Supplementary Material for Batch Monte Carlo Tree Search

Algorithm 1 gives the main PUCT search algorithm using a transposition table of evaluated states and the two trees. The GetBatch boolean is used to make the distinction between the first option to develop the tree and the second option to build the batch. The first tree is the main search tree while the second tree is only used to build the batch.

The modifications of the search algorithm that implement the Second Move heuristic are given in algorithm 3. Lines 33-39 modify the best move to try at the root when the most simulated move is unreachable. In this case the second most simulated move is preferred.

Algorithm 2 gives the modifications of the GetMove algorithm for using the Second Move heuristic. Lines 8-17 choose between the most simulated move and the second most simulated move according to their means.

Algorithm 1 The BatchPUCT algorithm

```
Function BatchPUCT ($s, GetBatch$)
  if isTerminal ($s$) then
    return Evaluation ($s$)
  end if
  if $GetBatch$ then
    $t \leftarrow treeBatch$
  else
    $t \leftarrow tree$
  end if
  if $s \notin t$ then
    if $s \notin$ transposition table then
      if $GetBatch$ then
        add $s$ to the batch
      end if
      return Unknown
    else
      add $s$ to $t$
      return value ($s$)
    end if
  end if
  $bestScore \leftarrow -\infty$
  for $m \in$ legal moves of $s$ do
    $\mu \leftarrow FPU$
    if $t.p(s, m) > 0$ then
      $\mu \leftarrow \frac{t.sum(s,m)}{t.p(s,m)}$
    end if
    $bandit \leftarrow \mu + c \times t.prior(s, m) \times \frac{\sqrt{t.p(s)}}{1+t.p(s,m)}$
    if $bandit > bestScore$ then
      $bestScore \leftarrow bandit$
      $bestMove \leftarrow m$
    end if
  end for
  $s_1 \leftarrow$ play ($s, bestMove$)
  $res \leftarrow$ BatchPUCT ($s_1$, $GetBatch$)
  if $GetBatch$ then
    UpdateStatisticsGet ($res, bestMove, s, t$)
  else
    UpdateStatistics ($res, bestMove, s, t$)
  end if
  return $res$
```

Algorithm 2 The GetMoveSecondHeuristic algorithm

```
Function GetMoveSecondHeuristic ($s, B$)
    $b \leftarrow size(batch)$
    for $i \leftarrow 0$ to $B$ do
        GetBatchSecond ($s, B \times b, i \times b$)
        $out \leftarrow$ Forward ($batch$)
        PutBatchSecond ($out, B \times b, i \times b$)
    end for
    $t \leftarrow tree$
    $best \leftarrow bestMove_m(t.p(s,m))$
    $\mu \leftarrow \frac{t.sum(s,best)}{t.p(s,best)}$
    $secondBest \leftarrow secondBestMove_m(t.p(s,m))$
    $\mu_1 \leftarrow \frac{t.sum(s,secondBest)}{t.p(s,secondBest)}$
    if $\mu_1 > \mu$ then
        return $secondBest$
    else
        return $best$
    end if
```

Algorithm 3 The BatchSecond algorithm

```
Function BatchSecond ($s, GetBatch, budget, i, root$)
    if isTerminal ($s$) then
        return Evaluation ($s$)
    end if
    if $GetBatch$ then
        $t \leftarrow treeBatch$
    else
        $t \leftarrow tree$
    end if
    if $s \notin t$ then
        if $s \notin$ transposition table then
            if $GetBatch$ then
                add $s$ to the batch
            end if
            return Unknown
        else
            add $s$ to $t$
            return value ($s$)
        end if
    end if
    $bestScore \leftarrow -\infty$
    for $m \in$ legal moves of $s$ do
        $\mu \leftarrow FPU$
        if $t.p(s,m) > 0$ then
            $\mu \leftarrow \frac{t.sum(s,m)}{t.p(s,m)}$
        end if
        $bandit \leftarrow \mu + c \times t.prior(s,m) \times \frac{\sqrt{1+t.p(s)}}{1+t.p(s,m)}$
        if $bandit > bestScore$ then
            $bestScore \leftarrow bandit$
            $bestMove \leftarrow m$
        end if
    end for
    if root then
        $b \leftarrow highestValue_m(t.p(s,m))$
        $b_1 \leftarrow secondHighestValue_m(t.p(s,m))$
        if $b \geq b_1 + budget - i$ then
            $bestMove \leftarrow secondBestMove_m(t.p(s,m))$
        end if
    end if
    $s_1 \leftarrow$ play ($s, bestMove$)
    $res \leftarrow$ BatchSecond ($s_1, GetBatch, budget, i, False$)
    if $GetBatch$ then
        UpdateStatisticsGet ($res, bestMove, s, t$)
    else
        UpdateStatistics ($res, bestMove, s, t$)
    end if
    return $res$
```