

モンテカルロ法に基づく Flood-It の AI に関する研究

周・伊藤研究室 学部 4 年 小田将也

1 はじめに

Flood-It とは, $n \times m$ の色分けされたグリッド上で行うゲームである. Flood-It はグリッドの一番左上にあるマスをもとに基準 (ピボット) として進めていくゲームであり, 1 回の操作でピボットをある色から別の色へ変更する. ピボットから上下左右に現在の色のみを辿ってたどり着くことができるマスの集合を, 自分の領地と呼び, ピボットの色の変更と同時に自分の領地の全てのマスの色が同じ色になる. 図 1 に操作の例を示す. 操作前図 (a) の灰色部分の 3 マスが自分の領地であり, ここにピボットの色を 1 から 2 へ変える操作を適用すると, 操作後図 (b) のグリッドが得られる. この操作を繰り返し $n \times m$ マス全てのマスを自分の領地にすることが目的である. このとき, 操作数が最小となるような操作列を求めることが Flood-It の問題である. Flood-It の問題に関しては, 既知の結果としてグリッドの大きさが $n \times n$ ならば NP 困難 [1] $3 \times n$ ならば NP 困難 [3], $2 \times n$ ならどんな色数でも多項式時間で解けることが知られている [2].

今回は, Flood-It を二人対戦用ゲームにしたものを考える. 二人用 Flood-It は, 同じく $n \times m$ の色分けされたグリッド上で行う. 一番左上のマスをプレイヤー 1 のピボット, 一番右下のマスをプレイヤー 2 のピボットとし, それぞれのピボットに対してプレイヤー 1 の領地, プレイヤー 2 の領地を定義する (図 2). プレイヤーが交互にピボットの色を変える操作を繰り返し, グリッドが 2 色になったらゲーム終了である. ゲーム終了時に自身の領地が広い方のプレイヤーの勝利, というルールである (図 3).

本研究では, 二人用の Flood-It に対して, できるだけゲームの勝率の高い AI を作成することを目標に, モンテカルロ法に基づく対戦用 AI を作成する.

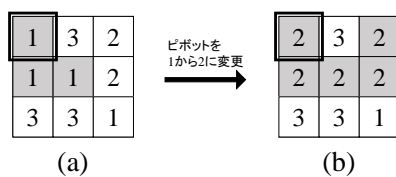


図 1: 1 回の操作の例. マス目の中の数字は色を表す.

1	3	5	1	5
1	4	2	4	4
1	3	5	1	5
3	2	3	2	1
4	3	5	5	5

図 2: 二人用ゲームの盤面の例. 実線内部がプレイヤー 1 の領地であり, 点線内部がプレイヤー 2 の領地である.

2	2	2	2	2
2	2	2	2	2
2	2	2	2	5
5	5	5	5	5
5	5	5	5	5

図 3: ゲーム終了時の盤面の例. この場合はプレイヤー 1 の領地の方がプレイヤー 2 の領地よりも広いのでプレイヤー 1 の勝利である.

2 定義・準備

2.1 作成すべき AI について

作成すべき二人用 Flood-It の AI について, 以下のよう

- 入力: グリッドの大きさ $n \times m$, グリッドのそれぞれのマスの色
- 出力: 次に変更すべき色

2.2 モンテカルロ法を用いた AI のアルゴリズム

モンテカルロ法とは, シミュレーションや数値計算を乱数を使用して行う手法の総称である. モンテカルロ法を用いた AI では, 受け取ったグリッドに対して, プレイヤーの次のそれぞれの色の選択肢からランダムな選択をし続けてゲームを終了させるプレイアウトという操作を繰り返す. プレイアウト後の勝敗数を記憶することで次のそれぞれの色の選択肢の勝率を求め, 勝率の一番高い色を出力する.

3 今後の方針

まずはモンテカルロ法を用いた AI とそのテスト用プログラムを作成し、テスト用プログラムで既存の AI との勝率を確認する。その後、多腕バンディットのアルゴリズムをモンテカルロ法に適用し勝率が上がるか確認する。

また、同じグリッドの状態に対する既存の AI の選択や人間の選択とモンテカルロの選択を比較し、モンテカルロの長所および短所を探る。その長所や短所をもとにさらなるアルゴリズムの改善をし、勝率の向上を試みる。

その他に、モンテカルロ法のアルゴリズムにおけるゲーム終了時に評価する値を勝敗だけでなく領地の広さも加えることで勝率が改善されるか確かめる。

参考文献

- [1] D. Arthur, R. Clifford, M. Jalsenius, A. Montanaro and B. Sash, The complexity of flood filling games, Proceedings of Fun with Algorithms 2010, LNCS 6099, pp. 307–318, 2010.
- [2] R. Clifford, M. Jalsenius, A. Montanaro, B. Sach, The complexity of flood filling games, Theory of Computing Systems 50(1), pp. 72–92, 2012.
- [3] K. Meeks, A. Scott, The complexity of flood filling games on graphs, Discrete Applied Mathematics 160(7–8), pp. 959–969, 2012.