State Space Search

07/01/2025

Koustav Rudra

Search Frameworks

- State space search
  - Uninformed/Blind search
  - Informed/Heuristic search

- Problem reduction search
  - Decompose the problem into parts
  - Solve parts best way
  - Integration by parts

- Game tree search

- Advances
  - Memory bound Search
  - Multi-objective Search
  - Learning how to search

State Space Search

- Basic Search Problem:
  - Given: [S,s,O,G] where
    - S is the [implicitly specified] set of states
    - s is the start state
    - O is the set of state transition operators
    - G is the set of goal states
  - To find a sequence of state transitions leading from s to a goal state

8 Puzzle Problem

1	3	5
2	7	4
6		8

Start state

→

1	2	3
4	5	6
7	8	

Goal state

8 Puzzle Problem

- State description (S)
  - Location of each of the eight tiles (and the blank)

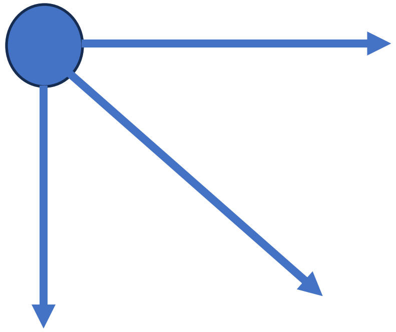
- Start state(s)
  - The starting configuration (s)

- Operators(O)
  - Four operators, for moving the blank left, right, up, or down
  - Not all the operators are applicable in each state

- Goal(G)
  - One or more goal state (given)

8 queens problem

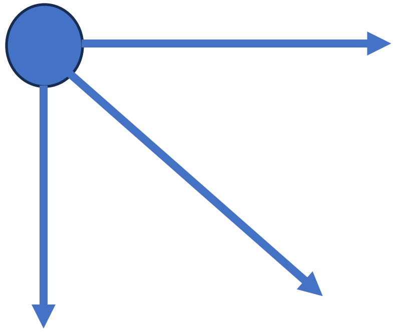
- Placing 8 queens on a chess board, so that none attacks other



- Formulation-1
  - A state is any arrangement of 0 to 8 queens on board
  - Operators add a queen to any square

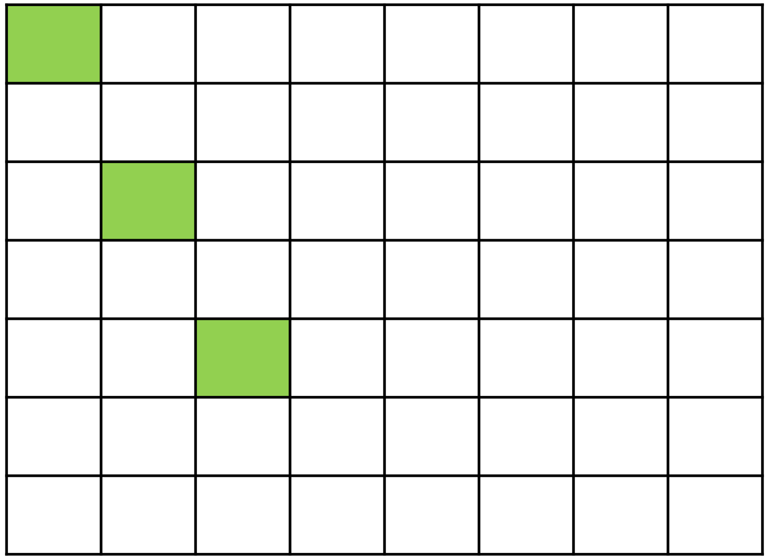
  - Unsystematic, putting anywhere
  - Not going row-wise or column-wise

8 queens problem



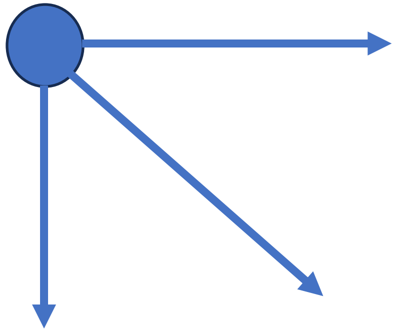
- Placing 8 queens on a chess board, so that none attacks other

- Formulation-2
    - A state is any arrangement of 0 to 8 queens with none attacked
    - Operators place a queen in the leftmost empty column



- Slightly more systematic

8 queens problem



- Placing 8 queens on a chess board, so that none attacks other

- Formulation-3
    - A state is any arrangement of 8 queens, one in each column
    - Operators move an attacked queen to another square in the same column

- Iterative refinement
- How to systematically solve such problems?
    - Good formulation reduces search effort
    - Bad formulation increases search effort

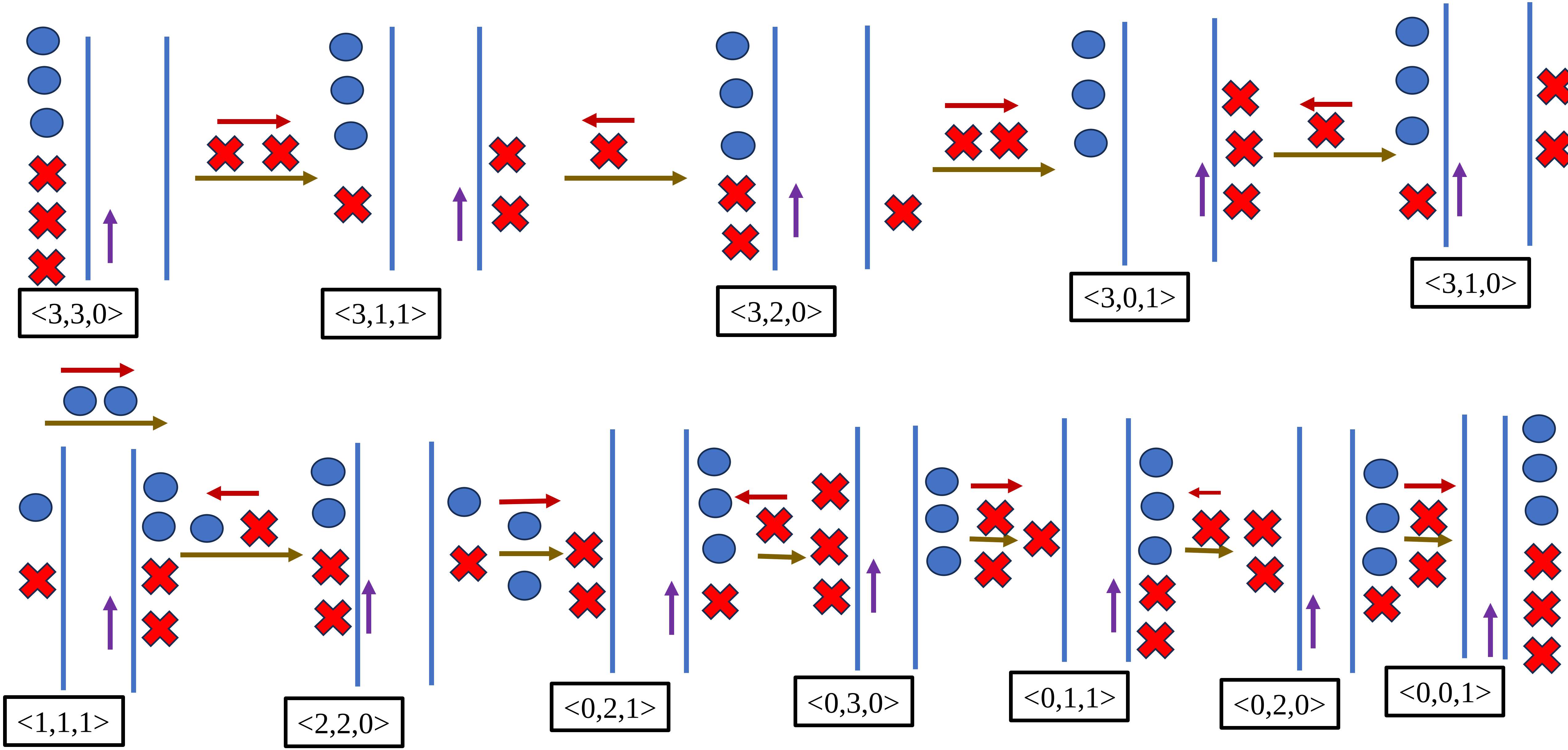
Missionaries and Cannibals

- Three missionaries and three cannibals are on one side of a river

- A boat that can hold one or two people

- Find a way to get everyone to the other side,
  - without ever leaving a group of missionaries outnumbered by cannibals

Missionaries and Cannibals



<3,3,0>

<3,1,1>

<3,2,0>

<3,0,1>

<3,1,0>

<1,1,1>

<2,2,0>

<0,2,1>

<0,3,0>

<0,1,1>

<0,2,0>

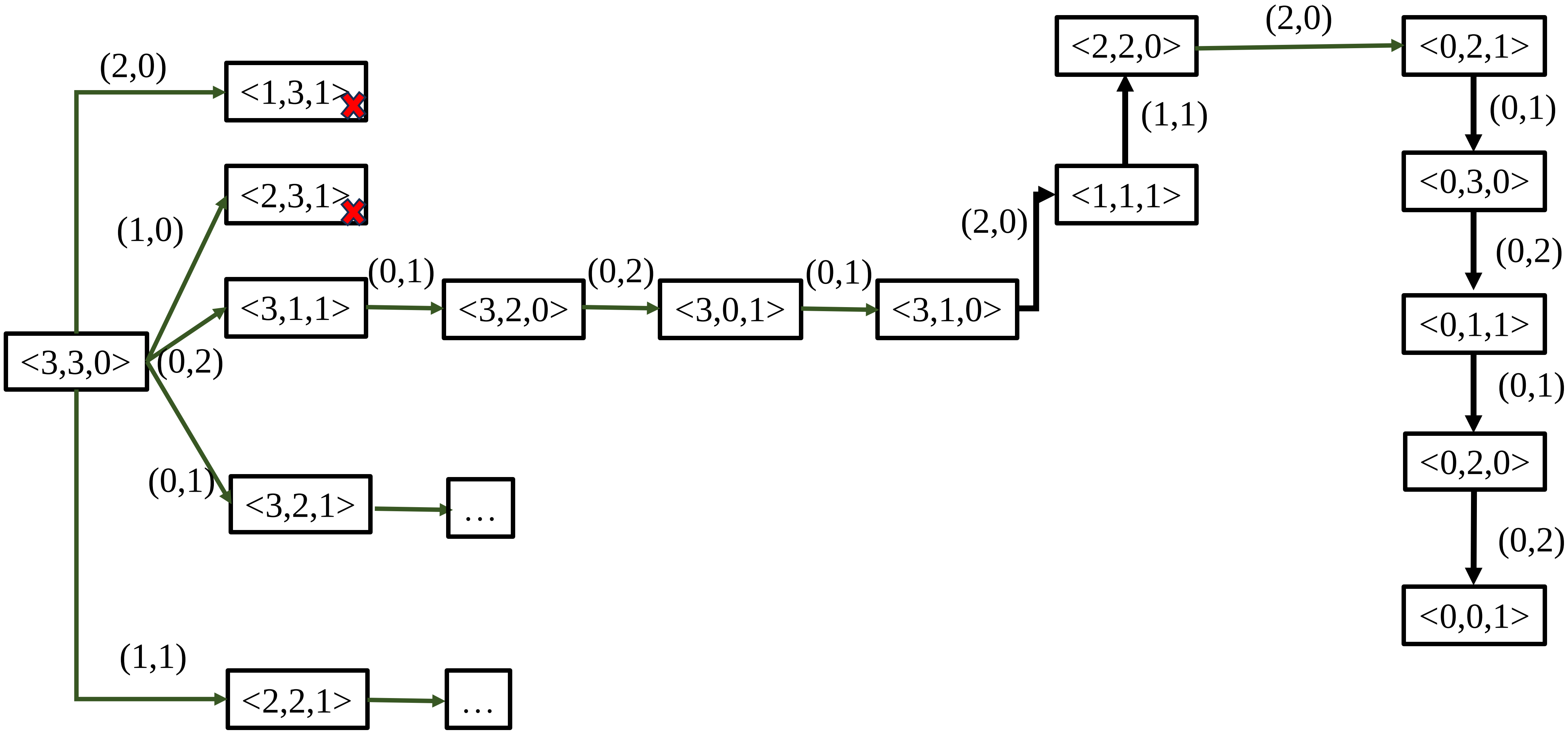
<0,0,1>

Missionaries and Cannibals

- State: (#m, #c, 1/0)
- #m: Number of missionaries in the first bank
- #c: Number of cannibals in the first bank
- The last bit indicates whether the boat is in the first bank

- Start state: (3,3,0)
- Goal state: (0,0,1)
- Operators: Boat carries (#missionaries, #cannibals)
- (1,0), (0,1), (1,1)
- (2,0), (0,2)

Missionaries and Cannibals: Search



Search: Challenges

- Is the search space a graph or tree?
    - While exploring the states same state may appear multiple times

- Is it important to identify such repetitive states?
    - Yes
    - It may lead to infinite loop

Thank You