

分割数と配分の問題

SHIMURA Masato
JCD02773@nifty.ne.jp

2010年6月28日

目次

1	分割数	1
1.1	分割数と組み合わせ	3
1.2	4個の蜜柑の配分/経過と解説	3
2	分割数と組み合わせの応用例	5
2.1	閉鎖待ち行列ネットワーク	5
2.2	修理連鎖 $M/M/\infty$	8
3	Reference	10
付録 A	Script	10

概要

分割数と順列を組み合わせると配分の問題や閉鎖待ち行列に応用できる。

Partition number and table of permutation make a pattern of distribution.

We can consider a simple model of Wealth distribution.

1 分割数

- 4個の蜜柑を分ける方法。数学的アプローチ

```
partition 4
+-----+-----+-----+-----+
|1 1 1 1|2 1 1|2 2|3 1|4|
+-----+-----+-----+-----+
```

- 組み合わせのテーブル

```

tap 3
0 1 2
0 2 1
1 0 2
1 2 0
2 0 1
2 1 0

```

```

tap=:i.@! A. i. NB. Table of all permutation (J Vocaburaly)
NB. A. anagram index

```

- 4つの蜜柑を A,B,C,D の4人で分ける方法。社会科学的アプローチで富の配分と考えればこのような小さな例でも 35 通りある。椅子取りゲームではなく配分は自由であるとする。2個取る人が出れば0が生じる。各人は $\frac{15}{35} = 42\%$ の割合で配分に預かれない。

```

classify0 part_tap 4 4

A B C D
+-----+-----+-----+-----+-----+
|4 0 0 0|3 1 0 0|2 2 0 0|1 3 0 0|0 4 0 0|
|      |3 0 1 0|2 1 1 0|1 2 1 0|0 3 1 0|
|      |3 0 0 1|2 1 0 1|1 2 0 1|0 3 0 1|
|      |      |2 0 2 0|1 1 2 0|0 2 2 0|
|      |      |2 0 1 1|1 1 1 1|0 2 1 1|
|      |      |2 0 0 2|1 1 0 2|0 2 0 2|
|      |      |      |1 0 3 0|0 1 3 0|
|      |      |      |1 0 2 1|0 1 2 1|
|      |      |      |1 0 1 2|0 1 1 2|
|      |      |      |1 0 0 3|0 1 0 3|
|      |      |      |      |0 0 4 0|
|      |      |      |      |0 0 3 1|
|      |      |      |      |0 0 2 2|
|      |      |      |      |0 0 1 3|
|      |      |      |      |0 0 0 4|
+-----+-----+-----+-----+-----+

```

分割数と組み合わせを用いる

- 4個の蜜柑を3人で分ける方法(15通り)

```

classify0 part_tap 4 3
+-----+-----+-----+-----+-----+
|4 0 0|3 1 0|2 2 0|1 3 0|0 4 0|
|      |3 0 1|2 1 1|1 2 1|0 3 1|
|      |      |2 0 2|1 1 2|0 2 2|
|      |      |      |1 0 3|0 1 3|
|      |      |      |      |0 0 4|
+-----+-----+-----+-----+-----+
1 2 3 NB. device

```

1.1 分割数と組み合わせ

分割数と組合せた配分の組み合わせ問題は重複を整理すると次の表のようになる。組合せ解は $n!$ で急激に大きくなり、配分の組合せは $n = 10$ で所謂計算問題の領域に入り、メモリ不足か解を得られない。

n	分割数	組合せ	配分の組合せ
2	2	2	2
3	3	6	10
4	5	24	35
5	7	120	126
6	11	720	462
7	15	5040	1716
8	22	40320	6435
9	30	362880	24310
10	42	3628800	-

1.2 4個の蜜柑の配分/経過と解説

1. 4個の蜜柑の分割数を求める

```

] p=. partition 4
+-----+-----+-----+-----+
|1 1 1 1|2 1 1|2 2|3 1|4|
+-----+-----+-----+-----+

```

2. 分割数の box 内の個数を求める

```

;#L:0 p
4 3 2 2 1

```

3. (4個=1 1 1 1)を検出する index

```
3>:/ ;#L:0 p
0 1 1 1 1
```

4. (1 1 1 1)を落として rotate(.)して開く(余白に0が入る)

```
tmp=.|. ;("1),. (3>:/ ;#L:0 p)#p
4 0 0
3 1 0
2 2 0
2 1 1
```

5. 3の組み合わせを求める

```
ind=: { tap 3
+-----+-----+-----+-----+-----+-----+
|0 1 2|0 2 1|1 0 2|1 2 0|2 0 1|2 1 0|
+-----+-----+-----+-----+-----+-----+
```

6. 4 0 0を ind により組み合わせる

```
ind { (L:0) 0{ tmp
+-----+-----+-----+-----+-----+-----+
|4 0 0|4 0 0|0 4 0|0 0 4|0 4 0|0 0 4|
+-----+-----+-----+-----+-----+-----+
```

7. tmpの4行に同じ計算を施して重複を整理する。(24 → 16)

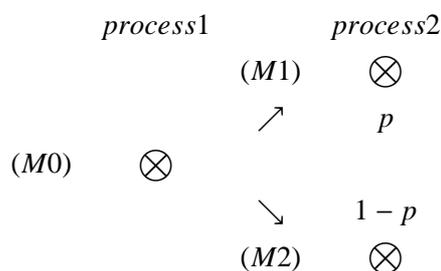
```
classify0 part_tap 4 3
+-----+-----+-----+-----+-----+
|4 0 0|3 1 0|2 2 0|1 3 0|0 4 0|
|    |3 0 1|2 1 1|1 2 1|0 3 1|
|    |    |2 0 2|1 1 2|0 2 2|
|    |    |    |1 0 3|0 1 3|
|    |    |    |    |0 0 4|
+-----+-----+-----+-----+-----+
```

2 分割数と組み合わせの応用例

閉鎖待ち行列は兎に角パレットや車両がエンドレスに（機械を止めるまで）駆けめぐるケースであるが分割数と組み合わせを応用できる。

2.1 閉鎖待ち行列ネットワーク

製造モデル



ある工場のプロセスと機械。工程は2のプロセスでパレットに一個の部品（一個限定）を載せてM0のプロセスを終えるとパレットは確率 p でM1, 確率 $1-p$ でM2に配分される。部品が2つのプロセスを終えるとパレットは回収され、また新しい部品を載せてM0の前に並ぶ。パレットの数は限られており、有限。

パレットの工程は閉じられた待ち行列で定常分布が存在する。

サービスパラメータ μ を

$$\mu_1 = \frac{1}{8}, \mu_2 = \frac{1}{9}, \mu_3 = \frac{1}{12} \quad \text{とする。}$$

移動を表す行列は次のようになる。

$$p(i, j) = \begin{bmatrix} 0 & 2/3 & 1/3 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

(Dullet EX7.1, EX7.4)

* 定常分布を求める。

$$\pi_1 = c, \rightarrow, \pi_2 = \frac{2}{3}c, \pi_3 = \frac{1}{3}c$$

$$c = \frac{1}{2}$$

* 待ち行列は1列なので

$$\psi_i(n) = \prod_{m=1}^n \varphi_1(m) = \mu_i^n$$

$$n_1 + n_2 + n_3 = N \quad \text{のとき}$$

$$\pi(n_1, n_2, n_3) = c_N \prod_{i=1}^3 \left(\frac{\pi_i}{\mu_i} \right)^{n_i} = c_N 4^{n_1} 3^{n_2} 2^{n_3}$$

それ以外は0

* 確率の和が1になるように c_N を決める。

$N = 4$ の場合。

$$n_i \geq 0$$

$n_1 + n_2 + n_3 = 4$ を満たす数 (n_1, n_2, n_3) から $4^{n_1} 3^{n_2} 2^{n_3}$ を求める。

ここで分割数 (partition number) と J の組み合わせイディオム tap を用いる。

* $N=4$ とする。

* $N=4$ 人が装置 M0, M1, M2 に並ぶパターン。分割数と J の tap (組み合わせ) を統合する。行列のパターンが現れる。

* 各パターンでの詳細釣り合い条件から計算した生起確率

経過と解説

1. part_tap

```
5{. part_tap 4 3
4 0 0
3 1 0
3 0 1
2 2 0
2 1 1
```

2. 4 3 2 ^ part_tap

```
4 3 2 ^ L:0 { 5{.part_tap 4 3
+-----+-----+-----+-----+-----+
|256 1 1|64 3 1|64 1 2|16 9 1|16 3 2|
+-----+-----+-----+-----+-----+
```

3. box 内を掛け合わせる

```
*/ (L:0) 4 3 2 ^ L:0 { 5{. part_tap 4 3
256 192 128 144 96
```

4. 1351 とおりのパターンがある

```
; */ (L:0) 4 3 2 ^ L:0 {part_tap 4 3
256 192 128 144 96 64 108 72 48 32 81 54 36 24 16
```

```
classify_que_74 4 3
```

```
+-----+-----+-----+-----+-----+
|4 0 0 256|3 1 0 192|2 2 0 144|1 3 0 108|0 4 0 81|
|          |3 0 1 128|2 1 1 96|1 2 1 72|0 3 1 54|
|          |          |2 0 2 64|1 1 2 48|0 2 2 36|
|          |          |          |1 0 3 32|0 1 3 24|
|          |          |          |          |0 0 4 16|
+-----+-----+-----+-----+-----+
```

* M0,M1,M2 で生じる待ち行列とパターン、頻度

```
classify_order_74 4 3
```

```
M0M1M2 度数
```

```
+-----+-----+-----+-----+-----+
|4 0 0 256|3 1 0 192|2 2 0 144|1 3 0 108|0 4 0 81 |
|          |3 0 1 128|2 1 1 96|1 2 1 72|0 3 1 54 |
|          |          |2 0 2 64|1 1 2 48|0 2 2 36 |NB. M0
|          |          |          |1 0 3 32|0 1 3 24 |
```

				0 0 4 16	
0 4 0 81	1 3 0 108	2 2 0 144	3 1 0 192	4 0 0 256	
	0 3 1 54	1 2 1 72	2 1 1 96	3 0 1 128	
		0 2 2 36	1 1 2 48	2 0 2 64	NB. M1
			0 1 3 24	1 0 3 32	
				0 0 4 16	
0 0 4 16	1 0 3 32	2 0 2 64	3 0 1 128	4 0 0 256	
	0 1 3 24	1 1 2 48	2 1 1 96	3 1 0 192	
		0 2 2 36	1 2 1 72	2 2 0 144	NB. M2
			0 3 1 54	1 3 0 108	
				0 4 0 81	

待ち行列の解析結果

	4p	3p	2p	1p	0
待ち行列 M0	256	320	304	260	211
待ち行列 M1	81	162	252	360	496
待ち行列 M2	16	56	148	350	781

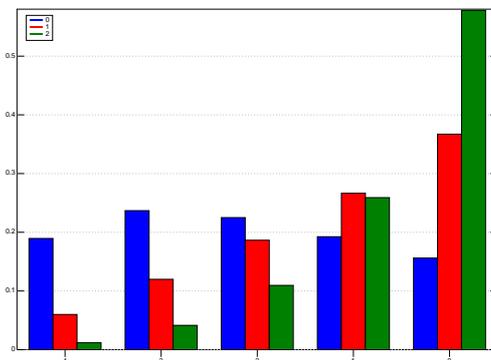


図1 工場連鎖

```
;( "1) order_que_74 4
```

256	320	304	260	211
81	162	252	360	496
16	56	148	350	781

```
0.189489 0.236862 0.225019 0.19245 0.156181
0.0599556 0.119911 0.186528 0.266469 0.367135
0.0118431 0.0414508 0.109548 0.259067 0.57809
```

2.2 修理連鎖 $M/M/\infty$

エンジンまたはタイヤの修理が必要なトラックの
修理

(Dullet EX7.2 Ex7.5)

1 週間あたりの修理パラメータ

故障パラメータ: $\mu_1 = 1$

修理パラメータ: $\mu_2 = 2, \mu_3 = 4$

修理を表す行列

$$p(i, j) = \begin{bmatrix} 0 & 1/4 & 3/4 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

確率 $\frac{1}{4}$ でエンジンの修理、確率 $\frac{3}{4}$ でタイヤの交換が必要である。

1. 定常分布を求める。

$$\pi_1 = c, \rightarrow, \pi_2 = \frac{1}{4}c, \pi_3 = \frac{3}{4}c$$

$$c = \frac{1}{2}$$

2. 最初の待ち行列 ($\mu = 1$) は 1 人あたりの客に対するサービスパラメータ 1 の $M/M/\infty$ 待ち行列に相応する

$$\psi_i(n) = \prod_{m=1}^n \varphi_1(m) = n!$$

- 2 番目と 3 番目の待ち行列 ($\mu = 2, \mu = 4$) はサービスパラメータ 2 と 4 で 1 個の窓口の待ち行列

$$\psi_2(n) = 2^n$$

$$\psi_3(n) = 4^n$$

$$n_1 + n_2 + n_3 = N$$

のとき

$$\begin{aligned} \pi(n_1, n_2, n_3) &= c_N \frac{(1/2)^{n_1} (1/8)^{n_2} (3/8)^{n_3}}{n_1! 2^{n_2} 4^{n_3}} \\ &= c_N \frac{N!}{n_1!} 16^{n_1} 2^{n_2} 3^{n_3} \end{aligned}$$

3. 確率の和が 1 になるように c_N を決める。

$N = 3$ とした場合。

$$\begin{cases} n_i & \geq 0 \\ n_1 + n_2 + n_3 = 3 \end{cases} \text{ を満たす数 } (n_1, n_2, n_3)$$

から $\frac{N! 16^{n_1} 2^{n_2} 3^{n_3}}{n_1!}$ を求める。

ここでも分割数と J の組み合わせイディオム Δtap を用いる。

1. $N=3$ とする
2. 待ち行列の総数は 10150 となる。
3. `calc_que_75 3 3`

```

3 0 0 4096
2 1 0 1536
2 0 1 2304
1 2 0 384
1 1 1 576

```

```

1 0 2 864
0 3 0 48
0 2 1 72
0 1 2 108
0 0 3 162

```

4. order_que_75 3 3
 0.403547 0.378325 0.179704 0.0384236

経過と解説

1. 3個の蜜柑を3人で配分(10通り) part_tap 3 3

```

5{. part_tap 3 3
3 0 0
2 1 0
2 0 1
1 2 0
1 1 1

```

2. $16^{n_1} 2^{n_2} 3^{n_3}$ を求める。

```

] a=. 16 2 3 ^ L:0 { 5{. part_tap 3 3
+-----+-----+-----+-----+
|4096 1 1|256 2 1|256 1 3|16 4 1|16 2 3|
+-----+-----+-----+-----+

```

3. $\frac{N!}{n_1!}$ を求める

```

] b=. (!3)%! 5{. {"1 part_tap 3 3
1 3 3 6 6

```

4. 上の2をかける

```

b* ;*/L:0 a
4096 1536 2304 384 576

```

故障したトラックの台数 $\begin{bmatrix} \text{台} & 0 & 1 & 2 & 3 \\ p = & 0.403547 & 0.378325 & 0.179704 & 0.0384236 \end{bmatrix}$

3 Reference

R. デュレット 今野 中村 曾雌 馬 訳「確率過程の基礎」Springer 2005

Miscellance

Download J language

<http://www.jsoftware.com>

Scrip Library

<http://japla.sakura.ne.jp>

付録 A Script

```
tap=: i.@! A. i. NB. J Vocaburaly A.
```

```
part_tap=: 3 : 0
```

```
NB. use partition number & tap
```

```
'N0 S0'= . y NB. N and column
```

```
NB. Usage: u 4 3/u 3 3
```

```
TMP0=: |. ;("1) ,.(S0 >:/;# L:0 tmp )# tmp=. partition N0
```

```
IND=: { tap S0
```

```
ANS=. <''
```

```
for_ctr. i. # TMP0 do.
```

```
TMP1=. {~. ;("1) ,. IND{ L:0 ctr { TMP0
```

```
ANS=. ANS,TMP1
```

```
end.
```

```
\:~ ;("1) ,. }.ANS NB. down sort
```

```
)
```

```
NB. from partition_shimura.ijs
```

```
partition=: 3 : 0
```

```
NB. partition number
```

```
NB. Usage: e.g. partition 10
```

```
ANS=. (<<1),(<1 1;2) NB. 0,1
```

```
if. y<3 do. ANS=. ;("1),. ANS goto_skip. end.
```

```
NR0=: }. >: i. y NB. first is 2
```

```

NB. -----
for_ctr. i. <:# NR0 do. NB. number exam
  TMP2=. ((>:ctr) {. NR0) calc_part_sub ANS
  ANS=. ANS,< (<tmp # 1),(TMP2),< tmp=.>: ctr { NR0
end.
NB. ANS=. ; 10 #. L:0 |. {: ;("1),. ANS NB. change on 10 sin-suu
ANS=. {: ;("1),. ANS NB. change on 10 sin-suu
label_skip.
ANS
)

calc_part_sub=: 4 : 0
RW=. |. i. # NR=. x NB. RAW for pickup e.g. 5->2 3 4
ANS2=. <'
  for_ctr. i. # NR do. NB. partial of number exam
    Y0=. (ctr{RW) { y
    if. 0=*/ (ctr{NR) check_major Y0
      do. Y0=. ((ctr{NR) check_major Y0) # L:1 Y0 end.
    TMP1=. >(< ctr{NR) , L:0 Y0 NB. connect with ,
    ANS2=. ANS2,<TMP1
  end.
  TMP=. ;}. ANS2
)

check_major=: 4 : ' ;*/(L:0) x >:/ L:0 ; y '

```