情報処理論2a

7章「待ち行列シミュレーション」

中島康彦

§7. 1 今日の作業ディレクトリを作る

1. % cd ⇒ ホームディレクトリへ移動
2. % mkdir chap19 ⇒ ディレクトリchap19を作成
3. % cd chap19 ⇒ ディレクトリchap19へ移動
4. % netscapeを使ってdata19をchap19へダウンロード
5. % tar xvf data19 ⇒ サンプルデータの複写

queue.c

§7. 2 時刻0～mにちょうどx人が到着する確率

時刻0～nに，互いに影響しないλn人がランダムに到着するとき，

- ある1人が時刻0～mに到着する確率: m/n
- ある1人が時刻m～nに到着する確率: $1-m/n$

時刻0～mに，ちょうどx人が到着する確率は，以下を全て乗じたものとして，求めることができる.

- λn人から，x人を取り出す組み合わせの数: ${}_{\lambda n}C_x$
- x人が時刻0～mに到着する確率: $(m/n)^x$
- λn−x人が時刻m～nに到着する確率: $(1-m/n)^{\lambda n-x}$

§7. 2 時刻0～mにちょうどx人が到着する確率(続き)

$$ {}_{\lambda n}C_x (m/n)^x (1-m/n)^{\lambda n-x} \quad \text{ただし } 0 \leqq x \leqq \lambda n $$

$$ = \frac{\lambda n(\lambda n-1)\cdots(\lambda n-x+1)}{x!} (m/n)^x (1-m/n)^{\lambda n-x} $$

$$ = \frac{1}{x!} m^x \lambda(\lambda-1/n)\cdots(\lambda-(x-1)/n)(1-m/n)^{\lambda n}(1-m/n)^{-x} $$

- n→∞とすると

$(1-m/n)^{\lambda n} \Rightarrow e^{-\lambda m}$ であるので，

$$ \Rightarrow \frac{1}{x!}(m\lambda)^x e^{-\lambda m} $$

§7. 3 次の1人の到着時刻が0～mである確率

時刻0～mに誰も来ない確率は，x=0とすると

$$e^{-\lambda m}$$

つまり，最初の1人の到着時刻が0～mである確率(分布関数)は，

$$1 - e^{-\lambda m}$$

mにより微分すると密度関数が得られる.

$$\lambda e^{-\lambda m}$$

§7. 4 ポアソン分布と指数分布

単位時間あたり平均λ人がランダムに到着する場合の，m時間にx人が到着する確率

$$\frac{1}{x!}(m\lambda)^x e^{-\lambda m}$$ xの分布: ポアソン分布

単位時間あたり平均λ人がランダムに到着する場合の，1人が到着するまでの時間mの密度関数

$$\lambda e^{-\lambda m}$$ mの分布: 指数分布

単位時間あたり平均μ人のサービスがランダムに完了する場合の，1人のサービスに要する時間mの密度関数

$$\mu e^{-\mu m}$$ mの分布: 指数分布

§7. 5 M／M／1モデル

M／M／1モデル

- 到着間隔およびサービス時間が指数分布に従い，サービス窓口が1つであるモデル

- 一度並んだら，いくら待たされても途中で逃げない．

待たされる確率:	λ/μ
平均待ち時間:	λ/(μ(μ−λ))
平均待ち客数:	λ*λ/(μ(μ−λ))
平均系内時間:	1/(μ−λ)
平均系内客数:	λ/(μ−λ)

§7. 6 シミュレーション

シミュレーション単位時間τを1とし，0<λ<μ≪1とする．

τの期間内に，たかだか1人が到着し，たかだか1人のサービスが終了する．

τごとに1人が到着する確率はλ，すなわち，τごとに，乱数r(0≦r<1)を求め，0≦r<λの場合に，1人を列に加える．

1人のサービスが終了する確率はμ．すなわち，τごとに，乱数s(0≦s<1)を求め，0≦s<μの場合に，並んでいる列から1人を削除する．

§7. 7 シミュレーションの具体化

```
simul_time: シミュレーション時間(入力)
l:          λ(入力)
m:          μ(入力)
queue_length:   待ち行列の長さ(初期値0)
n:          到着のべ人数(初期値0)
wait_length:    累積待ち行列長さ(サービス中除く)(初期値0)
stay_length:    累積待ち行列長さ(サービス中含む)(初期値0)

for (i=0; i<simul_time; i++) {
  if (queue_length && rand() < RAND_MAX*m)
    queue_length--;
  if (rand() < RAND_MAX*l) {
    queue_length++;
    n++;
  }
  wait_length += queue_length?(queue_length-1):0;
  stay_length += queue_length;
}
  平均待ち時間:     wait_length/n
  平均待ち客数:     wait_length/simul_time
  平均系内時間:     stay_length/n
  平均系内客数:     stay_length/simul_time
```

§7. 8 プログラム(queue.c)

```
static char RcsId[] = "$Header$";

#include <stdio.h>
#include <stdlib.h>

main()
{
  int    i, simul_time;
  int    queue_length, n;
  double l, m;
  double wait_length;
  double stay_length;

  while (scanf("%d %lf %lf", &simul_time, &l, &m) == 3) {
    queue_length = 0;
    n            = 0;
    wait_length  = 0.0;
    stay_length  = 0.0;
    for (i=0; i<simul_time; i++) {
      if (queue_length && rand() < RAND_MAX*m)
        queue_length--;
      if (rand() < RAND_MAX*l) {
        queue_length++;
        n++;
      }
      wait_length += queue_length?(queue_length-1):0;
      stay_length += queue_length;
      printf(" %d", queue_length);
    }
    printf("\n");
    printf("average_wait_time  =%6.2f\t(%6.2f)\n", wait_length/(double)n,        l/(m*(m-l)));
    printf("average_wait_length=%6.2f\t(%6.2f)\n", wait_length/(double)simul_time,l*l/(m*(m-l)));
    printf("average_stay_time  =%6.2f\t(%6.2f)\n", stay_length/(double)n,        1.0/(m-l));
    printf("average_stay_length=%6.2f\t(%6.2f)\n", stay_length/(double)simul_time,1/(m-l));
  }
  exit(0);
}
```

§7. 9 コンパイルと実行

1．コンパイルする.

```
% gcc queue.c -o queue
```

2．実行

```
% ./queue
100 0.1 0.12
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 2 2 2 2 2 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 2 2 2 2
average_wait_time  =  1.50       ( 41.67)
average_wait_length=  0.09       (  4.17)
average_stay_time  = 11.50       ( 50.00)
average_stay_length=  0.69       (  5.00)
```

▶ ()内は理論値

§7. 9 コンパイルと実行(続き)

```
500 0.1 0.12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 2 3 3 3 3 3 2 2 2 2
1 1 2 2 2 3 2 3 3 3 3 3 3 3 2 1 1 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 1 2 2 1
0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1
2 2 2 2 2 2 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
1 2 2 1 1 1 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 3 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 6 6
6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 6 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 9 10 10 10
10 10 11 11 10 10 10 10 10 10 11 11 11 10 10 10 11 11 10 10 10 10 11 11 12
12 12 12 12 12 12 12 12 12 12 12 13 13 14 15 15 15 15 15 15 15 16 17 16 16
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 15 15 15 15 15 15
15 15 15 15 15 15 15 15 14 14 14 13 13 13 13 13 12 12 12 12 12 12 12 11
11 11 11 11 11 11 11 12 12 12 11 11 10 11 11 11 11 11 11 11 11 11 11 11
12 11 11 11 11 11 11 11 11 11 11 11 11 12 12 12 12 11 12 12 12 12 13 13
13 13 13 13 13 13 14 13 13 13 13 12 12 12 12 12 12 13 13 13 14 14 14
14 14 14 14 14 14 14 13 13 13 13 13 13 13 13 13 12 12 12 11 12 12 12 12
12 12 12 12 13 14 14 14 14 14 14 15 15 15 15 14 15 15 15 15 14
average_wait_time  = 51.26       ( 41.67)
average_wait_length=  5.95       (  4.17)
average_stay_time  = 58.29       ( 50.00)
average_stay_length=  6.76       (  5.00)
```

▶ ()内は理論値

§7. 10 例題

あえてλ<μを破り，λ=0.1，μ=0.08を入力してみよ.

その際，シミュレーション時間を100，500，1000と変化させて，どのような現象が起こるか，また，その理由を簡単に述べよ.

ヒント:理論値は正しく表示されない

§7. 11 今日の課題

シミュレーション時間の前半はサービスが停止し，後半に倍速のサービスを提供するようソースプログラムを変更せよ.

また，シミュレーション時間を500，λ=0.1，μ=0.12とした場合，従来サービスと比べて平均待ち時間が何倍になるか測定せよ.

ヒント:500 0.1 0.12を入力するごとに結果は変わる.

宛先: nakashim@econ.kyoto-u.ac.jp
件名: unix2-学生番号

今日はここまで