

2010年3月24日
天野 史斎

Dias R7 工夫した点アピール文書

本文書は筆者が WCSC20 に出場させるコンピュータ将棋プログラム「Dias R7」の工夫した点をアピールするものです。ただし、今回はオリジナリティーを主張できれば良いものと考え、必ずしもプログラム内容紹介としての完全さは追求していません。

1. 出場プログラムのアピールポイント

1.1. 評価関数

駒の価値ベースの評価関数でなく、自己が負けないという制約の下で、ゲームの進行度をダイレクトに表すような新規の評価関数を用います。

物理量に近い意味をもち連続性を有するので、前向き枝刈り等のミスを敗着の兆候としてすみやかに察知可能です。

反面、複雑な計算のため差分計算が困難ですが、それは計算システムを工夫して回収を計ります(→1.4.項)。

1.2. 探索部

上記評価関数は勝率の追求というプレイヤー共通の価値観を完全には表現していないので、探索の最大の深さを偶数固定とし、探索の末端ノードが(最大の深さに達する前にゲーム木の葉に行き当たらない限り)必ず自己の手番になるようにします。また前向き枝刈りを自己の手番でのみ行う手番非対称なロジックとしています。

手のオーダリングは行いません。これは 1.1.項で述べた複雑な評価関数(差分計算困難)の高速計算を、巨大置換表とマルチコアによる並列計算を組み合わせた計算システム(→1.4.項)で成立させるためです(詳細略)。手のオーダリングを行わないということは反復深化手法を放棄することになりますが、このデメリットも同じシステムで(主に巨大置換表の効果により)回収できると見込んでいます。

また、不用意な探索延長でかえって最善手を誤る危険を冒すよりも評価関数の素性の良さに賭ける方が有利と考え、今回は探索延長も行いません。

1.3. 置換表

探索、df-pn 探索それぞれの置換表について、下記に対応させました。

- 置換が不用意な探索延長と同等になってしまう危険の除去

結果的に GHI 対策にもなっており、また置換表データの手番を超えた安全な持ち越しも可能になっています。

1.4. マルチコアによる高速計算

前記評価関数は計算量が大きいため、それを回収する目的で複数コア（含 HT）を動員して並列計算します。特徴を列挙すると次の通りです。

1. 巨大な単一置換表をスレッドで共有する。
2. キャッシュ汚染を引き起こさない形で置換表にアクセスする。
3. ロックを一切しない。
4. 評価関数計算中のスレッドを他のスレッドに「追い抜かせる」。
（ゲーム木上での全スレッドの計算箇所がほぼ集中した形で探索が推移します。）

並列計算といいましたが評価関数自体を分解せず、個々のスレッド内で計算を完結させます。評価関数計算と探索とをセットにしたジョブを複数コアで行い、上記巨大置換表の空きを塗りつぶして行くという、より大きい作業を並列に進めるというイメージです。CPU コア - 外部メモリ（DDR）間のトラフィック面は、両者間のデータ転送を置換表の読み書きに限定し、その際キャッシュを汚染させず、評価関数計算を極力キャッシュ上で安定して回すという設計になっています。

1.5. df-pn 探索

最良優先判断に 探索の置換表上のデータを使用します。

1.6. 予測読み

法、df-pn 探索とともに手番によらず常時回し続け、置換表をどしどし埋めさせます。もちろんその過程で詰みが証明できたら即座に着手に反映します。

1.7. 前向き枝刈りの自動調整

1.1.項の評価関数は指した結果から敗着の危険を早期に察知可能と見込まれるので、試合しながら、学習と引き替えに負けるというペナルティーを支払うことなく、悪手を指さなくなるようにじわじわ前向き枝刈りを調整させます。

以上