Nonserial Dynamic Programming: On the Optimal Strategy of Variable Elimination for the Rectangular Lattice

ALBERTO MARTELLI AND UGO MONTANARI

Consiglio Nazionale delle Ricerche, Istituto di Elaborazione della Informazione, 56100 Pisa, Italy

Submitted by Richard Bellman

The secondary optimization problem in dynamic programming consists of finding the "best" order of variable elimination. This problem can be completely stated in terms of an "interaction graph" representing the variable interaction structure of the objective function. In this paper, two general results about the variable elimination process are presented.

The class of problems having a rectangular lattice as interaction graph is then considered in detail, and two particular variable elimination strategies are both proved optimal. An application to picture processing by computer is finally shown. The same interaction graph, but with a different cost function, is also suitable for representing the topological structure of a (sparse) symmetric system of linear algebraic equations. Here the problem is to find the order of Gaussian elimination of the variables yielding the minimal number of multiplications. The results of this paper apply also partially to this type of variable elimination process.

1. INTRODUCTION

Dynamic programming [1] is a well-known optimization technique. It is efficient for objective functions of many variables which are sums of terms, each term depending from a few variables only. Dynamic programming was first introduced for simple serial problems. A "chain" of functional equations can be easily written, and solved with the so-called embedding technique. Here each stage of the solution algorithm corresponds to the optimization with respect to one variable, and all the stages are essentially of the same type.

However, a number of problems exists where the order of elimination of the variables is not obvious. In fact, different orders can lead to quite different computational costs of the solution. The problem of finding the best scheme is called secondary optimization problem (SOP). In general, the SOP itself can be solved with a dynamic programming algorithm [2]. Furthermore, many properties can be proved [3, 4] which allow one to reduce the combinatorics of the problem, and to find a solution in particular cases.
In this paper, some results are presented about the variable elimination process which allow easy computation of the new problem $P$ obtained after the elimination of any set of variables (regardless of the order; see [2]). Furthermore, it is possible to give lower bounds about the "complexity" of problem $P$. These properties allow one to solve the SOP for an interesting class of problems, namely, when the "interaction graph" of the variables is a rectangular lattice. A practical optimization problem of this type, occurring in picture processing by computer, is finally sketched.

2. Dynamic Programming

A cost function $f(x_1, x_2, ..., x_n)$ consisting of a sum of terms must be minimized. An important tool for understanding the structure of the problem is the interaction graph $G$ of function $f$. Vertices $V_i$ of $G$ are in one-to-one correspondence with variables $x_i (i = 1, ..., n)$ of $f$. An undirected arc between vertices $V_i$ and $V_j$ of $G$ characterizes the existence of at least one term of $f$ depending from both $x_i$ and $x_j$. For instance, the interaction graph of

$$f = f_1(x_1, x_2, x_3) + f_2(x_1, x_4, x_5) + f_3(x_3, x_4, x_5) + f_4(x_6)$$

is shown in Fig. 1(a).

If we want to eliminate the variable $x_1$, the original problem

$$C = \min_{x_1 \cdots x_6} f$$

can be modified as follows:

$$C = \min_{x_2 \cdots x_5} f_3(x_3, x_4, x_5) + f_4(x_6) + \min_{x_1} f_1(x_1, x_2, x_3) + f_2(x_1, x_4, x_5)$$

$$= \min_{x_2 \cdots x_5} f_3(x_3, x_4, x_5) + f_4(x_6) + f_5(x_2, x_3, x_4, x_5)$$

$$= \min_{x_2 \cdots x_5} f'$$

In other words, by partial optimization with respect to $x_1$, a reduced problem of the same type can be obtained. Note that a new term $f_5$ must be computed, depending from as many variables as was the degree $d_1$ of vertex $V_1$. If, for simplicity, all the variables can assume the same number $N$ of values, this stage requires

$$C_1 = N^{d_1}$$

elementary optimizations. The number $d_1$ is thus called the dimension of the stage, and is a measure of its computational cost. Now it is easy to see that
the interaction graph $G'$ of the reduced problem (Fig. 1b) can be obtained from the old graph $G$ by erasing vertex $V_1$ and connecting all the vertices adjacent to $V_1$ in $G$ with a complete graph.

The dynamic programming method consists of the sequential elimination of all the variables by the above technique. However, the order of elimination

![Diagram of an interaction graph before and after the elimination of vertex $V_1$.](image)

**Fig. 1.** An interaction graph (a) before and (b) after the elimination of vertex $V_1$. 
of the variables (or strategy) is essential in determining the dimensions of the stages and thus the overall computational cost of the process. For instance, in our example the strategy defined by

\[ S' = x_1, x_2, x_3, x_4, x_5, x_6 \]

has stage dimensions

\[ D' = 4, 3, 2, 1, 0, 0, \]

while the strategy

\[ S'' = x_2, x_1, x_3, x_4, x_5, x_6 \]

has stage dimensions

\[ D'' = 2, 3, 2, 1, 0, 0. \]

The overall computational cost is defined as some function of stage dimensions. Due to the exponential dependence of (2.2), the strategy dimension,

\[ d = \max d_i, \quad i = 1, \ldots, n, \]

is usually taken.¹

The secondary optimization problem (SOP) consists of finding a strategy \( S_{\min} \) leading to the minimal \( d \). The value \( d_{\min} \) is called the problem dimension.

3. THE ELIMINATION PROCESS

Let \( G = (N, A) \)² be an interaction graph, and let \( E \subseteq N \) be the set of the vertices eliminated at some stage of the optimization process. Let \( H = (E, A_H) \) be the subgraph of \( G \) corresponding to the set of vertices \( E \). The transitive closure of \( A_H \), an equivalence relation,³ gives us a partition \( \pi \) of the set \( E \). The elements \( E_K \subseteq E \) (\( K = 1, \ldots, p \)) of this partition have the meaning of connected components of \( H \) and are called holes.

¹ A related type of variable elimination process occurs in the solution by Gaussian elimination of sparse symmetric systems of linear algebraic equations [5, 6]. If the coefficient matrix is interpreted as incidence matrix of the interaction graph, the above concepts apply. However, the costs of the stages are quadratic instead of exponential, and thus the total cost is the sum of the individual costs of the stages.

² Formally, an undirected graph \( G \) consists of a set of vertices \( N \) and of a symmetric relation \( A \) between \( N \) and itself (i.e., \( A \subseteq N \times N \) and \( (V_i, V_j) \in A \iff (V_j, V_i) \in A \)). The elements of \( A \) are called arcs. A graph \( G' = (N', A') \) is called a subgraph of \( G \) iff \( N' \subseteq N \) and \( A' = A \cap (N' \times N') \), namely, if \( A' \) contains exactly the arcs of \( A \), whose both extremas are in \( N \).

³ The graph is undirected, and loops can be considered as always present.
The set
\[ B_K = \{ V_i \mid V_i \in N - E; \exists V_j \in E_K \text{ such that } (V_i, V_j) \in A \} \]
is called the boundary of hole \( E_K \). \( B_K \) is the set of noneliminated vertices which are adjacent, in \( G \), to some vertex of the hole \( E_K \). For instance, if in the graph of Fig. 2a we assume \( E = \{ V_4, V_5, V_6, V_7 \} \), the subgraph \( H \) is represented in Fig. 2b. Therefore we have two holes: \( E_1 = \{ V_4, V_5, V_6 \} \) and \( E_2 = \{ V_7 \} \). The corresponding boundaries are
\[ B_1 = \{ V_3 \} \quad \text{and} \quad B_2 = \{ V_1, V_3 \}. \]
Note that vertex \( V_3 \) belongs to two different boundaries, while vertex \( V_2 \) does not belong to any boundary.

We can now prove the following theorems.

![Fig. 2](image-url)
Theorem 3.1. The reduced graph $G' = (N - E, A')$ does not depend upon the order of elimination of the variables [2]. The graph $G'$ can be obtained taking the subgraph $F = (N - E, A_F)$ of $G$ and connecting with arcs all the pairs of vertices belonging to the boundary of the same hole. More formally,

$$A' = A_F \cup \bigcup_{k=1}^{p} (B_k \times B_k).$$

(3.1)

Note that $(B_k \times B_k) (K = 1,\ldots, p)$ is a complete relation, namely, in $G'$ every hole is "substituted" with a complete subgraph. In our example, the graphs $F$ and $G'$ are shown in Fig. 2c and d.

Proof. The reduced graph is unique [2]. Therefore a particular order of elimination of the variables can be devised, in which, if $K < h$, all the vertices of hole $E_K$ are eliminated before the vertices of hole $E_h$. Inside every hole $E_K$, an order is chosen such that at any stage (except the first stage of the hole) the eliminated vertex belongs to the boundary of the present part of hole $E_K$.

We can now prove our result by induction: We assume (3.1) before the elimination of the $h$-th vertex $V$ of the $K$-th hole, and we will prove that it holds also afterwards. In fact, if $h = 1$ the vertex $V$ to be eliminated is the first vertex of the $K$-th hole. Vertex $V$ cannot belong to any boundary $B_i$ ($i = 1,\ldots, K - 1$), because $E_K$ and $E_i$ are distinct connected components of $\pi$. Thus, according to (3.1), vertex $V$ is connected exactly with the same vertices as in the initial graph $G$. Therefore $V$ is substituted by a complete graph by definition, adding a new term to the summa in the right side of (3.1). If $h \neq 1$, $V$ belongs by construction to the boundary $B_K$ of the present part $E_K'$ of hole $E_K$ ($E_K'$ consists of the first $h - 1$ vertices of $E_K$, in the assumed elimination order). According to (3.1), $V$ is connected with: (i) all the vertices not yet eliminated with which it was connected in $G$ (let us call this set $S$); (ii) all the vertices $\in B_K'$. The elimination of $V$ causes the connection with a complete set of arcs of the union set $S \cup B_K'$. But this set is exactly the boundary $B_K^*$ of the first $h$ vertices of hole $E_K$. Q.E.D.

Theorem 3.2. Given an interaction graph, the following properties hold:

(a) the dimension of the stage $K$ (in which the vertex $V_{i_K}$ is eliminated) is equal to the cardinality of the boundary $B_j$ of the hole $E_j$, where $V_{i_K} \in E_j$, after the stage $K$;

(b) at any stage, the value

$$\max_{j=1,\ldots, p} |B_j|$$

is a lower bound to the dimensions of the preceding stages.
Proof. (a) We first remark that the subgraph corresponding to $E_j - V_{ik}$ is not necessarily connected. Thus let $E_j^i (i = 1, \ldots, q)$ be its connected components and let $B_j^i$ be the corresponding boundaries. It is now easy to see that $B_j$ is the union of the boundaries $B_j^i$ and of the set $S$ of noneliminated vertices adjacent to $V_{ik}$, minus $V_{ik}$. Furthermore, $V_{ik}$ belongs to all boundaries $B_j^i (i = 1, \ldots, q)$ because hole $E_j$ is connected. Therefore by Theorem 3.1, $V_{ik}$ is adjacent (before elimination) exactly to all the vertices $\in B_j^i (i = 1, \ldots, q)$ and to all the vertices $\in S$, i.e. to all the vertices, $\in B_j$.

(b) According to part (a), $|B_j|$ is the dimension of the last eliminated vertex $V_{ik} \in E_j$, $j = 1, \ldots, p$. In fact, after erasing $V_{ik}$, no vertex $V \in B_j$ can be eliminated, because otherwise $V_{ik}$ would not be the last eliminated vertex $\in E_j$ as assumed. Q.E.D.

4. The Rectangular Lattice

In this section, the secondary optimization problem for the interaction graph in Fig. 3 is considered.

The $i$-th row ($i = 1, \ldots, r$) of this graph is the set of vertices $V_{i,j}$, $j = 1, \ldots, c$ and the $j$-th column ($j = 1, \ldots, c$) is the set of vertices $V_{i,j}$, $i = 1, \ldots, r$. The

---

*Fig. 3. The rectangular lattice.*
first row and the last row are said to be, respectively, the N-border and the S-border of the lattice. Analogously, the first column and the last column are the W-border and the E-border, respectively. A hole \( E_k \) is said to touch a border, if at least one element of \( E_k \) belongs to this border.

Now we prove the following Lemmas.

**Lemma 4.1.** If two vertices \( V_{k,l} \) and \( V_{m,n} \) belong to the same hole \( E_h \), then lower bounds to the possible cardinality \( C_h \) of the boundary of \( E_h \) are:

1. \( C_h \geq |n - l| + 1 \) if \( E_h \) does not touch the N-border or the S-border (see, for instance, the lower hole in Fig. 4a);
2. \( C_h \geq |m - k| + 1 \) if \( E_h \) does not touch the E-border or the W-border;
3. \( C_h \geq 2 |n - l| + 2 \) if \( E_h \) does not touch the N-border and the S-border (see, for instance, the lower hole in Fig. 4b);
4. \( C_h \geq 2 |m - k| + 2 \) if \( E_h \) does not touch the E-border and the W-border.

**Proof.** (a) Let us assume that the hole \( E_h \) does not touch the N-border and \( l \leq n \). By definition, a hole is connected; therefore, there is a sequence of adjacent vertices of the graph, connecting \( V_{k,l} \) with \( V_{m,n} \), such that every element of the sequence belongs to \( E_h \) (see Fig. 4a).

The j-th column \((j = 1, \ldots, n)\) contains at least one element of this sequence, say \( V_{i,j} \), otherwise the sequence would not be connected. Now we move along the j-th column from the N-border toward \( V_{i,j} \) until we find a vertex \( V_{s,j} \) belonging to \( E_h \). The precedent vertex \( V_{s-1,j} \) cannot belong to another hole \( E_k \), otherwise \( E_h \) and \( E_k \) would be connected. Thus it belongs to \( N - E_h \), and it is an element of the boundary of \( E_h \), because it is adjacent to an element of \( E_h \). Since each j-th column \((j = 1, \ldots, n)\) contains at least one element of the boundary, its minimum cardinality is \( n - l + 1 \) (see the crossed vertices in Fig. 4a).

(b) Let us now assume that \( E_h \) does not touch the S-border too. The reasoning of part (a) can be repeated also for the S-border instead of the N-border. Therefore, if we move along the j-th column from the S-border toward \( V_{i,j} \), we find a distinct boundary vertex of \( E_h \). Since each j-th column \((j = 1, \ldots, n)\) contains at least two elements of the boundary, the minimum cardinality of it is \( 2(n - l + 1) \) (see Fig. 4b).

Let \( V_{i,j} \) \((i = 1, \ldots, r, j = 1, \ldots, c)\) be a vertex; its distance from the N-border, E-border, S-border, or W-border is given by \( i - 1 \), \( c - j \), \( r - i \), or \( j - 1 \), respectively.

---

4 We cannot have \( s = 1 \), otherwise the hole \( E_h \) would touch the N-border.
Fig. 4. Lower bounds for the cardinality of the boundary of a hole $E_h$ (lower right corner): Dotted vertices are noneliminated vertices; while crossed vertices are the boundary vertices of $E_h$ found by the procedure proved in Lemma 4.1. (a) Hole $E_h$ does not touch the $N$-border. (b) Hole $E_h$ does not touch the $N$- and the $S$-borders.

**Lemma 4.2.** If a hole $E_h$ touches only two contiguous borders and if $E_h$ contains a vertex whose distances from these borders are $n$ and $m$, then a lower bound to the cardinality $C_h$ of the boundary of $E_h$ is

$$C_h \geq m + n + 2.$$
Proof. Let us assume that \( E_h \) touches the \( N \)-border and the \( W \)-border only, and that the vertex \( V_{m+1,n+1} \) is an element of \( E_h \) (see Fig. 5). Lemma 4.1 applied twice proves that each \( i \)-th row \( (i = 1, \ldots, m + 1) \) and each \( j \)-th column \( (j = 1, \ldots, n + 1) \) contains at least one boundary element of \( E_h \) and gives an effective procedure for finding these vertices. Therefore, if they are shown to be all distinct, the Lemma is proved.

Let us consider the \( i \)-th row \( (1 < i < m + 1) \). According to Lemma 4.1, let \( V_{i,j} \) be the first boundary element of \( E_h \), starting from the \( E \)-border. If \( j > n + 1 \), \( V_{i,j} \) cannot coincide with any other considered boundary vertex; so, let us assume \( j < n + 1 \) (see, for instance, \( V_{1,3} \) in Fig. 5). Now we show that \( V_{i,i} \) is distinct from the boundary vertex \( V_{i,j} \) (vertex \( V_{3,5} \) in Fig. 5) found in column \( j \) by the second application of Lemma 4.1. In fact, if \( i = s \), we would have by construction

\[
V_{i,k} \notin E_h, \quad k = j, \ldots, c, \quad i \leq m + 1, \\
V_{k,j} \notin E_h, \quad k = i, \ldots, r, \quad j \leq n + 1,
\]

and no path belonging to \( E_h \) from \( V_{m+1,n+1} \) to the \( W \)- or \( N \)-border would exist.

Before going on to the main theorem, we remark that a hole can grow in two distinct ways. The hole can grow with continuity, that is its cardinality increases by one whenever a vertex \( V \) of its boundary is eliminated. This is the case if \( V \) does not belong to the boundary of any other hole. On the contrary, if \( V \) belongs to the boundary of more than one hole, its elimination causes the merging of the holes.
Now we can establish a lower bound for the dimension of the strategies for the rectangular lattice. From now on, we assume \( r \leq c \).

**Theorem 4.1.** Given any strategy for the interaction graph of Fig. 3, its dimension is not less than \( r \) \((r \leq c)\).

**Proof.** When the last variable is eliminated, there is only one hole, whose cardinality is \( rc \) and which touches all the borders. Therefore, there is some stage at which a hole touching four borders is generated. According to the two ways in which a hole can grow, we distinguish two cases:

(a) At some stage, a hole \( E_h \) touching less than four borders yields a hole touching four borders, by growing with continuity.

(b) At some stage, a hole touching four borders is generated by the merging of some holes, each touching less than four borders.

In case (a) we show first that the hole \( E_h \) touches three borders. In fact, there is only another possibility: that the hole touches two borders and that a vertex belonging to the other two borders is eliminated. Without lack of generality, let the hole touch only the \( E \)-border and the \( S \)-border and let the vertex \( V_{1,1} \) be eliminated. But \( V_{1,1} \) is not a boundary vertex of \( E_h \), because \( V_{1,1} \) is adjacent to \( V_{1,2} \) (belonging to the \( N \)-border) and to \( V_{2,1} \) (belonging to the \( W \)-border).

If hole \( E_h \) touches three borders, it certainly has two elements belonging to two opposite borders and no element belonging to one of the other borders. Therefore, according to Lemma 4.1a, a lower bound to the cardinality of hole \( E_h \) is either \( r - 1 + 1 = r \) or \( c - 1 + 1 = c \), in any case at least \( r \).

In case (b), we prove that at least one of the holes has a boundary cardinality \( \geq r \). If any of them touches three borders or two opposite borders, the assertion is proved (using Lemma 4.1a), as shown in the precedent case. Therefore, we assume that every hole touches either two contiguous borders or less than two borders. After the merging, the generated hole touches all the borders. Therefore, given any border, there is at least one hole which touches it.

Let \( V_{i,j} \) be the variable eliminated at this stage. Without lack of generality, we can assume

\[
i > \left\lfloor \frac{r}{2} \right\rfloor \quad \text{and} \quad j > \left\lfloor \frac{c}{2} \right\rfloor.
\]

Since \( V_{i,j} \) is on the boundary of all the holes to be merged, each of these holes contains at least one of the four vertices adjacent to \( V_{i,j} \). In what follows, three cases are examined and the worst of these vertices is considered in each case: The lower bounds given by Lemmas 4.1 and 4.2 are then computed.
(b1) A merging hole $E_h$ exists which touches the $N$-border ($W$-border) only.

(b2) A merging hole $E_h$ exists, which touches both the $N$-border and the $W$-border.

(b3) Two merging holes $E_h$ and $E_k$ exist, such that $E_h$ touches the $W$- and $S$-border, and $E_k$ touches the $N$- and $E$-border.

No other case exists, because the $N$- and $W$-border must be touched by some hole, and no hole can touch more than two borders.

In case (b1) (see Fig. 6a), the worst adjacent vertex is $V_{i-1,j}$ ($V_{i,j-1}$). By Lemma 4.1b, the first $i - 1$ rows ($j - 1$ columns) contain $p = 2(i - 1)$ ($p' = 2(j - 1)$) boundary vertices. Adding the vertex $V_{i,j}$, we have a lower bound of

$$q = 2(i - 1) + 1 \quad (q' = 2(j - 1) + 1).$$

From (4.1), we have, in the $N$-border case,

$$i - 1 \geq \left\lfloor \frac{r}{2} \right\rfloor \quad j \geq \left\lfloor \frac{c}{2} \right\rfloor + 1 \quad (4.2)$$

and, therefore,

$$q \geq 2 \left\lfloor \frac{r}{2} \right\rfloor + 1 \geq r - 1 + 1 = r.$$

Similarly, in the $W$-border case we have

$$q' \geq c.$$

Being $r \leq c$, in any case a lower bound is $r$.

In case (b2) (see Fig. 6b) the worst adjacent vertex is $V_{i-1,j}$. By Lemma 4.2, the lower bound is

$$q = (i - 1) + j.$$

From (4.2), we have

$$q \geq \left\lfloor \frac{r}{2} \right\rfloor + \left\lfloor \frac{c}{2} \right\rfloor + 1.$$

Finally, from $c \geq r$, we have

$$q \geq 2 \left\lfloor \frac{r}{2} \right\rfloor + 1 \geq r.$$

In case (b3) (see Fig. 6c), the worst adjacent vertex for hole $E_h$ is $V_{i,j-1}$ and for hole $E_k$ is $V_{i-1,j}$. By Lemma 4.2, we have

$$q_h = (j - 1) + (r - i + 1) = r + (j - i),$$

$$q_k = (i - 1) + (c - j + 1) = c - (j - i).$$
Fig. 6. A lower bound for the problem dimension of the rectangular lattice is $\min(r, c)$. 
Whatever the value of term \((j - i)\) we have

\[
\max(q_h, q_k) \geq r.
\]

We have shown that, at some stage, there is a hole such that the cardinality of its boundary is not less than \(r\). Therefore, according to Theorem 3.2b, \(r\) is also a lower bound to the dimension of the strategy. Q.E.D.

Now we show that some strategies can be devised, whose dimension is \(r\)

\[\begin{array}{c}
V_1 \\
V_2 \\
V_{K-1} \\
V_K \\
V_{K+1} \\
V_{r-1} \\
V_r
\end{array}\]

\[\begin{array}{c}
V_r \\
V_{r-1} \\
V_{K-1} \\
V_K \\
V_{K+1} \\
V_{r-1} \\
V_2
\end{array}\]

Fig. 7. Intermediate interaction graphs in the "by columns" (a); and "diagonal" (b) strategies.
(r ≤ c). Therefore, according to Theorem 4.1, their dimension is minimal. A first strategy is to eliminate vertices “by columns” (see Fig. 7a). In all the intermediate stages, according to Theorem 3.1, the vertex to be eliminated is connected with r - 1 boundary vertices and one nonboundary vertex. Thus, the dimension of the problem is r. The stages corresponding to the first and last column have less dimension. Also the “diagonal” elimination (see Fig. 7b) leads to the same dimension, for the same reason. An advantage of this second strategy is that the number of stages with dimension less than r is larger: in the case of the square r = c = n we have 2(n - 1) stages of dimension n, instead of (n - 1)^2.

5. An Application to Picture Processing

Optimization methods in picture processing try to define a figure of merit, which embodies the heuristics of the problem [7-9]. A dynamic programming

![Diagram](image)

**Fig. 8.** An optimization problem in picture smoothing yielding the rectangular lattice as interaction graph: (a) given picture.
algorithm is then used for determining the "best" processed figure. Here the secondary optimization problem is often relevant. For instance, a paper by one of the authors [8] is concerned with finding the "best" system of curves in a noisy picture. In this case, the interaction graph of the problem reflects the topological properties of the sought system of curves. In another paper [9], the authors apply the optimization concepts to the definition of a smoothing procedure: The smoothed image must optimize some figure of merit taking into account the "regularity" of the smoothed image itself and its "fidelity" to the given image. Being images discretized and represented by rectangular matrices, this problem generates a rectangular lattice as interaction graph. The results of this paper are then applicable, and the "by columns" strategy is used in practice, combined with an approximation technique. Figure 8 shows an example of application of this method to fingerprint processing: Fig. 8a shows a given stylized fingerprint during an intermediate stage of the recognition process, and Fig. 8b represents its smoothed version.

![Fig. 8. (b) smoothed picture.](image-url)
REFERENCES