BSc Thesis Applied Mathematics

Balancing balanceable matrices

Jorn de Jong

Supervisor: Matthias Walter

January, 2022

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

I would like to thank Dr. Matthias Walter for sharing his enthusiasm and for providing guidance and feedback.

Balancing balanceable matrices

Jorn de Jong*

January, 2022

Abstract

Totally unimodular matrices play an important role in combinatorial optimization. Testing if a matrix is totally unimodular requires checking if a matrix can become balanced by multiplying a subset of the entries with -1. A matrix with all entries in $\{-1, 0, 1\}$ is balanced if in every submatrix with 2 nonzero entries per row and per column, the sum of the entries is a multiple of 4. This paper describes an improved algorithm to create a balanced signing and a linear time algorithm to create a balanced signing for graphic matrices.

1 Introduction

A matrix with all entries in $\{-1, 0, 1\}$ is balanced if and only if the sum of the entries in each submatrix with exactly 2 nonzero entries per row and per column is a multiple of 4 [2]. A balanceable matrix is a matrix that can become balanced by multiplying a subset of entries with -1, such a signing is called a balanced signing. The quickest algorithm known to create a balanced matrix from a balanceable matrix is a polynomial time algorithm [5].

In Section 3 an improved algorithm to create a balanced signing for the balanceable matrix $[A \mid y]$ is presented, where y is a column vector and the matrix A is already balanced signed. Truempers algorithm [5] to sign an entire balanceable matrix, as well as an improved version of this algorithm, are presented in Section 4. The improved version can also be used to find a maximal balanceable submatrix. The paper by Truemper was missing a rigorous correctness proof, this is also stated in Section 4, together with a correctness proof for the improved version. In Section 5 graphic matrices, which are a subset of balanceable matrices, will be introduced as well as a linear time algorithm to create a balanced signing for these kinds of matrices.

Creating a balanced signing for a, not necessarily balanceable, matrix plays an important part in checking if a matrix is totally unimodular (TU). A matrix is TU if each square submatrix has a determinant of -1, 0 or 1. TU matrices are important in the field of combinatorial optimization [4]. When $A \in \mathbb{R}^{m \times n}$ is TU and $b \in \mathbb{Z}^n$, $\{\max x \mid Ax \leq b\}$ has integer solutions.

*Email: j.dejong-9@student.utwente.nl

2 Preliminaries & assumptions

We will use $A_{X,Y}$ for the submatrix of $A \in \{-1,0,1\}^{M\times N}$, a matrix of size $|M|$ by $|N|$, with row set $X \subseteq M$ and column set $Y \subseteq N$. If X or Y is a singleton set, for example only containing the element x, we use x instead of $\{x\}$. We will always use uppercase letters for sets and lowercase letters for elements of sets. We will use a colon to represent all rows or columns. For example, $A_{X,:}$ means the submatrix of A containing all columns and only the rows X. $A_{:,y}$ indicates column y of A. $z(A,X,Y)$ will be used as notation for a matrix induced by the rows X and columns Y of A, only those entries are kept, all other entries are put to zero. $t(A,F)$ will be used as notation for a matrix induced by a set $F \subseteq \{(i,j) : A_{i,j} \neq 0\}$, only those entries are kept, all other entries are put to zero. $supp(A)$ is a set containing the nonzero entries of A.

Let $G = (U,V,E)$ be a finite undirected simple bipartite graph with vertex set the union of U and V and edge set E such that each edge connects a vertex in U to one in V. A graph is connected if and only if any two vertices are connected by a path in the graph. A chordless cycle is a cycle such that no two vertices of the cycle are connected by an edge that does not belong to the cycle itself.

The bipartite graph of a matrix $A \in \{-1,0,1\}^{m\times n}$, $BG(A)$, has weighted adjacency matrix

$$\begin{array}{cc} & \begin{array}{cc} U & V \end{array} \\ \begin{array}{c} U \\ V \end{array} & \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \end{array}.$$

Let k be the number of nonzero entries in A, note that this is the same as the number of edges in $BG(A)$. Matrix A is connected if and only if $BG(A)$ is connected. We will use the rows and columns of A as notation for the vertices of $BG(A)$. A matrix A has the same number of components as $BG(A)$. We will use $d(u,v)$ to represent the distance between two vertices u and v.

Definition 2.1 (Chordless cycle submatrix)**.** A submatrix of a matrix A is a *chordless cycle submatrix* if it has 2 nonzero entries per row and per column. A chordless cycle submatrix is *covered by* a set $F \subseteq \{(i,j) : A_{i,j} \neq 0\}$ if it is a submatrix of $t(A,F)$.

The bipartite graph of a chordless cycle submatrix would represent a chordless cycle, hence the naming.

Signing a matrix means multiplying a subset of entries with -1. A matrix $A \in \{-1,0,1\}^{m\times n}$ is *balanced* if in every chordless cycle submatrix the sum of the entries is a multiple of 4. A matrix is *balanceable* if we can sign it such that it is balanced, we will call such a signing a *balanced signing.*

The goal of this thesis is to improve the signing algorithm for balanceable matrices. Whenever we talk about a matrix, A, we will therefore assume $A \in \{-1,0,1\}^{m\times n}$ and A is balanceable. We will also assume that each row and each column of A has at least one nonzero entry for simplicity. Rows or columns with only zero entries do not affect the signing, since they don't show up in any chordless cycle submatrix.

Lemma 1 (Balanced signing)**.** A balanceable matrix $A \in \{-1, 0, 1\}^{m \times n}$ with k nonzeros and c components is balanced signed if and only if there exists a subset F of $(m_{nz}+n_{nz}-c)$ entries of A, where m_{nz} and n_{nz} are the number of rows and columns respectively with at least one nonzero entry, and an ordering for the entries $e_1, \ldots, e_{k-|F|} \in supp(A) \setminus F$ for which the following holds: For each entry $e_i \in supp(A) \setminus F$ there exists a chordless cycle submatrix of A covered by $F \cup \{e_1, \ldots, e_i\}$ containing e_i such that the sum of all entries is a multiple of 4.

Proof. $\Leftarrow$ Assume that $A \in \{-1, 0, 1\}^{m \times n}$ is balanceable, there is a subset F of $(m_{nz} + n_{nz} - c)$ entries of A and an ordering for the entries $e_1, \ldots, e_{k-|F|} \in supp(A) \setminus F$ such that for each entry $e_i \in supp(A) \setminus F$ there exists a chordless cycle submatrix of A covered by $F \cup \{e_1, \ldots, e_i\}$ containing e_i such that the sum of all entries is a multiple of 4. We will prove by induction that $t(A, supp(A)) = A$ is balanced signed.

Base case: $t(A, F)$ is balanced signed since all entries of $t(A, F)$ are in F.

Inductive step: Let $i < k - |F|$ be given and assume $t(A, F \cup \{e_1, \ldots, e_i\})$ is balanced signed. We want to show that $t(A, F \cup \{e_1, \ldots, e_i, e_{i+1}\})$ is balanced signed as well. Since $t(A, F \cup \{e_1, \ldots, e_i\})$ is balanced signed, we only need to check that for each chordless cycle submatrix containing e_{i+1} induced by $F \cup \{e_1, \ldots, e_i, e_{i+1}\}$ the sum of the entries is a multiple of 4.

Since $t(A, F \cup \{e_1, \ldots, e_i, e_{i+1}\})$ is balanceable, there exists a signing for e_{i+1} such that there exists a chordless cycle submatrix containing e_{i+1} covered by $F \cup \{e_1, \ldots, e_i, e_{i+1}\}$ such that the sum of the entries is a multiple of 4. Since a chordless cycle submatrix has an even number of nonzero entries, the sum is congruent 0 or 2 mod 4. Changing the sign of e_{i+1} flips this sum between 0 and 2 mod 4. This signing of e_{i+1} implies that for each chordless cycle submatrices containing e_{i+1} covered by $F \cup \{e_1, \ldots, e_i, e_{i+1}\}$ the sum of the entries is a multiple of 4, since $t(A, F \cup \{e_1, \ldots, e_i, e_{i+1}\})$ is balanceable. Hence for this signing of e_{i+1}, $t(A, F \cup \{e_1, \ldots, e_i, e_{i+1}\})$ is balanced signed.

By induction $t(A, supp(A)) = A$ is balanced signed.

$\Rightarrow$ Assume $A \in \{-1, 0, 1\}^{m \times n}$ is balanceable and balanced signed. Let F be the entries corresponding to the edges of a spanning forest of $BG(A)$. Since $BG(A)$ has $(m_{nz} + n_{nz})$ vertices and c components, this spanning forest has $(m_{nz} + n_{nz} - c)$ edges. Hence F has $(m_{nz} + n_{nz} - c)$ entries.

Since A is balanced signed, the sum of the entries in each chordless cycle is a multiple of 4. We only still need an entry ordering such that for each entry $e_i \in supp(A) \setminus F$ there exists a chordless cycle submatrix of A covered by $F \cup \{e_1, \ldots e_i\}$ containing e_i. We can order the entries $e_1, \ldots, e_{k-|F|} \in supp(A) \setminus F$ such that for each e_i the number of chords in $BG(A)$ for the unique cycle in $BG(t(A, F \cup \{e_i\}))$ is smaller or equal to the number of chords in $BG(A)$ for the unique cycle in $BG(t(A, F \cup \{e_j\}))$ for all $j > i$.

For each entry $e_i \in supp(A) \setminus F$, we can create a chordless cycle. First, we take unique cycle C in $BG(t(A, F \cup \{e_i\}))$. Since the entries of F correspond to the edges of a spanning forest of $BG(A)$, such a unique cycle exists. As long as there are chords in C, we can replace C by a smaller cycle containing e_i and a chord. Eventually, this cycle will be chordless and will only use entries from F and e_j for $j \leq i$ because of the chosen entry ordering. $\square$

3 Signing a single column

If we have a balanced signing of a matrix A, and we append a new column y to this matrix such that the resulting matrix is balanceable, we want to be able to sign this column such that the resulting matrix is still balanced. This can be done by running a slightly changed version of breadth first search, BFS. The idea is to run BFS on $BG(A)$ starting from a nonzero row of y, and sign all nonzero entries $r_j \in y$ such that the sum of the entries in the r_i-r_j-path together with these two entries in y is a multiple of 4. Here r_i is a nonzero entry of y already visited during BFS such that $d(r_i, r_j) \leq d(r_h, r_j)$ for all nonzero $r_h \in y$ already visited during BFS. The entries used for this signing represent a chordless cycle.

When the modified BFS has signed all entries it can reach, but not all nonzero entries in y are signed, we simply run the algorithm again from a not yet signed nonzero entry until all entries in y are signed. The full algorithm is stated in Algorithm 1. When we have signed all nonzero entries in the column, we can stop, this is done at line 17.

Each entry $e \in A$ corresponds to an edge in $BG(A)$ connecting a row vertex with a column vertex. $r(e)$ is used to denote the row of e in A. $N_G(v)$ is the set containing all neighbors of vertex v in graph G.

Algorithm 1 *SignColumn*

Input: A balanceable matrix $A \in \{-1, 0, 1\}^{M \times N}$, a set of columns Y such that $A_{:,Y}$ is balanced signed and a column $y \in N \setminus Y$ that we want to sign.

Output: A balanced singing for $A_{:,Y \cup y}$.

Procedure:

1: $B := z(A, :, Y)$ ▷ B is a matrix containing already signed entries.
2: $P := \{x : |A_{x,y}| = 1\}$ ▷ P is a set containing all entries we need to sign.
3: **while** $P \neq \emptyset$ **do**
4:     Pick any $e_s \in P$
5:     $x_s := r(e_s)$, $E := \{e_s\}$ ▷ E is a set containing all explored vertices.
6:     $P := P \setminus \{x_s\}$
7:     Initialize a queue Q with (x_s, e_s).
8:     **while** $Q \neq \emptyset$ **do**
9:         Remove (v, d) from Q.
10:         **for all** $w \in N_{BG(B)}(v)$ **do**
11:             **if** $w \notin \textit{explored}$ **then**
12:                 $d = d + \begin{cases} A_{w,v} & \text{if } w \text{ is a row vertex} \\ A_{v,w} & \text{if } w \text{ is a column vertex} \end{cases}$
13:                 $E := E \cup \{w\}$
14:                 **if** $w \in P$ **then** ▷ Check if we need to sign w.
15:                     $P := P \setminus \{w\}$
16:                     $A_{w,y} = \begin{cases} -1 & \text{if } d = 1 \bmod 4 \\ 1 & \text{if } d = 3 \bmod 4 \end{cases}$
17:                     **if** $P = \emptyset$ **then return** A ▷ Stop if we have signed everything.
18:                     $d := A_{w,y}$
19:                 Append (w, d) to the queue.
20: **return** A

Making sure that the sum of the entries in a chordless cycle submatrix is a multiple of 4 can be done by keeping track of the sum of the entries since the last nonzero row of y was visited. If this is 1, sign the entry with -1 such that the sum is a multiple of 4. If this sum is 3, sign the entry with 1 such that the sum is a multiple of 4.

Lemma 2. Algorithm 1 produces a balanced singing for $A_{:,Y\cup y}$ for a balanceable matrix $A \in \{-1,0,1\}^{M\times N}$ with columns $Y \subset N$ already signed, $y \in N \setminus Y$ and k nonzero entries. Furthermore, it has time complexity of $O(|N| + |M| + k)$.

Proof. $A_{:,Y}$ is a balanced signing so there exists an entry ordering and a set F such that the conditions of Lemma 1 are satisfied. We have to extend this entry ordering and set F such that it also holds for $A_{:,Y\cup y}$.

All entries in the column y, for which the entire row in $A_{:,Y}$ is filled with zeros, are not changed. This is because there are no neighbors found at line 10. All these entries are added to F and the sign is not changed. The number of added entries to F is the same as the number of new rows with at least one nonzero entry, hence the cardinality of F is still within limits.

For each component of $BG(A_{:,Y})$ for which BFS is executed, x_s is added to F. The signing of this entry is not changed. All entries $e \in P \setminus F$ in the same component as x_s in $BG(A_{:,Y})$ will be signed such that for each e the sum of all the entries in a chordless cycle submatrix containing e is a multiple of 4. This only uses the entries in the other columns, and the entries signed before in column y. Hence a valid ordering for the entries is the ordering used before with the newly signed entries appended. Whenever we sign a new entry, we append it to the end of the entry ordering. By choosing the entries corresponding to the edges used in each chordless cycle, a chordless cycle submatrix can be constructed only using the allowed entries such that the sum is a multiple of 4.

For the first component, there is a new nonzero column, hence we can add x_s to F. For each component after that, 2 components become connected, hence the number of components decreases by 1 and we can add x_s to F.

Since we were able to extend both the entry ordering and F from Lemma 1, by induction, Algorithm 1 produces a balanced signing for $A_{:,Y\cup y}$.

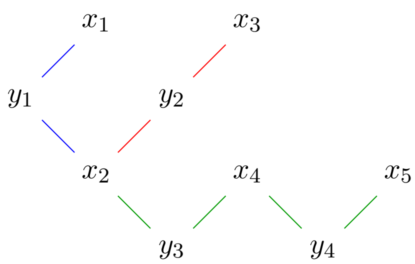
BFS will run once for each different component of $A_{:,Y}$. The time complexity of BFS of a single component is $O(m_c + n_c + k_c)$, where m_c, n_c and k_c are the number of nonzero columns, nonzero rows and nonzero entries in a component respectively. Since each vertex and each edge is part of exactly one component, in total this has a time complexity of $\sum_c O(m_c + n_c + k_c) = O(|M| + |N| + k)$. □

3.1 Example

In this section, an example that illustrates the working of Algorithm 1 is presented. Let A be as in Figure 1a. Figure 1b shows the corresponding bipartite graph, where all the edges have a weight of 1. We will now sign column y_5 using Algorithm 1.

$$\begin{array}{c c}
 & \begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & y_5 \end{array} \\
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1{*}{*} \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array},$$

(A) A balanceable matrix A, with the first four columns already signed, $Y = \{y_1, y_2, y_3, y_4\}$



(B) The corresponding bipartite graph of the $A_{:,Y}$, $BG(A_{:,Y})$

FIGURE 1

We pick e_s to be x_1 and add this to the queue together with the value $A_{1,5}$, which is 1. The only neighbor is y_1, which we do not need to sign, so we add $(y_1, 2)$ to the queue. The only unexplored neighbor of y_1 is x_2, which we do need to sign. We increase d by $A_{2,1}$, d now becomes 3. Since d is 3, we sign $A_{2,5}$ with $+1$ and we add $(x_2, 1)$ to the queue. The edges used are colored blue in Figure 1b as well as the corresponding entries in Figure 1a.

The unexplored neighbors of x_2 are y_2 and y_3, which we both do not need to sign. We add $(y_2, 2)$ and $(y_3, 2)$ to the queue. The only unexplored neighbor of y_2 is x_3, which we do need to sign. We increase d by $A_{3,2}$, d now becomes 3. Since d is 3, we sign $A_{3,5}$ with $+1$ and we add $(x_3, 1)$ to the queue. The edges used are colored red in Figure 1b as well as the corresponding entries in Figure 1a. Entries already used before that are used again are marked with a red asterisk.

The only unexplored neighbor of y_3 is x_4, which we do not need to sign, so we add $(x_4, 3)$ to the queue. x_3 has no unexplored neighbors. The unexplored neighbor of x_4 is y_4, which we do not need to sign, we add $(y_4, 0)$ to the queue. The only unexplored neighbor of y_4 is x_5, which we do need to sign. We increase d by $A_{5,3}$, d now becomes 1. Since d is 1, we sign $A_{3,5}$ with -1. The edges used are colored green in Figure 1b as well as the corresponding entries in Figure 1a. Entries already used before that are used again are marked with a green asterisk. Since we have signed all entries we need to sign, the algorithm stops.

We end up with the following balanced signing of A

$$\begin{array}{c c}
 & \begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & y_5 \end{array} \\
\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}
\end{array}.$$

4 Signing a balanceable matrix

Truemper presented an algorithm in [5] that creates a balanced signing for a balanceable matrix by signing the matrix column by column using Algorithm 1. This algorithm is stated in Algorithm 2. The columns need to be processed in a specific order, such that $BG(A_{X,Y})$ is always connected, when we have already signed columns Y with nonzero rows X, this is called *sequentially connected*. The algorithm is stated in Algorithm 2. Since the matrix must be sequentially connected, all entries of a single column are reached during a single BFS in Algorithm 1 and there will only be one iteration of the while loop at line 3. In fact, the while loop was not present in Truempers algorithm.

An altered version of the algorithm is stated in Algorithm 3. This version can create a balanced signing for any balanceable matrix without the requirement that it is sequentially connectable. The main difference with Truempers algorithm is that there can be multiple iterations of the while loop in Algorithm 1 at line 3.

Algorithm 2 *BalancedSigningTruemper* [5]

Input: A sequentially connectable balanceable matrix $A \in \{-1, 0, 1\}^{m \times n}$.
Output: A balanced signing for matrix A.
Procedure:

```
Compute a sequentially connected ordering of columns L.
Let Y be a set containing the first element ℓ of L.
for all y ∈ L \ y_0 in the order imposed by L do
    A := SignColumn(A, Y, y)
    Y := Y ∪ y
return A
```

Algorithm 3 *BalancedSigning*

Input: A balanceable matrix $A \in \{-1, 0, 1\}^{M \times N}$.
Output: A balanced signing for matrix A.
Procedure:

```
1: Y := {y_0 : for some y_0 ∈ N}
2: for all y ∈ N \ y_0 do
3:     A := SignColumn(A, Y, y)
4:     Y := Y ∪ y
5: return A
```

Theorem 3. Algorithm 3 produces a balanced signing for a balanceable matrix $A \in \{-1, 0, 1\}^{m \times n}$ and has a time complexity of $O(n(n + m + k))$.

Proof. **Base case:** Before the for all loop at line 2, $A_{:,Y}$ is a balanced signing.

All entries of $A_{:,Y}$ are in F. The number of components of $A_{:,Y}$ is 1 and the cardinality of F is exactly $m_{nz} = m_{nz} + 1 - 1$ as requested by Lemma 1 and hence $A_{:,Y}$ is a balanced signing.

Inductive step: Let $\ell < n$ be given and suppose Algorithm 3 gives a balanced signing after the first ℓ iterations of the for all loop at line 2. We have to show that after the next iteration of the for loop, $A_{:,Y \cup y}$ is still balanced signed.

We sign column y by Algorithm 1. Algorithm 1 creates a balanced signing for $A_{:,Y \cup y}$ hence we are done and by induction the algorithm produces a balanced signing for a balanceable matrix $A \in \{-1, 0, 1\}^{m \times n}$.

The for loop at line 2 will be executed $n - 1$ times, exactly once for each column except the first one. In each iteration of the for loop, Algorithm 1 is called once. Since this algorithm has a time complexity of $O(n + m + k)$, the time complexity of Algorithm 3 is $O(n(n + m + k))$. □

A rigorous correctness proof for Algorithm 2 was missing in [5]. Due to the way we presented Algorithms 1, 2 and 3 the proof presented above for Algorithm 3 also applies to Algorithm 2.

Theorem 4. (Truemper [5]) Algorithm 2 produces a balanced signing for a matrix $A \in \{-1, 0, 1\}^{m \times n}$ and has a time complexity of $O(n(n + m + k))$.

Instead of signing A, we can also sign A^T and take the transpose of the result. This gives a time complexity of $O(m(m + n + k))$. Hence by signing A^T whenever $m < n$ we get a time complexity of $O(min(m, n) \cdot (m + n + k))$.

5 Linear time algorithm for graphic matrices

Let G be a directed graph with vertex set V and edge set E_D. Let T be a directed tree on V with edge set E_T. Let $M \in \{-1, 0, 1\}^{E_T \times E_G}$ be a matrix such that for all $d = (v, w) \in E_G$ and $e \in E_T$, $M_{e,t}$ is equal to $+1$ if the unique v-w-path in T passes through e forwardly, -1 if the unique v-w-path in T passes through e backwardly and 0 if the unique v-w-path in T does not pass through e. M is called a *network matrix* and is balanced [4]. A *graphic matrix* is a matrix that can become a network matrix by signing the matrix.

Algorithm 4 BalancedSigningGraphic

Input: A graphic matrix $A \in \{0, 1\}^{T_v \times N}$, together with the tree T, where T_v is a set containing the vertices of the tree T.

Output: A balanced signing of A.

Procedure:

Run BFS on T from any vertex such that we have the depth and the predecessor of each edge.
for all $y \in N$ **do**
  $X := \{x : A_{x,y} = 1\}$
  $t := argmax_{x \in X}(x.depth)$
  $u := min_{x \in X}(x.depth)$
  while $t.depth \neq u$ **do**
    $A_{t,y} := -1$
    $t := t.predecessor$
  $A_{t,y} := -1$
return A

We have developed an algorithm to sign a matrix in linear time, if the matrix is graphic and we have access to the tree. The idea is to orient all edges such that the tree is rooted into a single vertex. After that, we run BFS once on the tree and save the distance from the root. To sign a column connecting vertices s and t, we can look at the distance of both s and t and go to the predecessor of whichever one is bigger. We continue like this until we arrive at the same vertex u and we have found the s-t-path in the tree by combining the s-u and t-u-path. All edges appear forward in the s-u and t-u-path. By combining them all entries in the s-u-path will appear forward and all entries in t-u-path will appear backward. Whenever we encounter an entry on the s-u-path, we don't change the sign and keep it $+1$, whenever we encounter an entry on the t-u-path, we sign it with -1.

Lemma 5. Algorithm 4 produces a balanced signing for a graphic matrix $A \in \{0,1\}^{T_v \times N}$ and has a time complexity of $O(|N| + k)$.

Proof. For each column, the s_v-t_v-path P is the path connecting the vertex t_v with the highest depth to s_v, via the vertex with the lowest depth u_v. In P, all entries on the t_v-u_v-path appear backwards, and all entries in s_v-u_v-path appear forward, hence signing all entries on the t_v-u_v-path with -1 gives a network matrix and hence a balanced signing. All entries on the t_v-u_v-path are exactly the entries we find in the while loop on line 6, together with the entry we find on line 9.

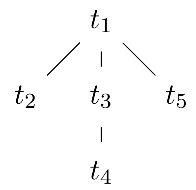
BFS has a time complexity of $O(v_t + e_t)$, where v_t and e_t are the number of vertices and edges in the tree, respectively. $v_t = |N|$ and $e_t = |N| - 1$ so this has a time complexity of $O(|N|)$. For each column y with k_y nonzero entries, finding the min and $argmax$ has time complexity $O(k_y)$. The while loop at line 6 will be iterated $O(k_y)$ times, each taking constant time. In total the for all loop at line 2 has a time complexity of $\sum_y O(k_y) = O(k)$. This means that in total the time complexity of Algorithm 4 is $O(|N| + k)$. $\square$

5.1 Example

In this section, an example that illustrates the working of Algorithm 4 is presented. Let A be as in Figure 2a. Figure 2b shows the corresponding tree. We will now sign column y_1 using Algorithm 4.

$$\begin{array}{c c} & \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{array}{c} t_1\text{-}t_2 \\ t_1\text{-}t_3 \\ t_3\text{-}t_4 \\ t_1\text{-}t_5 \end{array} & \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{array}$$

(A) A graphic matrix A



(B) The corresponding bipartite graph, $BG(A)$

FIGURE 2

Note that nonzero entries of each column of the matrix represent a path in the tree. We first run BFS from t_1 to get the distance and the predecessor for every edge in the tree. For column y_1, the minimum depth u is 0. The edge t with maximum depth is t_3-t_4 and has a depth of 1. Since this is not equal to 0, we sign $A_{3,1}$ with a -1. The predecessor of t_3-t_4 is t_1-t_3, which has a depth of 0, this is equal to u, so we sign $A_{2,1}$ with a -1 at line 9 and we are done with this column. The signing of the next column works similarly and results in $A_{2,2}$ and $A_{3,2}$ to be signed with -1. We end up with the following balanced signing of A

$$\begin{bmatrix} 1 & 0 \\ -1 & -1 \\ -1 & -1 \\ 0 & 1 \end{bmatrix}.$$

Note that the sum of the entries in the only chordless cycle submatrix is indeed a multiple of 4.

5.2 Nongraphic matrices

We can use a slightly altered version of Algorithm 4 in combination with Algorithm 1 to improve the running time to balance sign nongraphic matrices. This algorithm does not improve the order of the running time but may still be more efficient in practice. Determining the maximum column index set Y such that $A_{:,Y}$ is graphic together with the tree T can be done in almost linear time [1, 3]. We can sign all these columns in linear time with Algorithm 4. After this, we can sign all other columns with Algorithm 1. The algorithm is stated in Algorithm 5.

Algorithm 5 *BalancedSigningNonGraphic*

Input: A balanceable matrix $A \in \{0,1\}^{T_v \times N}$.
Output: A balanced signing for matrix A.
Procedure:

1: Determine the maximum column index set Y_g such that $A_{:,Y_g}$ is graphic together with the tree T.
2: Run BFS on T from any vertex such that we have the depth and the predecessor of each edge.
3: **for all** $y \in Y_g$ **do**
4:     $X := \{x : A_{x,y} = 1\}$
5:     $t := argmax_{x \in X}(x.depth)$
6:     $u := min_{x \in X}(x.depth)$
7:     **while** $t.depth \neq u$ **do**
8:         $A_{t,y} := -1$
9:         $t := t.predecessor$
10:     $A_{t,y} := -1$
11: $Y := Y_g$
12: **for all** $y \in N \setminus Y_g$ **do**
13:     $A := SignColumn(A, Y, y)$
14:     $Y := Y \cup y$
15: **return** A

Lines 2 till 10 create a balanced signing for $A_{:,Y_g}$. Lines 12 till 14 balance sign all other columns using Algorithm 1.

6 Summary

In Section 4 an improved algorithm to create a balanced signing for balanceable matrices is presented, together with a rigorous correctness proof. The algorithm by Truemper, on which this algorithm is inspired, is also proven. In Section 5, a linear time algorithm to create a balanced signing for graphic matrices is presented and proven.

References

[1] Robert E. Bixby and Donald K. Wagner. An almost linear-time algorithm for graph realization. *Mathematics of Operations Research*, 13(1):99–123, 1988.

[2] Paul Camion. Characterization of totally unimodular matrices. 1965.

[3] Satoru Fujishige. An efficient pq-graph algorithm for solving the graph-realization problem. *Journal of Computer and System Sciences*, 21(1):63–86, 1980.

[4] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume B. 01 2003.

[5] Klaus Truemper. A decomposition theory for matroids. v. testing of matrix total unimodularity. *Journal Of Combinatorial Theory*, pages 241–281, December 1987.