Jan Šindelář

Algorithmical complexity of some statistical decision processes. I

Kybernetika, Vol. 16 (1980), No. 4, (345)--356

Persistent URL: http://dml.cz/dmlcz/124890

KYBERNETIKA — VOLUME 16 (1980), NUMBER 4

Algorithmical Complexity of Some Statistical Decision Processes I

JAN ŠINDELÁŘ

Complexity of some types of statistical decision processes based on Bayesian decision functions is studied. Each process is expressed as a sequence of operations as number addition, multiplication, finding inverse matrices etc. Basic type of complexity of such process is characterized by a finite sequence of naturals $\langle n_1, n_2, \ldots \rangle$. Every natural n_i equals the number of executions of the corresponding operation occurring in the process. From this basic type of complexity some other types are derived.

Examples and applications will be given in part II.

1. In the following work these abbreviations (symbols) are used: $N = \{0, 1, 2, \ldots\}$, $N^+ = \{1, 2, 3, \ldots\}$, $Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

All the paper is divided into the sequence of sections, which are subsequently numbered.

STATISTICAL DECISION PROCESSES

2. In this paper algorithmical complexity of several types of decision processes will be studied using the concept of (statistical) decision function. Statistical decision functions have been introduced by Wald (cf. [1]). The most emphasize will be posed on Bayesian (optimal) decision functions. We will also study the change of algorithmical complexity of decision making caused by a simplification of structure dependence of the set of (observed) random variables.

3. Let us introduce an abstract model (borrowed from [4]) of statistical decision making abstract enough to cover all particular procedures which will be explained in the following chapters.

4. Definition D1. Statistical decision problem is a quadruple

$$\Delta = \langle \langle X, \mathscr{X}, P \rangle, \langle Y, \mathscr{Y}, \{P_x\}_{x \in X} \rangle, \langle D, \mathscr{D} \rangle, w \rangle , \tag{1}$$

where $\langle X, \mathscr{X}, P\rangle$ is a probability space over the parameter space X, every $\langle Y, \mathscr{Y}, P_x\rangle$, $x \in X$, is a probability space over the observation space Y, $\langle D, \mathscr{D}\rangle$ is a measurable space over the space of decisions D, and w is the weight function or loss function defined on Cartesian product $X \times D$ and taking non-negative real numbers as its values, here w is supposed to be measurable mapping of the measurable space $\langle X \times D, \mathscr{X} \times \mathscr{D}\rangle$ into the Borel line $\langle E_1, B\rangle$.

In our following investigation the members of the set Y always will be n-tuples of symbols from some set Y_0, i.e. $Y \subseteq Y_0^n$ $(n \in N^+)$. Instead of "observed value $y \in Y$" we will say "observed values $y_1, \ldots, y_n$ from Y_0". Very often the most simple case of loss function is used, i.e., $w(x, d) = 0$, if the decision d is "appropriate" or "the best" with respect to x, and $w(x, d) = 1$ otherwise. Also in the case, when $X = D = \{0, 1\}$ we shall often use this type of loss function, setting $w(0, 0) = w(1, 1) = 0$, $w(1, 0) = w(0, 1) = 1$.

5. Definition D2. Consider a statistical decision problem Δ from D. Decision function δ is measurable mapping from $\langle Y, \mathscr{Y}\rangle$ into $\langle D, \mathscr{D}\rangle$. The value

$$r(x, \delta) = \int w(x, \delta(y))\, \mathrm{d}P_x \tag{2}$$

is called the risk connected with Δ and δ under the condition that the value of parameter is x. Set

$$r_B(\Delta, \delta) = \int r(x, \delta)\, \mathrm{d}P = \iint w(x, \delta(y))\, \mathrm{d}P_x\, \mathrm{d}P\,, \tag{3}$$

$$r_M(\Delta, \delta) = \sup \{r(x, \delta) \mid x \in X\}\,.$$

Then $r_B(\Delta, \delta)$ is called the Bayes risk and $r_M(\Delta, \delta)$ the minimax risk connected with the problem Δ and decision function δ. A decision function δ_0 is called a Bayes solution (a minimax solution, resp.) to the statistical decision problem Δ, if

$$r_B(\Delta, \delta_0) \leqq r_B(\Delta, \delta) \tag{4}$$

$$(r_M(\Delta, \delta_0) \leqq r_M(\Delta, \delta)\,, \quad \text{resp.})$$

for all decision functions δ.

6. In case of the zero-one loss function the Bayes risk reduces to the probability of error weighted with respect to the apriori distribution and the minimax risk reduces to the maximum of probabilities of error of both types.

7. However, in actual decision making (in statistics) usually probabilities $P_x(y)$ are not known. It is necessary to calculate them, or in most cases to estimate them. So we can say, that real decision process consists of the three (or four) parts:

(theoretical choosing of a decision function)

I. observation of empirical values, calculation or estimation of the values of probabilities necessary for construction of the decision function.

II. construction of the decision function.

III. selection of a decision on the ground of application of decision function to the next observed empirical values.

8. Example 1. Suppose we have found that in some decision process the decision function is

$$\delta_{p,q}(y_1, y_2) = \begin{cases} 0, & \text{if } \frac{1}{2}(p + q^2) < y_1 + y_2 \\ 1 & \text{otherwise}, \end{cases}$$

where p, q are unknown parametres, y_1, y_2 are observed values. (It is the construction of the decision function.)

I. We calculate (or estimate) p, q.

II. We calculate $\frac{1}{2}(p + q^2) = d$, then the decision function, that we shall use (in practice) will be

$$\delta(y_1, y_2) = \begin{cases} 0 & \text{if } d < y_1 + y_2 \\ 1 & \text{otherwise}. \end{cases}$$

III. If $\bar{y}_1, \bar{y}_2$ are the next observed values, we find $\bar{y} = \bar{y}_1 + \bar{y}_2$ and then verify the inequality $d < \bar{y}$ if this inequality holds, the decision is 0; otherwise the decision is 1.

In this work we shall study (algorithmical) complexity of some parts (I, II or III) of several types of statistical decision processes. It is not possible always to determine precise boundaries and to divide explicitly a decision process into the three parts according to the rules mentioned above. The separations mentioned below are only some of all the possible. The details and comments on theoretical choosing of decision function see in [1].

9. In practice many times the following fact has been observed: if we assume, that $y_1, \ldots, y_n$ $(\in Y_0)$ were mutually statistically independently, then the decision process is more simple than in the case when the independence of observations is not assumed. The same effect has been observed also in the case when only some observations are independent from others.

More strictly we can say:

It has been observed that the simplification of dependence structure of random variables occurring in the decision process simplifies this decision process.

We suppose that when "the state of environment" is $x \in X$, then observations $y_1, \ldots, y_n$ are obtained by means of a random mechanism defined by random variables $Y_1, \ldots, Y_n$ (with distribution defined by probability P_x).

We shall distinguish these three cases:

Z1. The statistical dependence between $Y_1, \ldots, Y_n$ is arbitrary.

Z2. Random variables $Y_1, Y_2, \ldots, Y_n$ are divided into several blocks; the variables in every block can be dependent, variables from different blocks are independent; more precisely

a) we shall assume $2 \leqq q \leqq n$ numbers $i_1 \ldots, i_q \in N^+$ so that $i_1 + i_2 + \ldots + i_q = n$ and we put $i_0 = 0$;

b) we divide random variables $Y_1, \ldots, Y_n$ in q blocks

$$B_l = \{Y_{i_0+i_1+\ldots+i_{l-1}+1}, Y_{i_0+i_1+\ldots+i_2+i_l}\}, \quad \text{where} \quad l = 1, 2, \ldots, q$$

(then l-th block contains i_l variables and every Y_i $(i = 1, \ldots, n)$ occurs in just one block);

c) we shall assume, that blocks (sets) B_l $(l = 1, 2, \ldots, q)$ of random variables are mutually independent.

Z3. $Y_1, Y_2, \ldots, Y_n$ are mutually independent.

It is obvious that the case Z3 is a particular case of Z2, when $q = n$ and $i_1 = i_2 = \ldots = i_q = 1$.

ALGORITMICAL PROCESSES AND THEIR COMPLEXITY

10. The basis of every decision process (based on statistical decision functions) is a certain strictly defined method (procedure). To be able to study its complexity, we must specify the concepts "procedure" and "complexity".

The concept "procedure" has been precised and its properties formalized in many ways – as a normal (Markov) algorithm, Turing machine, recursive function etc. (cf. [2]).

We shall proceed in the following way. The basis of a statistical decision process is some calculation, i.e. a decision process (mostly) consists in a number of mathematical operations. Hence, as a base of the following considerations we choose a set $S = \{s_1, s_2, \ldots, s_k\}$ of these operations, and every decision process (or its part) we write as a sequence T of operations from S. By the complexity of this decision process (or its part) we shall understand the complexity of sequence T in this sense: for every operation $s \in S$ we determine how many times it has been used (executed) in realization of operations from T; if the operation s_1 was executed n_1-times, operation s_2 n_2-times, ..., operation s_k n_k-times, then as the complexity of the sequence T we shall understand k-tuple of naturals $\langle n_1, n_2, \ldots, n_k \rangle$.

11. It is obvious, that the complexity of a (statistical) decision process is not determined unambiguously by the method mentioned above.

For example, if we want to calculate the value of an expression $(a - b) \cdot (a + b)$ either we can calculate step by step $a - b$, $a + b$, $(a + b) \cdot (a - b)$ – it means to

execute the operation + one times, operation − one times and operation · one times, or we can use the equality $(a + b) . (a - b) = a^2 - b^2 = a . a - b . b$ and, step by step, calculate $a . a$, $b . b$, $a . a - b . b$ — it means to execute the operation + zero times, − once and · twice.

Remark. Some relation can exist between operations from S. For example, if s_i is the multiplication and s_j addition of real numbers and s_l is the scalar product of two n-dimensional vectors, then operation s_l can be written as some executions of s_i, s_j.

If we replace the set S of operations by a set of other operations, then it is possible that the complexity of investigated decision processes is also changed.

The type of complexity described above is some basic type. In the following we shall derive from this basic type other types of complexity, so this type we shall call elementary complexity.

Let us approach, now, to a precise description of the ideas mentioned above.

12. Definition D3. By the basic set of operations is called the set S_1, which elements are these operations:

1. operations on real numbers

$$a + b,\ a - b,\ a . b,\ a : b,\ |a|,\ a < b,\ a^2,\ \log a,\ {}_{\lceil}a_{\rfloor} \tag{5}$$

where

$${}_{\lceil}a_{\rfloor} = \min \{z \in Z \mid a \leqq z\} ;$$

other operations are used in usual sense and with usual domains,

2. operations on matrices over real numbers

$$A + B,\quad A - B,\quad A^{-1} = \operatorname{inv} A,\quad \det A \tag{6}$$

3. operation consisting in inscribing of a symbol c into the j-th cell of memory (storage)

$$U_j(c) .$$

Operations from S_1 are called basic operations.

We shall denote operation $a + b$ as s_1, $a - b$ as s_2, $a . b$ as s_3, $a : b$ as s_4, $|a|$ as s_5, $a < b$ as s_6, a^2 as s_7, $\log a$ as s_8, ${}_{\lceil}a_{\rfloor}$ as s_9, $A + B$ as s_{10}, $A . B$ as s_{11}, A^{-1} as s_{12}, $\det A$ as s_{13}.

We put $F_0 = 13$. An operation $U_j(c)$ we denote as $s_{13+j} = s_{F_0+j}$ $(j \in N^+)$.

Remark. In many calculations it is necessary to preserve some values in memory during some time, hence we introduce the operation of inscribing of some symbol into storage (memory); we do not ascribe any formal description to this operation, because they usually are not self-explanatory. For the purposes of this work the opera-

tion $U_j(c)$ can be understood as an identity mapping, i.e. we can put $U_j(c) = c$ for every symbol c (and every $j \in N^+$).

It is obvious, that the set S_1 is infinite. When studying a decision process we shall always use only a finite number of cells of memory and so we shall use only a finite number of operations from S_1.

13. For the purposes of this paper we shall understand under an algorithmical process any finite sequence of operations from S_1; more precisely.

Definition D4. A finite sequence $E = \langle e_1, e_2, \ldots, e_n \rangle$ will be called an algorithmical process, if:

1. $n \in N^+$,
2. each e_i is a real number or
   a pair of real numbers or
   a matrix of real numbers or
   a pair $\langle s, e \rangle$ where $s \in S_1$ and e is a result of one application of the operation S to some members of sequence $\langle e_1, e_2, \ldots, e_{i-1} \rangle$, $(i = 1, 2, \ldots, n)$,
3. at least one member of sequence E has a form $\langle s, e \rangle$ where $s \in S_1$.

Members of sequence E we shall call members of the algorithmical process E.

Remark. If we want to apply an operation s_i to the result of application of s_j on a number e, i.e. to find $s_i(s_j(e))$, we can describe it by

$$\ldots\langle s_j, s_j(e)\rangle, s_j(e), \langle s_i, s_i(s_j(e))\rangle \ldots .$$

Example 2. We want to calculate $a/b + a^2$ where a, b are real numbers, $b \neq 0$. We shall proceed like this:

1. numbers a, b are put into cells 1,2 of memory;
2. we calculate a/b, the result is put into the cell 2 of memory;
3. we calculate a^2 and put it into the cell 1;
4. we calculate $a/b + a^2$.

The corresponding algorithmical process can be like this:

$$(7) \qquad E = \langle a, b, \langle U_1, a\rangle, \langle U_2, b\rangle; \langle :, a/b\rangle, a/b, \langle U_2, a/b\rangle;$$
$$\langle \cdot^2, a^2\rangle, a^2, \langle U_1, a^2\rangle; \langle +, a/b + a^2\rangle; \rangle ;$$

the beginning of E can also be this:

$$(8) \qquad \langle a, b, \langle U_1, a\rangle, a, \langle U_2, b\rangle, b; \ldots\rangle$$

(particular parts of our calculation are in algorithmical process separated by semicolon).

For $a = 6$, $b = 3$ the algorithmical process mentioned above can be written like this:

$$E = \langle 6, 3, \langle U_1, 6\rangle, \langle U_2, 3\rangle; \langle :, 2\rangle, 2, \langle U_2, 2\rangle; \langle \cdot^2, 36\rangle, 36, \langle U_1, 36\rangle; \langle +, 38\rangle; \rangle . \tag{9}$$

A study of relations between algorithmical process and Turing machines (or normal algorithms, etc.) is rather complicated and so it is not included within the scope of this work.

Definition D5a. A result $V(E)$ of an algorithmical process $E = \langle e_1, \ldots, e_n\rangle$ is

1. e_n, if e_n is a real number of a pair of real numbers or a matrix,
2. a member e of pair $\langle s, e\rangle$ if $e_n = \langle s, e\rangle$ and $s \in S_1$.

14. Definition D5b. To an algorithmical process $E = \langle e_1, \ldots, e_n\rangle$ we ascribe a number $F(E)$ as follows:

$$F(E) = F_0 + \max\left[\{0\} \cup \{m \mid \langle U_m, e\rangle \in \{e_1, e_2, \ldots, e_n\}\}\right] . \tag{10}$$

It is obvious that in an algorithmical process only operations $s_1, s_2, \ldots, s_{F(E)}$ can occur but none of operations s_m for $m > F(E)$. If $F(E) > F_0$ then the operation $s_{F(E)} = U_{F(E)}$ occurs in E. The number F_0 is defined in the paragraph 12.

15. Definition D6. By elementary complexity of an algorithm process $E = \langle e_1, \ldots, e_n\rangle$ we shall understand $F = F(E)$-tuple $\langle n_1, \ldots, n_F\rangle$, where for $i = 1, 2, \ldots, n$ $n_i (\in N)$ equals to the number of pairs of a form $\langle s_i, \cdot\rangle$ occurring in E. The elementary complexity of E can be found, if we determine how many times we have applied every operation $s_1, \ldots, s_{F(E)}$ in E.

Remark. Here we commit to some simplification. We assume that the addition of two one-digit numbers is equally complicated (difficult) as an addition of two ten-digit (or thousand-digit) numbers, but in real computation it is not true. The same situation is with respect to the other operations.

16. Hence, when studying other types of complexity the complexity of an algorithmical process will be always a finite sequence of real numbers, which will be defined by the number of operations, necessary to the realization of the algorithmical process in question.

Remark. If $\langle n_1, \ldots, n_F\rangle$ is the (elementary) complexity of an algorithmical process E and $n_i = 0$ $(i \in \{1, 2, \ldots, F\})$, then the operation s_i does not occur in any member of E. Hence, important for us are only those n_i from $\langle n_1, \ldots, n_F\rangle$ which do not equal zero.

17. There are some natural requirements that should be satisfied by the concept of complexity (if we express the complexity as a finite sequence of real numbers):

a) If $\langle m_1, \ldots, m_F \rangle$ is the complexity of an algorithmical process, then $m_i \geqq 0$ $(i = 1, \ldots, F)$.

b) If E, E' are two algorithmical processes and the elementary complexity of E is n-times greater than the elementary complexity of E', then the complexity of E is n-times greater than the complexity of E.

c) We can easily see that if $E = \langle e_1, \ldots, e_n \rangle$, $E' = \langle e'_1, \ldots, e'_n \rangle$ are two algorithmical processes, then $E'' = \langle e_1, e_2, \ldots, e_n, e'_1, \ldots, e'_{n'} \rangle$ is an algorithmical process and elementary complexity of E'' is the sum of elementary complexities of E and E'.

If we have some procedure that from elementary complexity of E, E', E'' compute the complexity of E, E', E'', then the complexity of E'' is the sum of complexities of E and E', where E, E', E'' are as mentioned above.

d) If $\langle m_1, \ldots, m_F \rangle$, $\langle m'_1, \ldots, m'_F \rangle$ are the complexities of algorithmical processes E, E' and $m_i \leqq m'_i$ (for $i = 1, 2, \ldots, F$), and if we compute, by the same procedure, new (types of) complexities $\langle n_1, \ldots, n_{F'} \rangle$, $\langle n'_1, \ldots, n'_{F'} \rangle$ of E and E' then $n_i \leqq n'_i$ $(i = 1, 2, \ldots, F')$.

As can be shown, for every type of complexity defined below, this four requirements are fulfilled.

18. In many applications we can assume that the realization of an addition or substraction of two real numbers is equally difficult. Also the realizations of multiplication, division and square taking can be treated as equally difficult.

For a more detailed discussion on these problems cf. [3].

Let us consider a set of operations $S = \{t_1, \ldots, t_k\}$ $(k \geqq 2)$ and the complexity $\langle n_1, \ldots, n_k \rangle$ of an algorithmical process E with respect to the operations $t_1, \ldots, t_k$.

a) If we suppose (or if we determine or decide) that one realization of an operation from S (e.g. operation t_1) can be equally difficult as c_2 (joint) realizations of the operation t_2, c_3 realizations of the operation $t_3, \ldots, c_k$ realizations of the operations t_k, then we can consider the complexity of E only with respect to operations $S' = \{t_2, \ldots, t_k\}$ and write it as the $(k-1)$-tuple

$$\langle n_1 c_2 + n_2,\ n_1 c_3 + n_3, \ldots, n_1 c_k + n_k \rangle \quad \text{or as a } k\text{-tuple} \tag{11}$$

$$\langle 0,\ n_1 c_2 + n_2,\ n_1 c_3 + n_3, \ldots, n_1 c_k + n_k \rangle .$$

b) If we find out that one realization of t_1 can be replaced by c_2 (joint) realizations of t_2 and c_3 realizations of t_3 etc., then the way of transformation is the same as in the case a).

c) We can also put $c_1 = -1$, then $n_1 c_1 + n_1 = 0$ and we can write the complexity of E with respect to S' as $\langle n_1 c_1 + n_1,\ n_1 c_2 + n_2, \ldots, n_1 c_k + n_k \rangle$. Consequently,

if $\langle n'_1, \ldots, n'_k \rangle$ is the complexity of E with respect to S', then $n'_i = n_1 c_i + n_i$ $(i = 1, 2, \ldots, k)$ and the transition from operations from S to operations from S' implies the change of complexity of E' determined by k-tuple of numbers $\langle c_1, c_2, \ldots, c_k \rangle$ where $c_1 = -1$, $c_i \geqq 0$ for $i = 2, 3, \ldots, k$. If the transition from operations S to operations S' cannot be done or is not reasonable, we can characterize it by k-tuple $\langle 0, 0, \ldots, 0 \rangle$. For example if $S = \{U_j, +, -\}$ then it can be inappropriate to replace the operation U_j (inscribing to j-th cell of memory) by operations $+, -$. However, if we want to find the cost of some process, then this replacement can be of a sense.

d) With respect to the considerations in a)–c) we define:

Definition D7. The matrix $C = (c_{ij})_{i,j=1,k}$ will be called a transition tabel of order k, if

1. $k \in N^+$;
2. for $i \neq j$, $c_{ij} \geqq 0$ holds $(i, j = 1, 2, \ldots, k)$;
3. $c_{jj} = 0$ or $c_{jj} = -1$, $(j = 1, 2, \ldots, k)$;
4. if for some $j \in \{1, \ldots, k\}$ $c_{jj} = 0$, then $c_{ij} = 0$ for $i = 1, 2, \ldots, k$;
5. if for some $j \in \{1, \ldots, k\}$ $c_{jj} = -1$, then there is $i \in \{1, 2, \ldots, k\}$ so that $c_{ij} > 0$.

i-th row of C shall be written as c_i, so $c_i = \langle c_{i1}, c_{i2}, \ldots, c_{ik} \rangle$.

19. Definition D8. Let C be a transition table of the order F. For every $G \in N^+$ and every $i_1, i_2, \ldots, i_G \in \{1, 2, \ldots, F\}$ an operation $Pr_{i_1,\ldots,i_G} : E_1^F \to E_1^F$ is defined as follows:

1. if $G = 1$ then

$$Pr_{i_1}\langle n_1, n_2, \ldots, n_F \rangle = \langle n'_1, n'_2, \ldots, n'_F \rangle , \tag{12}$$

where

$$n'_{j_1} = n_j + c_{i_1 j} \cdot n_{i_1} , \tag{13}$$

$$\langle c_{i_1 1}, c_{i_1 2}, \ldots, c_{i_1 F} \rangle = c_{i_1}$$

2. if $G \geqq 2$ then

$$Pr_{i_1, i_2, \ldots, i_G}\langle n_1, n_2, \ldots, n_F \rangle = Pr_{i_G}\left(Pr_{i_1 \ldots i_{G-1}}\langle n_1, n_2, \ldots, n_F \rangle\right) . \tag{14}$$

Operation Pr_{i_1} is called an elementary transition, operation $Pr_{i_1 \ldots i_G}(G \in N^+)$ a transition.

The sense of this definition is obvious from D7 and paragraph 18.

20. Definition D9. Let E be an algorithmical process, $F = F(E)$ and $N_0 = \langle n_1, \ldots, n_F \rangle$ elementary complexity of E. Let C be a transition table of the order F.

By algorithmical complexity of E with respect to C we shall understand any $M = \langle m_1, m_2, \ldots, m_F \rangle$ satisfying the following conditions:

a) $M = N_0$, or

b) $M = Pr_{i_1, i_2, \ldots, i_G}(N_0)$ for some $i_1, \ldots, i_G \in \{1, \ldots, F\}$, $G \in N^+$.

We shall write

$$M = Com\, E \text{ (for } N = N_0) \tag{15}$$

$$M = Com\, C_{i_1, \ldots, i_G}(E).$$

Definition D10. Let E be an algorithmical process, $F(E) = F$, C – transition table of order F. Let $H \in N^+$ and $j_1, j_2, \ldots, j_H \in \{1, 2, \ldots, F\}$ be different naturals.

By algorithmical complexity of E with respect to the set $S = \{s_{j_1}, s_{j_2}, \ldots, s_{j_H}\}$ of operations (and with respect to C) we shall understand the H-th tuple $\langle m_1, \ldots, m_H \rangle$ iff there is a complexity $\langle n_1, \ldots, n_F \rangle$ of E with respect to C such that:

a) $n_{j_1} = m_{j_1}$, $n_{j_2} = m_{j_2}, \ldots, n_{j_H} = m_{j_H}$;

b) $n_i = 0$ for $i \neq j_1, j_2, \ldots, j_H$.

(Compare D10 with the remark in section 16.)

21. In our following study of statistical decision processes an important role plays the number of occupied cells of memory, but not a number of repetitive use of every cell. Hence we introduce a new type of complexity.

Definition D11. Let E be an algorithmical process, $F(E) = F$, $\langle n_1, \ldots, n_F \rangle = Com\, E$. By elementary reduced complexity (elementary r-complexity) of E we shall understand $(F_0 + 1)$-tuple $\langle n'_1, n'_2, \ldots, n'_{F_0+1} \rangle$, where

a) $n'_i = n_i \ldots$ for $i = 1, 2, \ldots, F_0$;

b) $n'_{F_0+1} = 0 \ldots$ for $F_0 = F$;

c) n'_{F_0+1} is the number of members of $\langle n_{F_0+1}, \ldots, n_F \rangle$ not equal to zero ... for $F_0 < F$.

Remark. F_0 is defined in section 12. Compare D11 with section 14.

Definition D12. Let E be an algorithmical process, $N_0 = \langle n_1, \ldots, n_{F_0+1} \rangle$ an elementary r-complexity of E, C a transition table of the order $F_0 + 1$. $M = \langle m_1, m_2, \ldots, m_{F_0+1} \rangle$ is called r-complexity of E (with respect to C), if

a) $M = N_0$ or

b) $M = Pr_{i_1, i_2, \ldots, i_G}(N_0)$ for some $i_1, i_2, \ldots, i_G \in \{1, 2, \ldots, F_0 + 1\}$, $G \in N^+$. We shall write

$$M = Com\, Pr(E) \quad \text{for} \quad M = N_0 \tag{16}$$

$$M = Com\ Pr\ C_{i_1 \ldots i_G}(E)\,.$$

Elementary r-complexity of every decision process is a $(F_0 + 1)$-tuple, so when studying every (statistical) decision process the same transition table can be applied.

Definition D13. Let E be an algorithmical process, C-transition table of the order $F_0 + 1$. Let $H \in N^+$ and $j_1, j_2, \ldots, j_H \in \{1, 2, \ldots, F\}$ be different integers.

By r-complexity of E with respect to the set $S = \{s_{j_1}, s_{j_2}, \ldots, s_{j_H}\}$ of operations (and with respect to C) we shall call the H-tuple $\langle m_1, \ldots, m_H \rangle$ iff there is the r-complexity $\langle n_1, \ldots, n_{F_0+1} \rangle$ of E (with respect to C) such that

(17) a) $\quad n_{j_1} = m_{j_1},\ n_{j_2} = m_{j_2}, \ldots, n_{j_H} = m_{j_H}$

b) $\quad n_i = 0 \quad \text{for} \quad i \neq j_1, j_2, \ldots, j_H\,.$

(Compare with D11.)

Remark. These two concepts of complexity are used in literature: space complexity and time complexity (cf. [3]). The concept of algorithmical complexity corresponds to the time complexity.

If $\langle n_1, \ldots, n_{F_0}, n_{F_0+1} \rangle$ is r-complexity of some process E then n_{F_0+1} corresponds to the space complexity of E.

Hence, our concepts of algorithmical complexity and r-complexity are generalizations of the concepts of time and space complexity.

As far as a reduction of space complexity to time complexity and vice versa is concerned, we can (roughly) say the same as in section 18 about (general) reduction between operations; cf. the last part of 18c.

22. When a misunderstanding is excluded, we shall say "complexity" instead of "algorithmical complexity" or "r-complexity", and "elementary complexity" instead of "elementary r-complexity" or "elementary algorithmical complexity".

Applying the transition operation to the complexity of some algorithmical process E we obtain (new) complexity of E. However, intuitively is obvious that a complexity should be determined unambigiously. In order to achieve this goal we always write the particular parts of a statistical decision process as a very detailed procedure (or computation), and then we describe how to compile the corresponding algorithmical process and combine its elementary complexity. We always use the same transition table and the same transition.

23. Every decision process will be separated into three parts I, II, III as described in section 8.

We distinquish between the three levels Z1, Z2, Z3 of the statistical dependence of random variables (cf. section 9), hence, there are three types of simplification of this dependence: we can transit from Z1 to Z2, from Z1 to Z3 and from Z2 to Z3. This types of simplifications will be denoted by Z12, Z13, Z23.

Studying statistical decision processes we consider the observed values $y_1, \ldots, y_n$ (cf. sections 4, 9). For different values of n we (usually) obtain different decision functions and, hence, also different algorithmical processes $E(n)$. So that to different values of n we (usually) obtain different complexities $M(n)$ of $E(n)$.

If we consider in addition the different kinds of dependence structure of random variables $Y_1, \ldots, Y_n$, we obtain two sequences $E(n)$, $E'(n)$ of algorithmical processes and two sequences of its complexities $M(n)$, $M'(n)$.

We want to characterize the change of complexity (of a statistical decision process) connected with the simplification of dependence structure of random variables (occurring in the decision process). So we define:

Definition D14. Let $F \in N^+$,

$$M(n) = \langle m_1(n), m_2(n), \ldots, m_F(n) \rangle , \tag{18}$$

$$M'(n) = \langle m'_1(n), m'_2(n), \ldots, m'_F(n) \rangle$$

be complexities of algorithmical processes $E(n)$, $E'(n)$, $n = 1, 2, 3, \ldots$. Let $i \in \in \{1, 2, \ldots, F\}$. We say, that the saving of operation s_i connected with the transition from the sequence $E(n)$ of algorithmical processes to the sequence $E'(n)$ of algorithmical processes is asymptotically characterized by a function $h : N \to E_1$, if

$$\lim_{n \to \infty} \frac{m_i(n)}{m'_i(n)} \cdot \frac{1}{h(n)} \tag{19}$$

exists and equals 1.

Remark. It is obvious that the function h from D14 need not be determined unambiguously. However, important (for us) is the type of this function (polynomial, exponential, etc.) and not a function itself.

(Received August 31, 1979.)

REFERENCES

The list of references will be presented in the part II.

Jan Šindelář, Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 182 08 Praha 8. Czechoslovakia.