CAT. 81
79N28048

NASA TECHNICAL MEMORANDUM NASA TM-76008

# DECISION MAKING AND PROBLEM SOLVING WITH COMPUTER ASSISTANCE

F. Kraiss

(NASA-TM-76008) DECISION MAKING AND PROBLEM SOLVING WITH COMPUTER ASSISTANCE (National Aeronautics and Space Administration) 61 p HC A04/MF A01 CSCL 05A

N80-15973

Unclas
G3/81 46599

Translation of "Entscheiden und Problemloesen mit Rechnerunterstuetzung," Forschungsinstitut fuer Anthropotechnik, Meckenheim, West Germany, Report No. FB-36, Feb. 1978, pp 1-73

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546 JANUARY 1980

# Table of Contents

# DECISION MAKING AND PROBLEM SOLVING WITH COMPUTER ASSISTANCE

F. Kraiss
Research Institute for Anthropotechnology
Meckenheim, West Germany

## Preface /2*

In this report the results achieved in a research project of the BMVg on the subject of "Man-Computer Interaction in Guidance and Control Systems" (T/RF 36 / 60024 / 61606) are summarized. In the period from February to November 1977 the author worked at the NASA Ames Research Center, Moffett Field, California in the Man-Machine Integration Research Division to conduct work for this project.

Numerous contributions from the research projects being conducted there and from many discussions with American colleagues find their echo in this report. Various important bibliographical sources first became accessible there, as did participation in internal courses and activities.

I extend a hearty thanks to the NASA Ames Research Center for these services and for the excellent working conditions.

Mr. Klaus-Peter Holzhausen and Mr. Willi Stein, both engineers, have taken the time and effort to review this report critically. I also wish to extend my thanks to them.

## 1. Introduction /3

Modern guidance and control systems serve for the control, coordination and communication in complex technical systems. The tasks delegated to the operator in his changing roles as manager, supervisor, problem solver, trouble shooter, etc. essentially require the following mental capabilities:

- observation (selection of data relevant to the task)
- correction
- decision making
- problem solving (finding the procedure for the solution and effectively representing the problem)
- association (analogous conclusions)
- deduction (understanding relationships, solving multiple meanings via comparisons with stored knowledge and experience).

This mental work must often be conducted under extreme work loads and pressure of time. As a rule large amounts of data have to be recorded and processed. Portions of these data contain mistakes. Changed aims and boundary conditions must continually be

* Numbers in the margin indicate pagination in the foreign text.

taken into consideration.

From this description of tasks it can be seen that the mental stress on the operator is very large in normal situations. On the other hand the work is facilitated by the fact that the working method of operator is already programmed to a great extent, i.e., the same work sequences and problem types occur and reoccur.

The process of decision making and problem solving in a situation typical in guidance and control systems may be divided into the following areas [1]:

- Does a decision have to be made or is a problem to be solved?
  - Observation of the process and surrounding situation
  - Selection of data relevant to the decision
  - Interpretation and reduction of data
  - Does the system condition observed deviate from expectations provided by internal models (experience, training)?
- Is this a new problem? /4
  - Is a routine solution possible?
  - May the problem be divided into partial problems with known solutions?
  - Has this situation already occurred before? What decision was made then and with what degree of success?
- Which paths of solution or alternatives for dealing with the problem are there?
  - Selection of a strategy (conservative, risky)
  - Is additional information necessary? What would this cost?
- Test and evaluation of possible consequences
  - How useful will the solution be?
  - Is the occurrence of the expected consequence certain or only probable?
  - How great is the risk of an undesired consequence?
  - How could an unfavorable consequence be met? (Calculating steps in advance)
- Carrying out the action with the greatest usefulness expected, that is with the least disadvantage.
- Observation of actual effects.

Experience teaches that a man cannot optimally conduct this complex sequence of decision making. He regularly deviates from behavior predicted on the basis of normative models, reasons being erroneous information processing and storing in the brain, lessening concentration, prejudice and inconsistant behavior. In addition a certain degree of uncertainty exists for the operator with respect to available alternatives, possible consequences, surrounding and boundary conditions and also individual preferences [2].

The present report studies the possibilities for actively supporting the operator in carrying out his tasks with guidance and control systems.

The guiding thought in this project is to reduce human error by means of computer support, to employ a computer for supporting operator intelligence. /5

A certain degree of intelligence (Artificial Intelligence, A.I.) is attributed to computer programs more or less capable of reproducing the above-mentioned mental performance of the operator [3]. Considerable progress has been made in this area in the past few years, especially with respect to pattern recognition and classification. This has led to the application of fully automated robots in manufacturing processes, which previously only men could carry out.

The transition to full automation, however. is substantially more problematic in guidance and control systems than in industrial procedures, one reason being that the available machines are not yet intelligent enough for many tasks. In addition it must be taken into consideration that the operator has to have the last word in many situations. It is, for example, difficult to imagine a fully automated air traffic guidance system without human control [4].

Therefore the replacement of man by intelligent machines is not to be discussed here, but on the contrary a symbiotic relationship between man and computer, in which the computer assumes the role of a partner, an intelligent associate. In this man-computer team each partner continuously assumes that portion of the total task fitting his powers and capabilities.

Notice that this approach substantially differs from a partial automation in which only routine work is delegated to the computer. In automated systems the operator only takes part sporadically, but does not receive any help from the computer in decision making in the usually critical situations.

It will be further demonstrated that intelligent support of the operator in interaction with him is possible on various levels of the decision making process. The applied artificial intelligence algorithms comprehend pattern recognition and classification, adaptation and machine learning as well as dynamic and heuristic programming. Elementary examples are presented to explain basic principles of individual procedures. /6

Consequences for design of future man-machine-systems, resulting from a cooperation between human and artificial intelligence, are discussed.

## 2. Cooperation with Computers capable of Learning and Adapting

A graphic representation of possibilities for the cooperation between humans and intelligent machines is presented in Figure 1.

Aside from the known man-machine controlled circuit a computer is presented here arranged parallelly to the operator.

Since the computer is first untrained, the machine must receive information about the intelligent behavior expected. This can be carried out explicitly by programming normative models for desired operator behavior or by presenting aims or strategies. An entire series of mathematical models is available for this purpose, e.g. for reading instruments, discovering signals, processing information and making decisions (theory of usefulness, analysis of decisions). Rouse gives a survey of the development of such models [5].

Another implicit possibility for instructing the computer consists in constructing an internal computer model capable of adapting by means of continuous observation of operator behavior. This behavior is expressed in the actions taken for an observed situation. In time a descriptive model of ever increasing reliability is created, ideally mirroring operator preferences adequately. Such models may therefore be employed for simulating humans and predicting what they intend to do. Building such a model is only possible if the operator behaves consistently for an extended period of time. This means that he makes comparable decisions in identical situations and retains his strategy.

Once the computer is trained for a defined manner of behavior, /7
cooperation between human and computer may be carried out on various levels: the computer is relegated to an advisory position, i.e. without independent responsibility it merely automatically indicates suggestions for dealing with problems. This may be applied, e.g. to the training of an operator along the guidelines of a normative model. Of course it is left to the discretion of the operator to call for further information from the computer. Among other things the following would be interesting for a pilot, "Which measures are to be carried out, if an engines fails in the next four minutes."

On the other hand the computer may also direct questions to the operator if it determines that additional information is lacking for the calculation of internal algorithms (dialogue initiated by computer).

The computer advances from an advisory function to a line function with independent responsibility when its suggestions are not only indicated to the operator but are directly carried out. The important question here is the division of tasks between man and machine and the interactive control. The various logical levels are expressed in the square "Task delegation" (Fig. 1):

- Where required manual delegation of partial tasks to the computer for autonomous processing.
- Computer independently assumes task portions when it determines that its internal model functions with a sufficient degree of reliability.
- Computer waits and only goes into action when the operator shows no reaction.

- Computer blocks doubtful operator decisions and gives the operator an alarm signal in cases of conflict.

The following sections of this report describe individual algorithms and procedures applied to instruct the computer in desired intelligent behavior and to organize cooperation between man and machine.

Possibilities for interaction in processing are studied for the following tasks: /9

- preparation and evaluation of information (Chapter 3)
- decision making (Chapter 4)
- problem solving (Chapter 5)

## 3. Computer Support in Preparation and Evaluation of Information

The derivation of probabilities for certain hypotheses under consideration of additional new information is a task frequently occurring in the process of decision making.

The influence of new data on the probability of an hypothesis is determined mathematically according to the rule of Bayes. This provides a normative model, a performance measurement for how well or poorly a man may estimate the consequence and significance of new information.

### 3.1 The Rule of Bayes [6]

It is assumed that there are two hypotheses $H_1$ and $H_2$, each excluding the other and also complementary. Then the valid formula for the a priori probability is:

$$p(H_1) + p(H_2) = 1$$

The relationship between the beginning probability for a hypothesis and the subsequent probability dependent on an observation D is given by the rule of Bayes:

$$p(H_1 \mid D) = \frac{p(D \mid H_1)p(H_1)}{p(D)}, \qquad p(H_2 \mid D) = \frac{p(D \mid H_2)p(H_2)}{p(D)}.$$

The sum of these probabilities may also be set equal to 1: p(D) may be substituted in the above equation:

$$p(H_1 \mid D) = \frac{p(D \mid H_1)p(H_1)}{p(D \mid H_1)p(H_1) + p(D \mid H_2)p(H_2)}.$$

This equation describes the influence of additional information D on the probability of an hypothesis. The subsequent dependent probability of an hypothesis can therefore be calculated if the /10 probability for the occurrence of a defined bit of information may be estimated under the assumption that the hypothesis is correct ($p(D|H)$).

A generalisation of the above described relationship for more than two (i) hypotheses and n repeated applications of the Bayes' Rule (several observations) results in the equation for independent observations and exclusive and all-encompassing hypotheses [6]:

$$p(H_i \mid D_1, D_2, \ldots, D_n) = \frac{p(D_n \mid H_i)\,p_{n-1}(H_i)}{\sum_j p(D_n \mid H_j)\,p_{n-1}(H_j)}$$

## 3.2 An Example for the Revision of Probabilities according to Bayes

A whale and a submarine have been sighted in an area of the ocean. The location of the submarine is to be determined. Two sensors are available for the search:

| Type of Sensor | Characteristics | Reliability | Cost |
|---|---|---|---|
| Sonar Buoy | reacts to objects, does not differentiate between whale and submarine | 0.7 | low |
| Helicopter | reacts only to submarine | 0.9 | very high |

The ocean area is divided into four quadrants for the search (compare page 13, upper section). It is assumed in this example that sensors applied cover each entire quadrant. For the positions of submarine and whale there are first a priori probabilities of equal weight $p_o(H_{u_i})$, $p_o(H_{w_i}) = 0.25$, which exclude each other and therefore add up to 1:

$$p_o(H_{u_1}) + p_o(H_{u_2}) + p_o(H_{u_3}) + p_o(H_{u_4}) = 1$$

$$p_o(H_{w_1}) + p_o(H_{w_2}) + p_o(H_{w_3}) + p_o(H_{w_4}) = 1$$

The search begins when a sonar buoy is positioned randomly in quadrant 2. A signal is given that an object is positioned there (whale or submarine). The reliability (probability) of this finding is 0.7, resulting in revised probabilities for the above-defined hypotheses: /11

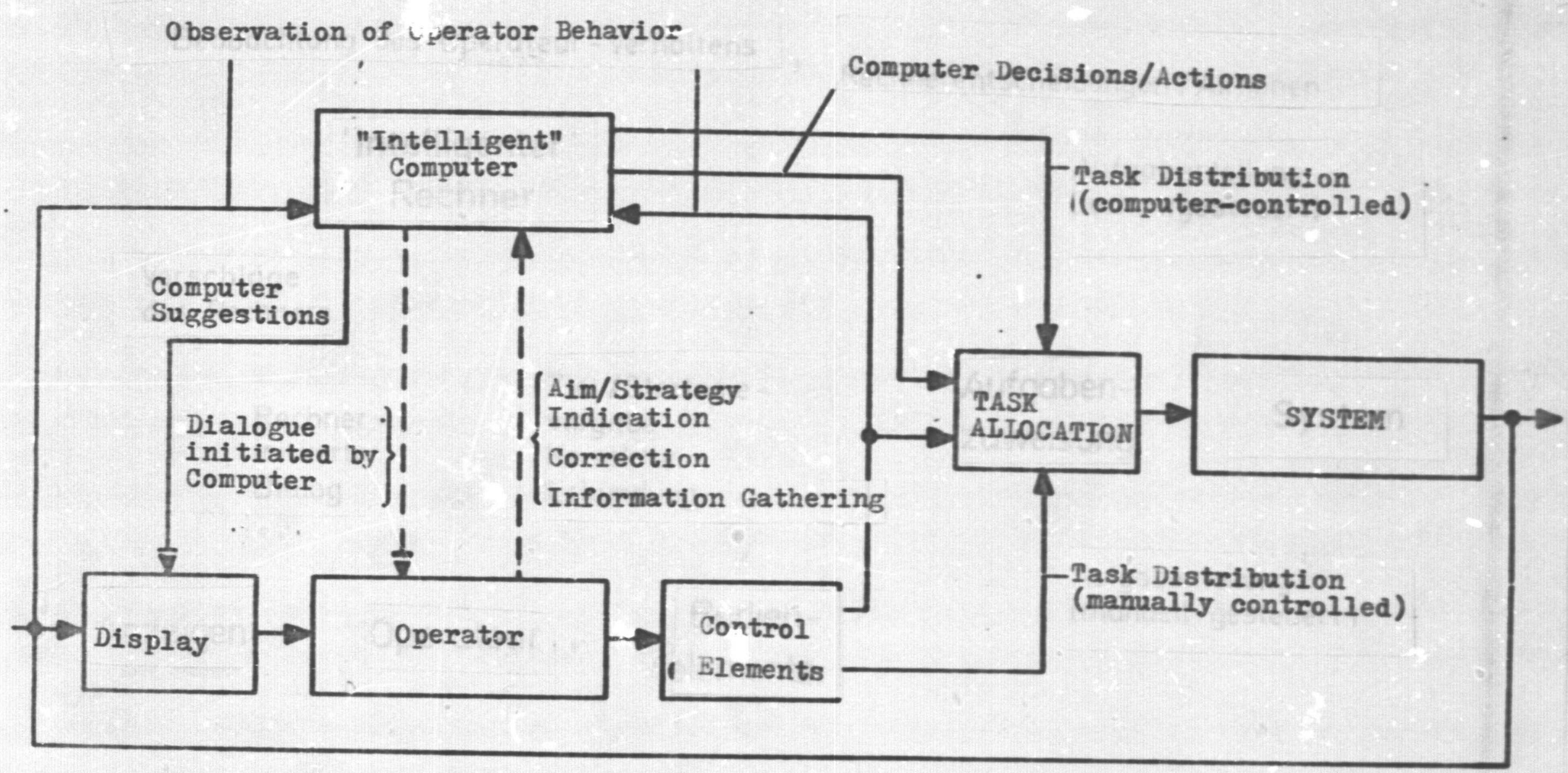



Fig. 1: Representation of Cooperation between Operator and adaptive Computer capable of Learning

$$p(H_{u_2} \mid D1) = \frac{0{,}7 \cdot 0{,}25}{0{,}7 \cdot 0{,}25 + 0{,}3 \cdot 0{,}25 + 0{,}3 \cdot 0{,}25 + 0{,}3 \cdot 0{,}25} = \frac{0{,}7}{1{,}6} \sim 0{,}43$$

$$p(H_{u_1} \mid D1) = \frac{0{,}3 \cdot 0{,}25}{0{,}3 \cdot 0{,}25 + 0{,}7 \cdot 0{,}25 + 0{,}3 \cdot 0{,}25 + 0{,}3 \cdot 0{,}25} = \frac{0{,}3}{1{,}6} \sim 0{,}19$$

$$= p(H_{u_3} \mid D1) = p(H_{u_4} \mid D1)$$

The same revised values result for the hypotheses with respect to the assumed position of the whale. $p(D1 \mid H_{u_2}) = 0.7$ is therefore the probability for the occurrence of observation D1 under the assumption that the hypothesis $H_{u_2}$ (submarine is in quadrant 2) is correct.

The next step is to search quadrant 2 with the helicopter, to ascertain whether the object positioned there is a submarine. Negative results are obtained. Reliability of this measurement is 90 %, resulting in the following revised probabilities:

$$p(H_{u_2} \mid D2) = \frac{0{,}1 \cdot 0{,}43}{0{,}1 \cdot 0{,}43 + 0{,}9 \cdot 0{,}19 + 0{,}9 \cdot 0{,}19 + 0{,}9 \cdot 0{,}19} = \frac{0{,}043}{0{,}56} \sim 0{,}08$$

$$p(H_{u_1} \mid D2 = p(H_{u_3} \mid D2) = p(H_{u_4} \mid D2)) = \frac{0{,}171}{0{,}556} \sim 0{,}31$$

The values for the position of the whale are not affected by this sensor report. It is then determined by a sonar buoy that there is an object in quadrant 1. This determination also has an error quotient of 30 %, resulting in:

$$p(H_{u_1} \mid D3) = \frac{0{,}7 \cdot 0{,}31}{0{,}7 \cdot 0{,}31 + 0{,}3 \cdot 0{,}08 + 0{,}3 \cdot 0{,}31 + 0{,}3 \cdot 0{,}31} = \frac{0{,}217}{0{,}405} \sim 0{,}53$$

$$p(H_{u_2} \mid D3) = 0{,}05 \,;\; p(H_{u_3} \mid D3) = p(H_{u_4} \mid D3) = 0{,}23$$

$$p(H_{w_1} \mid D3) = \frac{0{,}7 \cdot 0{,}19}{0{,}7 \cdot 0{,}19 + 0{,}3 \cdot 0{,}43 + 0{,}3 \cdot 0{,}19 + 0{,}3 \cdot 0{,}19} = \frac{0{,}133}{0{,}376} \sim 0{,}35$$

$$p(H_{w_2} \mid D3) = 0{,}34 \,;\; p(H_{w_3} \mid D3) = p(H_{w_4} \mid D3) = 0{,}15$$

Since there are a total of only two objects and these are probably in quadrants 1 and 2, it is now logical to search quadrant 1 with the helicopter for the submarine. A signal is given that a /12

| | Probability for the Position of | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Submarine | | | | Whale | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Starting situation and a priori probabilities for the hypotheses set down | .25 | .25 | .25 | .25 | .25 | .25 | .25 | .25 |
| 1. Sonar Buoy positioned: Information gained: Object in quadrant 2, 70 % certain | .19 | .43 | .19 | .19 | .19 | .43 | .19 | .19 |
| 2. Helicopter employed: Information gained: No submarine in quadrant 2, 90 % certain | .31 | .08 | .31 | .31 | | | | |
| 3. Sonar Buoy positioned: Information gained: Object in quadrant 1, 70 % certain | .53 | .05 | .23 | .23 | .35 | .34 | .15 | .15 |
| 4. Helicopter sent to quadrant 1: Information gained: Submarine in quadrant 1, 90 % certain. | .91 | ~.0 | .04 | .04 | | | | |

Table 1: Development of Probability Revision for the Example Submarine Location.

submarine was found there with a probability of 90 %, resulting in:

$$p(H_{u_1} \mid D4) = \frac{0{,}9 \cdot 0{,}53}{0{,}9 \cdot 0{,}53 + 0{,}1 \cdot 0{,}05 + 0{,}1 \cdot 0{,}23 + 0{,}1 \cdot 0{,}23} = \frac{0{,}48}{0{,}52} \sim 0{,}91$$

$$p(H_{u_2} \mid D4) = 0\,;\; p(H_{u_3} \mid D4) = p(H_{u_4} \mid D4) \sim 0{,}04$$

The submarine is therefore positioned in quadrant 1 with a probability of 91 %. Of course it depends on the strategy selected with what degree of certainty the reconnoissance is to be carried out, how much the search may cost and when the search is to be discontinued.

The development of probability revision for the positioning example is clearly summarized in Table 1. The reader should examine whether he would have come to the same conclusion subjectively.

Note also that the effect of an additional bit of information on the probability of individual hypotheses depends on the number of a priori hypotheses. The positioning of a sonar buoy, having a reliability of 70 %, increases e.g. in the case of a positive signal the probability of an hypothesis "submarine present" to the following values

| | | | | | | |
|---|---|---|---|---|---|---|
| from | 0.5 | to | 0.7 | in the case of | 2 | a priori hypotheses |
| from | 0.25 | to | 0.43 | in the case of | 4 | a priori hypotheses |
| from | 0.1 | to | 0.21 | in the case of | 10 | a priori hypotheses |
| from | 0.05 | to | 0.11 | in the case of | 20 | a priori hypotheses. |

## 3.3 The Probabilistic Information Processor of Edwards /14

Numerous experiments demonstrate [6], that men usually do not correctly estimate the significance of new additional information. Often a conservative behavior and a tendency to avoid extreme estimated values is observed. In general a considerable deviation from the values determined "correct" according to the Bayes' Rule is to be expected.

The probabilistic information processor (PIP) of Edwards (1963) is based on this observation [6,7]. The basic idea for this procedure is to have the Bayes' arithmetic processed by the computer while men still construct the basis for the a priori probabilities and for reliability of additional information (Fig. 2).

The function of this arrangement may be described as follows: on the lowest level information is collected, sorted and selected. On the next level experts attribute the individual data a significance in the form $p(D_k I H_i)$, an expression for the probability of

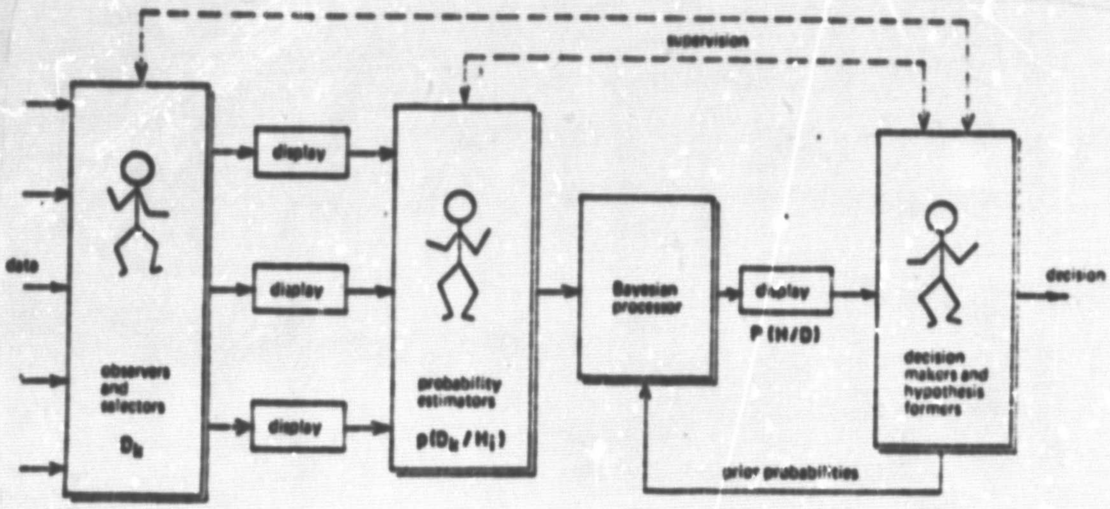



Fig. 2: Probabilistic Information Processor of Edwards (from [6])

information $D_k$ under the assumption that hypothesis $H_i$.

A computer (Bayes' Processor) carries out the revision for probability of a priori hypotheses with these data. On the basis of these revised probabilities other experts then make decisions, that is, formulate alternative a priori hypotheses. /15

Numerous studies show that the PIP system is a considerable aid in the decision making process. One example is given in Fig. 3 without a more exact description of the basic situation. The deviation of broken lines from the unbroken curve shows how much human error occurs in probability revision and how conservative the decisions are in each case.

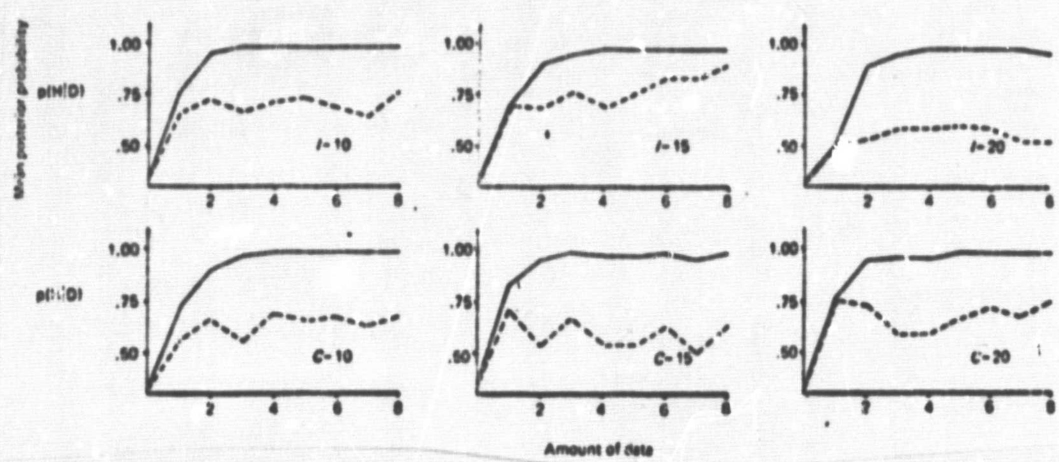



Fig. 3: Posterior Probabilities for the Correctness of an Hypothesis Estimated Subjectively (---) and Calculated with the Aid of PIP (——) [22]

It should be noted that the procedure described is only suitable for processing of relatively neutral technical questions in which the alternatives are well defined.

## 4. Computer Support in Decision Making /16

### 4.1 Decision Types and Usefulness

A one-dimensional decision situation is characterised by alternatives $A_i$ and by surrounding situations $B_j$. When the alternative $A_i$ is selected, while the surrounding conditions are $B_j$, the consequence is $C_{ij}$. This situation is demonstrated in the graphic of the matrix (Fig. 4).

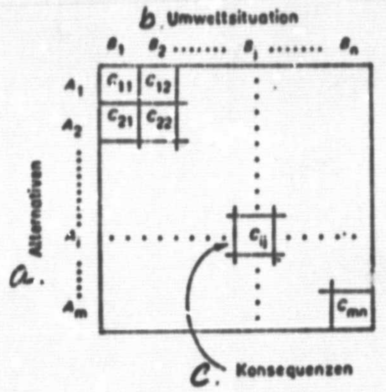



Fig. 4: Graph of a decision making situation: when $A_i$ is chosen at surrounding conditions $B_j$ the consequence is $C_{ij}$ (from [6])

Key: a. alternatives
b. surrounding situation
c. consequences.

Classification of decision types depends on the makeup of surrounding conditions $B_j$. A differentiation is made:

a. Decisions which are certain or without risk, when the surrounding conditions $B_j$ are known.

b. Decisions entailing some risks when for $B_j$ only probabilities may be given.
This is also termed the "game against nature " (the sum of all $p(B_j)$ is 1).

c. Uncertain decisions, where for $B_j$ no probabilities are known.

d. Games where $B_j$ is determined by another person. The players are opponents where the victory of one is equal to the loss of the other (games with a resulting sum of zero).

e. Dynamic decisions where a series of decisions is to be made and the matrix values A,B,C of individual decisions are influences by previously made decisions. Where no such influence occurs, the corresponding terminology is static decision.

Making a decision means, for all described decision types, selecting the consequence with the greatest advantage from several. The "usefulness" of a decision is a numerical index indicating how /17 strong the preference of a certain person is for this consequence.

The determination of usefulness index and the description and construction of models of human decision making behavior is subject of the decision analysis. The decision theory (Sheridan, Ferrell [6], Edwards, Tversky [8], Kaufmann [9] ) supplies normative, prescriptive models as aids in decision making and descriptive empirical models which can be tested for the description of an observed decision making behavior.

Within the structure of the decision making theory it is necessary to make certain plausible assumptions on how the preferences of a decision maker evolve. A simple usefulness structure, also employed in the following examples, may be based, for example, on the following assumptions:

- There must be several consequences $C_{ij}$. Formulation of these consequences is often difficult since the objectively identical facts may be subjectively evaluated differently.
- When comparing pairs of two consequences x,y from C, either x is preferred (xPy) or y (yPx) or neither of the two (x I y). (I = indifferent).
- For each consequence x there is a usefulness function EU(x).[1]
- If xPy, then EU(x) would be greater than EU(y).
- The preferences remain constant in time.
- The usefulness of multiple dimensional surrounding conditions may be determined by the usefulness of the elements. This means that the function of usefulness must make an interval scale possible.
- The usefulness of a determined alternative may be compared for various persons. This assumption makes it possible to combine the judgements of various persons.
- Usefulness structure is transitive, i.e. if xPy and yPz, then x is also preferred to the alternative z (xPz).

Here is a brief example for dealing with an usefulness function EU: in the case of uncertain decisions only the probability of a consequence is known, as was already explained. Under certain weather conditions (surrounding conditions $B_j$) it will, for example, rain with a probability $p_R$ and with a probability of $p_S = 1 - p_R$ the sun will shine. Assumed that a person has to decide under these weather conditions whether to take a raincoat or not, then each of the four logical alternatives in this lottery is apparently connected to a subjective usefulness expectation, as shown in the following matrix: /18

[1] Read: Expected Usefulness of Consequence x

| | Rain | Sun |
|---|---|---|
| Coat | $EU_{RC}$ | $EU_{SC}$ |
| No Coat | $EU_{RN}$ | $EU_{SN}$ |

A simple model for calculation of the total expected usefulness for action alternatives consists in adding the usefulness weighted by probability of occurrence of all possible consequences of an action:

$$EU_{Coat} = EU_{RC} \times p_R + EU_{SC} \times p_S$$

$$EU_{No\ Coat} = EU_{RN} \times p_R + EU_{SN} \times p_S$$

## 4.2 Static Decision Making Aids

A method for determining the preferences of a person is to play through all possible decision situations. The decision made may then be registered and stored in a computer which supplies the prepared solution for every decision to be made. This procedure has no practical application when the decision is based on surrounding conditions characterized by an entire series of factors.

A more rational method for determining a computer algorithm for evaluation of multiple dimensional alternatives is indicated by Yntema and Klem [10]. The procedure is demonstrated in the following simplified example of a certain decision: the decision /19 to be made consists in the selection of one of two nearby airports for an emergency landing. The surrounding situation is described by the parameters cloud height, visibility and fuel reserves, for which in each case 5 individual values are assumed. (Fig. 5). One-hundred-twenty-five different surrounding conditions result with this assumption.

| Level | Cloud Height | Visibility | Fuel Reserves |
|---|---|---|---|
| 1 | 100 | 1/4 | 15 |
| 2 | 300 | 1/2 | 30 |
| 3 | 1000 | 1 | 60 |
| 4 | 2000 | 2 | 120 |
| 5 | 5000 | 5 | 250 |

Fig. 5: Surrounding Conditions $B_{ji}$ for the Decision Making Task described in the Text (Yntema, Klem [10])

The "correct" airport for all situations could be experimentally determined by presenting experienced pilots with all possible parameter combinations for airports 1 and 2 in pairs.

The process would be so complicated that this method cannot be carried out practically. The alternative procedure of Yntema and Klem [10] consists in having experienced pilots determine working values in the form of an interval scale for the decisive values of individual surrounding parameter (see Fig. 6, left-hand side). Then the individual parameter are scaled (see Fig. 6, right). Scale x (cloud height), y (visibility) and z (fuel reserves) thus result. The expected usefulness (EU) which a certain parameter structure has for the decision to be made may be determined for each individual parameter by linear interpolation of the corresponding scale, for example:

$$EU(x) = C_1 + C_2 \, x \; (\cdot)$$

/20

Basic Value Scale Scales for Individual Parameters

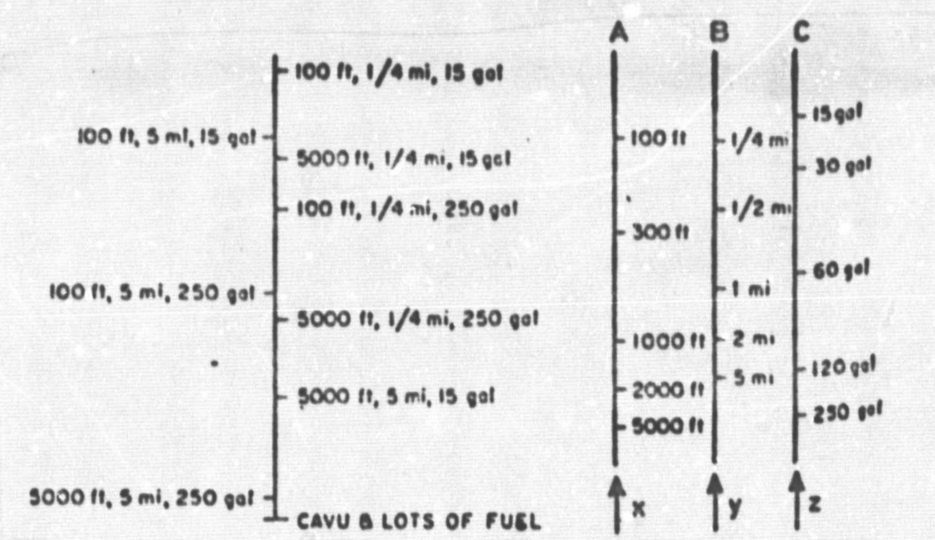



Fig. 6: Working Values (Interval Scales) for the Basic Values of Surrounding Parameter (left) and for Individual Parameter (right) according to Estimates made by Experienced Pilots (Yntema, Klem [10])

For the three-dimensional case in point in which three interpolated surrounding parameter are taken into consideration, the expected usefulness is calculated by the equation:

$$EU(x,y,z) = A + Bx + Cy + Dz + Exy + Fyz + Gxz + Hxyz$$

The constants A to H are selected in such a manner that the values calculated for the "basic combinations" are in agreement with those of the basic value scale (Fig. 6, left).

In a certain decision making situation the expected usefulness of each of the two alternative airports is determined by the computer corresponding to the above equation and then that one is selected with the greater expected usefulness.

When the decisions determined by the computer in this manner are compared with the decisions of pilots, it is demonstrated that the differences between individual pilots are greater than between any single pilot and the computer. The algorithm is therefore - on the average - a better model for decision making behavior of one pilot than any randomly chosen other pilot. The decisions of the algorithm are fully consistent. Pilot and computer algorithm deviate from one another merely in approx. 4 % of all cases. The application of a decision making aid with such a small error quotient could be helpful in cases in which any decision would be /21
better than none. Under such conditions the computer could be instructed to wait for a period of time until it takes an action appropriate for the operator, if the operator shows no reaction.

The interval scales in Fig. 6 are determined previously and usually not in genuine situations. The decision making strategy of the computer based on these scales is therefore fixed and not capable of adaptations. A continuously adjustment to new knowledge and information or a change of strategy is not possible with this procedure.

Computer algorithms capable of adapting, adjusting their decision making behavior continuously to the situation and to the strategy preferred by the operator, are discussed in the next section.

## 4.3 Dynamic Decision Making Aids Capable of Adaptation

In a decision making situation, characterized by various boundary conditions and consequences, there is as a rule a series of various situations leading to the same decision. These may then be categorized in classes in which each situation is characterized by a typical structure of certain parameter, i.e. by a characteristics vector. Division of classes changes with decision maker changes in strategy.

When the decision making analysis is to be transmitted to the computer, the necessity arises for automatically carrying out classification of characteristics vectors. In addition a continuous check is to be made whether the classification determined adequately reproduces the desired decision making behavior. Where necessary, the classes are to be adjusted to the altered boundary conditions adaptively.

There are numberous mathematical approaches to the problem of classification of characteristics. Solid introductions to these questions may be found in Nilsson [11], Duda and Hart [12], Young and Calvert [13] as well as Fu [14].

### 4.3.1 Classification of Characteristics Vectors /22

The function of a relatively simple classifier is to be explained here as far as this is required for understanding following examples. In principle we are dealing with the known Steinbuch learning matrix, the design of which is given in Fig. 7. This arrangement examines a series of characteristic vectors X and places them in two categories (+1,-1). Training of this learning matrix consists in determining the weight vector W in such a manner that each vector X is placed in the correct (desired) class.

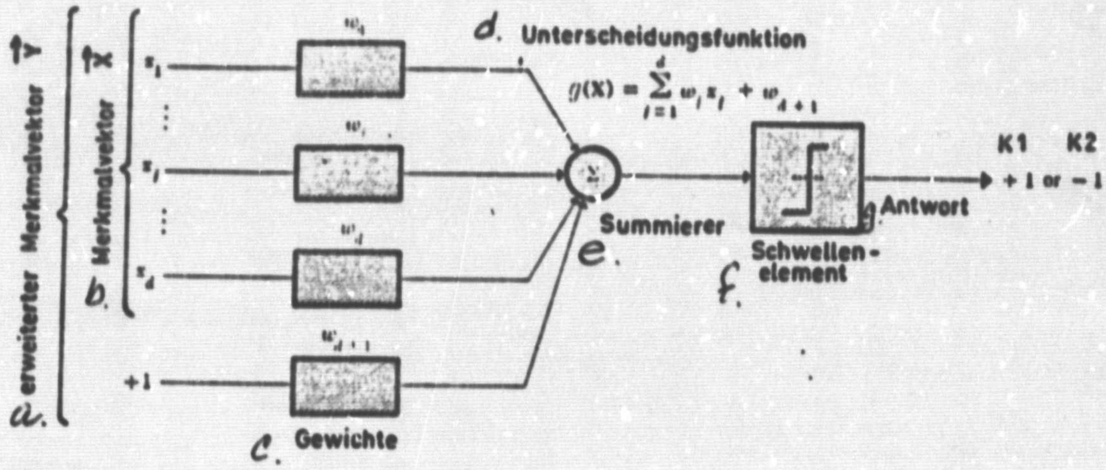



Fig. 7: Classifier for 2 Categories with Linear Differentiation Function (Learning Matrix) according to Nilsson [11]

Key: a. expanded characteristics vector
b. characteristics vector
c. weights
d. differentiation function
e. summation instrument
f. threshhold element
g. answer

Placement in a category is decided by the differentiation function g(X), resulting from multiplication of weight factor W with the characteristics vector expanded by "+1", as demonstrated in Fig. 7.

In training the learning matrix the actual placement of a characteristics vector is compared with the desired placement. When placement is correct, no correction is made and the next characteristics vector is examined. In the case of an erroneous placement a correction of weight vector is made according to the following rule:

W' = W + c.Y when placement is made to category 2, but placement to category 1 (+1) is desired.

W' = W - c.Y in the opposite case.

For selection of value of c there are various procedures, converging at different rates, but which lead to the same result. For linear differentiation functions g(X) (only this simplified case will be taken into consideration here) this training supplies a solution after a finite number of steps in every case. If the /23 categories may not be separated in a linear manner, there is a remainder error (Slagle [15]). In this case the training may also be discontinued after a finite number of steps when the remainder error has reached the desired size.

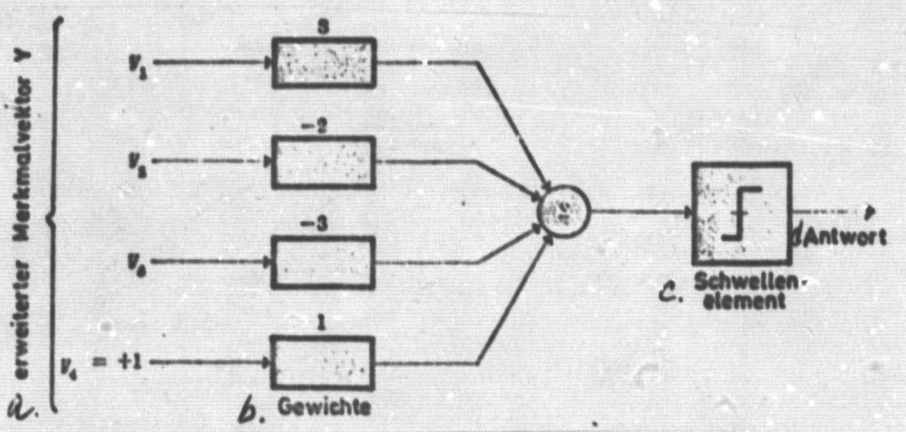
The following example (of Nilsson [11]) serves to demonstrate training procedure for learning matrix:
eight three-dimensional vectors are to be placed in the categories +1,-1 in the required manner (Table 2). The object of the search is a weight vector W which ensures that the learning matrix can perform this placement.

| Vector No. | Components $y_1$ | $y_2$ | $y_3$ | Expansion $y_4$ | desired Placement |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 | -1 |
| 2 | 0 | 0 | 0 | 1 | 1 |
| 3 | 1 | 0 | 0 | 1 | 1 |
| 4 | 1 | 0 | 1 | 1 | 1 |
| 5 | 0 | 0 | 1 | 1 | -1 |
| 6 | 1 | 1 | 0 | 1 | 1 |
| 7 | 1 | 1 | 1 | 1 | -1 |
| 8 | 0 | 1 | 0 | 1 | -1 |

Table 2: Characteristics Vectors and Desired Placement [11]

The course of training for a correction procedure with fixed increment c = 1 is shown in the Table below. The learning process is completed, as can be seen, after 29 patterns have been presented and after 13 corrections in weight vector have been made.

/24




Key:
a. Expanded characteristics vector Y
b. Weights
c. Threshhold element
d. Answer

Fig. 8: Trained Learning Matrix for the Example explained in the text [11]

| a. erweiterter Merkmalvektor | | | | b. Gewichtsvektor | | | | | c. Zuordnung | d. gewünschte Zuordnung | e. Korrektur | f. neuer Gewichtsvektor | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $y_1$ | $y_2$ | $y_3$ | $y_4$ | $w_1$ | $w_2$ | $w_3$ | $w_4$ | W · Y | | | | $w_1$ | $w_2$ | $w_3$ | $w_4$ |
| | | | | | | | | **Iteration 1** | | | | | | | |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | • | −1 | yes | 0 | −1 | −1 | −1 |
| 0 | 0 | 0 | 1 | 0 | −1 | −1 | −1 | −1 | −1 | 1 | yes | 0 | −1 | −1 | 0 |
| 1 | 0 | 0 | 1 | 0 | −1 | −1 | 0 | 0 | • | 1 | yes | 1 | −1 | −1 | 1 |
| 1 | 0 | 1 | 1 | 1 | −1 | −1 | 1 | 1 | 1 | 1 | no | 1 | −1 | −1 | 1 |
| 0 | 0 | 1 | 1 | 1 | −1 | −1 | 1 | 0 | • | −1 | yes | 1 | −1 | −2 | 0 |
| 1 | 1 | 0 | 1 | 1 | −1 | −2 | 0 | 0 | • | 1 | yes | 2 | 0 | −2 | 1 |
| 1 | 1 | 1 | 1 | 2 | 0 | −2 | 1 | 1 | 1 | −1 | yes | 1 | −1 | −3 | 0 |
| 0 | 1 | 0 | 1 | 1 | −1 | −3 | 0 | −1 | −1 | −1 | no | 1 | −1 | −3 | 0 |
| | | | | | | | | **Iteration 2** | | | | | | | |
| 0 | 1 | 1 | 1 | 1 | −1 | −3 | 0 | −4 | −1 | −1 | no | 1 | −1 | −3 | 0 |
| 0 | 0 | 0 | 1 | 1 | −1 | −3 | 0 | 0 | • | 1 | yes | 1 | −1 | −3 | 1 |
| 1 | 0 | 0 | 1 | 1 | −1 | −3 | 1 | 2 | 1 | 1 | no | 1 | −1 | −3 | 1 |
| 1 | 0 | 1 | 1 | 1 | −1 | −3 | 1 | −1 | −1 | 1 | yes | 2 | −1 | −2 | 2 |
| 0 | 0 | 1 | 1 | 2 | −1 | −2 | 2 | 0 | • | −1 | yes | 2 | −1 | −3 | 1 |
| 1 | 1 | 0 | 1 | 2 | −1 | −3 | 1 | 2 | 1 | 1 | no | 2 | −1 | −3 | 1 |
| 1 | 1 | 1 | 1 | 2 | −1 | −3 | 1 | −1 | −1 | −1 | no | 2 | −1 | −3 | 1 |
| 0 | 1 | 0 | 1 | 2 | −1 | −3 | 1 | 0 | • | −1 | yes | 2 | −2 | −3 | 0 |
| | | | | | | | | **Iteration 3** | | | | | | | |
| 0 | 1 | 1 | 1 | 2 | −2 | −3 | 0 | −5 | −1 | −1 | no | 2 | −2 | −3 | 0 |
| 0 | 0 | 0 | 1 | 2 | −2 | −3 | 0 | 0 | • | 1 | yes | 2 | −2 | −3 | 1 |
| 1 | 0 | 0 | 1 | 2 | −2 | −3 | 1 | 3 | 1 | 1 | no | 2 | −2 | −3 | 1 |
| 1 | 0 | 1 | 1 | 2 | −2 | −3 | 1 | 0 | • | 1 | yes | 3 | −2 | −2 | 2 |
| 0 | 0 | 1 | 1 | 3 | −2 | −2 | 2 | 0 | • | −1 | yes | 3 | −2 | −3 | 1 |
| 1 | 1 | 0 | 1 | 3 | −2 | −3 | 1 | 2 | 1 | 1 | no | 3 | −2 | −3 | 1 |
| 1 | 1 | 1 | 1 | 3 | −2 | −3 | 1 | −1 | −1 | −1 | no | 3 | −2 | −3 | 1 |
| 0 | 1 | 0 | 1 | 3 | −2 | −3 | 1 | −1 | −1 | −1 | no | 3 | −2 | −3 | 1 |
| | | | | | | | | **Iteration 4** | | | | | | | |
| 0 | 1 | 1 | 1 | 3 | −2 | −3 | 1 | −4 | −1 | −1 | no | 3 | −2 | −3 | 1 |
| 0 | 0 | 0 | 1 | 3 | −2 | −3 | 1 | 1 | 1 | 1 | no | 3 | −2 | −3 | 1 |
| 1 | 0 | 0 | 1 | 3 | −2 | −3 | 1 | 4 | 1 | 1 | no | 3 | −2 | −3 | 1 |
| 1 | 0 | 1 | 1 | 3 | −2 | −3 | 1 | 1 | 1 | 1 | no | 3 | −2 | −3 | 1 |
| 0 | 0 | 1 | 1 | 3 | −2 | −3 | 1 | −2 | −1 | −1 | no | 3 | −2 | −3 | 1 |

*undefined placement

Table 3: Course of training for the example explained in the text according to a correction procedure with fixed increment c = 1 [11]

Key: a. expanded characteristics vector
- b. weight vector
- c. placement
- d. desired placement
- e. correction ?
- f. new weight vector

The 2 category-classifier described carries out the most simple possible placement tasks. Normally the task consists in sorting all patterns of a group X in such a manner that these are placed in only one category and not two. For this purpose an expanded learning matrix is applied with the structure shown in Fig. 9. In this several elementary learning matrices can be discerned with exits additionally running through an maximum selector. /25

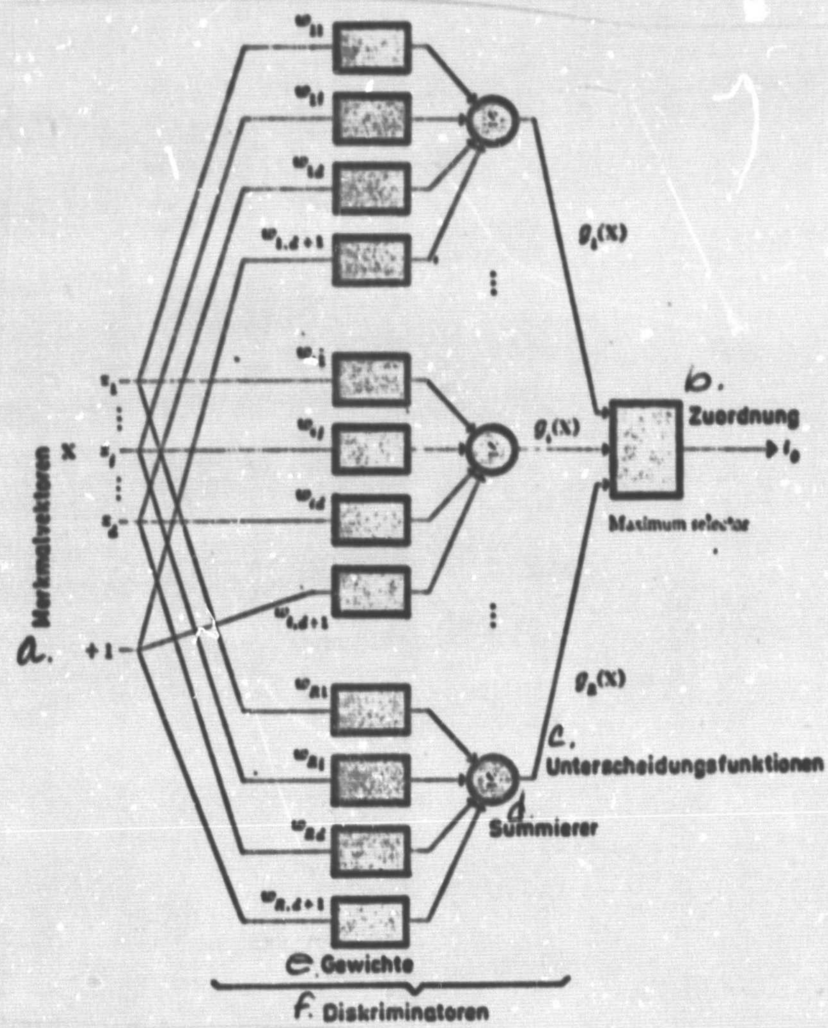



Fig. 9: Classifier for 1 Category with Linear Differentiation Function [11]

Key: a. Characteristics vectors
b. Placement
c. Differentiation functions
d. Summation instrument
e. Weights
f. Discriminators

The placement of a characteristics vector $Y_i$ in the category i is correct when the differentiation function $g_i(Y)$ for this vector becomes greater than all other $g_i(Y)$. In case any other weight function should become greater, the pattern is placed in an erroneous class, with the result that the weight vectors $W_i$ and $W_j$ have to be altered in pairs. For the correction the already known equations are applied ($W_i' = W_i + c.Y$ or $W_j' = W_j - c.Y$).

## 4.3.2 Example for Classification of Decision Making Situations /26

For matters of simplicity the example used in Section 3.2 for demonstrating revision of probabilities "submarine location" is further developed here. It can be seen from the representation of probability matrices for position "submarine" and "whale" (compare Table 1, p. 9), that the tactical situation on which a decision is based is described in each case by an eight-dimensional characteristics vector P:

$$P = \{ p(H_{u_1}), p(H_{u2}), p(H_{u_3}), p(H_{u_4}), p(H_{w_1}), p(H_{w_2}), p(H_{w_3}), p(H_{w_4}) \}$$

This characteristics vector may assume any number of values in the course of time. On the other hand only 8 different actions E1 - E8 may be chosen:

E1 : Search with helicopter in quadrant 1
E2 : Search with helicopter in quadrant 2
E3 : Search with helicopter in quadrant 3
E4 : Search with helicopter in quadrant 4
E5 : Search with sonar buoy in quadrant 1
E6 : Search with sonar buoy in quadrant 2
E7 : Search with sonar buoy in quadrant 3
E8 : Search with sonar buoy in quadrant 4

The task of an operator consists in making one of eight possible decisions after observation a certain situation characterized by a characteristics vector P. On the basis of his experience he will chose solution $E_i$, which holds promise for him of the greatest (reconnoissance) usefulness $EU_{E_i}$. Corresponding to Section 4.1, p. 14 -

$$EU_{E_i} = p(H_{u_1}) \times EU_{u1} + \dots\dots\dots\dots + p(H_{w_4}) \times EU_{w_4}$$

The operator therefore carries out a classification task as was dealt with in the previous section: classification of patterns in eight categories.

Notice that the differentiation function $g_i(X)$ is constructed in the same manner as the equation for the total expected usefulness of a decision under conditions of uncertainty:

$$g(x) = \sum_{i}^{d} w_i \times x_i + w_{d+1}$$ /27

The arrangement demonstrated in Fig. 9 may therefore be inserted as adaptive classifier capable of being trained.

The meaning of an expected usefulness EU, as is subjectively estimated by operator [16], may be attributed to the individual weights. Continuously decisions of the operator are observed for classifier training and assumed as desired values (desired placement). If no agreement can be made, a correction of weight factors, as described in the previous section, is made. When all tasks have been learned, the classifier makes the decision which the operator would also make. This arrangement therefore does not supply absolute optimal solutions but merely copies the operator behavior, including his mistakes.

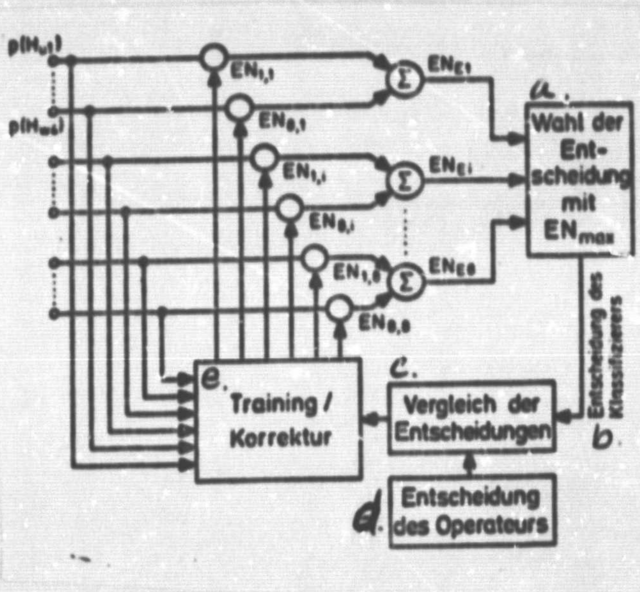



Fig. 10: Classifier for the Example "Submarine Location." See Text for Details (altered according to [16])

$EU_{i,j}$ = Subjective estimation of usefulness resulting in this situation from the information on quantity of $p(H_{n,m})$.

$EU_{Ei}$ = Expected usefulness of decision Ei.

Key: Note: All "EN" in this figure are understood as "EU".

a. Choice of decision with $EU_{max}$

b. Decision of classifier.

c. Comparison of decisions.

d. Decision of operator.

e. Training/correction.

## 4.3.3 Characteristics of Adaptive Usefulness Estimation /28

A special advantage of the method described is that usefulness estimation of the operator is determined here dynamically in a genuine situation and actual time. The arrangement is thus capable of reacting continuously to alterations in strategy or decision style of an operator, dependent on growing experience, lessening concentration or alteration of boundary conditions of problem, among other things.

Experiments show that this training rapidly leads to a lessening in error quotient under constant boundary conditions when the operator maintains consistent behavior. In general a stabile condition is achieved after a few decision cycles.

During the learning process two typical phases are differentiated (Fig. 11): In the beginning a rough adjustment occurs with a correction of weight factors after each decision. Subsequently corrections only occur after several decision cycles in each case. This second phase therefore represents a fine tuning and an adjustment to smaller variations in operator behavior.

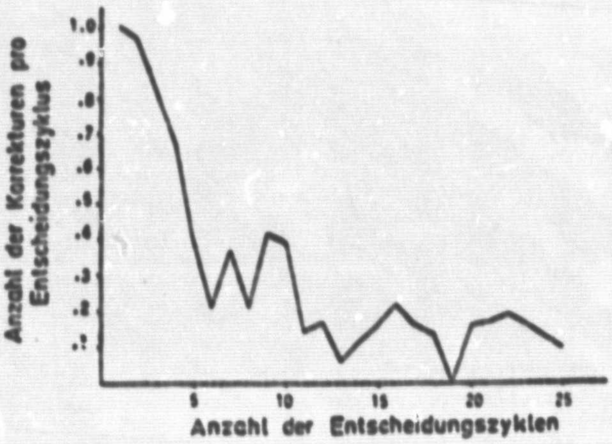



Fig. 11: Number of Corrections of Usefulness Estimated Values as a Function of Decision Cycles [17]

Key: a. Number of Corrections per Decision Cycle

b. Number of Decision Cycles

As soon as the classifier training has achieved the phase of fine tuning, and this is the case after a few cycles, most decisions of the operator are correctly predicted. When the strategy of the operator changes, a new adjustment process of classifier begins, in /29
which the learning matrix takes some new information more strongly into consideration than others and reacts in a manner similar to the human memory.

A normative significance may be attributed to the learned, estimated values of usefulness. The decision predictions of classifier based on these may be indicated to operator, who then employes them as decision-making aids or for the control of his decision-making behavior, whereby a closed circuit control is created as is demonstrated in Fig. 12.

## 4.3.4 The ADDAM System[1)]

The example "submarine location" in Section 4.3.2 was intentionally simplified in order to show the adaptive usefulness estimation better. Freedy et al. [18] conducted studies with a much more complex reconnoissance system, a relatively realistic experimental situation in which various decision aids may be tested in an actual time situation.

1) ADDAM = Adaptive Dynamic Decision Aiding Methodology

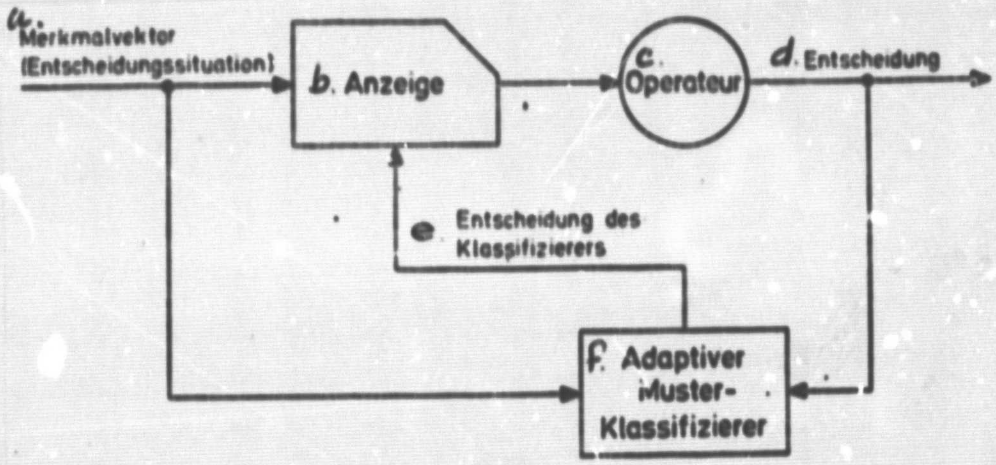



Fig. 12: Decision Aids and Controls on the Basis of Adaptive Usefulness Estimation in form of Closed Circuit Control [18]

Key: a. characteristics vector (decision making situation
b. display
c. operator
d. decision
e. decision of classifier
f. adaptive pattern - classifier

The reconnoissance situation selected is shown in Fig. 13. The position and direction of an enemy submarine and a second neutral object (whale) in an area of the ocean covering 25 sectors is to be determined. For reconnoissance a total of eight different sensor types may be positioned, in each sector, however, only one. The sensors differ from one another in sensitivity, reliability and in costs. One sensor is able to search the entire sector in which it was positioned in each case. Position reports are not transmitted continuously, but in intervals of 15 minutes. /30

Operator performance in this reconnoissance task is measured by means of an index taking into consideration correct and incorrect position reports as well as cost for the sensors deployed. This index is indicated to the operator and is the basis for reward at the end of the trial.

Aim of the experiments conducted was to test effect of adaptive decision aids on operator performance (test of ADDAM). The system contains two intelligent decision aids -

1. Help in gathering information - After each new report of deployed sensors a list of revised probabilities for target positions is processed and displayed (Bayes' Rule).
2. Help in selection of action (placement of sensors) - The computer determines expected usefulness of placement of a certain sensor in a certain sector for the operator /31

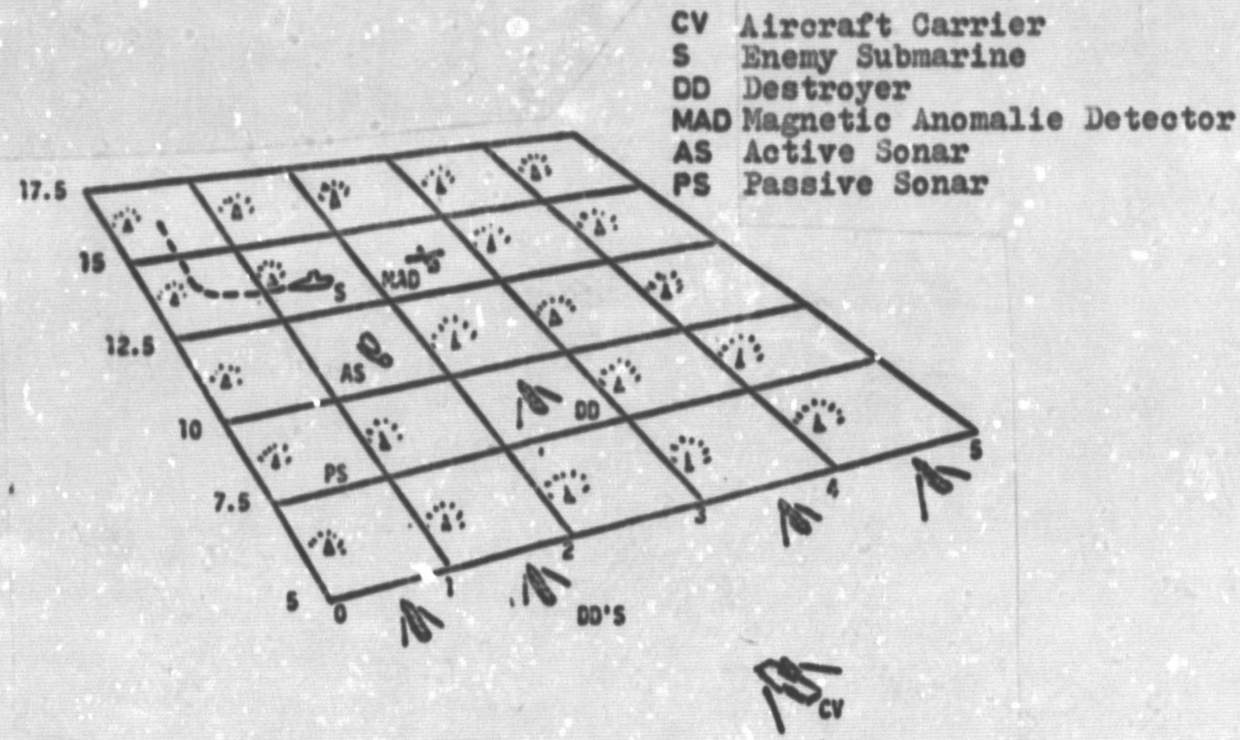



Fig. 13: The Situation Employed for Experimental Examination of ADDAM [18]

on the basis of adaptive usefulness estimation. Placement suggestions may be taken from a list of priorities for expected usefulness prepared by the computer. Suggestion for action by the computer represents the sensor placement with greatest expected usefulness. It is left to the discretion of the operator to deviate from this plan.

A block diagram containing both described computer aids is given schematically in Fig. 14.

Experiments demonstrate that the adaptive model already predicts the decisions of operator with a great degree of accuracy after a short period of time (correlation 82 %). Quality of decisions measured in the above indicated index is doubled. With computer aids higher sensor costs may be incurred, but the number of hits increases and there are fewer mistaken reports. The system is gladly accepted by the experimental personal and regarded as intelligent interactive partner after some practice. It may be decisive in this process that the trial personnel is informed about how suggestions of the computer result and that this is substantially a copying of their own behavior.

Operator decisions are more consistent with the aid of the computer, the interindividual variations are also reduced. In addition it is also observed that the decision cycle runs a more /32

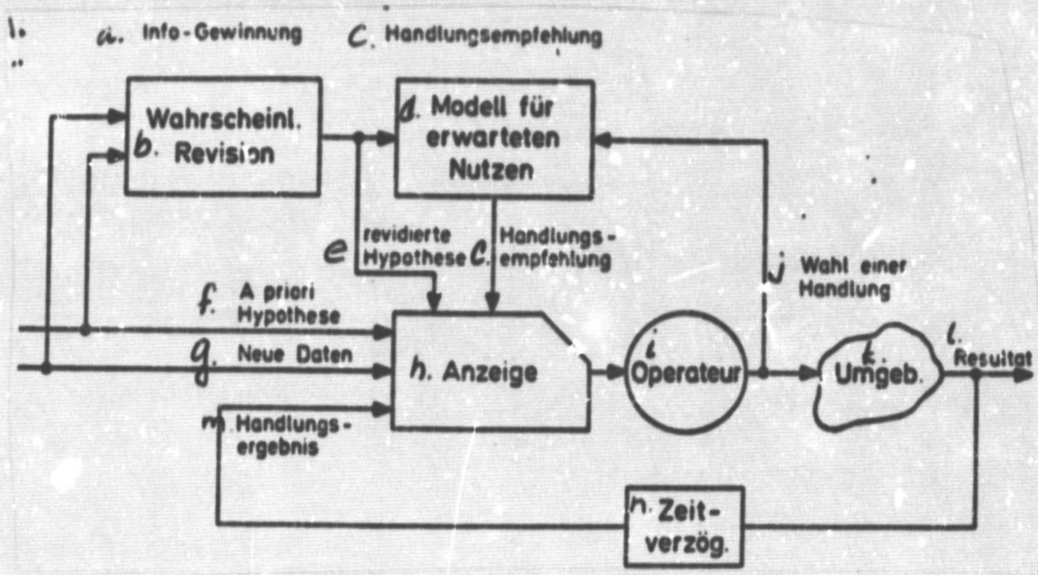



Fig. 14: The ADDAM System [18]

Key: a. Gaining information — b. Probability revision
c. Recommendation for action — d. Model for expected usefulness
e. Revised hypothesis — f. A priori hypothesis
g. New data — h. Display
i. Operator — j. Selection of action
k. Surroundings — l. Result
m. Result of action — n. Time delay

rapid course due to computer assumption of bookkeeping and sorting. Computer bookkeeping is a valuable support for human memory and concentration.

It is concluded from the experiments that the system described is especially suitable for tasks in which general guidelines may be given but the individual action is left to the operator.

It is left to remark on the value of the measures suggested by the computer. The rank series provided by the computer corresponding to expected usefulness typically shows that the first 3 - 5 suggestions in the list have a substantially higher EU value than the subsequent remainder. This result in various decision cycles and trial personnel. Since the placement suggestions of lesser priority have a lower expected usefulness but cost more, these suggestions may be suppressed by the computer in order to keep less helpful information from the operator who is already under great stress.

### 4.3.5 Task Distribution Man-Computer on the Basis of Adaptive Usefulness Estimation

Systems containing intelligent adaptive components capable of learning will be able to take over many tasks in the future which must now be carried out by the operator. Typical design is given in Fig. 15 in which parallel to the conventional man-man control circuit a second loop containing an adaptive computer is shown. This learns by means of observation of system behavior and is in a trained condition capable of taking action or giving recommendations on the basis of stored experience. Systems using this concept are already being applied. Typical examples are remote-controlled missiles, intelligent rockets, teleoperators and robots.

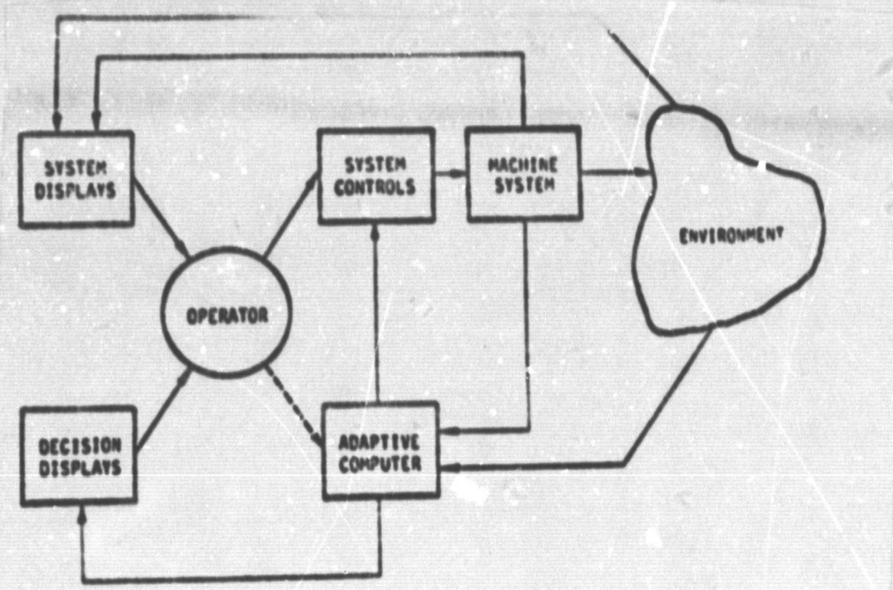



Fig. 15: System with adaptive Components capable of learning [19]

Application occurs in all these cases in non-deterministic situations characterized by a risk attached to all actions and decisions. This is valid for human actions as well as for those suggested by computers. /33

The operator is assisted by a system generating its own behavior patterns in time and giving suggestions which have only a certain degree of confidence.

The success of such an arrangement depends on whether the computer can win the operator's confidence by means of correct prognoses. When the operator realizes that the computer has learned enough to be a reliable aid, he will delegate an increasing amount of decision making tasks.

An example for an adaptive aid system capable of learning is indicated by Freedy et al., 1973 [19,21] and by Crooks et al, 1974 [20] (ACS - autonomous control system). This is a system applied for learning a matrix of dependent probabilities. The matrix permits prognoses via observation of present conditions and past system conditions. The probability of a correct prognosis increases with number of observations. The system decides in each case in favor of the action alternative with greatest probability of success.

The manner of function of this system is explained here using a greatly simplified example (Fig. 16). The task consists in assisting a missile to a avoid an obstacle and resulting collision. The task difficulty results from the threat probability represented by the obstacle, a non-linear function of distance from the missile (Fig. 17). Threat profile is known to the operator because of previous instruction. On the basis of this knowledge he attempts to place the missile on a path of most security. Depending on where the obstacle may appear he can select from 10 possible action alternatives (possible linear flight paths). /34

Path No. i,j

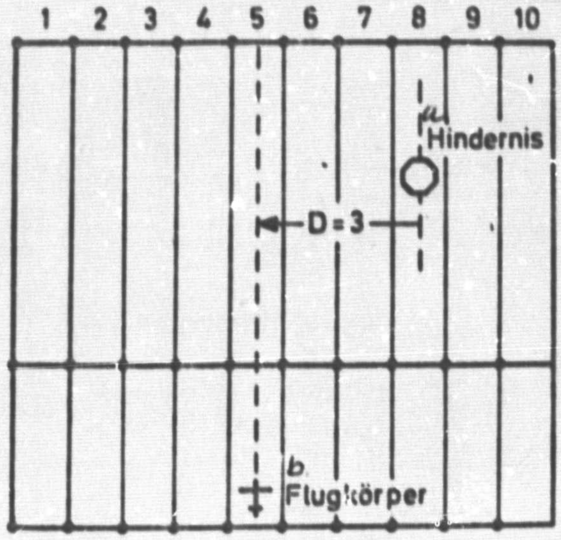



Fig. 16: Description of the Task applied by Crooks and Freedy [19,20]

i = Obstacle path
j = Missile path
D = (i-j) Distance between obstacle and missile

Key: a. Obstacle
b. Missile

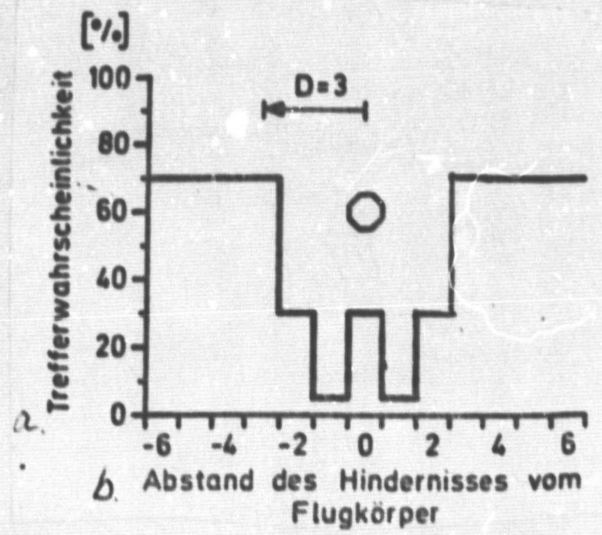



Fig. 17: Threat Profile of an Obstacle [19,20]

Key: a. Hit probability
b. Distance of obstacle to the missile

The computer observes actions of the operator and resulting consequences (hit/no hit). All possible combinations for the path i of obstacle and j of missile are a priori equally promising of success for the computer, expressed in an internally stored probability matrix (Fig. 18), showing in all fields above the value 0.1. The matrix is first untrained. /35

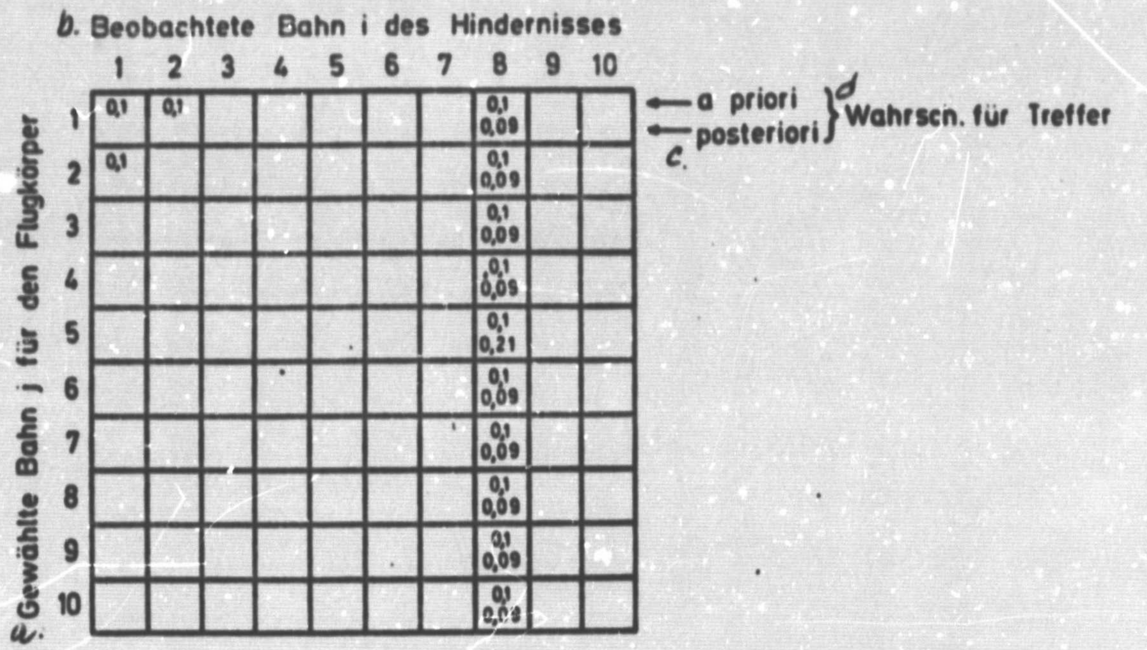



Fig. 18: Matrix for the Probability $p_{ij}$ of a Hit when the Obstacle is on Path i and the Missile on Path j

Key: a. Path j selected for missile
b. Path i observed, of the obstacle
c. Posterior
d. Probability for a hit

For the situation shown i = 8, j = 5, a hit probability of 70 % may be read from Fig. 17 in the case of a hit report. With this additional information the revised probabilities noted in the lower portion of column 8 in the matrix elements result (compare Section 3.1). It can be seen that for the next trial with this 8 - 5 configuration 9 alternatives are still equally promising of success (p=0.09). The learning process is continued with each observation. In time there is then for each configuration missile/obstacle a path j of relatively low hit probability. That is the path which the computer would chose. Reliability of this choice depends on the absolute value of corresponding probability. The p -matrix accumulates all information transmitted continuously in this manner. It continually becomes more clever in this manner and forgets nothing. When boundary conditions (threat profiles) /36
remain constant, it performs better than the operator after a defined point.

While the examples described may still be readily surveyed, the relationships studied by Crooks and Freedy using three obstacles simultaneously is substantially more complex.

For task distribution between operator and computer there first exists the possibility of determining a defined level of confidence. As soon as this level is exceeded the computer automatically assumes control. It is clear that determination of this level is decisive for system effectivity - when the level is low, computer decisions are speculative and liable to be erroneous; for high levels the computer is applied only seldom.

A second possibility consists in leaving the decision about task distribution to the operator where necessary. The disadvantage of this possibility is that the operator usually does not carry out this distribution optimally because he underestimates usefulness of computer assistance and his own work load. The operator has the choice of one of two lotteries -

1. To act himself with a defined probability of success
2. Delegation of decision to the computer which also promises a certain probability of success.

| | Successful | Unsuccessful |
|---|---|---|
| Lottery 1; computer decides | $p_{RE}$ | $p_{RM}$ |
| Lottery 2; operator decides | $p_{OE}$ | $p_{OM}$ |

In each decision making situation the operator makes subjective estimation of usefulness of all four possible consequences, resulting in expected usefulness (EU) of the two lotteries:

$$EN_{Lot1} = p_{RE} \times EN_{RE} + p_{RM} \times EN_{RM} \; ; \; EN_{Lot2} = p_{OE} \times EN_{OE} + p_{OM} \times EN_{OM}$$

Note: Read for "EN" = "EU".

The operator will chose the lottery with the greatest usefulness expectation for him.

The values $EU_{Lot1}$ and $EU_{Lot2}$ must be determined for an automatic adaptive task distribution between man and computer. It may be recognized that this again is a problem in classification. Characteristics vector $P = [p_{RE}\ p_{RM}\ p_{OE}\ p_{OM}]$ should be arranged in two categories. For this the usefulness parameter $EN\ (EN_{RE}\ EN_{RM}\ EN_{OE}\ EN_{OM})$ (Note; read "EU" for "EN") are to be learned by observation of operator behavior (compare Section 4.3.1). The weight vector which actually has no meaning in the case of pattern classification obtains in this application case the significance of usefulness expected subjectively by the operator from a certain possible consequence. /37

Where the characteristics vectors may be separated in a linear manner and when the operator acts in a completely consistent fashion, the training results in discret EU values. In reality remainder errors are observed, caused by memory and information processing mistakes of operator and by the fact that the patterns may not be separated in a linear manner.

The described method adaptive task distribution man/computer has proven itself in experiments. It may be shown that operator preferences are substantially influenced by costs in connection with the choice of a lottery. Usefulness estimation of computer decisions increases when costs of operator decisions increase and vice versa.

The indication of confidence level, i.e. of learning condition of classifier, intensifies the application frequency of automatic control systems considerably. The operator then has a better possibility for estimating the chances of success of the computer.

## 5. Computer Assistance in Problem Solving /38

### 5.1 Problems of the Type "Game against Nature"

The task of an operator in guidance and control systems frequently consists in conducting a system from a known beginning condition to a desired target condition. During the transition various boundary conditions (game rules) have to be adhered to as a rule. For this problem type numerous examples may be listed from the area of control technology, operations research, navigation, distribution of goods, procedure guidance etc.

An example from naval navigation is explained more closely here especially suitable for the problem solving procedure subsequently described:

A freighter is navigating in an heavily trafficed water way. The ship represents a dynamic system with a condition vector described at time t by a differential equation [23]:

$$\frac{d\underline{\xi}}{dt} = f[(\underline{\xi}, \underline{u}, t)]$$

In this equation $\underline{\xi}$ $(\alpha, \beta)$ describes the coordinates of location and $\underline{u}$ $(v, \vartheta)$ describes the steering vector including velocity and course.

Explicit equation for ship's motion:

$$\frac{d\alpha}{dt} = v(t) \cos \vartheta(t) \quad ; \quad \frac{d\beta}{dt} = v(t) \sin \vartheta(t)$$

For the motion of other ships in the waterway the analog equation is -

$$d\underline{\eta}^k / dt = \underline{g}^k (\underline{\eta}^k, \underline{q}^k, t) \qquad ; k = 1 \ldots r$$

With this formulation the problem may be defined as follows: observe the traffic situation, i.e. the position and motion of other ships and determine a control strategy $\underline{u}(t)$ based on this observation in such a way that your own ship is transported from a starting point $\xi_0$ to a target point $\xi_z$. Boundary conditions in this case are that a minimum distance $(\delta_{min})$ must be maintained at all times (avoidance of collision) and that travel is kept to a minimum. /39

Solution of this problem in closed form is possible (variation calculation) but difficult. Realisation is often not carried out because calculation may not be conducted in actual time.

### 5.1.1 Problem Representation as Decision Tree

Alternatively to the closed mathematical treatment the problem may be solved by means of a systematic analysis of successive system conditions (dynamic programming) [24]. Procedural method in the case of dynamic programming consists in a step-wise and continuous generation of new system conditions beginning with the starting condition. Boundary conditions and rules are to be observed in the transition to new system phases. Where the problem has a solution, the desired final condition will be found among the variants sooner or later. Once the target condition is reached, the path to solution may be retraced to the starting condition.

Graphs are suitable for representation of possible transitions between system conditions. The points of graph represent system phases, the connecting lines between the points are the operators conducting from one condition to another. The decision tree for a puzzle with 8 pieces is given as an example in Fig. 19 [25].

The game rules may be derived from the decision tree: shift the pieces in such a way that they come to lie in the correct order, compare the target point.

A form is to be chosen for problem representation within the graph, economical as possible, i.e. fitting to the problem. Symbol chains, vectors, fields and lists are possible.

Graphs are either represented explicitly in the form of tables /40
or implicitly via an algorithm, indicating in what manner the points of the graph are to be developed. In each case the portion of the graph leading to target point must be made explicit. Algorithms for development of graphs contain non-deterministic distributions and branching.

### 5.1.2 Searching the Decision Tree for a Solution

Problem solving consists in searching the graph for possible paths to the target phase while observing certain criteria such as path length and time. For this search application may be made of

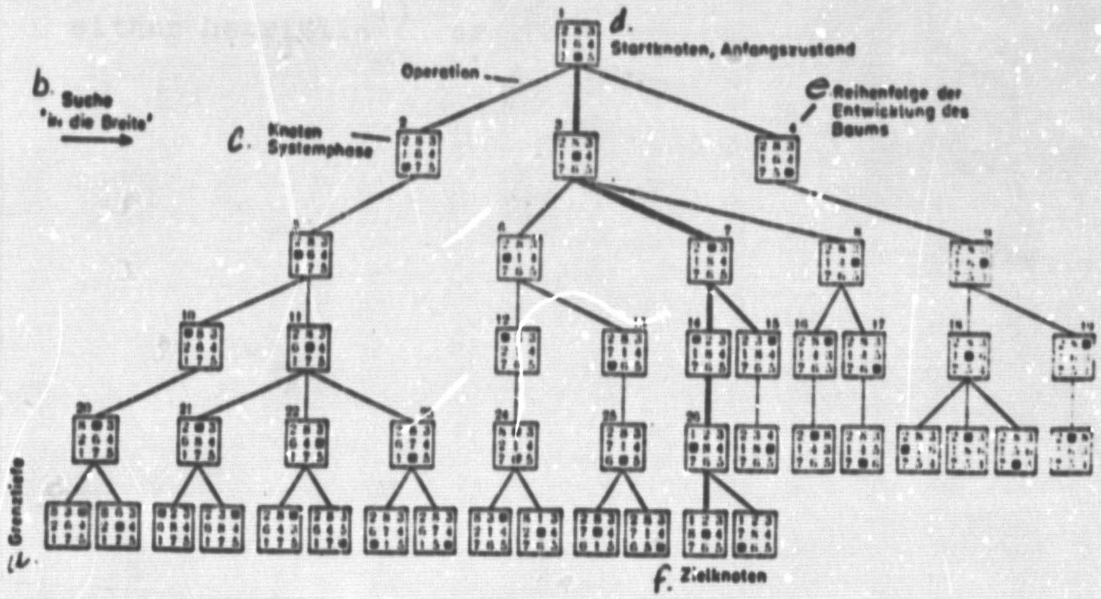



Fig. 19: Decision Tree for a Puzzle with 8 Pieces

Key:
- a. limit depth
- b. search "in breadth"
- c. point system phase
- d. starting point, beginning condition
- e. order of development of the tree
- f. target point

either exhaustive[1] or heuristic[2] techniques.

Complete procedures systematically search all branches of the graph until a solution is found or the computer capacity is exhausted. For this purpose varying strategies may be applied. In the case of /41 the search "in breadth" first all possible transitions are developed from one condition. An example for this is given in Fig. 19. In the case of the search "in depth" first a solution path is followed to a defined limit depth in the graph. Where no solution is found until then, the search is continued "further above" and again developed "in depth". With the two above named strategies paths through the graph are developed of equal length or equal cost. The parameter $c(n_i, n_j)$ designates the costs for the transition from one point $n_i$ to a point $n_j$ . The designation $g(n)$ is chosen for the cost of the total path from beginning point s to the point n.

With respect to the interactive solution of problems in man-machine systems the heuristic search methods have more significance than the exhaustive procedures. Heuristic information is employed in setting priorities with respect to the order in which points are to be developed. Heuristic information concerns special assumptions on the problem to be solved, suitable for simplifying the solution search,

---

1) Exhaustive = complete

2) Heuristic = "serving the discovery"

including intuitive assumptions, rules of thumb, generally valid principles and plausibility conclusions.

The heuristic search for a solution therefore represents a combination of systematic and intuitive procedures. That point of the graph, leading with the greatest probability to the target point, is developed first. The various points which may be selected are arranged in a rank series by means of an evaluation function.

The evaluation function f(n) indicates the costs of a path of minimum cost leading through point n. The costs consist of two components:

$$f(n) = g(n) + h(n)$$

g(n) = Cost of the path of optimal cost from starting point to point n. This value may be determined from the graph developed to point n.

h(n) = "heuristic function." Indicates the costs of path of optimal cost from point n to the target point. This value is estimated on the basis of heuristic assumptions.

In the course of an ordered search the point of the graph with the lowest f(n) is developed first in each case. As an example for this procedure the 8 piece puzzle in Fig. 19 is to be again taken into consideration. It is apparent that not all system phases represented there promise the same success, reach the target. A heuristic function h(n) is being looked for, expressing this situation. /42

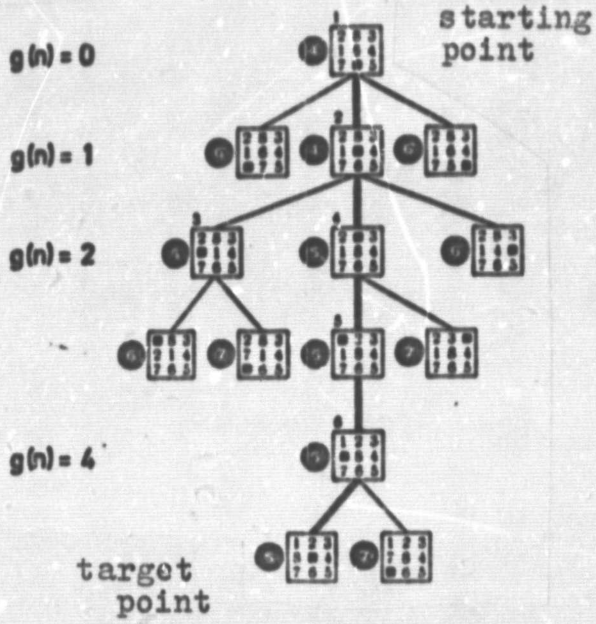
The simple heuristic function intuitively selected for this example under consideration of the game rules evaluates the number of falsely placed pieces at the point n. The value h(n) equals 4 is calculated for the beginning point (Fig. 19) using this assumption.

The costs of path g(n) are in addition included in the evaluation function to point n. In the case at hand the costs are set equal to number of transitions to point n. For the three points subsequent to starting point in Fig. 19 the evaluation functions 6,4,6 (see Fig. 20) are calculated with this rule. The point with the lowest f(n) is further developed in each case. It can be seen in the decision tree section in Fig. 20 that the heuristic has considerable reduced the search for a solution.

A further example for the same game, but with a complicated beginning situation, is represented is Fig. 21. The evaluation function is defined intuitively in this case as follows: /43

$$f(n) = g(n) + h(n) \text{ , where } h(n) = P(n) + 3S(n)$$

Here S(n) is the number of successive errors in game pieces, counted during a single circuit. S(n) increases by "2" for each case in which the piece with correct number does not follow. Chips in the middle count "1". P(n) indicates the sum of distances which




Evaluation function: f(n)=g(n)+h(n)

g(n)=path length to point n

h(n)=number of pieces with false position (costs of point n to target point)

Number example for point no. 2:

$$f(n) = 1+3 = 4$$

Fig. 20: Decision Tree for Puzzle with 8 Pieces Developed via "Ordered Search" [25]

each piece is removed from "its" location. The procedure is made clear by means of a number example for the following configuration:

S(n) = 15

| Game piece no. | Distance from desired position |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |
| 5 | 1 |
| 6 | 3 |
| 7 | 0 |
| 8 | 2 |
| Sum P(n) = | 11 |

h(n) = 11 + 3 x 15 = 56

Therefore it follows that $h(n) = P(n) + 3\ S(n) = 11 + 3 \times 15 = 56$

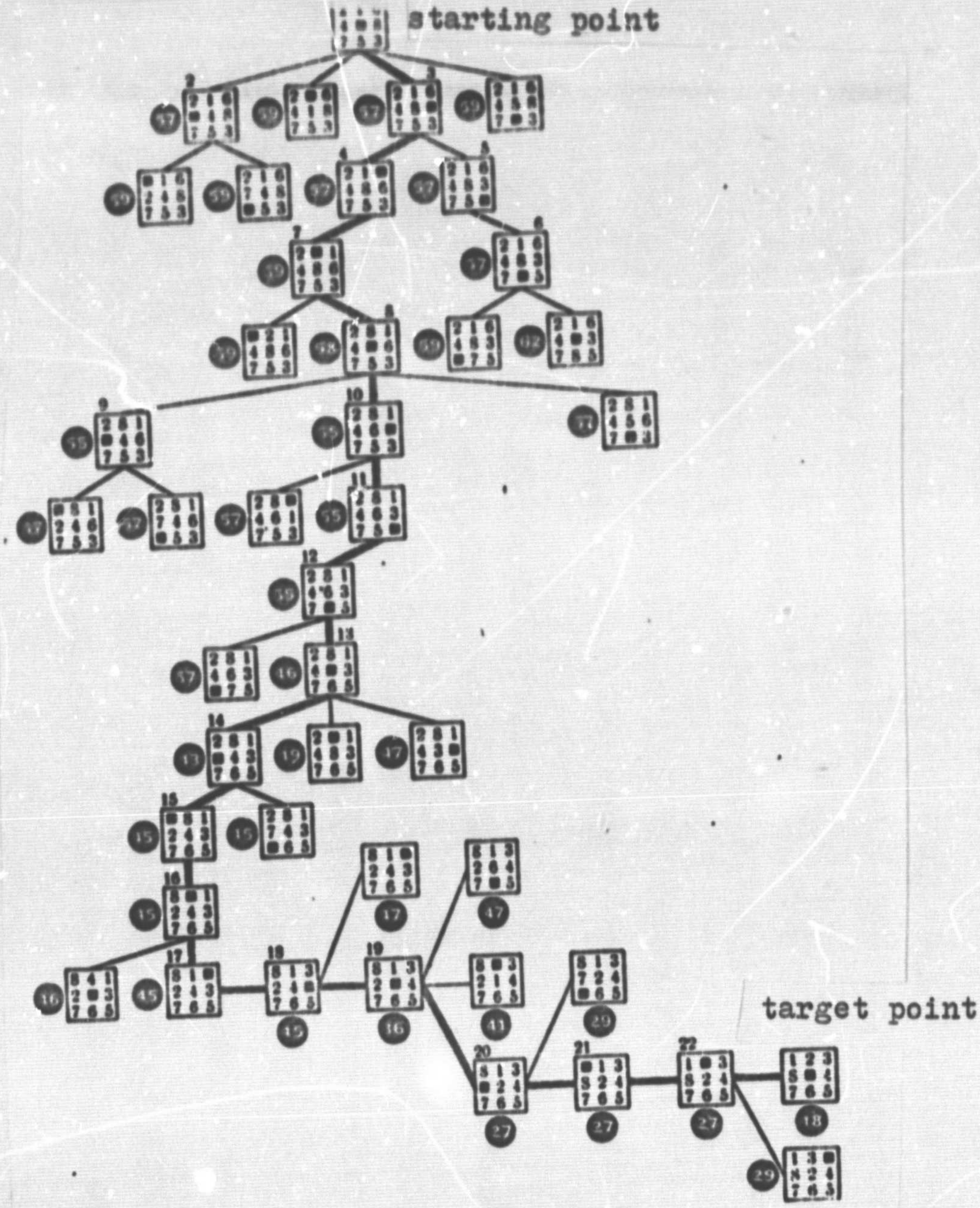



Fig. 21: Decision Tree developed by ordered Search with the Evaluation Function $f(n) = g(n) + P(n) + 3\,S(n)$ [25]

It is again clear here that the ordered search leads to a solution after a finite number of steps. There is, however, no assurance in this case that the solution path found has the lowest cost, as opposed to the blind, complete search. /44

The costs of a solution usually include in addition to the cost of solution path, the search costs. When these combined costs for method 1 are less than for method 2 it is said that method 1 has more "heuristic power". The "heuristic power" is determined by three factors: /45

- cost of solution path
- number of points developed
- amount of work in calculation of heuristic function $h(n)$

An attempt is made to deal with problems entailing high costs only in a few cases, so that in the case of routine tasks low costs are incurred. In many cases, however, calculation of combined costs is not possible, then only the intuitive choice of an heuristic approach remains.

The procedure of dynamic programming described may be employed for interactive solution of problems in man-machine systems in such a way that certain search strategies are put into the computer for which it then independently determines a suitable solution in each case. Solution alternatives and action consequences may be estimated in this way from the beginning and for various strategies. Decisive contribution of operator consists in the effective formulation of evaluation function so that desired strategy and solution approach are expressed correctly in this formulation. In the case in which e.g. any solution without consideration of cost is sought, $g(n) = 0$ may be inserted. In cases in which $h(n)$ is not adequately selected, the danger then exists that no solution will be found. When $g(n) \neq 0$, in time all parts of the graph (i.e. also in breadth) will be searched.

Dynamic programming for application in man-machine systems remains limited to such algorithms and problems which may be dealt with in approximately actual time.

### 5.1.3 Ship Navigation as Example for a Multiple Step Decision Process

In this section the solution for navigation task defined in Section 5.1 is carried out with the above explained method of dynamic programming. As a first step the possible paths of the own ship from beginning to target point are formulated, resulting in /46
a graph as in Fig. 22 if it is assumed that the ship moves forward in each section in a linear manner and at constant velocity. The point $x_n^i$ in the graph represents possible ship positions on the way to the target. On each of the N levels there are a finite number of possible positions $k_n$. The amount of possible system phases is therefore:

$$x_n = \{ x_n^1 \ldots\ldots\ldots x_n^{k_n} \} \text{ mit } n = 0\text{-}N$$

The geometry of the graph is determined by the consideration that deviation from correct course is to be limited where avoidance manoeuvers are undertaken. Further determining factors are natural obstacles and computer capacity. The maximum ship velocity is chosen under consideration of waves and currents, permitting calculation of expected arrival times for all points of the graph. Costs for phase transitions are calculated with transition time.

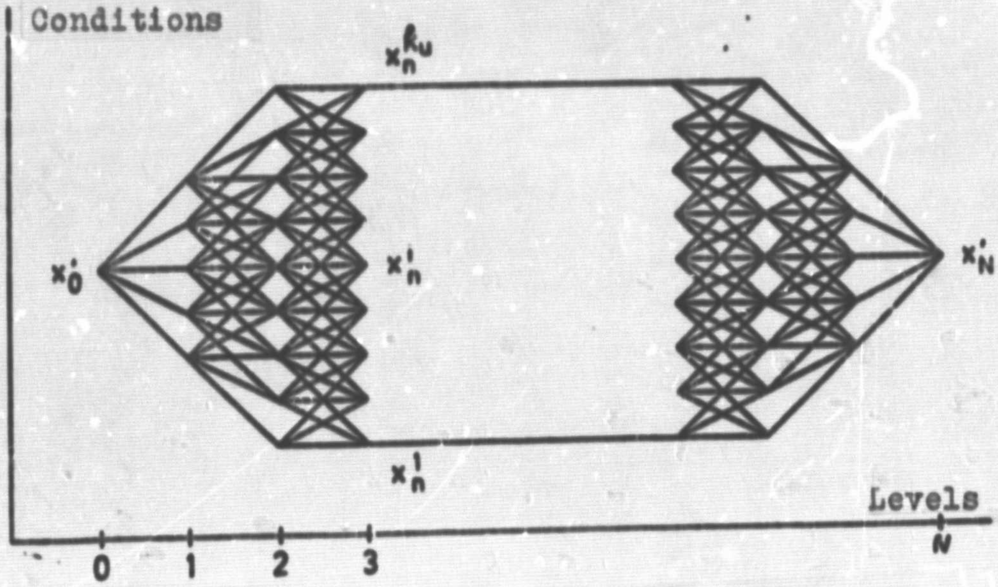



Fig. 22: Graph with N levels for representing all possible paths of the ship from the from the position of departure $x_0^1$ to the target $x_N^1$ [23]

Position and motion of other ships in the waterway ("targets") are known because of radar measurements and can be calculated for certain times. For simplification it will first be assumed here that the targets also move forward in the section in a linear manner as can be expected from merchant marine ships.

When considering Fig. 22 it is demonstrated that the graph is not a complete net. From a certain condition $x_n$ not all phases of the leven n+1 may be achieved. The plausible reason for this situation is again that the deviation from correct course cannot become randomly large. /47

For avoidance of a collision a minimum distance between ship and target must be maintained at all times. In order to define this distance the targets are observed periodically. Observation interval is in this case very much shorter than the transition between two points on the graph. On the basis of the last observation in each case a range is calculated for each target which can be achieved before the next observation is taken. If this calculation is made on the basis of motion equations of targets, permissible steering vectors $q(v, \delta)$ and measuring error of radar.

All non-permissible areas comprise a contour variable in time, the penetration of which could lead to a collision. The minimum distance ship/targets is calculated for all points and also for transition times between the points. Transitions not meeting the condition $\delta > \delta_{min}$ are disregarded. Note that after discovery of a collision danger in the transition to $x_n^1$ the entire path from start to that point may be eliminated. The original graph from Fig. 22 is greatly reduced by these calculations.

For all remaining paths without danger of collision the entire costs (here travel time) are now determined from start to the target and the fastest path is indicated as solution to the navigation problem. Figures 23a. and b. show trajectories determined in this manner. Fig. 23 b. demonstrates the recommended course correction becoming necessary when three targets alter their course.

Since there is no assurance that the simplifying assumptions about ship movements are accurate at all times, after each radar observation it must be examined whether deviation of the ship still remains within the secure time tolerance and whether the targets behave as predicted. Fig. 24 shows the program course for this periodic examination.

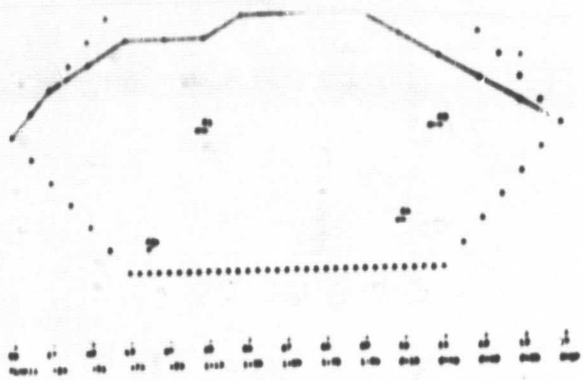


Fig. 23 a. : Example for Navigation optimal in Time in the Presence of 6 Ships [25]

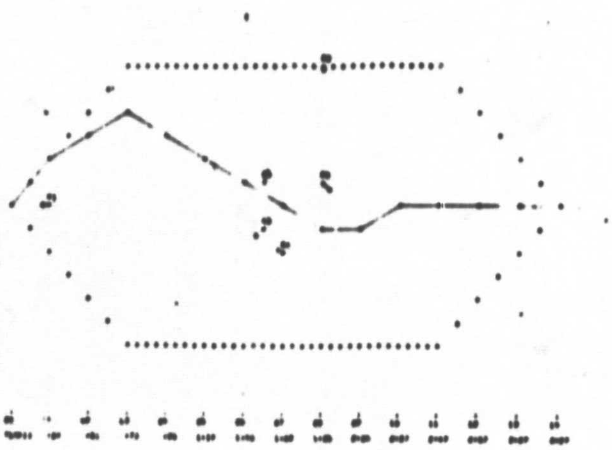


Fig. 23 b. : Altered optimal Strategy after 3 Ships have changed Course [25]

Note in this case that the operator is alarmed in the next /48
section only in the case of immediate collision danger. In cases of conflict lying further in the future the computer first attempts

to find a solution itself.

The navigation procedure described was developed for application in the Italian ship "Lloydiana". This system constructs a graph of 22 levels and takes into consideration up to 40 targets. The radar search interval was 2 - 3 minutes in this concrete case and calculation time was approx. 12 seconds.

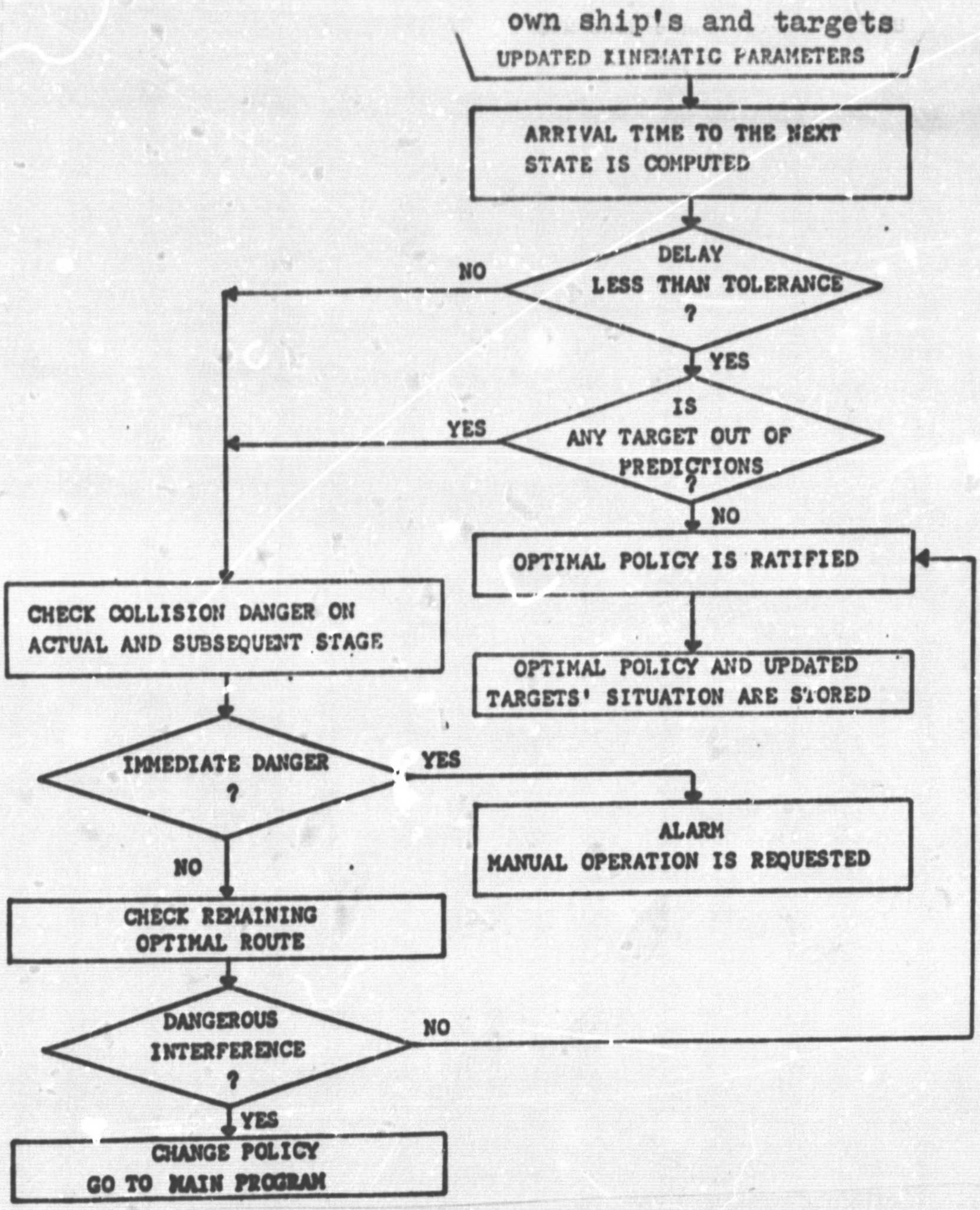



Fig. 24: Flow Diagram for Actual Time Optimation of Ship's Path [25]

Because the strategy may be changed, the risk altered and alarm threshhold varied in a simple manner, this procedure offers considerable advantages for the interactive application in man-machine systems. Other application possibilities are obvious. The computer-assisted optimation of air-traffic supervision and guidance should be mentioned. Work of air-traffic controllers could be substantially simplified if they receive suggestions from the computer on how holding loops could be avoided, which plane should accelerate for this and which should slow down etc. An example for this case is explained in the next section. /49

### 5.1.4 Computer-Assisted Air-Traffic Control /50

When a plane flies over land according to instrument rules, it is handed from one path control to the next. Task of air-traffic controller is to separate planes within his range of competency and to hold them on the desired course. Numerous boundary conditions have to be observed in this process. When a plane does not adhere to one of these boundary conditions, a conflict arises which must be resolved by corresponding directions from air-traffic controller to the pilot.

There are usually many various possibilities for solving the conflict. The quality of air-traffic controller work may then be estimated by the number and type of commands which he gives to the pilot via radio and by travel time of the planes.

This task is in times of heavy traffic very strenuous as is well known. It becomes especially difficult because expected events do not occur or not in the expected manner. Planes deviate from given course and height, do not maintain speed and new planes appear in the control range.

The experienced air-traffic controller has developed internal models permitting him to make predictions because of actual observed situations. His experience also puts him in a position to test hypothetical actions in a shortened time measure with his internal models. For the controller learning means in this connection reducing the number of surprises with which he is confronted in reality by means of continuous improvement of internal models.

It is possible with the method of dynamic heuristic programming to design a corresponding external decision model for assisting the controller. This model supplies action recommendations with exact time data on the basis of an heuristic calculation. The principle of implementation of this model is demonstrated with the following simplified example illustrated in Fig. 25 - on the basis of a traffic situation observed at 17:10 the further course of developments in the case of boundary conditions assumed unchanged is calculated to 17:30. /51 At 17:20 the picture given in the figure is received. Here radio fixed points (hexagons and + signs) and flight paths can be recognized. The broken lines mark the limit of an incoming control area. N111JD is a Lear Jet flying at 400 knots, SWAN-1 a military jet at 600 knots, N3953T a single motor Piper at 200 knots and AA-21 a Boeing 707 of American Airlines at 400 knots.

All machines are flying at a height of 11,000 feet.

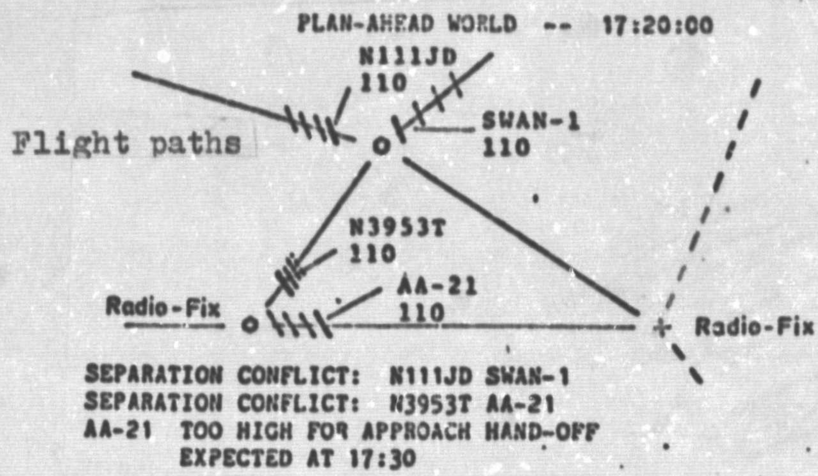



Fig. 25: An Air-Traffic Situation calculated 10 Minutes in the Future at 17:20:00 [26]

In the range of path control the following three situations may occur, requiring an action on the part of the controller:

a. Distance between planes i and j must longitudinally amount to at least 5 miles, otherwise a difference in height of at least 1,000 feet is required. This conflict may be resolve by the following alternatives -

- climb 1,000 feet
- reduce altitude by 1,000 feet
- a turn to the right
- a turn to the left
- reduce speed
- fly a holding pattern

b. When the next radio fixed point is an incoming control station, instructions on altitude may not exceed 6,000 feet. Corresponding instructions:

Reduce altitude to 6,000 feet or less.

c. Planes should not deviate from desired course by more than 4 miles. Possible instructions to pilot: /52

- Return to desired flight path
- Fly in the direction of the next radio fixed point if this is no further away than 40 miles
- First wait and give no instructions

It can be seen in Fig. 25 that there is a separation conflict at 17:20 between N111JD and SWAN-1 as well as between N3952T and AA-21. A conflict in altitude may further be predicted for AA-21 at 17:30, since this plane then approaches incoming control range.

Fig. 26 shows the solution of the separation conflict between Piper and Boeing 707 by means of a decision tree. Vertical lines signify here a course of events unaltered in time (vertical time axis), horizontal lines mark point in time and the type of alteration examined.

It can first be seen in the middle of the figure that the separation conflict 1 is determined at 17:20. For solution of this conflict all logical actions are played through on the decision tree, evaluating the final points again in the manner already explained.

For example a possibility consists in reducing Piper speed from 200 knots to 150 knots. This action would have to be taken already at 17:10. The same effect may be achieved, when the Boeing 707 reduces speed to 300 knots at 17:12. Both solutions, however, receive a low evaluation and are rejected. The complete development of decision tree supplies finally the optimal solution, setting the Piper on course 270 at 17:17.

A similar treatment of the other two conflict cases finally supplies the following computer suggestion:

17:17 N3953T Course correction to the right 45 degrees
17:19 N111JD Reduce altitude to 10,000 feet
17:24 AA-21 Reduce altitude to 5,000 feet
17:25 N111JD Reduce altitude to 6,000 feet
17:28 N3953T Course correction to the left 45 degrees.

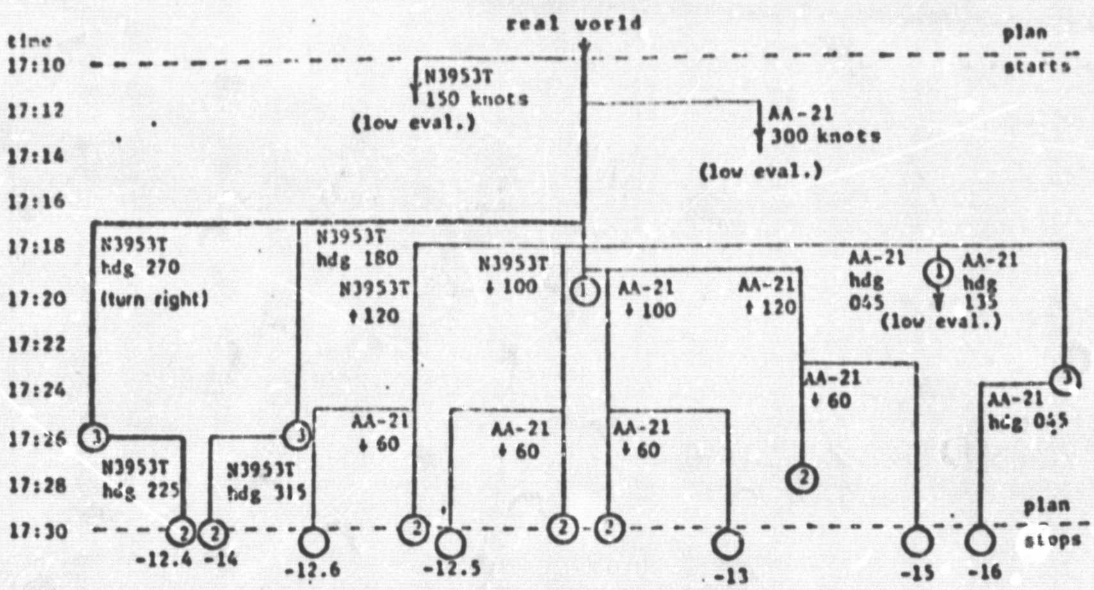



Fig. 26: Decision Tree for Avoidance of Separation Conflict between Plane N3953T and Plance AA-21 [26] 1)

---

1) The numerical values at the final points have been determined here by means of a function not explained in detail here.

This suggestion may be employed by the air-traffic controller for examining his own decisions. A direct transmission to the pilot is also possible. /53

A much more complex system of this type was studied experimentally at the University of Texas, comparing especially performance with that of air-traffic controllers. The computer model made better decisions in 16 of 19 cases and in no case worse than the controllers. In an especially difficult situation the computer required only 15 instructions for the pilot while the controller needed 40.

It may be concluded that such systems could be advantageously applied to practical situations. Large problems, however, above all of a psychological nature, are to be expected for the pilots, air-traffic controllers and general public. It still must be insured technically that the computer recognizes in time when it can not handle a problem and gives an alarm signal to air-traffic controller, before it is too late.

### 5.1.5 Interactive Problem Solving via Problem Reduction /54

In many case the problem to be solved permits so many variations that a complete decision tree could not be developed and searched. Determination of solution by means of complete permutation either takes too long or costs too much. Sometimes the task may be solved via division into partial problems (problem reduction). In this process the operator develops solution plans intuitively and delegates to the computer the solution of such partial problems as he deems necessary.

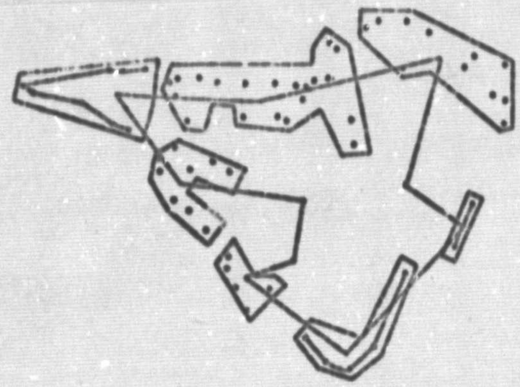
While fully automatic systems confronted with unsolvable tasks often react in an unexpected way, the operator retains his grasp of the solution path. The user controls the actions of the computer interactively so that the heuristic capabilities of the man and the algorithmic advantages of the computer complement in an optimal manner.

Tasks especially suitable for this type of problem reduction may be represented graphically. In such cases the unsurpassed capability of man to classify patterns is put to use.

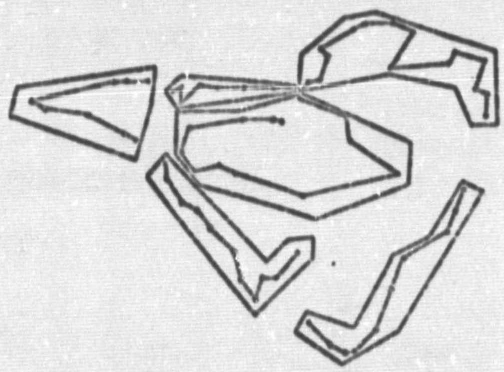
As an example an extensive problem of the type "Travelling Salesman" is considered. The task consists in visiting 68 cities one after the other in such a way that the total travel distance is kept to a minimum. Optimal solutions of this task cannot be found in a reasonable amount of time employing decision trees. Solutions applying heuristic approaches lead to solutions not completely optimal and take up approx. 22 minutes of computer time.

For the solution of this task by means of problem reduction the geographical position of individual cities is first displayed on a screen. The operator develops intuitive solution plans after observing the situation. A first plan consists in compiling groups of points to "supercities", drawing polygons around groups of cities with a light stylus, corresponding to the pattern classification promising the greatest success to the operator. The sub-problem arising in connecting the super-cities in an optimal manner is delegated to the computer (Fig. 27 a.).

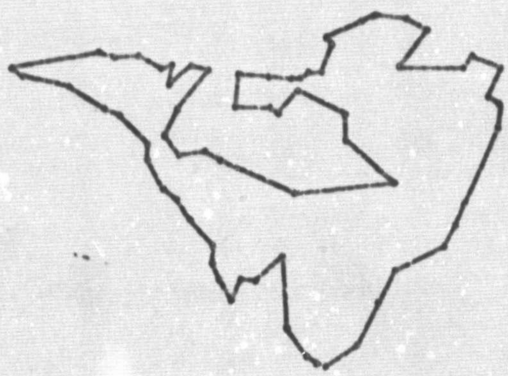
After observation of point distribution a second possible approach consists in determining an optimal inner and outer travel route which are then connected to form a total solution (Fig. 27 b.). /55



a.) Solution Plan 1: Compilation of Adjacent Cities to Super-Cities



b.) Solution Plan 2: Division of an Inner and Outer Route



c.) Optimal Solution found.

Fig. 27: Demonstration of Interactive Problem Reduction which may be represented graphically [27]

The solution found in this manner (Fig. 27 c.) is better than that found by means of the above-mentioned heuristic approach. It takes only 6 minutes of computer time, occupying an operator, however, for approx. 85 minutes. Applications of the interactive problem reduction procedure explained for problems represented graphically are possible among other in navigation, distribution of goods, traffic problems, searching for mistakes. /56

## 5.2 Two Person Games

As compared to the game against nature one's own actions in a duel situation have to be calculated for the most unfavorable consequence selected by the opponent instead of one of several possible consequences. Examples for this process evasive action, target aiming, tactical situations and many other tasks occurring in guidance and control systems.

Games may be divided into numerous categories [6,28]. Only methods for determining optimal strategy in two-person games using complete information are considered here. This game type is characterized by each player's knowledge of the game rules and the possible moves of the opponent (examples are checkers, chess, tic tac toe and go).

A suitable formulation for representing a series of one's own and the opponent's moves are AND/OR graphs which may be considered an extension of the OR graphs (decision trees) introduced in Section 5.1.1). The design and construction rules are expressed in Fig. 28:

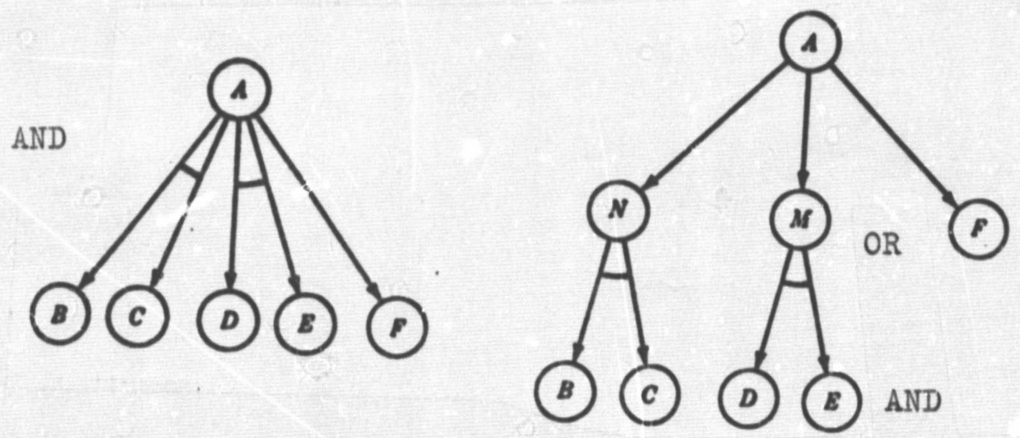



Fig. 28: Design of AND/OR Graphs [25]

The left-hand figure symbolizes a problem A with a number of alternative sub-problems. The figure demonstrates that the problem A is solved when either the sub-problems B <u>and</u> C or D <u>and</u> E or F are solved. N,M and F on the right-hand side are OR points (alternative sub-problems). At least one of these must be solved. B and C in the right-hand figure are again AND points. These are marked by a connecting arch. /57

For the representation of a game situation in an AND/OR graph the following notations are used for the two players:

PLUS (+)

MINUS (-)

X Game configuration

$X^+$ PLUS moves first in this configuration

$W(X^+)$ PLUS can win on the basis of situation X

The OR points represent the moves PLUS (the player selects a move from several alternatives). The AND points represent the moves of MINUS (opponent). The opponent may make many moves and the consequences of all these moves must be examined.

In the representation of the game OR (one's own moves) and AND points (opponent's moves) alternate with one another. The task now consists in search the AND/OR graph for a sequence of moves for PLUS, insuring that the game may be won from this starting situation or may at least remain a tie.

Blind search techniques seldom lead to a solution, because the graphs for many games become to large. The complete representation of a checker would have, e.g., $10^{40}$ points. One alternative consists again in limiting the design of graphs and the search for a solution with the aid of heuristic assumptions. This procedure does not guarantee finding the best move, but at least a good first move is found. Then the reaction of the opponent is considered and a good move is then determined on the basis of the new situation. AND/OR graphs demand another type of heuristic search technique than OR graphs. One of these procedures is described in the following section.

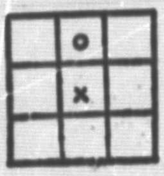
### 5.2.1 Determination of a good first Move with the MINIMAX Method /58

The MINIMAX method for searching game trees is based on the consideration of the most unfavorable reaction of opponent in the choice of a move. That move is made which results in the relatively smallest of all possible losses. The maximum expected loss is minimized.

The method is demonstrated with the aid of a tic tac toe game. Here three marks are to be made in a row - horizontally, vertically or diagonally - in a field of 3x3 fields. The player who first achieves this wins.

First a game tree for a move of player PLUS (marked by x) and an opponent move of the player MINUS (marked by o) is developed. Symmetrical moves are disregarded and for the game situations resulting at the end points a value e(p) is determined on the basis of heuristic assumptions. In the example described here the following rules apply for the formation of evaluation function e(p): e(p) evaluates the number of still open rows and columns for PLUS in a certain game situation subtracting the still open rows and columns for MINUS. An example:

Game Situation            Evaluation



$$e(p) = 6 - 4 = 2$$

e(p) is set at $-\infty$ when for the next move a victory for MINUS results and $+\infty$ in the opposite case.

The course of the game resulting via this heuristic evaluation may be followed for three moves of PLUS in Figures 29 and 30. The search depth after a good move amounts in each case to 2, i.e. one move and the next move of the opponent.

Search depth = 2

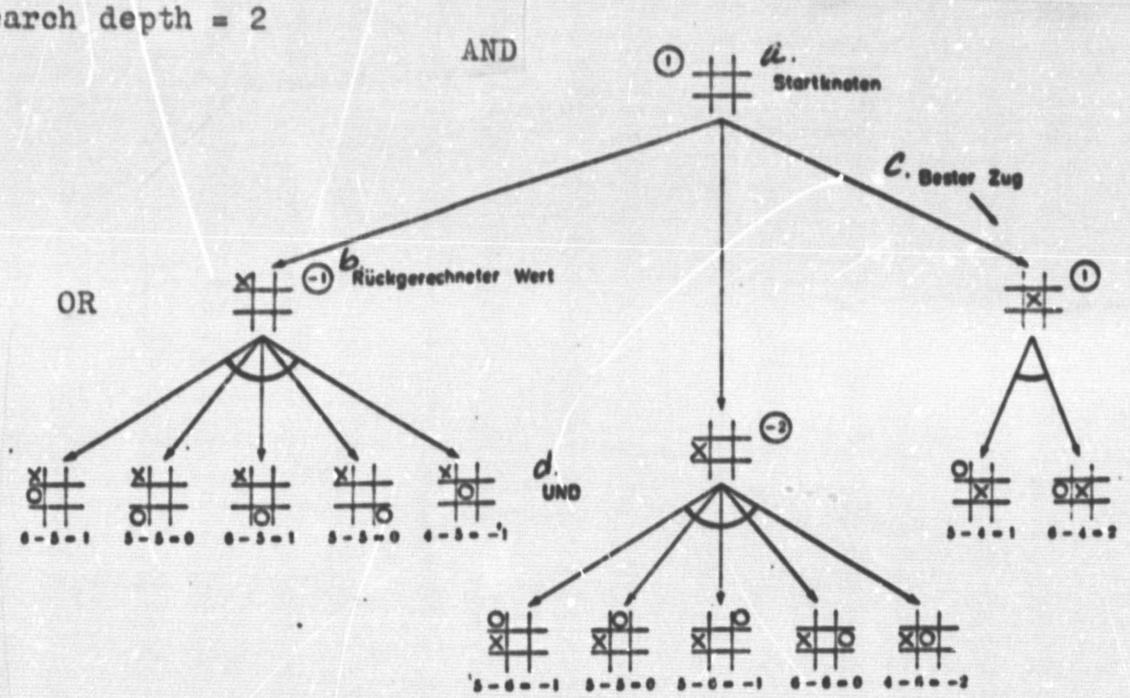



Fig. 29: Demonstration of MINIMAX Procedure using the Example of a Tic Tac Toe Game [25]

Key: a. Starting point            c. Best move
b. Value calculated from results     d. AND

The evaluation function takes into consideration the value of one's own move as well as the (negative) moves of the opponent. e(p) is smallest when the opponent reacts in the most unfavorable manner. One's own move is selected by means of the relatively greatest value calculated from the final results (greatest minimum). /59

It can be seen from Figures 29 and 30 that the graph was developed completely, i.e. with all branches to the limit depth given,

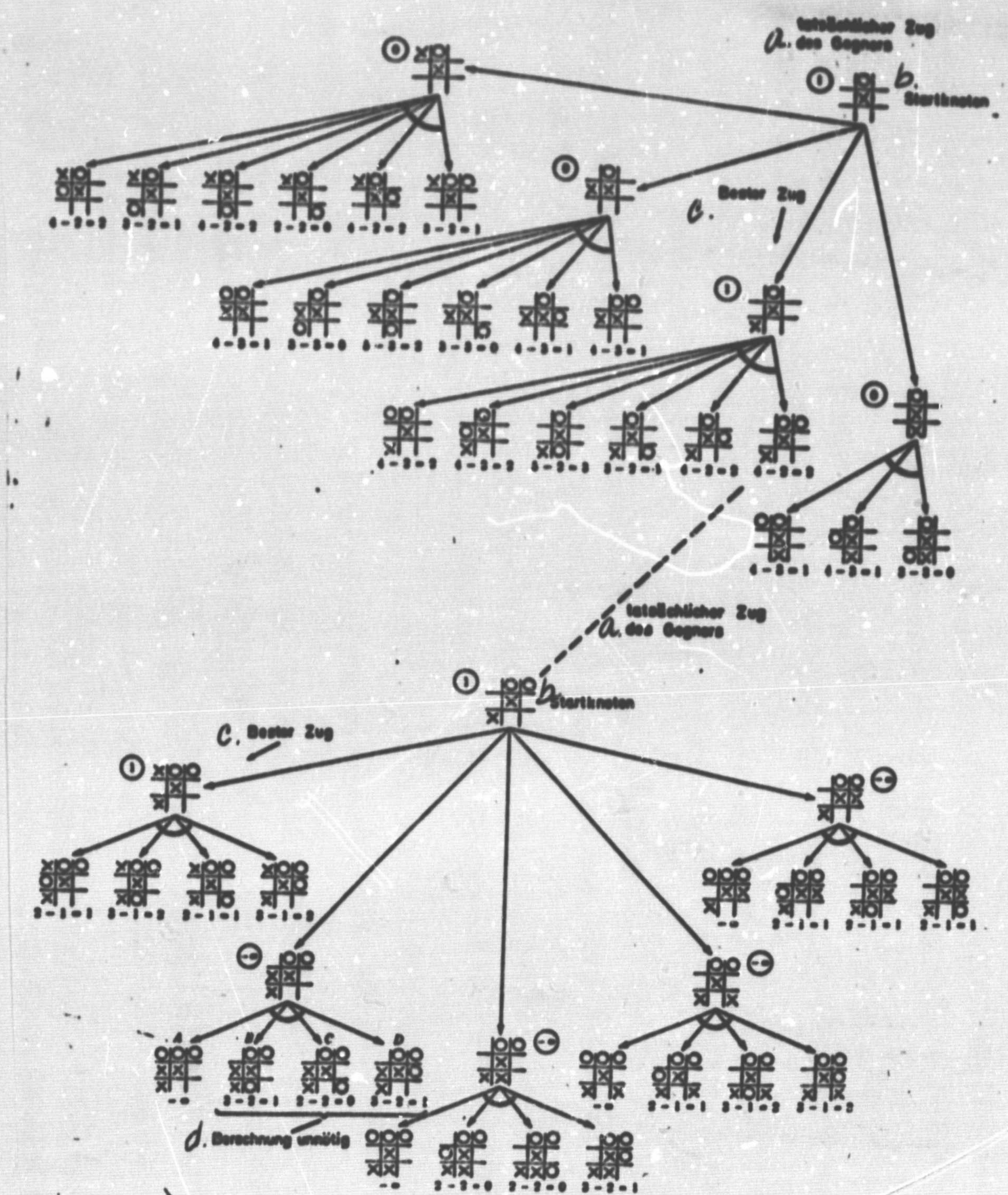



Fig. 30: Demonstration of MINIMAX Procedure using the Example of a Tic Tac Toe Game [25]

Key: a. Actual move of opponent
b. Starting Point
c. Best Move
d. Calculation unnecessary

before the evaluation of final points and the search for a solution begins. The work can be considerably reduced by calculating the evaluation during development of the graph and then using these calculations for control of further development of the graph. Fig. 31 shows this modified procedure. Differences become clear in a comparison with Fig. 29. Point B does not need to be developed further because the first opponent move examined supplies a value "-1", not better than the already present point A. Further moves after point B would either supply better values, not taken into consideration /61
corresponding to MINIMAX method, or worse values leading back again to point A.

In English literature the designation ALPHA values is employed for evaluation of AND points and BETA values for the values calculated from final results of the OR points. This is therefore also termed an ALPHA-BETA graph reduction.

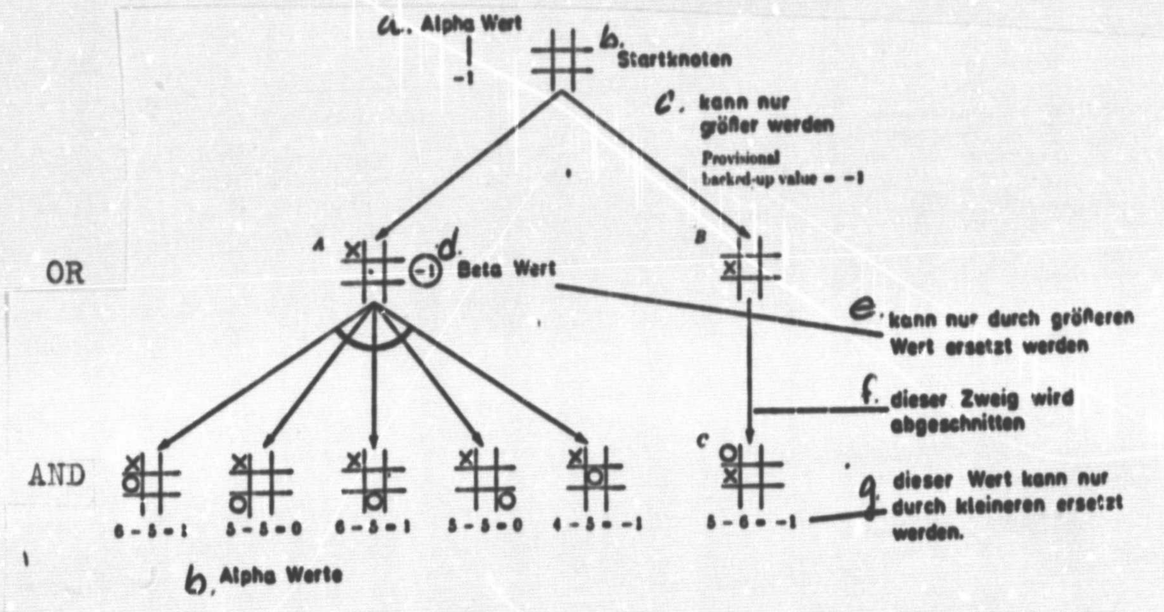



Fig. 31: Demonstration of ALPHA-BETA Development of a Game Tree (Nilsson 71)
ALPHA Values may only be replaced by greater values, BETA Values only be lower values [25]

Key:
a. Alpha value
b. Starting point
c. may only become greater
d. Beta value
e. may only be replaced by a greater value
f. this branch is cut off
g. this value may only be replaced by smaller ones

Fig. 32 provides an opportunity for following the ALPHA-BETA search procedure in the case of a more complex example. The graph shown is developed to a depth of three moves and the value for individual final points is indicated. The bolder type branches are those which are to be searched when applying the ALPHA-BETA procedure.

Searching the thinly drawn branches is disregarded.

The examples explained here were kept intentionally simple and clear. Ordinarily substantially more complex approaches are necessary for evaluation functions. It may even occur that several moves of equal value are determined. For the selection of a certain moves /62
additional criteria then become necessary.

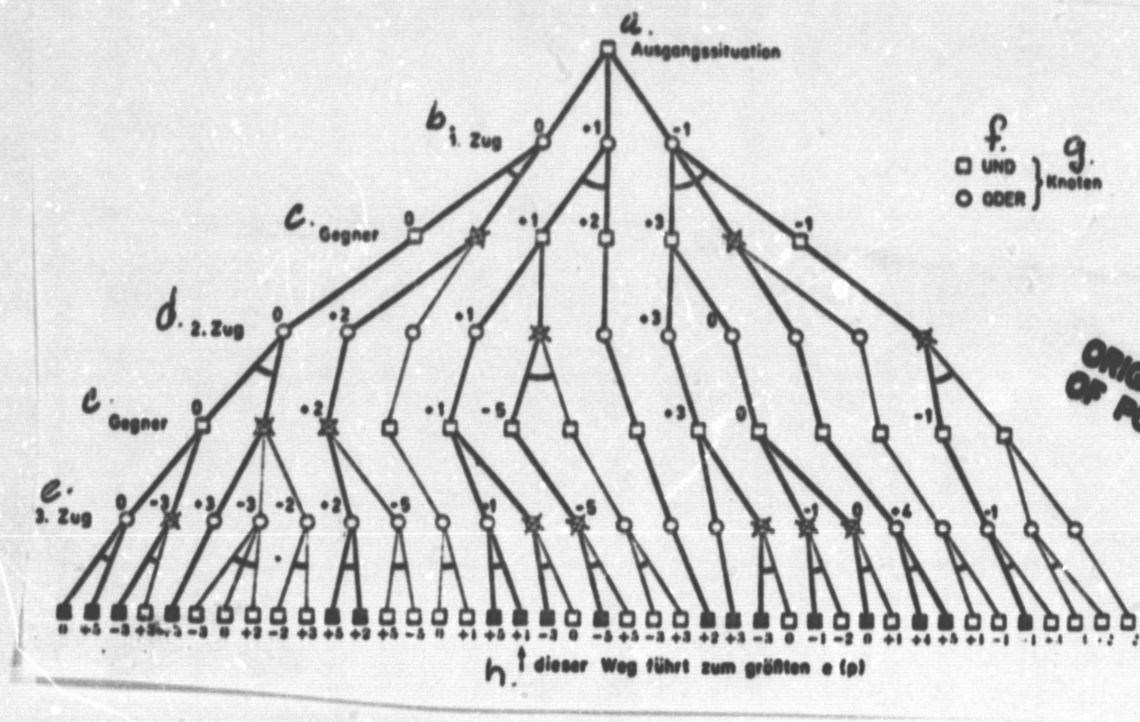





Fig. 32: Example for Illustrating the ALPHA-BETA Search Procedure (Nilsson 11 )

Key: a. Starting situation
b. First Move
c. Opponent
d. Second Move
e. Third Move
f. AND
OR
g. Points
g. This path leads to the largest e(p)

A further remark: when both players search "equally deep" in the graph, the opponent then employs one level deeper than PLUS in predicting his move in each case. MINUS as a result also plays another "worst expected" move than that calculated by PLUS.

## 5.2 Procedure for the Problem "Missile Defense" in the Form of a Two-Person Game

In this section the problem of warding off a missile attack is interpreted as a two-person game. Actions of opponent and one's own defense measures are represented in a graph and the adaptive decision aids for defender are derived from this in actual time.

The tactical situation may be described as follows [29]:

Missile silos $S_i$ are protected by interceptor rockets $I_i$. Each interceptor rocket protects several silos dependent on their positions. /63
The silos are attacked by the opponents rockets $R_i$ flying in succession. Because of the radar range only one attacker after the other may be discovered. Defense task consists in applying the available interceptor rockets in the most effective manner possible for protection of the silos. The concrete, greatly simplified game situation considered here as an example is shown in Fig. 33:

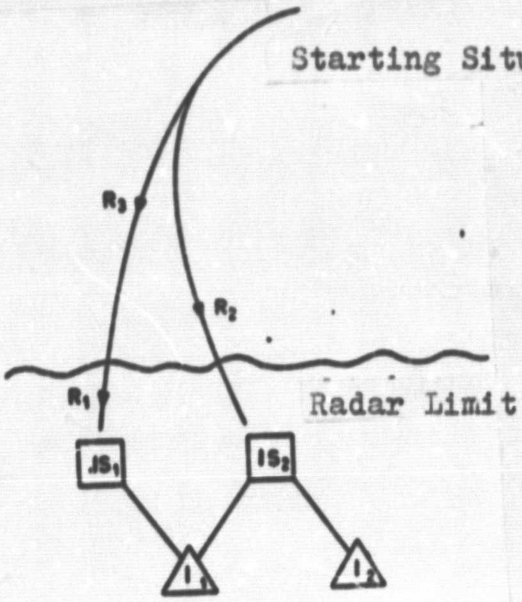



Fig. 33: Game Situation "Rocket Interception" [29]

Possible moves of the opponent consist in attacking either silo $S_1$ or silo $S_2$. The defense may react to this by defending the silo under attack with an interceptor rocket or surrendoring the silo, i.e. saving the rockets until further attacks occur (silos which have been destroyed require no further defense).

It is now assumed for further development of the example that an attack on $S_1$ is discovered from radar observation (first move of the opponent). Possible reactions are either to destroy $R_1$ (assumed hit probability = 1) or to do nothing (x), i.e. to surrender $S_1$. In order to determine the "correct" move of these two, a graph is developed with all possible future opponent reactions and defense measures. Fig. 34 shows this game tree for three successive attack and defense actions.

As the next step the final points of the graph are evaluated heuristically. In formulating the heuristic function the following parameters may be taken into consideration: /64

- S Number of remaining silos
- R Estimated number of remaining attack rockets

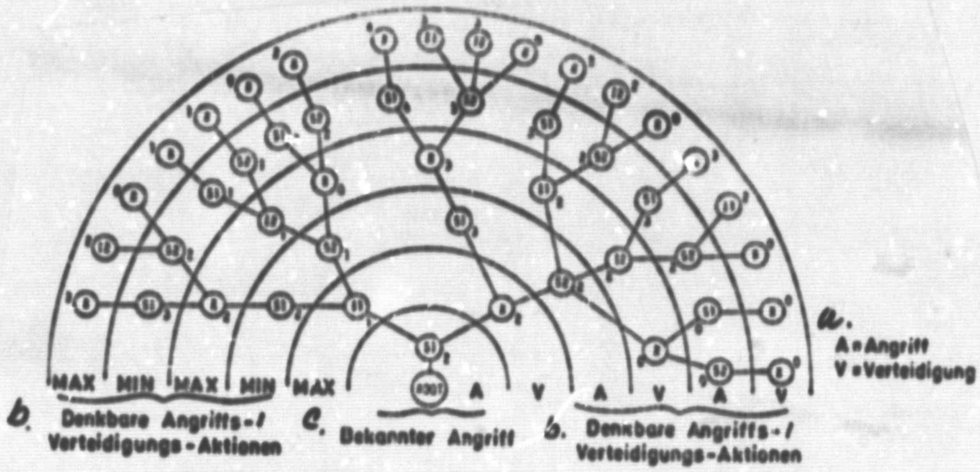



Fig. 34: Graph for three successive Attack and Defense Actions [29]

Key: a. A=attack
V=defense
b. Possible Attack/Defense Actions
c. Known Attack

- $I$ Number of remaining interceptor rockets
- $C$ Total protection of all interceptor rockets (sum of the number of silos covered by each I, redundancies are also counted)
- $S_T$ Sum of the silos at the beginning
- $R_T$ Number of opponent rockets at the beginning (estimated)
- $k$ Portion of silos to be saved ($k = 0 \ldots 1$).

Here are several examples for expressing different defense strategies in heuristic function:

$h = S$ ; Saving the silos is more important than all other considerations.

$h = 2S+C$ ; Saving the silos is more important than the remaining entire protection.

$h = ISC/R$ ; Number of silos saved, remaining interceptor rockets, remaining protection and remaining attack rockets are taken equally into consideration. /65

$h = \dfrac{R_T(S-kS_T)}{S_T(1-k)}$ ; k.S silos should remain after the battle.

For the present example $h = 2S+C$ is selected, resulting in the starting situation shown in Fig. 33, $h = 2 \times 2 + 3 = 7$.

For determination of a "good first move" the final points of the graph in Fig. 34 are now evaluated. With these values the optimal defense move may be determined according to the MINIMAX method explained in the previous section and by means of application of Alpha-Beta reduction. In the present case the largest value results for the alternative x (do nothing, surrender silo) with h = 2. Employing the heuristic function (strategy) selected the recommendation is therefore presented to surrender $S_1$.

In the meantime radar has registered an attack on $S_2$ (second opponent move). A decision tree is developed from the beginning situation given now to a depth defined by computer time and capacity (contained in Fig. 34). As defense move an interception is now recommended - whether with $I_1$ or $I_2$. This procedure may be continued in this manner.

Experimental studies show that the same defense effectivity is achieved with only 60 % of the interceptor rockets otherwise required, when applying this decision aid. It is assumed in addition that the opponent is always as well informed as the defender. This is not normally the case, so that in an actual situation even better results may be expected [34].

The most important advantage of decision aid described is that the computer suggestions are adjusted adaptively to changing surrounding conditions. Information gathered during the duel and council of experts as well as a change in strategy may be continuously taken into account in formation of heuristic function. By this means the operator receives a valuable aid in decision-making situations in which he would otherwise be dependent on intuitive solutions.

## 6. Final Remarks /66

In this report various possibilities were studied for reducing the cognitive stress of operator in complex man-machine systems via application of computers for intelligence assistance. Nickerson [30] states already in 1977,

> "Some system builders have envisioned systems that will have the ability to develop very sophisticated models of their users. Such systems would presumably be able to serve as extensions of their users' memories, to implement their problem-solving and decision-making capabilities, and generally to augment their cognitive performance in a variety of ways. But this is speculation ...".

This statement seems somewhat too conservative. The examples employed in this report underline the fact that the development in this area has already led to results which are no longer designated as speculation, although this compilation is in no way complete. The examples were also greatly simplified intentionally. There are a number of other approaches in the area of artificial intelligence with possible application for this research not taken into consideration at all here.

It must, however, be recognized that the development of an "intelligent partner relationship" between man and computer is just beginning. In addition it must be stated that this approach probably be employed only for exactly defined cognitive tasks.

As (the only?) alternative to the increasing automation in complex technical systems this approach, however, deserves considerably more attention than it has received up to the present. The possibility for substantially increasing cognitive performance of man without increasing stress and without turning to full automation is opening here.

## REFERENCES


[1] Sackmann, H., "Outlook for Man-Computer Symbiosis," Toward A General Theory of Man-Computer Problem Solving, Advanced Study Institute on Man-Computer Interaction, Mati Greece, 1976.

[2] Nickerson, R.S. and C.E. Feehrer, "Decision Making and Training: A Review of Theoretical and Empirical Studies of Decision Making and their Implications for the Training of Decision Makers," Bolt Beranek and Newman, Tech. Report NAVTRAEQUIPCEN 73-C-0128-1, July 1975.

[3] Firschein, O. et al., "Intelligent Machines are on the Way," IEEE Spectrum, 41-48 (July 1974).

[4] Voysey, H., "Problems of mingling Men and Machines," New Scientist, 416-417 (18 August 1977).

[5] Rouse, W.B., "Design of Man-Computer Interfaces for On-Line Interactive Systems-Proc. of the IEEE," Vol. 63/No. 6 (June 1975).

[6] Sheridan, T.B. and W.R. Ferrell, Man-Machine Systems, MIT Press, 1974.

[7] Edwards, W., "Emerging technologies for making decisions," New Directions in Psychology II, Holt, Rinehart and Winston Inc. (January 1967)

[8] Edwards, W.A. and A. Tversky (eds.), Decision Making, Penguin Books, Baltimore, 1967.

[9] Kaufmann, A., The Science of Decision Making, World University Library, McGraw Hill, 1968 (reprinted 1974).

[10] Yntema, D.B. and Klem, L., "Telling a Computer How to Evaluate Multidimensional Situations," IEEE Transactions on HFE-6:3-13, 1965.

[11] Nilsson, N.J., Learning Machines, McGraw Hill Book Comp., 1965.

[12] Young, T.Y. and T.W. Calvert, Classification, Estimation and Pattern Recognition, American Elsevier Publishing Company, Inc., 1974.

[13] Duda, R.O. and P.E. Hart, Pattern Classification and Scene Analyses, John Wiley & Sons Inc., 1973.

[14] Fu, King-Sun, "Learning Control Systems - Review and Outlook," IEEE Transactions on Automatic Control, 210-221, April 1970.

[15] Slagle, J.R., Artificial Intelligence: The Heuristic Programming Approach, McGraw-Hill, New York, 1972.

[16] Steeb, R., G. Weltman and A. Freedy, Man/Machine Interaction in Adaptive Computer-Aided Control, Perceptronics, Woodland Hill, Ca. 91364, 1977.

[17] Weisbrod, R.L., K.B. Davis and A. Freedy, "Adaptive Utility Assessment in Dynamic Decision Processes: An experimental Evaluation of Decision Aiding," IEEE Trans. SMC-7, No. 5, May 1977.

[18] Freedy, A. etal., "Adaptive Computer Aiding in Dynamic Decision Processes: Methodology, Evaluation and Application," Final Report to Arpa Order Nr. 2347, Perceptronics, Woodland Hill, Ca. 91364, August 1977.

[19] Freedy, A., R. Weisbrod and G. Weltman, "Self-Optimization of Task Allocation in Shared Man/Computer Control," Proc. of the Conference on Decision and Control, S. Diego, Dec. 5-7, 1977.

[20] Crooks, W.H. et al., "Man/Machine Interaction in Adaptive Computer-Aided Control; Analyses of Automated Control Allocation," Perceptronics, Report PATR-1008-74-12/30, Dec. 30,1974.

[21] Weltman, G. and A. Freedy, "Functional Allocation in Adaptively Aided Decision Making and Control," Proc. of the 12th DRG Seminar on the Optimum Balance between Man and Machine in Man-Machine Systems, Utrecht NL, May 1973.

[22] Kaplan, R.J. and J.R. Newman, "Studies in Probalisitc Information Processing," IEEE Trans. Human Factors in Electronics, HFE-7, 49-63, 1966.

[23] Tiano, A., P. Dagnino and M. Piattelli, "Determination of an Optimal Trajectory in the Presence of Risk, in Automation in Manned Aerospace Systems," AGARD-CP-114, p. 5: 1-19, October 1972.

[24] Wentzel, J.S., Elements of dynamic Programming, Oldenbourg Verlag, Munich and Vienna, 1966.

[25] Nilsson, N.J., Problem Solving Methods in Artificial Intelligence, Computer Science Series, McGraw-Hill Book Company, 1972.

[26] Wesson, R.B., "Planning in the World of the Air-Traffic Controller," 5th International Joint Conference on Artificial Intelligence, MIT Boston, 473-479, August 1977.

[27] Howden, W.E., "Solution Plans and Interactive Problem Solving," Comput. & Graphics, Vol. 1, Great Britain, 1975, pp. 21-26.

[28] Owen, G., Game Theory (Spieltheorie), Textbook Series, Springer Verlag, 1971.

[29] Leal, A., "Adaptive Decision in Ballistic Missile Defense," IEEE Trans. on SMC, pp. 398-403, May 77.

[30] Nickerson, R.S., "On Conversational Interaction with Computers," Bolt Beranek and Newman Inc. Report No. 3499, January 1977.