

# 回胴式遊技機の天井到達時における収支の考察

芝浦工業大学 数理科学研究会

平成 25 年 11 月 3 日

※何か不明な点や計算ミス等がありましたら加筆修正しますので指摘をお願いします

制作: 岡田 拓真

## 1 研究背景

パチスロにはある一定の回数の中でボーナス(当たり)が出ないと天井状態と言って、ボーナスが出やすい状態になる機能がついている機種がある。そんな機種を打つ場合は天井状態に近い台を狙って打つ事で勝ちを狙いに行くという立ち回りが考えられる。しかし天井状態まであと何回の状態からなら勝てるのかという線引きがとても難しい。そこで今回数学の力を使い、そのラインを見極めたいと考えた。

## 2 実験機種の仕様

今回の実験に使用する機種は、「新世紀エヴァンゲリオン～約束の時～」(Bisty)とした。選んだ理由は、天井機能を搭載したもので、挙動が単純であるためである。この機種の詳細は以下の通りである。

### 1. 天井状態突入の条件

- BIG 終了後 1000G 消化
- REG 終了後 800G 消化

### 2. 天井状態

この機種の天井状態は RT(コインが少しずつ増えていく状態) への突入確率が上がるというものがある。この機種の RT は暴走モード (1G あたりの純増 0.3 枚) と覚醒モード (1G あたりの純増 0.5 枚) の 2 種類がある。この 2 種類の RT は一旦突入するとボーナスまで継続する。

表 1: 通常状態と天井状態の RT 突入確率

	1	2	3	4	5	6
暴走 (通常)	1/2849.39	1/2849.39	1/2849.39	1/2849.39	1/2849.39	1/2849.39
暴走 (天井)	1/127.25	1/122.5	1/118.08	1/113.98	1/110.14	1/106.56
覚醒 (通常)	1/4369.07	1/4369.07	1/4369.07	1/4369.07	1/4369.07	1/4369.07
覚醒 (天井)	1/885.62	1/885.62	1/885.62	1/885.62	1/885.62	1/885.62

## 3 コンピュータによるシミュレーション

今回はシミュレーションに C を使用した。C は乱数の生成に偏りがでるので、それを少なくするため rand() の値を [0,1) の実数に正規化し、k 倍して整数部をとるようにした。また天井狙いの立ち回りをする時は設定を看破するための情報があまり得られないことを考慮し、設定は毎回ランダムで実行されるようにした。

### 3.1 プログラムの実装

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#define K //回転数
#define M 30//標本数
int setteil(int coin,int joutai,int N)
```

```

{
int i,a,b,k=0;
for(i=0;i<N;i++ )
{
a=(rand()/(RAND_MAX+1.0))*10000;
b=(rand()/(RAND_MAX+1.0))*10000;
if(joutai==0)//通常状態
{
if(a>=0 && a<8) coin+=9;//強スイカ
if(a>=8 && a<18) //1 枚役
{
i=-1; //ループをリセット
coin=0;
}
if(a>=18 && a<153) coin-=2;//チェリー
if(a>=153 && a<294) coin+=9;//弱スイカ
if(a>=294 && a<1515) coin+=7;//ベル
if(a>=1515 && a<2882) coin+=0;//リプレイ
if(a>=2882 && a<2886)//暴走
{
i=-1;
coin=0;
}
if(a>=2886 && a<2888)//覚醒
{
i=-1;
coin=0;
}
if(a>=2888 && a<10000) coin-=3;//ハズレ
}
if(joutai==1)//天井状態
{
i=0;
if(a>=0 && a<8)//強スイカ
{
if(b>=0 && b<1000)
{
printf("赤 7 成立\n");
coin+=409;
return coin;
}
if(b>=1000 && b<2000)
{
printf("青 7 成立\n");
coin+=409;
return coin;
}
if(b>=2000 && b<3000)
{
printf("黄 7 成立\n");
coin+=295;
return coin;
}
if(b>=3000 && b<5000)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=5000 && b<10000) coin+=9;
}
if(a>=8 && a<18) //1 枚役
{
if(b>=0 && b<1971)
{
printf("赤 7 成立\n");
}
}
}
}

```

```

    coin+=409;
    return coin;
}
if(b>=1971 && b<3097)
{
    printf("青7 成立\n");
    coin+=409;
    return coin;
}
if(b>=3097 && b<5210)
{
    printf("黄7 成立\n");
    coin+=295;
    return coin;
}
if(b>=5210 && b<10000)
{
    printf("REG 成立\n");
    coin+=113;
    return coin;
}
}
if(a>=18 && a<153)//チェリー
{
    if(b>=0 && b<247)
    {
        printf("赤7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=247 && b<337)
    {
        printf("青7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=337 && b<674)
    {
        printf("黄7 成立\n");
        coin+=295;
        return coin;
    }
    if(b>=674 && b<1011)
    {
        printf("REG 成立\n");
        coin+=113;
        return coin;
    }
    if(b>=1011 && b<10000) coin-=2;
}
if(a>=153 && a<294)//弱スイカ
{
    if(b>=0 && b<54)
    {
        printf("赤7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=54 && b<86)
    {
        printf("青7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=86 && b<194)
    {

```

```

    printf("黄7 成立\n");
    coin+=295;
    return coin;
}
if(b>=194 && b<302)
{
    printf("REG 成立\n");
    coin+=113;
    return coin;
}
if(b>=302 && b<10000) coin+=9;
}
if(a>=294 && a<1515)//ベル
{
    if(b>=0 && b<1)
    {
        printf("赤7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=1 && b<2)
    {
        printf("青7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=2 && b<3)
    {
        printf("黄7 成立\n");
        coin+=295;
        return coin;
    }
    if(b>=3 && b<4)
    {
        printf("REG 成立\n");
        coin+=113;
        return coin;
    }
    if(b>=4 && b<10000) coin+=7;
}
if(a>=1515 && a<2799)//リプレイ
{
    if(b>=0 && b<1)
    {
        printf("赤7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=1 && b<2)
    {
        printf("青7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=2 && b<3)
    {
        printf("黄7 成立\n");
        coin+=295;
        return coin;
    }
    if(b>=3 && b<4)
    {
        printf("REG 成立\n");
        coin+=113;
        return coin;
    }
}

```

```

    if(b>=4 && b<10000) coin+=0;
}
if(a>=2799 && a<2878)//暴走
{
    printf("暴走\n");
    joutai=2;
}
if(a>=2878 && a<2889)//覚醒
{
    printf("覚醒\n");
    joutai=3;
}
if(a>=2889 && a<10000) coin-=3;//ハズレ
}
if(joutai==2)//暴走モード
{
    i=0;
    if(a>=0 && a<8)//強スイカ
    {
        if(b>=0 && b<1000)
        {
            printf("赤7 成立\n");
            coin+=409;
            return coin;
        }
        if(b>=1000 && b<2000)
        {
            printf("青7 成立\n");
            coin+=409;
        }
    }
return coin;
}
if(b>=2000 && b<3000)
{
    printf("黄7 成立\n");
    coin+=295;
    return coin;
}
if(b>=3000 && b<5000)
{
    printf("REG 成立\n");
    coin+=113;
    return coin;
}
if(b>=5000 && b<10000) coin+=9;
}
if(a>=8 && a<18) //1 枚役
{
    if(b>=0 && b<1971)
    {
        printf("赤7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=1971 && b<3097)
    {
        printf("青7 成立\n");
        coin+=409;
        return coin;
    }
    if(b>=3097 && b<5210)
    {
        printf("黄7 成立\n");
        coin+=295;
        return coin;
    }
}
if(b>=5210 && b<10000)

```

```

{
printf("REG 成立\n");
coin+=113;
return coin;
}
}
if(a>=18 && a<153)//チェリー
{
if(b>=0 && b<247)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=247 && b<337)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=337 && b<674)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=674 && b<1011)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=1011 && b<10000) coin-=2;
}
if(a>=153 && a<294)//弱スイカ
{
if(b>=0 && b<54)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=54 && b<86)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=86 && b<194)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=194 && b<302)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=302 && b<10000) coin+=9;
}
if(a>=294 && a<1515)//ベル
{
if(b>=0 && b<1)
{

```

```

printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=1 && b<2)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=2 && b<3)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=3 && b<4)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=4 && b<10000) coin+=7;
}
if(a>=1515 && a<7328)//リプレイ
{
if(b>=0 && b<1)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=1 && b<2)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=2 && b<3)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=3 && b<4)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=4 && b<10000) coin+=0;
}
if(a>=7328 && a<7414) //覚醒
{
printf("覚醒\n");
joutai=3;
}
if(a>=7414 && a<10000) coin-=3;//ハズレ
}
if(joutai==3)//覚醒モード
{
i=0;
if(a>=0 && a<8)//強スイカ
{
if(b>=0 && b<1000)
{

```



```

printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=1000 && b<2000)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=2000 && b<3000)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=3000 && b<5000)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=5000 && b<10000) coin+=9;
}
if(a>=8 && a<18) //1 枚役
{
if(b>=0 && b<1971)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=1971 && b<3097)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=3097 && b<5210)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=5210 && b<10000)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
}
if(a>=18 && a<153)//チエリー
{
if(b>=0 && b<247)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=247 && b<337)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=337 && b<674)

```

```

{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=674 && b<1011)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=1011 && b<10000) coin-=2;
}
if(a>=153 && a<294)//弱スイカ
{
if(b>=0 && b<54)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=54 && b<86)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=86 && b<194)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=194 && b<302)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=302 && b<10000) coin+=9;
}
if(a>=294 && a<1515)//ベル
{
if(b>=0 && b<1)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=1 && b<2)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=2 && b<3)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=3 && b<4)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
}

```

```

}
if(b>=4 && b<10000) coin+=7;
}
if(a>=1515 && a<8093) coin+=0;//リプレイ
{
if(b>=0 && b<1)
{
printf("赤7 成立\n");
coin+=409;
return coin;
}
if(b>=1 && b<2)
{
printf("青7 成立\n");
coin+=409;
return coin;
}
if(b>=2 && b<3)
{
printf("黄7 成立\n");
coin+=295;
return coin;
}
if(b>=3 && b<4)
{
printf("REG 成立\n");
coin+=113;
return coin;
}
if(b>=4 && b<10000) coin+=0;
}
if(a>=8093 && a<10000) coin-=3; //ハズレ
}
//printf("%d,%d,%d\n", coin,i,k);
}
return coin;
}

int main()
{
int i,c,N,coin=0;
srand((unsigned)time(NULL));
N=K;
for(i=0;i<M;i++)
{
c=(rand()/(RAND_MAX+1.0))*6;
coin=0;
if(c==0)
{
printf("設定 1\n");
coin=settei1(coin,0,N);
coin=settei1(coin,1,N);
printf("%d\n", coin);
}
if(c==1)
{
printf("設定 2\n");
coin=settei2(coin,0,N);
coin=settei2(coin,1,N);
printf("%d\n", coin);
}
if(c==2)
{
printf("設定 3\n");
coin=settei3(coin,0,N);

```

```

coin=settei3(coin,1,N);
printf("%d\n",coin);
}
if(c==3)
{
printf("設定 4\n");
coin=settei4(coin,0,N);
coin=settei4(coin,1,N);
printf("%d\n",coin);
}
if(c==4)
{
printf("設定 5\n");
coin=settei5(coin,0,N);
coin=settei5(coin,1,N);
printf("%d\n",coin);
}
if(c==5)
{
printf("設定 6\n");
coin=settei6(coin,0,N);
coin=settei6(coin,1,N);
printf("%d\n",coin);
}
}
}
}

```

このプログラムは、天井状態まで  $N$  回転の状態から、ボーナスを引くまでに必要なコインの枚数を  $M$  回シミュレートするものである。上記のプログラムには設定 1 の数値しか入力されていないが、実際には設定 2~6 の数値も入力しシミュレーションを行った。

## 4 母平均の区間推定

今回はシミュレーションによって得られたは標本から母平均の区間推定を行う。

中心極限定理

$X_1, X_2, \dots, X_n$  を互いに独立で同じ確率分布に従う確率変数列とし、その平均を  $\mu$ 、分散を  $\sigma^2$ 、標本平均を  $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$  とすると、

$$Z_n = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

の確率分布は、 $n \rightarrow \infty$  のとき標準正規分布  $N(0, 1)$  に収束する。

この中心極限定理より、大標本 ( $n \geq 30$ ) のとき任意の母集団分布に対して母平均の区間推定を行うことができる。また母分散が未知の場合でも母分散  $\sigma^2$  を不偏分散  $u^2$  で十分に近似できると考えることができる。従って、母平均の信頼度  $100(1 - \alpha)\%$  の信頼区間は、

$$\bar{x} - z \left( \frac{\alpha}{2} \right) \frac{u}{\sqrt{n}} < \mu < \bar{x} + z \left( \frac{\alpha}{2} \right) \frac{u}{\sqrt{n}}$$

となる。

今回は信頼度  $90\% (\alpha = 0.1)$  とし、この区間が負の値を含まない部分を勝ちが見込める範囲とした。

### 4.1 プログラムの実装

```
{
```

```

int i,j,a[M],N;
double right,left,heikin,bunsan;
srand((unsigned)time(NULL));
for(j=0;j<=1;j++)
{

    /////前項 main 関数略/////

    for(i=0;i<M;i++)
    {
        printf("%d\n",a[i]);
        bunsan=bunsan+a[i]*a[i];
        heikin=heikin+a[i];
    }
    heikin=heikin/M;
    bunsan=(M/(M-1))*((bunsan/M)-(heikin*heikin));
    left=heikin-1.645*(sqrt(bunsan)/sqrt(M));
    right=heikin+1.645*(sqrt(bunsan)/sqrt(M));
    if(left<0)
    {
        j=0;
        N=N-1;
    }
    printf("%lf<母平均<%lf\n",left,right);
}

printf("%d 回前からなら勝てます.\n",N);
}

```

このプログラムは与えられた配列から、母平均を区間推定するプログラムである。left が負の値をとった場合回転数を-1 して再び区間推定をするようになっている。実際には先程のプログラムの main 関数の中で実行する。

## 4.2 プログラムの実行

表 2: 天井までの回転数と母平均の区間

回転数	左側	右側
177	-16.206100	0.276257
178	-10.160531	64.441233
179	-3.996182	70.018989
180	-45.567051	27.567279
181	-0.660443	79.660445
182	-6.564577	58.554577
183	-63.520581	9.340481
184	-27.855520	62.773719
185	-32.301322	38.890504
186	12.1471	109.218701

プログラムを実行した結果、母平均の区間が負の値を含まないのは天井までの回転数が 186 回転からという結果がでた。

## 5 考察

今回得られた結果は天井まで 186 回転の状態からなら勝てるという結果だった. しかしこのシチュエーションに出会うこと極稀であり, 実践的な結果ではない. またこの 186 回転という数字もプログラムを実行する度に変わるのであまり信頼性のある数字とは言えない

## 6 今後の課題

今回は単純化のために天井に突入してから当たりを引くというようにとても状況を限定して進めた. 今後はもっと一般的な状況について考察したり, 設定判別の考察をしたりしたい. もっと深くスロットの解析を進め, 必勝法を模索していきたい.

## 参考文献

- [1] 穴太克則, 講義:確率・統計, 学術図書出版
- [2] <http://slot-777.net/kaiseki/evangelion3.html>