Nonso Iwu

CISC481 HW1

1.

3 Missionaries, 3 Cannibals

Init State: MMMCCC|

Goal State: |MMMCCC

Actions: Left X or Left XX e.g. Left C, Left CM, Left L M

Right X or Right XX e.g. Right C, Right CC, Right MM

Path Cost: Uniform

Traversal Rules: If resulting state (# of M >= 1 and #M >= #C) or # of M == 0 then it’s a valid move

Breadth First Search:

RCC -> RMM -> RMC

2.

Init State: Sink 0 Gallons full

Goal State: Sink 4 Gallons full

Goal Test: If sink is 4 Gallons full then it is the goal state

Actions: Fill/Empty? Fill 3 Gallons, Fill Five Gallons, Empty 3 Gallons, Empty Five Gallons

Path Cost: Uniform

3.

a.

Successor-State function:

This function assumes:

A State has a cache unseated guests “*unseatedGuests*” for visited states exists

Expand(State *S*) returns a list of States *S’[ ]*:

S’ = [ ]

nextEmptySeat = -1

//Find an empty seat to fill

for every seat in S.table:

if seat is Empty:

nextEmptySeat = the index of seat

if nextEmptySeat is -1

return S’

for each guest in S.unseatedGuests:

newState = State.placeGuestAt(nextEmptySeat)

S’.add(newState)

return S’

b.

Heuristic(State *S*) returns int *h*:

*h* = 0

for every guest in *S*:

*h* += guest.currentHate - guest.lowestPossibleHate

return *h*

4.

Heurisitc(State S) return float *h*:

distX = (S.redCar.position.x – exit.position.x)

distY = (S.redCar.position.y – exit.position.y)

*h* = sqrt(distX^2 + distY^2)

return *h*

5.

a) Local Beam Search with k=1

-Hill climbing

b) Local beam search with an initial state and no limit on the number of states returned

-Breadth first search

c) Simulated annealing with T= 0 at all times (and omitting termination test)

-Hill Climbing but chooses the best successor

d) Simulated annealing with T = infinity

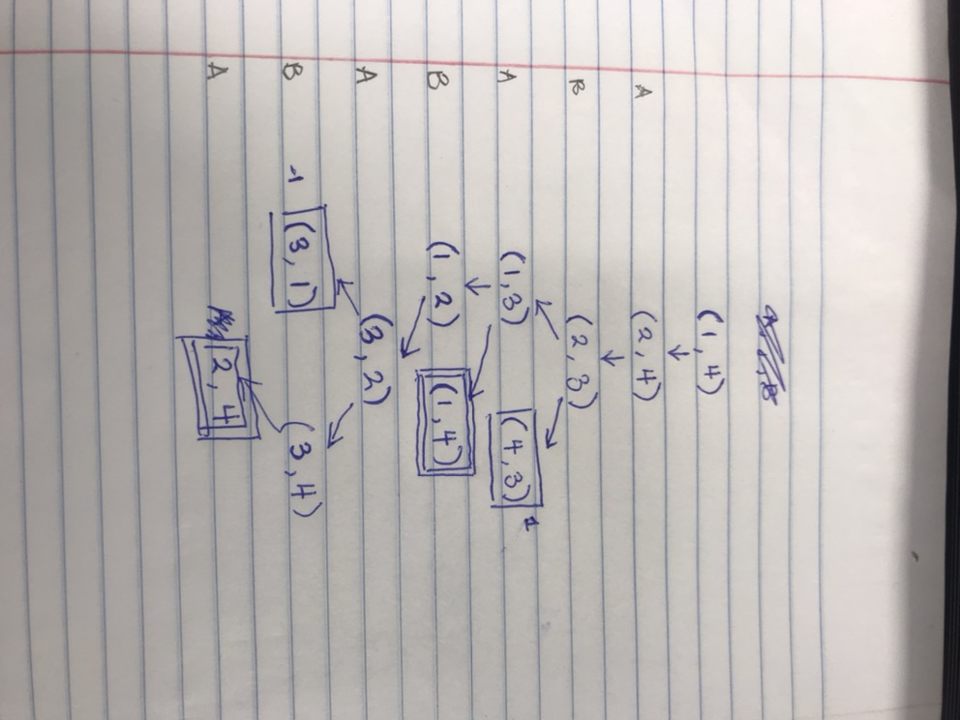
-Random Search

e) Genetic algorithm with the population size N = 1

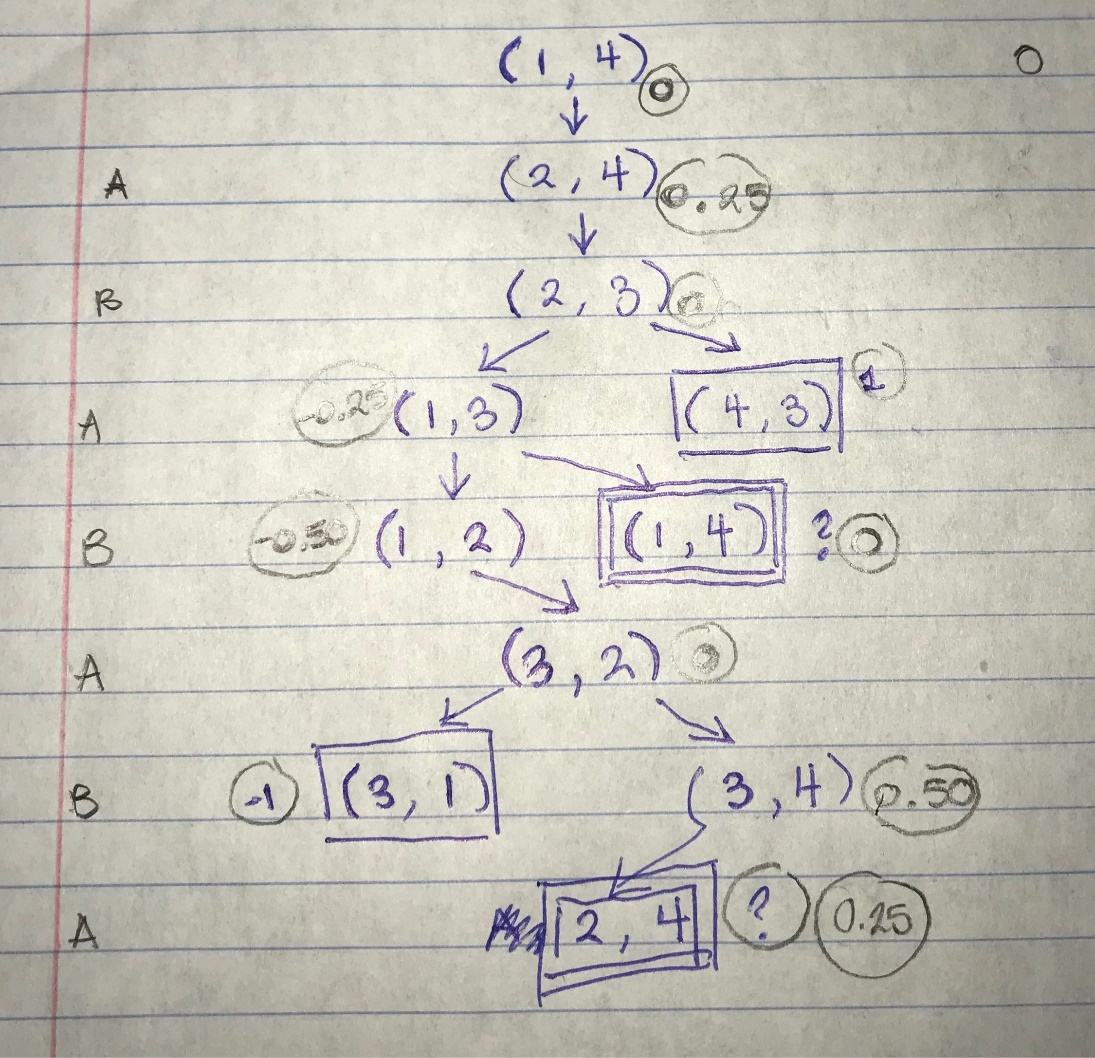
-Random Search

6.

a.



b.



c.

7.

a. Approximately: S \* M

# of possible starting points (S): (n\*m) choose 2

# of possible moves (M): (n\*m – 2) choose ((n\*m)-2/2) -

b.

A close up of text on a white background

Description automatically generated

c.

A close up of a whiteboard

Description automatically generated

d. Best move at starting position is left from white position

e.

A close up of text on a white background

Description automatically generated

8.

a.

Homework(x) => not isFun(x)

Homework(thisAssignment) => not isFun(thisAssignment)

b.

∃x(Guy(x) => hasLuck(x))

∃x(Guy(x) => hasPain(x))

∃x(Guy(x) => getBreaks(x))

∃x(Guy(x) => Complains(x))

c.

for every Dog(x) => hasBreed(x)

Mutt(x) <=> not Purebred(x)

Purebred(x) =>

( Purebred(y) and Father(y, x) ) and

( Purebred(z) and Mother(z, x) ) and

( Breed(y) ⬄ Breed(z) )

Yellow Labrador(x) ⬄ Purebred(x)

9.

Knowledge Base in CNF:

(¬Dog(x) ∨ Breed(y)) ∧ (¬Dog(x) ∨ hasBreed(y, x)));

(¬Mutt(x) ∨ ¬Purebred(x)) ∧ (Purebred(x) ∨ Mutt(x));

( ¬Purebred(x) ∨ ( Purebred(y) ∧ Father(y, x) )

∧ (¬Purebred(x) ∨ ( Purebred(z) ∧ Mother(z, x) )

∧ ( (¬Breed(y) ∨ Breed(z)) ∧ (¬ Breed(z) ∨ Breed(y)) );

Dog(Brandi)

Mother(Tabatha, Brandi)

Father(Moondog Moses, Brandi)

Yellow Labrador(Moondog Moses)

Yellow Labrador(Tabatha)

Brandi was not a mutt

not Mutt(Brandi)?

not Mutt(Brandi) ⬄ not not Purebred(Brandi)

not Mutt(Brandi) ⬄ Purebred(Brandi)