

Τεχνικές Βελτιστοποίησης

Project: Γενετικοί Αλγόριθμοι

Θεολόγης Γεώργιος
AEM:10413
email: gtheolog@ece.auth.gr

Ιανουάριος 2024

1 Εισαγωγή

Αντικείμενο της παρούσας εργασίας είναι η υλοποίηση γενετικού αλγορίθμου μέσω του οποίου θα καθοριστούν οι παράμετροι μοντέλου ενός συστήματος. Για τον προσδιορισμό του μοντέλου, έχουμε στην διάθεση μας το πραγματικό σύστημα και έτσι μπορούμε να αξιοποιήσουμε μετρήσεις εισόδου-εξόδου με σκοπό την αξιολόγηση της εγγύτητας του μοντέλου στο πραγματικό σύστημα.

Για το πραγματικό σύστημα μας δίνεται ότι μπορούμε να προσημειώσουμε την διαδικασία άντλησης των μετρήσεων εισόδου-εξόδου με την χρήση της μαθηματικής σχέσης:

$$f(u_1, u_2) = \sin(u_1 + u_2) \cdot \sin(u_2^2)$$

Το εύρος των τιμών εισόδου που μας ενδιαφέρει είναι τα εξής :

$$u_1 \in [-1, 2]$$

$$u_2 \in [-2, 1]$$

Το μοντέλο που θέλουμε να κατασκευάσουμε αποτελείται απο γραμμικό συνδυασμό N δισδιάστατων Gaussian συναρτήσεων μέσω των τιμών c_1, c_2 και τυπικών αποκλίσεων σ_1, σ_2 . Η i -στη Gaussian συνάρτηση είναι η :

$$G_i(u_1, u_2) = e^{-\left(\frac{(u_1 - c_{i,1})^2}{2\sigma_{i,1}^2} + \frac{(u_2 - c_{i,2})^2}{2\sigma_{i,2}^2}\right)}$$

Ενώ η συνάρτηση του μοντέλου είναι η :

$$\hat{f}_m(u_1, u_2) = \sum_{i=1}^N a_i G_i(u_1, u_2)$$

Κάθε άτομο του πληθυσμού παρίσταται ως ένα cell array 5 πινάκων, ο πίνακας **A** για τις τιμές των συντελεστών a_i της κάθε Γκαουσιανής, οι πίνακες **C1, C2** που περιέχουν τις μέσες τιμές της κάθε Γκαουσιανής και τέλος οι πίνακες **S1, S2** που περιέχουν τις τιμές των τυπικών αποκλίσεων της κάθε Γκαουσιανής. Ο γενετικός

A	C1	C2	S1	S2
---	----	----	----	----

Table 1: Άτομο του Πληθυσμού

αλγόριθμος αποτελεί είδος εξελικτικού αλγορίθμου του οποίου τα λειτουργικά χαρακτηριστικά και λογική είναι βασισμένα σε φυσικές διεργασίες όπως η εξέλιξη, η φυσική επιλογή καθώς και τυχαιότητα στην επιβίωση. Συγκεκριμένα, ο αλγόριθμος χειρίζεται ένα **πληθυσμό** στον οποίο κάθε άτομο καλείται **χρωμόσωμα**. Κάθε χρωμόσωμα σε αυτή την εργασία, είναι μια πιθανή μορφή του μοντέλου. Ο αλγόριθμος εφαρμόζει στην προσπάθεια του να μιμηθεί τις φυσικές διεργασίες, την επιλογή (selection) κρατώντας κάποια χρωμοσώματα που θεωρεί

βέλτιστα με κάποια πιθανότητα και στη συνέχεια εξίσου στοχαστικά τα διασταυρώνει (crossover) και τα μεταλλάσσει (mutate). Κάθε εφαρμογή αυτών των διαδικασιών στον πληθυσμό ονομάζεται γενιά (Generation). Σκοπός, μας είναι σε πεπερασμένο αριθμό γενεών να παραχθεί στον πληθυσμό ένα βέλτιστο άτομο-χρωμόσωμα του οποίο μας παρέχει ένα ικανοποιητικό μοντέλο για το σύστημα. Η αξία του κάθε χρωμοσώματος θα μπορεί να υπολογίζεται από την fitness function της οποίας ο καθορισμός είναι μείζον σημασίας για την λειτουργία του αλγορίθμου. Στη υπόλοιπη εργασία θα συνεχιστεί η περαιτέρω ανάλυση της λειτουργίας κάθε μιας από τις προαναφερόμενες διεργασίες σε κωδικά αλλά και η σημασία τους προς την όλη λειτουργικότητα του αλγορίθμου. Το κύριο script της εργασίας είναι το project_ga.m το οποίο ακολουθεί παρακάτω:

```

1 clear;
2 pop_size=100;
3 Initial_Pop=cell(pop_size);
4 num_Gaussian=15;
5
6 for i=1:pop_size
7     A=rand(1,num_Gaussian);
8     C1=zeros(1,num_Gaussian);
9     C2=zeros(1,num_Gaussian);
10    S1=0.5*ones(1,num_Gaussian);
11    S2=0.5*ones(1,num_Gaussian);
12    Initial_Pop{i}={A,C1,C2,S1,S2};
13 end
14
15 Pop=Initial_Pop;
16 Generations=1000;
17
18 for j=1:Generations
19     j
20     MSE=evaluate_fitness(Pop);
21     minMSE(j)=min((1000./MSE-0.0001).^(2/3));
22     sel=Selection(MSE);
23     crossed_Pop=Crossover(sel,Pop);
24     mutated_Pop=mutate_Pop(crossed_Pop);
25     Pop=mutated_Pop;
26 end
27 MSE=evaluate_fitness(Pop);
28 [max_val,index] = max(MSE(:));
29 Best=Pop{index};
30 A=Best{1};
31 C1=Best{2};
32 C2=Best{3};
33 S1=Best{4};
34 S2=Best{5};
35
36 figure;
37 [U1,U2] = meshgrid(linspace(-1,2,100),linspace(-2,1,100));
38 plot_est=fun_est(U1,U2,A,C1,C2,S1,S2);
39 surf(U1,U2,plot_est);
40 title('Plot of estimated model');
41 xlabel('u1');
42 ylabel('u2');
43 zlabel('fmodel')
44 figure;
45 Z=sin(U1+U2).*sin(U2.^2);
46 surf(U1,U2,Z);

```

```

47 title('Plot of system input-output measurements');
48 xlabel('u1');
49 ylabel('u2');
50 zlabel('f(u1,u2)')
51
52 figure;
53 plot(minMSE);
54 title('Plot of Sum of Square Errors');
55 xlabel('Generations');
56 ylabel('Sum of Square Errors');
57
58
59 norm_position=find((A(:)<10)&(A(:)>-10));
60 figure;
61 A_est=A(norm_position);
62 C1_est=C1(norm_position);
63 C2_est=C2(norm_position);
64 S1_est=S1(norm_position);
65 S2_est=S2(norm_position);
66 plot_est=fun_est(U1,U2,A_est,C1_est,C2_est,S1_est,S2_est);
67 surf(U1,U2,plot_est);
68 title('Plot of estimated model (Minimal Implementation)');
69 xlabel('u1');
70 ylabel('u2');
71 zlabel('fmodel(u1,u2)')
72
73 sum_square_error=sum(sum((plot_est-sin(U1+U2).*sin(U2.^2)).^2));
74 mean_square_error=sum_square_error*0.03^2;

```

Παρατηρούμε στις γραμμές 7-13 παράγεται ο αρχικός πληθυσμός όπου όλες οι μέσες τιμές επιλέγονται να ξεκινάνε από το σημείο (0,0), ενώ οι παράμετροι s_1, s_2 ξεκινάνε στο (0.5,0.5) και τυχαία επιλέγεται στο διαστήμα $[0,1]$ η α παράμετρος. Στην συνέχεια έχουμε την for loop των γραμμών 19-27 για τις διαδοχικές διεργασίες που επαναλαμβάνονται σε κάθε γενιά. Στη συνέχεια κρατάμε το βέλτιστο χρωμόσωμα της τελευταίας γενιάς (αυτό με την μεγαλύτερη τιμή της fitness function) και κάνουμε το plot της στο χώρο που μας ενδιαφέρει καθώς και το plot της μαθηματικής έκφρασης εισόδου-εξόδου του συστήματος που μας είχε δωθεί αλλά και ένα plot της μεταβολής της τιμής του μικρότερου error όπως θα το ορίσουμε σε κάθε γενιά.

Η fitness function που χρησιμοποιούμε είναι η evaluate_fitness(Population). Αυτή σε κάθε γενιά καλείται με όρισμα το τρέχων πληθυσμός και φτιάχνει με τρόπο που θα περιγραφεί στην συνέχεια μια μετρική για το σφάλμα του μοντέλου-χρωμοσώματος σε σχέση με το πραγματικό σύστημα και στη συνέχεια το μικρότερο σφάλμα θα το αντιστοιχεί σε μεγαλύτερη fitness τιμή.

Στη συνέχεια, πηγαίνουν οι fitness τιμές και δίνονται στη συνάρτηση Selection. Αυτή, βάσει των τιμών της fitness function με στοχαστικό τρόπο κατασκευάζει πιθανά ζευγάρια για διασταύρωση με τρόπο που θα αναλυθεί. Με αυτή την διαδικασία επίσης προχωράει στην επόμενη γενιά και άτομα του προηγούμενου πληθυσμού χωρίς να διασταυρωθούν απαραίτητα.

Το αποτέλεσμα της Selection προχωράει στη συνάρτηση Crossover ώστε να πραγματοποιηθεί η διαδικασία της διασταύρωσης σε κάθε ζευγάρι με κάποια πιθανότητα.

Η Crossover τροποποιεί τον αρχικό πληθυσμό και φτιάχνει έναν καινούργιο επιζώντα και διασταυρωμένο ενιαίο πληθυσμό ο οποίος στη συνέχεια μπαίνει στην συνάρτηση μετάλλαξης mutate_Pop(crossed_Pop) με σκοπό την πιθανή άλλα μη σιγουρή μετάλλαξη του κάθε χρωμοσώματος. Αυτή, θα επιστρέψει και τον τελικό πληθυσμό της γενιάς.

Τελικά για τον γενετικό αλγόριθμο που υλοποιήθηκε έγιναν οι εξής επιλογές:

n=100 (τα χρωμοσώματα-άτομα του πληθυσμού).

Generations=1000 (ο αριθμός των γενεών).

2 Συνάρτηση Ικανότητας (Fitness Function)

Στον κώδικα μας αυτή είναι η συνάρτηση `evaluate_fitness.m`.

```
1 function MSE = evaluate_fitness(Population)
2 MSE=[];
3 for i=1:length(Population)
4 Chromosome=Population{i};
5 A=Chromosome{1};%list of constants multiplied with Gaussian
6 C1=Chromosome{2};
7 C2=Chromosome{3};
8 S1=Chromosome{4};
9 S2=Chromosome{5};
10 mse=0;
11 for u1=linspace(-1,2,15)
12     for u2=linspace(-2,1,15)
13         mse=mse+(sin(u1+u2)*sin(u2^2)-fun_est(u1,u2,A,C1,C2,S1,S2))^2;
14     end
15 end
16 mse=(1000/(mse^1.5+0.0001))^1;
17 MSE=[MSE mse];
18 end
19 end
```

Έχουμε μια for loop στην οποία κάθε χρωμόσωμα έχουμε για συγκεκριμένες τιμές εισόδων $u_1 \in [-1, 2]$ και $u_2 \in [-2, 1]$ τον υπολογισμό του του αθροίσματος των τετραγωνικών σφαλμάτων των τιμών που παίρνει το χρωμόσωμα-μοντέλο σε σχέση με τις τιμές εξόδου που δίνει το πραγματικό σύστημα. Αυτά τα τετραγωνικά σφάλματα αθροίζονται συνολικά στην μεταβλητή `mse` και υπολογίζονται σε $15 \times 15 = 225$ σημεία. Στο τέλος του υπολογισμού για κάθε χρωμόσωμα, η `mse` χρησιμοποιείται για την άντληση τιμής ικανότητας μέσω της σχέσης:

$$fit_{value} = \frac{1000}{mse^{1.5} + 0.001}$$

Παρατηρούμε ότι με αυτόν τον τρόπο μικρή τιμή σφάλματος αντιστοιχεί σε μεγαλύτερη τιμή ικανότητας και μάλιστα η συγκεκριμένη μορφή επιλέχτηκε μετά από πολλές δοκιμές και βασίστηκε στην λογική της εκμετάλλευσης της πιο απότομης κλίσης στις μικρότερες τιμές αθροίσματος σφάλματος. Η απότομη κλίση μας συμφέρει καθώς αρκετά κοντινά σφάλματα διαφοροποιούνται μεταξύ τους σημαντικά ως προς την τιμή ικανότητας. Αυτό βέβαια ελλοχεύει κινδύνους γρήγορης σύγκλισης και εγκλωβισμού σε τοπικό ελάχιστο αλλά η συγκεκριμένη επιλογή φάνηκε ικανή για το πρόβλημα ειδικά μαζί με την πιθανότητα για μετάλλαξη.

Η `fun_est` συνάρτηση που χρησιμοποιούμε είναι η συνάρτηση του τρέχοντος μοντέλου-χρωμοσώματος με ορίσματα τις κατάλληλες εισόδους.

```
1 function val= fun_est(u1,u2,A,C1,C2,S1,S2)
2 val=0;
3 for i=1:length(A)
4     val=val+A(i)*Gaussian_2d(u1,u2,C1(i),C2(i),S1(i),S2(i));
5 end
6 end
7
8 function out = Gaussian_2d(u1,u2,c1,c2,s1,s2)
9 out=exp(-(((u1-c1).^2)/(2*s1.^2))+((u2-c2).^2)/(2*s2.^2)));
10 end
```

3 Συνάρτηση επιλογής (Selection)

Η διαδικασία της επιλογής είναι και αυτή σημαντική για τον αλγόριθμο.

Η συνάρτηση υλοποιείται με τον εξής κώδικα.

```
1 function sel=Selection(MSE)
2 P=MSE/sum(MSE);
3 n=length(P);
4 sel=cell(n/2);
5 for k=1:n/2
6     s1=Roulette_Wheel(P);
7     s2=Roulette_Wheel(P);
8     sel{k}=[s1 s2];
9 end
10 end
```

Σκοπός της συνάρτησης έχει την δημιουργία $\frac{n}{2} = \frac{\text{Population_size}}{2}$ ζευγαριών που είναι πιθανώς να διασταυρωθούν στην συνέχεια. Η κατασκευή αυτών των ζευγαριών γίνεται μέσω του λεγόμενου "τροχού της τύχης" (Roulette_Wheel()). Σε κάθε κλήση της συνάρτησης αυτής που γίνεται 2 φορές για την κατασκευή του κάθε ζευγαριού, η πιθανότητα επιλογής του i -στού χρωμοσώματος είναι η

$$P_i = \frac{I_i}{\sum_{j=1}^N I_j}$$

όπου I_i η τιμή ικανότητας για το κάθε χρωμόσωμα. Αυτή η διαδικασία είναι πιθανό να μας δώσει ταυτόσημα ζευγάρια άμα για παράδειγμα επιλεχθούν πολλές φορές τα ίδια 2 άτομα. Η λογική πίσω από αυτό είναι ότι "ικανά" άτομα μπορεί να διασταυρωθούν περισσότερες φορές από λιγότερο ικανά. ή πιθανώς να οδηγήσει στην επιλογή του ίδιου ατόμου με τον εαυτό του ως ζευγάρι. Ένα επίσης ενδεχόμενο είναι η επιλογή ως ζευγάρι ενός χρωμοσώματος με τον εαυτό του το οποίο ενδέχεται να συμβεί κυρίως στα πιο ικανά άτομα και μοντελοποιεί στην πράξη το ενδεχόμενο επιβίωσης του χρωμοσώματος αυτούσιου στην επόμενη γενιά αν δεν μεταλλαχθεί βεβαία. Πρέπει να σχολιαστεί ότι γιατί η επιλογή είναι στοχαστική δεν είναι σίγουρο ότι πάντα το ικανότερο άτομο θα επιλεχθεί και αυτό είναι λογικό συγκριτικά με την φύση αφού ποτέ δεν είναι εγγυημένο ότι ο πιο προσαρμοσμένος και ικανός οργανισμός θα επιβιώσει και θα αφήσει απογόνους. Η συνάρτηση του τροχού της τύχης είναι η παρακάτω:

```
1 function sel = Roulette_Wheel(Propability_fit)
2 rand_val=rand;
3 prev=0;
4 sel=0;
5 for i=1:length(Propability_fit)
6     if(rand_val<prev+Propability_fit(i))
7         sel=i;
8         break;
9     end
10    prev=prev+Propability_fit(i);
11 end
12 end
```

Η λογική της είναι ξεκάθαρη και μπορεί να αναπαρασταθεί εικονικά αντιστοιχώντας την πιθανότητα επιλογής για κάθε άτομο ως τομέα του τροχού. Μεγαλύτερη πιθανότητα αντιστοιχεί σε μεγαλύτερο τομέα. Περιστρέφοντας τον τροχό ένας νοητός δείκτης θα δείχνει στο τέλος της περιστροφής το τομέα/άτομο που θα επιλεγεί.

4 Διασταύρωση (Crossover)

Η διαδικασία της διασταύρωσης είναι απαραίτητη για την λειτουργία του γενετικού αλγορίθμου και την παραγωγή νέων λύσεων/χρωμοσωμάτων. Κάθε ζευγάρι έχει πιθανότητα 0.7 (σύμφωνα με την βιβλιογραφία) να διασταυρωθούν τα χρωμοσώματα που το αποτελούν και να δώσουν δύο απογόνους οι οποίοι επιστρέφονται στη θέση του ζευγαριού. Αν δεν επιλεγθούν για διασταύρωση το ζευγάρι επιστρέφεται αναλλοίωτο. Ο τρόπος που γίνεται όμως η διασταύρωση είναι με εφαρμογή της συνάρτησης Cross πάνω σε πίνακες των δύο χρωμοσωμάτων που αντιστοιχούν στην ίδια παράμετρο. Μέσα στη συνάρτηση αυτή επιλέγεται τυχαίος αριθμός r τέτοιος ώστε οι δύο πίνακες της ίδιας παραμέτρου να παρεμβληθούν στοιχείο ως προς στοιχείο με τον εξής τρόπο:

$$list_{child}(j) = r \cdot list_{parent_1}(j) + (1 - r) \cdot list_{parent_2}(j)$$

Γιατί το r επιλέγεται τυχαία στο εύρος $[-0.5, 1.5]$, οι τιμές που προκύπτουν από την παρεμβολή δεν παίρνουν μόνο τις ενδιάμεσες τιμές μεταξύ των στοιχείων αλλά και τιμές εξωτερικά από το εύρος που ορίζουν αυτές και συνεπώς οι πιθανές τιμές που προκύπτουν από την διασταύρωση είναι πιο διευρυμένες. Η επιλογή αυτού του εύρους λήφθηκε μετά από πολλές επιλογές με σκοπό την πιο αξιόπιστη και γρήγορη σύγκλιση του αλγορίθμου σε επιθυμητό αποτέλεσμα αν και λόγω των τιμών του ελλοχεύει κινδύνους ως προς την ευστάθεια των τιμών που προκύπτουν. Αν και προβληματική, εδώ θεωρώ ότι αξιοποιήθηκε ορθά για αυτό το πρόβλημα αυτή η επιλογή. Οι κωδικές είναι :

```
1 function crossed_Pop = Crossover(sel, Population)
2 crossed_Pop=cell(length(Population));
3 sel_count=0;
4 for i=1:2:length(Population)
5 rand_p=rand;
6 sel_count=sel_count+1;
7 s=sel{sel_count};
8 s1=s(1);
9 s2=s(2);
10 if(rand_p<0.7)
11     Chromosome_1=Population{s1};
12     Chromosome_2=Population{s2};
13
14     A_1=Chromosome_1{1}; %list of constants multiplied with Gaussian
15     C1_1=Chromosome_1{2};
16     C2_1=Chromosome_1{3};
17     S1_1=Chromosome_1{4};
18     S2_1=Chromosome_1{5};
19     A_2=Chromosome_2{1}; %list of constants multiplied with Gaussian
20     C1_2=Chromosome_2{2};
21     C2_2=Chromosome_2{3};
22     S1_2=Chromosome_2{4};
23     S2_2=Chromosome_2{5};
24
25     An1=Cross(A_1, A_2);
26     C1n1=Cross(C1_1, C1_2);
27     C2n1=Cross(C2_1, C2_2);
28     S1n1=abs(Cross(S1_1, S1_2));
29     S2n1=abs(Cross(S2_1, S2_2));
30     crossed_Pop{i}={An1, C1n1, C2n1, S1n1, S2n1};
31     An2=Cross(A_1, A_2);
32     C1n2=Cross(C1_1, C1_2);
33     C2n2=Cross(C2_1, C2_2);
34     S1n2=abs(Cross(S1_1, S1_2));
35     S2n2=abs(Cross(S2_1, S2_2));
```

```

36     crossed_Pop{i+1}={An2,C1n2,C2n2,S1n2,S2n2};
37
38 else
39     crossed_Pop{i}=Population{s1};
40     crossed_Pop{i+1}=Population{s2};
41 end
42 end
43 end

```

```

1 function list_out = Cross(list_1,list_2)
2 d=0.5;
3 for j=1:length(list_1)
4     rand_num=-d+(1+2*d)*rand;
5     list_out(j)=list_1(j)*rand_num+list_2(j)*(1-rand_num);
6 end
7 end

```

Αν τύχει το ζευγάρι να αποτελείται από το ίδιο άτομο προφανώς είτε διασταυρωθεί είτε όχι πάλι επιστρέφονται ταυτόσημα δυο αντίγραφα του αρχικού ατόμου καθώς από την μορφή της παρεμβολής έχουμε : $list_{child}(j) = r \cdot list_{parent}(j) + (1-r) \cdot list_{parent}(j) = list_{parent}(j)$. Η δυνατότητα αυτή ένα άτομο να περάσει ένα άτομο στην επόμενη γενιά χωρίς να διασταυρωθεί είναι αρκετά σημαντική για την αποφυγή εγκλωβισμού του αλγορίθμου σε τοπικά ελάχιστα που μπορεί να οδηγηθεί απότομα άμα επιλέγονται και επιβιώνουν μόνο αυτοί οι βέλτιστοι απόγονοι στην γειτονία του τοπικού ελαχίστου. Η κατασκευή των δύο παιδιών γίνεται με τον ίδιο τρόπο αλλά απλά αλλάζουμε την σειρά με την οποία δίνω τις λίστες/παραμέτρους στην συνάρτηση Cross.

5 Μετάλλαξη (Mutation)

Στο τέλος τα παραγόμενα αποτελέσματα από της επιλογής και διασταύρωσης τα περνάμε από διαδικασία στοχαστικής μετάλλαξης. Σε αυτή την διαδικασία κάθε χρωμόσωμα έχει πιθανότητα 0.05 να υποστεί μετάλλαξη. Η πιθανότητα αυτή επιλέχθηκε μετά απο αρκετές δοκιμές και είναι κοντά στις τιμές που προτείνει η βιβλιογραφία. Αν επιλεχθεί για μετάλλαξη το χρωμόσωμα μεταβάλλεται με την συνάρτηση Mutate.

```

1 function Mutated_Pop = mutate_Pop(Pop_in)
2 Mutated_Pop=cell(length(Pop_in));
3 for i=1:length(Pop_in)
4     mut_prop=rand;
5     if (mut_prop<0.05)
6         Mutated_Pop{i}=Mutate(Pop_in{i});
7     else
8         Mutated_Pop{i}=Pop_in{i};
9     end
10 end
11 end

```

```

1
2 function Mutated_Chromosome = Mutate(Chromosome)
3     A=Chromosome{1};%list of constants multiplied with Gaussian
4     C1=Chromosome{2};
5     C2=Chromosome{3};
6     S1=Chromosome{4};
7     S2=Chromosome{5};
8     pros=rand;
9     if (pros<0.5)
10         pros=1;

```

```

11     else
12         pros=-1;
13     end
14     A =A+normrnd(pros,2,[1,length(A)]);
15     C1 =C1+normrnd(pros,2,[1,length(C1)]);
16     C2 =C2+normrnd(pros,2,[1,length(C2)]);
17     S1 =abs(S1+normrnd(pros,1,[1,length(S1)]));
18     S2 =abs(S2+normrnd(pros,1,[1,length(S2)]));
19     Mutated_Chromosome={A,C1,C2,S1,S2};
20 end

```

Εκεί βλέπουμε ότι οι λίστες παραμέτρων των χρωμοσωμάτων τροποποιούνται με την πρόσθεση πίνακα ίδιου μεγέθους που αποτελείται από δείγματα κανονικής κατανομής με τις τυπικές αποκλίσεις που φαίνονται παραπάνω ενώ η μέση τιμή για τις παραμέτρους a, c_1, c_2 επιλέγεται τυχαία και ισοπίθανα $+1$ ή -1 για να έχω δυνατότητα επαρκής μεταβολής αμφίπλευρα. Κάτι τέτοιο δεν χρειάζεται για την τυπική απόκλιση για την οποία για να παραμείνει θετική χρησιμοποιούμε το απόλυτο.

Η χρήση της κανονικής κατανομής για τα δεδομένα μας είναι απαραίτητη ώστε να είναι πιο πιθανές οι προσθετικές μεταλλάξεις γύρω από το μηδέν με μεγαλύτερη πιθανότητα από ότι μια πιο μεγαλύτερη-απότομη μετάλλαξη. Άρα μικρές αλλαγές είναι πιο πιθανό να συμβεί από μεγαλύτερες.

Η σημασία της μετάλλαξης είναι σημαντική για τον γενετικό αλγόριθμο γιατί αποτελεί έναν σημαντικό παράγοντα διαφοροποίησης των δεδομένων και βοηθάει στην αποτροπή εγκλωβισμού του αλγορίθμου σε τοπικό ελάχιστο αφού επενεργεί τυχαίες αλλαγές που μπορεί να προσδώσουν σε άτομα του πληθυσμού την απαραίτητη διαφοροποίησή ώστε ο αλγόριθμος να οδηγηθεί σε ακόμα καλύτερες λύσεις.

6 Tuning

Κατά την σχεδίαση του γενετικού αλγορίθμου που θα επιλύσει το ζητούμενο πρόβλημα έρχομαστε αντιμέτωποι με αρκετές επιλογές. Οι επιλογές αυτές σχετίζονται με τις τιμές που θα διαλέξουμε για τις λειτουργικές πιθανότητες του αλγορίθμου (διασταύρωσης, μετάλλαξης) αλλά και άλλες παραμέτρους που σχετίζονται με το ποσό της μετάλλαξης ή διασταύρωσης. Για αυτό πρέπει να κάνουμε το κατάλληλο tuning στις διάφορες παραμέτρους δοκιμαστικά για να οδηγήσουμε πιο εύκολα την σύγκλιση του αλγορίθμου σε καλύτερα τοπικά ελάχιστα. Συνεπώς, στην διαδικασία αυτή ξοδεύσαμε αρκετό χρόνο μέχρι να οδηγηθούμε στις παραπάνω επιλογές. Πρέπει να τονισθεί επίσης ότι λόγω της στοχαστικής φύσης του γενετικού αλγορίθμου, αυτός δεν μπορεί να συγκρίνει πάντα στο ίδιο αποτέλεσμα. Από δοκιμή σε δοκιμή μπορεί να παίρνουμε χειρότερα ή καλύτερα αποτελέσματα. Στη τωρινή μορφή ο αλγόριθμος θα πρέπει να επιστρέφει μετά από λίγες δοκιμές ένα βέλτιστο αποτέλεσμα σαν αυτό που θα παρουσιαστεί παρακάτω. Ωστόσο, θα παρατεθούν στον φάκελο της εργασίας (συγκεκριμένα στο φάκελο Fit) και αποτελέσματα άλλων δοκιμών για να φανεί η διαδικασία επιλογών και σκέψεων μέχρι να καταλήξουμε στην τελική μορφή.

7 Βέλτιστο Αποτέλεσμα

Μετά από δοκιμές και αφού φέραμε τις παραμέτρους και πιθανότητες του αλγορίθμου στις τιμές που παρουσιάστηκαν παίρνουμε το plot (του Best στο workspace) που ακολουθεί:

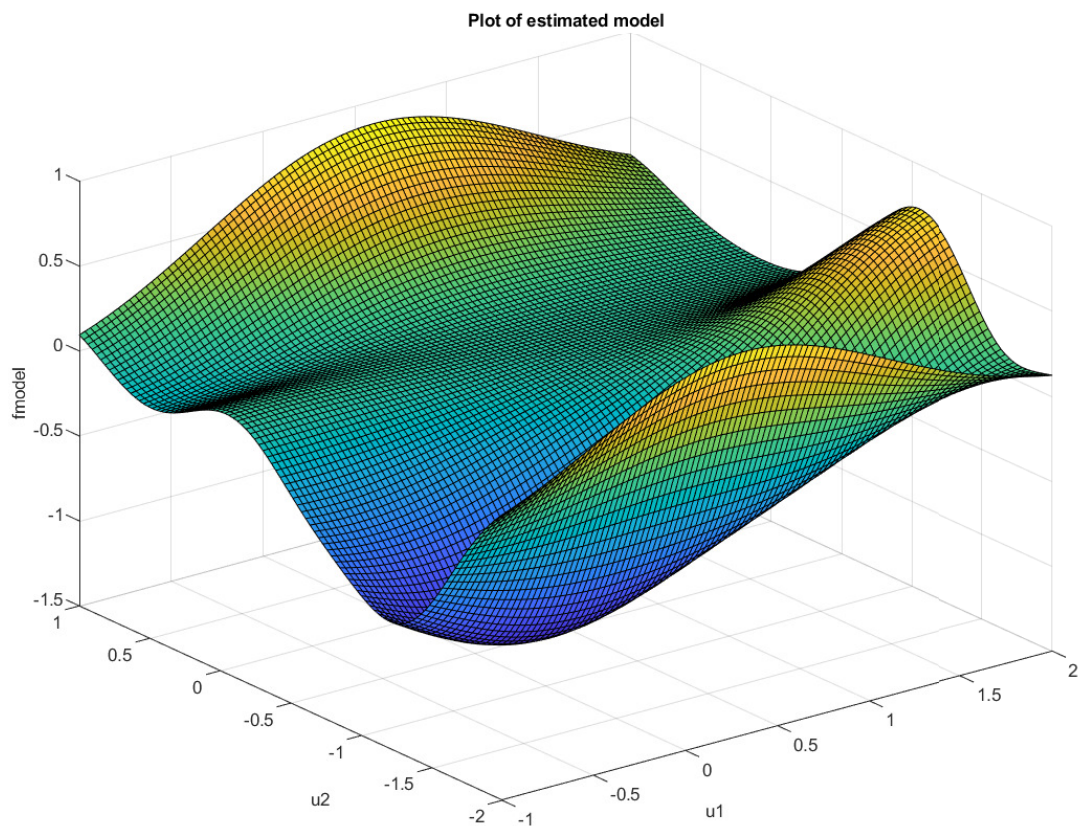
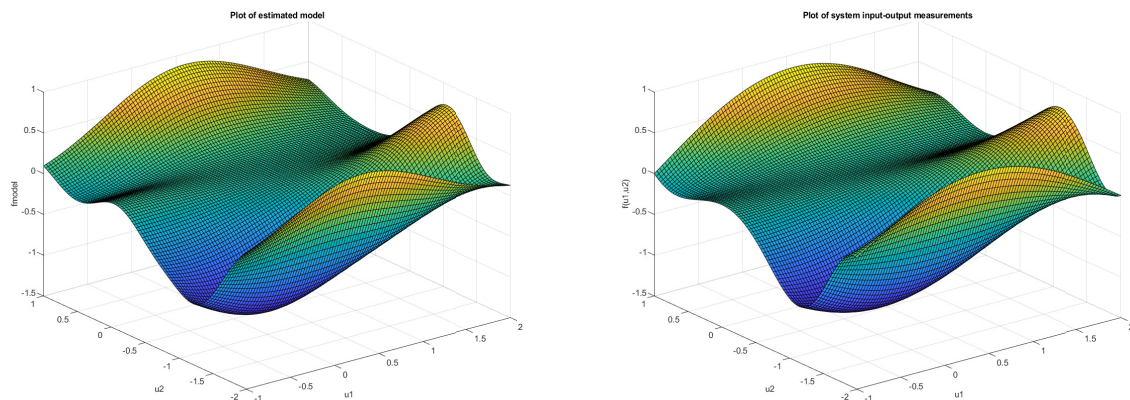


Figure 1: Βέλτιστο αποτέλεσμα

Μάλιστα, παρατίθεται και σε σύγκριση με το plot των μετρήσεων εισόδου-εξόδου του πραγματικού συστήματος. Αλλά παρατίθεται και το ελάχιστο άθροισμα τετραγωνικών σφαλμάτων του πληθυσμού σε κάθε γενιά:



((a)) Είσοδος ελέγχου για τετραγωνική είσοδο αναφοράς ((b)) Είσοδος ελέγχου για ημιτονοειδή είσοδο αναφοράς

Figure 2: Είσοδος ελέγχου για σχεδίαση με Παλινδρομητή. (Έλεγχος βασισμένος στην παθητικότητα).

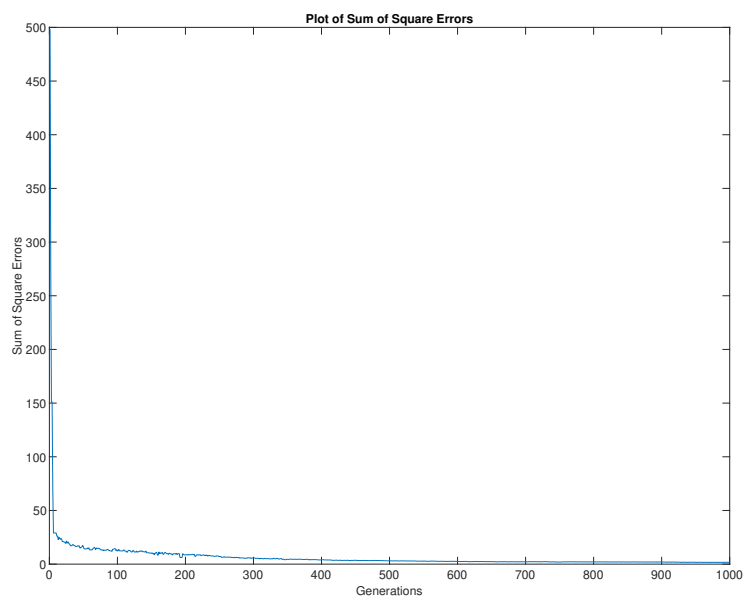


Figure 3: Άθροισμα τετραγώνου σφάλματος ανά γενιά

Παρατηρούμε ότι φθίνει αρχικά γρήγορα και μετά με την πάροδο των γενεών η βελτίωση είναι όλο και πιο μικρή.

Οι παράμετροι που προκύπτουν είναι οι εξής:

Gaussian	A	C1	C2	S1	S2
1	3.4088964	0.31483779	-3.55432	0.8382	0.95712
2	1.17855e+07	8.2766426e+02	-6.83748e+02	10.4897	0.0763
3	0.932112	0.543003	1.11346	0.95266	0.395789
4	-0.867939	-0.8645414	-0.212681	0.761197	0.66604
5	0.80804	2.2532323	-1.07028	0.7971151	0.367462
6	-5.23606e+05	8.731542e+06	2.4660216e+07	1.9344	0.1384
7	1.00115e+07	-1.29692e+06	-1.711098e+06	0.63632	5.7496
8	8.5986642e+02	-2.7918e+03	-3.44442e+03	20.962982	0.9959
9	3.023534e+07	1.1035797e+06	-1.368662e+06	5.0056	12.2663
10	-0.4069678	0.93313	-1.61914	0.6927344	0.207
11	-3.871957e+07	-1.51888e+05	2.05609e+07	4.37228	47.86177
12	-1.054949	-0.066034	-1.3209522	0.84429	0.31994
13	94.29203	-2.74335e+05	2.71536e+06	1.3012	0.87712
14	0.87770135	-0.851956	-0.1045917	0.72994	0.394363
15	-3.617132e+05	-3.806013e+05	4.261758	57.4120605	9.0635

Παρατηρούμε ότι ο κίνδυνος να ξεφύγουν οι τιμές των παραμέτρων παρά πολύ εκτός της επιθυμητής περιοχής εξαιτίας του εύρους $r \in [-0.5, 1.5]$ στη παρεμβολή της διασταύρωσης επαληθεύτηκε και στην πράξη. Ωστόσο αυτό τελικά ήταν προς όφελος μας αφού βλέπουμε ότι οι γκαουσιανές 2,6,7,8,9,11,13,15 ξεφεύγουν πολύ από την επιθυμητή περιοχή (βλέπουμε την κορυφή τους στο $(c1,c2)$) η συνεισφορά τους σε αυτή είναι ουσιαστικά ανεπαίσθητη. Αυτό μπορεί να επαληθευτεί κρατώντας τις Γκαουσιανές με κορυφές $(c1,c2)$ στην ή κοντά στην ζητούμενη περιοχή και παίρνοντας ξανά τα απαραίτητα plot. Θα κρατήσουμε δηλαδή τις:

Gaussian	A	C1	C2	S1	S2
1	3.4088964	0.31483779	-3.55432	0.8382	0.95712
2	0.932112	0.543003	1.11346	0.95266	0.395789
3	-0.867939	-0.8645414	-0.212681	0.761197	0.66604
4	0.80804	2.2532323	-1.07028	0.7971151	0.367462
5	-0.4069678	0.93313	-1.61914	0.6927344	0.207
6	-1.054949	-0.066034	-1.3209522	0.84429	0.31994
7	0.87770135	-0.851956	-0.1045917	0.72994	0.394363

Το plot που παίρνουμε με αυτή την "ελάχιστη υλοποίηση" είναι :

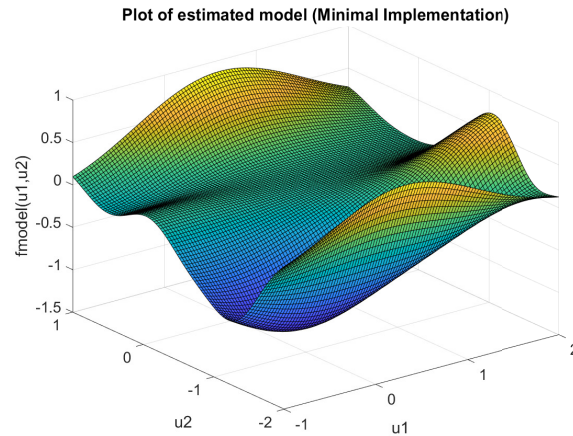


Figure 4: Ελάχιστη υλοποίηση Βέλτιστου αποτελέσματος

Παρατηρούμε ότι είναι πανομοιότυπο με το προηγούμενο plot που προέκυπτε με το μοντέλο με τις 15 γκαουσιανές. Σαφώς θα κρατήσουμε το πιο απλό μοντέλο καθώς γλιτώνουμε από πολυπλοκότητα έχοντας μόνο 7 γκαουσιανές ενώ το μοντέλο στην ενδιαφερόμενη περιοχή είναι αρκετά ακριβές. Αυτό φαίνεται άμα πάρουμε άθροισμα (`sum_square_error` στο `workspace`) σε παραπάνω σημεία του τετραγώνου του σφάλματος με τις μετρήσεις του συστήματος σε $100 \times 100 = 10000$ σημεία:

$$SE = \sum_{u_1=-1}^2 \sum_{u_2=-2}^1 (f_{model}(u_1, u_2) - \sin(u_1 + u_2) \cdot \sin(u_2^2))^2 = 11.811$$

Άμα κάνουμε και ένα scaling μαζί με τα βήματα άθροισης 0.03 παίρνουμε το "μέσο" τετραγωνικό σφάλμα:

$$MSE = 11.811 \cdot 0.03 \cdot 0.03 = 0.01063$$

Με scaling βάσει του αριθμού των δειγμάτων 10^4 έχουμε:

$$MSE = 11.811 \cdot 0.01 \cdot 0.01 = 0.00118$$

Πράγματι το "μέσο" τετραγωνικό σφάλμα είναι πολύ μικρό και για αυτό μπορούμε να πούμε ότι το μοντέλο είναι μια ικανοποιητική προσέγγιση του συστήματος. Καταφέραμε τελικά με την βοήθεια του γενετικού αλγορίθμου να βρούμε μια κατάλληλη μορφή για τις παραμέτρους του μοντέλου.

(Κύριο script κωδικα το `project_ga.m`)

(Καλύτεροι παράμετροι αποθηκευμένοι στο `BEST_DATA.m`)