カードゲーム「アコーディオン」の先読みに関する考察

新谷　敏朗*

Some Consideration on Look Ahead in the Card Game “Accordion”

Toshio SHINTANI*

ABSTRACT

“Accordion” is a solitaire we play with one deck. The game starts with a row of 52 cards. All cards are dealt with face up. A card may be moved upon its left-hand neighbor or upon the card third to its left, if the two cards concerned are of same suit or rank. The theoretical rate of success is nearly 100%, but it is very difficult for human player to solve the game. In this paper, I made some consideration about the property of the game. Then I used the simple brute-force method and “look ahead” to solve the game. I calculated what the success rate would be when the remaining card number(R=1, 2, ... , 50, 51) at the time of starting the “look ahead”. As a result, I clarified that the success rate is very low while R is small, but the success rate rises rapidly as R increases, and when R is larger than 40, it is asymptotic to 100%. In addition, I proposed the hypothesis that “If the order of cards is random in the initial state, it is almost 100% able to succeed”.

キーワード: トランプ, ひとり遊び, アコーディオン, 力まかせ, 先読み
Key words: Card game, Solitaire, Accordion, Brute-force, look ahead

1. まえがき

「アコーディオン」はトランプの一人遊びゲームのひとつである．トランプ一組 52 枚のカードをよくシャッフルして，すべて表向きに 1 列に並べた状態から始めることができるので，完全情報ゲームとして扱うことができる．このゲームは人間がプレイした場合，成功することが非常に難しいとされている.[1] ただし文献[2]では，「成功率は極端に低くもないように思えます．根気のよいパズリストに向いているゲームといえるでしょうか.」とされていて，難易度については，「成功率は 10 回に 1 回かそれ以下の難易度 C」よりもさらに難しい「特 C」に分類されている．しかし筆者の計算機実験[3]によると，ほぼ 100%成功するという結果も得ている．つまり，先読みが得意なコンピューターにとっては，それほど難しくはないが，「先読みそれほど得意ではない人間のプレイヤーにとっては難しい」というゲームであると言ってよい．ここでは，最初は，可能な着手を単純に実行していく，「力まかせ」の解き方でプレイし，途中から先読みを行った場合の成功率を，C++プログラムによる計算機実験によって求めた．その結果から，先読みを始める時点の残りカード枚数によって，成功率がどう変化するのか，また，人間がプレイする際に，どの時点から先読みをするのがよいのかなどの考察を行う．

*情報工学科

2. ルールと簡単な性質

以下ではトランプのスートを H,D,S,C と表し，ランク（数値）を A，2，3，… ，9，0，J，Q，K と表記する．ただし, 10 は桁数を 1 に揃えたほうが状態や計算結果を表示する際に見やすいので，0 と表記している．

2.1 ルール

アコーディオンのルールは文献[2]によると以下のとおりである．スートが同じであるか，数字が同じである 2 枚のカードを「合う」カード(match)と呼ぶ．

(1) 1 組 52 枚のカードをシャッフルして, すべて表向きにして 1 列に並べる．

(2) あるカード X が，左隣りのカード，あるいは 2 枚おいて左のカード Y と「合う」ときは，Y を捨てることができる．捨てたカード Y の場所には X を置き，X があった場所より右にあったカードをすべて左に移動して詰める．これらの操作を「move」と呼ぶことにする．

(3) 操作 move を続けていき, 1 枚だけ残った状態になれば,「成功」である．「成功」でない状態で「合う」カードが全くなくなれば,「失敗」である．

操作(1)では，シャフルしたカードを手札として，任意の枚数ごとに表向けていってもよい．「アコーディオン」という名称は，表向きにするカード数の任意性による．つまり, 並べたカードの枚数がゲームの進行過程で「(アコーディオンのように) 伸び縮みする」ことに由来する．初期状態として, 52 枚のうち一部のみ表向きにしてプレイする場合は, 完全情報ゲームではなくなる．ここでは，最初からすべてのカードを表向きにした完全情報ゲームとして扱う．

S7 S4 S0 D3 S3 C9 S9 SA CQ H3 SJ DA C3
H7 CK C4 C0 D7 H4 H5 H9 S6 D9 DJ D6 H2
CJ S8 H0 C7 HQ C2 SQ C8 D2 DQ D4 C6 HJ
D0 S2 H8 DK HK SK H6 S5 D8 HA CA D5 C5

図 1　初期状態の例

Fig.1 Example of the initial states

本来は一列に並べるが, スペースの都合上図 1 では, 13 枚ずつ 4 列にしている．この例であれば，**S7** と **S4** は隣り合っていて, スートがどちらもスペードなので, **D3** と **S3** はランクがどちらも 3 なので, それぞれ,「合う」カードである． また，**S3 C9 S9 SA** という並びの **S3** と **SA** は 2 枚置いて「合う」カードなので，**SA** を **S3** の場所に移動し，**S3** は捨てて，**SA** があった場所は **CQ** から右をすべて左に詰めて,

… S9 SA C9 S9 CQ H3 …

とすることができる．

2.2 性質

アコーディオンのプレイ中に，操作 move を行うと，捨てられたカードを除いた残りカードの枚数 R は 1 だけ減る．従って， move を行った回数を手数とすると，次の性質が成り立つ．

性質 アコーディオンにおいて，成功する場合の手数は 51 であり，探索アルゴリズムの種類によらない．

通常，深さ優先探索を行うと，最短手数の解は得られないが，このゲームでは幅優先探索やそのほかの最短手数を得られる探索法によらなくてもよいことになる．さらに以下のようなことも考えられる．例えば，次に示す図 2 は初期状態の例であるが，スートは，**S, D, H, C** を繰り返し，ランクは，**A, 3, 2, 4, … , J, 0, Q, K** を繰り返した並びになっているので,「合う」カードが存在しないことは明らかである．

SA D3 H2 C4 S5 D7 H6 C8 S9 DJ H0 CQ SK
DA H3 C2 S4 D5 H3 C6 S8 D9 HJ C0 SQ DK
HA C3 S2 D4 H5 C3 S6 D8 H9 CJ S0 DQ HK
CA S3 D2 H4 C5 S3 D6 H8 C9 SJ D0 HQ CK

図 2　成功不可能な初期状態の例

Fig.2 Example of the initial state unable to succeed

このように成功不可能な初期状態がいくつか考えられるので，アコーディオンの成功率は 100%ではない．しかし，ルールの操作(1)に従って，ランダムにシャッフルした場合に，この例のように，いわば「人為的」に作られた初期状態が生成される確率は非常に低い．このことより，次のような仮説を提唱することができる．

仮説 初期状態が，ある程度ランダムなカードの並びであれば，アコーディオンはほぼ成功可能である

文献[3]では，1 万個の初期状態で計算機実験を行った

結果，すべて成功可能であったが，現在，初期状態の個数を増やして，計算機実験を継続中である．途中経過としては，55 万個以上の初期状態について，すべて成功可能であることがわかっている．従って，上の仮説が成り立つことは大いにあり得ると思われる．本論文では，先読みは行わないで，「合う」カードがあれば，必ず合わせて捨てるという「力まかせ法」の後に全探索による「先読み」を実行する方法を採用した．

3. アルゴリズムとプログラム

今回採用したアルゴリズムは，最初のうちは，初期状態におけるカードの並びを左端から順に見ていき，「合う」カードをルールに従って捨てていく「力まかせ法」によりプレイする．同じカードが「隣」と「2 枚おき」の両方で「合う」カードに該当する場合は，「隣」を優先させた．これは，人間にとっては，すぐ隣にあるカードのほうが，2 枚おいて右にあるカードよりは「合う」のかどうかを判定しやすいからである．そして，残り枚数が R になった時点で，「力まかせ法」ではなく，「先読み」をするように戦略を変更する．

プログラムは，筆者が文献[3]で作成した C++プログラムで，最初の「力まかせ法」による部分は，子節点を 1 個のみ作成するように変更し，「先読み」をする部分は，深さ優先探索によってゲーム木を作成して解を見つけるか，全探索して解がないことを確認するようにしたものを使用した．

4. 計算結果

実際の計算は，主に OS が Windows 10 Education 21H2，CPU が Core i7-6800K で主記憶 128GB のマシンを用いた．一部の初期局面で生成された状態数が 1 億個を超える場合は，OS が Windows 11 Education 21H2，CPU が Xeon W-2223 で主記憶 256GB のマシンで計算した．全体としては，文献[3]で使用したのと同じ初期状態 10000 個に対する計算機実験である．先読みを開始する時点の残りカード枚数 R を 1 から 52 までとして，1 万個の初期状態に対する成功可能性に関する結果は，表 1 と図 3 のようになった．表 1 のデータを図示したものが図 3 である．図 3 を見ると，先読みを開始する時点が遅い(R が小さい)とほとんど成功しないが，R を増加するにつれて，成功可能性が高くなっていき，R>40 で 100%になることがわかる．

表 1　R と成功可能性の関係

Table 1 Relation between R and Feasibility

R	成功	R	成功	R	成功	R	成功
1	0	14	896	27	9121	40	9999
2	18	15	1248	28	9388	41	9999
3	18	16	1684	29	9571	42	10000
4	20	17	2280	30	9705	43	10000
5	34	18	2955	31	9808	44	10000
6	45	19	3800	32	9863	45	10000
7	74	20	4640	33	9918	46	10000
8	101	21	5486	34	9947	47	10000
9	143	22	6360	35	9962	48	10000
10	212	23	7226	36	9979	49	10000
11	315	24	7816	37	9987	50	10000
12	440	25	8398	38	9994	51	10000
13	638	26	8810	39	9998	52	10000

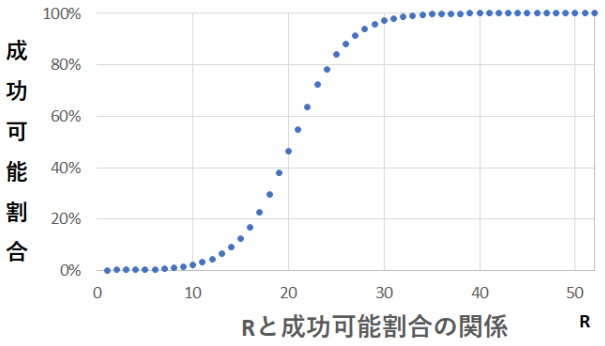

図 3 R と成功可能性の関係

Fig.3 Relation between R and Feasibility

図 3 で「成功」列の数値は 1 万個に対する成功可能な初期状態数なので，10000 が 100%を表す．ここで現れたグラフはソフトウェア工学でいうところの「信頼性成長曲線」に対応していると考えられる．信頼性成長曲線

はバグの累積発見個数が増加する過程を表しているので，ロジスティクス曲線やゴンペルツ曲線に似ているのは当然であると思われる．次に，成功可能な場合に生成された状態数に関する統計値を表 2 と図 4 に示す.

表 2 成功の場合に生成された状態数(1)

Table 3 Number of states in case of success (1)

R	最小値	中央値	平均値	最大値
2	2	2	2	2
3	3	4	4	4
4	5	6	6	8
5	7	9	10	14
6	10	15	17	29
7	12	28	30	60
8	16	46	50	125
9	19	68	81	307
10	27	102	127	519
11	31	156	229	1,298
12	33	264	400	3,292
13	33	471	658	3,866
14	40	714	1,164	19,842
15	48	1,131	1,987	25,381
16	51	1,803	3,316	41,899
17	55	2,876	5,377	103,485
18	66	3,954	8,221	211,424
19	55	5,737	13,014	249,183
20	76	7,923	19,593	1,029,803
21	76	10,258	28,966	1,074,927
22	89	12,496	42,157	2,803,911
23	90	14,954	55,453	4,511,758
24	97	17,119	74,121	14,022,511
25	101	19,766	93,442	18,014,671

表 2 成功の場合に生成された状態数(2)

Table 3 Number of states in case of success (2)

R	最小値	中央値	平均値	最大値
26	138	21,913	106,517	37,267,076
27	138	23,281	112,180	25,682,135
28	115	24,619	126,141	45,761,419
29	118	25,764	136,904	82,203,718
30	136	26,523	232,524	981,513,355
31	126	27,573	129,101	21,874,575
32	127	27,862	139,974	113,919,122
33	112	29,005	136,837	113,919,143
34	114	29,727	128,127	23,756,409
35	124	29,522	129,939	26,523,050
36	185	30,308	147,849	164,044,332
37	180	31,008	141,417	52,740,723
38	196	31,592	132,101	18,225,505
39	209	31,832	138,259	59,318,622
40	222	32,358	139,354	59,318,649
41	237	32,528	141,997	59,318,676
42	307	32,346	138,068	36,942,818
43	226	32,218	137,906	36,942,844
44	299	31,684	134,897	34,585,730
45	343	32,430	140,033	34,585,756
46	260	32,484	141,559	34,585,783
47	276	32,396	134,442	12,389,931
48	296	32,484	130,664	12,107,427
49	312	31,902	130,636	15,120,224
50	320	31,530	135,869	15,120,317
51	342	31,953	132,976	15,120,285
52	365	31,530	135,869	15,120,317

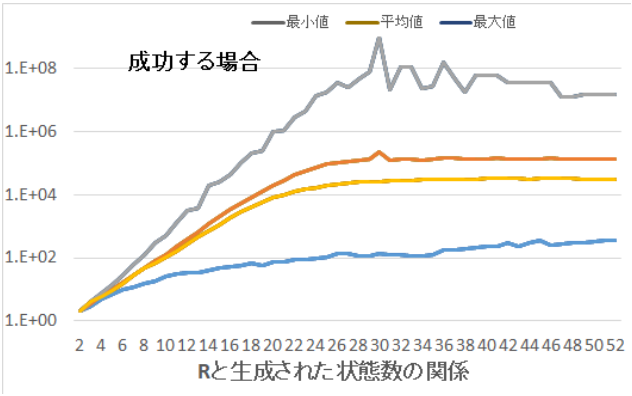

図 4 成功の場合に生成された状態数

Fig.4 Number of states in case of success

表 2 と図 4 で，R=1 の場合は，成功しないので値は存在しない．また，R=30, 32, 33, 36 の場合は，最大値が 1 億を超えたので，上で述べたように別マシンで再計算した値である．これらの値のために最大値の曲線はなめらかにはなっておらず，また平均値が大きくなっているので，中央値に注目する方がよい．

さらに，全探索しても成功状態が存在しなかった「失敗する」場合に生成された状態数に関する統計値を図 5 と表 3 に示す．

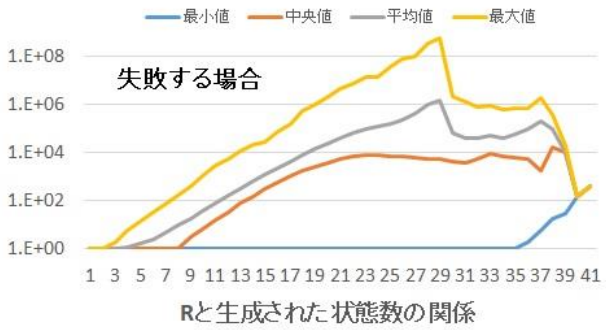

図 5 失敗の場合に生成された状態数

Fig.5 Number of states in case of failure

図 5 と表 3 を見てわかることは，失敗する場合に生成される状態数は，成功する場合と比べるとかなり少ないことである．ただし，R=27, 28, 29 の場合は，最大値が 1 億を超えたので，別マシンで再計算した値である．なお，R>41 では，表 2 からわかるようにすべて成功するので，値はすべて 0 であり，表 3 は R>41 の部分は省略している．失敗の場合は，R>29 の部分のグラフの形がやや不規則な感じに見える．このグラフの解釈はどう考えればよいのかは今後の課題である．なお，生成された状態数の最小値と最大値の差が大きいので，図 4 と図 5 では縦軸を対数目盛にした．

表 3 失敗の場合に生成された状態数(1)

Table3 Number of states in case of failure (1)

R	最小値	中央値	平均値	最大値
1	1	1	1	1
2	1	1	1	1
3	1	1	1	2
4	1	1	1	6
5	1	1	2	14
6	1	1	3	35
7	1	1	5	70
8	1	1	9	177
9	1	3	19	417
10	1	7	39	1,153
11	1	15	79	2,977
12	1	34	163	5,491
13	1	76	327	10,982
14	1	156	649	21,964
15	1	322	1,271	29,323
16	1	603	2,392	77,240
17	1	1,047	4,369	163,446
18	1	1,776	8,038	548,977
19	1	2,624	14,602	1,026,161
20	1	3,731	25,289	2,096,573
21	1	5,154	40,496	4,589,131
22	1	6,795	62,176	7,865,765
23	1	7,684	93,711	13,712,832
24	1	7,628	127,802	13,770,123
25	1	6,691	166,970	36,794,052

表 3 失敗の場合に生成された状態数(2)
Table3 Number of states in case of failure (2)

R	最小値	中央値	平均値	最大値
26	1	6,635	226,822	80,382,678
27	1	6,603	403,625	108,497,563
28	1	5,487	998,625	369,297,531
29	1	5,289	1,560,819	559,863,786
30	1	4,340	67,353	2,126,043
31	1	3,851	41,465	1,329,917
32	1	5,549	41,502	833,772
33	1	8,730	54,455	896,822
34	1	6,707	41,802	604,294
35	1	6,570	59,309	671,851
36	2	5,677	93,452	674,317
37	6	1,838	201,988	1,944,936
38	18	17,808	99,560	363,823
39	29	9,695	9,695	19,360
40	155	155	155	155
41	423	423	423	423

5. 考察

文献[3]では，ゲーム木を作成して全探索すると，ほぼ100%成功することを示し，文献[4]では，単純な「力まかせ法」によってプレイした場合は，成功する割合が非常に低いことが明らかにした．ここでは，両者をミックスした方法を試みた．その結果，ほぼ成功可能であるための閾値を 95%とすると，表 1 から R=29 の時点から先読みを始めればよいことになる．その際の状態数は，表 2 から 25,764（中央値）である．つまり，29 手を読み切るためには，概ね 2 万 5 千個の局面（状態）を検討すればよいことになる．しかし，ゲーム理論でいうところの「枝刈り」を行えば，より少ない局面を読むことで，成功に至ることができる．

もし，成功可能性が 80%弱でよければ，R=24 の時点から先読みを始めればよく，成功する場合，状態数の中央値は 17,119 であり，失敗する場合，状態数の中央値は 7,628 である．さらにいうと，50%弱でもよいことにすれば，R=20 から先読みを始めて，成功する場合の状態数は中央値が 7,923 であり，失敗する場合の状態数は中央値が 3,731 である. 見方を変えて, 状態数を 1000 程度とすると，R=15 であるが，成功可能性は 12%程度しかないことになる．人間がプレイする場合には，これらのことを考慮して，R を決めればよいと考えられる．

6. あとがき

本論文では，トランプの一人遊びである「アコーディオン」について，先読みを始める時点の残りカード枚数 R が 1 から 52 までの値を取る場合に，成功率がどうなるかを計算した．その結果，成功率は R が小さい間はほとんど成功しないが，R が増加するにつれて成功率が急激に上昇して, R=40 を過ぎると 100%に漸近していくことが明らかになった．

また，「初期状態で，カードの並びがランダムであれば, ほぼ 100%成功可能である」という仮説を提唱した．

今後, 人間がプレイする場合に, 着手可能な手のうち, どの手を選んで実行していけばよいのか，つまり，成功するために有効な戦略について考察していきたい．

参考文献

[1] Morehead & Mott-Smith, “The Complete Book of Solitaire and Patience,” Bantam, pp. 390-395, 1966

[2] 野崎，トランプひとり遊び 88 選，朝日選書 416, pp.235-242(1990)

[3] 新谷, カードゲーム「アコーディオン」の成功率, 第 71 回電気・情報関連学会中国支部連合大会, R20-23-01 (2020)

[4] 新谷，カードゲーム「アコーディオン」の力まかせ法による解に関する考察，福山大学工学部紀要 Vol.44, pp.69-72(2020)

[5] 新谷，カードゲーム「アコーディオン」の戦略, 第 72 回電気・情報関連学会中国支部連合大会, R21-23-03 (2021)