Revue française d'informatique et de recherche opérationnelle. Série verte

Umberto Bertelè
Francesco Brioschi

Parametrization in nonserial dynamic programming

Revue française d'informatique et de recherche opérationnelle. Série verte, tome 5, nº V2 (1971), p. 87-102
<http://www.numdam.org/item?id=RO_1971__5_2_87_0>

R.I.R.O.
(5e année, V-2, 1971, p. 87-102)

PARAMETRIZATION IN NONSERIAL DYNAMIC PROGRAMMING (1)

by Umberto BERTELÈ (2) and Francesco BRIOSCHI (2)

Abstract. — *New developments in the theory of nonserial dynamic programming are described in this paper. The new, more general, decomposition technique which uses parametrization allows, in general, solving the given optimization problem with less computationl effort. Also interesting connections with graph theory are pointed out.*

1. INTRODUCTION

Nonserial dynamic programming is a new branch of mathematical programming. Essentially it exploits decomposition, as expressed by Bellman's principle of optimality, for breaking the optimization problem into many smaller subproblems and is concerned with finding a decomposition which is optimal from the point of view of the computing time and the memory requirements (secondary optimization problem). The solution of the secondary optimization problem necessitates graph theoretical considerations. The works in this field up to now are [1, 2, 3, 4].

This paper introduces parametrization in nonserial dynamic programming. The basic idea is simple. It derives from the concept of « cut state » introduced in [5] and reported in [6] and [7]. Let $X = \{ x_1, x_2, ..., x_M \}$ be the set of variables of the optimization problem. Parametrization consists in selecting a proper subset $X' \subset X$, considering the simpler optimization problem for each assignment of the variables of X', and finally searching for an optimal solution through the assignments of X'.

This is equivalent to renouncing the use of decomposition for the variables of X'. Surprisingly enough it is shown that parametrization may be effective for reducing the computational complexity of the problem.

(1) This work has been supported by C.N.R.
(2) The authors are with the Istituto di Elettrotecnica ed Elettronica, Laboratorio di Controlli Automatici, Politecnico di Milano, Milano, Italy.

The organization of the paper is as follows. The basic concepts of nonserial dynamic programming are recalled in section 2. Parametrization is introduced first by means of an example (section 3) and, next, formally, in section 4. A procedure for determining, in some cases, an optimal parametrization is given in section 5. Here also some interesting connections with a well known property of graph theory (Mason's index) are reported. Finally section 6 contains some examples.

2. NONSERIAL DYNAMIC PROGRAMMING

Consider the following optimization problem

$$\min_X F(X) = \min_X \sum_{i \in I} f_i(X^i)$$

where

$$X = \{ x_1, x_2, ..., x_M \}$$

is a set of discrete variables,

$$I = \{ 1, 2, ..., n \}$$

and

$$X^i \subset X.$$

Each *component* $f_i(X^i)$ of the *cost* function $F(X)$ is specified by means of a stored table with $\sigma^{|X^i|}$ rows. For simplicity it has been assumed that all variables have the same range, namely that each variable can assume σ values.

Clearly the optimization problem stated above can be solved exhaustively by a straight-forward approach which consists in trying all σ^M possible assignment for X. Since this approach may require a very large computing time, it is convenient to consider solutions to this problem based on decomposition. Let $x_i \in X$ and $x_j \in X$. The two variables x_i and x_j are said to *interact* if there exists a component $f_k(X^k)$ such that both x_i and x_j belong to X^k.

The set of all the variables interacting with a member $x \in X$ is denoted with $\Gamma(x)$. Then consider a non empty subset $Y \subset X$. The set of all the variables, other then those in Y, which interact with at least one member of Y is denoted by $\Gamma(Y)$ and called the set of variables interacting with Y. Note the difference of this definition $\Gamma(Y) = \bigcup_{x \in Y} \Gamma(x) - Y$ with the usual one (see, for instance, [8]) : $\Gamma(Y) = \bigcup_{x \in Y} \Gamma(x)$.

One ordered partition, among all the possible ones, of the variables of the set X is selected. Let $Y_1, Y_2, \ldots, Y_m$ $(m \leqslant M)$ be such ordered partition. *For this partition* the optimization problem may be solved by dynamic programming.

More specifically, the subset Y_1 is considered first. For minimizing $F(X)$ w.r.t. (with respect to) Y_1 it is sufficient to compute

$$\min_{Y_1} \sum_{i \in I_1} f_i(X^i) = g_1(\Gamma(Y_1))$$

where

$$I_1 = \{ i : X^i \cap Y_1 \neq \varnothing \}.$$

and to store the optimizing assignment of Y_1 as a function of $\Gamma(Y_1)$, namely $Y_1(\Gamma(Y_1))$.

The minimization of $F(X)$ w.r.t. Y_1, for all possible assignments of $\Gamma(Y_1)$, is called the *elimination* of the subset Y_1.

The problem remaining after the elimination of Y_1

$$\min_{X - Y_1} g_1(\Gamma(Y_1)) + \sum_{i \in I - I_1} f_i(X^i)$$

is of the same form of the original one since the function $g_1(\Gamma(Y_1))$ may be regarded as a component of the new cost function.

Let $\Gamma(Y_j \mid Y_1, Y_2, \ldots, Y_{j-1})$ be the set of variables interacting with Y_j in the problem obtained after the elimination of $Y_1, Y_2, \ldots, Y_{j-1}$ in that order. Clearly $\Gamma(Y_m \mid Y_1, Y_2, \ldots, Y_{m-1}) = \varnothing$.

Then, according to the philosophy of dynamic programming, an optimal assignment for X can be obtained in two steps :

a) eliminating the subsets Y_j in the order $Y_1, Y_2, \ldots, Y_m$ and storing the optimizing assignment $Y_1^*(\Gamma(Y_1)), Y_2^*(\Gamma(Y_2 \mid Y_1), \ldots, Y_m^*$

b) operating « backwards » determining successively $Y_m^*, Y_{m-1}^*, \ldots, Y_1^*$ i.e. the optimizing assignment for X from the stored tables.

It is clear now that another optimization problem, *the secondary optimization problem*, emerges. An optimal assignment for X can be equally obtained by all the ordered partitions of the variables of the set X. Which, then, among those ordered partitions is the best from the point of view of minimizing the number of operations required (i.e. the computing time) with the constraint that the storage space does not exceed a prescribed level?

The elimination of the subset Y_j implies the construction and storage of a table which, in correspondence to each of its $\sigma^{|\Gamma(Y_j \mid Y_1, Y_2, \ldots, Y_{j-1})|}$ rows, gives the values of the optimizing assignment $Y_j{}^*$ and of the new component g_j.

The number of table look-ups required is $\sigma^{|\Gamma(Y_j \mid Y_1, Y_2, ..., Y_{j-1})|} \cdot \sigma^{|Y_j|}$ times the number of components which contain at least one member of Y_j.

Since the exponential factor is, usually, the most decisive, the integer $|\Gamma(Y_j \mid Y_1, Y_2, ..., Y_{j-1})| + |Y_j|$ may be assumed as a reasonable index of the computational effort for eliminating Y_j while $|\Gamma(Y_j \mid Y_1, Y_2, ..., Y_{j-1})|$ is an index of the memory space needed.

Solving the secondary optimization problem consists in finding one ordered partition $Y_1, Y_2, ..., Y_m$ for which the largest integer

$$|\Gamma(Y_j \mid Y_1, Y_2, ..., Y_{j-1})| + |Y_j|$$

is minimal, subject to the constraint that $|\Gamma(Y_j \mid Y_1, Y_2, ..., Y_{j-1})|$ does not exceed a prescribed integer h.

Formally, letting K be the set of ordered partitions of X and letting $k \in K$, it is possible to assign to each optimization procedure k the two integers :

$$\gamma(k) = \max_j (|Y_j^k| + |\Gamma(Y_j^k \mid Y_1^k, Y_2^k, ..., Y_{j-1}^k)|)$$

and

$$\delta(k) = \max_j |\Gamma(Y_j^k \mid Y_1^k, Y_2^k, ..., Y_{j-1}^k)|$$

where the indexing w.r.t. k refers to partition k.

Then the secondary optimization problem can be stated as

$$\min_{k \in K} \gamma(k) = C_h$$

subject to

$$\delta(k) \leqslant h.$$

The integers $|Y^k| + |\Gamma(Y_j^k \mid Y_1^k, Y_2^k, ..., Y_{j-1}^k)|$ and $|\Gamma(Y_j^k \mid Y_1^k, Y_2^k, ..., Y_{j-1}^k)|$ are called respectively *cost of eliminating the subset Y_j^k* and *dimension of the stored table in the elimination of Y_j^k* in the ordered partition k.

Finally C_h is called the *h-cost* of the optimization problem. It must be noted that, in the definition of the cost of eliminating the subset Y_j, the implicit assumption that Y_j is eliminated exhaustively has been made.

A special case of great importance is the one when variables are eliminated one by one. It can be shown [1], in fact, that whenever there are no storage limitations, there exists an ordered partition, whose blocks consist of a single variable, which is a solution of

$$\min_{k \in K} \gamma(k).$$

Letting $K' \subset K$ be the set of the $M!$ ordered partitions, whose blocks consist of a single variable, it is clear that, for $k \in K'$, $\gamma(k) = \delta(k) + 1$,

Then, obviously, the order of elimination which minimizes $\delta(k)$ also minimizes $\gamma(k)$.

The integer

$$\min_{k \in K'} \delta(k) = D$$

is called the *dimension* of the optimization problem and the integer

$$\min_{k \in K'} \gamma(k) = C$$

is called the *cost* of the problem. Then $C = D + 1$.

It is now shown that this problem becomes a problem in graph theory.

The interaction graph of the original (primary) optimization problem $G(X, \Gamma)$ is an undirected graph defined by :

1) The vertex set of the graph is the set X of the variables of the primary problem.

2) Two vertices are connected with an edge if and only if the corresponding variables interact.

The elimination of a subset Y_1 from the original problem implies a new one in which all the tables containing at least a member of Y_1 are replaced by a new table containing all the variables interacting with Y_1.

Hence the interaction graph of the new problem is obtained from the original one deleting the vertices of the set Y_1 and all the edges emanating from them and connecting all the previously unconnected vertices in $\Gamma(Y_1)$. An example is given in figure 1.

When variables are eliminated one by one it is clear that he secondary optimization problem is finding an order of elimination of the vertices of $G(X, \Gamma)$ such that the largest degree of the eliminated vertices is minimal.

3. AN INTRODUCTORY EXAMPLE

This section introduces the idea of parametrization by means of an example which, for simplicity, considers only eliminations of one variable at a time.

Let $F(X) = \sum_{i=1}^{5} f_i(X^i)$ where $X = \{ x_1, x_2, x_3, x_4, x_5 \}$ and $X^1 = \{ x_1, x_2 \}$,

$X^2 = \{ x_2, x_3 \}$, $X^3 = \{ x_3, x_4 \}$, $X^4 = \{ x_4, x_5 \}$ and $X^5 = \{ x_5, x_1 \}$.

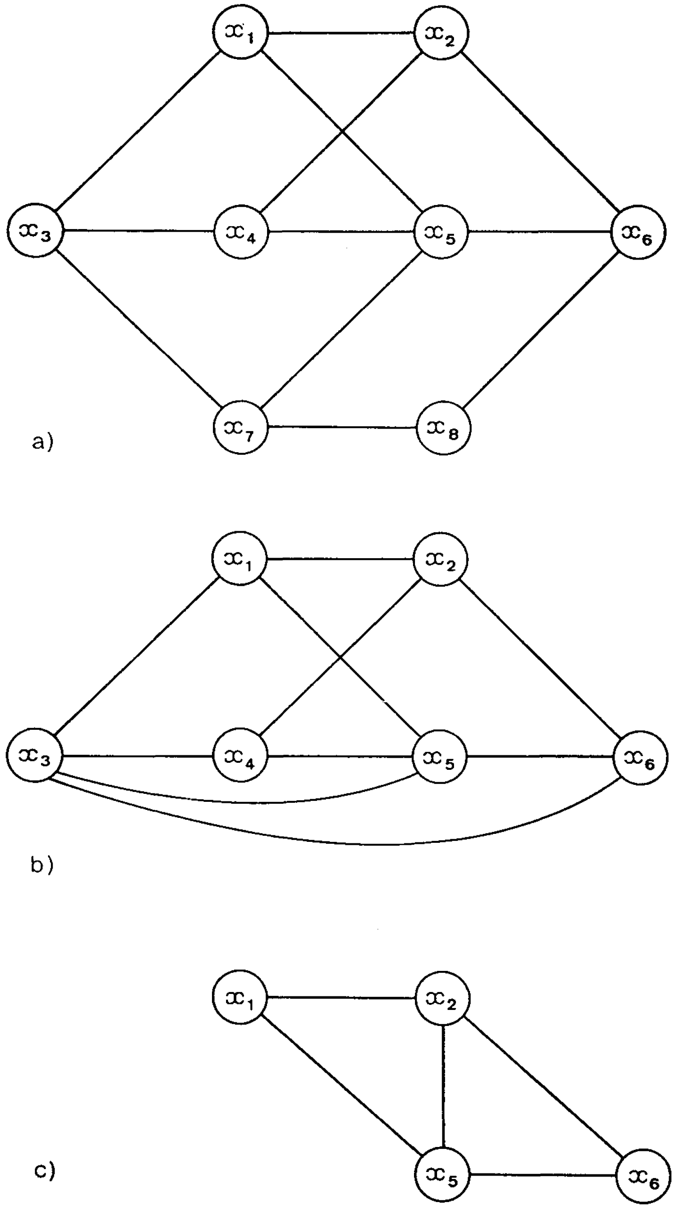

Figure 1

An interaction graph (*a*) and the graphs resulting after the elimination of the sets $\{ x_7, x_8 \}$ (*b*) and successively $\{ x_3, x_4 \}$ (*c*)

The interaction graph is shown in figure 2 *a*.

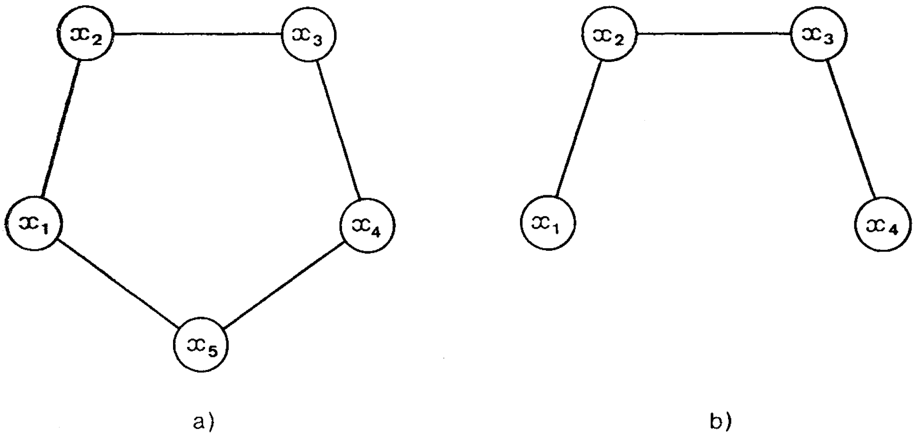

Figure 2

An interaction graph (*a*) and the graph resulting after the parametrization of x_5 (*b*)

The order of elimination x_1, x_2, x_3, x_4, x_5 (which is clearly an optimal one) necessitates a number of table look ups (see section 2 and the example given in [1] and [3]) équal to

$$2 \cdot \sigma^3 + 2 \cdot \sigma^3 + 2 \cdot \sigma^3 + 2 \cdot \sigma^2 + 1 \cdot \sigma.$$

For this order of elimination it results $C = 3$ and it is clear that it is a reasonable index of the computational complexity of the problem. Note the difference between the exhaustive approach to optimization which necessitates $5 \cdot \sigma^5$ table looks-ups and the decomposition procedure just employed.

Next a procedure, in which x_5 is parametrized, is considered. This means that the simple optimization problem, in which x_5 assumes one among its possible σ values, is solved σ times. The interaction graph of the derived simpler problem is given in figure 2 *b*. In this problem consider the order of elimination x_1, x_2, x_3, x_4. The number of table looks-ups needed are :

$$2 \cdot \sigma^2 + 2 \cdot \sigma^2 + 2 \cdot \sigma^2 + 2 \cdot \sigma.$$

Hence the original optimization problem necessitates a number of table look-ups equal to

$$\sigma(2 \cdot \sigma^2 + 2 \cdot \sigma^2 + 2 \cdot \sigma^2 + 2 \cdot \sigma).$$

Again the integer 3 may be taken as a reasonable index of the computational complexity for this decomposition procedure. Also note that, for the decomposition which does not employ parametrization, there are at most two variables, in the new components formed by the elimination procedure while, in the decomposition which employs parametrization, there is at most one variable in the new components.

This suggests the possibility of using parametrization for meeting the memory limitations imposed.

4. PARAMETRIZATION

In section 2 a technique for the solution of the primary optimization problem by decomposition and a criterion for ranking all the possible decomposition procedures have been recalled.

A new, more general, technique for the decomposition solution of the primary optimization problem and, correspondingly, a new criterion for ranking all the possible decomposition procedures are here developed.

Following the ideas introduced in section 3 by means of an example consider a subset $P \subset X$. For each assignment $\bar{P}$ of the variables in P the primary optimization problem (1) becomes

$$\min_{X-P} F(X - P, \bar{P}) = \min_{X-P} \sum_{i \in I} f_i(X^i \cap (X - P), \overline{X^i \cap P}) \tag{2}$$

with obvious meaning of the symbol $\overline{X^i \cap P}$.

Then the solution to the primary optimization problem (1) is obtained considering $\sigma^{|P|}$ assignments of the variables of P, solving problem (2) by means of the procedures of section 2 for each assignment and, finally, selecting an assignment for P and consequently for X-P, which minimizes the cost function $F(X)$.

Thus the *parametrization* of the set of variables P has been used as a step of the new optimization procedure by decomposition.

The problem of ranking the new decomposition procedures now emerges. Let $\Gamma(Y_j \mid P; Y_1, Y_2, ..., Y_{j-1})$ be the set of variables interacting with Y_j in the problem obtained by the parametrization of P, after the elimination of $Y_1, Y_2, ..., Y_{j-1}$ in that order.

Since in the search through the $\sigma^{|P|}$ assignment for P only a best assignment at each time is recorded the storage space required is practically the one needed for problem (2).

Also it is reasonable to assume the integer

$$|P| + \max_j (|Y_j| + |\Gamma(Y_j \mid P ; Y_1, Y_2, ..., Y_{j-1})|)$$

as an index of the compting time since $\sigma^{|P|}$ problems must be solved where the elimination of each subset Y_j requires a number of table look-ups equal to $\sigma^{|\Gamma(Y_j | P; Y_1, Y2, \ldots, Y_{j-1})|}$. $\sigma^{|Y_j|}$ times the number of components which contain at least one member of Y_j at the time of its elimination.

Formally, letting k be an ordered partition of X defined by

$$X = \{ P^k, Y_1^k, Y_2^k, \ldots, Y_l^k \}$$

and letting $\tilde{K}$ be the set of all possible decompositions of this kind, it is possible to assign to each decomposition procedure k the two integers :

$$\tilde{\gamma}(k) = |P^k| + \max_j (|Y_j^k| + \Gamma(Y_j^k | P^k; Y_1^k, Y_2^k, \ldots, Y_{j-1}^k)|)$$

and

$$\tilde{\delta}(k) = \max_j |\Gamma(Y_j^k | P^k; Y_1^k, Y_2^k, \ldots, Y_{j-1}^k)|$$

Then the secondary optimization problem in this case is

$$\min_{k \in \tilde{K}} \tilde{\gamma}(k) = \tilde{C}_h$$

subject to

$$\tilde{\delta}(k) \leqslant h.$$

The integers $\tilde{\gamma}(k)$ and $\tilde{\delta}(k)$ are called respectively *cost* and *dimension* of ordered partition k *when parametrization is allowed.* And, similarly, $\tilde{C}_h$ is the *h-cost of the problem* in such case.

By definition

$$\tilde{C}_h \leqslant C_h.$$

In section 6 an example, for which the strict inequality holds, will be shown. This demonstrates that parametrization is an efficient tool of decomposition.

The interaction graph of problem (2) is obtained from the original interaction graph by deleting all the vertices corresponding to the variables of P and all the edges emanating from them. For graphical convenience, in the sequel, the vertices of P will not be canceled from the interaction graph G but they will be coloured in black.

5. OPTIMAL PARAMETRIZATION FOR A SPECIAL CLASS OF DECOMPOSITIONS

Section 4 describes a very general decomposition procedure for the solution of the primary optimization problem and a corresponding statement of the criterion of the secondary optimization problem which ranks such decompositions. Unfortunately it is not possible, for the time being, to give rules for finding an optimal choice of the set P in the general case nor, on the other hand, an algorith for finding a best decomposition, when variables are not eliminated one by one.

In this section a special class of decompositions, defined by subsets Y_j consisting of a single variable, is dealt with. For this class it is possible to determine the optimal parametrization sets P corresponding to the memory limitation h. Also since the variables of X-P are, by definition, eliminated one by one an optimal decomposition is then obtained by the methods given in [1, 2, 3, 4].

It is worth noting that using decompositions of this kind may be regarded as an expedient for satisfying the memory limitations employing methods which, per se, do not have the capability of handling such limitation. Thus, in this case, the task of meeting the memory requirements falls entirely upon parametrization.

Let $\tilde{K}' \subset \tilde{K}$ be the set of ordered partitions defined above, namely

$$X = \{ P^k, \{ y_1^k \}, \{ y_2^k \}, \ldots, \{ y_l^k \} \}.$$

Then, for $k \in \tilde{K}'$,

$$\tilde{\gamma}(k) = |P^k| + 1 + \tilde{\delta}(k)$$

and the secondary optimization problem is

$$\min_{k \in \tilde{K}'} \tilde{\gamma}(k) = C_h^*$$

subject to

$$\tilde{\delta}(k) \leqslant h.$$

The integer C_h^* is the h-cost of the problem when variables in the set X-P must be eliminated one by one. Clearly $C_h^* \geqslant C_h$.

Lemma 1. *Let $G(X, E)$ be an interaction graph with dimension D. Let $G'(X', E')$ be another graph with $X' = X \cup \{ x \}$ and $E' = E \cup E_x$ where x is a new vertex and E_x is the set of edges emanating from it. Then letting D' be the dimension of G' it results $D \leqslant D' \leqslant D + 1$.*

Proof. The first inequality is trivial. In order to prove the second one, let $y_1, y_2, ..., y_M$ be an optimal order for $G(X, E)$ whose dimension is, hence, D. Then it is easy to see that, for the graph $G'(X', E')$, the order $y_1, y_2, ..., y_M$, x has dimension less or equal to $D + 1$.

Q.E.D.

Lemma 2. *Let $G(X, E)$ be an interaction graph with dimension D. Consider a new graph obtained from $G(X, E)$ deleting a vertex $x \in X$ and all the edges emanating from it. Letting D'' be the dimension of this new graph it results*

$$D - 1 \leqslant D'' \leqslant D.$$

Proof. The second inequality is trivial. In order to prove the first one it is sufficient to note that the possibility $D'' < D - 1$ contradicts lemma 1.

Q.E.D.

Définition 1. Let $G(X, \Gamma)$ be an interaction graph with dimension D. Let $h = 1, 2, ..., D - 1$. The minimal number of vertices which must be canceled so that the dimension of the resulting graph is equal to h is called *h-index* of $G(X, \Gamma)$ and denoted by q_h. Correspondingly a set of q_h vertices whose cancellation makes the dimension equal to h is called *h-index set* and denoted by Q_h.

Theorem 1. *Let $G(X, \Gamma)$ be an interaction graph with dimension D. The h-cost C_h^* is obtained with a set P given by ;*

a) for $h \geqslant D$

$$P = \varnothing \text{ and } C_h^* = C = D + 1$$

b) for $1 \leqslant h \leqslant D - 1$

$$P = Q_h \text{ and } C_h^* = 1 + h + q_h.$$

Proof. a) Consider a partition $k_0 \in \tilde{K}'$ defined by $\{ P, \{ y_1 \}, \{ y_2 \}, ..., \{ y_M \}\}$ where $P = \varnothing$ and $y_1, y_2, ..., y_M$ is one optimal order for the problem in which no parametrization is allowed and variables are eliminated one by one. This partition yields $\tilde{\gamma}(k_0) = 1 + D = C$.

Consider all partitions $k_{1j} \in \tilde{K}'$ with $|P| = 1$. By lemma 2 it follows that $\tilde{\gamma}(k_{1j}) \geqslant \gamma(k_0)$. By repeated use of lemma 2 it is clear that all partitions $k_{ij} \in \tilde{K}'$ with $|P| = i$ have $\tilde{\gamma}(k_{ij}) \geqslant \tilde{\gamma}(k_0)$.

Hence $C_h^* = \tilde{\gamma}(k_0) = C$.

b) Consider a partition $k \in \tilde{K}'$ defined by $\{ Q_h, \{ y_1 \}, \{ y_2 \}, ..., \{ y_l \}\}$ where $y_1, y_2, ..., y_l$ is one optimal order for the graph obtained from $G(X, \Gamma)$

by the cancellation of the vertices of the set Q_h. By definition 1 the dimension of this order is h and $\tilde{\gamma}(k) = 1 + h + q_h$.

The proof follows by repeated use of lemma 2 in a way similar to the one employed above.

Q.E.D.

Since an interaction graph has dimension one if and only if it is a tree [2] the 1-index q_1 is the analogous of Mason's index [9] for undirected graphs The indexes q_h ($h \geqslant 2$) may be regarded as generalizations.

Clearly the determination of the h-index set Q_h is crucial for determining a partition $k \in \tilde{K}'$ for which $\tilde{\gamma}(k) = C_h^*$. The determination of an 1-index set Q_1 may be regarded, for instance, as a covering problem [10] : namely a minimal set of nodes, whose cancellation cuts all circuits of the graph, is searched for.

It is worth while noting that the circuit is regarded as the elementary « structure » which does not allow the graph having dimension equal to one and which, hence, must be « cut ».

For $h \geqslant 2$ no method for determining Q_h is now available. However it is conceivable that it is possible to specify all elementary « structures » which do not allow the graph having dimension equal to h and to determine Q_h by means of a covering algorithm.

An interesting property of the interaction graph G is now easily demonstrated.

Theorem 2. *Consider an interaction graph $G(X, \Gamma)$ with dimension D. Let $h \leqslant D - 1$ and q_h be the h-index of $G(X, \Gamma)$. Then $D \leqslant q_h + h$.*

Proof. The proof follows directly from theorem 1. In fact, clearly, for $h' > h''$ it results $C_{h'}^* \leqslant C_{h''}^*$. Hence, by theorem 1, for $h \leqslant D - 1$, $C_h^* \geqslant C$ or $q_h + h \geqslant D$.

It is also possible to give a direct proof of this theorem. The case $h = 1$ is, for simplicity, considered. Consider a 1-index set Q_1. The section graph G' of $G(X, \Gamma)$ w.r.t. $X - Q_1$ (namely the graph obtained from $G(X, \Gamma)$ deleting the vertices of Q_1 and all the edges emanating from them), is, by definition, a tree (or a forest). Letting $y_1, y_2, \ldots, y_l$ be an optimal order of elimination ($D = 1$) in G' consider an order for G given by $y_1, y_2, \ldots, y_l, y_{l+1}, \ldots, y_M$ where $y_{l+1}, y_{l+2}, \ldots, y_M$ is any order in the set Q_1.

Since a vertex y_j ($1 \leqslant j \leqslant l$) at the time of its elimination is connected to at most one vertex in the set $\{ y_{j+1}, y_{j+2}, \ldots, y_l \}$ and to at most q_1 vertices in Q_1 it is clear that

$$D \leqslant q_1 + 1.$$

The same line of reasoning applies for showing that

$$D \leqslant q_h + h, \qquad h \leqslant D - 1.$$

Q.E.D.

Theorem 2 establishes an interesting relation between the dimension of a graph which plays a central role in nonserial dynamic programming and Mason's index. Also it sets an upper bound to the dimension D which might be used in branch and bound type algorithms for finding optimal orders of elimination.

6. EXAMPLES

EXAMPLE 1 (fig. 3). It results $D = 2$ ($C = 3$) and one optimizing order is $x_1, x_4, x_2, x_3, x_5, x_6$. Clearly it is $C_1 = 6$ (cost when no parametrization is allowed) obtained, for instance, with a partition

$$k = \{\{ x_1, x_2, x_3, x_5, x_6 \}, \{ x_4 \}\}.$$

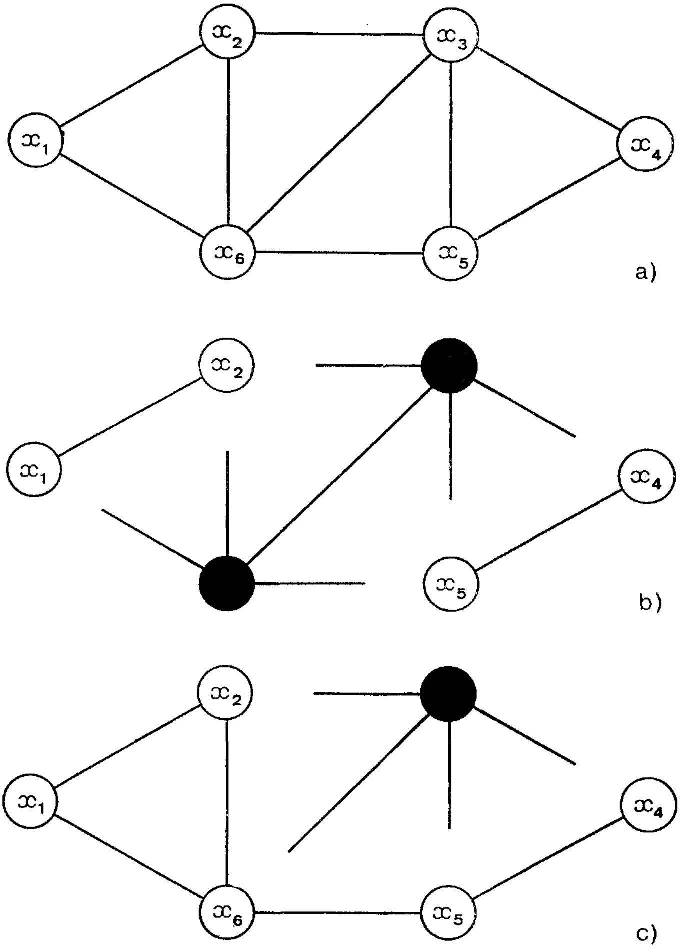

Figure 3
Exemple 1 of section 6

The 1-index q_1 is equal to 2 and 1-index set is, for instance, $Q_1 = \{ x_3, x_6 \}$ (fig. 3 *b*). Then $C_1^* = 4$. Also $\tilde{C}_1 = 4$. This cost is also obtained, for instance, with the partition

$$k = \{\{ x_3 \}, \{ x_4 \}, \{ x_5 \}, \{ x_1, x_2, x_6 \}\} \text{ where } \{ x_3 \} = P^k \quad \text{(fig. 3 } c\text{)}.$$

This example shows that parametrization may be crucial for reducing the computing time for the solution of the optimization problem.

EXAMPLE 2 (fig. 4). It results $D = 2$ ($C = 3$) and one optimizing order is x_1, x_5, x_2, x_3, x_6, x_4. $C_1 = 3$ is obtained, for instance, with a partition $k = \{\{ x_1, x_5\}, \{ x_2 \}, \{ x_3, x_4, x_6 \}\}$. It results $q_1 = 2$ and, for instance, $Q_1 = \{ x_2, x_3\}$ (fig. 4 *b*). Then $C_1^* = 4$. The cost $\tilde{C}_1$ is obtained without parametrization ($\tilde{C}_1 = C_1 = 3$).

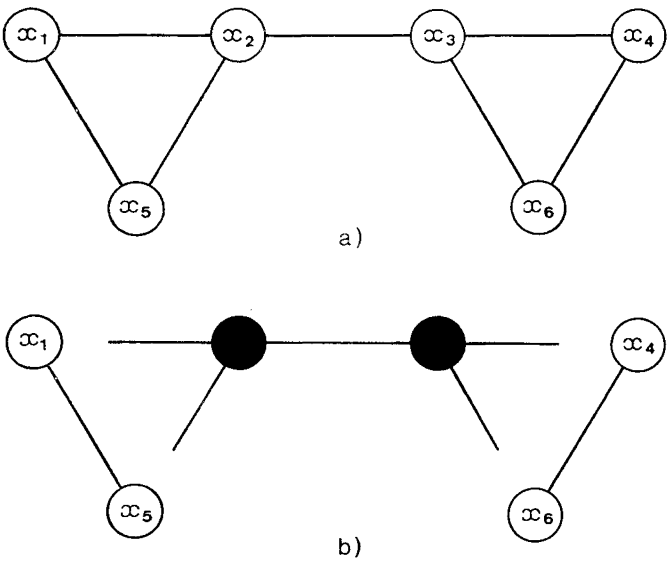

Figure 4
Exemple 2 of section 6

This example shows that parametrization may not help in this case. More than that it is easy to see that parametrization is harmful i.e. it increase the cost. Thus, in order to employ a procedure in which variables are eliminated one by one and the storage limitation $h = 1$ is met, a penalty must be paid.

EXAMPLE 3 (fig. 5). It results $D = 3$, $q_1 = 3$ and a $Q_1 = \{ x_2, x_3, x_7 \}$ (fig. 5 *b*). $q_2 = 1$ and a $Q_2 = \{ x_7 \}$ (fig. 5 *c*).

This example illustrates the relations of theorem 2. The relation $D \leqslant q_h + h$ is a strict inequality for $h = 1$ and an equality for $h = 2$.

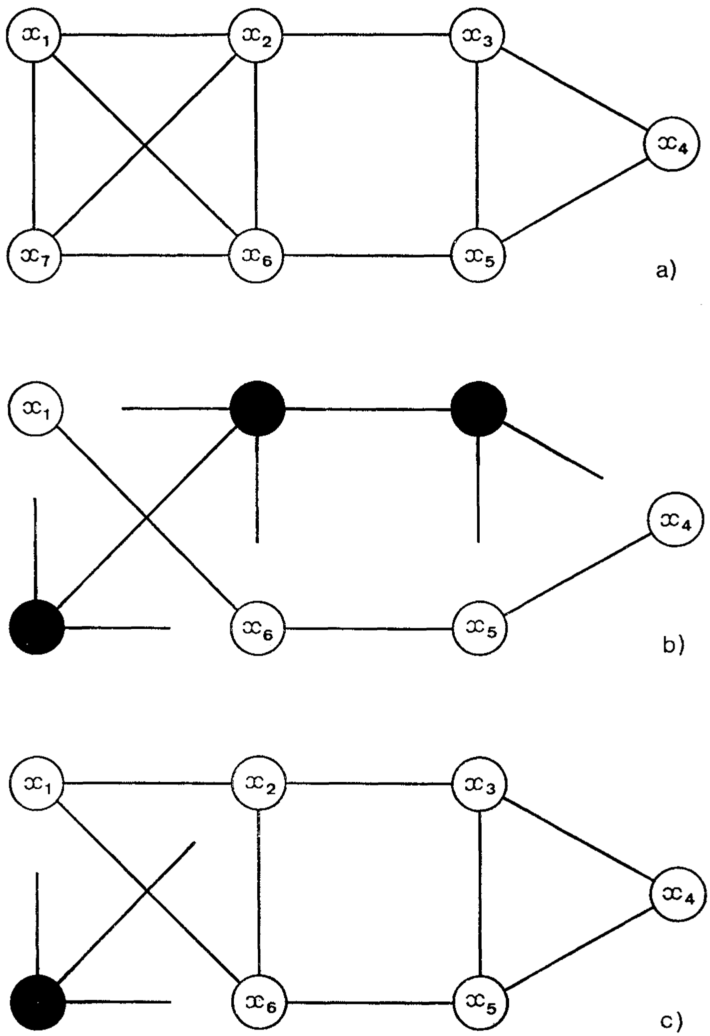

Figure 5

Exemple 3 of section 6

REFERENCES

[1] F. Brioschi et S. Even, « Minimizing the Number of Operations in Certain Discrete Variable Optimization Problems », *Operations Research*, vol. 18, n° 1, jan. 1970, 66-81.

[2] U. Bertele' et F. Brioschi, « A New Algorithm for the Solution of the Secondary Optimization Problem in Nonserial Dynamic Programming », *Journal of Mathematical Analysis and Applications*, vol. 27, n° 3, sept. 1969, 565-574.

[3] U. BERTELE' et F. BRIOSCHI, « Contribution to Nonserial Dynamic Programming », *Journal of Mathematical Analysis and Applications*, vol. 28, n° 2, nov. 1969, 313-325.

[4] U. BERTELE' et F. BRIOSCHI, « A Theorem in Nonserial Dynamic Programming », *Journal of Mathematical Analysis and Applications*, vol. 29, n° 2, feb. 1970, 351-353.

[5] R. ARIS, G. NEMHAUSER et D. WILDE, « Optimizazion of Multistage Cyclic and Branching Systems by Serial Procedures », *A.I.Ch.E. Journal*, 10, 6, november 1964.

[6] G. NEMHAUSER, *Introduction to Dynamic Programming*, Wiley, 1966.

[7] D. WILDE et C. BEIGHTLER, *Foundations of Optimization*, Prentice-Hall, 1967.

[8] C. BERGE, *The Theory of Graphs and its Applications*, Wiley, 1962.

[9] S. MASON, *Some Properties of Signal Flow Graphs*, P.I.R.E., vol. 41, sept. 1953.

[10] L. DIVIETI, « A Method for the Determination of the Paths and the Index of a Signal Flow Graph », VIII International Automation and Instrumentation Exhibition and Convention, Milano, nov. 1964.