

**Ε.Μ.Π., Σχολή Η.Μ.& Μ.Υ.**

Ημερ/νια: 01-05-2021

**Συστήματα Αναμονής (Queuing Systems) - Ακαδ.Ετος 2020-2021**

**3η Ομάδα Ασκήσεων**

1. Προσομοίωση συστήματος M/M/1/10

Παραδοτέα: 05-05-2021

Του προπτυχιακού φοιτητή:

**Νικήτας Τσίννας,      ΑΜ : 03118187**

## Προσομοίωση Συστήματος M/M/1/10

- 1) Ακολουθεί ο κώδικας που χρησιμοποιήθηκε για την προσομοίωση του συστήματος M/M/1/10 καθώς και η έξοδος του debugging της προσομοίωσης για τις πρώτες 30 μεταβάσεις:

```
%M/M/1/10 simulation | debugging mode: ON
```

```
clc;
clear all;
close all;
(rand "seed", 1);

max_clients = 10;
lamdas = [1, 5, 10];
m = 5;
num_fig = 0; #number of figure for plotting

for j=1:1:1
    lamda = lamdas(j);
    threshold = lamda / (lamda + m); # threshold to determine arrival or departure
    total_arrivals = 0; # total arrivals in the system
    arrival = zeros (1,10+1); # number of arrivals in every state
    count = 0; # number of transitions
    state = 0; # state of queue
    P = zeros (1,10+1) # ergodic propability of every state
    index = 0; # index for points on plot of mean_clients
    previous_mean = 0; #keep previous mean_value

    while count >= 0
        if mod(count,1000) == 0
            for i=1:1:length(arrival) %calculate ergodotic propabilities of states
                P(i) = arrival(i) / total_arrivals;
            endfor
        end
    end
end
```

```

Pblocking(j) = P(max_clients+1); #the last state is equal to blocking a client

mean_clients = 0;
for i=1:1:length(arrival) %calculate mean of system
    mean_clients += (i-1)*P(i);
endfor

index +=1;
points(index) = mean_clients;

if (abs(mean_clients - previous_mean) < (0.001/100) || count >= 1000000)
    ET(j) = mean_clients / (lamda*(1-Pblocking(j)));
    break;
endif

previous_mean = mean_clients;

endif

if count <= 30
    disp("new transition");
    disp ("current system state: "), disp(state);
endif

count = count + 1;
random = rand(1); #generate random number (0,1) uniform distribution
if (state == 0 || random < threshold) %arrival

    if count <= 30
        disp ("next transition: arrival ");
    endif

    total_arrivals += 1;

```

```

    arrival(state + 1) += 1;

    if (state != 10) state += 1;
    else continue;
endif

if count <= 30
    disp ("total arrivals in state:"), disp(arrival(state));
endif

else # departure

    if count <= 30
        disp ("next transition: departure ");
    endif

    state -= 1;
endif

endwhile

num_fig += 1;
figure(num_fig)
plot (points);
title (strjoin({"Average number of clients in the system for lamda = ", num2str(lamda)}));
ylabel("number of clients");
xlabel("transitions * 1000");

num_fig += 1;
figure(num_fig)
bar(P);
title (strjoin({"Ergodotic Propabilities of system states for lamda = ", num2str(lamda)}));
ylabel("propability");

```

```

xlabel("state");

endfor

num_fig += 1;
figure(num_fig)
bar (Pblocking)
title ("Pblocking of system for lamda = 1,5,10 respectively")

num_fig += 1;
figure (num_fig)
bar (ET)
title ("mean delay time of system for lamda = 1,5,10 respectively")

```

Ενδεικτικό output των πρώτων 8 μεταβάσεων:

*(ο κώδικας εκτύπωσε 30 μεταβάσεις, ωστόσο παραλείπονται στην αναφορά για οικονομία έκτασης αρχείου)*

```

new transition
current system state:
0
next transition: arrival
total arrivals in state:
1
new transition
current system state:
1
next transition: arrival
total arrivals in state:
1
new transition

```

current system state:

2

next transition: departure

new transition

current system state:

1

next transition: departure

new transition

current system state:

0

next transition: arrival

total arrivals in state:

2

new transition

current system state:

1

next transition: departure

new transition

current system state:

0

next transition: arrival

total arrivals in state:

3

new transition

current system state:

1

next transition: departure

- 2) Από τα παραπάνω, η προσομοίωση φαίνεται να λειτουργεί σωστά. Οπότε απενεργοποιούμε τις γραμμές κώδικα που αφορούν το debugging θέτοντάς τες ως comments. Οπότε προκύπτει ο τελικός κώδικας που συμπεριλαμβάνει τις προσομοιώσεις του συστήματος και για τις 3 τιμές του  $\lambda = \{1, 5, 10\}$ :

```
%M/M/1/10 simulation

clc;
clear all;
close all;
(rand "seed", 1);

max_clients = 10;
lamdas = [1, 5, 10];
m = 5;
num_fig = 0; #number of figure for plotting

for j=1:1:3
    lamda = lamdas(j);
    threshold = lamda / (lamda + m); # threshold to determine arrival or departure
    total_arrivals = 0; # total arrivals in the system
    arrival = zeros (1,10+1); # number of arrivals in every state
    count = 0; # number of transitions
    state = 0; # state of queue
    P = zeros (1,10+1) # ergodic propability of every state
    index = 0; # index for points on plot of mean_clients
    previous_mean = 0; #keep previous mean_value
    points = 0; #points for mean number of clients figure (for plotting)

    while count >= 0
        if mod(count,1000) == 0
            for i=1:1:length(arrival) %calculate ergodotic propabilities of states
                P(i) = arrival(i) / total_arrivals;
            endfor
        end
        count = count + 1;
        state = state + 1;
        arrival(state) = arrival(state) + 1;
        if state == 10+1
            state = 0;
        end
    end
    num_fig = num_fig + 1;
end
for j=1:1:3
    figure(num_fig);
    plot(points, lamdas(j));
    title('Mean number of clients for lamda = ' + num2str(lamdas(j)));
    xlabel('Number of clients');
    ylabel('Mean number of clients');
    hold on;
end
```

```

Pblocking(j) = P(max_clients+1); #the last state is equal to blocking a client

mean_clients = 0;
for i=1:1:length(arrival) %calculate mean of system
    mean_clients += (i-1)*P(i);
endfor

index +=1;
points(index) = mean_clients;

if (abs(mean_clients - previous_mean) < (0.001/100) || count >= 1000000)
    ET(j) = mean_clients / (lamda*(1-Pblocking(j)));
    break;
endif

previous_mean = mean_clients;

endif

#if count <= 30
# disp("new transition");
# disp ("current system state: "), disp(state);
#endif

count = count + 1;
random = rand(1); #generate random number (0,1) uniform distribution
if (state == 0 || random < threshold) %arrival

    #if count <= 30
    # disp ("next transition: arrival ");
    #endif

    total_arrivals += 1;
    arrival(state + 1) += 1;

```



```

    if (state != 10) state += 1;
    else continue;
endif

    #if count <= 30
    #   disp ("total arrivals in state:"), disp(arrival(state));
    #endif

    else # departure

        #if count <= 30
        #   disp ("next transition: departure ");
        #   endif

        state -= 1;

    endif

endwhile

num_fig += 1;
figure(num_fig)
plot (points);
title (strjoin({"Average number of clients in the system for lamda = ", num2str(lamda)}));
ylabel("number of clients");
xlabel("transitions * 1000");

num_fig += 1;
figure(num_fig)
bar(P);
title (strjoin({"Ergodotic Propabilities of system states for lamda = ", num2str(lamda)}));
ylabel("propability");

```

```

xlabel("state");

endfor

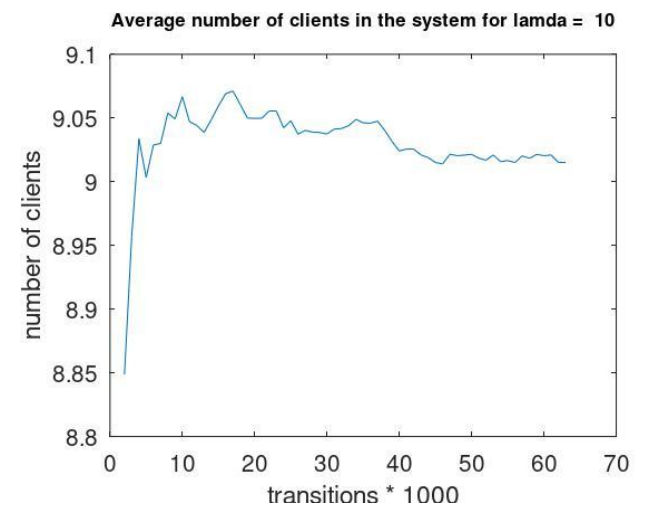
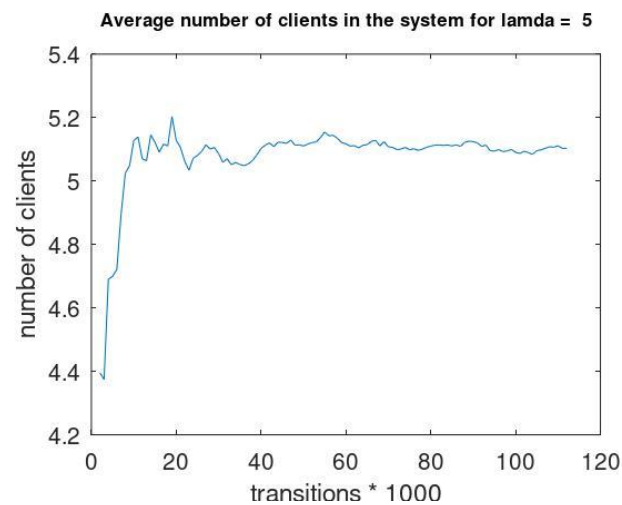
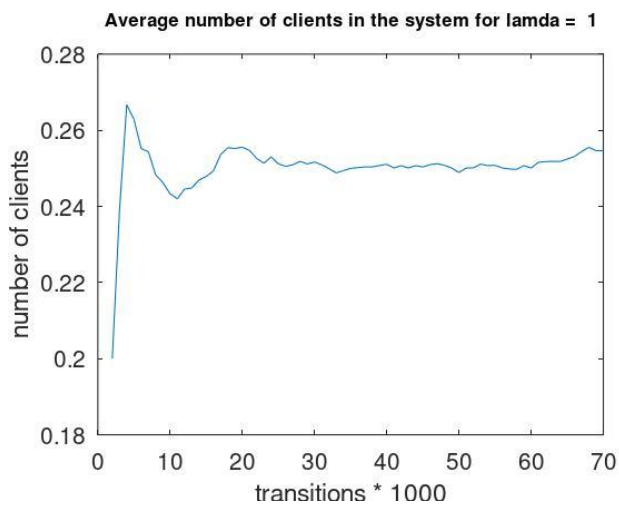
num_fig += 1;
figure(num_fig)
bar (Pblocking)
title ("Pblocking of system for lamda = 1,5,10 respectively")

num_fig += 1;
figure (num_fig)
bar (ET)
title ("mean delay time of system for lamda = 1,5,10 respectively")

```

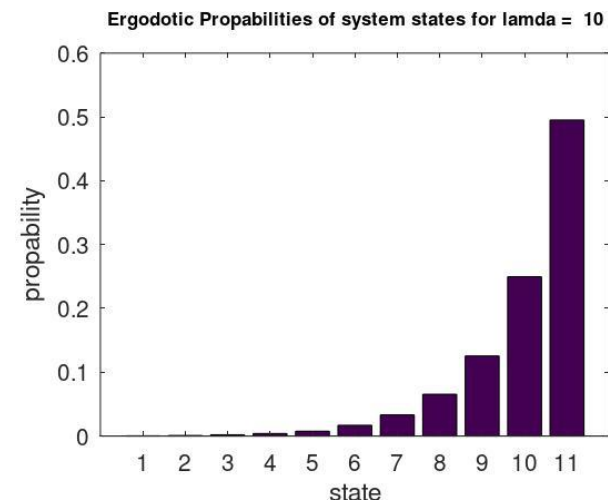
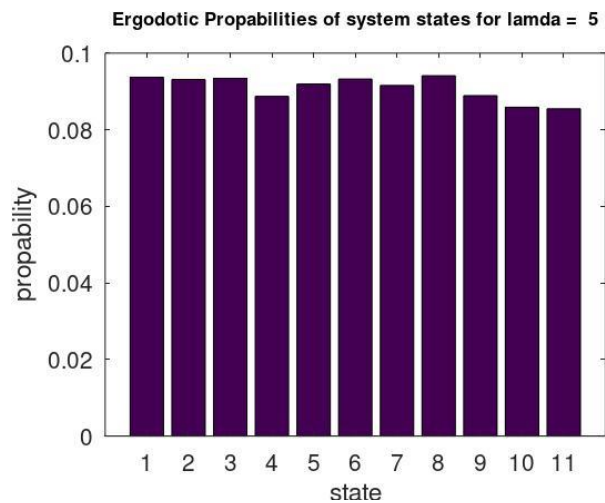
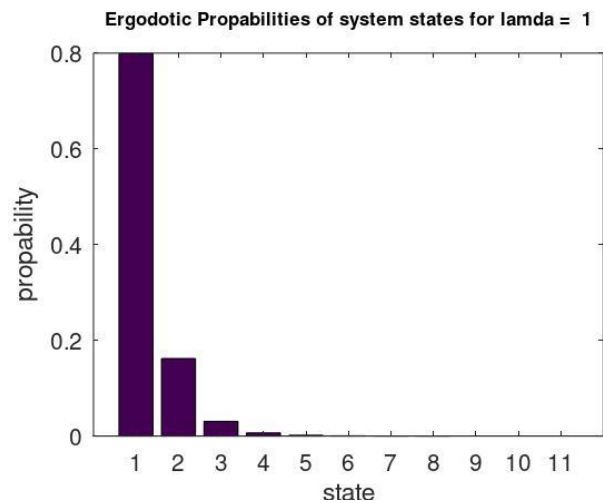
Εκτελώντας τον κώδικα προκύπτουν τα παρακάτω διαγράμματα:

*Εξέλιξη μέσου αριθμού πελατών στο σύστημα:*

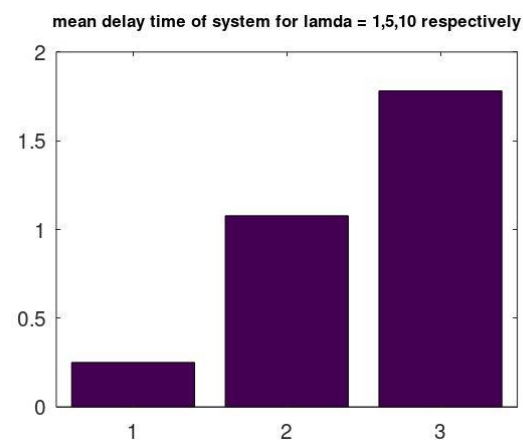
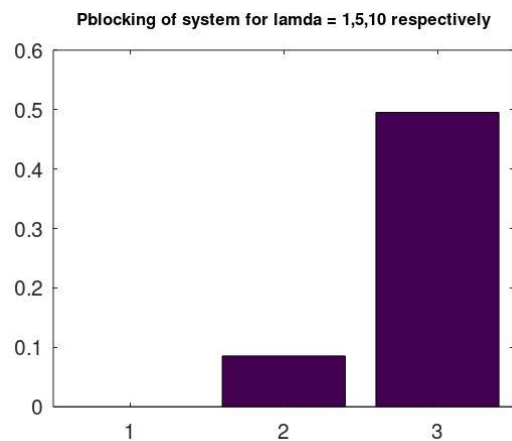


Εργοδικές πιθανότητες συστήματος:

\*στον άξονα x το state 1 αντιστοιχεί στο state  $\theta$  κ.ο.κ.



Πιθανότητα απόρριψης πελάτη και μέσος χρόνος καθυστέρησης στο σύστημα:



- 3) Μπορούμε μέσω παρατήρησης να συμπεράνουμε γενικευμένα πως όσο μειώνεται η απόλυτη τιμή της διαφοράς  $|\lambda - \mu|$ , δηλαδή όσο μικραίνει η διαφορά μεταξύ ρυθμού εξυπηρέτησης και ρυθμού άφιξης, τόσο μεγαλύτερο χρόνο χρειάζεται προσομοίωση ώστε να επέλθει το κριτήριο σύγκλισης και να ολοκληρωθεί η προσομοίωση. Επομένως για κάθε ακολουθία τυχαίων αριθμών στην προσομοίωση, μπορούμε να περιμένουμε πως τις περισσότερες μεταβάσεις θα χρειαστεί η προσομοίωση για  $\lambda=5$  (εφόσον  $\mu=5$  και άρα  $\lambda - \mu = 0$ ).

Θα μπορούσαμε με ασφάλεια να αγνοήσουμε τις πρώτες 10 χιλιάδες μεταβάσεις στο σύστημά μας καθώς αυτές χρειάζονται ώστε να σταθεροποιηθεί το σύστημα από την αρχική κατάσταση σε ένα επίπεδο γύρω από έναν αριθμό πελατών που χαρακτηρίζει το σύστημα.

Επίσης αξίζει να παρατηρήσουμε πως η πιθανότητα απόρριψης πελάτη καθώς και ο μέσος χρόνος καθυστέρησης του συστήματος αυξάνονται για μεγαλύτερες τιμές του  $\lambda$ .

- 4) Αυτό που θα αλλάζαμε θα ήταν απλώς το threshold, δηλαδή το κριτήριο άφιξης ή αποχώρησης. Πιο συγκεκριμένα θα αλλάζαμε την γραμμή που αφορά το threshold ως εξής:

$$\text{threshold} = \text{lamda} / (\text{lamda} + m * (\text{state} + 1) )$$

Και έπειτα θα την τοποθετούμε μέσα στο while () ώστε να γινόταν εκ νέου η ανάθεση της τιμής threshold σε κάθε κατάσταση συστήματος.