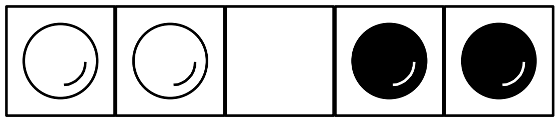
Well-defined formulation of the stone puzzle:



states: square content - 5 variables, 3 values each
white (O), black (X), empty (–)

initial state: (O O – X X)

goal test: (X X – O O)

operators:

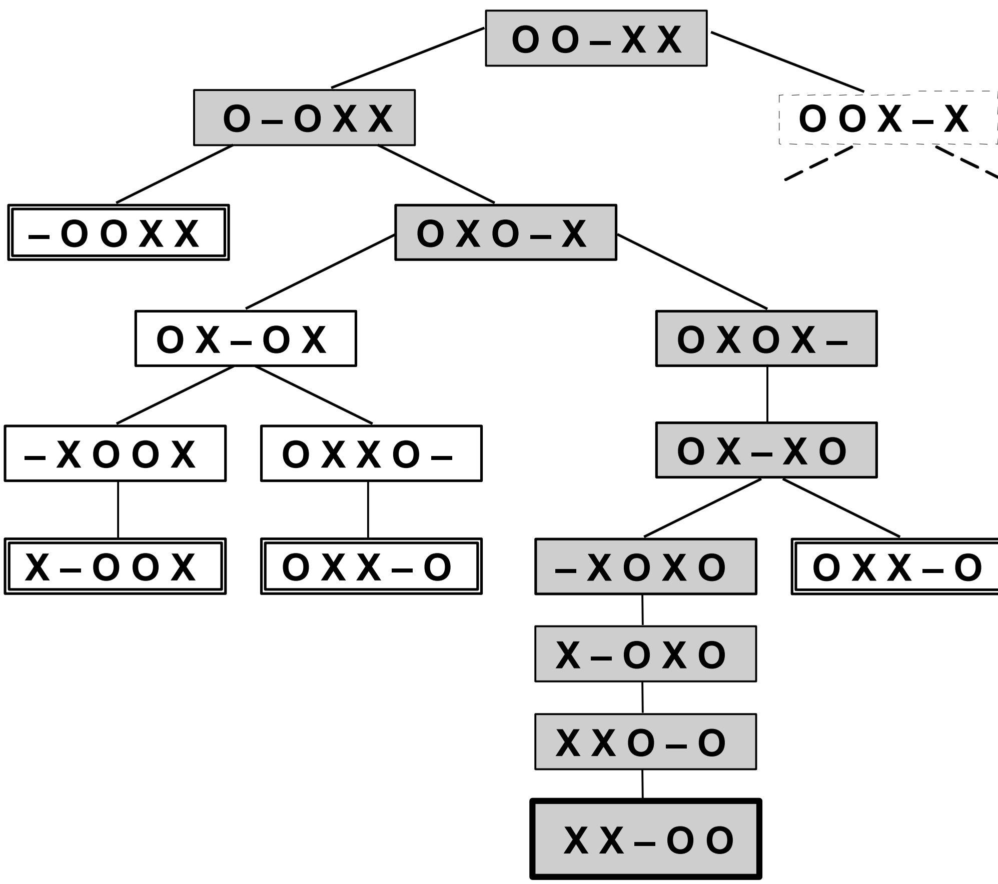
- MoveToRight: (O –) → (– O)
- MoveToLeft: (– X) → (X –)

- JumpToRight: (O X –) → (– X O)
- JumpToLeft: (– O X) → (X O –)

path cost: number of operators used (1 for all ops)

Problem search tree and solution:

- valid, reachable states only (subset of the state space)
- symmetric portion of the search tree not shown



Characteristics of the search space:

nb of branches: $2 * 15 = 30$
non-terminal nodes: $1 + 2 * 10 = 21$

average branching factor: $30 / 21 \approx$ 1.43

depth of the 2 solutions: 8

space complexity:

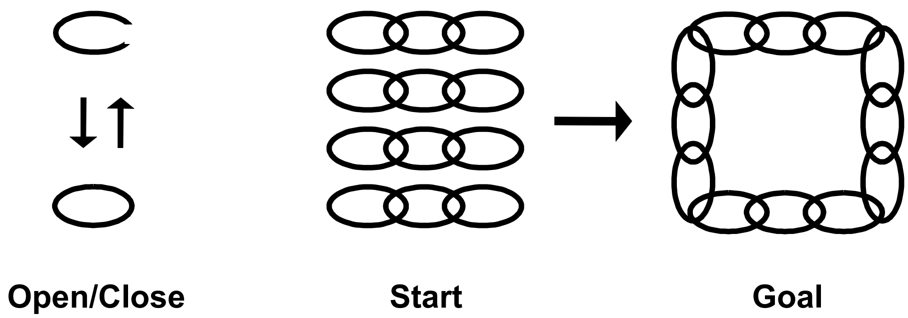
- actual space required = 31 nodes
- theoretical = $1 + 1.43 + 1.43^2 + \dots + 1.43^8 \approx 55$

Most suitable search algorithm:

(note: for small problems, *any* algorithm will do!)

- heuristic function? no → non-informed search
- any solution ok? low branching factor → DFS
- optimal solution? low branching factor → BFS,
- variable operator cost? → UCS

Formulation of the chain problem:



states: - set of n chains
- chains of k links, circular or not (l = 0 or 1)
- links open or closed (c = 0 or 1)

→ { … (k, l, c) … } (note: c=0 for k>1)

initial state: { (3,0,0) (3,0,0) (3,0,0) (3,0,0) }

goal state: { (12,1,0) }

operators: OS: *open a single link*
"*open*" (1,0,0) → (1,0,1)

OE: *open a link at the end of a chain*
(k,l,0) → (1,0,1) + (k–1,0,0)k > 1

OM(m): *open a link in the middle of a chain*
(k,0,0) → (1,0,1) + (m,0,0) k > 2
+ (k–m–1,0,0) k–1>m>0

operators: CS: *close a single link*
"*close*" (1,0,1) → (1,0,0)

CE(*l*): *close a link at the end of a chain*
(1,0,1) + (*k*,0,0) → (*k*+1,*l*,0)

CM: *close a link in between two chains*
(*k*,0,0) + (*m*,0,0) + (1,0,1) → (*k*+*m*+1,0,0)

path cost: number of operators applied (1 for all ops)

Optimal solution to the chain problem:

{ (3,0,0), (3,0,0), (3,0,0), (3,0,0) }

OM(1): { (3,0,0), (3,0,0), (3,0,0), (1,0,1), (1,0,0), (1,0,0) }

OS(): { (3,0,0), (3,0,0), (3,0,0), (1,0,1), (1,0,1), (1,0,0) }

OS(): { (3,0,0), (3,0,0), (3,0,0), (1,0,1), (1,0,1), (1,0,1) }

CM(): { (7,0,0), (3,0,0), (1,0,1), (1,0,1) }

CM(): { (11,0,0), (1,0,1) }

CE(1): { (12,1,0) } *6 steps only*