

アルゴ AI の設計と実装

k.seiya28@gmail.com

2017 年 6 月 16 日

0.1 アルゴのルール

アルゴ (algo) は市販の数あてゲームの一種で, 0~11 の数字が書かれたカード 12 枚が白と黒で計 24 枚のカードを使って 2 人でプレイする. ルールは以下の通りである.

- カードは基本ルール通りに並べ, 数字がわからないように伏せておく
対戦が始まる前に, シャッフルされた山札から 1 人 4 枚カードをとって自分の手札として伏せて配置する. このとき, 手札は自分の左から見て数字の小さい順に並べる. 同じ数字があった場合は, 黒のほうが小さいものとする.
- 対戦相手の, 伏せてあるカードの数字を推理して当てる (アタック)
山札からアタックカード引いてアタックする. アタックに成功したとき, 続けてアタックするかやめる (ステイする) かを選べる. カードが小さい順に並んでいることや, 相手のアタックなどの情報からどの位置にどの数字があるかを推理する.
- 相手のカードを先に全部当てたほうが勝つ

0.2 アルゴの性質

ボードゲームの分類では, アルゴは不完全情報ゲームである. 相手の手札は完全には定められず, 山札はシャッフルされたものを上から順に引いていく形式であり, プレイヤーは全ての情報を参照することができない.

1 回の対戦の勝敗ではプレイヤーの優劣を決定できないことがアルゴの特徴の一つとして挙げられる. 全ての初期状態についての対戦を行えば AI の性能を測ることができるが, アルゴの初期状態は山札の並びだけでも ${}_{24}P_{16}$ パターン存在し, 10^{19} を超えることがわかる.

0.3 手札候補の生成と絞り込み

このアルゴリズムの目的は, アタックして成功する確率が 0 でないアタックを列挙することである. 過程としては,

1. 相手の手札のカード数, 自分の手札カードから, 相手の手札がとりうる全てのカードの並びを生成
2. 相手の手札の見える分から候補を削る
3. 過去の自分のアタック失敗履歴から候補を削る
4. 残った候補から可能なアタックを列挙

過程 1 では, 相手の手札のカード数 k , 自分の手札のカード, アタックカードの情報をういて計算を行う. まず, ゲーム上の全 24 枚のカードから自分の手札のカードとアタックカードを除いた集合 A を考える. 例えば, 自分の手札が 4 枚だとすると, アタックカードは必ず 1 枚なので, A の要素数は $24 - 4 - 1 = 19$ となる. このとき A は, 相手の手札に存在しうるカードの集合になっている. つぎに, A に対する長さ k の全ての組み合わせを考え, その集合を D とす

る. D の要素数は A の要素数を a として ${}_aC_k$ である. アルゴのルールにおいては, カードの組み合わせが定まるとその並びが一意に決定される. よって, 全ての組み合わせを列挙する計算結果 D は全ての順列を列挙している計算結果に等しい. カードの並びの集合 D は相手の手札がとりうる全パターンを含んでいる.

過程 2 では, 過程 1 で生成した D を, 自分から見えているカードの情報を用いて削除し, 候補を絞り込む. 例えば, 相手の手札の左から 1 番目のカードがオープンになっていて, 数字が 1 であった場合, D の要素のうち左から 1 番目のカードの数字が 1 でない要素が削除される.

過程 3 では, 過去の自分のアタック失敗履歴から得られる情報をもとに D を絞り込む. 例えば, ゲームが始まってから相手の手札の左から 2 番目に 5 でアタックして失敗したとき, 相手の手札の左から 2 番目のカードは 5 ではないという情報が得られる. D の要素で 2 番目のカードが 5 のものを削除することで候補を絞り込むことができる.

過程 4 では, 残った候補から, アタックして失敗する確率が 0 でないアタックを得る. 相手の手札上の位置のうちカードがオープンになっていない位置について D 上から抽出する処理を行うことで有効なアタックを抽出できる.

手札の情報からアタック候補を列挙するまでの流れを簡単に示したものが図 1 である.

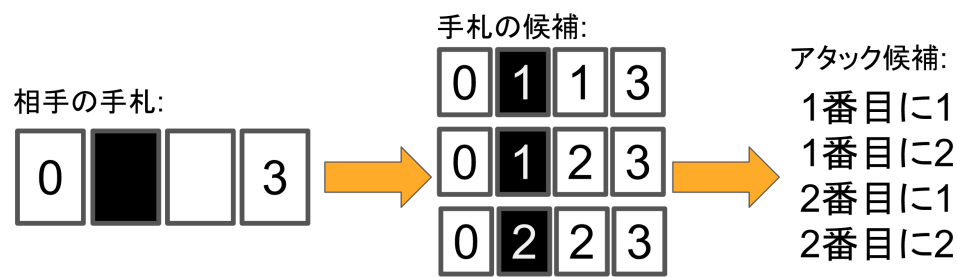


図 1 カード列で候補を保持するアルゴリズム

0.4 次のターンでのアタック候補数の期待値の計算

このアルゴリズムの目的は, アルゴリズム (5.1) で生成したアタック候補から, 最終的に実行するアタックを決定することである.

1. 列挙されたアタックについてアタック成功確率を計算
2. 列挙されたアタックについての次のターンの可能なアタック数の削減数の期待値を計算

手順 1 の方法について説明する. 手札候補の生成と絞り込み (6.1) のアルゴリズムを用いて生成した候補を D とする. 候補 D 内での重複に注目することで, アタックの成功確率を計算することができる. 具体的には, 候補 D 内のそれぞれの要素の i 番目に注目し, i 番目に数字 n が出現する頻度 r , D の要素数 l を用いて, 相手の手札の i 番目に n でアタックした場合の成功確率は

$$P((i, n)) = \frac{r}{l} \quad (1)$$

と計算できる.

手順 2 の方法について説明する. 可能なアタック数の削減数は, アタックが成功したと仮定したときの相手の手札を生成し, そのカード列についての可能なアタック数を計算することによって得られる. 具体的には, i 番目に n のアタックが成功したと仮定したカード列を d として, 手札候補の生成と絞り込み (6.1) のアルゴリズムを用いてカード列に対する可能アタック数 a_{next} を計算する. 現在の可能アタック数を a_{now} としたとき, 可能アタック数の削減数は $a_{\text{now}} - a_{\text{next}}$ となる. ここで, 手順 1 で計算したアタックの成功確率 $P((i, n))$ を用いると, 可能なアタック数の削減数の期待値は

$$P((i, n))(a_{\text{now}} - a_{\text{next}}) \quad (2)$$

となる．ここで， a_{now} はどのアタックを選ぶかによらずに決まる．相手の手札の情報から次のターンの可能アタック数を求める方法を簡単に示したものが図 2 である．

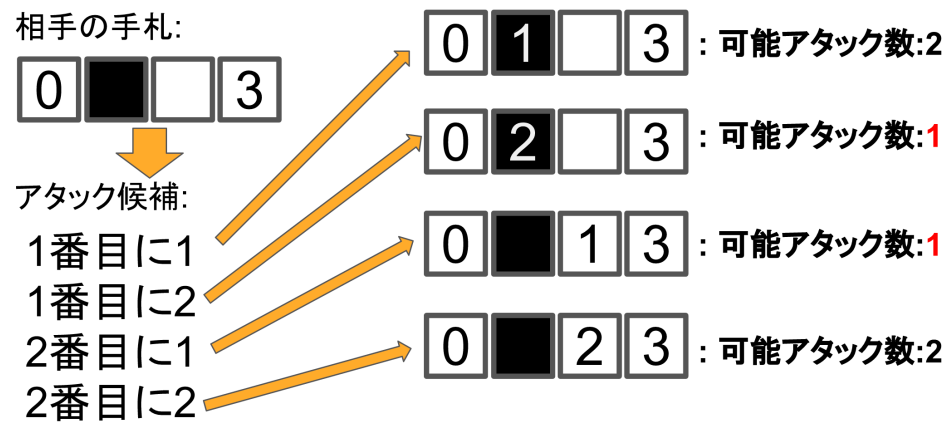


図 2 次のターンの可能アタック数を計算するアルゴリズム