Vol.2019-GI-41 No.18
2019/3/9

大貧民における モンテカルロ法の報酬値に関する研究

土橋 康希[1,a)]　大久保 誠也[2,b)]　若月 光夫[1,c)]　西野 哲朗[1,d)]

概要：大貧民には，階級制度と呼ばれるルールがあり，順位間の価値差は順位点以上に広がる．コンピュータ大貧民においてモンテカルロ法を用いる場合，報酬値は順位点ではなく，このルールの効果を加味した値にすることで，シミュレーションの精度向上が望める．報酬値を工夫した対戦プログラムの例として，snowl は順位点を 2 乗した値を，wisteria は階級ごとに異なる期待値を算出し，この値を順位点と足し合わせた値を報酬値に用いている．本研究では，既存の対戦プログラムが行っている報酬値の工夫によって，報酬値を順位点とした場合より強くなるかを対戦実験にて検証した．順位点に期待値を足し合わせた値を報酬値とした場合は，純粋なモンテカルロ法を用いている対戦プログラム (MSM04) でのみ強くなる傾向がみられ，順位点を 2 乗した値を報酬値に用いた場合では，2 つの対戦プログラム (MSM04，snowlmono) において強くなる傾向がみられた．また，snowl の報酬値について，大富豪の報酬値を更に上げた場合により強くなる傾向があったことから，順位点を 2 乗した値をベースに大富豪の報酬値を更に高くすることで，モンテカルロ法を用いている大貧民対戦プログラムを強化できることがわかった．

1. はじめに

ゲームは，プレイヤーがすべての情報を共有して行う完全情報ゲームと，一部の情報が隠されて行われる不完全情報ゲームの 2 種類に分けられる．不完全情報ゲームの例として，麻雀やトランプを使ったゲームがある．そのトランプゲームの一つである大貧民について，コンピューター同士で大貧民をプレイさせる UEC コンピューター大貧民大会（以降，UECda と呼ぶ）が，2006 年から開催されており，大会では数千試合における合計得点によって対戦プログラムの優劣を競い合う [1]．計算時間を必要とする部門における対戦プログラムの多くはモンテカルロ法を搭載している．

大貧民には，前の試合の順位を受けて初期手札でカード交換を行う階級制度と呼ばれるルールがある．大貧民においてモンテカルロ法を用いる場合，シミュレーション時にもらえる得点（報酬値と呼ぶ）は実際の試合における得点に加えて，各階級ごとに異なる期待値を加味した値にした方が真の利得に近くなり，シミュレーションの精度が向上すると考えられる．

[1] 電気通信大学大学院情報理工学研究科 情報学専攻
[2] 静岡県立大学経営情報学部経営情報学科
[a)] y-dobashi@uec.ac.jp
[b)] s-okubo@u-shizuoka-ken.ac.jp
[c)] wakatsuki.mitsuo@uec.ac.jp
[d)] nishino@uec.ac.jp

2. 研究目的

大貧民対戦プログラムにおける報酬値の例として，須藤氏が作成した snowl は順位点を 2 乗した値を，大渡氏によって開発された wisteria は階級による利得差を 14 試合分まで考慮した得点を，それぞれ報酬値として使用している [2][3]．我々はこの 2 つの対戦プログラムにおいて報酬値を変更して対戦実験を行い，wisteria では報酬値を順位点に 1 から 3 試合分の得点期待値を足し合わせた値にすると元の対戦プログラムより強くなる結果を得ている [4][5]．しかし，対戦実験の試合数が不十分であったり，同じモンテカルロ法を用いたプログラムでもこの手法が有効でない理由が明らかになっていないなどの課題が挙げられる．

本研究では，上記 2 つの対戦プログラムに加え，単純なモンテカルロ法を用いている MSM04 を用いて，報酬値変更が強さに与える影響を更に詳しく解析することを目的とする．具体的には，snowl 及び wisteria が用いている報酬値や実験過程で有力とされた報酬値を 3 つの対戦プログラムに適用し，報酬値を順位点に設定した元の対戦プログラムとの対戦実験により，大貧民において報酬値を変更することが対戦プログラムの強化につながるかどうかの検証を行った．

3. 大貧民におけるルール

本研究では，大貧民のルールに UECda 標準ルールを用いる [6]．獲得した順位によって次の試合で行われる初期手札のカード交換制度及び順位点を表 1 に示す．

表 1：UECda 標準ルールにおける階級制度と順位点

順位	階級名	カード交換の内容	順位点
1 位	大富豪	大貧民に任意のカードを 2 枚渡す	5 点
2 位	富豪	貧民に任意のカードを 1 枚渡す	4 点
3 位	平民	カード交換なし	3 点
4 位	貧民	富豪に最も強いカードを 1 枚渡す	2 点
5 位	大貧民	大富豪に強いカードを順に 2 枚渡す	1 点

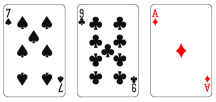
4. モンテカルロ法

モンテカルロ法とは，シミュレーションなどを乱数を用いて実行する手法の総称である [7]．基本的なアルゴリズムとして，囲碁や将棋のある局面において指し手を選ぶとき，各指し手に対してその後自他共にランダムに手を指し続けることで終局における勝敗を得る．この一連の動作をプレイアウトと呼ぶ．このプレイアウトを何度も行って各指し手の勝率を計算し，この結果勝率が一番高い手を次の指し手とする手法である．多人数ゲームにおいては，最終的な順位を数値化した値を使ってシミュレーションを行った後のそれぞれの指し手における合計点を比較することで，それぞれの手の優劣を決める．順位を数値化した値を報酬値という．

図 1 は大貧民におけるモンテカルロ法の例であり，プレイアウトを計 50 回行った時の各行動の点数を例示している．報酬値は表 1 の順位点をそのまま使用しており，図 1 の例では一番平均点が高い ♠7 を選択する．以後，本稿において，各階級ごとの報酬値を (5,4,3,2,1) などと表すことがある．

場のカード

自分の手札



計 50 回のプレイアウト結果

順位	♠7	♣9	◇A	パス
1 位	12	3	1	0
2 位	3	1	2	1
3 位	2	4	0	1
4 位	1	3	3	2
5 位	2	2	4	3
合計点	82	39	23	14
平均点	4.1	3.0	2.3	2.0

図 1 大貧民におけるモンテカルロシミュレーションの例

5. 大貧民対戦プログラム

本節では，実験に用いた 3 つの対戦プログラムについて説明する．

5.1 MSM04

MSM04 は石巻専修大学の学生によって作成された第 6 回 UECda の出場プログラムである．snowl や wisteria とは異なり，シミュレーションの際事前に学習されたデータ等は用いていないため，ランダムシミュレーションを行う純粋なモンテカルロ法を用いている対戦プログラムとして本研究では利用した．本研究における提案手法を正しく反映できるようにソースコードの playgame.c 内の一部記述の削除したり書き換えたりして，枝刈りや特定盤面において報酬値の変更は行わずに，各行動に対して 1000 回のシミュレーションを行うように改変を行った．

5.2 snowl

snowl は第 5 回 UECda で優勝したプログラムであり，須藤らによって開発された [2]．手番における行動をモンテカルロ法で決定する．具体的には，手番で取りうる行動それぞれに対して終局までランダムシミュレーションを 2000 回程度行い，得られた得点の平均が最も高かった行動をその手番における行動として選択する．

- シミュレーション回数の分配
  より見込みのある手を重点的にシミュレーションする方法として，UCT アルゴリズムの一つである UCB1-TUNED を搭載している [8]．これは現在のシミュレーション回数を n, 行動 j のシミュレーション回数を T_j, 現在の得点の平均を $\overline{X_j}$ とすると UCB（Upper Confidence Bound）値は以下の式で表される．

$$UCB(j) = \overline{X_j} + \sqrt{V \times \ln(n)/T_j} \tag{1}$$

  ただし，$V = \min\{0.25, \overline{X_j^2} - \overline{(X_j)}^2 + \sqrt{2\ln(n)/T_j}\}$ とし，V を最大化する行動を次にシミュレーションするという方策である．これによって，シミュレーションを均等に分配した場合と比べて良い行動が得られる．
- シミュレーション中の方策
  snowl ではシミュレーション中の行動決定に，Policy Gradient Simulation Balancing により得た評価関数をもとにした方策を用いている [9]．評価関数で考慮する特徴は，手札に存在する各強さの単体と 2〜4 枚のペア及び 3〜5 枚の階段の数とした．各役に対して，通常時と革命時の 2 通りのパラメータを用意し，状態の評価に用いている．
- シミュレーション中に用いる報酬値
  snowl は，報酬値の工夫として，各階級ごとの報酬値を 2 乗値にする工夫が施されている．報酬値を変更する際，この 2 乗の工夫によって本来意図した値と異なる値になってしまうため，実験ではこの 2 乗の工夫を取り除いた snowl を用いる．本稿ではこの snowl を snowlmono と呼ぶ．

5.3 wisteria

wisteria は，第 10 回 UECda の無差別級で優勝したプログラムであり，大渡氏によって開発された [3]. 行動決定のアルゴリズムとして，まず必勝手探索を行い，見つからなかった場合は方策関数とモンテカルロ法を使用して指し手を選択している.

- モンテカルロシミュレーション
  使用する手札配置サンプルを選び，方策関数によって 1 手ずつシミュレーションを進める．シミュレーションの最低回数は 5000 と定められており，snowl 同様にシミュレートする行動は UCB1-TUNED によって決定しているが，(1) 式の $\sqrt{V \times \ln(n)/T_j}$ に $\sqrt{6}$ を掛けて使用している．残り 2 人になった場合は探索してどちらが勝つか調べ，シミュレーションを終了する．3 人以上残っている場合でも簡単な必勝手の探索を行って，見つかればその手でシミュレーションを進める. また報酬値については，階級リセットまでの試合数が 14 試合以上の時は 14 試合分の期待値を，それ以下の場合はその試合数分の期待値を順位点に加えた値を使用している.
- 方策関数
  カード交換や役の提出のための方策関数はいずれも線形関数による softmax 方策としている. 状態 s で行動 a をとる際の特徴ベクトルを $\phi(s,a)$，各特徴に対する重みベクトルを θ で表す．この時，状態 s で可能な行動の集合を A としたとき，行動 $a \in A$ を取る確率は次のように表される.

$$\pi_\theta(s,a) = \frac{e^{\phi(s,a)\cdot\theta}/T}{\sum_{b\in A} e^{\phi(s,b)\cdot\theta}/T}$$

  ただし T は温度パラメータと呼ばれ，この値を大きくすると方策はランダムに近づく. 方策関数でプレイする場合は $T = 0$，手札推定とシミュレーションでは $T = 1$ としている.（特徴ベクトルの詳細については省略する.）

6. 対戦プログラムごとの期待値の算出

大貧民における期待値は，順位だけでなく，その対戦プログラムの相対的な強さによっても異なる．本手法の検証のために行う対戦実験は，元の対戦プログラムと対戦させて比較する方法を採用したため，対戦相手は自分と同様の強さを持つ対戦プログラムと仮定し，期待値の算出には同じ対戦プログラムを 5 体対戦させた統計を用いた．なお wisteria に関しては，期待値込みの報酬値が元々搭載されているため，この値を実験で用いた.

6.1 階級遷移確率の算出

期待値を求める準備として，各階級間の遷移確率を統計によって算出する．snowlmono，及び MSM04 について，同じ大貧民対戦プログラム 5 体を 10001 回対戦させ，階級から階級への各遷移回数を調べ，各値を母数で割った値を遷移確率として用いた．表 2・表 3 に各対戦プログラムの階級遷移確率を示す.

表 2 統計によって算出した MSM04 における階級遷移確率

確率	大富豪	富豪	平民	貧民	大貧民
大富豪	0.467	0.273	0.136	0.064	0.060
富豪	0.301	0.309	0.198	0.107	0.085
平民	0.124	0.200	0.265	0.226	0.187
貧民	0.061	0.128	0.228	0.303	0.279
大貧民	0.046	0.093	0.173	0.299	0.389

表 3 統計によって算出した snowlmono における階級遷移確率

確率	大富豪	富豪	平民	貧民	大貧民
大富豪	0.530	0.267	0.117	0.055	0.031
富豪	0.295	0.353	0.189	0.104	0.060
平民	0.103	0.191	0.290	0.233	0.183
貧民	0.040	0.128	0.231	0.331	0.290
大貧民	0.031	0.082	0.172	0.279	0.437

表 2〜表 3 の見方の例として，例えば富豪から平民になる確率は MSM04 なら 19.8%，snowlmono なら 18.9%といった形である. snowlmono と MSM04 の階級遷移確率を比較すると，大富豪から大富豪・富豪から富豪・平民から大富豪・平民から平民・大貧民から大貧民のそれぞれの確率が他の値に比べて 3%〜6%程異なっている．本稿には示されてないが，対戦プログラムの強さは snowlmono の方が MSM04 より強く，対戦プログラムの実力が強いほど同じ階級を維持する傾向があると推察される.

6.2 得点期待値の算出

得点期待値の算出例として，表 2 の MSM04 の階級遷移確率を用いて説明する．階級遷移確率を行列として表したものを階級遷移行列 R とし，表 2 から R を (2) 式のように表す.

$$R = \begin{bmatrix} 0.467 & 0.273 & 0.136 & 0.064 & 0.060 \\ 0.301 & 0.309 & 0.198 & 0.107 & 0.085 \\ 0.124 & 0.200 & 0.265 & 0.226 & 0.187 \\ 0.061 & 0.128 & 0.228 & 0.303 & 0.279 \\ 0.046 & 0.093 & 0.173 & 0.299 & 0.389 \end{bmatrix} \quad (2)$$

次に，現在行っている試合の結果によって獲得する各階級において，現在行っている試合から n 試合後の試合で見込まれる得点期待値を行列 E_n とし，(3) 式のように計算する．R^n は，現在の試合から n 試合後の階級遷移確率を表しており，(4) 式は，E_n の各行の値について説明している.

$$E_n = R^n \times \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} \tag{3}$$

$$E_n = \begin{bmatrix} \text{大富豪における n 試合目の得点期待値} \\ \text{富豪における} \quad 〃 \\ \text{平民における} \quad 〃 \\ \text{貧民における} \quad 〃 \\ \text{大貧民における} \quad 〃 \end{bmatrix} \tag{4}$$

また，各階級における現在の試合から n 試合後までの得点期待値を P_n とすると，P_n は (5) 式で求められる．本稿では，この P_n の値を順位点に期待値を加味した報酬値として用いる．

$$P_n = \sum_{i=0}^{n} E_i \tag{5}$$

上記の方法によって求めた P_n の値を，MSM04 については表 4，snowlmono については表 5 に示す．また便宜上，E_0 及び P_0 の値を順位点として $E_0 = P_0 = (5\ 4\ 3\ 2\ 1)^T$ とする．表 6 には wisteria が元々搭載している報酬値を示す．wisteria は，5.3 項の末尾で述べたように 1 から 14 試合分の各期待値を順位点に足した値を報酬値として搭載している．

期待値 E_n は，いずれの対戦プログラムにおいても約 15 試合ほどで平均点の 3 点に収束するため，表には 15 試合先までの期待値を掲載した．また，本研究では 15 試合先までの期待値を累計した P_{15} を，期待値を最大まで考慮した報酬値として利用した．

7. 検証する各報酬値の名称

実験で検証する報酬値は，順位点に期待値を加えた値以外に，実験過程で有力とされた順位点を 2 乗した値やそれに準じた値についても検証した．

7.1 検証した各報酬値の名称と詳細

- ori 版
  報酬値に UECda ルールにおける順位点である (5,4,3,2,1) を用いた場合に ori 版と名付ける．snowlmono の ori 版，MSM04 の ori 版などと呼ぶ．
- f*n* 版
  報酬値に 6 節で求めた Pn を用いた場合に f*n* 版 (n は 1 から 15 の数字) と名付ける．対戦プログラムによって得点期待値は異なるため，MSM04 なら表 4，snowlmono なら表 5 の値を使用する．なお wisteria については，表 6 に示したように元々搭載されている報酬値を使用する．

  また f*n* 版におけるすべてのパターンで対戦実験を行うのは困難であったため，snowlmono，MSM04 については f1・f3・f5・f15 版を，wisteria は f1・f3・f5・f14 版のみを実験で検証した．
- sq 版，sq-版，sq+版
  snowl が元々用いている順位点の 2 乗値も対戦プログラムを強くする報酬値の一つと推察したため，順位点の 2 乗値，すなわち (25,16,9,4,1) も有力な値として実験した．

  また sq 版と f15 版の違いとして，7.2 項でに示す表 7〜表 9 を用いて比較すると，大富豪と富豪の差が大きく異なり，sq 版の方が倍近く大きいことが分かる．snowlmono の実験結果も踏まえ，大富豪の価値をより

表 4 現在の試合から n 試合先までの得点期待値の累計 P_n(MSM04)

試合数	大富豪	富豪	平民	貧民	大貧民
0	5.000	4.000	3.000	2.000	1.000
1	9.024	7.636	5.846	4.388	3.106
2	12.563	10.970	8.753	7.062	5.654
3	15.845	14.145	11.703	9.891	8.420
4	18.993	17.236	14.677	12.802	11.298
5	22.070	20.284	17.663	15.756	14.234
6	25.111	23.309	20.656	18.732	17.201
7	28.132	26.323	23.653	21.719	20.184
8	31.144	29.331	26.652	24.713	23.176
9	34.150	32.335	32.651	27.710	26.172
10	37.154	35.338	32.651	33.709	29.170
11	40.156	38.340	35.652	33.709	32.169
12	43.158	41.342	38.652	36.709	35.169
13	46.160	44.343	41.653	39.710	38.170
14	49.161	47.345	44.654	42.711	41.170
15	52.162	50.346	47.655	45.711	44.171

表 5 現在の試合から n 試合先までの得点期待値の累計 P_n(snowlmono)

試合数	大富豪	富豪	平民	貧民	大貧民
0	5.000	4.000	3.000	2.000	1.000
1	9.209	7.720	5.800	4.279	2.9928
2	12.947	11.158	8.6530	6.829	5.415
3	16.394	14.421	11.560	9.555	8.070
4	19.665	17.579	14.503	12.390	10.863
5	22.828	20.674	17.469	15.290	13.739
6	25.927	23.732	20.448	18.230	16.663
7	28.986	26.767	23.435	21.194	19.618
8	32.022	29.788	26.428	24.172	22.591
9	35.044	32.800	29.423	27.159	25.574
10	38.057	35.808	32.421	30.151	28.564
11	41.065	38.813	35.419	33.146	31.558
12	44.069	41.815	38.418	36.143	34.555
13	47.072	44.817	41.417	39.141	37.552
14	50.074	47.818	44.417	42.140	40.551
15	53.075	50.819	47.417	45.140	43.550

表 6　wisteria に搭載されている報酬値

試合数	大富豪	富豪	平民	貧民	大貧民
0	400	300	200	100	0
1	627	472	283	122	0
2	766	576	327	130	0
3	852	640	352	134	0
4	906	679	368	137	0
5	940	704	378	138	0
6	961	719	384	139	0
7	975	729	388	139	0
8	983	735	390	140	0
9	988	739	392	140	0
10	991	741	393	140	0
11	993	743	393	140	0
12	995	744	394	140	0
13	996	744	394	140	0
14	996	745	394	140	0

高く見積もった方が強くなると推測された．

そのため，類似の値として f15 版と sq 版の間を取った，大富豪の報酬値を下げて sq の大富豪・富豪間の報酬値を 0.75 倍にしたバージョンを sq-，sq 版の大富豪の報酬値を上げることで，大富豪・富豪間の報酬値を 1.5 倍にしたバージョンを sq+として，大富豪・富豪間の報酬値の変更が強さに関係した要素かどうかを sq 版・sq-版・sq+版を用いて調べた．

7.2　各報酬値の正規化

検証する各報酬値の違いをこのままの値で比較することは困難であるため，大富豪と大貧民の差が 1 になるように各階級の報酬値を等倍し，階級間の値の差の比率を用いて比較を行った．表 7〜表 9 に MSM04・snowlmono・wisteria の各対戦プログラムにおいて，本稿で検証する各報酬値をを正規化し，階級間の差の値を表にしたものを示す．

f1・f3・f5・f15 版では，報酬値に含める期待値の累計が大きくなるほど富豪と平民間の差が大きくなり，貧民と大貧民間の差が小さくなっている．しかし大富豪と富豪間・平民と貧民間の差はほとんど変化しない．期待値を考慮することは，相対的に大富豪・富豪の価値を高く見積もり，平民・貧民・大貧民の価値を小さく見積もるものの，大富豪と富豪間及び平民と貧民間の価値差はほとんど変わらないことがわかる．

8.　対戦実験とその結果

f*n* 版・sq 版及びその派生の sq-，sq+版の報酬値を変更した対戦プログラムに対して，報酬値に順位点を用いた ori 版 4 体を対戦相手として対戦させ，10000 試合を 1 セットとして各 12 セット行う．帰無仮説 H_0 を「平均点は 30000 点である」，対立仮説 H_1 を「平均点は 30000 点より大きい」もしくは「平均点は 30000 点より小さい」として有意水準 0.05 で報酬値を変更した対戦プログラムに対して片側 t 検定を行い，報酬値を順位点とした対戦プログラムより強くなるかどうかを検証する．なお，f*n* 版については f1・f3・f5・f15 版のみを実験し，wisteria については元々搭載されている f*n* 版に相当する報酬値を用いて f1・f3・f5・f14 版を実験した．図 2〜図 4 に各対戦プログラムにおける，それぞれの報酬値変更プログラムの平均点を散布図で示す．またエラーバーは標準偏差である．表 10 に t 検定時の統計検定量 T を示し，対立仮説「平均点は 30000 点より大きい」が採択された場合には青色を，対立仮説「平均点は 30000 点未満である」が採択された場合には橙色を付けた．

表 7　各報酬値における階級間の値の差 (MSM04)

名称	大富豪・富豪間	富豪・平民間	平民・貧民間	貧民・大貧民間
ori 版	0.25	0.25	0.25	0.25
f1 版	0.23	0.30	0.25	0.22
f3 版	0.23	0.33	0.24	0.20
f5 版	0.23	0.33	0.24	0.19
f15 版	0.23	0.34	0.24	0.19
sq-版	0.31	0.32	0.23	0.14
sq 版	0.38	0.29	0.21	0.13
sq+版	0.47	0.25	0.18	0.11

表 8　各報酬値における階級間の値の差 (snowlmono)

名称	大富豪・富豪間	富豪・平民間	平民・貧民間	貧民・大貧民間
ori 版	0.25	0.25	0.25	0.25
f1 版	0.24	0.31	0.24	0.21
f3 版	0.24	0.34	0.24	0.18
f5 版	0.24	0.35	0.24	0.17
f15 版	0.23	0.36	0.24	0.17
sq-版	0.31	0.32	0.23	0.14
sq 版	0.38	0.29	0.21	0.13
sq+版	0.47	0.25	0.18	0.11

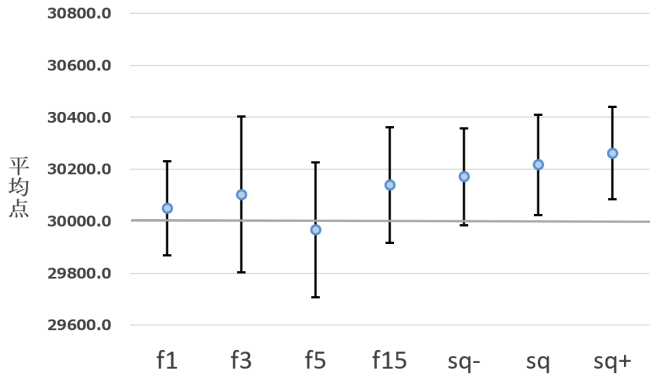


図 2　MSM04 の各報酬値変更版における平均点と標準偏差

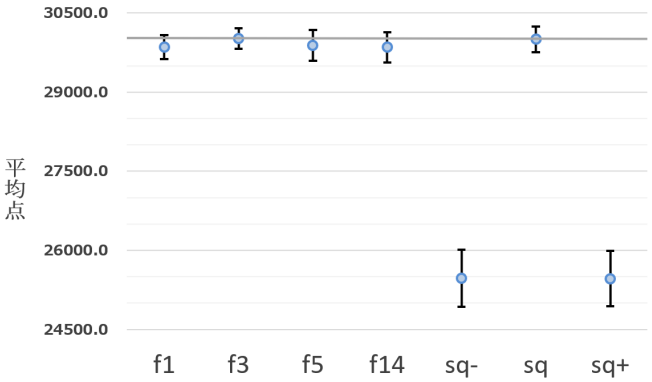


図 4　wisteria の各報酬値変更版における平均点と標準偏差

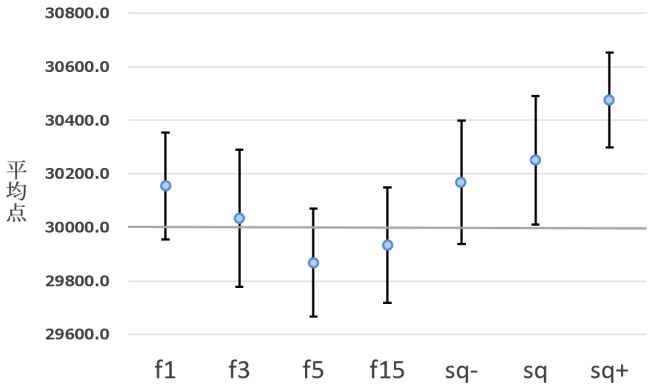


図 3　snowlmono の各報酬値変更版における平均点と標準偏差

9. 考察

考えられる対戦プログラムによって結果が異なる原因を 3 つ列挙する．

1 つ目は，シミュレーション回数と割り振りの違いである．MSM04 は各提出手に対して同じ 1000 回のシミュレーションを行うのに対し，snowlmono と wisteria は合計でそれぞれ約 2000 回，最低 5000 回と定められている．そのため，選択肢が多い序盤において snowl や wisteria で十分なシミュレーションが行えていないことが考えられる．

またシミュレーションの割り振りの工夫として，見込みのある手に重点的にシミュレーションを割り振る UCB1-TUNED を snowlmono と wisteria は用いている．

snowl は，大富豪の報酬値が 1 になるよう各報酬値を大富豪の報酬値で割った値を UCB1-TUNED に用いている．順位点に期待値を加えた値を報酬値に用いると階級間の報酬値の差は大きくなるが，UCB1-TUNED で使用する際の報酬値の最大値と最小値，すなわち大富豪と大貧民の報酬値の差は小さくなっていくため，(1) 式の $\sqrt{V \times \ln(n)/T_j}$ で表される誤差の値が相対的に小さくなり，シミュレーショ

表 9　各報酬値における階級間の値の差 (wisteria)

名称	大富豪・富豪間	富豪・平民間	平民・貧民間	貧民・大貧民間
ori 版	0.25	0.25	0.25	0.25
f1 版	0.24	0.30	0.25	0.21
f3 版	0.25	0.33	0.25	0.18
f5 版	0.25	0.34	0.25	0.17
f15 版	0.25	0.35	0.24	0.16
sq-版	0.31	0.32	0.23	0.14
sq 版	0.38	0.29	0.21	0.13
sq+版	0.47	0.25	0.18	0.11

表 10　各報酬値変更プログラムの t 検定における, 母平均の検定統計量 T

名前	f1	f3	f5	f15（f14）	sq-	sq	sq+
MSM04	0.969	1.192	-0.439	2.158	3.165	3.901	5.106
snowlmono	2.675	0.460	-2.250	-1.073	2.529	3.623	9.272
wisteria	-2.240	0.286	-1.334	-1.816	-29.099	-0.004	-29.949

ンの割り振りがうまく機能していない可能性が考えられる．

これは snowlmono と MSM04 において，順位点の 2 乗とそれに似た報酬値を用いた sq-・sq・sq+版でほぼ同じ結果が出たことも助長しており，この 3 つの報酬値変更プログラムについては，UCB1-TUNED で適用される報酬値の最大値と最小値の差は sq+，sq，sq-版の順に大きい．

wisteria では，UCB1-TUNED の上記の問題を解決する方法として，表 6 のように大貧民の報酬値が 0 になるよう，各階級における報酬値を大貧民の報酬値で引いた値を報酬値に用いている．しかし，wisteria はこの工夫に加えて snowlmono よりシミュレーション回数が倍以上に多いにも関わらず wisteria と snowlmono の fn 版に関する結果は類似した結果となっている．先ほど述べたシミュレーション回数の不足やシミュレーションの割り振りが上手く機能していないことを否定する結果だが，そもそも大貧民の報酬値を 0 となるように報酬値を変更する wisteria の工夫が与える UCB1-TUNED への有効性は示されていないことや，複合的な要因がかみ合っている可能性も十分考えられる．

どちらにしても，大富豪のみを高くする報酬値はシミュレーション回数に関する工夫を問わず対戦プログラムを強化できると考えられるが，大富豪・富豪を高く見積もる fn 型の報酬値は UCB1-TUNED と相性が悪いと推測された．しかし snowlmono の f1 版は順位点とした場合より強くなっているため，大富豪と富豪の両方を高く見積もる報酬値は対戦プログラムを強化する報酬値の方針として一理あると推測される．

2 つ目に，シミュレーションを実践的な内容に近づける工夫が関与していると考えられる．snowl 及び wisteria は，いずれも予め学習した評価関数の方策を用いてシミュレーション内のプレイヤの提出手を決定している．wisteria はこれに加えて，現在の試合進行と類似した状況を導くような相手手札を推定したうえでプレイアウトを行うため，よりシミュレーションの質は向上していると推測できる．このような工夫は限られた回数でより精度の高いシミュレーションをする場合に有効であるが自身で定めた方策によって対戦相手の提出手を選択するため，想定するシミュレーションの幅が狭くなることが考えられる．MSM04 のようなランダムシミュレーションの方が，想定される試合進行を広く均等にシミュレートできるため，報酬値への期待値加味の効果が現れやすいのではないかと考えられる．

3 つ目に，対戦相手の強さの違いが考えられる．3 種類の対戦プログラムにおける強さの関係は，MSM04 が一番弱く，wisteria が一番強い．本論文における実験の対戦相手は，報酬値を順位点に変更した元の対戦プログラムであり，3 種類の対戦プログラムそれぞれの対戦実験において対戦相手の強さが異なっている．そのため，対戦相手が snowl 程度のある一定レベル以上の場合，期待値を考慮する手法が有効でないことが考えられる．

10. おわりに

モンテカルロ法を用いた大貧民対戦プログラムにおいて，報酬値変更が強さに与える影響を調べるため，対戦プログラムを 3 種類用いて，いくつかの報酬値について対戦実験を行って検証した．各階級ごとの報酬値を，順位点にそれぞれの期待値を加えた値にすることでより実際の利得を反映させた *fn* 版では，純粋なモンテカルロ法を用いている MSM04 においては有効な場合が見受けられたが，シミュレーションに工夫をした snowlmono や wisteria については期待値を考慮した値にするほど弱くなる傾向があった．順位点の 2 乗値及びそれに準ずる報酬値は 2 つの対戦プログラムについて有効であり，大富豪の報酬値を大きくすると対戦プログラムの強さを向上できる．シミュレーション回数や UCB1-TUNED の適用を変更した対戦プログラムによる実験が今後の課題として挙げられる．

参考文献

[1] UECda-2018 コンピュータ大貧民大会 : http://www.tnlab.inf.uec.ac.jp/daihinmin/2018/ (2019.02.13).
[2] 須藤 郁弥, 成澤 和志, 篠原 歩: UEC コンピュータ大貧民大会向けクライアント「snowl」の開発, 第 2 回 コンピュータ大貧民シンポジウム (2010)
[3] 大渡勝己, 田中哲郎: 方策勾配を用いた教師有り学習によるコンピュータ大貧民の方策関数の学習とモンテカルロシミュレーションへの利用, IPSJ SIG Technical Report, vol.2016, G135, No.10 (2017)
[4] Mitsuo Wakatsuki, Yasuki Dobashi, Tasuku Mitsuishi, Seiya Okubo and Tetsuro Nishino: Strengthening methods of computer Daihinmin programs, Proceedings of CAINE 2017, ISCA, pp.229-236, 2017.10.
[5] 土橋 康希: コンピュータ大貧民における階級制度の効果に関する研究, 電気通信大学総合情報学科卒業論文 (2017)
[6] UECda 標準ルール : http://www.tnlab.inf.uec.ac.jp/daihinmin/2018/document_rules.html (2019-02-13)
[7] 小谷善行: ゲーム計算メカニズム, コロナ社 pp.149-154 (2010)
[8] P.Auer, N.Cesa-Bianchi, and P.Fischer: Finite time Analysis of the Multiarmed Bandit Problem, Machine Learning, vol.47, pp235-256, 2002
[9] D.Silver, and G.Tesauro. Monte-Carlo simulation balancing. 26th International Conference on Machine Learning (ICML), 945-952 (2009)