

## プログラムのページ

担当 伊 理 正 夫

### 6404. 待ち行列のシミュレーション

伏見 正則 (東大大学院数物系研究科)

典型的な待ち行列の性質をシミュレーションによって調べたい。窓口は  $S$  個あって、どの窓口も同じ能率で同じサービスを提供する。客は先着順にサービスを受け、しかも窓口が空いた瞬間にサービスを受け始める。客の到着の仕方は、ポアソン到着であるとし、一人の客が受けるサービスの時間は指数分布に従うとする。以下に、東大計算センターの OKITAC 5090 のための ALGOL である ALGOLIP によるプログラムを示す。

```

begin integer I, I0, J, J1, Q, Q0, Q1, Q2,
           Q3, QMAX, L, S, R, REP, N,
           K, H, NOWT;
real LAMBDA, MU, TEND, WT, SWT, ST,
     TIN, T, A, S0I, MINEND, EMPT,
     EMPTYJ, S1, SENDJ, TIN1, W, W1, DT;
array SEND, EMPTY [1:100], S0 [1:1000],
     TQ [0:1000];
integer array NWT [0:50];
procedure REXP;
  begin real RANDOM;
    I0:=I0*L;
    RANDOM:=FLOAT(I0)*1.010-12;
    T:=-LN(RANDOM)/A
  end;
START:
  I0:=READINTEGER;
  L:=READINTEGER;
  REP:=READINTEGER;
  K:=READINTEGER;
  DT:=READREAL;
INPUT:
  S:=READINTEGER;
  LAMBDA:=READREAL;
  MU:=READREAL;
  Q0:=READINTEGER;
  TEND:=READREAL;

  for R:=1 step 1 until REP do
  begin
    SPACE(2);
    PRINTSTRING('  _ _ _ S _ = ');
    PRINTINTEGER(S); SPACE(2);
    PRINTSTRING('  _ _ Q0 _ = ');
    PRINTINTEGER(Q0); CRLF(1);
    PRINTSTRING('  _ _ _ LAM ');
    PRINTSTRING(' BDA _ = _ ');
    PRINTREAL(LAMBDA); SPACE(2);
    PRINTSTRING('  _ MU _ = _ ');
    PRINTREAL(MU); SPACE(1);
    PRINTSTRING('  _ _ _ _ T ');
    PRINTSTRING(' END _ = _ ');
    PRINTREAL(TEND); CRLF(3);
  PREP:
    I:=Q:=QMAX:=NOWT:=0;
    S0I:=TIN:=TIN1:=WT:=0;
    SWT:=EMPT:=EMPTYJ:=0;
    for J:=1 step 1 until S do
      SEND[J]:=EMPTY[J]:=0;
    for Q:=0 step 1 until 1000 do
      TQ[Q]:=0;
    for N:=1 step 1 until K+1 do
      NWT[N]:=0;
  ARRIVE:
    I:=I+1;
    if I≤Q0 then go to L1;
    TIN1:=TIN;
    A:=LAMBDA; REXP;
    TIN:=TIN+T;
    if TIN>TEND then go to CLOSE;
  L1:MINEND:=SEND[1]; J1:=1;
    for J:=2 step 1 until S do
      begin W:=SEND[J];
        if W<MINEND then
          begin MINEND:=W; J1:=J
            end
        end
  end

```

```

    end;
    if TIN<MINEND
        then go to WAITE;
    if I≤Q0 then go to SERVE;
    WT:=0; S0I:=TIN;
    EMPT:=TIN-MINEND;
    EMPTYJ:=EMPTY[J 1] +EMPT;
    EMPTY[J 1]:=EMPTYJ;
    Q2:=Q; Q3:=0; go to L4;
WAITE:
    WT:=MINEND-TIN;
    S0I:=MINEND; EMPT:=0;
    EMPTYJ:=EMPTY[J 1];
    NOWT:=NOWT+1; W:=WT;
    for N:=1 step 1 until K do
        begin W:=W-DT;
            if W≤0 then
                begin
                    NWT[N]:=NWT[N]+1;
                    go to L2
                end
            end;
        NWT[K+1]:=NWT[K+1]+1;
L2:if I≤Q0 then
    begin
        QMAX:=Q:=Q+1;
        S0[Q]:=S0I; go to SERVE
    end;
    for Q1:=1 step 1 until Q do
        if S0 [Q 1]>TIN then
            begin
                Q2:=Q1-1; go to L3
            end;
        Q2:=Q;
L3:Q3:=Q-Q2+1;
    if Q3>QMAX then QMAX:=Q3;
    if QMAX>1000 then
        begin CRLF(2); SPACE(2);
            PRINTSTRING(' TOO_MA ');
            PRINTSTRING(' NY_CUS ');
            PRINTSTRING(' TOMERS ');
            CRLF(5); go to IRR
        end;
    S0[Q3]:=S0I;
L4:if Q2=0 then
    begin
        TQ[Q]:=TQ[Q]+T; Q:=Q3;
        go to SERVE
    end;
    TQ[Q]:=TQ[Q]+S0[1]-TIN1;
    for Q1:=1 step 1 until Q2-1 do
        TQ[Q-Q 1]:=TQ[Q-Q 1]
            +S0[Q 1+1]-S0[Q 1];
    TQ[Q-Q 2]:=TQ[Q-Q 2]
        +TIN-S0[Q 2];
    if Q3≥2 then
        for Q1:=1 step 1 until Q3-1 do
            S0[Q 1]:=S0[Q 2+Q 1];
        Q:=Q3;
SERVE:
    A:=MU; REXP; ST:=T;
    S1:=S0I+ST; SWT:=SWT+WT;
    SEND[J 1]:=S1;
PRINT:
    PRINTINTEGER(I); PRINTREAL(TIN);
    PRINTREAL(S0I); PRINTREAL(S1);
    CRLF(1); SPACE(6);
    PRINTREAL(WT); PRINTREAL(ST);
    PRINTREAL(WT+ST);
    CRLF(2); SPACE(4);
    PRINTINTEGER(Q);
    PRINTINTEGER(J 1);
    PRINTREAL(EMPT);
    PRINTREAL(EMPTYJ);
    CRLF(3);
    go to ARRIVE;
CLOSE:
    CRLF(3);
    for J:=1 step 1 until S do
        begin
            SENDJ:=SEND[J];
            EMPT:=TEND-SENDJ;
            EMPTYJ:=EMPTYJ
                +(if EMPT>0 then
                    EMPT else 0);
            PRINTINTEGER(J);
            PRINTREAL(SENDJ);
            PRINTREAL(EMPT);

```

```

PRINTREAL(EMPTYJ);
PRINTREAL(100.0
      * EMPTYJ/TEND);
CRLF(1)
end;
CRLF(3);
PRINTREAL(SWT);
PRINTINTEG(NOWT);
PRINTREAL(SWT/FLOAT (NOWT));
PRINTREAL(SWT/FLOAT (I-1));
CRLF(3);
if Q=0 then
  TQ[0]:=TQ[0]+TEND -TIN 1
else
  begin
    TQ[Q]:=TQ[Q]+S 0[1]-TIN 1;
    for Q 1:=1 step 1 until Q-1 do
      TQ[Q-Q 1]:=TQ[Q-Q 1]
        +S 0[Q 1+1]-S 0[Q 1];
    W:=TEND-S 0[Q];
    if W>0 then TQ[0]:=TQ[0]+W
  end;
W:=0;
for Q:=0 step 1 until QMAX do
  begin
    W:=W+TQ[Q];
    PRINTINTEG(Q);
    PRINTREAL(TQ[Q]);
    PRINTREAL(W);
    CRLF(1)
  end;
CRLF(3);
NWT[0]:=I-1-NOWT;
W:=FLOAT(I-1); H:=0;
for N:=0 step 1 until K+1 do
  begin
    H:=H+NWT[N];
    PRINTINTEG(N);
    PRINTINTEG(NWT[N]);
    PRINTINTEG(H);
    PRINTREAL(FLOAT(H)/W);
    CRLF(1)
  end;
end;

```

```

CRLF(5);
IRR:
  end;
  go to INPUT;
end

```

データ・テープには、乱数発生用の定数 I 0, L, シミュレーションの繰返し数 REP, 待ち時間のヒストグラムを作るための定数 K, DT を記す。次に、窓口の数 S, 単位時間当たりの客の平均到着数 LAMBDA, 窓口のサービス能力 MU, サービス開始時の客の数 Q 0, 窓口を開いている時間 TEND を 1 組として、必要なだけ記す。

プログラムが実行されると、まずこれらのパラメタの値が印刷され、つづいて客の到着順の番号 I, 到着時刻 TIN, サービスを受け始める時刻 S 0 I, 受け終わる時刻 S 1 が第 1 行に、待ち時間 WT, サービスを受けていた時間 ST, システムの中にいた時間 (WT+ST) が第 2 行に、待ち行列の長さ Q, その客がサービスを受ける窓口の番号 J 1, その窓口がその客の到着時刻迄ひき続いて手空きであった時間 EMP-T, その窓口の手空き時間の累積和 EMPTYJ が第 3 行に印刷され、これが繰返される。最後の処理としては、窓口の番号 J とともに、そのサービス終了時刻 SENDJ, その時迄の継続手空き時間 EMPT (SENDJ が受付を中止する時刻 TEND を越えたときは、その差に負の符号をつけたもの)、手空き時間の累積和 EM-PTYJ, それが、窓口を開いている時間の中で占める割合 (百分率) が印刷され、次に待ち時間の合計 SWT, 待たされた客の数 NOWT, 実際に待たされた客 1 人当たりの平均待ち時間、待つ必要のなかった客も含めて 1 人当たりの平均待ち時間が印刷される。つづいて、待ち行列の長さが Q であった時間 TQ[Q] の分布および累積分布が印刷され、最後に待ち時間の分布 NWT[N], 累積分布および相対度数の累積分布が印刷される。ここに NWT[N] は待ち時間が (N-1)・DT を越え、N・DT 以下であった客の数である。K・DT を越えるものはまとめて NWT[K+1] で表わされる。

なお、プログラム中の SEND [J] は、I 番目より前の客に対して窓口 J のサービスが終わるべき (あるいは終わった) 時刻を表わし、S 0 [Q] は行列の Q 番目の客がサービスを受け始めるはずの時刻を表わしている。

(昭和 39 年 1 月 17 日受付)

## 訂正

### 6404. 待ち行列のシミュレーション

伏見正則（東大大学院数物系研究科）

プログラムの一部を次のように訂正する．

5 巻, 2 号 (94 頁) 左欄最下行の命令  $S0[Q\ 3]:=S0I$ ; を削除し, 同一の命令を右欄 3 行目および 15 行目の命令  $Q:=Q\ 3$ ; の後に挿入する．