Distributed Artificial Intelligence

Putting the artifice back in AI

Aidan O'Neill[1]

[1]Mathematics and Computer Science
Davidson College

March 5, 2024

Table of Contents

1. Hex
2. Monte Carlo Tree Search
3. Dsys
4. Results and Conclusions

Table of Contents

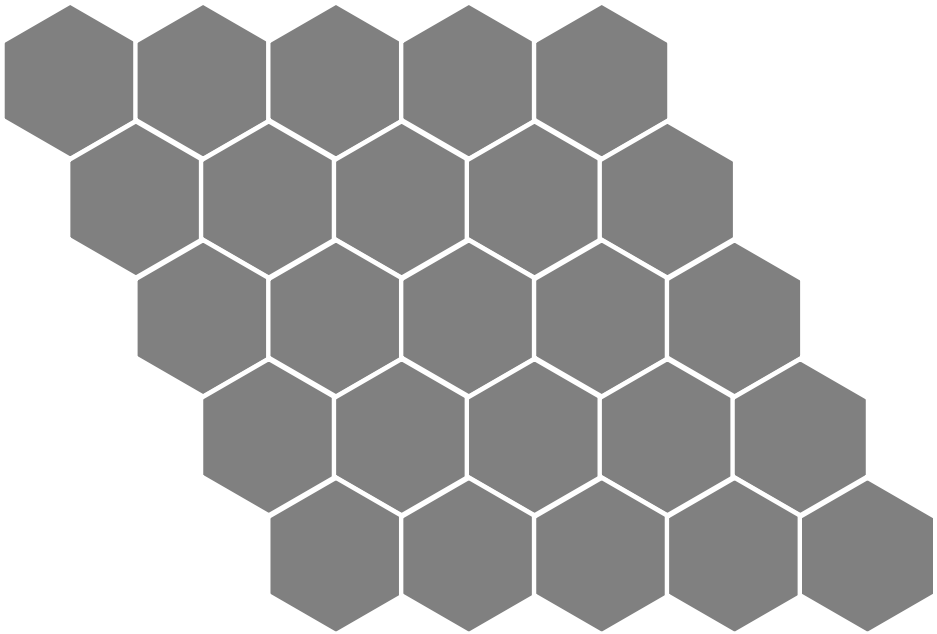
1 Hex

2 Monte Carlo Tree Search

3 Dsys

4 Results and Conclusions

Hex



Stockfish and Heuristics

```
constexpr int QueenSafeCheck = 780;
constexpr int RookSafeCheck = 1080;
constexpr int BishopSafeCheck = 635;
constexpr int KnightSafeCheck = 790;
```

State Space

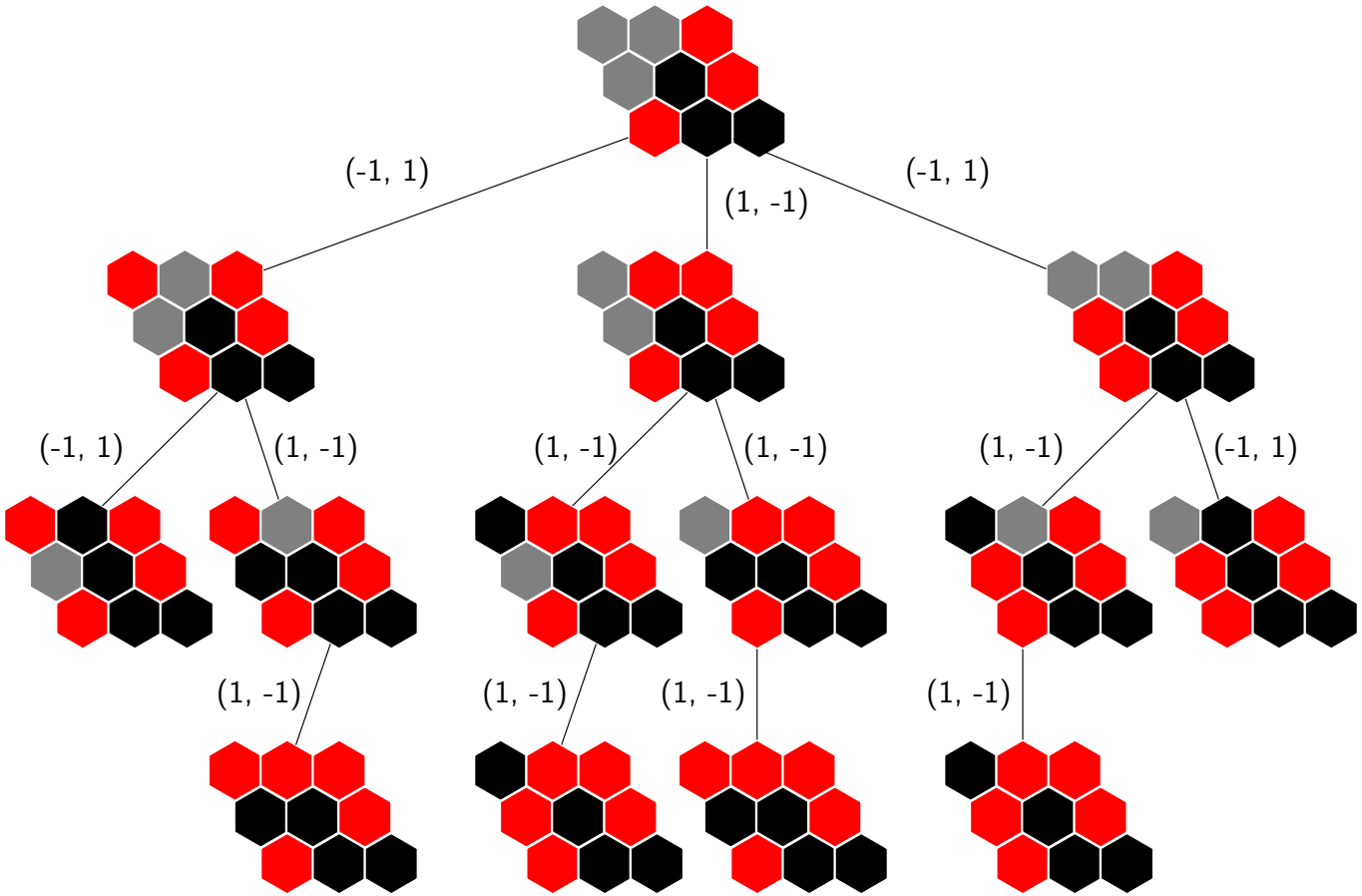


Table of Contents

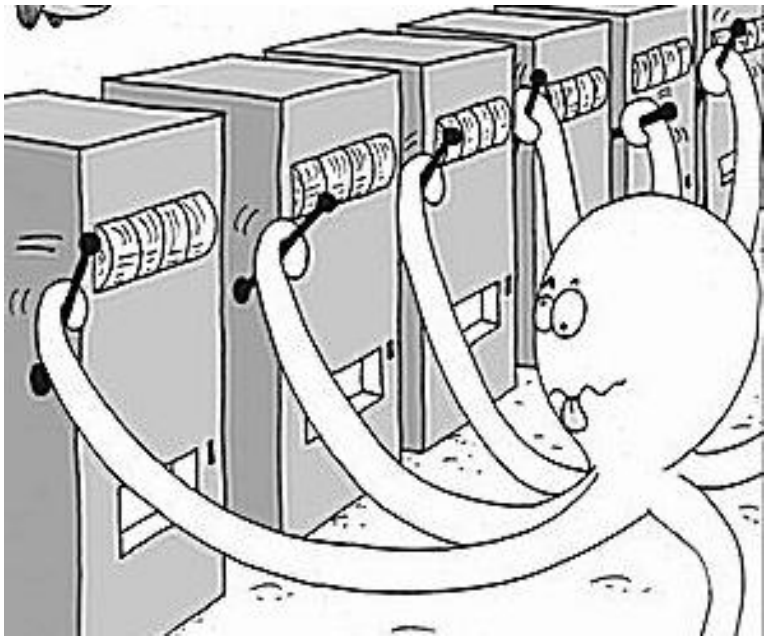
1 Hex

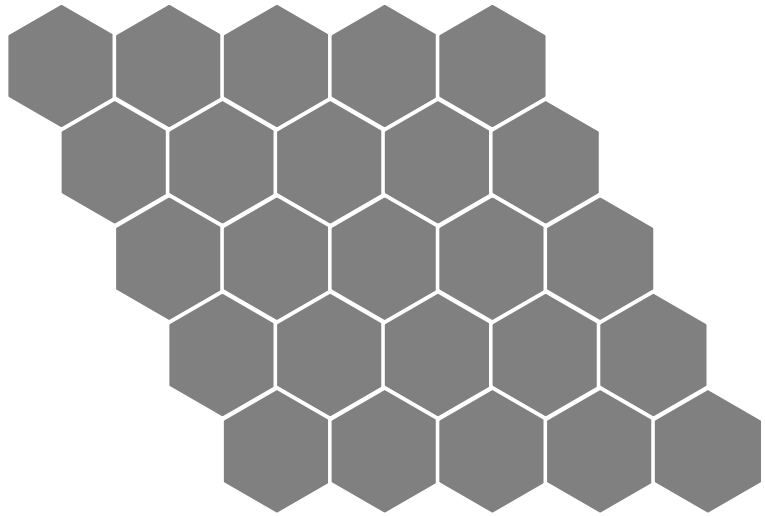
2 Monte Carlo Tree Search

3 Dsys

4 Results and Conclusions

Multi-Armed Bandits



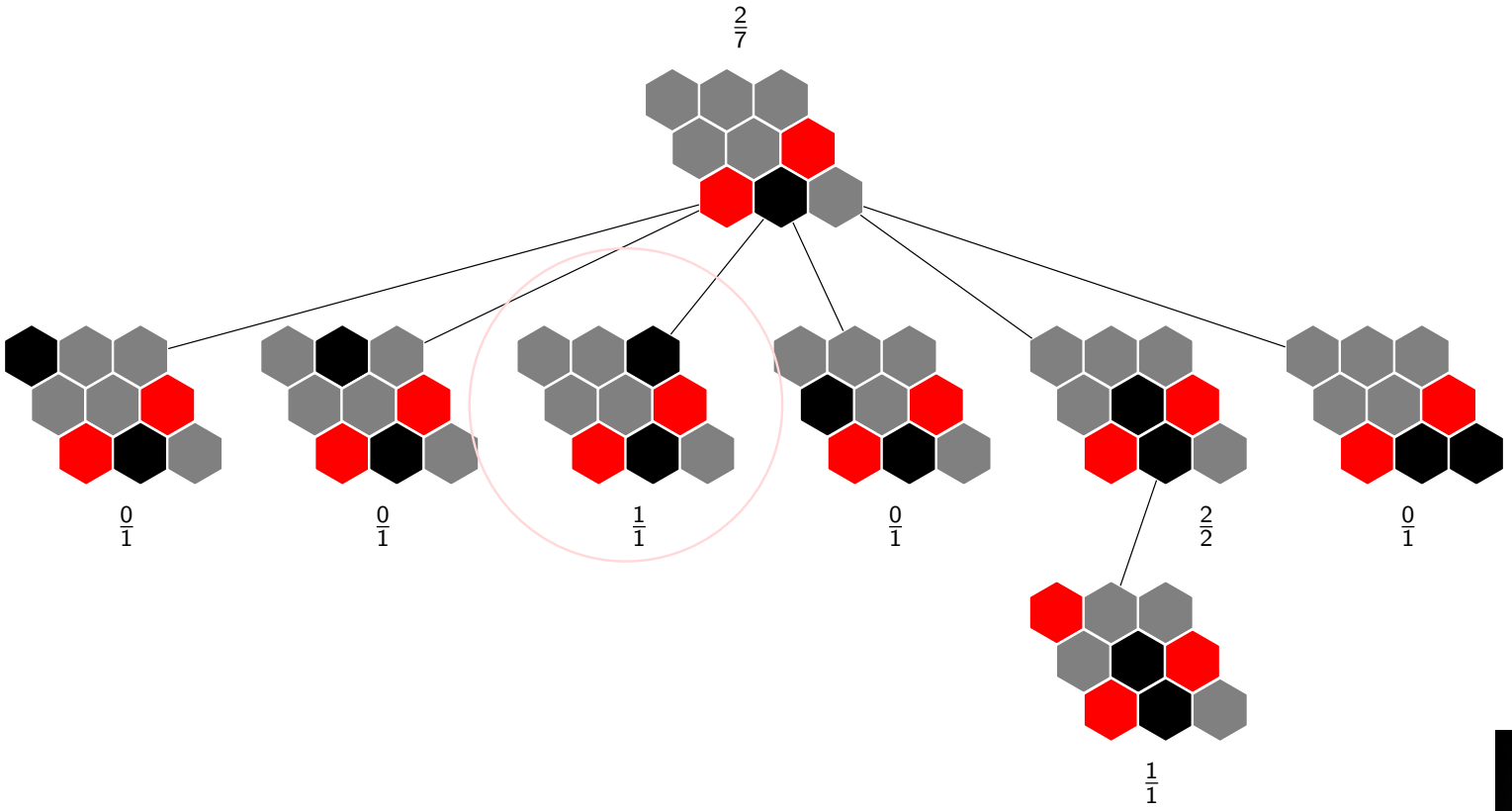


Exploration-Exploitation

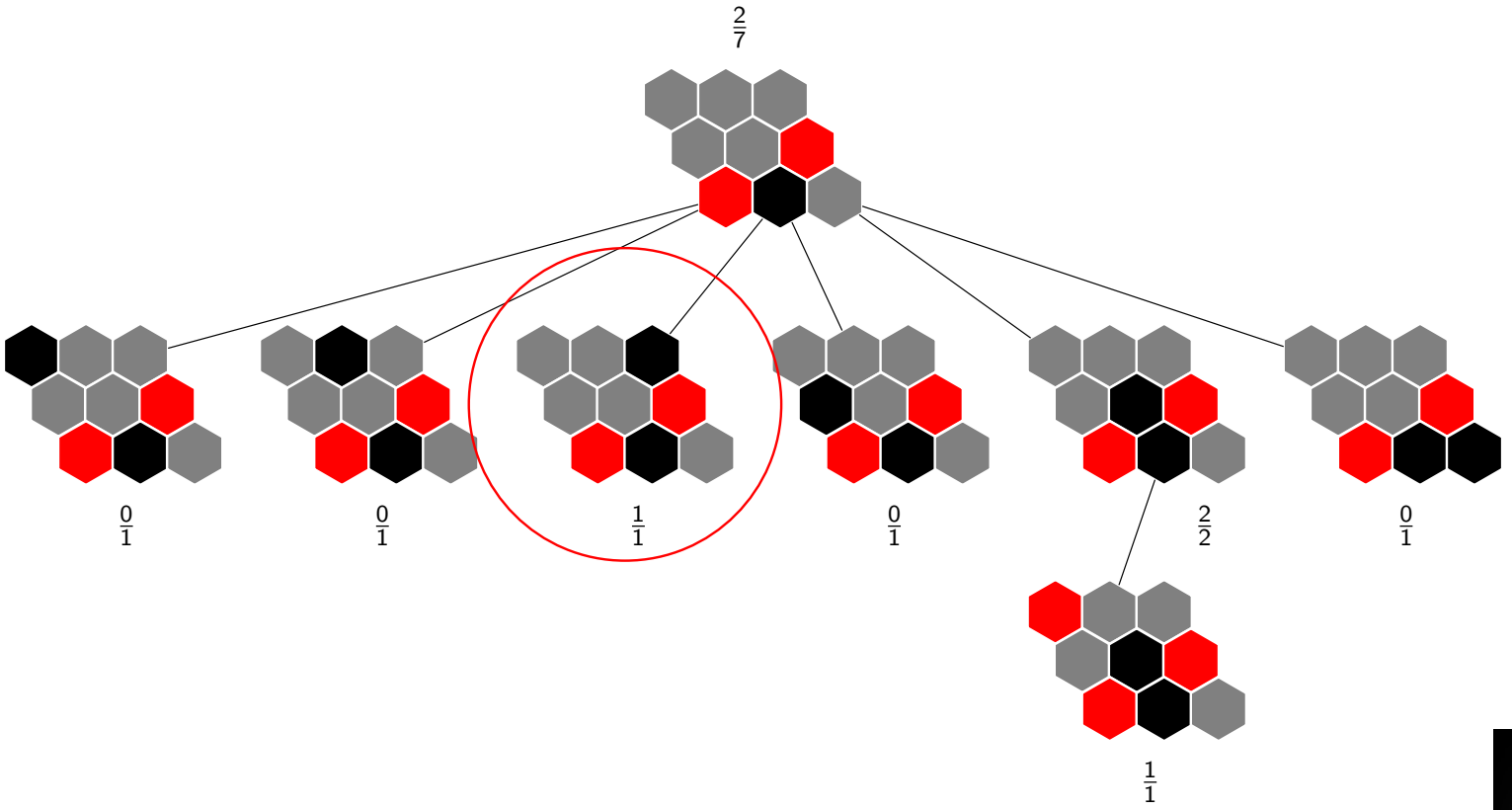
UCB1

$$\frac{Q(v_i)}{N(v_i)} + c\sqrt{\frac{\log N_v}{N(v_i)}}$$

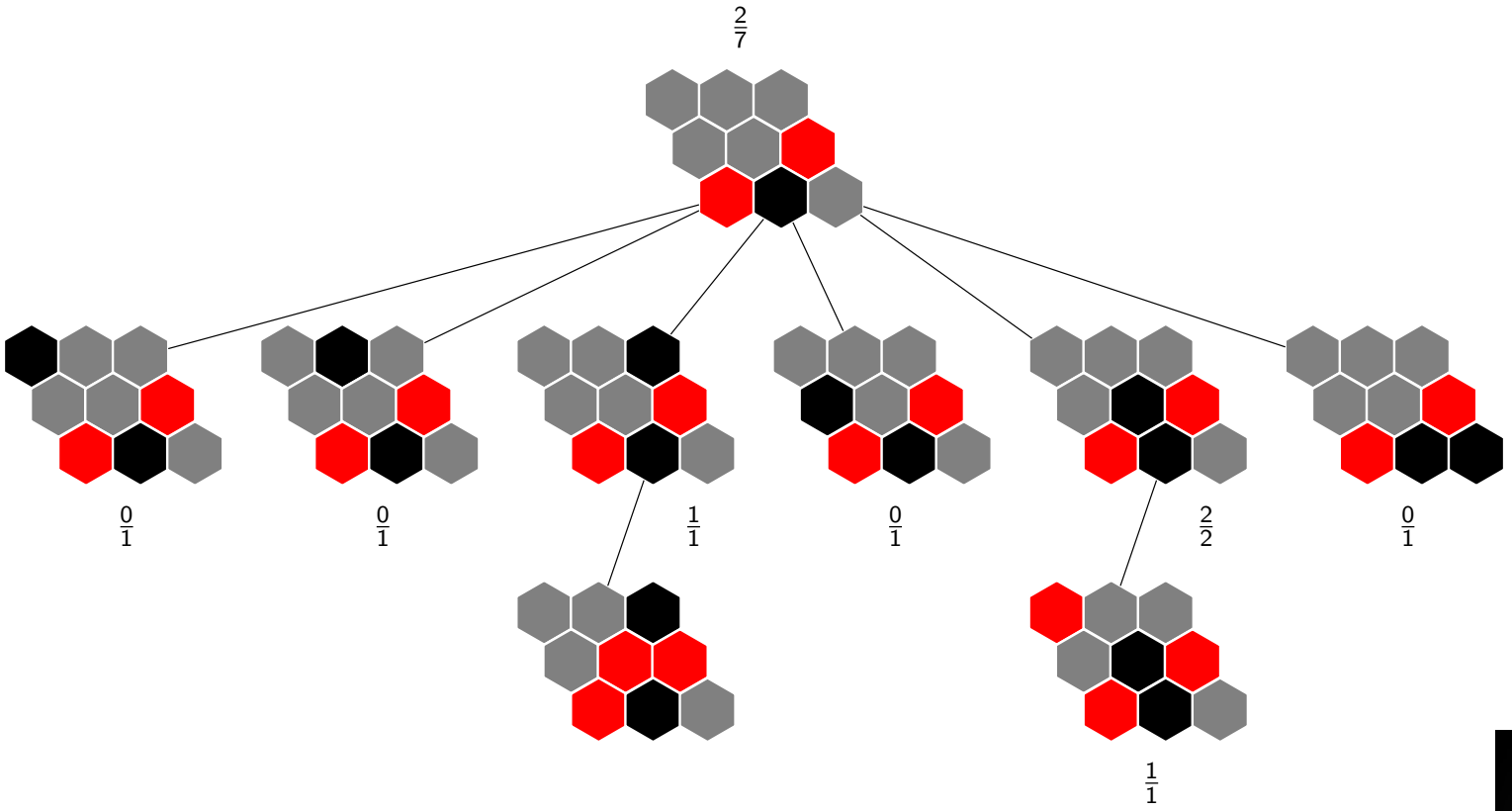
Selection



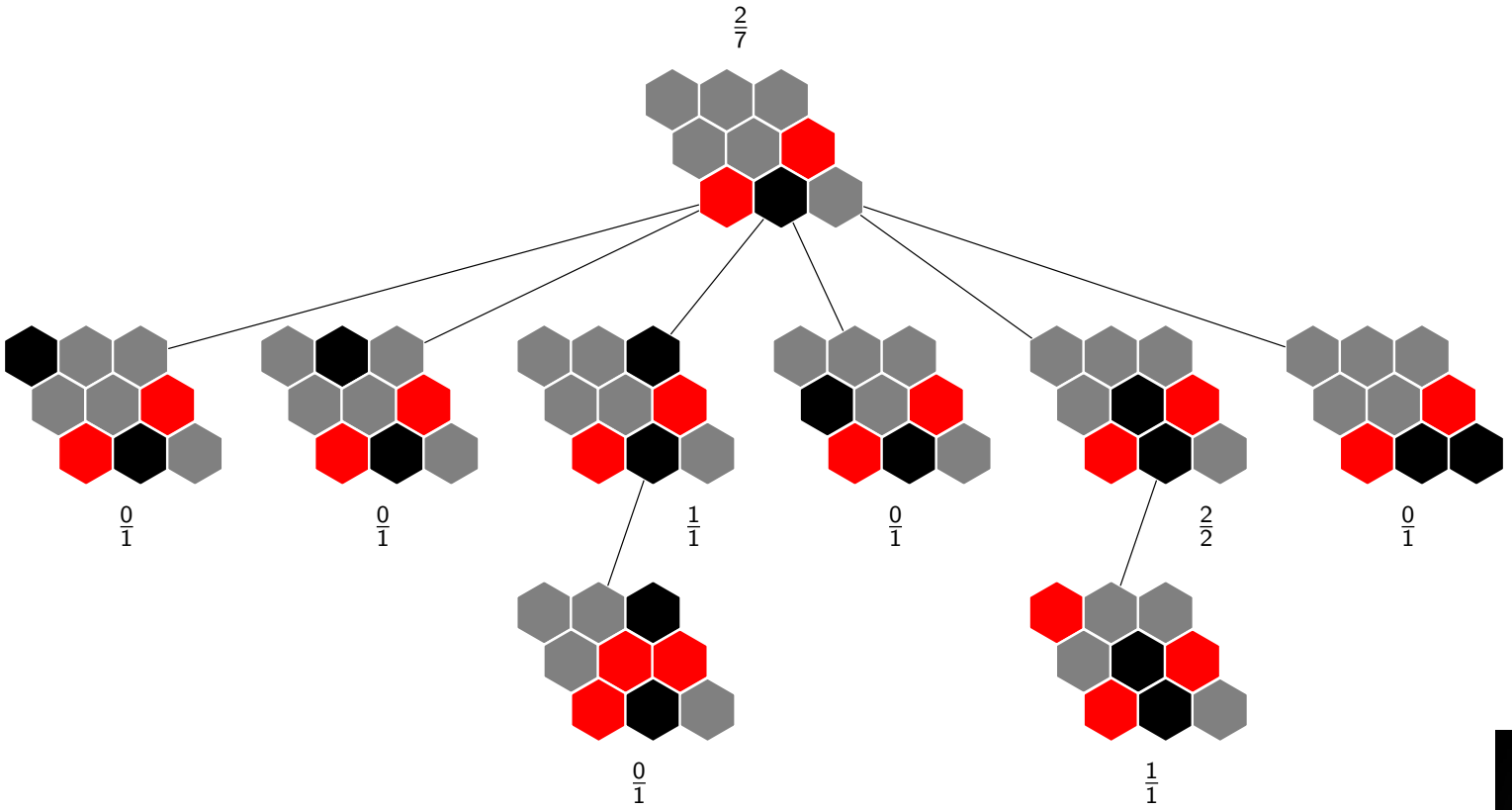
Selection



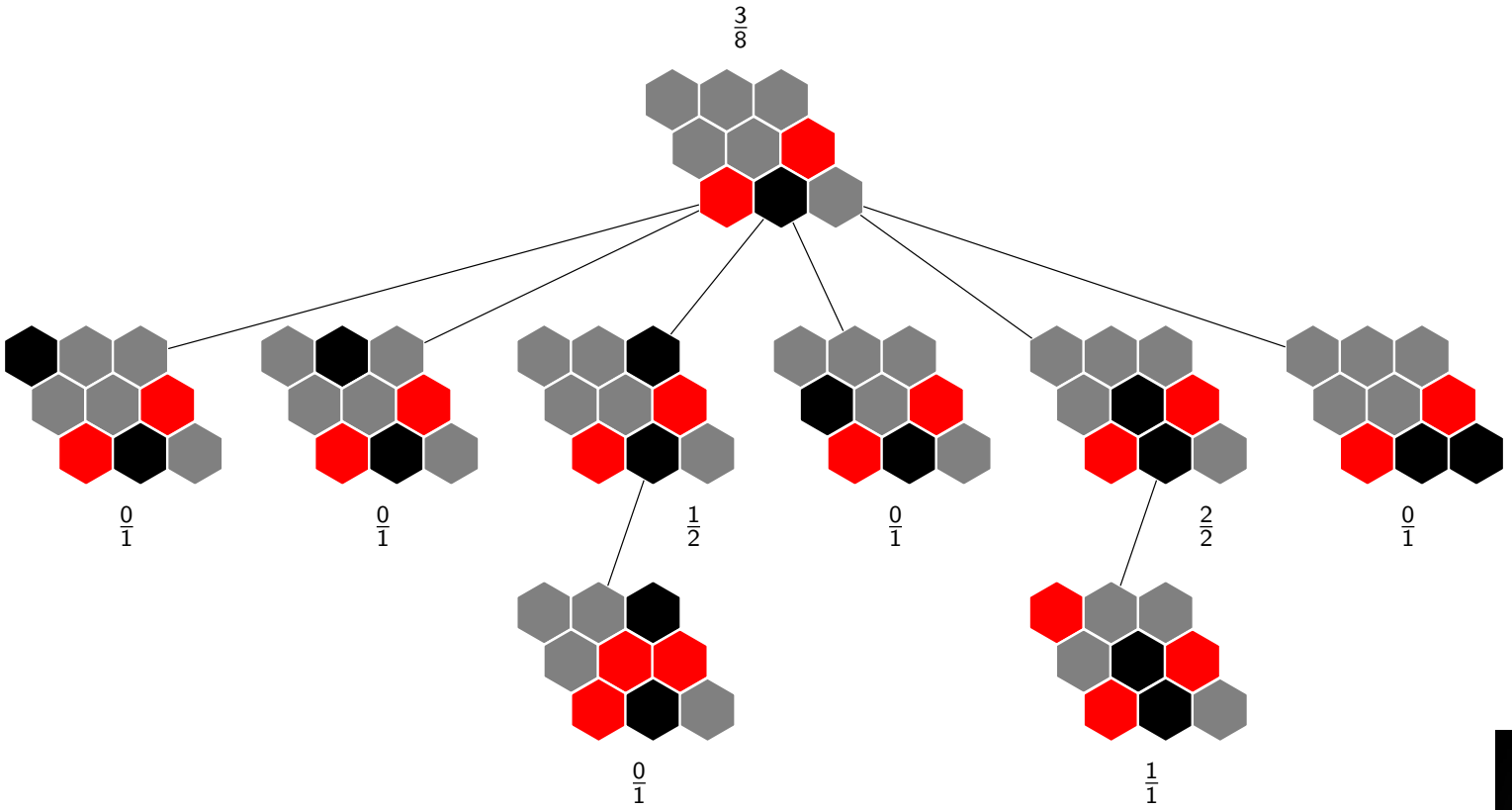
Expansion



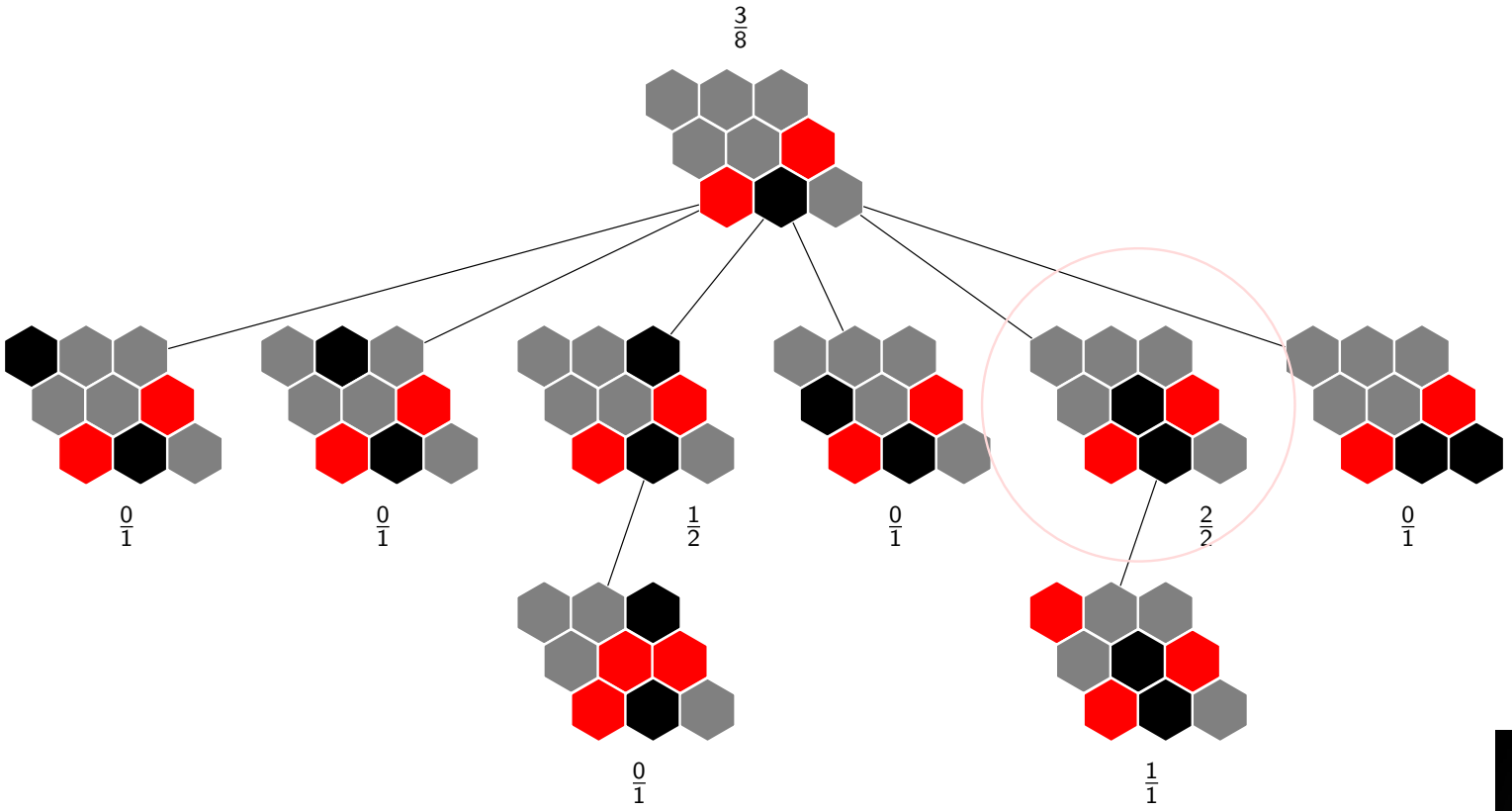
Simulation



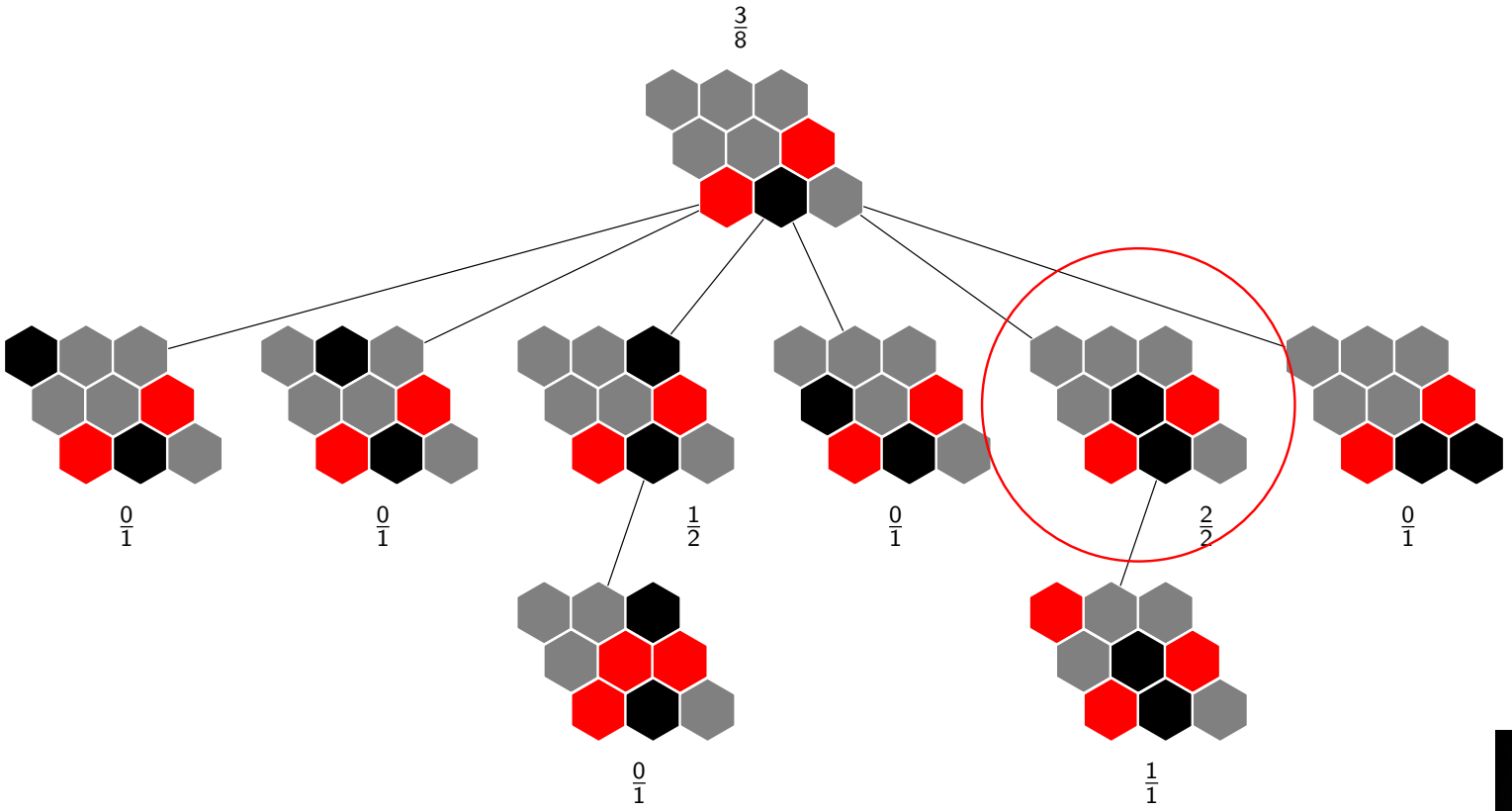
Backpropagation



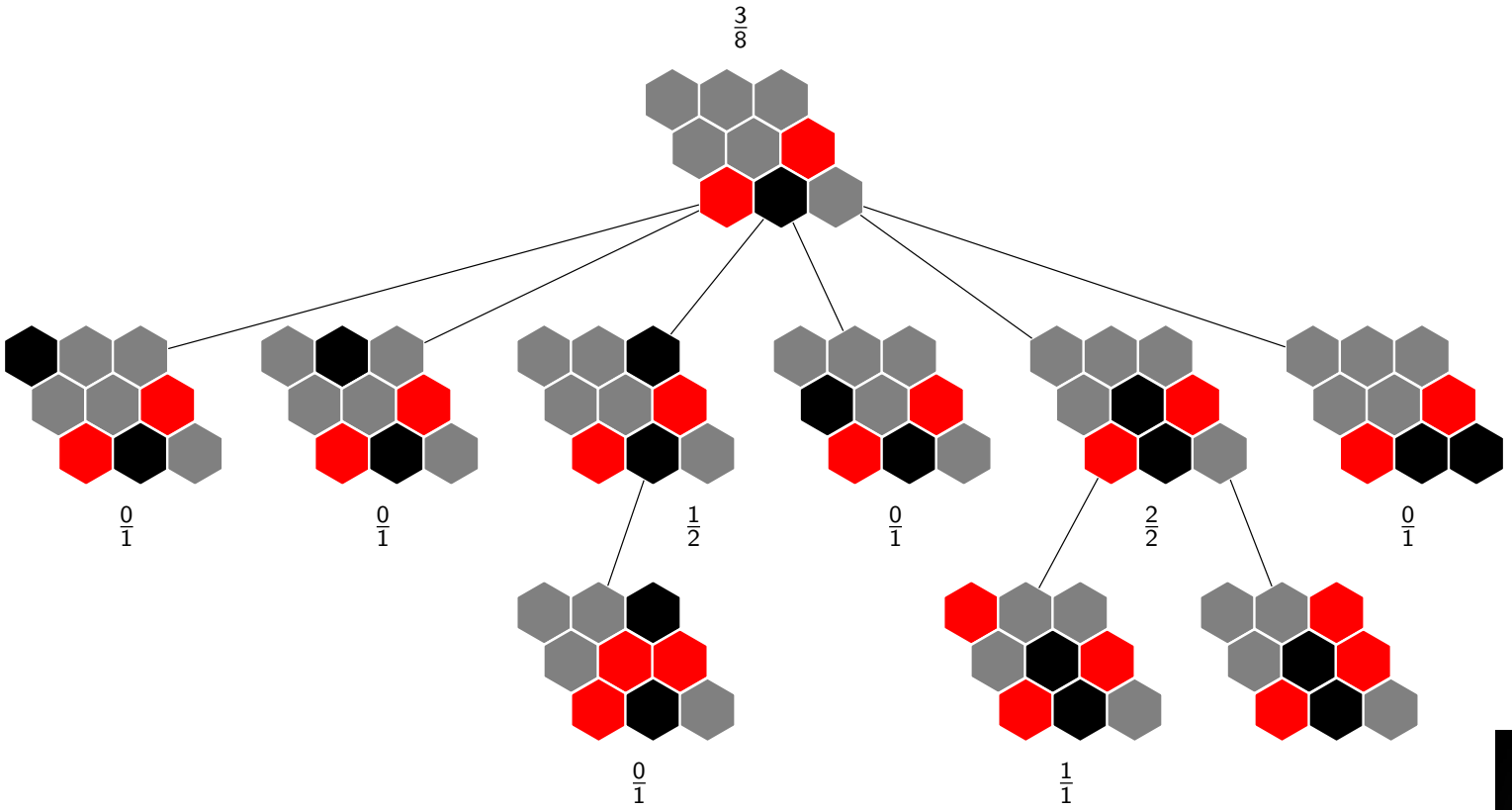
Selection



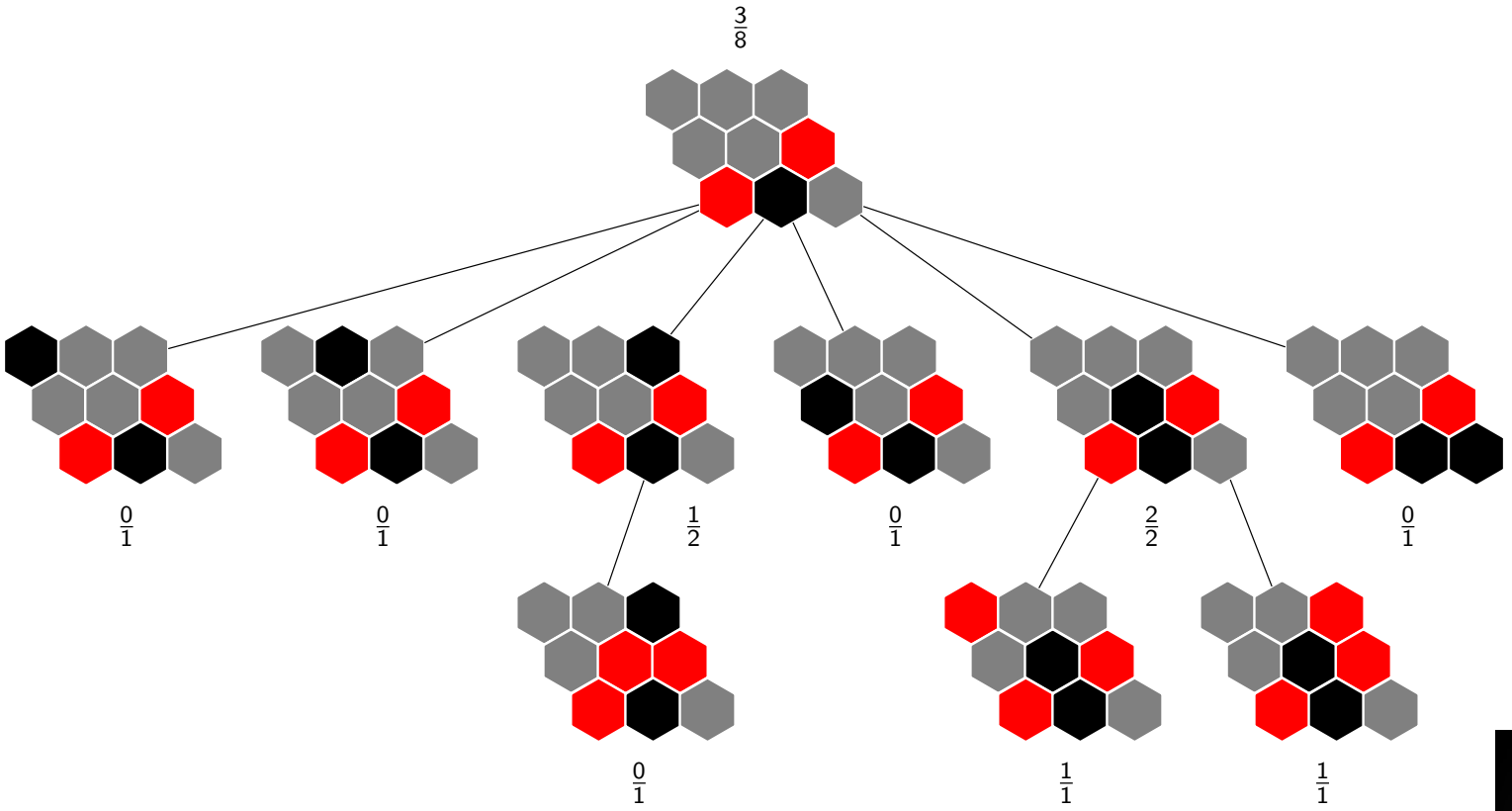
Selection



Expansion



Simulation



Backpropagation

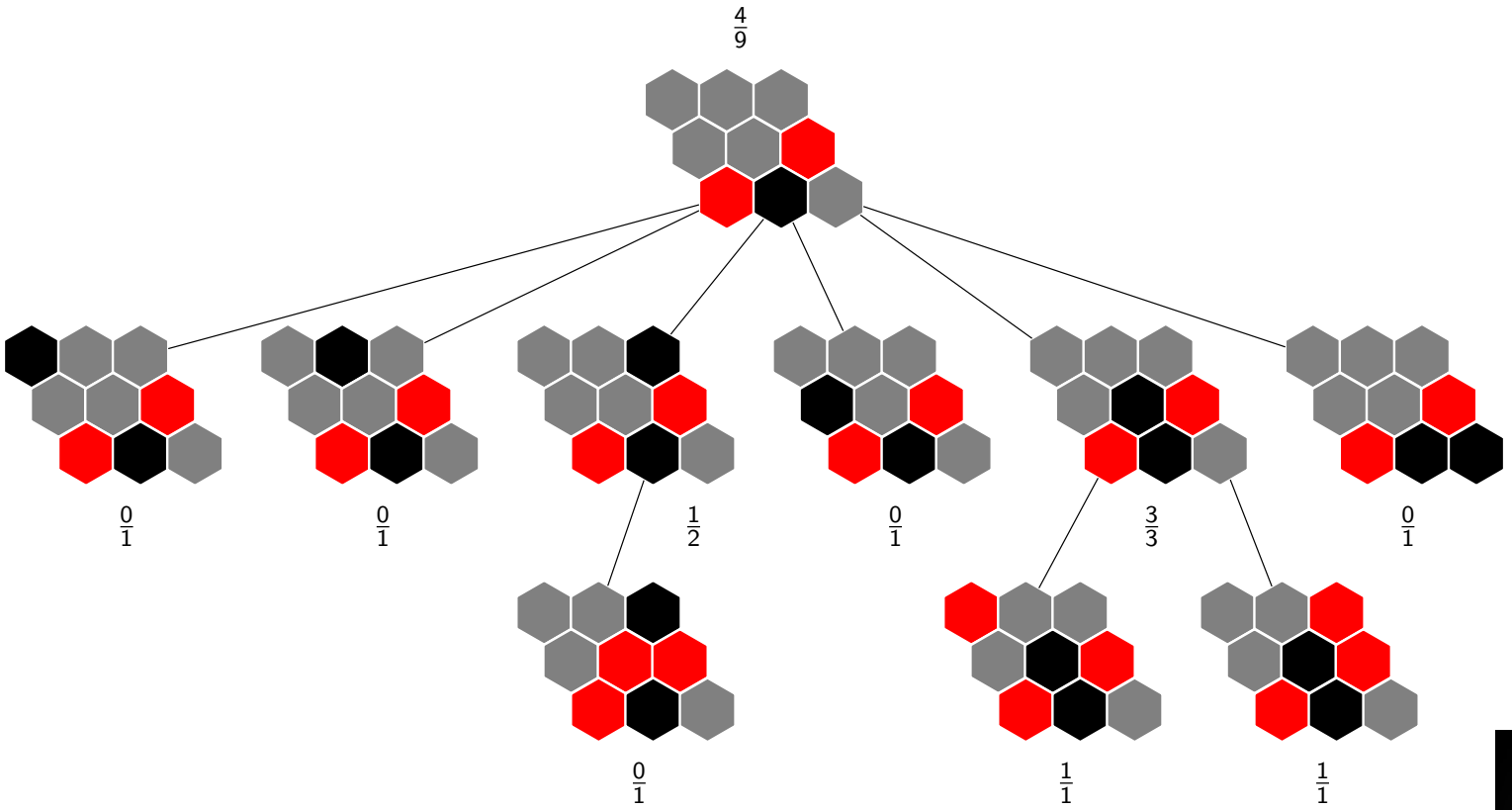


Table of Contents

1 Hex

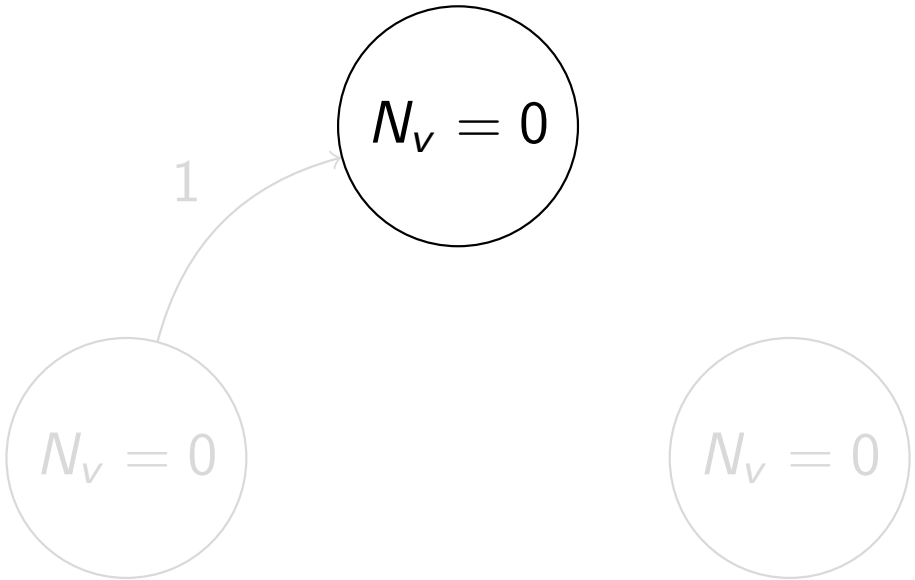
2 Monte Carlo Tree Search

3 Dsys

4 Results and Conclusions

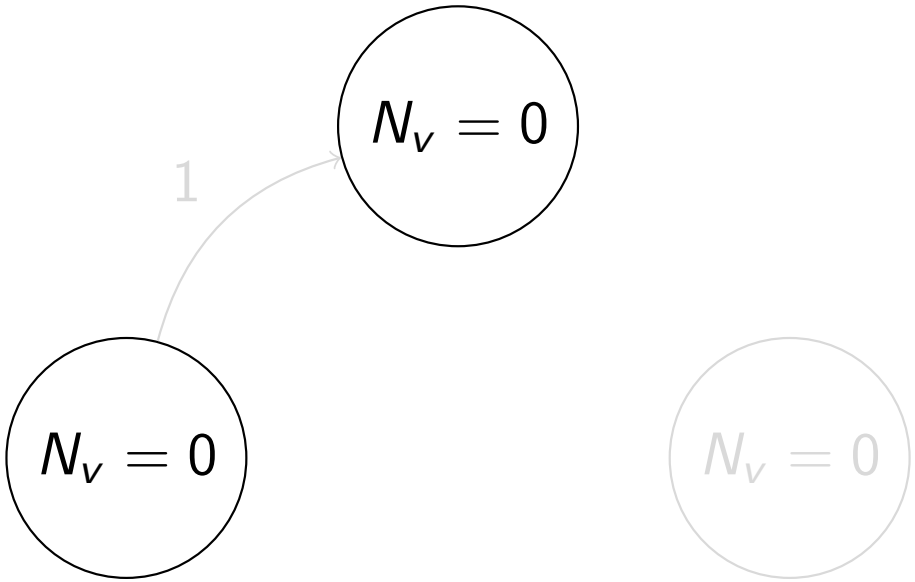
Paradigms

- Multithreading



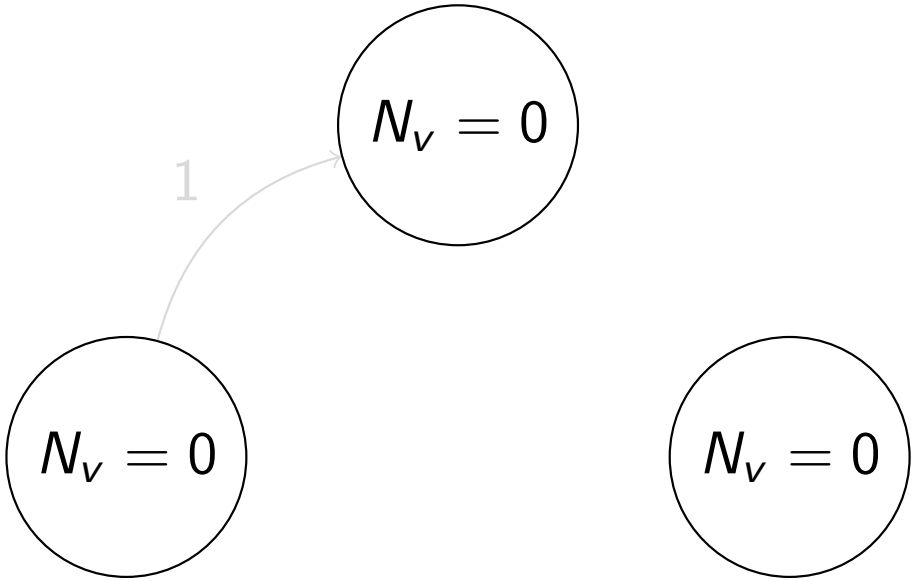
Paradigms

- Multithreading



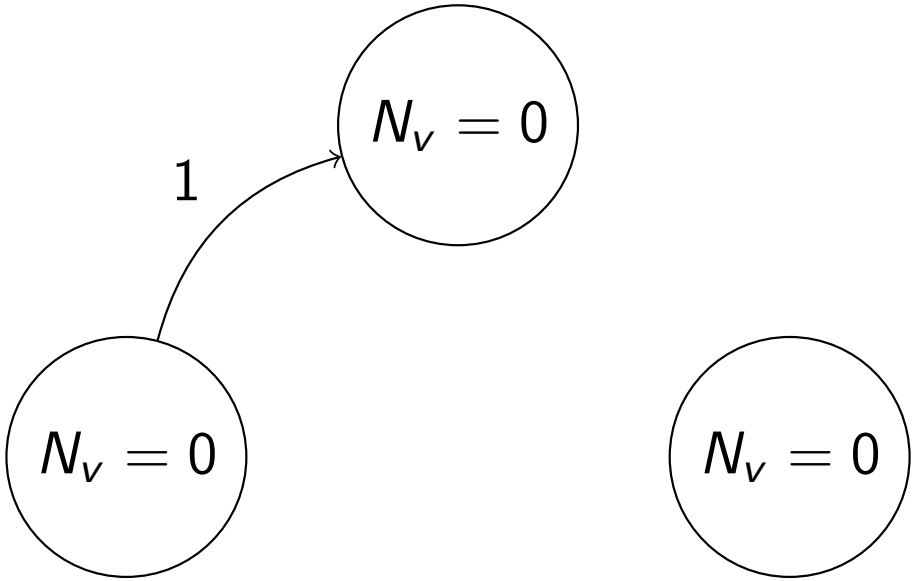
Paradigms

- Multithreading



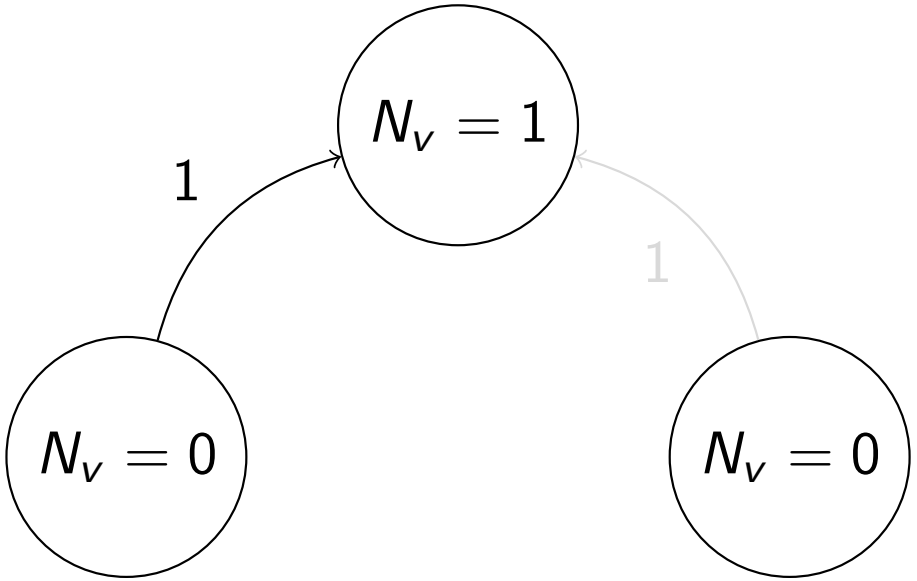
Paradigms

- Multithreading



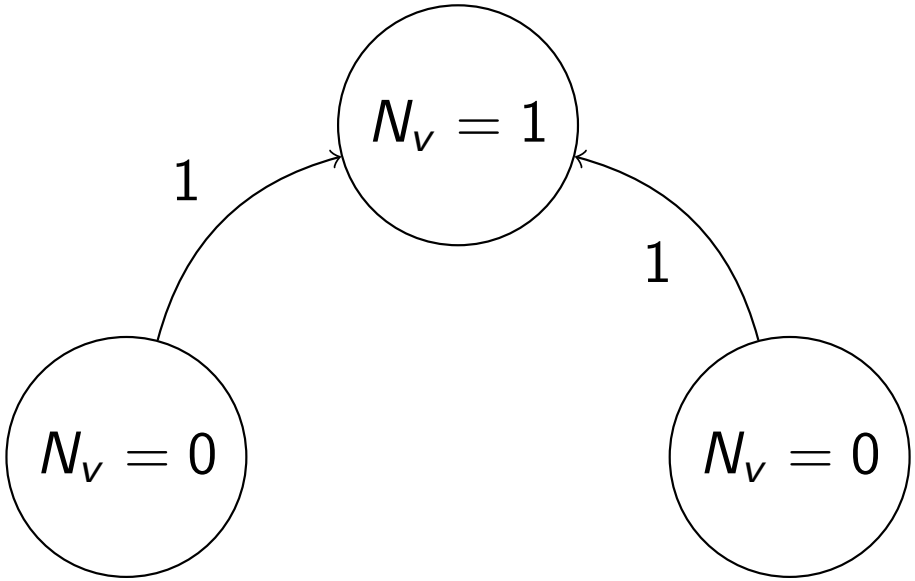
Paradigms

- Multithreading



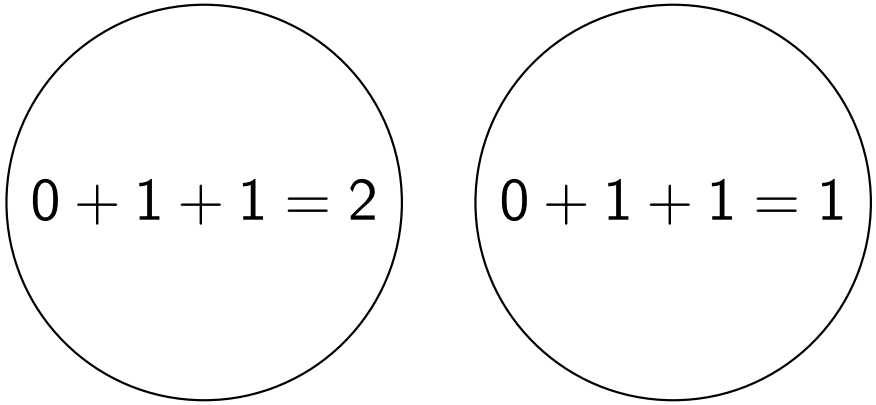
Paradigms

- Multithreading



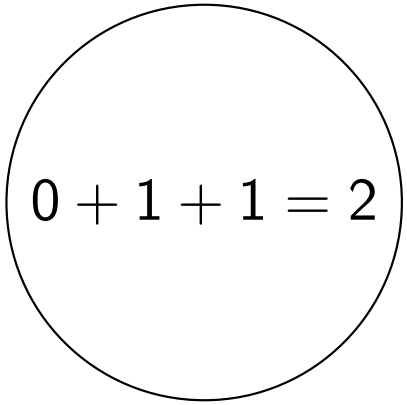
Paradigms

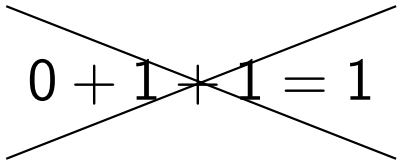
- Multithreading



Paradigms

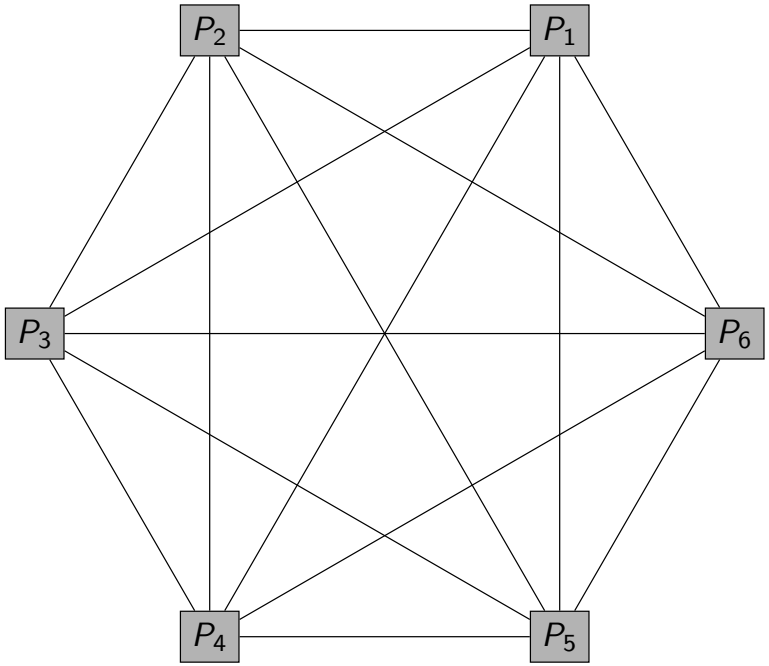
- Multithreading





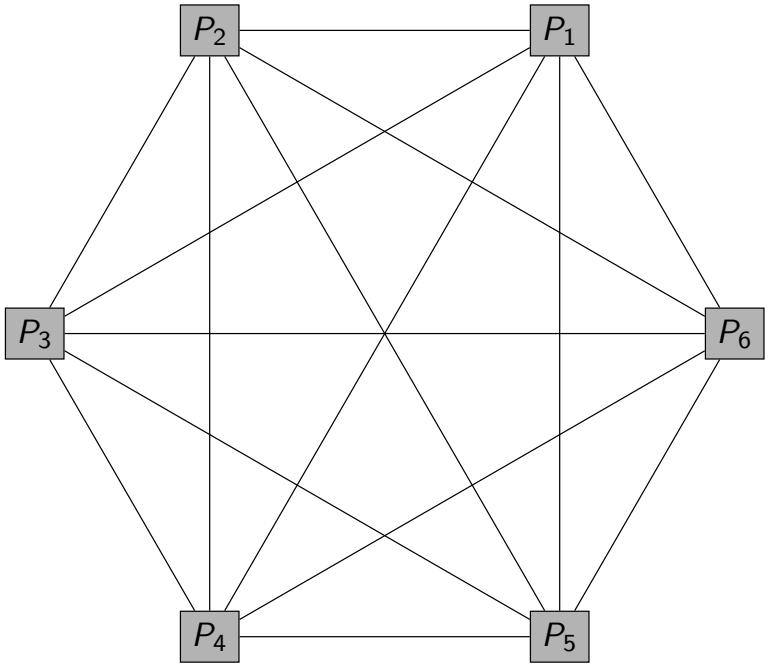
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



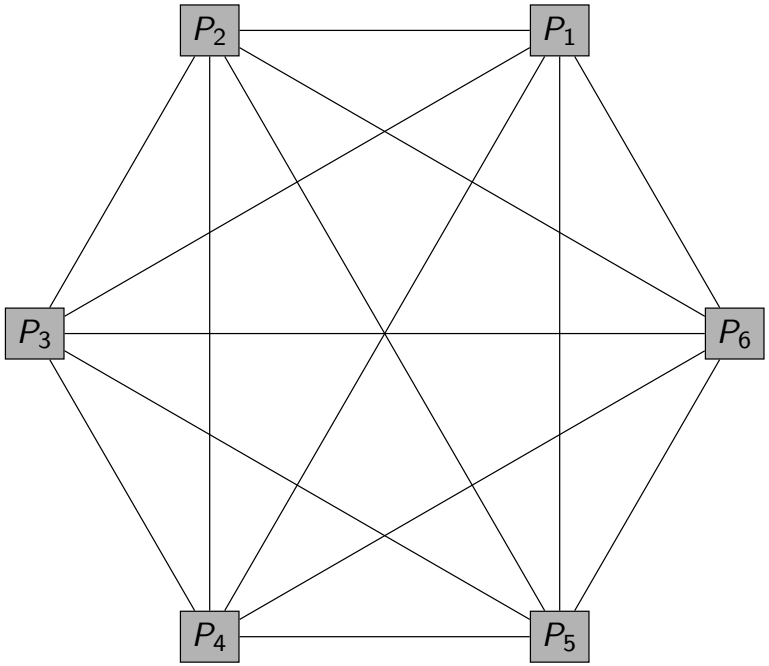
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



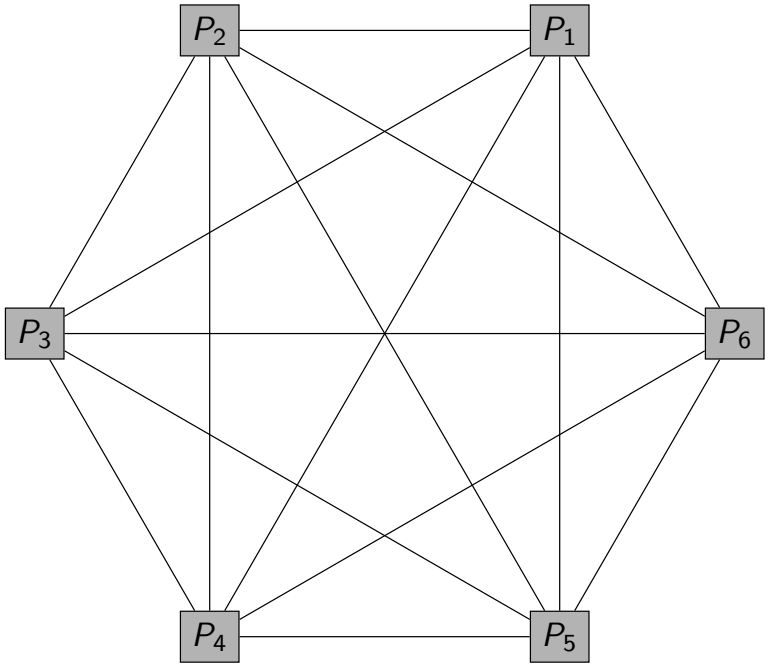
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



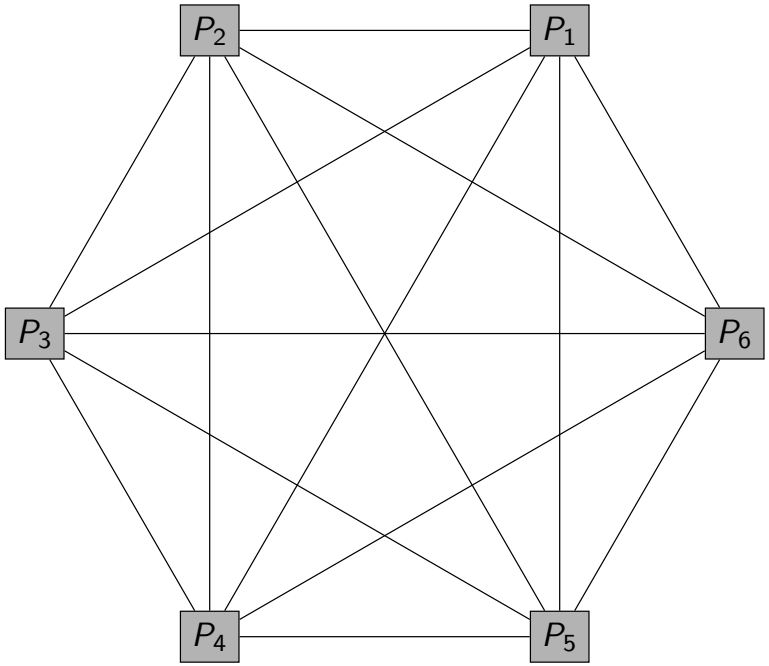
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



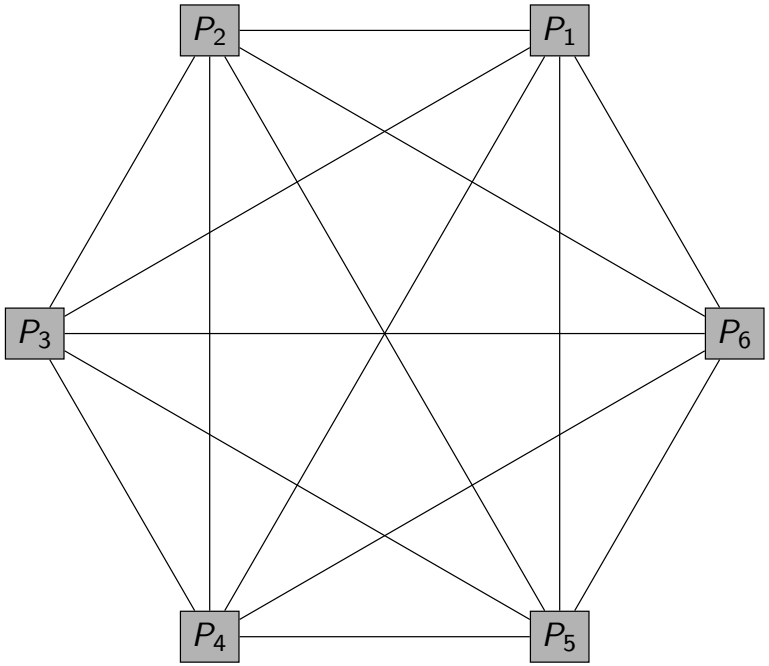
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



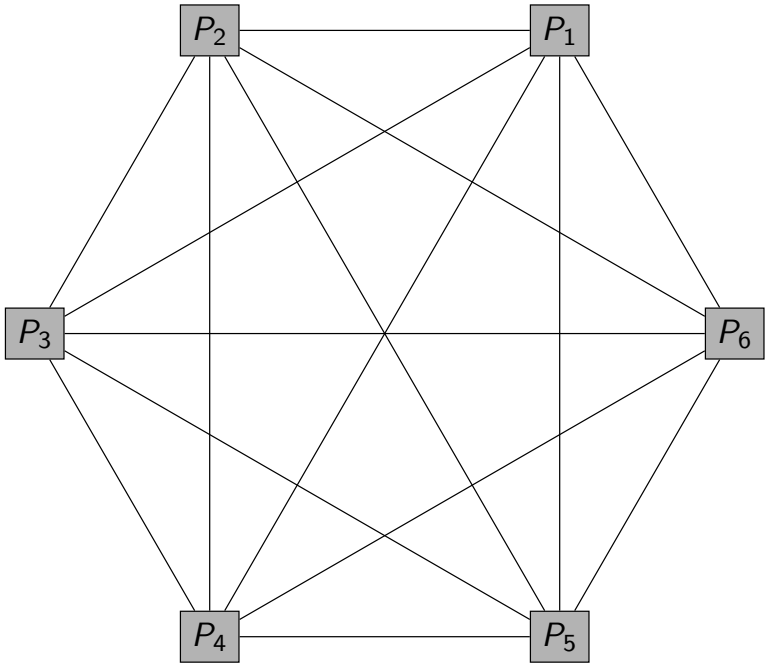
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



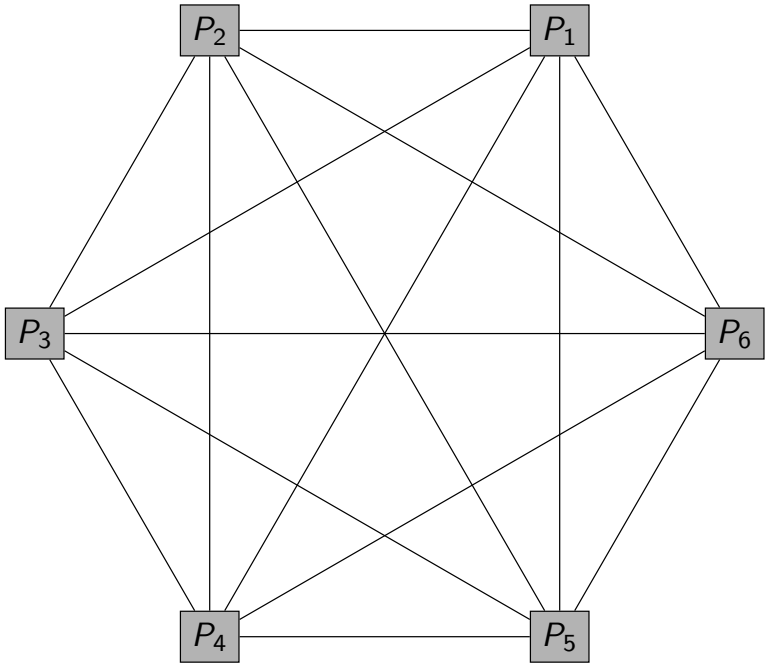
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



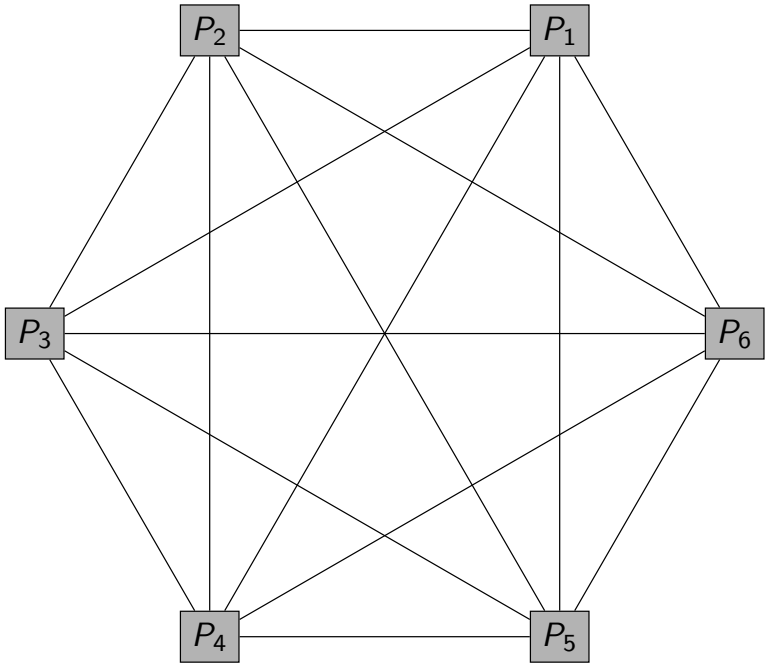
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa

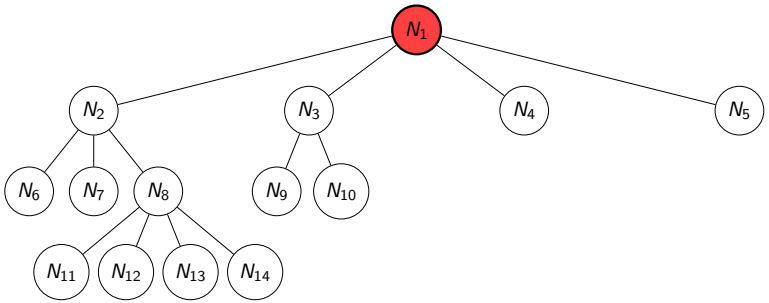


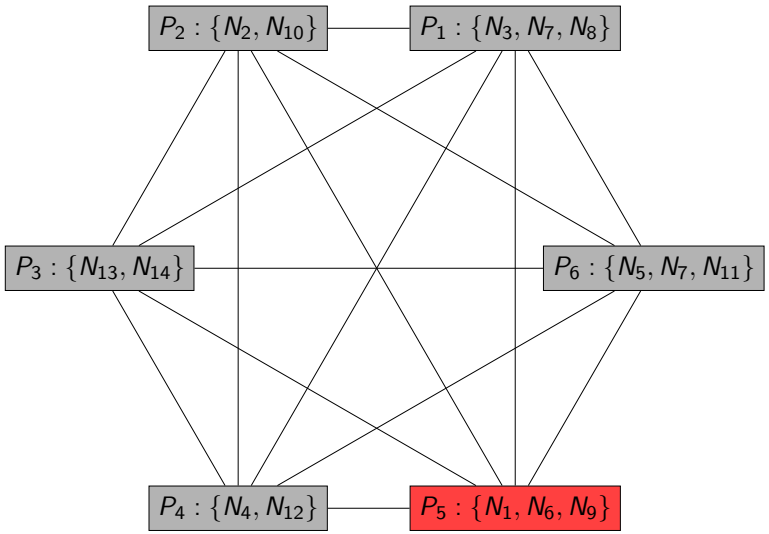
Paradigms

- Multithreading
  - Locks
    - Coarse-Grained
    - Fine-Grained
  - Atomic Variables
- Multiprocessing
  - MPI
  - Grappa



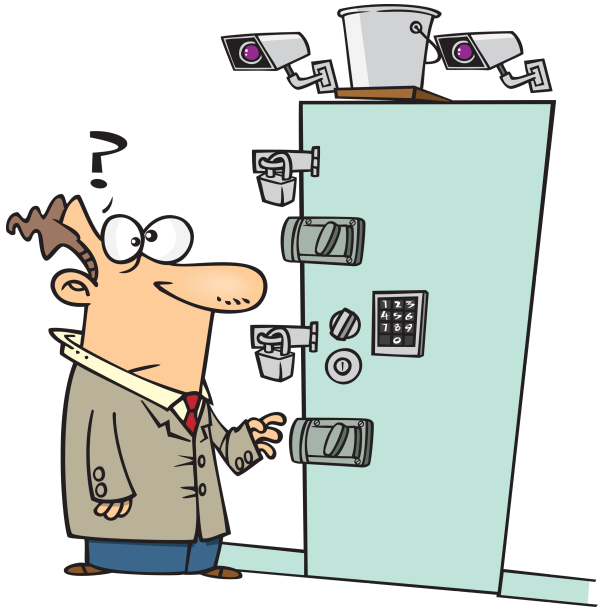
Distributed MCTS





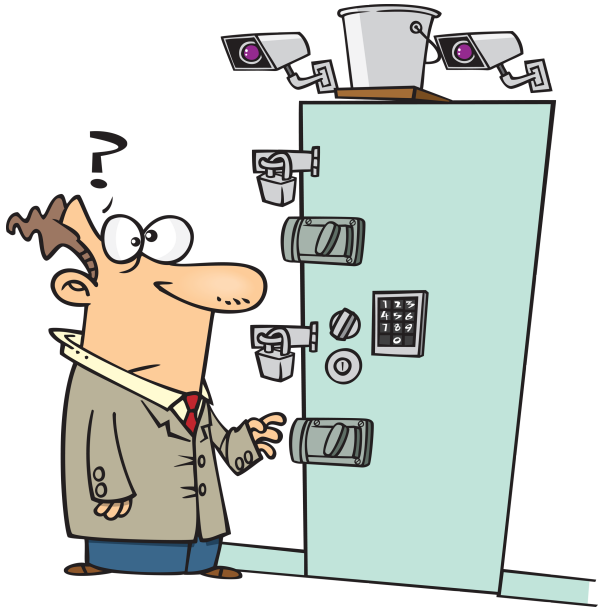
Synchronizer

- Condition Variable
- Mutex
  - Deadlocks
- Number of operations



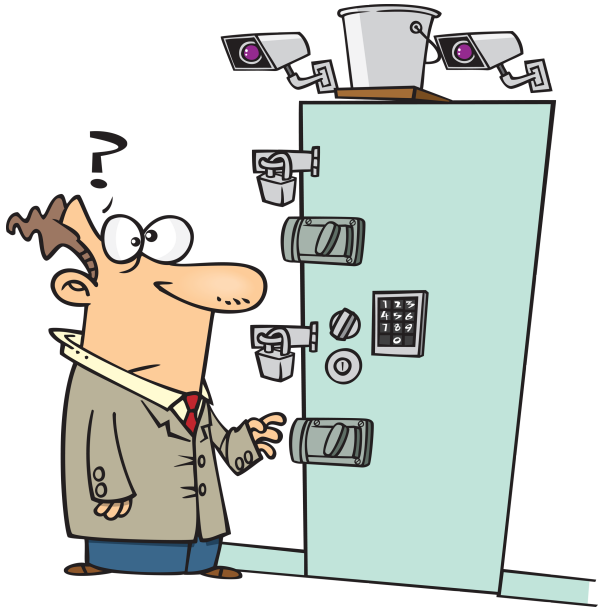
Synchronizer

- Condition Variable
- Mutex
  - Deadlocks
- Number of operations



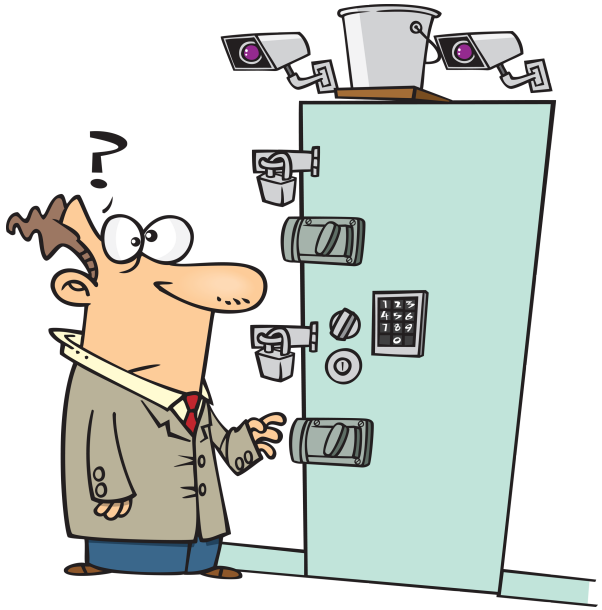
Synchronizer

- Condition Variable
- Mutex
  - Deadlocks
- Number of operations

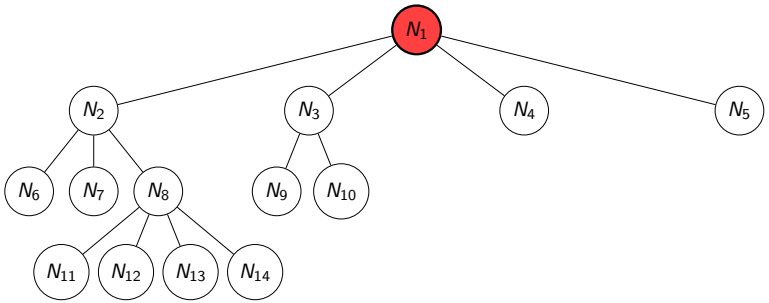


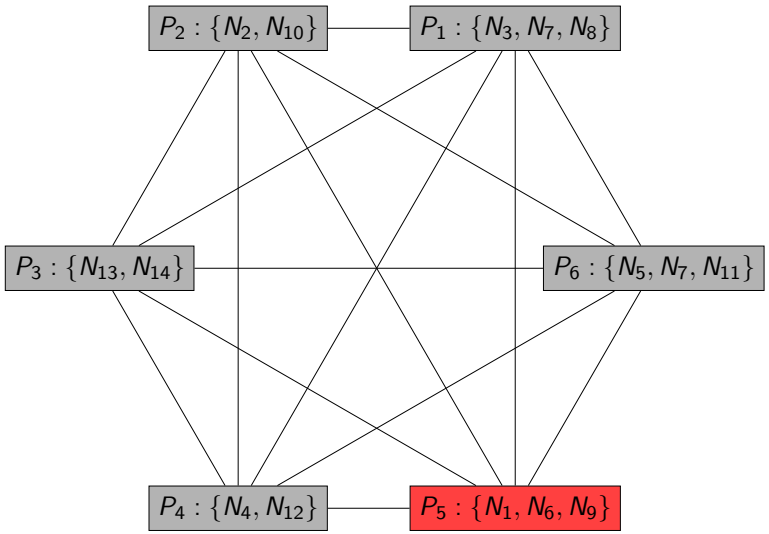
Synchronizer

- Condition Variable
- Mutex
  - Deadlocks
- Number of operations

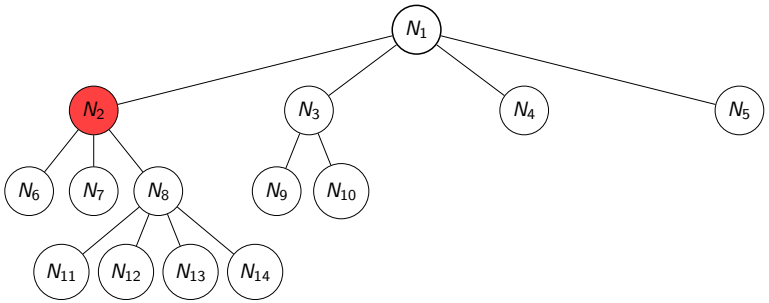


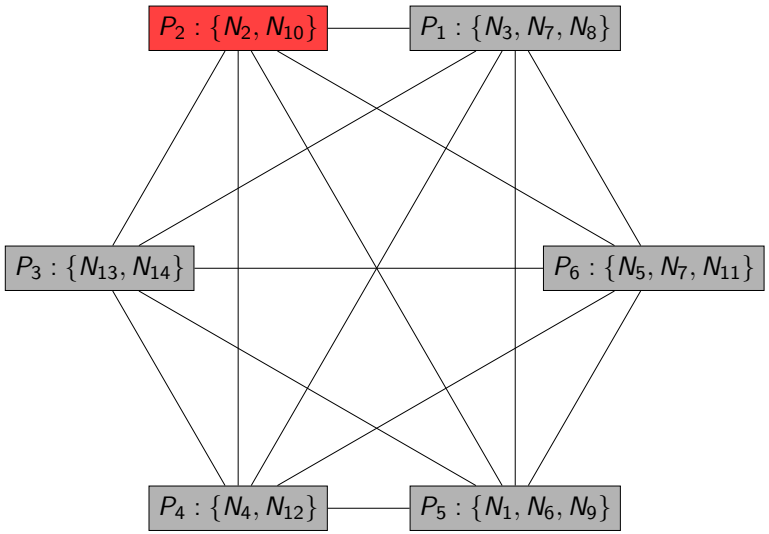
Distributed MCTS



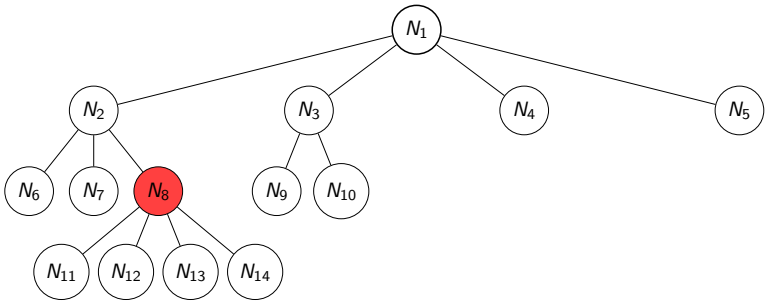


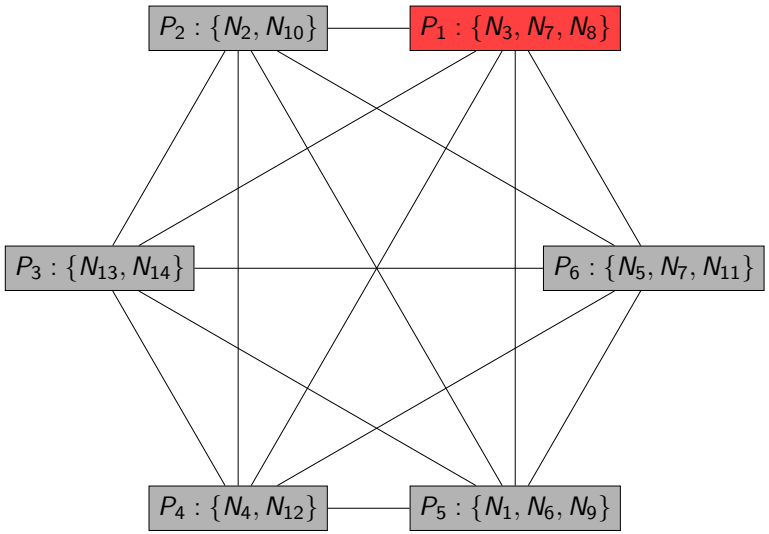
Distributed MCTS



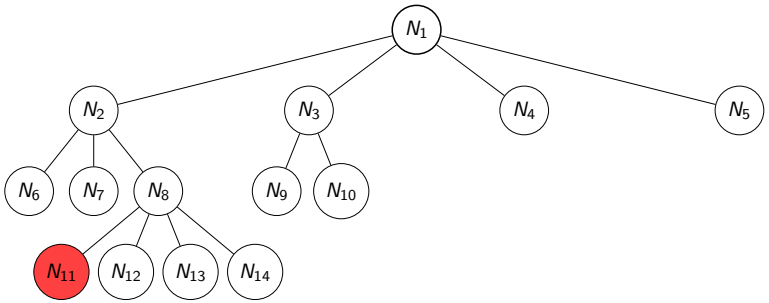


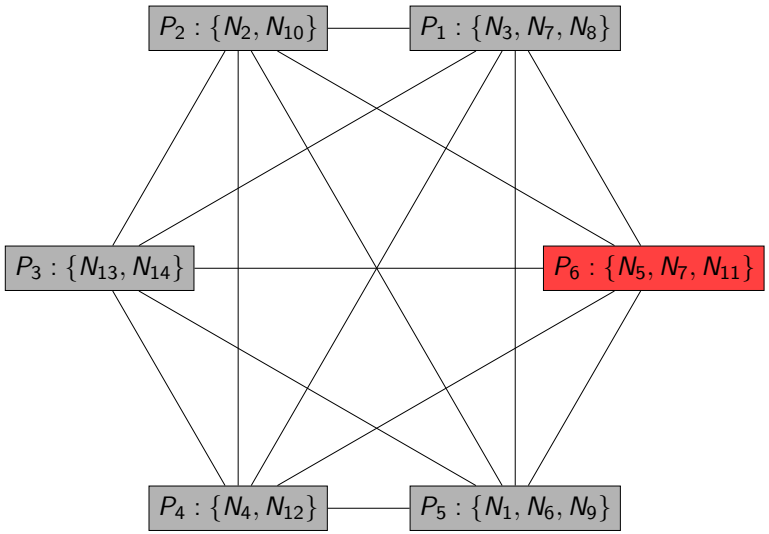
Distributed MCTS





Distributed MCTS





Returning Values

Returning Values

Returning Values

Returning Values

Returning Values

Returning Values

Returning Values

Returning Values

Returning Values

Returning Values

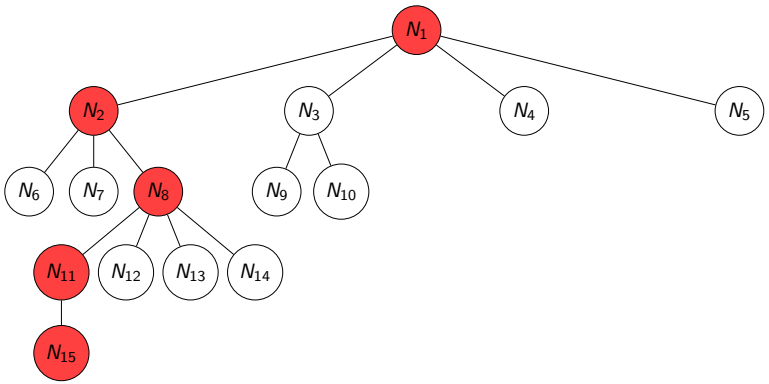
Returning Values

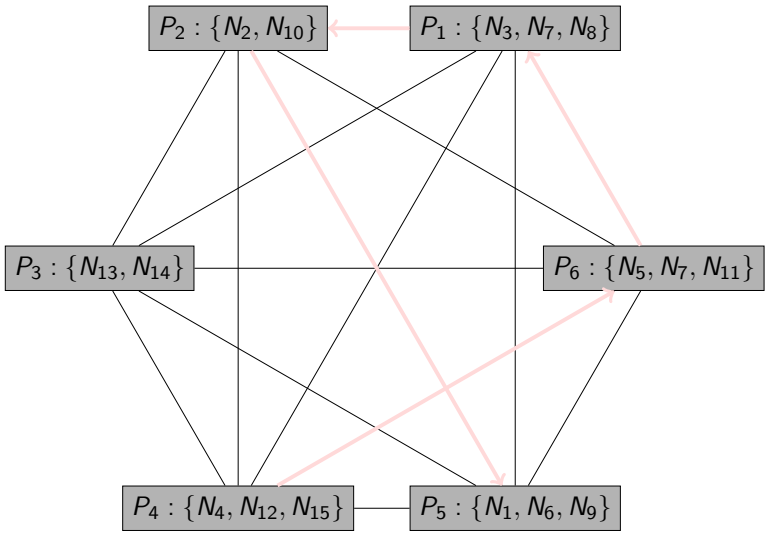
Returning Values

Returning Values

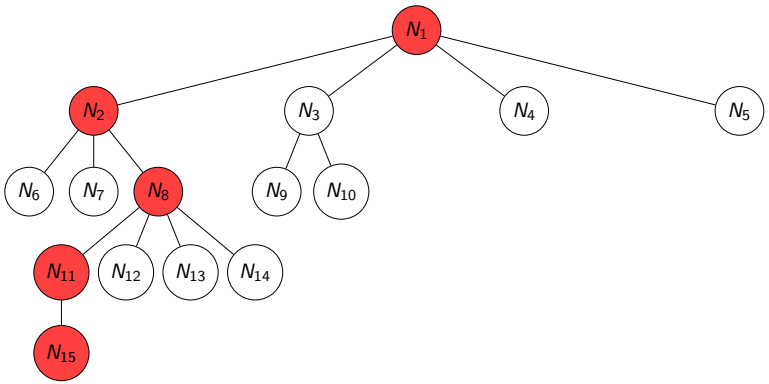
Returning Values

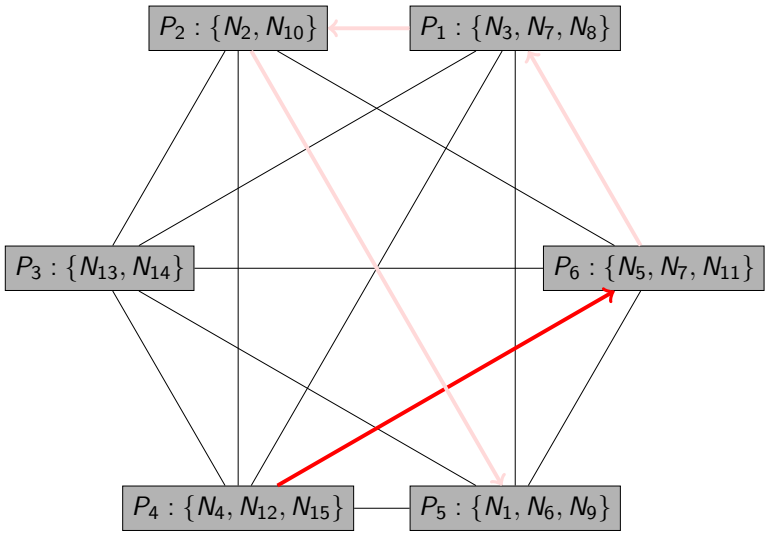
Distributed MCTS



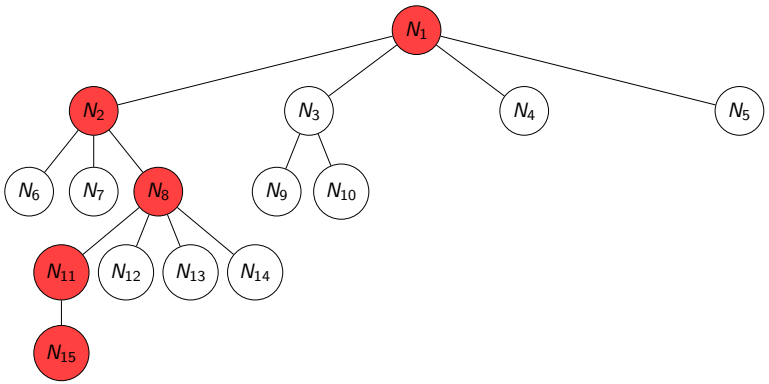


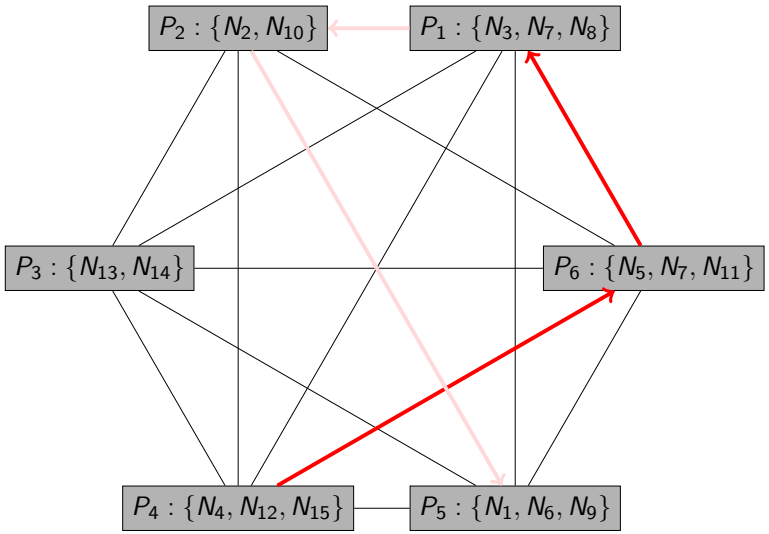
Distributed MCTS



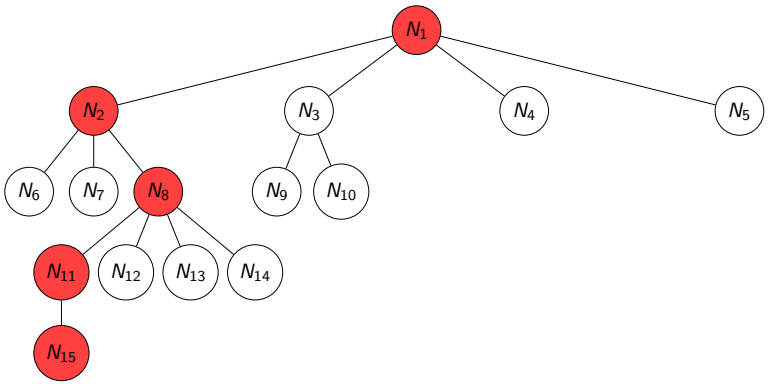


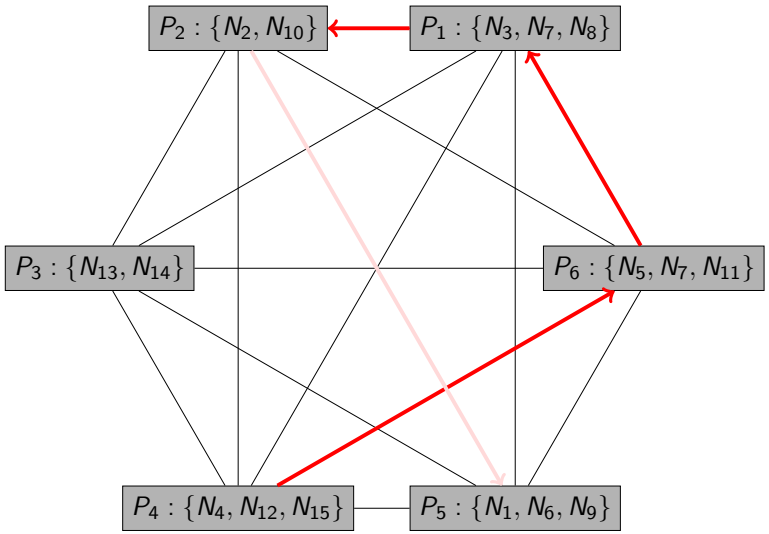
Distributed MCTS



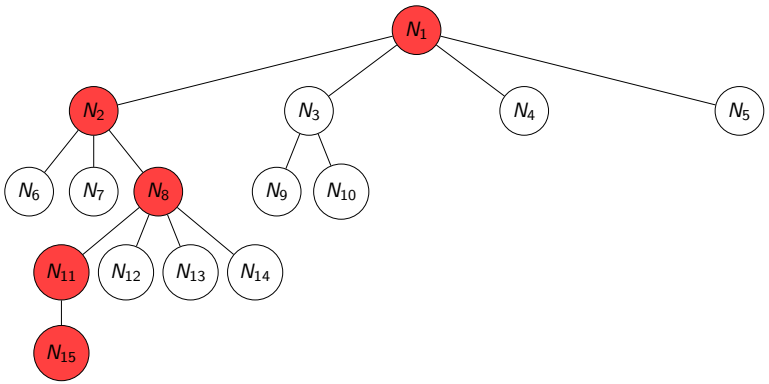


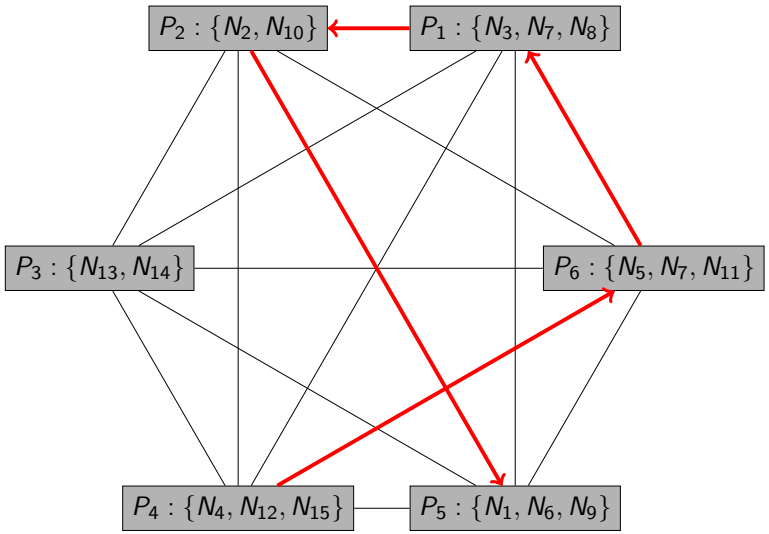
Distributed MCTS





Distributed MCTS





Dsys Services

- Remote Calls
- Synchronization primitives
- Global Addressing
- Collective Operations

Dsys Services

- Remote Calls
- Synchronization primitives
- Global Addressing
- Collective Operations

Dsys Services

- Remote Calls
- Synchronization primitives
- Global Addressing
- Collective Operations

Dsys Services

- Remote Calls
- Synchronization primitives
- Global Addressing
- Collective Operations

Collective Operations

- gather
- scatter
- allGather
- broadcast
- allToAll
- allToAllv

Collective Operations: MPI_THREAD_Helper_allGather

Thread	SendBuffer	RecvBuffer
0	[0]	[0 1 2 3 4 5 6 7]
1	[1]	[0 1 2 3 4 5 6 7]
2	[2]	[0 1 2 3 4 5 6 7]
3	[3]	[0 1 2 3 4 5 6 7]
4	[4]	[0 1 2 3 4 5 6 7]
5	[5]	[0 1 2 3 4 5 6 7]
6	[6]	[0 1 2 3 4 5 6 7]
7	[7]	[0 1 2 3 4 5 6 7]

Table of Contents

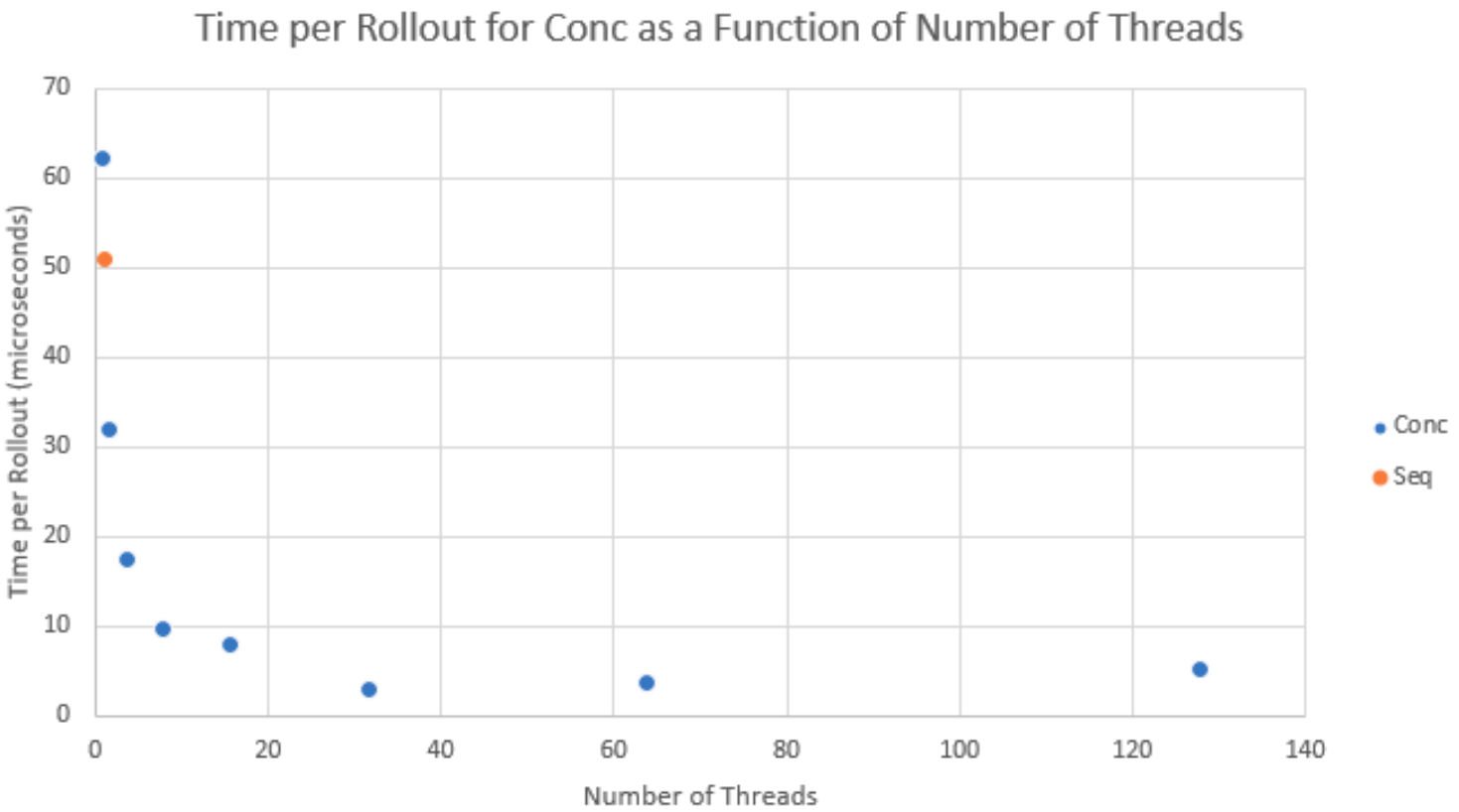
1 Hex

2 Monte Carlo Tree Search

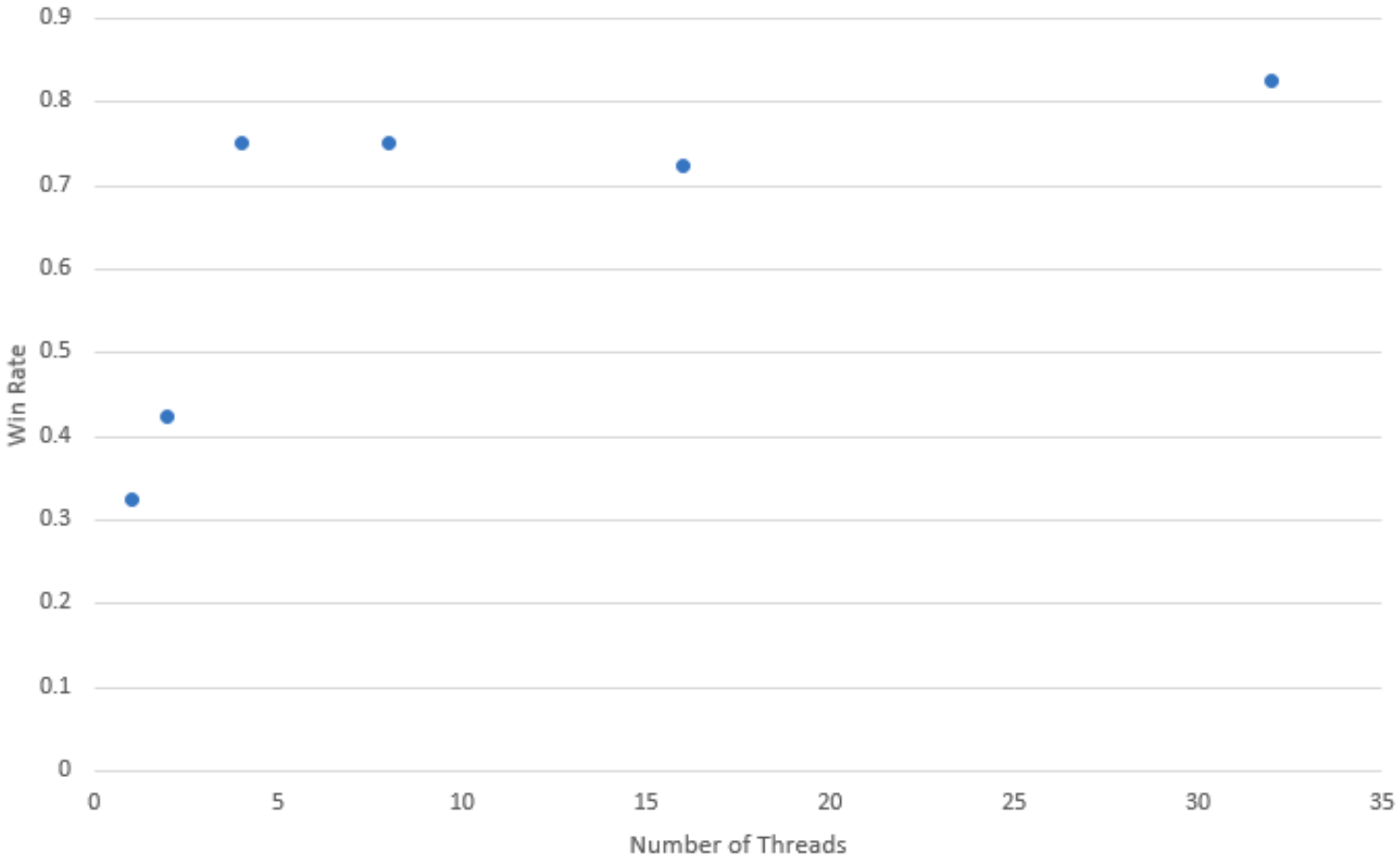
3 Dsys

4 Results and Conclusions

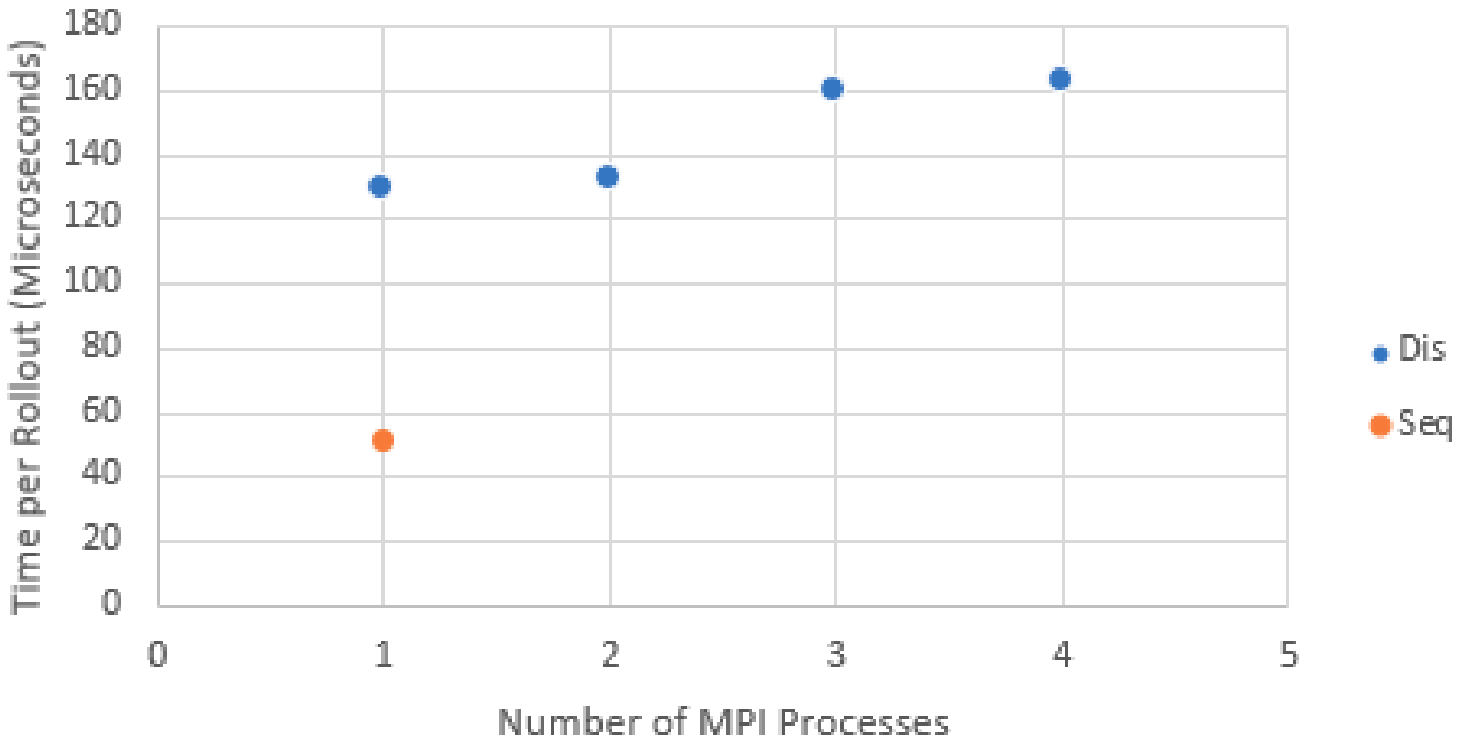
Seq vs. Conc


Time per Rollout for Conc as a Function of Number of Threads
Time per Rollout (microseconds)
70
60
50
40
30
20
10
0
0
20
40
60
80
100
120
140
Number of Threads
Conc
Seq

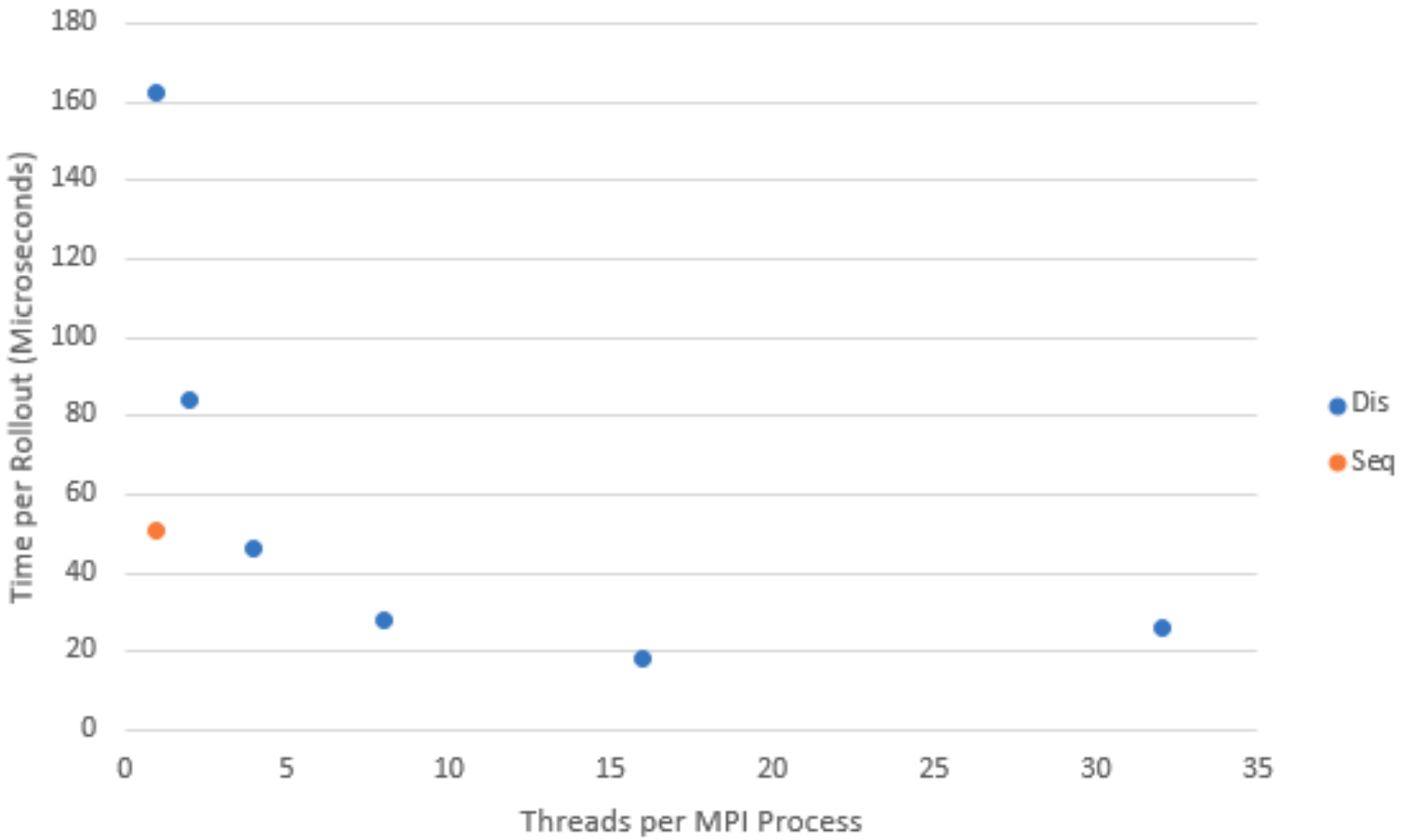
Seq vs. Conc


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Win Rate
0
5
10
15
20
25
30
35
Number of Threads

Dist


Time per Rollout (Microseconds)
180
160
140
120
100
80
60
40
20
0
0
1
2
3
4
5
Number of MPI Processes
Dis
Seq

Dist


180
160
140
120
100
80
60
40
20
0
Time per Rollout (Microseconds)
0
5
10
15
20
25
30
35
Threads per MPI Process
Dis
Seq

Conclusions

Conclusions

Questions

Conclusions

Conclusions

Questions