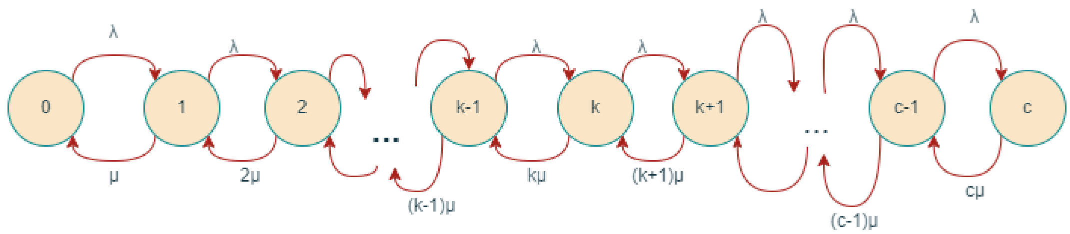
ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ
4Η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ

Ονοματεπώνυμο: Κουστένης Χρίστος
Α.Μ. : el20227
Ακαδημαϊκό Έτος: 2022-2023

Ανάλυση και Σχεδιασμός τηλεφωνικού κέντρου

Στην άσκηση αυτή θα ασχοληθούμε με την ανάλυση και το σχεδιασμό ενός τηλεφωνικού κέντρου, που μοντελοποιείται ως μία ουρά M/M/c/c, δηλαδή ουρά που διαθέτει c εξυπηρετητές ίδιων δυνατοτήτων και μέγιστη χωρητικότητα c πελάτες. Οι αφίξεις στο σύστημα ακολουθούν την κατανομή Poisson με ομοιόμορφο μέσο ρυθμό λ και οι εξυπηρετήσεις ακολουθούν την εκθετική κατανομή με ομοιόμορφο μέσο ρυθμό μ.

(1)



Εικόνα 1(Διάγραμμα ρυθμού μεταβάσεων M/M/c/c)

Έστω Pn η πιθανότητα να βρίσκεται το σύστημα στην κατάσταση n δηλαδή n πελάτες να εξυπηρετούνται από n εξυπηρετητές. Η πιθανότητα απόρριψης(blocking probability)P_b ενός πελάτη ισούται με P_c δηλαδή με την πιθανότητα να είναι πλήρες το σύστημα. Στην περίπτωση του συστήματος μας αλυσίδα Markov είναι πάντα εργοδική οπότε η στατική κατανομή πιθανότητας με βάση τη γενική λύση των εξισώσεων ισορροπίας προκύπτει ως εξής:

$$P_n = P_0 \prod_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}} \text{ , } n = 1,2,\ldots \quad \Rightarrow P_n = P_0 \frac{\rho^n}{n!} \quad n = 0,1,2,\ldots,c$$

και $P_0 = [1+ \sum_{n=1}^{\infty} \prod_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}}]^{-1} \Rightarrow P_0 = [\sum_{i=0}^{c} \frac{\rho^i}{i!}]$

Επομένως για P_c=P_b ισχύει:

$$P_{blocking} = \frac{\rho^c/c!}{1+\rho+\rho^2/2! + \ldots + \rho^c/c!} = B(\rho,c) \text{ όπου } \rho = \frac{\lambda}{\mu}$$

Για τον μέσο ρυθμό απωλειών πελατών από μια ουρά έχουμε την πιθανότητα απόρριψης πελάτη πολλαπλασιασμένη με τον ρυθμό άφιξης πελατών, επομένως έχουμε:

$\lambda\text{-}\gamma = \lambda - \lambda(1- P_{blocking}) = \lambda \,.\, P_{blocking} = \lambda \,.\, B(\rho,c)$

Στη συνέχεια υλοποιούμε τη ζητούμενη συνάρτηση erlangb_factorial στο Octave όπως φαίνεται στον Κώδικα 1.

```
% c : number of servers
% r = lambda / mu

function p = erlangb_factorial (r,c)
  sum = 0;
  for k = 0:1:c
    sum = sum +(power(r,k)/factorial(k));
  endfor
  p = (power(r,c)/factorial(c))/sum;
endfunction
```

Κώδικας 1(erlangb_factorial)

```
erlangb_factorial(5,5):
0.2849
erlangb(5,5):
0.2849
>>
```

Έξοδος 1(Έλεγχος ορθότητας erlangb_factorial)

(2)

Για τον τύπο Erlang-B ισχύει ο ακόλουθος επαναληπτικός τύπος:

$$B(\rho, 0) = 1$$

$$B(\rho, n) = \frac{\rho B(\rho, n-1)}{\rho B(\rho, n-1) + n}, n = 1,2, \dots, c$$

Με βάση την παραπάνω σχέση υλοποιούμε την erlangb_iterative:

```
function p = erlangb_iterative (r,c)
  p = 1;
  for i=0:1:c
    p = ((r*p)/((r*p)+i));
  endfor
endfunction
```

Κώδικας 2(erlangb_iterative)

```
erlangb_iterative(5,5):
0.2849
erlangb(5,5):
0.2849
```

Έξοδος 2(Έλεγχος ορθότητας erlangb_iterative)

(3)

Δοκιμάζουμε να τρέξουμε τις συναρτήσεις erlangb_factorial και erlangb_iterative με παραμέτρους ρ = 1024 και c = 1024. Το αποτέλεσμα εκτέλεσης του προγράμματος μας φαίνεται στην Έξοδο 3.

```
erlangb_factorial(1024,1024):
NaN
erlangb_iterative(1024,1024):
0.024524
```

Έξοδος 3

Παρατηρούμε ότι στην περίπτωση της συνάρτησης erlangb factorial(1024,1024), ο επιμέρους υπολογισμός του 1024! καταλήγει σε έναν πολύ μεγάλο αριθμό πριν τον υπολογισμό του τελικού πηλίκου οπότε δεν έχουμε κάποιο αποτέλεσμα λόγω του overflow. Από την άλλη για την συνάρτηση erlangb iterative(1024,1024) υπολογίζεται κανονικά η πιθανότητα αφού οι επιμέρους υπολογισμοί δεν οδηγούν σε overflow.

(4)

(α)

Καλούμαστε να σχεδιάσουμε από την αρχή το τηλεφωνικό δίκτυο μίας εταιρείας στην οποία απασχολούνται 200 εργαζόμενοι. Κάθε εργαζόμενος διαθέτει και μία εξωτερική γραμμή, δηλαδή η εταιρεία πληρώνει για 200 γραμμές. Για να σχεδιάσουμε ένα πιο οικονομικό δίκτυο, κάνουμε μετρήσεις στην ώρα αιχμής και βρίσκουμε ότι ο πιο απαιτητικός χρήστης χρησιμοποιεί συνολικά το τηλέφωνο του για εξωτερικές κλήσεις κατά μέσο όρο 23 λεπτά σε μία ώρα. Επομένως, η συνολική ένταση του φορτίου που καλείται να εξυπηρετήσει το τηλεφωνικό δίκτυο της εταιρείας είναι :

$$\rho = 200\,\frac{23}{60} \approx 76.667 \text{ Erlangs}$$

(β)

Στη συνέχεια, σχεδιάζουμε το διάγραμμα της πιθανότητας απόρριψης πελάτη από το σύστημα ως προς τον αριθμό των τηλεφωνικών γραμμών, επιλέγοντας από 1 έως 200 τηλεφωνικές γραμμές και για τον υπολογισμό της πιθανότητας χρησιμοποιούμε τη erlangb_iterative.

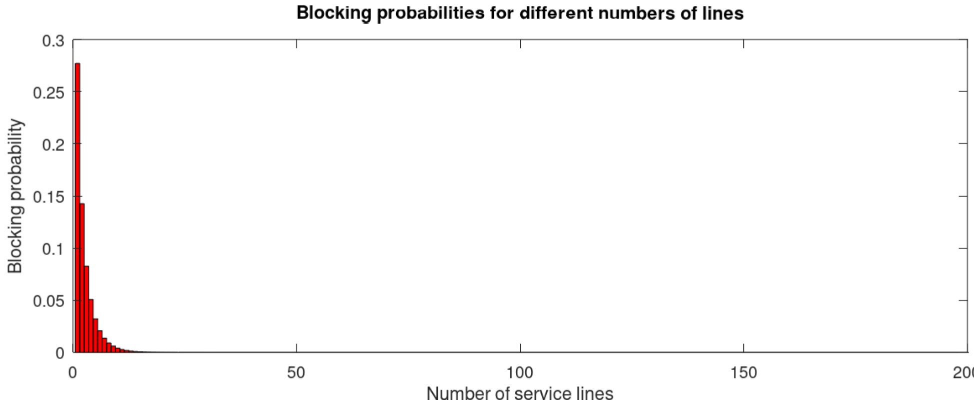


Figure 1

(γ)

Μεγεθύνοντας το Figure 1 παρατηρούμε ότι πιαθνότητα απόρριψης μικρότερη από 0.01% επιτυγχάνεται για περισσότερες από 8 τηλεφωνικές γραμμές. Επομένως, με βάση το απαιτούμενο GoS (Grade of Service) το οικονομικότερο δίκτυο επιτυγχάνεται με 8 τηλεφωνικές γραμμές.

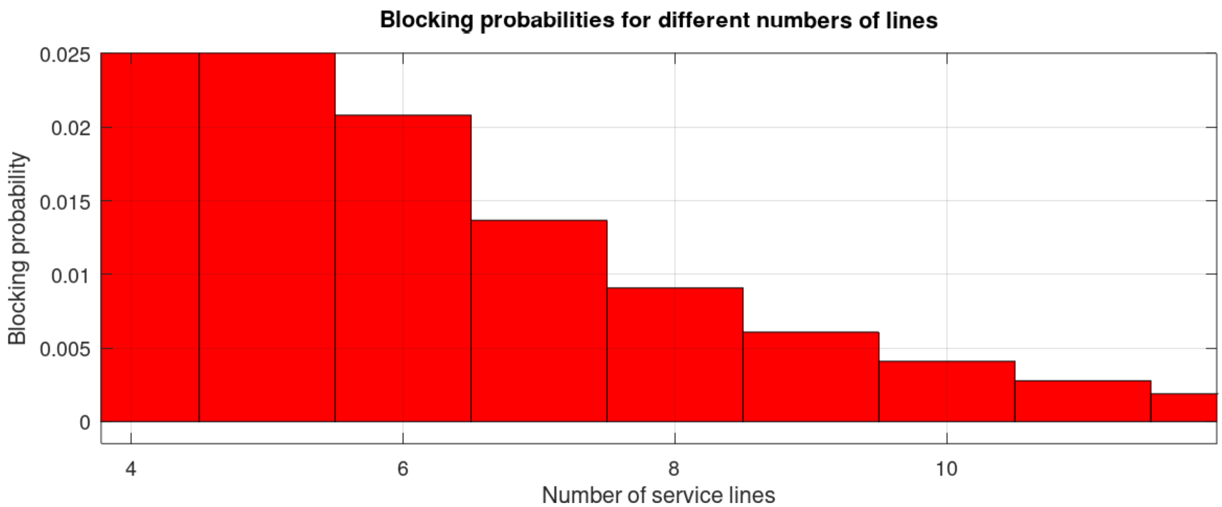


Figure 2

Ακολουθεί ο κώδικας που αντιστοιχεί στα παραπάνω ερωτήματα που βρίσκεται στο αρχείο « erlangb_tele_service.m ».

```
clc;
clear all;
close all;
pkg load queueing;
% c : number of servers
% r = lambda / mu

%(1)
function p = erlangb_factorial (r,c)
  sum = 0;
```

```
  for k = 0:1:c
    sum = sum +(power(r,k)/factorial(k));
  endfor
  p = (power(r,c)/factorial(c))/sum;
endfunction

% Testcase


display("erlangb_factorial(5,5): ");
disp(erlangb_factorial(5,5));

display("erlangb(5,5): ")
disp(erlangb(5,5));


% (2)
function p = erlangb_iterative (r,c)
  p = 1;
  for i=0:1:c
    p = ((r*p)/((r*p)+i));
  endfor
endfunction


display("erlangb_iterative(5,5): ");
disp(erlangb_iterative(5,5));

display("erlangb(5,5): ")
disp(erlangb(5,5));


%(3)


display("erlangb_factorial(1024,1024): ");
disp(erlangb_factorial(1024,1024));

display("erlangb_iterative(1024,1024): ");
disp(erlangb_iterative(1024,1024));

%(4)

%(b)
P = zeros(200,1);
for k = 1:1:200

P(k)=  erlangb_iterative(k*(23/60),k)

endfor

figure(1);
bar(P,'r','hist');
xlim([0,200]);
title("Blocking probabilities for different numbers of lines");
xlabel("Number of service lines");

ylabel("Blocking probability");
```

Σύστημα εξυπηρέτησης με δύο ανόμοιους εξυπηρετητές

Θεωρούμε ένα σύστημα εξυπηρέτησης που αποτελείται από δύο εξυπηρετητές 1 και 2 χωρίς δυνατότητα αναμονής πελατών. Όταν και οι δύο εξυπηρετητές είναι διαθέσιμοι, ένας πελάτης δρομολογείται πάντα στον εξυπηρετητή 1. Σε περίπτωση που ο εξυπηρετητής 1 δεν είναι διαθέσιμος, ένας νέος πελάτης δρομολογείται στον εξυπηρετητή 2. Σε περίπτωση που και οι δύο εξυπηρετητές δεν είναι διαθέσιμοι, ένας νέος πελάτης απορρίπτεται χωρίς επανάληψη προσπάθειας εξυπηρέτησης.Οι αφίξεις πελατών στο σύστημα ακολουθούν την κατανομή Poisson με μέσο ρυθμό $\lambda = 1$ πελάτες/sec. Οι εξυπηρετήσεις είναι εκθετικά κατανεμημένες με μέσους χρόνους εξυπηρέτησης $1/\mu_1 = 1.25$ sec και $1/\mu 2 = 2.5$ sec αντίστοιχα, δηλαδή ο εξυπηρετητής 2 είναι χαμηλότερων δυνατοτήτων από τον εξυπηρετητή 1.

(1)

(α)

Οι ρυθμοί εξυπηρέτησης μ_1 και μ_2 είναι ίσοι με $\mu_1 = 1/1.25 = 0.8$ πελάτες/sec και $\mu_2 = 1/2.5 = 0.4$ πελάτες/sec.

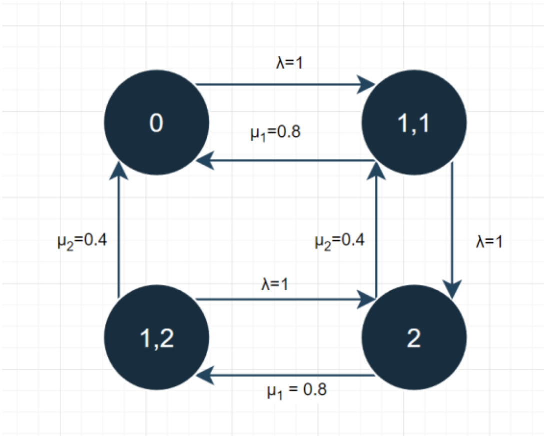


Figure 3(Διάγραμμα ρυθμού μεταβάσεων)

Το σύστημα εξισώσεων που προκύπτει για το σύστημα από τις εξισώσεις ισορροπίας είναι το ακόλουθο:

- $\lambda\cdot P_0 = \mu_1 P_{11} + \mu_2 P_{12}$
- $(\lambda+\mu_1)\cdot P_{11} = \lambda P_0 + \mu_2 P_2$
- $(\lambda+\mu_2)\cdot P_{12} = \cancel{\lambda\cdot P_0} + \mu_1 P_2$
- $P_0 + P_{11} + P_{12} + P_2 = 1$

$(\mu_1=0.8,\ \mu_2=0.4,\ \lambda=1) \rightarrow$

- $P_0 = (0.8)P_{11} + (0.4)P_{12}$
- $(1.8)P_{11} = P_0 + (0.4)P_2$

- $(1.4)P_{12} = (0.8)P_2$
- $P_0 + P_{11} + P_{12} + P_2 = 1$

Με χρήση του Mathematica για διευκόλυνση των υπολογισμών βρίσκουμε τις ακόλουθες τιμές για τις εργοδικές πιθανότητες:

```
In[1]:= eq1 = P0 == 0.8 P11 + 0.4 P12;
        eq2 = 1.8 P11 == P0 + 0.4 P2;
        eq3 = 1.4 P12 == 0.8 P2 ;
        eq4 = P0 + P11 + P12 + P2 == 1;
        Solve[{eq1 && eq2 && eq3 && eq4}, {P0, P11, P12, P2}]
        λύση εξισώσεων και ανισώσεων

Out[5]= {{P0 → 0.249513, P11 → 0.214425, P12 → 0.194932, P2 → 0.341131}}
```

Κώδικας 3

P_0 = 0.24951
P_{11} = 0.21442
P_{12} = 0.19493
P_2 = 0.34113

(β)

Η πιθανότητα απόρριψης πελάτη από το σύστημα ισούται με **P_2 = 0.34113**

(γ)

Ο μέσος αριθμός πελατών στο σύστημα είναι:

$$\sum_{k=0}^{2} k\, P_k = 0 \cdot P_0 + 1 \cdot P_{11} + 1 \cdot P_{12} + 2 \cdot P_2 = 1.0917$$

(2)

(α)

- **threshold_1a:** Από την κατάσταση 1,1 μπορούμε να έχουμε είτε αναχώρηση είτε άφιξη και να μεταβούμε στην κατάσταση 0 με ρυθμό μ_1 είτε στην κατάσταση 2 με ρυθμό λ αντίστοιχα.Επομένως, ο παρονομαστής του threshold ισούται με $\lambda + \mu_1$ και ο αριθμητής με λ.
- **threshold_1b**: Από την κατάσταση 1,2 μπορούμε να έχουμε είτε αναχώρηση είτε άφιξη και να μεταβούμε στην κατάσταση 0 με ρυθμό μ_2 είτε στην κατάσταση 2 με ρυθμό λ αντίστοιχα. Επομένως, ο παρονομαστής του threshold ισούται με $\lambda + \mu_2$ και ο αριθμητής ισούται με λ.
- **threshold_2_first**: Από την κατάσταση 2 μπορούμε να έχουμε άφιξη και απόρριψη του πελάτη με ρυθμό λ, επομένως ο αριθμητής του threshold_2_first θα είναι λ και ο παρονομαστής $\lambda+\mu_1+\mu_2$ *.

- **threshold_2_second**: Από την κατάσταση 2 μπορούμε να μεταβούμε στην κατάσταση 1,2 με αναχώρηση και ρυθμό μ_1, επομένως ο αριθμητής του threshold_2_second θα είναι $\lambda+\mu_1$ και ο παρονομαστής $\lambda+\mu_1+\mu_2$ *.

  ****Μπορούμε να έχουμε αναχώρηση και να πάμε στην κατάσταση 1,1 με ρυθμό μ_2, άρα ο παρονομαστής και των δύο thresholds, τα οποία σχετίζονται με την κατάσταση 2 ισούται με $\lambda+\mu_1+\mu_2$.***

Έτσι, προκύπτουν τα παρακάτω thresholds.

```
threshold_1a = lambda/(lambda+m1);
threshold_1b = lambda/(lambda+m2);
threshold_2_first = lambda/(lambda+m1+m2);
threshold_2_second = (lambda+m1)/(lambda+m1+m2);
```

(β)

Το μοναδικό κριτήριο σύγκλισης της προσομοίωσης μας είναι η διαφορά μεταξύ δύο διαδοχικών μέσων αριθμών πελατών να είναι μικρότερη από 0.001%

(γ)

Παρατηρούμε ότι οι αριθμοί της προσομοίωσης συμφωνούν με τους παραπάνω που υπολογίσαμε με μικρές αποκλίσεις εντός του αναμενόμενου με βάση το κριτήριο σύγκλι σης που ορίσαμε.

```
0.2499
0.2144
0.1930
0.3427
>>
```

Έξοδος 4

Ακολουθεί ο συνολικός κώδικας της Άσκησης 2 του αρχείου «System_with_2_different_servers.m»

```
% Σύστημα εξυπηρέτησης με δύο ανόμοιους εξυπηρετητές
clc;
clear all;
close all;


lambda = 1;
m1 = 0.8;
```

```
m2 = 0.4;

threshold_1a = lambda/(lambda+m1);
threshold_1b = lambda/(lambda+m2);
threshold_2_first = lambda/(lambda+m1+m2);
threshold_2_second = (lambda+m1)/(lambda+m1+m2);

current_state = 0;
arrivals = zeros(1,4);
total_arrivals = 0;
maximum_state_capacity = 2;
previous_mean_clients = 0;
delay_counter = 0;
time = 0;

while 1 > 0
  time = time + 1;

  if mod(time,1000) == 0
    for i=1:1:4
      P(i) = arrivals(i)/total_arrivals;
    endfor

    delay_counter = delay_counter + 1;

    mean_clients = 0*P(1) + 1*P(2) + 1*P(3) + 2*P(4);

    delay_table(delay_counter) = mean_clients;

    if abs(mean_clients - previous_mean_clients) < 0.00001
       break;
    endif
    previous_mean_clients = mean_clients;
  endif

  random_number = rand(1);

  if current_state == 0
      current_state = 1;
      arrivals(1) = arrivals(1) + 1;
      total_arrivals = total_arrivals + 1;
  elseif current_state == 1
    if random_number < threshold_1a
      current_state = 3;
      arrivals(2) = arrivals(2) + 1;
      total_arrivals = total_arrivals + 1;
    else
      current_state = 0;
  endif
  elseif current_state == 2
    if random_number < threshold_1b
      current_state = 3;
      arrivals(3) = arrivals(3) + 1;
      total_arrivals = total_arrivals + 1;
    else
      current_state = 0;
    endif
  else
      if random_number < threshold_2_first
```

```
        arrivals(4) = arrivals(4) + 1;
        total_arrivals = total_arrivals + 1;
      elseif random_number < threshold_2_second
        current_state = 2;
      else
        current_state = 1;
      endif
   endif

endwhile

display(P(1));
display(P(2));
display(P(3));
display(P(4));

figure(1);
bar(P);
```

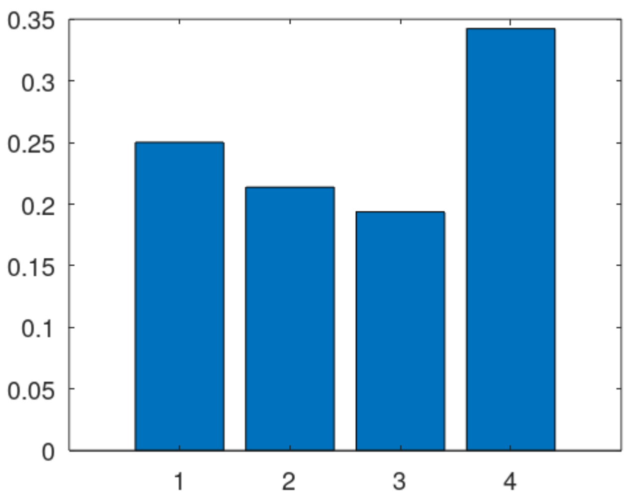


Figure 4(Γραφική αναπαράσταση των πιθανοτήτων του ερωτήματος (γ))