記号処理 43－4
（1987. 11. 13）

カードゲーム「計算」の計算機用アルゴリズム

花澤正純
東海大学理学部情報数理学科

カードの一人ゲーム「計算」を行う計算機用アルゴリズムを紹介する。今の所、２０００回の試行で約８５％の成功率を記録している。

An algorithm to play the card solitaire Calculation

Masazumi HANAZAWA
Department of Mathematical Sciences, Tokai University, Hiratuka, 259-12

We present a computer algorithm for the well-known card solitaire "Calculation", with which we have recorded 85% successes in two thousand test plays.

1．序．　本稿はカードの一人ゲーム「計算」の高い完成率を目指す計算機用（を頭に置いた）アルゴリズムを紹介することを目的とする．このアルゴリズムによるプレーでは島内剛一の乱数アルゴリズムでシャッフルした２０００組のカードに対して１７０８（８５．４％）回成功している。アルゴリズムの説明は理解し易いことと追試可能な程度に正確・詳細であるように記述したい。「計算」のプレーの規則は省略する。用語は文献［１］に合わせるようにした。

2．用語　　台札を積む４個の台にはそれぞれ公差 $1 \sim 4 \pmod{13}$ の上昇列を作るが，この公差を速さと呼び v で表す。場札を積む４個の場に $1 \sim 4$ の番号を付けておき，この番号をその場の重さと呼び w で表す。札の数値（$J=11$ $Q=12$，$K=13$）を札の位数と呼び r で表す。採用する戦略は次の原則に従う。
（i）手札を場に出すときは将来どの台に移すかを決めてしまう。
（ii）手札を台に移せるときは（そこに予定されている場札がない限り）移す。
速さ v の台に移す予定の位数 r の札を (v,r) で表す。したがって上の原則に従う限り各場札にはこの順序対 (v,r) が定まっている。(v,r) より先に v の台に乗る札を (v,r) の先行札，後から乗る札を後続札と呼ぶ．例えば札 $(4,5)$ の後続札は $(4,9)$，$(4,K)$ の２枚ある。$(3,8)$ は $(3,4)$ の先行札である。また (v,r) の後続札のうち場に出ているものを (v,r) の後続場札と呼ぶ。特に後続場札のうちで（台 v の札順序の意味で）(v,r) に最も近いものを (v,r) の次候補場札と呼ぶ。

3．基本的アイデア　　上述の原則により場札のトップカードをどの台に移すかは定まっており移せるときは直ちに移す．また手札を台に移すか場に出すかという選択手続きも不要になる。残るのは次の二種の決定手続きである。
（i）手札を移せる台が複数あるときいずれを選ぶか。
（ii）手札を場に出すとき，どの場 w に出すか，また将来どの台 v に乗せるか。
前者の方は成功率に大きな影響を与えないようなので（しかも今の戦略は当面の場札が一番沢山はけるように移す策を大筋としているが最善と思えないので）解説を省略する。後者の戦略を述べる。
まず札 (v,r) に対し距離 d を定義する

$$d = (\text{後続札の枚数}) - (\text{後続場札の枚数})$$

札 (v,r) が場札として出されると，その上に少くとも d 枚の札が乗せられない状況が起きる。その意味で距離 d は札 (v,r) の悪役度を表している。これを用いて試験的な原理「１の場には距離０の札を，２の場には距離１の札を，…，４の場には距離３の札を出す」を考えてみる．
この原理の理想的な例を下に挙げる。

	$w=1$	$w=2$	$w=3$	$w=4$
高さ＝１　…	$(1,K)$	$(2,J)$	$(2,7)$	$(4,A)$
高さ＝２　…	$(1,Q)$	$(1,10)$	$(1,8)$	
高さ＝３　…			$(3,7)$	

図１．　こちら側から積んでいる

この戦法は明らかにすぐ駄目になるが，この考え方を基本的に採用したい。その

ため各場wに対して sd 値（stack degree のつもり）と呼ぶものを次のように定める：

$sd(w) =$ 場w内の札の距離dの最大値

で，戦略の原理として「場wにはwのsd値に等しい（あるいはなるべく近い）距離を持つ札を乗せる」を採用する。すなわち位数rの手札が出たとき，v，wを設定して札(v,w)の距離を求め，場wのsd値と比較し，その一致度が最大になるようにする。残る問題はその一致度の評価の仕方である。例えば，v_1, w_1とv_2, w_2という二つの選択について前者では(v_1, r)の距離が0，場w_1のsd値が0となり，後者では(v_2, r)の距離が2，場w_2のsd値が2のときどちらがよいかの判定法を用意しておく必要があるし，実際これの良し悪しが，（この方式の戦略では）成功率に大きな影響を与える（ようである）。ここで採用した方式は，いずれを選択するかの判定をこの距離dとsd値sdの2種のみで行なうもので他の周囲の状況とかの情報を考慮しないものである。従って予めdとsdの組合せに対する評価値の一覧表を与えておいてその評価の良い組合せを単純に選択するという手段で済ませられる。その一覧表を下に挙げる。

sd ＼ d	0	1	2	3	4	5	6
0	0	121	122	123	124	125	126
1	120	5	65	95	200	250	300
2	120	6	10	40	60	75	90
3	120	8	6	15	58.78	59.04	59.21
4	120	11.77	11.76	11.76	58.75	59.02	59.19
5	120	11.81	11.81	11.81	11.80	59.00	59.18

図2．障害度表（距離dとSd値Sdの組合せのプレーがそれ以後のプレーに与える障害の大きさを評価したもの）

注．本稿の準備に際して初めて気づいたのであるが，上述の表は明らかに（少くとも筆者の意に反して）変な箇所がある。Sd値＝0の欄と1の欄が，$d=3$から4に移る所で障害度が逆転してしまっている。また$d=0$の欄が$Sd=0$の所を除いて同一の数値で区別できていない。しかし，この表は現在使用中のプログラムの中の障害度表を出力させたものであり，今の所そのプログラムによる実行結果しか得ていないので，不備ではあるものの敢えてそのまま載せておく。結果から考えて，この辺りの数値の変動は大きな意味を持たないのかも知れぬ。逆なら面白いが…。$d \geq 7$，$sd \geq 6$については重要度が低いので省略した。原則としてd，sdは大きくなるほど障害度は大に取ってあり，$d \geq sd$の範囲は障害度≥ 50，$d < sd$の範囲では障害度< 12となっている。障害度表はもともとdとsdの組合せの良し悪しの比較に用いるものであるので，表の中の数値の代りにその数値の（小大順の）順位を用いてよい。

4．実際の戦略　実は札(v, r)の距離および場wのsd値の定義に重大な修正を施こして使っている。

(i) Sd値の変更．まず，場wの高さhに位置する札αのsd値$Sd(w,h)$を定義する。まずαの次候補場札をα^*で表し，α^*の属する場をw^*，その高さをh^*とおき，（αの属する台の札順の意味で）αとα^*の間に来る札の枚数を

αとα^*のギャップと呼ぶ．このとき，$Sd(w, h)$は次式によって帰納的に定義される：

$$Sd(w, h) = \max\{Sd(w, h-1),\ Sd(w^*, h^*) + (\alpha と \alpha^* のギャップ)\}$$

但し，$h=1$のときは$Sd(w, h-1)=0$と置き，αの後続場札がない場合は

$$Sd(w, h) = \max\{Sd(w, h-1),\ (\alpha の後続札の枚数)\}$$

と定める。これを用いて，場wのSd値$Sd(w)$を

$$Sd(w) = (場 w のトップカードの Sd 値)$$

と定める。場wが空のときは$Sd(w)=0$と定める。

注．ここで気がついたことであるが，上の定義はまずい定義で，次のようにした方がよいと思われる。まず集合$S(w, h)$を帰納的に次のように定める。

$$S(w, h) = S(w, h-1) \cup S(w^*, h^*) \cup (\alpha と \alpha^* の間に来る札全部)$$

そこで　$Sd(w, h) = \#S(w, h)$　と定義してやる。少くともこちらの方が意味の明確な関数であろう。

(ii) 距離の概念の変更．　さきほど定義した距離dの定義をより精密にする。位数rの手札をどの台vに使うかを決定する判定資料として，各$v=1,2,3,4$について札(v, r)の距離$d(v)$を求める。

$$d(v) = \begin{cases} \infty & \text{----札}(v, r)\text{が使用済の場合} \\ (\text{札}(v, r)^* \text{の Sd 値}) + ((v, r)\text{と}(v, r)^*\text{とのギャップ}) \\ \quad \text{---それ以外の場合で}(v, r)\text{の次候補場札}(v, r)^*\text{がある場合} \\ (v, r)\text{の後続札の枚数} \\ \quad \text{---それ以外の場合} \end{cases}$$

(iii) 前回のSd値，距離の代りに，このSd値と距離を用いて，先程の障害度表に当てはめて，最適なvとwを抽出するわけである。蛇足になるがアルゴリズム全体を要約すると次のようになる。

位数rの手札に対して

(1) 台札に移せるときは移す（場札に札(v, r)があるときは台vに移せるとは認めない）。

そうでない場合には

(2) 前述の手段によって，それぞれの場wについてsd値$sd(w)$を算出し，各$v=1, 2, 3, 4$について

(3) 前述の式によって，札(v, r)の距離$d(v)$を求め，

(4) 前述の障害度表によって，順序対$(sd(w), d(v))$に対する障害度評価を求め，

(5) 最小障害度を与える組合せ(v, w)を選び，

その位数rの手札を，将来vの台へ移す札(v, r)と定め，場wへ乗せる。ただし文献[1]で言う「もつれ状態」を引き起こし，その時点で完成不能が判明する組合せ(v, w)は対象から除く。

5、後記　上述のアルゴリズムではKの行先さえ予め指定しているが，文献[1]のように自由度を残しておいた方がよいはず。その欠点を補うため，本稿の戦術でKを場に出すとき，他の場札の状況を若干参照するような小細工を実際のプログラムに組み込んでいるが，その効果はプラス面マイナス面の両面に現れて全体としては余りプラス効果がなかったと記憶している。障害度の評価表は成

功率に大きな影響を与える。いろいろに変更して（自動変更による学習ルーチンを加えて）みれば（と言っても数値自体でなく，数値によって引き起こされる大小順位のみが重要なので，その意味での変更だけが意味を持つのであるが）まだ進歩が見られるかと思う。また Sd 値の定義，d の定義も本文中で述べたように若干妥当性に欠ける気味があり，それを正した場合の働きに興味が残る。言うまでもないかも知れないが，各場札の行先の台を予め固定しない方式と結びつけた場合の結果にはそれ以上の興味が残る。なお場の重み w の意味づけが文献［１］と反転している…念のため。

文献

［１］ System5，「計算」について（続），数学セミナー　２４巻７号（１９８５）５３－５７