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 filling ゲームとは，色の操作を繰り返すことで，全体を一つの色に変えるゲームの総称である．本研究では，Flood filling ゲームの中でも特に研究のされている，Flood-It について扱う．**Flood-It** では，$n \times m$ の色分けされたグリッドが与えられる．ここで，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を含めたグリッドの状態を**盤面**と呼ぶ．盤面の一番左上にあるマスを**基準 (ピボット)** と呼び，ピボットから上下左右に現在の色のみを辿って辿り着くことができるマスの集合を，**自分の領地**と呼ぶ．1 回の**操作**では，ピボットを現在の色から別の色へ変更する．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変わる．図 1 に操作の例を示す．操作前図 1(a) は操作前の盤面であり，灰色で示した部分が自分の領地である．この盤面にピボットの色を 1 から 2 へ変える操作を適用すると，図 1(b) の盤面が得られる．ここで，色の変更によって，自分の領地が増えたことに注意しよう．このように操作を繰り返すことで，グリッドの全てのマスを自分の領地にすれば終了である．この Flood-It に対し，色分けされたグリッドが与えられた際に，終了までに必要な最小の操作数を求める問題について以下のような研究がされてきた．Arthur らが示した結果によると，グリッドの大きさが $n \times n$ である場合には，使用される色の数が 3 色の場合ですら NP 困難である [1]．また，2012 年，Meeks と Scott は，$3 \times n$ のグリッドにおいて，使用される色の数が 4 色のときですら NP 困難であることを示した [3]．一方で，2012 年，Clifford らは，$2 \times n$ のグリッドであれば，使用される色の数が何色であったとしても多項式時間で解けるということを示した [2]．

従来研究されてきた Flood-It は一人用のゲームであったが，今回は，Flood-It を二人用対戦ゲームに拡張したものを考える．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．一番左上のマスを先手のピボット，一番右下のマスを後手のピボットとし，それぞれのピボットから上下左右に，同じ色のマスのみを辿って辿り着くことができるマスの集合を，それぞれ先手の領地，後手の領地と定義する (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，グリッドの全てのマスが，先手の領地もしくは後手の領地になった場合にゲーム終了となる (図 3)．ゲーム終了時に自分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．

本研究では，二人用 Flood-It の，強い 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．

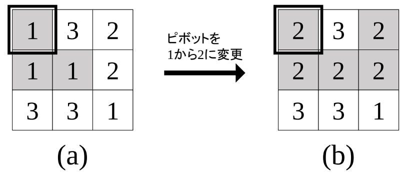

図 1: 1 回の操作の例．(a) は操作前の盤面，(b) は操作後の盤面である．マスの中の数字はそのマスに割り当てられている色を表し，灰色のマスは自分の領地を表している．

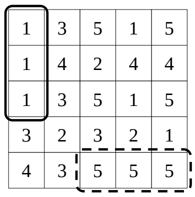

図 2: 二人用 Flood-It の盤面の例．実線内部が先手の領地であり，点線内部が後手の領地である．

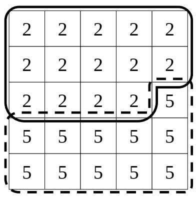

図 3: ゲーム終了時の盤面の例．この場合は先手の領地の方が後手の領地よりも広いので先手の勝利である．

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，以下のように定義する．

入力: グリッドの行数 n
グリッドの列数 m
色集合 $C = \{1, 2, \ldots, k\}$
グリッドの各マスへの色の割当 $f : [1, n] \times [1, m] \to C$
$f(n', m') \in C$

出力: 次の操作での変更先の色 $i \in C$

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．Flood-It におけるモンテカルロ法を用いた AI では，受け取った盤面の情報に対して，プレイヤーの次の各色の選択肢からランダムな操作をし続けてゲームを終了させるプレイアウトという動作を繰り返す．プレイアウト後の勝敗数を記憶することで次の各色の選択肢毎の勝率を求め，勝率の最も高い色を出力する (図 4)．

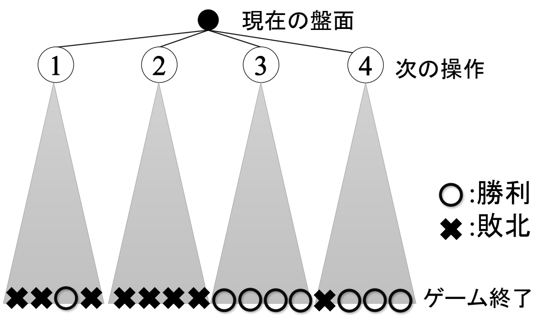

図 4: 各選択肢について 4 回ずつプレイアウトを行った例．この場合は，勝率の最も高い 3 を出力する．

3　今後の方針

まずはモンテカルロ法を用いた AI を作成し，既存の AI との勝率を確認する．その後，モンテカルロ法の改善アルゴリズムを AI に適用し勝率が上がるか確認する．

また，同じ盤面に対する既存の AI の操作や人間の操作とモンテカルロの操作を比較し，モンテカルロの長所および短所を探る．その長所や短所を基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に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, Lecture Notes in Computer Science 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.