# Heuristics cont'd

Friday, September 16, 2016 10:31 AM

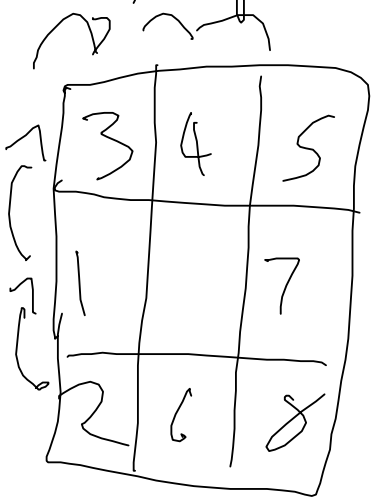
- Consistent heuristics are admissable
  - not always true other way
  - Mostly true though

## Design

- More an art than science

- Some strategies for design

- Relax the problem




constraints

1) Can only move A if adj to B

~~2) A or B must be blank~~

- Remove constraint 2, Manhattan Distance

- Remove 1 & 2
  - # of tiles out of place
- Remove 1
  - Gaschnig's distance

# Heuristic Design

Friday, September 16, 2016 10:41 AM

Subproblems

- Find subproblem that's not too hard to compute
- Solve that
- 8 puzzle, solve $\frac{1}{2}$ tiles
- Provides lower bound on full problem complexity

- Make sure h() isn't too hard to calculate

Review

Monday, September 19, 2016 9:03 AM

$f(n) = g(n) + h(n)$

rank states on the open list

A* – admissible heuristic

## Handling shortcuts

```
WHILE (NOT isgoal(current) AND open ≠ NIL) DO:
    closed <— closed + {current}
    FOREACH n ∈ successors(current, ops) DO:
        IF n is not on open or closed THEN DO:
            compute g(n)
            Insert n into open (ranked on g())
        ELSE IF n is on open AND n is reached by a shorter path THEN DO:
            n.parent <— current
            update g(n)
            resort open
    END FOREACH
    current <— pop(open)
END WHILE
```

shortcuts

Best-First: h() can be admissible or not

## Handling shortcuts for Best-First

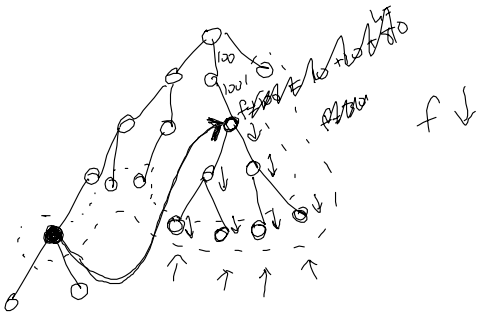
```
WHILE (NOT isgoal(current) AND open ≠ NIL) DO:
    closed <— closed + {current}
    FOREACH n ∈ successors(current, ops) DO:
        IF n is not on open or closed THEN DO:
            compute f(n)
            Insert n into open (ranked on g())
        ELSE IF n is on open AND not on closed AND n is reached by a shorter path THEN DO:
            n.parent <— current
            update f(n)
            resort open
        ELSE IF n is on closed AND n is reached by a shorter path THEN DO:
            n.parent <— current
            update f(n)
            Propagate new f-values to all descendants using DFS
                (stop generating successors if a node is on open)
    END FOREACH
    current <— pop(open)
```

shortcuts

hasn't generated successors yet

start w/ changed parent

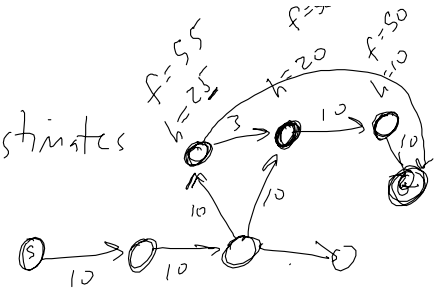
Depth bounded



END WHILE

Admissible

h() never overestimates



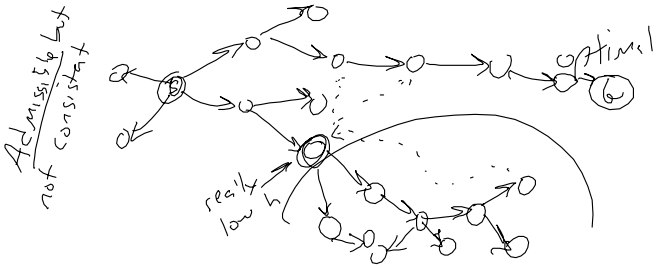

If you don't have admissibility → Non-optimal alg.

Consistency → Admissible

Admissible h under-estimates

Consistent h under-estimates too much growing up

$h(A) - h(B) \leq K(A,B)$

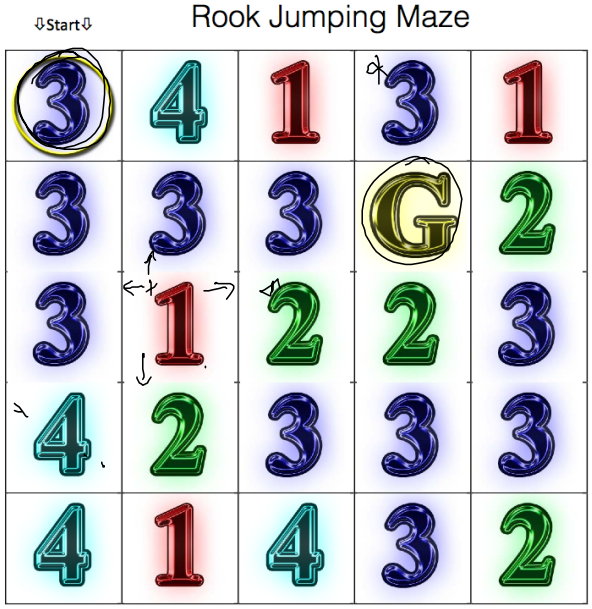



h(n) = 0 for all n ← Admissible

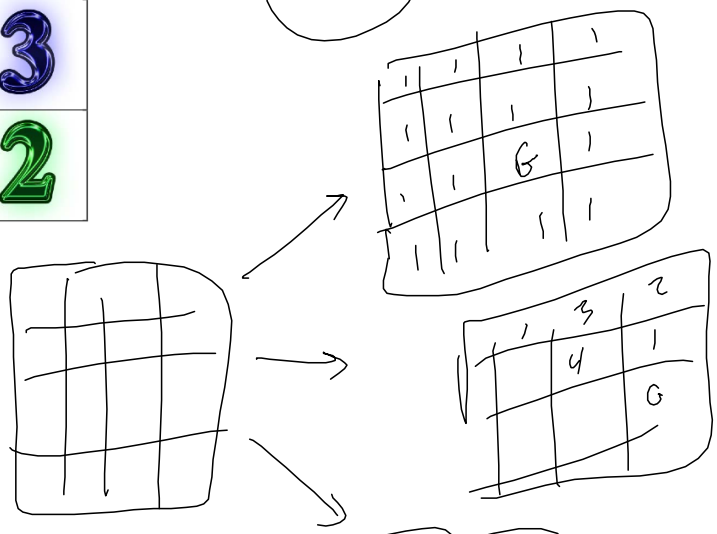
↳ ~~Breadth-first~~ Uniform-cost

# Rook Jumping

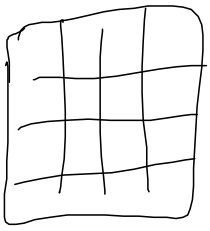
Monday, September 19, 2016 9:24 AM







Solution: DRLUDLRULLRDU



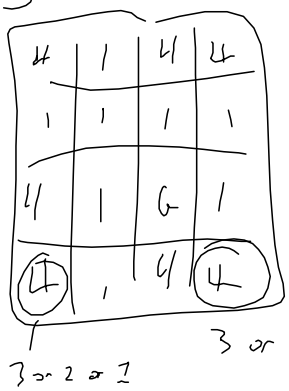
Complete config
set of cells
w/ numbers
(or empty)

state:

init: random state, or grid of 1s or empty grid

goal: (?)

Actions: picked a cell & changed the number

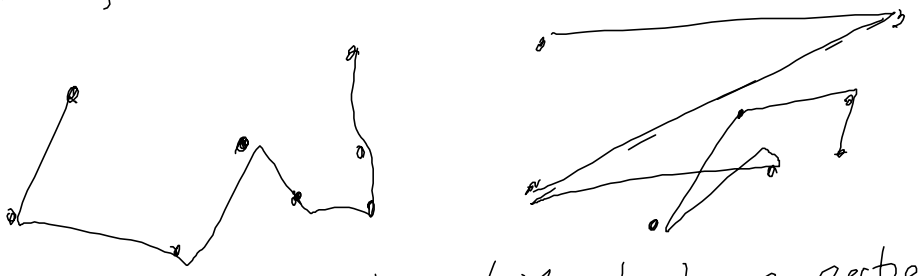


# Hill Climbing Search

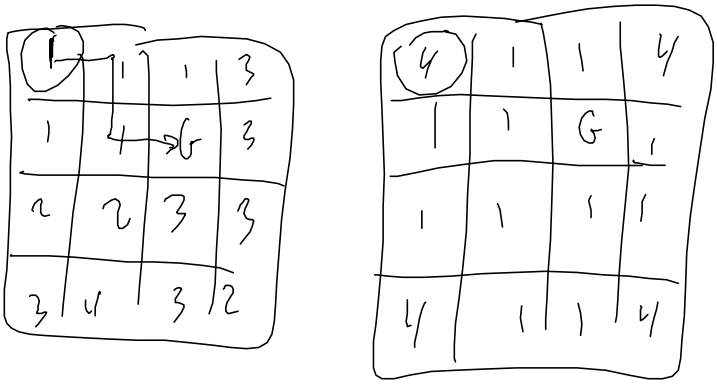
Monday, September 19, 2016 9:26 AM

state ✓
init ✓
goal ?
intuition that some states better than others

Traveling Sales Person

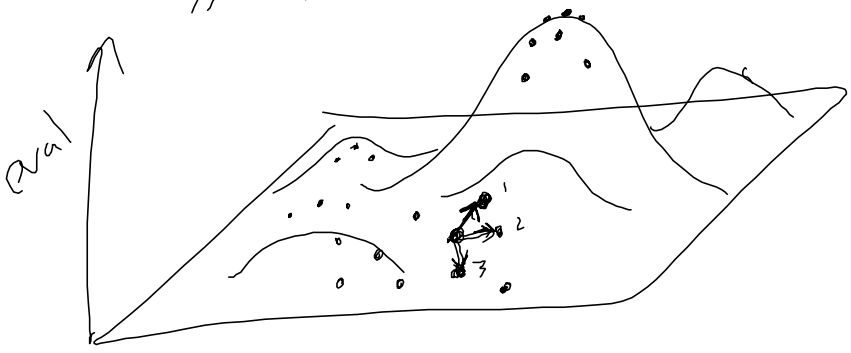


Search for the state w/ the best properties



---

intuition
evaluation fn () = captures this intuition
- bigger the # the better the state

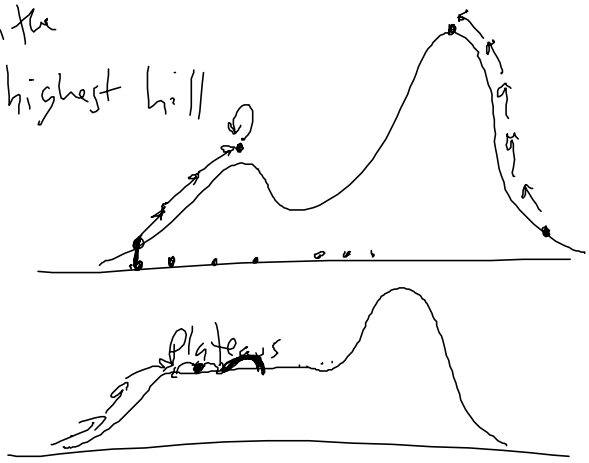



Blind Hill Climbing
- start w/ a random state

- Start with a [illegible]
- Generate K successors
- Sort successors by eval fn
- Move to best successor
- Repeat

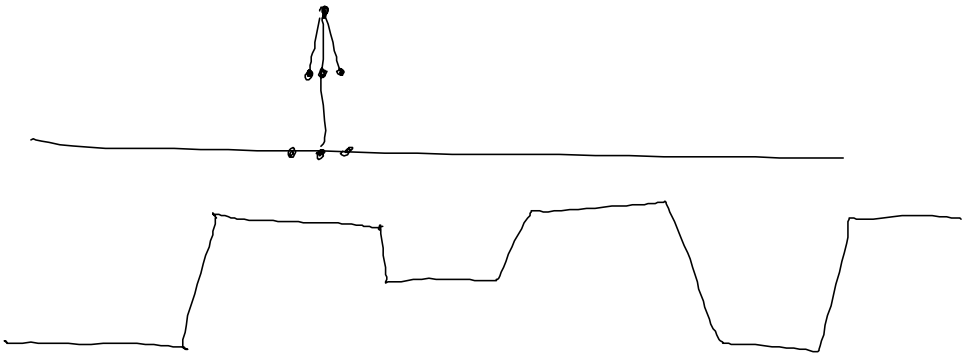
Problems:

- you might not be on the highest hill
- local maxima
- plateaus - ∞ loops
- No open or closed




Fixes!

- Random bad moves (down-hill moves)
- Multiple restarts (keep multiple states)
- Gradient measure
- Keep history (beam search for limited open list)



Genetic Algorithms (Game AI) 4731

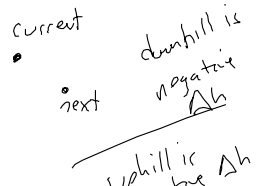
Simulated Annealing

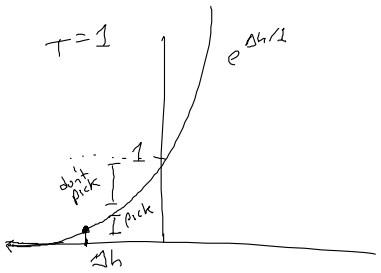
~~Successors~~ Neighborhood

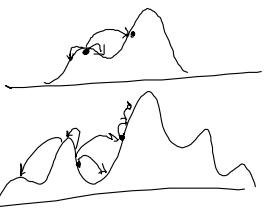
Simulated annealing

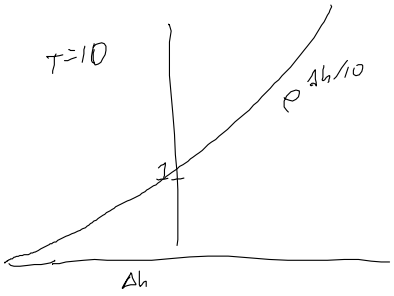
Monday, September 19, 2016 9:28 AM

1. T <— starting temperature
2. current <— random starting node
3. next <— pick a ~~random move and apply it~~ 1 neighbor
4. Δh <— h(next) - h(current)
5. If Δh > 0 then current <— next, goto 3
6. current <— next with probability $e^{\Delta h/T}$ or goto 3
7. Reduce T by some given schedule
8. If T > 0 goto 3, or terminate with current as solution

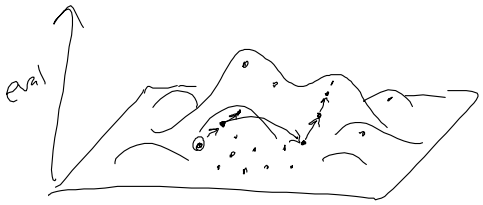


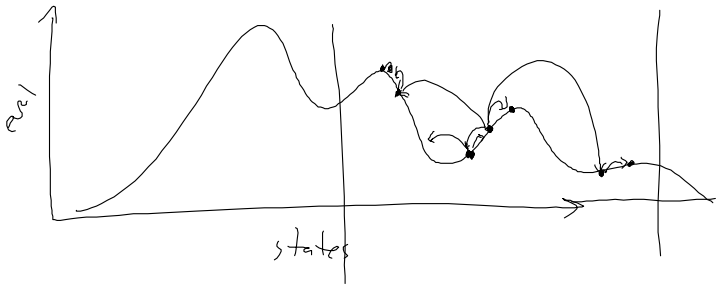



Hill-climbing




Random downhill move

Simulated Annealing
- Define a neighborhood
- If you find a successor better do it
- Prob you allow move downhill
  - ↳ decreases as time goes on




Temperature variable — decreases over time

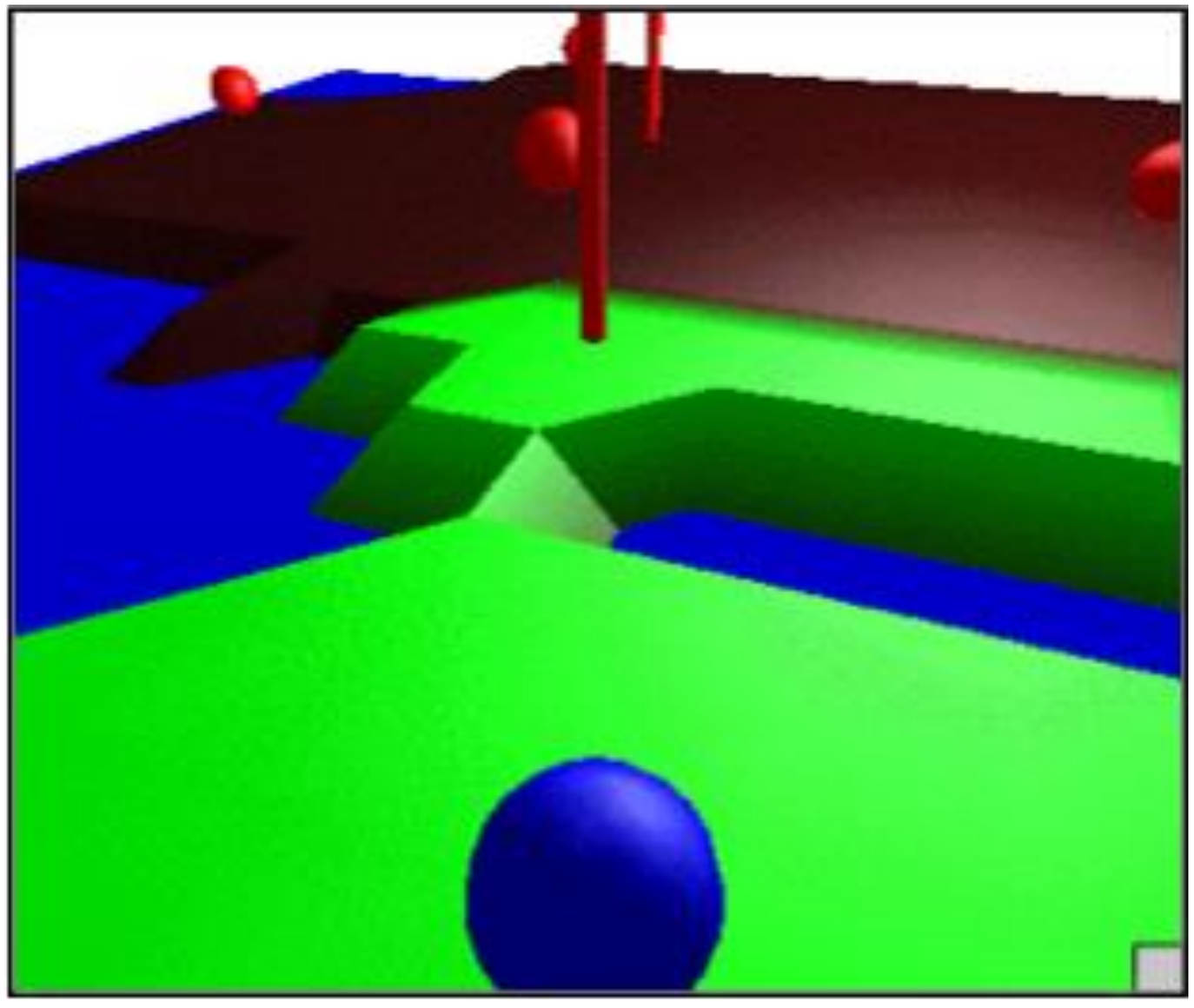
How fast to decrease?

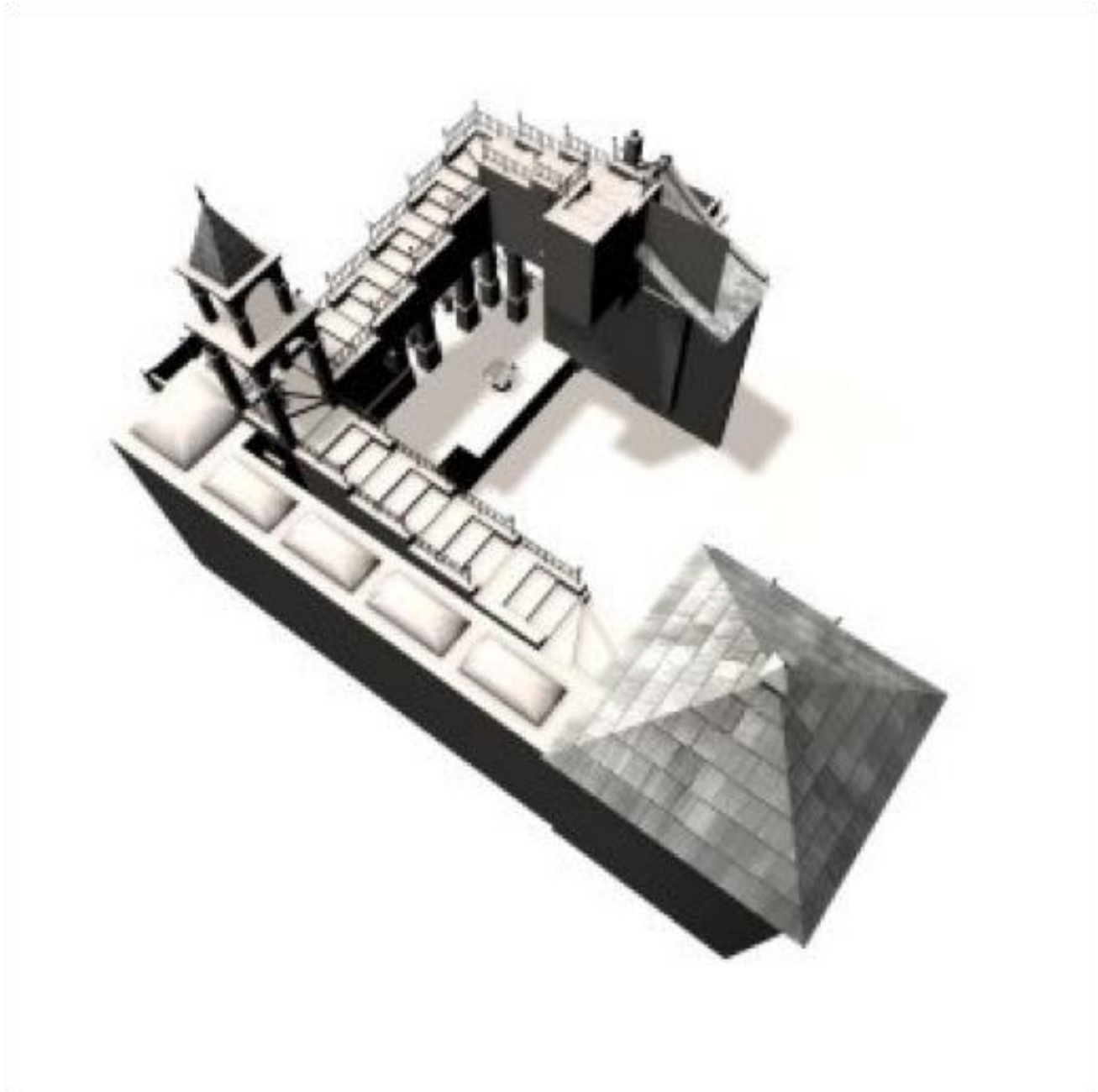
Tradeoff: time & quality of solution

Decrease slow enough → increase likelihood of a good solution

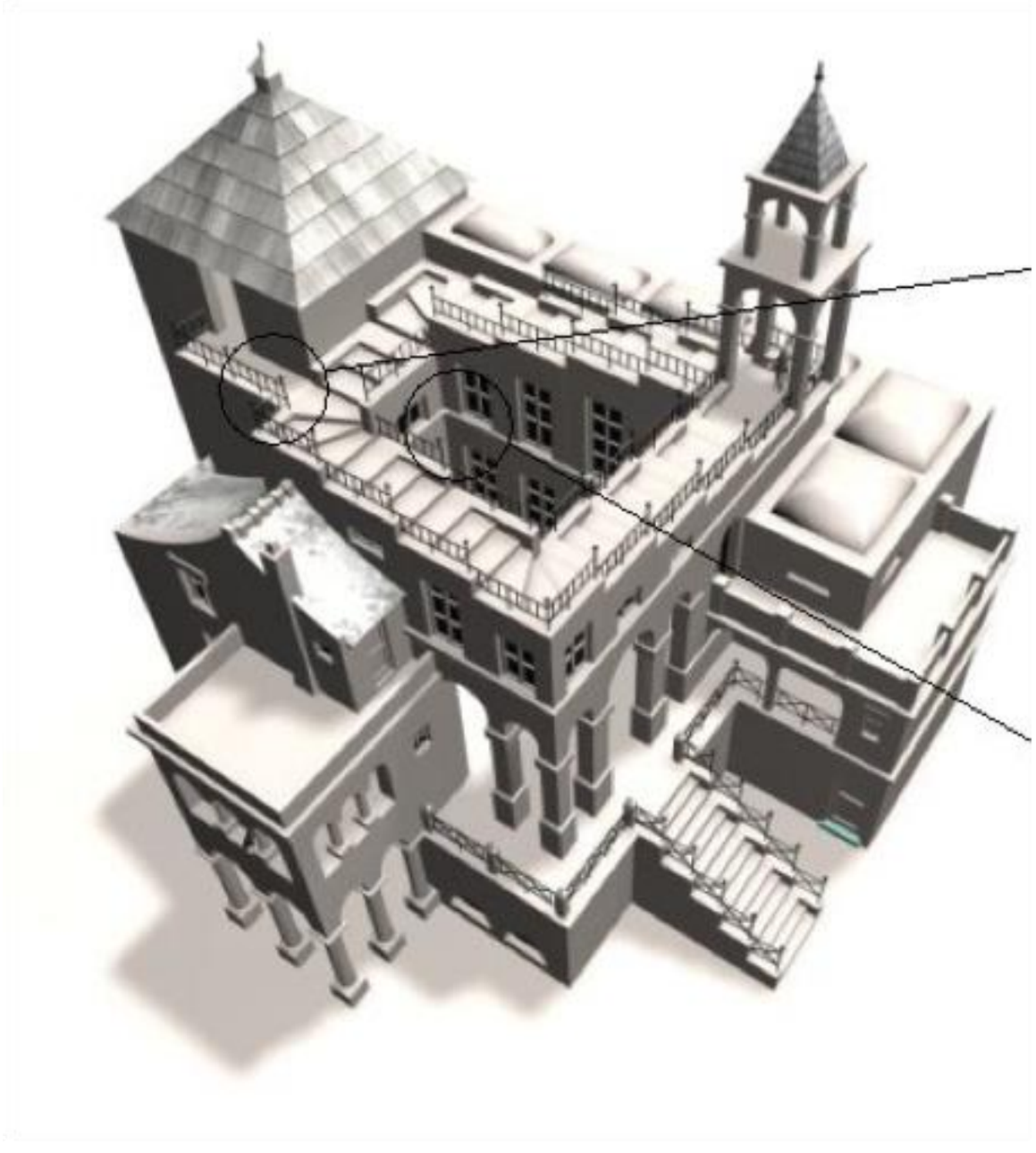
Satisficing Search - any alg where more time → better quality solution
- okay to return sub-optimal solutions

Just-in-time - Can be interrupted at any time and still return some solution





Halper and Olivier. (2000). Smart Graphics.




MARIO 03
COIN
07
WORLD
none
TIME
183