ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

4η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028

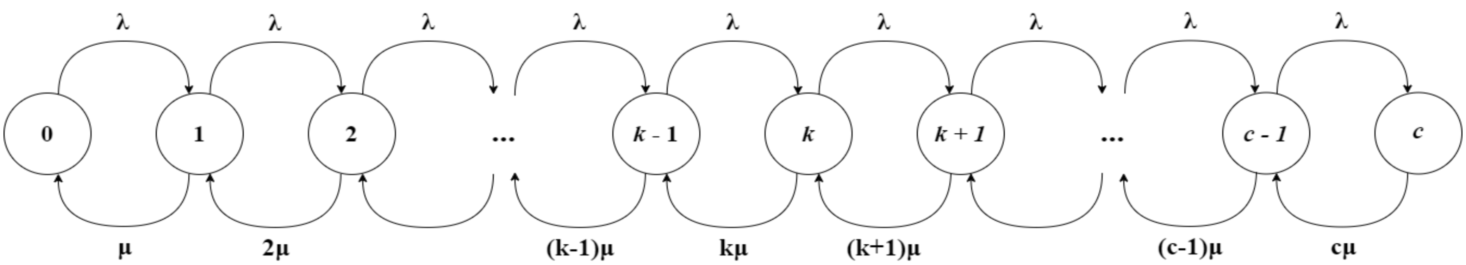
28 ΜΑΙΟΥ, 2023

Ονοματεπώνυμο: Θοδωρής Αράπης	**ΑΜ:** el18028

Ανάλυση και Σχεδιασμός τηλεφωνικού κέντρου

(1)

Σχεδιάζουμε αρχικά το διάγραμμα ρυθμού μεταβάσεων του συστήματος M/M/c/c:



Για την απόδειξη του τύπου της πιθανότητας απόρριψης θα βασιστούμε στις εξισώσεις ισορροπίας:

- $\lambda P_0 = \mu P_1,\ \lambda P_1 = 2\mu P_2, \dots ,\ \lambda P_{k-1} = k\mu P_k \implies$

$$\boxed{P_k = \left[\frac{\lambda}{k\mu}\right] P_{k-1} = \left(\frac{\rho^k}{k!}\right) P_0} \quad \gamma\iota\alpha\ k = 1,2,\dots,c \quad \kappa\alpha\iota \quad \rho \triangleq \frac{\lambda}{\mu}$$

- $P_0 + P_1 + P_2 + \cdots + P_{c-1} + P_c = 1 \implies \boxed{P_0 = \frac{1}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}}$

Οπότε, συνδυάζοντας τις παραπάνω σχέσεις για $k = c$ προκύπτει:

$$\boxed{P_c = P_{blocking} = \frac{\frac{\rho^c}{c!}}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}}$$

Ο μέσος ρυθμός απωλειών πελατών από την ουρά δίνεται από την σχέση:

$$\boxed{\lambda P_{blocking} = \frac{\frac{\lambda \rho^c}{c!}}{\sum_{k=0}^{c} \frac{\rho^k}{k!}}}$$

Η υλοποίηση σε κώδικα Octave της συνάρτησης erlangb_factorial είναι η ακόλουθη:

```
function P_blocking = erlangb_factorial(r, c)
  denominator = 0;
  numerator = power(r, c) / factorial(c);

  for k = 0 : c
    denominator = denominator + (power(r, k) / factorial(k));
  endfor

  P_blocking = 100 * (numerator / denominator);
  printf("(Factorial Method) The probability of rejecting a client when r = %d snd c = %d is: %f%%\n", r, c, P_blocking)
endfunction
```

(2)

Η υλοποίηση σε κώδικα Octave της συνάρτησης erlangb_iterative είναι η ακόλουθη

```
function P_blocking = erlangb_iterative(r, c)
  B = ones(1, c + 1);
  for i = 2 : (c + 1)
    numerator = r * B(i - 1);
    denominator = numerator + (i - 1);
    B(i) = numerator / denominator;
  endfor
  P_blocking = 100 * B(c + 1);
  printf("(Iterative Method) The probability of rejecting a client when r = %d and c = %d is: %f%%\n", r, c, P_blocking)
endfunction
```

(3)

Εκτελούμε τις ζητούμενες εντολές και λαμβάνουμε τις εξής εξόδους των συναρτήσεων:

```
>> erlangb_factorial(1024, 1024)
(Factorial Method) The probability of rejecting a client when r = 1024 snd c = 1024 is: NaN%
ans = NaN
>> erlangb_iterative(1024, 1024)
(Iterative Method) The probability of rejecting a client when r = 1024 and c = 1024 is: 2.452426%
ans = 2.4524
```

Όπως βλέπουμε από τις εξόδους των συναρτήσεων, η erlangb_iterative μας δίνει το αναμενόμενο αποτέλεσμα, η erlangb_factorial όμως μας επιστρέφει NaN (Not a Number). Ο λόγος γι' αυτό είναι το γεγονός ότι η μέθοδος με το παραγοντικό χρειάζεται να υπολογίσει αριθμούς που οδηγούν το πρόγραμμα σε υπερχείλιση (1024!), οπότε αποτυγχάνει.

(4)

(α)

Θεωρούμε ως πρότυπο τον πιο απαιτητικό χρήστη, επομένως, ισοδύναμα ο κάθε χρήστης χρησιμοποιεί τη γραμμή του 23 λεπτά ανά ώρα. Άρα η συνολική ένταση του φορτίου που καλείται να εξυπηρετηθεί από το δίκτυο της εταιρείας είναι:

$$\rho = 200\frac{23}{60} \implies \boxed{\rho = 76.67\ Erlangs}$$

(β)

Αξιοποιώντας την συνάρτηση erlangb_iterative που υλοποιήσαμε προηγουμένως, χρησιμοποιούμε τον ακόλουθο κώδικα και παράγουμε τη ζητούμενη γραφική:

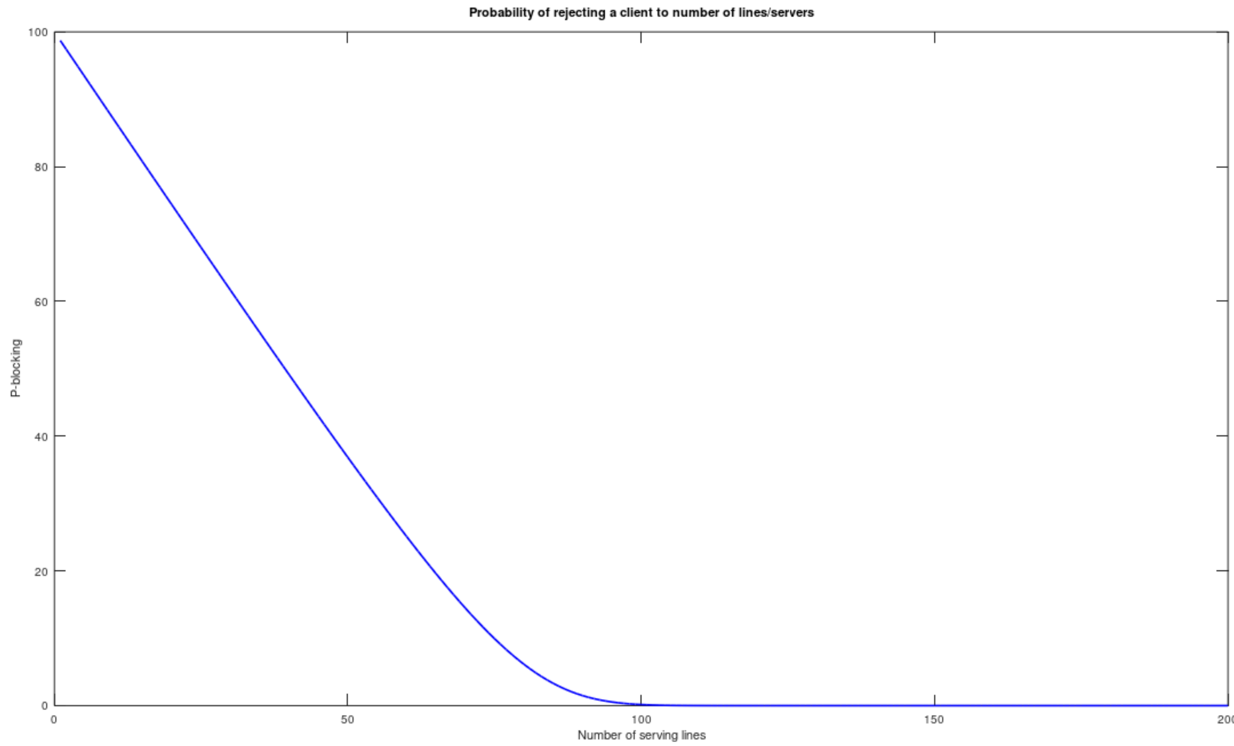
```
clc;
clear all;
close all;

# Task 4
#(4)(β)

r = 200 * (23/60);
c = 1 : 200;
P_blocking = zeros(1, 200);

for i = 1 : 200
  P_blocking(i) = erlangb_iterative(r, i);
endfor

plot(c, P_blocking, "b", "linewidth", 1.3)
xlabel("Number of serving lines")
ylabel("P[blocking]")
title("Probability of rejecting a client to number of lines/servers")
```



(γ)

Το πλήθος των τηλεφωνικών γραμμών, που απαιτείται, έτσι ώστε η πιθανότητα απόρριψης τηλεφωνικής κλήσης να είναι μικρότερη του 1% είναι 92 γραμμές.

```
P[blocking] is less than 1% for 92 serving lines
```

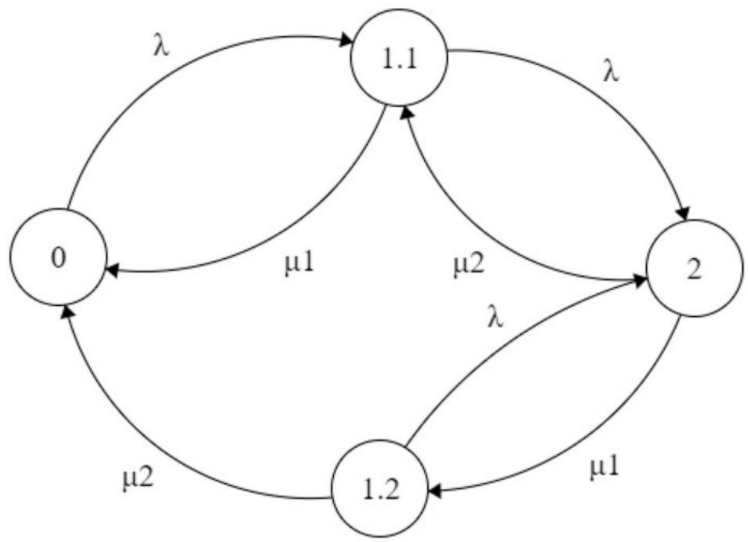
Το παραπάνω αποτέλεσμα δίνεται προσθέτοντας τον ακόλουθο κώδικα στον κώδικα του προηγούμενου ερωτήματος:

```
# Task 4
#(4)(γ)
i = find(P_blocking < 1, 1);
printf("P[blocking] is less than 1%% for %d serving lines\n", i - 1)
```

<u>Σύστημα εξυπηρέτησης με δύο ανόμοιους εξυπηρετητές:</u>

(1)

Το ζητούμενο διάγραμμα των ρυθμών μεταβάσεων του συστήματος είναι:



(α)

Θα υπολογίσουμε τώρα τις εργοδικές πιθανότητες. Ισχύουν οι ακόλουθες εξισώσεις ι-σορροπίας:

- $\lambda P_0 = \mu_1 P_{1.1} + \mu_2 P_{1.2}$
- $(\lambda + \mu_1)P_{1.1} = \lambda P_0 + \mu_2 P_2$
- $(\lambda + \mu_2)P_{1.2} = \mu_1 P_2$
- $\lambda(P_{1.1} + P_{1.2}) = (\mu_1 + \mu_2)P_2$
- $P_0 + P_{1.1} + P_{1.2} + P_2 = 1$

Αντικαθιστώντας στις παραπάνω εξισώσεις για $\lambda = 1\ \pi\varepsilon\lambda\acute{\alpha}\tau\eta\varsigma/sec$, $\mu_1 = 0.8\ \pi\varepsilon\lambda\acute{\alpha}\tau\varepsilon\varsigma/sec$ και $\mu_2 = 0.4$ πελάτες/sec και έχουμε:

- $P_0 = 0.8P_{1.1} + 0.4P_{1.2}$
- $1.8P_{1.1} = P_0 + 0.4P_2$
- $1.4P_{1.2} = 0.8P_2$
- $P_{1.1} + P_{1.2} = 1.2P_2$
- $P_0 + P_{1.1} + P_{1.2} + P_2 = 1$

Λύνοντας το παραπάνω σύστημα έχουμε:

$$\boxed{P_0 = 24.06\%}$$

$$\boxed{P_{1.1} = 21.70\%}$$

$$\boxed{P_{1.2} = 19.72\%}$$

$$\boxed{P_2 = 34.51\%}$$

(β)

Η πιθανότητα απόρριψης ενός πελάτη από το σύστημα είναι:

$$\boxed{P_{blocking} = P_2 = 34.51\%}$$

(γ)

Ο μέσος αριθμός πελατών στο σύστημα είναι:

$$E[n(t)] = \sum_{k=0}^{2} kP_k = 0 \cdot P_0 + 1 \cdot P_{1.1} + 1 \cdot P_{1.2} + 2 \cdot P_2 = 1 .1044$$

(2)

(α)

Συμπληρώνουμε αρχικά τα κενά του δοσμένου κώδικα ως εξής:

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

Τρέχοντας τώρα την προσομοίωση (έχοντας τροποποιήσει κατάλληλα τις εξόδους ώστε να φαίνονται ξεκάθαρα τα αποτελέσματα) λαμβάνουμε τα ακόλουθα:

```
The ergodic Probability of P(0) = 24.882874%
The ergodic Probability P(1) = 21.486613%
The ergodic Probability P(2) = 19.528118%
The ergodic Probability P(3)/Rejection = 34.102395%
The mean number of clients in the system is: 1.092195
```

Παρατηρούμε ότι επιβεβαιώνονται τα θεωρητικά αποτελέσματά μας, με μία μικρή απόκλιση, η οποία οφείλεται στην ακρίβεια του κριτηρίου σύγκλισης.

(β)

Το κριτήριο σύγκλισης της που παρατηρούμε στην προσομοίωσή μας είναι το ακόλουθο:

```
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
```

Σύμφωνα με το κριτήριο αυτό, η προσομοίωση συγκλίνει όταν η διαφορά του μέσου αριθμού πελατών την τωρινή στιγμή (επανάληψη του εξωτερικού βρόχου) με την προηγούμενη είναι μικρότερη του 0.00001. Η σύγκριση αυτή γίνεται κάθε 1000 επαναλήψεις.