**ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ**

ΝΤΟΝΤΟΡΟΣ ΗΛΙΑΣ

el19206

**ΑΣΚΗΣΗ 1η**

Ερώτημα 1ο

Για να μπορέσουμε να μοντελοποιήσουμε τους συνδέσμους σαν ουρές Μ/Μ/1 θα πρέπει :

1. Η ροή εισερχόμενων πελατών να είναι διαδικασία Poisson με παράμετρο λ και η ροή αυτή να διασπάται τυχαία και να παράγονται διαδικασίες Poisson με παραμέτρους και .
2. Οι δύο γραμμές 1 και 2 να μοντελοποιούνται σαν ουρές Μ/Μ/1 με μέσους ρυθμούς άφιξης λ1 και λ2 και μέσους ρυθμούς εξυπηρέτησης μ1 και μ2 αντίστοιχα.

Για την γραμμή 1 έχουμε :

Και :

Για την γραμμή 2 έχουμε :

Και :

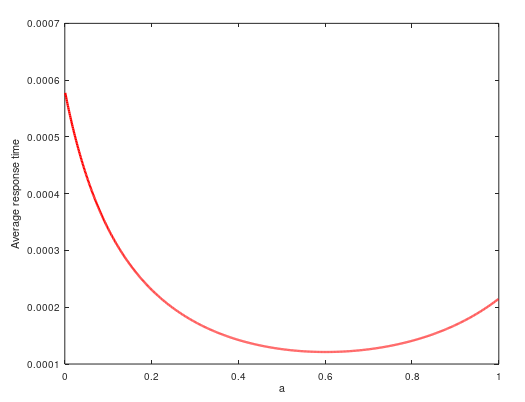
Για να είναι εργοδικές οι ουρές πρέπει:

και

Ερώτημα 2ο

Από το νόμο του Little ο μέσος χρόνος καθυστέρησης Ε[Τ] υπολογίζεται

από τον τύπο οπου είναι ο μέσος αριθμός πελατών και γ είναι οι εξωτερικές ροές δηλαδή λ. το διάγραμμα που προκύπτει από το Octave είναι :



Και ο ελάχιστος χρόνος καθυστέρησης είναι :

1. Minimun value of E(T)
2. 1.2118e-04
3. for a=
4. 0.6020

Ο κώδικας για την άσκηση 1 :

1. clc;
2. clear all;
3. close all;
4. pkg load queueing
6. # Exercise 1
8. # Task 2
10. a = 0.001:0.001:0.999;
11. l = 10000;
12. l1 = 10000\*a;
13. m1 = 14650;
14. l2 = 10000\*(1-a);
15. m2 = 11720;
17. [U1, R1, Q1, X1, P1] = qsmm1(l1, m1);
18. [U2, R2, Q2, X2, P2] = qsmm1(l2, m2);
20. totClients = Q1 + Q2;
21. totTime = totClients/l;
22. figure(1);
23. plot(a, totTime,"g","linewidth",2);
24. xlabel("a");
25. ylabel("Average response time");
27. [minval, mina] = min(min(totTime,[],1));
28. display("Minimun value of E(T)")
29. disp(minval)
30. display("for a=")
31. disp(0.001\*(mina+1))

**ΑΣΚΗΣΗ 2η**

Ερώτημα 1ο

Σύμφωνα με το θεώρημα Jackson για να μελετήσουμε το δίκτυο ως ανοιχτό δίκτυο πρέπει να ισχύουν οι παρακάτω προϋποθέσεις :

1. Η κάθε ουρά αναμονής Qi για i = 1, 2, 3, 4, 5 να είναι δικτυακός κόμβος εξυπηρέτησης κορμού με εκθετικούς ρυθμούς εξυπηρέτησης µi για i = 1, 2, 3, 4, 5.
2. Για τις αφίξεις πελατών που προέρχονται από εξωτερικές πηγές που είναι άμεσα συνδεδεμένες στους δικτυακούς κόμβους κορμού Q1, Q2, ενώ προσανατολίζονται προς τους εξωτερικούς προορισμούς που είναι άμεσα συνδεδεμένοι στους δικτυακούς κόμβους κορμού Q4, Q5. Οι ροές μεταξύ των δικτυακών κόμβων να είναι ανεξάρτητες ροές Poisson με μέσο ρυθμό γij για i,j = 1, 2, 3, 4, 5 και η συνολική εξωγενής ροή Poisson στην ουρά Q­­i είναι ίση με .
3. Για την δρομολόγηση των πελατών μεταξύ δύο ουρών Qi, Qj πρέπει να γίνονται με τυχαίο τρόπο και με πιθανότητα που ισούται με rij. Έχουμε : r12=2/7, r13=4/7, r14=1/7, r35=1/2 και r34=1/2.
4. Οι ροές που διαπερνούν τον δικτυακό κόμβο Qi να έχουν συνολικό μέσο ρυθμό ίσο με
5. Οι χρόνοι εξυπηρέτησης των πελατών στις ουρές να έχουν την ιδιότητα της έλλειψης μνήμης και η τιμή τους να είναι εξαρτημένη από την κατανομή του κάθε εξυπηρετητή.

Ερώτημα 2ο

Για την ένταση του φορτιού ισχύει . Χρησιμοποιώντας το θεώρημα Bruke, δηλαδή ότι η έξοδος πελατών από μια ουρά Μ/Μ/1 ακολουθεί κατανομή Poisson με ρυθμό ίσο με τον ρυθμό εισόδου λ, έχουμε :

Ερώτημα 3ο

Η συνάρτηση mean\_clients θα επιστρέφει τις τιμές μέσου αριθμού πελατών Qi για i=1,2,3,4,5 σε κάθε ουρά

Ερώτημα 4ο

Για τιμές : οι εντασεις των φορτίων που δέχεται η κάθε ουρά είναι από το Octave :

1. r1=
2. 0.6667
3. r2=
4. 0.4286
5. r3=
6. 0.2857
7. r4=
8. 0.2449
9. r5=
10. 0.5476

Ο μέσος χρόνος καθυστέρησης είναι :

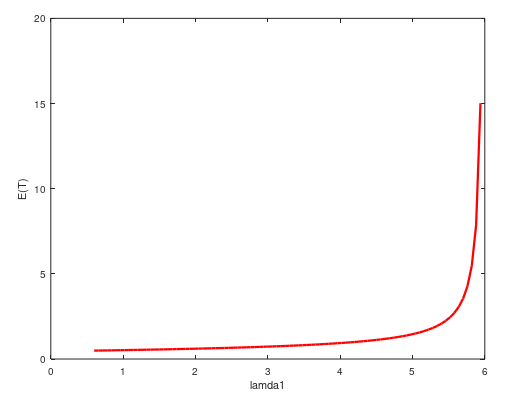
1. E(T)=
2. 0.9370

Ερώτημα 5ο

Στενωπός του δικτύου είναι η ουρά 1 αφού σε αυτή εμφανίζεται η μεγαλύτερη ροή φορτιού. Η μέγιστη τιμή λ1 που μπορούμε να έχουμε με το σύστημα να παραμείνει εργοδικό είναι .

Ερώτημα 6ο

Για το διάγραμμα μέσου χρόνου καθυστέρησης ενός πελάτη από άκρο σε άκρο είναι :



Ο κώδικας για την άσκηση 2 :

1. # Exercise 2
3. # Task 2
5. clc;
6. clear all;
7. close all;
8. pkg load queueing

11. function [r1, r2, r3, r4, r5, e] = intesities(lamda1, lamda2, mu1, mu2, mu3, mu4, mu5)
12. r1 = (lamda1/mu1);
13. r2 = ((lamda2+(2/7)\*lamda1)/mu2);
14. r3 = ((4/7)\*lamda1/mu3);
15. r4 = ((3/7)\*lamda1/mu4);
16. r5 = (((4/7)\*lamda1+lamda2)/mu5);
17. if((r1<1) && (r2<1) && (r3<1) && (r4<1) && (r5<1))
18. e = 1;
19. else
20. e = 0;
21. endif
22. display("r1=")
23. disp(r1)
24. display("r2=")
25. disp(r2)
26. display("r3=")
27. disp(r3)
28. display("r4=")
29. disp(r4)
30. display("r5=")
31. disp(r5)
32. endfunction
34. # Task 3
36. function [r1, r2, r3, r4, r5, e] = intesities\_no\_display(lamda1, lamda2, mu1, mu2, mu3, mu4, mu5)
37. r1 = (lamda1/mu1);
38. r2 = ((lamda2+(2/7)\*lamda1)/mu2);
39. r3 = ((4/7)\*lamda1/mu3);
40. r4 = ((3/7)\*lamda1/mu4);
41. r5 = (((4/7)\*lamda1+lamda2)/mu5);
42. if((r1<1) && (r2<1) && (r3<1) && (r4<1) && (r5<1))
43. e = 1;
44. else
45. e = 0;
46. endif
47. endfunction

50. function [Q1, Q2, Q3, Q4, Q5] = mean\_clients(lamda1, lamda2, mu1, mu2, mu3, mu4, mu5)
51. [r1, r2, r3, r4, r5, e] = intesities\_no\_display(lamda1, lamda2, mu1, mu2, mu3, mu4, mu5);
52. Q1 = r1/(1-r1);
53. Q2 = r2/(1-r2);
54. Q3 = r3/(1-r3);
55. Q4 = r4/(1-r4);
56. Q5 = r5/(1-r5);
57. endfunction
59. # Task 4
61. lamda1 = 4;
62. lamda2 = 1;
63. mu1 = 6;
64. mu2 = 5;
65. mu3 = 8;
66. mu4 = 7;
67. mu5 = 6;
68. [r1,r2,r3,r4,r5,e]=intesities(lamda1, lamda2, mu1, mu2, mu3, mu4, mu5);

71. [Q1, Q2, Q3, Q4, Q5] = mean\_clients(lamda1, lamda2, mu1, mu2, mu3, mu4, mu5);
72. display("E(T)=")
73. disp((Q1+Q2+Q3+Q4+Q5)/(lamda1+lamda2))
75. # Task 6
77. maxlamda1 = 6;
78. for i = 1:1:90;
79. lamda1 = (0.1\*maxlamda1)+(i-1)\*0.01\*maxlamda1;
80. [Q1, Q2, Q3, Q4, Q5] = mean\_clients(lamda1, lamda2, mu1, mu2, mu3, mu4, mu5);
81. E(i)= (Q1+Q2+Q3+Q4+Q5)/(lamda1+lamda2);
82. endfor
84. lamda1 = (0.1\*maxlamda1):(0.01\*maxlamda1):(0.99\*maxlamda1);
85. figure(2);
86. plot(lamda1, E,"r","linewidth",2);
87. xlabel("lamda1");
88. ylabel("E(T)");