Extracting Policies from Replays to Improve MCTS in Real Time Strategy Games

Zuozhi Yang, Santiago Ontanón

Drexel University, Philadelphia, Pennsylvania

RTSy > gry planszowe

- Większy współczynnik rozgałęzienia drzewa (ang. *branching factor*)
  - Szachy: ~20
  - Go: <361
  - Starcraft: 30^{50} – 30^{200}

- Rozgrywka w czasie rzeczywistym – ograniczony czas na obliczenia

µRTS

- Minerały 

- Budynki:
  - Baza 

  - Baraki 
- Jednostki
  - Lekka 
  - Ciężka
  - Zasięgowa 

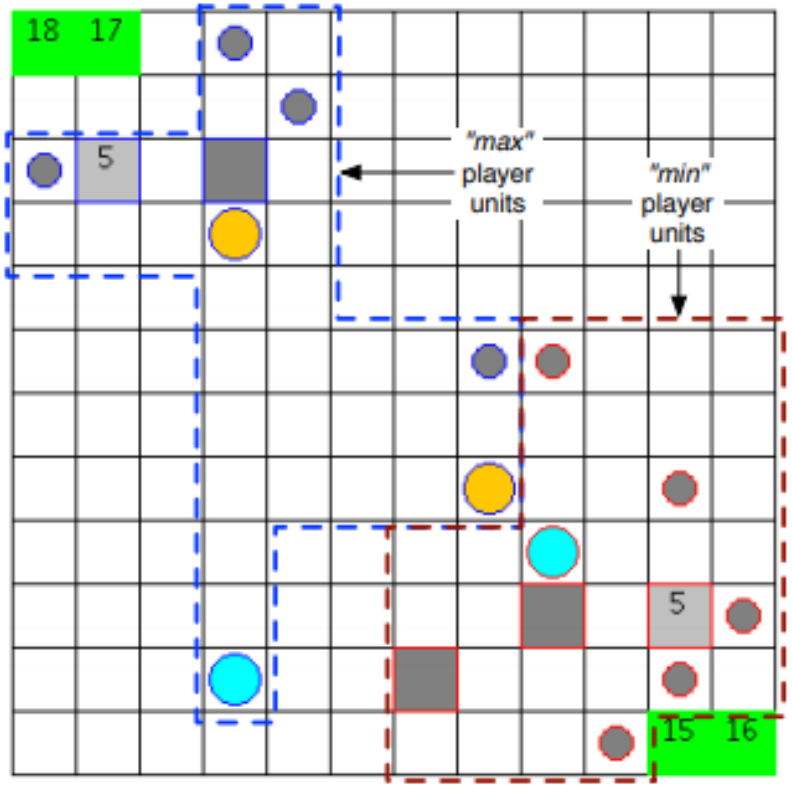


Figure 1: A Screenshot of μRTS.

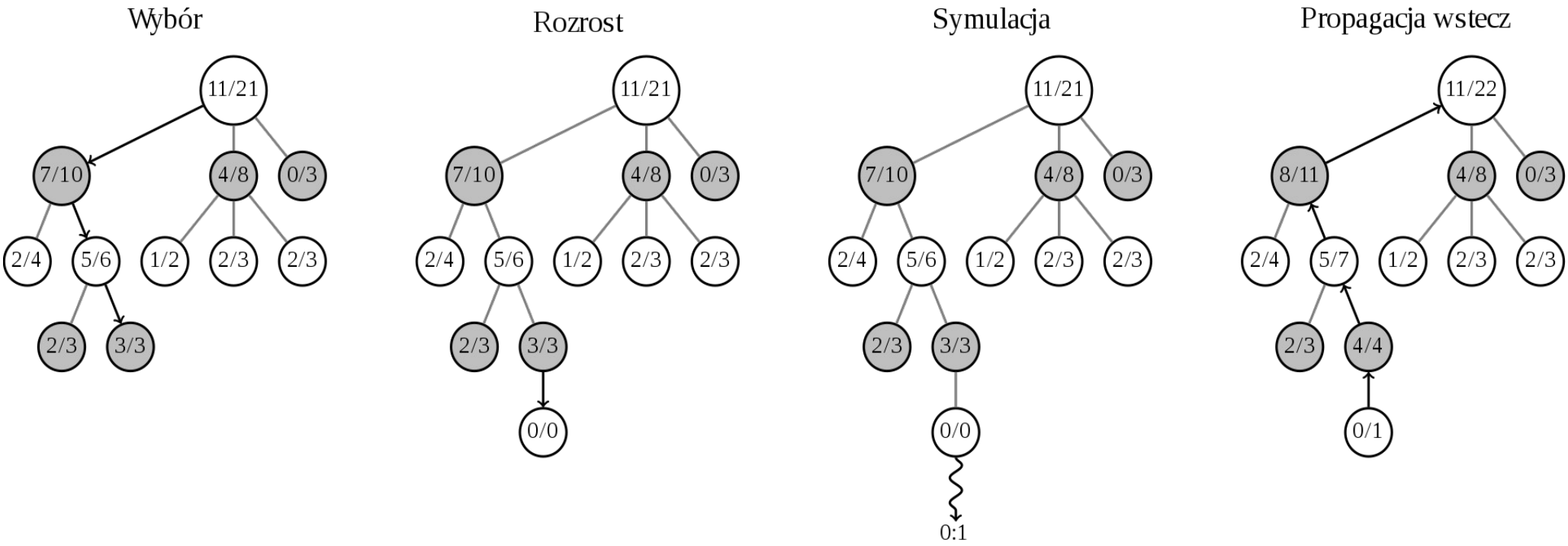
Monte Carlo Tree Search

- Oceniamy sytuację wykonując symulowane rozgrywki.
- Budujemy drzewo gry (na początku składające się z jednego węzła – stanu przed ruchem komputera)
- Dla każdego rozwiniętego węzła utrzymujemy statystyki, mówiące o tym, kto częściej wygrywał gry rozpoczynające się w tym węźle
- Rozwijamy wybrany węzeł dodając jego dzieci i przeprowadzając rozgrywkę.

Monte Carlo Tree Search

- Wybór (ang. *selection*): zaczynając od korzenia drzewa **R**, wybieraj kolejne węzły potomne, aż dotrzesz do liścia drzewa **L**. (*tree policy*)
- Rozrost (ang. *expansion*): o ile L nie kończy gry, utwórz w nim jeden lub więcej węzłów potomnych i wybierz z nich jeden węzeł **C**
- Symulacja (ang. *playout*): rozegraj losową symulację z węzła C (*playout policy*)
- Propagacja wstecz (ang. *backpropagation*): na podstawie wyniku rozegranej symulacji uaktualnij informacje w węzłach na ścieżce prowadzącej od **C** do **R**.

Monte Carlo Tree Search



Tree Policy Learning

- *Supervised learning* na podstawie działań botów
- Wynik – model $p_u(a_i \mid s)$
  - Wejście: s – wektor stanu gry „z punktu widzenia” jednostki u
  - Wyjście: $(a_1, a_2, \ldots, a_n)$ – prawdopodobieństwa wszystkich n akcji, jakie jednostka może wykonać
- Model jest wykorzystywany z każdym użyciem *tree policy*
- 8 cech do reprezentowania stanu gry

Naiwny klasyfikator bayesowski (ang. *Naive Bayes classifier*)

- Zakładamy, że wszystkie cechy są niezależne.

$$p(\boldsymbol{x}|y=c,\boldsymbol{\theta}) = \prod_{i=1}^{D} p(x_i|y=c,\boldsymbol{\theta}_{ic})$$

- Założenie o niezależności cech jest zwykle nieprawdziwe.
- Semi-Naive Bayes – prawdopodobieństwo dla danej cechy, zależy od n innych cech

Drzewa decyzyjne

- **Budowanie drzewa**
  - Podział węzła według cechy dającej największy zysk informacji (entropia)
  - Rozkłady prawdopodobieństwa w liściach
  - Pruning (a threshold confidence interval – 25%)
  - Liście zdominowane przez jedną klasę – można zamienić prawdopodobieństwa na 0/1
  - Laplace smoothing

Drzewa decyzyjne - ulepszenia

- **Bootstrapped aggregating (bagging)**

- **Random forest**
  - log(n) + 1 cech
  - 100 drzew