Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Ακ. έτος 2020-2021, 6ο Εξάμηνο: Συστήματα Αναμονής

Παναγιώτα-Μικαέλα Ξυλιά
ΑΜ:03118859

3η Ομάδα Ασκήσεων

Προσομοίωση συστήματος M/M/1/10

1.Debugging

Σε κάθε πίνακα η πρώτη στήλη αναφέρεται στις μεταβάσεις, η δεύτερη στην τρέχουσα κατάσταση, η τρίτη στις αφίξεις στην τρέχουσα κατάσταση και οι δύο τελευταίες στο αν είναι άφιξη ή αναχώρηση. Τρέχουμε τον κώδικα για λ=1,5,10.

(α)λ=1

debug_array =

0	0	0	1	0
2	1	1	0	1
0	0	0	1	0
4	1	2	1	0
5	2	1	0	1
6	1	2	0	1
0	0	0	1	0
8	1	3	1	0
9	2	2	0	1
10	1	3	1	0
11	2	3	0	1
12	1	3	0	1
0	0	0	1	0
14	1	4	0	1
0	0	0	1	0
16	1	5	0	1
0	0	0	1	0
18	1	6	0	1
0	0	0	1	0
20	1	7	0	1
0	0	0	1	0
22	1	8	1	0
23	2	4	0	1
24	1	8	0	1
0	0	0	1	0
26	1	9	0	1
0	0	0	1	0
28	1	10	0	1
0	0	0	1	0
30	1	11	1	0

(β)λ=5

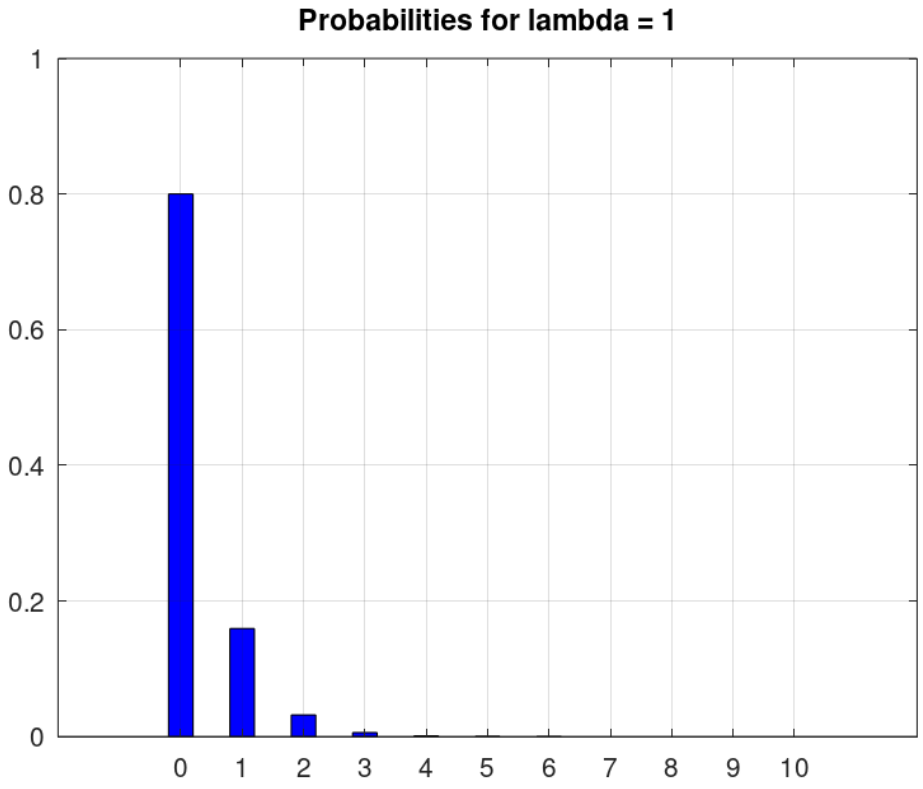
debug_array =

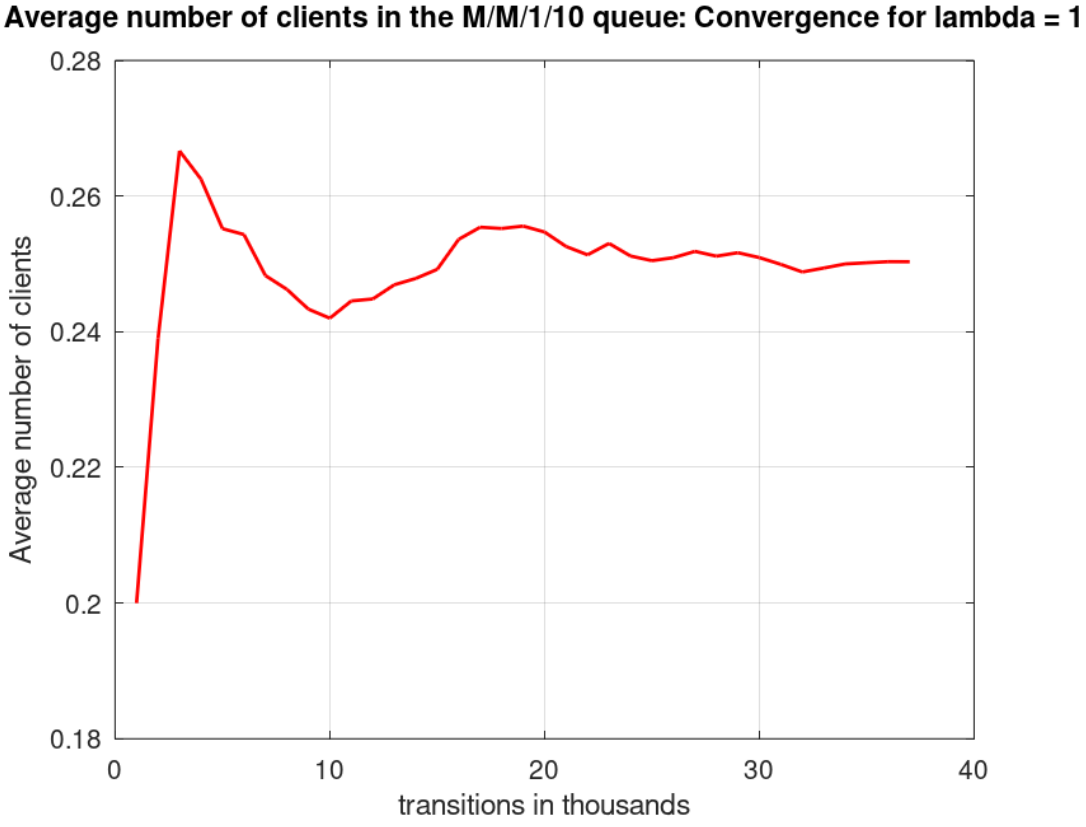
0	0	0	1	0
2	1	1	0	1
0	0	0	1	0
4	1	2	1	0
5	2	1	0	1
6	1	2	0	1
0	0	0	1	0
8	1	3	1	0
9	2	2	0	1
10	1	3	1	0
11	2	3	0	1
12	1	3	1	0
13	2	4	1	0
14	3	1	1	0
15	4	1	1	0
16	5	1	1	0
17	6	1	1	0
18	7	1	1	0
19	8	1	1	0
20	9	1	0	1
21	8	1	0	1
22	7	1	1	0
23	8	2	0	1
24	7	1	1	0
25	8	3	0	1
26	7	1	1	0
27	8	4	1	0
28	9	2	1	0
29	10	1	1	0
30	10	1	1	0

(γ) λ=10

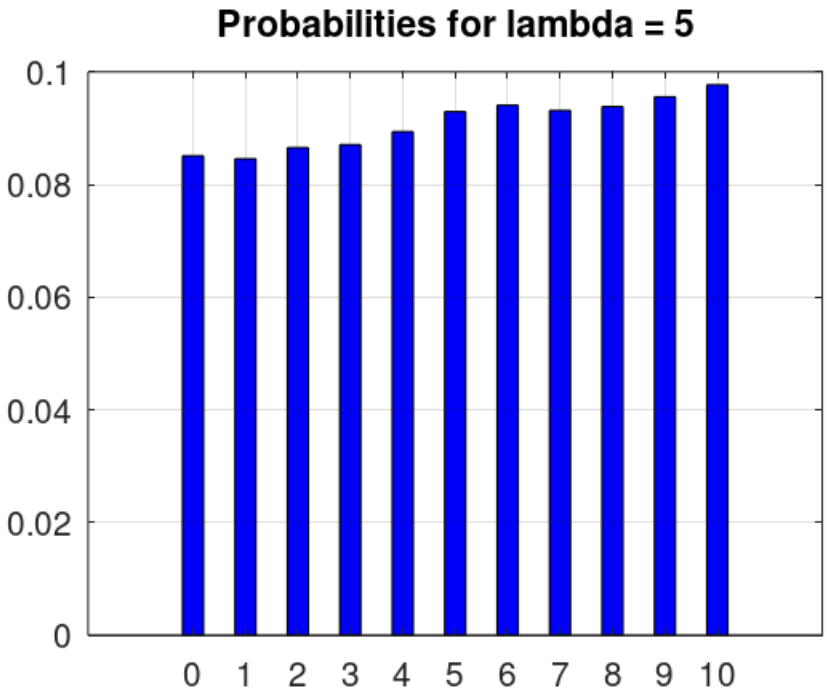
```
debug_array =

     0     0     0     1     0
     2     1     1     1     0
     3     2     1     0     1
     4     1     1     1     0
     5     2     2     1     0
     6     3     1     1     0
     7     4     1     0     1
     8     3     1     0     1
     9     2     2     1     0
    10     3     2     0     1
    11     2     2     1     0
    12     3     3     1     0
    13     4     2     1     0
    14     5     1     1     0
    15     6     1     1     0
    16     7     1     1     0
    17     8     1     1     0
    18     9     1     1     0
    19    10     1     1     0
    20    10     1     1     0
    21    10     1     1     0
    22    10     1     1     0
    23    10     1     1     0
    24    10     1     0     1
    25     9     1     1     0
    26    10     2     1     0
    27    10     2     0     1
    28     9     1     1     0
    29    10     3     1     0
    30    10     3     0     1
```

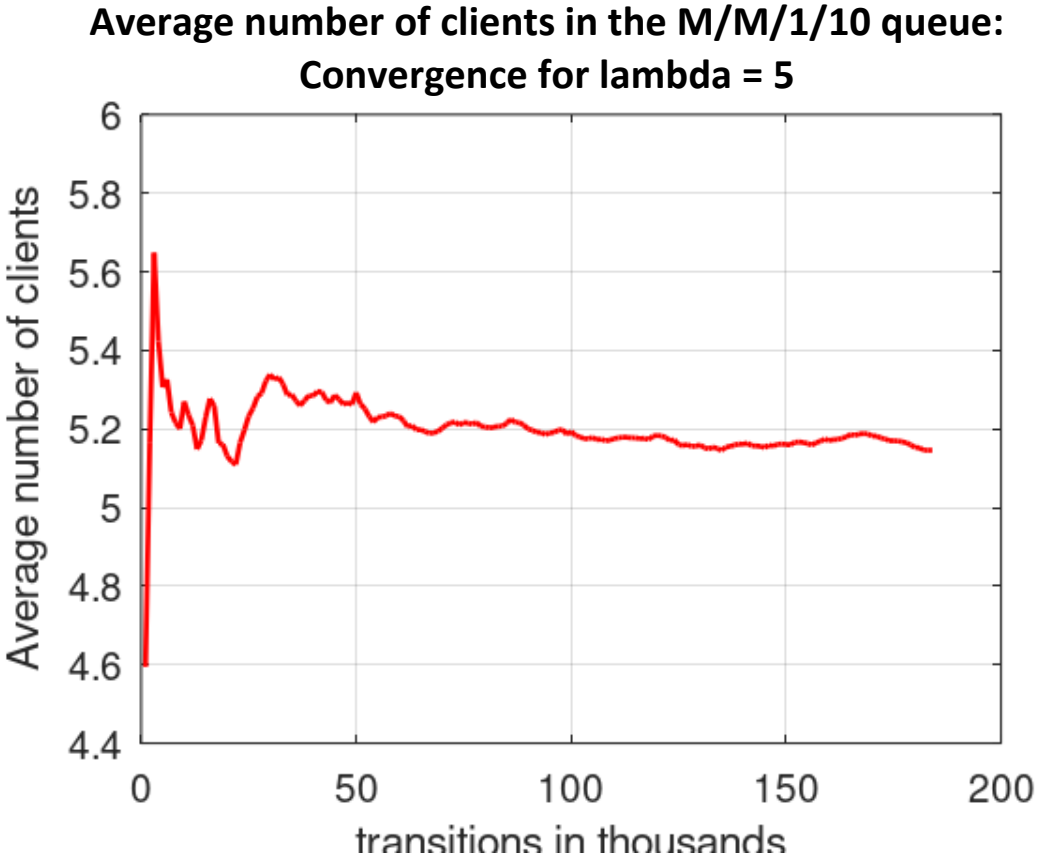
(2) (α)λ=1


Probabilities for lambda = 1
1
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8 9 10

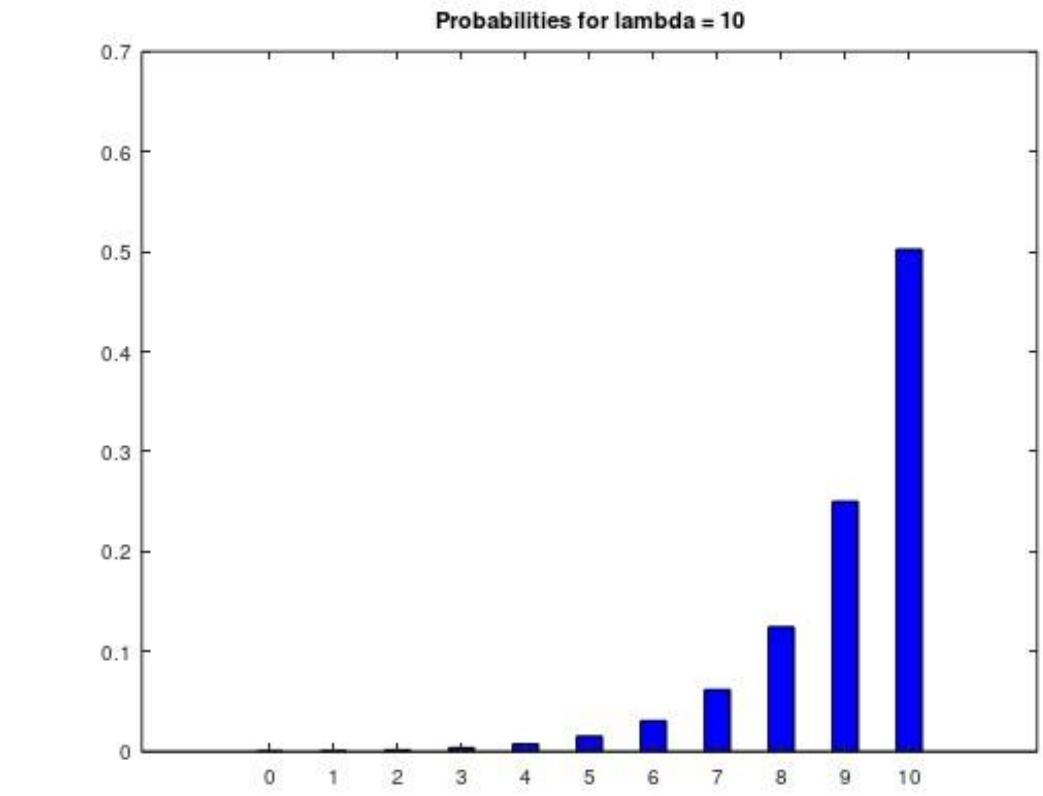

Average number of clients in the M/M/1/10 queue: Convergence for lambda = 1
0.28
0.26
0.24
0.22
0.2
0.18
Average number of clients
0
10
20
30
40
transitions in thousands

(β) λ=5

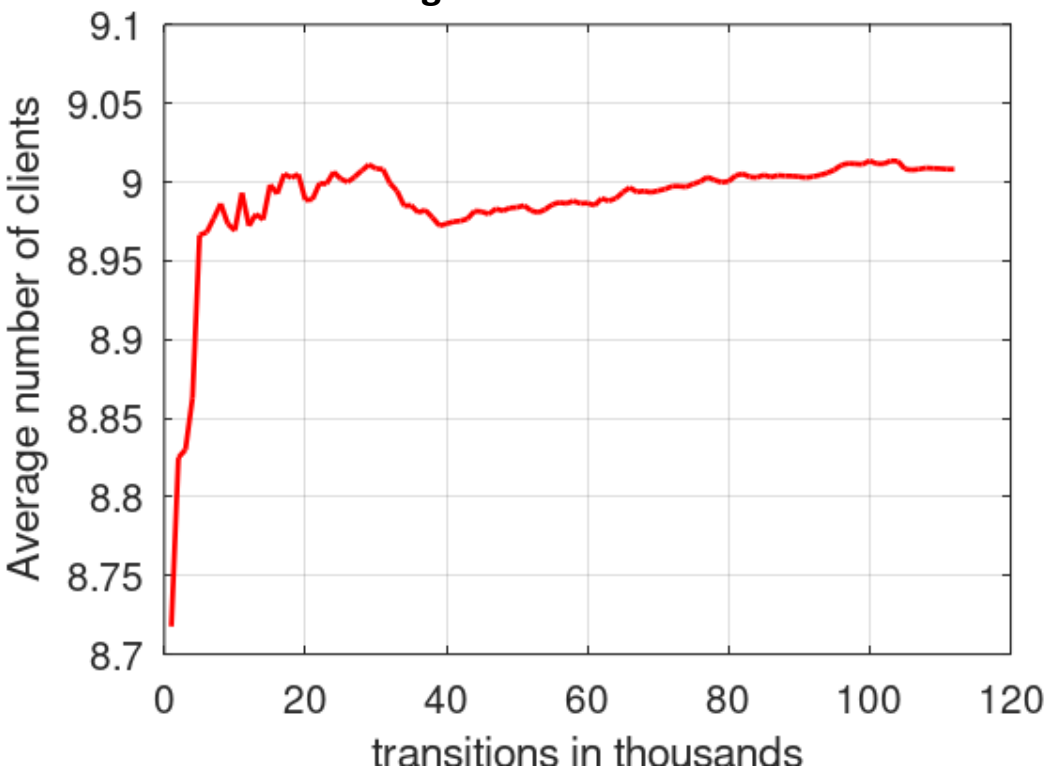

Probabilities for lambda = 5
0.1
0.08
0.06
0.04
0.02
0
0 1 2 3 4 5 6 7 8 9 10


Average number of clients in the M/M/1/10 queue:
Convergence for lambda = 5
Average number of clients
6
5.8
5.6
5.4
5.2
5
4.8
4.6
4.4
0
50
100
150
200
transitions in thousands

(γ) λ=10



Average number of clients in the M/M/1/10 queue: Convergence for lambda = 10



(3) Συγκρίνοντας τις γραφικές παραστάσεις, μπορούμε να παρατηρήσουμε πως για μεγαλύτερο λάμδα, η προσομοίωση συγκλίνει πιο γρήγορα. Μπορούμε να αγνοήσουμε τις πρώτες 10000 μεταβάσεις, λόγω της ύπαρξης μεταβατικού φαινομένου.

(4) Για μεταβλητό μέσο ρυθμό εξυπηρέτησης μi = μ * (i+1), θα χρειαζόταν να ενημερώνουμε συνέχεια το threshold έτσι ώστε να ισχύει ο τύπος $threshold = \frac{\lambda}{\lambda + current.state + 1}$

Κώδικας που χρησιμοποιήθηκε:

```
% M/M/1/10 simulation. We will find the probabilities of the first states.
% Note: Due to ergodicity, every state has a probability >0.

clc;
clear all;
close all;

rand('seed',1);

lambdaAll = [1,5,10];
for i = lambdaAll
    clear to_plot;
    arrivals = [0,0,0,0,0,0,0,0,0,0,0]
    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;
    mu = 5;
    threshold = (i)/((i) + mu);

    transitions = 0;
  %debugging
  %my_i = 0
  %debug_array = [];

   while  transitions >= 0 % && transitions < 30
        transitions = transitions + 1;

      %{
       my_i = my_i + 1
        if 0 < my_i < 31 && current_state > 0
            debug_array(my_i,1) = my_i;
            debug_array(my_i,2) = current_state;
            debug_array(my_i,3) = arrivals(current_state);
        endif
      %}

        if mod(transitions,1000) == 0 % check for convergence every 1000 transitions ste
ps
            index = index + 1;
            for j=1:1:length(arrivals)
```

```
                P(j) = arrivals(j)/total_arrivals; % calcuate the probability of every s
tate in the system
            endfor

            mean_clients = 0; % calculate the mean number of clients in the system
            for j=1:1:length(arrivals)
                mean_clients = mean_clients + (j-1).*P(j);
            endfor

            to_plot(index) = mean_clients;

            if abs(mean_clients -
 previous_mean_clients) < 0.00001 || transitions > 1000000 % convergence test
            break;
            endif

            previous_mean_clients = mean_clients;

        endif

        random_number = rand(1);
        if current_state == 0 || random_number < threshold % arrival

           %{
            if 0 < my_i < 31
                debug_array(my_i,4) = 1
            endif
            %}

            total_arrivals = total_arrivals + 1;

           % x = arrivals(current_state + 1) + 1
           % arrivals(current_state + 1) = x; % increase the number of arrivals in the c
urrent state

          arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
           if (current_state != 10)
                current_state = current_state + 1;
            endif
            else % departure
            %{
            if 0 < my_i < 31
                debug_array(my_i,5) = 1
            endif
            %}

            if current_state != 0 % no departure
            current_state = current_state - 1;
            endif
        endif
    endwhile

   % for j=1:1:length(arrivals)
   %    display(P(j));
   % endfor

    figure(1);
    plot(to_plot,"r","linewidth",1.3);
    title(strjoin({"Average number of clients in the M/M/1/10 queue: Convergence for lam
bda = ",num2str((i))},""));
    xlabel("transitions in thousands");
```

```
    ylabel("Average number of clients");


    figure(2);
    bar(0:1:(length(arrivals)-1),P,'b',0.4);
    title(strjoin({"Probabilities for lambda = ",num2str((i))},""));


endfor
```