Cognitively Realistic Problem Solving through Causal Learning

Seng-Beng Ho
Department of Social and Cognitive Computing
Institute of High Performance Computing, A*STAR
Singapore

Abstract - Traditionally, search algorithms such as the A* algorithm have been used for general problem solving in AI, including looking for paths that circumvent obstacles. However, even with the assistance of (often hand-coded) heuristics, the search space is still often very large. The problem solving process is not cognitively realistic because natural intelligent systems such as human beings do not solve problems in this manner. Building on our previous work on effective causal learning, this paper demonstrates how an intelligent system can learn about the physical properties of objects in the environment through identifying the causes of, say obstruction, and use that knowledge to solve problems rapidly without having to resort to extensive and unintelligent search processes.

Keywords: Problem solving; causal learning; rapid learning; cognitively realistic problem solving; intelligent agents; spatial movement

1 Introduction

The visual sense allows an intelligent system to learn about the inherent physical properties of objects as well as how they interact with each other or respond to causal entities such as forces. In our previous work [1] we formulated an effective causal learning framework that allows causalities such as those that could be present in the physical environment to be learned through visual observation. Causal rules that are learned using this method could be used for problem solving. Thus, this method could contribute to an intelligent system’s problem solving process in two major ways. Firstly, the causal rules are learned from observing the environment, so if the environment changes, such as if the physical laws that govern physical entities and their attendant interactions change, the intelligent system would learn and encode a new set of causal rules for problem solving and would then formulate different solutions accordingly. This allows the intelligent system to be adaptive. Secondly, unlike traditional methods of learning such as reinforcement learning [2] and the various kinds of supervised and unsupervised learning [3], effective causal learning requires very few training examples. Hence the system can learn the causal rules involved rapidly and deploy them rapidly for problem solving.

Traditionally, a general method for problem solving in AI is search, such as the A* search [4]. In A* search, if the process is applied to solve a problem such as one that involves physical entities, typically some knowledge of the physical world is built-in (such as knowing that some objects are impenetrable and could impede the movement of other objects, thus functioning as an obstacle), and the search process then uses this knowledge to generate a search tree to find a solution to a problem. One example would be a physical movement problem in which an agent is supposed to move from a starting location to a goal location. The search process generates many possible paths, and if there are obstacles, the built-in physical knowledge would inhibit the generation of the search tree in certain directions accordingly. There is also the use of heuristics, which is the measure of how close a certain search node is to the goal. This helps to cut down the search – only the most promising parts of the search tree as measured by the heuristics are searched. However, the search process is not cognitively realistic as it generates a large amount of unnecessary search space, and no attempt is made to learn useful causal knowledge about the physical properties of the environment that may help to drastically reduce the problem solving effort.

In this paper, we describe a method that overcomes the shortcomings of the traditional search method and apply it to a spatial movement problem that involves obstacles. As mentioned earlier, firstly the causal rules governing the behavior of the physical entities in the environment are learned and used in the problem solving process. These causal rules are learned as a result of a small number of training instances. Secondly, using the causal rules, the problem solving process can obviate a large amount of unnecessary search efforts and produce a solution rapidly, and this is due to the fact that the causal rules encode powerful generalizations about the physical properties of entities in the environment. This computational problem solving process is not only rapid, but also cognitively realistic as it parallels the process used by intelligent systems such as human beings for solving the same problem. The learning and problem solving method is also general, much like A* is a general problem solving method that can be applied to a wide range of problems.
1.1 Comparisons between A* and Current Method

In the earlier work [1] we have demonstrated the application of the effective causal learning method to solve the spatial movement to a goal without obstacle (SMG) problem. That process demonstrated mainly the learning of domain specific heuristics to obviate a large amount of search effort. In the current spatial movement to a goal with obstacle (SMGO) problem, the emphasis is on learning the causal rules associated with the obstacle to allow the intelligent system to reason a way to overcome the obstacle.

Figure 1 shows a comparison between the problem solving process generated by the A* algorithm and one that is expected to be generated by our causal learning method for the SMGO problem. In Figure 1(a), it can be seen that the A* search process has to search a relatively wide area of space, most of which is actually irrelevant to the problem at hand. The process is hence relatively “blind.” In Figure 1(b), we show the path that would be followed by the causal learning method that is also cognitively realistic. Basically, the Agent involved first heads toward the goal in a straight line (thinking that the earlier straight-line path learned in a SMG problem would work as it does not have any experience with obstacles yet), and then when it encounters an impediment to its movement – the Obstacle - it attempts a small amount of search/trial-and-error around the point of impediment to encode the cause of its impediment. Then it reasons out a method to counter the cause of impediment, and this knowledge is used by it to rapidly plan a path around the Obstacle as shown. Natural intelligent systems such as human beings would do likewise in a first encounter with an obstacle.

2 General Problem Solving Framework

In this section we use an example to develop a general framework for causal learning and problem solving that will be applied to the SMGO problem in the ensuing discussion.

In Figure 2(a), we show an irregular boundary representing the “environment.” There is an Agent in the environment. This environment is made up of a set of elemental objects (EOs). Suppose the Agent actuates a force, F(A), within itself to propel itself to generate a movement. The following causal rule would be learned through the method of effective causal learning as described in [1]:

\[
\text{At} (\text{Agent}(A), x_1) \land \\
\exists \text{All-EO}(S_1) = \{ \text{EO}(x) \}, \forall x, x \in \text{Environ}(E_1) \land \\
\exists \text{All-RD}(S_2) = \{ \text{RD}(\text{Agent}(A), \text{EO}(x)), \forall x, x \in \text{Environ}(E_1) \}
\]

Figure 1: SMGO problem. (a) A* search space. (b) “Intelligent” path generated by causal learning.

In the case of Figure 1(b), the path shown around the Obstacle is what is expected in a first encounter with the Obstacle. Having learned the movement impediment property of the Obstacle, subsequently the Agent would just head straight toward one of the corners of the Obstacle from the START position.

Figure 2: (a) An Agent actuates a force to move itself in Environ(E_1). (b) The Agent actuates a force in Environ(E_2). (c) The Agent actuates a force in Environ(E_2) at a different location but experiences no movement. “x” represents the absolute location and “t” the time of the action/event.
∀x ∈ Environ(E_i) \ [SYN] \ ∧
Actuate(F(A), x_1, t_1) \ [DIA] 
→ Move(Agent(A), t_1+Δ)

(1)

In Rule 1, Environ(E_i) is the environment shown in Figure 2(a) and RD is the relative distance between the Agent and an EO in the environment. We are assuming that a visual system would provide this and other visual information to the system. SYN and DIA are the synchronic and diachronic causal preconditions respectively as discussed in [1]. Basically SYN is the static causal context that must be present before an effect can take place and DIA is the change of state of some variable that must be present before an effect can take place. At the first observation of a change of state of the Agent, after an elemental time Δ, in the form of movement at time \( t_1+Δ \) – Move(Agent(A), \( t_1+Δ \)) – the system formulates causal Rule 1 by assuming that the three SY conditions are necessary. Firstly, the Agent must be at the absolute location \( x_1 = \text{At}(\text{Agent}(A), x_1) \). (Note that \( x_1 \) is a 2D vector.) Secondly, the existence of Environ(E_i), represented by the set of all EOs in it, is necessary. S_1 stands for this set. Thirdly, the existence of all the RDs from the Agent to all the EOs of Environ(E_i) is necessary. S_2 stands for this set of values. The DIA condition Actuate(F(A), x_1, t_1) is identified to be the diachronic causal precondition of the effect Move(Agent(A), \( t_1+Δ \)), based on the procedures discussed in [1].

Note that in our universe, the physical law is such that the force law – the movement of an object as a result of a force application – is not dependent on the presence of and the distance to other objects in the environment, unless the object involved and other objects in the environment are, say, electrically charged. In our case here, the causal learning process will discover if these synchronic conditions initially thought necessary are indeed necessary.

Now, suppose the force is actuated again after the Agent has moved to location \( x_2 \). The following rule is formulated:
\[
\begin{align*}
\text{At}(\text{Agent}(A), x_2) \ [\text{SYN}] \ ∧ \\
\exists \text{All-EO}(S_3) = \{\text{EO}(x), \forall x \in \text{Environ}(E_i)\} \ [\text{SYN}] \ ∧ \\
\exists \text{All-RD}(S_3) = \{\text{RD}(\text{Agent}(A), \text{EO}(x))\), \\
\forall x \in \text{Environ}(E_i)\} \ [\text{SYN}] \ ∧ \\
\text{Actuate}(F(A), x_2, t_2) \ [\text{DIA}] \\
→ \text{Move}(\text{Agent}(A), t_2+Δ)
\end{align*}
\]

(2)

There are three parameters that are different between Rules 1 and 2: \( x_1 \neq x_2 \), \( S_2 \neq S_3 \), and \( t_1 \neq t_2 \). This is because the Agent is now at a different location and the event takes place at a different time, and because the Agent is at a different location, the set of RDs to the EOs in Environ(E_i) is also different. The set of EOs, \( S_1 \), remains the same. A generalization can now be effected using Rules 1 and 2. Depending on the urgency of the situation, if an Agent is not desperate to derive a general rule to encode the phenomenon (in this case the force law) involved, it can wait for more instances of observation. On the other hand, if it is desperate, it can effect generalization with fewer instances. The derivation of the level of desperation in a given situation requires a separate computational module which we will not describe here. Here we assume that at least two instances are necessary to effect a generalization - we term this “dual instance generalization” [1]. Hence, given Rules 1 and 2, the following generalized rule is obtained:
\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{\text{ANY}}) \ [\text{SYN}] \ ∧ \\
\exists \text{All-EO}(S_3) = \{\text{EO}(x), \forall x \in \text{Environ}(E_i)\} \ [\text{SYN}] \ ∧ \\
\text{Actuate}(F(A), x_{\text{ANY}}, t_{\text{ANY}}) \ [\text{DIA}] \\
→ \text{Move}(\text{Agent}(A), t_{\text{ANY}}+Δ)
\end{align*}
\]

(3)

In Rule 3, the SYN condition associated with RD has been removed because \( S_2 \neq S_3 \) means that the RD condition is not necessary. In principle, the \( \text{At}(\text{Agent}(A), x_{\text{ANY}}) \) condition should also be removed as \( x_1 \neq x_2 \), which implies that it does not matter where the Agent is located. However, in order to have an explicit statement about the Agent’s location, we retain that term and instead use \( \text{At}(\text{Agent}(A), x_{\text{ANY}}) \) to indicate the fact that the Agent can be anywhere for this rule to hold. The \( x_{\text{ANY}} \) and \( t_{\text{ANY}} \) parameters in the DIA condition indicate that this “force law” is applicable anywhere and anytime in Environ(E_i).

Suppose now the entire physical “experiment” (actuating the force twice on the Agent) takes place in a different environment - Environ(E_2) – as shown in Figure 2(b). A general rule similar to that of Rule 3 would be formulated as follows:
\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{\text{ANY}}) \ [\text{SYN}] \ ∧ \\
\exists \text{All-EO}(S_4) = \{\text{EO}(x), \forall x \in \text{Environ}(E_2)\} \ [\text{SYN}] \ ∧ \\
\text{Actuate}(F(A), x_{\text{ANY}}, t_{\text{ANY}}) \ [\text{DIA}] \\
→ \text{Move}(\text{Agent}(A), t_{\text{ANY}}+Δ)
\end{align*}
\]

(4)

Now the set of EOs - \( S_4 \) – is different from the earlier \( S_1 \) because the environment is different. Combining Rules 3 and 4 a further generalized rule is obtained:
\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{\text{ANY}}) \ [\text{SYN}] \ ∧ \\
\text{Actuate}(F(A), x_{\text{ANY}}, t_{\text{ANY}}) \ [\text{DIA}] \\
→ \text{Move}(\text{Agent}(A), t_{\text{ANY}}+Δ)
\end{align*}
\]

(5)

The general rule is applicable anywhere, anytime, and in any environment. This is like the force law as we know it in our physical reality (provided that the Agent is not, say, electrically charged and the environment does not contain any charged objects).

Now, suppose at a specific location, \( x_1 \), the force law of Rule 5 does not hold, which is that the actuation of a force does not cause the Agent to move, as shown in Figure 2(c). An exceptional condition will then be added to Rule 5:
\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{\text{ANY}}) \ [\text{SYN}] \ ∧ \\
\text{Actuate}(F(X), x_{\text{ANY}}, t_{\text{ANY}}) \ [\text{DIA}] \\
→ \text{Move}(\text{Agent}(A), t_{\text{ANY}}+Δ)
\end{align*}
\]

\text{EXCEPT}

The tussle between the movement and NOT(Movement) situations for rule formulation could be resolved if additional information in the environment is available. Figure 3 shows that there are colors of the “patches” on the “ground” of the environment that the visual system can provide that can assist with the formulation of causal rules. Suppose a few instances of physical experiments are performed each on the WHITE and GRAY areas of the environment, and it is discovered that when the Agent is on the WHITE area, the movement rule is obtained and when the Agent is on the GRAY area, NOT(Movement) is obtained, then a set of rules as follows can be formulated using the extra SYN conditions provided by the ground color:

\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{\text{any}}) \ [\text{SYN}] \land \\
\exists \text{all-RD}(S_i) = \{\text{RD}(\text{Agent}(A), E_O(x)) \}, \\
\forall x \in \text{Environ}(E_i) \ [\text{SYN}] \land \\
\text{Actuate}(F(A), x, t) \ [\text{DIA}] \\
\rightarrow \text{NOT}(\text{Move}(\text{Agent}(A), t_{\text{any}+\Delta}))
\end{align*}
\]

(6)

2.1 Inductive Competition for Rule Generalization

A seemingly conflicting situation can arise as a result of applying the generalization process to the NOT(Movement) situation. Suppose now there is yet another location, say \(x_4\), at which the NOT(Movement) of the Agent is obtained when a force is actuated on it. Using the dual instance generalization process described above, a general rule could be obtained that states that an actuation of a force anywhere, anytime, and in any environment will lead to NOT(Movement) of the Agent, which is in conflict with the movement rule, Rule 5.

The process of generalization is an inductive process. Rules can change when new observations are made. And how we formulate the rules is also dependent on how we plan to use them. If there are not many situations under which Not(Movement) occurs, then when the situation of the Agent being at \(x_4\) mentioned above occurs, it can be added to Rule 6 as yet another exception. However, if more and more locations are found at which Not(Movement) is true, then it will be better to state that it is generally true that NOT(Movement) holds anywhere, anytime, and in any environment except at a few locations movement holds. We term this “inductive competition for rule generalization.”

Suppose in a planning process, the system wishes to plan a path across the environment, then the way the rule is formulated based on observations so far has an impact on the planning process. Suppose currently Rule 6 is the best characterization of the situation. Then the planning process can assume that the Agent would be relatively unimpeded when it moves across the environment, and at a few of the locations when this is expected not to be true, the planning process will formulate the plan accordingly. On the other hand, suppose the Not(Movement) version of Rule 6 is true, then the planning process would have to assume that at most locations the Agent will be impeded.

\[
\begin{align*}
\text{At}(\text{Agent}(A), x_3) \ [\text{SYN}] \land \\
\exists \text{all-E}(S_i) = \{E_O(x), \forall x \in \text{Environ}(E_i) \} \ [\text{SYN}] \land \\
\exists \text{all-RD}(S_i) = \{RD(\text{Agent}(A), E_O(x)), \\
\forall x \in \text{Environ}(E_i) \} \ [\text{SYN}] \land \\
\text{Actuate}(F(A), x_3, t_3) \ [\text{DIA}] \\
\rightarrow \text{NOT}(\text{Move}(\text{Agent}(A), t_{3+\Delta})) \\
\end{align*}
\]

The spatial movement to a goal (SMGO) problem is illustrated in Figure 4(a). Basically, an Agent, starting from a START location, is to move to a GOAL location, and in between these two locations, there is an Obstacle. In this section, we apply the general framework developed in the previous section as regards causal learning to the problem.

![Figure 3: Ground colors of environment provide extra synchronic (SYN) causal conditions.](image)

![Figure 4: (a) SMGO problem. (b) An attempted solution that results in the thwarting of a plan based on a script.](image)
Suppose the Obstacle has an identifiable color, GRAY, supplied by the visual system, the following causal rule is formulated:

\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{11}) & \quad \text{[SYN]} \land \\
\exists \forall \text{EO}(S_{11}) = \{\text{EO}(x), \forall x \in \text{Obstacle}(O_1)\} & \quad \text{[SYN]} \land \\
\exists \forall \text{RD}(S_{12}) = \{\text{RD}(\text{Agent}(A), \text{EO}(x)), \\
\forall x \in \text{Obstacle}(O_1)\} & \quad \text{[SYN]} \land \\
\text{Contact}(\text{Agent}(A), \text{Obstacle}(x_{11}+\delta)) & \quad \text{[SYN]} \land \\
\text{Color(Obstacle}(x_{11}+\delta), \text{GRAY}) & \quad \text{[SYN]} \land \\
\text{Actuate}(F(A, RA=0), x_{11}, t_1) & \quad \text{[DIA]} \\
\to \neg \text{NOT(Move}(\text{Agent}(A), RA=0, t_1 + \Delta))
\end{align*}
\]

This is similar to the formulation of Rule 8 using the color information of the ground. (Here, it is the color of the Obstacle, which is GRAY, that is the SYN condition.) Rule 9 is basically a logical description of the thwarting of the attempt (impediment of the movement) to reach the GOAL using the straight-line script. \(x_{11}\) is the location of the Agent, represented by its center. Because of the finite size of the Agent, the point that it contacts the Obstacle is slightly further away, at \(x_{11}+\delta\). Similar to the environment, say Environ\(E_i\) in the earlier example, which provides two SYN conditions to the causal rule involved (say, equation (1)), here it is the Obstacle that provides the sets of EOs and the RDs as synchronous causal conditions to the Agent’s movement impediment. We use \(S_{11}\) and \(S_{12}\) to represent these two sets of parameters respectively. In contrast to the parameters associated with the force, \(F(A)\), in the earlier discussion, here there is an additional parameter associated with \(F\), which is the relative angle \(RA\) subtended by the direction of the force and a line joining the center of the Agent and the GOAL. \(RA=0\) means that the force is applied directly toward the GOAL. This parameter and its associated value are derived from the script of the solution of the SMG problem [1, 5] that stipulates a series of forces pointed in the direction of the GOAL as the solution. The \(\neg\text{NOT(Movement)}\) consequence is also in the direction of the GOAL.

The DIA part of Rule 9 specifies that the actuation of a force on the Agent in the direction of the GOAL leads to a \(\neg\text{NOT(Movement)}\) consequence. In order to achieve movement, we need to negate the \(\neg\text{NOT(Movement)}\). We consider negating the SYN conditions to achieve this:

\[
\begin{align*}
\neg \text{NOT(At}(\text{Agent}(A), x_{11})) & \lor \\
\neg \text{NOT(}\exists \forall \text{EO}(S_{11}) = \{\text{EO}(x), \forall x \in \text{Obstacle}(O_1)\}) & \lor \\
\neg \text{NOT(}\exists \forall \text{RD}(S_{12}) = \{\text{RD}(\text{Agent}(A), \text{EO}(x)), \\
\forall x \in \text{Obstacle}(O_1)\}) & \lor \\
\neg \text{NOT(}\text{Contact}(\text{Agent}(A), \text{Obstacle}(x_{11}+\delta))) & \lor \\
\neg \text{NOT(}\text{Color(Obstacle}(x_{11}+\delta), \text{GRAY}))
\end{align*}
\]

In words, these mean:

(i) Cause the Agent not to be at \(x_{11}\).
(ii) Cause the Obstacle not to exist.
(iii) Cause the RDs from Agent to all the EOs on the Obstacle to be of a different set.

(iv) Cause “not-contact” between Agent and the Obstacle at \(x_{11}+\delta\).
(v) Cause the color of the Obstacle to be “not-GRAY” at the point of contact \(x_{11}+\delta\).

If (ii) could be achieved, the problem is solved right away. We will assume that this is not an option for the Agent – there are no known means that the Agent can dematerialize the Obstacle. (i), (iii), and (iv) could be achieved by changing the location of the Agent. (iii) and (iv) could also be achieved by moving the Obstacle - again, we assume this is not an option available for the Agent. We also assume that there is no ability on the Agent’s part to change the color of the Obstacle – option (v) – at the point of contact. So, what remains would be to change the location of the Agent. This process is carried out automatically computationally by checking the causal rule base to see whether the corresponding rules that can effect the various conditions (i) – (v) are available.

Note that any of the possible actions above is not guaranteed to work. For example, even if one can dematerialize the Obstacle, there could be another invisible force that impedes the movement of the Agent. So, even if the Agent’s location were to change, as recommended by option (i), there is no guarantee that it will work. However, it is still a good option to try.

The Agent now carries out a small number of physical experiments – a small blind search process – to determine the consequences of changing the location of the Agent, using the known means: the actuation of a force on the Agent. This force is applied in all possible directions to determine the various consequences of location change so that a good one could be selected. This process is not only applicable to the spatial movement scenario – in general, when a process is thwarted in the same manner as in the current case of spatial movement, all available actions will be tried in a small blind search process. Similarly, other non-spatial movement scenarios should also execute the ensuing process described for the current spatial movement scenario.

Figure 5: (a) Various directions of force actuation to derive the consequences of moving the Agent in these directions. (b) Direction \(d\) is chosen and another straight-line movement toward the GOAL is attempted from the new location.
directions, the Agent is free to move. This is learned and encoded in the following rules:

\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{11}) & \ [\text{SYN}] \land \\
\exists \forall \text{-EO}(S_{11}) & = \{\text{EO}(x), \forall x \in \text{Obstacle}(O_{1})\} \ [\text{SYN}] \land \\
\exists \forall \text{-RD}(S_{12}) & = \{\text{RD}(\text{Agent}(A), \text{EO}(x)), \forall x \in \text{Obstacle}(O_{1})\} \ [\text{SYN}] \land \\
\text{Contact}(\text{Agent}(A), \text{Obstacle}(x_{11}+\delta)) & \ [\text{SYN}] \land \\
\text{Color}(\text{Obstacle}(x_{11}+\delta), \text{GRAY}) & \ [\text{SYN}] \land \\
\text{Actuate}(F(A, 270^\circ < RA < 90^\circ), x_{11}, 1) & \ [\text{DIA}] \\
\rightarrow \text{NOT(Move(\text{Agent}(A), RA=0, t_{1}+\Delta))} \\
\end{align*}
\]

(11)

\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{11}) & \ [\text{SYN}] \land \\
\exists \forall \text{-EO}(S_{11}) & = \{\text{EO}(x), \forall x \in \text{Obstacle}(O_{1})\} \ [\text{SYN}] \land \\
\exists \forall \text{-RD}(S_{12}) & = \{\text{RD}(\text{Agent}(A), \text{EO}(x)), \forall x \in \text{Obstacle}(O_{1})\} \ [\text{SYN}] \land \\
\text{Contact}(\text{Agent}(A), \text{Obstacle}(x_{11}+\delta)) & \ [\text{SYN}] \land \\
\text{Color}(\text{Obstacle}(x_{11}+\delta), \text{GRAY}) & \ [\text{SYN}] \land \\
\text{Actuate}(F(A, 90^\circ \leq RA \leq 270^\circ), x_{11}, 1) & \ [\text{DIA}] \\
\rightarrow \text{Move(\text{Agent}(A), RA=0, t_{1}+\Delta)} \\
\end{align*}
\]

(12)

From these rules, it can be seen that the force actuation for the directions between c and d, and toward the GOAL (e.g., directions a and b, corresponding to \(270^\circ < RA < 90^\circ\), will lead to \text{NOT(Movement)} of the Agent, and the directions between c and d, and away from the GOAL (e.g., directions e, f, and g, corresponding to \(90^\circ \leq RA \leq 270^\circ\), will lead to movement. And among the directions in the latter case, directions c or d (depending on the point of contact between the Agent and the Obstacle) will give rise to a minimum distance to the GOAL. (For scenarios other than this movement, there are equivalent measures of “minimum distance” to the GOAL.) We apply the usual minimum distance to goal heuristic (often also used in A*) and select, say, d as the direction of force actuation to change the location of the Agent. This results in the situation shown in Figure 5(b) – the Agent is moved elementally in the direction d – and then another attempt is made to apply the straight-line movement script to the Agent to move it toward the GOAL to see if now the problem (of reaching the GOAL) can be solved.

The Agent will discover that its action of moving in direction d will again result in an impediment of movement in the direction of the GOAL much like before. However, this effort is not completely a waste as the Agent now learns a more general causal rule. Because in this second instance of impediment, the Agent is at a different location and the RDs to the EOs on the Obstacle are hence different, Rule 9 is generalized to the rule below (the “\(\exists \forall \text{-RD}(S_{12})\)” line is removed and \(x_{11}\) is generalized to \(x_{\text{ANY}}\)):

\[
\begin{align*}
\text{At}(\text{Agent}(A), x_{\text{ANY}}) & \ [\text{SYN}] \land \\
\exists \forall \text{-EO}(S_{11}) & = \{\text{EO}(x), \forall x \in \text{Obstacle}(O_{1})\} \ [\text{SYN}] \land \\
\text{Contact}(\text{Agent}(A), \text{Obstacle}(x_{\text{ANY}}+\delta)) & \ [\text{SYN}] \land \\
\text{Color}(\text{Obstacle}(x_{\text{ANY}}+\delta), \text{GRAY}) & \ [\text{SYN}] \land \\
\text{Actuate}(F(A, RA=0), x_{\text{ANY}}, 1) & \ [\text{DIA}] \\
\rightarrow \text{NOT(Move(\text{Agent}(A), RA=0, t_{\text{ANY}}+\Delta))} \\
\end{align*}
\]

(13)

The corresponding negated SYN conditions are:

\[
\begin{align*}
\text{NOT(At(\text{Agent}(A), x_{\text{ANY}}))} \lor \\
\text{NOT(\exists \forall \text{-EO}(S_{11}) = \{\text{EO}(x), \forall x \in \text{Obstacle}(O_{1})\})} \lor \\
\text{NOT(\text{Contact}(\text{Agent}(A), \text{Obstacle}(x_{\text{ANY}}+\delta))))} \lor \\
\text{NOT(\text{Color}(\text{Obstacle}(x_{\text{ANY}}+\delta), \text{GRAY}))} \\
\end{align*}
\]

(14)

The first condition means to bring about a situation in which the Agent is not anywhere. This is not a viable option. The second condition means to dematerialize the Obstacle and again we assume this is not an available option. Again, we assume there is no means for the Agent to change the color of the Obstacle. Therefore, what remains is to execute the third condition, which is for the Agent not to contact the Obstacle anywhere.

There are many ways to achieve a non-contact situation with the Obstacle. One way is for the Agent to stay next to the Obstacle and move along the length of it until the Agent moves slightly beyond one of the “corners” and arrives at a location at which there is an unobstructed path toward the GOAL. The other is simply to move away from the Obstacle. However, if we impose the minimum distance heuristic earlier as a result of which we picked the d direction for optimal movement, then it makes sense that for every step of a proposed long sequence of actions to achieve an unobstructed path to the GOAL, direction d is selected (adding a number of minimum distances will lead to an overall minimum distance).

At this point after the Agent has moved elementally in the d direction, similar exploratory elemental force actuations that lead to Rules 11 and 12 can be effected, and more general forms of Rules 11 and 12, which apply to any location along the Obstacle, would be formulated.

Figure 6: The repeated mental simulation of earlier learned elemental solution that leads to a complete solution.

Figure 6 shows that the Agent system carries out a series of computational mental simulations, selecting the optimal direction of movement, d, at every step and moving the Agent until it moves “beyond” the “right corner”, at which location (labeled X in the figure) the Agent no longer contacts the Obstacle. At every step of the simulation, there is no need to carry out any further physical experiments to learn about which directions of movement would give rise to obstruction, as the generalized versions of Rules 11 and 12 (that specify that at any location along the Obstacle, there are similar obstructions in the direction of the GOAL) provide that
information and imply that the optimal direction is \( d \) at every location next to and along the length of the \textit{Obstacle}.

There are two situations that can obtain after this first stage mental simulation that generates a straight-line path along the \textit{Obstacle} as the first steps of movement toward the \textit{GOAL}. Suppose the \textit{Obstacle} has some thickness as shown in Figure 6 and the \textit{Agent} has knowledge about it. After the mental simulation brings the \textit{Agent} to location \( X \), it then attempts to move the \textit{Agent} in the direction of the \textit{GOAL} and the effort fails because the thickness impedes the movement of the \textit{Agent} in the same manner as before. Then, another stage of mental simulation similar to that executed earlier would bring the \textit{Agent} to location \( Y \), at which point the \textit{Agent} can move straight to the \textit{GOAL} unobstructed. This entire path – move to \( X \), then to \( Y \), and then to the \textit{GOAL} is returned as a solution to be executed physically. Another situation is that the \textit{Agent} does not have the knowledge of the thickness of the \textit{Obstacle} from its earlier locations (because its view is obstructed). The straight line movement leading to location \( X \) is then executed physically after the first stage mental simulation, at which point the \textit{Agent} can “see” the thickness of the \textit{Obstacle} and it then executes a second stage of mental simulation to bring it to location \( Y \). Now, executing this straight-line movement to \( Y \) physically would then allow it to move straight to the \textit{GOAL} unobstructed after that.

The above mental simulation steps have been encoded in a general problem solving algorithm that can be applied to general situations.

4 Conclusions

We have successfully shown that the causal learning process as described in [1] can be applied to the SMGO problem to obtain a cognitively realistic solution to the problem, in contrast with the traditional AI approach of using the \textit{A*} algorithm that leads to blind and unintelligent extensive search effort. Causal learning, like \textit{A*}, is a general method for problem solving.

This research provides not only a computer implementable process for AI but also a framework and model for psychologists to investigate problem solving processes in natural intelligent systems such as human beings.

5 Acknowledgement

This research is supported by the Social Technologies+ Programme funded by \textit{A*STAR} Joint Council Office.

6 References


