

## 1. 序論

リバーシプログラムは一般に数手先を先読みし、その中で最も評価関数が高い手を採用する。しかし、先読み手数増加に伴い、可能な盤面の組み合わせは指数的に増大するため、先読み数を 10 手程度に制限しても探索には膨大な時間がかかる。

そこで、本研究では、先読み数にかかる時間を短縮するためにマルチスレッド処理を用いる Java プログラムを作成する。マルチスレッド処理は先読みにおける探索作業をサブスレッドで処理することにより並列計算が可能であり、一定時間内に行える先読み数を増やす事ができる。そこで探索作業のみをマルチスレッド化することで先読み数を増やし、並列計算にする有効性について考える。

## 2. 研究内容

本研究では、リバーシで計算機との対戦が行える Java プログラムを作成した。本研究に用いたプログラムはある局面で打つ手は局面を何手か先読みし、先読み後の局面で最も評価値が高くなる手を用いる。また、先読み数は序盤・中盤・終盤とすることで先読み数を分けている。これはゲーム進行に伴い採るべき戦略が変化することから、先読みの評価関数も変化するためである。本研究では上記の戦略に基づくシングルスレッド AI(以下 SSAI)とマルチスレッド AI(MSAI)を作成した。SSAI はある盤面で可能な各候補手に対しその手を指した場合の盤面を再帰的に探索し、評価値を求める操作を全ての候補手に対し逐次的に行う。MSAI は同様の操作をマルチスレッドを用いて並行処理する。また、本研究では盤面の評価基準として着手可能手数、開放度、ウィング、確定石の個数、危険な C 打ち(隅がとられていない状態での C 打ち)、危険な X 打ち(隅がとられていない状態での X 打ち)を用いる。SSAI, MSAI の先読み数  $L$  は事前に手を調べて探索順序を決定するための先読み  $L_p$ 、序盤・中盤の先読み数  $L_o$ 、終盤の必勝読みを始める残り手数  $L_m$ 、終盤完全読み  $L_e$ 、4 個組で定義される。表 1 に先読み数の初期値を示す。

表 1 先読み数の初期値

	$L_p$	$L_o$	$L_m$	$L_e$
SSAI	1	2	5	3
MSAI	1	10	15	13

SSAI,MSAI の性能評価を行うための対戦相手は、候補手からランダムに選択するランダム AI(以下 RAI) を用いる。SSAI,MSAI の先読み数を変化させながら RAI と対戦させ、先読みにかかる時間と勝率を調査する。

先読み数は表 1 の値で RAI と 100 回対戦し、実行時間を計測する。次に 4 つの先読み数の要素を 1 大きくして再度 RAI との 100 回の対戦  $L_p=20$  まで繰り返し行う。以下に実験を行った計算機のスペックを示す。

- プロセッサ: Intel(R)Core2 Duo CPU L7800 2.0GHZ
- メモリ(RAM): 2.00GB

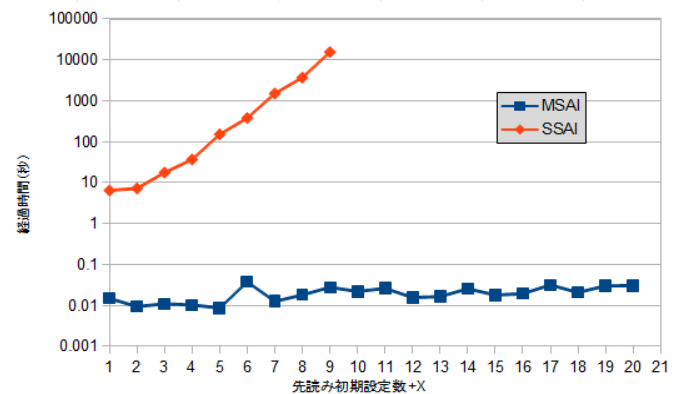


図 1: 先読み時間と先読み手数の関係

## 3. 結果・考察

図 1 に SSAI,MSAI それぞれの先読み時間と先読み手数の関係を示す。SSAI では  $L_p=10$  以上は実行時間が長すぎるため測定不能とした。一方、MSAI はマルチスレッド化することにより、探索時間が縮んでおり、先読み数をかなり大きくすることができた。また、このことからマルチスレッドを用いれば更に複雑な評価関数を用いて高度な先読みを行うことが可能であることが示される。また、勝率は先読み数に関係なく  $66\pm 5\%$  程度だった。

## 4. まとめ及び今後の課題

本研究では MSAI と SSAI の先読みにおける時間と勝率の計測を行った。RAI に負ける事ケースがあることから常に最良の選択が採れてるわけではないことが示された。よって今後の課題として、更に強い評価関数を導入することが挙げられる。評価関数は様々なものが提案されているが、最良な評価関数、またはその組み合わせ等は未解決である。よって評価関数の性能を検討し、最良となる評価関数を得ることが今後の課題である。

## 参考文献

- 1) Seal Software : リバーシのアルゴリズム,工学社(2003)
- 2) 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], SoftBankCreative (2006)
- 3) 大筆豊 : オセロプログラムの評価関数の改善について, 情報処理学会研究報告 2004-GI-11 p.15-p.20 (2004)