Monte Carlo Tree Search and Levy Flight Search: Solutions to Search and Detection

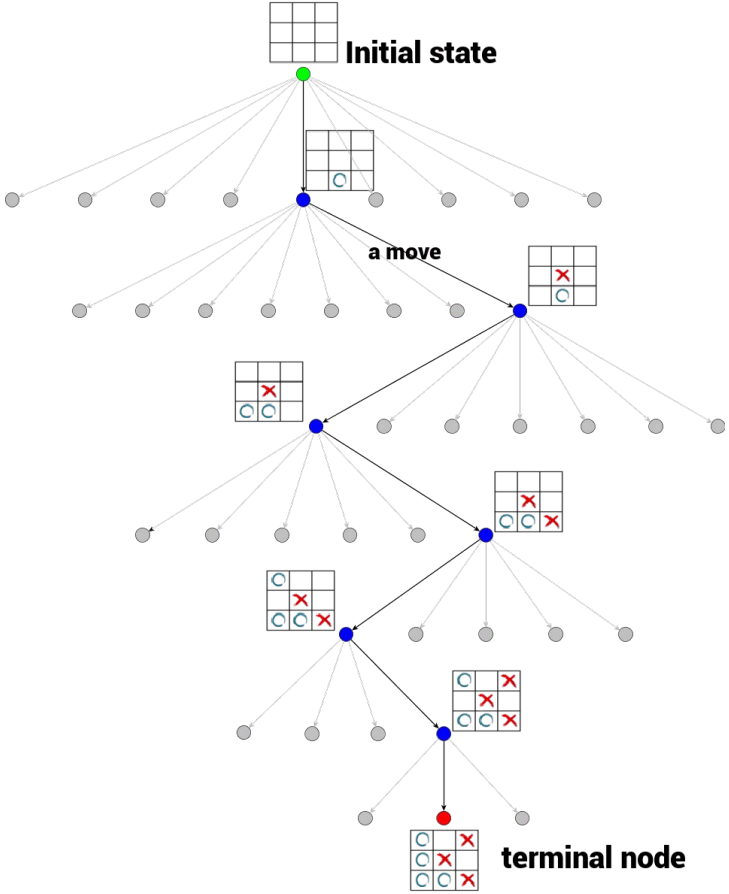
Elana Kozak
United States Naval Academy

Motivation: Game AI

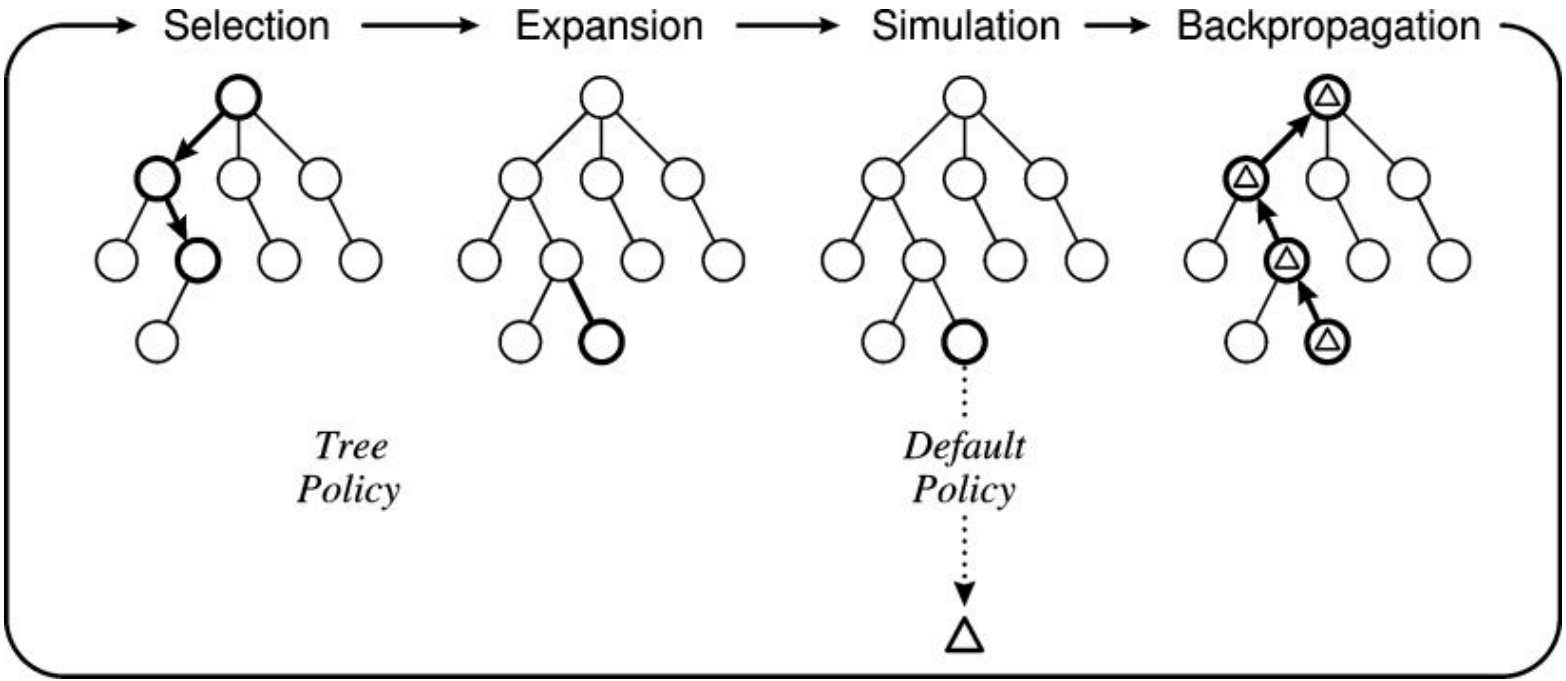
How does a computer choose its moves for a game like chess or tic-tac-toe?

With so many possible game playouts it can't possibly know the effects of each move…. Right?

Monte Carlo Tree Search provides a solution!



Monte Carlo Tree Search Overview



Selection Policy:
Upper Confidence Bound for Trees (UCT)

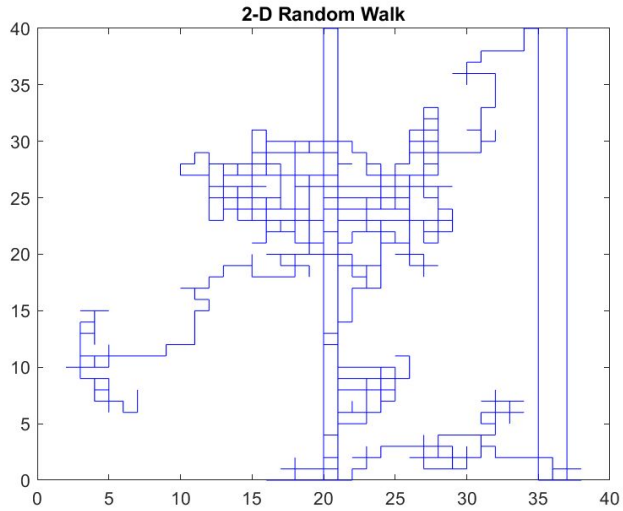
$$\mathbb{UCT}(v_i, v) = \frac{Q(v_i)}{N(v_i)} + c\sqrt{\frac{\log(N(v))}{N(v_i)}}$$

V_i: node
V: parent node
Q: win count
N: visit count
C: exploration constant

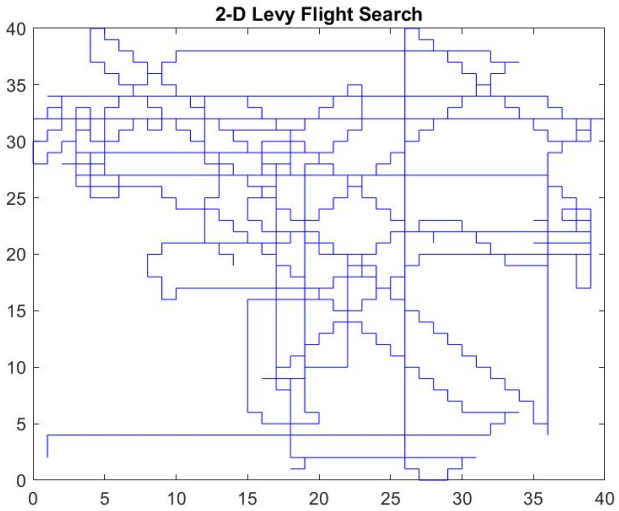
After each iteration, the program chooses the v_i that maximizes UCT.

Default/Rollout Policies

Random Walk



Levy Flight Search

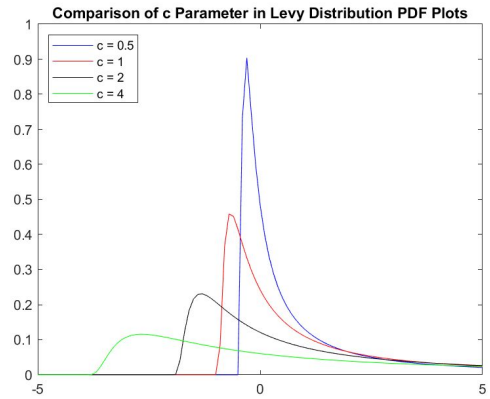


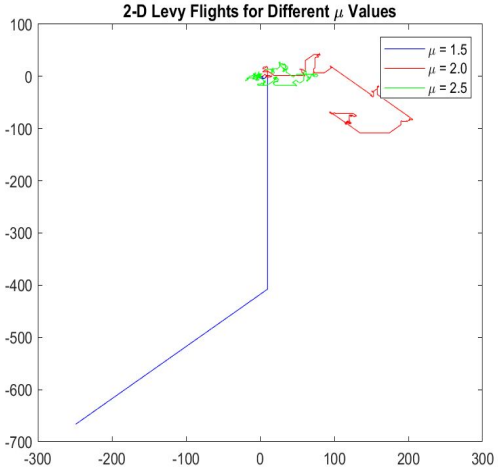
Levy Flight Search: Definition and Distribution

- Levy Distribution: a special case of the stable distribution with α=½, β=1, scale parameter *c* and shift parameter *m*
- Levy Flight Search (LFS) draws its step lengths from a stable distribution with μ=α+1 (not really Levy)

$$P(l_j) \sim l_j^{-\mu} \qquad 1 < \mu \leq 3$$

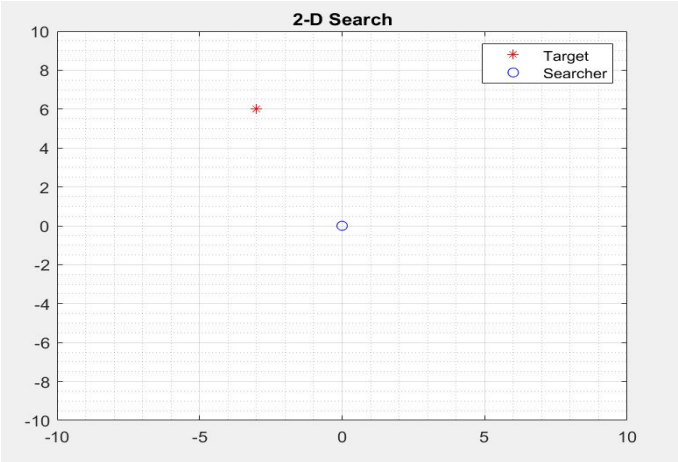
- Direction is chosen randomly
- Path is characterized by many small steps with a few large jumps

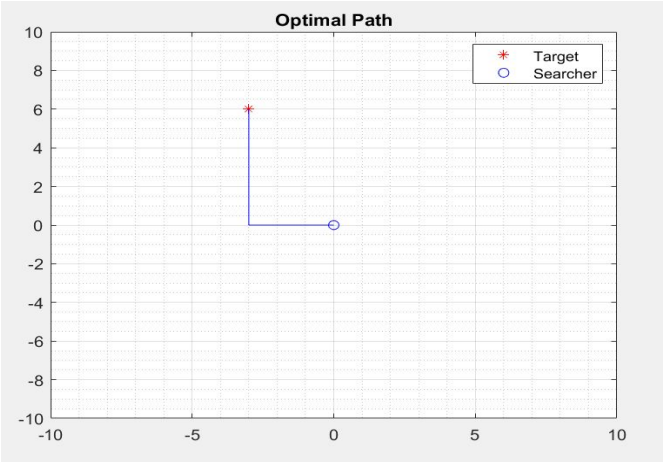




Single Target Search Problem

Goal: find an efficient path to the target





2-D Lattice with periodic boundaries

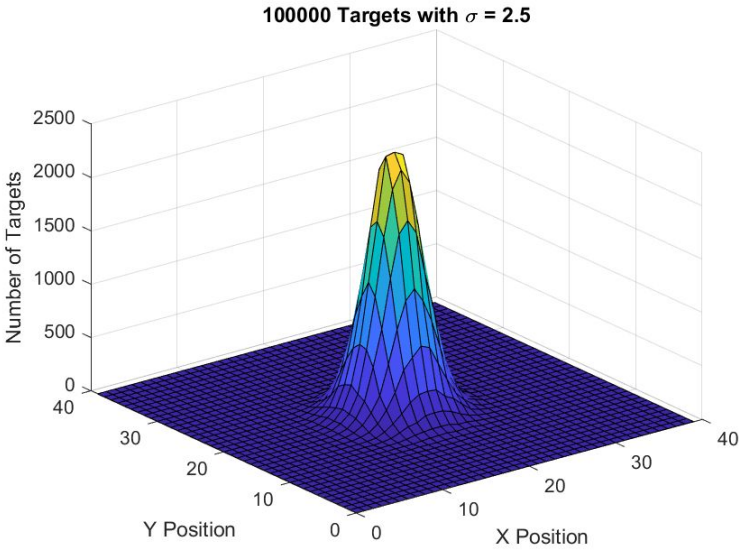
Methodology

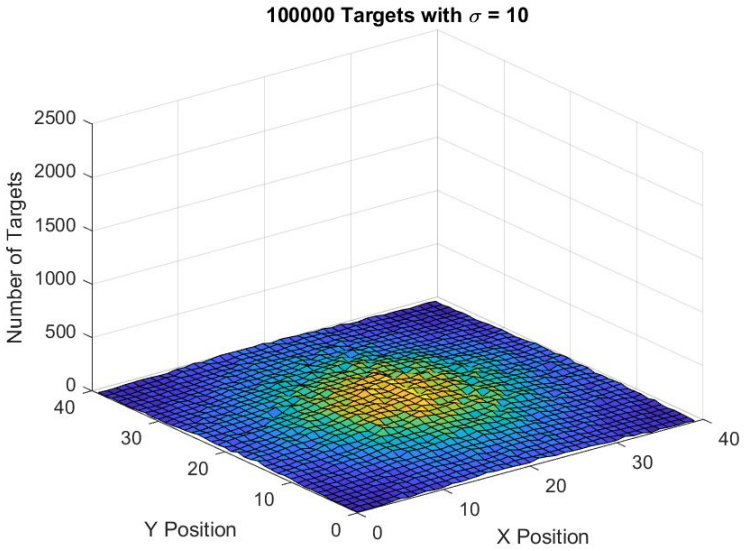
1. Set parameters
2. Place target according to distribution
3. Start searcher at (1,1) and calculate optimal steps to target
4. Use the MCTS method to play out simulations/loops (L times)
    a. Place simulated target according to same distribution
    b. Choose move based on UCT algorithm
    c. Use default policy to find target
5. Choose best move and update searcher location
6. Repeat steps 4-5 until distance is less than vision radius (Rv)
7. Repeat steps 2-5 for desired number of trials (T)

Parameter Legend

N = grid size
T = # of trials
L = # of loops
Rv = vision radius
C = exploration constant
Limit = forced stop

Single Target Gaussian Distributions: Target Density Plots





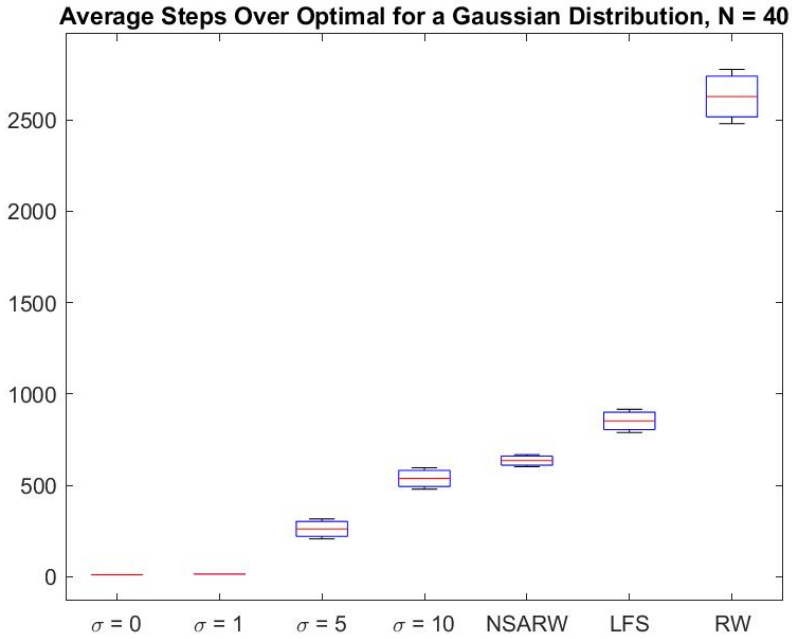
Smaller σ means a more well-defined (a.k.a. “known”) target, large σ means a more random (a.k.a. “unknown”) target.

Simulation Results

As sigma gets larger, the MCTS approaches a nearly self-avoiding random walk.

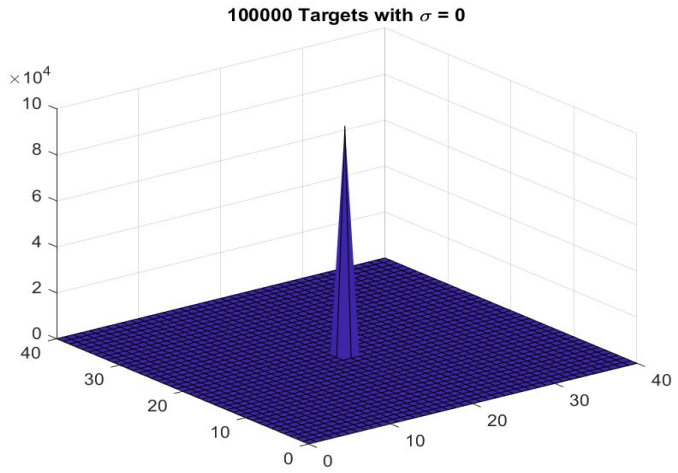
MCTS always outperforms a truly random walk or a generic Levy Flight Search.

Note: for these results and all following, N=40, Loops=100, Trials=1000, c=2, and rv=1. The graph shows confidence intervals of 95%.

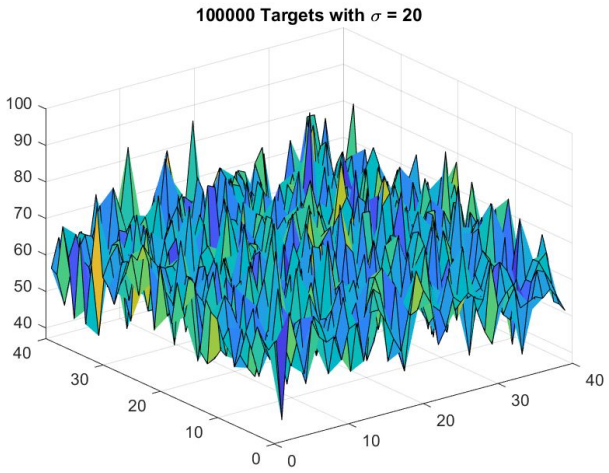


Gaussian Distribution: Special Cases

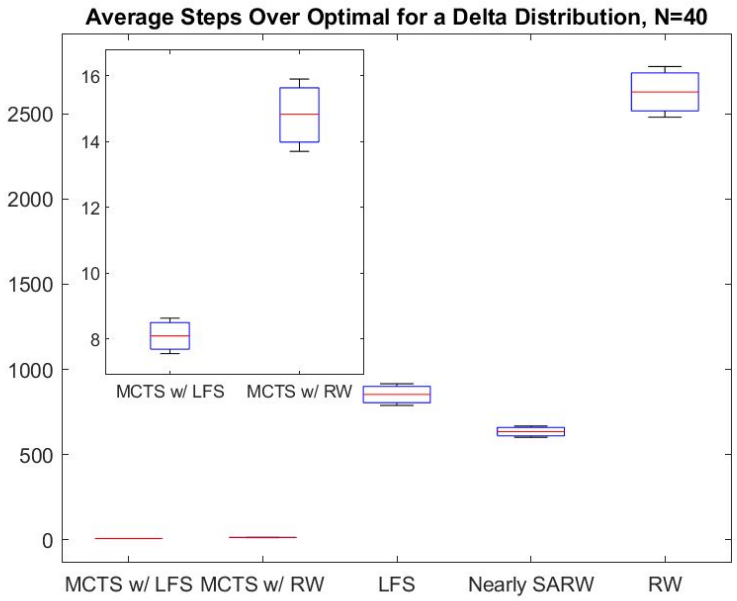
Delta Distribution

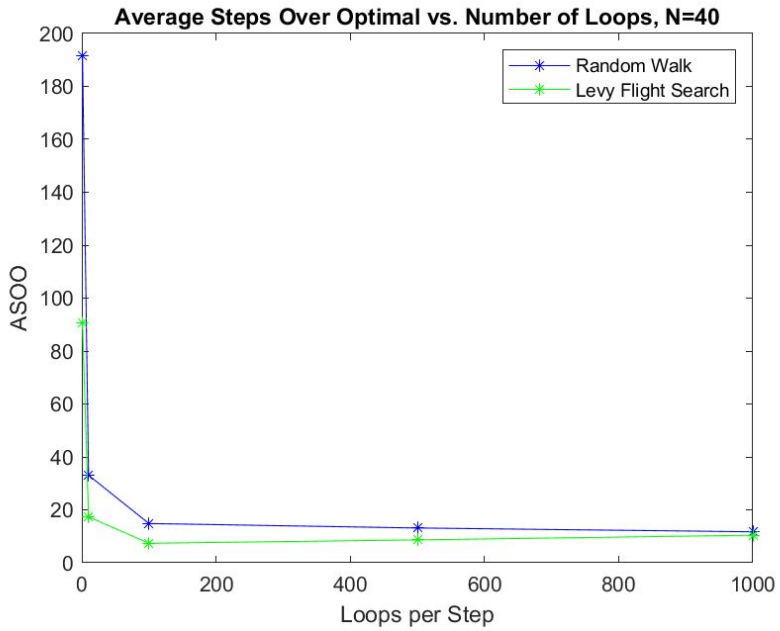


Uniform Distribution



Delta Distribution Simulation Results





*Data for 90% Confidence Intervals

Delta Distribution: Theoretical Results

Theorem: As the number of loops goes to infinity, a Monte Carlo Tree Search with UCT algorithm will converge to an optimal path.

Main idea of proof:

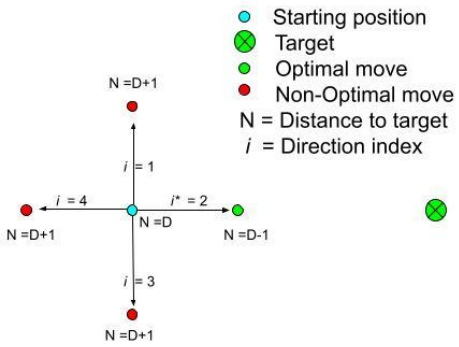
$$UCT(v_i, v) = \frac{Q(v_i)}{N(v_i)} + c\sqrt{\frac{log(N(v))}{N(v_i)}}$$

Central Limit Theorem: after many trials the average steps to target from a certain distance will be the expected value for that distance.

Define: X_i= steps to target after taking move *i*, Y_i= inverse of average X_i

$p_{i,l}$ = probability of choosing move *i* after *l* loops

Delta Distribution: Theoretical Results



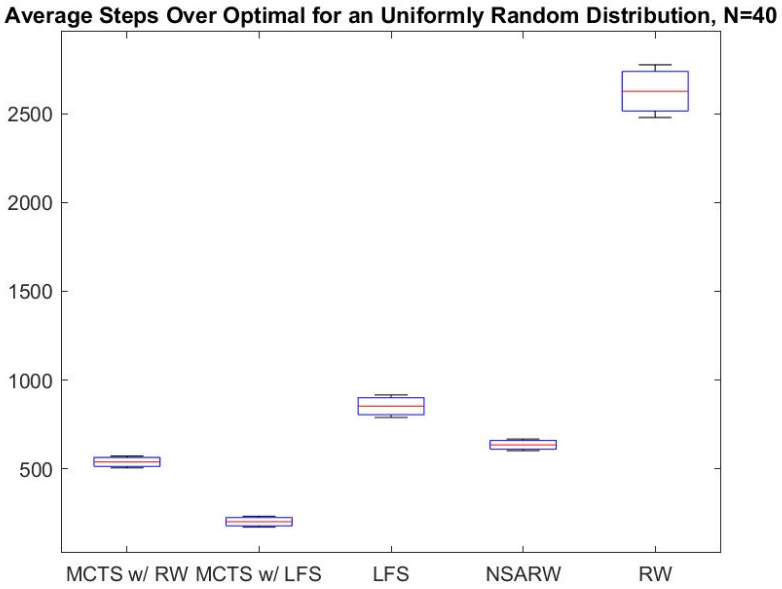
Proof (continued):

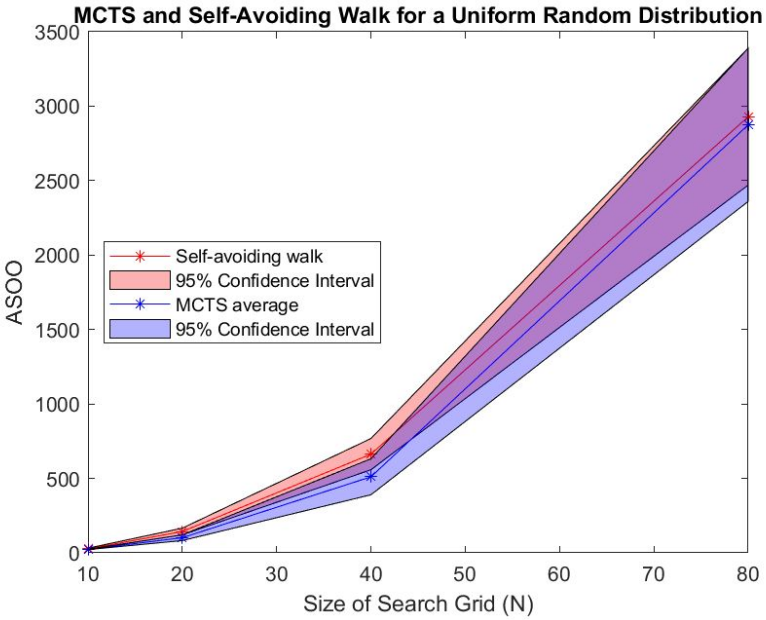
Probability of choosing move *i* depends on average step count. $p_{i,l} = \mathbb{P}(Y_{i,T_{i,T_i(l)}} \geq Y_{*,T_*(l)})$

For a sufficiently large number of loops, the probability of selecting the wrong move will be small. $p_{i,l} \leq \mathbb{P}(X_{*,T_*(l)} \geq E[\tau_{D-1,N}] + \epsilon) + \mathbb{P}(X_{i,T_i(l)} \leq E[\tau_{D+1,N}] - \epsilon)$

As the loops increases, this probability will go towards zero. Thus, for each step, the algorithm will choose a move that decreases its distance from the target by 1 (i.e., the optimal move).

Uniform Distribution Simulation Results





Uniform Distribution Theoretical Results

Theorem: As the search grid size goes to infinity, Monte Carlo Tree Search will converge to a nearly self avoiding random walk.

Main idea of proof:

When N is large enough, probability of target being near searcher is very small so average steps to target for each direction (reward function) will be roughly equal.

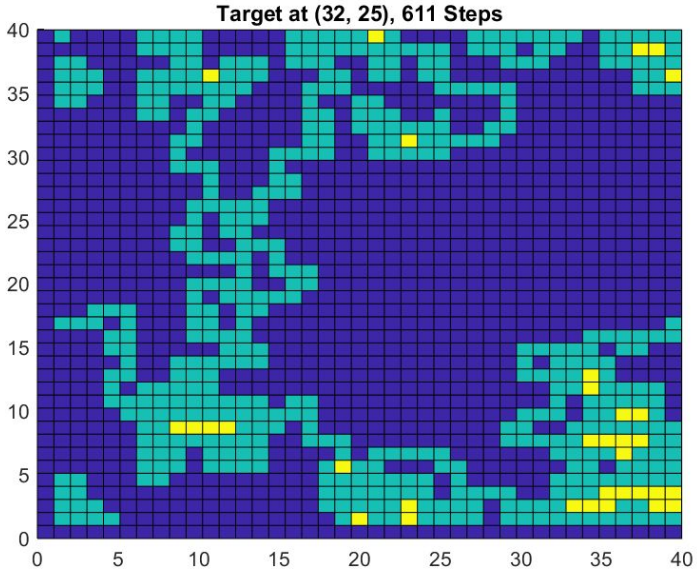
$$UCT(v_i, v) = \cancel{\frac{Q(v_i)}{N(v_i)}} + c\sqrt{\frac{log(N(v))}{N(v_i)}}$$

Uniform Distribution: Theoretical Results

Proof (continued):

Exploration term will be smaller for nodes already visited.

Thus, the algorithm will choose randomly between the nodes that have been visited an equal number of times but will avoid repeat visits.

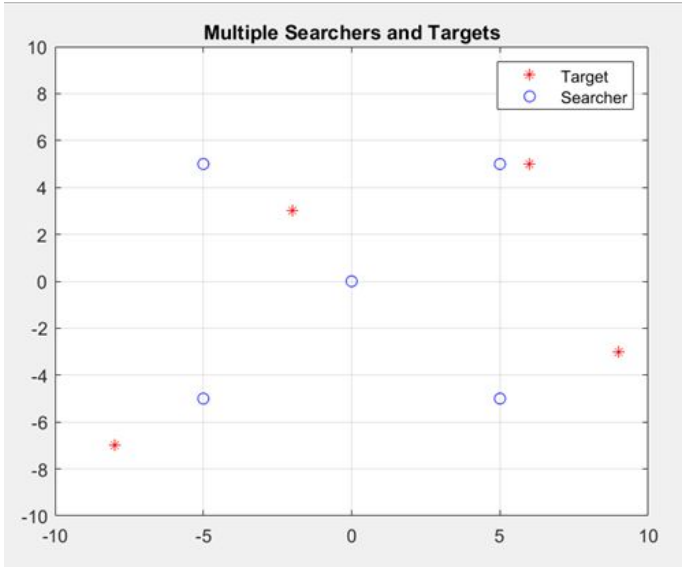


Summary of Findings

1. Monte Carlo Tree Search is a good choice for a known target scenario
    a. Theorem 1: For a delta target distribution, as the number of loops goes to infinity, the searcher steps over optimal goes to zero.
    b. This is promising for future applications!
2. Monte Carlo Tree Search is not a great choice for an unknown target scenario
    a. Theorem 2: For a uniform target distribution, as the search grid size goes to infinity, the search path converges to a nearly self-avoiding random walk.
    b. More computationally efficient to use a self-avoiding random walk!

Next Steps

- Multi-target scenario
  - Destructive, non-destructive, and regenerative
- Multiple search agents
  - Swarm behavior analysis



Thank You

contact: m213624@usna.edu