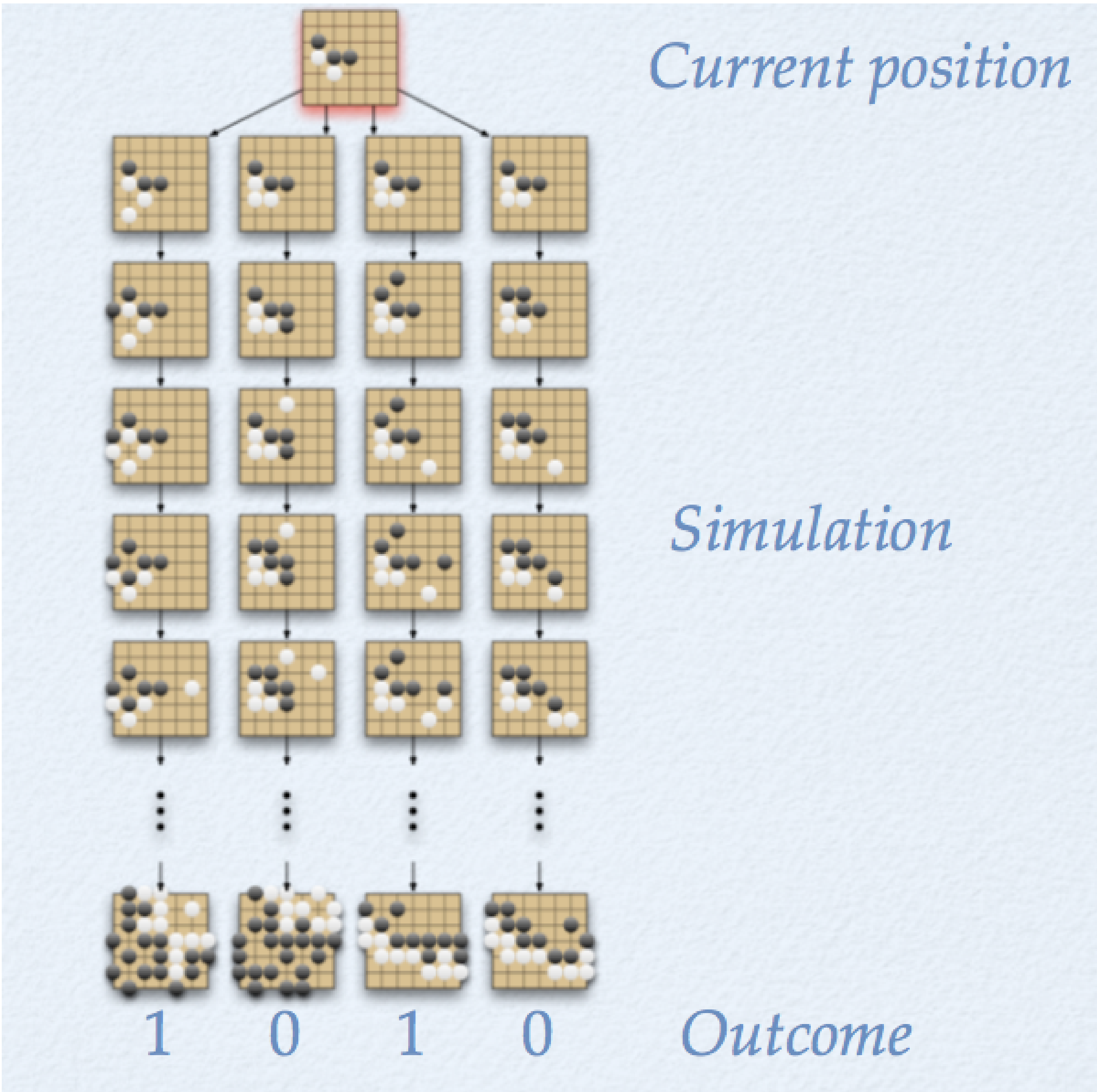
Computer Go: Enhancing Monte Carlo Tree Search with RAVE

Walt Javins, Peter Drake, Yung-Pin Chen
LEWIS & CLARK COLLEGE, PORTLAND, OREGON

Abstract

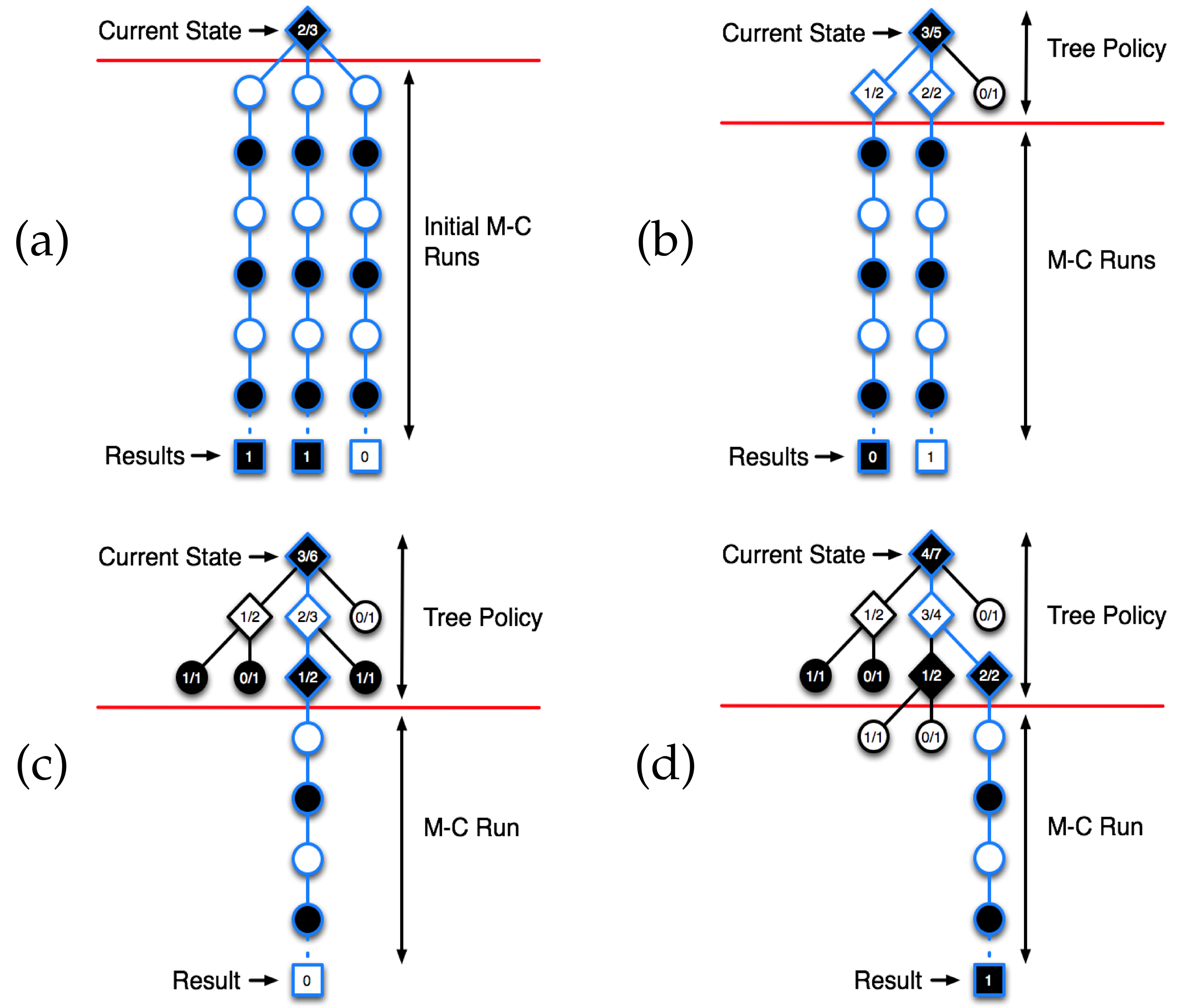
The Monte-Carlo approach has significantly strengthened the performance of computer Go programs. We examine and improve the RAVE (**R**apid **A**ction **V**alue **E**stimation) algorithm proposed by Gelly and Silver in 2008 and enhance the Monte Carlo tree search with two revised RAVE algorithms.

Monte-Carlo Simulation



Orego uses the Monte Carlo approach by playing random moves to the end of a Go game thousands of times, and then choosing the initial move leading to the best result.

Monte Carlo Tree Search



(a) MC simulations are run from the current state.

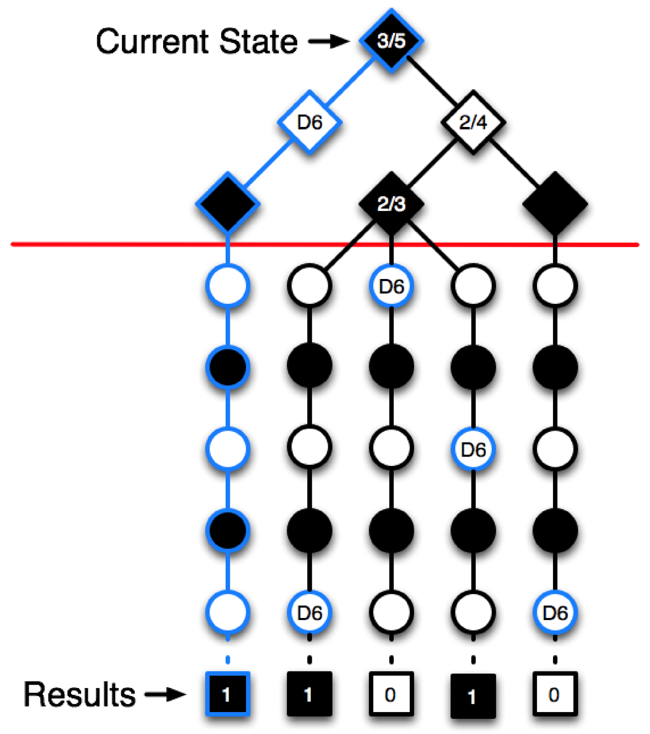
(b) Successful moves are incorporated into a game tree and simulations are run again.

(c) At each level of the tree, the branch with the highest MC win rate is selected. Once the bottom of the tree is reached, a MC simulation is run.

(d) Thus unpromising moves are ignored and promising ones are explored further.

RAVE

Monte-Carlo Tree Search needs a large number of simulations to estimate the value of a move, and each move value is learned independently. RAVE (**R**apid **A**ction **V**alue **E**stimation) generalizes data from related positions in parallel playouts and forms a rapid estimate of each move's value. In the figure on the right, even though only one MC simulation has been run through the point D6, RAVE includes all other occurrences of D6 in the play outs for a total of 5 data points.



Experimental Results

Orego model 1 has win rates ranging from 64% to 67%, very comparable to the performance result of Gelly and Silver's model.

Future Work

- Run experiments on *Orego* model 2
- Initialize the RAVE values at nodes with a heuristic function

Theory: Enhancing MCTS with RAVE

Let $Q_M(v)$ and $Q_R(v)$ be the Monte Carlo value and the RAVE value at node v, respectively. Let

b_M = MC bias; b_R = RAVE bias; σ_M^2 = MC variance; σ_R^2 = RAVE variance

Consider $Q(v) = \beta Q_R(v) + (1-\beta)Q_M(v)$ where β is chosen to minimize the mean square error

$MSE[Q(v)] = \beta^2\sigma_R^2 + (1-\beta)^2\sigma_M^2 + 2\beta(1-\beta)Cov_{R,M} + (\beta b_R + (1-\beta)b_M)^2$.

Gelly and Silver's model: $b_M = 0$ and $Q_M(v)$ and $Q_R(v)$ are independent: $\beta = \sigma_M^2/(\sigma_M^2 + \sigma_R^2 + b_R^2)$

Orego model 1: Assume $b_M = b_R$ and $Q_M(v)$ and $Q_R(v)$ are uncorrelated: $\beta = \sigma_M^2/(\sigma_M^2 + \sigma_R^2)$

Orego model 2: Assume $b_M = b_R$ and $Q_M(v)$ and $Q_R(v)$ are dependent:

$$\beta = (\sigma_M^2 - Cov_{R,M})/(\sigma_M^2 + \sigma_R^2 - 2\,Cov_{R,M})$$

References

- Kocsis and Szepesvari, 2006. *Bandit based Monte Carlo planning*.
- Gelly and Silver, 2007. *Combining online and offline learning in UCT*.

Acknowledgments

- Thanks to NSF for funding our research under the Willamette Valley REU-RET Consortium.
- Thanks to the Lewis & Clark College Math Department.