78

ニップの並列化

情報論理工学研究室　　松浦 美里

1． はじめに

ニップ 3) とは, リバーシの一種である. ニップは正方形の盤で行われる通常のリバーシとは違い, 円形型の盤上でプレイされる. 盤が円形のため, 確定石, すなわちリバーシの四隅においた石のように色がもう二度と変わらない石が無い. そのため, 確定石を増やすことが勝利につながるリバーシと違い, 終盤で逆転することが可能な遊戯であると言える.

リバーシや将棋, チェスといった完全情報遊戯の AI 作成において, 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打つ手を決定する手法がよく用いられる. より強い AI を作成するためには, この先読みする手数を増やす必要があるが, 手数を増やすと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, 十数手先程度の先読みでも処理に膨大な時間がかかってしまう. そ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 作成のためには先読みの手数を増やさなければならないが, これは上記の遊戯と同様に時間がかかる. 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による時間短縮を目指す.

2． 研究内容

ニップの並列化に先立ち, 本研究ではまず逐次のニップ AI を作成した.

本研究で作成したニップ AI は, 打てる手が複数存在する場合, その手を打った後に得られる局面を先読みし, そこで得られた局面の評価値を用いて打つ手を決定する. 局面の評価の代表的な手法として, 評価値マップがある. 評価値マップとは, 各マスに正負の値を価値として付加し, 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値を正負反転し足す, というものである. リバーシの場合は角である四隅のマスの値を高価値とし, 四隅に隣接するマスを低価値とすると良いことが知られている. 一方ニップにおいては, 各マスにどのような価値を付加すればより良い結果になるのかは確立されていないが, 外周に隣接するマスは低価値であるというのはリバーシと同様だと思われる. 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.

また, リバーシでは相手の選択肢を減らす手がより良いとされている. そこで本研究の評価関数は評価値マップによる局面の評価に加えて相手の選択肢の数も用いる.

		30	30	30	30		
	30	-10	-10	-10	-10	30	
30	-10	-2	3	3	-2	-10	30
30	-10	3	0	0	3	-10	30
30	-10	3	0	0	3	-10	30
30	-10	-2	3	3	-2	-10	30
	30	-10	-10	-10	-10	30	
		30	30	30	30		

図 1　各マスの評価値

表 1　対戦結果 (試行回数 100 回)

対戦	先手勝ち	後手勝ち	引き分け
AI 対ランダム	75	25	0
ランダム対 AI	38	62	0
AI 対 3)	67	33	0
3) 対 AI	32	68	0

3． 結果・考察

本研究で作成したニップ AI の性能を検証するため, ランダムで打つ AI および既存のニップ AI との対戦を先手後手それぞれ 100 回行った.

表 1 に対戦結果を示す. 表 1 よりランダムには先手でも後手でも七割ほどの確率で作成した AI が勝利している. 既存のニップ AI にも六割ほどの確率で作成した AI が勝利している.

4． おわりに

本研究ではニップ AI を作成した. 本研究で作成したニップ AI は六割以上の確率でランダムや既存の AI に勝利しているが, 三割ほどは負けている.

今後の課題は先読み数を増やすと動作が遅くなり一局を終えるのに時間がかかった. 今回は 4 手先までしか動かすことができなかった. 今後このプログラムの探索の部分を並列化して高速化を図りたい.

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/