情報システム評価学レポート

B5TB1058 小田将也

2019 年 1 月 23 日

1

Describe one of the algorithms given in the lecture (Segment intersection, Convex polygon intersection, Polygon area computation, Art Gallery problem, Polygon triangulation, Max Min-angle triangulation)

ここでは，Art Gallery problem の入力である頂点数 n のポリゴン P において，高々$[\frac{n}{3}]$ 人の警官の配置を出力するアルゴリズムについて述べる．

アルゴリズムは，以下の 3 つのステップによって構成されている．

1. 与えられたポリゴン P について，三角形分割を行う．
2. P と同様の頂点を頂点とし，P の辺と分割の際に引かれた線を辺とするグラフ G を作成し，その G について，3 彩色を求める．
3. ステップ 2 で得られた 3 彩色の 3 色の中で，最も使用された数が少ない色で塗られている頂点集合を警官の配置として出力する．

ステップ 1 の三角形分割については，既に知られているアルゴリズムを使って行う．

ステップ 2 について，一般のグラフにおいては常にそのグラフが 3 彩色が可能であるというわけではないが，今回のアルゴリズムで作成した G は 2 縮退グラフであることから，必ず 3 彩色が可能となっている．

ステップ 3 について，出力について考える．三角形の頂点に配置されている警官は，三角形が凸な多角形であることから，その三角形の内部の全ての範囲を警備することができる．

隣接した頂点が違う色で塗られていなければならないという彩色の条件から，それぞれの三角形の頂点は違う色で塗られているはずなので，3彩色の 3 色のうちのある色で塗られている頂点全てを出力として選べば，それぞれの三角形について，その範囲を警備できる警官が一人は居る事になるため，P の全ての範囲を警備することができる．

また，3 色の中で最も使用された数が少ない色を選択するので，その頂点の数は必ず $n/3$ 以下になっているはずである．従って，このアルゴリズムは，頂点数 n のポリゴン P において，高々$[\frac{n}{3}]$ 人の警官の配置を出力するアルゴリズムとなっている．

このアルゴリズムによって，Art Gallery problem に，$[\frac{n}{3}]$ という必要な警官の数の上界を与えることができる．

2

Explain your own research topic(how interesting it is, how important it is etc.) so that a high school student with interest in science will be attracted.

私の研究テーマは,「モンテカルロ法に基づく領地拡大型ゲームの対戦アルゴリズムに関する研究」である．私の研究では，領地拡大型ゲームの一つである,「Flood-It」という 2 人完全情報ゲームを用いて,「モンテカルロ法」という手法に基づいたコンピュータの意思決定に関するアルゴリズムを研究している．

ゲームという題材は，勝ち負けという目的がはっきりしているという性質から，人工知能分野の研究題材として広く用いられてきた．

ゲームの中でも，完全情報ゲームと呼ばれる，リバーシ (オセロ) や将棋などの運の要素が全くないようなゲームにおいては，完全にゲームの現在から先の局面を先読みすることができるため，コンピュータにとって特に扱いやすい題材となっている．

近年では，様々なゲームにおいて，コンピュータとそのゲームのプロが対戦するというシーンが多く見られ，そのゲームのプロにコンピュータが勝つといったことが，人工知能の発展を象徴するような出来事となっているともいえる．

先述した完全情報ゲームにおいては，コンピュータがゲーム終了までの未来を完全に先読みすることができれば，自分が必ず勝てるような手

を選択することができる最強の対戦アルゴリズムを簡単につくることができると考えられるが，実際はどうなのであろうか．リバーシを例にして考えてみる．

リバーシは，将棋やチェスなどの他のゲームに比べて，1 手の選択肢が少なく，一度置かれた駒が盤面から消えることはないため，8×8 の盤面では最大でも 60 手でゲームが終了する．しかしそんなリバーシですら，ゲームの終了盤面の個数は，10^{54} 程度と推定されている．これは，2018 年 11 月の時点で最も性能が良いとされる，毎秒 20 京回の計算が可能なスーパーコンピュータ「サミット」が一秒間に 20 京 $(2 * 10^{17})$ 通りの盤面を読むことができたとしても，全ての盤面を読むためには 10^{26} 年以上かかってしまうほど莫大な数字である．一手を打つのにこんなに待たされてしまっては，いくら強くても対戦アルゴリズムとしては不適切である．では，実際のアルゴリズムはどのような仕組みになっているのであろうか．

ゲームの対戦アルゴリズムとして広く用いられている手法に，mini-max 法と呼ばれるものが存在する．これはどういった手法かというと，先程述べたようにゲーム終了まで全ての盤面を先読みしていたのでは時間がかかりすぎるため，例えば 8 手先までといったような，現実的な時間で先読みできる範囲までで先読みを打ち切る．そして，その盤面での「評価値」と呼ばれる値を計算し，その値が，数手先の未来で最も良くなるような手を選ぶ，というようなアルゴリズムになっている．ここで,「評価値」というのは，高ければ高いほど自分にとって良い値とし，例えばリバーシで言うと，自分の石の数が最も多い，であったり，次の自分の選択肢が最も多い，といったような場合に高くなるような値を設定する．こういったアルゴリズムによって，ある程度強い対戦アルゴリズムを作ることが可能となっているが，このような手法を使うと，近い未来では最適になる手を選べているとしても，最終的に逆転されてしまうような手を打ってしまったり，囲碁のような複雑なゲームでは，そもそも「評価値」の設定自体が難しい，といった問題点も存在する．

このような問題に対して，簡単にアプローチできるような手法が，モンテカルロ法である．それでは，モンテカルロ法という手法がどのようなものであるかを簡単に説明する．

先程述べた mini-max 法と同様に，モンテカルロ法でも，最善に見える手を選ぶのには変わりはないのだが，その選び方に違いがある．mini-max 法が局所的に最善な手を打っているのに対して，モンテカルロ法では，最

終的に勝率の高そうな手を選択するのである．

具体的にどのようなことをするのかというと，まず，次に打てる手の選択肢それぞれから，自分と相手の手をランダムに選び続けてゲームを終了させ，その勝敗をカウントする．そしてこの試行を十分繰り返した後，それぞれの次の手の選択肢の中で，この試行で最も勝率の高かった手を選ぶ．このような操作をすることによって，最終的に勝率の高そうな手を選択するということを行う．この手法のメリットとしては，先ほどの mini-max 法と違い，ランダムではあるがゲーム終了までゲームを読むので，局所的には良くてものちに逆転されてしまうような手を選ぶ可能性が減るということがあげられる．また，モンテカルロ法であれば，ただ「ゲーム終了時の勝敗」を見るだけで済むので，mini-max 法で必要だった，ゲームごとの「評価値」の設定が必要なくなるため，様々なゲームに簡単に適用することができるというメリットもある．

そんな乱数を使ったある種適当な選択の仕方でちゃんと強い対戦アルゴリズムができるのかと疑問に思う人もいるかもしれないが，不思議なことに，ゲームのコツをしらないようなこの手法でも，ゲームによってはかなり強くなることが確かめられている．実際に，囲碁で結果を出した「AlphaGo」も，このモンテカルロ法を基にしたものとなっている．

私の研究では，このようなモンテカルロ法に基づいて，Flood-It というゲームのより強い対戦アルゴリズムの作成や，より高速で効率の良い先読みの手法の開発を行っている．