Συστήματα Αναμονής Άσκηση 3^{η}

Αυγερινός Πέτρος 03115074

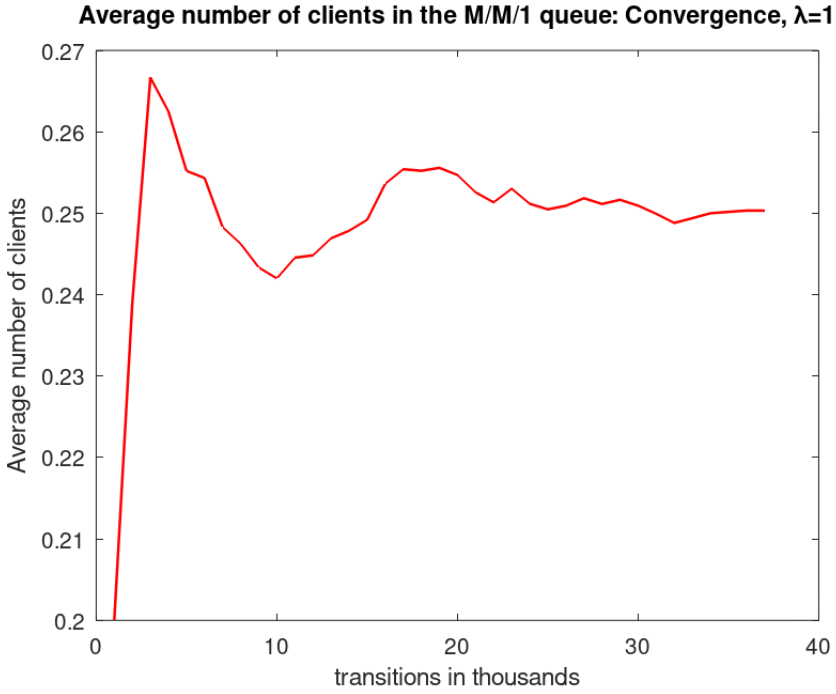
Contents

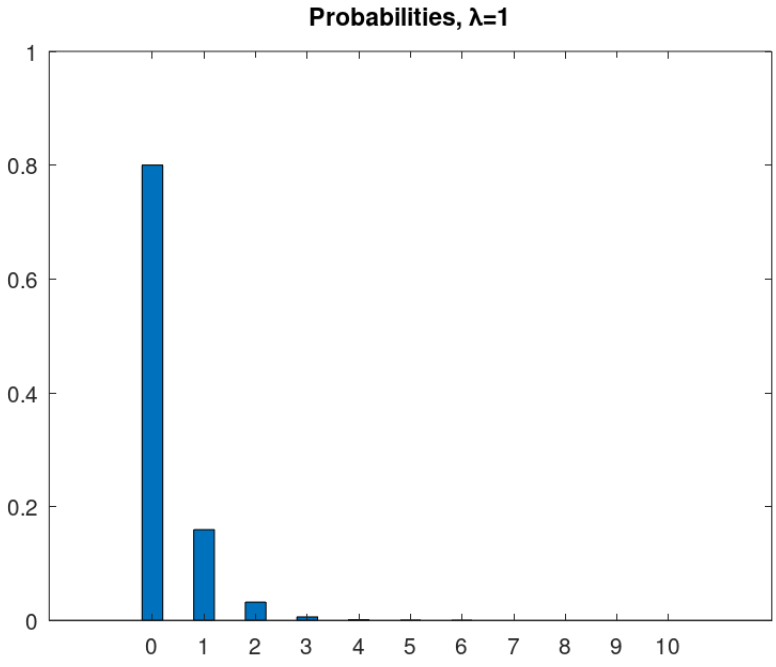
1 Προσομοίωση 1

1.1 Ερώτημα 1

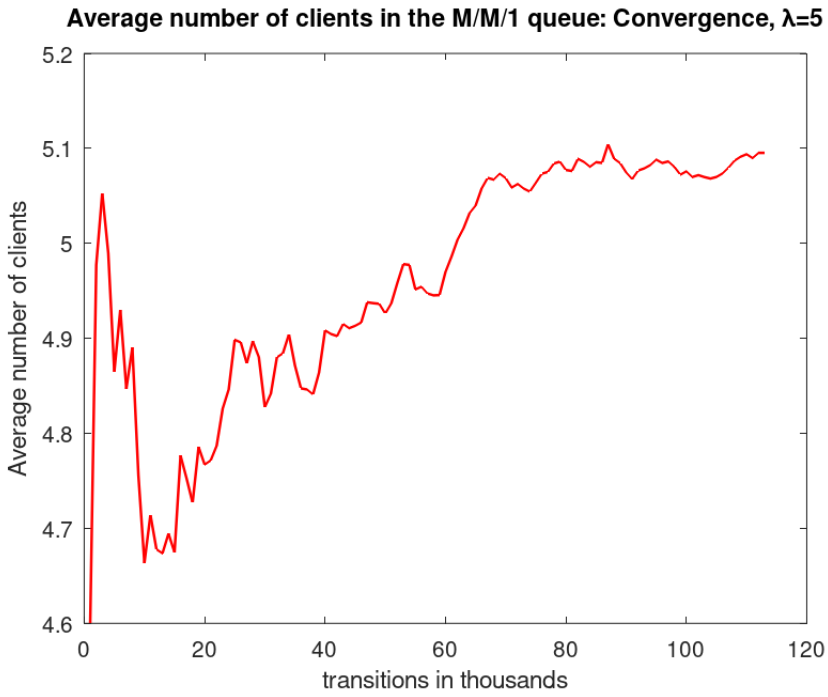
Στο πρώτο ερώτημα ζητείται να υλοποιηθεί η προσομοίωση μίας ουράς M/M/1/10 για τρεις διαφορετικές τιμές του $\lambda = \{1, 5, 10\}$.

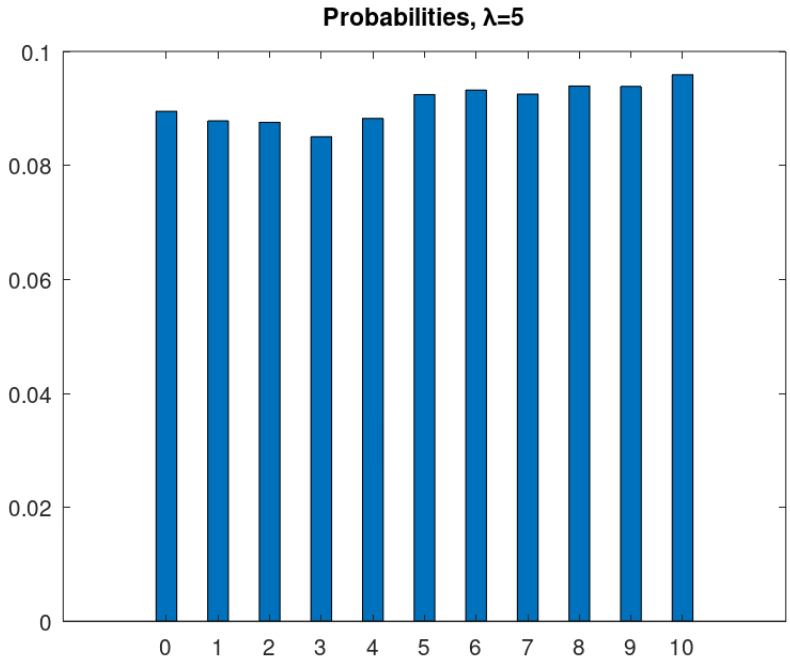
1. Για το $\lambda = 1$ έχουμε τα παρακάτω αποτελέσματα:



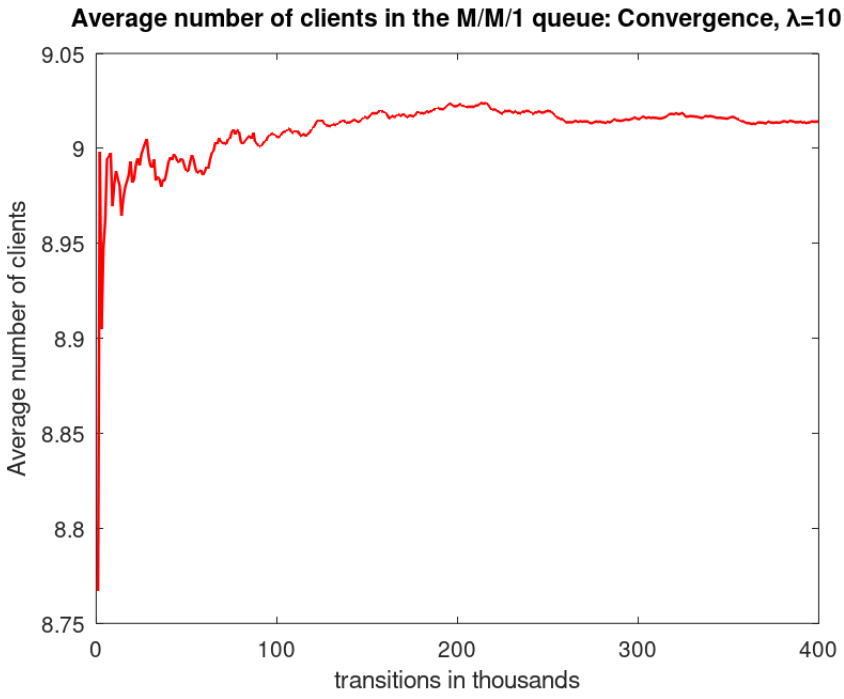


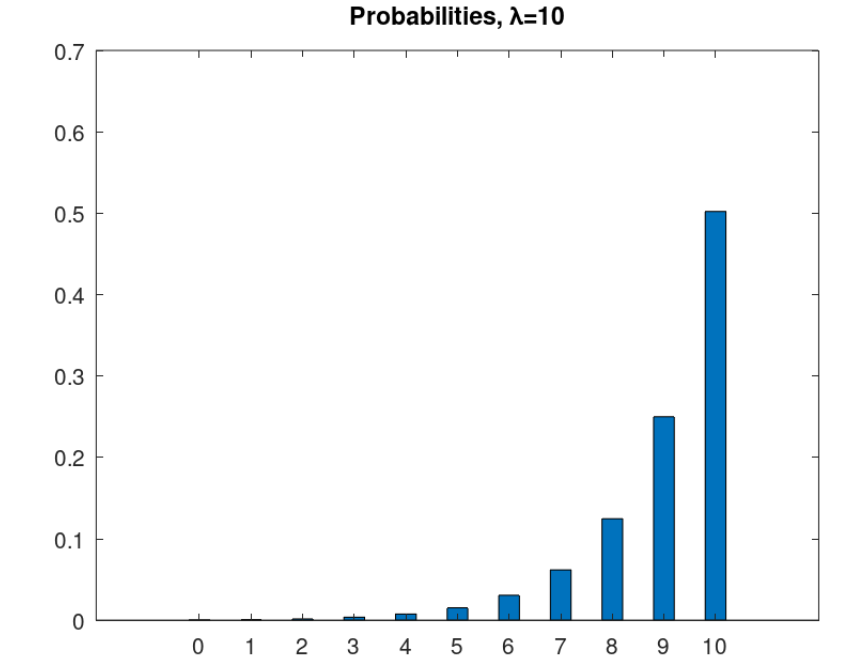
2. Για το $\lambda = 5$ έχουμε τα παρακάτω αποτελέσματα:





3. Για το $\lambda = 10$ έχουμε τα παρακάτω αποτελέσματα:





1.2 Ερώτημα 2

Για την ταχύτητα σύγκλισης για κάθε διαφορετική τιμή του λ έχουμε τα εξής:

1. Για το $\lambda = 1$ έχουμε $Transitions = 37000$, $Delay = 0.25$ και $Blocking = 0$
2. Για το $\lambda = 5$ έχουμε $Transitions = 113000$, $Delay = 1.127178$ και $Blocking = 0.095912$
3. Για το $\lambda = 10$ έχουμε $Transitions = 400000$, $Delay = 1.810885$ και $Blocking = 0.502224$

Βλέπουμε ότι με την αύξηση του λ αυξάνεται και το πλήθος των μεταβάσεων που χρειάζονται για να συγκλίνει το σύστημα. Αυτό σημαίνει πως μειώνεται ο χρόνος σύγκλισης, δηλαδή απαιτείται περισσότερος χρόνος ώστε το σύστημα να βγει από τη μεταβατική κατάσταση και να έρθει σε

ισορροπία. Αυτό συμβαίνει διότι με την αύξηση του λ αυξάνεται και το πλήθος των πελατών που εισέρχονται στο σύστημα, παράλληλα όμως διατηρείται το μ σταθερό και προκύπτει bottleneck στο σύστημα το οποίο απαιτεί περισσότερο χρόνο για να απορροφήσει τους πελάτες.

1.3 Ερώτημα 3

Οι αλλαγές που πρέπει να γίνουν στον κώδικα είναι ότι κάθε φορά που αλλάζει το current_state με κάποια προσθαφαίρεση, πρέπει να αλλάζει το μ σύμφωνα με τον τύπο $\mu_i = \mu(i+1)$ και με αυτή την αλλαγεί να αλλάζει σαφώς και το threshold για αφίξεις και αναχωρήσεις.

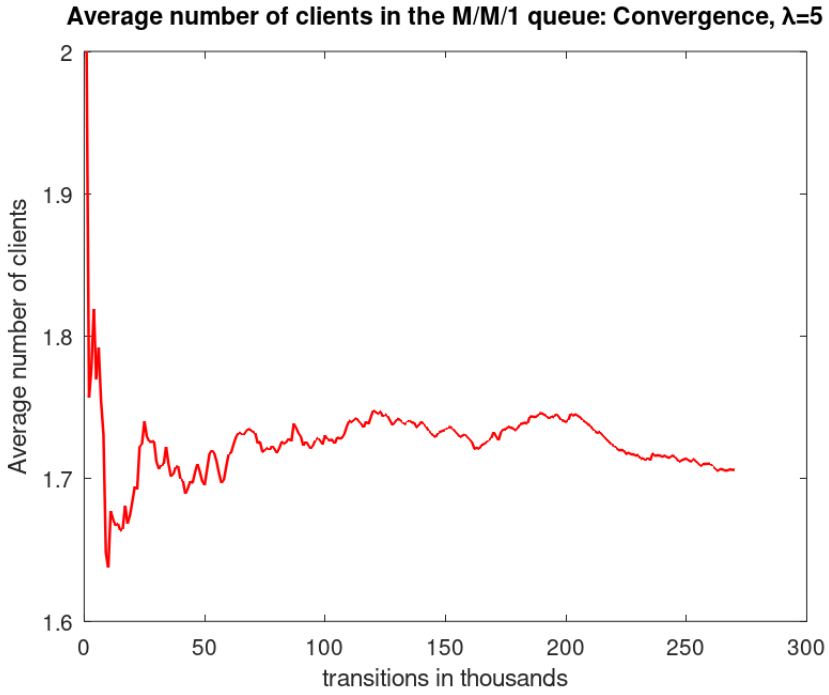
Ο κώδικας για όλα τα ερωτήματα φαίνεται παρακάτω:

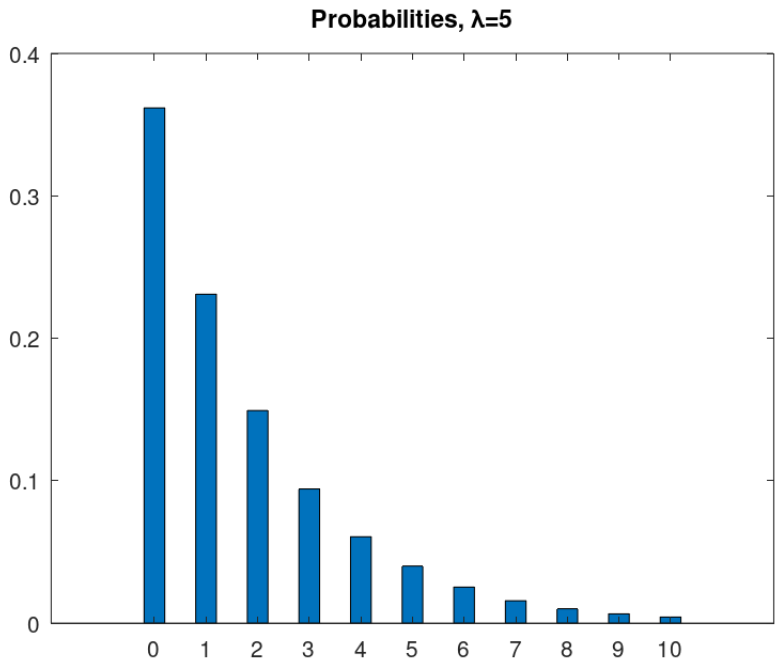
```
clc;
clear all;
close all;

rand("seed",1);

P = [0,0,0,0,0,0,0,0,0,0,0];
arrivals = [0,0,0,0,0,0,0,0,0,0,0];
total_arrivals = 0;
current_state = 0;
previous_mean_clients = 0;
index = 0;

lambda = str2num(argv(){1});

if isnan(lambda)
    display("Please provide a valid lambda value");
    return;
endif

mu = 5;
threshold = lambda/(lambda + mu);

transitions = 0;

while transitions >= 0
  transitions = transitions + 1;

  if mod(transitions,1000) == 0
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/total_arrivals;
    endfor

    mean_clients = 0;
    for i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    endfor

    to_plot(index) = mean_clients;

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

    previous_mean_clients = mean_clients;

  endif

  random_number = rand(1);
  if current_state == 0 || random_number < threshold
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
  else
    if current_state != 0
      current_state = current_state - 1;
    endif
  endif
endwhile


fidfilename = sprintf("../results/results_%d.txt", lambda);

fid = fopen(fidfilename, "w");
tid = fopen("../results/transitions.txt", "a");


fprintf(fid, "State propabilities:\n");
for i=1:1:length(arrivals)
    fprintf(fid, "P(%d) = %f\n", i-1, P(i));
endfor

g = lambda*(1-P(11));
average_delay_time = mean_clients / g;
fprintf(fid, "Average delay time = %f\n", average_delay_time);
fprintf(fid, "Blocking propability = %f\n", P(11));
```

```
fprintf(tid, "Transitions = %d\n", transitions);

figure(1);
plot(to_plot,"r","linewidth",1.2);
titlename = sprintf("Average number of clients in the M/M/1 queue: Convergence,  =%d", lambda);
title(titlename);
xlabel("transitions in thousands");
ylabel("Average number of clients");
filename = sprintf("../images/task1ask1_convergence_%d.png", lambda);
print("-dpng", filename);

x=[0,1,2,3,4,5,6,7,8,9,10];
figure(2);
bar(x,P,0.4);
titlename = sprintf("Probabilities,  =%d", lambda);
title(titlename);
filename = sprintf("../images/task1ask1_probabilities_%d.png", lambda);
print("-dpng", filename);
```

2 Προσομοίωση 2

Η μέση καθυστέρηση του συστήματος είναι 0.342785 ενώ η blocking πιθανότητα είναι 0.004405.
Τα διαγράμματα για την δεύτερη προσομοίωση φαίνονται παρακάτω:





Παρατηρώ ότι η ταχύτητα σύγκλισης είναι μικρότερη από ότι στο σύστημα της πρώτης προσομοίωσης για το ίδιο λ. Αυτό συμβαίνει διότι στο σύστημα οι εξυπηρετητές "κουράζονται" και αυτό επηρεάζει την ταχύτητα εξυπηρέτησης και την ταχύτητα σύγκλισης του συστήματος.

Ο κώδικας για τα παραπάνω φαίνεται παρακάτω:

```
clc;
clear all;
close all;

rand("seed",1);

P = [0,0,0,0,0,0,0,0,0,0,0];
arrivals = [0,0,0,0,0,0,0,0,0,0,0];
total_arrivals = 0;
current_state = 0;
previous_mean_clients = 0;
counter = 0;
index = 0;

lambda = 5;

max_mu = 8;
min_mu = 3;
mu = max_mu;
threshold = lambda/(lambda + mu);

transitions = 0;

while transitions >= 0
  transitions = transitions + 1;

  if mod(transitions,1000) == 0
    index = index + 1;
    for i=1:1:length(arrivals)
        P(i) = arrivals(i)/total_arrivals;
    endfor

    mean_clients = 0;
    for i=1:1:length(arrivals)
       mean_clients = mean_clients + (i-1).*P(i);
    endfor

    to_plot(index) = mean_clients;

    if abs(mean_clients - previous_mean_clients) < 0.00001 || transitions > 10000000
      break;
    endif

    previous_mean_clients = mean_clients;

  endif

  random_number = rand(1);
  if current_state == 0 || random_number < threshold
    if current_state < 11
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
      if current_state < 10
        current_state = current_state + 1;
      endif
    endif
```

```
  else
    if (mu == min_mu)
      counter = counter + 1;
    else
        mu = 0.98 * mu;
    endif

    threshold = lambda/(lambda + mu);

    if counter == 10
        mu = max_mu;
    endif

    if (mu < min_mu)
      mu = min_mu;
      counter = 0;
    endif

    if current_state != 0
      current_state = current_state - 1;
    endif
  endif
endwhile


fidfilename = sprintf("../results/results2_%d.txt", lambda);

fid = fopen(fidfilename, "w");
tid = fopen("../results/transitions2.txt", "a");


fprintf(fid, "State propabilities:\n");
for i=1:1:length(arrivals)
    fprintf(fid, "P(%d) = %f\n", i-1, P(i));
endfor

g = lambda*(1-P(11));
average_delay_time = mean_clients / g;
fprintf(fid, "Average delay time = %f\n", average_delay_time);
fprintf(fid, "Blocking propability = %f\n", P(11));
fprintf(tid, "Transitions = %d\n", transitions);

figure(1);
plot(to_plot,"r","linewidth",1.2);
titlename = sprintf("Average number of clients in the M/M/1 queue: Convergence,   =%d", lambda);
title(titlename);
xlabel("transitions in thousands");
ylabel("Average number of clients");
filename = sprintf("../images/tasklask2_convergence_%d.png", lambda);
print("-dpng", filename);

x=[0,1,2,3,4,5,6,7,8,9,10];
figure(2);
bar(x,P,0.4);
titlename = sprintf("Probabilities,   =%d", lambda);
title(titlename);
filename = sprintf("../images/task1ask2_probabilities_%d.png", lambda);
print("-dpng", filename);
```