Discrete Comput Geom 14:331–358 (1995)

Discrete & Computational
Geometry
© 1995 Springer-Verlag New York Inc.

Visibility Graphs of Staircase Polygons and the Weak Bruhat Order, I: from Visibility Graphs to Maximal Chains*

J. Abello,[1] O. Egecioglu,[2] and K. Kumar[3]

[1] Department of Computer Science, Texas A & M University,
College Station, TX 77843-3112, USA
abello@cs.tamu.edu

[2] Department of Computer Science, University of California,
Santa Barbara, CA 93106, USA
omer@cs.ucsb.edu

[3] Department of Mathematics and Computer Science, Colby College,
Waterville, ME 04901, USA
k_kumar@colby.edu

Abstract. The recognition problem for visibility graphs of simple polygons is not known to be in NP, nor is it known to be NP-hard. It is, however, known to be in *PSPACE*. Further, every such visibility graph can be dismantled as a sequence of visibility graphs of convex fans.

Any nondegenerate configuration of n points can be associated with a **maximal chain** in the weak Bruhat order of the symmetric group S_n. The visibility graph of *any* simple polygon defined on this configuration is completely determined by this maximal chain via a one-to-one correspondence between maximal chains and **balanced tableaux** of a certain shape.

In the case of staircase polygons (special convex fans), we define a class of graphs called **persistent graphs** and show that the visibility graph of a staircase polygon is persistent. We then describe a polynomial-time algorithm that recovers a representative maximal chain in the weak Bruhat order from a given persistent graph, thus characterizing the class of persistent graphs.

The question of recovering a staircase polygon from a given persistent graph, via a maximal chain, is studied in the companion paper [4]. The overall goal of both papers is to offer a characterization of visibility graphs of convex fans.

* The research of J. Abello was supported by NSF Grants Nos. DCR 8603722 and DCR 8896281. This research was done while K. Kumar was at the Department of Computer Science, Texas A & M University.

1. Introduction

A **polygon** P is a sequence $p_0, \ldots, p_{n-1}$ of n distinct points in the plane, together with the open line segments $p_i p_{i+1} \bmod n$ joining points in P. The polygon is called **simple** if no two of the segments intersect. Two vertices of the polygon are called internally visible if the closed line segment between them is either an edge of the polygon or lies entirely in the interior of the polygon. The internal visibility graph of a simple polygon is the graph whose vertices correspond to the vertices of the polygon and whose edges correspond to internally visible pairs of vertices in the polygon.

The problem of characterizing internal visibility graphs of arbitrary simple polygons, hereafter called visibility graphs, and the related algorithmic problem of efficiently recognizing such graphs have remained important open problems in computational geometry [13], [16], [21]. The recognition problem for visibility graphs of simple polygons: to decide whether a given graph is the visibility graph of some simple polygon, is known to be in *PSPACE* [14]. However, at the time of writing, the problem is not known to be NP, nor is it known to be NP-hard. From the characterization standpoint, Ghosh [16] obtained four necessary conditions for visibility graphs of simple polygons but it has been shown that they are not sufficient even for triconnected graphs [6]. Further necessary conditions were developed by Coullard and Lubiw [9] but they are not sufficient. Abello *et al.* [6] strengthened these results by showing they are not sufficient, even for triconnected graphs, and in the case of the conditions of [9], even for planar graphs. A stronger set of necessary conditions is offered in [7]. For every graph satisfying these conditions a uniform rank 3 oriented matroid is constructed. When this matroid is coordinatizable a simple polygon can be recovered whose visibility graph is isomorphic to the given graph. The only complete characterizations obtained to date have been for internal visibility graphs of spiral polygons [15]. Positive results have also been obtained by extending the notion of visibility to other geometric objects such as edges in orthogonal polygons [21] and collections of vertical line segments in the plane [23], [24].

In this paper and its companion, we study visibility graphs of staircase polygons [21] also called orthogonal convex fans. This class of graphs is a proper subclass of the class of persistent graphs which were introduced originally in [3]. The main contribution of this paper is an efficient algorithmic characterization of persistent graphs in terms of balanced tableaux of a certain shape. Our characterization offers a novel graph theoretical partition of the set of maximal chains in the weak Bruhat order of the symmetric group. This is a coarser partition than the one induced by semispace equivalence [18] which in turn is the same as the one derived from the closure operator defined in [2]. In the sequel paper [4] we study the geometric realization of persistent graphs as visibility graphs of convex fans.

1.1. *Overview*

For any polygon whose n vertices constitute a nondegenerate configuration of points, the global ordering on the pairwise slopes of the vertices may be represented combinatorially by **a balanced tableau** of shape $n-1, \ldots, 1$. This is done by relating the **circular sequences** of Goodman and Pollack [11], [17], [18] to **maximal chains** in the weak Bruhat order of the symmetric group [1], [2], [12]. A geometric interpretation of a one-to-one correspondence between these chains and balanced tableaux (due to Edelman and Greene [12]) gives the desired combinatorial representation (Section 3).

The visibility graph of *any* simple polygon defined on a nondegenerate configuration of points in the plane can be completely determined from the combinatorial representation of the underlying configuration (see [5] and [20] for this and other related results). For the staircase polygons considered here, the mapping that produces the visibility graph from the underlying balanced tableau has a particularly simple structure. The obtained graph satisfies a special condition called persistence (Sections 2 and 3). From it several key combinatorial properties follow, which are then used to exhibit an algorithm that generates any given persistent graph from the complete graph (Section 4). This algorithm is then used as a guide for a second algorithm that reconstructs, from a given persistent graph, a representative balanced tableau, in polynomial time (Section 5). Therefore, balanced tableaux, and their corresponding maximal chains in the weak Bruhat order, are partitioned into equivalence classes by persistent graphs. This is a combinatorial view of one of the main contributions of this paper. If each such equivalence class contains a realizable chain, then persistent graphs are precisely visibility graphs of convex fans. It is precisely this geometric interpretation what constitutes the main contribution of the sequel paper [4].

2. Definitions

A monotone, decreasing sequence of a finite number of alternating horizontal and vertical line segments that connects a point p on the positive y-axis to a point q on the positive x-axis is called a **staircase path.** A **staircase polygon (orthogonal convex fan)** is a staircase path together with the segments from the origin to p and the origin to q. A staircase path with $n-1$ vertical or horizontal line segments determines a staircase polygon with $2n$ vertices which we call a staircase polygon of order n. Suppose u_0 is the origin, $u_1, u_2, \ldots, u_{n-1}$ are the points from left to right where the staircase path changes from horizontal to vertical. Let $v_1 = p$, $v_n = q$ and suppose the points where the path changes from a vertical segment to a horizontal one are labeled $v_2, v_3, \ldots, v_{n-1}$.

The **vertex visibility graph** or simply the **visibility graph** $\Gamma_P = (V, E)$ associated with a staircase polygon P of order n is constructed as follows. Put $V = \{v_1, \ldots, v_n, u_0, u_1, \ldots, u_{n-1}\}$. For two distinct elements $u, v \in V$, $(u, v) \in E$ if and only if the line segment connecting the corresponding points in the plane is either a segment of P itself, or lies completely in the interior of the closed region determined by P. It is clear that each of the vertices u_i, $1 \le i \le n-1$, in Γ_P is of degree 3 and that u_i is adjacent to precisely v_i, v_{i+1}, and u_0, for $1 \le i \le n-1$. Also u_0 is adjacent to all the remaining vertices. Thus the visibility graph is actually determined by the subgraph induced by the vertices $v_1, \ldots, v_n$. This induced subgraph is called the **core** of Γ_P and is denoted by $\mathscr{C}_P$. For a staircase polygon P, or order n, the core $\mathscr{C}_P$ is a connected graph on n vertices, and since $(v_1, v_2), \dots, (v_{n-1}, v_n)$ are all edges in Γ_P it has a Hamiltonian path. The problem of studying the combinatorial properties of Γ_P is reduced to studying those of $\mathscr{C}_P$.

Given a graph G, let $M(G)$ denote the adjacency of G. Since $M(G)$ is symmetric with zero diagonal, to simplify our exposition in what follows, we only consider the array consisting of the subdiagonal entries of the adjacency matrix and refer to it as the **matrix** of G. It is convenient to identify the graph G with its matrix and drop the subscript G when the context makes it clear which object is being referred to. In this setting the rows and columns of M are indexed as $2, \dots, n$ from top to bottom and $1, \dots, n-1$ from left to right, respectively. $M(i,j)$ refers to the entry on row i and column j of M with $j < i$. The entries $M(i, i-1)$ for $i = 2, 3, \dots, n$ are one.

2.1. *Weak Bruhat Order*

For $n \geq 2$, let S_n denote the symmetric group of all permutations of the set $\{1, \dots, n\}$. As a Coxeter group, S_n is endowed with a natural partial order called the weak Bruhat order [1], [2], [12]. The weak Bruhat order is generated by the following immediate successor relation. A permutation σ is an immediate successor of a permutation τ if and only if τ can be obtained from σ by interchanging a pair of noninverted elements of σ. For example, $\sigma = 2413$ has two immediate successors, 4213 and 2431. The partial order $\leq_{WB}$ is the transitive closure of this relation. $(S_n, \leq_{WB})$ is a ranked poset where the rank of a permutation is simply its inversion number $i(\sigma) = |\{(\sigma_i, \sigma_j)\colon i < j \text{ and } \sigma_i > \sigma_j\}|$.

Now, let s_i denote the adjacent transposition of the letters in **positions** i and $i+1$ of a permutation in S_n. Given a permutation $\sigma \in S_n$, σs_i is the permutation obtained by switching the symbols σ_i, σ_{i+1} of σ. Every permutation $\sigma \in S_n$ is representable as a word over the alphabet $\{s_1, \dots, s_n\}$ where the juxtaposition of letters serves to express σ as a product of the s_i's, the multiplications being performed from left to right. Among these representations, the words that involve exactly $i(\sigma)$ transpositions are called **reduced words** for σ. Reduced words that represent the maximal element $n \; n-1 \; \cdots \; 2 \; 1$ have length $N = \binom{n}{2}$ and determine **maximal chains** in $(S_n, \leq_{WB})$ in the interval from the identity permutation to its reverse.

2.2. *Balanced Tableaux*

Let $\lambda = (\lambda_1 \geq \lambda_2 \geq \cdots \lambda_k > 0)$ be a partition of N. The **Ferrers' diagram** of λ is a figure obtained from k left justified columns of squares of lengths $\lambda_1, \dots, \lambda_k$. The special partitions of the form $(n-1, n-2, \dots, 1)$ give rise to special diagrams of a staircase shape that we refer to as $SS(n)$. A **tableau** of shape λ is a filling of the cells of the Ferrers' diagram of λ with the distinct integers from the set $\{1, 2, \dots, N\}$. Consider a tableau T of staircase shape $SS(n)$ filled with distinct integers from

$1, \ldots, \binom{n}{2}$. A pair of cells $T(i, j)$ and $T(j, k)$, $k < j < i$, are called **mates** with respect to the cell $T(i, k)$. A tableau T is said to be **balanced** if for any three entries $T(i, j)$, $T(j, k)$, $T(i, k)$ we have either $T(i, j) < T(i, k) < T(j, k)$ or $T(i, j) > T(i, k) > T(j, k)$. In particular, mate cells in a balanced tableau T cannot have consecutive values. The tableau in Fig. 1 is balanced.

Balanced tableaux of shape $SS(n)$ may be thought of as an alternative encoding for maximal chains in the weak Bruhat order $(S_n, \leq_{WB})$. The balanced tableau T corresponding to a maximal chain can be constructed by setting $T(i, j) = l$ if in going from the $(l-1)$th permutation in the chain to the lth permutation the symbols that are interchanged are i and j, $j < i$. Edelman and Greene [12] proved that this mapping defines a one-to-one correspondence between balanced tableaux of shape $SS(n)$ and maximal chains in $(S_n, \leq_{WB})$.

2.3. *Skeletons of Balance Tableaux*

Given a balanced tableau T, the **hook** belonging to a cell $T(i, j)$ is the collection of all cells which lie above it in the same column and lie weakly to its right in its row. Given a balanced tableau T of shape $SS(n)$ we can construct a binary array M of shape $SS(n)$ by putting $M(i, j) = 1$ if and only if $T(i, j)$ is larger than all the elements that lie above it in the same column in its hook. Note that by the balance property this means that $T(i, j)$ is smaller than all the elements that lie to its right in its hook. The resulting array is called the **skeleton** of the balanced tableau. Figure 1 shows an example of a balanced tableau and its skeleton. Skeletons of balanced tableaux when considered as graphs turn out to belong to a special class of graphs, introduced in [3], called persistent graphs.

2.4. *Persistent Graphs*

Let G be a graph and let M be its adjacency matrix. A maximal length sequence of consecutive zeros in a row of M is called a **maximal horizontal interval of zeros**. Similarly, a maximal length sequence of consecutive zeros in a column of M is called a **maximal vertical interval of zeros**. A matrix M with $M(i, i-1) = 1$ is defined to

1							
1	2						
1	1	3					
1	1	1	4				
1	0	0	1	5			
1	0	0	1	1	6		
1	0	0	0	0	1	7	
1	0	0	1	0	1	1	8

1							
1	2						
2	3	3					
4	27	28	4				
5	7	8	6	5			
9	17	18	16	26	6		
10	13	14	12	15	11	7	
19	21	22	20	24	23	25	8

Fig. 1. The skeleton of a balanced tableau.

be **persistent** if, for every triple $k < j < i$ such that $M(i, j) = 1$ and $M(i, y) = 0$ for every y in the range $k < y < j$, the following conditions hold:

- $M(x, y) = 0$ for $j < x < i$, $k < y < j$.
- $M(j, k) = 1$ if $M(i, k) = 1$.

The matrix in Fig. 1 is a persistent matrix. We say that G is a **persistent graph** if an ordering of the vertices exists such that the adjacency matrix of G is persistent with respect to the ordering.

Intuitively, in a persistent matrix, maximal horizontal intervals of zeros "propagate upward." It can be shown that the above definition also implies that maximal vertical intervals of zeros "propagate to the right" as summarized in the following proposition.

Proposition 2.1. *If M is a persistent matrix, then for every triple $k < j < i$ such that $M(j, k) = 1$ and $M(x, k) = 0$ for every x in the range $j < x < i$:*

- *$M(x, y) = 0$ for every y in the range $j < x < i$, $k < y < j$.*
- *$M(i, j) = 1$ if $M(i, k) = 1$.*

The propagation of maximal horizontal intervals of zeros is called **horizontal persistence** and the propagation of maximal vertical intervals of zeros is called **vertical persistence**. The two notions can be shown to be equivalent and we refer to them interchangeably as the **persistence conditions** when the context makes it clear which of the two is being used.

3. Visibility Graphs of Staircase Polygons are Persistent

In this section we show that visibility graphs of staircase polygons are persistent. We begin by showing that to any staircase polygon, we can associate a balanced tableau whose skeleton is identical to the core graph $\mathscr{C}_P$ of the polygon.

Let C be a nondegenerate configuration of n points in the plane. Number the points from 1 through n in increasing order of their x-coordinates. The points in P determine $\binom{n}{2}$ lines. We may construct a tableau T of shape $SS(n)$ that encodes the linear order on the slopes of these lines by putting $T(i, j) = s$ if and only if the rank of the slope of the line through i and j in this linear order is s. This is called the *slope table* of the configuration C.

Lemma 3.1. *Let C be a nondegenerate configuration of points in the plane. Then the slope table of C is a balanced tableau.*

Proof. Compute the circular sequence associated with the configuration using the method of Goodman and Pollack [17] with the initial direction of rotation being the horizontal. The first half-period of this sequence can be seen to be a maximal chain in the weak Bruhat order since the points are numbered according to their projections on the x-axis. For any two pairs of points i, j and k, l the transposition between symbols i and j occurs before that of the pair k and l if and only if the slope of the line determined by i and j is less than that of the line determined by k and l. Since the order of transpositions in the second half-period of the circular sequence is completely determined by the order of transpositions in the first half-period, the order in which the slopes are encountered is completely encoded in the corresponding maximal chain in the weak Bruhat order. If T is the balanced tableau obtained by applying the bijection of Edelman and Greene [12] to this maximal chain, then a straightforward geometric interpretation of the bijection shows that the value stored in $T(i, j)$ is s if and only if the rank of the slope of the line through i and j is s. Thus T is identical to the slope table of C. □

Now let P be a staircase polygon and consider the configuration of points $v_1, \dots, v_n$ that constitute the *core* of P. By the previous lemma, the slope tableau of this configuration is balanced and we call this the *balanced tableau of the staircase polygon P*.

Lemma 3.2. *If P is a staircase polygon and T is its balanced tableau, then the skeleton of T is identical to the adjacency matrix of $\mathscr{C}_P$.*

Proof. Consider a staircase polygon P. Let m_{ik} denote the *magnitude* of the slope between points v_i and v_k. By definition of visibility, v_k is visible from v_i if and only if the line segment joining the two points lies entirely inside P. It is easy to see that for staircase polygons this implies that v_i is visible from v_k if and only if there is no index j, $k < j < i$ such that $m_{ik} \leq m_{ij}$. Thus (v_i, v_k) is an edge of $\mathscr{C}_P$ if and only if $m_{ik} > m_{jk}$ for $j = i - 1, i - 2, \dots, k + 1$. The ordering of the slopes is completely encoded by the balanced tableau of the polygon and thus we have that v_i is visible from v_k iff $T(i, k)$ is larger than all the entries that lie above it in its hook. Thus the core of the visibility graph of a staircase polygon is identical to the skeleton of the balanced tableau of P. □

In the remainder of this section we show the first main result of this paper, that the skeleton of a balanced tableau is a persistent graph.

Lemma 3.3. *Let T be a balanced tableau and let M be its skeleton. For a triple $k < j < i$ with $M(i, j) = M(i, k) = 1$, if $M(i, y) = 0$ for $k < y < j$, then $T(i, j) < T(i, y)$ for $k < y < j$.*

Proof. By induction on y. The base case is $y = j - 1$. Since $M(i, j) = 1$, $T(i, j)$ is smaller than all the elements to its right on row i. So suppose $T(i, j - 1) < T(i, j)$. It follows that $T(i, j - 1)$ is smaller than all the elements which lie to its right on row i. Thus $M(i, j - 1)$ must be one, which is a contradiction.

Now, suppose that the statement is true for all $y \geq y'$, $k < y' < y$. Let $y = y' - 1$ and assume $T(i, y' - 1) < T(i, j)$. By the induction hypothesis, $T(i, y) < T(i, j)$ for $y' \leq y < j$. Since $T(i, j)$ is smaller than all the elements which lie to its right on row i we have that $T(i, y' - 1)$ is also smaller than all the elements which lie to its right on row i. Thus $M(i, y' - 1) = 1$ which is a contradiction. □

Lemma 3.4. *If M is the skeleton of a balanced tableau T, then M is a persistent matrix.*

Proof. Let i, j, k with $k < j < i$ be any triple with $M(i, j) = M(i, k) = 1$ and $M(i, y) = 0$ for $k < y < j$ as in the definition of the persistence condition. To prove that the first condition for persistence holds, suppose that $M(x, y) = 1$ for some x with $j < x < i$. Now, $T(i, j) < T(i, y)$ by Lemma 3.3 and $T(i, y) < T(j, y)$ by the balance property of the mates of $T(i, y)$. However, $T(j, y) < T(x, y)$ since $M(x, y) = 1$ by assumption, and $T(x, y) < T(x, j)$ by the balance property of the mates of $T(x, y)$. Finally, $T(x, j) < T(i, j)$ since $M(i, j) = 1$. Thus we conclude that $T(i, j) < T(i, j)$ which is a contradiction.

The proof for the second persistence condition is similar and is left to the reader. □

From Lemmas 3.4 and 3.2 and the definition of persistent graphs we have the following theorem.

Theorem 3.5. *If P is a staircase polygon, then the visibility graph of P is a persistent graph.*

4. Combinatorial Properties of Persistent Graphs

In this section we describe several interesting properties of persistent graphs. The results are described by the properties of the corresponding persistent matrices. The main result we show here is that these matrices can be partially ordered in a natural way that allows us to generate any of them in a canonical manner. The generation algorithm described here is then used in Section 5 to show the converse of Lemma 3.4, i.e., that every persistent matrix is the skeleton of a balanced tableau. In the following discussion we consider matrices which are implicitly assumed to be adjacency matrices of graphs. Thus we use some graph theoretic terminology in referring to these matrices.

Let $G = (V, E)$ be a graph and let M be its adjacency matrix. We associate **vertex** $v_j \in V$ with row j and column j of M which represent the adjacency information of this vertex. A **path** $(v_{i_0}, \dots, v_{i_k})$ is a sequence of entries in M such that $M(i_j, i_{j+1}) = 1, 0 \leq j \leq k$. A path is **ordered** if $i_j < i_{j+1}, 0 < j < k$. An ordered path $\Pi = (v_{i_0}, \dots, v_{i_k})$ is defined to be **concave** if the subgraph induced by $v_{i_0}, \dots, v_{i_k}$ in G is Π. The persistent matrix in which every entry is one is called the **clique matrix.**

4.1. Factors of a Persistent Matrix

Suppose $M[i, k]$ is a zero entry in a persistent matrix M. A vertex v_j, $k < j < i$, is defined to be a **blocking vertex** if and only if it lies on the *shortest* ordered path from v_k to v_i in G. It can be shown from the persistence conditions that such a path must be concave.

The following proposition follows directly from the definition of blocking vertices and the persistence conditions.

Proposition 4.1. *If M is a persistent matrix and if v_j is a blocking vertex for $M(i, k)$, then v_j is a blocking vertex for every $M(i, r)$, $k < r < j$, and every $M(s, k)$, $j < s < i$.*

As a consequence of Proposition 4.1 it can be seen that a vertex v_j is a blocking vertex for some entry $M(i, k)$ if and only if $M(j + 1, j - 1) = 0$. Let $v_{i_0}, \dots, v_{i_m}$ be the blocking vertices in a persistent matrix M and define matrices M_{i_k}, $0 \leq k \leq m$, called the **factors** of M as follows. $M_{i_k}(x, y) = 0$ if and only if $M(x, y) = 0$ and v_{i_k} is a blocking vertex for $M(x, y)$, $1 \leq x \leq n$, $2 \leq y \leq n - 1$. It can be readily verified that if M is a persistent matrix and M_j is its factor corresponding to vertex v_j, then:

1. The subgraphs induced by vertices $v_1, \dots, v_j$ and vertices $v_j, \dots, v_n$ are cliques.
2. In every row $i > j$ either $M_j(i, x) = 0$ for all $x < j$ or an entry $M_j(i, l)$, $l < j$, exists such that $M_j(i, x) = 1$ for $1 \leq x \leq l$ and $M_j(i, x) = 0$ for $l < x < j$.
3. If $M_j(i, l_1)$ and $M_j(i + 1, l_2)$, $i > j$ and $l_1, l_2 < j$, are the entries as described in item 2, then $l_1 \leq l_2$.
4. M_j is persistent.

The relationship between a matrix and its factors is summarized by the following lemma.

Lemma 4.2. *If M is a persistent matrix and M_{i_k}, $0 \leq k \leq m$, are its factors, then $M = M_{i_0} \cap M_{i_1} \cap \cdots \cap M_{i_m}$ where the binary intersection operation is interpreted as a bitwise AND on the corresponding positions of two factors.*

4.2. Reversible Entries

Given a persistent matrix M, an entry $M(i, k) = 1$, $k < i - 1$, is defined to be **reversible** if and only if the matrix obtained by changing the entry $M(i, k)$ to 0 is persistent. For example, the entry $M(8, 4)$ is reversible in the matrix in Fig. 1. A similar definition can be made for reversible zero entries.

A key combinatorial property of persistent matrices is that every persistent matrix with at least n one entries has at least one reversible one entry. Intuitively, this implies that given any persistent matrix it should be possible to change zero entries to one successively until the resulting matrix is the clique matrix, such that each intermediate matrix is persistent. We describe an algorithm that works in the reverse direction. Starting from a clique matrix K_n we present an algorithm that generates any given persistent matrix by changing a sequence of reversible one entries to zero.

Lemma 4.3. *Let $M(i, k)$ be an entry in a persistent matrix M, and let $M(i, l)$ and $M(i, r)$ be the first one entries in row i to the left and right of $M(i, k)$, respectively. Similarly, let $M(a, k)$ and $M(b, k)$ be the first one entries in column k above and below $M(i, k)$. Then $M(i, k)$ is reversible if and only if $r = a$, $M(a, l) = 1$ and $M(b, a) = 1$.*

Proof. It is easy to see that if the entry satisfies the given conditions it is reversible. Thus we have to show the converse, i.e., that if an entry is reversible it satisfies the above conditions. Since $M(i, r)$ is the first one entry to the right of $M(i, k)$ we apply persistence conditions to the maximal interval of zeros $M(i, x)$, $k < x < r$, and obtain that $M(r, k) = 1$. We have to show that this is the first one entry above $M(i, k)$ in column k. So suppose $M(y, k)$ is the first one entry in column k above $M(i, k)$ and that $y > r$. Since the entry $M(i, k)$ is reversible the matrix M' obtained by changing only $M(i, k)$ to zero is persistent. However, in this new matrix, the entries from $M'(j + 1, k)$ to $M'(i, k)$ are contained in a maximal column of zeros and applying persistence conditions we get that $M'(i, r) = 0$ which is a contradiction. That $M(j, l) = 1$ follows from the fact that $M(i, k)$ is reversible and that $M'(i, l + 1), \dots, M'(i, j - 1)$ is a maximal horizontal interval of zeros in the resulting persistent matrix M'. That $M(b, j) = 1$ can be checked similarly. □

4.3. *Generation of a Persistent Matrix*

We now describe the algorithm that generates any given persistent matrix from the clique matrix K_n.

Algorithm I
Input: A persistent matrix S.
Output: A sequence $M_0, \dots, M_r$ of persistent matrices such that $M_0 = K_n$, $M_r = S$, and each matrix in the sequence has exactly one more zero entry than its predecessor.

```
$M := K_n$ ($M$ is the current matrix in the sequence)
For $j := 2, n - 1$ do
  For $i := j + 1, n$ do
    For $k := j - 1$ downto 1 do
      If $S(i, k) = 0$ and $M(i, k) = 1$ and
        $v_j$ is a blocking vertex for $S(i, k)$ Then
      Begin
        $M(i, k) := 0$
        Output $M$
      End
```

In the remainder of this section we show that when the above algorithm terminates, $S = M$ and that each time an entry in M is changed from one to zero it is in fact a reversible entry. Every iteration of the outermost do loop in the above algorithm is called a **phase** of the algorithm. The proof of correctness is by induction on the phases of the algorithm. For phases in which v_j is not a blocking vertex no changes are made to the matrix M. The key idea behind the proof is the following statement.

Lemma 4.4. *Let $S_{i_0}, \dots, S_{i_m}$ be the factors of the input matrix S. If v_{i_u} is a blocking vertex, then at the end of the i_u*th *phase $M = K_n \cap S_{i_0} \cap \cdots \cap S_{i_u}$.*

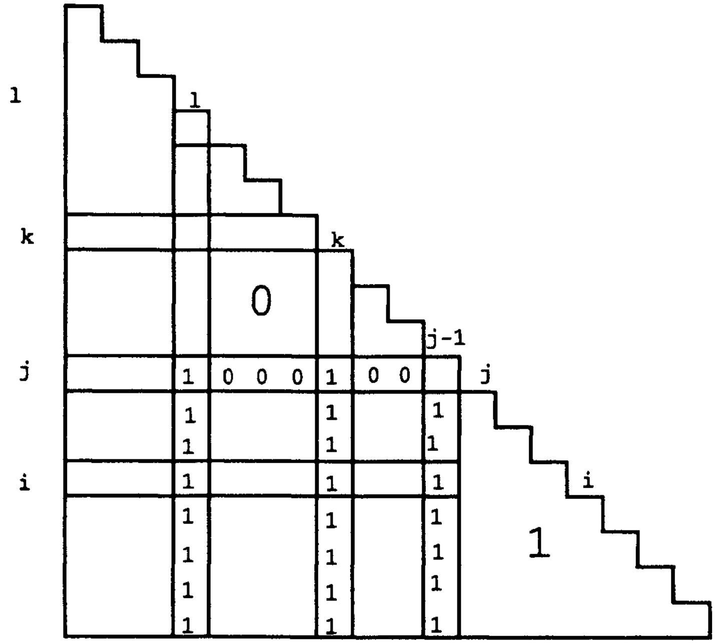


Fig. 2. The matrix S at the start of a phase j.

Proof. By induction on u. The base case is the clique matrix and is trivial. Thus, we suppose, inductively, that at the end of the i_{u-1}th iteration, the matrix constructed is $K_n \cap S_{i_0} \cap \cdots \cap S_{i_{u-1}}$. We will show that the matrix constructed at the end of the i_uth phase is $K_n \cap S_{i_0} \cap \cdots \cap S_{i_u}$.

Consider initially, the situation at the beginning of the i_uth phase. The matrix M at this point is shown in Fig. 2. In what follows, let $i_u = j$ for notational convenience. The rectangle defined by the entries $M(x, y)$, $j + 1 \leq x \leq n$, $1 \leq y \leq j - 1$, is called the **rectangle of influence** in phase j. These are the only entries changed in this phase. We make the following observations:

1. The subgraph induced by $v_{j-1}, \ldots, v_n$ is a clique.
2. The entries $M(x, j - 1)$, $j < x \leq n$, are all one.
3. For $j > y$, if $M(j, y) = 1$, $j > y$, then $M(x, y) = 1$ for all $x > j$.

Now, assume that the algorithm has performed correctly so far within the phase and is currently changing $M(i, k)$ from a one to a zero (see Fig. 3). Since the change occurs at this phase, $S(i, k) = 0$ and v_j is a blocking vertex for $S(i, k)$. By Proposition 4.1 v_j is a blocking vertex for every $S(i, r)$, $j < r \leq k$, and $S(s, k)$, $i \leq s < j$. Under the assumption that the algorithm has performed correctly so far all these entries are also zero in the current matrix. Applying the persistence conditions to M we have that $M(j, k) = 1$. By observation 3 above, we have that the first one entry in column k below $M(i, k)$ is $M(i + 1, k)$ since the entry was one at the beginning of the phase and it has not yet been processed by the algorithm in this phase. By

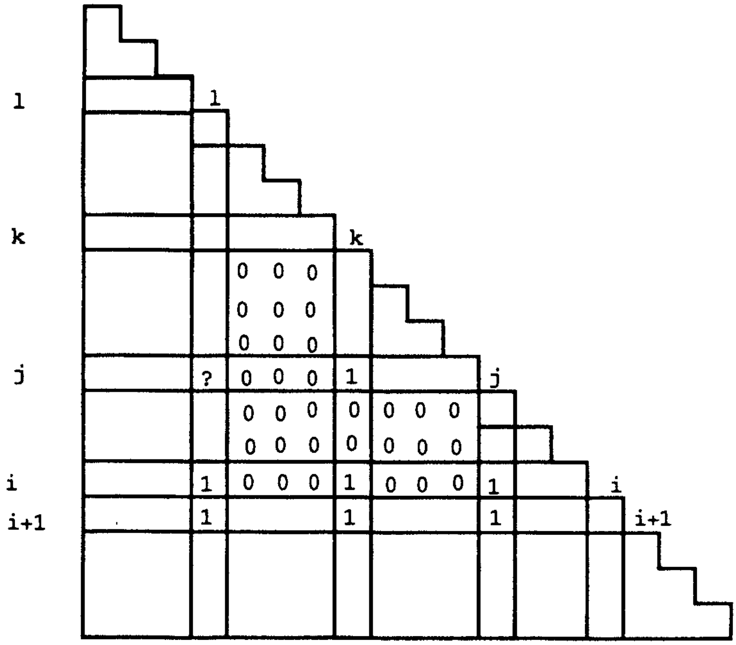


Fig. 3. A generic step within phase j.

observation 2 above we have that $M(i+1, j) = 1$. We have thus satisfied almost all the conditions for the reversibility of $M(i, k)$ as stated in Lemma 4.3. The only condition remaining to be checked is if $M(j, l) = 1$ where $M(i, l)$ is the first one entry to the left of $M(i, k)$ on row i.

So suppose $M(j, l) = 0$. This implies $S(j, l) = 0$. Applying persistence conditions to the maximal horizontal interval of zeros $M(i, l+1), \dots, M(i, k-1)$ we have that $M(k, l) = 1$. Also it was shown above that $M(j, k) = 1$. These entries must be one in the target matrix S since if they were zero their blocking vertices would have to be smaller than j and so they would already have been changed to zero in the current matrix. Thus $S(k, l) = 1$ and $S(j, k) = 1$. Since $S(j, l) = 0$ we have that v_k is a blocking vertex for $S(j, l)$. Also, v_j is a blocking vertex for $S(i, k)$.

We now show that the above implies that $S(i, l) = 0$ and that v_k is a blocking vertex for $S(i, l)$. Suppose, to the contrary, that $S(i, l) = 1$. Then, since $S(k, l) = 1$, $S(j, k) = 1$, and $S(j, l) = 0$, the persistence property implies that every $S(x, l)$, $k < x < j$, must be zero. Further, every $S(y, l)$, $j < y < i$, must be zero. Applying persistence conditions to the resulting maximal vertical interval of zeros $M(k+1, l), \dots, M(i-1, l)$ we have that $S(i, k) = 1$ which contradicts the hypothesis that $S(i, k) = 0$. Thus $S(i, l) = 0$. Further, it is clear that any ordered path from v_l to v_i in S must pass through v_k. Thus v_k is a blocking vertex for $S(i, l)$. However, this implies that $M(i, l)$ would have been changed to zero in phase k and thus contradicts the fact that $M(i, l) = 1$ in the current matrix. Thus we have that $M(j, l)$ must be one. Thus $M(i, k)$ is a reversible entry.

The algorithm changes any one entry to zero if and only if v_j is a blocking vertex for the entry. Thus an entry is zero in the rectangle of influence if and only if it is zero in the factor S_j. The elements from $v_j, \dots, v_n$ still constitute a clique since they are outside the rectangle of influence and thus the matrix constructed at the end of this phase is $K_n \cap S_{i_0} \cap \cdots \cap S_{i_u}$. □

The time complexity of a naive implementation of the above algorithm can be seen to be $O(n^4)$. The most expensive step within the innermost loop is to check if v_j is a blocking vertex for $S(i, k)$. However, it can be shown that this step is equivalent to checking if $M(i, k)$ is reversible. By Lemma 4.3 this can be done in a straightforward way in $O(n)$ time, which gives the overall complexity of $O(n^4)$. However, this can be reduced to $O(n^3)$ by use of a data structure that allows us to check if $M(i, k)$ is reversible in constant time. This is done by maintaining for each entry $M(i, k)$ the position of the first one entry to its right and left in row i and the position of the first one entry above and below it in column j. Such a data structure can be initialized in $O(n^2)$ time for the clique matrix. It can be seen that each time an entry is changed from one to zero the data structure can be updated in constant time using Lemma 4.3. Thus the overall algorithm can be implemented to run in $O(n^3)$ time.

5. A Reconstruction Algorithm for Balanced Tableaux

We show in this section that every persistent matrix is the skeleton of a balanced tableau. The main idea is to use Algorithm I developed in the last section to guide an algorithm that moves through maximal chains in the weak Bruhat order of the symmetric group in such a way that the changes made to the matrix in Algorithm I are mirrored in the changes made to the skeletons of the tableaux corresponding to these chains.

5.1. *Coxeter Transformations*

Recall from Section 2.1 that maximal chains in $(S_n, \leq_{WB})$ are uniquely determined by their reduced words. There are two types of transformations defined on maximal chains via their reduced word representations.

Proposition 5.1. *If W is the reduced word corresponding to a maximal chain in $(S_n, \leq_{WB})$, then*:

- *The word obtained by interchanging two adjacent symbols $s_i s_j$ in W such that i and j differ by more than one is also the reduced word of a maximal chain.*
- *The word obtained by changing three adjacent symbols in W of the form $s_i s_{i+1} s_i$ to $s_{i+1} s_i s_{i+1}$ is also the reduced word of a maximal chain.*

The first transformation is called a Coxeter type I *transformation and the second is called a Coxeter type* II *transformation.*

Coxeter proved that maximal chains in $(S_n, \leq_{WB})$ form a single orbit under these transformations, in the sense that given two maximal chains with reduced words W and W' a sequence of Coxeter transformations exists mapping W to W' [10].

We first interpret the Coxeter transformations on the balanced tableau representations of maximal chains. The following proposition follows directly from the definitions.

Proposition 5.2. *Suppose T is a balanced tableau of shape $SS(n)$.*

- *A type* I *Coxeter transformation corresponds to switching two entries x and $x + 1$ provided they are not in the same row or column.*
- *A type* II *Coxeter transformation corresponds to interchanging the entries $x - 1$ and $x + 1$ in two mate cells in T.*

The following proposition describes the effect of Coxeter transformations on the skeletons of balanced tableaux. Its proof is straightforward and is omitted.

Proposition 5.3. *Suppose a balanced tableau T' is obtained from a balanced tableau T by a Coxeter transformation ρ. If ρ is of type* I, *then T and T' have the same skeleton. If ρ is of type* II *and the interchanged mates are $T(i,j)$ and $T(j,k)$, then T and T' have the same skeleton if and only if there is an index $r \neq j$ with $k < r < i$ and $T(r,k) > T(i,k)$.*

Thus Coxeter transformations applied to balanced tableaux can be used to define new balanced tableaux with possibly different skeletons. This is the underlying idea behind the algorithm that constructs a balanced tableau for a given persistent skeleton matrix. We start with the clique matrix which has a particularly simple balanced tableau. We then apply certain specific sequences of Coxeter transformations in such a way that the underlying skeletons change precisely in the order defined by Algorithm I, which derives the given matrix from the clique matrix. These sequences of transformations are defined in the following section.

5.2. *Constructing a Balanced Tableau from a Persistent Matrix*

Let W be the reduced word of a maximal chain in $(S_n, \leq_{WB})$. A **simplicial sequence** is a maximal subword of adjacent letters in W of the form $s_y \; s_{y-1} \; \cdots \; s_1$ for some $y \leq n - 1$. We abbreviate such a sequence as π_y. The effect of π_y on a permutation σ is to move the symbol in position $y + 1$ to the first position.

Now we define two sequences of Coxeter transformations on maximal chains (augmentations and flushes) that contain simplicial sequences. They are the basic building blocks of the reconstruction algorithm.

Lemma 5.4. *If $W = us_x\pi_y u'$ is a maximal chain containing a simplicial sequence π_y and u, u' are (possibly empty) substrings of W and $x < y$, then $W' = u\pi_y s_{x+1}u'$ is a maximal chain.*

Proof. W can be written as $us_x s_y s_{y-1} \cdots s_{x+2}s_{x+1}s_x \dots s_1 u'$. Since $x < y$, the sequence of Coxeter type I transformations $s_x s_y, s_x s_{y-1}, \dots, s_x s_{x+2}$ transforms W into the reduced word $us_y s_{y-1} \cdots s_{x+2}s_x s_{x+1}s_x \cdots s_1 u'$. A Coxeter type II transformation at $s_x s_{x+1} s_x$ gives the maximal chain $us_y \cdots s_{x+1}s_x s_{x+1}s_{x-1} \cdots s_1 u'$. Now the sequence of Coxeter type I transformations $s_{x+1}s_{x-1}, s_{x+1}s_{x-2}, \dots, s_{x+1}s_1$ produces the chain $us_y s_{y-1} \cdots s_1 s_{x+1}u'$ which is precisely $W' = u\pi_y s_{x+1}u'$. □

Consider a subword $s_x\pi_y$ with $x < y$ in the reduced word of a maximal chain. The operation that transforms the chain $W = us_x\pi_y u'$ to the chain $W' = u\pi_y s_{x+1}u'$ is called an **augmentation** of W at π_y. For example, the augmentation of the chain $\pi_2 s_2 s_3 \pi_4 s_3 s_2$ at π_4 produces the chain $\pi_2 s_2 \pi_4 s_4 s_3 s_2$.

From the proof of Lemma 5.4, an augmentation is really a composition of Coxeter transformations. Further, any augmentation involves a sequence of Coxeter type I transformations followed by a **single** Coxeter type II transformation which is then followed by another sequence of Coxeter type I transformations. We refer to the type I transformations as the **initial** and **final** sequences of an augmentation and the Coxeter type II transformation is unambiguously referred to as the type II step of an augmentation. Note that the initial and final sequences of an augmentation may be empty if $x = y - 1$ or $x = 1$, respectively.

Similarly, let $W = us_x v\pi_y u'$ be a maximal chain in $(S_n, \leq_{WB})$ where u, u' are (possibly empty) substrings of W and v is a nonempty substring of W. If, for each symbol s_j in v, $x < j$ and $(j - x) > 1$, then $W' = uvs_x\pi_y u'$ is a maximal chain. We refer to the above operation that transforms the chain W into the chain W' as **flushing** the symbol s_x toward π_y. It, too, is a composition of Coxeter transformations, this time restricted only to transformations of type I.

We are now in a position to describe the algorithm that reconstructs a balanced tableau for a given persistent matrix (Algorithm II). The basis of the algorithm is the procedure developed in the last section for generating the given matrix from the clique matrix K_n. The clique matrix is the skeleton of the balanced tableau whose maximal chain is $\pi_1\pi_2 \cdots \pi_{n-1}$. As a one entry is changed to zero in the generation procedure (Algorithm I), we interpret this operation via a sequence of flushes and augmentations on the maximal chain corresponding to the current tableau such that the resulting maximal chain corresponds to a balanced tableau whose skeleton differs from the previous skeleton in precisely the entry changed from one to zero in Algorithm I.

Note the close correspondence between Algorithms I and II. Both algorithms examine the matrix entries in exactly the same order. Each iteration of the outermost loop for j in the above algorithm is defined as a **phase** of the algorithm. Each iteration of the loop for i is defined as the processing of row i in phase j. The algorithm modifies the tableau only when the conditions of the if statement are satisfied, therefore we say a **step** of the algorithm occurs when it executes the corresponding block of statements. The proof of correctness of this algorithm takes up the rest of this section.

```
Algorithm II
Input: A persistent matrix $S$.
Output: A balanced tableau $T$ whose skeleton is $S$.

$M := K_n$ ($M$ is the current matrix in the sequence)
$W := \pi_1\pi_2 \cdots \pi_{n-1}$
$T := Tableau(W)$
For $j := 2, n-1$ do
  For $i := j+1, n$ do
    For $k := j-1$ downto 1 do
      If $M(i,k) = 1$ and $S(i,k) = 0$ and
        $v_j$ is a blocking vertex for $S(i,k)$ Then
      Begin  {What we call a step of the algorithm
                begins here}
        $r := 0$
        $s :=$ column of the first one entry ot the left of
               $M(i,k)$ (or zero if no such entry exists)
        Repeat
          $x = i - k - 1 + r$
          Flush $s_x$ toward $\pi_{i-1}$
          Augment $W$ at $\pi_{i-1}$
          $T := Tableau(W)$
          $M := Matrix(T)$
          $r := r + 1$
        Until $r = k - s$
      End  {of a step of the algorithm}
```

5.3. *The Invariant for Algorithm II*

Consider a balanced tableau T with skeleton M. Let $M(i,k)$ be a one entry on row i, and let $M(i,l)$ be the first one entry to its left (assume $l = 0$ if no such entry exists). Since $M(i,k) = 1$, it follows that $T(i,k)$ is larger than all the entries that lie above it in its column, and, by the balance property, we can conclude that it is smaller than all the elements to its right on its row. In particular, if $M(i,r)$ is the first one entry to the right of $M(i,k)$, then $T(i,k) < T(i,r)$. We say that row i of the tableau is **well distributed** if, in addition, $T(i,k) < T(i,z) < T(i,r)$ for all z, $l < z < k$, and this property holds for each one entry $M(i,k)$ on row i. We say that row i of the tableau is **well distributed from** m_0 if m_0 is the smallest value on row i of T. Row i is said to be **strongly well distributed** if it is well distributed and in addition satisfies the property that the $i-1$ values on the row are consecutive. A vertex j in a persistent matrix is defined to be **simple** if $M(i,j) = 1$ for all $i > j$.

Given a maximal W in $(S_n, \leq_{WB})$ that contains a simplicial sequence π_q let $\theta(\pi_q)$ denote the *position* at which the first letter (i.e., s_q) of π_q occurs within W. For example, suppose the reduced word of a chain is

$$s_1s_2s_1s_3s_4s_3s_2s_1s_5s_6s_5s_4s_3s_2s_1s_5s_4s_3s_7s_6s_5s_4s_3s_2s_1s_3s_6s_5,$$

which can be written as $\pi_1\pi_2s_3\pi_4s_5\pi_6s_5s_4s_3\pi_7s_3s_6s_5$. $\theta(\pi_7) = 19$ and $\theta(\pi_6) = 10$.

Let W be a maximal chain in $(S_n, \leq_{WB})$ and let M be the skeleton of its associated balanced tableau T. W (resp. T) is said to be **good** with respect to j when, for every q, $j < q < n$:

- If q is a simple vertex in M, then there is a unique simplicial sequence π_q in W and row $q + 1$ is strongly well distributed from $\theta(\pi_q)$.
- If $j < q < p$ and p, q are simple vertices, then $\theta(\pi_q) < \theta(\pi_p)$.

Note that if a maximal chain (resp. balanced tableau) is good with respect to j, then it is good with respect to j' for $j < j' < n$.

Recall that a step of Algorithm II is said to occur when the conditions of the if statement are met, i.e., $M(i,k) = 1$, $S(i,k) = 0$, and v_j is a blocking vertex for $S(i,k)$ for some values of i, j, k such that $k < j < i$. Each step of the algorithm can thus be associated with a unique triple of values of the variables i, j, k and the value of j determines the phase in which the step occurs. The steps of the algorithm can be ordered according to the sequence of the values taken by these variables. Also note that for every step of Algorithm II there is a corresponding step in Algorithm I in which exactly one entry in the matrix was changed from one to zero.

The invariant maintained by Algorithm II is described in the following lemma.

Lemma 5.5. *At the start of each step of Algorithm* II:

1. *The current matrix M is identical to the matrix produced by Algorithm* I *at the corresponding step and M is the skeleton of the current balanced tableau T.*
2. *T is good with respect to j, where j is the phase in which the step occurs.*

The proof that the above invariant is maintained, is the main task in proving the correctness of Algorithm II and it takes up most of the rest of the paper.

5.4. *Proof of Lemma 5.5*

The proof proceeds by induction on the steps of the algorithm.

Basis. Consider the start of the very first step of the algorithm. Using the order in which the matrix entries are processed by the algorithm, it is readily verified that the first step occurs when the algorithm encounters an entry $M(j+1, j-1) = 1$ such that $S(j+1, j-1) = 0$ for the first time (i.e., the smallest such j). By Proposition 4.1, such an entry must always exist (unless the target matrix is the clique matrix) and since v_j is the only blocking vertex for this entry, this step occurs in phase j.

At this point no changes have been made to the initial matrix or tableau. Thus the current matrix is the clique matrix K and the current chain is $W = \pi_1 \pi_2 \cdots \pi_{n-1}$. The first invariant condition is thus trivially satisfied and it can be verified that the tableau corresponding to this chain is good with respect to j, thus satisfying the second invariant condition. At this step the repeat loop of the algorithm executes once and it augments s_1 at π_j. It can be verified that the current chain after this step is

$$\pi_1 \pi_2 \cdots s_{j-1} \cdots s_2 \pi_j s_2 \pi_{j+1} \cdots \pi_{n-1},$$

whose matrix has exactly one zero entry in $M(j+1, j-1)$.

Inductive Hypothesis. Assume that the algorithm has performed correctly for $t-1$ steps, $t \geq 1$, and that at the start of the tth step the two conditions of the above invariant hold. Also suppose that at this step the algorithm is currently in phase j where it is processing entry $M(i,k)$. At this point, $M(i,k)=1$, $S(i,k)=0$, and v_j is a blocking vertex for $S(i,k)$.

We have to show that when the repeat loop terminates at the end of the tth step:

1. The matrix M changes in precisely the same entry that would have been changed by Algorithm I at step t.
2. M is the skeleton of the current tableau, which is good with respect to j.

Inductive Step. In the following claims and proofs we assume implicitly that the inductive hypothesis holds at the start of the tth step. For each step $u<t$, this implies that the matrix M at the *end* of that step was identical to the corresponding matrix produced by Algorithm I at that step. Since each step in Algorithm I changes exactly one matrix entry from one to zero, for steps $u<t$ we may assume that exactly one entry in the matrix was changed from one to zero by Algorithm II in each step. Thus, at the start of a phase the current matrix will be as shown in Fig. 2, and at an intermediate point in a phase it will be as shown in Fig. 3.

Two easy facts about the order in which these entries are changed are implicitly used in the proof that follows. They follow directly from the proof of Algorithm I.

1. If $M(i,k)$ and $M(i+1,k)$ are zero in the current matrix, then they were changed to zero in the same phase.
2. If $M(i,k)=0$ and $M(j,k)$ is the first one entry above $M(i,k)$ in the current matrix, then $M(i,k)$ was changed to zero in phase j.

In the initial chain constructed by the algorithm each π_i is adjacent to π_{i+1} for $1 \leq i \leq n-2$. However, as the algorithm proceeds, successive augmentations at π_i produce some symbols "in between" π_i and π_{i+1}. These symbols may then be "consumed" by augmentations at π_{i+1}. When an entry $M(i,k)$ is changed from one to zero, the symbol s_{i-k} appears in between π_{i-1} and π_i (from the description of the algorithm, s_{i-k-1} is flushed toward π_{i-1} and the subsequent augmentation at π_{i-1} produces the symbol s_{i-k} as per Lemma 5.4). In Claims 5.6–5.9 we use the fact that the inductive hypothesis holds in all steps $u<t$ to deduce the order in which these symbols were produced and consumed in all the previous steps. This allows us to deduce the structure of the chain W at the start of the tth step.

Claim 5.6. *If $M(i,k)$ is changed from one to zero in a step $u<t$, within a phase $j' \leq j$, then every symbol s_x, $i-k \geq x \geq i-j'-2$, has been flushed toward π_{i-1} and subsequently augmented at π_{i-1} in phase j'.*

Proof. Let $M(i,q_0),\ldots,M(i,q_t)$, $t \leq j'-k-1$, be the entries to the right of $M(i,k)$ changed to zero in phase j' (in that order). Note that q_0 is always $j'-1$ in phase j'. If $t=j'-k-1$, then every element to the right of $M(i,k)$ is changed to zero precisely in phase j' and at each step when an entry $M(i, l)$, $k + 1 \leq l \leq j' - 1$, was changed to zero, $M(i, l - 1)$ was one. Thus, the repeat loop in Algorithm II executed exactly once in each case, the symbols flushed are precisely s_{i-l-1}, $k + 1 \leq l \leq j' - 1$, and we have the required sequence of augmentations. If $t < j' - k - 1$, then some of the entries to the right of $M(i, k)$ were changed to zero in phases $< j'$. Thus, for some of the entries $M(i, l)$, $k + 1 \leq l \leq j' - 1$, changed to zero in phase j', $M(i, l - 1) = 0$, and s (the position of the first one entry to the left of $M(i, l)$) is greater than $l - 1$. However, for each such entry, the repeat loop of the algorithm executes $l - s$ times and the symbols flushed are $s_{i-l-1+r}$, $r = 0, \ldots, l - s - 1$, which correspond precisely to the zero entries to the immediate left of $M(i, l)$. Thus, every element in the required sequence is flushed and subsequently augmented. □

Claim 5.7. *If a symbol s_x with $x = i - y - 1$ was augmented by the algorithm when processing row i at step u of the algorithm, with $u < t$ (the current step), then $M(i, y) = 0$ at the end of step u.*

Proof. Suppose, in step u, that s_x was augmented by the algorithm when it was processing an entry $M(i, k')$ in phase j'. At this point $M(i, k') = 1$, $S(i, k') = 0$, and j is a blocking vertex for $S(i, k)$. If $k' = y$, then the entry $M(i, y)$ (which was changed to zero by Algorithm I at that step) will be zero in the current matrix at the end of step u. If $k' \neq y$, then s_x would have been augmented by the algorithm at step u only if $M(i, y)$ belonged to the block of zero entries in between $M(i, k')$ and the first one entry to its left. Therefore, the entry had already been changed to zero in an earlier step of the algorithm. □

Claim 5.8. *Let $M(i, y) = 0$ in the current matrix and let $M(i, r)$ be the first one entry to its right. If the symbol s_{i-y-1} was augmented at π_{i-1} by the algorithm at step u with $u < t$, then exactly one of the entries $M(i, q)$, $y \leq q < r$, was changed to zero in that step.*

Proof. s_{i-y-1} is augmented by the algorithm either when $M(i, y)$ is changed to zero, or the entry $M(i, y)$ is a zero entry that appears in the sequence of consecutive zeros between an entry being changed to zero and the first one entry to its left. Clearly, such an entry has to be to the left of $M(i, r)$ and to the right of $M(i, y)$. □

From the above claims we obtain the following important result characterizing when a symbol appears between two simplicial sequences π_{i-1} and π_i.

Claim 5.9. *At the start of step t, a symbol s_x with $x = i - y$ appears in between π_{i-1} and π_i in the current chain constructed by the algorithm if and only if $M(i, y) = 0$ and an entry $M(i, r)$, $y \leq r$, exists such that $M(i, q) = 0$, $y \leq q \leq r$, and $M(i + 1, r) = 1$.*

Proof. For the forward direction, suppose an entry $M(i, r)$ exists that satisfies the required conditions. If there is more than one such entry, choose r to be as small as possible. In the consecutive sequence $M(i, y), \ldots, M(i, r)$ of zero entries, in the current matrix, let $M(i, q)$ be the last entry changed from one to zero by the algorithm before the current step. We first show that $r = q$. First note that it is impossible that $q > r$ since the order in which the entries in a row are processed ensures that when $M(i, r)$ was changed to zero, all the entries $M(i, y), \dots, M(i, r + 1)$ were already changed to zero. Therefore suppose $q < r$. Since $M(i, r) = 0$ and it could have been changed from one to zero only after phase r, it follows that $M(i, q)$ was also changed from one to zero in a phase $r' > r$. Therefore, in the current matrix there is a one entry $M(r', q)$ with $r < r' < i$. However, by the persistence condition, since $M(i + 1, r) = 1$ this implies $M(i + 1, q) = 1$ in the current matrix. Nevertheless, this contradicts our choice of r as the smallest index satisfying the conditions above. Therefore, $M(i, r)$ was the last entry in this consecutive sequence of entries changed to zero. The sequence s_{i-q}, $y \le q \le r$, was produced between π_{i-1} and π_i at this point. The s_{i-y} which was produced in the above step cannot be flushed toward π_i unless s_{i-r} is first flushed and augmented at π_i. However, if this happens Claim 5.7 implies that $M(i + 1, r)$ must be zero in the current matrix. Therefore, s_x remains in between π_{i-1} and π_i in the current matrix.

Conversely, suppose s_x appears in between π_{i-1} and π_i. By Claim 5.7 $M(i, y) = 0$ in the current matrix. Let $M(i, r)$ be the first one entry to its right in the current matrix. If there is a one entry in the positions $M(i + 1, y), \dots, M(i + 1, r - 1)$ we are done. So assume that all these entries are zero. Each of the corresponding entries on row i are also zero. Therefore, each pair $M(i, q), M(i + 1, q)$, $y \le q < r$, was changed in the same phase. From Claim 5.8 the only time symbol s_x could appear in between π_{i-1} and π_i is when one of the entries $M(i, q)$ $y \le q < r$, is changed to zero. However, each time this happens the symbol is removed by augmentation at π_i when the corresponding entry on row $i + 1$ is changed to zero. Therefore none of these symbols remain in between π_{i-1} and π_i which is a contradiction. □

Using Claims 5.6–5.9 we now deduce the structure of the chain W at the start of step t.

Claim 5.10. *At the start of step t the chain W has the form*

$$w v s_{i-s} s_{i-s-1} \cdots s_{i-k} s_{i-k-1} v' \pi_{i-1} w',$$

where v, v' are (possibly empty) substrings of W, and every symbol $s_y \in v'$ satisfies $x < y$ and $y - x > 1$, where $x = i - k - 1$.

Proof. First consider the situation when $i - 1 = j$. This corresponds to the case when step t occurs while processing the first row of entries in phase j. At the start of phase j the matrix M has the form given in Fig. 2. Since vertices $j - 1$ and j are both simple at this point, the chain at this point had the form $w\pi_{j-1}v'\pi_j w'$ where v' is a (possibly empty) sequence of symbols produced by augmentations at π_{j-1} in phases $< j$. Each symbol in v' corresponds to a zero entry on row j which was produced by an augmentation at π_{j-1} and since the symbol was not augmented at π_j the corresponding entry on row $j + 1$ is 1. Also when the entries in row $j + 1$ are processed in phase j all augmentations occur at π_j.

Let $M(j+1,k)$ with $k<j$ be any entry on row $j+1$ changed to zero in phase j. Since the entry is reversible at the time it is processed, the conditions of Lemma 4.3 imply that any one entry on row $j+1$ with a zero entry in the corresponding column of row j must occur strictly to the left of the first one entry to the left of $M(j+1,k)$. Therefore by Claim 5.9 $M(j+1,k)$ is changed to zero in phase j, and s_y is a symbol that was in between π_{j-1} and π_j at the start of phase j, then s_x with $x=j-k$ satisfies the property that $y-x>1$. Further each such symbol s_x is present in π_{j-1} at the start of phase j.

Now, whenever an entry is processed on row $j+1$ in phase j the chain must have the form $ws_{j-1}s_{j-2},\dots,s_x v'\pi_j w$, the symbols in v' being identical to those at the start of the phase. The symbols $s_{j-1}s_{j-2},\dots,s_x$ are obtained when the symbols $s_{x-1},\dots,s_1$ are removed from π_{j-1} by augmentations at π_j (which must occur according to Claim 5.6) in phase j. Therefore the chain has the required form.

Now suppose $i-1>j$. Since $M(i-1,k)=0$, and j is its smallest blocking vertex, the entry was changed from one to zero in phase j. Thus the symbol s_{i-k-1} was produced in between π_{i-2} and π_{i-1} at that time. Since $M(i,k)$ is one this symbol has not yet been augmented by the algorithm in this phase and thus still remains in between π_{i-2} and π_{i-1} when processing the current entry. If $M(i,k-1)=0$, then $M(i-1,k-1)$ is changed to zero in phase k. Likewise, $M(i,k-1)$ is changed to zero in phase k. The entries $M(i-1,k-2),\dots,$ $M(i-1,s+1)$ are changed to zero in phases $\leq k$. Finally, if $M(i-1,s)$ is zero, then it is changed to zero in phase j.

Thus, in phase j, when $M(i-1,k)$ was changed to zero, all the entries $M(i-1,k-1),\dots,M(i-1,s+1)$ were zero and all the symbols $s_{i-k-1},s_{i-k-2},$ $\dots,s_{i-s}$ would have been flushed and subsequently augmented at π_{i-2} by the algorithm *in that order* and appear in between π_{i-2} and π_{i-1}. Thus the chain, at the start of processing row i in phase j, can be written as

$$w\pi_{i-2}us_{i-s}s_{i-s-1}\cdots s_{i-k-2}s_{i-k-1}u'u''\pi_{i-1}w',$$

where u consists of symbols coming from elements changed to the left of $M(i-1,k)$ in phase j, the symbols in u' come from elements changed to the right of $M(i-1,k)$ in phase j, and the symbols in u'' come from elements changed in phases $<j$. The symbols in u'' come from elements changed in phases $<j$. The symbols in u'' can also be divided into those coming from entries to the right and left of $M(i-1,k)$. The symbols s_y in u'' that come from entries changed to the left of $M(i-1,k)$, trivially satisfy the property that $y>x$ and $y-x\geq 1$. However, we need the second inequality to be strict. To see that this is so we argue as follows. Suppose that the symbol s_{i-k-2} appears in u'' (this is probable since $M(i-1,k-1)$ was changed in phase $k<j$). However, since $M(i,k-1)$ was also changed precisely in phase k, the symbol created when $M(i-1,k-1)$ was changed to zero would have been consumed when $M(i,k-1)$ was changed to zero in the same phase and thus augmented at π_{i-1} in phase k. Thus the symbol cannot appear in between π_{i-2} and π_{i-1}.

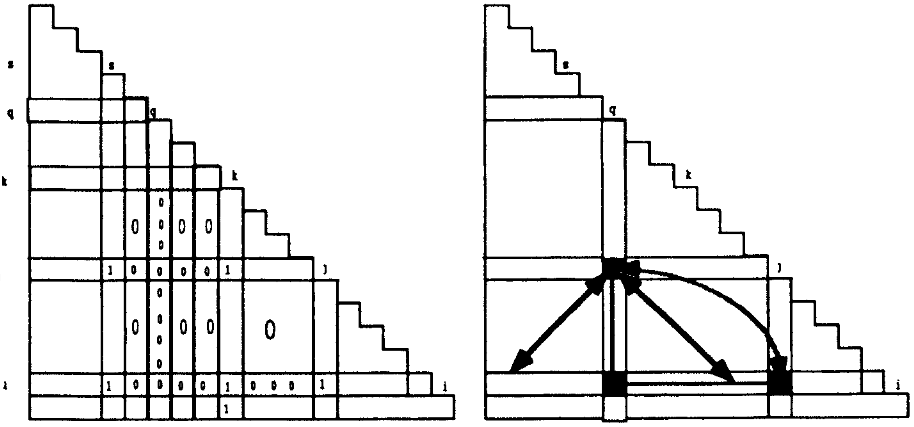
Now, the symbols in u' and the symbols in u'' coming from entries to the right of $M(i-1,k)$ are flushed and augmented when processing the entries to the right of $M(i,k)$ in phase j as per Claim 5.6. Thus the chain has the required form. □

We are now ready to examine the behavior of the algorithm once the repeat loop begins to execute. This loop executes $k - s$ times, where $M(i, s)$ is the first one entry to the left of $M(i, k)$. Figure 4 shows an abstract picture of the changes that occur in the tableau in any of these iterations. Figure 5 shows an example of the tableau modifications that occur when an augment step is performed, and Fig. 6 shows the relationship between the changes in the tableau from one iteration of the loop to the next. In these figures the shaded rectangle in the matrix indicates the entry currently being changed from one to zero.

We show that in the first $k - s - 1$ iterations of the repeat loop, the 0–1 matrix remains unchanged. However, the Coxeter type II transformations involved in the augmentations occur "closer" to the entry $T(i, k)$ after each iteration. Finally, in the $(k - s)$th iteration, the Coxeter type II transformation involves the mates of entry $T(i, k)$ thus changing the matrix at position $M(i, k)$. We show that the tableau remains good with respect to j at the end of this iteration. In the intermediate iterations, however, the values on row i are *not* well distributed. We therefore need a slightly more general invariant that holds during each iteration of the repeat loop.

Claim 5.11. *At the start of the* rth *iteration of the repeat loop* $0 \leq r \leq (k - s)$ *the following hold*:

1. *The value in* $T(i, j)$ *is the* xth *largest value on row* i *where* $x = i - k - 1 + r$.
2. *The number of entries* $T(i, l)$, $s < l < k$, *such that* $T(i, k) < T(i, l) < T(i, j)$ *is exactly* $k - s - r$.
3. $T(i, j) - 1$ *lies in some column* l *on row* i *such that* $s < l < k$.
4. *Row* i *is well ordered to the left of* $T(i, s)$.
5. *For each row* $q > j$ *the values in row* q *lie in between* $\theta(\pi_{q-1})$ *and* $\theta(\pi_{q-1}) + q - 1$.
6. $M(i, j)$ *is unchanged.*



Matrix

Balanced Tableau

Fig. 4. Tableau modifications within an iteration.

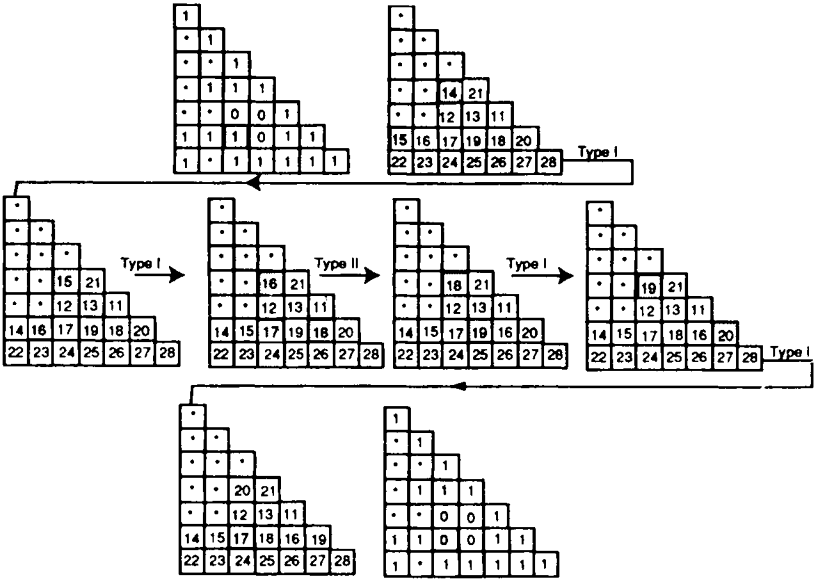


Fig. 5. Change in tableau during an augmentation.

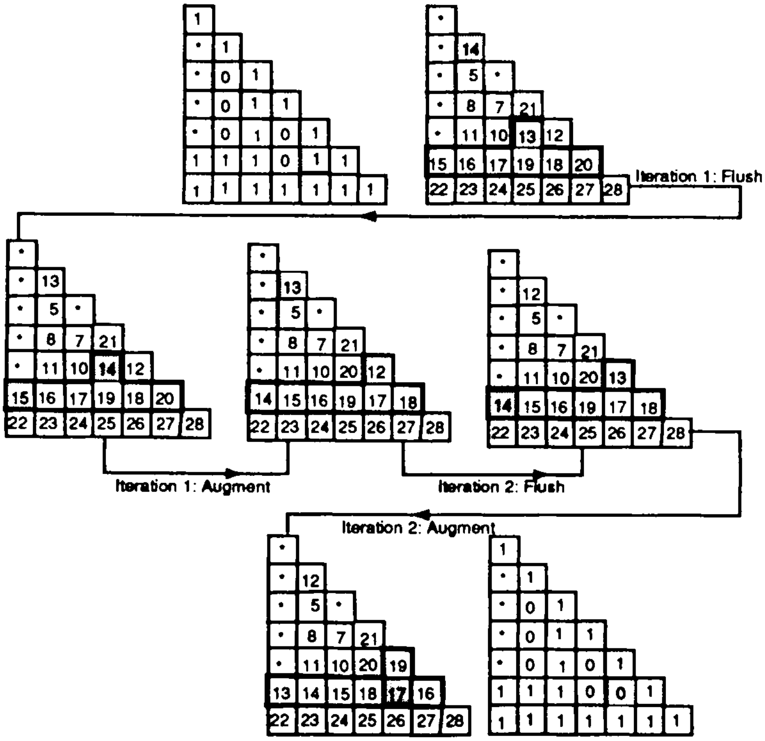


Fig. 6. Flushes and augmentations during different iterations.

Proof. We prove statements 1–6 simultaneously by induction on r.

Basis: $r = 0$. By the induction hypothesis of the overall algorithm, the balanced tableau and the chain at this point are good after j. To prove the statement 1 for $r = 0$, we note that since the values on row i are well distributed, the numbers larger than $T(i, j)$ are precisely the $i - j - 1$ one entries to its right and the $j - k - 1$ zero entries to its immediate left. Thus $T(i, j)$ is the $(i - k - 1)$th largest value on row i. Statement 2 also holds since the $k - s$ zero entries between $T(i, k)$ and $T(i, j)$ are precisely the entries that satisfy $T(i, j) < T(i, l) < T(i, j)$. Statements 3 and 4 also follow from the fact that row i is well distributed. Statement 5 follows from the fact that for every $q > j$ vertex q is simple and so every row is well distributed from $\theta(\pi_q)$. Statement 6 is obvious.

Inductive Hypothesis (IH1): At the start of the rth iteration $0 < r < k - s - 1$ statements 1–6 hold.

So, consider the rth iteration (Fig. 4). The symbol flushed and augmented by the algorithm in the rth iteration is s_x, $x = i - k - 1 + r$. By Claim 5.10 the flush step can be correctly performed at this step and since this involves only Coxeter type I transformations, the matrix remains unchanged at this point.

Now, we claim that no tableau entry involved in an interchange in the flush step lies in a row $> j$. For each row $q > j$, the values in row q of the tableau lie in the range $\theta(\pi_{q-1}), \dots, \theta(\pi_{q-1}) + q - 1$. Now, s_x, $x = i - k - 1$, lies between π_{i-2} and π_{i-1} and if p_x denotes the position of s_x in the chain, then $\theta(\pi_{i-2}) < p_x < \theta(\pi_{i-1})$. Since the values interchanged by the Coxeter transformations in the flush step are $p_x, p_x + 1, \dots, \theta(\pi_{i-1}) - 1$, none of these values can come from rows $> j$. Therefore, no row after j is affected by the flush step, and hence all the above statements hold after the flush step has been performed.

By IH1, the xth largest value lies in $T(i, j)$. Let m denote this value. From the definition of Coxeter transformations, we note that if $p_0 = \theta(\pi_{i-1})$, then the entries in the balanced tableau interchanged in the augmentations are precisely those containing the values $p_0 - 1, p_0, \dots, p_0 + i - 1$.

We now identify where the value $p_0 - 1$ lies. Within π_{i-1} the symbol s_x occurs x positions to the left of the symbol s_1. The type II Coxeter transformation involves the symbols s_x, s_{x+1}, s_x. The position of the symbol s_x within π_{i-1} is $m = \theta(\pi_{i-1}) + i - x$. Thus the type II Coxeter transformation involves the values m, $m - 1$, and $m - 2$ in the balanced tableau. By IH1, $m - 1$ lies in an entry $T(i, l)$, $s < l < k$, and $M(i, l) = 0$. Thus the Coxeter type II transformation interchanges the mates $T(i, j)$ and $T(j, l)$ of the entry $T(i, l)$. By Proposition 5.3 the matrix remains unchanged by this transformation which proves statement 6 above.

Now, it can be seen that after the augmentation:

- $T(i, l)$ is unchanged.
- $T(i, j)$ is reduced by two.
- $T(l, j)$ is increased by $i - 1$.
- Every other value on row i is uniformly reduced by 1.

There were exactly r entries in between $T(i, s)$ and $T(i, k)$ that were larger than $T(i, j)$. Since $T(i, j)$ is reduced by two and all other entries in this range are reduced by one, each of these remains larger than $T(i, j)$. However, since $T(i, l)$ was equal to

$T(i, j) - 1$ before augmentation and remains unchanged after the augmentation, the value in $T(i, l)$ after augmentation is larger than the value in $T(i, j)$ after augmentation. Thus there is one additional value in between $T(i, s)$ and $T(i, k)$ that is larger than $T(i, j)$. Thus, at the end of the (r)th iteration, the number of entries such that $T(i, k) < T(i, l) < T(i, j)$ is precisely $(k - s) - r$. This proves statement 2. To prove statement 5, we note that the only row after j affected by the augmentation is row i, and that the values in this row remain in the consecutive interval from $\theta(\pi_{i-1})$ to $\theta(\pi_{i-1}) + i - 1$ after the above arithmetic operations are performed. Statement 4 is true since all the values to the left of $T(i, s)$ were uniformly decreased by one. Statement 1 was true at the start of the rth iteration by IH1. The value in $T(i, j)$ is smaller than the $i - k - 1$ entries that were larger than it at the start of the 0th iteration and the $r - 1$ entries between $T(i, s)$ and $T(i, k)$. Since $T(i, l)$ is the only element in this process that has become larger than $T(i, j)$ in the rth iteration we have the required value for statement 1. Statement 3 is obvious. □

The Final Iteration. By the arguments in Claim 5.11, at the start of the $(k - s)$th iteration the value in $T(i, k)$ is $T(i, j) - 1$ and the Coxeter type II transformation occurs about the reversible entry $M(i, k)$ in the $(k - s)$th iteration and thus the matrix changes precisely at $M(i, k)$ at the end of the $(k - s)$th iteration.

We now complete the proof by showing that the row remains well distributed at the end of the last iteration of the repeat loop. Since every value to the left of $T(i, k)$ is uniformly reduced by one, we see that if $M(i, l_0)$ is the leftmost one entry on row i, then $T(i, l_0) = \theta(\pi_{i-1})$ and every element of the tableau to the left of $T(i,k)$ remains well ordered. To see the effect on the rest of the row consider the following. If $M(i, k - 2) = 1$, the value $m - 2$ was originally stored in $T(i, k - 1)$ and $T(i, j) - T(i, k - 1) = 2$ before the augmentation. Now after the augmentation $T(i, j)$ is reduced by two and $T(i, k - 1)$ is reduced by one. Thus $T(i, j) - 1$ is contained in $T(i, k - 1)$ after the augmentation as required by the well ordering. Further, since $T(i, k)$ remains unchanged, $T(i, j) + 1$ is contained in the sequence of zero entries to the immediate right of $T(i, j)$ as required. The relative distribution of the remaining elements remains unaffected as they are uniformly decreased by one. If $M(i, k - 2) = 0$, then the only difference is that the value $m - 2$ was originally stored in one of the zero entries to the left of $T(i, k - 2)$. It is easy therefore to establish that the row remains well distributed.

The only row after j that was affected by the flush and augmentation steps is row i, and thus every other row remains well distributed after this step. We have thus established that after the repeat loop terminates:

1. The matrix M changes in precisely the entry $M(i, k)$ which was changed from one to zero by Algorithm I at step t.
2. M is the skeleton of the tableau T which is good after j.

This concludes the proof of Lemma 5.5. □

Since Algorithm II produces the same matrix as Algorithm I after each step, it follows that the final tableau produced by the algorithm when it terminates is the target matrix S. Thus we have proved that Algorithm II correctly generates a balanced tableau whose skeleton is the given input matrix. This is summarized in the following lemma.

Lemma 5.12. *If M is a persistent matrix, then a balanced tableau T exists such that M is the skeleton of T.*

Lemmas 3.4 and 5.12 give us the main theorem of this paper.

Theorem 5.13. *G is a persistent graph if and only if a balanced tableau T exists such that $M(G)$ is the skeleton of T.*

5.5. Complexity of Algorithm II

Algorithm II thus allows us to reconstruct a balanced tableau from a given persistent matrix. The algorithm can be implemented to run in $O(n^5)$ time. In the innermost loop, techniques used implementing Algorithm I can be applied to check if v_j is a blocking vertex for $S(i, k)$ in constant time. The repeat loop executes at most $n - 1$ times. If we had to recompute the tableau and matrix at each step, then each of these computations could take $\theta(n^2)$ time thus giving us an overall complexity of $O(n^6)$. However, using the proof of Lemma 5.5, it can be seen that the changes made to the tableau and matrix are in fact localized. There are at most $n - 1$ flushes that need to be made in any iteration of the repeat loop and the augmentation involves at most $n - 1$ interchanges in the tableau. At the end of the augmentation the matrix changes in precisely one entry. Thus the overall repeat loop can be implemented to run in $O(n^2)$ time without explicit recomputation of the tableau and matrix. This gives an algorithm with overall complexity in $O(n^5)$.

6. Closing Remarks

We have proven that visibility graphs of staircase polygons are included in the class of persistent graphs. It follows from our characterization of persistent graphs (Theorem 5.13) that they partition the set of maximal chains in the weak Bruhat order; call this a persistent partition. This is a coarser partition than the one induced by semispace equivalence [18] which is in turn the same as the one derived from the closure operator defined in [2]. In the companion paper [4], we study the existence of a realizable chain in each equivalence class of our persistent partition. The existence of such a realizable chain plus the results of this paper give us an efficient algorithmic characterization of visibility graphs of staircase polygons. As a byproduct we obtain a characterization of visibility graphs of convex fans. This was considered a major stumbling block for the general problem because any visibility graph can be decomposed into a sequence of visibility graphs of convex fans. It may be interesting to find an easier proof of the main result of this paper and to investigate the optimality of the algorithms presented here.

We mention in closing that a relation between general visibility graphs and oriented matroid realizability has been established in [7].

Acknowledgments

We want to express our appreciation to Joseph O'Rourke, Richard Pollack, Bernd Sturmfels, and the anonymous referees for their constructive criticisms of preliminary versions of this work. Our gratitude to Kamal Abdali and NSF for financial assistance.

References

1. J. Abello, A study of an independence system arising in group choice via the weak Bruhat order, Ph.D. Dissertation, University of California, San Diego, CA, 1985.
2. J. Abello, The weak Bruhat order of S_σ, consistent sets, and Catalan numbers, *SIAM Journal of Discrete Mathematics*, **4** (1991), 1-16.
3. J. Abello and O. Egecioglu, Visibility graphs of staircase polygons with uniform step length, *International Journal of Discrete and Computational Geometry*, **3**(1) (1993), 27-37.
4. J. Abello, O. Egecioglu, and K. Kumar, Visibility graphs of staircase polygons and the weak Bruhat order, II: From maximal chains to polygons, Preprint.
5. J. Abello, O. Egecioglu, and K. Kumar, A combinatorial view of visibility graphs of simple polygons, *Proceedings of the Fifth International Conference on Computing and Information*, Ontario, 1993, pp. 87-92.
6. J. Abello, L. Hua, and S. Pisupati, On visibility graphs of simple polygons, *Congressus Numerantium* **90** (1992), 119-128.
7. J. Abello and K. Kumar, Visibility graphs and oriented matroids, *Proceedings of the Workshop on Graph Drawing*, Princeton, NJ, October 1994. Lecture Notes in Computer Science, Vol. 894, Springer-Verlag, Berlin, 1995, pp. 147-158.
8. D. Avis and G. T. Toussaint, An efficient algorithm for decomposing a polygon into star shaped polygons, *Pattern Recognition* **13** (1981), 295-298.
9. C. Coullard and A. Lubiw, Distance visibility graphs, *Proceedings of the Seventh ACM Symposium on Computational Geometry*, 1991, pp. 290-302.
10. H. S. M. Coxeter and W. O. J. Moser, *Generators and Relations for Discrete Groups*, 2nd edition Springer-Verlag, New York, 1965.
11. H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, New York, 1987.
12. P. Edelman and C. Greene, Combinatorial correspondences for Young tableaux, balanced tableaux, and maximal chains in the weak Bruhat order, *Contemporary Mathematics* **34** (1984), 220-255.
13. H. A. El-Gindy, Hierarchical decompositions of polygons with applications, Ph.D. Dissertation, School of Computer Science, McGill University, 1985.
14. H. Everett, Visibility graph recognition, Ph.D. Dissertation, Department of Computer Science, University of Toronto, 1990.
15. H. Everett and D. G. Corneil, Recognizing visibility graphs of spiral polygons, *Journal of Algorithms* **11** (1990), 1-26.
16. S. K. Ghosh, *On Recognizing and Characterizing Visibility Graphs of Staircase Polygons*, Lecture Notes in Computer Science, Vol. 318, Springer-Verlag, Berlin, 1988, pp. 132-139.
17. J. E. Goodman and R. Pollack, On the combinatorial classification of nondegenerate configurations in the plane, *Journal of Combinatorial Theory, Series A* **29** (1980), 212-233.
18. J. E. Goodman and R. Pollack, Semispaces of configurations, cell complexes of arrangements, *Journal of Combinatorial Theory, Series A* **37** (1984), 84-102.

19. J. E. Goodman and R. Pollack, Upper bounds for configurations and polytopes in R^d, *Discrete and Computational Geometry* **1** (1986), 219–227.
20. K. Kumar, Combinatorial aspects of point visibility, Ph.D. Dissertation, Department of Computer Science, Texas A & M University, 1993.
21. J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.
22. F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer-Verlag, New York, 1985.
23. R. Tamassia and I. Tollis, A unified approach to visibility representations of planar graphs, *Discrete and Computational Geometry* **1** (1986), 312–337.
24. S. Wismath, Characterizing bar line-of-sight graphs, *Proceedings of the 1st ACM Symposium on Computational Geometry*, 1985, pp. 113–145.

Received June 12, 1992, and in revised form May 14, 1994, and December 13, 1994.