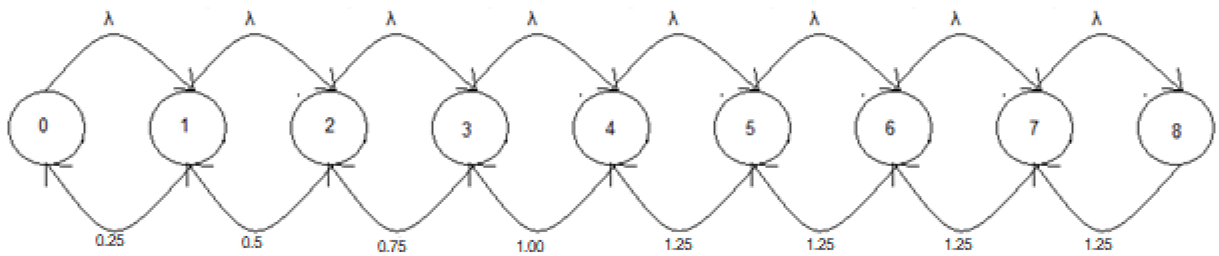
Συστήματα Αναμονής

4η Εργαστηριακή Άσκηση

Όνομα: Σταύρος Σταύρου
ΑΜ: 03115701
Εξάμηνο: 6ο-ΣΗΜΜΥ

Σύστημα M/M/N/K (call center)

(1) Έχουμε το εξής διάγραμμα μεταβάσεων:



(2) Με τον κώδικα qsmm58 .m (στο παράρτημα παίρνουμε τα εξής αποτελέσματα:

Για $\lambda = \frac{1}{4}\frac{\pi\varepsilon\lambda\acute{\alpha}\tau\varepsilon\varsigma}{min}$:

```
Ergodic probability of state 0 is 0.36782
Ergodic probability of state 1 is 0.36782
Ergodic probability of state 2 is 0.18391
Ergodic probability of state 3 is 0.061303
Ergodic probability of state 4 is 0.015326
Ergodic probability of state 5 is 0.0030652
Ergodic probability of state 6 is 0.00061303
Ergodic probability of state 7 is 0.00012261
Ergodic probability of state 8 is 2.4521e-005
```

Για $\lambda = 1\frac{\pi\varepsilon\lambda\acute{\alpha}\tau\varepsilon\varsigma}{min}$:

```
Ergodic probability of state 0 is 0.0168
Ergodic probability of state 1 is 0.0672
Ergodic probability of state 2 is 0.1344
Ergodic probability of state 3 is 0.1792
Ergodic probability of state 4 is 0.1792
Ergodic probability of state 5 is 0.14336
Ergodic probability of state 6 is 0.11469
Ergodic probability of state 7 is 0.091751
Ergodic probability of state 8 is 0.0734
```

(3) Έχουμε $P[A\nu\alpha\mu o\nu\acute{\eta}] = P[5] + P[6] + P[7] + P[8]$.

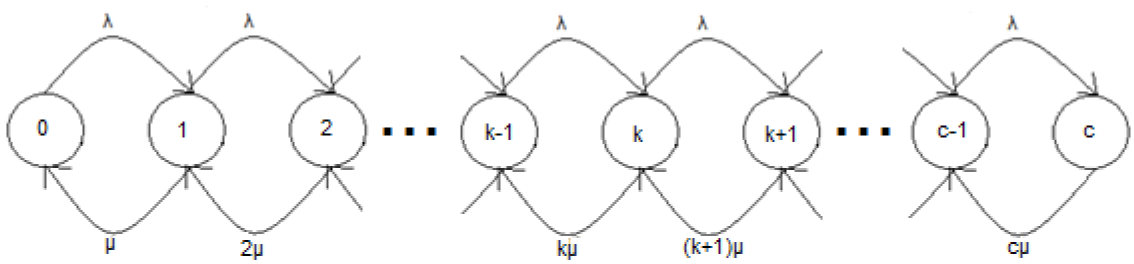
Για $\lambda = \frac{1}{4}\frac{\pi\varepsilon\lambda\acute{\alpha}\tau\varepsilon\varsigma}{min}$ έχουμε $P[A\nu\alpha\mu o\nu\acute{\eta}] =$ 0.0038253. Η συνάρτηση erlangc δίνει 0.0038314.

Για $\lambda = 1\frac{\pi\varepsilon\lambda\acute{\alpha}\tau\varepsilon\varsigma}{min}$ έχουμε $P[A\nu\alpha\mu o\nu\acute{\eta}] =$ 0. 0.4232. Η συνάρτηση erlangc δίνει 0.55411.

Παρατηρούμε, πως παρότι η συνάρτηση erlangc αφορά συστήματα άπειρης χωρητικότητας, εν τούτοις δίνει εξαιρετικά ακριβή αποτελέσματα (με πολύ μικρή απόκλιση) και για συστήματα πεπερασμένης χωρητικότητας εφόσον το φορτίο που λαμβάνει είναι αρκετά μικρό.
Από την άλλη για μεγαλύτερα φορτία ρ αυτό δεν ισχύει, καθώς η απόρριψη κάποιων πακέτων σε ένα σύστημα πεπερασμένης χωρητικότητας, μειώνει την αναμονή κάποιου μελλοντικού πελάτη.

<u>Ανάλυση και Σχεδιασμός τηλεφωνικού κέντρου</u>

(1) Έχουμε το εξής διάγραμμα μεταβάσεων:



Παίρνοντας τις εξισώσεις ισορροπίας του τυχαίου κόμβου k έχουμε $k\mu P_k = \lambda P_{k-1}$ ➔ $P_k = \frac{\lambda}{k\mu} P_{k-1} = \frac{1}{k}\rho P_{k-1} = (\frac{\rho^k}{k!})P_0$.
Επίσης από συνθήκη κανονικοποίησης $P_0 + P_1 + \cdots + P_c = 1$➔ $P_0 = \frac{1}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$.
Και από τις 2 πιο πάνω σχέσεις $P_c = P[Blocking] = B(\rho, c) = \frac{\rho^c}{c!} \times \frac{1}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$.
Ο μέσος ρυθμός απωλειών δίδεται από τη σχέση $\lambda - \gamma = \lambda \times P[Blocking] = \frac{\rho^c}{c!} \times \frac{\lambda}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}$.

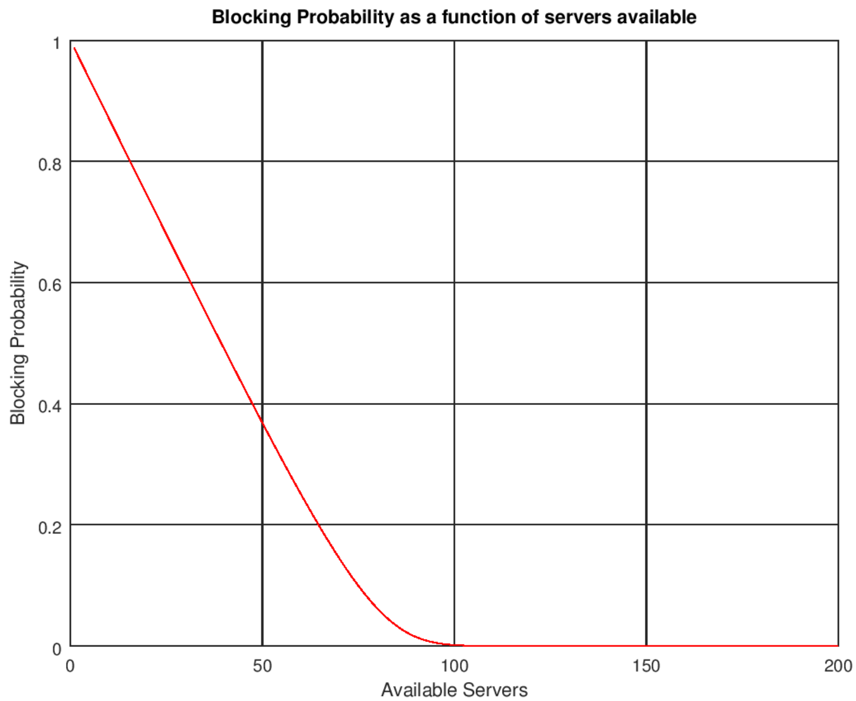
(2) Έχουμε $B(\rho, c) = \frac{\rho^c}{c!} \times \frac{1}{\sum_{k=0}^{c} \frac{\rho^k}{k!}} = \frac{1}{\sum_{k=0}^{c} \frac{c!\rho^k}{\rho^c k!}} = \frac{1}{\sum_{k=0}^{c} \frac{c!}{\rho^{c-k} k!}}$. Παίρνουμε για c+1 την εξής σχέση:
$B(\rho, c+1) = \frac{1}{\sum_{k=0}^{c+1} \frac{(c+1)!}{\rho^{c+1-k} k!}} = \frac{1}{1+\sum_{k=0}^{c} \frac{(c+1)!}{\rho^{c+1-k} k!}} = \frac{1}{1+\frac{c+1}{\rho}\sum_{k=0}^{c} \frac{c!}{\rho^{c-k} k!}} = \frac{1}{1+\frac{c+1}{\rho}\times\frac{1}{B(\rho,c)}} = \frac{\rho B(\rho,c)}{\rho B(\rho,c)+c+1}$ και αντικαθιστώντας το c+1 με n παίρνουμε την τελική σχέση $B(\rho, n) = \frac{\rho B(\rho,n-1)}{\rho B(\rho,n-1)+n}$. Από την προηγούμενη σχέση παίρνουμε και την «βάση» της αναδρομής $B(\rho, 0) = \frac{\rho^0}{0!} \times \frac{1}{\sum_{k=0}^{0} \frac{\rho^k}{k!}} = 1$.

(3) Παρατηρούμε πως πράγματι η erlangb_iterative μας δίνει τη σωστή τιμή 0.024524. Όμως η erlanbg_factorial δίνει τιμή NaN. Αυτό συμβαίνει, επειδή στην erlangb_factorial έχουμε τον υπολογισμό πολύ μεγάλων τιμών (πχ 1024!) με αποτέλεσμα να έχουμε υπερχείλιση στην Octave και αδυναμία υπολογισμού κάποιου αποτελέσματος.

(4) Μοντελοποιώ το σύστημα ως εξής: $\lambda = 1\frac{\kappa\lambda\acute{\eta}\sigma\eta}{\acute{\omega}\rho\alpha}$ με διάρκεια $\frac{1}{\mu} = \frac{23}{60}\acute{\omega}\rho\varepsilon\varsigma$ ➔ $\rho = \frac{\lambda}{\mu} = \frac{23}{60} = 0.38333\, Erlangs$. Με διαφορετική μοντελοποίηση καταλήγω στο ίδιο αποτέλεσμα. Για παράδειγμα $\lambda = \frac{23}{60}\frac{\kappa\lambda\acute{\eta}\sigma\varepsilon\iota\varsigma}{\lambda\varepsilon\pi\tau\acute{o}}$ με διάρκεια $\frac{1}{\mu} = 1\, \lambda\varepsilon\pi\tau\acute{o}$ ➔ $\rho = \frac{\lambda}{\mu} = \frac{23}{60} = 0.38333\, Erlangs$. Αυτό

όμως είναι η «συνεισφορά» μόνο ενός χρήστη. Συνεπώς το προσφερόμενο φορτίο είναι $\rho_{o\lambda} = 200 \times \rho = 76.66667\ Erlangs$.

Με χρήση της Octave παίρνουμε το εξής διάγραμμα:



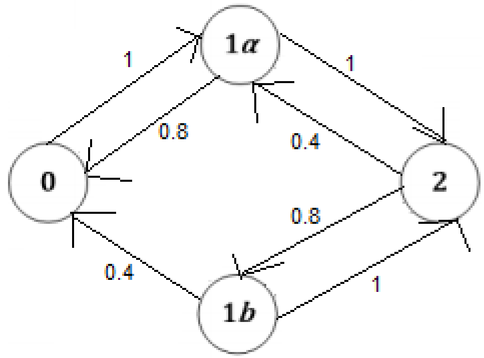
Και την εξής έξοδο για εύρεση της τιμής που δίνει πιθανότητα απόρριψης κλήσης <1%:

```
We want 93 servers for a probability less than 1%. The probability is 0.008368
The probability for 92 servers is 0.010236
```

Όλοι οι κώδικες που χρησιμοποιήθηκαν, καθώς και οι συναρτήσεις που ζητούνται, βρίσκονται στο παράρτημα στο τέλος.

Σύστημα εξυπηρέτησης με δύο ανόμοιους εξυπηρετητές

(1) Έχουμε το εξής διάγραμμα καταστάσεων:



Έχουμε $1 \times P_0 = 0.8 \times P_{1a} + 0.4 \times P_{1b}$, $1.8 \times P_{1a} = 1 \times P_0 + 0.4 \times P_2$, $1.4 \times P_{1b} = 0.8 \times P_2$ και τη συνθήκη κανονικοποίησης $P_0 + P_{1a} + P_{1b} + P_2 = 1$. Λύνοντας παίρνουμε $P_0 = 24.951\%$, $P_{1a} = 21.442\%, P_{1b} = 19.493\%$ και $P_2 = 34.113\%$.
Έχουμε, επίσης $P[Blocking] = P_2 = 34.113\%$.
Τέλος, ο μέσος αριθμός πελατών δίνεται από τον πιθανοτικό τύπο $E[n(t)] = 1 \times (0.19493 + 0.21442) + 2 \times 0.34113 = 1.09161 \pi\varepsilon\lambda\acute{\alpha}\tau\varepsilon\varsigma$.

(2) Η προσομοίωση στην Octave δίνει:

```
The ergodic probabilities of the system are (P0, P1a, P1b, P2):
   0.24832   0.21370   0.19474   0.34323
The blocking probability of the system is 0.34323
The average number of clients in the system is 1.0949
```

Παρατηρούμε τα αποτελέσματα της προσομοίωσης να είναι αρκετά κοντά στις τιμές που υπολογίσαμε στο προηγούμενο ερώτημα.

Ο κώδικας της προσομοίωσης ακολουθεί στο παράρτημα.

ΠΑΡΑΡΤΗΜΑ

qsmm58.m:

```
clc;
clear all;
close all;


lambda = 1;
mu = 1/4;
states = 0:1:8;
initial_sate = [1, 0, 0, 0, 0, 0, 0, 0 , 0];   # initial state of the system

genniseis = [lambda, lambda, lambda, lambda, lambda, lambda, lambda, lambda];
thanatoi = [mu, 2*mu, 3*mu, 4*mu, 5*mu, 5*mu, 5*mu, 5*mu];

metavatikos = ctmcbd(genniseis, thanatoi);


ergodic_prob = ctmc(metavatikos);

format short g;
for i = 1:1:9
  message = cstrcat ("Ergodic probability of state ", num2str(i-1), " is ", num2str(ergodic_prob(i)));
  disp(message);
endfor


P_waiting1 = ergodic_prob(6) + ergodic_prob(7) + ergodic_prob(8) + ergodic_prob(9);
disp(cstrcat("The possibility of every server being busy is ", num2str(P_waiting1)));
P_waiting = erlangc(lambda/mu, 5);
disp(cstrcat("The possibility of every server being busy (for an infinite system) is ", num2str(P_waiting)));
```

erlangb_factorial.m:

```
function pblock = erlangb_factorial(rho, c)
  if ( nargin != 2 )
    print_usage();
  endif
  sum = 0;
  for k = 1:1:c+1
    sum = sum + (rho**(k-1))/(factorial(k-1));
  endfor

  pblock = (rho**c)/((factorial(c))*sum);
endfunction
```

erlangb_iterative.m:

```
function pblock = erlangb_iterative(rho, c)
  if ( nargin != 2 )
    print_usage();
  endif
  results = zeros(1, c + 1);
  results(1) = 1;
  for i = 2:1:c+1
    results(i) = rho*results(i-1) / (rho*results(i-1) + i-1);
  endfor
  pblock = results(c + 1);
endfunction
```

Call_center.m:

```
clc;
clear all;
close all;

blocking_prob = zeros(1, 200);
rho = 76.66667;
c = 1:1:200;


#calculating the blocking probabilities
for i = 1:1:200;
  blocking_prob(i) = erlangb_iterative(rho, i);
endfor

#finding the first probability less than 1%
for i = 1:1:200
  if blocking_prob(i) < 0.01
    break;
  endif
endfor

disp(cstrcat("We want ", num2str(i), " servers for a probability less than 1%. The probability is ",
num2str(blocking_prob(i))));
disp(cstrcat("The probability for ", num2str(i-1), " servers is ", num2str(blocking_prob(i-1))));
figure(1);
plot(c, blocking_prob, 'r');
grid on;
title("Blocking Probability as a function of servers available");
xlabel("Available Servers");
ylabel("Blocking Probability");
```

qsmm2:

```
clc;
clear all;
close all;

state1b = 0;
state1a = 0;
total_arrivals = 0;
current_state = 0;
previous_mean_clients = 0;
sigklisi = false;
transitions = 0;
arrivals = zeros(1, 4);
P = zeros(1, 4);



while transitions <= 300000 && !sigklisi
  decision = rand(1);
  transitions = transitions + 1;
  if current_state == 0
    total_arrivals = total_arrivals + 1;
    current_state = 1;
    arrivals(1) = arrivals(1) + 1;
    state1a = 1;
  elseif state1a && (decision < 1 / 2.2)
    total_arrivals = total_arrivals + 1;
    current_state = 2;
    arrivals(2) = arrivals(2) + 1;
    state1a = 0;
  elseif state1a && (decision > 1 / 2.2) && (decision < 1.8 / 2.2)
    current_state = 0;
    state1a = 0;
  elseif state1b && (decision < 1 / 2.2)
    total_arrivals = total_arrivals + 1;
    current_state = 2;
    arrivals(3) = arrivals(3) + 1;
    state1b = 0;
  elseif state1b && (decision > 1.8 / 2.2)
    current_state = 0;
    state1b = 0;
  elseif (current_state == 2) && (decision < 1 / 2.2)
    total_arrivals = total_arrivals + 1;
    arrivals(4) = arrivals(4) + 1;
  elseif (current_state == 2) && (decision > 1.8 / 2.2)
    current_state = 1;
    state1a = 1;
  elseif (current_state == 2) && (decision > 1 / 2.2) && (decision < 1.8 / 2.2)
```

```
    current_state = 1;
    state1b = 1;
  endif

  if mod(transitions, 1000) == 0
    for i = 1:1:4
      P(i) = arrivals(i)/total_arrivals;
    endfor
    mean_clients = P(2) + P(3) + 2 * P(4);

    if abs(mean_clients - previous_mean_clients) < 0.00001
      sigklisi = true;
    endif
    previous_mean_clients = mean_clients;
  endif
endwhile

disp("The ergodic probabilities of the system are (P0, P1a, P1b, P2):");
disp(P);
disp(cstrcat("The blocking probability of the system is ", num2str(P(4))));
disp(cstrcat("The average number of clients in the system is ", num2str(mean_clients)));
```