

モンテカルロ木探索における シミュレーションの GPU を用いた並列化

明治大学理工学部情報科学科
知的情報処理システム研究室
4 年 15 組 28 番 高野昂平

目次

1	はじめに	2
1.1	研究概要	2
1.2	研究背景	2
2	関連研究	2
3	モンテカルロ木探索について	2
3.1	UCT について	3
3.2	並列化手法	3
4	実験	4
4.1	パラメータについて	4
4.2	実行環境	4
5	終わりに	4

1 はじめに

1.1 研究概要

モンテカルロ木探索 (Monte Carlo Tree Search, MCTS) におけるシミュレーション部分を並列化し、高速化を目指す。オセロを対象に実験を行った。

1.2 研究背景

モンテカルロ木探索の処理速度向上を図るうえで、シミュレーション部分が全体の処理速度のボトルネックとなっているのではと考え、シミュレーション部分を並列化することが全体の高速化につながるのではないかと考えた。

高性能な CPU を用いることで高速化を図ることも考えられるが、近年 CPU の性能向上率が以前に比べて伸び悩んでおり、高速化を図るためには CPU に依存した手法では高止まりになってしまう。そこで、別のアプローチで高速化を目指す必要があるが、その手法として GPU を用いた並列化を選択した。近年、GPU を画像処理だけでなく、汎用計算に用いて GPGPU (General Purpose GPU) として活用する動きも見られるため、モンテカルロ木探索に対しても GPGPU として活用できないものかと考えた。

2 関連研究

3 モンテカルロ木探索について

本研究のテーマであるモンテカルロ木探索 (Monte Carlo Tree Search, MCTS) とは、乱数を用いた計算手法であるモンテカルロ法を木探索に応用した手法である。

まず、評価関数を用いないという特徴を持つため、様々なゲームのプレイヤーに対してモンテカルロ木探索を実装することができる。例えば、General Game Playing (GGP) という汎用ゲームに対して人間の介入なしで強いプレイヤーを作るという試みがあるが、そのプレイヤーとしてモンテカルロ木探索を用いることが可能である。

また、不要な探索を行わないという特徴を持つ。合法手が膨大なゲームに対して木探索を行う場合、そのすべてのノードを探索することによって強いプレイヤーの作成するのは計算量が多くなってしまい、現実的でない。例えば、本研究ではオセロのプレイヤーに対して、モンテカルロ木探索を施したが、そのオセロの合法手は 10^{28} あると推測されており、すべてのノードを探索することが困難であることがわかる。しかし、モンテカルロ木探索の場合、すべてのノードを探索するのではなく、有効と考えられる手を優先的に深く探索するため、無駄な探索をすることがなく、計算量を削減することができる。

モンテカルロ木探索の流れとしては、選択、シミュレーション、展開、逆伝播の 4 ステップからなっており (図 1)、本研究では 2 ステップ目のシミュレーションを並列化することを目標としている。

3.1 UCT について

選択ステップにおいて、次に訪問するノードを決めるアルゴリズムとして、UCT アルゴリズムを採用した。このアルゴリズムは、以下の式が最大となるようなノードを次の訪問ノードとするアルゴリズムである。

以下の式において、 w はそのノードにおける評価値、 n はそのノードにおける訪問回数、 N はそのノードを親としたときの子ノードの n の値の合計値、 C は任意の値で探索がうまくいくように調整すべきパラメータとなっている。

$$UCT = \frac{w}{n} + C \sqrt{\frac{\ln N}{n}} \quad (1)$$

この式が意味するのは、ノードの評価値だけで訪問先を決めるのではないということで、第 1 項自体は評価値の平均を取っているが、第 2 項において訪問回数が少ないノードを訪問するように UCT 値を調整しているのがわかる。変数 C はこの訪問回数の少ないノードへの訪問をどれくらい重く見るべきか定めるパラメータと言える。

3.2 並列化手法

本研究では、シミュレーション部分を並列化するが、具体的には以下の図 1 のように、4 ステップあるモンテカルロ木探索の 2 ステップ目を複数回のシミュレーションを同時に実行することで実現する。

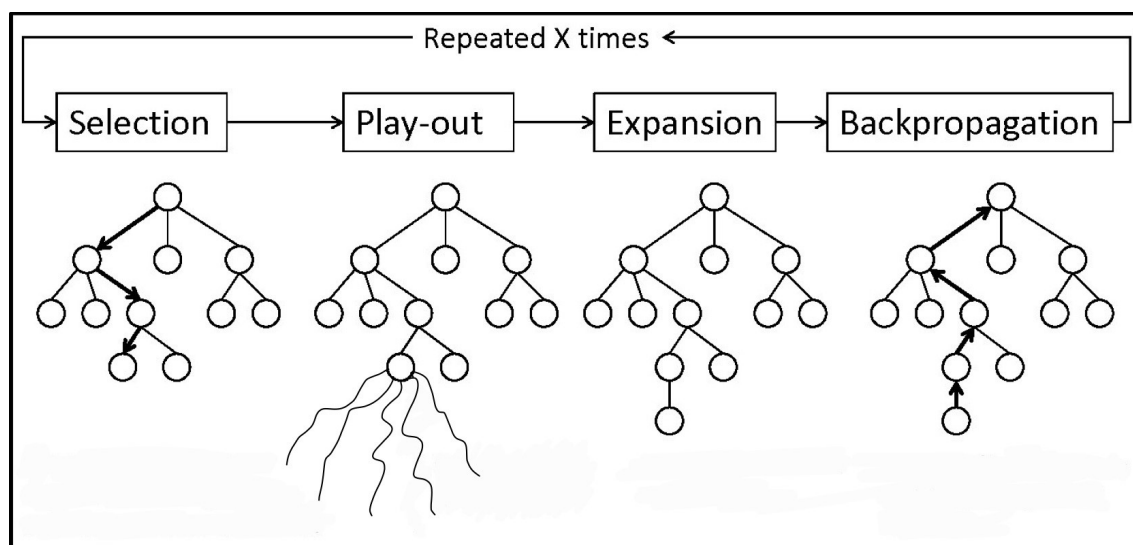


図 1 シミュレーションの並列化

並列化にあたって、NVIDIA 社の CUDA(Compute Unified Device Architecture) を用いた。CUDA は NVIDIA 社の GPU 上で動作する汎用並列コンピューティングプラットフォームである。なお、言語は C++ とし、CUDA C を用いて実装を行った。

4 実験

4.1 パラメータについて

4.2 実行環境

CUDA を用いる関係上 GPU は NVIDIA 社のものとなっている。

OS	Ubuntu 20.04.2 LTS
CPU	Intel(R) Core(TM) i9-10900X CPU @ 3.70GHz
GPU	GeForce RTX 3080
メモリ	32GB

5 終わりに