ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

4η Ομάδα Ασκήσεων

Νικόλαος Δημητριάδης
03114016
ΗΜΜΥ 8ο

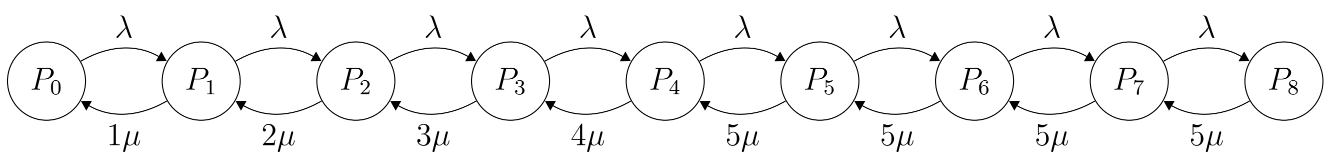
Εισαγωγή

Τα διάφορα μέρη παραδίδονται σε διαφορετικά σε διαφορετικά αρχεία .m και τα επιμέρους ερωτήματά τους χωρίζονται σε sections στον κώδικα για καλύτερη κατανόηση/διόρθωση. Το μέρος 1 βασίστηκε στο demo1.m που αναρτήθηκε στο site του μαθήματος. Χρησιμοποιήθηκε `MATLAB`.

1 Σύστημα M/M/N/K (call center)

ερώτημα 1

Το διάγραμμα ρυθμού μεταβάσεων για το σύστημα M/M/5/8 είναι το ακόλουθο:



ερώτημα 2

Με τη βοήθεια των συναρτήσεων `ctmcbd` και `ctmc` οι εργοδικές πιθανότητες των καταστάσεων του συστήματος για $\lambda = 0.25$ πελάτες/min είναι:

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \end{bmatrix} = \begin{bmatrix} 0.36782 \\ 0.36782 \\ 0.18391 \\ 0.061303 \\ 0.015326 \\ 0.0030652 \\ 0.00061303 \\ 0.00012261 \\ 2.4521 \cdot 10^{-5} \end{bmatrix} \tag{1}$$

Επιπλέον, οι εργοδικές πιθανότητες των καταστάσεων του συστήματος για $\lambda = 1$ πελάτες/min είναι:

$$\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \\ P_7 \\ P_8 \end{bmatrix} = \begin{bmatrix} 0.0168 \\ 0.0672 \\ 0.1344 \\ 0.1792 \\ 0.1792 \\ 0.143360 \\ 0.114688 \\ 0.0917510 \\ 0.0734 \end{bmatrix} \tag{2}$$

ερώτημα 3

Για μία ουρά M/M/N/K γνωρίζουμε ότι:

$$P_{waiting} = \sum_{i=N}^{K} P_i \tag{3}$$

Για το σύστημα που μελετούμε έχουμε:

$$P_{waiting} = \sum_{i=5}^{8} P_i \tag{4}$$

Πιο συγκεκριμένα προέκυψαν τα παρακάτω:

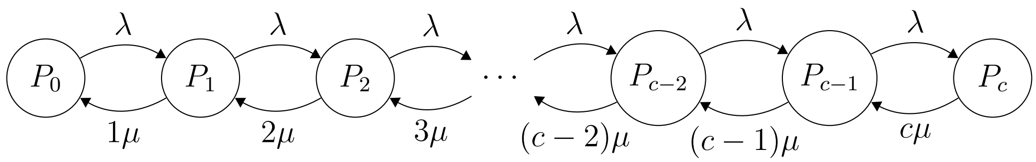
	$P_{waiting}$	$P_{erlangc}$
$\lambda = 0.25$	0.0038253	0.0038314
$\lambda = 1$	0.4232	0.554411

Τότε για τη συνάρτηση `erlangc` παρατηρούμε ότι επιστρέφει αξιόπιστα αποτελέσματα **μόνο για εργοδικά συστήματα** ($\rho < 1 \implies \lambda < \mu$). Αυτό συμβαίενι, διότι η συνάρτηση `erlangc` υπολογίζει την εργοδική πιθανότητα για ένα σύστημα με m πανομοιότυπους εξυπηρετητές, **άπειρη χωρητικότητα**, ρυθμό αφίξεων λ και ρυθμό εξυπηρέτησης του κάθε εξυπηρετητή μ, όταν όλοι οι εξυπηρετητές είναι απασχολημένοι. Επομένως, μόνο για $\lambda = 0.25$ πελάτες/min τα αποτελέσματα της `erlangc` συμπίπτουν με τα θεωρητικά.

2 Ανάλυση και Σχεδιασμός τηλεφωνικού κέντρου

ερώτημα 1

Το διάγραμμα ρυθμού μεταβάσεων για το σύστημα M/M/c/c είναι το ακόλουθο:



Για το σύστημα έχουμε σύμφωνα με το παραπάνω διάγραμμα:

$$P_k = \frac{\lambda}{k\mu} P_{k-1} = \frac{\rho^k}{k!} P_0 \text{ για } k = 1, 2, \cdots, c \tag{5}$$

Επιπλέον, από τον τύπο ολικής πιθανότητας έχουμε:

$$\sum_{k=0}^{c} P_k = 1 \overset{(5)}{\Longrightarrow} P_0 = \frac{1}{\sum_{k=1}^{c} \frac{\rho^k}{k!}} \tag{6}$$

Επιπλέον, το σύστημα απορρίπτει πελάτη όταν όλοι οι εξυπηρετητές είναι απασχολημένοι:

$$P_{blocking} = P_k \overset{(5),(6)}{=} \frac{\frac{\rho^c}{c!}}{\sum_{k=1}^{c} \frac{\rho^k}{k!}} \qquad \blacksquare$$

Η συνάρτηση `erlangb_factorial` που υλοποιεί τον παραπάνω υπολογισμό της $P_{blocking}$ παρουσιάζεται παρακάτω:

Listing 1: `erlangb_factorial`

```
function  B  = erlangb_factorial(rho, c)
    sum = 0;
    for k = 0 : c
        sum = sum + (rho^k)/(factorial(k));
    end
    B = ((rho^c)/factorial(c))*(1/sum);
end
```

ερώτημα 2

ϑδο

$$B(\rho, n) = \frac{\rho B(\rho, n-1)}{\rho B(\rho, n-1) + n} = \frac{\frac{\rho^c}{c!}}{\sum_{k=1}^{c} \frac{\rho^k}{k!}} \tag{7}$$

Θα αποδείξουμε το ζητούμενο με χρήση μαθηματικής επαγωγής:

Βάση
Για $k = 1$ η (7) ισχύει.

Υπόθεση

Υποθέτουμε ότι ισχύει για $k = \mathcal{X}$:

$$B(\rho, \mathcal{X}) = \frac{\frac{\rho^{\mathcal{X}}}{\mathcal{X}!}}{\sum_{k=1}^{\mathcal{X}} \frac{\rho^k}{k!}} \tag{8}$$

Βήμα

Θδο ισχύει για $\mathcal{X} + 1$:

$$\begin{aligned}
B(\rho, \mathcal{X}+1) &= \frac{\rho B(\rho, \mathcal{X})}{\rho B(\rho, \mathcal{X}) + \mathcal{X} + 1} \\
&= \frac{\rho \frac{\frac{\rho^{\mathcal{X}}}{\mathcal{X}!}}{\sum_{k=1}^{\mathcal{X}} \frac{\rho^k}{k!}}}{\rho \frac{\frac{\rho^{\mathcal{X}}}{\mathcal{X}!}}{\sum_{k=1}^{\mathcal{X}} \frac{\rho^k}{k!}} + \mathcal{X} + 1} \\
&= \frac{\frac{\rho^{\mathcal{X}+1}}{\mathcal{X}!}}{\frac{\rho^{\mathcal{X}+1}}{\mathcal{X}!} + (\mathcal{X}+1) \sum_{k=1}^{\mathcal{X}} \frac{\rho^k}{k!}} \\
&= \frac{\frac{\rho^{\mathcal{X}+1}}{(\mathcal{X}+1)!}}{\frac{\rho^{(\mathcal{X}+1)}}{(\mathcal{X}+1)!} + \sum_{k=1}^{\mathcal{X}} \frac{\rho^k}{k!}} \\
&= \frac{\frac{\rho^{\mathcal{X}+1}}{(\mathcal{X}+1)!}}{\sum_{k=1}^{\mathcal{X}+1} \frac{\rho^k}{k!}} \qquad \blacksquare
\end{aligned}$$

και η απόδειξη ολοκληρώθηκε, καθώς καταλήξαμε στον τύπο του προηγούμενου ερωτήματος.

Η συνάρτηση `erlangb_iterative` που υλοποιεί τον παραπάνω αναδρομικό υπολογισμό της $P_{blocking}$ παρουσιάζεται παρακάτω:

Listing 2: `erlangb_iterative`

```
function p = erlangb_iterative( rho, n )
    if n==0
        p = 1;
    else
        a = erlangb_iterative(rho, n-1);
```

```
        p = rho*a / (rho*a + n);
    end
end
```

ερώτημα 3

Παρατηρούμε ότι η `erlangb_iterative` επιστρέφει σωστά αποτελέσματα, ενώ η `erlangb_factorial` δεν επιστρέφει αποτέλεσμα (το `MATLAB` αναγράφει Nan:= Not a Number). Αυτό συμβαίνει, διότι η υλοποίηση της `erlangb_factorial` απαιτεί τη διατήρηση στη μνήμη πολλών πολύ μεγάλων αριθμών (πχ. 1024!), κάτι το οποίο αδυνατεί να κάνει ο υπολογιστής. Αντίθετα, η `erlangb_iterative` απλώς χρειάζεται 1024 θέσεις στη στοίβα κλήσεων και εκτελεί μόνο 1024 πράξεις.

ερώτημα 4

Σύμφωνα με την εκφώνηση, η εταιρία διαθέτει 200 γραμμές και ο πιο απαιτητικός χρήστης χρησιμοποιεί το τηλέφωνο για 23 λεπτά της ώρας.

υποερώτημα (α)

Από τα παραπάνω, συμπεραίνουμε ότι η ένταση φορτίου για ένα χρήστη είναι:

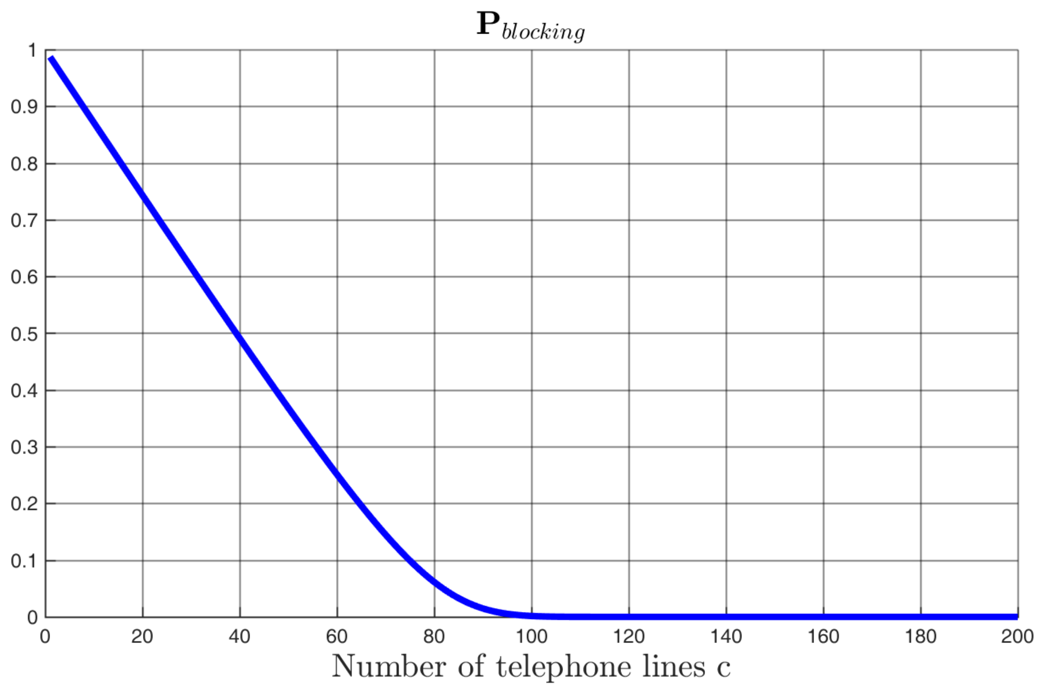
$$\rho_{single} = \frac{23}{60} = 0.38\overline{33} \quad Erlang \tag{9}$$

Επομένως, η συνολική ένταση φορτίου είναι:

$$\rho_{total} = 200 \cdot \rho_{single} = 76.\overline{66} \quad Erlang \tag{10}$$

υποερώτημα (β)

Το ζητούμενο διάγραμμα είναι:



υποερώτημα (γ)

Από το παραπάνω διάγραμμα και με τη βοήθεια του `MATLAB` βρίσκουμε ότι για $c = 93$ τηλεφωνικές γραμμές επιτυγχάνεται πιθανότητα απόρριψης τηλεφωνικής κλήσης μικρότερη του 1%

3 Σύστημα εξυπηρέτησης με δύο ανόμοιους εξυπηρετητές

ερώτημα 1

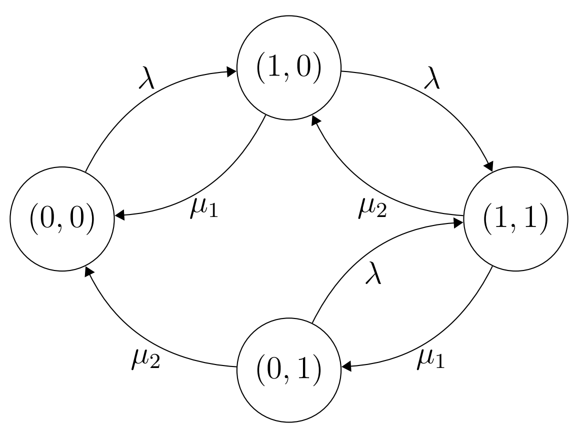


Figure 1: Διάγραμμα ρυθμού μεταβάσεων συστήματος

Από το παραπάνω διάγραμμα προκύπτουν οι εξισώσεις ισορροπίας:

$$\lambda P_{00} = \mu_1 P_{10} + \mu_2 P_{01}$$
$$(\lambda + \mu_1) P_{10} = \lambda P_{00} + \mu_2 P_{11}$$
$$(\lambda + \mu_2) P_{01} P_{01} = \mu_1 P_{11}$$
$$(\mu_1 + \mu_2) P_{11} = \lambda P_{10} + \lambda P_{01}$$

Επιπλέον, από τον τύπο ολικής πιθανότητας έχουμε:

$$\sum P_k = 1 \quad k \in \{(0,0), (0,1), (1,0), (11)\}$$

Επιλύοντας το σύστημα προκύπτει ότι

$$\begin{bmatrix} P_{00} \\ P_{01} \\ P_{10} \\ P_{11} \end{bmatrix}_{theoretical} = \begin{bmatrix} 0.249513 \\ 0.194932 \\ 0.214425 \\ 0.341131 \end{bmatrix} \tag{11}$$

$$P_{blocking}^{theoretical} = 0.341131$$

$$\mathbb{E}[n(t)]_{theoretical} = 1.0916$$

ερώτημα 2

Ο κώδικας της προσομοίωσης παρουσιάζεται στο Appendix (βλ. simulation). Τα πειραματικά αποτελέσματα είναι τα ακόλουθα και η απόκλισή τους από τις θεωρητικές τιμές είναι αμελητέα.

$$\begin{bmatrix} P_{00} \\ P_{01} \\ P_{10} \\ P_{11} \end{bmatrix}_{experimental} = \begin{bmatrix} 0.2494 \\ 0.1968 \\ 0.2144 \\ 0.3393 \end{bmatrix} \tag{12}$$

$$P_{blocking}^{experimental} = 0.3393$$

$$\mathbb{E}[n(t)]_{experimental} = 1.0872$$

4 Appendix

Listing 3: erlangb_factorial

```
function p = erlangb_factorial( rho, n )
    if n==0
        p = 1;
    end
    a = erlangb_factorial(rho, n-1);

    p = rho*a / (rho*a + n);
end
```

Listing 4: erlangb_iterative

```
function p = erlangb_iterative( rho, n )
    if n==0
        p = 1;
    else
        a = erlangb_iterative(rho, n-1);
        p = rho*a / (rho*a + n);
    end
end
```

Listing 5: qsmmmk

```
function [U, R, Q, X, p0, pK] = qsmmmk( lambda, mu, m, K )

  lambda = lambda(:)'; % make lambda a row vector
  mu = mu(:)'; % make mu a row vector
  m = m(:)'; % make m a row vector
  K = K(:)'; % make K a row vector

%   U = 0; R = 0; Q = 0; X = 0; p0 = 0; pK = 0;
  for i=1:length(lambda)
    % Build and solve the birth-death process describing the M/M/m/k system
    birth_rate = lambda(i)*ones(1,K(i));
    death_rate = [ linspace(1,m(i),m(i))*mu(i) ones(1,K(i)-m(i))*m(i)*mu(i)
       ];
    p = ctmc(ctmcbd(birth_rate, death_rate));
    p0(i) = p(1);
    pK(i) = p(1+K(i));
    j = [1:K(i)];
    Q(i) = dot( p(1+j),j );
  end
  % Compute other performance measures
```

```
  X = lambda.*(1-pK);
  U = X ./ (m .* mu );
  R = Q ./ X;
end
```

Listing 6: `simulation` για μέρος 3 (Σύστημα εξυπηρέτησης με δύο ανόμοιους εξυπηρετητές)

```
clc;
clear all;
close all;
%%
rng(1);     % seed
P = zeros(4,1);
arrivals = zeros(4,1);
total_arrivals = 0; % to measure the total number of arrivals
current_state = 0;  % holds the current state of the system
previous_mean_clients = 0; % will help in the convergence test
index = 0;
file = [];
covergence_criterion =  0.00001 ;
maxTransitions = 300000;
lambda = 1;
mu1 = 1/1.25;
mu2 = 1/2.5;
threshold1 = lambda/(lambda + mu1); % the threshold used to calculate
    probabilities
threshold2 = lambda/(lambda + mu2); % the threshold used to calculate
    probabilities
p = mu1/(mu1 + mu2);
threshold = lambda/(lambda + mu1 + mu2);
transitions = 0; % holds the transitions of the simulation in transitions
    steps

while transitions >= 0
  transitions = transitions + 1; % one more transitions step

  if mod(transitions,1000) == 0 % check for convergence every 1000
      transitions steps
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/total_arrivals; % calcuate the probability of
            every state in the system
    end

    mean_clients = 0; % calculate the mean number of clients in the system
```

```

    mean_clients = mean_clients + [0,1,1,2]*P;


    to_plot(index) = mean_clients;

    if abs(mean_clients - previous_mean_clients) <  covergence_criterion || transitions > maxTransitions % convergence test
      break;
    end

    previous_mean_clients = mean_clients;

  end

  random_number = rand(1); % generate a random number (Uniform distribution)
  if current_state == 0
        total_arrivals = total_arrivals + 1;
        arrivals(current_state+1) = arrivals(current_state+1) + 1;
        current_state = 1;
  elseif current_state == 1
      if random_number < threshold1 % arrival
          total_arrivals = total_arrivals + 1;
          arrivals(current_state+1) = arrivals(current_state+1) + 1;
          current_state = 3;
      else
          current_state = 0;
      end
  elseif current_state == 2
      if random_number < threshold2 % arrival
          total_arrivals = total_arrivals + 1;
          arrivals(current_state+1) = arrivals(current_state+1) + 1;
          current_state = 3;
      else
          current_state = 0;
      end
  elseif current_state == 3
    if random_number < threshold
        total_arrivals = total_arrivals + 1;
        arrivals(current_state+1) = arrivals(current_state+1) + 1;
    else
%         random_number = rand(1);
        if random_number < (lambda+mu1)/(lambda + mu1 + mu2); % arrival
              current_state = 2;
        else
```

```
                current_state = 1;
            end
        end
    end


 end

display(P)
```

Listing 7: part 2

```
clc;
close all;
clear all;


%%

rho = 76.66;
c = 200;
for i = 1:200
    p(i) = erlangb_iterative(rho, i);
end


fontsize = 12;
lineWidth = 1;
plotLineWidth = 4;
width=1024;
height=568;

fig1 = figure();
fig1.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
plot(p,'Color','b','LineWidth',plotLineWidth);
temp = gca;

temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
hold off
```

```
% ylim([0 50]);

xlabel(['Number of telephone lines c'],'FontSize',20,'Interpreter','latex');
title(['$\mathbf{P}_{blocking}$'],'FontSize',20, 'Interpreter','latex');
grid on;
saveas(gcf,'figure2_4c.png');
%% 2.4(c)
x = find(p < 0.01);
result = x(1)
```