Automatica, Vol. 6, pp. 207-219. Pergamon Press, 1970. Printed in Great Britain.

Heuristic Self-Organization in Problems of Engineering Cybernetics*

Auto-organisation heuristique dans les problèmes de cybernétique technique

Heuristische Selbstorganisation bei Problemen der technischen Kybernetik

Эвристическая самоорганизация В проблемах технической кибернетики

A. G. IVAKHNENKO†

An analysts of engineering cybernetics shows that the current deterministic approach can only solve comparatively simple problems. A new approach called heuristic self-organization is needed for solving complex problems.

Summary—The systems, or programs, of heuristic self-organization are defined as those which include the generators of random hypotheses, or combinations, and several layers of threshold self-sampling of useful information. The complexity of combinations increases from layer to layer. A known system, Rosenblatt's perceptron, may be taken as an example.

The Group Method of Data Handling (GMDH) based on the principles of heuristic self-organization is developed to solve complex problems with large dimensionality when the data sequence is very short. Two examples are given to illustrate how this method applies to problems of predicting random processes and to identifying characteristics of a multiextremum plant.

One: Heuristics are groundless decisions which have no mathematical proofs. They give us the results which are only good enough for practice, but they are not the best ones.

The other: No ! Heuristics are decisions in afield irrelevant to the subject and competence of mathematics. The results of heuristics are often much better than those which can be obtained from a formalized approach.

HEURISTIC SELF-ORGANIZATION

A DISPROPORTIONAL development of two basic parts of cybernetics may now be seen: a dominance of work using deterministic approaches and an almost complete lack of work concerning practical use of heuristic self-organization. Although the ideas of self-organization have been discussed many times in the well known papers of N. Wiener, J. von Neumann, G. Pask, R. Ashby, S. BEER [1], A. Ya. Lerner, V. S. Fain and others, the papers and books of the sixties only repeat concepts which had already been stated 10 or 20 years ago. There has been almost no progress in this field.

But it is clear that only self-organization and ideas associated with it can justify the very existence of cybernetics as a science on the general approach to problems which are different by their nature. The present-day deterministic approach is associated with the analysis of system inputs and outputs. The specific features of each particular problem is of main importance, and this results in a situation where all problems related to computers are related to cybernetics. Such a viewpoint and the more universal original idea of cybernetics, given by N. Wiener, are at variance. Certain methods often associated with cybernetics, such as the "black box" idea, are now considered not to be constructive. Instead, self-organization concepts must re-establish the general ideas of cybernetic sand show their constructiveness.

Moreover, too much confidence in the deterministic approach nonpluses us. It is now clear that it is impossible to solve many practical problems, such as the problem of automatic synchronous translation from one language into another, or the problem of classification when 200-300 classes are involved, and so on, by deterministic methods. Self-organization must be used to find a way out of this impasse. However, in order to do this, it is necessary to begin with practical problems, and having decided to make an attempt, we took some first steps by solving various problems of pattern recognition, of random processes prediction, and multiextremum plant identification [2–6].

For the present we cannot given an exact mathematical definition of "self-organization", but it is clear that self-organization is necessary when it is impossible to trace all input-output relations

* Received November 1968; revised 22 April and 1 October 1969. The original version of this paper was not presented at any IFAC meeting. It was recommended for publication in revised form by associate editor B. Gaines.

† Institute of Cybernetics, Kiev, USSR.

throughout an entire system which is too complex for the purpose [1]. Therefore, we must use the notion of general "integral influences" which act upon a network of components, each having its own "elementary algorithm" of action.

The integral influence is defined to be one which is not found from an analysis of a complex system. It does not use information about the state of each particular component of the system, but it is chosen by the summary result of active responses.

Automatic control theory, as mathematics itself, has been developed as a pure deductive theory for investigating causes and consequences, inputs and outputs. The arrows and squares of block diagrams are embedded in our consciousness so deeply that we may say for certain that the control theory actually impedes the development of spontaneous processes control.

As previously indicated, self-organization is the art of controlling spontaneous processes through use of integral influences.

An income tax is a good example of integral influence, because market spontaneity can be controlled by changing a nonlinearity—the income tax. If the nonlinearity is high, the income tax may become an integral action of a threshold type: nothing from the poor and all from the rich.

The simplest realization of integral influences in cybernetics, for example, in the perceptron [7], is a threshold unit permitting only some inputs to pass. In fact, we have used this simplest type of integral influence in solving the three interpolation problems mentioned above.

Finally, self-organization should be associated with heuristics which we mean to be conjectures in evaluating a course of problem solution by man. In this respect, self-organization resembles a sandwich: after mathematical processing of information, a "layer" of heuristic evaluation of the results follows, and this process is repeated several times. Man controls the course of the solution by continuously directing its way to desired results by means of integral influences. That is why heuristic self-organization ensures an accuracy which could not be reached by the use of routine mathematical methods. The influence of heuristics is so potent that it is possible to apply a mathematical tool of less sophistication than those which are usually used. Heuristics are creative thought processes of men, and their results are decisions. They are connected with the wishes of man, with factors associated with his motives. They pertain neither to the subject, nor to competence of mathematics, therefore no mathematical tool can be perfected to compensate for them nor can one even be compared with them with regard to their effect on the accuracy of a solution.

The history of civilization is full of examples where various control problems have been solved by self-organization. For example, the problem of raising the yield of farms with a minimum of human labour has been solved so successfully that soon only 5 per cent of world population will be involved in farming.

Some scientists tell us that the problem of "large" or "complex" systems is a new one. but it is only natural to advise the scientists interested in the complex plant control to investigate the experience of the mankind in this respect. It may seem strange, but mathematics has no tool capable of solving practical complex system problems. Mathematics is not prepared to meet the challenge of problems involving self-organization.

"HYPOTHESIS OF SELECTION" IN COMPLEX SYSTEMS THEORY

The threshold type of integral influence, which may be considered as "an examination" are widely used in the mass selection of plants and animals. To obtain plants which have certain desired characteristics, for example, a portion of seeds is selected from several generations of the plants in which these properties are more predominant than in others. In what follows we use a similar "hypothesis of selection" process to solve engineering cybernetics problems. This hypothesis states that methods of selection are the best for solving interpolation problems of prediction, pattern recognition or identification.

The "hypothesis of selection" has a probabilistic character. The more the value of a given variable exceeds the threshold value, the more is the probability that it is just this variable which provides us with information about the best, or optimal, decision. Therefore, each threshold has a single optimal setting corresponding to the maximum of the accuracy in the result.

For example, when selecting plants the following three questions are answered in a purely heuristic way:

(1) Which seeds must be used for the first crop? This is the first heuristic: Choosing elementary algorithms for producing input signals.

(2) Which criteria should be used to select the best plants ? This is the second heuristic: Choosing criteria for self-samplings.

(3) According to which laws are the crossing of the plants to be determined? This is the third heuristic: Choosing laws for generating combinations.

Having answered these questions we can also answer the next two:

(4) What portion of seeds is to be selected in each generation? This illustrates that there is an optimum level for each threshold of self-sampling.

(5) After which generation is the selection to be stopped? This question must be answered because after a certain number of generations, the desired plant characteristics begin to degenerate.

The systems of heuristic self-organization considered below are based on the same sort of heuristics that have been illustrated above.

THE PERCEPTRON AND OTHER EXAMPLES OF SYSTEMS HAVING HEURISTIC SELF-ORGANIZATION

Let us define the system, or a program, of heuristic self-organization as a system which has a multilayered or a hierarchical algorithm, i.e. a structure where self-sampling thresholds of useful data are used in each layer. To make these self-samplings more effective, one or several generators of random combinations of arguments are provided, so that the complexity of variables increases with each successive layer. If the combinations are not numerous, all of them may be subject to exhaustive search. It is clear from the example above, that the selection of seeds may be used as an algorithm for a heuristic self-organization system.

Figure 1 shows several other examples of self-organizing systems which are known in engineering. The first example in Fig. la is the well-known perceptron, the model of the brain perception function, designed by ROSENBLATT [7]. Random connections of links between perceptron layers are considered to be a kind of generation of new combinations, to complete the analogy previously made.

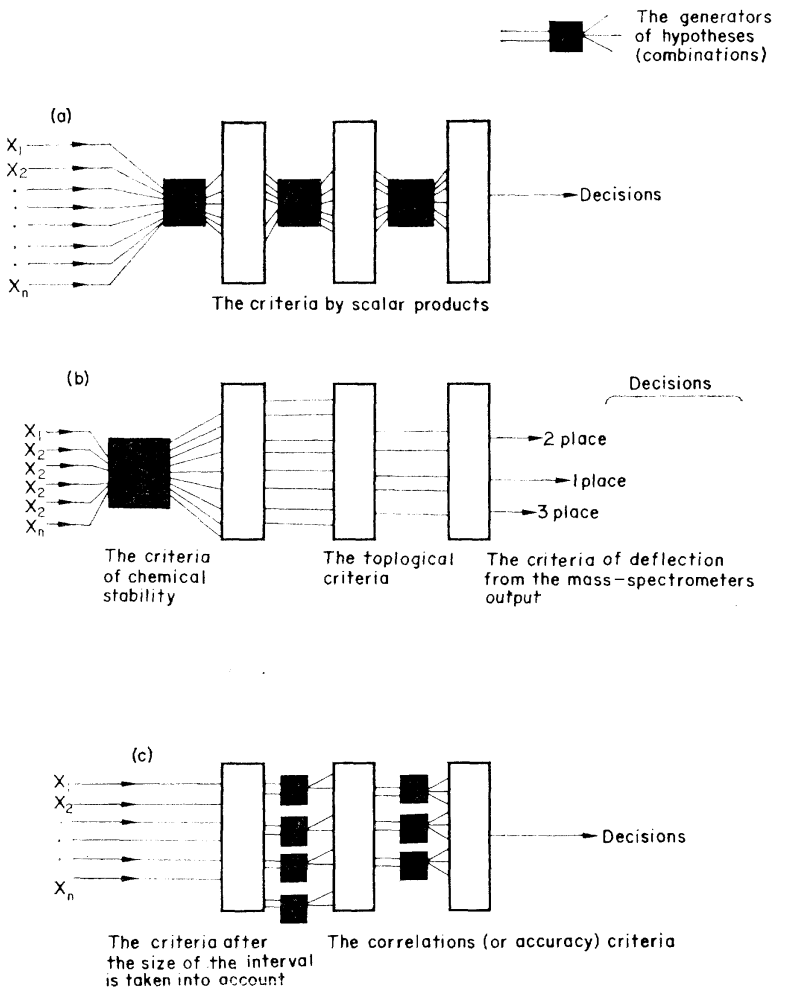

FIG. 1. Examples of system structures having heuristic self-organisation: (a) the Perceptron; (b) the Stanford University system and (c) the structure of GMDH

The second example shown in Fig. 1b is the structure of a system designed at the Stanford University. The problem is solved to predict the structure of organic molecules [8]. Only one

"generator of hypotheses" and three threshold for self-samplings are used here each having a different heuristic criterion.

The third example in Fig. 1c is the structure of algorithms of the Group Method of Data Handling (GMDH). Here the combination generators receive as their inputs only small groups of arguments. This algorithm will be thoroughly explained below.

Some other examples of the self-organizing systems are also known. For instance, the method of the *S*-matrix in theoretical physics and the algorithms of the so-called "evolutionary programming" may also be considered as examples of systems of heuristic self-organization [9, 10].

HEURISTICS IN THE GROUP METHOD OF DATA HANDLING (GMDH)

The GMDH is developed for solving various interpolation problems of engineering cybernetics. Here in place of selecting seeds for growing plants with desired characteristics, we deal with some functions of inputs and intermediate variables. The selection heuristics, formulated above, may be interpreted for this system as follows:

The first heuristic. "Elementary algorithms" are to be chosen. This term in GMDH case concerns laws of non-linear transformations of arguments and intermediate variables. For example, we shall use hereafter the covariations and the first, second, second, third, and fourth powers of input arguments.

The second heuristic. Heuristic criteria are to be chosen for threshold self-samplings. We use the mean square error criterion and the correlation criterion.

The third heuristic. Laws for constructing a complete description of the plant or processes are chosen according to several partial descriptions. In other words: it is necessary to choose the GMDH algorithm. Each GMDH algorithm has a complete description of the complex plant or process in some form which is replaced by several partial descriptions. The complete description takes into account all the arguments, the partial description only a group of them, perhaps only two for example.

Examples of GMDH Algorithms

The GMDH can be realized by many algorithms which differ with respect to construction of the complete description of a complex plant. About twenty algorithms have been proposed up to the present time. Let us consider, for brevity, only three which seem to be most significant. The three will be considered for the case where only four input arguments are avilable as discussed below.

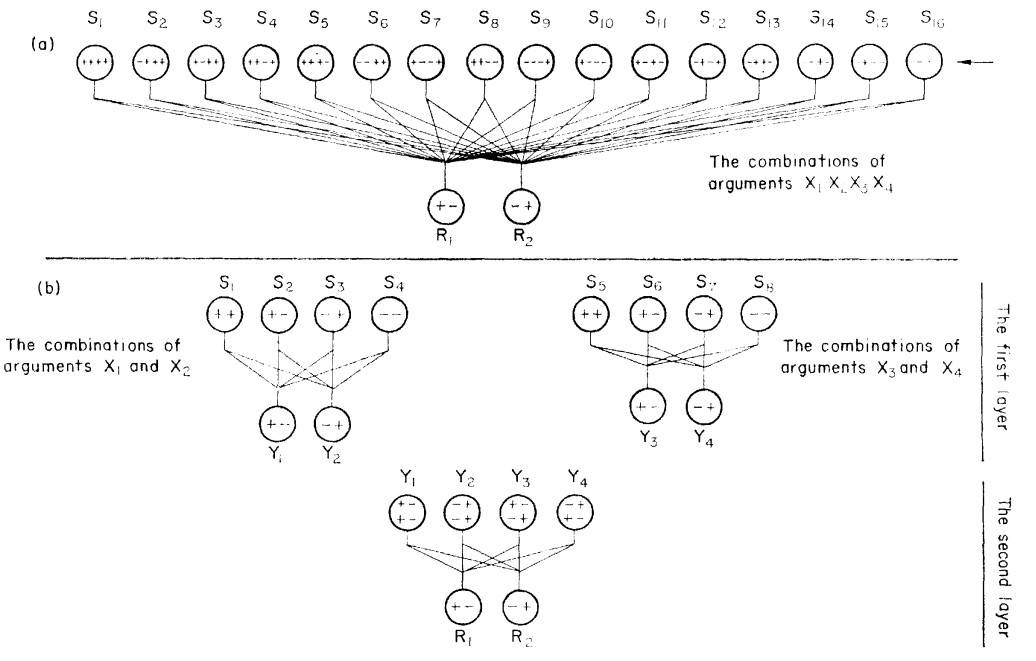

FIG. 2. The algorithm of the Group Method of Data Handling (GMDH) with probabilistic graphs: three partial graphs are used in (b) instead the one complete graph in (a).

1. *A GMDH algorithm using probabilistic graphs.* Four binary arguments x_1, x_2, x_3, x_4 can give us sixteen combinations which we call the input states. These states can be connected with two binary responses of the automata $Z = R_1$ and $Z = R_2$ by the use of the transmition probability graph shown in Fig. 2a. This graph is the complete description of some complex automaton.

According to the GMDH method, this complete graph is to be replaced by three partial graphs, each for two arguments only, and Fig. 2b shows an

example of graphs for the combination of arguments $x_1 - x_2$ and $x_3 - x_4$. Such graphs can be constructed for all combinations of arguments, particularly for $x_1 - x_4$ and $x_2 - x_3$ or $x_1 - x_3$ and $x_2 - x_4$ in this example. To learn the structure of graphs, the probabilities of connecting input states and responses are calculated. When calculating the probabilities, we assume $y_1 = y_2 = Z$, determine variables $y_1 y_2$ as functions of time, and use them in the last graph. Note that the calculation of probabilities for two arguments in the partial graph requires a much shorter learning sequence of data than the calculation of probabilities for four arguments which is needed for the complete graph. The algorithm has a multilayered structure. Therefore it is possible to insert a threshold self-sampling after each layer to select the useful information.

2. *A GMDHalgorithm using the Bayesformulas.* The Bayes formula in a complete form

$$Z = K\frac{P(x_1/R_i)}{P(x_1)} \cdot \frac{P(x_2/R_i)}{P(x_2)} \cdots \frac{P(x_1x_2/R_i)}{P(x_1x_2)} \cdot \frac{P(x_1x_3/R_i)}{P(x_1x_3)} \cdots \frac{P(x_1x_2x_3x_4/R_i)}{P(x_1x_2x_3x_4)}$$

can be replaced by three partial Bayes formulas, for example to combine the arguments $x_1 - x_2$ and $x_3 - x_4$:

$$y_1 = K_1\frac{P(x_1/R_i)}{P(x_1)} \cdot \frac{P(x_2/R_i)}{P(x_2)} \cdot \frac{P(x_1x_2/R_i)}{P(x_1x_2)},$$

$$y_2 = K_2\frac{P(x_3/R_i)}{P(x_3)} \cdot \frac{P(x_4/R_i)}{P(x_y)} \cdot \frac{P(x_3x_4/R_i)}{P(x_3x_4)}$$

$$Z = K_3\frac{P(y_1/R_i)}{P(y_1)} \cdot \frac{P(y_2/R_i)}{P(y_2)} \cdot \frac{P(y_1y_2/R_i)}{P(y_1y_2)},$$

where y_1 and y_2 are decisions made from the two first formulas. Such formulas may also be written for the other combinations of arguments: $x_1 - x_4$ and $x_2 - x_3$ or $x_1 - x_3$ and $x_2 - x_4$. When calculating the probabilities we assume $y_1 = y_2 = Z$, then we determine variables y_1 and y_2 as a function of time, and we use them in the third formula. Note that for the calculation of probabilities for two arguments, such as $P(x_1x_2/R_i)$, $P(x_3x_4/R_i)$ or $P(y_1y_2/R_i)$, much shorter sequences of learning data are necessary as compared to the calculation of probabilities for three of four arguments.

The algorithm is multilayered, and thus it is possible to use self-sampling thresholds.

3. *A GMDH algorithm using polynomials of second order.* The algorithm using polynomials of second order actually uses several short partial polynomials instead of one very long discrete Kolmogorov–Gabor polynomial which is usually used to approximate an unknown decision function. For example, with four arguments x_1, x_2, x_3, x_4, the complete polynomial including terms of all powers and all covariations of arguments has 70 terms.

The complete polynomial is:

$$\begin{aligned}
Z = {} & a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + a_5x_1^2 + a_6x_2^2 \\
& + a_7x_3^2 + a_8x_4^2 + a_9x_1x_2 + a_{10}x_1x_3 + a_{11}x_1x_4 \\
& + a_{12}x_2x_3 + a_{13}x_2x_4 + a_{14}x_3x_4 + a_{15}x_1^3 \\
& + a_{16}x_2^3 + a_{17}x_3^3 + a_{18}x_4^3 + a_{19}x_1^2x_2 + a_{20}x_1^2x_3 \\
& + a_{21}x_1^2x_4 + a_{22}x_2^2x_1 + a_{23}x_2^2x_3 + a_{24}x_2^2x_4 \\
& + a_{25}x_3^2x_1 + a_{26}x_3^2x_2 + a_{27}x_3^2x_4 + a_{28}x_4^2x_1 \\
& + a_{29}x_4^2x_2 + a_{30}x_4^2x_3 + a_{31}x_1x_2x_3 + a_{32}x_1x_2x_4 \\
& + a_{33}x_1x_3x_4 + a_{34}x_2x_3x_4 + a_{35}x_1^4 + a_{36}x_2^4 \\
& + a_{37}x_3^4 + a_{38}x_4^4 + a_{39}x_1^2x_2^2 + a_{40}x_1^2x_3^2 \\
& + a_{41}x_1^2x_4^2 + a_{42}x_2^2x_3^2 + a_{43}x_2^2x_4^2 + a_{44}x_3^2x_4^2 \\
& + a_{45}x_1^3x_2 + a_{46}x_1^3x_3 + a_{47}x_1^3x_4 + a_{48}x_1^2x_2x_3 \\
& + a_{49}x_1^2x_2x_4 + a_{50}x_1^2x_3x_4 + a_{51}x_2^3x_1 \\
& + a_{52}x_2^2x_3x_4 + a_{53}x_2^2x_1x_4 + a_{54}x_2^2x_3 + a_{55}x_2^3x_4 \\
& + a_{56}x_2^2x_3x_4 + a_{57}x_3^2x_1x_2 + a_{58}x_3^3x_1 + a_{59}x_3^3x_2 \\
& + a_{60}x_3^3x_4 + a_{61}x_3^2x_1x_4 + a_{62}x_2^2x_2x_4 \\
& + a_{63}x_4^2x_1x_2 + a_{64}x_4^2x_1x_3 + a_{65}x_4^3x_1 \\
& + a_{66}x_4^2x_2x_3 + a_{67}x_4^3x_2 + a_{68}x_4^3x_3 \\
& + a_{69}x_1x_2x_3x_4.
\end{aligned}$$

Learning consists in determining coefficients of this polynomial. To determine the coefficients by solving Gaussian normal equations, it would be necessary to invert matrices with dimension of 70 x 70 components and to use learning sequences having no less than 70 data points of interpolation. Such an extensive number of calculations generally exceeds the capabilities of the most modern computers. However, if there are more than four inputs, the solution becomes completely impossible, for example, when there are ten inputs, the polynomial contains about 200,000 terms. This is the source of Bellman's "curse of multidimensionality" which explains why no actual complex problem has yet been solved.

In the example with four inputs, or arguments, the GMDH uses three partial second order polynomials instead of one complete polynomial.

Partial polynomials for the combination of arguments $x_1 - x_2$ and $x_3 - x_4$ are:

$$y_1 = b_0 + b_1x_1 + b_2x_2 + b_3x_1^2 + b_4x_2^2 + b_5x_1x_2,$$

$$y_2 = c_0 + c_1 x_3 + c_2 x_4 + c_3 x_3^2 + c_4 x_4^2 + c_5 x_3 x_4,$$
$$Z = d_0 + d_1 y_1 + d_2 y_2 + d_3 y_1^2 + d_4 y_2^2 + d_5 y_1 y_2.$$

The other combinations are $x_1 - x_4$ and $x_2 — x_3$ or $x_1 — x_3$ and $x_2 — x_4$. We can choose any combination which gives us better accuracy. When calculating the coefficients, we assume $y_1 = y_2 - Z$ and then determine the variables y_1 and y_2 as functions of time, which are used in the third polynomial.

After substituting the first and the second polynomials into the third, we obtain a polynomial in which sixteen covariations of the 4th power would be omitted. It is known that the omission of any term of the complete polynomial or the superposition of any links upon its coefficients can decrease the accuracy of approximation although the decrease is small when these links are "optimal" or the polynomials are orthogonal. The basic result of our calculation is that the use of heuristic thresholds for the self-sampling of useful information provides an increase in accuracy which cannot be even compared with that which can be attained by perfecting a mathematical tool of approximation. Thus, the heuristics employed in the field of self-organization are more effective than the heuristics used to perfect a mathematical tool.

Let us consider the case when all the four arguments are binary thus taking on only two values: — 1 and + 1. Then the complete polynomial is:

$$\begin{aligned} Z = {} & a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_4 + a_5 x_1 x_2 \\ & + a_6 x_1 x_3 + a_7 x_1 x_4 + a_8 x_2 x_3 + a_9 x_2 x_4 + a_{10} x_3 x_4 \\ & + a_{11} x_1 x_2 x_3 + a_{12} x_1 x_2 x_4 + a_{13} x_1 x_3 x_4 \\ & + a_{14} x_2 x_3 x_4 + a_{15} x_1 x_2 x_3 x_4. \end{aligned}$$

The partial polynomials, for the combinations of x_1-x_2 and x_3-x_4, are:

$$y_1 = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2,$$
$$y_2 = c_0 + c_1 x_3 + c_2 x_4 + c_3 x_3 x_4,$$
$$Z = d_0 + d_1 y_1 + d_2 y_2 + d_3 y_1 y_2.$$

Using this combination of equations, the coefficients of the complete polynomial can obviously be constructed by the following formulas:

$$\begin{aligned} a_0 &= d_0 + d_1 b_0 + d_2 c_0 + d_3 b_0 c_0, & a_8 &= d_3 b_1 c_1, \\ a_1 &= d_1 b_1 + d_3 b_1 c_0, & a_9 &= d_3 b_2 c_1, \\ a_2 &= d_1 b_2 + d_3 b_1 c_0, & a_{10} &= d_3 b_2 c_2, \\ a_3 &= d_1 b_2 + d_3 b_0 c_1, & a_{11} &= d_3 b_3 c_1, \\ a_4 &= d_2 c_2 + d_3 b_0 c_2, & a_{12} &= d_3 b_3 c_4, \\ a_5 &= d_1 b_3 + d_3 b_3 c_0, & a_{13} &= d_3 b_3 c_1, \\ a_6 &= d_2 c_3 + d_3 b_0 c_3, & a_{14} &= d_3 b_2 c_3, \\ a_7 &= d_3 b_1 c_1, & a_{15} &= d_3 b_2 c_3. \end{aligned}$$

It is easy to find similar formulas from the other two combinations of arguments if they prove to be more accurate than those shown above. However, note that not a single term of the complete polynomial is lost. However, this does not mean that there are no additional limitations on the choice of coefficients. Using the complete polynomial does give more freedom when we attempt to minimize the mean square error.

Coefficients of partial polynomials may also be found by solving the Gaussian normal equations. The minimum number of interpolation points is equal to number of unknown coefficients whereas, in the last example, only four points instead of 16 were needed.

CRITERION OF OPTIMALITY FOR THE GMDH ALGORITHM WITH POLYNOMIALS OF SECOND DEGREE

The GMDH algorithm with polynomials of second order guarantees a choice of coefficients of the partial polynomials whereby the minimum mean square error may be obtained. Then the coefficients of the complete polynomial may be calculated from the formulas determined by the chosen structure of the algorithm.

To raise the accuracy and to get the well-conditioned matrices, all possible combinations of arguments are tried. Only combinations producing the smallest error are allowed to pass through a threshold to the next layer. For accuracy, the number of equations averaged, according to the Gaussian rule, is increased as much as possible for stationary processes. If $N \geqslant n$, the complete polynomial and partial polynomials give the same value of mean square error.

DIFFERENCES BETWEEN THE PERCEPTRON AND GMDH ALGORITHMS WITH POLYNOMIALS OF SECOND DEGREE

The perceptron has a multilayer structure. Instead of accepting final decisions in the first layer of data processing, as is recommended by the modern theory of statistical decision, the signals pass through several layers each of which consists of links with variable gains, summators, and threshold units. The above integral influences, acting "without human influence", are realized only by means of threshold units. The value of each threshold is high enough to permit the sampling of about only 40 per cent of the most probable decisions beyond each layer; the rest of the signals are not allowed to pass. This is just "the principle of nonfinal decisions", which is realized by the perceptron contrary to conclusions reached through statistical decision theory. The idea of nonfinal decisions enables different heuristic criteria to act on the information flow several times, and this results in the exceptionally high accuracy of systems using heuristic self-organization.

Our modifications of the perceptron are as follows:

(1) Coefficients of the perceptron links are calculated by solving Gaussian normal equations, formulated for a small group of input signals, instead of a random search or adaptation.

(2) Integral influences are realized by different heuristic criteria and by the use of threshold units, more often, by the correlation of signals and teaching data, contrary to a scalar multiplication in the Rosenblatt's perceptron [7]. Instead of the linear perceptron decision function

$$\sum = (\bar{a}\bar{x}) = a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_nx_n$$

a more developed non linear polynomial is used:

$$\sum = a_0 + \sum_0^N a_nx_n + \sum\sum a_{n_1}a_{n_2}x_{n_1}x_{n_2} + \sum\sum\sum a_{n_3}a_{n_4}a_{n_5}x_{n_3}x_{n_4}x_{n_5} + \ldots .$$

This polynomial is often called the Kolmogorov-Gabor polynomial [11]. It is stated in N. J. NILSSON'S book [12] that non-linear decision functions were proposed by I. Koford. In fact they were introduced by D. Gabor around 1960. Note that for Gaussian random processes, the optimum filter is linear and the perceptron decision function is the best function.

(3) The continuous optimization of threshold values is used to get the highest accuracy.

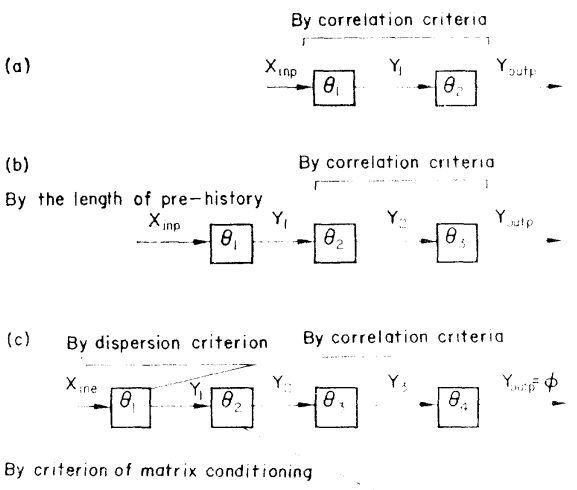

FIG. 3. Structures of algorithms for self-sampling of useful information:
(a) in pattern recognition;
(b) in forecasting prediction;
(c) in identification.

The distinctions between the perceptron and the GMDH are not great nor fundamental; therefore we often call our systems perceptrons or "systems of the perceptron type".

Other features of our systems are the same as those of Rosenblatt's perceptron. The simplest perceptron is used for pattern recognition. It involves two thresholds only, both determined by correlation coefficients (Fig. 3a). The perceptron for a random process prediction is more complicated. Here the self-sampling of the length of a current interval of prehistory is added (Fig. 3b). And finally, two preliminary thresholds are used in identification problems: First, for choosing the most active variables, and second, for choosing data which do not repeat the previous information. Data which repeat are omitted, and then the two main correlation thresholds are used.

FOUR REASONS FOR USING MULTILAYERED ALGORITHMS OF GMDH

There are at least four reasons why the perceptron-like multilayered structures of the GMDH algorithms are much better than usual, singlelayered structures:

(1) Only short learning sequences are ever available when we attempt to predict a process or try to find the mathematical model of a complex plant. Thus we must use *one and the same points of interpolation several times,* the number of points is less than the number of members of the complete polynomial. Only two methods are known which will work under such conditions: Methods of stochastic approximation and the GMDH. That is why we have called them rival methods in Ref. [2]. But stochastic approximations cannot solve the problem of identifying the global maximum of a multiextremum hill and they do not permit us to organize self-sampling thresholds to omit "harmful information". Therefore, the GMDH is the superior method.

(2) The interval of the data observation is always limited. Therefore the input data, "features", can not be useful or only neutral, but even harmful too. This statement contradicts Shannon's information theory, but it is true. We can give the following definition of the harmful feature. A given feature is harmful if its average value and other statistical characteristics in the learning sequence differ from those in the testing sequence. Such features are poorly correlated with the output and they must be eliminated in order to increase the accuracy. The thresholds provide self-sampling of the useful information, but they do not allow "harmful" information to pass.

(3) Even if we could obtain very long learning sequences, we would not be able to find computers large enough to solve normal equations based on complete polynomials.

(4) The coefficient matrix of the equations for a complete polynomial is always ill-conditioned. However, among many combinations of small

Deviations $\Delta y = y - y_{med}$ are shown in Fig. 4. They are included in the input data, or features, for our perceptron.

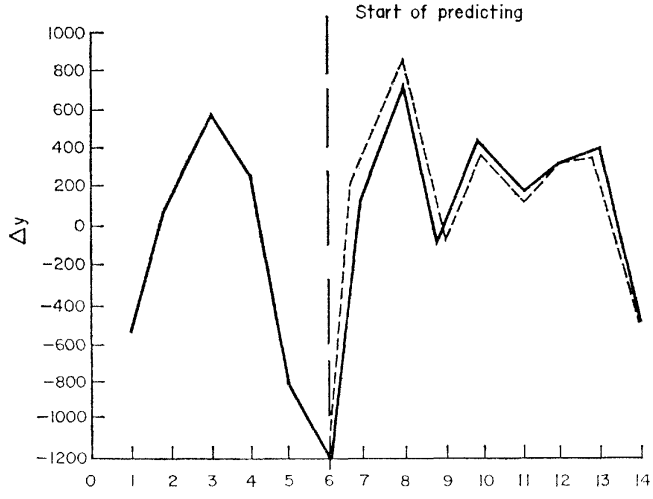

FIG. 4. Time variation of Δy: continuous line—the real variation of y: dotted line—the predicted variation.

partial equations, we can always find well-conditioned matrices of small dimensionality.

Let us draw attention to the main fact that the GMDH solves not only the problem of dimensionality, but it also permits the use of very short learning sequences consisting of only six interpolation points as a minimum, and, with linear operators, only three [6].

Many examples of solving various interpolation problems encountered in engineering cybernetics by the GMDH have already been published in the Ukrainian journal "Avtomatika" [2-6 and on]. This journal is now translated into English as the "Soviet Automatic Control" journal by the Institute of Electrical and Electronic Engineers, Inc., 345 East 47 Street, New York, N.Y. 10017. Therefore, we shall consider below only the main results of two examples to show the application of one GMDH algorithm, particularly the algorithm with second order polynomials, to the solution of two rather different interpolation problems.

Example of random process predicting [2]

In Table 1, data about the size of areas used to grow wheat and other produce for a period of 14 years in one district of the Ukraine are given. Here y is the area used to grow wheat, $hzwv$ are the areas used to grow other produce. The problem is to predict the area which will be used to grow wheat $y(t)$ for at least one year in advance.

The first heuristic, the choice of an "elementary algorithm" of input features. The preliminary data processing consists in calculating deviations of the variable $y(t)$ from the non-linear trend described by the 3rd order equation:

$$y_{med} = a_0 + a_1 t + a_2 t^2 + a_3 t^3.$$

The second heuristic, the choice of criteria for threshold self-samplings. Three thresholds for self-sampling of useful information were used. The first one by the length of the prehistory being considered, the second and the third by the correlation coefficients of the intermediate and predicted variables.

The third heuristic, the choice of the GMDH algorithm. An algorithm using polynomials of second order was chosen. The method of constructing the complete description by a series of partial descriptions was conditioned by this choice.

The results of calculations for one definite value of three thresholds shown in Fig. 5 are as follows: when the prehistory length being considered equals 5 years ($\theta_1 = 5$), 35 input quantities pass through the first threshold. (It is easy to calculate that the six variables being taken into account plus one variable of deviation shown in Fig. 4 give us 35 ordinates for 5 years.)

Each of the 35 features is considered to be random function, and therefore we can calculate their correlation with the deviation $y(t)$ from the average trend. The threshold value of the first correlation was taken as $\theta_2 = 0{\cdot}443$. The second self-sampling passed the following features:

1. Sowing area of wheat 5 years ago $x_1 = y_{K-5}$
2. Sowing area of wheat 4 years ago $x_2 = x_{K-4}$
3. Sowing area of produce v 5 years ago $x_3 = v_{K-5}$
4. Total area 5 years ago $x_4 = \Sigma_{K-5}$
5. Sowing area of produce w 2 years ago $x_5 = w_{K-2}$
6. Total area 4 years ago $x_6 = \Sigma_{K-4}$

TABLE 1. INPUT DATA ABOUT SOWING AREAS (LEARNING SEQUENCE)

Produce		Sowing year after year (in hectares)													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
White	y	2500	5500	7700	8334	7800	7400	8647	8795	7400	6200	6060	6370	6380	5700
Other produce	n	80	140	280	500	630	1140	1880	2430	3300	3040	2990	3500	3800	3500
	z	380	600	1180	1100	1020	920	860	1150	1520	1800	1840	1970	2530	2980
	w	630	2740	4530	3400	1390	1280	750	370	380	450	660	1170	1690	1900
	v	160	540	980	800	630	780	900	670	740	1090	1050	1170	1430	1370
Total area	Σ	3750	9250	14,670	14,134	11,470	11,520	13,037	13,415	13,340	12,580	12,600	14,180	15,830	15,450
Deviation from trend	Δy	-4714	214·7	795·2	408·8	−641·8	−1150·0	302·1	373·2	24·1	−602·6	−237·1	415·3	509·5	−740·0

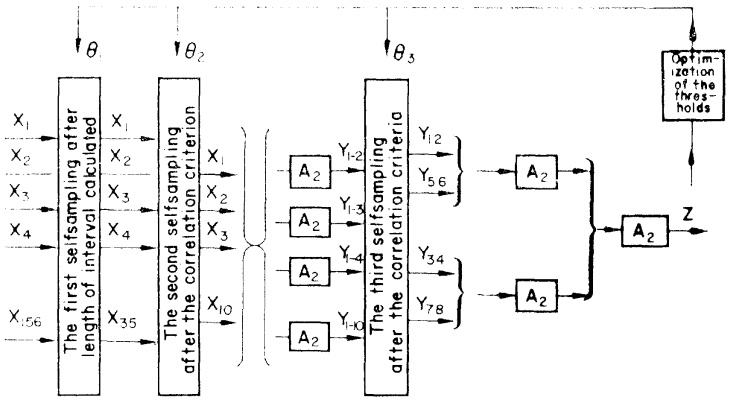

FIG. 5. The algorithm of the prediction, the GMDH with polinomials of second degree.

7. Sowing area of produce w 3 years ago $x_7 = w_{K-3}$
8. Total area 2 years ago $x_8 = \Sigma_{K-2}$
9. Deviation from the trend of sowing area occupied by wheat 2 years ago $x_9 = \Delta y_{K-2}$
10. Sowing area of produce w 5 years ago $x_{10} = w_{K-2}$

Note that *features chosen by the threshold are quite unexpected, and they cannot be found by any deductive reasoning. This is why the given method is said to be one of self-organization.* Data on the fifth, eighth and ninth variables is obtained for the most recent past, 2 years ago. Therefore the optimum prediction should be for 2 years in the future.

Ten variables chosen by the correlation threshold enable us to construct 45 Kolmogorov-Gabor polynomials of second order each having 2 arguments. Each polynomial can be written thirteen times, according to the length of the learning sequence, by substituting the input data. After the Gaussian averaging, we get 45 systems of normal equations, each having a small matrix of 6 x 6 elements. The solution of normal equations determines 45 intermediate variables.

Then the correlation coefficients between the intermediate variables and centred deviations of the predicted variable are calculated. The threshold value for the second correlation was $\theta_3 = 0{\cdot}9$, and this allowed only 4 variables to pass the third self-sampling threshold namely: y_{34}, y_{78}, y_{12}, y_{56}. Four variables make it possible to find two variables at the next level of complexity, and after combining result in the output Kolmogorov-Gabor polynomial:

$$y = a_0 + a_1 y_{1256} + a_2 y_{3478} + a_3 y_{1256}^2 + a_4 y_{3478}^2 + a_5 y_{1256} \cdot y_{3478}.$$

Having written this polynomial thirteen times, inserting the data and averaging by the Gaussian rule, we obtain the last system of normal equations, with a 6 x 6 matrix. Its solution gives us the prediction formula:

$$y(t) = 120 - 78{\cdot}7 + 3529{\cdot}9t - 442t^2 + 15{\cdot}9t^2 + 1{\cdot}352y_{1256} - 0{\cdot}23y_{3478} - 0{\cdot}0053y_{1256}^2 - 0{\cdot}0065y_{3478}^2 + 0{\cdot}0012y_{1256}y_{3478}.$$

Using this formula we can predict the sowing area occupied by wheat for the fourteenth year which is used for the testing. Predicting for each successive year the formula is redeveloped from the very beginning to evolve the formula coefficients. The predictions for the sixth to the fourteenth year are shown in Fig. 4 by a dashed line. The accuracy of prediction proved to be unusually high since the mean-root-square error was $\delta = 0{\cdot}0009$.

Optimization of thresholds indicated in Fig. 6. The above algorithm realizes a feedforward heuristic self-organization method according to the principle "by inputs". To realize the feedback principle "by outputs" a procedure of threshold optimization should be used. The portion of the signals passed by each threshold is to be chosen so that the accuracy of the results, for a sequence, be maximal. This optimal portion is equal to about 40 per cent in the first layer and decreases very rapidly in the next layers of the perceptron. Papers have been written in which the solution of this problem is obtained using a probabilistic approach [7 and 13]. However, we prefer to solve it using the data of a given testing sequence by the simple calculation of several variants of threshold values.

The accuracy decreases if thresholds are too low or too high. So the problem is to find the single extremum value of accuracy in the space of the thresholds θ_1, θ_2, θ_3, using, for example, the Fibonacci method. The variation of thresholds mentioned above increases the accuracy even more.

Example of identification of static characteristic of multiextremum plant [4 and 6]

The value of extremum index ϕ, the manipulating variable μ, and the main disturbance λ measured

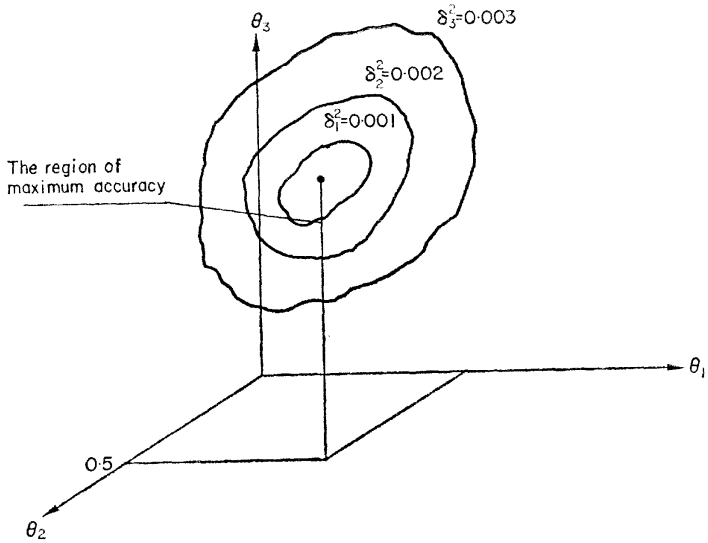

Fig. 6. The optimisation of thresholds.

for 6 last instants, are input information in the second example. If memory devices permit the storage of data for more than 6 instants, the accuracy will be multiplied due to reduction in measurement noise.

The first heuristic is the construction of an "elementary algorithm", i.e. simple non-linear functions of inputs. When identifying the static characteristic we have used the first, second and third powers of the inputs

$$\mu, \quad \lambda, \quad \mu^2, \quad \lambda^2, \quad \mu^3, \quad \lambda^3.$$

When identifying the dynamic characteristic we have used the integrals of these functions [6].

The second heuristic is connected with the choice of criteria for the threshold self-samplings of useful information. As previously stated, we used correlation coefficients between every intermediate variable and the extremum index.

organize the process of self-sampling, we again used the four-layer perceptron shown in Fig. 7. The first layer gives us those polynomials of second order which are best suited to approximate the complex surface of the multiextremum hill. Only about 40 per cent of total number of polynomials are used to construct intermediate variables of the second layer. In the second layer, new polynomials of second order are selected again, but here they are constructed from intermediate variables of the first layer, and because of that they are of fourth order with respect to the input arguments. Those polynomials ensuring the best approximation of the extremum hill surface pass the new threshold and new polynomials are constructed again and so on until a degeneracy begins, i.e. until the accuracy of approximation starts to diminish. Finally, only the single best decision is chosen in the last layer. As a result the surface of the

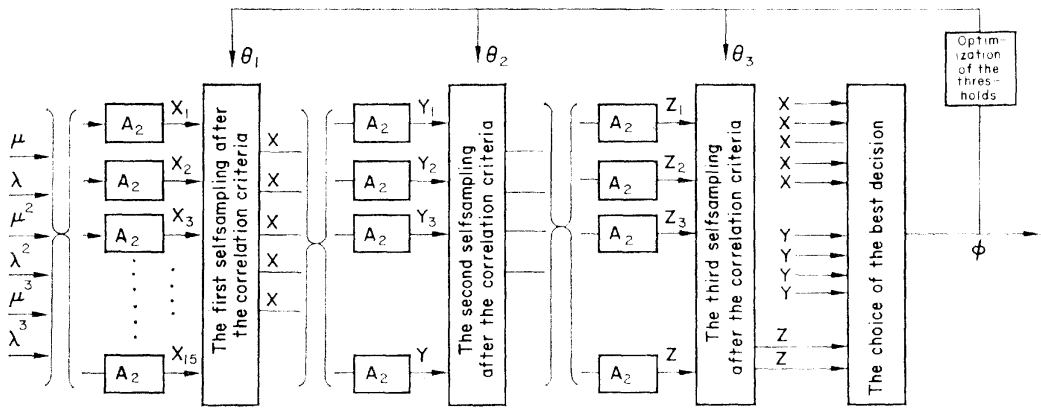

FIG. 7. The algorithm of the GMDH with polynomials of second degree in the problem of identification of a two modal extremum plant.

The third heuristic is concerned with the choice of the GMDH algorithm. The algorithm using second order polynomials was chosen again. To

extremum hill will be described by several, optimal, polynomials of second order, chosen in all layers. These comparatively simple polynomials

represent the statical characteristics of the extremum plant, replacing a very long Kolmogorov-Gabor complete polynomial.

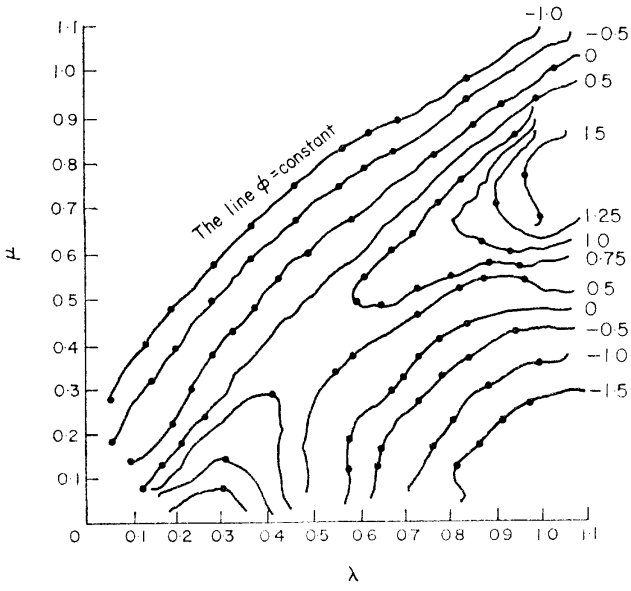

FIG. 8. The static characteristics of a two modal extremum plant.

Let us point out some results: 60 points of the multiextremum hill, shown in Fig. 8, were used as a learning sequence of data and only 10 as a testing or examining sequence. The first threshold was passed by 11 of 15 variables, the second, 10, and the third, 2. The last threshold passed only one polynomial of the "fourth generation". The accuracy can be evaluated by the correlation coefficient $K_{Z\phi}=0{\cdot}9815$. It is very high.

Optimization of the thresholds. This accuracy was reached by the optimum values of all the thresholds which were found by the Fibonnacci search for the maximum accuracy.

CONCLUSION

The examples of the GMDH application to the solution of different interpolation problems show the high accuracy of this method. In some cases, e.g. in the case of random process prediction, the accuracy is quite fantastic. The unusual prediction accuracy of a process which seems to be quite unpredictable makes us change our estimate of the role of randomness in our environment. It seems now that perhaps Laplace was almost correct. The whole world around us is perhaps more deterministic than we usually think. The randomness exists but it shows up at the 4th or 5th decimal place.

This high accuracy can be explained in a very simple way: Everybody who has used prediction theory knows that the accuracy is higher when the process itself is well correlated, i.e. when the autocorrelation of the process is high. In the GMDH, the thresholds select only useful variables, i.e. those which are well correlated with the output. This is the first reason why the accuracy is so high.

The second reason is that the GMDH, having multilayered algorithms, enables us, despite of brevity of data readings, to take into account the high order covariations in the Kolmogorov-Gabor polynomials, or dependent inputs in the Bayes formula. Present-day statistical decision theory has a single-layered structure of algorithms, and therefore it requires learning data sequence which are too long to be obtained and used in practice.

We can recommend the following method for verifying the accuracy: All data except those for the next predicted moment are always used as the learning sequence. The testing sequence consists of only one future point. So we are to repeat all calculations from the beginning before each next prediction. We call this method "the method predicting formula evolution" because it continuously changes.

Note that it would not be correct to verify the accuracy by means of the learning sequence itself because in this case we cannot reach "degeneracy" of formulas: more algorithm layers may be taken —more the accuracy will increase. We must use a separate testing sequence of data and only then it is possible to find an optimum number of layers. The accuracy first increases with each subsequent layer but then, after exceeding the optimal number of layers, the accuracy begins to decrease. There are an optimal number of generations just as in the process of plant or animal selection.

This is correct not only for prediction but also for pattern recognition and for identification problems. Let me point out finally that only the GMDH enables us to solve the problem of identifying a multiextremum plant directly because this method is specially developed for solving high dimensional problems when the data sequences are very short.

REFERENCES

[1] S. BEER: *Cybernetics and Management.* English University Press, London (1963).

[2] A. G. IVAKHNENKO: The group method of data handling—A rival of the method of stochastic approximation, *Avtomatika,* No. 3 (1968).

[3] A. G. IVAKHNENKO, V. B. KONOVALENKO, YU. M. TULUPCHUK and I. K. TIMCHENKO: The group method of data handling in the problem of pattern recognition and decisions making. *Avtomatika,* No. 5 (1968).

[4] A. G. IVAKHNENKO, Yu. S. KOPPA and N. A. IVAKHNENKO: The group method of data handling in the problem of identification of the multiextremum plant. *Avtomatika,* No. 2 (1969).

[51 A. G. IVAKHNENKO, V. D. DIMITROV and S. G. MGELADSE: The probability algorithms of the group method of data handling in the problem of the prediction of random events. *Avtomatika,* No. 3 (1969).

[6] A. G. IVAKHNENKO, YU. S. KOPPA: The regularisation of the discriminat functions in the GMDH. *Avtomatika*, No. 2, (1970).
[7] F. ROSENBLATT: *Principles of Neurodynamics*. Spartan Books, Washington (1962).
[8] D. PATERSON: Computers that hypothesize. *New Scient.* **39,** No. 612, 29 Aug. (1968).
[9] W. HEISENBERG: *Introduction to the Unified Field Theory of Elementary Particles*. Interscience, New York (1966).
[10] L. FOGEL, A. OUENS and M. UOMY: *Artificial Intelligence and Evolutionary Programming*, London (1966).
[11] D. GABOR, W. WILDY and R. WOODCOCK: Universal non-linear filter predictor which optimises itself by learning. *IEE Proc.* V. 108, Part B (1961).
[12] N. J. NILSSON: *Learning Machines*. McGraw-Hill, New York (1967).
[13] V. S. AMIRBEKYAN and S. V. DAYAN: Some questions of choosing optimal structure of A-unit of perceptron. *Voprosy Radioelectroniky*, Seria EVT, No. 7 (1967).

Résumé—Les systèmes ou programmes d'auto-organisation heuristique sont définis comme ceux contenant les générateurs d'hypothèses ou de combinaisons aléatoires et plusieurs couches d'auto-échantillonage à seuil de l'information utile. La complexité des combinaisons augmente de seuil en seuil. Un système connu, le perceptron de Rosenblatt, peut être pris comme exemple.

La méthode des Groupes du Traitement de l'Information (MGTI) basée sur les principes de l'auto-organisation heuristique est réalisée pour la solution des problèmes complexes à grande échelle lorsque la sequence d'information est très courte. L'article donne deux exemples pour illustrer la manière dont cette méthode s'applique aux problèmes de prédiction de procédés aléatoires et d'identification des caractéristiques d'un système réglé à extrema multiples.

Zusammenfassung—Die Systeme oder Programme heuristischer Selbstorganisation sind in der Weise definiert, daß sie die Generatoren von stochastischen Annahmen oder Kombinationen und verschiedene Schichten von Schwellenselbstabtastung nützlicher Information einschließen. Die Komplexität der Kombinationen wächst von Schicht zu Schicht. Ein bekanntes System, das Perzeptron von Rosenblatt, mag als Beispiel dienen.

Die Gruppenmethode der Datenverarbeitung auf der Grundlage der Prinzipien heuristischer Selbstorganisation wird entwickelt, um komplexe Probleme hoher Dimension zu lösen, wenn die Datenfolge sehr kurz ist. Zwei Beispiele veranschaulichen, wie diese Methode auf Probleme der Vorhersage zufälliger Prozesse und der Identifizierung der Charakteristiken einer Anlage mit vielen Extrema anzuwenden ist.

Резюме—Системы или программы эвристической самоорганизации определены как имеющие генераторы случайных гипотез или комбинаций и несколько слоев самоотборов по критерию полезности информации. Сложность комбинаций растет от уровня к уровню. Примером может быть известная всем системаперцептрон ф. Розенблатта.

Метод Группового Учета Аргументов (МГУА), основанный на принципах эвристической самоорганизации, служит для решения многомерных задач, когда интервал наблюдения очень короток. В статье даны два примера, иллюстрирующие, каким образом этот метод применеятся к проблемам предсказания случайных процессов и идентификации характеристик управляемого об'' екта со многими экстремумами.