80　　ニップの並列化について

情報論理研究室　　井藤 雄太

1．はじめに

ニップとはリバーシの一種であり, 円形リバーシと呼ばれることもある．通常のリバーシは正方形のゲーム盤上に石を配置するため, 角に石を置けるかどうかでほぼ勝敗が決まってしまう．しかしニップでは盤面が円形になっているため, どの石も最後までひっくり返される可能性がある．そのため, リバーシよりも終盤での逆転が起こり易い．

リバーシや将棋といった完全情報ゲームの AI を作成する場合, 数手先の局面を先読みし, 先読み後の局面の評価値によりどの手を打つか決定することが多い．強い AI を作るにはこの先読み手数を多くする必要があるが, 先読みしなければならない局面の数は指数関数的に増えていくため, 処理に膨大な時間がかかってしまいます．このため, 先読み処理を並列化することで処理時間の短縮を図ることが多い．

これはニップも同様であり, 強い AI を作るためには先読み手数を増やさなければならないが, それには膨大な時間がかかる．そこで, 本研究ではニップの探索を並列化することによってプログラムの高速化を目指す．

2．研究内容

本研究で作成したニップ AI は, 打てる手が複数ある場合, その手を打った場合に得られる局面を先読みし, 先読みで得られた局面の評価値を用いて手を決定する．

局面の評価の代表的な手法としては，評価値マップの使用がある．評価値マップとは，各マスに正負の価値を付加し，マスに自石が置かれた場合その価値を足し相手石が置かれた場合その価値を引く，というものである．リバーシの場合は角のマスを高価値とし，角に隣接するマスを低価値とすると良いことが一般に知られている．一方, ニップは各マスにどのような価値を付加するかは確立されていない．そこで本研究では，各マスに図 1 に示す価値を付加し，その有用性を検証する．

リバーシでは一般に，相手の選択肢を減らす手が良いとされている．そこで本研究の評価関数は評価値マップによる局面評価に加えて 相手の選択肢の数を用いた．本研究で作成したプログラムの実行の様子を図 2 に示す．

3．結果・考察

探索アルゴリズムを並列化することによって探索の高速化が成されれば、現在解析の完了していないゲームの解析が進むことが予想される．

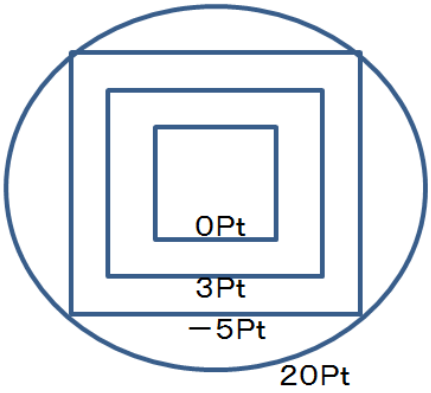

図 1　評価マップ

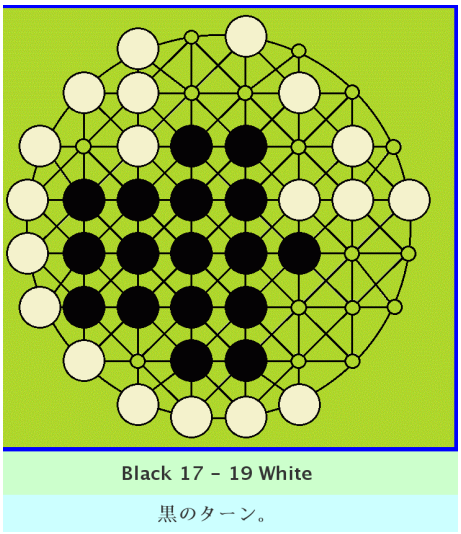

図 2　プログラムの実行画面

4．おわりに

現在プログラムの完成に向け取り組み中であり, プログラムを完成させ並列化の効果を確認する必要がある．

参考文献

1) 結城　浩：Java 言語で学ぶデザインパターン入門【マルチスレッド編】, ソフトバンク クリエイティブ (2006)
2) Seal Software：リバーシのアルゴリズム, 工学社 (2007)
3) OpenNip(オープンニップ) プロジェクト
http://sourceforge.jp/projects/opennip/