Algorithmica (2019) 81:47–68
https://doi.org/10.1007/s00453-018-0432-7

On Almost Monge All Scores Matrices

Amir Carmel[1] · **Dekel Tsur**[1] ·
Michal Ziv-Ukelson[1]

Received: 29 October 2016 / Accepted: 19 March 2018 / Published online: 27 March 2018

Abstract The all scores matrix of a grid graph is a matrix containing the optimal scores of paths from every vertex on the first row of the graph to every vertex on its last row. This matrix is commonly used to solve diverse string comparison problems. All scores matrices have the Monge property, and this was exploited by previous works that used all scores matrices for solving various problems. In this paper, we study an extension of grid graphs that contain an additional set of edges, called bridges. Our main result is to show several properties of the all scores matrices of such graphs. We also apply these properties to obtain an $O(r(nm+n^2))$ time algorithm for constructing the all scores matrix of an $m \times n$ grid graph with r bridges and bounded integer weights.

Keywords Sequence alignment · Longest common subsequences · DIST matrices · Monge matrices · All path score computations · Multiple-source shortest-paths

1 Introduction

String comparison is a fundamental problem in computer science that has applications in computational biology, computer vision, and other areas. String comparison is often

A preliminary version of this work appeared in the proceeding of CPM 2016 conference.

✉ Amir Carmel
karmela@post.bgu.ac.il

Dekel Tsur
dekelts@cs.bgu.ac.il

Michal Ziv-Ukelson
michaluz@cs.bgu.ac.il

1 Department of Computer Science, Ben-Gurion University of the Negev, Beersheba, Israel

performed using *sequence alignment*: The characters of two input strings are aligned to each other, and a *scoring function* gives a score to the alignment according to pairs of the aligned characters and unaligned characters. The goal of the string alignment problem is to seek an alignment that maximizes a similarity score or minimizes a distance score. Common scoring functions are the *edit distance* score, and the *LCS* (longest common subsequence) score.

All scores matrices were introduced by Apostolico et al. [3] in order to obtain fast parallel algorithms for LCS computation. The *all scores matrix* of two strings A and B is a $(|B|+1) \times (|B|+1)$ matrix that stores the optimal alignment scores between A and every substring of B. More precisely, the element at row i and column j in the matrix is the optimal alignment score between A and $B[i..j]$. All scores matrices are also called DIST matrices [3] or semi-local score matrices [35].

The problem of efficiently constructing the all scores matrix of two strings has been studied in several papers [2,3,20,22–25,29,33–35]. All scores matrices provide a very powerful tool that can also be used for solving many problems on strings: optimal sequence alignment computation [11], approximate tandem repeats [27,33], approximate non-overlapping repeats [6,18,33], common substring alignment [26, 28], sparse spliced alignment [19,32], alignment of compressed strings [14], fully-incremental string comparison [17,35], and other problems.

The alignment problem on strings A and B can be represented by using an $(|A|+1)\times(|B|+1)$ grid graph, known as the *alignment graph* (cf. [33]). Vertical (respectively, horizontal) edges correspond to alignment of a character in A (respectively, B) with a gap, and diagonal edges correspond to alignment of two characters in A and B. A path from the j-th vertex on row i to the j'-th vertex on row i' corresponds to an alignment of $A[i..i']$ and $B[j..j']$. The all scores matrix is therefore a matrix that contains the maximum (or minimum) scores of paths from vertices on the first row of the alignment graph to the vertices on its last row.

For an $n \times n$ matrix D, its *density* matrix [35], denoted by $D^{\square}$, is an $(n-1)\times(n-1)$ matrix, where $D^{\square}[i, j] = D[i-1, j-1] + D[i, j] - D[i-1, j] - D[i, j-1]$. A matrix is called *Monge* if its density matrix is either non-negative or non-positive, and *unit Monge* if every row or column of the density matrix contains at most one non-zero element, and all the non-zero elements are equal to 1. All scores matrices of grid graphs are Monge matrices, this follows from the *crossing paths property* of the grid graph: If P_1 and P_2 are two paths from vertices on the first row to vertices on the last row of the graph, where on the first row the endpoint of P_1 appears before the endpoint of P_2, and on the last row the endpoint of P_1 appears after the endpoint of P_2, then the paths P_1 and P_2 must cross. This is illustrated in Fig. 1. An equivalent, and a more common definition of a Monge matrix, is the condition that either $\Delta^{i,j}_{i',j'} = D[i', j'] + D[i, j] - D[i', j] - D[i, j']$ is non-negative for every $i' \le i$ and $j' \le j$, or $\Delta^{i,j}_{i',j'}$ is non-positive for every $i' \le i$ and $j' \le j$. In the rest of the paper we will make use of both definitions. The Monge property is crucial for many of the algorithms for constructing all score matrices and for their applications. When the scoring function is the LCS score (namely, horizontal and vertical edges have weight 0, and diagonal edges have weight 1), the all scores matrix is unit Monge [34]. A detailed survey of LCS scoring can be found in [4].

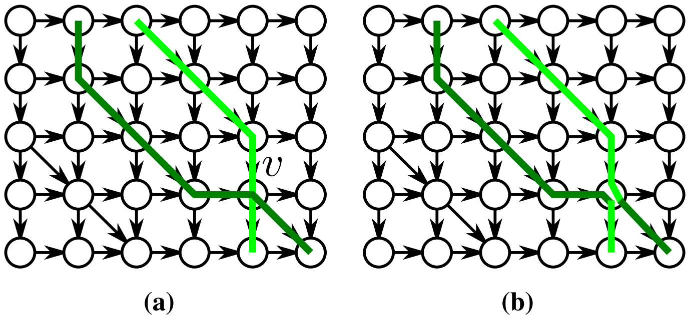


Fig. 1 The crossing paths property yielding the Monge property in grid graphs. In **a**, the dark green path is a maximum-score path from (0, 1) to (4, 5), and the light green path is a maximum-score path from (0, 2) to (4, 4). These two paths cross at the vertex v. **b** Shows that a path from (0, 1) to (4, 4) can be obtained by taking the prefix of the dark green path until v, and the suffix of the light green path from v. Similarly, a path from (0, 2) to (4, 5) can be obtained by taking the prefix of the light green path until v, and the suffix of the dark green path from v. The sum of scores of the new paths is equal to the sum of scores of the former paths, which is equal to $D[1, 5]+D[2, 4]$. Since the new paths are not necessarily of maximal score, we obtain that $D[1, 4] + D[2, 5] \geq D[1, 5] + D[2, 4]$, and thus, $D^{\square}[2, 5] = D[1, 4] + D[2, 5] - D[1, 5] - D[2, 4] \geq 0$ (Color figure online)

The Monge property plays an important role in combinatorial optimization [8]. For example, if the distance matrix of a travelling salesman problem fulfills the Monge property, then the problem can be solved in linear time [31]. Moreover, searching and selecting becomes particularly fast in Monge matrices. Aggarwal et al. [1] proposed a very efficient algorithm for computing all row minima of a Monge matrix. Recently, Hou and Prékopa [15] showed that Monge matrices can be used for bounding multivariate probability distribution functions and in that way opened a new area of applications for Monge matrices.

In this paper we extend the grid graphs that are used in sequence alignment to include an additional set of edges. These additional edges are of form $((i, j), (i', j'))$ where $i' \geq i$ and $j' \geq j$, and either $i' > i+1$ or $j' > j+1$ (see Fig. 2a). We call these edges *bridges*. The bridges represent correspondence between pairs of substrings, one per each input sequence, which could be precomputed using an auxiliary adviser. In grid graphs enhanced with bridges, the crossing paths property no longer holds, and so the all scores matrix does not necessarily have the Monge property (see Fig. 2).

Motivating examples of grid graphs enhanced with bridges are found in the domain of computational biology. Here, bridges are often used to incorporate additional information that is known about the function and the physical structure of the aligned biomolecules and of their components [7,13,30]. One such example is found in a problem denoted “sequence alignment guided by motifs”. Here, each one of the input sequences is first subjected to a parsing step in which meaningful substrings within it are identified and labeled. Substrings sharing the same label could be instantiations of the same motif shared by members of a protein family [16], particular DNA or RNA substrings of similar structure or function [5], or conserved molecular binding sites shared by multiple sequences that are combinatorially regulated in some biological pathway. Note that two substrings identified as belonging to the same motif family could be quite diverged in sequence, as it is the function, rather than the exact sequence, that is conserved in functional motifs. Yet, pairs of substrings sharing the same motif label are expected to be highly conserved in their location and order of occurrences within homologous genomic sequences. To incorporate this information, the alignment grid graph is enhanced with bridges reflecting pairs of substrings belonging to the same motif family, one from each sequence, and weights are assigned to these additional edges based on some a-priori scoring scheme expressing the importance of conserving the motifs in the alignment [5,10].

Our Contribution and Roadmap In this paper, we consider grid graphs with bridges, and we assume that the non-bridge edges have 0/1 weights. We note that grid graphs that incorporate a string alignment scoring table can be reduced to grid graphs with 0/1 weights [35], and thus we will only consider the 0/1 weights scheme. However, this reduction is quasi-polynomial: If the weights of non-bridge edges in the original grid graph are integers between $-C$ and C, then the reduction increases the size of graph by a factor of $\Theta(C^2)$.

Our main result is to show the following properties of the non-zero values in the density matrix of an all scores matrix of a grid graph with r bridges (see Fig. 2 for an example).

1. All the non-zero values in the density matrix are -1 or 1, except for $O(r^2)$ values.
2. In every row or column, except for r specific rows and r specific columns, the number of non-zero values is $O(r)$.

In particular, the number of non-zero values in the density matrix is $O(rn)$. Thus, if $r = o(n)$, the all scores matrix is "almost Monge". Property 1 is proven in Sect. 2 (Theorem 1). Property 2 is proven in Sect. 3 for the case of a single bridge. Then, in Sect. 5, the proof is extended to the general case. Finally, in Sect. 6 we show that the second property above is asymptotically tight by giving a construction of grid graphs whose density matrices contain $\Theta(n)$ rows and columns with $\Theta(r)$ non-zero values.

As a consequence of our main result, we obtain an algorithm for computing the all scores matrix of an $m \times n$ grid graphs with r bridges in $O(r(nm + n^2))$ time. This algorithm is based on Schmidt's algorithm [33] for grid graphs with no bridges, and utilizes the properties described above. See below for a comparison of this algorithm with previous results. The algorithm is given in Sect. 4 (Theorem 3).

Related Work Our algorithm mentioned above computes the optimal scores of paths from every vertex in a specific set of vertices (the vertices on the first row) to every vertex in the graph. This problem is called the *multiple source shortest paths* (MSSP) problem. Algorithms for solving MSSP were proposed by several previous works. Schmidt [33] gave an MSSP algorithm for grid graphs with general weights. This algorithm constructs the all scores matrix in $O((mn + n^2)\log n)$ time. For grid graphs with bounded integer weights, Schmidt gave an algorithm that constructs the all scores matrix in $O(mn)$ time. Tiskin [35] gave an MSSP algorithm for grid graphs with bounded integer weights that constructs the all scores matrix in $O(mn(\log\log n/\log n)^2)$ time. The results on grid graphs have been extended to general planar graphs. Klein [21] gave an algorithm for MSSP on planar graphs with general weights. The algorithm constructs the all scores matrix of a grid graph in $O((mn + n^2)\log(mn))$ time. Eisenstat and Klein [12] gave an algorithm for MSSP on undirected planar graphs with bounded integer weights, which constructs the all scores matrix of a grid graph in $O(mn + n^2)$. Cabello et al. [9] extended the result of Klein to graphs that can be embedded on a surface with genus g. Since a grid graph with r bridges can be embedded on a surface with genus r, the algorithm of Cabello et al. constructs the all scores matrix of a grid graph with r bridges and general weights in $O(rmn \log^2(mn) + n^2 \log(mn))$ time. Cabello et al. also gave a randomized algorithm whose running time is $O(r \log(mn)(mn + n^2))$ with high probability. Our algorithm improves upon both of the results of Cabello et al. and constructs the all scores matrix of a grid graph with r bridges in $O(r(mn + n^2))$ time for the case of grid graphs with bridges and bounded integer weights.

2 Preliminaries and Basic Problem Properties

A *grid graph with bridges* is a directed graph $G = (V, E)$ whose vertex set is $V = \{(i, j) : 0 \leq i \leq m, 0 \leq j \leq n\}$, and whose edge set consists of four types of edges:

1. Horizontal edges: $((i, j), (i, j + 1))$ for every pair of indices i, j satisfying $0 \leq i \leq m$ and $0 \leq j < n$.
2. Vertical edges: $((i, j), (i+1, j))$ for every pair of indices i, j satisfying $0 \leq i < m$ and $0 \leq j \leq n$.
3. Diagonal edges: Edges of the form $((i, j), (i + 1, j + 1))$.
4. Bridges: Edges of the form $((i, j), (i', j'))$ where $i \leq i'$ and $j \leq j'$, and either $i + 1 < i'$ or $j + 1 < j'$.

In our framework, the horizontal and vertical edges have weight 0, the diagonal edges have weight 1, and each bridge has a positive integer weight. The *score* of a path is the sum of the weights of its edges. The 0/1 weights of the non-bridge edges correspond to the LCS scoring scheme for the Sequence Alignment problem.

Let G be a grid graph with bridges $f_1, \ldots, f_r$. For a path P in G, we say that P is an *s-path*, if f_s is the first bridge that P passes through. If P does not pass through bridges, we say that P is a *0-path*. Note that we focus on the first bridge of path P in order to obtain a variant of the crossing paths property which will be given in Lemma 7.

We define matrices D, $D^{\square}$, and D_{first} as follows (see Fig. 2).

1. For $0 \leq i \leq j \leq n$, $D[i, j]$ is the maximum score of a path from $(0, i)$ to (m, j). For $i > j$, $D[i, j] = j - i$. This extension ensures that the properties discussed in the introduction will hold for the lower triangular part of D as well. The matrix D is called the *all scores matrix of* G.
2. For $1 \leq i, j \leq n$, $D^{\square}[i, j] = (D[i, j] + D[i - 1, j - 1]) - (D[i - 1, j] + D[i, j - 1])$. The matrix $D^{\square}$ is called the *density matrix of* D.
3. Let $S = \{0, 1, \ldots, r\}$ be a set of *symbols*. For $0 \leq i, j \leq n$, $D_{\text{first}}[i, j]$ is a subset of S such that for every $s \in S$, $s \in D_{\text{first}}[i, j]$ if and only if there is an s-path from $(0, i)$ to (m, j) with score $D[i, j]$.

To illustrate the importance of the D_{first} matrix, consider a region in D_{first} in which all entries contain the same symbol s. Then, the crossing paths property holds for indices in this region (since every path encoded in this region passes through f_s), so we obtain that the Monge property holds inside this region.

Next, we point out the entries in D and in $D^{\square}$ that are affected by a bridge in G. We refer the reader to Fig. 2 for an example of the definitions given below. Henceforth, a pair (i, j) of integers referring to a matrix entry will be called an index. For some bridge $f_k = ((i_s, j_s), (i_e, j_e))$, we define $\text{start}(f_k) = j_s$ and $\text{end}(f_k) = j_e$. We also define $E_k = \{(i, j) : 0 \leq i \leq \text{start}(f_k), \text{end}(f_k) \leq j \leq n\}$. In other words, E_k contains all indices (i, j) in D such that paths from $(0, i)$ to (m, j) can pass through f_k. The *boundary* of f_k is a set of indices in $D^{\square}$, defined as $B_k = \{(i, \text{end}(f_k)) : 1 \leq i \leq \text{start}(f_k) + 1\} \cup \{(\text{start}(f_k) + 1, j) : \text{end}(f_k) \leq j \leq n\}$. The two sets in the definition of B_k are called the *left boundary* and *bottom boundary* of f_k, respectively. We say that an index (i, j) is a *boundary index* in $D^{\square}$ if it is contained in the boundary of some f_k. An index (i, j) is an *intersection index* if there are k, k' (possibly $k = k'$) such that (i, j) is in the left boundary of f_k and in the bottom boundary of $f_{k'}$.

In the introduction we gave two properties of the density matrix. We now restate these properties using the definitions above. fk : bridge, Ek : Entries, Bk : boundary

Property 1 *Non-zero values other than* $-1, 1$ *can appear only at intersection indices (Theorem 1).* -1,1 以外的非零值只能出現在交叉點索引處

Property 2 *In every row or column, the number of* -1 *values in non-boundary indices is at most* r*, and the number of* 1 *values in non-boundary indices is at most* r *(Theorem 4).* 在每一行或列中，非邊界索引中的-1/1值的數量最多為 r

We will assert Property 2 only for columns. Hence, the lemmas that will follow will be formulated and proved solely for columns. The same properties can be asserted symmmetrically for rows as well.

In what follows we achieve preliminary results that we will later utilize in Sect. 3 to prove Property 2 for the restricted case of a single bridge (Theorem 2), and in Sect. 4 for the purpose of the all scores construction algorithm (Theorem 3). Section 5 is designated to prove Property 2 for the general case of r bridges (Theorem 4).

In the rest of the paper we implicitly assume that matrix entries (i, j) satisfy $i < j$. Note that the two properties stated above are satisfied for (i, j) with $i \geq j$ due to the following observations:

- If $i > j+1$, $D^{\square}[i, j] = (j-i)+((j-1)-(i-1))-(j-(i-1))-((j-1)-i) = 0$.
- If $i = j+1$ then $D^{\square}[i, j] = (j-(j+1))+((j-1)-((j+1)-1))-D[j, j]-((j-1)-(j+1)) = -D[j, j]$, so in this case $D^{\square}[i, j] = 0$ unless there is a bridge f_k with $\text{start}(f_k) = \text{end}(f_k) = j$, in which case (i, j) is an intersection index.
- Similarly, for $i = j$, $D^{\square}[i, j] \in \{0, 1\}$ unless one of the following two cases occurs: (1) There is a bridge f_k with $\text{start}(f_k) = j - 1$ and $\text{end}(f_k) = j$. (2) There are bridges f_k and $f_{k'}$ with $\text{start}(f_k) = \text{end}(f_k) = j - 1$ and $\text{start}(f_{k'}) = \text{end}(f_{k'}) = j$. In both cases (i, j) is an intersection index.

We now give a proof for Property 1. To this end, we need the following definition and lemma.

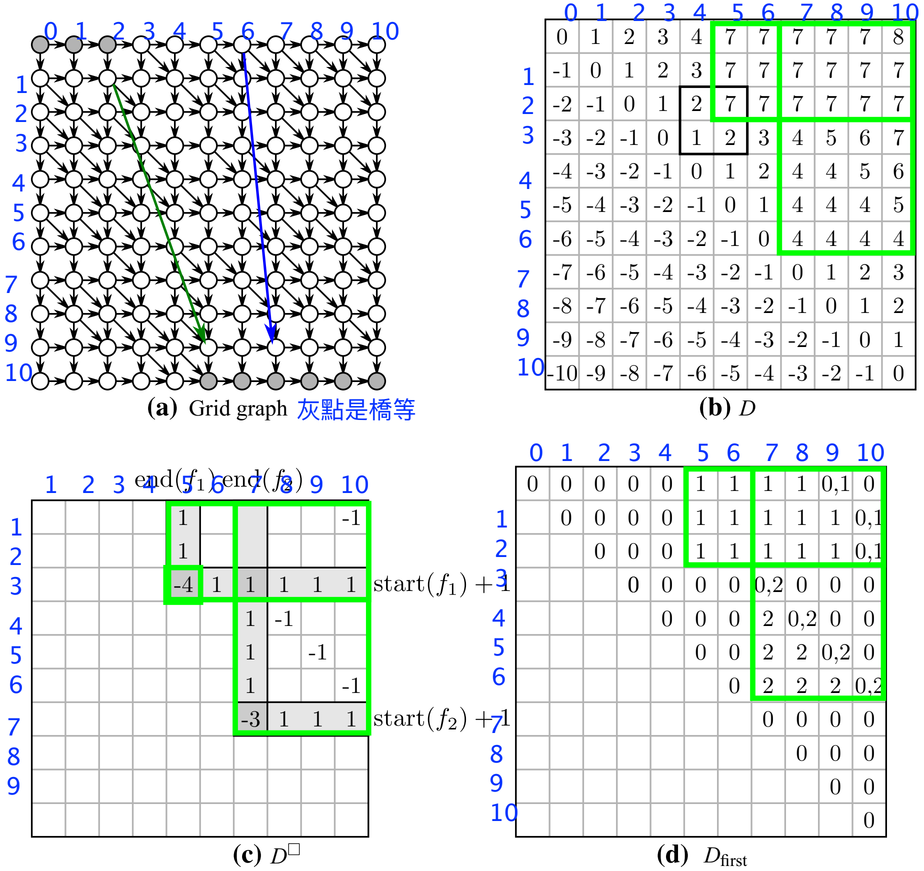


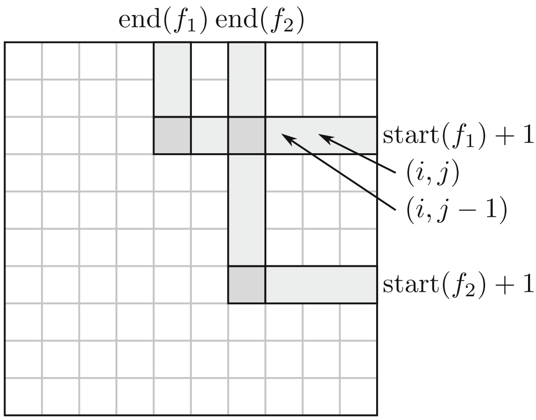
Fig. 2 **a** Contains an example of a grid graph with two bridges. The weight of the bridge $f_1 = ((1, 2), (9, 5))$ is 7, and the weight of the bridge $f_2 = ((0, 6), (9, 7))$ is 4. The matrices D, $D^{\square}$, and D_{first} of the graph are shown in **b–d**, respectively. Only the non-zero values of the density matrix are shown. The boundary indices are marked in gray, and the intersection indices are marked with darker gray. As stated in the text, each column or row of the density matrix can contain at most two negative values in non-boundary indices, and these values must be -1. The only pairs of vertices (where one vertex belongs the first row and the other vertex belongs to the last row of the grid graph) that can utilize the bridge f_1, are the pairs composed of the vertices highlighted in gray in (**a**). Thus, the entries that are affected by the bridge f_1 are $E_1 = \{(i, j) : 0 \leq i \leq 2, 5 \leq j \leq 10\}$. The boundary of this region is $B_1 = \{(1, 5), (2, 5), (3, 5)\} \cup \{(3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (3, 10)\}$. Note that the index $(3, 5)$ is an intersection index and that the value of $D^{\square}[3, 5]$ is $2 + 2 - 7 - 1 = -4$, and these four values in D are marked in **b** with a bold rectangle. The cause for the negative value in $D^{\square}[3, 5]$ is that the cell $(2, 5)$ in D is the only one of the four corresponding cells that can utilize the bridge f_1

都在同一個橋上

Definition 1 A pair of indices $(i_1, j_1), (i_2, j_2)$ in the matrix D are said to be *bridge equivalent* if for every $1 \leq k \leq r$, $(i_1, j_1) \in E_k$ if and only if $(i_2, j_2) \in E_k$. In other words, $(i_1, j_1), (i_2, j_2)$ are bridge equivalent if any bridge that can be visited by a path from $(0, i_1)$ to (m, j_1), can also be visited by a path from $(0, i_2)$ to (m, j_2), and vice versa.

Lemma 1 *For every* i, j,

Fig. 3 An example of the proof of Theorem 1. The pair of indices $(i, j-1)$, (i, j) are bridge equivalent (both indices belong to E_2 and not to E_1), and the pair of indices $(i-1, j-1)$, $(i-1, j)$ are bridge equivalent (both indices belong to E_1 and to E_2)



水平 1. *If* $(i, j-1)$ *and* (i, j) *are bridge equivalent,* $D[i, j-1] \leq D[i, j] \leq D[i, j-1]+1$.
垂直 2. *If* $(i-1, j)$ *and* (i, j) *are bridge equivalent,* $D[i, j] \leq D[i-1, j] \leq D[i, j]+1$.

The lemma follows due to the fact that a path from $(0, i)$ to (m, j) can be obtained by appending the edge $((m, j-1), (m, j))$ to a path from $(0, i)$ to $(m, j-1)$. Also, since $(i, j-1)$ and (i, j) are bridge equivalent, there is no bridge f_k with $\text{end}(f_k) = j$. Therefore, a path from $(0, i)$ to $(m, j-1)$ can be obtained by truncating a path from $(0, i)$ to (m, j). The first part of Lemma 1 is now obtained.

Theorem 1 *Non-zero values other than* -1 *or* 1 *can appear only at intersection indices.*

Proof If (i, j) is not an intersection index then either (i, j) does not belong to any left boundary, or (i, j) does not belong to any bottom boundary. Assume without loss of generality the former case. In this case, $(i, j-1)$, (i, j) are bridge equivalent (see Fig. 3). Moreover, either $(i, j-1)$ does not belong to any left boundary, or $(i-1, j)$ is the bottom index of some left boundary (namely $(i-1, j) = (\text{start}(f_k)+1, \text{end}(f_k))$ for some $1 \leq k \leq r$). In both cases, $(i-1, j-1)$, $(i-1, j)$ are bridge equivalent. Hence, we can rearrange the terms in the definition of $D^{\square}[i, j]$ and obtain that $D^{\square}[i, j] = \Delta_1 - \Delta_2$, where $\Delta_1 = D[i, j] - D[i, j-1]$ and $\Delta_2 = D[i-1, j] - D[i-1, j-1]$. By Lemma 1, $\Delta_1, \Delta_2 \in \{0, 1\}$, and thus $D^{\square}[i, j] \in \{-1, 0, 1\}$. □

Note that in the density matrix, the number of non-zero elements within the boundaries can be immediately bounded by $O(rn)$. Hence, in the rest of the paper we will focus on non-zero elements that are not within the boundaries, and show that the number of such elements is also $O(rn)$. Therefore, we will obtain that the total number of non-zero elements in the density matrix is $O(rn)$. Moreover, we will show in Sect. 6 that the $O(rn)$ bound is tight. We next give several lemmas which will be used later to prove this upper bound in Sects. 3 and 5.

Definition 2 An index (i, j) in $D^{\square}$ which is not a boundary index and for which $D^{\square}[i, j] < 0$ (resp. $D^{\square}[i, j] > 0$) is called a *negative injury* (resp. *positive injury*). The submatrices $D[i-1..i, j-1..j]$ and $D_{\text{first}}[i-1..i, j-1..j]$ are called the submatrices of D and D_{first} corresponding to the injury, respectively.

An injury induces a unique structure in the corresponding submatrices D and D_{first}, which we will characterize next. Due to similarity, we omit the proofs for the positive injuries. In the next lemma we deduce the structure of a submatrix of D that corresponds to an injury.

Lemma 2 – *For a negative injury* (i, j), $D[i-1..i, j-1..j] = \left(\begin{smallmatrix} x & x+1 \\ x & x \end{smallmatrix}\right)$ *for some* x.
– *For a positive injury* (i, j), $D[i-1..i, j-1..j] = \left(\begin{smallmatrix} x & x \\ x-1 & x \end{smallmatrix}\right)$ *for some* x.

Proof As in the proof of Theorem 1, $D^{\square}[i, j] = \Delta_1 - \Delta_2$, where $\Delta_1 = D[i, j] - D[i, j-1]$ and $\Delta_2 = D[i-1, j] - D[i-1, j-1]$. The lemma follows since $D^{\square}[i, j] < 0$ and $\Delta_1, \Delta_2 \in \{0, 1\}$. 水平關係 □

Corollary 1 *If* (i, j) *is a negative (resp. positive) injury then* $(i+1, j)$ *and* $(i, j+1)$ *are not negative (resp. positive) injuries.*

The next lemma gives a general property for adjacent cells in D_{first} with regard to their respective values in the matrix D.

Lemma 3 *For every* i, j,

水平 1. *If* $(i, j-1)$ *and* (i, j) *are bridge equivalent,*
   (a) *If* $D[i, j-1] = D[i, j]$ *then* $D_{\text{first}}[i, j-1] \subseteq D_{\text{first}}[i, j]$.
   (b) *If* $D[i, j-1] + 1 = D[i, j]$ *then* $D_{\text{first}}[i, j] \subseteq D_{\text{first}}[i, j-1]$.

垂直 2. *If* $(i-1, j)$ *and* (i, j) *are bridge equivalent,*
   (a) *If* $D[i, j] = D[i-1, j]$ *then* $D_{\text{first}}[i, j] \subseteq D_{\text{first}}[i-1, j]$.
   (b) *If* $D[i, j] + 1 = D[i-1, j]$ *then* $D_{\text{first}}[i-1, j] \subseteq D_{\text{first}}[i, j]$.

Proof We first prove 1.(a). Choose a symbol $s \in D_{\text{first}}[i, j-1]$, and let P be an s-path from $(0, i)$ to $(m, j-1)$ with score $D[i, j-1]$. The path P' obtained by appending the vertex (m, j) to P is an s-path from $(0, i)$ to (m, j) with score $D[i, j-1] = D[i, j]$ Therefore, $s \in D_{\text{first}}[i, j]$.

We next prove 1.(b). Let $s \in D_{\text{first}}[i, j]$, and let P be an s-path from $(0, i)$ to (m, j) with score $D[i, j]$. Since $(i, j-1)$,(i, j) are bridge equivalent, P cannot pass through a bridge f with $\text{end}(f) > j-1$, so P has vertices on column $j-1$. Denote by k the maximal index such that $(k, j-1) \in P$. The path P' obtained by taking the prefix of P until $(k, j-1)$, and appending the vertices $(k+1, j-1), \ldots, (m, j-1)$ is an s-path from $(0, i)$ to $(m, j-1)$ with score at least $D[i, j] - 1 = D[i, j-1]$. It follows that $s \in D_{\text{first}}[i, j-1]$.

The proofs of 2.(a) and 2.(b) are symmetrical to the proofs of the first two parts, and thus they are omitted. □

Now let us consider the structure of a submatrix of D_{first} that corresponds to an injury, denote this as $D_{\text{first}}[i-1..i, j-1..j] = \left(\begin{smallmatrix} \gamma & \beta \\ \alpha & \delta \end{smallmatrix}\right)$. The following lemma follows directly from Lemmas 2 and 3.

Lemma 4 *For an injury at* (i, j) *with the corresponding submatrix:* $D_{\text{first}}[i-1..i, j-1..j] = \left(\begin{smallmatrix} \gamma & \beta \\ \alpha & \delta \end{smallmatrix}\right)$,

1. *If* (i, j) *is a negative injury, then* $\alpha \subseteq \gamma \cap \delta$ *and* $\beta \subseteq \gamma \cap \delta$.

2. *If* (i, j) *is a positive injury, then* $\gamma \subseteq \alpha \cap \beta$ *and* $\delta \subseteq \alpha \cap \beta$.

In order to restrict values of D in indices for which the corresponding entries in D_{first} contain the same symbol s, we define a matrix D_s as follows. For a symbol $s \in S$, let D_s be a matrix in which for every $(i, j) \in E_s$, $D_s[i, j]$ is the maximum score of an s-path from $(0, i)$ to (m, j). For $s = 0$, D_s is defined as above, except that $D_s[i, j]$ is defined for every $0 \leq i, j \leq n$. Note that $D_s[i, j] \leq D[i, j]$ for every (i, j) for which $D_s[i, j]$ is defined.

Lemma 5 *For every* $s \in S$, *the matrix* D_s *has the Monge property.*

Proof For $s = 0$ the lemma is true due to the crossing paths property for grid graphs with no bridges. For $s > 0$ we also have the crossing paths property: For every index (i, j), a maximum score s-path from $(0, i-1)$ to (m, j) must cross a maximum score s-path from $(0, i)$ to $(m, j-1)$ as both paths pass through f_s. Thus, the lemma follows. □

Our next goal is to show that every column in the density matrix contains at most r negative injuries and at most r positive injuries. As the proofs are symmetrical for these cases, we will just discuss the former case and so from now on we will simply refer to negative injuries as 'injuries'. Consider a fixed column, and assume that this column has k injuries, let $D_i = \begin{pmatrix} \gamma_i & \beta_i \\ \alpha_i & \delta_i \end{pmatrix}$ be the submatrix of D_{first} corresponding to the i-th injury, where the injuries are numbered in increasing row indices. Note that the submatrices D_i are disjoint (by Corollary 1). Our approach for proving that $k \leq r$ is based on showing properties of the D_{first} matrix. One of our techniques is showing that there are forbidden *structures* in D_{first}. For example, Lemma 6 below states that a structure consisting of a symbol $s \in \beta_i$ and $s \in \alpha_j$ for $j \geq i$ is forbidden. For the case of $r = 1$, applying this lemma with $i = j$ implies that there are only two possible cases for α_i, β_i: either $\{0\}, \{1\}$ or $\{1\}, \{0\}$. If we assume conversely that there are $k = 2$ injuries, then there are four possible cases for $\alpha_1, \beta_1, \alpha_2, \beta_2$. We then use Lemma 6 and an additional lemma (Lemma 7) that gives another forbidden structure in D_{first}, and show that each of these four cases cannot occur. This is a contradiction, and therefore there cannot be two injuries.

Lemma 6 *For every* $1 \leq i \leq j \leq k$, $\beta_i \cap \alpha_j = \emptyset$.

否證 *Proof* Fix $i \leq j$, and assume conversely that $s \in \beta_i \cap \alpha_j$. By Lemma 2, the submatrices of D corresponding to injuries i and j are $D' = \begin{pmatrix} x & x+1 \\ x & x \end{pmatrix}$ for some x, and $D'' = \begin{pmatrix} y & y+1 \\ y & y \end{pmatrix}$ for some y, respectively (see Fig. 4). Let D'_s and D''_s be the submatrices of D_s that correspond to D' and D'', respectively. From the assumption $s \in \beta_i$ and Lemma 4, we have that $s \in \gamma_i$. Thus, the first row of D'_s is equal to the first row of D'. Similarly, we have that $s \in \delta_j$ and therefore the last row of D''_s is equal to the last row of D''. By taking the first row of D'_s and the last row of D''_s, we obtain that D_s contains a submatrix $\begin{pmatrix} x & x+1 \\ y & y \end{pmatrix}$ and therefore D_s does not have the Monge property. This contradicts Lemma 5. □

Finally, we give another forbidden structure in D_{first}, based on a variant of the crossing paths property.

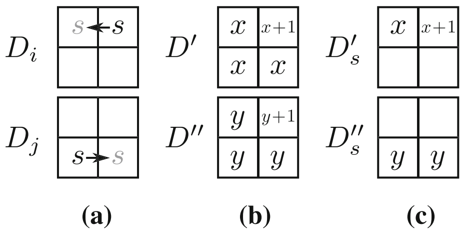


Fig. 4 An illustration of the proof of Lemma 6. The gray s symbols in **a** represent values that are obtained using Lemma 4. **a** D_{first}, **b** D, and **c** D_s

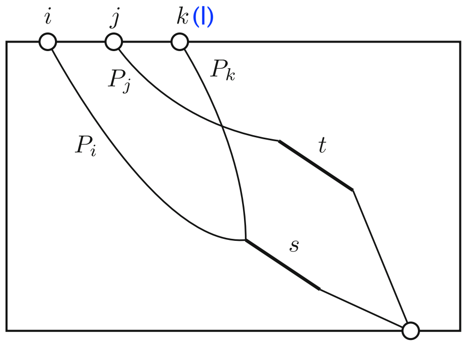


Fig. 5 An illustration of the proofs of Lemmas 7 and 12

Definition 3 Let $\preceq$ be a linear order on $S = \{0, 1, \ldots, r\}$ defined as follows. For every $i \neq j$, $i \preceq j$ if and only if $\text{start}(f_i) \leq \text{start}(f_j)$, where $\text{start}(f_0) = \infty$.

Lemma 7 *Let d_i, d_j, d_ℓ be subsets on rows i, j, ℓ of some column i' of D_{first}, where $i < j < \ell$. Then, there are no $s, t \in S$ such that $s \preceq t$, $s \in d_i \cap d_\ell$, $t \notin d_i \cup d_\ell$, and $t \in d_j$.*

Proof Assume conversely that there are $s, t \in S$ such that $s \preceq t$, $s \in d_i \cap d_\ell$, $t \notin d_i \cup d_\ell$, and $t \in d_j$. Note that $s \neq 0$ since by definition, $0 \npreceq t$.

Let P_i, P_ℓ be maximum score s-paths from $(0, i)$ and $(0, \ell)$ to (m, i'), respectively. Let P_j be a maximum score t-path from $(0, j)$ to (m, i'). Since f_s is the first bridge in P_i, P_ℓ and f_t is the first bridge in P_j, and also $s \preceq t$, then, in the subgraph of G that contains the vertices above and to the left of the start vertex of f_s, the paths P_i, P_j, P_ℓ do not pass through bridges (see Fig. 5). Thus, P_j must cross one of the paths P_i and P_ℓ. Assume without loss of generality that P_j crosses P_ℓ.

Let P_j^1, P_ℓ^1 denote the prefixes of P_j, P_ℓ until the crossing point, and let P_j^2, P_ℓ^2 denote the suffixes of P_j, P_ℓ from the crossing point. Let y, z denote the scores of the paths P_j, P_ℓ, respectively, and let a, b denote the score of the paths P_j^1, P_ℓ^1, respectively.

We have that the path $P_\ell^1 \cup P_j^2$ is a t-path from $(0, \ell)$ to (m, i'). Since $t \notin d_\ell$, we conclude that $b + (y - a) < z$. Furthermore, due to the path $P_j^1 \cup P_\ell^2$ we have $a + (z - b) \leq y$. Summing the two inequalities above we obtain $y + z < y + z$, a contradiction. □

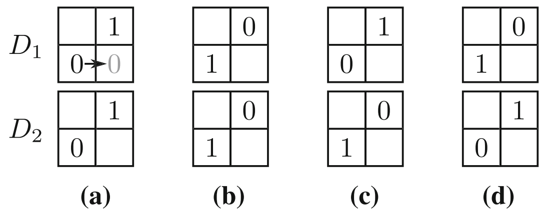


Fig. 6 The four cases for two injuries in the proof of Theorem 2. The gray 0 in **a** represents a value that is obtained using Lemma 4

3 Properties of the One Bridge Case

In this section we assume the grid graph has a single bridge, $f = ((i_{\text{start}}, j_{\text{start}}), (i_{\text{end}}, j_{\text{end}}))$, and show that there is at most one injury in every column of $D^{\square}$. As discussed in the previous section, this corresponds to $O(n)$ non-zero elements in $D^{\square}$.

Theorem 2 *In the case of a single bridge, there is at most one injury in every column of* $D^{\square}$.

否證 *Proof* Fix some column of $D^{\square}$, and suppose conversely that there are at least two injuries in this column. Recall that $D_i = \begin{pmatrix} \gamma_i & \beta_i \\ \alpha_i & \delta_i \end{pmatrix}$ is the submatrix of D_{first} corresponding to the i-th injury. By Lemma 6, $\alpha_i \cap \beta_i = \emptyset$, and since α_i and β_i are non empty subsets of $S = \{0, 1\}$, it follows that either D_i is of the form $\begin{pmatrix} \cdot & 1 \\ 0 & \cdot \end{pmatrix}$ or D_i is of the form $\begin{pmatrix} \cdot & 0 \\ 1 & \cdot \end{pmatrix}$. Considering the first two injuries, there are four possible cases (see Fig. 6):

1. D_1, D_2 are of the form $\begin{pmatrix} \cdot & 1 \\ 0 & \cdot \end{pmatrix}$.
2. D_1, D_2 are of the form $\begin{pmatrix} \cdot & 0 \\ 1 & \cdot \end{pmatrix}$.
3. D_1 is of the form $\begin{pmatrix} \cdot & 1 \\ 0 & \cdot \end{pmatrix}$ and D_2 is of the form $\begin{pmatrix} \cdot & 0 \\ 1 & \cdot \end{pmatrix}$.
4. D_1 is of the form $\begin{pmatrix} \cdot & 0 \\ 1 & \cdot \end{pmatrix}$ and D_2 is of the form $\begin{pmatrix} \cdot & 1 \\ 0 & \cdot \end{pmatrix}$.

We now show that each of the cases above yields a contradiction. In Case 1, we have from Lemma 4 that $0 \in \delta_1$. Note that by Corollary 1, D_1 and D_2 are disjoint. Thus, we can apply Lemma 7 on $\beta_1, \delta_1, \beta_2$ and obtain a contradiction (taking $s = 1$ and $t = 0$). Case 2 yields a contradiction using similar arguments. In Cases 3 and 4, we have $1 \in \beta_1 \cap \alpha_2$ and $0 \in \beta_1 \cap \alpha_2$, respectively, which is a contradiction to Lemma 6. □

4 Algorithm for Constructing All-Scores Matrices

In this section we give an algorithm for computing the all scores matrix of a grid graph with bridges. Our algorithm is an extension of the algorithm of Schmidt for a grid graph without bridges [33]. We follow the presentation of Schmidt's algorithm which was given in Matarazzo et al. [29]. For clarity of presentation, we will first describe an algorithm for the case of a single bridge, and we will later handle the case of $r > 1$ bridges.

Let $f = ((i_{\text{start}}, j_{\text{start}}), (i_{\text{end}}, j_{\text{end}}))$ be the single bridge of the grid graph, and let W_f denote its weight.

Let $G_0, \ldots, G_m$ be grid graphs, where G_i is the subgraph of G induced by all the vertices (i', j) with $0 \le i' \le i$ and $0 \le j \le n$. Let $D_0, \ldots, D_m$ be the all scores matrices of $G_0, \ldots, G_m$, respectively.

For $0 \le k \le n$, define:

水平 col 垂直 row

$$\text{DIFFC}_{i,j}(k) = D_i[k, j+1] - D_i[k, j] \quad \text{and} \quad \text{DIFFR}_{i,j}(k) = D_{i+1}[k, j] - D_i[k, j].$$

The following lemma follows from the definition above.

Lemma 8 *For $i \le m$, if all $\text{DIFFC}_{i,j}(k)$ values are known for all j and k, then the matrix D_i can be constructed in $O(n^2)$ time.*

Proof Construct the columns of D_i from left to right. The $(j+1)$-th column is constructed in $O(n)$ time from the j-th column and the values of $\text{DIFFC}_{i,j}$. □

Our algorithm for constructing the all-scores matrix of G computes all $\text{DIFFC}_{m,j}$ functions and then applies Lemma 8. The algorithm is based on the following observations on the $\text{DIFFC}_{i,j}$ and $\text{DIFFR}_{i,j}$ functions.

1. 壓縮 Most $\text{DIFFC}_{i,j}$ and $\text{DIFFR}_{i,j}$ functions have compact representations of size $O(1)$.
2. 更新 The compact representations of $\text{DIFFC}_{i+1,j}$ and $\text{DIFFR}_{i,j+1}$ can be computed efficiently from the compact representations of $\text{DIFFC}_{i,j}$ and $\text{DIFFR}_{i,j}$.

The first observation is shown in Lemma 9 and the second observation is shown in Lemma 10. Similar properties were used in the algorithm of Schmidt for grid graphs with no bridges. In that case, *all* the $\text{DIFFC}_{i,j}$ and $\text{DIFFR}_{i,j}$ functions have compact representations, and the size of each representation is *exactly* 1. In the case of a grid graph with a single bridge, we need additional steps to handle the $\text{DIFFC}_{i,j}$ and $\text{DIFFR}_{i,j}$ functions that do not have compact representations.

We now give a compact representation for the $\text{DIFFR}_{i,j}$ and $\text{DIFFC}_{i,j}$ functions. An index k for which $\text{DIFFC}_{i,j}(k-1) \ne \text{DIFFC}_{i,j}(k)$ will be called a *step index* of $\text{DIFFC}_{i,j}$. For $i \ne i_{\text{end}} - 1$, the *compact representation* $\text{SC}_{i,j}$ of $\text{DIFFC}_{i,j}$ is an array whose first element is $\text{DIFFC}_{i,j}(0)$ and the rest of its elements are the step indices of $\text{DIFFC}_{i,j}$ in increasing order. For $i = i_{\text{end}} - 1$, $\text{SC}_{i,j}$ is an array containing the values $\text{DIFFC}_{i,j}(k)$ for $0 \le k \le n$. We say that $\text{SC}_{i_{\text{end}}-1,j}$ is non-compact. The arrays $\text{SR}_{i,j}$ are defined similarly.

Lemma 9 *For every $i \ne i_{\text{end}} - 1$ and $j \ne j_{\text{end}} - 1$, the arrays $\text{SC}_{i,j}$ and $\text{SR}_{i,j}$ have $O(1)$ elements each. Moreover, the functions $\text{DIFFC}_{i,j}$ and $\text{DIFFR}_{i,j}$ can be computed from the arrays $\text{SC}_{i,j}$ and $\text{SR}_{i,j}$ in $O(n)$ time, respectively.*

Proof We first prove the lemma for the $\text{DIFFC}_{i,j}$ functions. By definition,

$$\text{DIFFC}_{i,j}(k) = \text{DIFFC}_{i,j}(0) + \sum_{l=1}^{k} D_i^{\square}[l, j+1].$$

Thus, an index k is a step index of $\text{DIFFC}_{i,j}$ if and only if $D_i^{\square}[k, j+1] \ne 0$. Therefore, by Property 2, the size of $\text{DIFFC}_{i,j}$ is $O(1)$.

By definition, if k_1 and k_2 are consecutive step indices then $\textsc{DiffC}_{i,j}$ is constant on the range $[k_1, k_2 - 1]$. By Lemma 1, $\textsc{DiffC}_{i,j}(k) \in \{0, 1\}$ for all k. Therefore, the step indices of $\textsc{DiffC}_{i,j}$ partition the range $[0, n]$ into regions such that the value of $\textsc{DiffC}_{i,j}$ is constant in each region and alternates between 0 and 1. The second part of the lemma follows.

To prove the lemma for the $\mathrm{SR}_{i,j}$ functions, we use the equality:

$$\textsc{DiffR}_{i,j}(k) = \textsc{DiffR}_{i,j}(0) + \sum_{l=1}^{k} \overline{D}_j^{\square}[l, i+1],$$

where $\overline{D}_j$ is a matrix in which $\overline{D}_j[k, i]$ is the maximum score path from $(0, k)$ to (i, j). The matrix $\overline{D}_j$ satisfies Property 1 and Property 2. This can be shown by constructing a new grid graph whose density matrix contains $\overline{D}_j$ as a submatrix (cf. [35], Chapter 4.3, Definition 4.8). The claims of the lemma on $\textsc{DiffR}_{i,j}$ now follow using the same arguments used to prove the lemma on $\textsc{DiffC}_{i,j}$. □

In the following lemma we show that $\mathrm{SC}_{i+1,j}$ and $\mathrm{SR}_{i,j+1}$ can be computed efficiently from $\mathrm{SC}_{i,j}$ and $\mathrm{SR}_{i,j}$. For every $(i, j) \neq (i_{\mathrm{end}} - 1, j_{\mathrm{end}} - 1)$ and $k \leq j$, the optimal path from $(0, k)$ to $(i + 1, j + 1)$ passes through either $(i + 1, j)$, (i, j), or $(i, j + 1)$. Thus,

$$D_{i+1}[k, j+1] = \max\{D_{i+1}[k, j], D_i[k, j] + W_{i,j}, D_i[k, j+1]\},$$

where $W_{i,j} = 1$ if there is a diagonal edge entering (i, j) and $W_{i,j} = 0$ otherwise. From the equality above, the following formulas for $\textsc{DiffC}_{i+1,j}$ and $\textsc{DiffR}_{i,j+1}$ are obtained (see [29], Lemma 4).

Lemma 10 *Let* $(i, j) \neq (i_{\mathrm{end}} - 1, j_{\mathrm{end}} - 1)$ *and* $0 \leq k \leq j$. *If* $W_{i,j} = 1$ *then* $\textsc{DiffC}_{i+1,j}(k) = 1 - \textsc{DiffR}_{i,j}(k)$ *and* $\textsc{DiffR}_{i,j+1}(k) = 1 - \textsc{DiffC}_{i,j}(k)$. *Otherwise,* $\textsc{DiffC}_{i+1,j}(k) = \max(0, \textsc{DiffC}_{i,j}(k) - \textsc{DiffR}_{i,j}(k))$ *and* $\textsc{DiffR}_{i,j+1}(k) = \max(0, \textsc{DiffR}_{i,j}(k) - \textsc{DiffC}_{i,j}(k))$.

Our algorithm for computing the arrays $\mathrm{SC}_{m,j}$, traverses every i, j and computes $\mathrm{SC}_{i+1,j}$ and $\mathrm{SR}_{i,j+1}$ from $\mathrm{SC}_{i,j}$ and $\mathrm{SR}_{i,j}$ using Lemma 10. When $i \notin \{i_{\mathrm{end}} - 1, i_{\mathrm{end}}\}$ and $j \notin \{j_{\mathrm{end}} - 1, j_{\mathrm{end}}\}$, this computation takes $O(1)$ time by Lemma 9. There are two cases which require a special treatment. The first case is $(i, j) = (i_{\mathrm{end}} - 1, j_{\mathrm{end}} - 1)$. In this case Lemma 10 cannot be applied and thus $\mathrm{SC}_{i+1,j}$ and $\mathrm{SR}_{i,j+1}$ must be computed differently. Here we compute $D_{i+1}[k, j]$, $D_i[k, j+1]$, and $D_{i+1}[k, j+1]$, for every $0 \leq k \leq n$. Then, we use these values to compute $\textsc{DiffC}_{i+1,j}(k)$ and $\textsc{DiffR}_{i,j+1}(k)$ for all k, and finally we compute $\mathrm{SC}_{i+1,j}$ and $\mathrm{SR}_{i,j+1}$ from $\textsc{DiffC}_{i+1,j}$ and $\textsc{DiffR}_{i,j+1}$. The values $D_{i+1}[k, j]$ for all k are computed in $O(n^2)$ time by computing the j leftmost columns of D_{i+1} as in Lemma 8 (note that we already computed $\textsc{DiffC}_{i+1,j'}$ for all $j' < j$). Similarly, the values $D_i[k, j+1]$ for all k are computed in $O(n^2)$ time. To compute the $D_{i+1}[k, j+1]$ values, we use the equality

$$\begin{aligned} D_{i+1}[k, j+1] = \max\{&D_i[k, j+1], D_i[k, j] + W_{i,j}, \\ &D_{i+1}[k, j], D_{i_{\mathrm{start}}}[k, j_{\mathrm{start}}] + W_f\}. \end{aligned}$$

The second special case occurs when $i \in \{i_{\text{end}} - 1, i_{\text{end}}\}$ or $j \in \{j_{\text{end}} - 1, j_{\text{end}}\}$. In this case Lemma 9 does not apply. Therefore, we can only bound the time to compute $\text{SC}_{i+1,j}$ and $\text{SR}_{i,j+1}$ by $O(n)$. Since there are $O(n+m)$ pairs i, j for which this case occurs, the total contribution of this case to the time complexity of the algorithm is $O(n^2 + nm)$.

Extension to r Bridges The algorithm presented above can be extended to the case of $r > 1$ bridges. Denote the k'th bridge by $f_k = ((i^k_{\text{start}}, j^k_{\text{start}}), (i^k_{\text{end}}, j^k_{\text{end}}))$ for $1 \le k \le r$. In this case, Property 2 implies that for every pair i and j such that $i \ne i^k_{\text{end}} - 1$ and $j \ne j^k_{\text{end}} - 1$ for all $1 \le k \le r$, $\text{DIFFC}_{i,j}$ and $\text{DIFFR}_{i,j}$ have $O(r)$ step indices. Therefore, the computation of $\text{SC}_{i,j}$, $\text{SR}_{i,j}$ for indices i and j, such that $i \notin \{i^k_{\text{end}} - 1, i^k_{\text{end}}\}$ and $j \notin \{j^k_{\text{end}} - 1, j^k_{\text{end}}\}$, for all $1 \le k \le r$, takes $O(rnm)$ time. As for the pairs i, j that require special treatment, the technique remains as in the case of one bridge, only that now there are at most r end-points (for which $\text{SC}_{i,j}$ and $\text{SR}_{i,j}$ have to be computed exhaustively) and at most $r(n+m)$ such indices. Summing the above, the following theorem is obtained.

Theorem 3 *The all scores matrix for an $m \times n$ grid graph with r bridges can be constructed in $O(r(nm + n^2))$ time.*

5 Properties of the *r* Bridges Case

In this section we assume that the grid graph G has r bridges $f_1, \ldots, f_r$ and show that each column of $D^{\square}$ has at most r injuries. Recall that in Sect. 2, Lemmas 6 and 7 gave forbidden structures in D_{first}. The following lemmas, Lemmas 11 and 12, are analogous to Lemmas 6 and 7, but give different forbidden structures in D_{first}. In what follows we consider a fixed column in $D^{\square}$ and assume that this column has k injuries numbered in increasing row indices.

Lemma 11 *For every $1 \le i \le j \le k$, there is no $s \in S$ such that $s \in \beta_i \cap \gamma_j$ and $s \notin \beta_j$.*

否證 *Proof* Fix $i \le j$, and assume conversely that there is $s \in S$ such that $s \in \beta_i \cap \gamma_j$ and $s \notin \beta_j$. We use the same notations as in the proof of Lemma 6. the submatrices of D corresponding to injuries i and j are $D' = \begin{pmatrix} x & x+1 \\ x & x \end{pmatrix}$ for some x, and $D'' = \begin{pmatrix} y & y+1 \\ y & y \end{pmatrix}$ for some y, respectively. Let D'_s and D''_s be the submatrices of D_s that correspond to D' and D'', respectively. As in the proof of Lemma 6, we have that the first row of D'_s is equal to the first row of D'. Now consider the first row of D''_s. The first element of this row is equal to the first element in the first row of D'' since $s \in \gamma_j$. Furthermore, $s \notin \beta_j$, so the second element of this row is less than the second element of the first row of D''. Since each row of D_s is monotonically non-decreasing, we conclude that the first row of D''_s is $(\,y\ y\,)$. We obtain that D_s contains a submatrix $\begin{pmatrix} x & x+1 \\ y & y \end{pmatrix}$ and therefore D_s does not have the Monge property, a contradiction. □

Lemma 12 *Let d_i, d_j, d_ℓ be the values on rows i, j, ℓ of some column i' of D_{first}, where $i < j < \ell$. Then, there are no $s, t \in S$ such that $s \preceq t$, $s \in d_i \cap d_\ell$, $s \notin d_j$, and $t \in d_j$.*

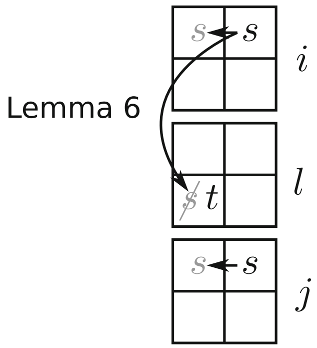


Fig. 7 An illustration of the proof of Lemma 13. The gray symbols represent values that are obtained using Lemmas 4 and 6

Proof Using the same notation as in the proof of Lemma 7 (see also Fig. 5), we have that the path $P^1_\ell \cup P^2_j$ is a path from $(0, \ell)$ to (m, i'), and therefore the score of this path is less than or equal to the maximum score of a path between these vertices. Therefore, $b + (y - a) \leq z$. Moreover, the path $P^1_j \cup P^2_\ell$ is an s-path from $(0, j)$ to (m, i'). Since $s \notin d_j$, the score of this path is strictly less than the maximum score of a path between these vertices, so $a + (z - b) < y$. Summing the two inequalities above, we obtain $y + z < y + z$, a contradiction. □

In Lemma 13, Corollary 2 and Lemma 14, we use the former lemmas to show additional, more complex, forbidden structures in D_{first}.

Let $\alpha[i, j] = \bigcup_{l=i}^{j} \alpha_l$, $\beta[i, j] = \bigcup_{l=i}^{j} \beta_l$, and $\alpha\beta[i, j] = \alpha[i, j] \cup \beta[i, j]$.

Lemma 13 *For every* $1 \leq i \leq j \leq k$,

1. *There are no* $s, t \in S$ *such that* $s \preceq t$, $s \in \beta_i \cap \beta_j$, *and* $t \in \alpha[i, j-1]$.
2. *There are no* $s, t \in S$ *such that* $s \preceq t$, $s \in \alpha_i \cap \alpha_j$, *and* $t \in \beta[i+1, j]$.

Proof To prove the first part of the lemma, take $s \in \beta_i \cap \beta_j$ and t in some α_l, where $i \leq l \leq j - 1$ (see Fig. 7). We have that (1) $s \in \gamma_i$ and $s \in \gamma_j$ (from Lemma 4), (2) $s \notin \alpha_l$ (from Lemma 6), (3) $t \in \alpha_l$. Therefore, by Lemma 12, $s \not\preceq t$.

The proof of the second part is symmetric to the proof of the firs part. Take $s \in \alpha_i \cap \alpha_j$ and t in some β_l, where $i + 1 \leq l \leq j$. Now, (1) $s \in \delta_i$ and $s \in \delta_j$ (from Lemma 4), (2) $s \notin \beta_l$ (from Lemma 6), (3) $t \in \beta_l$. Thus, $s \not\preceq t$. □

Corollary 2 *1. There are no indices* $i' < i \leq j' < j$ *such that* $\alpha_{i'} \cap \alpha_{j'} \neq \emptyset$ *and* $\beta_i \cap \beta_j \neq \emptyset$.
2. There are no indices $i' \leq i < j' \leq j$ *such that* $\beta_{i'} \cap \beta_{j'} \neq \emptyset$ *and* $\alpha_i \cap \alpha_j \neq \emptyset$.

Proof To prove the first part, assume conversely that there are such indices. Arbitrarily select $u \in \alpha_{i'} \cap \alpha_{j'}$ and $v \in \beta_i \cap \beta_j$. From Lemma 13 (part 1) applied on $v \in \beta_i \cap \beta_j$, $v \not\preceq u$, and from Lemma 13 (part 2) applied on $u \in \alpha_{i'} \cap \alpha_{j'}$, $u \not\preceq v$. The conditions of Lemma 13 hold since $i < i'$ and $j' < j$. The second part is proved using the same arguments. □

Lemma 14 *For every* $1 \leq i \leq j \leq k$,

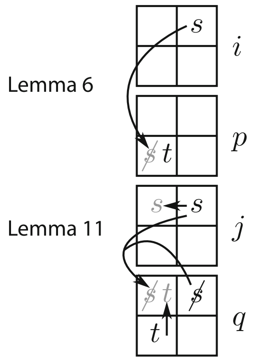


Fig. 8 An illustration of the proof of Lemma 14. The black s and $\not\preceq$ symbols are due to the conditions of case 1 of the lemma. The black t symbols are due to the assumption that there is a symbol $t \in \alpha[1, j-1] \cap \alpha\beta[j+1, k]$. The gray symbols represent values that are obtained using Lemmas 4, 6, and 11

1. *If* $s \in \beta_i \cap \beta_j$ *and* $s \notin \beta[j+1, k]$ *then* $\alpha[i, j-1] \cap \alpha\beta[j+1, k] = \emptyset$.
2. *If* $s \in \alpha_i \cap \alpha_j$ *and* $s \notin \alpha[1, i-1]$ *then* $\beta[i+1, j] \cap \alpha\beta[1, i-1] = \emptyset$.

Proof The proofs for the two parts of the lemma are similar, so we will prove only the first part. Assume conversely that the first part is not true, and pick $t \in \alpha[i, j-1] \cap \alpha\beta[j+1, k]$. Let $p \in [i, j-1]$ and $q \in [j+1, k]$ be the indices such that $t \in \alpha_p$ and $t \in \alpha_q \cup \beta_q$ (see Fig. 8). We will show a contradiction by showing that $s \not\preceq t$ and $t \not\preceq s$.

From Lemma 13 (part 1), $s \not\preceq t$ (since $t \in \alpha_p$ and $i \leq p \leq j-1$). We next show that $t \not\preceq s$ using Lemma 7 on $\alpha_p, \gamma_j, \gamma_q$. We have $t \in \alpha_p$ (by the definition of p), $s \notin \alpha_p$ (by Lemma 6), $s \in \gamma_j$ (by Lemma 4), $t \in \gamma_q$ (by Lemma 4), and $s \notin \gamma_q$ (by Lemma 11 applied on injuries j and q). Therefore, $t \not\preceq s$. □

The following Lemmas 15 and 16 use our previous results in order to show that k injuries in a column of $D^{\square}$ impose a certain constraint in D_{first}, that is, there are $k+1$ distinct symbols of S. Since $|S| = r+1$, this implies that $k \leq r$, thus proving our main result of this section, Theorem 4.

Lemma 15 *For every* $1 \leq i \leq j \leq k$,

1. *If* $s \in \beta_i \cap \beta_j$ *and* $s \notin \beta[j+1, k]$ *then there is a set* $T \subseteq \alpha[i, j-1] \cup \beta[i, j]$ *such that* $|T| = j-i+1$ *and* $T \cap \alpha\beta[j+1, k] = \emptyset$.
2. *If* $s \in \alpha_i \cap \alpha_j$ *and* $s \notin \alpha[1, i-1]$ *then there is a set* $T \subseteq \alpha[i, j] \cup \beta[i+1, j]$ *such that* $|T| = j-i+1$ *and* $T \cap \alpha\beta[1, i-1] = \emptyset$.

Proof We will prove only the first part of the lemma, as the second part is similar. The proof is done using induction on $j-i$. The main idea is to partition the interval $[i, j]$ into disjoint intervals, and then apply the induction hypothesis on each of these intervals to obtain sets $T_1, \ldots, T_p$. We will then show that the set $T = \{s\} \cup T_1 \cup \cdots \cup T_p$ satisfies the requirement stated in the lemma.

數學歸納法 Consider the base case of the induction, namely $j-i=0$. Let $T = \{s\}$. We have $s \notin \alpha[j+1, k]$ (from Lemma 6 and from the condition $s \in \beta_j$ of the lemma) and $s \notin \beta[j+1, k]$. Therefore, $T \cap \alpha\beta[j+1, k] = \emptyset$.

We next prove the induction step. Assume the correctness of the Lemma for $0 \leq j-i \leq q$, and consider i, j satisfying $j-i = q+1$. Construct a partition

$[j'_1, j_1], [j'_2, j_2], \ldots, [j'_p, j_p]$ of $[i, j-1]$ as follows (see Fig. 9a). Let $j_1 = j-1$, and let j'_1 be the minimum index such that $i \le j'_1 \le j_1$ and $\alpha_{j'_1} \cap \alpha_{j_1} \neq \emptyset$. Continue with this process on $[i, j'_1 - 1]$. For every $1 \le \tilde{p} \le p$, pick arbitrarily an element $t_{\tilde{p}} \in \alpha_{j'_{\tilde{p}}} \cap \alpha_{j_{\tilde{p}}}$. Note that for every $\tilde{p}$, the conditions of the second part of the lemma are satisfied for the symbol $t_{\tilde{p}}$ and injuries $j'_{\tilde{p}}, j_{\tilde{p}}$, since (1) $t_{\tilde{p}} \in \alpha_{j'_{\tilde{p}}} \cap \alpha_{j_{\tilde{p}}}$, (2) $t_{\tilde{p}} \notin \alpha[i, j'_{\tilde{p}} - 1]$ (by the definition of $j'_{\tilde{p}}$), and (3) $t_{\tilde{p}} \notin \alpha[1, i-1]$ [by Corollary 2 (part 1) applied on injuries $l, i, j_{\tilde{p}}, j$ for every $l \in [1, i-1]$, as shown in Fig. 9b]. Thus, we can use the induction hypothesis on every $t_{\tilde{p}}$ and obtain sets $T_1, \ldots, T_p$ such that for every $1 \le \tilde{p} \le p$, $T_{\tilde{p}} \subseteq \alpha[j'_{\tilde{p}}, j_{\tilde{p}}] \cup \beta[j'_{\tilde{p}} + 1, j_{\tilde{p}}]$, $|T_{\tilde{p}}| \ge j_{\tilde{p}} - j'_{\tilde{p}} + 1$, and $T_{\tilde{p}} \cap \alpha\beta[1, j'_{\tilde{p}} - 1] = \emptyset$. The sets $T_1, \ldots, T_p$ are disjoint, since for every $\tilde{p}_1 < \tilde{p}_2$ we have $T_{\tilde{p}_1} \cap \alpha\beta[1, j'_{\tilde{p}_1} - 1] = \emptyset$, $T_{\tilde{p}_2} \subseteq \alpha[j'_{\tilde{p}_2}, j_{\tilde{p}_2}] \cup \beta[j'_{\tilde{p}_2} + 1, j_{\tilde{p}_2}]$, and $j_{\tilde{p}_2} \le j'_{\tilde{p}_1} - 1$.

Define $T' = \cup_{\tilde{p}=1}^{p} T_{\tilde{p}}$. Since $T_1, \ldots, T_p$ are disjoint, $|T'| = \sum_{\tilde{p}=1}^{p} |T_{\tilde{p}}| = \sum_{\tilde{p}=1}^{p} (j_{\tilde{p}} - j'_{\tilde{p}} + 1) = j - i$. Note that $s \notin T'$, since for every $1 \le \tilde{p} \le p$, (1) $s \notin \alpha[j'_{\tilde{p}}, j_{\tilde{p}}]$ (by Lemma 6), (2) $s \notin \beta[j'_{\tilde{p}} + 1, j_{\tilde{p}}]$ [if $j'_{\tilde{p}} = j_{\tilde{p}}$ this property is trivial, and otherwise it follows from Corollary 2 (part 1) applied on the injuries $j'_{\tilde{p}}, l, j_{\tilde{p}}, j$ for every $l \in [j'_{\tilde{p}} + 1, j_{\tilde{p}}]$, as shown in Fig. 9c]. Thus, for $T = T' \cup \{s\}$, $|T| = j - i + 1$. It remains to prove that $T \cap \alpha\beta[j+1, k] = \emptyset$.

Take $x \in T$. If $x = s$ then we already shown that $s \notin \alpha\beta[j+1, k]$ in the proof of the base case of the induction. Otherwise ($x \neq s$), let $\tilde{p}$ be the index such that $x \in T_{\tilde{p}}$. There are two cases: either $x \in \alpha[j'_{\tilde{p}}, j_{\tilde{p}}]$ or $x \in \beta[j'_{\tilde{p}} + 1, j_{\tilde{p}}]$. In the former case, from Lemma 14 (part 1) we have $\alpha[i, j-1] \cap \alpha\beta[j+1, k] = \emptyset$, and in particular, $x \notin \alpha\beta[j+1, k]$. In the latter case, let $l \in [j'_{\tilde{p}} + 1, j_{\tilde{p}}]$ be the index such that $x \in \beta_l$. From Lemma 6, $x \notin \alpha[j+1, k]$. From Corollary 2 (part 1) applied on injuries $j'_{\tilde{p}}, l, j_{\tilde{p}}, l'$ for every $l' \in [j+1, k]$, $x \notin \beta[j+1, k]$ (see Fig. 9d). Therefore, $x \notin \alpha\beta[j+1, k]$. Since this is true for every x, we obtain that $T \cap \alpha\beta[j+1, k] = \emptyset$. □

Lemma 16 $|\alpha\beta[1, k]| \ge k + 1$.

數學歸納法

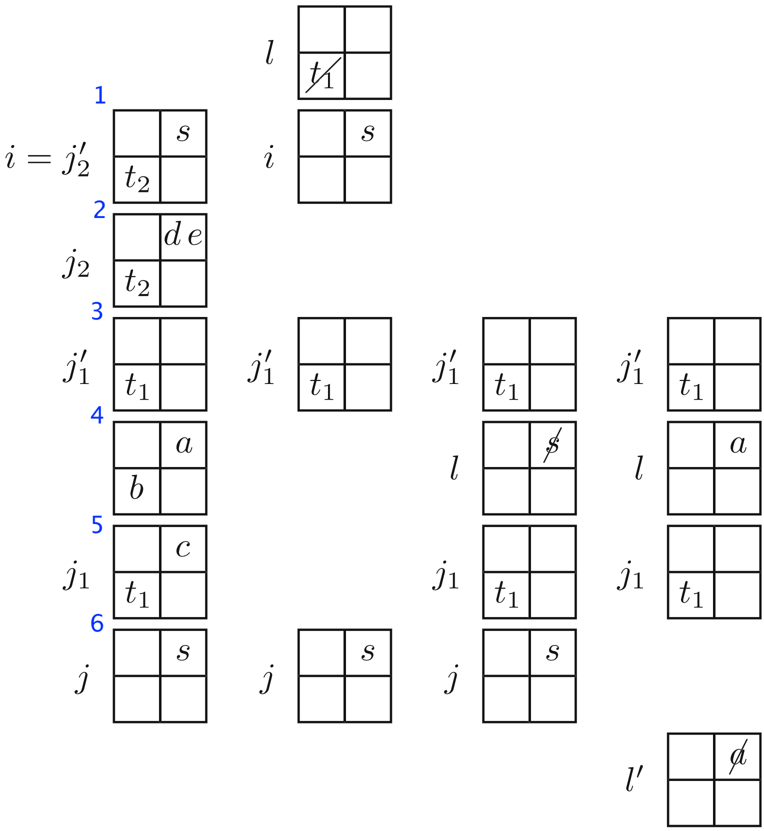
Proof We prove the lemma using induction on k. If $k = 1$ then since $\alpha_1 \cap \beta_1 = \emptyset$ (by Lemma 6), we get $|\alpha\beta[1, 1]| = |\alpha_1| + |\beta_1| \ge 2$. We now assume that $k > 1$. Let i be the maximum index such that $\beta_1 \cap \beta_i \neq \emptyset$ and let $s \in \beta_1 \cap \beta_i$.

There are two cases. The first case is when $i = k$. Choose $t \in \alpha_k$. From Lemma 6, $t \notin \beta[1, k]$ and from Corollary 2 (part 2), $t \notin \alpha[1, k-1]$. From the induction hypothesis $|\alpha\beta[1, k-1]| \ge k$, and since $t \notin \alpha\beta[1, k-1]$ we get $|\alpha\beta[1, k]| \ge k + 1$.

In the second case $i < k$. We apply Lemma 15 (part 1) and obtain a set $T \subseteq \alpha[1, i-1] \cup \beta[1, i]$ of size i. Let $k' = |\alpha\beta[i+1, k]|$. From the induction hypothesis $k' \ge k - i + 1$, and since $T \cap \alpha\beta[i+1, k] = \emptyset$, we conclude that $|\alpha\beta[1, k]| = i + k' \ge k + 1$. □

Using Lemma 16 we obtain our main theorem.

Theorem 4 *There are at most r injuries in every column of $D^{\square}$.*



(a) **(b)** **(c)** **(d)**

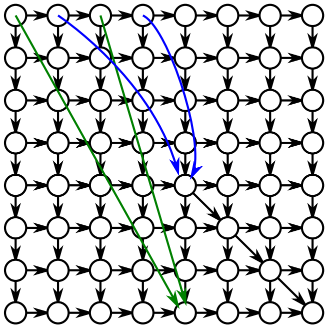
Fig. 9 An illustration of the proof of Lemma 15. **a** Shows an example for the induction step. Suppose that $i = 1$ and $j = 6$. In this example, we partition $[i, j-1] = [1, 5]$ into two intervals: $[\hat{j}_1, j_1] = [3, 5]$ and $[\hat{j}_2, j_2] = [1, 2]$ and pick $t_1 \in \alpha_3 \cap \alpha_5$ and $t_2 \in \alpha_1 \cap \alpha_2$. We then use induction to obtain sets $T_1 = \{a, b, c\} \subseteq \alpha[3, 5] \cup \beta[4, 5]$ and $T_2 = \{d, e\} \subseteq \alpha[1, 2] \cup \beta[2, 2]$. **b–d** Show the different applications of Corollary 2 in the proof

Proof If there are more than r injuries, then by Lemma 16, $|\alpha\beta[1, r+1]| \geq r+2$. However, $\alpha\beta[1, r+1]$ is a subset of S, and $|S| = r+1$. It follows that there are at most r injuries. □

6 Lower Bound

In this section we give a construction of grid graphs with r bridges in which the corresponding density matrices have $\Theta(n)$ rows and columns, each containing $\Theta(r)$ 1 values and $\Theta(r)$ -1 values. Such rows and columns will be called *saturated*. This construction shows that Property 2 is asymptotically tight.

For given r, n, with $r \leq n$, we build a graph $\mathrm{TG}_{r,n}$ as follows (see Fig. 10). The vertex set of the graph is $\{(i, j) : 0 \leq i \leq 2n-1, 0 \leq j \leq 2n-1\}$, and the set



(a) Grid graph

0	0	0	0	16	16	16	16
-1	0	0	0	12	13	14	15
-2	-1	0	0	8	8	8	8
-3	-2	-1	0	4	5	6	7
-4	-3	-2	-1	0	1	2	3
-5	-4	-3	-2	-1	0	1	2
-6	-5	-4	-3	-2	-1	0	1
-7	-6	-5	-4	-3	-2	-1	0

(b) D

1			-4	1	1	1
	1		-4	-1	-1	-1
		1	-4	1	1	1
			-3			

(c) $D^{\square}$

Fig. 10 An example showing the grid graph $\text{TG}_{4,4}$, its all scores matrix D, and the density matrix $D^{\square}$. The gray region in $D^{\square}$ consists of rows of ones and rows of minus ones

of diagonal edges is $\{(k, k+1) : n \le k < 2n-1\}$. The bridges of the graph are $f_0, f_1, \ldots, f_{r-1}$ where $f_k = ((0, k), (2n-1, n))$ if k is even and $f_k = ((0, k), (n, n))$ if k is odd. The weight of the bridge f_k is $W_{f_k} = n^2 - kn$.

Consider some grid graph $\text{TG}_{r,n}$ and let D be its all scores matrix. We next show that the density matrix $D^{\square}$ has $\Theta(n)$ saturated columns.

Fix $i < r$ and $j \ge n$, and note that: (1) For every $k < k'$, $W_{f_k} - W_{f_{k'}} = (k' - k)n \ge n$, (2) The number of diagonal edges is $n-1$, (3) For every k, $W_{f_k} \ge n-1$. Thus, the optimal path from $(0, i)$ to $(2n-1, j)$ utilizes the bridge f_i. Therefore, $D[i, j] = n^2 - in$ if i is even and $D[i, j] = n^2 - in + (j - n)$ if i is odd, (since in the latter case, the bridge's endpoint is (n, n), and thus diagonal edges can be utilized). Finally, we have that $D[i, j+1] - D[i, j] = 0$ if i is even and $D[i, j+1] - D[i, j] = 1$ if i is odd. We conclude that for every $i < r$ and $j > n$, $D^{\square}[i, j] = 1$ if i is odd and $D^{\square}[i, j] = -1$ if i is even.

Consider the graph $\text{TG}^2_{r,n}$ that is obtained by reversing the edges of $\text{TG}_{r,n}$ and mapping vertex (i, j) to $(2n-1-j, 2n-1-i)$. Row k in the all scores matrix of $\text{TG}^2_{r,n}$ is equal to column $2n-1-k$ in the all scores matrix of $\text{TG}_{r,n}$. Therefore, the density matrix of $\text{TG}^2_{r,n}$ has $\Theta(n)$ saturated rows. Consider now the graph $\text{TG}^3_{r,n}$ that is obtained by concatenating the graph $\text{TG}^2_{r,n}$ to the right of $\text{TG}_{r,n}$, and denote by D_3 its all scores matrix. The submatrices $D_3[0..2n-1, 0..2n-1]$, $D_3[2n..4n-1, 2n..4n-1]$ are equal to the all scores matrices of $\text{TG}_{r,n}$, $\text{TG}^2_{r,n}$, respectively. Thus we have that $\text{TG}^3_{r,n}$ has $\Theta(n)$ saturated rows and $\Theta(n)$ saturated columns.

7 Conclusions

We considered all scores matrices of grid graphs extended with a set of r additional edges, and studied their combinatorial implications on all scores matrices, focusing on the Monge property of such graphs. Our main observation is that the number of non-zero values in a density matrix corresponding to such graph is $O(rn)$. Thus, if $r = o(n)$, the all scores matrix is "almost Monge". Furthermore, we showed that all the non-zero values in the density matrix are confined to -1 or 1, except for the values in $O(r^2)$ specific locations in the matrix. Based on this analysis, we gave an algorithm for computing the all scores matrix, for grid graphs extended with r bridges, in $O(r(n^2 + nm))$ time.

We note that in some applications, the all scores matrix also includes the optimal alignment scores between every suffix of A and every prefix of B, and between every prefix of A and every suffix of B. Our results generalize to these applications as well.

An interesting challenge would be to extend our analysis and approaches to other applicative problem domains where "almost Monge" matrices can be identified.

Acknowledgements We thank the anonymous reviewers, that helped us improve the readability of the paper through their many helpful suggestions. The research of A.C and D.T was partially supported by ISF Grant No. 981/11. The research of A.C and M.Z-U was partially supported by ISF Grant 179/14.

References

1. Aggarwal, A., Klawe, M.M., Moran, S., Shor, P., Wilber, R.: Geometric applications of a matrix-searching algorithm. Algorithmica **2**(1–4), 195–208 (1987)
2. Alves, C.E.R., Cáceres, E.N., Song, S.W.: An all-substrings common subsequence algorithm. Discrete Appl. Math. **156**(7), 1025–1035 (2008)
3. Apostolico, A., Atallah, M.J., Larmore, L.L., McFaddin, S.: Efficient parallel algorithms for string editing and related problems. SIAM J. Comput. **19**(5), 968–988 (1990)
4. Apostolico, A.: String editing and longest common subsequences. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, pp. 361–398. Springer (1997)
5. Arslan, A.N.: Sequence alignment guided by common motifs described by context free grammars. In: Biotechnology and Bioinformatics Symposium (BIOT-2007). Colorado Springs, CO (2007)
6. Benson, G.: A space efficient algorithm for finding the best nonoverlapping alignment score. Theor. Comput. Sci. **145**(1&2), 357–369 (1995)
7. Bodlaender, H.L., Fellows, M.R., Evans, P.A.: Finite-state computability of annotations of strings and trees. In: Proceedings of the 7th Symposium on Combinatorial Pattern Matching (CPM), pp. 384–391. Springer (1996)
8. Burkard, R.E., Klinz, B., Rudolf, R.: Perspectives of monge properties in optimization. Discrete Appl. Math. **70**(2), 95–161 (1996)
9. Cabello, S., Chambers, E.W., Erickson, J.: Multiple-source shortest paths in embedded graphs. SIAM J. Comput. **42**(4), 1542–1571 (2013)
10. Comet, J.-P., Henry, J.: Pairwise sequence alignment using a prosite pattern-derived similarity score. Comput. Chem. **26**(5), 421–436 (2002)
11. Crochemore, M., Landau, G.M., Ziv-Ukelson, M.: A sub-quadratic sequence alignment algorithm for unrestricted cost matrices. SIAM J. Comput. **32**(5), 1654–1673 (2003)
12. Eisenstat, D., Klein, P.N.: Linear-time algorithms for max flow and multiple-source shortest paths in unit-weight planar graphs. In: Proceedings of the 45th ACM Symposium on Theory of Computing (STOC), pp. 735–744 (2013)
13. Evans, P.A.: Algorithms and Complexity for Annotated Sequence Analysis. Ph.D. thesis, Citeseer (1999)
14. Hermelin, D., Landau, G.M., Landau, S., Weimann, O.: A unified algorithm for accelerating edit-distance computation via text-compression. In: Proceedings of the 26th Symposium on Theoretical Aspects of Computer Science (STACS), pp. 529–540 (2009)
15. Hou, X., Prékopa, A.: Monge property and bounding multivariate probability distribution functions with given marginals and covariances. SIAM J. Optim. **18**(1), 138–155 (2007)
16. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., De Castro, E., Langendijk-Genevaux, P.S., Pagni, M., Sigrist, C.: The prosite database. Nucleic Acids Res. **34**(suppl 1), D227–D230 (2006)
17. Ishida, Y., Inenaga, S., Shinohara, A., Takeda, M.: Fully incremental LCS computation. In: Proceedings of the 15th Symposium on Fundamentals of Computation Theory (FCT), pp. 563–574 (2005)

18. Kannan, S., Myers, E.W.: An algorithm for locating nonoverlapping regions of maximum alignment score. SIAM J. Comput. **25**(3), 648–662 (1996)
19. Kent, C., Landau, G.M., Ziv-Ukelson, M.: On the complexity of sparse exon assembly. J. Comput. Biol. **13**(5), 1013–1027 (2006)
20. Kim, S.-R., Park, K.: A dynamic edit distance table. J. Discrete Algorithms **2**(2), 303–312 (2004)
21. Klein, P.N.: Multiple-source shortest paths in planar graphs. In: Proceedings of the 16th Symposium on Discrete Algorithms (SODA), vol. 5, pp. 146–155 (2005)
22. Krusche, P., Tiskin, A.: String comparison by transposition networks, In: Proceedings of the of London Algorithmics, Workshop, pp. 184–204 (2008)
23. Krusche, P., Tiskin, A.: New algorithms for efficient parallel string comparison. In: Proceedings of the 22nd Symposium on Parallel Algorithms and Architectures (SPAA), pp. 209–216 (2010)
24. Landau, G.M., Myers, E.W., Schmidt, J.P.: Incremental string comparison. SIAM J. Comput. **27**(2), 557–582 (1998)
25. Landau, G.M., Myers, E.W., Ziv-Ukelson, M.: Two algorithms for LCS consecutive suffix alignment. J. Comput. Syst. Sci. **73**(7), 1095–1117 (2007)
26. Landau, G.M., Schieber, B., Ziv-Ukelson, M.: Sparse LCS common substring alignment. Inf. Process. Lett. **88**(6), 259–270 (2003)
27. Landau, G.M., Schmidt, J.P., Sokol, D.: An algorithm for approximate tandem repeats. J. Comput. Biol. **8**(1), 1–18 (2001)
28. Landau, G.M., Ziv-Ukelson, M.: On the common substring alignment problem. J. Algorithms **41**(2), 338–359 (2001)
29. Matarazzo, U., Tsur, D., Ziv-Ukelson, M.: Efficient all path score computations on grid graphs. Theor. Comput. Sci. **525**, 138–149 (2014)
30. Mozes, S., Tsur, D., Weimann, O., Ziv-Ukelson, M.: Fast algorithms for computing tree lcs. Theor. Comput. Sci. **410**(43), 4303–4314 (2009)
31. Park, J.K.: A special case of the n-vertex traveling-salesman problem that can be solved in o (n) time. Inf. Process. Lett. **40**(5), 247–254 (1991)
32. Sakai, Y.: An almost quadratic time algorithm for sparse spliced alignment. Theory Comput. Syst. **48**(1), 189–210 (2011)
33. Schmidt, J.P.: All highest scoring paths in weighted grid graphs and their application to finding all approximate repeats in strings. SIAM J. Comput. **27**(4), 972–992 (1998)
34. Tiskin, A.: Semi-local longest common subsequences in subquadratic time. J. Discrete Algorithms **6**(4), 570–581 (2008)
35. Tiskin, A.: Semi-local string comparison: algorithmic techniques and applications. Math. Comput. Sci. **1**(4), 571–603 (2008)