

CGP アピール文章

2016/3/28
大熊 三晴

主な特徴

- 1から作成
- 非ビットボード型
- 高 NPS (1 スレッド 1.5~2Mnps, 4 スレッド 4~5Mnps @ core i7 6700K)
- 局面構造体に各マスへの利きの状態を保持
- 局面構造体に評価関数の演算途中結果のうち変化の頻度が少ないものを中心に保持
- 探索(枝刈り)は Stockfish をベースに将棋向けの独自手法を追加したもの
- 評価関数の形は Bonanza ベースのオリジナル
- 評価関数は現在様々な設定で学習を試行中
- SIMD 命令をはじめとした拡張命令を活用
- 一般的にはあまり使われていない機能を使用

• 1から作成, 高 NPS

強さをあまり考えずに高 NPS を目指して自作したプログラムをベースとしております。
現時点でも高 NPS が武器ですが、マルチスレッド化方式は YBWC ですが実装のまずさがあり、マルチスレッドではあまり NPS が出ておりません。マルチスレッド効率を改善するアイデアはあり、またさらに高速化するアイデアも複数ありますが、他の改良を優先するためにまだ未着手です。

• 非ビットボード型, 利き等を保持

非ビットボードだとビットボードに比べ遅くなる処理もありますが、複雑な情報を持てることにより速く処理できる可能性もあります。ビットボードに比べ遅い処理をうまく避けるために利きを保持したり、局面構造体の配置をビット位置を含めて工夫しております。

また利きの保持以外にも、演算途中のデータを保持することによりメモリアクセス待ち時に演算を回す事により高速化を狙っております。

• 探索(枝刈り)は Stockfish ベース, 評価関数は Bonanza ベース

いわゆる Bonafish に属するソフトです。

•SIMD 等の活用

高速化のため SIMD を活用しております。SIMD は現在評価値の算出、オーダリング、構造体のコピーが主な使用箇所です。

オーダリングの一部は VPMAX 命令で次に試す手を抜き出す方式を取っております。この方式は条件分岐なく複数の手を比較できるため、挿入ソートを通常の x86 命令で行うより高速化できております。SIMD を用いたソートの使用は試していないのでこちらのほうがより高速になる可能性もあります。

コピーに関して言えば通常の x86 命令だとメモリアクセスの能力を使いきれないため、SIMD 命令を使用しております。

また置換表や評価値テーブルのページテーブルにラージページ(Windows での言い方 Linux 用語だと Huge Page)を使用し、高速化を図っております。