# Observation Sensitive MCTS for Elevator Transportation

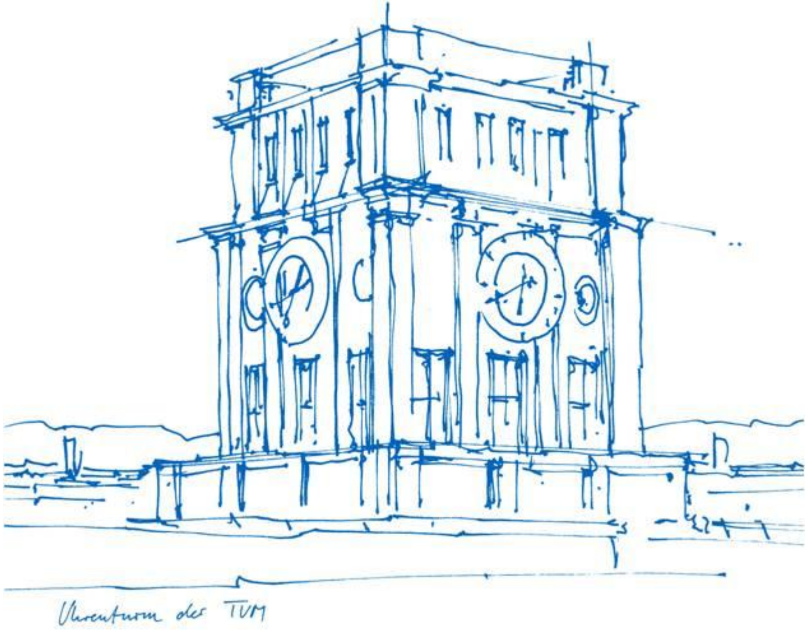
**Group 5:** Maximilian Rieger, Tim Pfeifle

Advisor: Sebastian Bachem

Technical University Munich

Advanced Deep-Learning in Robotics

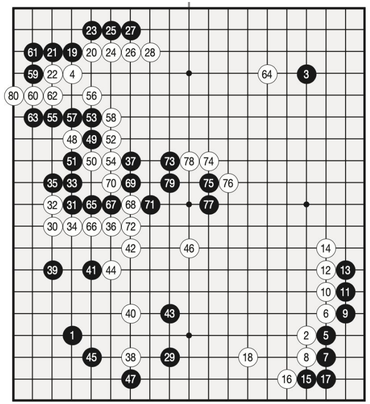
Munich, 18. June 2020



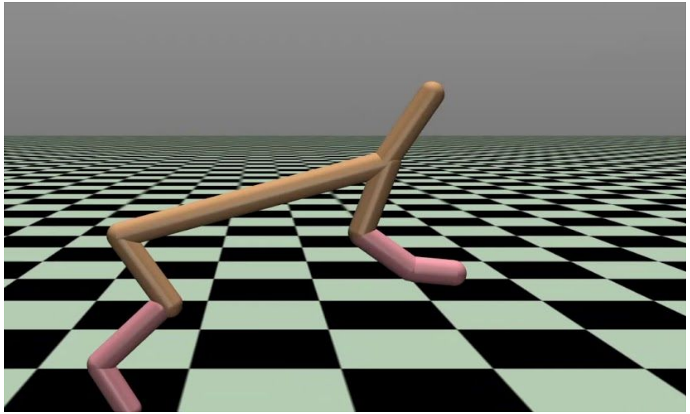
# Motivation

**Combination of Search Algorithm with RL shown to be successful in AlphaZero** [1]

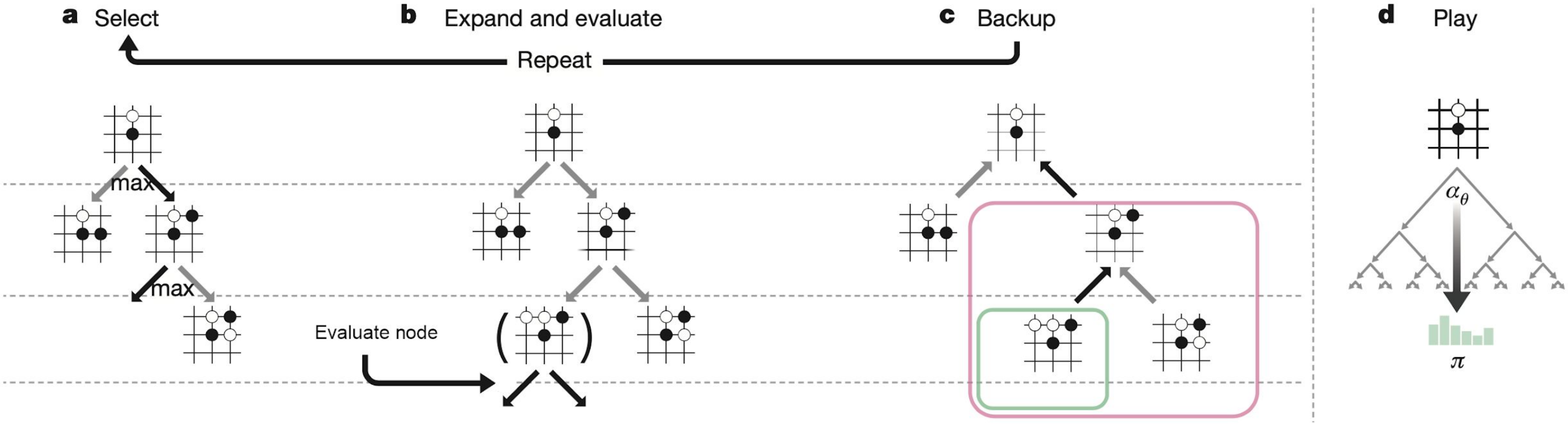
AlphaZero relies on **simulated** 2-player-games with a reward of {-1, 1} **at the end**



Our approach: use search for **simulated** problems with **continuous rewards**



# Monte-Carlo-Tree-Search (AlphaZero [1])



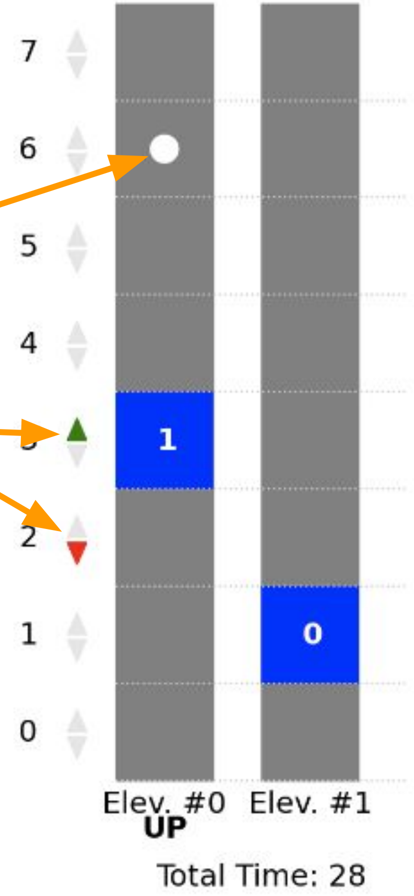

Our Modification: Use observed rewards at each step

# Elevator Transportation

**Example task with Continuous rewards**

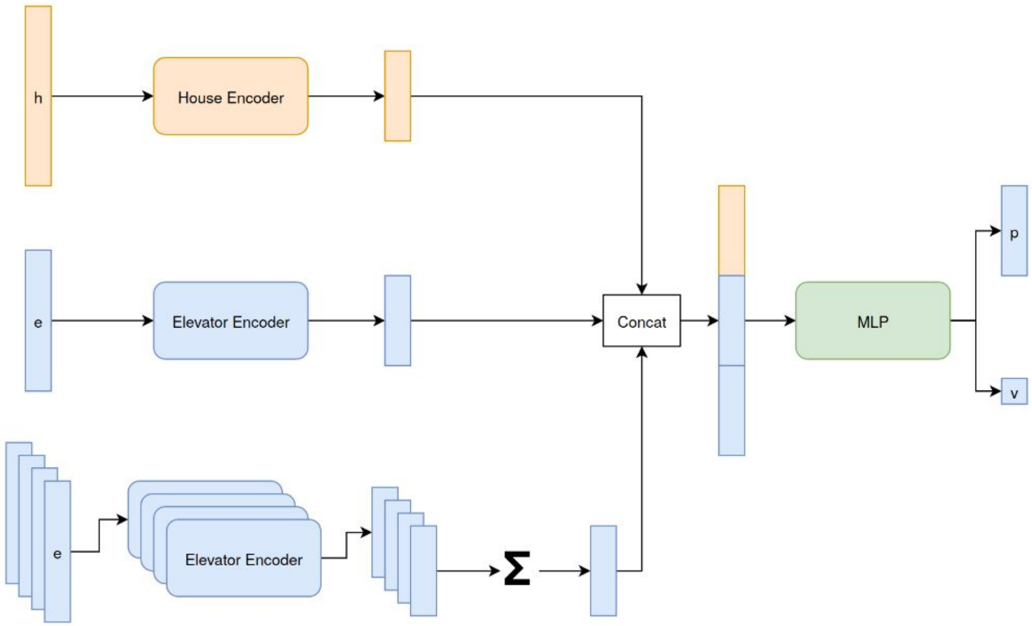
Passenger-Requests

Elevator-Requests




- 3 Actions per Elevator:
  - Up
  - Down
  - Open
- **Goal: Minimize Passenger Waiting Time**
- **Difficulties:**
  - State not fully observable!
  - Large exploration space

# Model Architecture

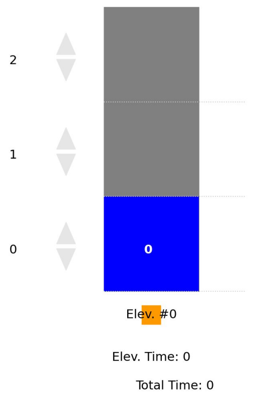


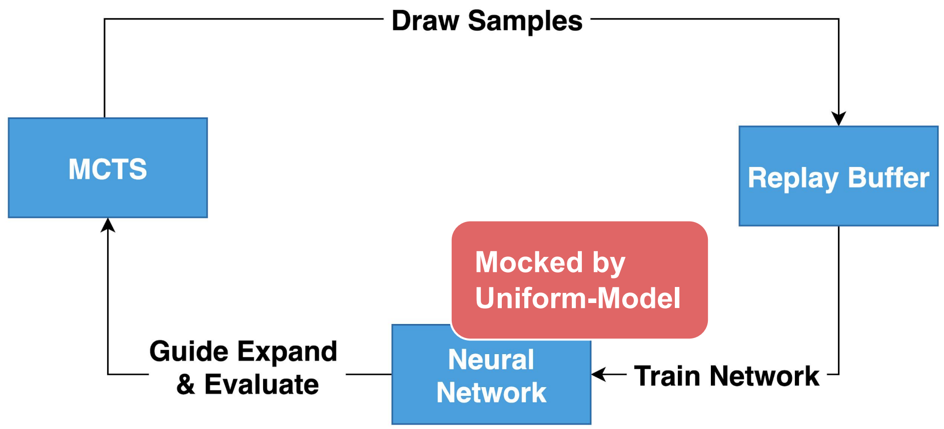
# Ranked Reward [2]

- Scaling to a range of [-1, 1] makes training value net a lot easier
- Achieving a good rank becomes gradually harder
- Resembles improving opponent in AlphaZero

$$\text{Ranked reward} = \begin{cases} +1 & \text{if result is better than 75\% of previous} \\ -1 & \text{else} \end{cases}$$

# Experiments: **Pure MCTS without NN**

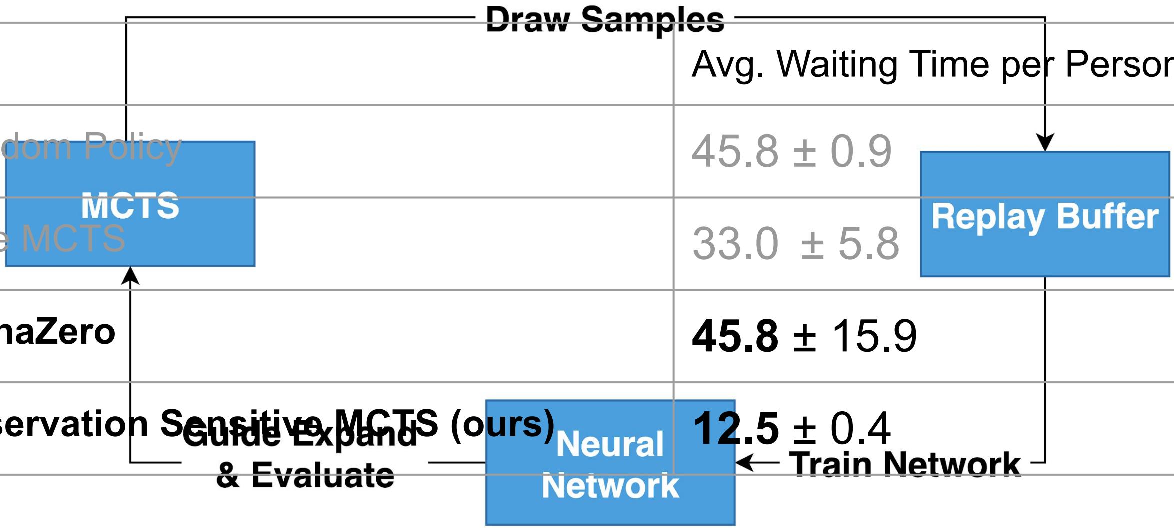



| | **Avg. Waiting Time per Person** |
|---|---|
| Random Policy | 45.8 ± 0.9 |
| **Pure MCTS** | **33.0** ± 5.8 |

# Experiments: **Observation Sensitive MCTS** vs **AlphaZero**

| | Avg. Waiting Time per Person |
|---|---|
| Random Policy | 45.8 ± 0.9 |
| Pure MCTS | 33.0 ± 5.8 |
| **AlphaZero** | **45.8** ± 15.9 |
| **Observation Sensitive MCTS (ours)** | **12.5** ± 0.4 |

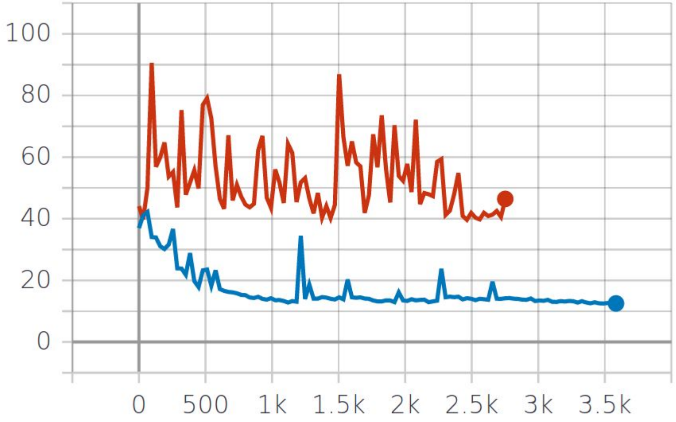


# Discussion: AlphaZero does not work here?

- State at single time step does not show quality of whole episode
  ⇒ Episode could be divided as many subtasks (Transporting passengers to their targets)

  **AlphaZero** tries to figure out how to perform a good complete episode

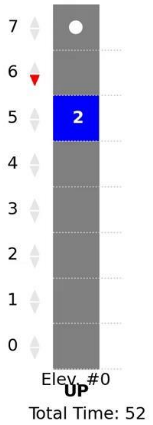
  **Our approach** learns to perform well in short term (using short-term information of waiting-times)
  → much easier




Average Waiting Time per Person over Training: AlphaZero (Red), Observation Sensitive MCTS (Blue)

# Collective Control [3] Heuristic

Used in most buildings:
1. Stop at the nearest call in their running direction
2. Switch direction if exhausted requests in current direction




Multi-Elevators:
→ **Bunching**

https://drive.google.com/file/d/1nvZ1ltgfhdZ1XwiicA5EdT4jcBMYMXV-/view?usp=sharing
https://drive.google.com/file/d/16-jooGr1U-D_gs6bZU0RQdlxo4VU98oQ/view?usp=sharing

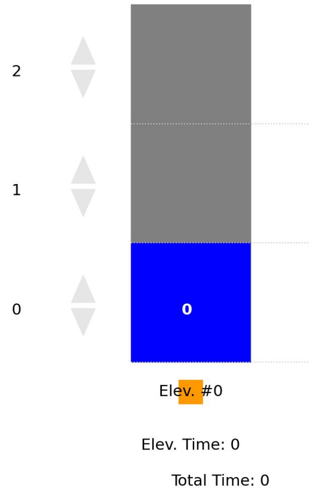
# Experiments: Observation Sensitive MCTS vs **Heuristic**

| | Avg. Waiting Time per Person |
|---|---|
| Random Policy | 45.8 ± 0.9 |
| Pure MCTS | 33.0 ± 5.8 |
| AlphaZero | 45.8 ± 15.9 |
| **Observation Sensitive MCTS (ours)** | **12.5** ± 0.4 |
| **Collective Control (Heuristic)** | **10.5** ± 0.3 |

# Discussion: Why don’t we reach performance of Heuristic?

- Simple Problem → Collective Control performs nearly perfect
- Hyperparameters might not be optimal

- More complex Scenarios (more elevators, more floors) are more promising for RL
    - Requires new hyper-parameter search and simulation time grows



# Conclusion & Discussion

- Environment + Passenger-Generator
  ⇒ Stochastic environment → can model passenger distributions and do not leak state through MCTS

- Heuristic Baseline (Collective Control)
  ⇒ Very strong for single elevator environments
  ⇒ “bunching” in multi-elevator environments

- Observation sensitive MCTS
  ⇒ Extends AlphaZero to continuous reward problem-sets
  ⇒ Training time dominated by simulation (same as in Alphazero)
    ⇒ Difficult to evaluate on large environments

# Thank you!

# Q&A

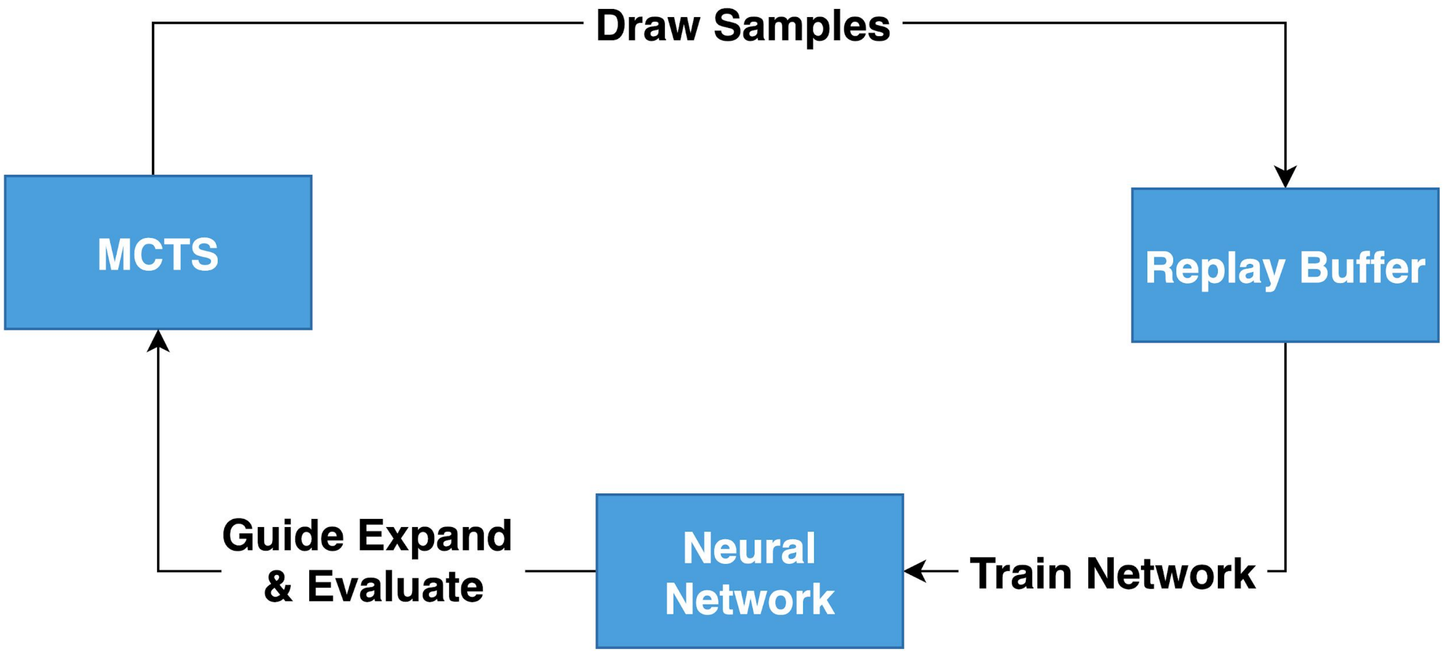
# References


[1] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. In: Science 362.6419 (2018), pp. 1140–1144

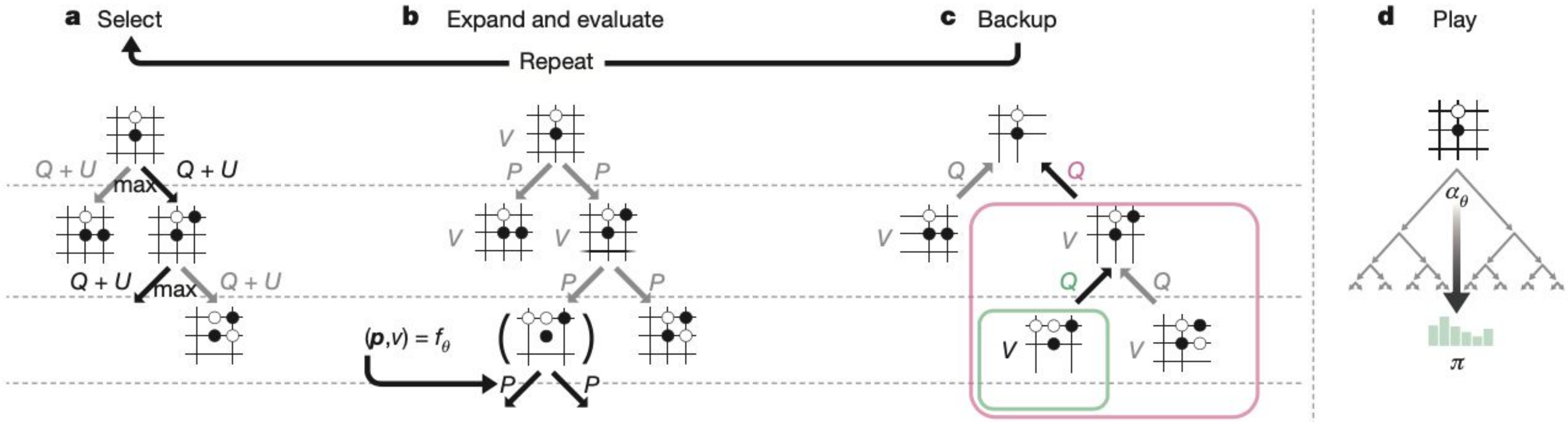
[2] A. Laterre, Y. Fu, M. K. Jabri, A.-S. Cohen, D. Kas, K. Hajjar, T. S. Dahl, A. Kerkeni, and K. Beguir. “Ranked reward: Enabling self-play reinforcement learning for combinatorial optimization”. In: arXiv preprint arXiv:1807.01672 (2018)

[3] M. Siikonen, “Elevator traffic simulation”. In: Simulation Sage Publications Sage CA Volume 61.4 p.257-267 (1993)

# **Algorithm** (Overview)



# **MCTS** (AlphaZero)




| | | |
|---|---|---|
| • Action Value: | Q(s, a) | How good is the action a? |
| • Upper Confidence Bound: | U(s, a) | Should I explore a further? |
| • Visit Counter: | N(s, a) | How often was a visited? |

# **MCTS** (AlphaZero) + Our Modification

$$a = \arg\max_a Q(s,a) + U(s,a)$$

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{s'} v(s')$$

$$U(s,a) \propto \frac{p(s,a)}{1 + N(s,a)}$$

$$Q_{new}(s,a) = \frac{1}{N(s,a)} \sum_{s'} c_{obs} \cdot f_{norm}\left(\frac{r(\pi_{s,s'})}{|\pi_{s,s'}|}\right) + (1 - c_{obs}) \cdot v(s')$$

Length of path from s to s'

$$f_{norm}(x) = \tanh\left(\frac{x}{10}\right)$$

| | | |
|---|---|---|
| ● Action Value: | Q(s, a) | How good is the action a? |
| ● Upper Confidence Bound: | U(s, a) | Should I explore a further? |
| ● Visit Counter: | N(s, a) | How often was a visited? |

# Experiments Configurations

| | |
|---|---|
| MCTS-Samples | 40 |
| Observation-Weight | 0.5 |
| Replay Buffer Size | 7,000 |
| Iterations | 150 |
| Episodes | 16 |
| Batch-Size | 128 |
| Ranked-Reward Buffer | 250 |
| Ranked-Reward Threshold | 0.75 |