FUNDAMENTAL DISCUSSION OF HEURISTIC FUNCTION AND ITS INTRODUCTION INTO LEARNING CONTROL

MORIYA ODA

Automatic Control Laboratry

(Received May 30, 1968)

1. Introduction

Heuristic function as well as *learning function* is one of the important intelligent functions evolved by a human being. There are, however, left many unsolved problems as to its engineering meanings and the method of its utilization. This paper aims to clarify the meaning of heuristic function and to introduce the function into learning control.

Heuristic is defined in Webster's New Twentieth Century Dictionary as follows; "adjective—(to invent, discover)" helping to discover or learn: sometimes used to designate a method of education in which the pupil is trained to find out things for himself.

It was some ten years that Heuristics was first noticed in the fields of engineering and science from the general viewpoint of heuristic programs. In pedagogy, heuristics has been reinvestigated both at home and abroad under the term of "heuristic learning", "discovery learning", or "learning by discovery".

The practical necessity of heuristics arises in the following cases: The heuristic solutions will be necessary in problems where the conventional sequential or analytical methods are regarded as unable, or very difficult in a practical sense, to find out a correct algorithm from a huge amount of possibilities to be searched.

As a result of fundamental discussions[1)-5)] on heuritsics developed by the author and Dr. Nakamura, *heuristics* is divided into the two basic concepts of *heuristic element* and *heuristic system* (*method*). We will start with these two concepts to investigate the backgrounds of heuristics, its history, practical examples of its application, its relation to learning, and so on, and will try to settle some cues for introducing heuristics into the learning control.

2. Background of Heuristics

2.1. Fields Needing a Heuristic Program

First of all, we will survey the engineering fields which necessiate heuristics. The fields concerned with heuristic program (that is, programs which work on discovering solving methods for themselves) are listed in Table 1.

The common characteristic of the structure of such programs is not that a heuristic program consists of complete series of computing statements (algorithm) which guarantee a solution of some given task, but that it consists of both (1) much *characteristic information* which provides the characteristics of the task, and

TABLE 1. Examples of Heuristic Programmes

No.	Applied Field	Example
1	Solution of strategic games	Chess[6], Checkers[7)63], Tic-tac-toe[8], Go*, Shogi*, etc.
2	Establishment of Proofs	Proofs of Geometrical Problem[9)10], Symbolical Logic Equation[11], Group Theory[58], Algebraical Equation, etc.
3	Labyrinth Search	Searches of Maze[12], and Fleight Path[14]; Constructions of Traffic Map[13], and Pattern of Printed Network[15]
4	Solution of Integral Equation	Definite and Indefinite Integrations[44]
5	Establishment of Time-Tables	Time Tables of Lessons, Test[17], Train, etc.; Scheduling of a Flow Shop[18]
6	Determination of Important Terms in the Network	Self-Organization of Imcomplete, Nonlinear, Multi-threshold Network[19]
7	Solution of Optimum Control	Optimum Control for First and Second Order Systems[20)21]
8	Search of Optimum Point on Nonlinear Surface	Extraction of Heuristics evolved by Human Searchers[1)-5]

* A kind of Japanese game

(2) *logical rules* which make optimum the possibility of success at the branches on the way through a computing process[8].

As listed in Table 1, the research fields of heuristic programs cover not only (1) such a problem as finding the optimum path of connecting its starting point (premise) and its end point (conclusion), where both the premise and conclusion are given clearly (*e.g.* No. 2 in Table 1), but also (2) such a problem as finding both the optimum path and its end (or starting) point, where only the start (or end) point is given (*e.g.*, others except No. 2 in Table 1).

It may seem apparent that only No. 7 and 8 in Table 1 have any relation to control engineering. There are, however, many common phases among No. 1 through No. 8, in the details of their techniques. Actually, in the IFAC Symposium to be held at Yerevan, USSR, in Oct. 1968, the agendum of technical problems include a subject of heuristic programming concerning No. 1 through No. 3 in Table 1. This fact may be enough for us to understand a new trend of introducing heuristics into control engineering.

2.2. Pedagogical Background of Heuristics

In pedagogy, there has been concept of heuristic education since old times, in which the pupil is trained to re-discover for himself a truth, principle, theory, law, rule, and/or so on. Especially, since about 1960 the importance of *heuristic learning* (learning by discovery) has been re-noticed in USA, USSR[*1], Japan, and so on. This has the backgound of necessity that the pupil must learn to think, to create, and to discover things for himself instead of being trained in special techniques in the age of rapid progress of science and technology[23].

[*1] At the International Conference of Psychology held at Moscow in 1966, there were presented 22 papers in the session of "heuristics in thought"[22].

As a recent general trend in elementary school, junior and senior high schools (partly in some universities), almost all kinds of textbooks have been or will be reorganized[23].

An example of heuristic learning is illustrated in the learning of the Archimedes' principle: First the teacher presents such decomposed elements of curriculum as an iron ball, a beaker full of water, a balance, a measure cylindar, and then lets the pupil make a series of preassigned experiments. The pupil thinks according to the results of experiments, and is instructed to re-discover for himself Archimedes' principle which is implied (but not stated) by the facts observed by the pupil himself.

As for a definition of heuristic learning in pedagogy, there is not yet an established definition even in USA. This means that heuristics is generally accepted as a fairly wide concept due to its versatile characteristics embedded within. Here, however, two moderate definitions are cited as follows:

Definition 1. *Heuristic Learning* is the learning method which reorganizes pedagogically the process of original discovery into an easy way for the pupil of average intelligence[23].

Definition 2. *Heuristic Learning* is the learning method which is based upon the *processism* where scientific uderstanding is recognized as the end point of learning to be achieved, and which lets the pupil trace the process of reproduction and reorganization by the pupil's positive activities[24].

Prof. R. Hiro-oka pointed out from a macroscopic point of view that the process of heuristic learning can be classified into the following four levels[23]: (1) The level where a concrete fact is observed by consciousness of question, (2) the level where some assumption is conceived, (3) the level where the assumption is developed into a theory, and (4) the level where the theory is transferred to an actually applicable ability. Especially in the near future, the microscopic phases of learning process (*e.g.*, the processes of prediction and testing an assumption in the heuristic learning) must be researched in more detail.

2.3. History of Research of Heuristics

It is said that the father of heuristics might be the philosopher Socrates (469–399 BC) in Ancient Greece, or the mathematician Pappus (about 300 AC). Afterwards, there appeared two excellent mathematicians as well as philosophers, Rene Descartes (1596–1650) and Gottfried Wilhelm von Leibnitz (1646–1716), the mathematician as well as philosopher, Bernard Bolzano (1781–1848) who revived heuristics in a modern style. Recently, there are the physicist as well as philosopher Ernst Mach, the mathematicians G. Polya[26]–[28] and J. Hadamard[29], the psychologists M. Wertheimer[30], William James, Wolfgang Koehler, and so on. Especially, G. Polya is the person in the present age who has revived heuristics in "research of technical method of problem-solving"[26]–[28].

All the researches mentioned so far are concerned with the subject of human being. However, after the digital computer ENIAC was born in 1946, many engineers have joined the research of heuristics[31]–[33]. That is, there are two big ideas: one is that a human being himself makes a higher discovery and creation by fully using the function of computer[34], and the other is that the computer which is considered to be an artificial intelligence should be given a heuristic ability in order to make such higher intellectual activities. As examples for the latter case, M. Minsky[35][36] explained Heuristic in connection with Search, Learning, and Problem-solving; A. Feigenbaum and J. Feldman[37] edited the book "Computers and Thought" which had collected many important papers on heuristic reported by that time from the following points of view of (1) game played by machine, (2) computer solution of mathematical problem, (3) pattern recognition, (4) problem-solving, (5) learning, and so on; A. L. Charnyavskii[38] surveyed Heuristic Programming used in the process of solving complex logical problems. Afterwards, many papers included in the categories No. (1) through No. (5) described above have been published.

In our country, Japan, there are several papers, surveys, and/or books on heuristics by the pedagogists, Hiro-oka[23][24], Onda, *et al.*[60)-62)], and Abe (Education of Mathematics)[25][39]; the engineers, Iwaki[40], Sakai and Nagao[13], Chikatsuji[18], Nakano, *et al.*[12a][12b], Oda and Nakamura[1)-5)], and so on.

On the other hand, there are very few examples of introducing heuristics into control engineering. Research in this field awaits future developments.

3. Engineering Discussion on Heuristics

3.1. Engineering Necessity of Heuristics

Here, our discussion will be focused on the engineering necessity of heuristics.

Recently in the treatment of complicated and large-scale systems, there exist many problems whose solutions can not (or hardly) be approached by a conventional sequential method. Therefore, the application of heuristics is necessary for an approximate solution of such cases.

This can be said in the following way from the viewpoint of computer science: For the large-scale, complicated, and difficult class of computation (or thought) considered to be beyond the capability of a human programmer, the computer can not be given in advance a complete algorithm for a program. In such a case a heuristics (or heuristic programming) is necessary as a technique of programming which will teach the computer only the heuristics suitable to each problem, and the computer is left alone to achieve the task for itself.

3.2. Engineering Definition of Heuristics

There is as yet no established definition of heuristics in engineering as in pedagogy (cf. Section **2.2.**). However, the engineering definitions on heuristics cited below are fairly different from the pedagogical ones:

Definition 3. *Heuristic* is to determine a principle of restricted generalization whereby a given machine learns to apply a "generalized" from of a previously successful algorithm when it is subsequently presented with a "similar" problem[41].

Definition 4. *Heuristics* are principles or devices that contribute, on the average, to reduction of search in problem-solving activity[42][43].

Definition 5. A *heuristic method* (or simply a *heuristic*) is a method which helps in discovering a problem's solution by making plausible but fallible*2 guesses

*2 From the viewpoint of fundamental theory of mathematics, such a point as "not to guarantee a sure solution" is indicated as fatal weakness.

as to what is the best thing to do next[44].

Definition 6. *Heuristic* is to refer to things related to problem solving. In particular we tend to use the term in describing rules or principles which have not been shown to be universally correct*3 but which often seem to be of help[45].

3.3. Various Terms on Heuristics

Here, various terms on heuristics are listed as in Table 2[1]. Table 2 shows the versatile characteristics.

4. The Essence of Heuristics and the Relation between Heuristics and Learning

4.1. The Essence of Heuristics (Author's Opinion)

What on earth might be considered to be the essence of heuristics of such versatile characteristics? The author would like to propose the following definition:

Definition 7. A *heuristic method* is a method which solves a problem using a *heuristic algorithm* which utilizes a series of *heuristic elements* having a special order of *priority* of these elements.

As stated in the above difinition, heuristics is divided into two main concepts: (1) heuristic elements and (2) heuristic system (method, algorithm). Which of the two above concepts the term "heuristics" means has depended greatly upon each paper reported in the past[1] (cf. Table 2)*4. However, Definition 7 makes it clear that both concepts should be separated and recognized.

Definition 7 is illustrated in Figs. 1 and 2. In Fig. 1, the problem is to search a (an optimum*5) path to reach from a starting point (initial state, premise) to an end point (object, conclusion), where there are so many unknown branches on the way of any path that it is impractical to check the whole set of possible paths sequentially. In such a problem, a "*heuristic element*" means the branches a_{ij}'s ($i=1, 2, \ldots, j=1, 2, \ldots, J_i$) which have a high possibility of gaining a success*6 of reaching to an end (cf. in Fig. 1, $a_{11}=b_{13}$, $a_{12}=b_{18}, \ldots$, $a_{21}=b_{23}$, $a_{22}=b_{25}$, $a_{23}=b_{29}$, $\ldots$, $a_{31}=b_{32}$, $a_{32}=b_{33}, \ldots$). On the other hand, a "*heuristic method*" means a method of connecting a series of the branches, which can pass through from the starting point to the end point, under some proper priority of such branches with a high possibility of success.

In Fig. 1, the following problems must be considered according to real cases: What kinds of heuristic elements must be chosen for a practical case? What kind of priority system must be adopted? From which side must the connection

*3 From the viewpoint of fundamental theory of mathematics, such a point as "not to guarantee a sure solution" is indicated as fatal weakness.

*4 Generally speaking, it seems rather often that heuristics is accepted as stressing the phase of process of (2).

*5 When there are so many branches to be searched, it is an urgent necessity to seek out one of possible paths. It is another problem to select an optimum path in a lot of possible paths.

*6 The measure of success depends upon real problems.

TABLE 2. Examples of Terms on Heuristics

No.	Term		Main reference
1	heuristic	(adjective)	dictionary
		(noun)	〃
2	heuristics	(noun)	〃
3	heuristically	(adverb)	〃
	Usage as Noun		
4	additional heuristics		Gelernter, 9), p. 136
5	basic learning heuristic(s)		Slagle, 44), p. 197
			Minsky, Mesarovic, 46)
6	basic heuristic		Slagle, 44)
7	extended heuristic		Slagle, 44)
8	choice of heuristics		Gelernter, 9), p. 137
9	general learning heuristic		Mesarovic, 46)
10	general problem-solving heuristics		NSS*1, 47), p. 189
11	general-purpose heuristic		FF*2, 37)
12	gross heuristics		NSS, 47), p. 188
13	heuristic for learning		Mesarovic, 46)
14	heuristics for subproblem selection		
15	heuristic in a learning process		Mesarovic, 46), p. 392
16	human problem-solving heuristics		Minsky, 37), p. 470
17	intersection heuristic		Mesarovic, 46)
18	learning heuristic		Mesarovic, 46), p. 406
19	means-ends analysis heuristics		NSS, 47)
20	modern heuristic		Polya, 26)
21	non-heuristic		Tonge, 43), p. 172
22	particular heuristic		Gelernter, 9), p. 137
			NSS, 47), p. 155
23	planning heuristics		NSS
24	possible heuristic		Mesarovic, 46), p. 393
25	regrouping heuristics		Tonge, 43), p. 179
26	selection heuristics		Gelernter, *et al.*, 10), p. 156
27	selective heuristics		NSS, 47), p. 175
28	semantic heuristic		Gelernter, 9), p. 140
29	set of heuristics		Tonge, 43), p. 173
30	single heuristic		Gelernter, 9), p. 137
31	specific geometry heuristic		
32	special-purpose chess heuristic		Minsky, 37), p. 412
33	special-purpose heuristic		FF, 37)
34	successful heuristic		Slagle, 44), p. 197
35	union heuristic		Mesarovic, 46)
36	weight size heuristic		Mucciardi-Gose, 19), p. 261
	Usage as Adjective or Adverb		
37	heuristic ability		Goto, 48)
38	heuristic algorithm		Oda
39	heuristic approach		Polya, 26), Tonge, 43)
			Waltz-Fu, 49), Wilson-Wilson, 50)
40	heuristic argument		Polya, 28)
41	heuristic assumption		Polya, 26), p. 221
			Chernyavskii, 58), p. 154
42	Heuristichen Beweisgruende		Polya
43	heuristic capability		Carne, 8), p. 122
44	heuristic character		Polya, 26)
			Mesarovic, 46), p. 393
45	heuristic computer*3		Gelernter, 9), p. 139

*1 NSS = Newell-Shaw-Simon.

*2 FF = Feigenbaum-Feldman.

*3 It is the program which works on discovering a solving method for itself.

TABLE 2. (*Continued*)

No.	Terms	Main reference
46	heuristic conclusion	Polya, 26)
47	heuristic connection	Minsky, 35)
48	heuristiche Darsteillung	Polya
49	heuristic device	Gelernter, 9), p. 138
50	heuristic education	dictionary
51	heuristic effectiveness	Minsky, 37), p. 412
52	heuristically effective definition	Minsky, 37), p. 412
53	heuristic element	Oda
54	heuristich erhaltene Ergebnis	Polya
55	heuristic goal	Slagle, 44), p. 199
56	heuristic hint	Von Neumann, 34), p. 4
57	heuristic hypothesis	Polya, 27), p. 221
58	heuristic insight	
59	heuristic justification	Polya, 28), p. 173
60	heuristic learning	Hiro-oka, 23), 24)
61	heuristic machine	NSS, 47), p. 188
62	heuristic means	NSS, 47), p. 153
63	heuristic method	Slagle, 44), p. 193 Chernyavskii, 58)
64	heuristic model	Minsky, 45)
65	heuristic modeling	Fogel, *et al.*, 51), p. 8
66	heuristic motive	
67	heuristic point of view	Polya
68	heuristic policy	Mucciardi-Gose, 19), p. 257
69	heuristic procedure	NSS, 11), p. 114
70	heuristic process	NSS, 11), p. 114
71	heuristic problem-solving procedure	Tonge, 43), p. 172
72	heuristic problem solving program	NSS, 47), p. 155
73	heuristic program	Iwaki, 40)
74	heuristic programming	Zadeh, 52) Fogel, *et al.*, 51), p. 7
75	heuristic property	Gelernter, 9), p. 137
76	heuristic reasoning	Polya, 26)
77	heuristic retrieval	Chorafas, 53), p. 196
78	heuristic rule	Gelernter, *et al.*, 10), p. 155
79	heuristic search(ing)	Sklansky, 54), p. 7 and 17
80	heuristic selection	Waltz-Fu, 49)
81	heuristic solution	Polya
82	heuristic suggestion	Polya
83	heuristic syllogism	Polya, 26)-28)
84	heuristic synthesis	Wilson-Wilson, 49), p. 308
85	heuristic technique	Carne, 8), p. 134
86	heuristic transformation	Slagle, 44), p. 197
87	heuristic treatment	
88	heuristic value (of the diagram)	Gelernter, 9), p. 138
	New Terms proposed by the Author	
89	heuristic clue (or cue)	Oda-Nakamura, 1)
90	heuristic element	〃 , 3)
91	heuristic factor	〃 , 1)
92	heuristic jump	〃 , 1)
93	heuristic system	〃 , 3)

be started, from the starting point or from the end point ?, and so on.

Fig. 2 illustrates the relation between the heuristic elements a_{ij}'s and their corresponding domains searched by them. This figure shows that only special subsets of whole domain are searched by the special heuristic elements, a_{ij}'s. By a heuristic search method, some special subsets $\{R_{ij}\}$ are searched corres-

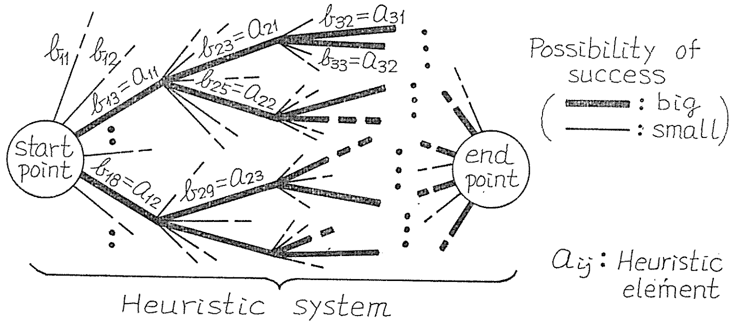

FIG. 1. Heuristic elements and heuristic system—First illustration of Definition 7.

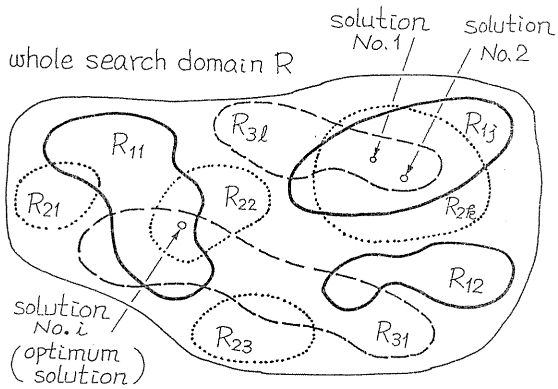

FIG. 2. Heuristic element a_{ij} and its corresponding search domain R_{ij}—Second illustration of Definition 7.

ponding to a series of $\{a_{ij}\}$, and finally the search domain is narrowed down to the intersection domain $\bigcap_i R_{ij}$. Thus, the huristic method, where only the special subsets of search domain are searched according to some heuristic elements with higher possibility, has the possibility of reducing greatly the necessary amount of search in comparison with other methods of searching the whole domain R.

4.2. Examples of Heuristic Elements

Real examples of the heuristic elements stated in the previous section appear in fairly different forms depending upon each practical case. Some practical examples of heuristic element are here collected from the various research fields as follows[1)3)]:

(1) It is better for a first move of "GO" (a Japanese game similar to checkers) to select the point of four-four (or its equivalent point)[*7].

(2) In chess or "Shōgi" (a Japanese game), (a) always take a check—it may be mate[43)], (b) cease to think the ply which expose the King to danger[37)], and soon[*8].

---

*7, *8 Rules in the games of "GO", "Shōgi", or chesss are considered here to be heuristic elements from the viewpoints of Definitions 5 through 7.

(3) In the problem of proving a theorem of elementary geometry, (a) draw a diagram[26)-28)43)], (b) reduce everything to sines and cosines in proving trigonometric identities[43)], (c) introduce an auxiliary element (auxiliary line, etc.)[26)-28)], and so on.

(4) An unknown function must first be assumed (a) to be linear, (b) to be monotonous[26)-28)] and so on[*9].

(5) In general problem-solving, (a) exchange elements, (b) reverse the order of elements, (c) combine elements, (d) apply the previously successful method of solving a similar problem (utilization of similarity, repetition of success), (e) decompose a problem into smaller subproblems, and so on[*10].

Examples in control engineering will be stated later in Sections 5.3. and 5.4.

4.3. Kinds of Heuristic Algorithms

There are various kinds of heuristic algorithm described in Section 4.1, which have been developed to solve specific real problems. The main kinds of them are listed in Table 3.

The general explanation of Table 3 is as follws: The general-purpose heuristic and special-purpose heuristic are developed for so-called general purpose and special purpose, respectively.

Means-ends analysis heuristics[8)38)50)57)] is a method which analyzes heuristically the relation between the end(s) and the mean(s) necessary to achieve it (them) in such a problem[*11] as to find out a complete path passing through from a given initial state to a given end(s). The general explanation of means-ends analysis heuristic is as follows: Assume that an initial state is given as $X=(x_1, x_2, \ldots)$ and an end as $Z=(z_1, z_2, \ldots)$. Then, its difference $D=(d_1, d_2, \ldots)$ where $d_i=z_i-x_i$, $i=1, 2, \ldots$.

TABLE 3. Main kinds of Heuristic Algorithm

No.		Kind	Proposer
1	a	general-purpose heuristic	Feigenbaum-Feldman[37)]
	b	special-purpose heuristic	
2*	a	means-ends analysis heuristics	Newell-Shaw-Simon[47)57)]
	b	planning heuristics	
3	a	intersection heuristic	Mesarovic[46)]
	b	union heuristic	
4		heuristic syllogism	Polya[26)-28)]
5**		weight size heuristic	Mucciardi-Gose[19)]

*, ** See Chapter 5 for the practical examples.

---

*9 Such heuristic elements as in the case are sometimes called "heuristic assumption" 26)-28).

*10 No. (a) through (c) in (5) have close relations with items of the check list in "creative thinking" such as (i) are there any other way to utilize?, (ii) if it is reconstructed?, (iii) if it is modified?, (iv) if it is expanded?, (v) if it is reduced?, (vi) if it is reorganized?, (vii) if it is reversed?, (viii) if it is recombined?, and so on[55)56)]. This is an example to show a similarity between heuristics and creation (or synectics[59)]).

*11 This is the same type of problem as a two-point boundary-value problem.

These two vectors are compared to determine the difference $D=(d_1, d_2, \ldots)$; and operators[8)38)57)] or effectors[50)] *12 $f_{ij} \in F(i, j=1, 2, \ldots)$ are combined to transform X into X', where $X'=(x_1', x_2', \ldots)$, $f_{ij}x_i = x_{ij} \triangleq x_i'$. If the new difference $d_i' = z_i - x_i' = 0$, then the operator $f_{ij} \triangleq f_i^\circ : f_i^\circ = \{f_{ij} : d_i' = 0\}$. After the process of checking all x_i's, the desired means $F^\circ = (f_1^\circ, f_2^\circ, \ldots)$ will be found.

Planning heuristics is used for a more complex problem where the difference between an initial state and an end can not be determined so easily as in the case of *means-ends analysis heuristics.* The planning heuristics is a method that (1) first omits the details of the initially settled complex problem, (2) makes a new simpler problem, (3) formulates and solves the simplified problem, (4) plans to solve the original problem and finds out the final solution by the help of the simplified solution.

Intersection heuristic and *union heuristic*[46)] are the methods which develop the thinking in such processes of set theory as convergence and divergence, respectively. In other words, *intersection heuristic* is a method which chooses the sequential search domain within the previous search domain in order to narrow down gradually the search domain with progress in thinking *13. On the other hand, *union heuristic* is a type of method which extends gradually the search domain into a similar and larger domain.

Heuristic syllogism[26)-28)] is abstracted by G. Polya, which is based upon the principle that "reasoning gains more reliability with a new fact" and has the following structure of reasoning:

Heuristic Syllogism
- Premise
  - (1) If $\boldsymbol{A}$ is true, then $\boldsymbol{B}$ is true.
  - (2) $\boldsymbol{B}$ is right.
- Conclusion (3) $\boldsymbol{A}$ gains more reliability.

4.4. Relation between Heuristics and Learning

As both heuristics and learning are fairly wide concepts (functions), it is not so easy for us to clarify the relation between them. However, referring to Definition 3 which relates heuristics and learning indirectly, and noting the existence of the terms *learning heuristic*[46)] and inversely *heuristic learning*[23)24)], we can conclude that heuristics and learning are different (but partially common) functions, but have such mutually helping properties as introducing the other function into the higher intelligent functions (wide heuristics and/or wide learning), cf. Fig. 3.

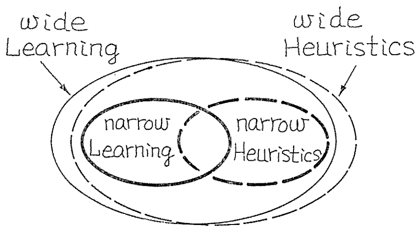

FIG. 3. Relation between heuristics and learning.

The standpoint of the above learning heuristic is based upon the introduction of

---

*12 *Operator* or *Effector* stated here is corresponding to ***heuristic element*** proposed by the author.

*13 Finally the search domain is to be narrowed into an intersection domain of all possible search domains at each stage.

learning into heuristic. Practically speaking, it is to choose and alter the heuristic algorithm (priority) by learning. On the contrary, that of heuristic learning is based upon the introduction of heuristic into learning. For example, it is to endow a heuristical property (cf. Definition 3 through 6) into the process of modification by learning.

From the viewpoint stated above, the introduction of the heuristic function into conventional learning control will make a powerful tool to solve more complex systems, or will make a new step to a higher learning control.

5. Survey of Examples of Heuristics

In this chapter, a survey of typical examples of heuristics is presented, and it will show us a possibility of introducing heuristics into control engineering.

5.1. Heuristics used for Proving a Logical Equation

Newell-Shaw-Simon[47][57] tried to prove a logical equation by the means-ends analysis heuristics. The essential part of the heuristical proof is illustrated as follows.

The problem is to prove the logical equation of Eq. (5.1) under the given condition of the following twelve rules of logical transformation.

Logical equation to prove:

$$(R \supset \sim P)\cdot(\sim R \supset Q) = \sim(\sim Q\cdot P) \tag{5.1}$$

Twelve logical rules given here:

R 1. $A\cdot B \rightarrow B\cdot A,\ A \vee B \rightarrow B \vee A$

R 2. $A \supset B \rightarrow \sim B \supset \sim A$

R 3. $A\cdot A \leftrightarrow A,\ A \vee A \leftrightarrow A$

R 4. $A\cdot(B\cdot C) \leftrightarrow (A\cdot B)\cdot C,\ A \vee (B \vee C) \leftrightarrow (A \vee B) \vee C$

R 5. $A \vee B \leftrightarrow \sim(\sim A\cdot \sim B)$

R 6. $A \supset B \leftrightarrow \sim A \vee B$

. . . . . .

R 12. $A \supset B,\ B \supset C \leftrightarrow A \supset C$

We will now investigate this problem from the viewpoint of means-ends analysis heuristics described in Section 4.3. The left side and right side of (5.1) correspond to the initial state X and the end Z, respectively; and the twelve rules correspond to operators (heuristic elements) f_{ij}'s. Besides, in this problem seven d_i's are to be the elements of the difference D[38]. To find out the relation between these d_i's and the necessary operators f_{ij}'s and to eliminate the d_i's are the key of heuristical solution of the problem, and it is shown as "connection table" in Table 4*14.

*14 The connection relation of Table 4 agrees as a whole with the result of proving experiment by a subject.

TABLE 4. Table of Connection between the Differences d_i's and the Operators f_{ij}'s

No.	No. of Logical Rule→ Elimination of difference ↓	1	2	3	4	5	6	·····	12	remarks	
1	add terms			0					0	difficult ↑	start ↓
2	delete terms			0					0		
3	change connective					0	0				
4	change sign					0					
5	change lower sign		0			0	0				
6	change grouping				0						
7	change position	0	0							easy	end

In Table 4, as written in a remarks column, the process of difference elimination develops from a difficult one gradually to a simpler one.

5.2. Heuristics used for Selecting the Important Terms in a Self-Organizing Nonlinear Threshold Network

Mucciardi-Gose[19] propose "*weight size heuristic*", which determines the specially important terms and their weight values from a lot of terms in a self-organizing network for pattern recognition.

For example, the problem is as follows: In the nonlinear threshold network of Fig. 4, assume the input is $X_j=(x_{j1}, x_{j2}, \dots, x_{jN})$ with elements $x_{jn}=1$ or 0, and choose M linearly-independent terms $f_m(X_j)$, $m=1, 2, \dots, M$. This is a problem of selecting M proper terms from the 2^N possible linearly-independent terms, and the total number of combination is $\binom{2^N}{M}$. The number $\binom{2^N}{M}$ grows very rapidly as the number N increases. Therefore, it is very difficult to select the M proper terms among them.

In the method of weight size heuristic proposed by Mucciardi-Gose, M terms $f_m(X_j)$ are first chosen at random to make a system (network), then the system is trained by a series of J training patterns X_j, $j=1, 2, \dots, J$. After the training

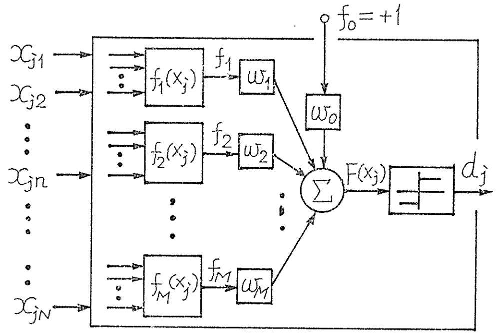

FIG. 4. Nonlinear threshold network.

the M_1 terms with higher values of weight w_m are kept in the network, while the $M_2=(M-M_1)$ terms with lower values of weight are discarded to exchange for the M_2 randomly-chosen new terms. Then, the network is trained again. As stated above, if such a process of both training and selecting phases is repeated, then the M desired terms can be discovered, and their weight values can also be decided.

Such a series process is called an *evolutional technique* by Mucciardi-Gose, who have experimented on the effectiveness of the technique. Formerly, Samuel[7] had made a checker-playing program on the basis of the same idea of heuristics.

5.3. Heuristics used for Solving an Optimum Control

Tou-Thomas[20,21] have researched the heuristic method of solving an optimum control from two points of view, namely experiments by subjects and by a digital computer. Although the dynamics of the controlled plant investigated were written as first order and/or second order, here we will consider only the case of the former.

In their experiment, as illustrated in Fig. 5, a subject reads the present values of six variables*15 on meters, and chooses heuristically a current optimum control y_k in order to minimize the fuel consumption (performance index) C_N at the terminal point. The subject in this case, however, is not informed about the relations of Eqs. (5.2) through (5.4):

$$\text{Dynamics:}\quad \left.\begin{aligned} &V_{k+1}=aV_k+by_{k+1}\\ &k=0,1,2,\ldots,N-1\\ &a,b=\text{const.},\ V_0=\text{initial velocity}\end{aligned}\right\} \tag{5.2}$$

$$\text{Performance Index:}\quad \left.\begin{aligned} &C_N=A\textstyle\sum_{i=1}^{N}(V_i-V)^2+B(V_N-V_f)^2\\ &A,B=\text{const.},N=\text{maximum trial number}\\ &V_f=\text{desired final value}\end{aligned}\right\} \tag{5.3}$$

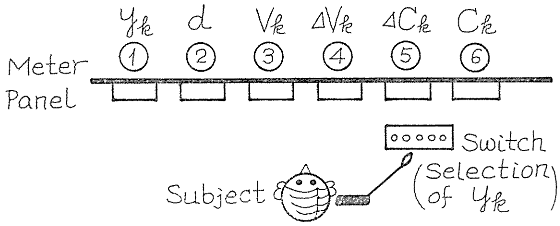

FIG. 5. Experiment of obtaining heuristically an optimum control by a human operator.

*15 Six variables are (1) current control choice, y_k; $k=0, 1, 2, \ldots$, (2) the remaining number of decision, d, (3) current velocity, V_k, (4) difference between desired final velocity and current velocity, $\Delta V_k=V_f-V_k$, (5) fuel consumption (cost) incurred during current decision interval, ΔC_k, and (6) current cumulative fuel consumption (cumulative cost), C_k.

$$\text{Indication of Meters: } \left.\begin{array}{l} d = N-k,\ \Delta C_k = A(V_k - V)^2 \\ C_k = \sum_{i=1}^{k} \Delta C_i = A\sum_{i=1}^{k}(V_k - V)^2 \\ 1 \leqq k \leqq N-1 \end{array}\right\} \quad (5.4)$$

The subject should choose a series of controls $y_k \in Y = \{-2, -1, 0, 1, 2\}$ so as to minimize the *PI*, C_N.

As a result of the experiment, a new method of heuristic search which a subject evolves was extracted as follows: "A subject pays his attention to the six indications of meters in order to detect if there is any of nine kinds of invariants illustrated in Table 5". If there is any one of invariants, a control y_k is chosen according to the invariant, as illusrated in Table 5. On the other hand, if there are two (or more than two) invariants, y_k is chosen according to the priority designated in Table 5.

Furthermore, Tou-Thomas, *et al.* made a computer simulation with the new search program which combined the above written heuristic search mode and three conventional search modes (a gradient mode, a terminal mode, and a probing mode), where the heuristic mode could be used when three other modes could not be used.

TABLE 5. List and Priority of Possible Heuristics to Minimize Fuel Consumption

Priority	Heuristics (Heuristic Elements)
1	Choose controls $\{y_k\}$ so that V_k = constant.
2	〃 〃 ΔV_k = constant.
3	Choose a control so that $y_k = y_{k-1}$*.
4	Choose controls $\{y_k\}$ so that $V_k/\Delta V_k = V_k$ = constant.
5	〃 〃 $y_k/(d\Delta V_k)$ = constant.
6	〃 〃 $\Delta C_k/C_k$ = constant.
7	Choose controls $\{y_k\}$ in order to minimize ΔC_k.
8	Choose controls $\{y_k\}$ so that C_k = constant.
9	〃 〃 d = constant.

* It means to choose a certain control repeatedly.

5.4. Heuristic used for Searching an Optimum Point on Multi-dimensional and Multi-modal Hills

The author and Prof. Nakamura have studied on the heuristics in "the search of an optimum point on multi-dimensional and multi-modal hills" as one of the important tasks which appear generally in the modern control problems. The outline of the research is described below, while its details will be reported somewhere in the near futnre.

As illustrated in Fig. 6, an experimenter and a subject sit opposite to each other to be partitioned with a screen. The subject can choose any point, as a trial point, on a search domain (a search paper which is quantized vertically 18 × horizontally 22 = 396 cells). For example, the subject chooses the i-th ($i = 1, 2, \ldots, N$) trial point (x_{1i}, x_{2i}); then the experimenter informs the subject about the corresponding height value y_i of the point on the test hill, which can not be seen by the subject due to the interception of the screen. The subject writes down the value y_i as $(y_i)_i$ in the cell (x_{1i}, x_{2i}). The search process stated above continues N times. Afterwards, the experimenter analyzes the process of searching an optimum point in order to extract heuristics evolved by the subject.

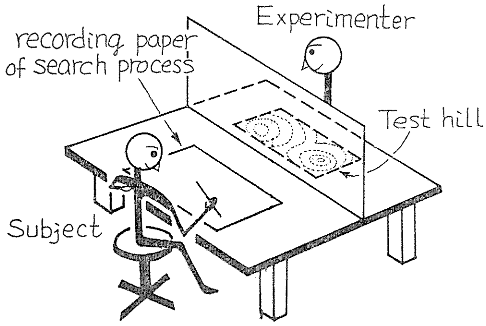

FIG. 6. Experiment of searching heuristically an optimum point on a multi-dimensional and multimodal test hill.

As for the conditions of the experiment, the followings are currently under investigation: (1) the conditions where the subject can choose the $(i+1)$-th trial point after either (1 a) *visually seeing* the whole results written previously on the search paper, or (1 b) *tactually recognizing* (touching) the previous results without visual observation[5]; (2) the conditions where the maximum trial number N is (2 a) fixed, and (2 b) free; (3) the condition where the test hills have various regularities (rules)—twenty test hills have already been made[3)-5)]; (4) the condition where the optimum point of the test hill is changed suddenly*[16].

From the results of our preliminary experiment, the author has already extracted the typical heuristics concerning (1) the first selection point, (2) the global search, (3) the local search (within a local search domain, and between local search domains), and (4) the convergent peak-confirming search. These heuristics will be a great aid to the construction of heuristic programs.

6. Conclusion

In this paper are presented both the fundamental discussion on heuristic function which is a higher intelligent function as well as learning function in a human thought process, and also the discussion on the possibility of introducing heuristics into control engineering (especially into learning control): In the first two Chapters, we investigated, as the background of heuristics, the engineering fields which need heuristics, the pedagogical background, and the history of research of heuristics. In Chapter 3, as the engineering discussion on heuristics, we considered the engineering necessity, various definitions, and various terms of heuristics. In Chapter 4, it was shown by the author that heuristics should be divided into two parts: "heuristic elements" and "heuristic algorithm", and the relationship between heuristics and learning was emphasized. In Chapter 5, the essential parts of heuristics were surveyed in four examples. Especially, the last two examples are the ones concerning control engineering.

The necessity of taking up large-scale and complicated systems with a lot of indetermination will gradually increase in the future of control engineering (especially, the learning control). In such a case, the heuristic function discussed

*16 This sudden change corresponds that by a sudden load change in Learntrols.

in this paper will grow up to be a powerful tool as well as learning in the near future.

The author expresses his great gratitude to Prof. Kahei Nakamura, Automatic Control Laboratory, Faculty of Engineering, Nagoya University, for his continuous guidance for this research, and also thanks the faculty and the graduate students of the Laboratory for their helpful discussions and aids.

References

1)* M. Oda and K. Nakamura: Fundamental Discussion on the Meaning of Heuristic Function in Automatic Control Engineering, Tech. Rept. Inst. Auto. Cont., Faculty of Eng., Nagoya Univ., Vol. 14, 61–107, Dec. 1966.
2)* M. Oda and K. Nakamura: The Meaning of Heuristic Function in Control Engineering, 1967 Joint Conv. Record of Four Inst. Elect. Eng., Japan, No. 2600 (April 1967).
3)* M. Oda and K. Nakamura: Fundamental Discussion of Heuristic Function and its Application to Learning Control, Sixth Conv. Record of Soc. Cont. Eng., No. 224 (Oct. 1967).
4)* M. Oda and K. Nakamura: On the Heuristics used in the Search of Optimum Point on Multi-dimensional and Multi-modal Hill, Tenth Joint Conv. Record of Auto. Cont., No. 116 (Nov. 1967).
5)* M. Oda and K. Nakamura: Establishment of Test Hills used in the Experiment of extracting Heuristic Function and Preliminary Experiment, 1967 Joint Conv. Rec. of Four Inst. Elect. Eng., Japan, Tokai Branch, No. 4 p-D-8 (Nov. 1967).
6) A. Newell, J. C. Shaw, and H. A. Simon: Chess-Playing Program and the Problem of Complexity, in 37), 39–70.
7) A. L. Samuel: Some Studies in Machine Learning using the Game of Checkers, in 37), 71–108.
8) E. B. Carne: Artificial Intelligence Techniques, Spartan (1965).
9) H. Gelernter: Realization of Geometry-Theorem Proving Machine, in 37), 134–152.
10) H. Gèlernter, J. R. Hansen, and D. W. Loveland: Empirical Explorations of the Geometry-Theorem Proving Machines, in 37), 153–163.
11) A. Newell, J. C. Shaw, and H. A. Simon: Empirical Explorations with the Logic Theory Machine—A Case Study in Heuristics, in 37), 109–133.
12 a)* K. Nakano, T. Sato, and S. Miki: Generation of Maze Pattern by Computer, Rept. at Research Meeting of Automata and Auto. Cont., Inst. Elect. Commu. Eng., Japan, Dec. 24, 1966.
12 b)* K. Nakano, T. Sato and S. Miki: Mazes constructed by Computer, Mathematical Sciences, 5-2, 49–57 (Feb. 1967).
13)* T. Sakai and M. Nagao: Storing and Processing Traffic Maps in Digital Computer, Jour. Inst. Elect. Commu. Eng. Japan, **50**-4, 629–636 (Apr. 1967).
14) H. Freeman and S. P. Morse: On Searching a Contour Map for a Given Terrain Elevation Profile, Jour. Franklin Inst., **284**-1, 1–25 (July 1967).
15) T. Kitamura: Automation of Design of IC Logical Network Package, **7**-8, 73–79 (Aug. 1967).
16) Mechanisation of Thought Processes, 2 Vols., Proc. of Sympo. at National Physical Lab., Nov. 1958, HMSO (1959).
17) A. D. Hall, Jr. and F. S. Acton: Scheduling University Course Examinations by Computer, Commu. of the ACM, **10**-4, 235–238 (Apr. 1967).
18)* K. Chikatsuji: Automatic Scheduling in a Flow Shop, Rept. Government Mech. Lab., No. 63 (Mar. 1967).
19) A. N. Mucciardi and E. E. Gose: Evolutionary Pattern Recognition in Incomplete Nonlinear

* written in Japanese.

Multithreshold Networks, IEEE Trans. on Elect. Comp., EC-15-2, 257-261 (Apr. 1966).
20) J. T. Tou, R. E. Thomas, and R. J. Cress: Development of a Mathematical Model of the Human Operator's Decision-Making Functions, Battelle Memorial Inst. Rept. (Oct. 1966).
21) R. E. Thomas and J. T. Tou: Evolution of Heuristics by Human Operators in Control Systems, 1967 IEEE International Conv. Record, Part 9, 179-192 (Mar. 1967).
22)* H. Minami: Character and Tasks of Behavior Science, Thought, Nov. 1966, 28-41.
23)* R. Hiro-oka: Modernization of the Contents of Education, Meiji Books Co. (Feb. 1967).
24)* R. Hiro-oka: Modification of Teaching, Meiji Books Co., (Feb. 1965).
25)* K. Abe: New School Mathematics and Discovery Learning, Memiors of Educational University of Osaka, C—Educational Science, No. 7, 85-103 (1965).
26)* G. Polya (trans. by Y. Kakiuchi): How to Solve it ?, Maruzen Books Co. (1954).
27)* G. Polya (trans. by W. Shibagaki): Mathematics and Plausible Reasoning (Vol. 1), Maruzen Books Co. (1959).
28)* G. Polya (trans. by W. Shibagaki): Mathematics and Plausible Reasoning (Vol. 2), Maruzen (1959).
29)* J. Hadamard (trans. by K. Fushimi, T. Ozaki): An Essay on the Psychology of Invention in Mathematical Field, Misuzu Books Co. (Nov. 1959).
30)* M. Wertheimer (trans. by T. Yatabe): Productive Thinking, Iwanami Modern Books Co., (July 1952).
31)* N. Wiener (trans. by S. Ikehara, S. Iyanaga, S. Muroga): Cybernetics, Iwanami Books Co. (1957).
32)* A. M. Turing: Computing Machinery and Intelligence, Mind, 59, 433-460 (Oct. 1959).
33)* J. Von Neumann (trans. by T. Iijima, S. Inomata, M. Kumata): Computer and Brain, Maruzen Books Co. (1964).
34) J. Von Neumann: Theory of Self-Reproducing Automata. Univ. of Ill. Press (1966), edited and completed by A. W. Burks.
35) M. Minsky: Steps toward Artificial Intelligence, Proc. IRE, 49, 8-30 (Jan. 1961).
36) M. Minsky: A Selected, Descriptor-Indexed Bibliography to the Literature on Artificial Intelligence, IRE Trans. on HFE, HFE-2-1, 39-55 (Mar. 1961).
37) E. A. Feigenbaum and J. Feldman (eds.): Computers and Thought, McGraw (1963).
38) A. L. Chernyavskii: Computer Simulation of the Process of Solving Complex Logical Problems (Heuristic Programming), Automation and Remote Cont., No. 1, 145-167 (Jan. 1967).
39)* K. Abe: On "Heuristic", Memoirs of Educational Univ. of Osaka, C—Educational Science, No. 6, 96-108 (1964).
40)* S. Iwaki: On Heuristic Program (1), Information, Information Processing, Information Retrieval, 1-1, 18-21 (Apr. 1965); same (2), 1-2, 13-17 (May 1965); same (3), 1-3, 22-29 (June 1965); same (4), 1-4, 41-46 (Jul. 1965); same (5), 1-5, 20-25 (Aug. 1965); same (6), 18-21 (Sep. 1965); same (7), 1-7, 18-21 (Nov. 1965); same (8), 2-3, 34-36 (Apr. 1966).
41) G. Pask: Learning Machines, IFAC, Survey Paper, Basle (Sep. 1963).
42) A. Newell and H. A. Simon: Heuristic Problem-Solving, the Next Advance in Operations Research, Jour. of Operations Research Society of America, 6-1 (1958).
43) F. M. Tonge: Summary of Heuristic Line Balancing Procedure, in 37), 168-190.
44) J. R. Slagle: A Heuristic Program that Solve Symbolic Integration Problems in Freshman Calculus, in 37), 191-203.
45) M. L. Minsky: Some Methods of Artificial Intelligence and Heuristic Programming, in 16), 5-36.
46) M. D. Mesarovic: A Unified Theory of Learning and Information, in Computer and Informaton Sciences, ed. by J. T. Tou and R. H. Wilcox, Spartan (1964), 392-409.
47) A. Newell, J. C. Shaw, and H. A. Simon: A Variety of Intelligent Learning in a General Problem Solver, in Self-Organizing Systems, ed. by M. C. Yovits, S. Cameron, Pergamon

* written in Japanese.

(1960), 153-189.
48)* E. Goto: Difficulties in Realizing Artificial Intelligence, Jour. Inst. Elect Commu. Eng., Japan (Automata Issue), **46**-11, 39-45 (Nov. 1963).
49) M. D. Waltz and K. S. Fu: A Heuristic Approach to Reinforcement Learning Control Systems, IEEE Trans. Auto. Cont., **AC-10**-4, 390-394 (Oct. 1965).
50) I. G. Wilson and M. E. Wilson: Information, Computers and System Design, Wiley (1965).
51) L. J. Fogel, A. J. Owens and M. J. Walsh: Artifical Intelligence through Simulated Evolution, Wiley (1966).
52)* M. Aoki: Electrical Engineering of California University, Berkley (Introduction of Laboratory), Jour. of Soc. Instr. and Cont. Eng., **5**-1, 56-57 (Jan. 1966).
53) D. N. Chorafas: Control Systems Functions and Programming Approaches, Vol. A—Theory, Academic (1966).
54) J. Sklansky: Learning Systems for Automatic Control, IEEE Trans. Auto. Cont., **AC-11**-1, 6-19 (Jan. 1966).
55)* E. K. Von Fange (trans. by Y. Kato and K. Okamoto): Professinal Creativity, Iwanami Books Co. (1963).
56)* A. F. Osborn (trans. by I. Ueno): Principles and Procedures of Creative Thinking, Diamond Co. (1958).
57) A. Newell and H. A. Simon: GPS, A Program that Simulates Human Thought, in 37), 279-296.
58) F. V. Anufriev, V. V. Fedyurko, A. A. Letichevskii, Z. M. Asel'derov, and I. I. Didukh: One algorithm for Proving Theorems in Group Theory, Kibernetika, No. 1 (1966).
59)* Z. Katagata: Techniques for Developing the Creative Thinking, Lattice (Nov. 1967).
60)* A. Onda (ed.): Education of Creativity (Vol. 1: Basic Theory of Creative Ability), Meiji Books Co. (Oct. 1967).
61)* A. Onda (ed.): Education of Creativity (Vol. 2: Development and Estimation of Creativity) Meiji Books Co. (Sept. 1967).
62)* A. Onda (ed.): Education of Creativity (Vol. 3: Planning and Practice of Creativity), Meiji Books Co. (Sep. 1967).
63) A. L. Samuel: Some Studdies in Machine Learning Using the Game of Checkers, II—Recent Progress, IBM J. Vol. 11, No. 6, 601-617 (Nov. 1967).

* written in Japanese.