StreamTable: An Area Proportional Visualization for Tables with Flowing Streams

Jared Albert Espenant
Department of Computer Science, University of Saskatchewan, Canada

Debajyoti Mondal
Department of Computer Science, University of Saskatchewan, Canada

Abstract

Let T be a two-dimensional table with each cell weighted by a nonzero positive number. A StreamTable visualization of T represents the columns as non-overlapping vertical streams and the rows as horizontal bands such that the intersection between a stream and a band is a rectangle with area equal to the weight of the corresponding cell. To avoid large wiggle of the streams, it is desirable to keep the consecutive cells in a stream to be adjacent. The difference between the area of the bounding box containing the StreamTable and the sum of the weights of T is referred to as the excess area. We attempt to optimize various StreamTable aesthetics (e.g., minimizing excess area, or maximizing cell adjacencies in streams).

- If the row permutation is fixed and the row heights are given, then we give an $O(rc)$-time algorithm to optimize these aesthetics, where r and c are the number of rows and columns, respectively.
- If the row permutation is fixed but the row heights can be chosen, then we discuss a technique to compute a StreamTable with small area and required cell adjacencies by solving a quadratically-constrained quadratic program, followed by iterative improvements. If the row heights are restricted to be integers, then we prove the problem to be NP-hard.
- If the row permutations can be chosen, then we show that it is NP-hard to find a row permutation that optimizes the area or adjacency aesthetics.

Keywords and phrases Geometric Algorithms, Table Cartogram, Streamgraphs

Digital Object Identifier 10.57717/cgt.v3i1.24

Related Version A preliminary set of results were presented at the 16th International Conference and Workshops on Algorithms and Computation [9].

Funding This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund and to the Natural Sciences and Engineering Research Council of Canada (NSERC).

1 Introduction

Proportional area charts and cartographic visualizations commonly represent data values as geometric objects. The table cartogram [10] is a brilliant way to visualize tables as cartograms, where each table cell is mapped to a convex quadrilateral with area equal to the cell's weight. Furthermore, the visualization preserves cell adjacencies and the quadrilaterals are packed together in a rectangle with no empty space in between (e.g., see Figure 1(e)). However, since the cells in a table cartogram are represented with convex quadrilaterals, it may sometimes become difficult to follow the rows and columns [14]. This motivated us to examine the scenario where each row is represented with a horizontal *band* (i.e., a region bounded by two horizontal lines) and the cells in each row are represented with axis-aligned rectangles inside the corresponding band. A Streamgraph is one such example where often the columns (instead of rows) are presented as bands.

Given a set of variables, a *streamgraph* visualizes how their values change over time by representing each variable with a flowing river-like stream (e.g., an x-monotone polygon).

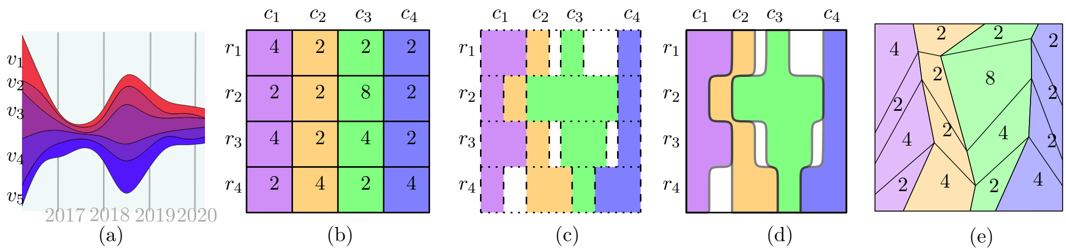


Figure 1 (a) A streamgraph. (b) A table T. (c) A StreamTable for T. (d) A StreamTable visualization with smooth streams. (e) A table cartogram for T.

WA 6.725	MT 0.989	ND 0.673	MN 5.304	WI 5.687	NY 19.378	VT 0.626	ME 1.328
OR 3.831	ID 1.568	SD 0.814	IA 3.046	NI 9.884	PA 12.702	NH 1.316	MA 6.548
NV 2.701	WY 0.564	NE 1.826	IL 12.831	IN 6.484	OH 11.537	CT 3.574	RI 1.053
UT 2.764	CO 5.029	KS 2.853	MO 5.989	KY 4.339	WV 1.853	MD 5.774	NJ 8.792
CA 37.254	NM 2.059	OK 3.751	AR 2.916	TN 6.346	SC 4.625	VA 8.001	DE 0.898
AZ 6.392	TX 25.146	LA 4.533	MS 2.967	AL 4.780	GA 9.688	FL 18.801	NC 9.535

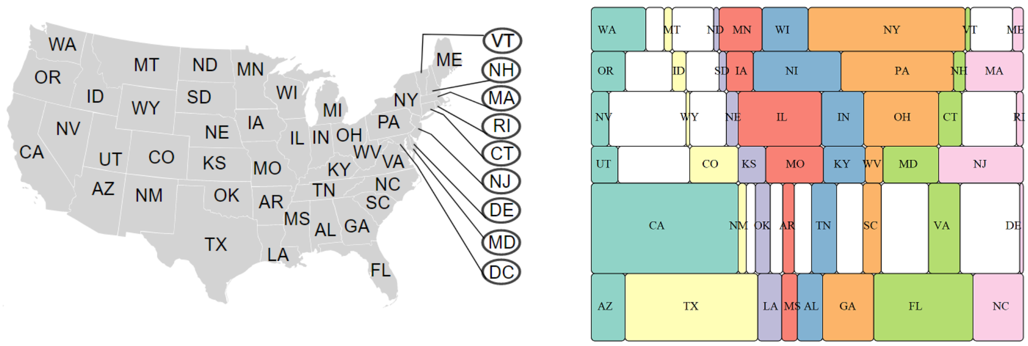


Figure 2 A grid map of the USA [8] and the corresponding StreamTable for the population of the states in 2010.

The width of the stream at a timestamp is determined by the value of the variable at that time. Figure 1(a) illustrates a streamgraph with five variables. Streamgraphs are often used to create infographics of temporal data [5], e.g., box office revenues for movies [3], various statistics or demographics of a population over time [15], etc. In this paper, we introduce StreamTable that extends this idea of a streamgraph to visualize tables or spreadsheets. Figure 2 shows a grid map of the USA [8] and the corresponding StreamTable for the population of the states in 2010. We now formally define a StreamTable.

1.1 StreamTable

Let T be an $r \times c$ table with $r \geq 1$ rows and $c \geq 2$ columns, where each cell is weighted by a nonzero positive number. A *StreamTable* visualization of T is a partition of an axis-aligned rectangle R into r consecutive horizontal bands that represent the rows of T, where each band is further divided into not necessarily adjacent rectangles to represent the cells of its corresponding row. A column q of T is thus represented by a sequence of rectangles corresponding to the cells of q, which we refer to as a *stream*. Furthermore, a StreamTable

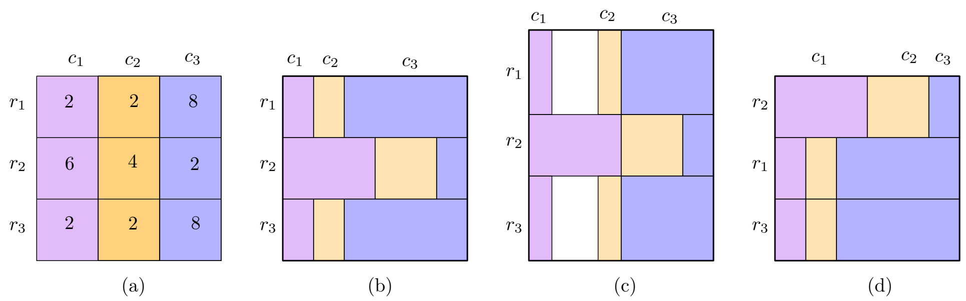


Figure 3 (a) A table. (b) A StreamTable with no excess area and 2 splits. (c) A StreamTable with non-uniform row heights, non-zero excess area, but no splits. (d) A StreamTable with no excess area and 1 split (obtained by reordering rows).

must satisfy the following properties.

P_1. The left side of the leftmost stream (resp., the right side of the rightmost stream) must be aligned to the left side (resp., right side) of R.

P_2. For each cell of T, the area of its corresponding rectangle in the StreamTable must be equal to the cell's weight.

Property P_1 ensures an aesthetic alignment with the row labels. Property P_2 provides an area proportional representation of the table cells. Figure 1(b) illustrates a table and Figure 1(c) illustrates a corresponding StreamTable. The bands (rows) are shown in dotted lines and the partition of the bands are shown in dashed lines. Figure 1(d) illustrates an aesthetic visualization of the streams after smoothing the corners.

Note that a StreamTable may contain rectangular regions that do not correspond to any cell. We refer to such regions as *empty regions* and the sum of the area of all empty regions as the *excess area*. While computing a StreamTable, a natural optimization criterion is to minimize this excess area. However, minimizing excess area may sometimes result into split or disconnected streams. Figure 3(b) illustrates a StreamTable with zero excess area, where the consecutive rectangles for column c_2 are not adjacent (i.e., no two consecutive rectangles of c_2 share a common boundary point). If a pair of cells are consecutive in a column but the corresponding rectangles are nonadjacent in the stream, then they *split* the stream. To maintain the stream connectedness, it is desirable to minimize the number of such splits. As illustrated in Figure 3(c)-(d), one may choose non-uniform row heights or reorder the rows to optimize the aesthetics. Such reordering operations also appear in matrix reordering problems [16] where the goal is to reveal clusters in matrix data. StreamTable computation also relates to floorplanning [6, 19] and area-universal rectangular layout problems [4, 7], where the horizontal adjacencies are not mandatory but vertical adjacencies must be preserved.

It is an intriguing question to study how StreamTable compares to Streamgraphs when it comes to human interpretation, perception and task performances. However, in this paper, our focus is entirely on computing a StreamTable by minimizing some natural optimization functions such as excess area and number of splits.

In the rest of the paper, we associate the terms 'rows' and 'columns' of an input table with 'bands' and 'streams' of its corresponding StreamTable, respectively.

1.2 Our contribution

We examine StreamTable from a theoretical perspective and explore several variants considering the following two questions.

Question 1 (StreamTable with no Split, Minimum Excess Area, and Fixed Row Ordering). Given an $r \times c$ table T, can we compute a StreamTable for T in polynomial time with no splits and minimum excess area? Note that in this problem, the StreamTable must respect the row ordering of T.

If the row heights are restricted to be integers, then we show that finding a minimum excess area no-split StreamTable is NP-hard (Section 2.2.2). In general, the problem of computing a no-split StreamTable with minimum excess area can be modeled leveraging a quadratically-constrained quadratic program, and a solution computed by non-linear programming solver may be iteratively improved by adjusting the row heights. However, this only provides a heuristic solution (Section 2.2.1). While Question 1 remains open, if the input additionally specifies an ordered set $(h_1, \ldots, h_r)$ of nonzero positive numbers to be chosen as row heights, then we can compute a StreamTable with minimum excess area in $O(rc)$ time (Section 2.1). Since choosing a fixed row height helps to obtain a fast algorithm and to compare the cell areas more accurately, we examined whether one can leverage the row ordering to further improve the StreamTable aesthetics.

Question 2 (Row-Permutable StreamTable with Uniform Row Heights). Given a table T and a non-zero positive number $\delta > 0$, can we compute a StreamTable in polynomial time by setting δ as the row height, and minimizing the excess area (or, the number of splits)? Note that while answering this question, the row ordering can be chosen.

We show that Question 2 is NP-hard (Section 3) in the following two settings: first, computing a StreamTable with no excess area and minimum number of splits is NP-hard, and second, computing a StreamTable with no splits and minimum excess area is NP-hard. The following table summarizes the results.

Table 1 Summary of the results.

	StreamTable with Fixed Row Ordering and Variable Row Heights	Row-Permutable StreamTable with Uniform Row Heights
Constraint: no-split Minimization: excess area	**Open**; but computable in $O(rc)$ time if row heights are given (Th. 2), and NP-hard if row heights are restricted to integers (Th. 3)	NP-hard (Th. 4)
Constraint: no excess area Minimization: num. of splits	Computable (if it exists) in $O(rc)$ time, trivial	NP-hard (Th. 5)

2 StreamTable with no split, min. excess area, fixed row order

In this section we compute StreamTables by respecting the given row ordering of the input table. We first explore the case when the row heights are given, and then the case when the row heights can be chosen.

2.1 Fixed row heights

Let T be an $r \times c$ table and let $(h_1, \ldots, h_r)$ be an ordered set of nonzero positive numbers to be chosen as row heights. We now introduce some notation for the rectangles and streams in the StreamTable. Let $w_{i,j}$ be the weight for the (i, j)th entry of T, where $1 \leq i \leq r$ and

$1 \leq j \leq c$, and let $R_{i,j}$ be the rectangle with height h_i and width $(w_{i,j}/h_i)$. Let $a_{i,j}$ and $b_{i,j}$ be the x-coordinates of the left and right side of $R_{i,j}$.

We now show that a StreamTable $\mathcal{R}$ for T with no splits and minimum excess area can be constructed using a greedy algorithm. For simplicity, we first describe the high-level steps of the algorithm and prove that the algorithm can produce a StreamTable with no-split and minimum excess area. We next show that these high-level steps can be implemented in $O(rc)$ time.

Algorithm $\mathcal{G}$ (Greedy-StreamTable)
Input: An $r \times c$ table T, where $r \geq 1$ and $c \geq 2$.
Output: A StreamTable $\mathcal{R}$ with no-split and minimum excess area.

Step 1. Draw the rectangles $R_{i,1}$, where $1 \leq i \leq r$, of the first column such that they are left aligned.

Step 2. For each $j < c$, draw the jth stream by minimizing the sum of x-coordinates $a_{i,j}$, and ensure that the stream remains connected, i.e., place the rectangles $R_{i,j}$ so that no splits appears in the stream.

Step 3. Draw the rectangles $R_{i,c}$ of the last column by minimizing the maximum x-coordinate over $b_{i,c}$, and ensuring that the rectangles are right aligned.

For every column j, let $A(\mathcal{R}, j)$ be the orthogonal polygonal chain determined by the left side of $R_{i,j}$. Similarly, we define (resp., $B(\mathcal{R}, j)$) for the right side of $R_{i,j}$. We now have the following lemma.

▶ **Lemma 1.** *Given an $r \times c$ table T, where $r \geq 1$ and $c \geq 2$, Algorithm $\mathcal{G}$ computes a StreamTable $\mathcal{R}$ with no splits and minimum excess area.*

Proof. We employ an induction on the number of columns of $\mathcal{R}$. For $c = 2$, it is straightforward to verify the lemma. We now assume that the lemma holds for every table T with j columns where $j \geq 2$. Consider now a table with c columns, where $c \geq 3$. We now show that the StreamTable $\mathcal{R}$ computed by $\mathcal{G}$ coincides with an optimal StreamTable $\mathcal{R}^*$, i.e., with no splits and minimum excess area.

We first show that the first two streams of any optimal StreamTable $\mathcal{R}^*$ can be replaced with the corresponding streams of $\mathcal{R}$. To observe this first note that the stream for the first column must be drawn left-aligned, and since the rectangle heights are given, the right side of the streams $B(\mathcal{R}, 1)$ must coincide with $B(\mathcal{R}^*, 1)$. Consider now the left sides of the second streams. If $A(\mathcal{R}, 2)$ does not coincide with $A(\mathcal{R}^*, 2)$, then there must be non-zero area between them. Let A be an orthogonal polygonal chain constructed by taking the left envelope of these two chains. In other words, for each row, we choose the part of the chain that have the minimum x-coordinate. Since the streams for $\mathcal{R}$ and $\mathcal{R}^*$ do not contain any split, the stream determined by A is a no-split stream. Since the sum of x-coordinates is smaller for A, the polygonal chain $A(\mathcal{R}, 2)$ must coincide with A. Thus the right side of the stream, i.e., the polygonal chain $B(\mathcal{R}, 2)$, must remain to the left of $B(\mathcal{R}^*, 2)$.

Consider now an $r \times (c-1)$ table T' which is obtained by treating the polygonal chain $B(\mathcal{R}, 2)$ as $B(\mathcal{R}, 1)$. By induction, $\mathcal{G}$ provides a StreamTable $\mathcal{R}'$ with no splits and minimum excess area. We can obtain the StreamTable $\mathcal{R}$ by replacing the first stream of $\mathcal{R}'$ with the two streams constructed using Steps 1 and 2 of $\mathcal{G}$. If there exists an optimal StreamTable $\mathcal{R}^*$ with a smaller excess area than that of $\mathcal{R}$, then there must be a StreamTable with a smaller excess area than that of $\mathcal{R}'$, a contradiction. ◀

We now show that Algorithm $\mathcal{G}$ can be implemented in $O(rc)$ time.

Implementation of Step 1. It is straightforward to compute the drawing of the first column in $O(r)$ time.

Implementation of Step 2. To compute **Step 2**, i.e., to draw the jth stream, we assume that the drawings of the previous streams are given. We compute two sequences of rectangles R_t and R_b, and then process them to find the stream for the jth column.

We construct R_t from top-to-bottom (e.g., see Figure 4(a)), and refer to a placement of the rectangle $R(i,j)$ in R_t as $R_t(i,j)$. We first place $R_t(1,j)$ starting at $b_{1,j-1}$, and then greedily place $R_t(i,j)$, where $i > 1$, such that each rectangle lies as much to the left as possible maintaining the adjacency with the previously placed rectangle. The adjacency between subsequent rectangles is broken when the $b_{i,j-1}$ is larger than the x-coordinate of the right side of the previously placed rectangle $R_t(i-1,j)$. We construct R_b from bottom-to-top symmetrically (Figure 4(b)). It is straightforward to compute R_t and R_b in $O(r)$ time. We now construct the stream of the jth column by taking for each i, the rectangle with the larger starting x-coordinate among $R_t(i,j)$ or $R_b(i,j)$ (breaking ties arbitrarily), which also takes $O(r)$ time. Intuitively, the choice of taking the rectangle with the maximum x-coordinate is to satisfy the no-split constraint. We now show that this construction results in no-split streams and minimizes the sum of x-coordinates $a_{i,j}$ (as required for Step 2). Figure 4(c) illustrates the resulting drawing of the jth stream.

Let $R(i,j)$ be a rectangle in $\mathcal{R}$. We call $R(i-1,j)$ *a parent of* $R(i,j)$ if $a_{i-1,j} = b_{i,j}$, as shown in Figure 4(c). Similarly, we call $R(i+1,j)$ *a parent of* $R(i,j)$ if $a_{i+1,j} = b_{i,j}$. By our construction, if a rectangle (in R_t or B_b) does not have a parent, then it must start exactly at the ending x-coordinate of the previous rectangle in the same row. We refer to such a rectangle as *root rectangle*.

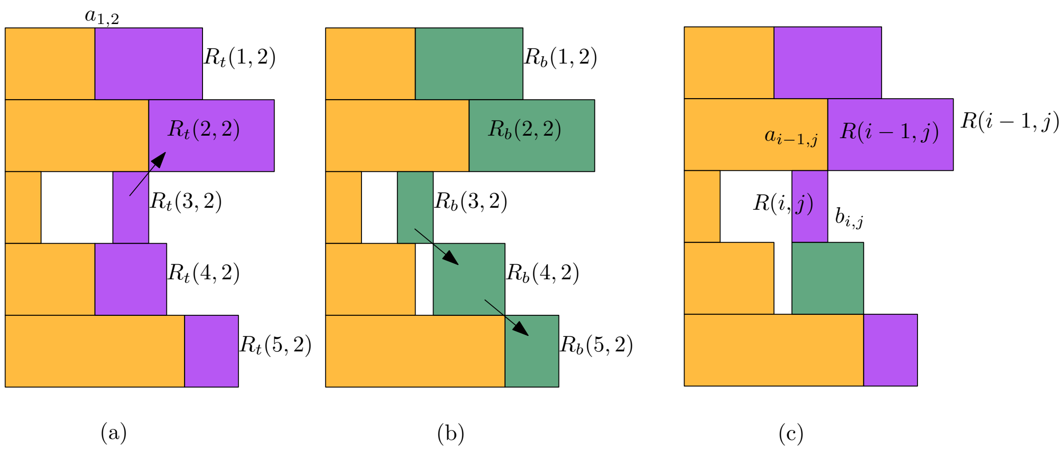


Figure 4 (a) The computation of R_t (purple). An arrow is drawn from a child to its parent. (b) The computation of R_b (green). (c) The drawing of the jth stream computed from R_t and R_b.

Let $A(\mathcal{R},j)$ be the left side of the jth stream. We now show that $A(\mathcal{R},j)$ determines a no-split stream and minimizes the sum of x-coordinates.

The constructed jth stream is a no-split stream: Suppose for a contradiction that $R(i-1,j)$ and $R(i,j)$ are not adjacent, and without loss of generality assume that $R(i-1,j)$ comes from R_t, i.e., $R(i-1,j) = R_t(i-1,j)$. The case when $R(i-1,j) = R_b(i-1,j)$ is symmetric considering a vertical flip. We now distinguish two cases depending on whether $R(i,j) = R_t(i,j)$ or $R(i,j) = R_b(i,j)$.

Case 1 ($R(i,j) = R_t(i,j)$): Since each rectangle in R_t lies as much to the left as possible maintaining the adjacency with the previously placed rectangle, $R(i-1,j)$ and $R(i,j)$ must be adjacent.

Case 2 $(R(i,j) = R_b(i,j))$**:** Since $R(i,j)$ and $R(i-1,j)$ are not adjacent, either the starting x-coordinate of $R_b(i,j)$ is larger than the right side of $R_t(i-1,j)$ (Figure 5(a)), or the ending x-coordinate of $R_b(i,j)$ is smaller than the left side of $R_t(i-1,j)$. In the former scenario, the starting x-coordinate of $R_b(i-1,j)$ must be larger than that of $R_t(i-1,j)$, which contradicts the assumption that $R(i-1,j)$ comes from R_t. In the latter scenario, the starting x-coordinate of $R_t(i,j)$ must be larger than that of $R_b(i,j)$, which contradicts the assumption that $R(i,j)$ comes from R_b.

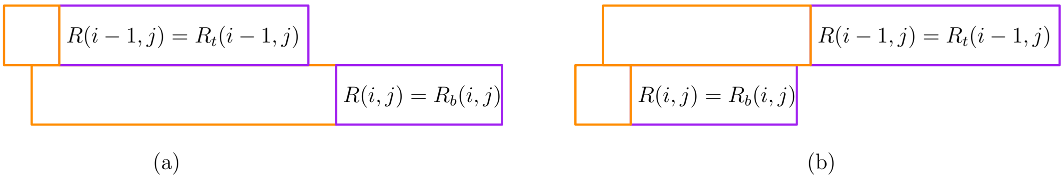


Figure 5 Illustration for Case 2.

The constructed jth stream minimizes the sum of $a_{i,j}$: Suppose for a contradiction that there exists a no-split drawing $\mathcal{R}'$ for the jth stream such that the sum of x-coordinates of $A(\mathcal{R}',j)$ is smaller than that of $A(\mathcal{R},j)$. Then there must exist a rectangle $R(i,j)$ in R such that the starting x-coordinate of $R(i,j)$ is larger than the corresponding rectangle $R'(i,j)$ in $\mathcal{R}'$. In the following we show that such a scenario cannot exist.

If $R(i,j)$ is a root rectangle, then the starting x-coordinate of $R(i,j)$ cannot be larger than that of $R(i,j)$. Hence we may assume that $R(i,j)$ has a parent. We follow the parent repeatedly until we reach a root $R(q,j)$. Without loss of generality assume that $q > i$. The case when $q < i$ is symmetric. Since $R(q,j)$ is a root rectangle, the starting x-coordinate of $R(q,j)$ cannot be larger than that of $R'(q,j)$. The ending x-coordinate of the child rectangle $R(q-1,j)$ is exactly the starting x-coordinate of $R(q,j)$. Therefore, the starting x-coordinate of $R(q-1,j)$ cannot be larger than that of $R'(q-1,j)$. By following this chain of constraints determined by the child relations, we observe that the starting x-coordinate of $R(i,j)$ cannot be larger than that of $R'(i,j)$. This contradicts our initial assumption that the starting x-coordinate of $R(i,j)$ is larger than that of $R'(i,j)$.

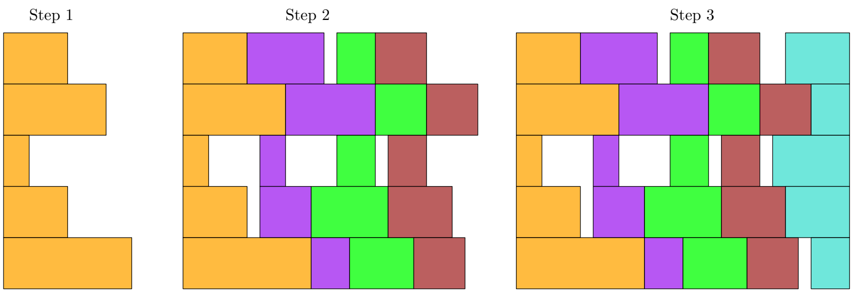


Figure 6 A simple run of the greedy algorithm, computing a no-split minimum excess area StreamTable with 5 rows and columns.

Implementation of Step 3. It is straightforward to compute **Step 3** by first following **Step 2** and then moving the rectangles rightward to make them right aligned (Figure 6).

The following theorem summarizes the result of this section.

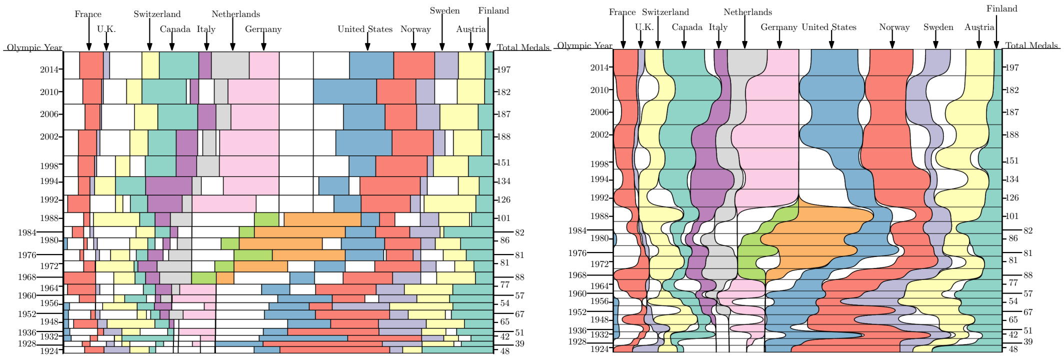


Figure 7 StreamTables of a Winter Olympics dataset (left) using a linear program with row height proportional to the row sum, and (right) using Gurobi with a fixed total height and with corner smoothing.

▶ **Theorem 2.** *Given an $r \times c$ table T and a height for each row, a StreamTable $\mathcal{R}$ for T with no splits and minimum excess area can be computed in $O(rc)$ time such that $\mathcal{R}$ respects the row ordering of T.*

2.2 Variable row heights

We now show how to formulate a system of linear equations to compute a StreamTable for T with no splits and minimum excess area such that the height of the ith row is set to h_i, where $1 \le i \le r$. Here $h_1, \dots, h_r$ are fixed constants. Let $d_{i,j}$ be a variable to model the adjacency between $R_{i,j}$ and $R_{i+1,j}$, where $1 \le i \le r-1$ and $1 \le j \le c$. We minimize the excess area: $\sum_{j=1}^{r} \sum_{k=1}^{c-1} h_j(a_{j,k+1} - b_{j,k})$, subject to the following constraints.

1. $a_{j,1} = a_{j+1,1}$ and $b_{j,c} = b_{j+1,c}$, where $j = 1, \dots, r-1$. This ensures StreamTable property P_1.
2. $b_{j,k} - a_{j,k} = (w_{j,k}/h_j)$, where $j = 1, \dots, r$ and $k = 1, \dots, c$. This ensures property P_2.
3. $a_{j,k} \le d_{j,k} \le b_{j,k}$ and $a_{j+1,k} \le d_{j,k} \le b_{j+1,k}$, where $1 \le j \le r-1$ and $1 \le k \le c$. This ensures that there are no splits in the streams.

Since $h_1, \dots, h_r$ are fixed constants, the above system with the constraint that the variables must be non-negative can be modeled as a linear program, e.g., see Figure 7 (left).

We now consider the case when $h_1, \dots, h_j$ are variables. Here the objective and constraint functions yield a quadratically-constrained quadratic program. Note that scaling down the height of a StreamTable by some $\delta \in (0, 1]$ and scaling up the width by $1/\delta$ do not change the excess area. Therefore, a non-linear program solver may end up generating a final StreamTable with bad aspect ratio. Hence we suggest to add another constraint: $h_1 + \ldots + h_k = H$, where H is the desired height of the visualization. Figure 7 (right) shows an example (not necessarily optimal) solution computed using a non-linear program solver Gurobi [13].

2.2.1 Local improvement

We now show how a non-optimal StreamTable may be improved further by examining each empty cell individually, while deciding whether that cell can be removed by shrinking the height of the corresponding row. By $E_{i,j}$ we denote the empty rectangle between the rectangles $R_{i,j}$ and $R_{i,j+1}$. We first refer the reader to Figure 8(a)–(b). Assume that we want to decide whether the empty cell $E_{i,j}(= E_{2,4})$ can be removed by scaling down the height of the second row. The idea is to grow the rectangles to the left (resp., right) of $E_{i,j}$ towards the right (resp., left) respecting the adjacencies and area.

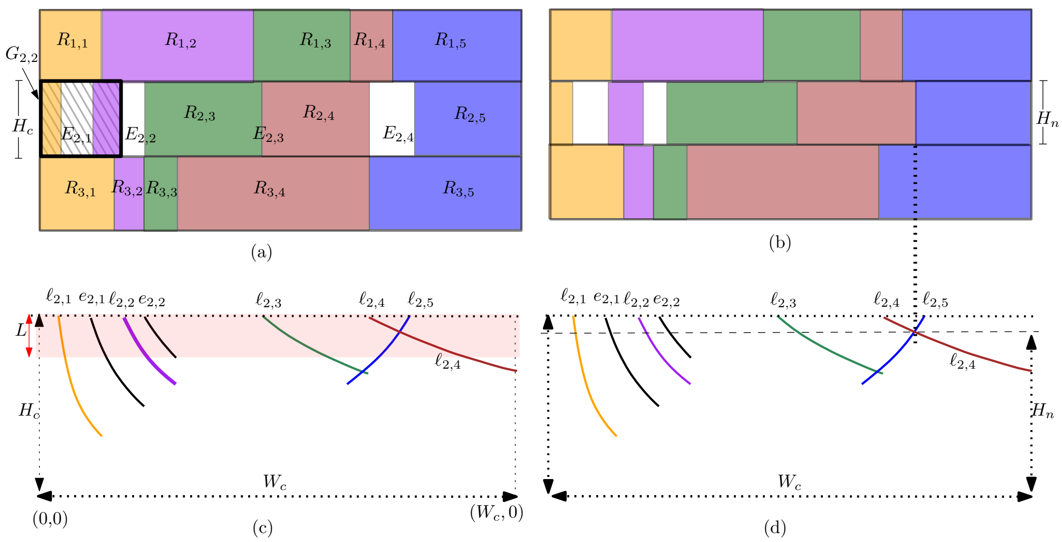


Figure 8 (a) A StreamTable with width W_c and height H_c. (b) Removal of the empty rectangle $E_{2,4}$ (c)–(d) Illustration for computing the new height H_n of the second row.

Now consider a rectangle $R_{i,k}$ before $E_{i,k}$ (e.g., $R_{2,2}$ before $E_{2,4}$ in Figure 8(a)). Let $G_{i,k}$ be the rectangle determined by the ith row with left and right sides coinciding with the left and right sides of $R_{i,1}$ and $R_{i,k}$, respectively. Figure 8(a) shows $G_{2,2}$ using a rectangle with thick boundary and falling pattern. Let $\ell_{i,k}$ be the length of $G_{i,k}$. Let $A_{i,k}$ be the initial area of $G_{i,k}$, and our goal is to keep this area fixed as we scale down the height of the ith row. The height of $G_{i,k}$ is defined by $f(\ell_{i,k}) = A_{i,k}/\ell_{i,k}$. Since the rectangles of the $(i-1)$th and $(i+1)$th rows do not move, $f(\ell_{i,k})$ does not split the $(k+1)$th stream as long as $\ell_{i,k}$ is upper bounded by the right sides of $R_{i-1,k+1}$ and $R_{i+1,k+1}$. Figure 8(c) plots these functions, where H_c is the current height of the second row. The height function for $G_{2,2}$ is drawn in thick purple in the interval $[\ell_{2,2}, \min\{b_{1,3}, b_{3,3}\}]$, where $b_{1,3}$ and $b_{3,3}$ are the right sides of $R_{1,3}$ and $R_{3,3}$, respectively.

We construct such functions also for all the empty rectangles $E_{i,k}$, where $1 \le k < j$. These are labeled with $e_{i,k}$. Finally, we construct these functions symmetrically for the rectangles that appear after $E_{i,j}$. Let L be an interval corresponding to the intersection of the projections of all the plots determined by these height functions on a vertical line (Figure 8(c)). Let Ξ be a horizontal slab determined by the L, which is shaded in red. We then compute the amount by which the ith row can be shrunk by determining the topmost intersection (if any) in Ξ, as illustrated in Figure 8(d). If no such intersection point exists, then we can shrink the row by an amount equal to the length of the interval L.

We iterate over the empty rectangles as long as we can find an empty rectangle to improve the solution, or to a maximum number of iterations. However, this only provides a heuristic algorithm, and thus Question 1 remains open.

2.2.2 Restricting row heights to integers

We now show that if the row heights are restricted to be positive integers, then finding a minimum-area no-split StreamTable respecting a given height H is NP-hard.

▶ **Theorem 3.** *Given a table T and a positive integer H, it is NP-hard to compute a minimum-area no-split StreamTable of height H with row heights as integers respecting the row ordering of T.*

Proof. We reduce the NP-hard problem CLIQUE [12], where the input is a graph G and a positive integer k and the goal is to find a *clique of size* k, i.e, a set of k vertices that are pairwise adjacent. The problem remains NP-hard even when $1 < k < n$. Given an instance G of the CLIQUE problem with n vertices and m edges, we construct a table T with n rows and m columns as follows.

1. For each edge $e \in E_G$, we create a column called an *edge column*, and label it by e.
2. We insert an additional column at the left and right sides of the table and also between every pair of adjacent columns. We refer to these columns as *line columns*. Each cell of a line column has a weight of $\epsilon = \frac{1}{n(m+1)}$.
3. For each vertex $v \in V_G$, we create a row and assign it the label v.
4. We now partition each cell $T_{v,e}$ into two cells (Figure 9), as follows.
   a. If vertex v is an end point of edge e, then the weight of the left and right cells are 2 and 4, respectively. We refer to these as a $(2,4)$-*group*.
   b. Otherwise, the weight of the left and right cells are 2 and 2, respectively. We refer to these as a $(2,2)$-*group*.

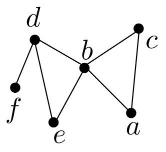
It now suffices to show that G admits a clique of size k if and only if there exists a no-split StreamTable of height $H = (n+k)$ and width at most $(6m - 2\binom{k}{2}) + (m+1)\epsilon$, where the row heights are integers.

Assume first that G has a clique C of size k. We then draw the StreamTable such that each row corresponding to the vertices of C has a height of 2 and every other row has a height of 1 (Figure 9). For each edge column $e = (v, w)$, there can now be two cases: (A) If $v \in C$ and $w \in C$, then $T_{v,e}$ and $T_{w,e}$ will be $(2,4)$ groups and all the other cells in this edge column are $(2,2)$ groups. Hence this edge column can be drawn with a width of 4 units. (B) If at least one of v and w are not in C, then without loss of generality assume $w \notin C$. Since $T_{w,e}$ is a $(2,4)$-group and the row height for w is 1, we can draw the cells in $T_{w,e}$ using a width of 6 units. It is straightforward to draw the remaining cells of this edge column within the same width such that we obtain a no-split stream.

Since we have k mutually adjacent vertices, we will have $\binom{k}{2}$ edge columns with width 4 and $(m - \binom{k}{2})$ edge columns with width 6. Thus the total width is at most $6(m - \binom{k}{2}) + 4\binom{k}{2} = (6m - 2\binom{k}{2})$. Together with the line columns the width is at most $(6m - 2\binom{k}{2}) + (m+1)\epsilon$.

Assume now that there exists a StreamTable for T with height $(n+k)$ and width $(6m - 2\binom{k}{2}) + (m+1)\epsilon$, where all the row heights are integers.

Since the height of T is $(n+k)$, the number of rows with height 2 or more is at most k. Let C' be the set that consists of the vertices corresponding to these rows. We now establish a lower bound on the width of a cell. Let $e = (v, w)$ be an edge in G. Assume that C' includes both v and w. Since $k < n$, there must be at least one row of height 1. Let z be the corresponding vertex. The width required for the corresponding cell $T_{z,e}$ is at least 4 units. If C' does not include both v and w, then without loss of generality assume that $w \notin C'$. Since the row corresponding to w is of height 1, the width required for the corresponding cell $T_{w,e}$ is at least 6 units.



	ac	ab	de	bc	db	be	df
a	(2, 4)	(2, 4)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)
b	(2, 2)	(2, 4)	(2, 2)	(2, 4)	(2, 4)	(2, 4)	(2, 2)
c	(2, 4)	(2, 2)	(2, 2)	(2, 4)	(2, 2)	(2, 2)	(2, 2)
d	(2, 2)	(2, 2)	(2, 4)	(2, 2)	(2, 4)	(2, 2)	(2, 4)
e	(2, 2)	(2, 2)	(2, 4)	(2, 2)	(2, 2)	(2, 4)	(2, 2)
f	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 4)

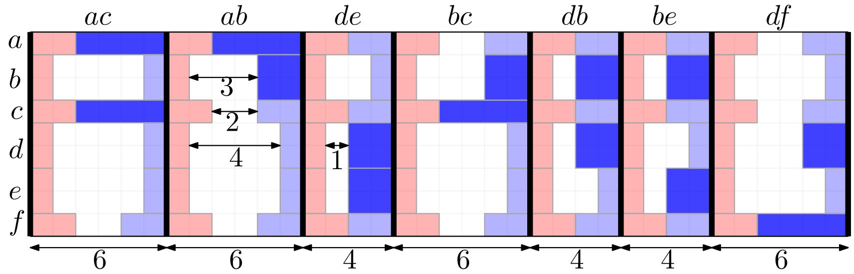


Figure 9 Illustration for the proof of Theorem 3. Given a clique $\{b, e, d\}$, one can construct a StremTable with $\binom{3}{2}$ edge columns of width 4, where the remaining edge columns are of width 6. The line columns are shown in thick vertical lines.

We now show that the vertices in C' are mutually adjacent in G. If C' does not correspond to a clique, then the total width of the table must be at least $6(m - m') + 4m' - n(m+1)\epsilon$, where m' is the number of edges of G with both endpoints in C' and the negative term $n(m+1)\epsilon$ compensate for the shift that may occur due to the line columns. For example, consider the sequence of rectangles corresponding to a line column. The rectangle at the first row allows the rectangle on the second row to move to the left or right by at most ϵ. Therefore, the stream corresponding to a line column may span a horizontal interval of size $n\epsilon$, and for the $(m+1)$ line columns, this shift is bounded by $n(m+1)\epsilon$.

Since $\epsilon = \frac{1}{n(m+1)}$, the width is at least $6(m-m')+4m'-1 = 6m-2m'-1$. If C' does not correspond to a clique, then m' is smaller than $\binom{k}{2}$. Therefore, $6m - 2m' - 1 \geq 6m - 2(\binom{k}{2} - 1) - 1 = 6m - 2\binom{k}{2} + 1$. Since $(m+1)\epsilon = \frac{1}{n}$, we have $6m - 2\binom{k}{2} + 1 > 6m - 2\binom{k}{2} + (m+1)\epsilon$. We thus reach a contradiction to our initial assumption on the width of T. ◀

3 StreamTable with uniform row heights and variable row order

In this section we show that computing StreamTables with no splits (resp., minimum excess area) while minimizing the excess area (resp., number of splits) by reordering the rows is NP-hard.

3.1 NP-hardness — no split, minimum excess area

▶ **Theorem 4.** *Given a table T and a non-zero positive number $\delta > 0$, it is NP-hard to compute a StreamTable with no splits and minimum excess area, where each row is of height δ and the ordering of the rows can be chosen.*

Proof. We reduce the NP-complete problem Betweenness [18], where the input is a set of ordered triples over r elements, and the problem is to decide whether there exists a total order σ of these r elements, with the property that for each given triple, the middle element in the triple appears somewhere in σ between the other two elements.

Let S be a set of c integer triples (an instance of BETWEENNESS) over r elements (integers), where $r, c \geq 5$. We now construct an $r \times (4c+1)$ table T (Figure 10(a)), as follows:

1. For every triple $t \in S$, we make a column (labeled with t). We refer to these columns as *triple columns*. Each of these columns will later be split into three more columns. For every element e, we create a row (labeled with e).
2. We insert an additional column at the left and right sides of the table and also between every pair of adjacent triple columns. We refer to these columns as *line columns*. Each cell of a line column has a weight of $\epsilon = \frac{1}{r(c+1)}$.
3. For every triple t and row i, we further partition the cell (t, i) into three cells and assign weights based on a parameter w to be chosen later, as follows:
   a. If i is the left element of t, then the weight of the left, middle and right cells are $\frac{2w}{3}$, $\frac{w}{6}$ and $\frac{w}{6}$, respectively.
   b. If i is the right element of t, then the weight of the left, middle and right cells are $\frac{w}{6}$, $\frac{w}{6}$ and $\frac{2w}{3}$, respectively.
   c. If i is the center element of t, then the weight of the left, middle and right cells are $\frac{w}{6}$, $\frac{2w}{3}$ and $\frac{w}{6}$, respectively.
   d. Finally, if i does not belong to t, then the weight of the left, middle and right cells are $\frac{5w}{12}$, ϵ and $\frac{5w}{12}$, respectively.

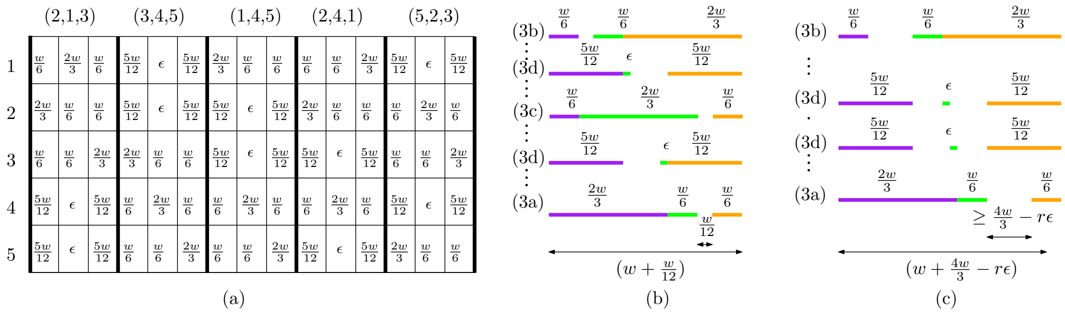


Figure 10 (a) A table T obtained from a set of triples $\{(2,1,3), (3,4,5), (1,4,5), (2,4,1), (5,2,3)\}$. Here the thick black lines represent the line columns. (b)–(c) Illustration for the required width for different row orderings.

We set δ, i.e., the height of each row, to be 1. It now suffices to show that the BETWEENNESS instance S admits a total order σ, if and only if there exists a StreamTable with no splits and at most $\frac{kw}{12} + (rc - k)(\frac{w}{4} - \epsilon)$ excess area, where k is the number of cells satisfying (3a), (3b) or (3c).

Assume first that S admits a total order σ and we show how to compute the required StreamTable. We draw the rectangles of each line column on top of each other (vertically aligned) and allocate a width of $(w + \frac{w}{12})$ for the triple columns. We now show how to complete the drawing of the rectangles of the three streams corresponding to a triple column within the allocated width without any split.

If we draw the left (right) rectangle in each cell as much to the left (right) as possible within the allocated width for the cell, then we obtain two no-split streams. We now show that the middle rectangles of the cells can be drawn to obtain a no-split stream. If a cell satisfies conditions (3a) or (3b), then we make its middle rectangle adjacent to the largest among the other two rectangles. Since we order the rows by σ, no pair of cells in a triple column where one satisfies (3a) and the other satisfies (3b) can be adjacent.

If a cell satisfies condition (3c) or (3d), then we will leverage the ordering σ to show that the middle rectangle can always be drawn so that it touches the middle rectangles of the (upper and lower) adjacent cells. In fact, for a cell that satisfies (3c), its middle rectangle is large enough to satisfy the required adjacencies. Figure 10(b) illustrates such adjacencies using a schematic representation.

For a cell q that satisfies (3d), if one of its adjacent cells x satisfies (3a) and the other cell z satisfies (3b), then we do not have a position for the middle rectangle of q that would make it adjacent to the other middle rectangles of x and z. However, since we order the rows by σ, we must have a cell y' that satisfies (3c), i.e., q cannot be adjacent to both x and z. In all other cases, finding a position for the middle rectangle of q is straightforward so that it touches the middle rectangles of the adjacent cells.

The excess area for each cell satisfying (3a), (3b) or (3c) is $\frac{w}{12}$, and for each cell satisfying (3d) is $(\frac{w}{12} + (w - \frac{5w}{6} - \epsilon)) = (\frac{w}{12} + \frac{w}{6} - \epsilon) = \frac{w}{4} - \epsilon$. Therefore, the total excess area is $\frac{kw}{12} + (rc - k)(\frac{w}{4} - \epsilon)$, where k is the number of cells satisfying (3a), (3b) or (3c). Figure 11 illustrates the construction for the table from Figure 10(a).

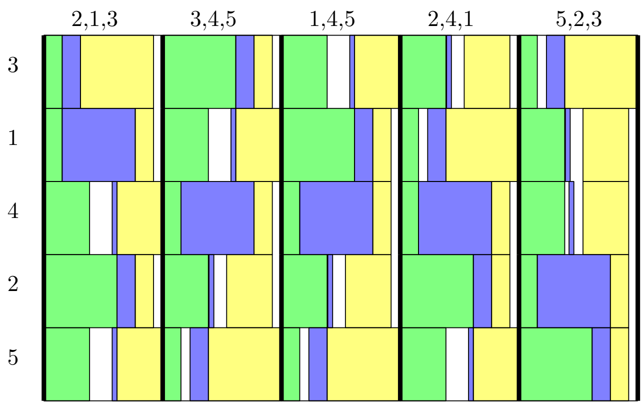


Figure 11 A StreamTable for the T, where $\sigma = \{3, 1, 4, 2, 5\}$.

We now show that if there is a StreamTable for T with at most $\frac{kw}{12} + (rc - k)(\frac{w}{4} - \epsilon)$ excess area, then the corresponding row ordering will yield the total order for the BETWEENNESS instance. Suppose for a contradiction that for some triple $t = (x, y, z)$, the cell y satisfying $(3c)$ does not appear between the cells satisfying $(3a)$ and $(3b)$ (Figure 10(c)). Without loss of generality assume that x satisfies (3a) and z satisfies (3b). We now show that there exists a cell satisfying (3d) with at least $(\frac{w}{3} - r(c+2)\epsilon)$ excess area. Note that the left rectangle of x and the right rectangle of z each has a width of $\frac{2w}{3}$.

- If x and z are consecutive in the triple column, then they must require a width of $\frac{4w}{3}$, and thus at least $(\frac{w}{3} - r(c+1)\epsilon)$ excess area for sufficiently small ϵ. Since each line column allows a cell to move by at most $r\epsilon$, the $r(c+1)\epsilon$ term accommodates the potential contribution of the $(c+1)$ line columns. Consequently, there must be a cell satisfying (3d) in this triple column with at least $(\frac{w}{3} - r(c+1)\epsilon) - r\epsilon = (\frac{w}{3} - r(c+2)\epsilon)$ excess area.
- If x and z are not consecutive, then they may have one or more cells in between that satisfy (3d). Therefore, the middle rectangles of weight ϵ can help the right rectangle of z to start at most $r\epsilon$ units earlier. Hence, the width of the cell is at least $(\frac{4w}{3} - r\epsilon - r(c+1)\epsilon)$. Here the term $r\epsilon$ corresponds to the contribution of the middle rectangles of the cells, and the $r(c+1)\epsilon$ term accommodates the potential contribution of the $(c+1)$ line columns. Thus the excess area is at least $(\frac{4w}{3} - r\epsilon - r(c+2)\epsilon) - \frac{5w}{6} > (\frac{w}{3} - r(c+2)\epsilon)$.

For every triple t, exactly one cell satisfies ($3a$), one cell satisfies ($3b$), and one cell satisfies ($3c$). Since we assumed $r \geq 5$, there exists a cell q in t that satisfies ($3d$) and is adjacent to a cell satisfying either (3a) or (3b). Without loss of generality assume that q is adjacent to a cell p where p satisfies (3a). The left rectangle of p and the right rectangle of q are of size $\frac{2w}{3}$ and $\frac{5w}{12}$, respectively. Therefore, the width of p is at least $(\frac{2w}{3} + \frac{5w}{12} - r(c+1)\epsilon)$, and the excess area is at least $(\frac{w}{12} - r(c+1)\epsilon)$ for the cells with weight w and at least $(\frac{w}{4} - \epsilon - r(c+1)\epsilon)$ for the cells satisfying (3d).

We now compute the total excess area. If there is a triple $t = (x, y, z)$ such that the cell y satisfying ($3c$) does not appear between the cells satisfying ($3a$) and ($3b$), then there is a cell satisfying (3d) with an excess area of $(\frac{w}{3} - r(c+2)\epsilon)$. Every other cell in this triple column is forced to have an excess area of at least $(\frac{w}{3} - r(c+3)\epsilon) \geq (\frac{w}{12} - r(c+2)\epsilon)$. Similarly, if the cell y satisfying ($3c$) appears between the cells satisfying ($3a$) and ($3b$), then every cell of weight w in this column has an excess area of at least $(\frac{w}{12} - r(c+2)\epsilon)$ and every cell of weight $(\frac{5w}{12} - \epsilon)$ has an excess area of at least $(\frac{w}{4} - \epsilon - r(c+1)\epsilon)$. We set $\epsilon = \frac{1}{kr(c+2)}$. Now the total excess area is at least

$$
\begin{aligned}
&(\frac{w}{3} - r(c+2)\epsilon) + k(\frac{w}{12} - r(c+2)\epsilon) + (rc - 1 - k)(\frac{w}{4} - r(c+2)\epsilon) \\
&> (\frac{w}{3} - 1) + (\frac{kw}{12} - 1) + (rc - k)(\frac{w}{4} - 1) - (\frac{w}{4} - 1) \\
&> \frac{w}{3} + \frac{kw}{12} - 1 + (rc - k)(\frac{w}{4} - 1) - \frac{w}{12} \\
&= \frac{w}{4} + \frac{kw}{12} + (rc - k)\frac{w}{4} - (rc - k) - 1.
\end{aligned}
$$

For $w > 4(rc-k) - 4(rc-k)\epsilon + 4$, this implies an excess area larger than $\frac{kw}{12} + (rc-k)(\frac{w}{4} - \epsilon)$, a contradiction. ◀

3.2 NP-hardness — minimum number of splits, zero excess area

▶ **Theorem 5.** *Given a table T and a non-zero positive number $\delta > 0$, it is NP-hard to compute a StreamTable with zero excess area and minimum number of splits, where each row is of height δ and the ordering of the rows can be chosen.*

Proof. We reduce the NP-complete problem HAMILTONIAN PATH in a cubic graph [11], where the input is a graph G with n vertices and m edges such that every vertex is of degree 3, and the problem is to decide whether there exists a total order of the vertices that determines a Hamiltonian path, i.e., a simple path of size n, in G.

Let G be a graph with n vertices, i.e., an instance of the HAMILTONIAN PATH problem. We now construct a table T, as follows.

1. For each edge $e \in E_G$, we create a column called an *edge column*, and label it by e.
2. For each vertex $v \in V_G$, we create a row and assign it the name v. For each edge column, we assign each cell a weight w.
3. We now partition each cell $T_{v,e}$ into three cells (Figure 12(a)–(b)), as follows.
   a. If vertex v is an end point of edge e, then the weight of the left, middle and right cells are $\frac{7}{12}w$, $\frac{1}{12}w$, and $\frac{4}{12}w$, respectively. We refer to these as an *L group*.
   b. Otherwise, the weight of the left, middle and right cells are $\frac{4}{12}w$, $\frac{1}{12}w$, and $\frac{7}{12}w$, respectively. We refer to these as an *R group*.

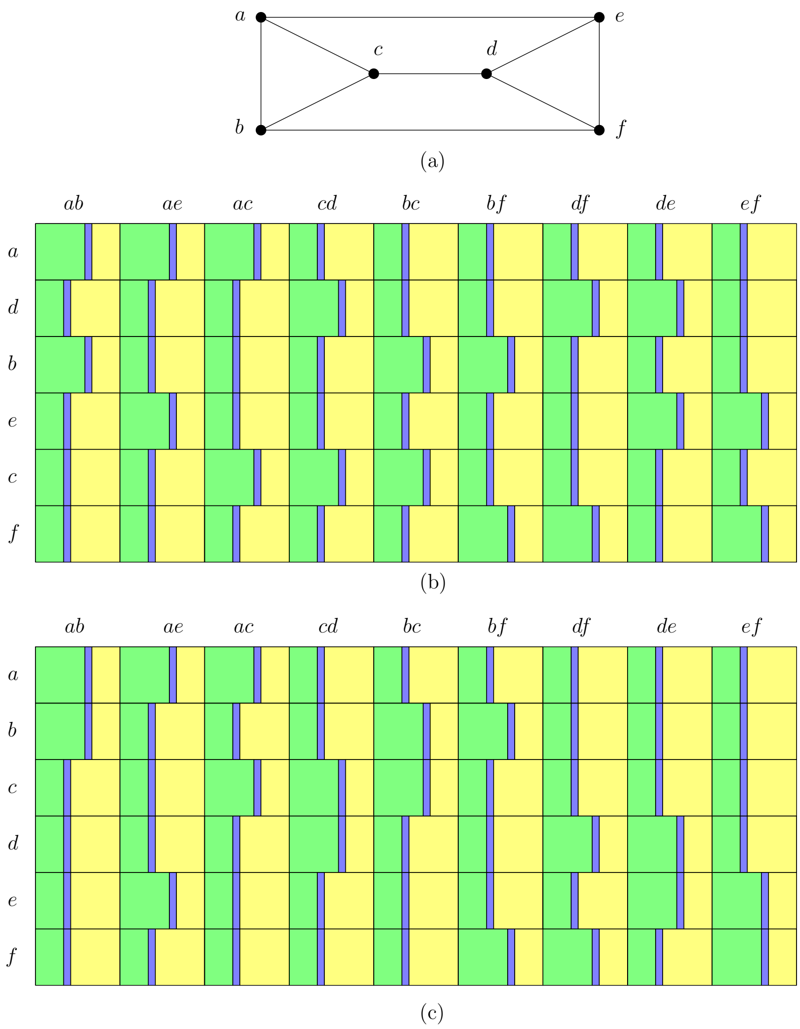


Figure 12 (a) A cubic graph G. (b) A visual representation of the table T corresponding to G. (c) Construction of a StreamTable from a given Hamiltonian path a, b, c, d, e, f.

It now suffices to show that G admits a Hamiltonian path if and only if there exists a StreamTable for T with zero excess area and at most $4(n-1)$ splits, where the height of each row is $\delta = 1$.

Assume first that G has a Hamiltonian path P. We then draw the StreamTable such that each row has a height of δ, each row is drawn in the order of the Hamiltonian path, and the cells within each row are drawn consecutively without leaving any gap in between (Figure 12(c)). By construction of the StreamTable, for every pair of vertices that are adjacent in P, the corresponding rows will be consecutive in the StreamTable. Consider a pair of consecutive vertices v, z on P. For each edge column e, there can now be three cases: (A) If $e = (v, z)$, then $T_{v,e}$ and $T_{z,e}$ will consist of L groups and hence no splits will appear. (B) If neither v nor z is an endpoint of e, then $T_{v,e}$ and $T_{z,e}$ will consist of R groups and hence no splits will appear. (C) Otherwise, one of $T_{v,e}$ and $T_{z,e}$ will represent an L group and the other is an R group, and hence a split will appear. Since G is cubic, there are exactly two edges incident to v and two edges incident to w (other than the edge (v, w)) that can generate splits. Therefore, the number of splits contributed by the rows representing v and

w is 4. Thus the number of splits overall is $4(n-1)$.

Assume now that there exists a StreamTable for T with zero excess area and at most $4(n-1)$ splits, where each row is of height δ. Note that the height of each row is 1 and the cells in an edge column are of weight w. Since there is no excess area, the edge columns must be drawn inside a vertical slab, i.e., a region bounded by two vertical lines. Therefore, a pair of adjacent L and R groups will generate a split. We now show that the row ordering in the StreamTable determines a Hamiltonian path in G. Suppose for a contradiction that a consecutive pair of rows exists in the StreamTable where the corresponding vertices are not adjacent in G. Let a, b be such a pair of vertices. Then every L group of a will occupy a horizontal interval shared with an R group of b, and vice versa. Hence this would contribute to at least 6 splits. Since any pair of consecutive rows must contribute to at least 4 splits, the number of splits will be at least $6 + 4(n-2) = 4n - 2 > 4(n-1)$. ◀

4 Conclusion

In this paper we have introduced StreamTable, which is an area proportional visualization inspired by streamgraphs. We formulated algorithmic problems that need to be tackled to produce aesthetic StreamTables and examined two aesthetic criteria – excess area and number of splits.

We have shown that if row heights and row ordering are given, then a StreamTable with no splits and minimum area can be computed via a linear program. However, the case when the row ordering is given but the row heights can be chosen needs further investigation. We only provided a quadratically constrained quadratic program to model the problem and an NP-hardness proof when the row heights are constrained to be integers. A future research direction can be to conduct a full-fledged experimental analysis of the proposed approaches alongside innovating other heuristics to gain an understanding of the scalability and performance trade-offs of these techniques. However, the original question remains open.

Open Problem 1: Given a table T and a positive integer H, does there exist a polynomial-time algorithm to compute a minimum-area no-split StreamTable of height H that respects the row ordering of T?

We also showed that if the row ordering can be chosen, then the problem of finding a minimum-area or a minimum-split StreamTable is NP-hard. In this setting, it would be interesting to find algorithms for computing zero-excess-area (resp., no splits) StreamTables with good approximation on the number of splits (resp., excess area).

Open Problem 2: Design polynomial-time algorithms to find good approximation for StreamTable aesthetics (excess area or number of splits) in both the fixed and variable row ordering settings.

Similar aestheic criteria can be considered even beyond tabular data such as in area propoortional circle packing [2] or community visualization in large networks [17]. Recently a framework for $\exists\mathbb{R}$-completeness of packing problems has been proposed in [1]. It would be interesting to investigate $\exists\mathbb{R}$-completeness in this context, where the rows need to be packed inside a rectangle maintaining column adjacencies.

Acknowledgements

We thank anonymous reviewers for insightful feedback that helped streamline the proofs and improve the presentation.

References

1 Mikkel Abrahamsen, Tillmann Miltzow, and Nadja Seiferth. Framework for ER-completeness of two-dimensional packing problems. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1014–1021. IEEE, 2020. doi:10.1109/FOCS46700.2020.00098.

2 Md. Jawaherul Alam, David Eppstein, Michael T. Goodrich, Stephen G. Kobourov, and Sergey Pupyrev. Balanced circle packings for planar graphs. In *Proceddings of the 22nd International Symposium on Graph Drawing (GD)*, volume 8871 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 2014. doi:10.1007/978-3-662-45803-7_11.

3 Marco Di Bartolomeo and Yifan Hu. There is more to streamgraphs than movies: Better aesthetics via ordering and lassoing. *Comput. Graph. Forum*, 35(3):341–350, 2016. doi:10.5555/3071534.3071571.

4 Kevin Buchin, David Eppstein, Maarten Löffler, Martin Nöllenburg, and Rodrigo I. Silveira. Adjacency-preserving spatial treemaps. *J. Comput. Geom.*, 7(1):100–122, 2016. doi:10.20382/jocg.v7i1a6.

5 Lee Byron and Martin Wattenberg. Stacked graphs - geometry & aesthetics. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1245–1252, 2008. doi:10.1109/TVCG.2008.166.

6 Temo Chen and Michael K. H. Fan. On convex formulation of the floorplan area minimization problem. In Majid Sarrafzadeh, editor, *Proceedings of the 1998 International Symposium on Physical Design (ISPD)*, pages 124–128. ACM, 1998. doi:10.1145/274535.274553.

7 David Eppstein, Elena Mumford, Bettina Speckmann, and Kevin Verbeek. Area-universal and constrained rectangular layouts. *SIAM J. Comput.*, 41(3):537–564, 2012. doi:10.1137/110834032.

8 David Eppstein, Marc J. van Kreveld, Bettina Speckmann, and Frank Staals. Improved grid map layout by point set matching. *Int. J. Comput. Geom. Appl.*, 25(2):101–122, 2015. doi:10.1109/PacificVis.2013.6596124.

9 Jared Espenant and Debajyoti Mondal. Streamtable: An area proportional visualization for tables with flowing streams. In *Proceedings of the 16th International Conference and Workshops on Algorithms and Computation (WALCOM)*, pages 97–108, 2022.

10 William S. Evans, Stefan Felsner, Michael Kaufmann, Stephen G. Kobourov, Debajyoti Mondal, Rahnuma Islam Nishat, and Kevin Verbeek. Table cartogram. *Comput. Geom.*, 68:174–185, 2018. doi:10.1016/j.comgeo.2017.06.010.

11 M. R. Garey, David S. Johnson, and Robert Endre Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5(4):704–714, 1976. doi:10.1137/0205049.

12 Michael R. Garey and David S. Johnson. *Computers and intractability : a guide to the theory of NP-completeness.* W. H. Freeman, San Francisco, 1979.

13 LLC Gurobi Optimization. Gurobi optimizer reference manual, 2020. URL: https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/9.1/refman.pdf.

14 Mohammad Rakib Hasan, Debajyoti Mondal Jarin Tasnim, and Kevin A. Schneider. Putting table cartograms into practice. In *Proceedings of the 16th International Symposium on Visual Computing (ISVC)*, volume 13017, pages 91–102. Springer, 2021. doi:10.1007/978-3-030-90439-5_8.

15 Susan Havre, Elizabeth G. Hetzler, Paul Whitney, and Lucy T. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. Vis. Comput. Graph.*, 8(1):9–20, 2002. doi:10.1109/2945.981848.

16 Erkki Mäkinen and Harri Siirtola. Reordering the reorderable matrix as an algorithmic problem. In Michael Anderson, Peter C.-H. Cheng, and Volker Haarslev, editors, *Proceedings of the First International Conference on Theory and Application of Diagrams*, volume 1889 of *LNCS*, pages 453–467. Springer, 2000. doi:10.1007/3-540-44590-0_37.

17 Ehsan Moradi and Debajyoti Mondal. Biggraphvis: Visualizing communities in big graphs leveraging gpu-accelerated streaming algorithms. In *Proceedings of the 18th International Joint*

Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP), pages 195–202. SCITEPRESS, 2023. doi:10.5220/0011783700003417.

18 Jaroslav Opatrny. Total ordering problem. *SIAM J. Comput.*, 8(1):111–114, 1979. doi:10.1137/0208008.

19 Eric Rosenberg. Optimal module sizing in VLSI floorplanning by nonlinear programming. *Zeitschrift für Operations Research*, 33:131–143, 1989. doi:10.1007/BF01415168.