Logical Foundations of Artificial Intelligence

Angel Garrido
Faculty of Sciences, National University of Distance Education, Madrid, Spain
Paseo Senda del Rey, 9, 28040, Madrid, Spain
algbmv@telefonica.net

Abstract:
The procedures of searching solutions to problems, in Artificial Intelligence, can be brought about, in many occasions, without knowledge of the Domain, and in other situations, with knowledge of it. This last procedure is usually called Heuristic Search. In such methods matrix techniques reveal themselves as essential. Their introduction can give us an easy and precise way to the search for a solution. Our paper explains how the matrix theory appears and fruitfully participates in A I with feasible applications to Game Theory. Our paper is based on [2-4].
Key words: Artificial Intelligence, Fuzzy Logic, Game Theory, Infinite Games.

1. Introduction to Artificial Intelligence

The origin of A.I. can be traced back to the Dialogues of Plato, passing through Descartes and Leibnitz, until reaching three great thinkers: Janos von Neumann, Norbert Wiener and Alan Turing. And from then on, in the work of many scientists.

The problems analyzed by A.I. can be classified according to their level.

The first level includes problems of decision, learning, perception, planning and reasoning.

The second level includes tasks of classification, representation and search.

When we formulate a problem, we depart from its statement (explanation) in natural language. Fundamentally, its treatment is based in the "level of knowledge", introduced by Newell in 1981, as "abstract level of interpretation of systems, in A. I." An assertion called the "Rationality Principle" is also essential, stating that: "if a system has the knowledge according to which one of their actions leads to one of their goals, then such an action is carried out".

With regard to the techniques, we can distinguish between those dedicated to statements and those devoted to procedures. The former alludes to allowing the description of the known aspects of the problem. Their usual name is heuristic treatment. The latter specifies the process of finding the searched solutions. It is the algorithmic treatment.

The posing of a problem consists in constructing the solution(s) of such a problem. This needs:

1) an agent (the system or program to be executed), with a sequence of objectives
2) a set of actions, which allows one to reach such goals
3) a procedure of election, applicable to the different ways of reaching the goals

At first there were Descartes and Leibnitz, and then came three great thinkers: Janos von Neumann, Norbert Wiener and Alan Turing.

2. Search Methods

In the searching process, we can distinguish between:
- without information about the domain (Blind Search), and
- with information about of the domain (in this case, called Heuristic Search).

<u>Blind Search</u>

We can choose, according to the kind of problem, between:
- Extended Search
- Deep Search

together with other methods; some of these being derived. So, the latter is not the same as searching with the possibility of backward motion (backtracking).

Also, there are different methods, obtained from the previous, such as:

Searching in Progressive Depth (where the limit of depth is increased with each step), and the *Bidirectional Searching* (from the root-node to the final node and vice versa, crossing both graphs through a common node; if there is any solution, this will find it). Both names are sufficiently evocative of their nature.

We can also find another method, in this case not derived: the General Graph Search. In such a procedure, the possibility of immediate translation to matricial raising, with their incidence matrices is obvious.

All these methods would be joined to their corresponding algorithms.

The searching without information of the domain (blind search, BS) appears with the initial attempts to solve, through idealizations of the real world, playing problems, or the obtaining of automatic proofs. The searching process could be in state spaces. Such a searching procedure (BS) has applicability to problems with certain characteristics, such as:

- we can associate a state to each different situation of the domain
- there are series of initial states (in the searching process)
- there are operators, which allow the step between the successive states
- there is a final state

In such processes, there is an obvious correspondence between:

state ↔ *node of the graph*

arc (or link, into the graph) ↔ *operator*

And jointly to each graph, and their changes through the evolution of the problem, the corresponding matrices. So, we construct one path between the initial node and the final node. This exploration is achieved inside the state spaces. Such a space contains all the reachable states, if we activate the set of operators in all the nodes. Because this search is without information of the domain, we need to make an exhaustive exploration of the state space. Such an exhaustivity results to be unnecessary in Heuristic Search, because it will be possible to select each time the subsequent step by the available information.

Extended *Search*

We advance in the tree (or the graph) by levels. So, we obtain the lowest-cost, if it exists.

Deep Search

We expand only one link each time, from the root-node. If we reach a blind alley (cul de sac) in the graph, we come back to the nearest node and from this we take a ramification (or alternative branch) in the tree.

In this type of search, it is normal to establish an exploration limit, or depth limit (*dl*), by fixing the maximal length of the path, from the root.

Also, in this case the direction assumed is from left to right. It can be interpreted as the ordered ($l \rightarrow r$) and exhaustive journey of an imaginary ship, visiting each fiord, on this imaginary coast. We are avoiding the algorithms associated with each one of these searching processes.

3. Heuristic Search

Relative to searching with knowledge of the domain, in an initial phase, it was generally thought that all paths can be explored by the computer. But this is too optimistic: such an exploration can often be very difficult, because of the phenomenon of "combinatory explosion" of the ramifications, when we expand the search. Their spatial and temporal complexity can advise us against their realization. For this reason, we need to select, firstly, the most promising trajectories. In this way, we cannot obtain the best solution (optima), but an efficient approach to it.

Now, we introduce a new mathematical tool: the heuristic evaluation function, f. By such a function, we assign a value to each node, n. So, f(n) gives us the estimation of the real distance (unknown), from the actual node, n, until the final node, m.

One of these procedures is called the *Gradient Method,* or *Climbing Search.* According to this, in the expansion of each node, we must select the link which connects with the node of the subsequent level where the value of f is maximal, in the supposition of f reaching the greatest value in the finish node. We can also proceed in the reversed sense: reaching the lesser value, in each step, until the minimal, in the last node.

There are critics on the Heuristic Search, because of its unpredictability. It can find good solutions, but not necessarily the best.

The A algorithm.* This is a convention for the introduction of this algorithm with their useful properties of *completeness* and *admissibility.* According to the last property, if there is any solution, it will find it.

*A** is a particular case of the searching procedure of "first the best", into the strategies of "alternative explorations". It belongs to the Procedures of General Graph Search. In each step, we go revisiting the Open List. Initially, such a set is empty. Our successive elections would be based on the previous assignment to each node of the value of f in it. The selection of each node is attained according to the lesser value of the heuristic function on the nodes of their level, as a general rule. The comparison is carried out into the Open List, independent of the original level of each node. Generally, we prefer the solution of lesser cost. All the explored nodes, then, pass to be stored in the Closed List. Such nodes remain inactive for the rest of the process.

The heuristic function, f, can be decomposed in two parts or components, g and h:

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

for each node k. Where $g(k)$ gives us the real cost (known) of the best path found from the "root", or initial node, until the actual node, k. And $h(k)$ is the estimation of the length of the optimum path (unknown, until now), from k to the final node m.

4. Search problems with two adversaries

There are also strategies specially designed for the treatment of such problems.

Their general purpose is to select the necessary steps to winning the game (chess, generally; in fact, this was their origin).

We assume alternate moves. In each move, the ideal would be when the player knows his possibilities and even realizes the worst move for his adversary. But it is impossible to control it completely, generally because of the mentioned "combinatory explosion".

For this, we need to develop a tree of depth searching, with limited depth. Suppose each player always makes the best move in each turn. To estimate this goal, we need to introduce a more sophisticated function, which would measure, for each node, the possibilities of being a winner, a looser or a draw.

Let us go back to the algorithm A*, as in the "first the best" (which is a particular case of it), we mention the *Open List*, with the disposable nodes, in the successive steps, joining the value of the heuristic function in them.

Each time we choose a node, according to the minimal distance, we store such a node into the *Closed List*. And so on, until reaching the final node in the tree search.

More procedures of Heuristic Search: We can apply the exploration by graphs *And/Or*, where we represent a main problem and their sub-problems, as nodes pending from the root-node. In this case, we dispose of two types of links: *Or*, which indicates different options, and the link type *And*, which connects the father-nodes to the sub-problems. We can also consider the MINIMAX method, which makes an exhaustive exploration of the Search Tree.

And the method of "pruning", where the purpose is to reach the reduction of the number of visited nodes, detecting through a double counter, the pair (α,β). So, I can cut, leaving the remaining nodes pending from such a node.

5. Fuzzy and Modal Logic

Successive attempts were directed towards the summit of very complex problems. Not only in formal worlds, but for the real world as well. This requires new mathematical ideas, which are still in evolution today. Therefore, some essential formalisms appear, such as: Logic, Rules, Associative Networks, Frames, Scripts and so on. For each formalism here included, there are methods of handling the knowledge, which allows for our approach [7].

As you known, the step from disposable information to new information is called Inference, or Reasoning. The Classical Logic reveals promptly its inadequacy for A.I., where more sophisticated Logic is needed, extending the Classical. So, the Predicate Logic with Identity and then the Modal Logic (ML, in acronym). This extension (ML) of the Predicate Logic made evaluating arguments possible which included the concepts of necessity and possibility. Their symbols (as you know) are □ and ◇, respectively. Both of them would be called "modes of truth". This gives the surname of "Modal" for such a Logic.

Desirable properties for such Theories: To formulate the knowledge of the domain, *D*, in a very effective and efficient way, as theory, three characteristics could be necessary: *completeness, consistency* and *tractability.* According to the first of them (completeness), each formula must be demonstrable into the theory. For the second one (consistency), the new contributions to the system must not generate inner contradictions with the previous assertions or axioms. The tractability must give us a moderate complexity. That is, manipulating laws and premises in the Derivate Calculus, through the Inference process, must not result in excessive temporal and spatial complexities.

6. New trends in Algebra and Non-Monotonic Logic

There exists the possibility of obtaining an extended theory, in Linear Algebra, by the introduction of concepts and methods of Fuzzy Logic ([5], [8]), where the idea of sets, relations and so on, must be modified in the sense of adequately covering the indeterminacy or imprecision of the real world. In the problems concerning this "real world", only one of the "possible worlds", the Monotonic Logic seldom works. Such a type of Logic is Classical Logic in formal worlds, such as in the Mathematical fields of the past, because for the future we need to introduce uncertainty and their related problems. Instead of Monotonic Logic, the Non-Monotonic Logic ([1], [6]) must be developed and applied where the extension of the set of sentences can now modify consequences. This happens frequently in the real world: for instance, in the medical sciences. With these and similar techniques, we hope to obtain each time more interesting developments which may contribute to expand some classical and new fields in Game Theory.

References

[1] Brewka: *Non-monotonic Reasoning. Logical Foundations of Commonsense.* Cambridge University Press, 1991.
[2] Garrido, Angel: "Some deep connections between Linear Algebra and Fuzzy Logic". Contributed talk to the *ILAS ´04,* the "International Conference on Linear Algebra and Its Applications", held in the University of Coimbra (Portugal), 2004.
[3] Garrido, Angel: "Matrix Theory and Searching Methods in Artificial Intelligence" *International Congress on Matrix Theory,* held in the Center of Math. *Studies of the Technion*. Haifa, 2005.
[4] Garrido, Angel: "Logics in Artificial Intelligence" *Congress "Logic in Hungary 2005".* Organized by the Janos Bolyái Mathematical Society. Budapest, 2005.
[5] Klir & Yuan: *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice-Hall Inc., 1995.
[6] Mc Dermott & Doyle: "Non-monotonic logic I". *Artificial Intelligence,* 13, pp. 41-72, 1980.
[7] Mira: *Aspectos Básicos de la Inteligencia Artificial.* Sanz y Torres. Madrid, 1995.
[8] Zadeh: "Fuzzy Sets". *Information and Control,* 8, pp. 338-55, 1965.