まったりゆうちゃんのアピール文書 2025

いまのところ去年と同じシステムである。改良することができなかった。可能な限り出場続けたいとと思っている。以下は前回までと同様の説明である。

複数のシステムを動作させて、その間でデータをやり取りして並列動作をする方法を検討している。以下は前回とほぼ同様の説明である。

1990 年過ぎから開発を始めた。4 半世紀にわたって開発しているシステムである。当初のコードもたくさん含んでいる。完全独自開発であり、他のシステムを参考にしていない(考え方は参考にしている)。アイデア的にも独自工夫をしている。

今日、AI というとディープラーニングをはじめとしてパラメータ学習に基づくものが多い。コンピュータ将棋での駒価値学習もそうである。しかしそうでない進化論的計算などの方式を試そうとしている。またディープラーニングと多量パラメータ学習の中間的なメカニズムを考えたいと思っている。

実現できるかどうかわからないが、全面的に書き換えることを考えている。今日からみれば、適切でないコーディングもある。それを直すことで、新しい並列方式が実現できると考えていて、少しとりかかっている。

開発者の年齢がもっとも高いのではないか。可能な限りやめないで続けたいと思っている。コーディング能力がいつまで続くか試したい。

WCSC35選手権KURO651アピール文書（京都府 河原林）

１．開発の経緯

幼少から将棋を指しておりましたが、10代後半から熱中し

20才頃からアマチュア将棋大会にて上位入賞を行う事が出来ました。

同時に認知科学や人口知能にも興味を持ち、

コンピュータ将棋への開発を意識する様になりました。（1980年代）

以後、仕事（ソフト会社SE職）を続けながら構想を温めてはおりました。

2001年11回大会に開発したアプリにて出場を行いましたが、

開発途中で3手読みも完全に出来ない稚拙な状態でした。

昨年66才にて仕事を退職したのを契機に、コンピュータ将棋の開発を再開し

（ソースはゼロから開始）未だ開発途中ながら今回参加する予定です。

今後３年間（70才迄）は開発を続ける予定ですが、PG能力の衰えとの闘いです。

２．特徴

フルスクラッチにて、データ構造／検索方法／評価方法を設計し、

全く１から他のコードを参考にせずに作成しています。

特徴１：局面モード（序盤・中盤・終盤）により各モード別の選択検索、評価を行う。

特徴２：序盤は自陣目標の設定を行い（攻め駒前進、玉囲い完成ETC）目標優先で検索を行う。

特徴３：玉安全度は中・終盤（詰めろ未満）にて利用し、詰み危険度には使用しません。

特徴４：機械学習は全く利用していません。

３．仕様

定跡検索：DB化した定跡データから、完全一致検索及びLIKE検索を行う。（未実装）

序盤：自陣目標の設定を行い（攻め駒前進、玉囲い完成ETC）目標優先で検索を行う。（未実装）

中盤：位置別駒価値＋玉安全度を主評価とした、全幅検索＋選択検索（駒取り優先）での検索を行う。

終盤：位置別駒価値＋玉安全度を主評価とした、全幅検索＋選択検索（駒取り優先）での検索を行う。

詰めろ判定以上：詰め検索を主評価とした、詰み検索を行う。

４．目標

２０２５年5月 ６６歳 局面モード決定＋全幅検索＋一部選択検索＋詰み検索のみコード化

２０２６年5月 ６７歳 上記３．仕様の全機能追加

２０２７年5月 ６８歳 定跡自動収集＋自己学習

２０２８年5月 ６９歳 並列検索の上位機能（ＧＰＵ、ＣＬＵＤ？）

以上.

一閃将棋　WCSC35 アピール文章

チーム: 自堕落同好会

コンセプト

将棋専用アクセラレータの実装により、圧倒的な読みの速度を実現する

特徴

演算装置: FPGA (Amazon EC2 F2 インスタンスを使用予定)

> よく使われる CPU/GPU ではなく、FPGA 上に将棋専用の論理回路を構築する。

探索手法: αβ 法

評価関数: NNUE

並列化手法: LazySMP(予定)

技術要素

指し手生成

性能にきわめて大きな影響を与えると考えられる。~~3/31 時点では、1 秒間に約 200 万局面~~ 4/15 時点では、1 秒間に約 400 万局面に対して全指し手の生成が可能である。末端局面の 1 手前で指し手生成を行い、生成された指し手を評価関数部に流し込むとき、仮に評価関数部が即座に評価値を返し、かつ十分な並列度を持つならば、NPS はおよそ 400 万×生成された指し手数となる。

4/15 時点で未実装の機能

- 打ち歩詰め判定（指し手生成部ではなく専用ユニットで判定予定）
- 連続王手の千日手判定（指し手生成部ではなく探索部で判定予定）
- 静止探索用指し手生成
- 詰・不詰探索用指し手生成

探索

αβ 法を利用する。

陽にスタックメモリを用意して push/pop を行うことで再帰的探索が実現できる。4/15 現在未実装だが、オセロでは過去に実装実績がある[1]。

FPGA においては処理を物理的に並列に走らせることができる。そのため、いくつかの枝刈りは他の処理の裏で実行することで時間的コストを隠蔽することが可能であり、CPU 実装では有用でなかった枝刈り手法も有用な可能性がある。

置換表

4/15 現在実装中。

F2 インスタンスに搭載されている VU47P FPGA には 16GB の High Bandwidth Memory(HBM)が搭載されているが、探索速度を考えると帯域は過大だが容量が小さいため、通常の DDR4 メモリの方を使う予定。

評価関数

NNUE 評価関数では、評価関数テーブルはネットワーク構造により数十 MB〜数百 MB ものサイズになる。評価のたびに外部メモリへアクセスしていてはスループットが大きく低下してしまうので、高速なオンチップメモリを利用する。

UltraRAM は 72bit(9bytes)幅、4096word の疑似[2] 2-port RAM で、VU47P FPGA には 960 個搭載されている。合計容量は 9B * 4096 * 960 = 35389440B = 33.75MiB となる。この容量は、典型的な NNUE 評価関数のパラメータを全て載せるには不足しているため、以下の工夫を行っている。

- ネットワーク構造に halfKP_vm 256x2-32-32 を採用

FeatureTransformer 部の出力次元数はパラメータサイズに直結するため、比較的小さめの 256 次元にすることにした。_vm は玉が 6,7,8,9 筋にいる場合にそれぞれ 4,3,2,1 筋にいるとして扱う(vertical mirror, vm)という意味である。この手法により、パラメータサイズを通常の HalfKP に比べて 5/9 倍に減らすことができる。

- 12bit 化
  FeatureTransformer のパラメータは通常 16bit 整数型が用いられているが、推論時に 38 個の和をとることから、実際には 11〜12bit 程度に収まることがほとんどである。FPGA ではロジック回路のデータ幅を 1bit 単位で自由に決めることができるため、12bit 幅にすることで、

1 https://github.com/primenumber/FPGAOthello

2 2 つのポートはクロックを共有するため、厳密には 2-port RAM とは異なる

パラメータサイズを 3/4 倍に減らすことができる。また、12bit 幅は UltraRAM のデータ幅 72bit のちょうど 1/6 となっているため、回路を単純化できる。

並列化

並列化ができるかどうかは主に指し手生成・探索部のロジックリソースの余裕に依存するが、可能なら LazySMP による並列化を予定している。

EC2 F2 では最大で 8FPGA を搭載したインスタンスが提供されているが、複数 FPGA を利用する場合には、合議か ponder に用いる予定。

定跡

制御 CPU 側で対応予定。

アピール文書 2025ss

Yorkie 2025SS アピール文書

- 評価関数として NNUE を使います。
- 探索時に置換表を使わず、替わりにMCTS探索に類似した方法で探索経過を保持します。
- Quota で許される限りの CPU数、メインメモリー容量のPCを使用予定です。
- その他、以下による高速化を行います。
  - 実行環境をLinuxに限る
  - 極力、定数を定義する
  - NUMA を意識したメモリ割り当てを行う
  - 探索を実行するP上Cで gcc の `-march=native` に相当する最適化オプションを使ってビルドする
  - USIプロトコルを実装せず、CSA サーバプロトコルのみ実装する