UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

SCS_3547_006 Intelligent Agents and Reinforcement Learning - Prof. Larry Simon

# Monte Carlo Tree Search Applied to Finite-Capacity Scheduling Optimization

Bruno Schrappe - August 2020

# Problem Definition

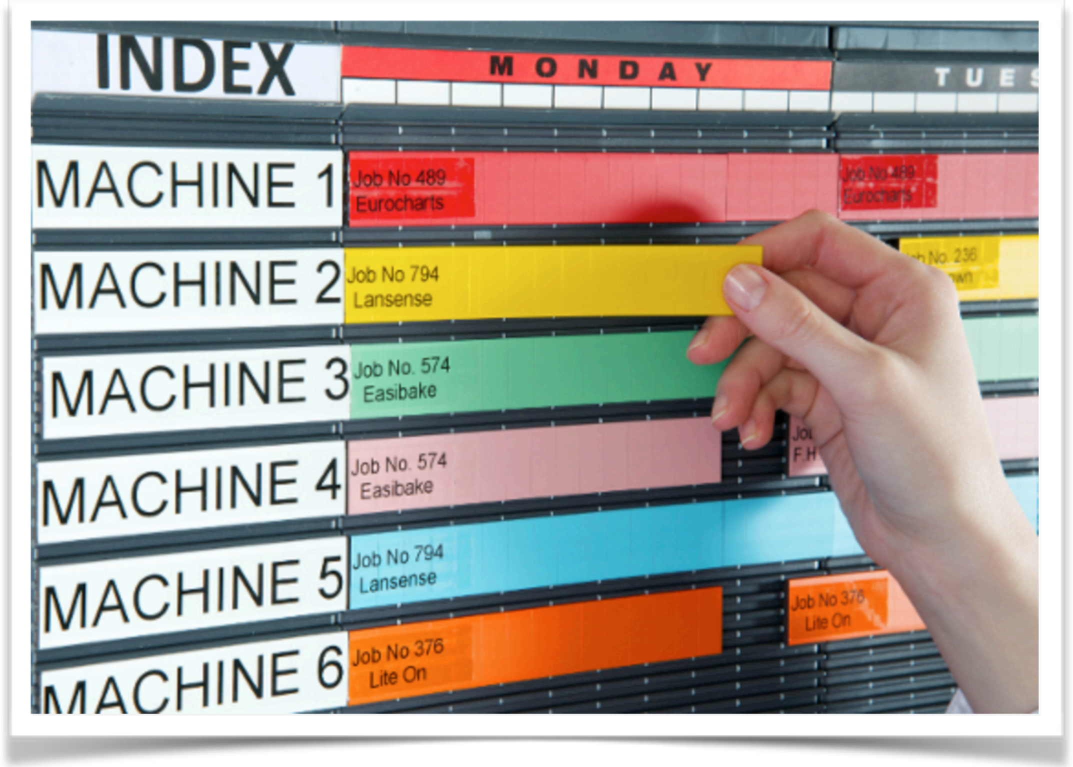


- Production scheduling involves planning tasks from a set of orders in such a way that resources are optimized by minimizing lead time or work-in-process or by maximizing throughput

- It is an NP-hard problem as the number of possible paths to a terminal state increases exponentially to an intractable number (i.e. $states \propto orders^{tasks}$)

- Industrial production scheduling has relied either on human planning or on expert systems requiring fine tuning for varying planning scenarios

- This project explores the feasibility of a universal production scheduling optimizer using Monte Carlo Tree Search as preparation for a more comprehensive solution including deep reinforcement learning

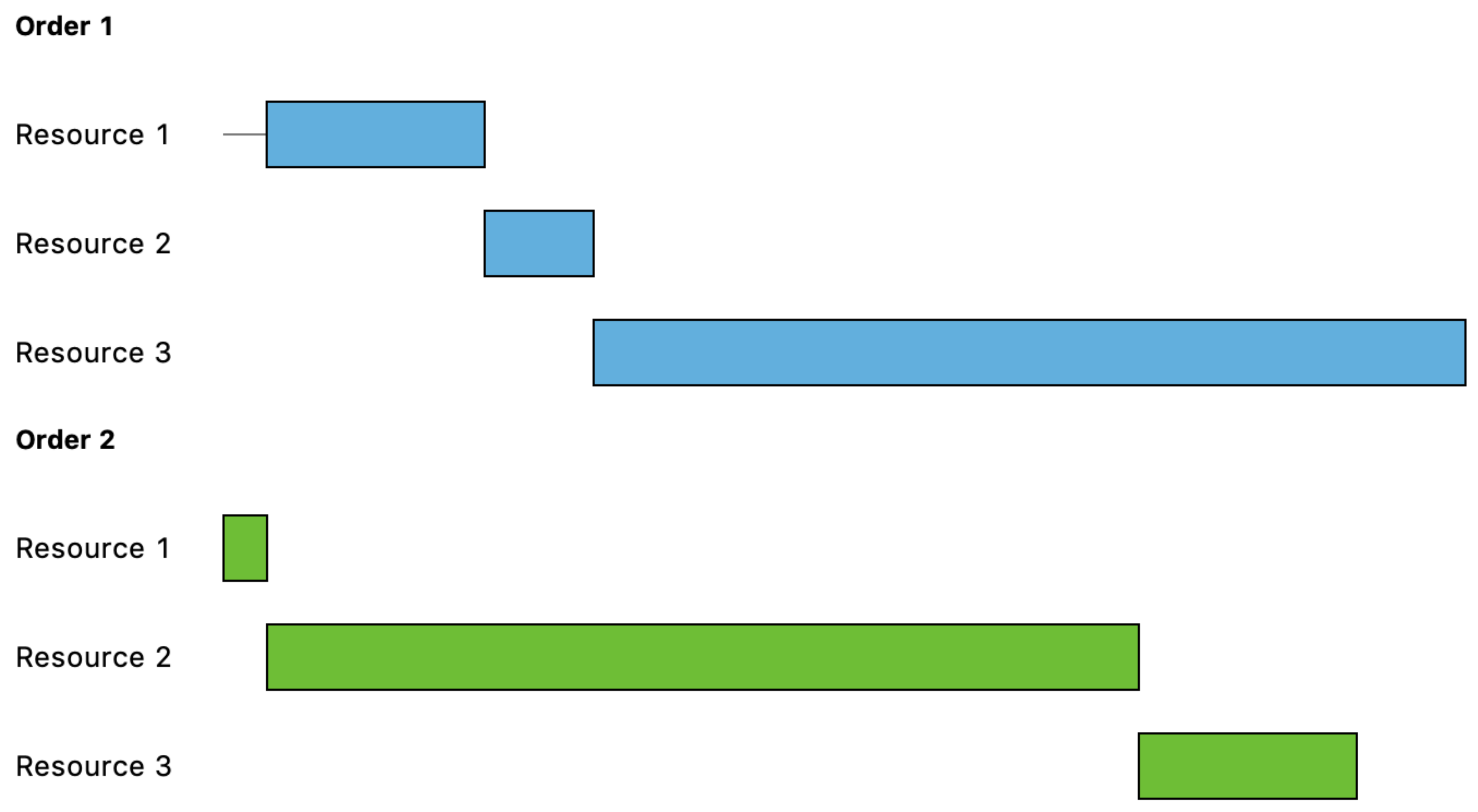


# Look-Ahead is Needed



Blue order cannot start right away (simbolized by waiting line)

Green order can start right away on Resource 1 and immediately thereafter on Resource 2



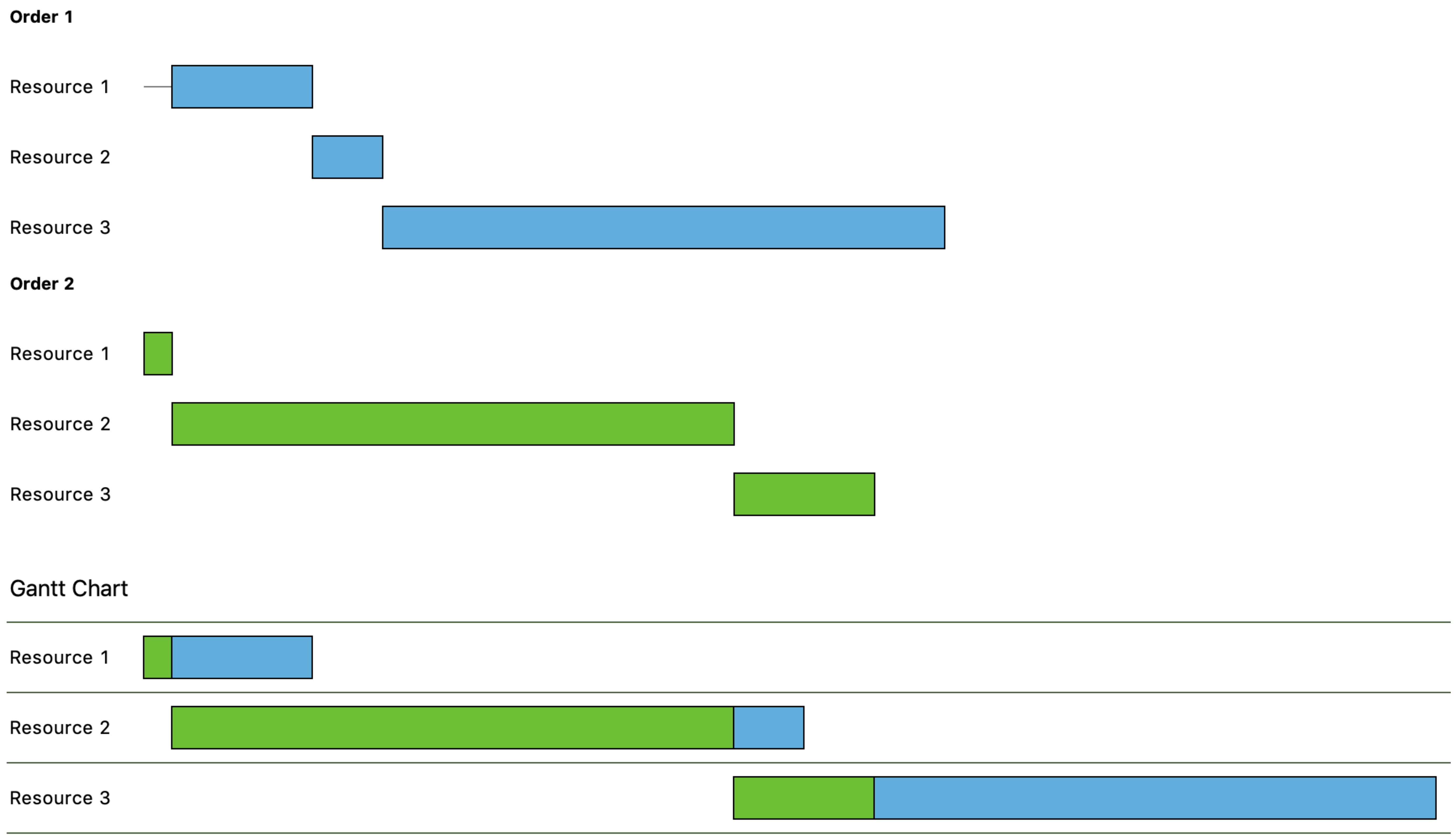
# Look-Ahead is Needed



Blue order cannot start right away (simbolized by waiting line)

Green order can start right away on Resource 1 and immediately thereafter on Resource 2

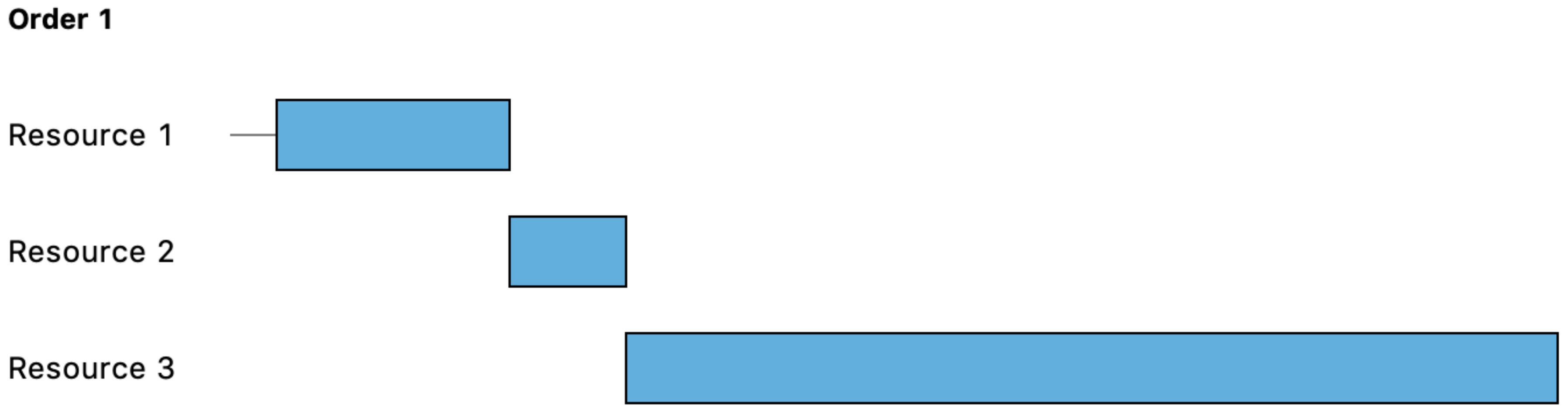
Using basic scheduling rules with no look-ahead capabilities (plan the earliest available task), a non-optimal plan is generated



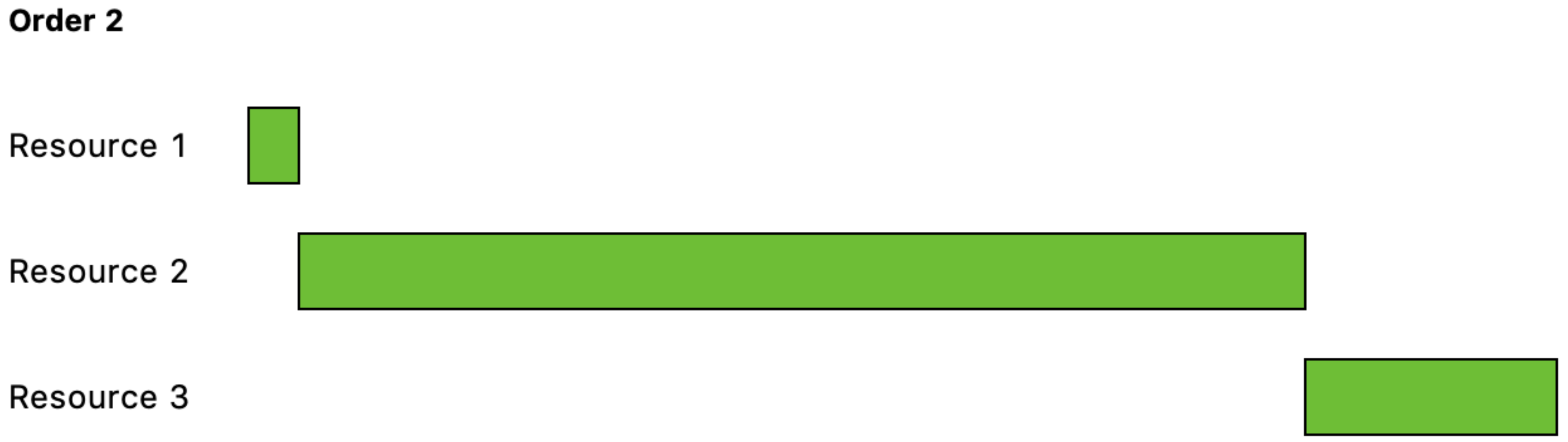
# Look-Ahead is Needed



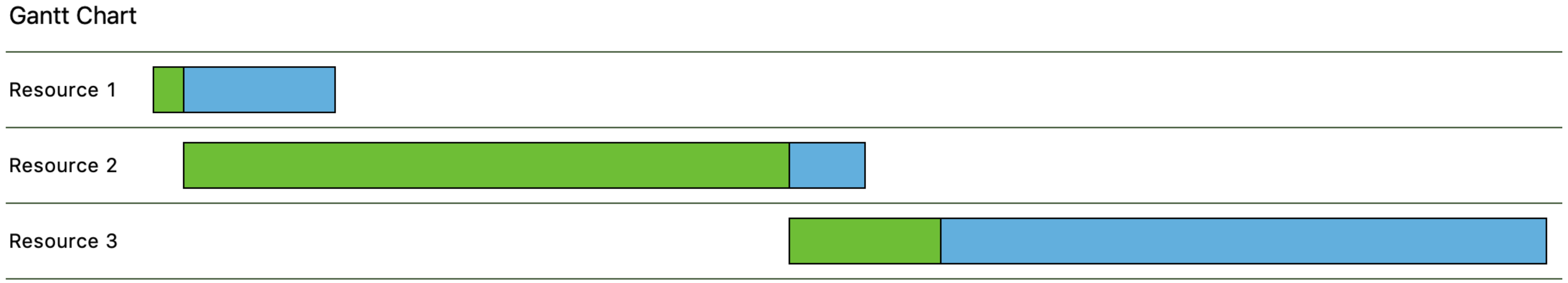
Blue order cannot start right away (simbolized by waiting line)



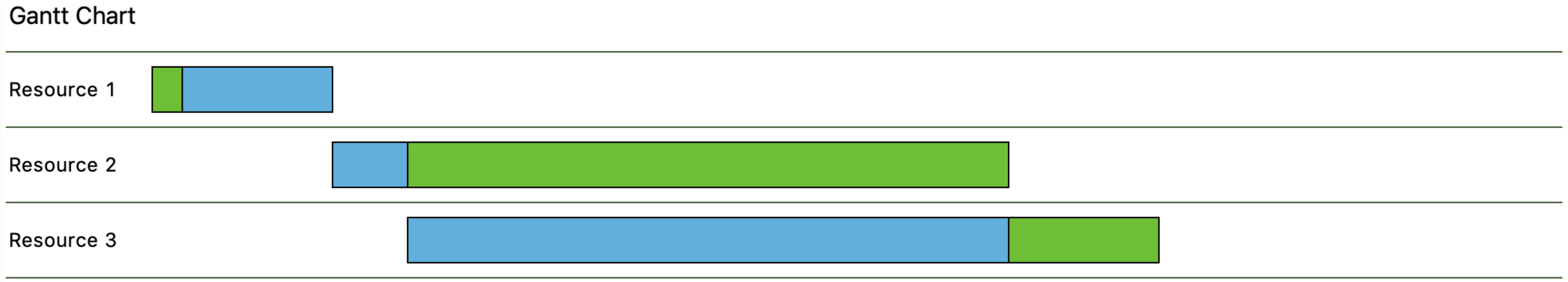

Green order can start right away on Resource 1 and immediately thereafter on Resource 2




Using basic scheduling rules with no look-ahead capabilities (plan the earliest available task), a non-optimal plan is generated

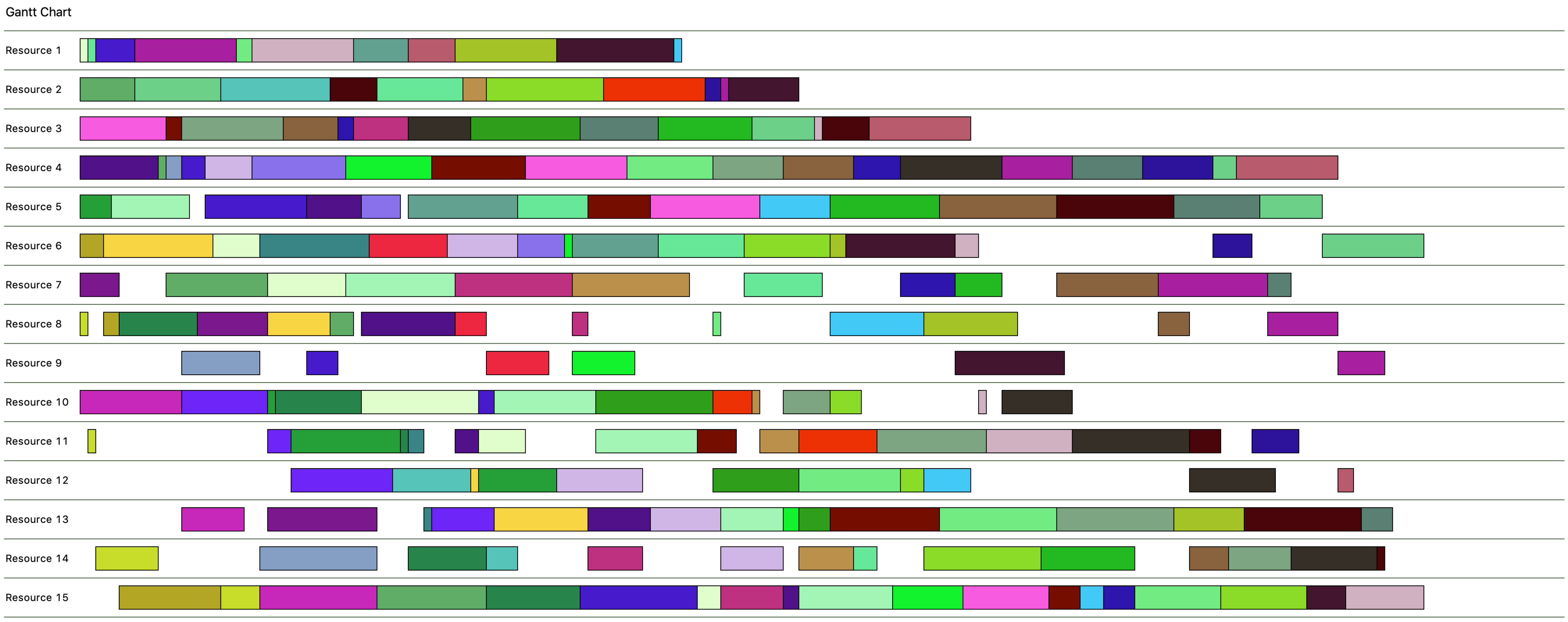



With look-ahead (i.e. full tree search) an optimal plan can be computed

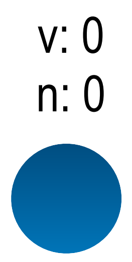


# With Many Choices for each Action…

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES



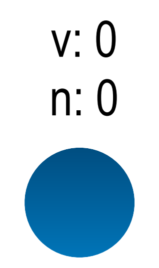
# Monte Carlo Tree Search



# Monte Carlo Tree Search



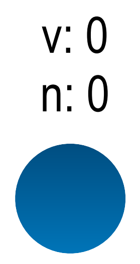
Selection



# Monte Carlo Tree Search



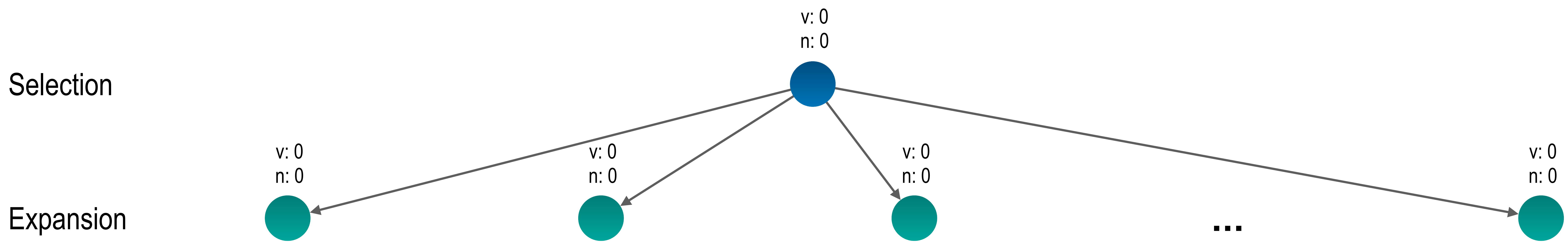
Selection




Expansion

# Monte Carlo Tree Search





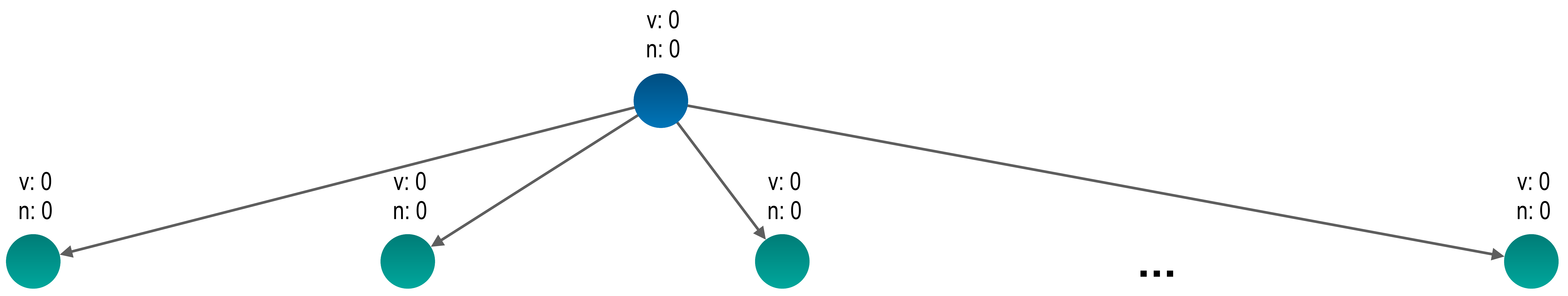
# Monte Carlo Tree Search



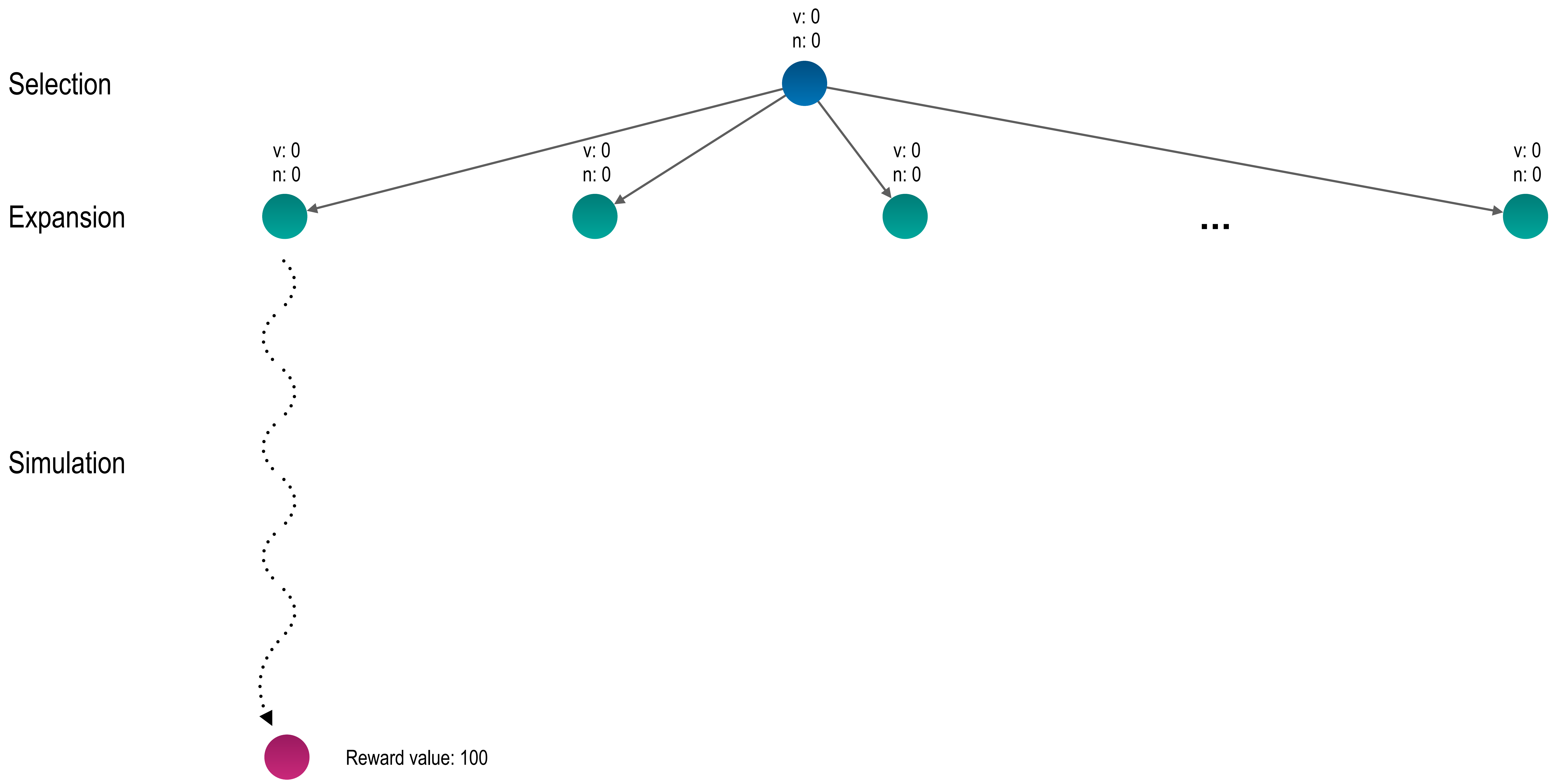
Selection

Expansion

Simulation

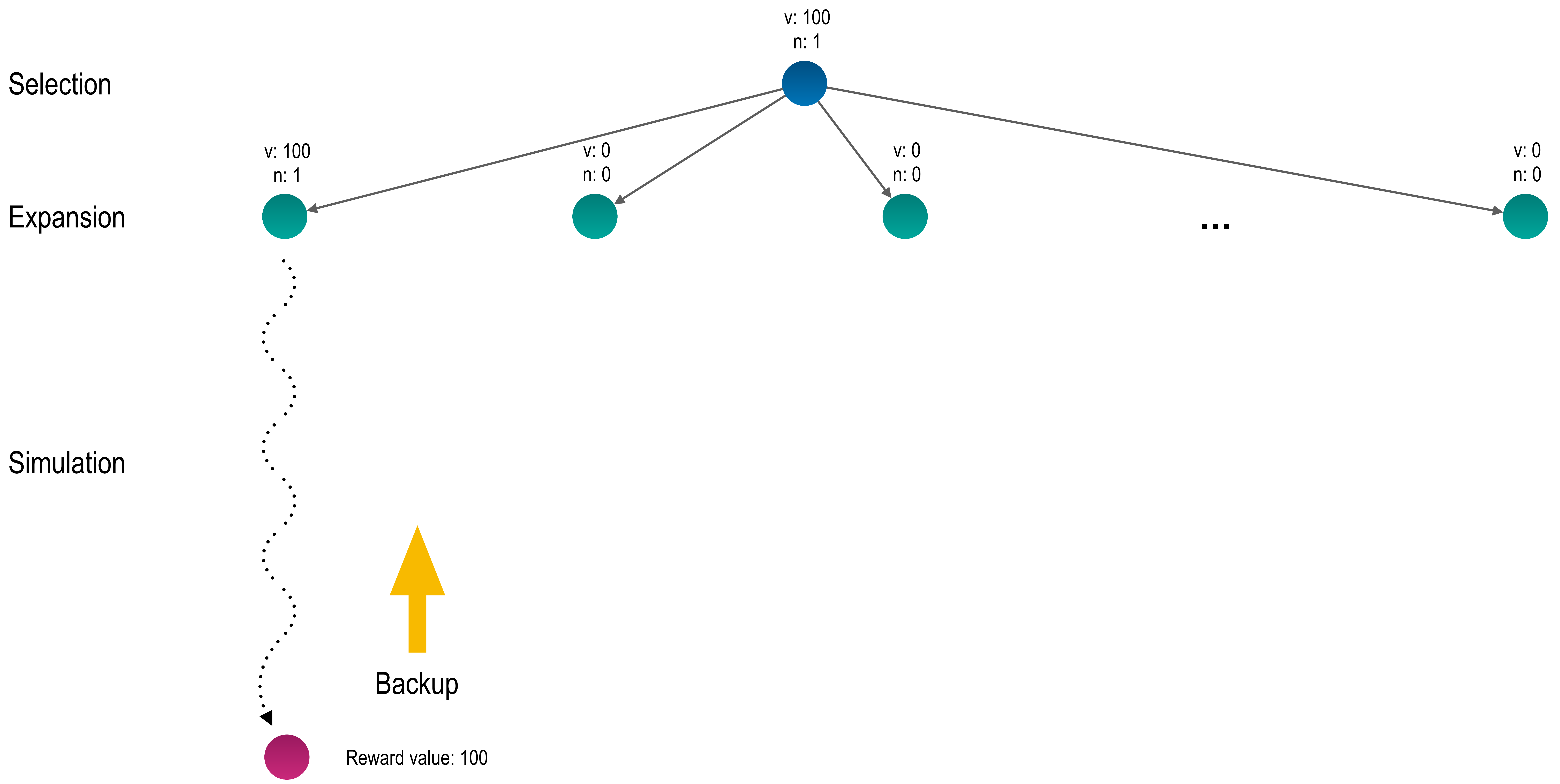


# Monte Carlo Tree Search

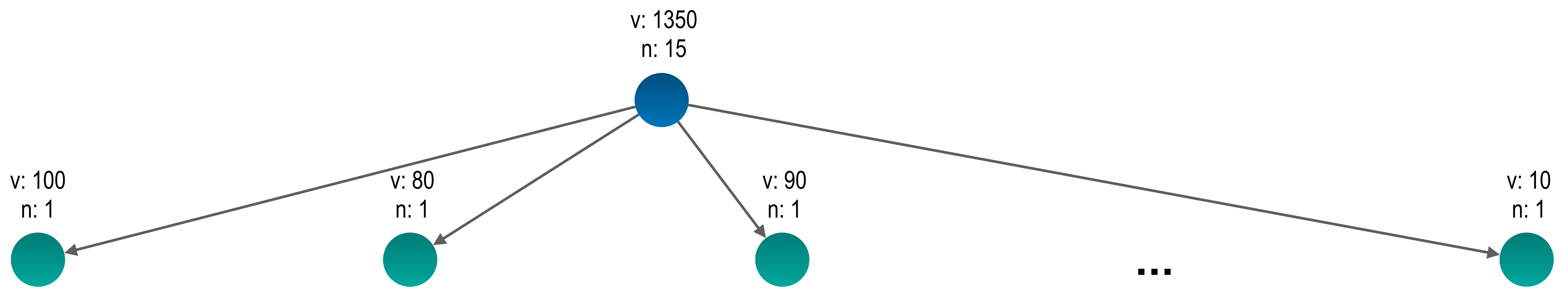


# Monte Carlo Tree Search



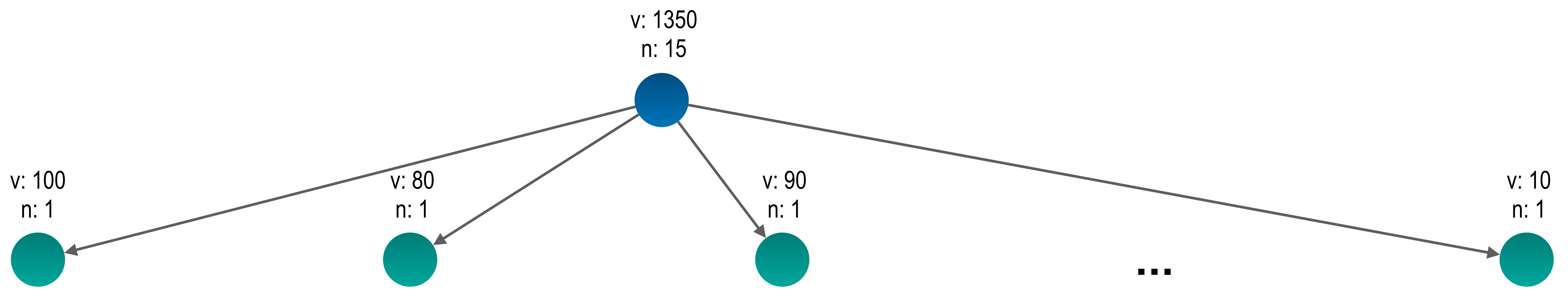


# Monte Carlo Tree Search



# Monte Carlo Tree Search

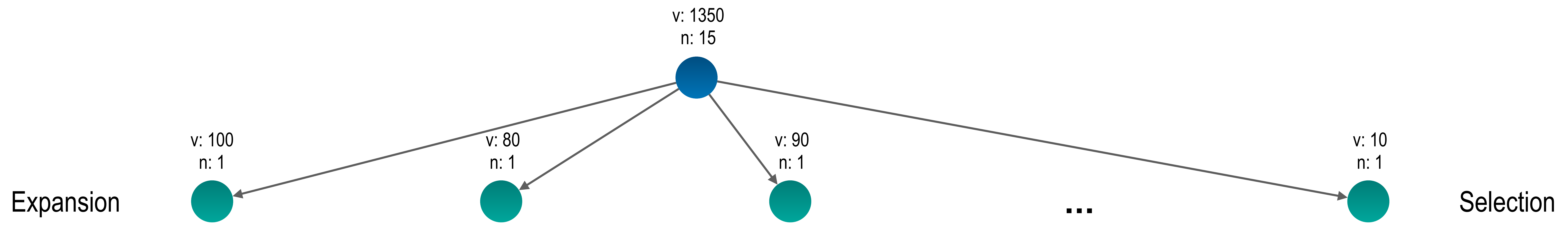





Selection

$$UCT(s_i) = \frac{v_i}{n_i} + c\sqrt{\frac{ln(N)}{n_i}}$$

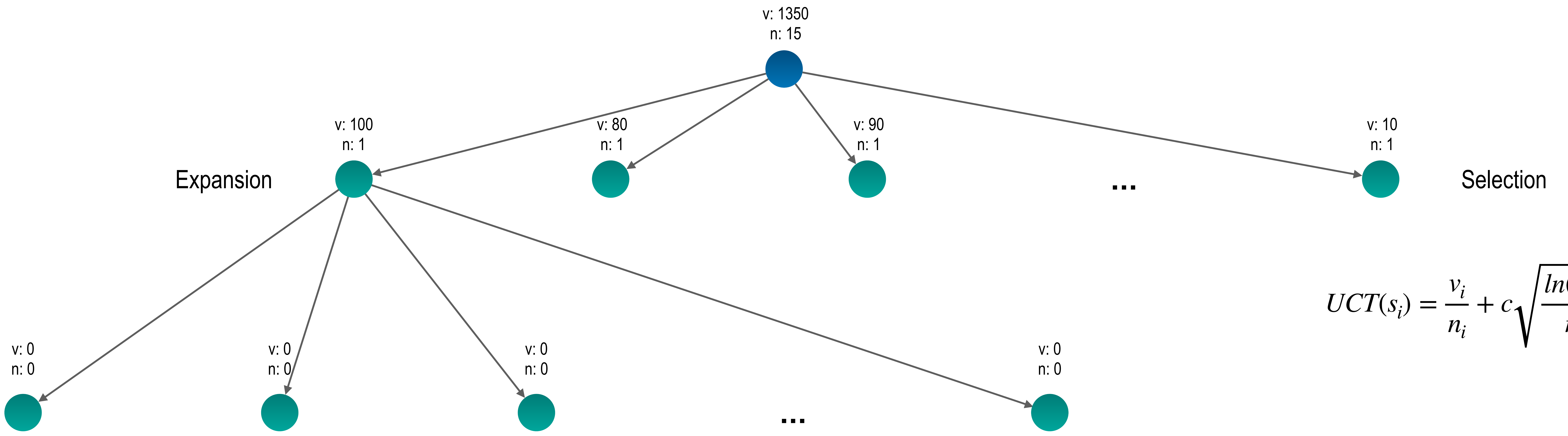
# Monte Carlo Tree Search




$$UCT(s_i) = \frac{v_i}{n_i} + c\sqrt{\frac{ln(N)}{n_i}}$$

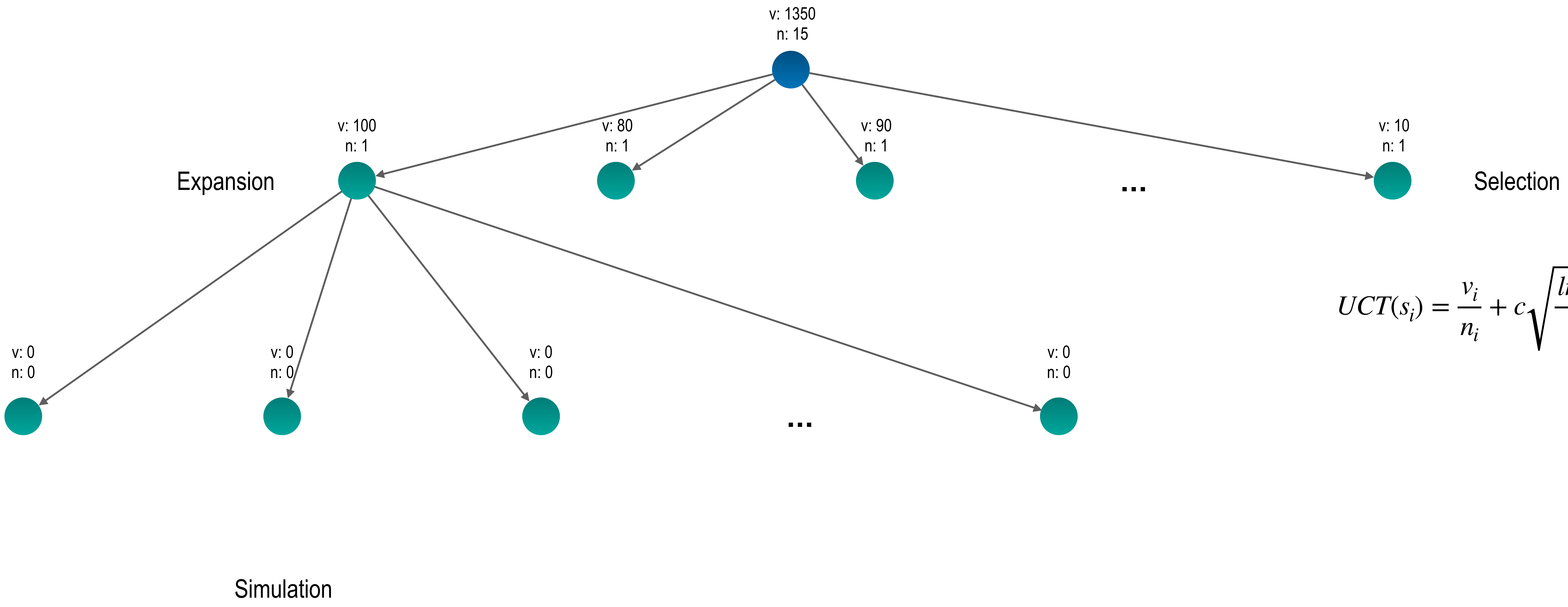
# Monte Carlo Tree Search






$$UCT(s_i) = \frac{v_i}{n_i} + c\sqrt{\frac{ln(N)}{n_i}}$$

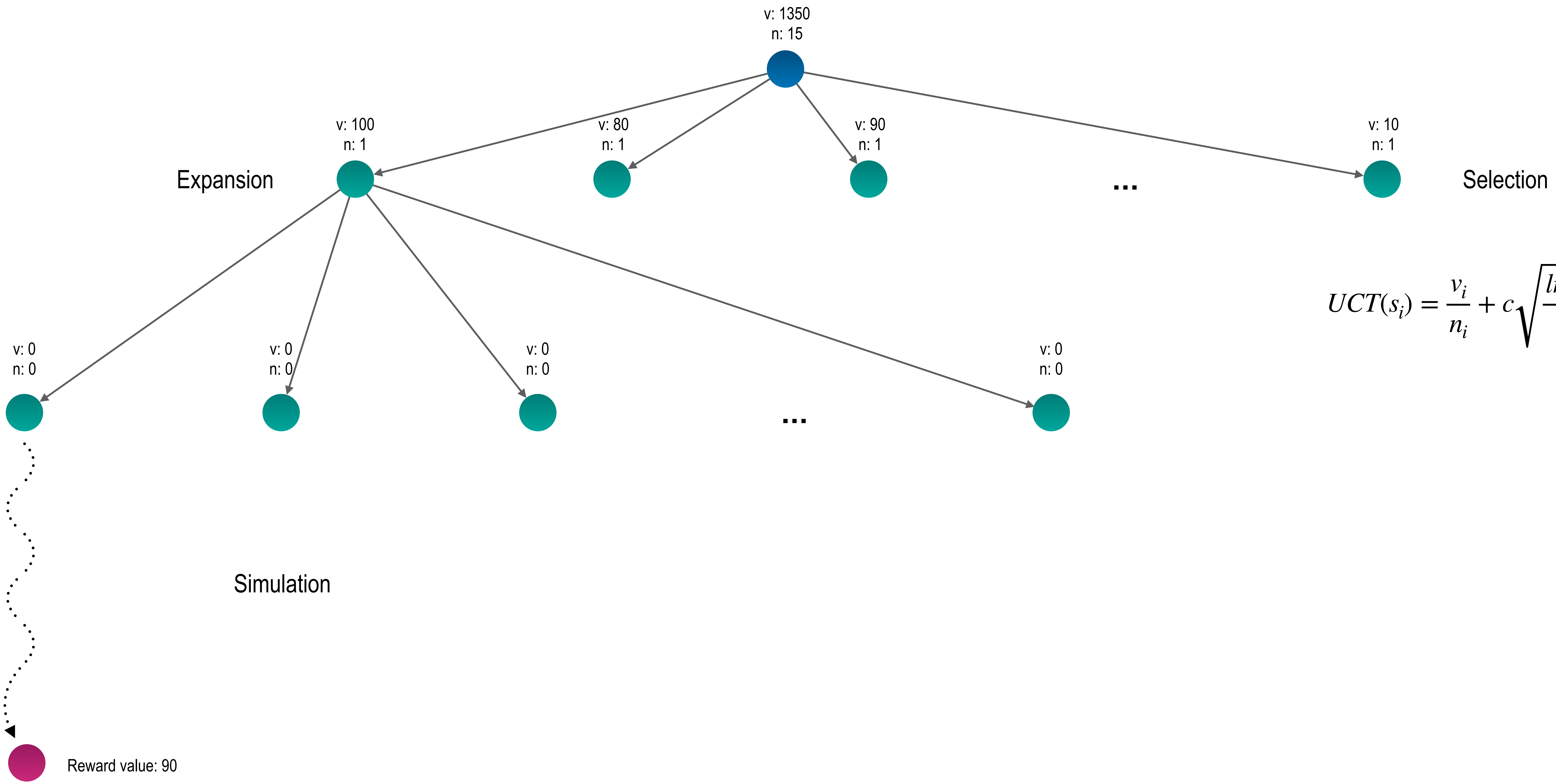
# Monte Carlo Tree Search




$$UCT(s_i) = \frac{v_i}{n_i} + c\sqrt{\frac{ln(N)}{n_i}}$$

# Monte Carlo Tree Search

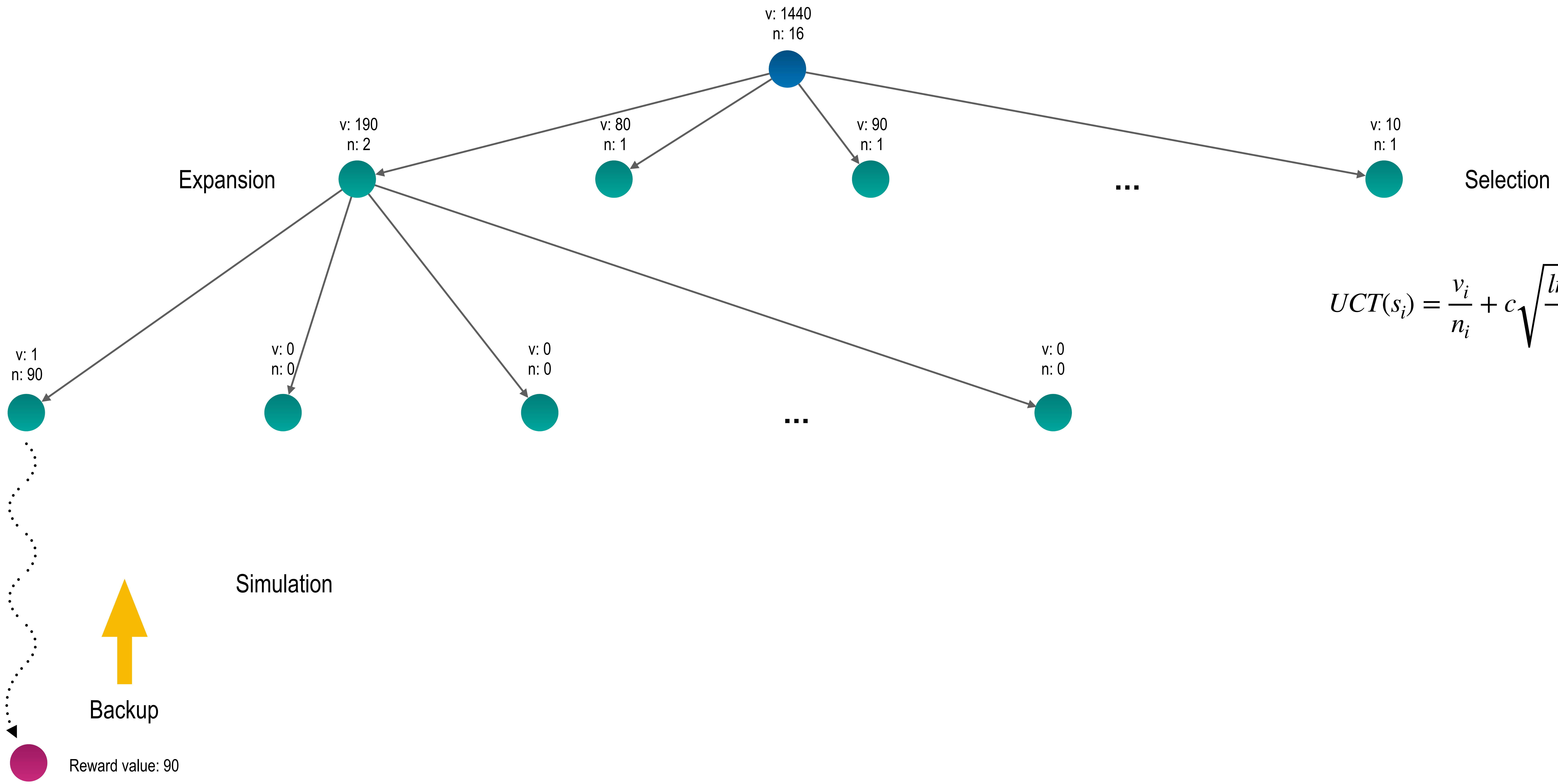





$$UCT(s_i) = \frac{v_i}{n_i} + c\sqrt{\frac{ln(N)}{n_i}}$$

# Monte Carlo Tree Search

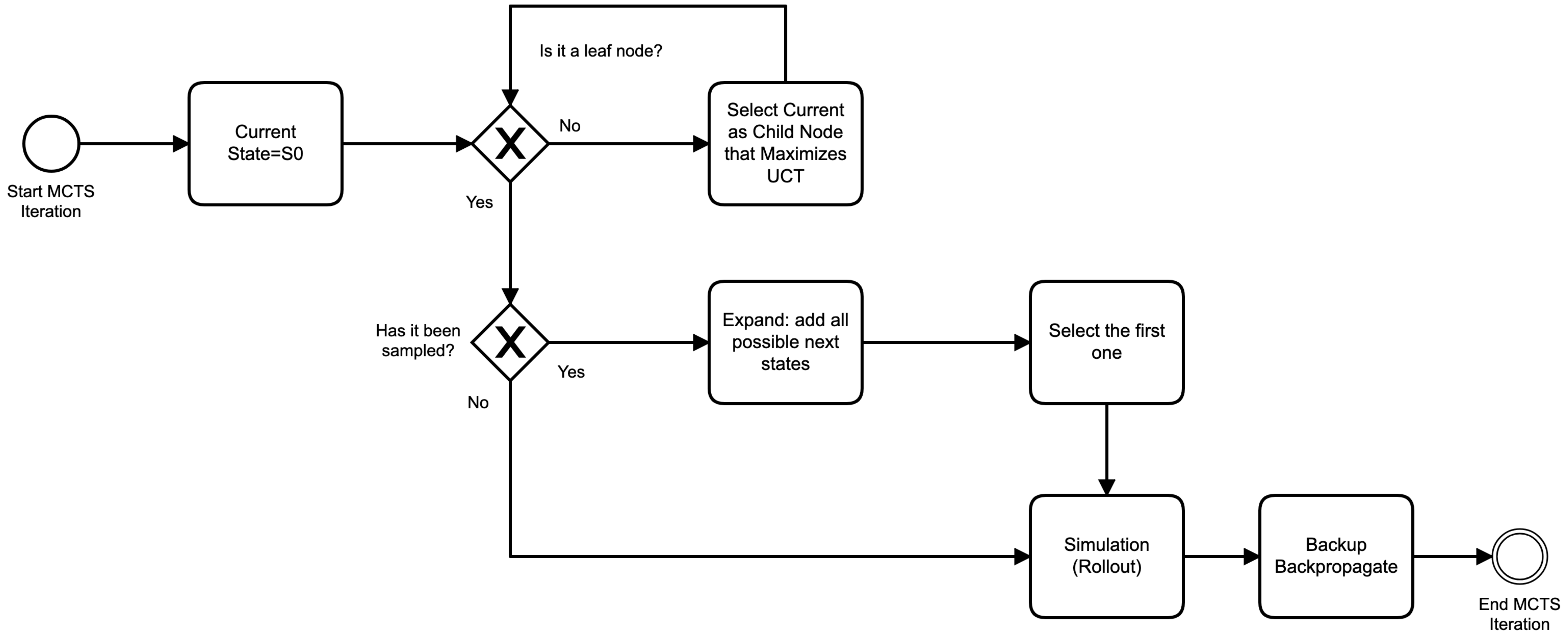




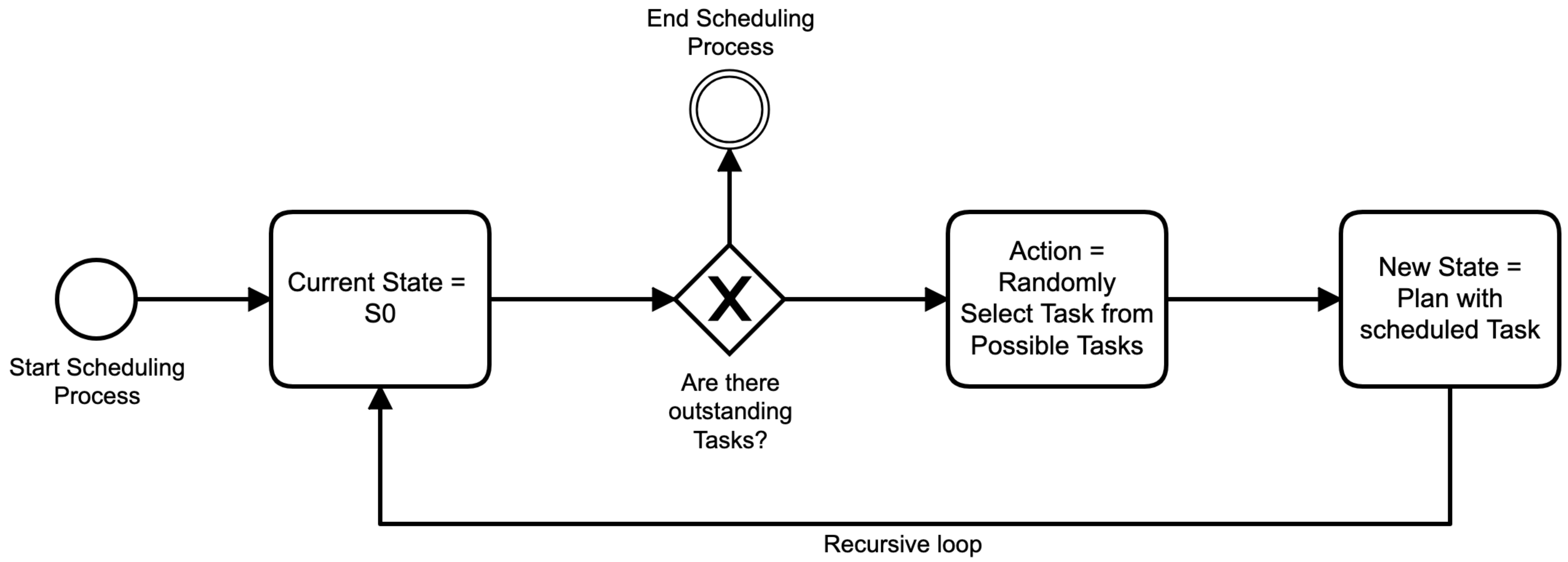

$$UCT(s_i) = \frac{v_i}{n_i} + c\sqrt{\frac{ln(N)}{n_i}}$$

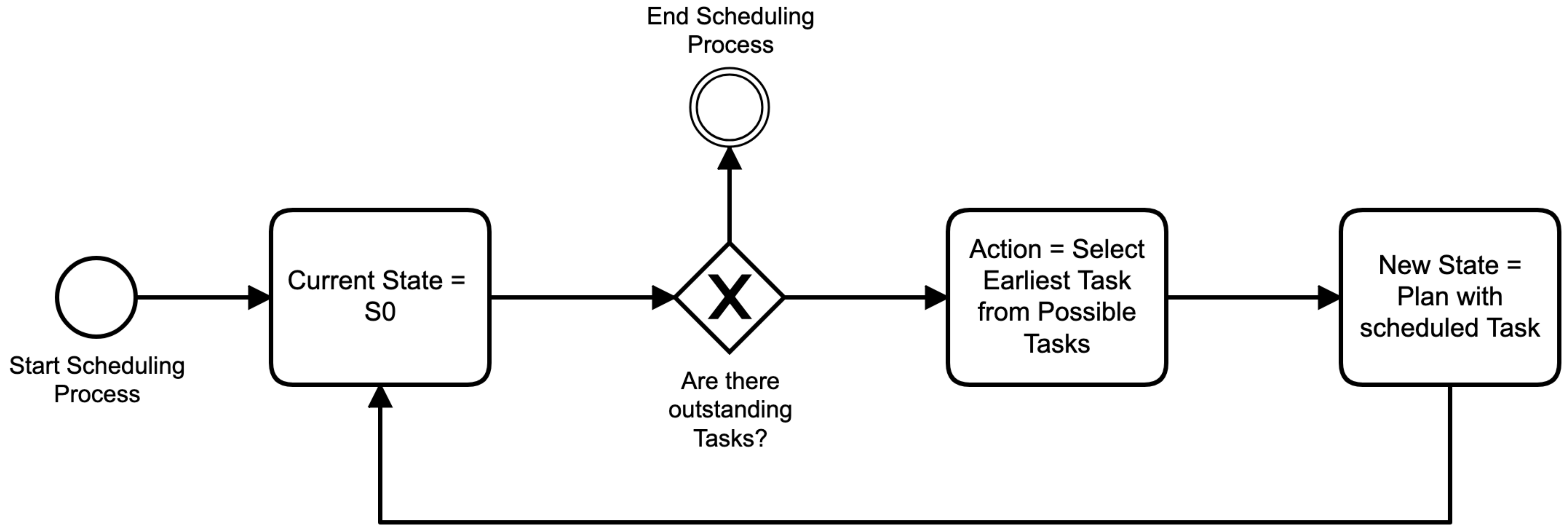
# MCTS Algorithm





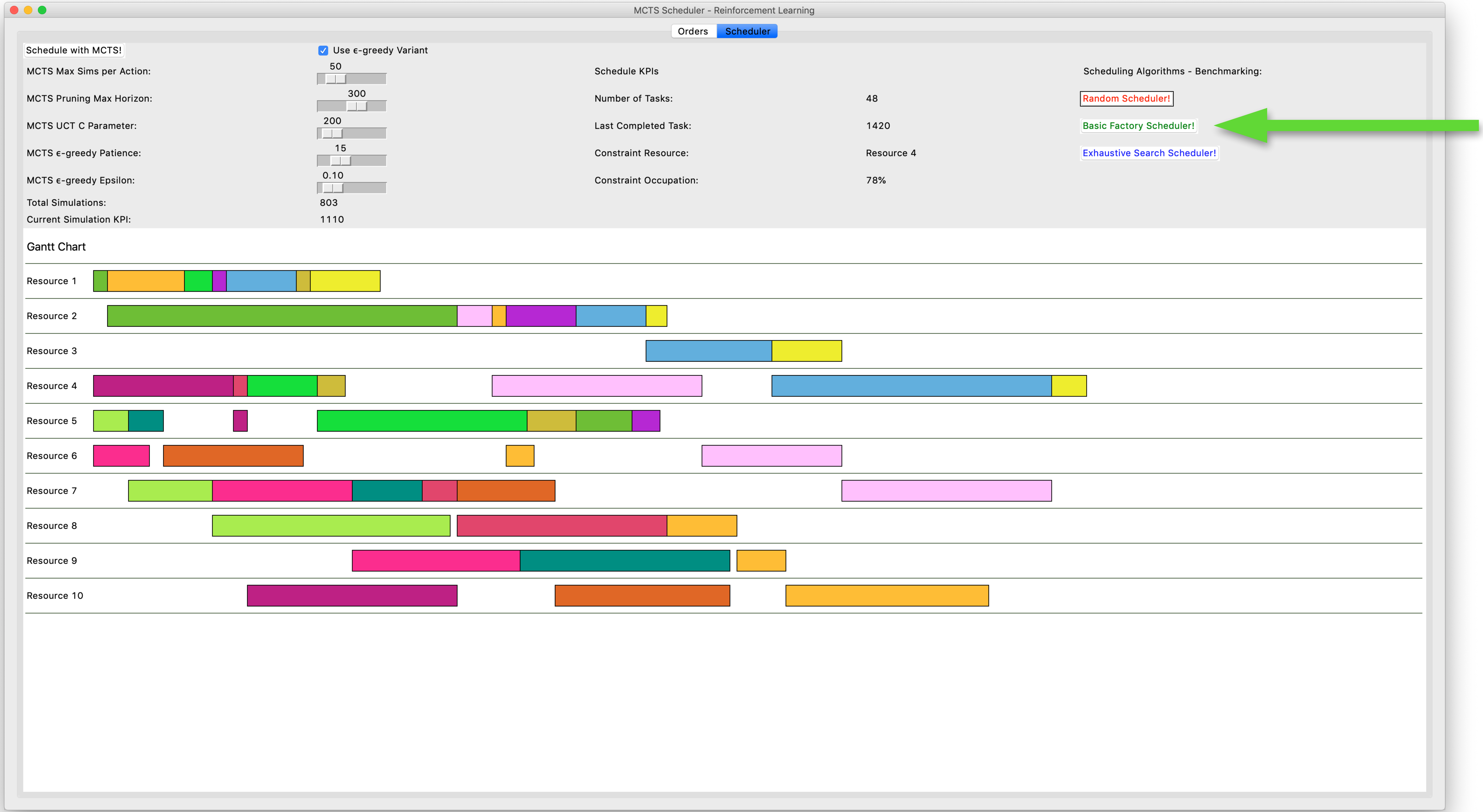
# Simulation (Rollout) & Basic Scheduling Algorithms



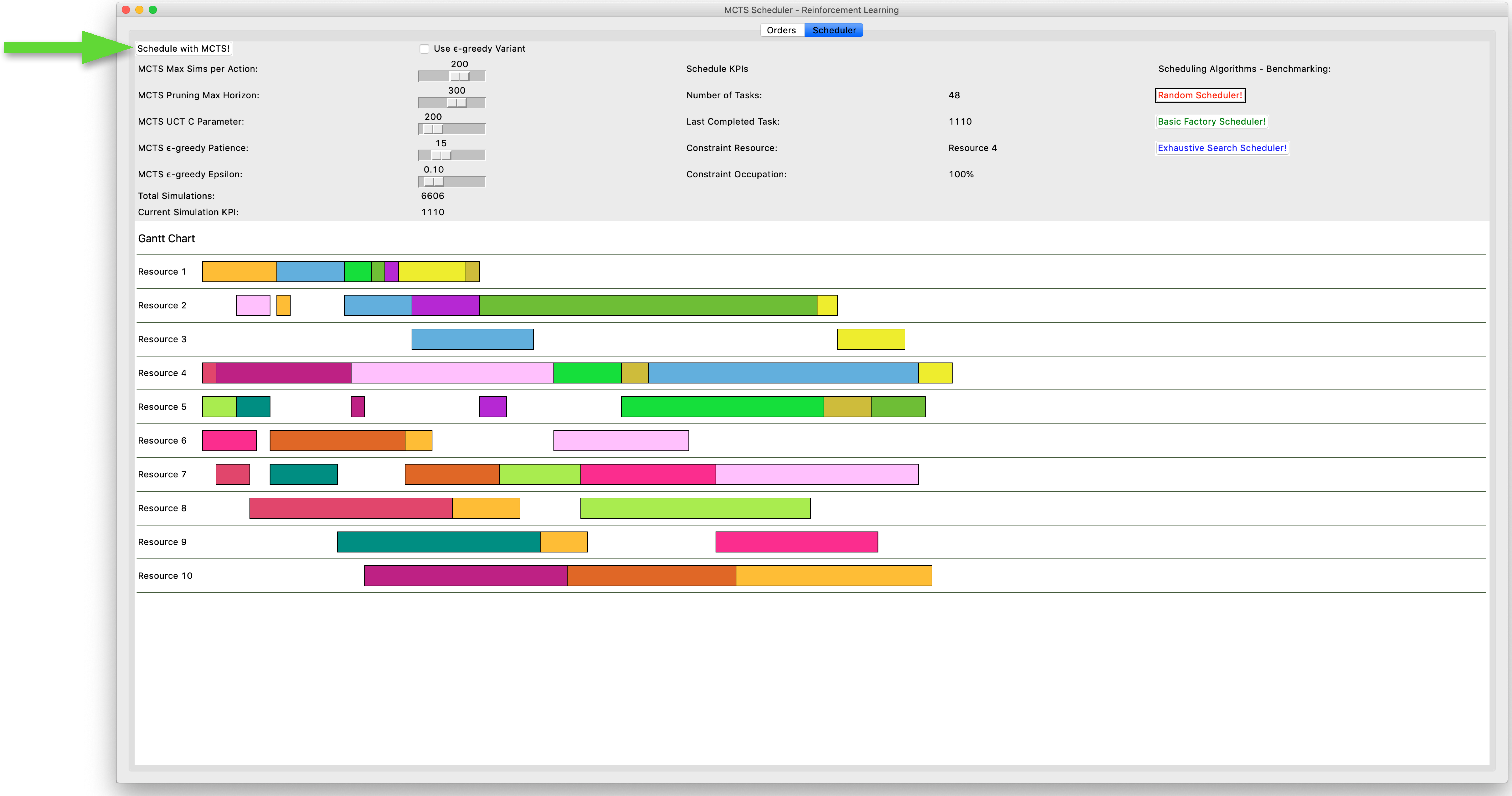



# Developed MCTS Application - Performance Comparisons

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

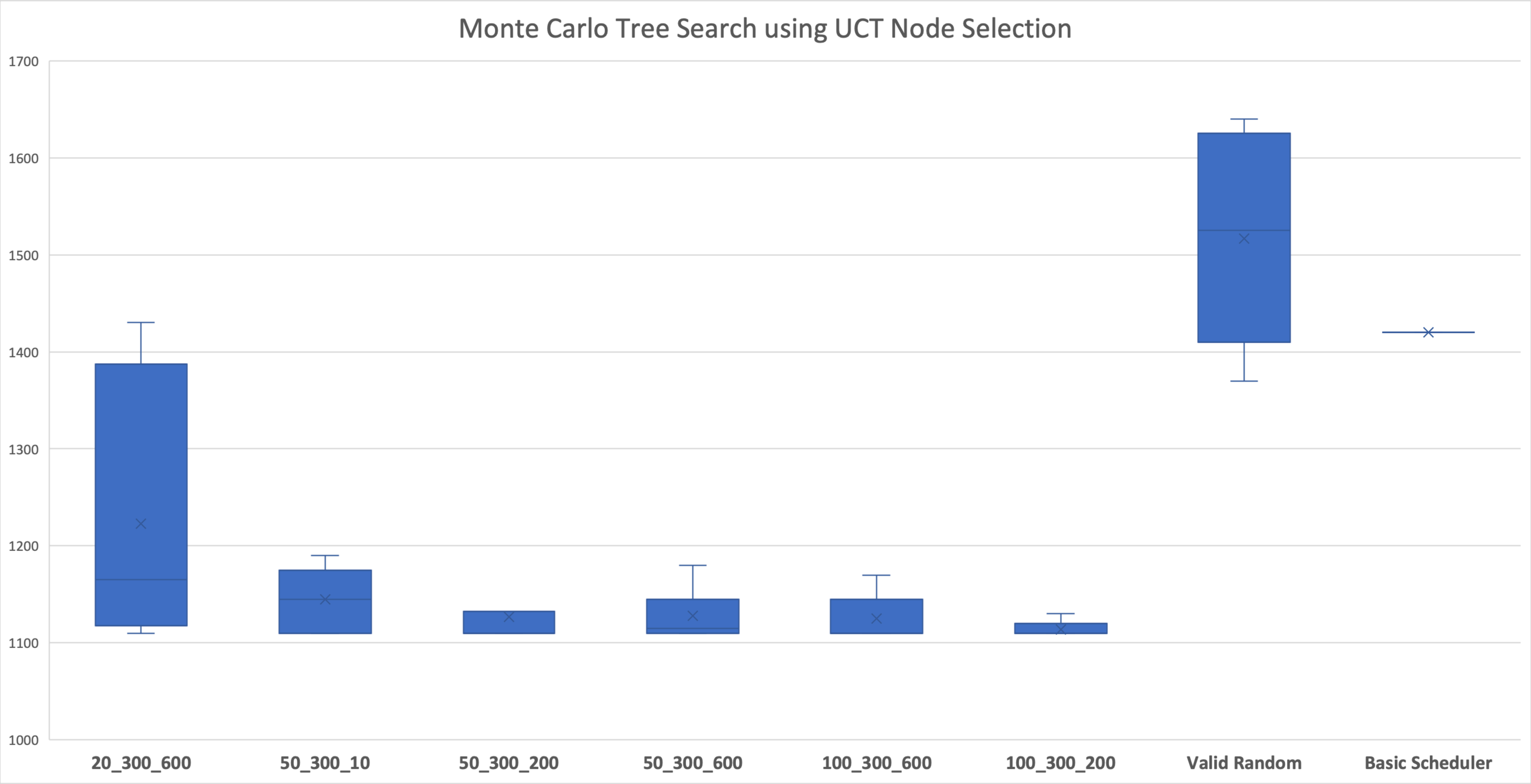


# Developed MCTS Application - Performance Comparisons



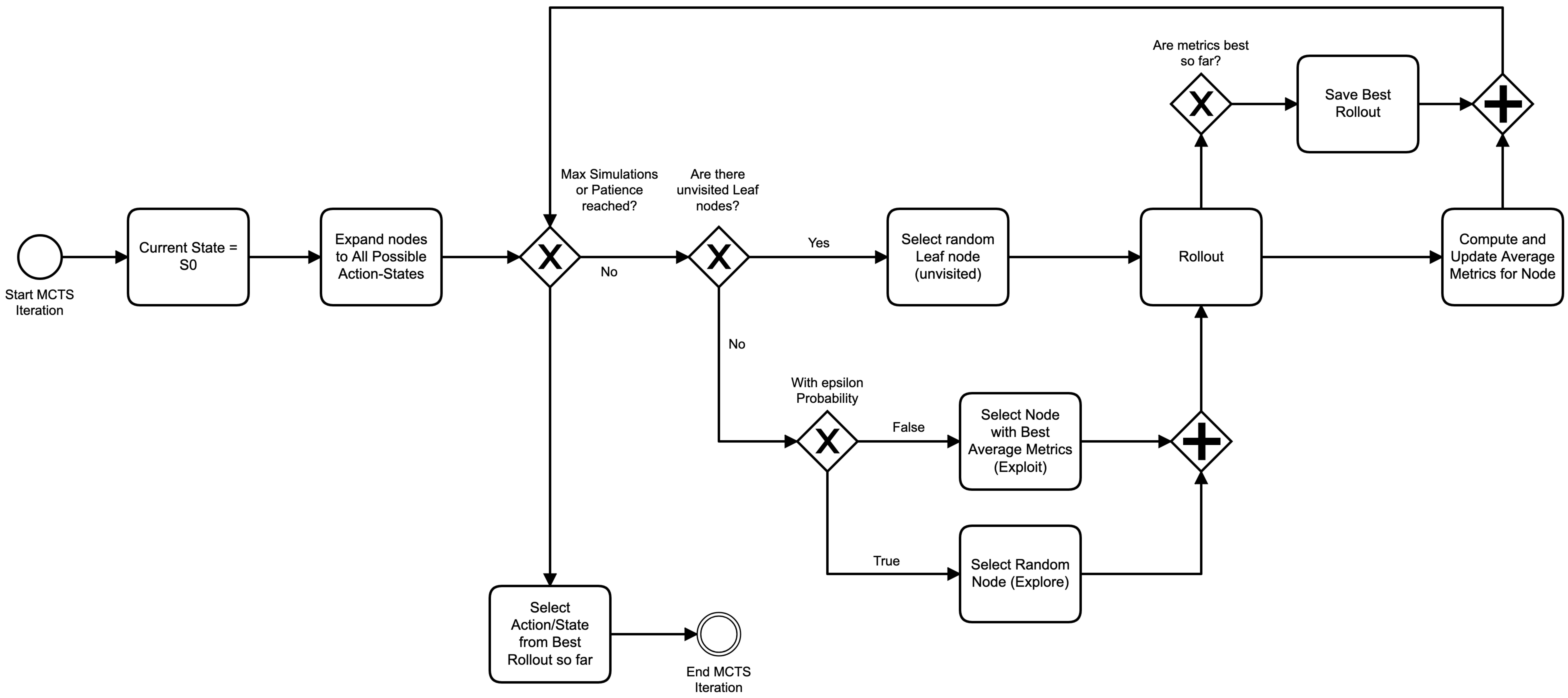
# MCTS with UCT Metrics - Parameter Tuning

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES



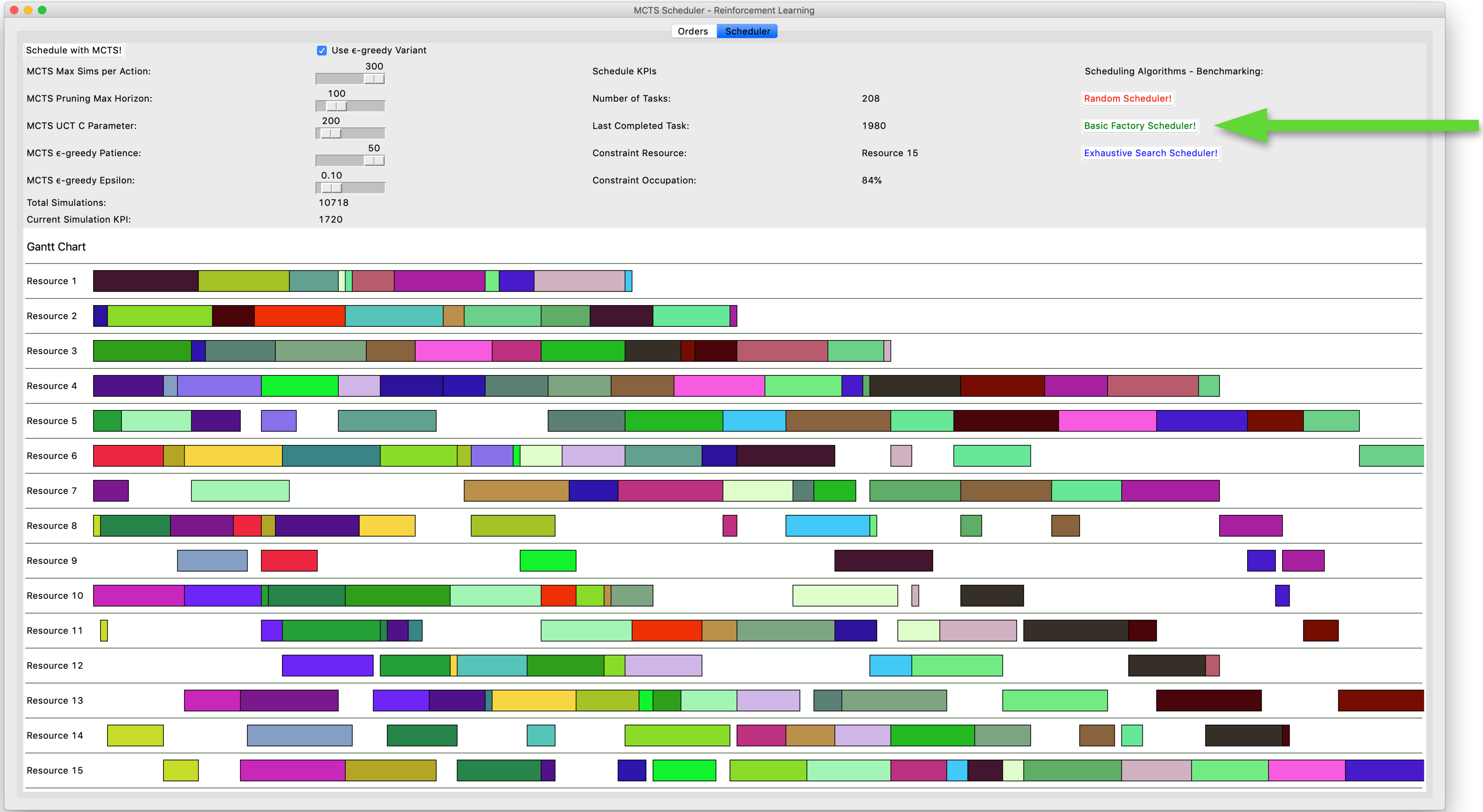

**# Simulations_Pruning Horizon_C**

# MCTS Variant with ε-greedy Decisions



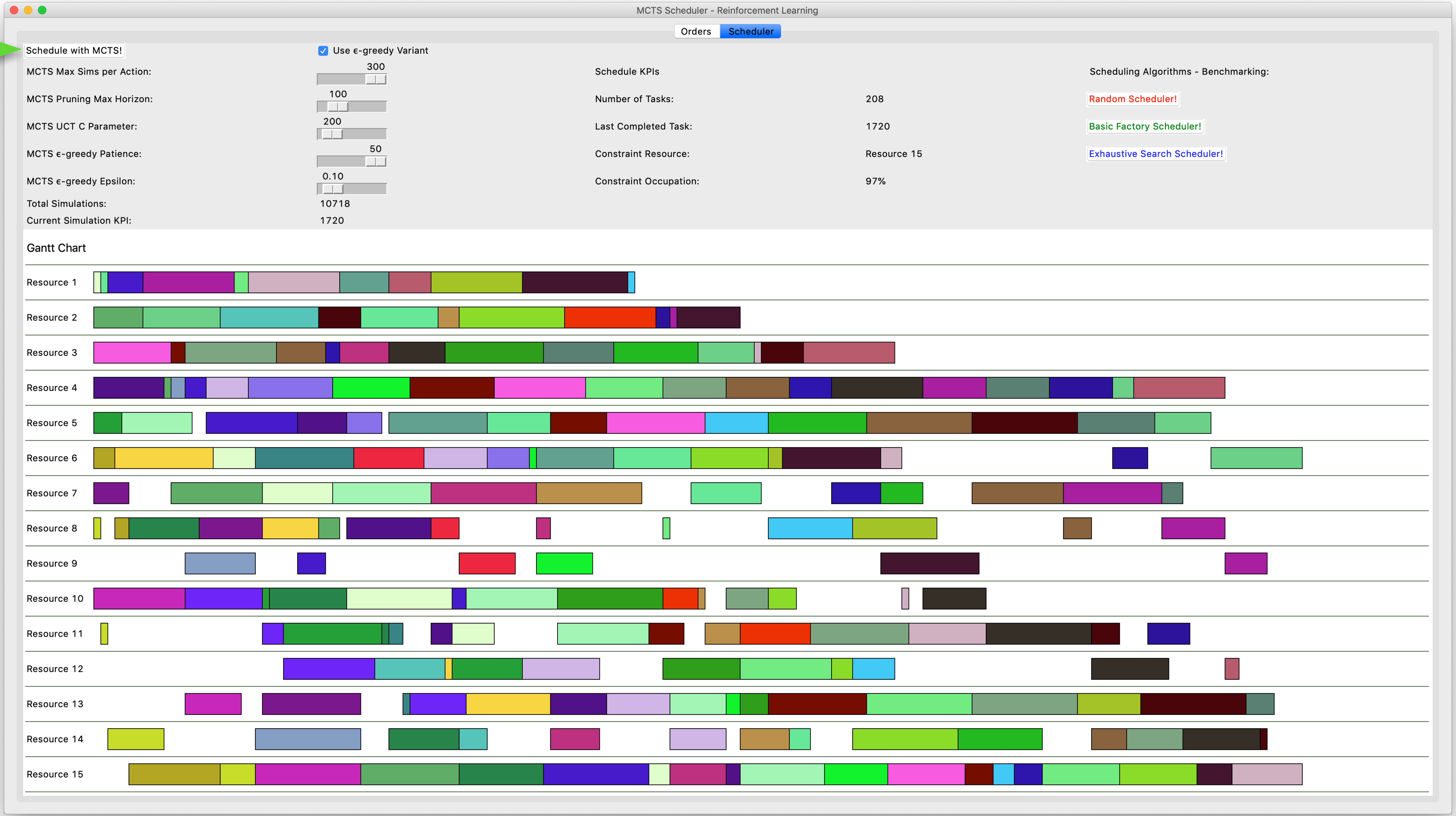
# Developed MCTS Application - Performance Comparisons

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES



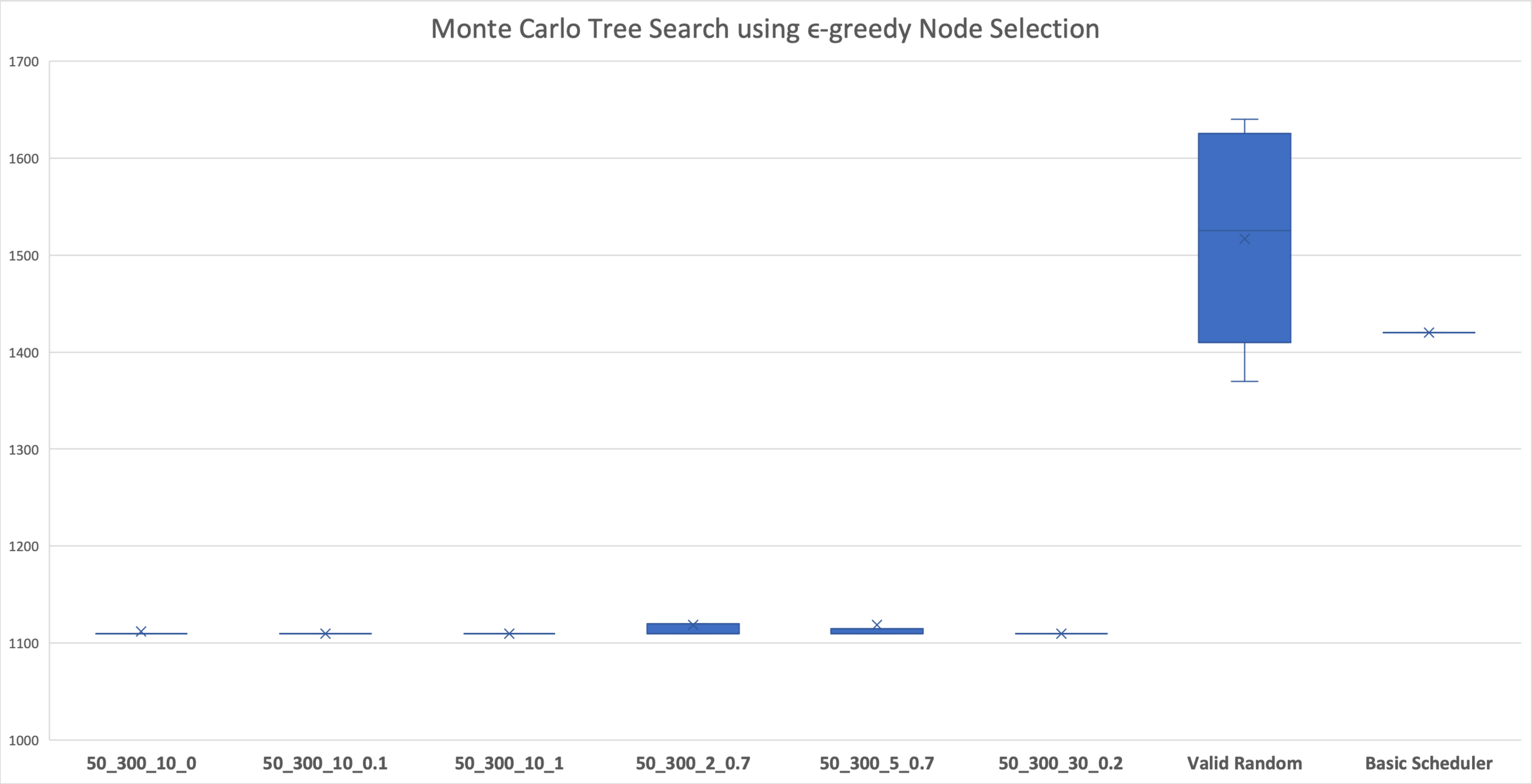
# Developed MCTS Application - Performance Comparisons

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES



# MCTS with ε-greedy Decisions Metrics






**# Simulations_Pruning Horizon_Patience_ε**

# Conclusions

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

- MCTS can be successfully applied to the industrial scheduling problem, especially if modified to leverage the fact that any simulation is not a probabilistic inference, but rather a real possibility.

- Alternative algorithms were proposed and proved to enhance performance.

- A possible next step is leveraging optimized scheduling plans to train a deep learning model and derive a Q function that in turn will speed up decisions on MCTS, in the same fashion as Alpha-Go implementations.

# References


- Yu. N. Sotskov, V. Shakhlevich. NP-hardness of shop-scheduling problems with three jobs. Discrete Applied Mathematics, Volume 59, Issue 3, 26 May 1995, Pages 237-266
- Levente Kocsis and Csaba Szepesvári, Bandit based Monte-Carlo Planning. European Conference on Machine Learning, ECML 2006 pp 282-293
- Marco Lubosch, Martin Kunath, Herwig Winkler. Industrial scheduling with Monte Carlo tree search and machine learning. 51st CIRP Conference on Manufacturing Systems
- Bernd Waschneck, André Reichstaller, Lenz Belzner, Thomas Altenmüller, Thomas Bauernhansl, Alexander Knapp, Andreas Kyek. Optimization of global production scheduling with deep reinforcement learning. 51st CIRP Conference on Manufacturing Systems
- Frank Benda, Roland Braune, Karl F. Doerner, Richard F. Hartl. A machine learning approach for flow shop scheduling problems with alternative resources, sequence-dependent setup times, and blocking. OR Spectrum (2019) 41:871–893. https://doi.org/10.1007/s00291-019-00567-8
- Chandrasekhar V. Ganduri. Rule Driven Job-Shop Scheduling Derived from Neural Networks Through Extraction. 2004 Thesis presented at College of Engineering and Technology of Ohio University
- Yeou-Ren Shiue, Ken-Chuan Lee, Chao-Ton Su. Real-time scheduling for a smart factory using a reinforcement learning approach. https://doi.org/10.1016/j.cie.2018.03.039
- Chaslot, Guillaume, de Jong, Steven, Takeshi-Saito, Jahn, Uiterwijk, Jos. Monte-Carlo Tree Search in Production Management Problems.
- Tara Elizabeth, Thomas Jinkyu Koo, Somali Chaterji, Saurabh Bagchi. MINERVA: A Reinforcement Learning-based Technique for Optimal Scheduling and Bottleneck Detection in Distributed Factory Operations. School of Electrical and Computer Engineering - Purdue University
- Sutton, Richard S., Barto, Andrew G. Reinforcement Learning - An Introduction. 2nd Edition. The MIT Press.
- Pumperla, Max, Ferguson, Kevin. Deep Learning and the Game of Go. Manning.

# Thank you!

https://github.com/schrappe/mctsscheduler