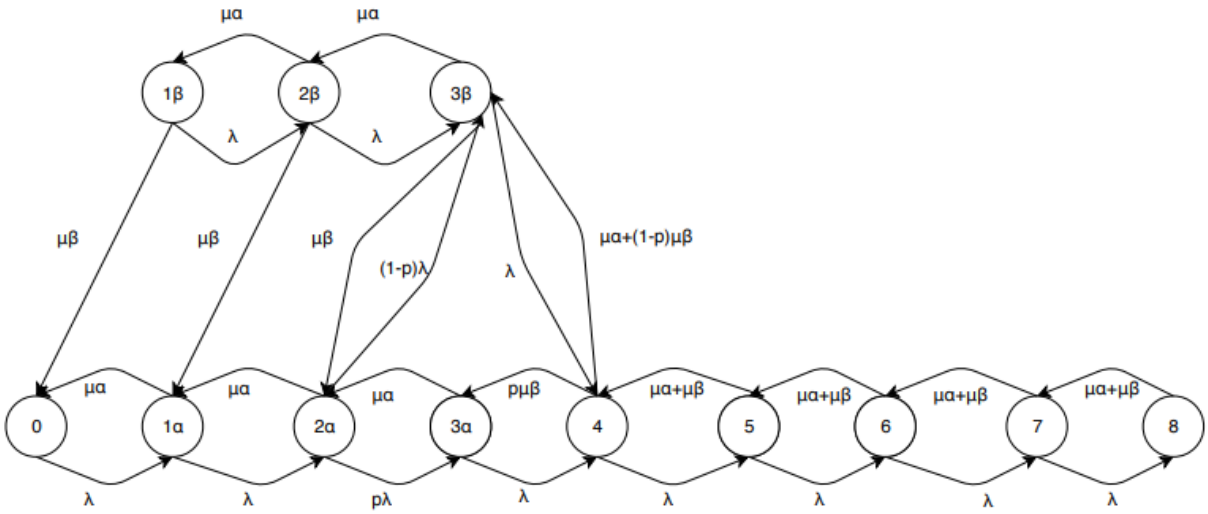
Άσκηση Προσομοίωσης

Συστήματα Αναμονής

Όνομα: Σταύρος
Επώνυμο: Σταύρου
ΑΜ: 03115701
Εξάμηνο: 6ο-ΗΜΜΥ
Ακαδημαϊκό Έτος: 2017-2018

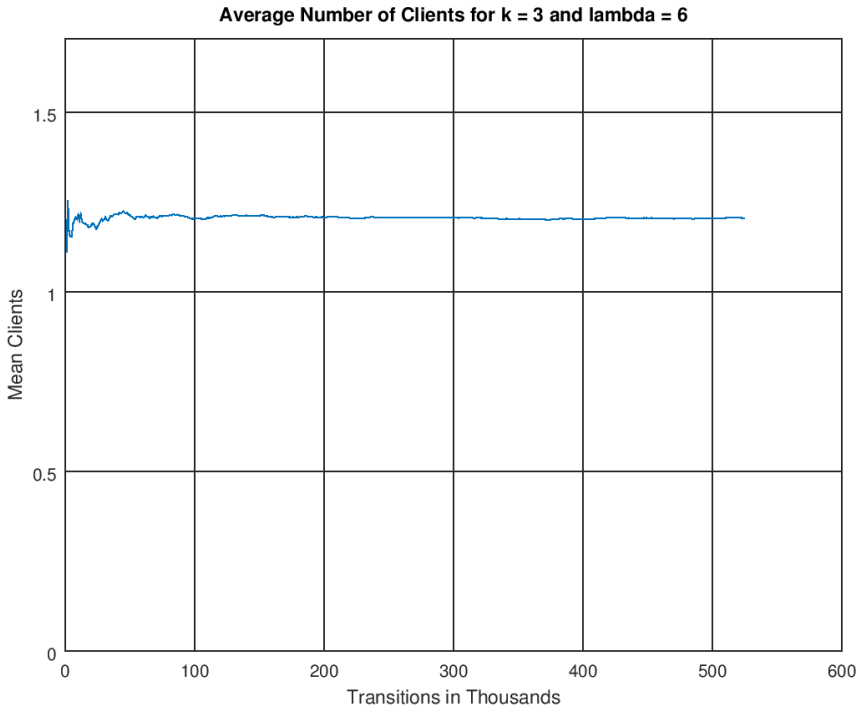
Με κατώφλι την κατάσταση 3 έχουμε το εξής διάγραμμα μεταβάσεων:

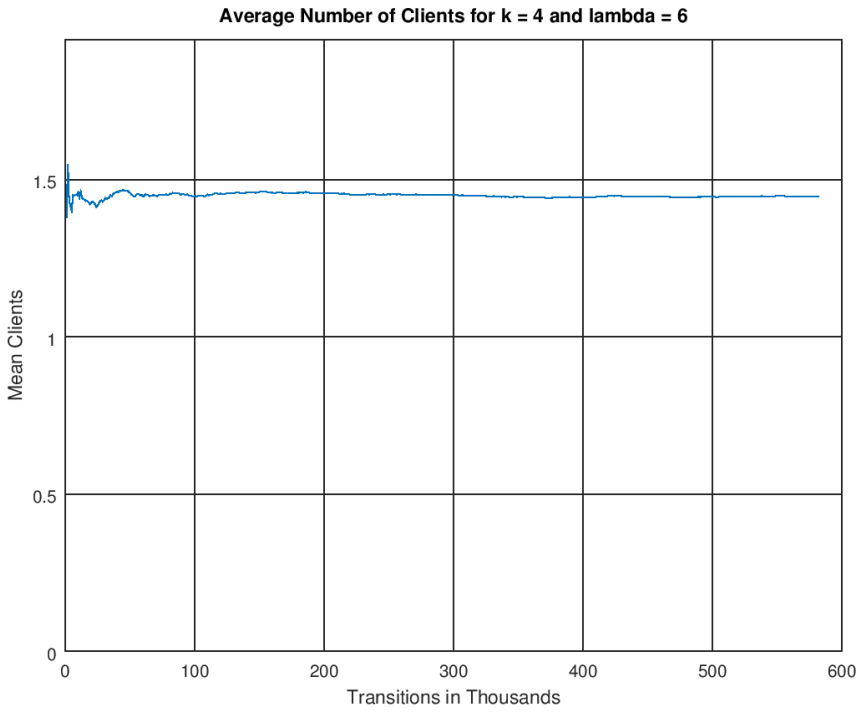


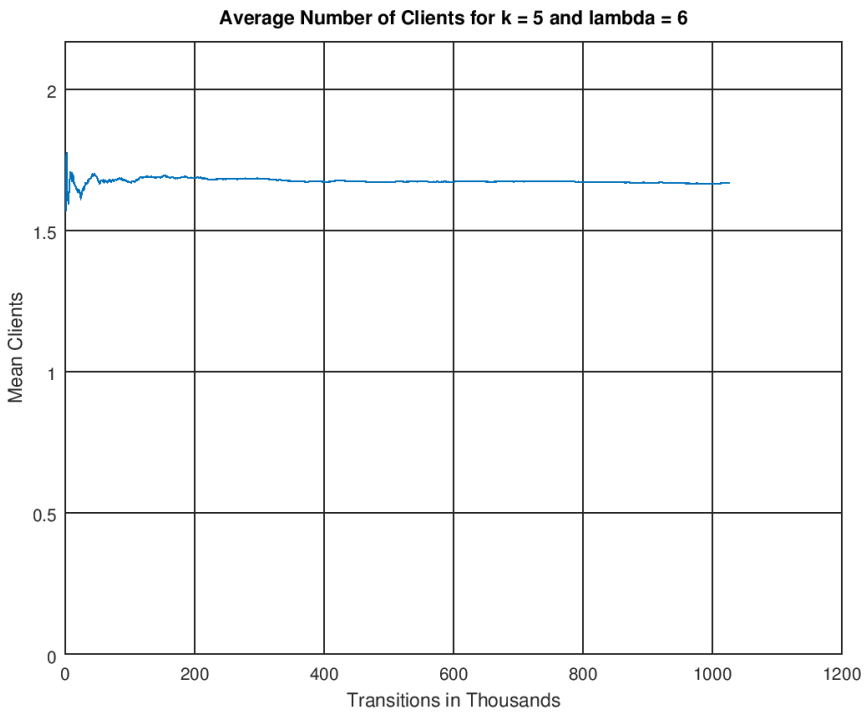
Ο κώδικας της προσομοίωσης βασίστηκε στο παραπάνω διάγραμμα μεταβάσεων. Αυτός επισυνάπτεται σε ξεχωριστό αρχείο και επίσης παρατίθεται στο παράρτημα στο τέλος. Επίσης, σαν κριτήριο σύγκλισης επιλέχθηκε η διαφορά 0.000001 ανάμεσα σε 2 διαδοχικές τιμές μέσων πελατών.

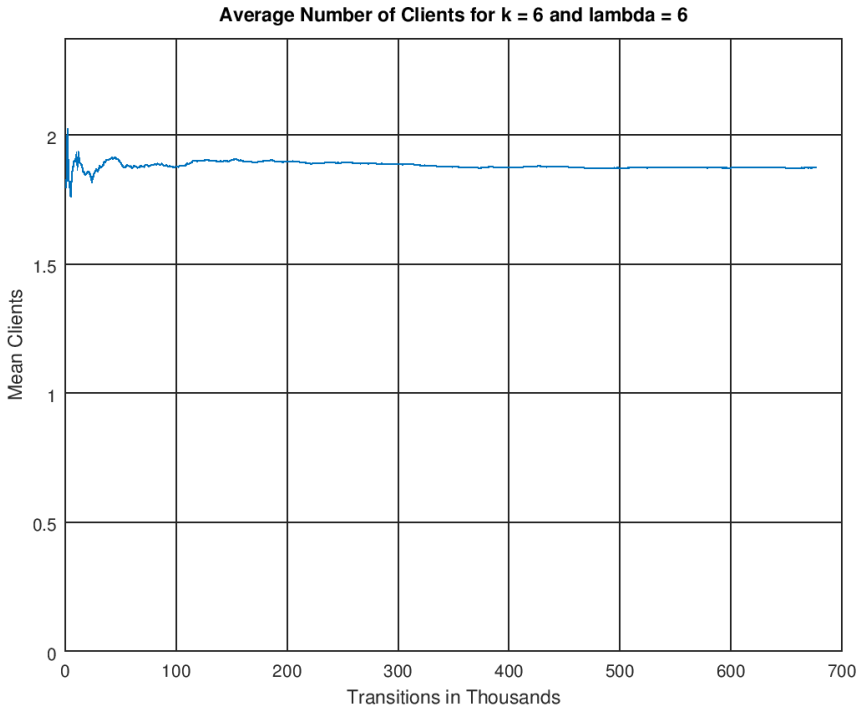
Ερώτημα 1

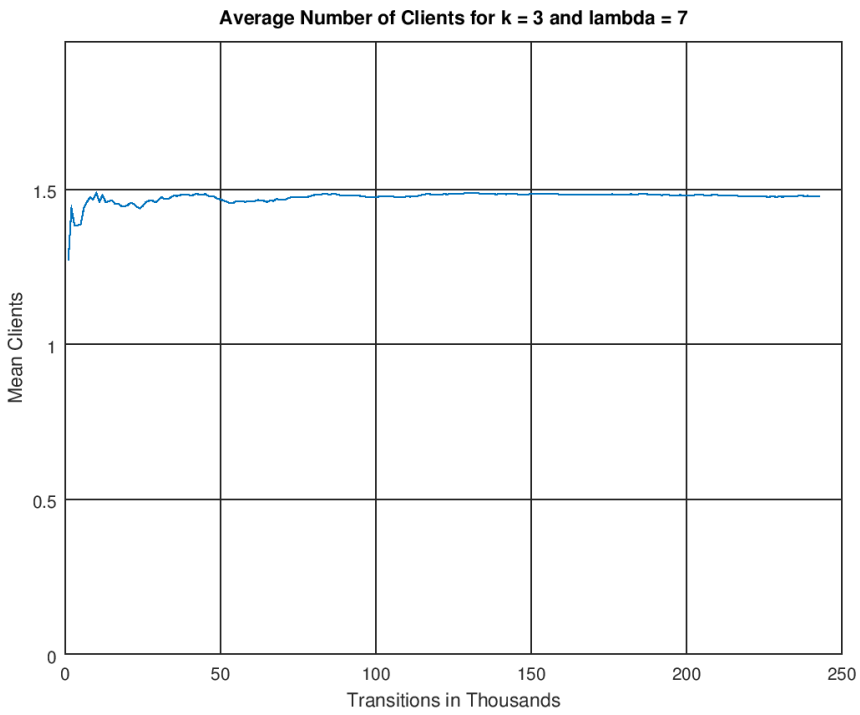
Η προσομοίωση στην Octave δίνει τις εξής 12 γραφικές παραστάσεις που απεικονίζουν την εξέλιξη του μέσου αριθμού πελατών στο σύστημα, μέχρι αυτό να φτάσει σε σύγκλιση.

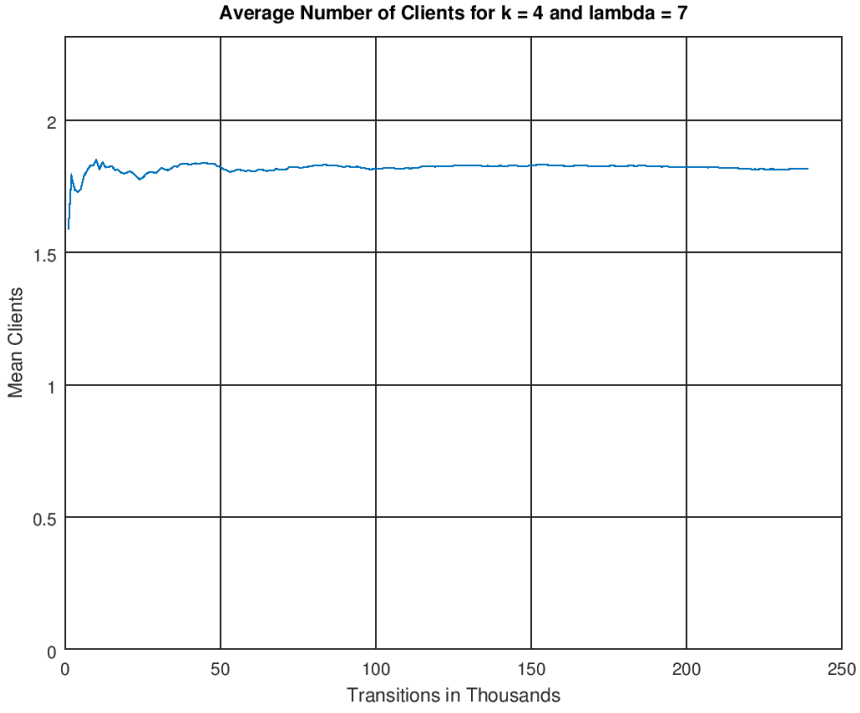


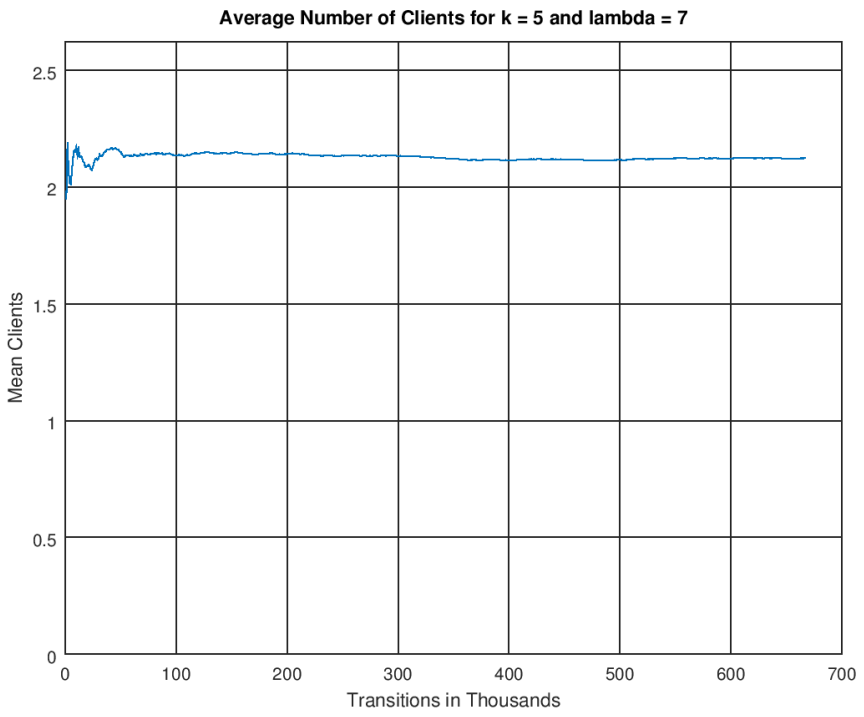


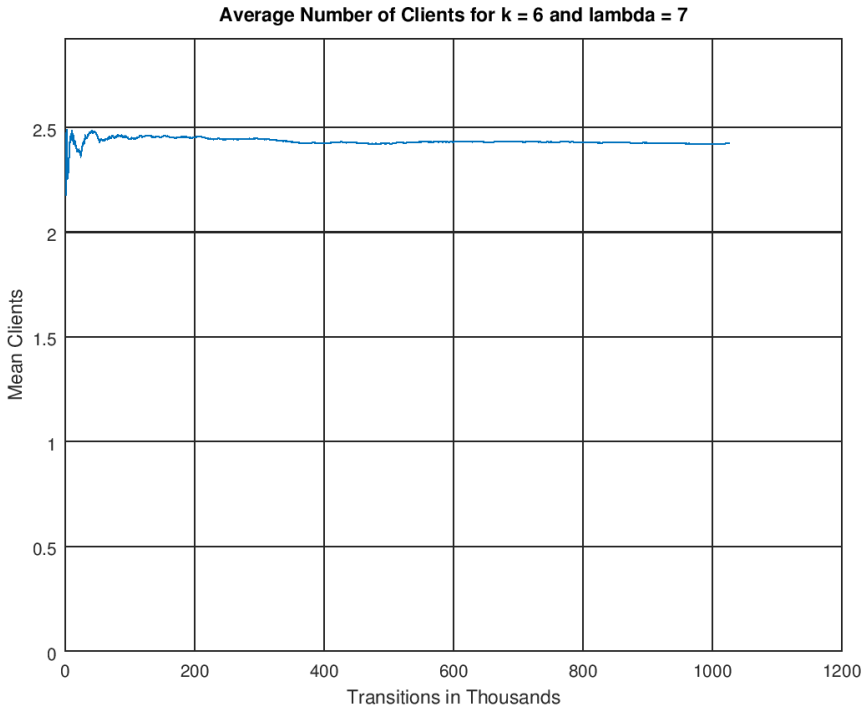

Average Number of Clients for k = 5 and lambda = 6
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
0
200
400
600
800
1000
1200

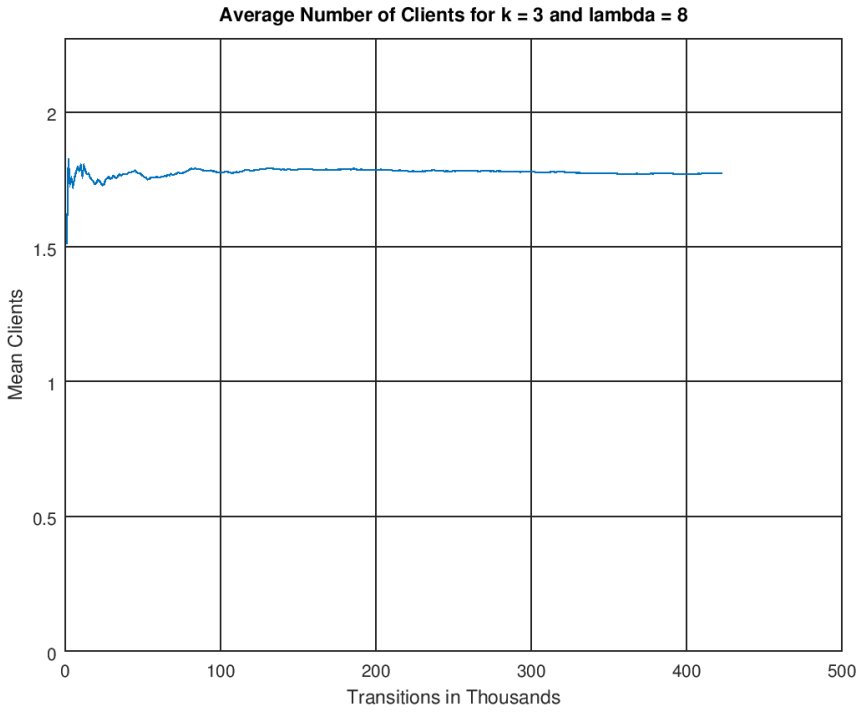

Average Number of Clients for k = 6 and lambda = 6
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
0
100
200
300
400
500
600
700

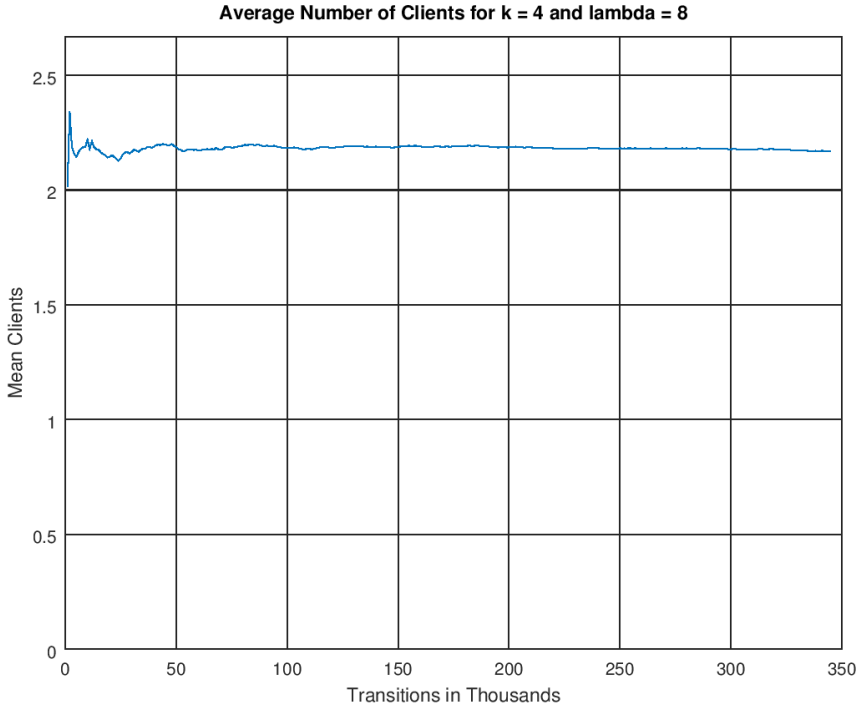

Average Number of Clients for k = 3 and lambda = 7
Mean Clients
0
0.5
1
1.5
0
50
100
150
200
250
Transitions in Thousands

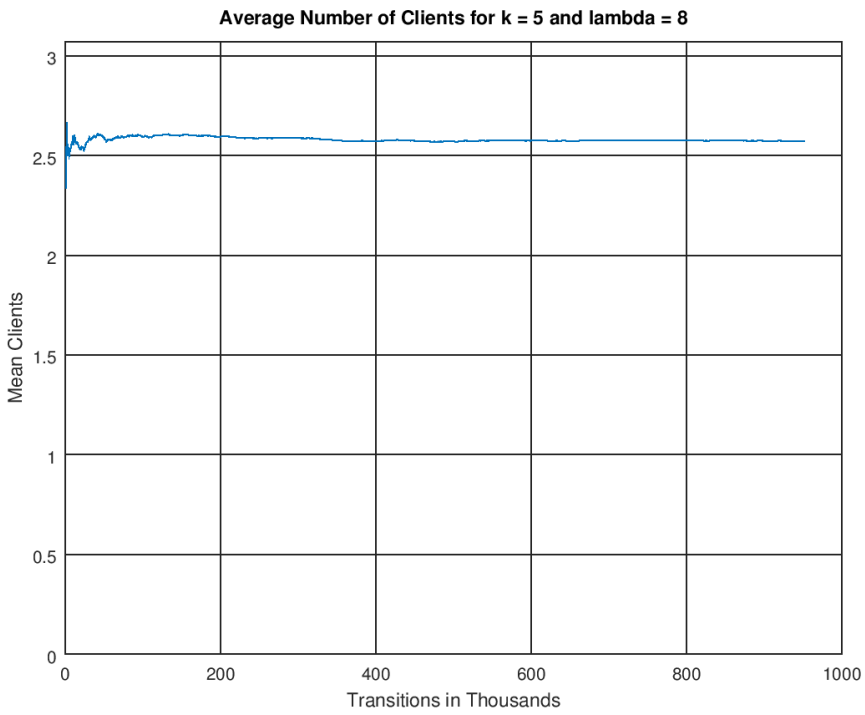

Average Number of Clients for k = 4 and lambda = 7
Mean Clients
0
0.5
1
1.5
2
0
50
100
150
200
250
Transitions in Thousands

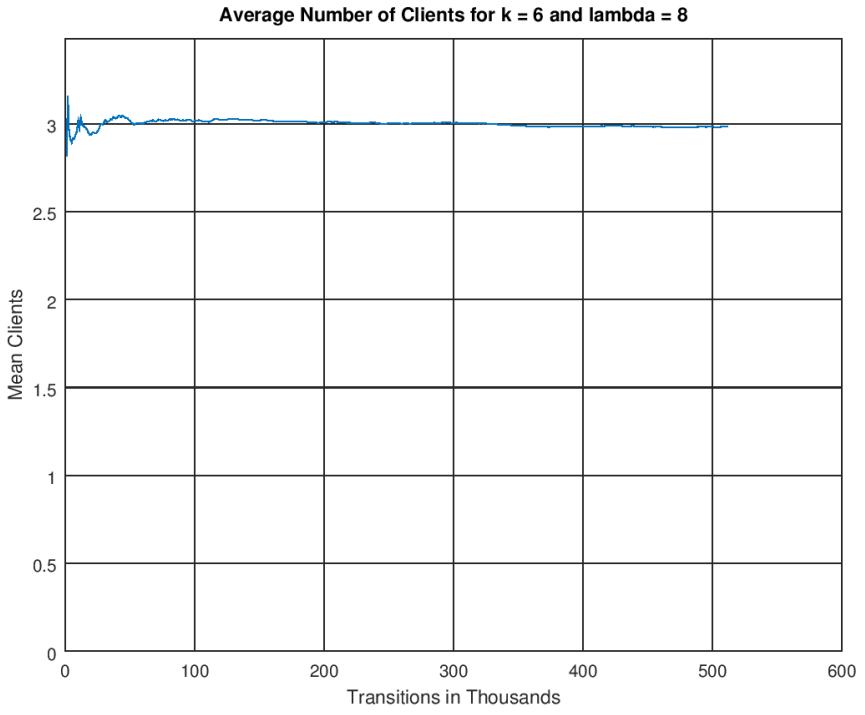

Average Number of Clients for k = 5 and lambda = 7
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
2.5
0
100
200
300
400
500
600
700


Average Number of Clients for k = 6 and lambda = 7
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
2.5
0
200
400
600
800
1000
1200


Average Number of Clients for k = 3 and lambda = 8
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
0
100
200
300
400
500


Average Number of Clients for k = 4 and lambda = 8
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
2.5
0
50
100
150
200
250
300
350


Average Number of Clients for k = 5 and lambda = 8
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
2.5
3
0
200
400
600
800
1000


Average Number of Clients for k = 6 and lambda = 8
Mean Clients
Transitions in Thousands
0
0.5
1
1.5
2
2.5
3
0
100
200
300
400
500
600

Ακόμη παίρνουμε το εξής από την Octave (που χρησιμοποιούμε για απάντηση των ερωτημάτων 2 και 3):

```
For k = 3 and lambda = 6 ThroughputA/ThroughputB is 4.4783
Average Number of Clients for k = 3 and lambda = 6 is 1.2058
For k = 4 and lambda = 6 ThroughputA/ThroughputB is 7.419
Average Number of Clients for k = 4 and lambda = 6 is 1.4467
For k = 5 and lambda = 6 ThroughputA/ThroughputB is 11.3897
Average Number of Clients for k = 5 and lambda = 6 is 1.6688
For k = 6 and lambda = 6 ThroughputA/ThroughputB is 17.4121
Average Number of Clients for k = 6 and lambda = 6 is 1.8745
For k = 3 and lambda = 7 ThroughputA/ThroughputB is 3.4889
Average Number of Clients for k = 3 and lambda = 7 is 1.4782
For k = 4 and lambda = 7 ThroughputA/ThroughputB is 5.0794
Average Number of Clients for k = 4 and lambda = 7 is 1.8173
For k = 5 and lambda = 7 ThroughputA/ThroughputB is 7.2117
Average Number of Clients for k = 5 and lambda = 7 is 2.1238
For k = 6 and lambda = 7 ThroughputA/ThroughputB is 9.9273
Average Number of Clients for k = 6 and lambda = 7 is 2.4247
For k = 3 and lambda = 8 ThroughputA/ThroughputB is 2.8297
Average Number of Clients for k = 3 and lambda = 8 is 1.7736
For k = 4 and lambda = 8 ThroughputA/ThroughputB is 3.8896
Average Number of Clients for k = 4 and lambda = 8 is 2.1692
For k = 5 and lambda = 8 ThroughputA/ThroughputB is 5.081
Average Number of Clients for k = 5 and lambda = 8 is 2.5723
For k = 6 and lambda = 8 ThroughputA/ThroughputB is 6.507
Average Number of Clients for k = 6 and lambda = 8 is 2.9862
```

Ερώτημα 2

Βάση του πιο πάνω έχουμε τον εξής πίνακα για τον μέσο αριθμό πελατών μετά τη σύγκλιση:

	k = 3	k = 4	k = 5	k =6
λ = 6	1.2058	1.4467	1.6688	1.8745
λ = 7	1.4782	1.8173	2.1238	2.4247
λ = 8	1.7736	2.1692	2.5723	2.9862

Ερώτημα 3

Παρατηρούμε πως για λ = 6 έχουμε $\frac{\gamma_\alpha}{\gamma_\beta} > 5$ για k = 4, για λ = 7 έχουμε $\frac{\gamma_\alpha}{\gamma_\beta} > 5$ για k = 4 και για λ = 8 έχουμε $\frac{\gamma_\alpha}{\gamma_\beta} > 5$ για k = 5. Θα πειραματιστώ με τις τιμές $p = \frac{1}{3}$ και $p = \frac{2}{3}$. Έχουμε τον εξής πίνακα για τις τιμές του λόγου $\frac{\gamma_\alpha}{\gamma_\beta}$.

	$p = \frac{1}{3}$	$p = \frac{2}{3}$
λ = 6, k = 4	6.924	7.9722
λ = 7, k = 4	4.8955	5.457
λ = 8, k = 5	4.9129	5.3329

Παρατηρούμε ότι μείωση της πιθανότητας p (δηλαδή ο β ενεργοποιείται πιο συχνά) φέρνει μείωση του λόγου, που είναι λογικό, καθώς ο β εξυπηρετεί περισσότερους πελάτες από πριν. Από την άλλη αύξηση του p (δηλαδή ο β ενεργοποιείται πιο αραιά) φέρνει αύξηση του λόγου, καθώς ο β εξυπηρετεί ακόμα λιγότερους πελάτες από πριν.

Ερώτημα 4

Για ακριβότερη απάντηση στα παρακάτω κάθε υπο-προσομοίωση είχε αρχικοποιημένο το seed τυχαίων αριθμών στο 1234. Με αυτόν τον τρόπο «απαλείψαμε» την τυχαιότητα από το σύστημα.

I. Παρόλο που αυτό δεν φαίνεται να ισχύει πάντα, εν τούτοις, φαίνεται η αύξηση του κατωφλίου να φέρνει μείωση της ταχύτητας σύγκλισης της προσομοίωσης. Αυτό ίσως να οφείλεται στο ότι αύξηση του κατωφλίου φέρνει επιπλέον καταστάσεις στο σύστημα.

II. Βλέπουμε πως αύξηση του κατωφλίου φέρνει μείωση της απόδοσης (ύπαρξη περισσοτέρων πελατών στο σύστημα). Αυτό φαίνεται λογικό για το λόγο ότι με ένα μεγαλύτερο κατώφλι, έχουμε τον 2^{ο} εξυπηρετητή του συστήματος ανενεργό για περισσότερο χρόνο. Αυτό προκαλεί και την εξυπηρέτηση λιγότερων πελατών στο ίδιο χρονικό διάστημα.

ΠΑΡΑΡΤΗΜΑ

```
clc;
clear all;
close all;

mu = 8;
figure_count = 1;
prob = 0.5;

% montelopoioume to sistima. Oi katastaseis einai k + 9. Exoume ti xrisi enos
tixaiou arithmou\


for lambda = 6:1:8
  for k = 3:1:6
    % exoume sinolika 12 epanalipseis, arxikopoioume oles tis metavlites pou
xreiazomaste
    rand("seed",1234); % to idio seed se kathe ipo-prosomiosi gia na mporoume
na elegksoume tin taxitita sigklisis
    clear to_plot;
    index = 0;
    transitions = 0;
    sigklisi = 0;
    total_arrivals = 0;
    P = zeros(1, k + 9);
    arrivals = zeros(1, k + 9);
    cur_state = 1;
    prev_mean = 0;
    threshold0 = lambda / (lambda + mu);
    threshold1 = lambda / (lambda + 2 * mu);
    threshold2 = (lambda + mu) / (lambda + 2 * mu);
    threshold3 = threshold0 * prob;
    while !sigklisi
      transitions++;
      decision = rand(1);
      if cur_state == 1
        arrivals(1)++;
        total_arrivals++;
        cur_state++;
      elseif (cur_state == k) && (decision > threshold0)
        cur_state--;
      elseif (cur_state == k) && (decision < threshold3)
        arrivals(cur_state)++;
        total_arrivals++;
        cur_state++;
      elseif (cur_state == k) && (decision > threshold3) && (decision <
threshold0)
        arrivals(cur_state)++;
        total_arrivals++;
        cur_state = k + 9;
      elseif (cur_state == k + 2) && (decision < threshold1)
        arrivals(cur_state)++;
        total_arrivals++;
        cur_state++;
      elseif (cur_state == k + 2) && (decision > threshold1) && (decision <
(lambda + prob * mu) / (lambda + 2 * mu))
```

```
        cur_state--;
      elseif (cur_state == k + 2) && (decision > (lambda + prob * mu) /
(lambda + 2 * mu))
        cur_state = k + 9;
      elseif (cur_state == 9) && (decision < threshold1)
        arrivals(cur_state)++;
        total_arrivals++;
      elseif (cur_state == 9) && (decision > threshold1)
        cur_state--;
      elseif (cur_state == k + 9) && (decision < threshold1)
        total_arrivals++;
        arrivals(cur_state)++;
        cur_state = k + 2;
      elseif (cur_state == k + 9) && (decision > threshold1) && (decision <
threshold2)
        cur_state = k;
      elseif (cur_state == k + 9) && (decision > threshold2)
        cur_state--;
      elseif (cur_state == 10) && (decision < threshold0)
        total_arrivals++;
        arrivals(cur_state)++;
        cur_state++;
      elseif (cur_state == 10) && (decision > threshold0)
        cur_state = 1;
      elseif (cur_state == k + 1) && (decision < threshold0)
        arrivals(cur_state)++;
        total_arrivals++;
        cur_state++;
      elseif (cur_state == k + 1) && (decision > threshold0)
        cur_state--;
      elseif (cur_state < k) && (decision < threshold0)
        arrivals(cur_state)++;
        total_arrivals++;
        cur_state++;
      elseif (cur_state < k) && (decision > threshold0)
        cur_state--;
      elseif (cur_state < 9) && (decision < threshold1)
        arrivals(cur_state)++;
        total_arrivals++;
        cur_state++;
      elseif (cur_state < 9) && (decision > threshold1)
        cur_state--;
      elseif (cur_state > 10) && (decision < threshold1)
        total_arrivals++;
        arrivals(cur_state)++;
        cur_state++;
      elseif (cur_state > 10) && (decision > threshold1) &&(decision <
threshold2)
        cur_state -= 9;
      elseif (cur_state > 10) && (decision > threshold2)
        cur_state--;
      endif

% kathe  1000 metavaseis elegxoume gia sigklisi
      if mod(transitions,1000) == 0
        index++;
```

```
        for i=1:1:length(arrivals)
          P(i) = arrivals(i) ./ total_arrivals;
        endfor

        mean_clients = 0;
        for i=1:1:9
          mean_clients += P(i) .* (i - 1);
        endfor
        for i = 10:1:length(arrivals)
          mean_clients += P(i) .* (i - 9);
        endfor
        to_plot(index) = mean_clients;
        if abs(mean_clients - prev_mean) < 0.000001
          sigklisi = 1;
        endif

        prev_mean = mean_clients;
      endif
    endwhile

% ipologizoume ta throughput kai parousiazoume ta apotelesmata mas gia kathe
sindiasmo lambda kai k
    utilA = 1 - P(1) - P(10);
    utilB = 1;
    for i = 1:1:k+1
      utilB -= P(i);
    endfor
    tptA = utilA * mu;
    tptB = utilB * mu;
    logos = tptA / tptB;
    disp(cstrcat("For k = ", num2str(k), " and lambda = ", num2str(lambda), "
ThroughputA/ThroughputB is ", num2str(logos)));
    title_str = cstrcat("Average Number of Clients for k = ", num2str(k), "
and lambda = ", num2str(lambda));
    disp(cstrcat(title_str, " is ", num2str(mean_clients)));
    figure(figure_count++);
    plot(to_plot);
    grid on;
    title(title_str);
    ylim([0 mean_clients+0.5]);
    xlabel("Transitions in Thousands");
    ylabel("Mean Clients");
  endfor
endfor
```