

Συστήματα Αναμονής

Εργασία Προσομοίωσης

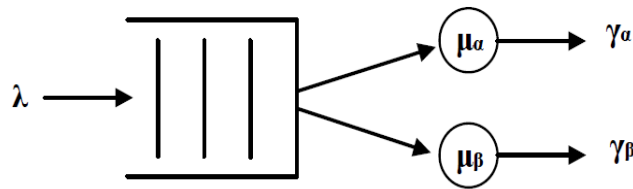
Κατσαραγάκης Μανώλης
el13059

Κυτέας Απόστολος
el13209

13 Ιουλίου 2018

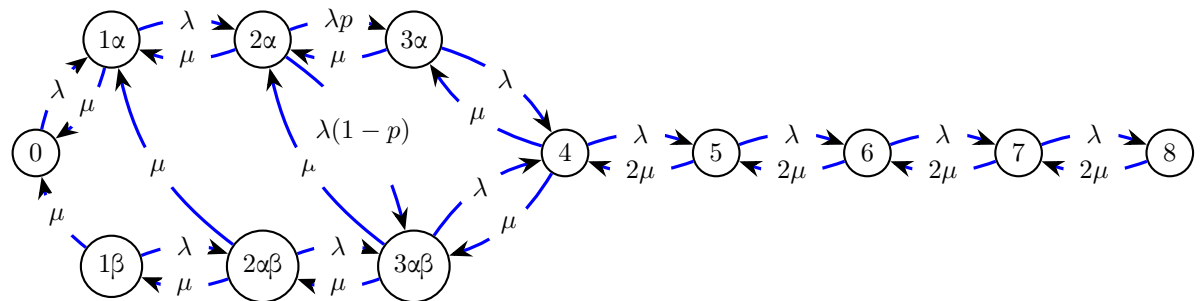
Ουρά M/M/2/8 με Κατώφλι (Threshold) και Τυχαία Ενεργοποίηση

Στην άσκηση αυτή κληθήκαμε να προσομοιώσουμε και να μελετήσουμε ένα M/M/2/8 σύστημα. Σε αυτό το σύστημα αναμονής με 2 εξυπηρετητές και μέγιστο αριθμό πελατών 8 συμπεριλαμβανομένων αυτών που εξυπηρετούνται, εφόσον ο αριθμός των πελατών στο σύστημα είναι μικρότερος του $k(k=3,4,5,6)$ οι αφίξεις δρομολογούνται στον εξυπηρετητή α , ενώ την ίδια στιγμή ο β παραμένει ανενεργός(idle). Όταν ο αριθμός των πελατών στο σύστημα είναι ίσος με k , τότε ο β παραμένει ανενεργός με πιθανότητα p ή ενεργοποιείται με πιθανότητα $1-p$. Αυτό σημαίνει ότι μια νέα άφιξη θα δρομολογηθεί στον εξυπηρετητή β με πιθανότητα $1-p$ είτε θα παραμείνει στην ουρά αναμονής με πιθανότητα p . Όταν ο αριθμός των πελατών στο σύστημα είναι μεγαλύτερος από $k+1$, τότε και οι 2 εξυπηρετητές είναι ενεργοί. Για την κατάσταση του συστήματος με ενεργοποιημένο τον εξυπηρετητή β και με αριθμό πελατών $(k+1)$, η εξυπηρέτηση πελάτη από τον β οδηγεί το σύστημα είτε στην περίπτωση που ο εξυπηρετητής β παραμένει ανενεργός με πιθανότητα p είτε στην κατάσταση που ο εξυπηρετητής β είναι ενεργός με πιθανότητα $1-p$. Στο σύστημα έχουμε αφίξεις Poisson, ρυθμού $\lambda=6,7,8$ πελάτες/sec και εκθετικές εξυπηρετήσεις ρυθμού $\mu_\alpha=\mu_\beta = 8$ πελάτες/sec. Για τη πιθανότητα ενεργοποίησης έχουμε ότι $p=0.5$. Στην επόμενη εικόνα παρουσιάζεται το εν λόγω σύστημα:



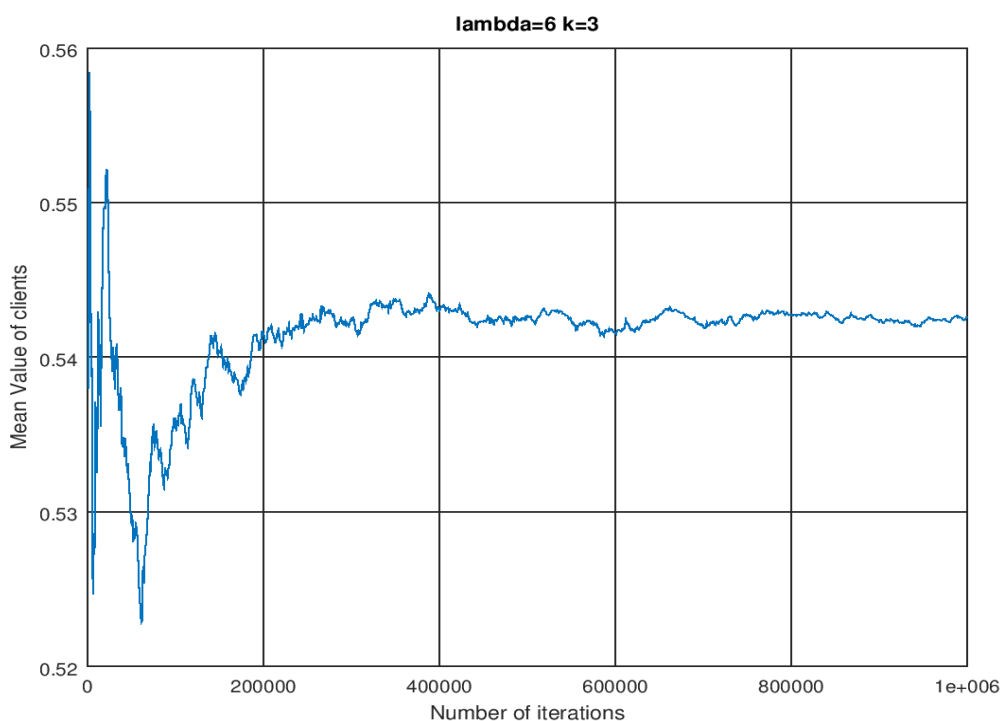
Σχήμα ρυθμών μεταβάσεων καταστάσεων του συστήματος για $k = 3$.

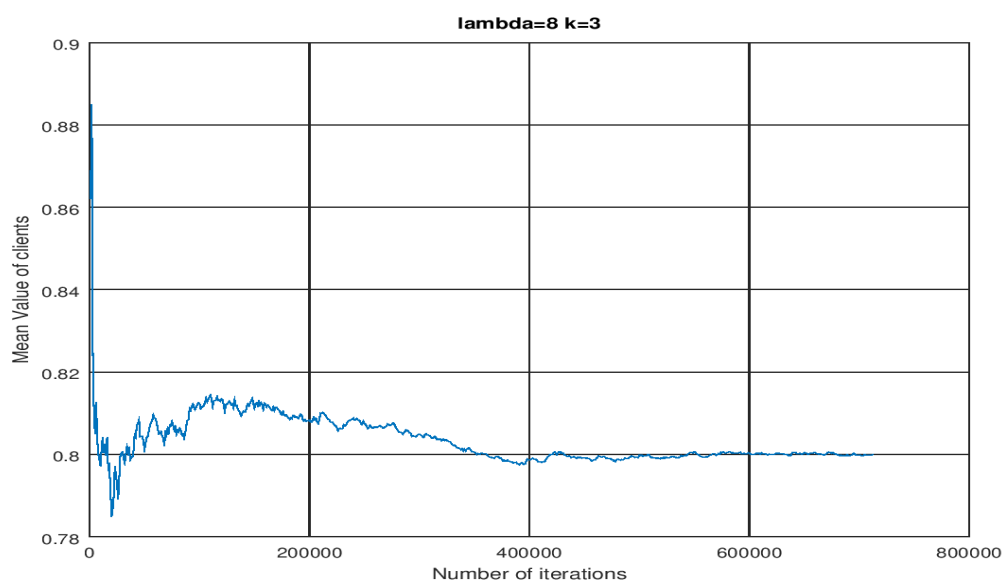
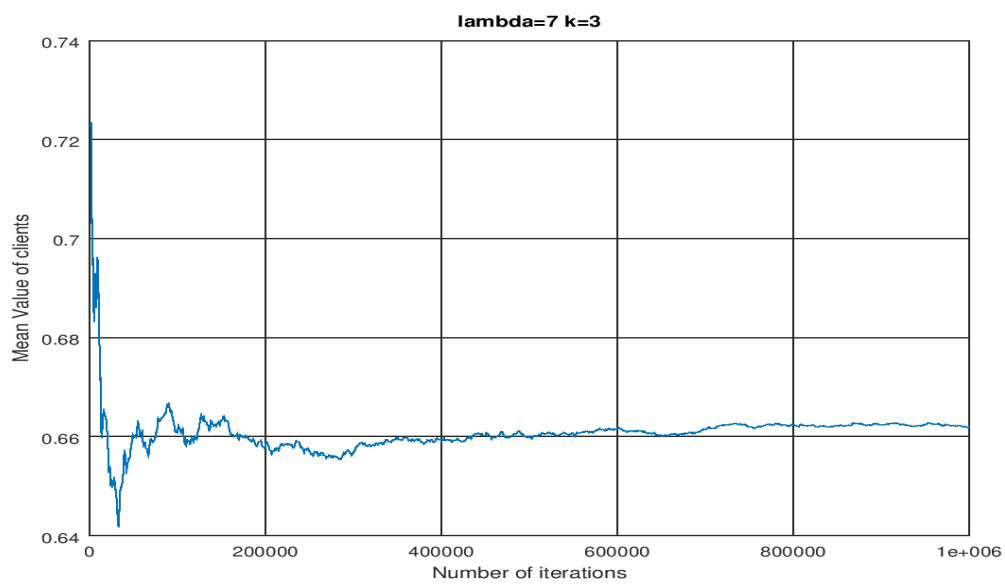
Αρχικά, παρατίθεται το διάγραμμα ρυθμού μεταβάσεων καταστάσεων του συστήματος για τη τιμή του κατωφλίου $k=3$.

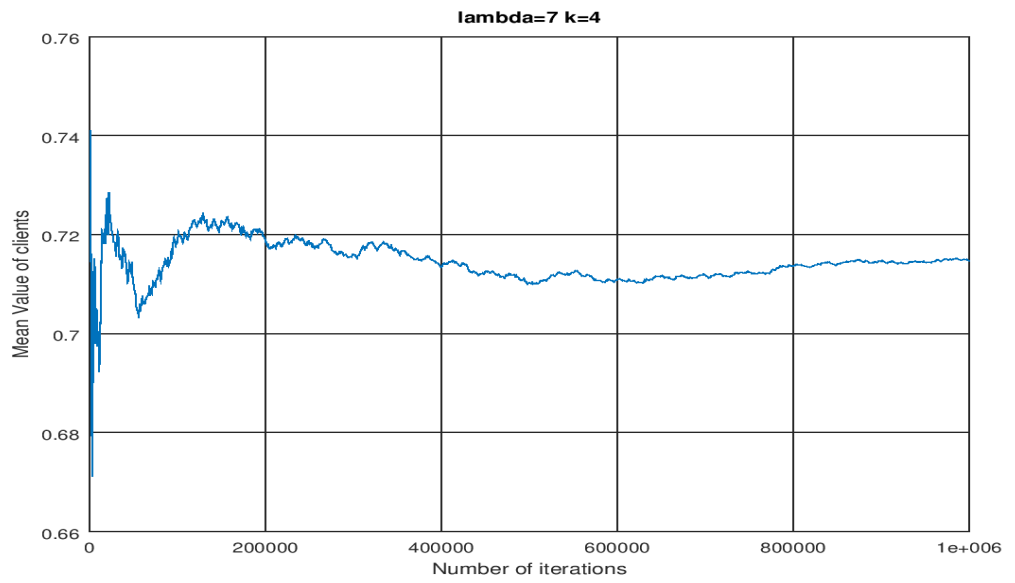
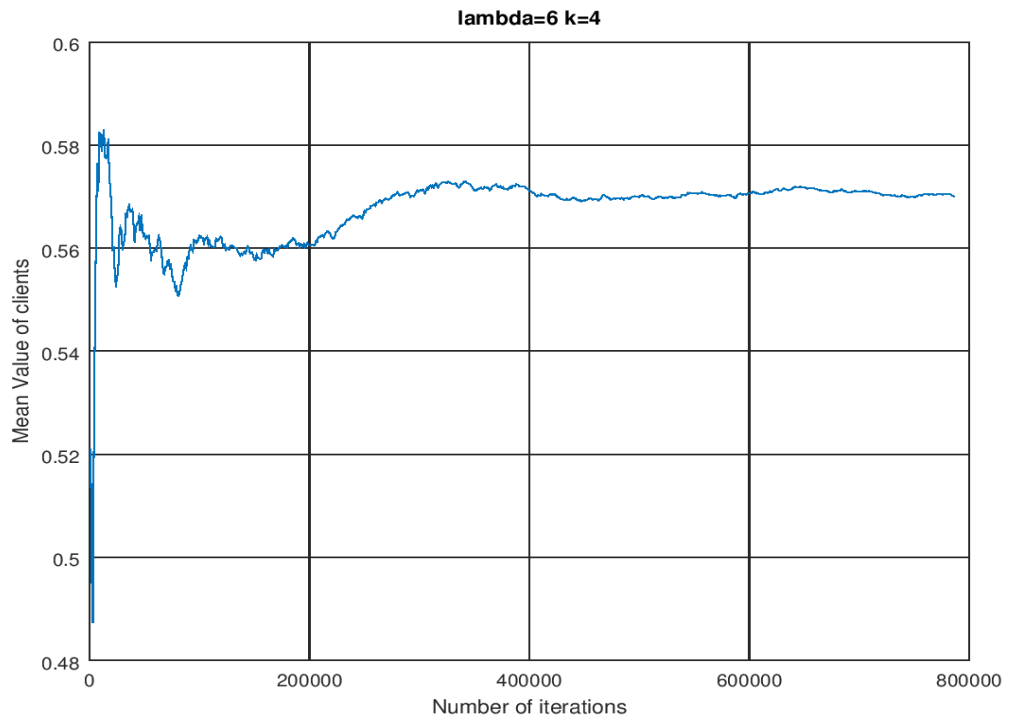


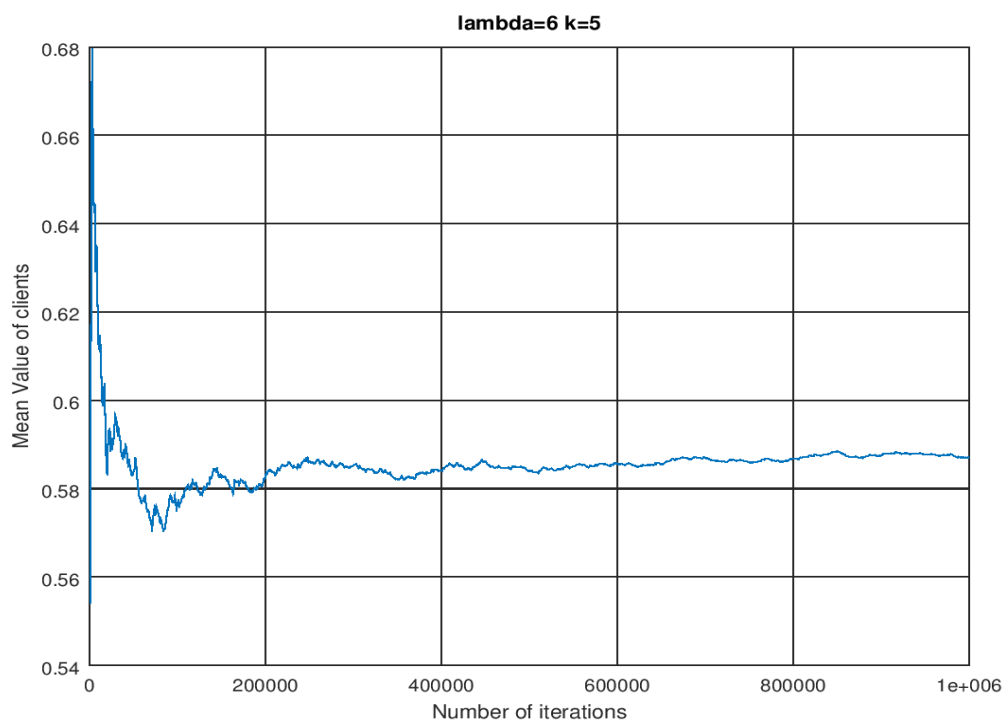
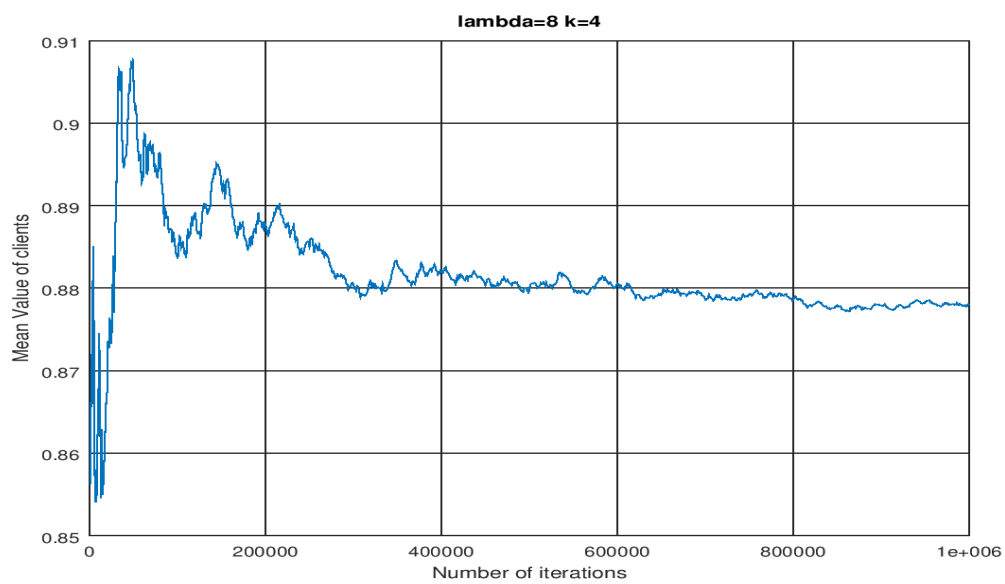
1. Μέσος Αριθμός Πελατών στο Σύστημα για $k=3, \dots, 6$

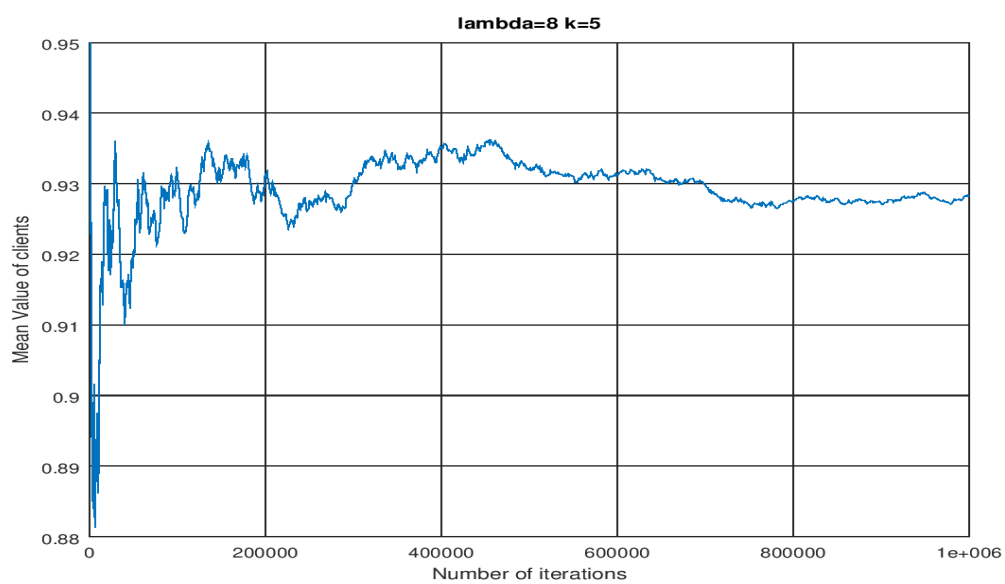
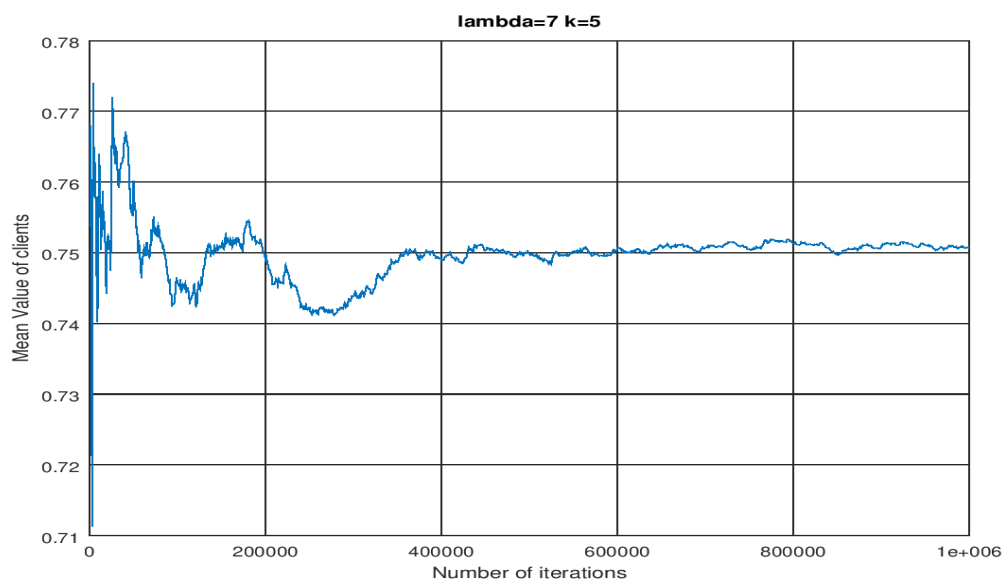
Σε αυτό το ερώτημα ζητάται ο αριθμός των πελατών στο σύστημα για $k=3, \dots, 6$, και για τις 3 περιπτώσεις του ρυθμού εισόδου, όπως αυτό εξελίσσεται κατά τη διάρκεια της προσομοίωσης. Η προσομοίωση σταματάει όταν ικανοποιηθεί το κριτήριο σύγκλισης, το οποίο είναι να έχουν δύο διαδοχικές μετρήσεις του μέσου αριθμού πελατών διαφορά μικρότερη από 0.0000001 ή να έχουμε ξεπεράσει τις 1000000 μεταβέσεις καταστάσεων. Τα αποτελέσματα παρουσιάζονται στις επόμενες γραφικές παραστάσεις για τις διάφορες τιμές των κ και λ :

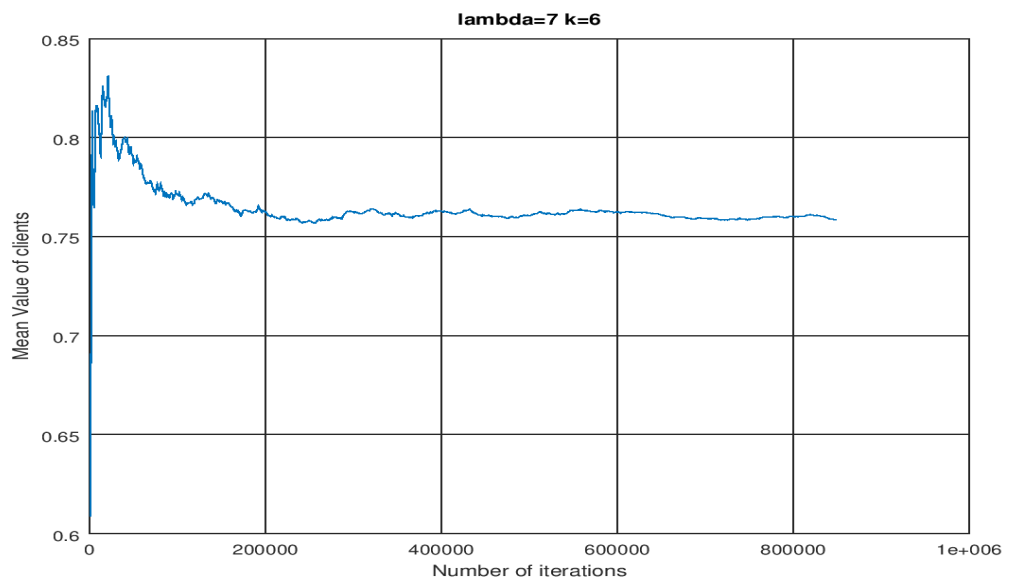
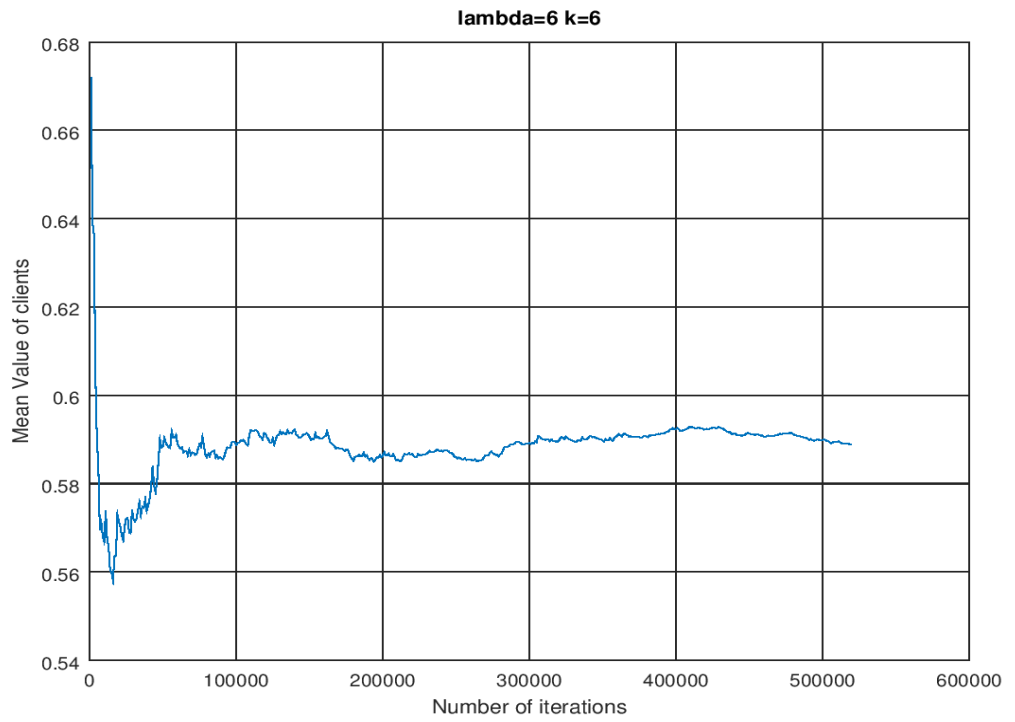


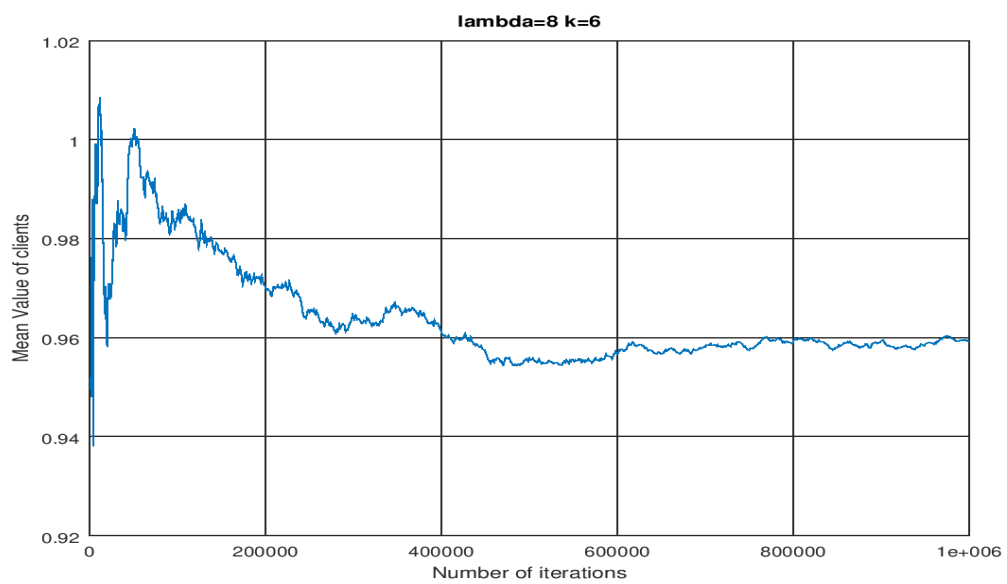






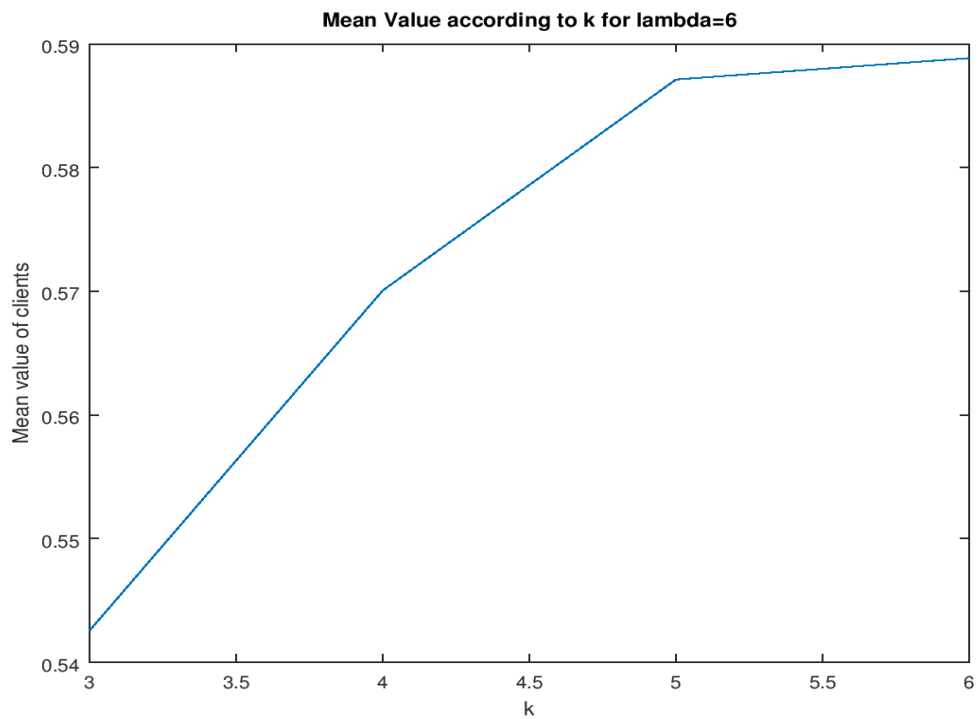


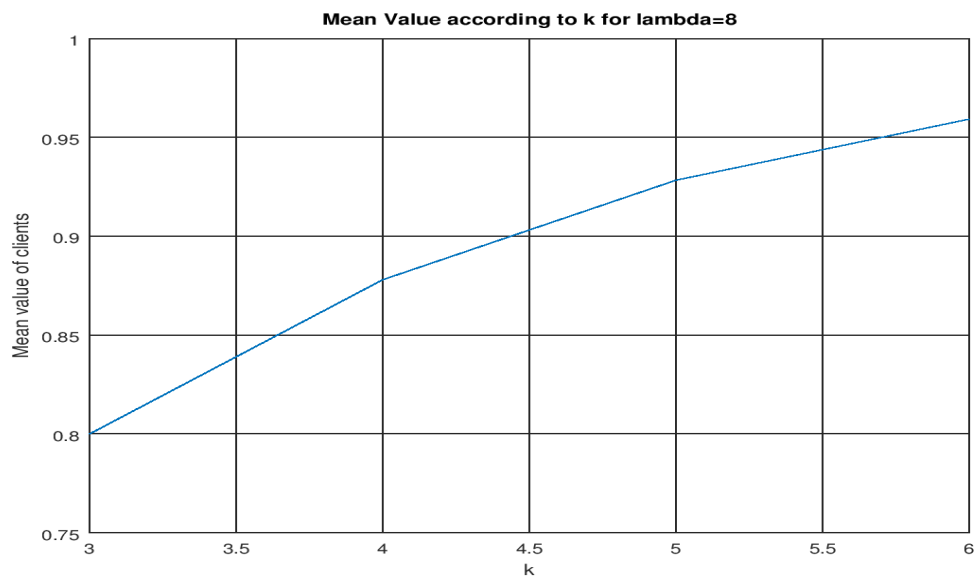
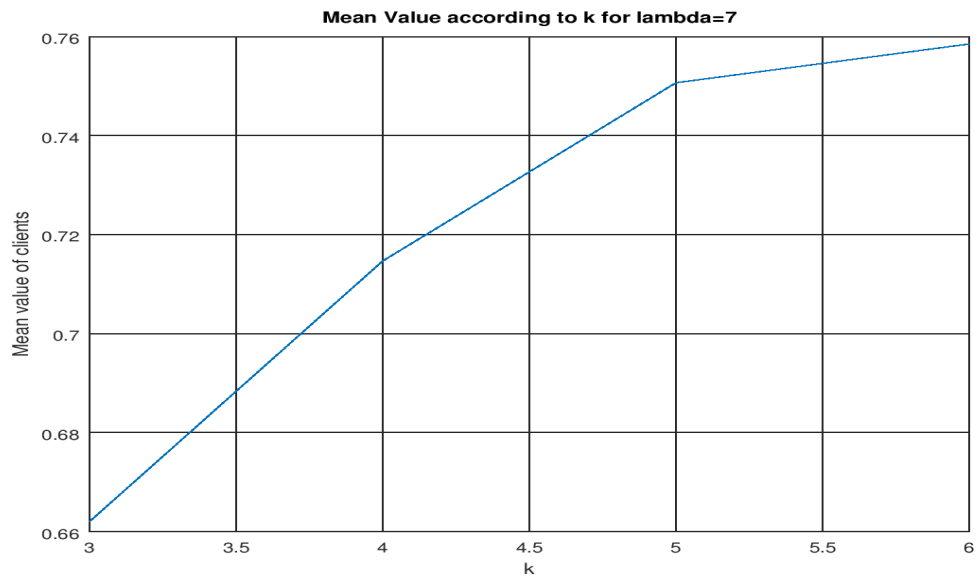




2. Μέσος αριθμός πελατών, μετά τη σύγκλιση ανωτέρω, σαν συνάρτηση του k για κάθε τιμή του ρυθμού εισόδου

Σε αυτό το ερώτημα παρουσιάζεται ο μέσος αριθμός των πελατών, αφού το σύστημα έχει φτάσει σε σύγκλιση. Τα αποτελέσματα παρατίθενται και στις επόμενες γραφικές παραστάσεις :

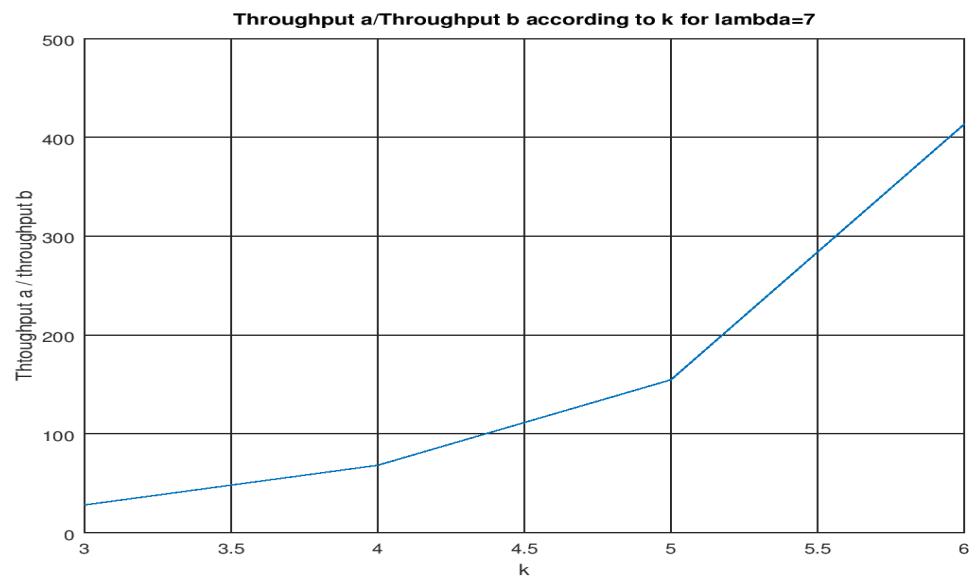
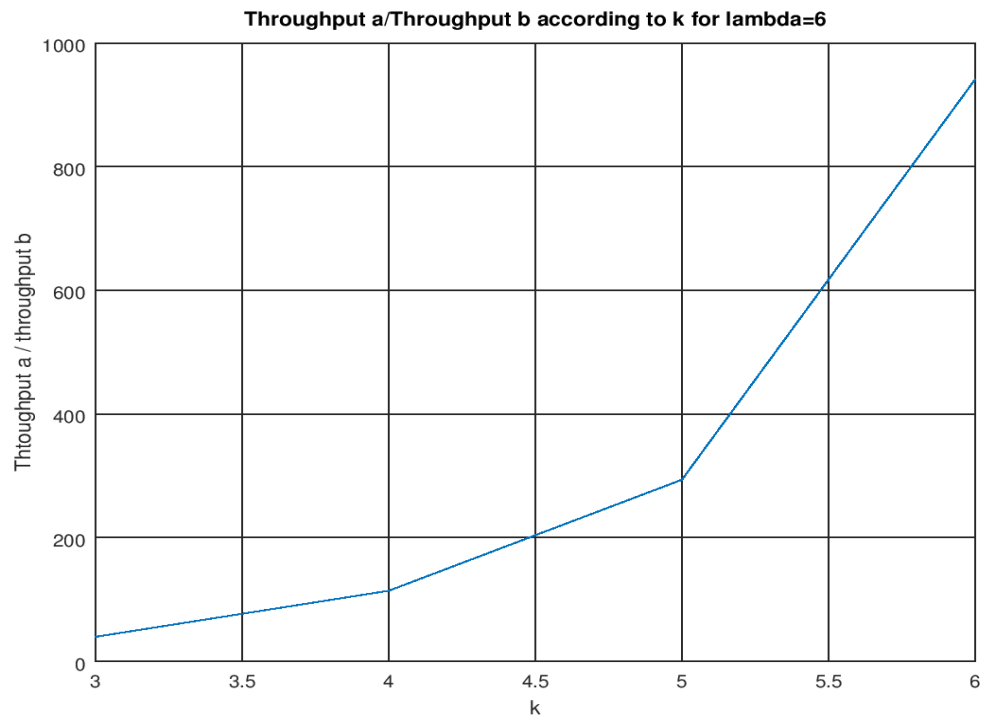


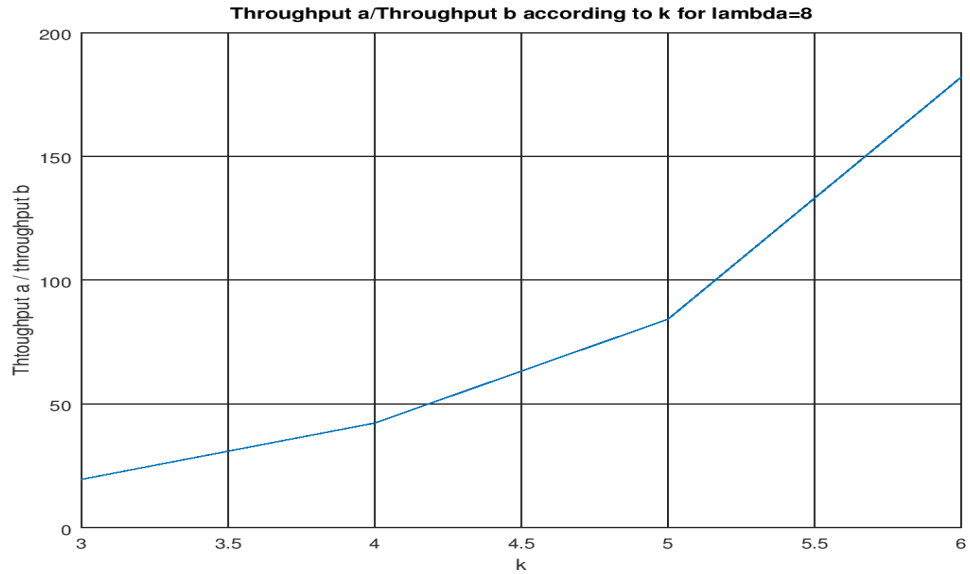


3. Λόγος ρυθμών απόδοσης

Σε αυτό το ερώτημα κληθήκαμε για τις 3 περιπτώσεις του ρυθμού εισόδου να προσδιορίσουμε την ελάχιστη τιμή του κατωφλίου k , για την οποία ο λόγος των ρυθμών απόδοσης των δύο εξυπηρετητών $\gamma_\alpha/\gamma_\beta$ γίνεται μεγαλύτερος από 5, δηλαδή όταν ο εξυπηρετητής α διεκπεραιώνει κατά μέσο όρο 5 φορές περισσότερους πελάτες/sec από τον εξυπηρετητή β . Οι λόγοι των ρυθμών απόδοσης για τις διαφορετικές τιμές

του ρυθμού εισόδου φαίνονται στις παρακάτω γραφικές παραστάσεις, για $p=0.5$:





Σε όλες τις παραπάνω περιπτώσεις παρατηρούμε ότι ο λόγος $\gamma_\alpha/\gamma_\beta$ είναι πάντα μεγαλύτερος από 5. Πειραματιζόμενοι με τις διάφορες τιμές του p , παρατηρούμε ότι καθώς μεγαλώνει το p , ο λόγος αυτός μικραίνει. Στις περισσότερες των περιπτώσεων και για τις διαφορετικές τιμές του λ , ο λόγος είναι πάνω από 5. Ωστόσο, παραδείγματος χάριν, για $p=0.9$, παρατηρούμε ότι για ρυθμό εισόδου $\lambda = 8$ και $k=3$, παρατηρούμε ότι ο λόγος των ρυθμών απόδοσης πέφτει κάτω από 5. Αντίστοιχα παρατηρούμε και για ακόμα μεγαλύτερες τιμές του p , οπότε θα μπορούσαμε να ισχυριστούμε ότι το ζητούμενο κατώφλι είναι το $k = 3$. Αξίζει, τέλος να σχολιαστεί το γεγονός ότι ο λόγος αυτός αυξάνεται, καθώς το k αυξάνεται και αυτό.

4. Σχολιασμός αποτελεσμάτων

Αρχικά, όπως αναφέρθηκε και παραπάνω, ο ρυθμός απόδοσης του α είναι σαφώς μεγαλύτερος από τον ρυθμό απόδοσης του β και, καθώς η τιμή του k αυξάνεται, αυξάνεται και ο λόγος μεταξύ των 2 ρυθμών απόδοσης.

Σε ό,τι αφορά την **ταχύτητα σύγκλισης** της προσομοίωσης, αρχικά σημειώνεται ότι από τον κώδικα που δίνεται και στο παράρτημα, η εξαγωγή των αποτελεσμάτων μας γίνεται ανά 1000 επαναλήψεις. Η προσομοίωση σταματάει όταν το πλήθος των επαναλήψεων ξεπεράσει τις 1000000 ή όταν η διαφορά 2 διαδοχικών μετρήσεων του μέσου αριθμού πελατών γίνει μικρότερη από 0.0000001.

Στον επόμενο πίνακα παρουσιάζονται οι αριθμοί επαναλήψεων για τις διάφορες τιμές των k , λ :

k	3	4	5	6
lambda				
6	1000000	565000	1000000	1000000
7	1000000	1000000	808000	1000000
8	587000	1000000	909000	1000000

Παρατηρούμε ότι στις περισσότερες των περιπτώσεων η προσομοίωση σταματάει μόλις φτάσουμε στις 1000000 επαναλήψεις. Αυτό σημαίνει ότι το σύστημα δε συγκλίνει ιδιαίτερα γρήγορα.

Τέλος, σε ό,τι αφορά την **απόδοση του συστήματος**, παρατηρούμε ότι όσο μικρότερη είναι η τιμή του k , τόσο καλύτερη είναι η συνολική ρυθμαπόδοση του συστήματος, δηλαδή τόσο μικρότερη είναι η τιμή του λόγου των ρυθμών απόδοσης των δύο εξυπηρετητών. Όσο ο λόγος αυτός πλησιάζει τη μονάδα, τότε σημαίνει ότι ο φόρτος του συστήματος είναι επιμερισμένος περίπου ισόποσα στους δύο εξυπηρετητές και, συνεπώς, το σύστημα λειτουργεί αποδοτικά.

ΠΑΡΑΡΤΗΜΑ ΜΕ ΤΟΝ ΚΩΔΙΚΑ ΤΗΣ ΠΡΟ- ΣΟΜΟΙΩΣΗΣ

```
1 close all;
2 clear all;
3 clc;
4 k_vector=[3 4 5 6]; %vector of all k values
5 ma=8;
6 mb=8;
7 p=0.9;
8
9 for l=6:1:8 % loop for all lambdas
10     averages_k = [];
11     ga_k = []; % throughput of a
12     gb_k = []; % throughput of b
13     for k=3:1:6 % loop for all k
14         state = 0; % initial state = 0
15         arrivals_counter = 0;
16         total_counter = 0;
17         upper_chain = 1;
18         conversion_old = 0.0; %about convergence
19         conversion_new = 100.0; %about convergence
20         arrivals_a = zeros([1,12]);
21         arrivals_ab = zeros([1,12]);
22         prob_arrival_a = zeros([1,12]);
23         prob_arrival_ab = zeros([1,12]);
24         averages_vector = [];
25         totalcounter_vector = [];
26         %————SIMULATION && PLOTTING
27         %—————%
28         while (abs(conversion_old - conversion_new) >
29             0.0000001 && (total_counter < 1000000)) %chack
30             convergence
31             i = rand(1); %random generator
32             if (state==0) %Zero state
33                 arrivals_counter=arrivals_counter+1;
34                 arrivals_a(state+1)=arrivals_a(state+1) + 1;
35                 state=state+1;
36                 upper_chain=1;
37             elseif (state == 8) %Final State(queueing system is
38                 full)
39                 if i < (p*1/(p*1+ma+mb)) %Arrival
40                     arrivals_counter = arrivals_counter + 1;
41                     arrivals_ab(state+1) = arrivals_ab(state+1) + 1;
42                 else
```

```

39         state = state -1;
40     end
41 elseif ((state<k+1) && (upper_chain==1) && (state
42     ~=0)) %on state k-1 and a is active
43     if (i < p*1/(p*1+ma)) %Arrival
44         arrivals_counter = arrivals_counter + 1;
45         arrivals_a(state+1)=arrivals_a(state+1) + 1;
46         state = state+1;
47         if (state == k+1)
48             upper_chain=0;
49         end
50     else %Departure
51         state = state -1;
52     end
53 elseif ((state<=(k+1)) && (upper_chain==0) && (
54     state~=0) && (state~=1)) %both a and b are
55     working
56     if (i< p*1/(p*1+ma+mb))
57         arrivals_counter = arrivals_counter + 1;
58         arrivals_ab(state+1) = arrivals_ab(state+1) +1;
59         state = state +1;
60     elseif ((i>=(1/(1+ma+mb))) && (i<((ma+p*1)/(1+ma
61         +mb)))) %departure from a
62         upper_chain = 0;
63         state = state -1;
64     else %departure from b
65         upper_chain = 1;
66         state = state - 1;
67     end
68 elseif ((state == 1) && (upper_chain == 0)) %State
69     1, both are working
70     if (i< p*1/(p*1+mb)) %Arrival
71         arrivals_counter = arrivals_counter + 1;
72         arrivals_ab(state+1)=arrivals_ab(state+1)+1;
73         state = state + 1;
74     else %Departure from b
75         state = state -1;
76         upper_chain=1;
77     end
78 elseif ((state>k+1) && (upper_chain == 0)) %>k+1,
79     both are working
80     if (i < p*1/(p*1+ma+mb))
81         arrivals_counter = arrivals_counter +1;
82         arrivals_ab(state+1) = arrivals_ab(state+1) +1;
83         state = state+1;
84     else

```



```

79         state = state -1;
80     end
81 end
82 total_counter = total_counter+1;
83 if ((mod(total_counter,1000) == 0))
84     conversion_old = conversion_new;
85     conversion_new = 0;
86     for iter = 1:1:12
87         prob_arrival_a(iter)=arrivals_a(iter)/
            arrivals_counter;
88         prob_arrival_ab(iter)=arrivals_ab(iter)/
            arrivals_counter;
89         conversion_new = conversion_new + (iter-1)*(
            prob_arrival_a(iter) +
            prob_arrival_ab(iter));
90     end
91     averages_vector=[averages_vector conversion_new];
92     %Count new mean values
93     totalcounter_vector = [totalcounter_vector
94         total_counter];
95 end
96 end
97 k
98 l
99 total_counter
100 totalcounter_vector;
101 averages_k = [averages_k averages_vector(end)];
102 figure();
103 plot(totalcounter_vector,averages_vector);
104 grid();
105 title(['lambda=',int2str(l), ' k=',int2str(k)]);
106 ylabel('Mean Value of clients');
107 xlabel('Number of iterations');
108 Pa = 1-(prob_arrival_a(1) + prob_arrival_ab(2));%

109 ga = Pa*ma; %
110 ga_k=[ga_k ga];
111 Pb=0; %
112 for j=2:1:12
113     Pb = Pb+prob_arrival_ab(j);
114 end
115 gb=Pb*mb;
116 gb_k=[gb_k gb];
117 end
118 g_ratio= ga_k./gb_k
119 figure();

```

```

120     plot(k_vector, averages_k);
121     grid();
122     title(['Mean Value according to k for lambda=', int2str(
123         1)]);
124     ylabel('Mean value of clients');
125     xlabel('k');
126     figure();
127     plot(k_vector, ga_k);
128     grid();
129     title(['Throughput of a according to k for lambda=',
130         int2str(1)]);
131     ylabel('Throughput a');
132     xlabel('k');
133     figure();
134     plot(k_vector, gb_k);
135     grid();
136     title(['Throughput of b according to k for lambda=',
137         int2str(1)]);
138     ylabel('Throughput b');
139     xlabel('k');
140     figure();
141     plot(k_vector, g_ratio);
142     grid();
143     title(['Throughput a/Throughput b according to k for
144         lambda=', int2str(1)]);
145     ylabel('Throughput a / throughput b');
146     xlabel('k');
147 end

```