MANİSA CELAL BAYAR UNIVERSITY

CSE 3111

Artificial Intelligence

FALL 2018

Missionaires And Cannibles Problem

Report

Group

Erdoğan ABACI – 150315025

Mert ÇALIŞKAN – 150315012

26 November 2018

Table Of Content

1 – Problem formulation

1-1 Initial State:

This state has a six-variable of integers listing the number of missionaries, cannibals, and boats on the first side, and then the second side of the river which has a **MissonairesCanniblesEnvironment** class.

1-2 Possible Actions:

MissionairesCanniblesProblem detailed in this class.

1-3 Goal Test:

3 Missionaries and 3 Cannibals will pass from the first side to the second side with the bot. **MissonairesCanniblesEnvironment** detailed in this class.

1-4 Path Cost:

Cost function is one per action, and the successors of a state are all the states that move 1 or 2 people and 1 boat from one side to another. **PrintInstrumentation** detailed in this function.

2-Missionaires and Cannibles Class Explaning Details

2-1 MissionairesCanniblesGeneralProblem class

saves the current state of the problem, this class get initialState from super class and we implement missionaries,cannibles and bot like this=> (0,3,0,3,0,1),take actions and results from MissionairesCanniblesProblem class .Finally, MissionairesCanniblesEnvironment class return ok if isResultOk.

2-2 MissonairesCanniblesEnvironment class

is a class where missionaries and cannibals are identified.This class has variable of missioanaries,cannibles,boat.Also has setter and getter method.Then; We can check the accuracy with **isResultOk** function. **isStateOk** function check the condition which given to us.

2-3 MissionairesCanniblesAction class

is a class extends aima.core.agent.impl.DynamicAction (representing possible actions with dynamicly)

2-4 MissionairesCanniblesProblem class

has actions,problem from **MissionairesCanniblesGeneralProblem class** .

Implements an ACTIONS function for the createProblem formulation of the MissonairesCanniblesProblems

Assumes that MissionairesAndCannibles are placed , starting with an empty boat, and provides MissionairesAndCannibles placing actions for all non-placed positions of the boat and we checked states if state ok , we adding listActions.We checked five times,because we have 5 possible action to reach myState.

2-5 Actions:

Firstly;both the missionaries (M) and cannibals(C) are on the same side of the river.

Left Right

Initially the positions are : 0M , 0C and 3M , 3C (B)

Now let's send 2 Cannibals to left of bank : 0M , 2C (B) and 3M , 1C

Send one cannibal from left to right : 0M , 1C and 3M , 2C (B)

Now send the 2 remaining Cannibals to left : 0M , 3C (B) and 3M , 0C

Send 1 cannibal to the right : 0M , 2C and 3M , 1C (B)

Now send 2 missionaries to the left : 2M , 2C (B) and 1M . 1C

Send 1 missionary and 1 cannibal to right : 1M , 1C and 2M , 2C (B)

Send 2 missionaries to left : 3M , 1C (B) and 0M , 2C

Send 1 cannibal to right : 3M , 0C and 0M , 3C (B)

Send 2 cannibals to left : 3M , 2C (B) and 0M , 1C

Send 1 cannibal to right : 3M , 1C and 0M , 2C (B)'

Send 2 cannibals to left : 3M , 3C (B) and 0M , 0C

Here (B) shows the position of the boat after the action is performed. That is, how many missionaries and cannibals are in each side of the river and where is the boat (left or right).

Therefore, all missionaries and cannibals passed the river safely.

2-6 Implements a RESULT

function for the MissionairesAndCannibles. Supports MissionairesAndCannibles placing,and MissionairesAndCannibles movement actions.

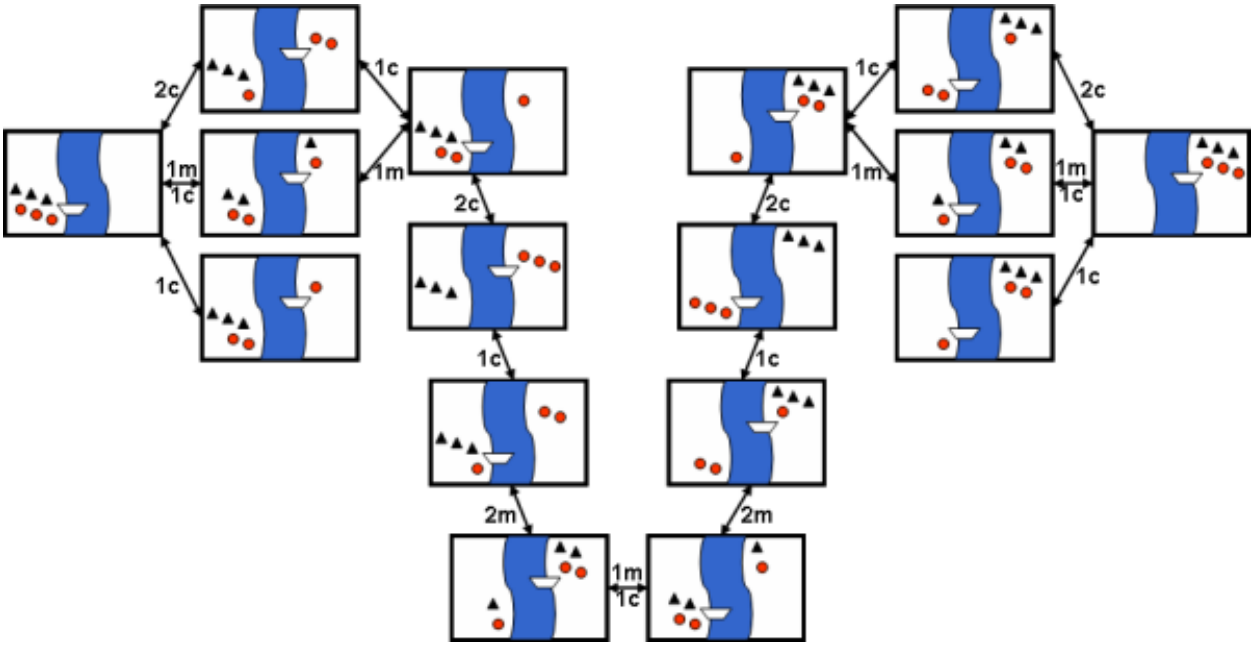

Figure-1 All Possible Actions

2-7 Goal Test Function:

Implements a GOAL-TEST for the MissionairesAndCannibles problem.This function checked state whether state reach testGoal.

3– Heuristic functions for informed search algorithms

3-1 Describe

Estimates the distance to goal by the number of positon pairs of MissionairesAndCannibles on the boat. I gave to my function closer to my target position.(2,2,1)

3-2 An admissible heuristic

Never overestimates the cost to reach the goal, A heuristic $h(n)$ is admissible if for every node n,because this rule => $h(n) \leq A^*$

Our heuristic is **consistent**,we have 31 queue size and we expanded just 17 nodes.This searchin result is relatively fast compared to other search trees.

4– Discussion on the results

4-1 Uninformed Searched

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening
Complete?	Yes	Yes	No	No	Yes
Time	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil})$	$O(b^m)$	$O(b^l)$	$O(b^d)$
Space	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil})$	$O(bm)$	$O(bl)$	$O(bd)$
Optimal?	Yes	Yes	No	No	Yes

Figure 2 Compare Uninformed Search

4-1-1 Missionaires&Cannibals BFS

```
Missionaires&Cannibals Demo BFS -->
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_M]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_M]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
pathCost : 9.0
nodesExpanded : 1919
queueSize : 3575
maxQueueSize : 3575
```

Completeness=Yes(Completed because It came to the goal node without approaching the end node.)

Optimal=NO(because,did not find the first node)

Space Complexity=Less than dfs,dls(dept limited),dıs(dept iterative)=>**2^3575+1** according to the above table.

Time=Same as space complexity.Every node keep memory until reach the goal state. **2^3575+1** according to the above table.

4-1-2 Missionaires&Cannibals Uniform-cost search

```
Missionaires&Cannibals Demo Uniform Cost Search -->
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_M]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
pathCost : 9.0
nodesExpanded : 3424
queueSize : 6546
maxQueueSize : 6547
```

Completeness= Yes(Completed because It came to the goal node without approaching the end node.)

Optimal=Yes(nodes expanded in increasing order of real path)

Space Complexity=More than breadth-first-search.

Time= More than breadth-first-search.

4-1-3 Missionaires&Cannibals Depth Limited Search

```
Missionaires&Cannibals Demo Depth Limited Search -->
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_M]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_M]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
pathCost : 9.0
nodesExpanded : 1205
```

Completeness=No(overflow the target limit**)**

Optimal=NO(the goal state is below the limit.**)**

Space Complexity=We can't decision

Time=too much than bfs and its not completed.

4-1-4 Missionaires&Cannibals Iterative DLS Search

```
Missionaires&Cannibals Demo Iterative DLS Search -->
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_M]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_M]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
pathCost : 8.0
nodesExpanded : 1688
```

Completeness= Theoretical completed yes but in this example **No(**overflow the target limit**)**

Optimal= NO(the goal state is below the limit.**)**

Space Complexity=We can't decision

Time= too much than bfs and its not completed.

4-1-5 Missionaires&Cannibals Greedy Best First Search

```
Missionaires&Cannibals Demo Greedy Best First Search -->
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_M]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_C_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_M_C]
pathCost : 9.0
nodesExpanded : 17
queueSize : 30
maxQueueSize : 31
```

Completeness=Yes(in this example not looping**)**

Optimal= Theoretical No but in this example.**Yes**(less nodes expanded)

Space Complexity=like dfs **2^31+1** according to the above table.

Time= too much than bfs and its not completed.

4-1-6 Missionaires&Cannibals AStar Search

```
Missionaires&Cannibals AStar Search -->
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_C_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_M_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_C_C]
Action[name=GO_LEFT_C]
Action[name=GO_RIGHT_M_C]
pathCost : 9.0
nodesExpanded : 145
queueSize : 274
maxQueueSize : 275
```

Completeness=Yes(problem is finite which is given**)**

Optimal=Yes

Space Complexity=2^275

Time=Keeps all nodes in memory

Compare Uninformed and informed Searches

If we compare between Uninformed and informed search,In this example informed search better than other searches. Greedy Best First Search has less node expanded.It benefit us to space and time.

5-References

https://github.com/marianafranco/missionaries-and-cannibals/blob/master/java/src/State.java (some code borrowed)

http://www.aiai.ed.ac.uk/~gwickler/missionaries.html (image taken in this link other image taken in book)

https://github.com/aimacode/aima-java/blob/AIMA3e/aima-gui/src/main/java/aima/gui/demo/search/NQueensDemo.java (some code borrowed)