Local and global heuristic functions

We have already seen that the local maxima or minima is a serious problem. That problem also has some relation to the heuristic function chosen. If the heuristic function chosen is local in the sense that it does not really generates the value using overall situation but local situation, it is possible for it to stuck in local maxima. Unlike that, if the function chosen takes into consideration the overall situation and not confined to local parameters it has more chances to reach to a final state.

To understand the difference between a global and a local function,

let us take an example from a domain called blocks world.

The blocks world problems include a plain surface and some blocks of same size. The problem is about rearranging blocks from a given position to some other position. For example we might have two blocks resting on the surface called A and B, we may like to have A on top of B or vice versa. The other constraint is that only one block may be moved at a time, may be atop the surface or some other block. A block which does not have any other block on top of it is called a free block. One cannot move a block directly if it is not free. He has to move all the blocks resting on top of that block first, one by one, and only then it can move that block. The blocks world problem sounds simple enough for KG kids to play around. Blocks world problems and solutions proved quite useful to teach robots to stack and unstack things to get desired result. For example if a robot is asked to pick up a box which is lying somewhere in the room, may be under some other box and place it at some other place, may be on top of some other box, the blocks world algorithms are helpful.

Let us take an example of a blocks world problem shown in figure 1. We have blocks from A to F arranged as shown in the initial stage. We want them to be arranged in the form shown in the final state. As we are only able to move one block at a time we would like to find out the solution such that we should reach to final state ASAP. We will use some heuristic function for that.

The first part is to represent the initial and final state. We will use a form of predicate logic which we have used earlier to represent both the initial and final states as well as the state space.

Accordingly the initial state is defined as

On(-,A) // the – indicates the surface
On(-,E) // On (X,Y) means Y on top of X
On (-,F)^On(F,G)^On (G,D)^On(D,C)^On(C,B)

And the final state is defined as
On(-,G)^On(G,F)^On(F,E)^On(E,D)^On(D,C)^On(C,B)^On(B,A)

What will be the state space? We must state that all blocks are either has a block on top of them (can be any other block except the – which represents the surface) or are free (nothing on top of them).

$\forall$, $X \in A..G, -$ $Y \in A..G$ On(X,Y) V Free(X)

What will be the rules?
On(X,Y) and Free(Y) -> Free(X) and On(_,Y) //putting the block sitting on top of other block on table
Free(X) ^ Free (Y) -> On(X,Y) and ~Free(X) //when two blocks have nothing on top of them,
// putting one on top of other

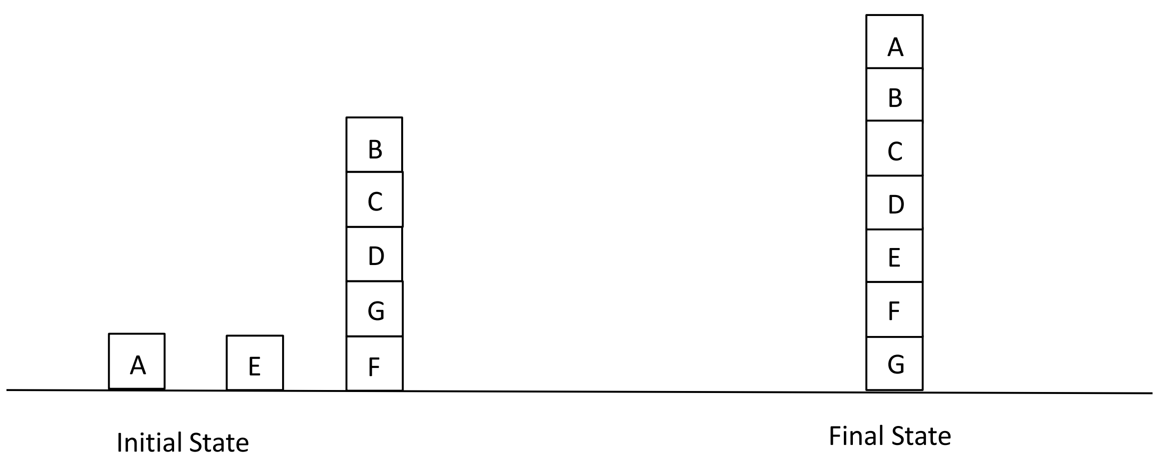

Figure 1 The blocks world problem

Using these two simple rules we can march forward from initial state. We can simplify the search using a heuristic function. Rich and knight has a similar problem with two different heuristic functions. We will be using those functions here. The first heuristic function adds one point to a block resting on the block it should be resting on and -1 otherwise. Looking at which the initial state heuristic function values are calculated as follows.

-1 for A, -1 for E, -1 for F and -1 for G and -1 for D as all of them are resting on something they should not be resting on. Similarly Both B and C are resting on the block they should be resting on, so both of them fetch 1 each. The total comes out to be -3. The Final state every block is resting on what it should so the total comes out to be 7.

Now let us look at all possible moves from the initial state. There are total five possible moves. Each of the resultant state's heuristic values are also calculated. Figure 2 shows all possible resulting states and the heuristic values of each resulting states. One can see that only one of the five possible states is better with heuristic value -1. Let us pick up that state. What are possible moves from it? You can see that all the states which result from it are not better than this state and thus hill climbing terminates at this point[1]. The figure shows all those states. One can see that none is better than previous.

You may consider the state-3 to have equivalent value so consider it as a successor. Even if we do that, the only move possible is to place E back on the surface which again has the same heuristic value -1. Each of the moves from now on has lesser heuristic value and so even this refinement does not lead us much further.

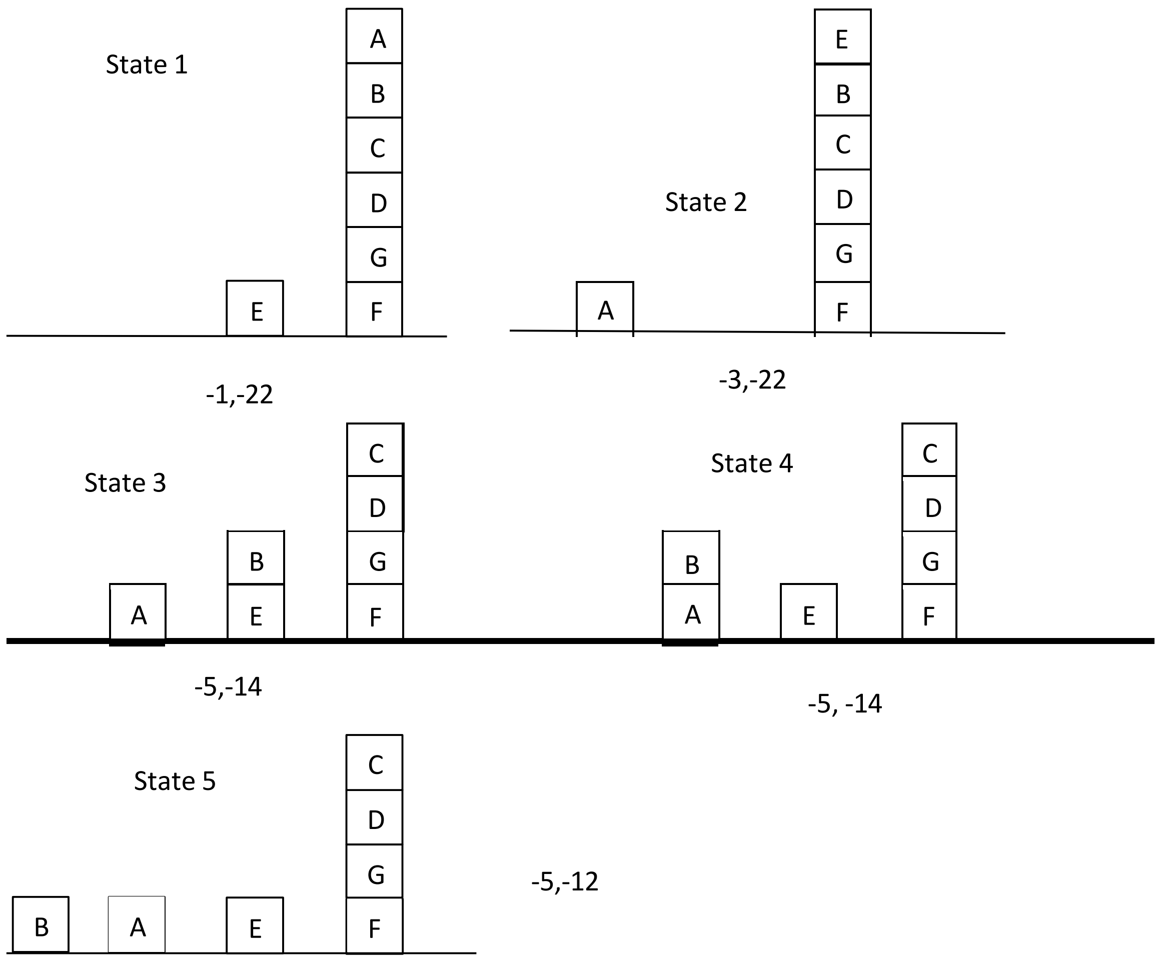

Figure 2 The heuristic function values, the local heuristic function values are written first and global heuristic function values are written next.

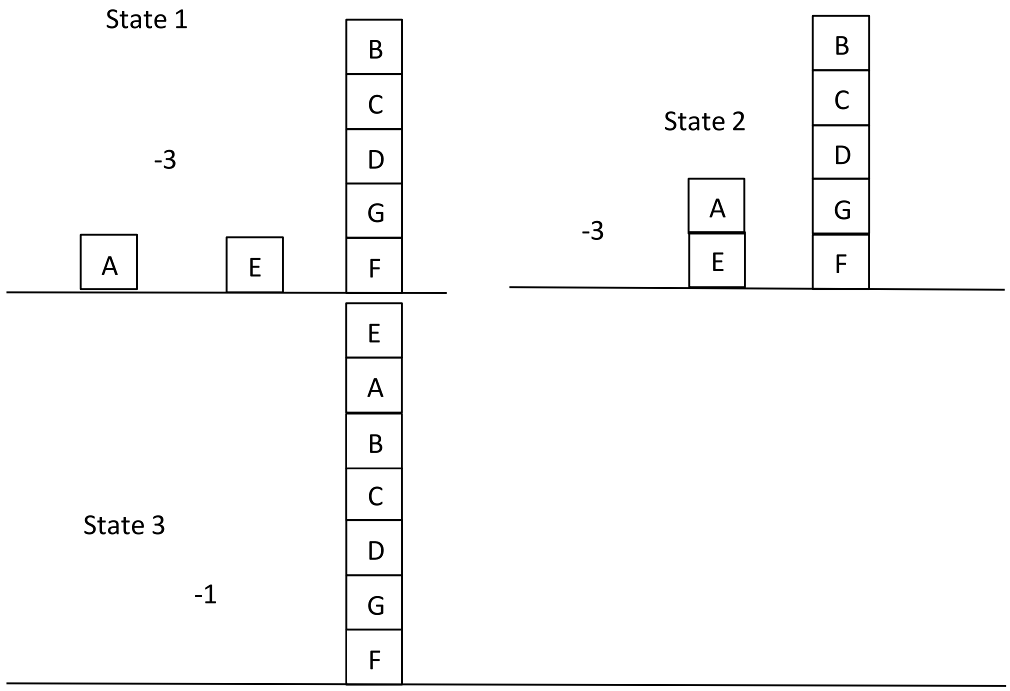

Figure3 three states resulting from the best move

You can remember our last module discussion about adding randomness to the movement and escaping local minima. Anyway, we would like to change the heuristic function to see what we can do. Let us take another heuristic function to see if the situation can be improved.

The second heuristic function does not consider the block on which the block under consideration is resting on but entire stack of blocks. Thus we will award one point for each of the block of the entire correct stack. If the stack contains even one block out of place, we will give a negative point for each ofthe blocks of the entire stack. Using the new heuristic function, the initial state's value calculated as

-1 for A, -1 for E, -1 for F, -2 for G, -3 for D, -4 for C, -5 for B, making it = -17

For the final state, 1 for G, 2 for F, 3 for E, 4 for D, 5 for C, 6 for B and 7 for A makes it =28

Now let us look at all possible moves from the initial state and calculate heuristic values of each of them. The state 1

-1 for E, -1 for F, -2 for G, -3 for D, -4 for C, -5 for B, -6 for A making it -22

State 2 will have identical value -22

State 3 -1 for A, -1 for E, -2 for B, -1 for F, -2 for G, -3 for D, -4 for C making it -14

State 4 will have identical value -14

While state 5 -1 for A, -1 for B, -1 for E , -1 for F, -2 for G, -3 for D, -4 for C making the total = -12

Figure 2 also depicts the heuristic value for the new function immediately after the value calculated for the first heuristic function.

Now the next best state is state 5 and not state 1. You can clearly see that out of all other possible moves, placing C on surface would be the best and so on till all blocks are placed on the surface. From there on placing right blocks on top of the structure not only increases the value of heuristic function but also would lead towards solution. We will never encounter the problem of local maxima ever if we use this function. The function will always be increasing for correct moves.

Why it is so different searching using different heuristic functions? If you look carefully you can get the difference. The first heuristic function is a local heuristic function. It only looks if the block is resting on the block it should and award point for the same irrespective of the global position of that block, considering other blocks either resting on top of them or below them. This heuristic function value does not change if the block it is resting on, contains correct stack of blocks or not. One can easily understand that if a block is resting on a correct block but have otherwise incorrect structure, it must be broken down and restructured. As this heuristic function does not consider that, it falls for local maxima.

Unlike that, the second heuristic function is a global function. The second function only allows complete support stack to be present. Even if one of the block is not correctly placed in the stack, it disallows the entire stack. Not only that, it also preferred such structures to be broken down by allowing more negative points for the blocks resting on top of incorrect structures.

In fact the example clearly indicates that global heuristic functions are to be chosen for avoiding local minima or maxima. Unfortunately not all domains and heuristic functions are simple enough for a designer to decide about the same. Many complex domains including chess contains some good heuristic functions but it is really hard for anybody to decide if the function is really global or not.