ELSEVIER

Information Processing Letters 68 (1998) 3–9

Information Processing Letters

An efficient algorithm for on-line searching of minima in Monge path-decomposable tridimensional arrays ☆

Alfredo García *, Pedro Jodrá [1], Javier Tejel [2]

Dpto. Métodos Estadísticos, Facultad de Ciencias, Universidad de Zaragoza, 50009 Zaragoza, Spain

Received 5 March 1998; received in revised form 10 August 1998
Communicated by T. Lengauer

Abstract

We consider the problem of computing the recurrence $E[i] = \min_{j=1,\ldots,m} \min_{1 \leqslant k \leqslant i} \{b(i,j) + c(j,k) + E[k-1]\}$, $i = 1, \ldots, n$, where $E[0]$ is known and $B = \{b(i,j)\}$ and $C = \{c(j,k)\}$ are known weight Monge matrices of size $n \times m$ and $m \times n$, respectively. We provide an $\Theta(m+n)$-algorithm for calculating the $E[i]$ values. This algorithm allows us to linearly solve the two following problems: Finding the minimum Hamiltonian curve from point p_1 to point p_m for N points on a convex polygon, and solving the traveling salesman problem for N points on a convex polygon and a segment line inside it, improving the previous $\Theta(N \log N)$-algorithms for both these problems.

Keywords: Monge matrices; Dynamic programming; Computational geometry; Computational complexity; Traveling salesman problem

1. Introduction

Let $W = \{w(i,k)\}$ be an $n \times n'$ weight matrix, where each $w(i,k)$ can be calculated in constant time. Given an integer constant $c_1 \geqslant 1$ and the values $F[k]$, for $k = 1, \ldots, c_1$, the so-called one-dimensional dynamic programming problem consists in solving the on-line recurrence:

$$E[i] = \min_{1 \leqslant k \leqslant c_i} \{w(i,k) + F[k]\}, \quad i = 1, \ldots, n, \quad (1)$$

where c_i and $F[k]$, for $k = c_{i-1}+1, \ldots, c_i$, can be computed from $E[i-1]$ in constant time, and the integer constants $c_1, \ldots, c_n$ verify $1 \leqslant c_1 \leqslant \cdots \leqslant c_n \leqslant n'$.

Problems of this type, usually with the weight matrix verifying some additional property, arise in many fields: biology [8], economics [3], operation research [9,7], computational geometry [1], etc.

Solving (1) is equivalent to the problem of on-line searching of the minimum of each row of the partial matrix A with entries $\{a(i,k) = w(i,k) + F(k)\}$, defined when $k \leqslant c_i$. A partial matrix of this shape is called a generalized lower triangular matrix, and it is concave totally monotone if $a(i,k) > a(i,k')$ implies $a(i',k) > a(i',k')$, for $1 \leqslant i < i' \leqslant n$, $1 \leqslant k < k' \leqslant n'$, when these four entries are defined. If we denote by $k(i)$ the smallest column index where the minimum of row i is found, the main property of concave totally monotone matrices is that $k(1) \leqslant k(2) \leqslant \cdots \leqslant k(n)$ (monotonicity), and this property is also verified by the minima of any submatrix (total

☆ Partially supported by University of Zaragoza, Spain. Project UZ96-CIENT-09.

* Corresponding author. Email: olaverri@posta.unizar.es.

[1] Email: pjodra@posta.unizar.es.

[2] Email: jtejel@posta.unizar.es.

0020-0190/98/$ – see front matter
PII: S0020-0190(98)00137-9

monotonicity). For concave totally monotone matrices there are several $\Theta(n+n')$-algorithms which solve the on-line row minima search problem. We will use the one by Larmore and Schieber, described in [8], and we will call it the LARSCH algorithm.

The most frequent case of concave totally monotone matrices is that of Monge matrices. A full $n \times n'$ matrix A is Monge if

$$a(i,j)+a(i+1,j+1) \leqslant a(i,j+1)+a(i+1,j) \quad \text{for } 1 \leqslant i < n \text{ and } 1 \leqslant j < n'.$$

For a review on Monge properties and applications see [4]. The following properties of Monge matrices can be easily checked:

(i) the transpose of a Monge matrix is Monge,
(ii) the sum of two Monge matrices is a Monge matrix,
(iii) if A with elements $\{a(i,j)\}$ is Monge, $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^{n'}$, also B with elements $\{b(i,j) = a(i,j)+u_i+v_j\}$ is Monge, and
(iv) any generalized lower triangular submatrix of a Monge matrix is concave totally monotone.

The main result in this paper is a linear time algorithm solving the extension of problem (1) when the weight matrix is given by

$$w(i,k) = \min_{j=1,\dots,m} \{b(i,j)+c(j,k)\},$$

where $B = \{b(i,j)\}$ and $C = \{c(j,k)\}$ are known Monge matrices of size $n \times m$ and $m \times n'$, respectively. In the applications described later, c_i will be i, n' will be n and $F[k]$ will be $E[k-1]$. For this reason, we explain the algorithm only for this particular case, although it can be modified for solving the more general formulation. Then, the problem is that of calculating

$$E[i] = \min_{j=1,\dots,m} \min_{k=1,\dots,i} \{b(i,j)+c(j,k)+E[k-1]\}, \quad i = 1,\dots,n, \tag{2}$$

where $E[0]$ is given. In [2], a tridimensional array that can be expressed as $b(i,j)+c(j,k)$ is called a Monge path-decomposable tridimensional array. So, our problem is the on-line calculating of the minima of a partial (because $k \leqslant i$) Monge path-decomposable tridimensional array.

2. On-line algorithm

In order to solve (2), we use mainly the $m \times n$ Monge matrix

$$\overline{C} = \{\bar{c}(j,k) = c(j,k)+E[k-1]\}.$$

Given i, let $\overline{C}^i$ be the $m \times i$ submatrix formed by the first i columns of $\overline{C}$ and let A^i be the matrix obtained by adding $b(i,j)$ to each row j of $\overline{C}^i$. $E[i]$ is the global minimum of A^i and we will denote by $J(i)$ and $K(i)$ the row and column of A^i where this minimum is, i.e., $E[i] = a^i(J(i),K(i)) = b(i,J(i))+\bar{c}(J(i),K(i))$. For each i, we want to calculate $(J(i),K(i))$ and the key idea of our algorithm is that only $O(m+n)$ candidate positions (j,k) need to be considered for calculating $(J(i),K(i))$, $\forall i$. In addition, all the matrices involved in the algorithm are concave totally motonones.

Let $k^i(j)$, $j = 1,\dots,m$, be the column with the minimum of row j of A^i, and let $j^i(k)$, $k = 1,\dots,i$, be the row with the minimum of column k of A^i. Notice that $k^i(j)$ is also the position of the minimum of the row j of $\overline{C}^i$ because row j of A^i is obtained from the same row of $\overline{C}^i$ by adding the constant $b(i,j)$. When a row has more than one minimum, we will always take the one in the leftmost column. Similarly for columns, we will take the one in the topmost row and as a global minimum, we will take the one in the leftmost column and topmost row. The following lemma gives some additional properties about these row and column minima.

Lemma 1.
(1) $k^i(j) \leqslant k^i(j+1)$, $j = 1,\dots,m-1$.
(2) $j^i(k) \leqslant j^i(k+1)$, $k = 1,\dots,i-1$.
(3) $k^i(j) \leqslant k^{i+1}(j)$, $i = 1,\dots,n-1$.
(4) $j^i(k) \leqslant j^{i+1}(k)$, $i = k,\dots,n-1$.

Proof. Given i, A^i and its transpose are Monge matrices, hence (1) and (2) hold by monotonicity. On the other hand, as $\overline{C}^{i+1}$ is obtained from $\overline{C}^i$ by adding column $i+1$ of $\overline{C}$, $k^{i+1}(j)$ is either $k^i(j)$ or $i+1$, hence (3). Finally, given $k \leqslant i$, column k of A^{i+1} is obtained from column k of A^i by adding $b(i+1,j)-b(i,j)$ to each entry. As B is Monge, $b(i+1,j+1)-b(i,j+1) \leqslant b(i+1,j)-b(i,j)$, for $j = 1,\dots,m-1$, so, what is added is a decreasing

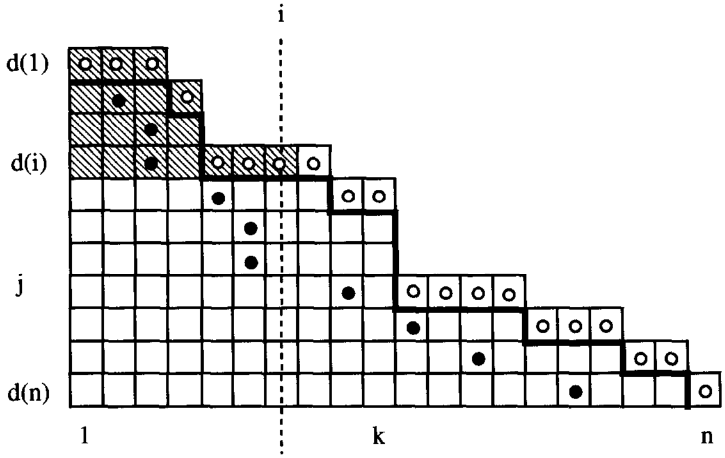

Fig. 1. Matrices $\overline{C}'$ and H.

amount in each row. Therefore, the minimum of the column k of A^{i+1} has to be in or after $j^i(k)$. □

The global minimum of A^i is always the minimum in its row and in its column, so we have $K(i) = k^i(j)$ for an index $j \leqslant m$, and $J(i) = j^i(k)$ for an index $k \leqslant i$. Hence, by previous lemma $J(i) \leqslant j^i(i)$. These values $j^i(i)$, $i = 1, \ldots, n$, can be precalculated as follows: let D be the $n \times m$ Monge matrix defined as $d(i, j) = b(i, j) + c(j, i)$ and let $d(i)$ be the column where row i of D has its minimum. Column i of A^i is obtained by adding the constant $E[i-1]$ to each element of row i of D, so $d(i) = j^i(i)$.

Lemma 2. *$K(i) \leqslant i$ and $d(K(i)) \leqslant J(i) \leqslant d(i)$ for $i = 1, \ldots, n$.*

Proof. Obviously, $K(i) \leqslant i$ because A^i is an $m \times i$ matrix. In addition, $J(i) = j^i(k)$ for a $k \leqslant i$, hence by Lemma 1, part (2),

$$J(i) \leqslant j^i(i) = d(i).$$

Similarly, as $K(i) \leqslant i$, using (4) of Lemma 1 we have:

$$d\big(K(i)\big) = j^{K(i)}\big(K(i)\big) \leqslant j^i\big(K(i)\big) = J(i). \qquad \square$$

In Fig. 1 the lemma is illustrated. Let $\overline{C}'$ be the partial matrix formed by the elements $\bar{c}(j, k)$ such that $d(k) \leqslant j \leqslant d(n)$, for $1 \leqslant k \leqslant n$. Given i, the previous lemma implies that the position $(J(i), K(i))$ is in R_i (shaded region in Fig. 1) defined as

$$R_i = \big\{(j, k) \ni d(k) \leqslant j \leqslant d(i),\ 1 \leqslant k \leqslant i\big\}.$$

Now, let H be the partial matrix of $\overline{C}$ defined as $h(j, k) = c(j, k) + E[k-1]$ if $d(k) < j \leqslant d(n)$, for $1 \leqslant k \leqslant n$ (submatrix delimited by the thick line in Fig. 1). Notice that each row j of H, $j = d(1) + 1, \ldots, d(n)$, is defined until the column $\max\{k \ni d(k) < j\}$ and that H is concave totally monotone. Let $h(j)$ be the position where the minimum of row j of H is achieved. Then, the following lemma shows that it suffices to consider $O(m+n)$ candidates for the positions of all the global minima.

Lemma 3. *Either $J(i) = d(K(i))$ or $K(i) = h(J(i))$ for $i = 1, \ldots, n$.*

Proof. Given i, we know that $K(i)$ is the column where the minimum of row $J(i)$ of $\overline{C}^i$ is found and, by Lemma 2, that $(J(i), K(i)) \in R_i$. Then, either the minimum of row $J(i)$ is in H, and hence $K(i) = h(J(i))$, or $K(i) \leqslant i$ is a column such that $J(i) = d(K(i))$. □

In Fig. 1, the positions $(j, h(j))$ of H are marked with a black dot and the positions $(d(i), i)$ with a white one. Lemmas 2 and 3 imply that, given i, candidate positions for containing $(J(i), K(i))$ are only those belonging to R_i and marked with a black or a white dot.

Now, let m' be the number of these positions that can contain a global minimum in any step i. For the moment, we assume that all of them have been calculated, i.e., we know where a black or white dot appears in $\overline{C}'$. We can enumerate these positions from 1 to m' beginning with the first row and then in each row, from left to right. If (j,k) is the candidate position with number p, $1 \leqslant p \leqslant m'$, we define $row(p) = j$ and $col(p) = k$. Given i, $E[i]$ is achieved in one candidate position of R_i, so, if $l(i)$ is the number of candidate positions in R_i, then

$$E[i] = \min_{1 \leqslant p \leqslant l(i)} b\big(i, row(p)\big) + \bar{c}\big(row(p), col(p)\big).$$

Therefore, if we define A' as the $n \times m'$ partial matrix with elements

$$a'(i,p) = b\big(i, row(p)\big) + c\big(row(p), col(p)\big) + E\big[col(p) - 1\big]$$
$$\text{for } i = 1, \ldots, n, \ p = 1, \ldots, l(i),$$

then $E[i]$ is the minimum of row i of A'.

Lemma 4. *The $n \times m'$ matrix A' is concave totally monotone.*

Proof. The values $l(i)$ are non decreasing, so A' is a generalized lower triangular matrix. We only need to prove that $a'(i,p) + a'(i+1,p+1) \leqslant a'(i+1,p) + a'(i,p+1)$ when these four entries are defined, which is equivalent to proving that $b(i, row(p)) + b(i+1, row(p+1)) \leqslant b(i+1, row(p)) + b(i, row(p+1))$. This last inequality is true because $row(p+1)$ is either $row(p)$ or $row(p)+1$, and B is Monge. □

Now, we are ready to solve the initial problem (2). In step i, let us assume that $E[1], \ldots, E[i-1]$ and $h(d(1)+1), \ldots, h(d(i-1))$ have been calculated. Then, the first i columns of $\overline{C}$ and H are defined. Hence, the rows $d(i-1)+1, \ldots, d(i)$ of H are also known and, if $d(i-1) < d(i)$, then we can calculate $h(d(i-1)+1), \ldots, h(d(i))$. In order to calculate $E[i]$, we need to know all the positions in R_i with black and white dots and the value of $a'(i,p)$ in these positions. This obviously can be done because the first i columns of $\overline{C}$ are defined and because $h(d(1)+1), \ldots, h(d(i))$ are known at step i. The code in Fig. 2 shows a way of doing the calculations.

Algorithm MINIMA
begin
  activate LARSCH over each row of D, obtaining
    $d(i), i = 1, \ldots, n$.
  initialize $p = 0$; $d(0) = d(1)$;
  for i **from** 1 **to** n **do**
    comment The i first columns of $\overline{C}$ are defined.
    if $(d(i) > d(i-1))$ **then**
      comment The rows from $d(i-1)+1$ to $d(i)$ of H are defined.
      for j **from** $d(i-1)+1$ **to** $d(i)$ **do**
        activate LARSCH over H for calculating $h(j)$
        $p = p+1$; $row(p) = j$; $col(p) = h(j)$;
      end for
    end if
    $p = p+1$; $row(p) = d(i)$; $col(p) = i$; $l(i) = p$;
    comment The first $l(i)$ columns of A' are defined.
    comment The row i of A' is complete.
    activate LARSCH over A' for calculating $E[i]$.
  end for
end

Fig. 2. The on-line algorithm.

Lemma 5. *The complexity of algorithm MINIMA is $\Theta(m+n)$.*

Proof. The algorithm LARSCH is activated for calculating the minima of D, H, and A'. D is a $n \times m$ full matrix, H is a partial matrix of, at most, size $(m-1) \times (n-1)$ and A' is a $n \times m'$ partial matrix, but $m' = n + d(n) - d(1) \leqslant n + m - 1$. As each entry of these matrices can be calculated in constant time, all these minima are obtained in $\Theta(m+n)$ steps. □

3. Applications

3.1. Finding the minimum Hamiltonian curve in a convex polygon

Given $N = m + n$ points on the plane forming a convex polygon P, we want to find the minimum Hamiltonian curve S, starting at point p_1 and finishing at point p_m, where p_1 and p_m are arbitrarily chosen. In [7], this problem is reformulated as one of dynamic

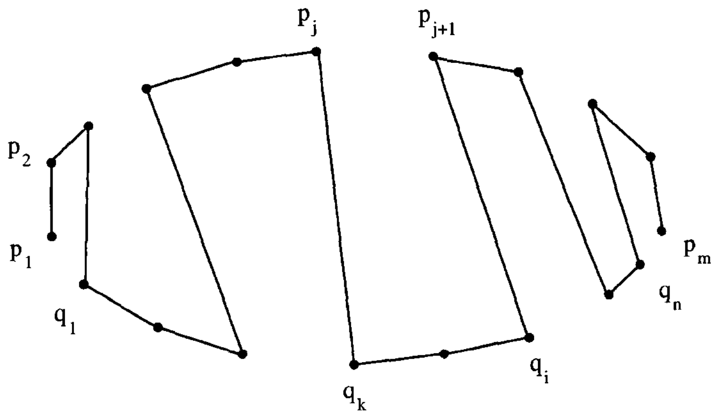

Fig. 3. The minimum Hamiltonian curve.

programming and an $O(N \log N)$ time and $O(N)$ space algorithm is given for solving it.

This reformulation is the following. Let $U_P = \{p_1, \ldots, p_m\}$ be the set of points on P between p_1 and p_m (both included), clockwise numbered, and $L_P = \{q_1, \ldots, q_n\}$ the set of points on P between p_1 and p_m (both excluded), counterclockwise numbered (see Fig. 3). Let $d(\cdot,\cdot)$ be the Euclidean distance between two points. Let $S(i)$ be the shortest Hamiltonian curve from p_1 to p_m visiting all the points of U_P and only the first i points of L_P and let $E[i]$ be the length of $S(i)$.

If we define

$$E[0] = \sum_{j=1}^{m-1} d(p_j, p_{j+1}),$$

then the following scheme of dynamic programming holds (see [7]):

$$E[i] = \min_{j=1,\ldots,m-1} \min_{1 \leqslant k \leqslant i} \Bigg\{ E[k-1] + d(p_j, q_k) + \sum_{l=k}^{i-1} d(q_l, q_{l+1}) + d(q_i, p_{j+1}) - d(p_j, p_{j+1}) \Bigg\}, \quad i = 1, \ldots, n.$$

By defining:

- $s(q_i) = \sum_{k=1}^{i-1} d(q_k, q_{k+1})$, for $1 \leqslant i \leqslant n$,
- $b(i, j) = s(q_i) + d(q_i, p_{j+1}) - d(p_j, p_{j+1})$, for $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m-1$,
- $c(j, k) = d(p_j, q_k) - s(q_k)$, for $1 \leqslant j \leqslant m-1$ and $1 \leqslant k \leqslant n$,

then the previous scheme is equivalent to:

$$E[i] = \min_{j=1,\ldots,m-1} \min_{k \leqslant i} \{ b(i, j) + c(j, k) + E[k-1] \}, \quad i = 1, \ldots, n. \tag{3}$$

Quantities $s(i)$, $i = 1, \ldots, n$, can be easily computed in $O(n)$ time and stored in $O(n)$ space and hence, given i and j, we can calculate $b(i, j)$ and $c(j, i)$ in constant time. In addition, matrices B and C are Monge and then, the on-line algorithm of the previous section can be applied directly.

3.2. *The convex-polygon-and-line TSP*

Now, we are interested in calculating the minimum tour that visits $N = m + n$ points, when m of them are on a convex polygon P and the other n are on a segment line SL inside P. In [7], this problem is solved in $O((m+n)\log n)$ time and $O(n)$ space, improving the previous $O(N^2)$-algorithm for this problem described in [6]. A simpler version of this problem, for points on three parallel lines, was studied by Cutler (see [5]) in relation to the problem of connecting nets in printed circuits and he solved it in $O(N^3)$ time.

In [7], a similar reformulation to the previous one is given for solving the convex-polygon-and-line TSP. Let CP_U (the upper convex polygon) and CP_L (the lower convex polygon) be the set of points $\{p_0, p_1, \ldots, p_{m_1+1}\}$ and the set of points $\{r_0, r_1, \ldots,$

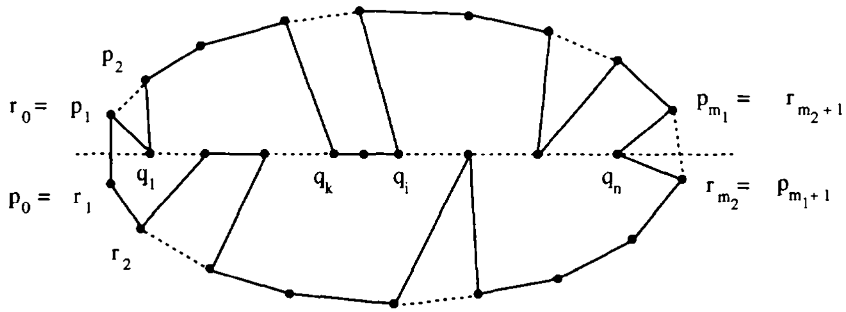

Fig. 4. The convex-polygon-and-line TSP.

$r_{m_2+1}\}$ clockwise and counterclockwise numbered, respectively (see Fig. 4). Notice that $p_1 \in CP_U$ is redefined as $r_0 \in CP_L$ because sometimes it must be considered as belonging to CP_L. The same happens with points r_1, p_{m_1} and r_{m_2}. $V_{SL} = \{q_1, \ldots, q_n\}$ is the set of points on SL numbered from left to right. Let $S(i)$ be the shortest tour that visits all the points of P and only the first i points of SL and let $G[i]$ be its length.

Now, let $E[i]$ and $F[i]$, respectively, be the length of the shortest tour that visits all the points of P and only the first i points of SL, with the constraint that the last zone of SL, a zone $[q_k, q_i]$, is linked with two points of CP_U and with two points of CP_L. Let us define

$$G[0] = \sum_{j=0}^{m_1} d(p_j, p_{j+1}) + \sum_{j=1}^{m_2-1} d(r_j, r_{j+1})$$

(the perimeter of P). Then, the following scheme of dynamic programming holds (see [7]):

$$E[i] = \min_{j=0,\ldots,m_1} \min_{1 \leqslant k \leqslant i} \left\{ \begin{array}{l} G[k-1] + d(q_k, p_j) \\ + d(q_k, q_i) + d(q_i, p_{j+1}) \\ - d(p_j, p_{j+1}) \end{array} \right\},$$
$$i = 1, \ldots, n,$$

$$F[i] = \min_{j=0,\ldots,m_2} \min_{1 \leqslant k \leqslant i} \left\{ \begin{array}{l} G[k-1] + d(q_k, r_j) \\ + d(q_k, q_i) + d(q_i, r_{j+1}) \\ - d(r_j, r_{j+1}) \end{array} \right\},$$
$$i = 1, \ldots, n,$$

$$G[i] = \min\{E[i], F[i]\}, \quad i = 1, \ldots, n,$$

where $d(\cdot, \cdot)$ is the Euclidean distance between two points except that we define

$$d(q_k, p_0) = d(q_k, r_0) = \infty \quad \text{if } k \neq 1, \quad \text{and}$$
$$d(q_k, p_{m_1+1}) = d(q_k, r_{m_2+1}) = \infty \quad \text{if } k \neq n.$$

If we define:

- $s(q_i) = d(q_1, q_i)$, for $1 \leqslant i \leqslant n$,
- $b(i, j) = s(q_i) + d(q_i, p_{j+1}) - d(p_j, p_{j+1})$, for $i = 1, \ldots, n$ and $j = 0, \ldots, m_1$,
- $c(j, k) = d(q_k, p_j) - s(q_k)$, for $j = 0, \ldots, m_1$ and $k = 1, \ldots, n$,
- $b'(i, j) = s(q_i) + d(q_i, r_{j+1}) - d(r_j, r_{j+1})$, for $i = 1, \ldots, n$ and $j = 0, \ldots, m_2$,
- $c'(j, k) = d(q_k, r_j) - s(q_k)$, for $j = 0, \ldots, m_2$ and $k = 1, \ldots, n$

then we have:

$$\begin{aligned} E[i] &= \min_{j=0,\ldots,m_1} \min_{1 \leqslant k \leqslant i} \{G[k-1] \\ &\quad + b(i, j) + c(j, k)\}, \quad i = 1, \ldots, n, \\ F[i] &= \min_{j=0,\ldots,m_2} \min_{1 \leqslant k \leqslant i} \{G[k-1] \\ &\quad + b'(i, j) + c'(j, k)\}, \quad i = 1, \ldots, n, \\ G[i] &= \min\{E[i], F[i]\}, \quad i = 1, \ldots, n. \end{aligned} \tag{4}$$

Each element of the matrices B, B', C and C' can be calculated in constant time and these matrices are all Monge. Then, by interleaving the computation of $E[i]$, $F[i]$ and $G[i]$, the algorithm of the previous section can be used to solve the convex-polygon-and-line TSP in $\Theta(N)$ time and $\Theta(N)$ space.

References

[1] A. Aggarwal, M.M. Klawe, S. Moran, P. Shor, R. Wilber, Geometric applications of a matrix searching algorithm, Algorithmica 2 (1987) 195–208.

[2] A. Aggarwal, J.K. Park, Notes on searching in multidimensional monotone arrays, in: Proceedings 29th IEEE Symp. on Foundations of Computer Science, October 1988, pp. 497–512.

[3] A. Aggarwal, J.K. Park, Improved algorithms for economic lot-size problems, Oper. Res. 41 (1993) 549–571.

[4] R.E. Burkard, B. Klinz, R. Rudolf, Perspectives of Monge properties in optimization, Discrete Appl. Math. 70 (1996) 95–161.

[5] M. Cutler, Efficient special case algorithms for the N-line planar traveling salesman problem, Networks 10 (1980) 183–195.

[6] V.G. Deineko, R. van Dal, G. Rote, The convex-hull-and-line traveling salesman problem: a solvable case, Inform. Process. Lett. 51 (1994) 141–148.

[7] A. García, J. Tejel, Using total monotonicity for two optimization problems on the plane, Inform. Process. Lett. 60 (1996) 13–17.

[8] L.L. Larmore, B. Schieber, On-line dynamic programming with applications to the prediction of RNA secondary structure, J. Algorithms 12 (1991) 490–515.

[9] J.K. Park, A special case of the n-vertex traveling-salesman problem that can be solved in $O(n)$ time, Inform. Process. Lett. 40 (1991) 247–254.