2023 年 1 月 24 日 B3 西村昭賢

進捗報告

1 今週やったこと

- B3 発表用資料, スライド加筆修正
- 先週考えた手法の有効性を示そうとした実験
  - DQN を用いてどのような戦略を持つ相手にも平均的に勝つ戦略を構築し, 調整対象のカード (強すぎるカードと弱すぎるカード) を検出する
  - GA でパラメータの変更量を抑えつつ, 勝率が 50 % に近づくように調整する.

2 DQN

2.1 対戦相手の戦略の修正

対戦相手の戦略としてアグロとコントロールを用意した. また, よりルールベースな AI へと改良した. Algorithm 1, 2 に疑似コードを示す.

Algorithm 1 対戦相手の行動ルーチン (コントロール)

```
手札から盤面にカードを出せるだけプレイ (ドロー順が古い方から)
for 盤面のカード (プレイ順が古い方から) do
    if 敵盤面にカードがない then
        敵プレイヤーを攻撃
    else
        敵盤面カードの総攻撃力を計算
        自盤面カードの総攻撃力を計算
        if 自盤面カードの総攻撃力 >= 敵プレイヤーの残り HP then
            敵プレイヤーを攻撃
        end if
        if 敵盤面に有利トレードできるカードがある then
            そのカードを攻撃
        else
            if 敵盤面に相討ちできるカードがある then
                そのカードを攻撃
            else
                敵盤面で最も攻撃力が高いカードを攻撃
            end if
        end if
    end if
end for
ターンを終了
```

Algorithm 2 対戦相手の行動ルーチン (アグロ)

```
手札から盤面にカードを出せるだけ出す (ドロー順が古い方から)
for 盤面のカード (プレイ順が古い方から) do
  if 敵の盤面にカードがない then
    敵プレイヤーを攻撃
  else
    敵盤面カードの総攻撃力を計算
    自盤面カードの総攻撃力を計算
    if 自盤面カードの総攻撃力 >= 敵プレイヤーの残り HP then
      敵プレイヤーを攻撃
    end if
    if 残り HP が 10 以上であり, かつ敵盤面の総攻撃力より多い then
      敵プレイヤーを攻撃
    end if
    if 残り HP が 10 未満 then
      if 敵盤面に有利トレードできるカードがある then
        そのカードを攻撃
      else
        敵プレイヤーを攻撃
      end if
    end if
    if 敵盤面の総攻撃力 >= 残り HP then
      if 敵盤面に有利トレードできるカードがある then
        そのカードを攻撃
      else
        if 敵盤面に相討ちできるカードがある then
          そのカードを攻撃
        else
          敵盤面で最も攻撃力が高いカードを攻撃
        end if
      end if
    end if
  end if
end for
ターンエンド
```

2.2 実験条件

DQN で強すぎるカードと弱すぎるカードをどのように検出するかどうか調べることが目的. そのため, 表 1 デッキにあえて強すぎるカードと弱すぎるカードを含めて学習を行った.

デッキは対戦相手にも同じものを持たせた. 先攻側で 1000000 ステップ学習した. コントロールとアグロに平均的に勝利する戦略を学習するため, 1 エピソードごとに対戦相手の戦略を乱数で等確率に決定した. また, 比較材料として対戦相手の戦略を変化させない場合も学習した.

3 結果

学習済みのモデルで 50000 回ゲームを実行し勝率を記録した. 表 2 に結果を示す.

また, 表 3 に 3 つの学習済みエージェントにおける選択された総数が多い行動上位 5 つを示す.

このことから 3 つのエージェントとも同じように学習の結果として相手プレイヤーに攻撃して早くゲーム終了を迎えるアグロの戦略を構築したと考えられる. デッキに入れた強いカードは 1 コスト (5, 5) のカードであるため, アグロの戦略が有利と考えられる. そのため, 学習時の対戦相手に関係なくこのような結果になったと考えられる.

また, 表 2 において最も勝率が高かったアグロ相手に学習したエージェントについて 50000 回ゲームを実行した結果を用いてバランス調整に相応しいカードを選択するための指標について検討した. 今回用いる指標

表 1: 実験で用いた強すぎるカードと弱すぎるカードを含んだデッキ (太字が変更点)

ID	攻撃力	HP	コスト	特殊効果	枚数
0	**4**	**4**	1	無し	2
1	2	2	2	無し	2
2	3	3	3	無し	2
3	4	3	4	無し	2
4	5	4	5	無し	2
5	2	2	2	召喚	2
6	2	3	3	召喚	2
7	1	1	1	取得	2
8	1	3	2	取得	2
9	2	1	2	速攻	2
10	3	1	3	速攻	2
11	1	2	2	攻撃	2
12	2	3	3	攻撃	2
13	1	1	1	治癒	2
14	**1**	1	**5**	治癒	2

表 2: 50000 回のゲーム実行における勝率

勝率計算時の対戦相手 ＼ 学習時の対戦相手	ランダムに戦略が変化する	コントロール	アグロ
コントロール	0.87850	0.87884	**0.88158**
アグロ	0.67630	0.67362	**0.67852**

は, WRD (Win Rate when Draw), WRP (Win Rate when Play), PlayRate (そのカードが盤面にプレイされたゲーム数 / 総ゲーム数) の 3 つを用いた. 表 4, 5 に各指標の上位 3 つ, 下位 3 つのカードとその値を示す.

PlayRate に関して, 上位 3 つは コスト 1 のカードで占められ, 下位 3 つは コスト 4 の ID 3, コスト 5 の ID 14, 4 となっており構築されたアグロ戦略への適応度を表す指標であると考えられる. 仮に DQN で作成した戦略がデッキの最適な戦略であればこの指標を用いてその戦略におけるカードパワーを測る指標と考えられる.

WRD は上位 3 つがコスト 1 のカードであることからも戦略を踏まえたカードパワーの指標になっていると考えられる. 下位 3 つにもコストが 5 の ID 14, 4 のカード, 2 コスト (1, 3) で 1 枚ドローの ID 8 と構築したアグロ戦略とは噛み合わないカードが並んでいる.

WRP は盤面に出たときの勝率であるため戦略よりも単純なカードパワーを示していると考えられる. ID 0 はもちろんのこと, 他の指標では下位にランクインしていた ID 4 のカードが WRP では上位 3 つに入っていることからも WRP が単純なカードパワーを示していることが伺える. ただ, WRP の下位には数値上では強い ID 3 のカードが入っていたり, ID 14 のカードが 下から 4 番目にランクインしているなど下位になるにつれよくわからない指標になっていた.

今回は深層強化学習で平均的に勝てる戦略を構築しそれを元に調整対象のカードを選択するアプローチを取るため, WRD と PlayRate が調整対象のカード選択の指標として相応しいと考えている. なお, いずれの指標においても 「バランスブレイカー」だったり「弱すぎる」という値は具体的にどのような値で示されるのかがまだ決まっていない. 単純に上位のものを順に GA でバランス調整にかけることが無難かもしれない.

表 3: 3 種類の学習済みエージェントの行動

ランダムに戦略が変化		コントロール		アグロ	
行動説明	総数	行動説明	総数	行動説明	総数
ターンエンド	270153	ターンエンド	270674	ターンエンド	271243
手札 1 を盤面に出す	247143	手札 1 を盤面に出す	247081	手札 1 を盤面に出す	248676
盤面 1 で相手プレイヤーに攻撃	212046	盤面 1 で相手プレイヤーに攻撃	212709	盤面 1 で相手プレイヤーに攻撃	212380
盤面 2 で相手プレイヤーに攻撃	120773	盤面 2 で相手プレイヤーに攻撃	120803	盤面 2 で相手プレイヤーに攻撃	120952
手札 2 を盤面に出す	74531	手札 2 を盤面に出す	74769	手札 2 を盤面に出す	74579

表 4: 各指標上位 3 カード (降順)

WRD		WRP		PlayRate	
0	0.86888	0	0.89930	0	0.57976
13	0.79735	10	0.82776	13	0.53634
7	0.79133	4	0.81306	7	0.52146

4 GA を用いたカードのパラメータ (HP, 攻撃力, コスト) の調整

前々から参考にしているハースストーン環境におけるバランス調整を GA を用いて試みた先行研究 [1] では, 各デッキ間の勝率を 50 % にするために 単一目的 GA, 更にパラメータの変化量を少なくなるようにする多目的 GA を用いていた. 先行研究では, 多目的 GA として NSGA-2 を用いていたため本環境でも NSGA-2 を実装してみた [2].

表 7 のパラメータを示す. NSGA - 2 で表 8 に示すデッキにおいてランダムに行動する同士の対戦環境において後攻で勝率が 50 % になるように学習した.

パラメータの総変更量が 6 で後攻の勝率が 0.50 となった個体が存在した. 表 9 にそのデッキを示す. 先週の単一目的 GA ではパラメータの総変更量は 25 となったので, パラメータの変更量を減らしながら目的を達成できている. ただ, 勝率計算におけるゲームの実行数が 1000 回と少ないため数を増やしてより厳密な解を得たい.

5 今後の課題

- 新規性

  多目的最適化を誰もやってないと勘違いして NSGA-2 で新規性生めると思っていた. DQN で変更するパラメータを絞るなら, 勝率の他の目的関数をデッキの平均マナレシオに近づけるなどに変えてみればいいかもしれない.

- 卒論テーマ決め & 執筆

参考文献

[1] Fernando de Mesentier Silva, Rodrigo Canaan, Scott Lee, Matthew C. Fontaine, Julian Togelius, and Amy K. Hoover. Evolving the Hearthstone Meta. *arXiv e-prints*, p. arXiv:1907.01623, July 2019.

[2] `https://github.com/baopng/NSGA-II`.

表 5: 各指標下位 3 カード (昇順)

WRD		WRP		PlayRate	
14	0.76367	8	0.77762	14	0.34086
8	0.77028	3	0.78577	4	0.36054
4	0.77150	1	0.79052	3	0.40752

表 6: 前回の単一目的 GA で扱ったデッキの内容

ID	攻撃力	HP	コスト	特殊効果	枚数
0	1	2	1	無し	2
1	2	2	2	無し	2
2	3	3	3	無し	2
3	4	3	4	無し	2
4	5	4	5	無し	2
5	2	2	2	召喚	2
6	2	3	3	召喚	2
7	1	1	1	循環	2
8	1	3	2	循環	2
9	2	1	2	速攻	2
10	3	1	3	速攻	2
11	1	2	2	攻撃	2
12	2	3	3	攻撃	2
13	1	1	1	治癒	2
14	2	1	3	治癒	2

表 7: NSGA - 2 のパラメータ

パラメータ	値
目的関数 1	パラメータの総変更量
目的関数 2	$(0.50 - \mathrm{r_{win}})^2$
世代数	100
個体数	100
交叉	2 点交叉
勝率計算の際の回数	1000

表 8: GA で扱ったデッキの内容

ID	攻撃力	HP	コスト	特殊効果	枚数
0	1	2	1	無し	2
1	2	2	2	無し	2
2	3	3	3	無し	2
3	4	3	4	無し	2
4	5	4	5	無し	2
5	2	2	2	召喚	2
6	2	3	3	召喚	2
7	1	1	1	循環	2
8	1	3	2	循環	2
9	2	1	2	速攻	2
10	3	1	3	速攻	2
11	1	2	2	攻撃	2
12	2	3	3	攻撃	2
13	1	1	1	治癒	2
14	2	1	3	治癒	2

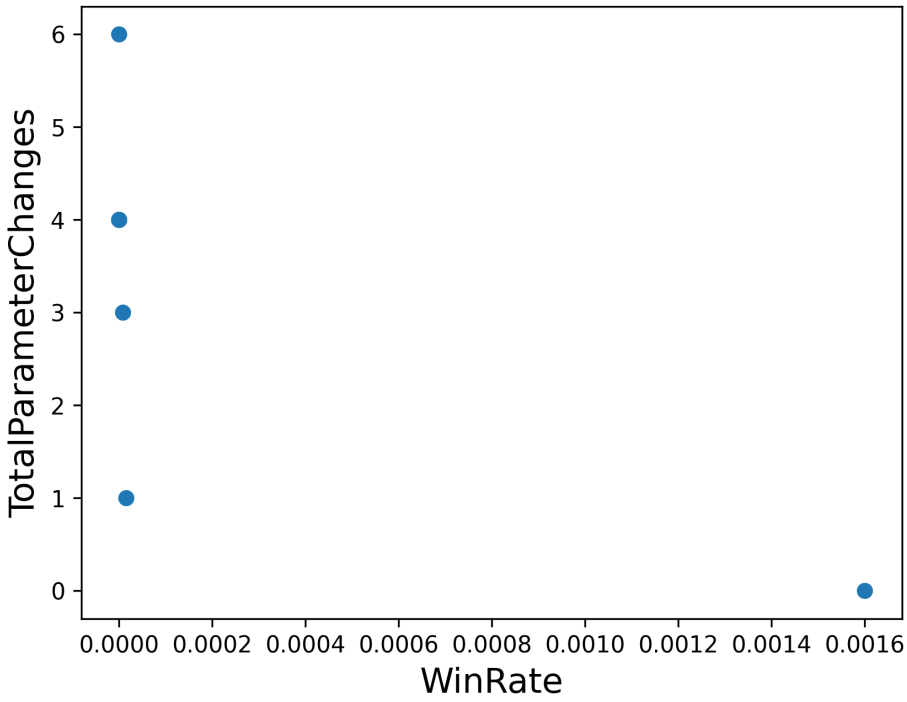

図 1: NSGA-2 を用いた最適化の結果 (青点はパレートフロント)

表 9: NSGA-2 で得た総変化量 6 のデッキの内容 (太字は表 8 からの変更箇所)

ID	攻撃力	HP	コスト	特殊効果	枚数
0	1	**1**	1	無し	2
1	2	2	2	無し	2
2	3	3	3	無し	2
3	4	3	4	無し	2
4	5	4	5	無し	2
5	2	2	2	召喚	2
6	2	3	**1**	召喚	2
7	1	1	1	循環	2
8	1	3	**1**	循環	2
9	2	1	2	速攻	2
10	3	1	3	速攻	2
11	1	**4**	2	攻撃	2
12	2	3	3	攻撃	2
13	1	1	1	治癒	2
14	2	1	**2**	治癒	2