Issues in Heuristics

by Luís MONIZ PEREIRA (Portugal),
Center for the Study of Cybernetics,
Instituto Superior Técnico,
School of Engineering, Lisbon.

Human problem-solving behaviour relies much on the so-called heuristic methods.

Given that our complex environment exhibits a certain regularity, it is wise of our species to profit from it. That regularity is imposed by its rules and entails that one may gain from using stored information for the solutioning of problems where that information is not given in the problem statement. This is the first distinction one may point out between heuristic and other well known methods : systematic (or algorithm like) and random (or chance).

When applying these latter methods there is no need for prior information. (One could for instance play chess either way, knowing only the rules of the game. A mechanical rat having thread a few mazes with some common property, will on the whole take less time if it explores a new maze heuristically than if it goes about it in a systematic or a random fashion).

Besides having the power to use the environment's regularity (like using some overall property of a function when hill-climbing), there are several other reasons for the eventual excellency of heuristic methods :

— first, they may yield approximate solutions that are satisfactory in view of a certain permissiveness of the environment and of our demands (like in everyday life) ;

— second, there may be no known or even unknown algorithms to apply ;

— third, even if there is one, it may not qualify to the practical needs for economy of means or speed. Moreover, random moves in tentative problem solving are usually poorly efficient and too risky.

— fourth, they may evolve through feedbacks incorporated in them (in contradistinction to the other methods).

Heuristics can be suggestively characterized by a mingling of systematic and chance methods. Indeed, it is characteristic of heuristic methods a selective attention focusing on *regions* of a problem (sub-statements, sub-problems, etc.) which are to be systematically dealt with, individually. Externally, it might seem as if arbitrary decisions are being made by those selections, although, actually, they are meeting some internal criteria. But, as we will see, there is a precise sense in which randomness does occur, for those criteria (and the degree to which they are to be met), can only be settled by other criteria (etc.), up to a level where arbitrary choice resolves the uncertainty. In this sense also, we may say that there is always a certain amount of trial-and-error in problem solving.

Because of this characterization we can regard heuristics as establishing a compromise between the more striking (and antagonic) features of both the other methods :

Systematic	*Chance*
Computationally reliable	Risky
Efficient	Non-efficient
Particular	General
Lengthy	Fast to appy
Costly	Economic to apply

Furthermore, systematic and random procedures are not but ideal ones, for even when applying an algorithm using a computer there is always an heuristic decision of why, when and where it is to be applied. Similarly of random methods (this is probably why humans are not good random numbers generators at all).

For the sake of completeness, it remains to be said that problem finding or objectives selecting is also guided by heuristic rules. The same applies (almost exclusively) to the appreciation of problem solvability before it is solved, and also to the case of giving up solving a problem along some line. Heuristic rules are once more present in the gathering of further data for an underspecified problem, or, on the contrary, they may remain responsable for the deleting of information in the case of redundant problem statements.

In order to gain some advantage from the regularity of the environment in which a given problem has a context, it is necessary to recognize in what ways profitable use can be made of information not given (explicitly) in the problem statement and which has been accumulated in memory. To do so entails the knowledge of (at least) how to accumulate it, what is meant by advantage and how to define and recognize the relevancy of information stored. (*Can this questioning of the environment be profitably related to the notion of H-computability, where H is a set, in Turing machines theory ?* is a question that might provide examples for what has been just said).

Another issue I would like to point out is the close relationship between heuristics and inductive reasoning.

Consider that we wished to infer from experience if *all crows are black*. We might go about it infering its logically equivalent formulation *all non-blacks are not crows*. But then we could do it without even getting out of the room !

The point is we feel that maybe 500 instances of black crows will suffice, but not even 100,000 instances of non-black non-crows will do. Why ?

Because we *already know* that crows exhibit a regularity, a consistency between themselves, which non-crows do not possess. It so happens that in the process of inductive inference we are making use of information not conveyed by the procedure used to tell crows from non-crows and blacks from non-blacks.

As Banerji points out in his introduction [1] : " The phenomenon " of generalization has received some attention from statisticians. " Their studies seem to indicate that the number of tagged objects " needed for establishing a degree of confidence in a description " is strongly dependent on the usefulness of the features used and " the resulting simplicity of description ".

The manner in which we select what information is relevant and the degree to the extent of which we have confidence in our sample of instances are both generated heuristically, according to our own experience and in a way that does not garantee certainty, but only the definition of confidence and the extent to which it is satisfied, which are, of course, acquired and up-dated with experience. This acquisition and this up-dating is accomplished by the higher levels of the mainly heuristic type of hierarchy of learning processes, as I will suggest next.

It seems to me one can distinguish at least three stages in heuristic reasoning :

1) recognizing or stating a problem,

2) recognizing which methods are appliable according to some criterious qualities and degrees of appliability,

3) carrying out the methods until certain criteria of solution attainement are satisfied, or, on the contrary, a certain measure of effort is surpassed, indicating to give up the problem or line of investigation.

Another stage might be present, in which the criteria, degrees and measures refered to are built up or reformulated. But the way this is done in a given instance is justified according to other criteria, degrees and measures which establish the pertinency of what is to be done. This way, we reach the notion of *hierarchy of heuristics*, a hierarchy structured in a manner similar to control hierarchies [4, 5]. (There is a recent paper on heuristics to generate heuristics in the game of poker, by Waterman [8]).

At the ultimate level of the hierarchy, however, a decision concerning the pertinency of pertinencies etc. has to be made arbitrarily if that level is to be ultimate (see [3, 6] for epistemological issues). This fact is a symptom of the necessity of trial-and-error in learning, whatever the level it must occur in. But heuristic induction is so important that we may postulate its legitimacy as in number theory.

Now, although we may state a problem in several ways, some of them make it amenable to readily recognized algorithms or heuristic rules. Clearly, the capacity to do so is closely linked to the problem of problem description, and to the general problem of pattern recognition. But the role of heuristics is specially important in *active* pattern recognition. By active is meant the generating of recognition hypothesis which are to be confirmed or infirmed by some kind of search. Clearly enough, not only the hypothesis making but also the particular search made to test it (and the threshold of confirmation), can be heuristically generated and improved (hill-climbing techniques provide an example of this, in the sense that the steepest slope may be found by exploratory movements).

To provide a means for recognizing the inclusion of a problem in a problem-class we *might* rely on the evaluation of special functionals giving us a measure of *overall* properties expressed by the problem statement. A set of functionals permitting maps from one description space to another will have to be closely tied up to the languages of problem description, and it seems apparent that these languages, sooner or later, will have to be put forward axiomatically, along with the calculus pertinent to them. Recent references to an axiomatic approach to artificial intelligence and problem solving are [1] and [2].

A very useful heuristic can be the confrontation of two or more different problem-solving procedures which have not quite succeeded ; their comparison may lead to a successfull solution method. But because it is unpredictable when or where that comparison should be made, it is useful to have all those procedures being computed at the same time so that partial results of one may be used by another. This leads to the notion of parallel computation [7, 10], and probably explains why most of the process of human heuristic reasoning is rather unconscious.

Heuristic procedures cannot properly be called algorithmic not only in the sense that not always do they provide a correct solution or even a solution, but also in the sense that they may include *dice-throwing* decisions along the way ; besides, there is always an arbitrary choice decision at the beginning of the solution process at the top of the hierarchy.

The development of heuristic programming research comes after the early works of Thorndike and Köhler in animal problem solving and of Newell, Shaw and Simon in human problem solving and its simulation. To day, one can study computer problem solving by providing it with artificial environments and making it play games in them. A general and formal theory of problem solving seems to be on its way. The consequences will regard psychology, teaching, industry, business and war. As Norbert Wiener pointed out [9], human beings should be used for their greatest value : creativity. But who will be creative enough to predict the consequences of a machine-creativity revolution ?

References

[1] Banerji, R., *Theory of Problem Solving*. Elvesier Publ. Co., 1969.

[2] Banerji, R., and Mesarovic, M. D., *Theoretical Approaches to Non-numerical Problem Solving*. Springer-Verlag, 1970.

[3] Koestler, A., *The Ghost in the Machine*. Hutchinson, 1967.

[4] Leondes, C. And Mendel, J., *Artificial Intelligence Control*. In Survey of Cybernetics, edited by J. Rose, Iliffe Books, 1969.

[5] Mesarovic, M. D., Macko, D. and Takahara, Y., *Theory of Multilevel Hierarchical Systems*. Academic Press, 1970.

[6] Monod, J., *Le hasard et la nécessité — essai sur la philosophie naturelle de la biologie moderne*. Éditions du Seuil, 1970.

[7] Selfridge, O., *Pandemonium : a Paradigm for Learning*. In the Mechanization of Thought Processes, H.M.S.O., 1959.

[8] Waterman, R., in 1-2, vol. I, Artificial Intelligence Journal, Elsevier Publ. Co., 1970.

[9] Wiener, N., *The Human Use of Human Beings*. Houghton Mifflin Co., 1950.

[10] Artificial Intelligence Journal, 1-2, vol. I, Elvesier Publ. Co., 1970.