

淺談人工智慧

71-280

作者：吳德旺

一、前言

人工智慧是研究設計「智慧型」的電腦系統，使電腦系統具有模擬人類智慧的行為。人工智慧（Artificial Intelligence）是麥卡錫（J. Mc Cathy）在 1956 年的一個學術研討會議中首先提出的，在 1980 年以前，這類學術研究工作並沒有受到特別的關注。

1981 年 10 月日本的第五代電腦計劃引起了全世界的震驚和關切，由於第五代電腦標榜的是人工智慧型的電腦，被認為將比以往任何一種電腦的功能更強大有力，它的真正力量不在於處理速度，而在於推理的能力。霎時間，人工智慧成為電腦科學發展的重點。

近年來，教育部和國科會推動了些人工智慧的課程和研討會，資策會和工研究院電子所也將人工智慧列入研究發展的目標。甚至民間的電腦團體也熱烈響應：舉辦「電腦風大賽」、「電腦圍棋比賽」，由此觀之，國內亦朝著人工智慧的領域邁進。

二、人工智慧的定義

人工智慧是電腦科學的一支，電腦科學對人工智慧所作的定義有：

Minsky：「人工智慧是一門科學，它使機器能從事那些需要人類智慧才能完成的工作」。

美國麻省理工學院人工智慧研究室主任 P. Henrr, Winston：「人工智慧乃是研究使電腦具有智慧的科學」。

Tesler：「人工智慧就是指那些目前還沒能做到（自動化）的事」。

由 Tesler 的定義來看，對人工智慧研究的對象—永遠在變的情形，做了很適當的詮釋。

三、人工智慧研究的領域

製造具有智慧的機器，也就是使機器的舉止像人類，一直是人工智慧這一科學領域的研究目標，人工智慧的目標是(1)使電腦更具功能(2)探究使電腦更具智慧的原理。

所謂智慧的行為，目前研究的項目有推理、解題、學習、識別、了解、表達……等

，因此人工智慧研究的領域有：

- 下棋、打牌、其他遊戲 (game)
- 造句、作詩、作文、寫小說
- 繪畫、作曲
- 證明幾何、代數中的定理
- 識別文字、圖形 (機器人之控制)
- 走迷宮、解謎
- 了解自然語言、回答人的問話
- 自動程式設計
- 專家系統

• 醫學診斷

• 化學分析

• 探礦

• 工程設計

四人工智慧解決問題的方法

人工智慧的程式主要就是解決問題，人工智慧所要處理的問題，應是那些還沒有被完全瞭解的問題。要把尚未瞭解的問題用電腦去處理、應該從人類如何去處理一些未能瞭解的問題上獲取一些啓示；例如，要具備知識、能獲得一些法則，並應用這些法則去解決問題。人工智慧解決問題的步驟是：

- 嚴謹地定義問題：包括啓始狀況及最後獲得結果之嚴謹定義。
- 分析問題：在解決問題時，有一些重要的特性對解決問題的一些技術的合適與否有很深遠的影響。
- 對一個特定問題造出最好的解決技巧。

1 以搜尋狀態空間的方式來定義問題：

水罐問題

給定兩個水罐，一個容量為 4 加侖，另一個為 3 加侖，兩者都沒有任何刻度。另外有個幫浦可以將水罐裝滿水，那麼，如何恰好將 2 加侖的水裝入 4 加侖的水罐呢？

這問題的狀態空間可以按次序的數字對 (X , Y) 的集合來表示，X 表示 4 加侖水罐中的水量，X = 0 , 1 , 2 , 3 或 4。Y 表示 3 加侖水罐中的水量，Y = 0 , 1 , 2 或 3。啓始狀態為 (0 , 0)，目標為 (2 , n)，n 為 0 至 3 的任意數。

圖 1 所描述的是解題的操作方式，它們的左邊表示的是現在的狀態，而右邊則是經過撮合規則操作後的狀態。

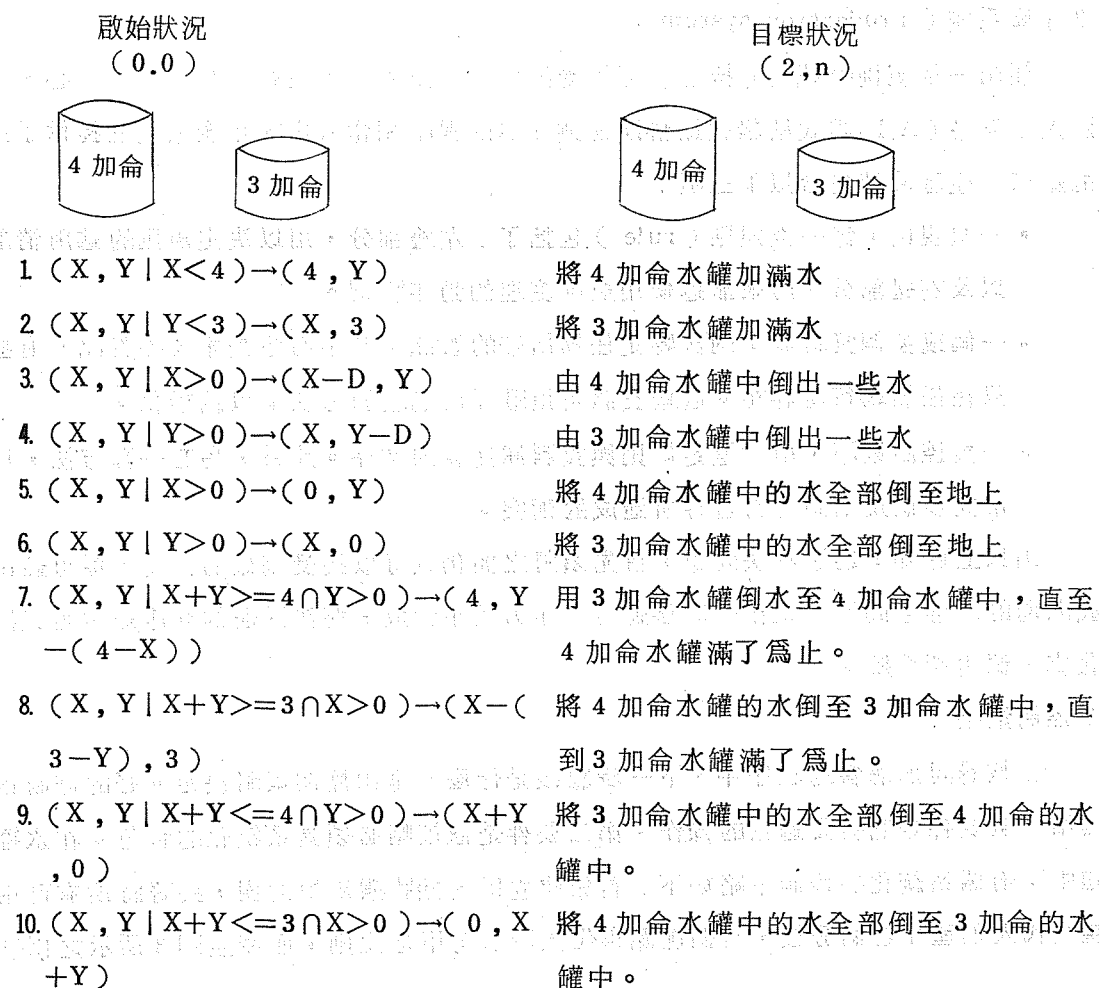


圖 1 根據水罐問題所得到的規則

在 4 加侖水罐中水量	在 3 加侖水罐中的水量	使用規則	或目標狀況
0	0	2	1
0	3	9	8
3	0	2	6
3	3	7	10
4	2	5	1
0	2	9	8
2	0		

圖 2 水罐問題的解決方法之一

在解決水罐問題時，跟其他問題一樣，有許多解決方法，通常所要找出的問題解決就是最短（或最便宜）的解法，其中一種解決如圖 2 所示。

2 生產系統（Production system）：

經由一組規則中找出（搜尋）可接受的解答狀態是許多有智慧的程序的核心，所以將人工智慧（A.I）程式結構化以幫助描述搜尋的程序相當有用。生產系統就提供了這樣的結構，生產系統包含以下三項：

- 一組規則，每一條規則（rule）包括了：左邊部分，用以決定規則的適用情形，以及右邊部分，用來描述使用規則處理的動作狀況。
- 一個或多個資料庫，內含特定任務所需的資訊，其中有些是永久的資訊，有些則只在任務執行時存在。這些資訊可以用任何適當的方法加以結構化。
- 一套控制策略，用來選定規則與資料庫比較的次序。此外，另有一套方法，用來解開多個規則同時符合時所造成的衝突。

由以上可知，為了解決問題，首先須將之簡化成可以接受精確的指述，這可經由定義問題的狀態空間及一組在空間移動的操作方式來達成，解決問題的程序通常可以模式化成一個生產系統。

3. 控制策略：

在找尋問題解答的程序中，下一步應該是什麼，全由控制策略決定。好的控制策略的第一個要件是它引發適當的操作，第二要件是該策略必須具系統化的特色。在水罐問題中，有個系統化的控制策略如下：首先建立以啓始狀態為根的樹，接著將所有可用的規則代入而產生各個分枝，每個節點再代入所有可用之規則，而產生圖 3 所示之樹狀圖。

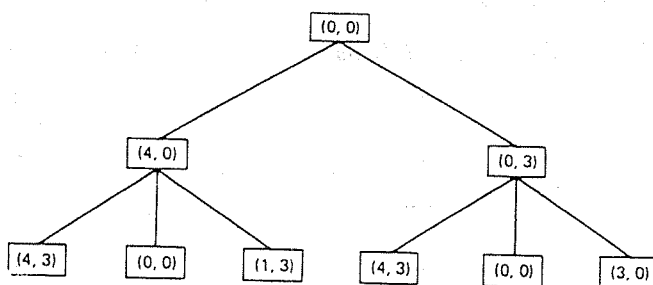


圖 3 兩層幅員優先的搜尋樹

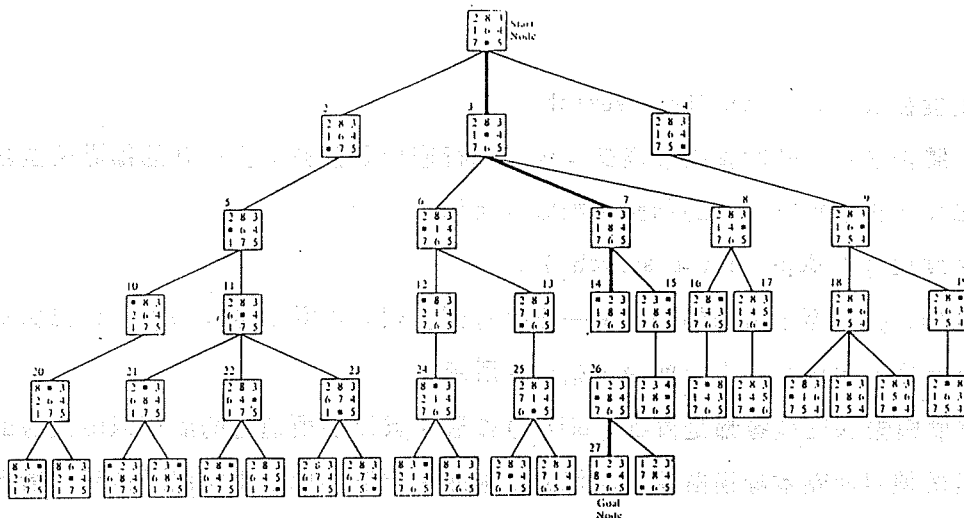


Fig 4

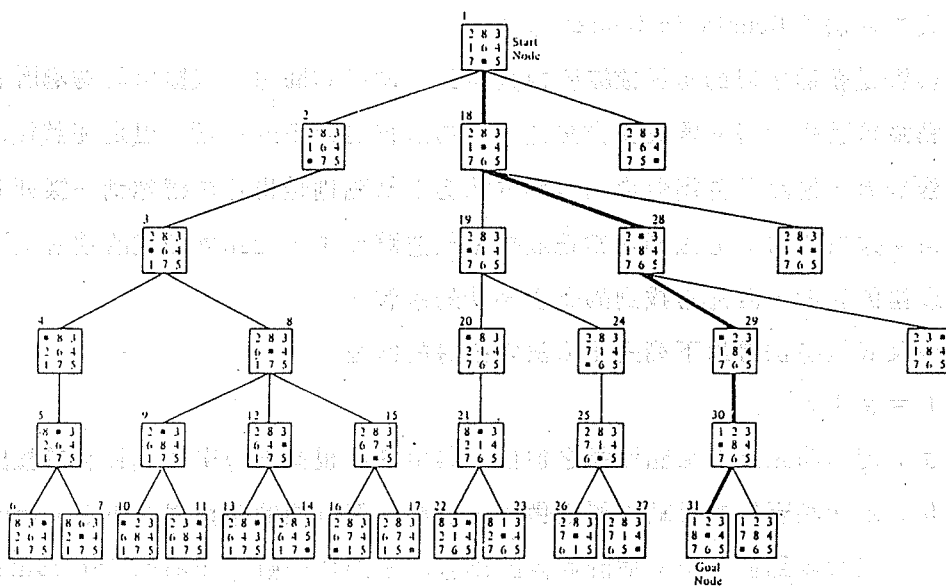


Fig 5

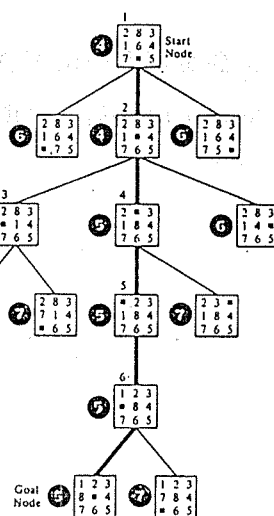


Fig 6

(1)先寬度搜尋 (bxeadth-first search) :

將搜尋樹的節點按階層高低、依序進行審視及擴充，上一層的節點完全審視擴充之後，再進行下一階的節點，如圖 4 所示。

(2)先深度搜尋 (depth-first search) :

先進行父至子方向的搜尋，一直搜尋到盡頭 (或所限定的深度) 仍無結果，才回到原處，再向另一方向搜尋，如圖 5 所示。

先寬度與先深度搜尋都是盲目 (blind) 搜尋，其搜尋所進行的順序只由搜尋的方法決定，與所搜尋節點本身所描述的狀況完全無關，完全不曾根據狀況所顯示的情況，所提供的資訊找出一條正確搜尋之道，而浪費許多不必要的搜尋步驟。

(3)啟發式搜尋法 (Heuristic Search) :

啟發是藉助任何的提示或經驗規則，以協助找尋問題，根據啟發誘導所作的搜尋，稱為啟發式搜尋。啟發式方法是一種增進搜尋效率的技術，但是完整性略差。它就像導遊，優點是會指引你到有趣的地方；缺點則是也可能帶你到一條死巷，走不出來。還有的啟發方法却常常把最佳路徑忽略掉了，因此啟發式的搜尋並不保證找到最佳的答案，但通常找到的是很不錯的解答。

啟發式搜尋法常以下列式子來決定搜尋的路徑：

$$f = g + h$$

g : 起始節點至該節點路徑之價格，其值通常根據搜尋樹的成長易於找出。

h : 該節點至目標節點路徑之價格，該節點至目標的距離很難決定，因此 h 值大多為估計值，估計 h 值的函數叫做評估函數 (evaluation function)。

f : 整個路徑之價格，每當要審視下一個開放節點時，通常選 f 值最小者。

例 1 圖 6 為 8 一方格 (pnzzle) 問題，用評估函數所解題的搜尋樹，其 h 值為方格不在目標狀況的數量和。

例2 圖7 為在正方格平面上，尋找由起始狀況(0, 0)至目標狀況(2, 4)的最短路徑。

$$f = g + h$$

$$g = |x| + |y|$$

$$h = \sqrt{(m-x)^2 + (n-y)^2}$$

$$h^* = |m-x| + |n-y|$$

$$0 \leq h \leq h^*$$

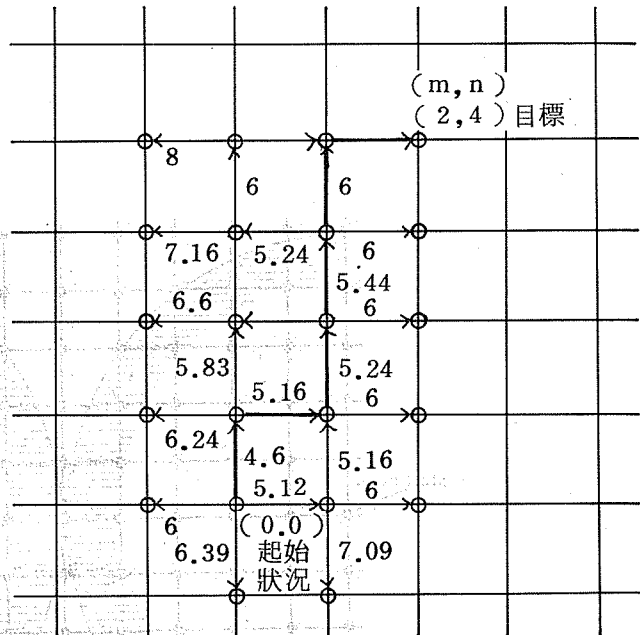


圖7 最短路徑搜尋

其他的啓示技術有：

- ①子女的次序，用啓示資訊估計那些子女最易於達到目標節點，就賦予優先審視的次序。
- ②限制子女產生的數量，當擴充一個節點時，將若干較有可能達到目標節點的子女保留，放棄其他子女，以減少無用的搜尋。
- ③樹的修剪，當搜尋一段時間仍未找到解答，使用啓示資訊來決定開放節點是可能的，然後剪掉 (prune) 所有不能到達這些節點的路徑。
- ④終止，用啓示資訊使不同節點有彈性的決定其深度限制，對於較有可能達到目標的節點，可設定較深的深度，反之則設定較淺的。

4. 搜尋（推論）的方向：

(1)從啓始狀態向前推理 (Forward chaining)：

以起始狀態為根，建立一個移動序列的樹狀圖，建立根部後，掃描所有的規則，若有合適的規則，則依規則的行動產生樹狀圖的下一層，重覆此一步驟，直到與目標形勢符合的節點出現為止。

(2)從目標狀態往後推理 (Backward Chaining)：

次目標狀態為根，建立一個移動序列的樹狀圖，建立根部後，掃描所有的規則，找出規則右邊部分適合根部結點的狀態，使用這些規則左邊，產生樹狀圖的下一

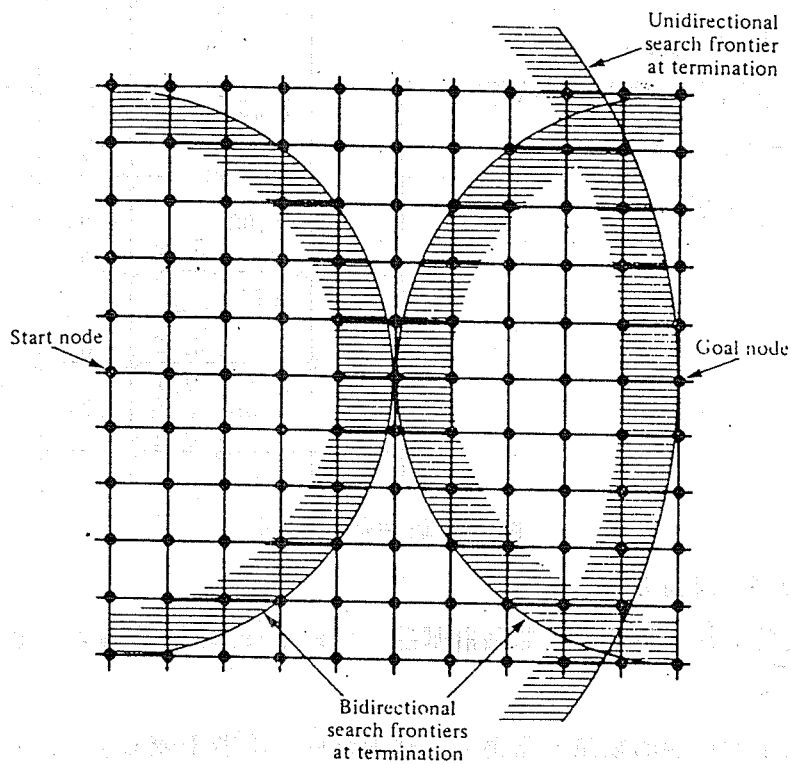


Figure 8 Bidirectional and unidirectional breadth-first searches.

層節點，再找出那些右邊部分符合此節點的規則，應用這些規則產生新的節點，繼續此程序，直到找到與啓始狀態相符合的節點出現爲止。這種由所求目標狀態逆向鏈結的方法稱爲以目標爲導向的推理方法（goal-directed reasoning）或逆向鏈結法（backchaining）。

(3) 雙向搜尋（bidirectional search）：

同時由啓始狀態正向推理及由目標狀態往後推理，直到兩條路徑在中央某處會合。

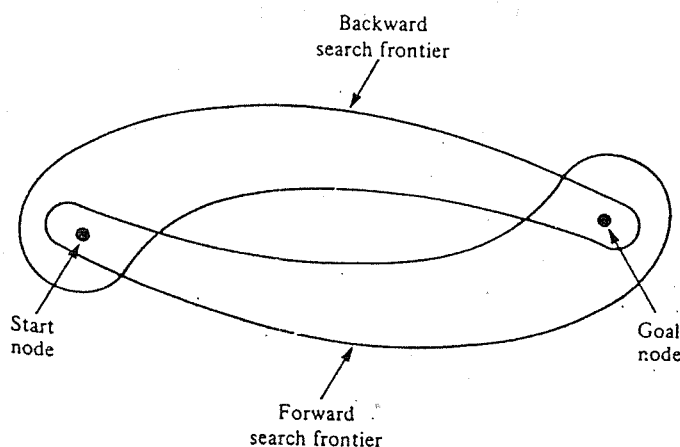


Fig. 9 : Forward search misses backward search.

五. 結論

電腦所以可能具備智慧，是因為我們採用啓發式的方法來編撰程式，相對的，一般電腦程式則都是根據既定的法則來編撰。使用既定法則來編撰程式，就像看食譜炒菜一樣，雖然做法可能很複雜，但是只要按部就班，一步一步往前做，就可以完成特定的工作，用這種方式來解決問題時，只要問題有答案存在，保證可以找到答案。至於啓發式的方法，則是參考人類的思維法則來設計，如果人類可以澈底瞭解本身智慧形成的經過，那麼就可以把它寫成程式，並教給電腦。所以說，智慧型電腦研究的進展緩慢，主要是因為目前人類的智慧尚不夠成熟。

六參考資料

- 1 Nils J. Nilsson: Principles of Artificial Intelligence 1980.
- 2 Elaine Rich: Artificial Intelligence 1982.
- 3 顏仁鴻譯：人工智慧，曉園出版社，1986年4月。
- 4 孫家麟著：人工智慧概論，第3波，民國75年。
- 5 施純協：資訊科技的發展與教育改革。
- 6 謝清俊：漫談人工智慧，科學月刊16卷11期。
- 7 張光耀：超級電腦與第五代，資訊與電腦，民國73年3月。
- 8 莊書彰：人工智慧基本概念初窺，0與1科技1986年7月。

