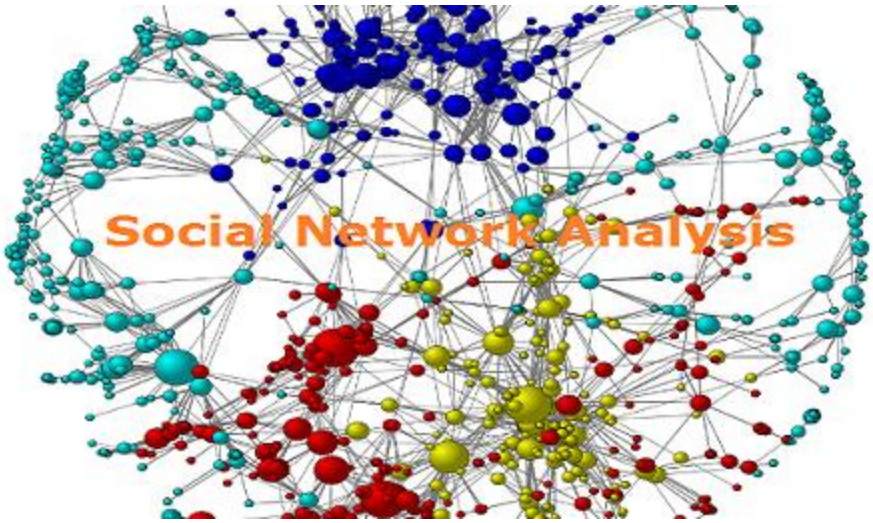
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση Κοινωνικών Δικτύων

3η Εργαστηριακή Άσκηση:

Γενετικοί Αλγόριθμοι & Επιδημιολογικά Μοντέλα



Ονοματεπώνυμο: Σταυρακάκης Δημήτριος

ΑΜ:03112017

Εξάμηνο: 9^{ο}

Ημερομηνία Παράδοσης: 3/2/2017

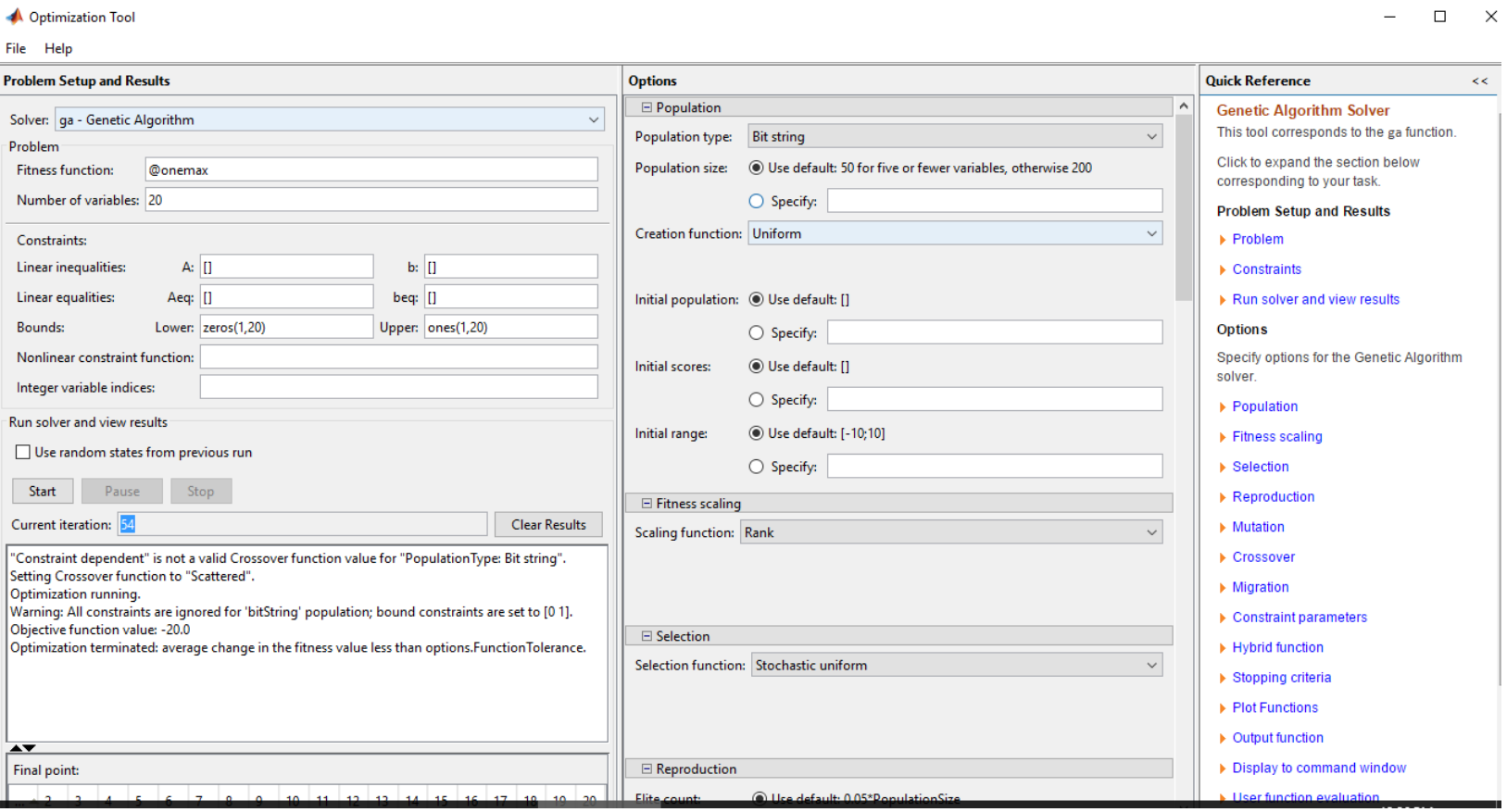
Μέρος Α

Άσκηση 1η:Εξοικείωση με τους Γενετικούς Αλγορίθμους

Στο μέρος αυτό της εργαστηριακής άσκησης καλούμαστε με χρήση γενετικού αλγορίθμου να λύσουμε το πρόβλημα OneMax. Το πρόβλημα αυτό αναζητεί τη δυαδική ακολουθία $x_1, x_2, \dots, x_n$ που να μεγιστοποιεί το άθροισμα $x_1 + x_2 + \dots + x_n$, Για εμάς n=20. Άρα η προφανής λύση είναι $x_1 = x_2 = \dots = x_n = 1$. Θα πειραματιστούμε με 3 παραμέτρους του γενετικού αλγορίθμου για να εξετάσουμε την ποιότητα της λύσης που μας δίνουν.

- Πληθυσμός χρωμοσωμάτων: 10 έως 100 με βήμα 10
- Πιθανότητα Διασταύρωσης: 0.3 έως 0.9 με βήμα 0.1
- Πιθανότητα Μετάλλαξης: 0.01 έως 0.2 με βήμα 0.01

Το πρόβλημα αρχικά το ορίσαμε με χρήση του Optimization Tool του Matlab όπως φαίνεται παρακάτω:



Στη συνέχεια το κάναμε export to workspace και με χρήση τριπλής for loop εξετάσαμε και αποθηκεύσαμε τα αποτελέσματα που μας έδωσε για κάθε συνδυασμό των παραμέτρων που θέλουμε. Ως fitness function του αλγορίθμου δημιουργήσαμε την onemax() που το μόνο που κάνει είναι να επιστρέφει το αντίθετο (-1*) το άθροισμα των 20 όρων της δυαδικής ακολουθίας καθώς με το γενετικό

αλγόριθμο βρίσκουμε ελάχιστη λύση. (για αυτό το λόγο βάζουμε και τον αντίθετο του sum σαν έξοδο, ώστε τελικά να μας δώσει μεγιστοποίηση του sum)

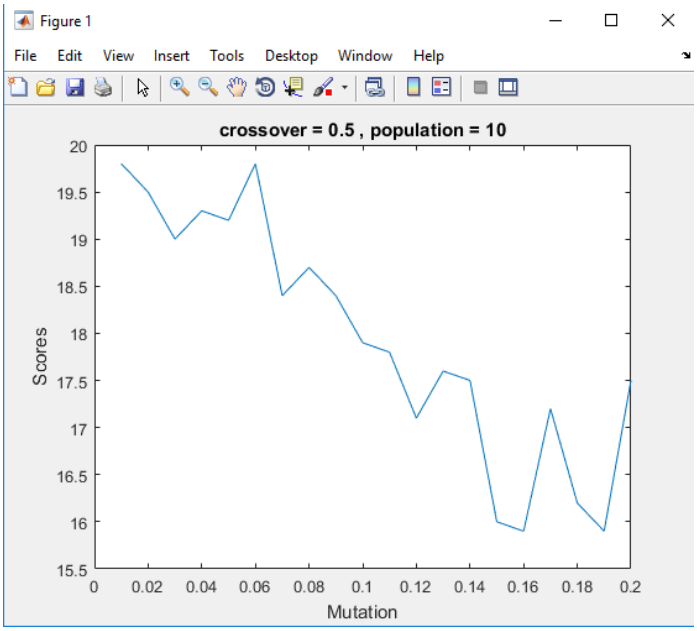
Για να δούμε πως επηρεάζει ο πληθυσμος, η πιθανότητα μετάλλαξης και η πιθανότητα διασταύρωσης την απόδοση του γενετικού αλγορίθμου για κάθε συνδυασμό πήραμε τα score του πληθυσμού (κάθε ενός απο τα χρωμοσώματα), υπολογίσαμε το μέσο όρο τους και έτσι μπορούμε να εκτιμήσουμε την απόδοση της λύσης που πήραμε καθώς γνωρίζουμε ότι το ιδανικό score για κάθε πληθυσμό είναι το 20.

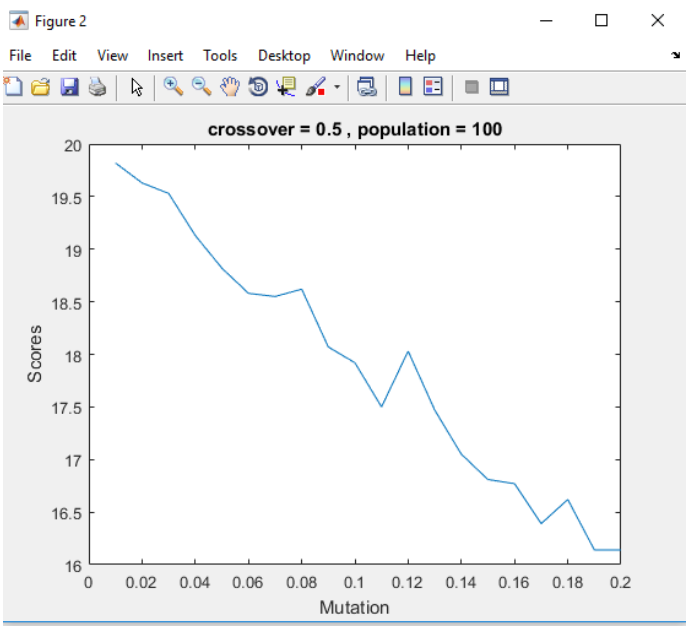
Παρατηρούμε ότι το πρόβλημα είναι πολύ απλό επομένως για μεγάλες τιμές του πληθυσμού η έξοδος του γενετικού αλγορίθμου βρίσκει ορθά τη βέλτιστη λύση.

Επομένως τα διαγράμματα για μεγάλες τιμές του πληθυσμού δεν έχει νόημα να τα παραθέσω καθώς όλα είναι πανομοιότυπα και προσεγγίζουν σχεδόν τέλεια τη βέλτιστη λύση.

Θα παραθέσω διαγράμματα κατάλληλα ώστε να επιδεικνύουν την επιρροή κάθε παραμέτρου στη λύση που μας δίνει ο γενετικός αλγόριθμος:

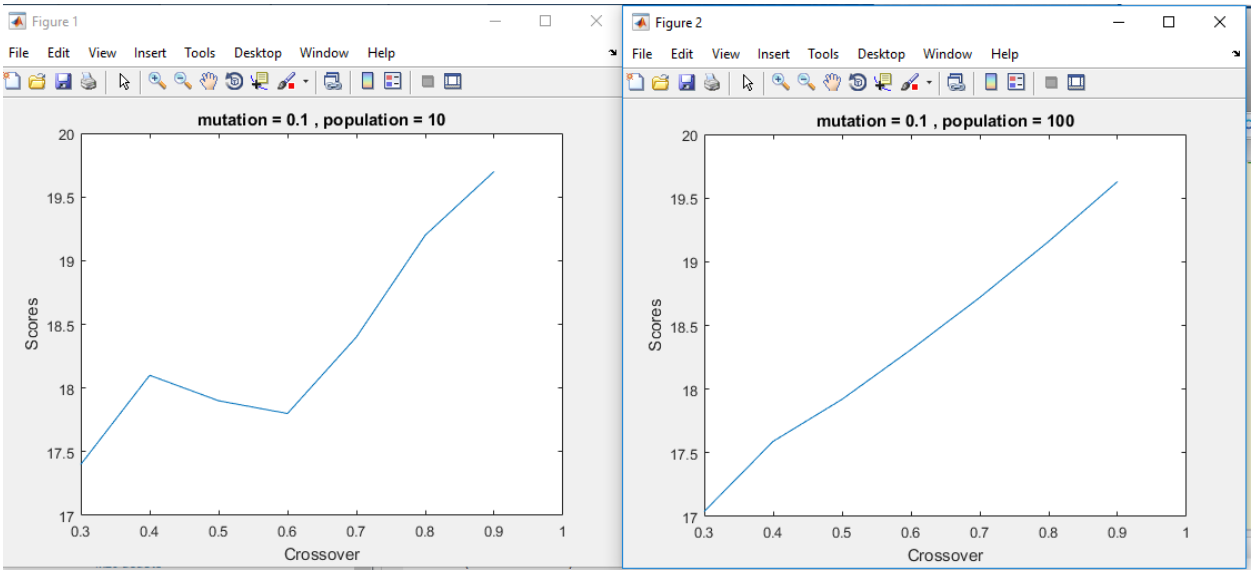
<u>Μεταβολή στην πιθανότητα μετάλλαξης:</u>





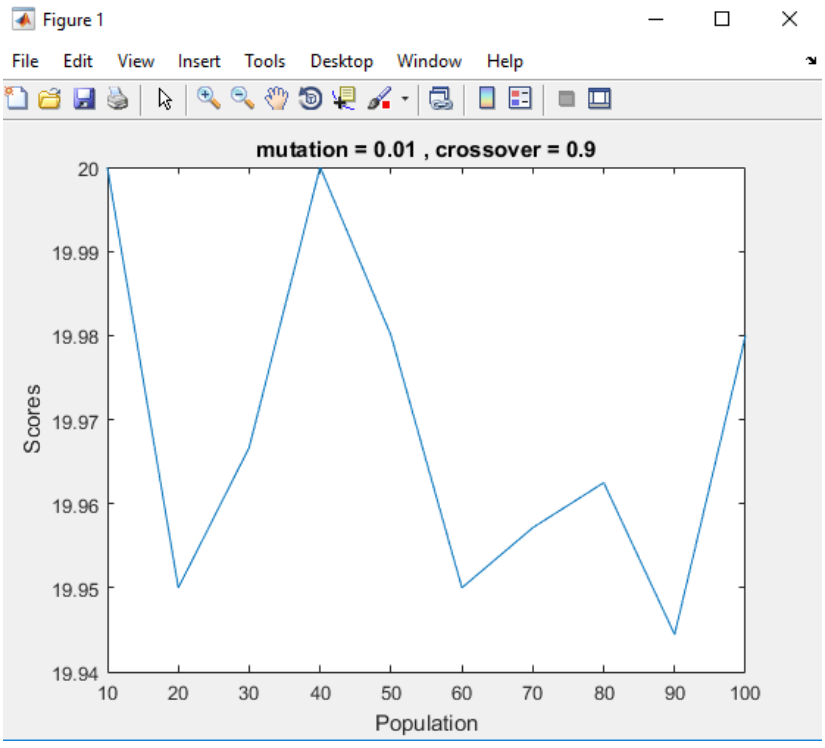
Παρατηρούμε σε γενικές γραμμές ότι οσο αυξάνουμε την πιθανότητα μετάλλαξης, τα scores μας μειώνονται. Κάποιες φορές παρατηρούμε Peaks που όμως θα μπορούσε να οφείλεται στην τυχαιότητα της μετάλλαξης. (να πετάμε κάτι καλό για κάτι κακό ή το αντίστροφο)

Μεταβολή στην πιθανότητα διασταύρωσης:



Εδω παρατηρούμε μια σχετική αύξηση των scores με την αύξηση του crossover. Έχουμε λοιπόν καλύτερο αποτέλεσμα.

Μεταβολή στον πληθυσμό:



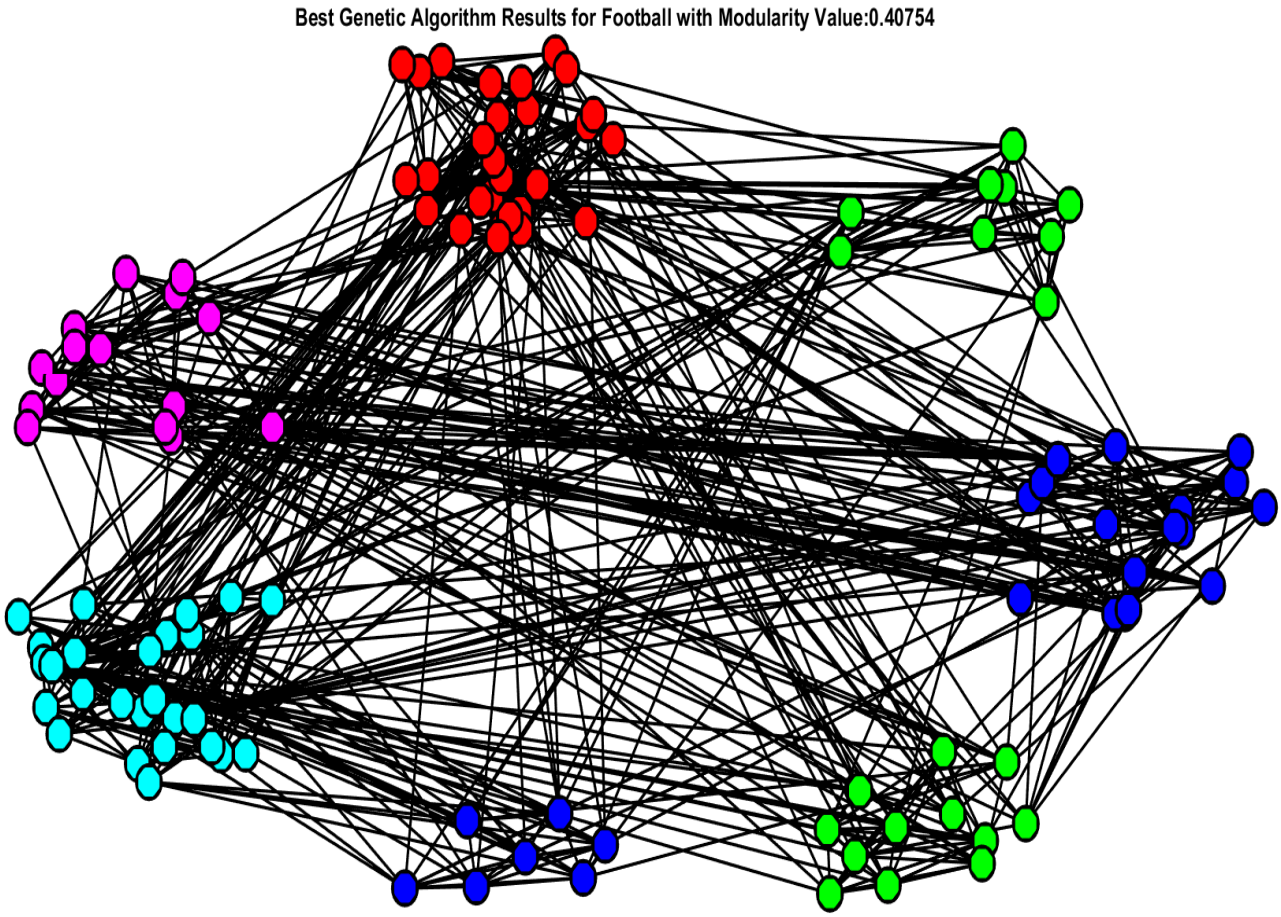
Παρατηρούμε κάποιες κλιμακώσεις με τη μεταβολή του πληθυσμού στα scores. Η λύση όμως που επιστρέφει ο γενετικός αλγόριθμος σε κάθε περίπτωση είναι η βελτιστη απο κάποιο σημείο και μετά.

Γενικά, στα παραπάνω διαγράμματα χρησιμοποιήθηκε ως μετρική το score και οχι η λύση που μας δίνει ο γενετικός αλγόριθμος για το πρόβλημα μας, καθώς το πρόβλημά μας σε γενικές γραμμές είναι πολύ απλό και σχεδόν με όλες τις παραμέτρους (αφου επιλέξαμε bitstring datatype στο optimization tool) μας επέστρεφε την προφανή και εύκολη λύση κι έτσι δεν προσφερόταν αυτό ως κριτήριο αξιολόγησης.

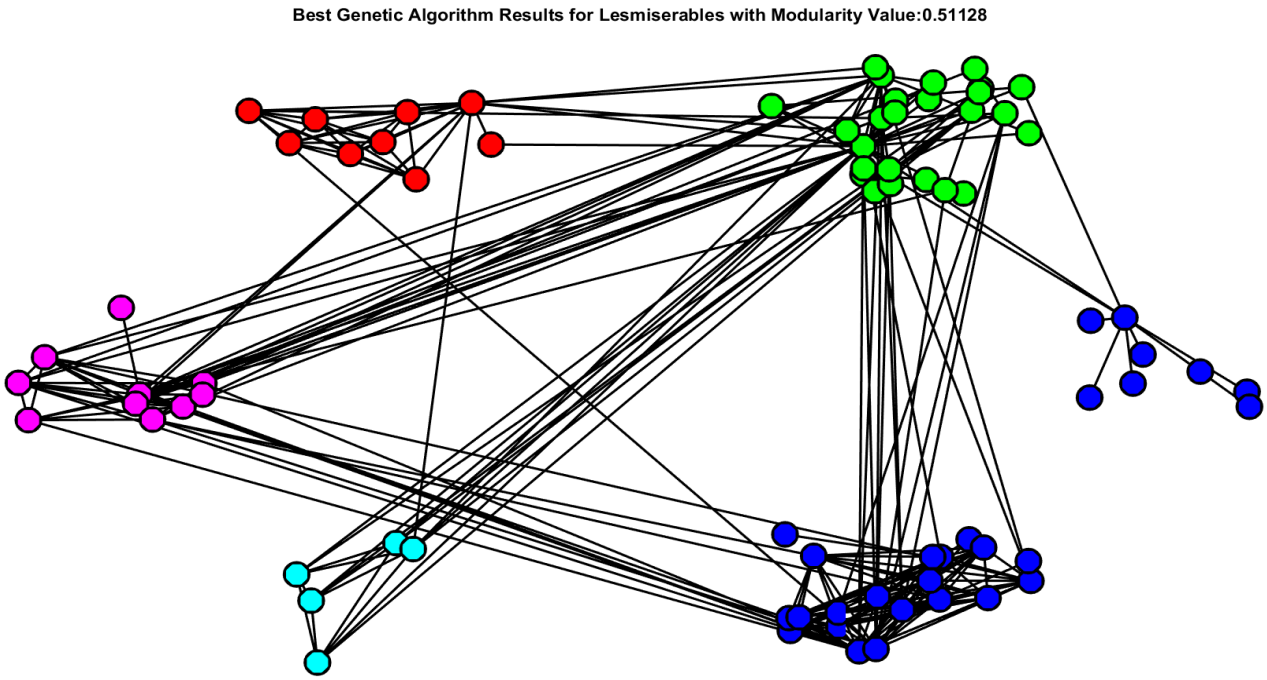
Άσκηση 2η:Εντοπισμός Κοινοτήτων σε Γράφους Κοινωνικών Δικτύων με Χρήση Γενετικών Αλγορίθμων

Σε αυτό το ερώτημα της άσκησης υλοποιήθηκε ο δικός μας γενετικός αλγόριθμος και χρησιμοποιηθήκε για τον εντοπισμό κοινοτήτων στα δίκτυα football, lesmis και dolphins που χρησιμοποιήθηκαν και στην προηγούμενη εργαστηριακή άσκηση. Η υλοποίηση του αλγορίθμου έγινε με βάση το σκελετό που μας δίνεται στην εκφώνηση της άσκησης και τις επεξηγήσεις που παρέχονται στις διαφάνειες του μαθήματος. Τα αποτελέσματα που μας προέκυψαν τα συγκρίναμε με βάση το modularity, μετρική που χρησιμοποιήθηκε και στην προηγούμενη άσκηση. Στη συνέχεια παραθέτουμε τον καλύτερο διαχωρισμό κοινοτήτων που πετύχαμε για κάθε δίκτυο με τη χρήση του γενετικού μας αλγορίθμου.

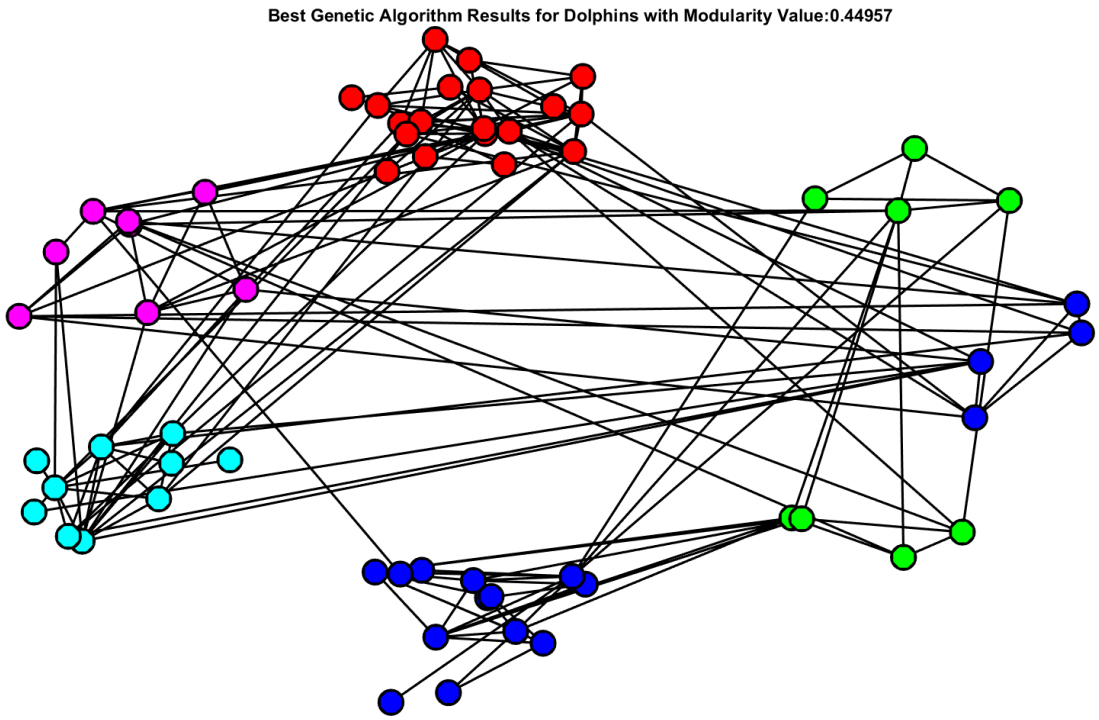
Football:



Les Miserables:



Dolphins:



Σχολιασμός Αποτελεσμάτων:

Στα παραπάνω σχήματα φαίνεται και η τιμή της μετρικής του Modularity που πήραμε για κάθε μια από αυτές τις ανιχνεύσεις κοινοτήτων. Αν συγκρίνουμε τα αποτελέσματα αυτά με εκείνα που πήραμε με τις μεθόδους της προηγούμενης άσκησης, μπορούμε να παρατηρήσουμε ότι οι τιμές του Modularity που προέκυψαν με τη χρήση του γενετικού αλγορίθμου είναι σε γενικές γραμμές χαμηλότερες από αυτές που προέκυψαν από τις μεθόδους εντοπισμού κοινοτήτων που χρησιμοποιήσαμε στη $2^{η}$ εργαστηριακή άσκηση και για τις 3 τοπολογίες δικτύου. Αυτό δεν είναι περίεργο, καθώς ο γενετικός αλγόριθμος σε κάθε του βήμα :

1) Χρησιμοποιεί για επιλογή τη μέθοδο της ρουλέτας κι έτσι μπορεί να «πετάξει» κάποιο καλό χρωμόσωμα και να κρατήσει κάποιο λιγότερο καλό(για να μετριάσουμε αυτο το φαινόμενο χρησιμοποιούμε και τη μέθοδο του ελιτισμού)
2) Υπάρχει πιθανότητα για mutation ενός καλού χρωμοσώματος

Γενικότερα, εισάγει τον παράγοντα τύχη που είναι πολύ καθοριστικός εδώ. Επομένως τα αποτελέσματα που θα λάβουμε από αυτή τη μέθοδο έχουν απόκλιση κάθε φορά που τρέχουμε τον αλγόριθμο ενώ στην προηγούμενη άσκηση κάθε μέθοδος έδινε παρόμοια αποτελέσματα όσες φορές και αν την δοκιμάζαμε.

Για να το βελτιώσουμε αυτό μπορούμε είτε να χρησιμοποιήσουμε μεγαλύτερες τιμές για τον ελιτισμό είτε να εφαρμόσουμε κάποια καλύτερη fitness function ή να βρούμε κάποιο καλύτερο κριτήριο τερματισμού. Ενδεικτικά, αν για fitness function χρησιμοποιούσαμε το modularity τα αποτελέσματα που θα είχαμε θα ήταν καλύτερα.

Στη συνέχεια παρατίθεταιτο κομμάτι του script που μας έδωσε τα αποτελέσματα, η function του γενετικού αλγορίθμου και η fitness function:

Script:

```
%% LESMIS

lesmis =importgml('lesmis.gml');
disp('*******Lesmis*********')
if isdirected(lesmis)
    B=undir(lesmis,size(lesmis,1));
    lesmis=B;
    clearvars B;
end
maxi_lesm=0;
for i=0.7:0.1:0.9
    for j=0.1:0.1:0.2
        for k=1:3
            res=cellcluster(my_ga(lesmis,i,j,k,30,300,5,1),77);
            if QFModul(res,lesmis)>maxi_lesm
                maxi_lesm=QFModul(res,lesmis);
                maxcluster_lesm=res;
            end
%            break;
```

```
        end
%         break;
    end
%    break;
end
%res=genetic(lesmis,77,0.7,0.2,2);
PlotGraph(lesmis,maxcluster_lesm);
frame_h = get(handle(gcf),'JavaFrame');
set(frame_h,'Maximized',1);
X = ['Best Genetic Algorithm Results for Lesmiserables with Modularity
Value:' num2str(maxi_lesm)];
title(X);
saveas(gcf,'Genetic_lesmis.png');

%% FOOTBALL
football =importgml('football.gml');
disp('*******Football*********')
if isdirected(football)
    B=undir(football,size(football,1));
    football=B;
    clearvars B;
end
maxi_foot=0;
for i=0.7:0.1:0.9
    for j=0.1:0.1:0.2
        for k=1:3
            res=cellcluster(my_ga(football,i,j,k,30,300,5,1),115);
            if QFModul(res,football)>maxi_foot
                maxi_foot=QFModul(res,football);
                maxcluster_foot=res;
            end
        end
    end
end
%res=genetic(lesmis,77,0.7,0.2,2);
%figure;
PlotGraph(football,maxcluster_foot);
frame_h = get(handle(gcf),'JavaFrame');
set(frame_h,'Maximized',1);
X = ['Best Genetic Algorithm Results for Football with Modularity Value:'
num2str(maxi_foot)];
title(X);
saveas(gcf,'Genetic_football.png');

%% DOLPHINS
dolphins =importgml('dolphins.gml');
disp('*******Dolphins*********')

if isdirected(dolphins)
    B=undir(dolphins,size(dolphins,1));
    dolphins=B;
    clearvars B;
end
maxi_dol=0;
for i=0.7:0.1:0.9
    for j=0.1:0.1:0.2
```

```
        for k=1:3
            res=cellcluster(my_ga(dolphins,i,j,k,30,300,5,1),62);
            if QFModul(res,dolphins)>maxi_dol
                maxi_dol=QFModul(res,dolphins);
                maxcluster_dol=res;
            end
        end
    end
end
figure;
PlotGraph(dolphins,maxcluster_dol);
frame_h = get(handle(gcf),'JavaFrame');
set(frame_h,'Maximized',1);
X = ['Best Genetic Algorithm Results for Dolphins with Modularity Value:'
num2str(maxi_dol)];
title(X);
saveas(gcf,'Genetic_dolphins.png');
```

Genetic Algorithm:

```
function [ communities ] =
my_ga(Adj,pc,pm,elitism,generations,chrom,fit_max_unchanged,order_for_fitness
_function)
    %my genetic algorithm for communities detection
    %%INPUT
    %Adj : adjacency matrix
    %pc : crossover probability
    %pm : mutation probability
    %elitism : number of conquering first chromosomes for the next generation
    %generations : number of generations
    %chrom : number of chromosomes
    %fit_max_unchanged : how many times fitness function maximum remains the
same

    %%OUTPUT
    %communities : communities as a cell array

    %%Initialization
    % n = numer of nodes
    n=size(Adj,1);
    B = cell(chrom,generations+1);
    %get the neighbours of each node and initialize our first generation
    %taking randomly one of them
    neighbours = cell(n);
    for i=1:n
        neighbours{i} = find(Adj(i,:)==1);
    end
    t = 1; %t represents our generation
    for i=1:chrom
        for j=1:n
            B{i,t}(j) = neighbours{j}(randi(size(neighbours{j},2)));
        end
    end

    %%Main Body of my Genetic Algorithm
```

```
    %create the new adjacency matrix according to the chromosomes to find the
connected components
    Adj_2 = zeros(n,n);
    %number of times max of fitness functions remains unchanged
    max_fitness_cnt = 0;
    max_fitness = -inf;
    while (t <= generations+1 && max_fitness_cnt < fit_max_unchanged)

        for i = 1:chrom
            for j = 1:n
                Adj_2(j,B{i,t}(j)) = 1;
                Adj_2(B{i,t}(j),j) = 1;
            end
            %we now find the connected components
            conn_comp = find_conn_comp(Adj_2);
            %calculation of fitness function and scores
            for j = 1:size(conn_comp,2)
                %take the appropriate subgraph
                Sub{j} = subgraph(Adj_2,conn_comp{j});
                %calculation of the CS of this component
                CStemp(j) =
fitness_function(Sub{j},order_for_fitness_function);
            end
            CS(i) = sum(CStemp);
            %clearvars CStemp  %%%%%%%%%%%%%%%%%%%%%%%%%%
            Adj_2 = zeros(n,n); %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        end

        %Selection with elitism
        [sorted_CS,indexes] = sort(CS,'descend');
        %keep the  #elitism first chromosomes
        for i=1:elitism
            B{i,t+1} = B{indexes(i),t};
        end
        %roulette method
        total_CS= sum(CS);
        for i=elitism+1:chrom %continue with the remaining chromosomes
            x=rand(1);
            k=1;
            while (k < chrom && x < sum(CS(1:i))/total_CS )
                k=k+1;
            end
            B{i,t+1} = B{k,t};
        end

        %one-point Crossover
        for i=1:2:chrom-1
            if (rand(1) <= pc)
                pos = randi(n-1);
                for k=pos+1:n
                    aux = B{i,t+1}(k);
                    B{i,t+1}(k) = B{i+1,t+1}(k);
                    B{i+1,t+1}(k) = aux;
                end
            end
        end
```

```
        %Mutation
        for i=1:chrom
            for k=1:n
                if (rand(1) <= pm)
                    B{i,t+1}(k) =
neighbours{k}(randi(size(neighbours{k},2)));
                end
            end
        end

        if (sorted_CS(1) == max_fitness)
            max_fitness_cnt = max_fitness_cnt + 1;
        else
            max_fitness = sorted_CS(1);
            max_fitness_cnt = 1;
        end

        %break;
        %clearvars CS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        t = t+1;
    end
    %get my results
    t = t - 1;
    Adj_2 = zeros(n,n);
    for j = 1:n
        Adj_2(j,B{indexes(1),t}(j)) = 1;
        Adj_2(B{indexes(1),t}(j),j) = 1;
    end
    communities = find_conn_comp(Adj_2);
end
```

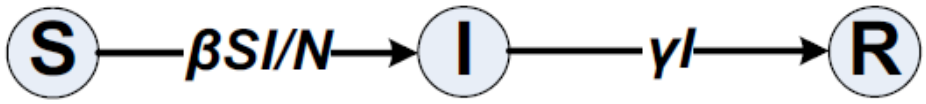
Fitness Function:

```
function CS= fitness_function(Sub,r)
    %Calculates the Community Score of subgraph S of our graph
    us = sum(sum(Sub));

    M = sum(mean(Sub,2).^r)/length(Sub);

    CS = M*us;
end
```

Μέρος Β

Ερώτημα Α:SIR Επιδημιολογικό Μοντέλο

Στο ερώτημα αυτό θα ασχοληθούμε με το επιδημιολογικό μοντέλο SIR. Το μοντέλο αυτό έχει το ακόλουθο διάγραμμα μεταβάσεων:



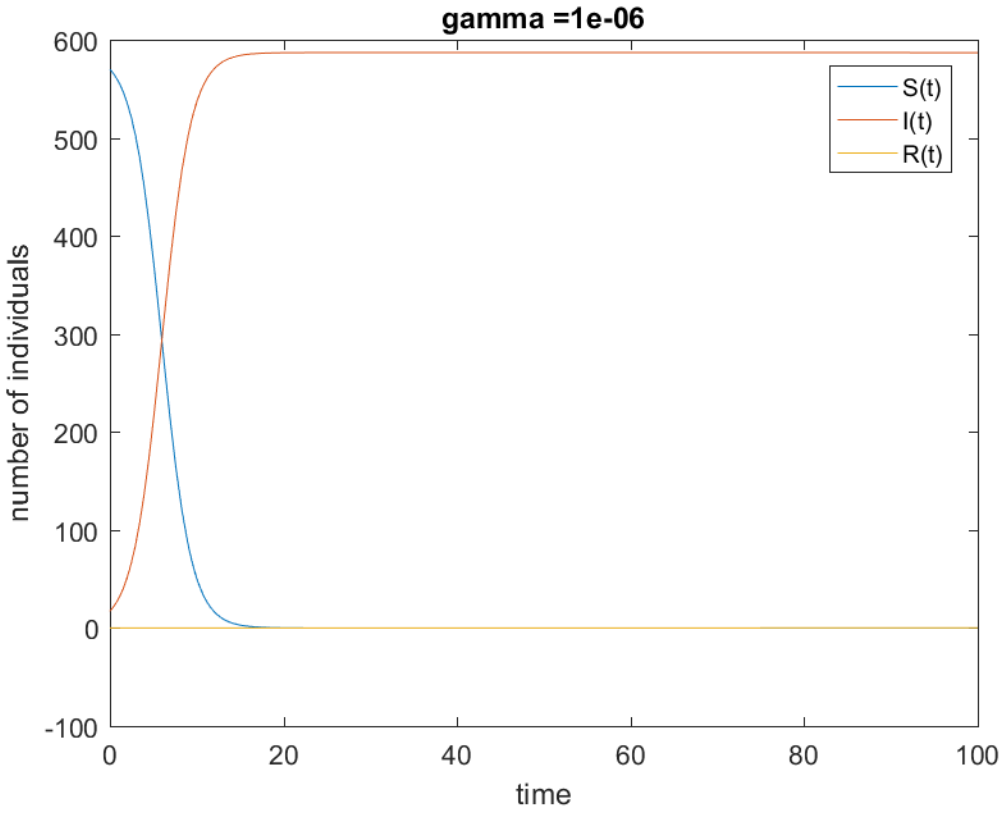
Το μοντέλο αυτό περιγράφεται από τις ακόλουθες διαφορικές εξισώσεις:

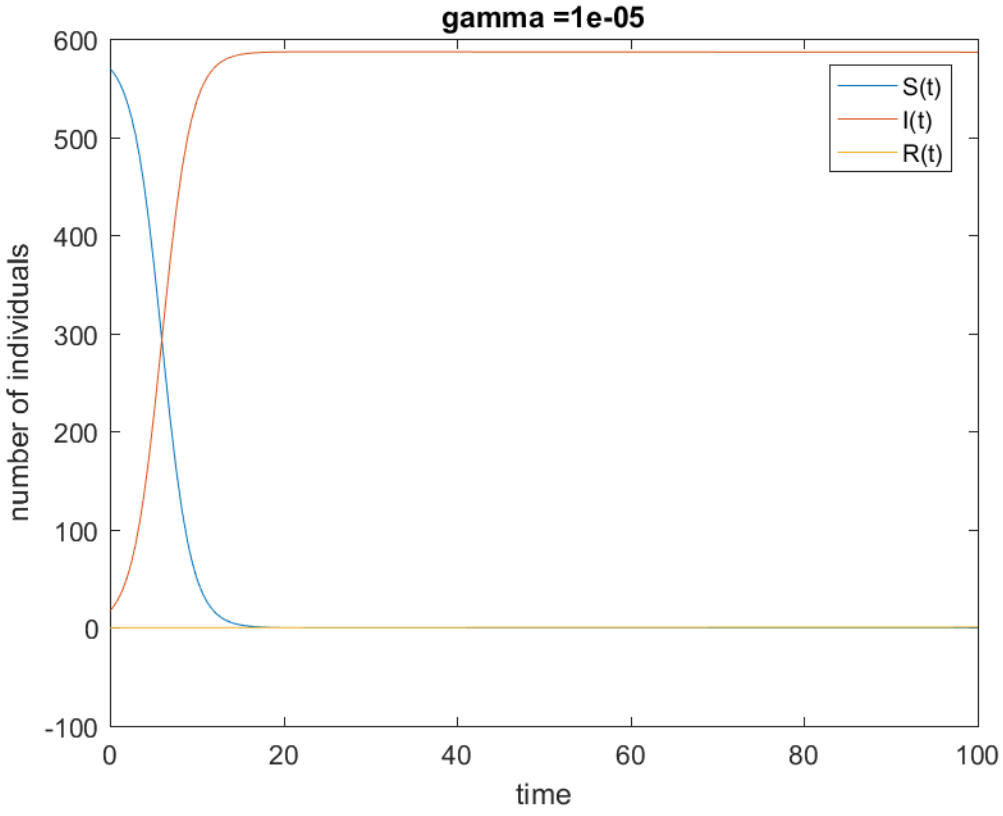
$$\frac{dS}{dt} = -\frac{\beta SI}{N} \qquad \frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I \qquad \frac{dR}{dt} = \gamma I$$

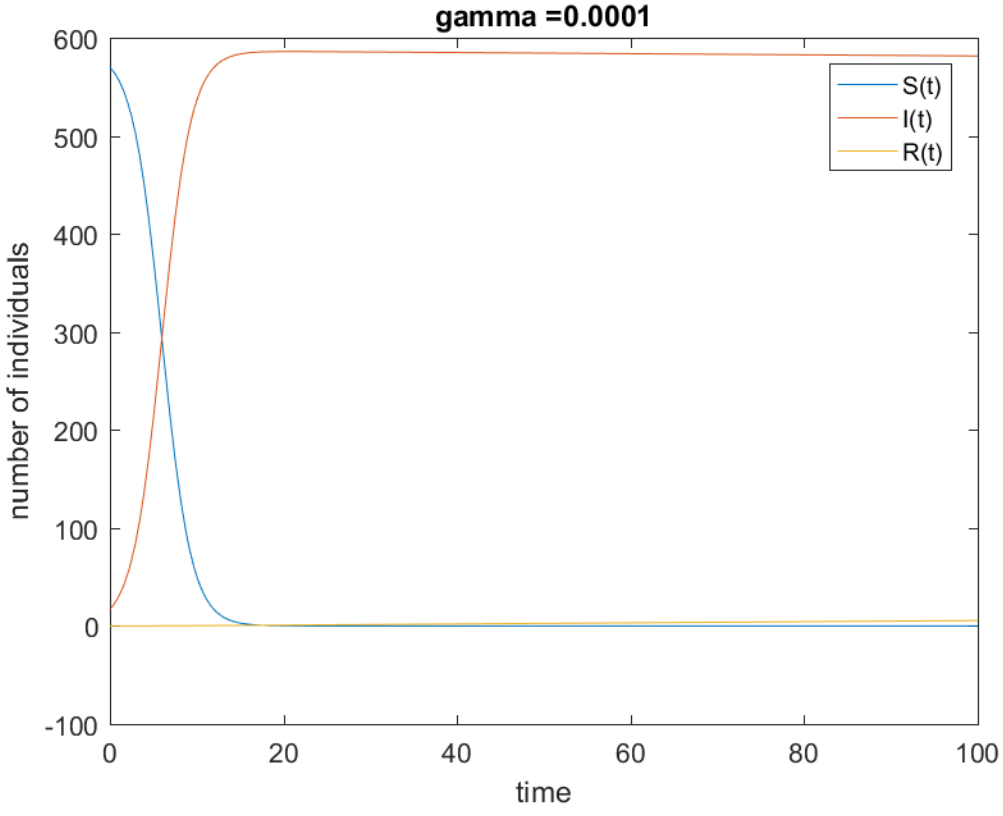
Στη συνέχεια παραθέτουμε τα S(t),I(t),R(t) σε κοινό διάγραμμα. Οι παράμετροι που χρησιμοποιήθηκαν είναι:

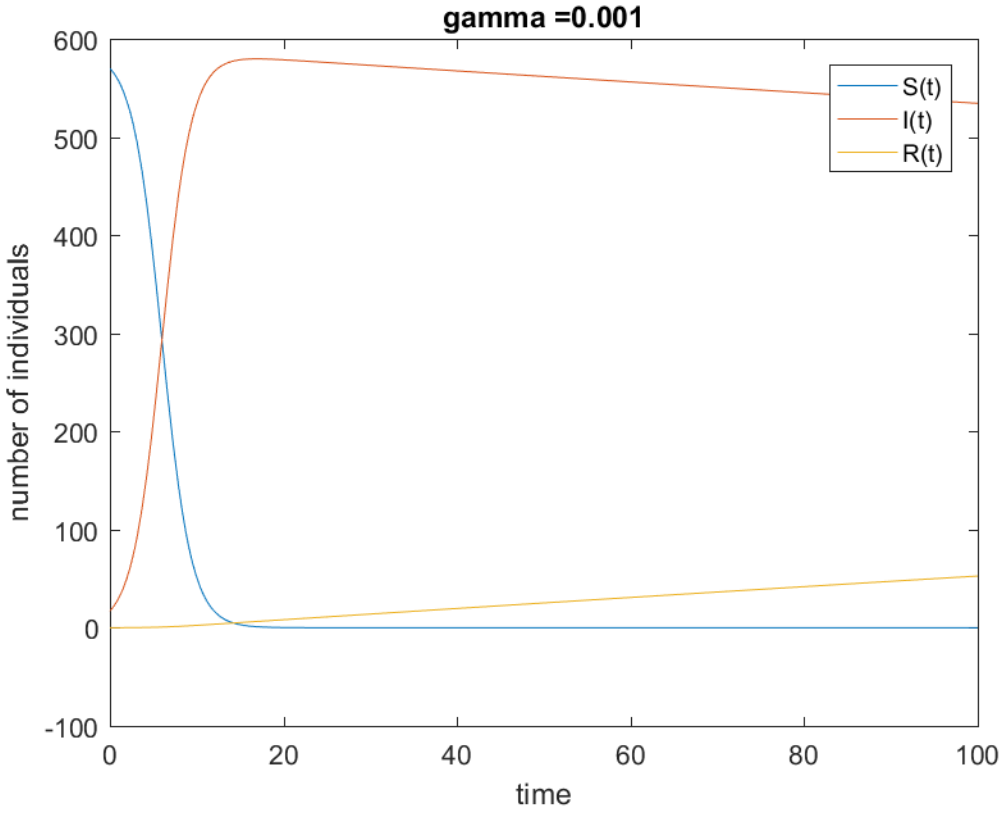
S(0) = 170 , I(0) = 17 , R(0) = 0 , β = 10^{-3}

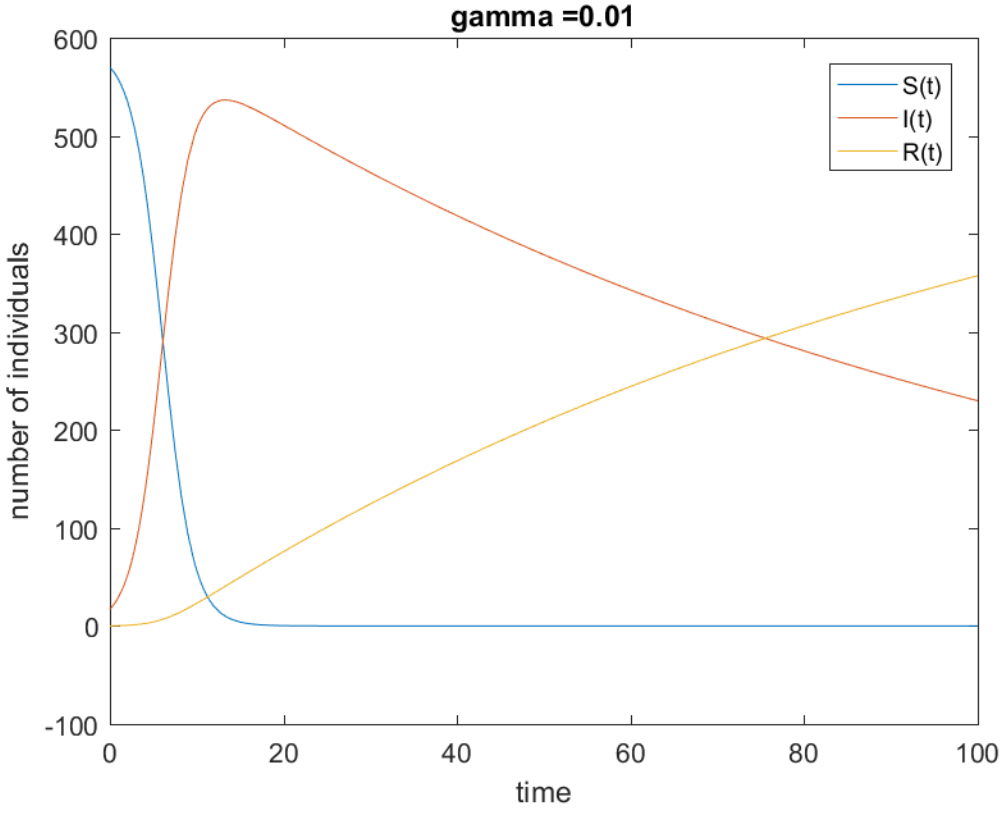
Οι τιμές του **γ** μεταβάλλονταν όπως ζητείται στην εκφώνηση και έτσι λάβαμε τα εξής αποτελέσματα:

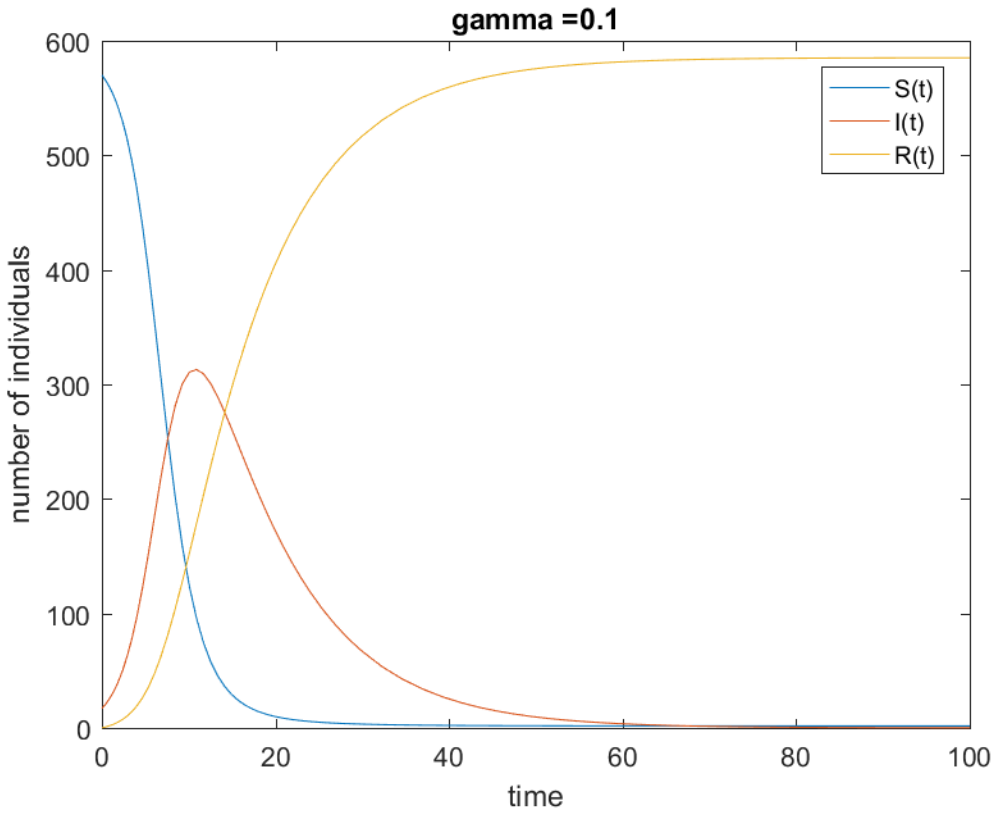



gamma =1e-05
600
500
400
300
200
100
0
-100
number of individuals
0
20
40
60
80
100
time
S(t)
I(t)
R(t)


gamma =0.0001
600
500
400
300
200
100
0
-100
number of individuals
0
20
40
60
80
100
time
S(t)
I(t)
R(t)


gamma =0.001
S(t)
I(t)
R(t)
number of individuals
time
600
500
400
300
200
100
0
-100
0
20
40
60
80
100


gamma =0.01
S(t)
I(t)
R(t)
number of individuals
time
600
500
400
300
200
100
0
-100
0
20
40
60
80
100



Σημ:οι τιμές του γ φαίνονται στους τίτλους κάθε γραφικής παράστασης

Σχολιασμός Αποτελεσμάτων:

S: Πληθυσμός που δεν εχει μολυνθεί ακόμα

I: Πληθυσμός που έχει μολυνθεί και μεταδίδει τον ιό στον πληθυσμό S

R: Πληθυσμός που δε μπορεί να μολυνθεί απο τον ιό

β: Ρυθμός με τον οποίο οι κόμβοι στο δίκτυο αλληλεπιδρούν (στο πείραμά μας ισούται με 0.001)

γ: Μέσος ρυθμός ανάρρωσης

Για το μοντέλο μας ακόμα ισχύει:

$N = S + I + R$

$R_0 = {}^{\beta}/_{\gamma}$

Από αυτούς τους τύπους σε συνδυασμό με τις διαφορικές εξισώσεις που ορίζουν το SIR μοντέλο προκύπτουν τα ακόλουθα:

$$\frac{dI}{dt} = \frac{R_0 S}{N-1}\gamma I$$

Έτσι διακρίνουμε τις ακόλουθες περιπτώσεις:

- $R_0 > {}^{N}/_{S(0)}$, $\frac{dI}{dt}(0) > 0$: ρυθμός ανάρρωσης > ρυθμός επαφής: καταπίεση του ιού
- $R_0 < {}^{N}/_{S(0)}$, $\frac{dI}{dt}(0) < 0$: ρυθμός ανάρρωσης < ρυθμός επαφής: εξάπλωση του ιού

Αναφερόμενοι στα παραπάνω διαγράμματα:
Στο πρώτο και το δεύτερο διάγραμμα έχουμε ότι γ>β, άρα ρυθμός ανάρρωσης < ρυθμός εξάπλωσης, το μεγαλύτερο μέρος του πληθυσμού θεραπεύεται με το πέρασμα του χρόνου μετά το αρχικό ξέσπασμα του ιού. Βλέπουμε ότι όσο μεγαλύτερο είναι το γ από το β η σύγκλιση είναι ταχύτερη. Στην περίπτωση του διαγράμματος όπου β=γ έχουμε σχετικά αργή θεραπεία του πληθυσμού. Τέλος, εκεί που έχουμε β>γ ο πληθυσμος μας τείνει να μολυνθεί εξ ολοκλήρου απο τον ιό.

Ερώτημα Β: SIS Επιδημιολογικό Μοντέλο

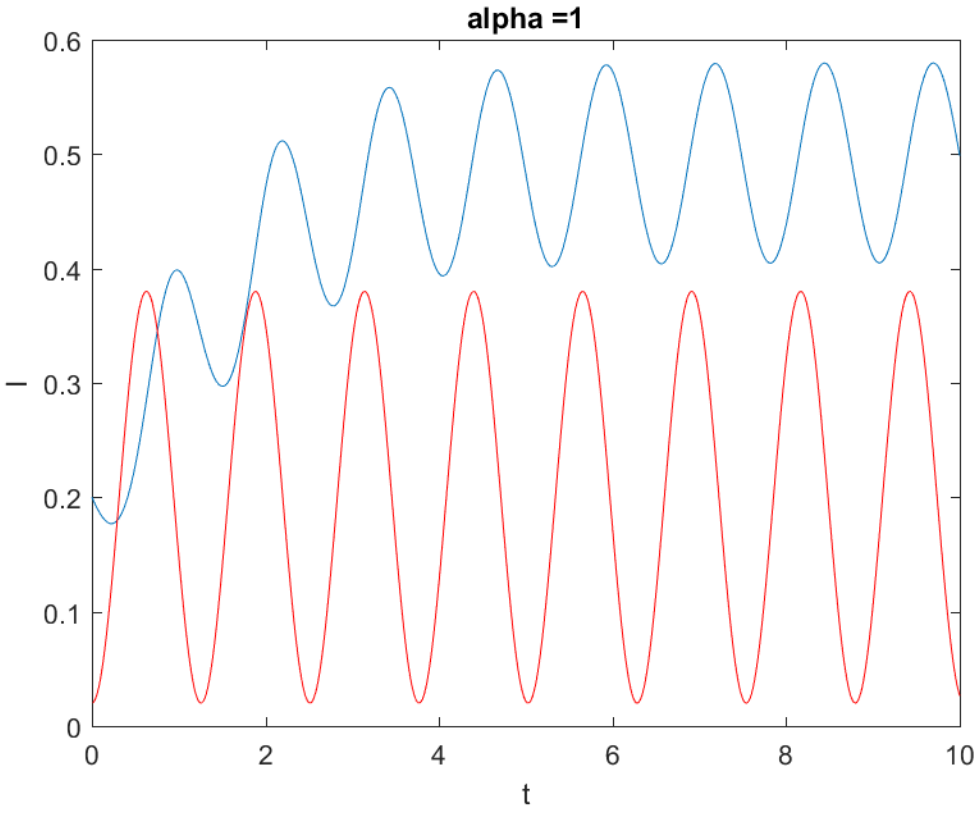
Στο ερώτημα αυτό θα ασχοληθούμε με το επιδημιολογικό μοντέλο SIR. Το μοντέλο αυτό περιγράφεται από τις ακόλουθες διαφορικές εξισώσεις:

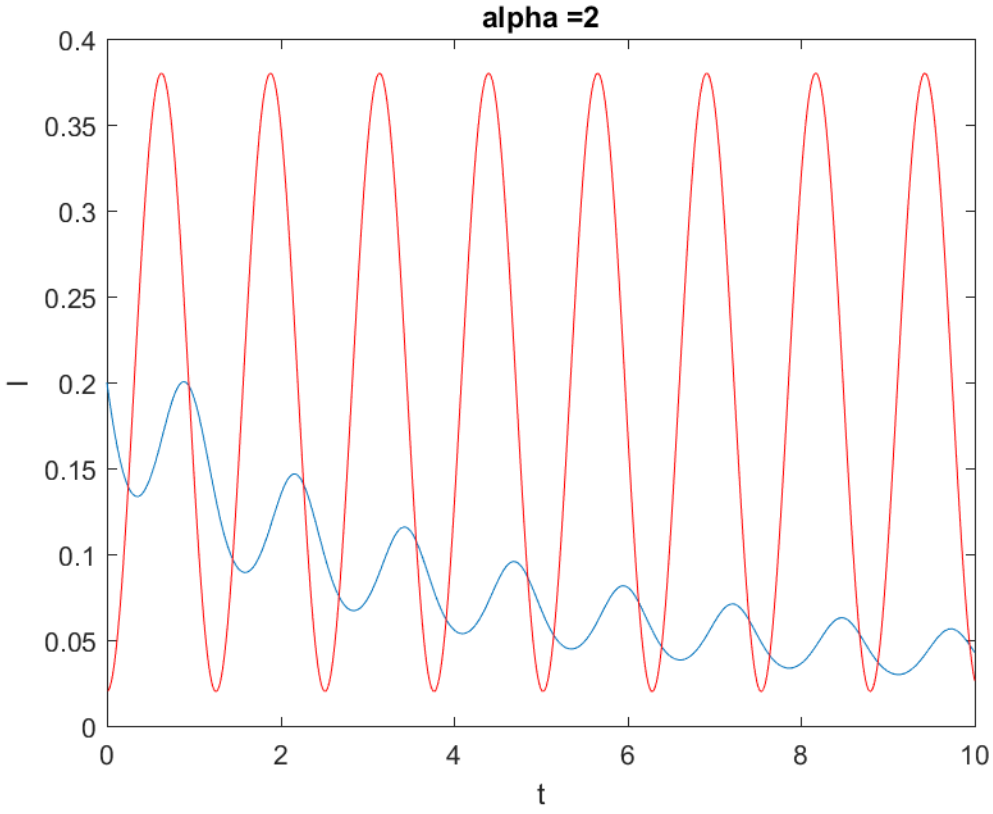
$$\frac{dI}{dt} = (\beta(t)N - a)I - \beta(t)I^2 \qquad \beta(t) = 2 - 1.8cos(5t)$$

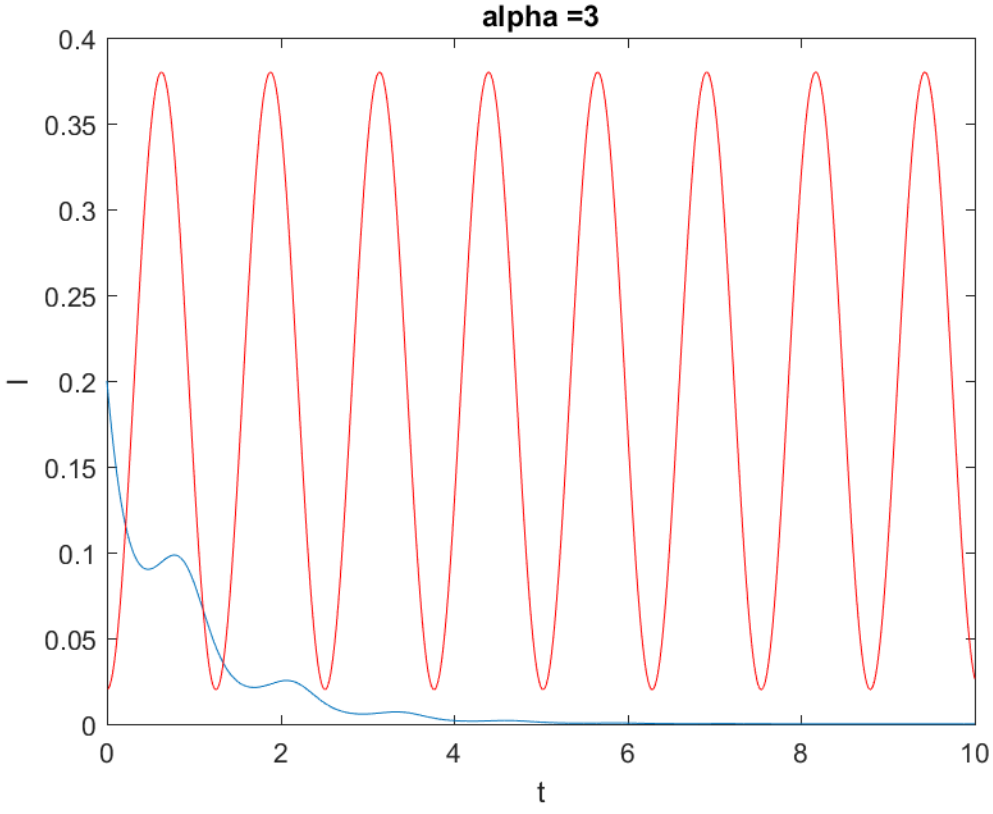
Στη συνέχεια παραθέτουμε τα I(t),β(t) σε κοινό διάγραμμα. Οι παράμετροι που χρησιμοποιήθηκαν είναι:

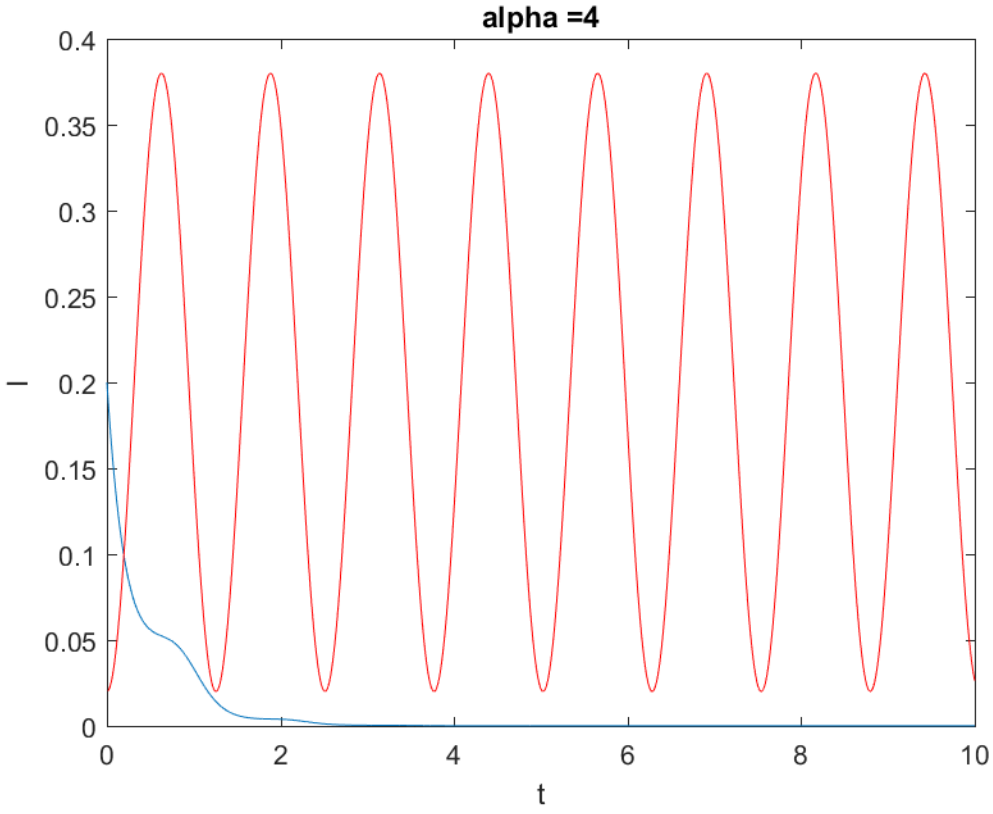
N=1

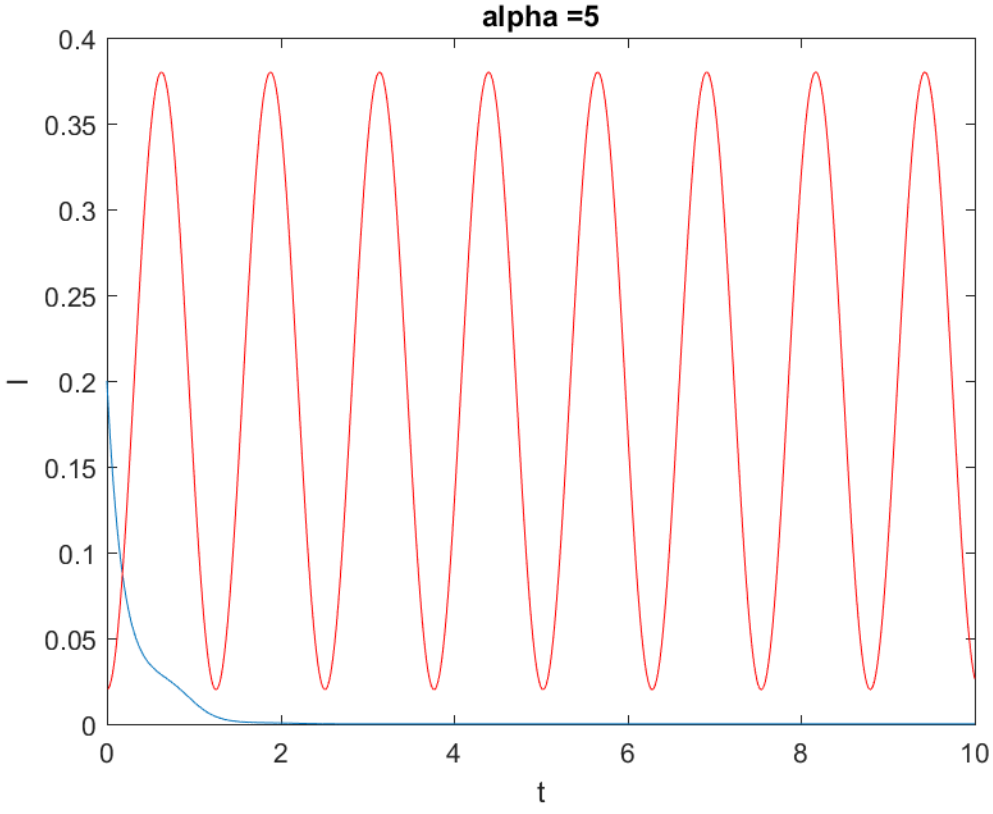
Οι τιμές του **α** μεταβάλλονταν όπως ζητείται στην εκφώνηση και έτσι λάβαμε τα εξής αποτελέσματα:

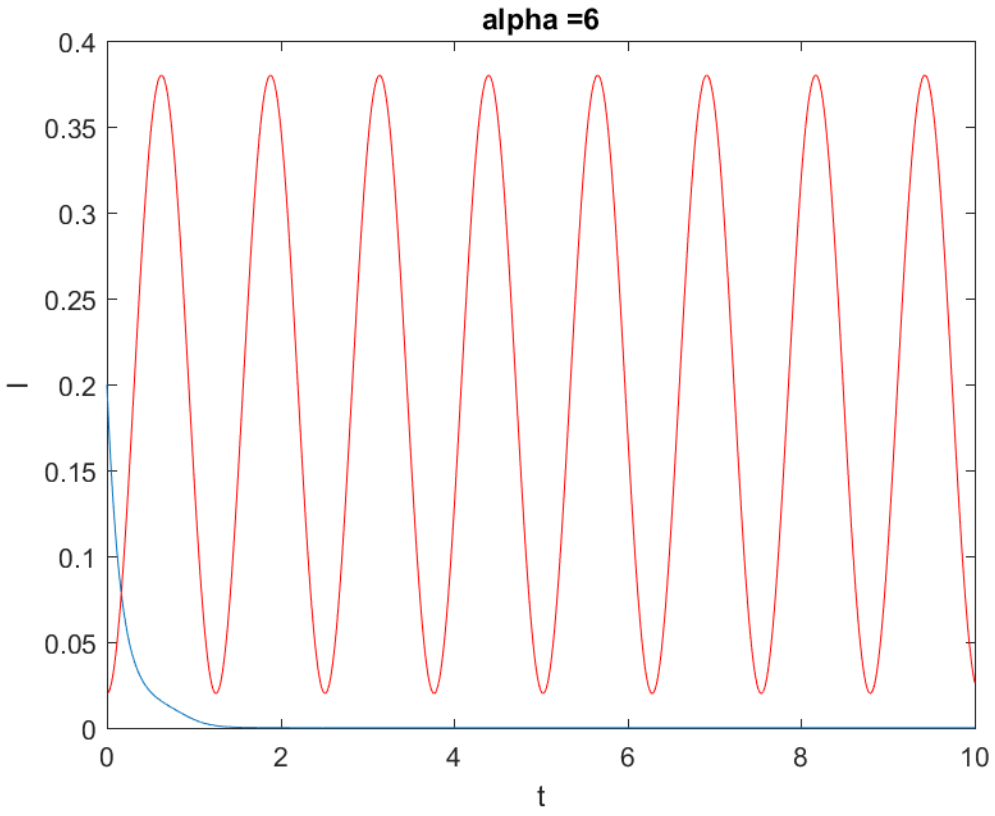



alpha =2
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
I
0
2
4
6
8
10
t


alpha =3
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
I
0
2
4
6
8
10
t


alpha =4
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
I
0
2
4
6
8
10
t


alpha =5
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
I
0
2
4
6
8
10
t



Σημ:οι τιμές του γ φαίνονται στους τίτλους κάθε γραφικής παράστασης

Σχολιασμός Αποτελεσμάτων:

Από τα παραπάνω διαγράμματα μπορούμε να συμπεράνουμε ότι καθώς η τιμή της παραμέτρου α του μοντέλου αυξάνεται διατηρώντας την τιμή του Ν σταθερή, ο αριθμός των μολύνσεων λαμβάνει χαμηλότερες τιμές με το πέρασμα του χρόνου και επίσης **δεν έχει τόσες διακυμάνσεις** (όπως έχει πχ στο πρώτο διάγραμμα). Επομένως αν επιθυμούμε να μην έχουμε διακυμάνσεις στον αριθμό των μολύνσεων μπορούμε απλά να εφαρμόσουμε μοντέλο με μεγάλη τιμή του α για σταθερό Ν. Ο λόγος, μαθηματικά, που συμβαίνει αυτό είναι επειδή μειώνεται η επίδραση που έχει στο μοντέλο μας το συνημίτονο που υπάρχει στις διαφορικές εξισώσεις του μοντέλου. Λόγω αυτού, μπορούμε να συμπεράνουμε ότι με αυξηση του α μειώνεται η πιθανότητα να προκύψει επιδημία. Γενικά αυτό το μοντέλο , όπως βλέπουμε, είναι κατάλληλο για την περιγραφή ιών οι οποίοι παρουσιάζουν εξάρσεις και περιόδους ηρεμίας (διακυμάνσεις για μικρές τιμές του α).