ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΛΥΣΗ ΠΡΟΣΑΡΜΟΣΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

ΕΙΣΗΓΗΤΗΣ: **ΜΑΝΩΛΗΣ ΔΙΑΜΑΝΤΗΣ**

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: **ΚΟΣΜΑΤΟΠΟΥΛΟΣ ΗΛΙΑΣ**

Χανιά Σεπτέμβριος 2003

Περιεχόμενα

Εισαγωγή

Τα προβλήματα βελτιστοποίησης συναντώνται σε κάθε ανθρώπινη δραστηριότητα. Οι διάφοροι ανταγωνιστικοί μεταξύ τους περιορισμοί, δημιουργούν πολυπλοκότητα στα προβλήματα αυτά. Ο μη γραμμικός προγραμματισμός είναι ένα χρήσιμο κομμάτι της θεωρίας βελτιστοποίησης, με τη βοήθεια του οποίου, καθώς και με τη χρήση του κατάλληλου αλγορίθμου, μπορούμε να προσεγγίσουμε την ιδανικότερη λύση για κάθε πρόβλημα.

Οι κλασικές μέθοδοι βελτιστοποίησης χρησιμοποιούν αναλυτική μορφή του διανύσματος κλίσης της συνάρτησης που επιθυμούμε να ελαχιστοποιήσουμε (ή μεγιστοποιήσουμε). Σε αρκετές περιπτώσεις συναντάμε προβλήματα στα οποία η αναλυτική μορφή του διανύσματος κλίσης δεν είναι γνωστή. Σε αυτές τις περιπτώσεις χρησιμοποιούμε αλγόριθμους στοχαστικής προσέγγισης. Η διαδικασία πεπερασμένων διαφορών (FDSA) και η μέθοδος ταυτόχρονων διαταραχών (SPSA) είναι δύο γνωστοί αλγόριθμοι στοχαστικής προσέγγισης. Και οι δύο αλγόριθμοι χρησιμοποιούν αριθμητικές μεθόδους για την εκτίμηση του άγνωστου διανύσματος κλίσης. Η μέθοδος FDSA, που είναι και η πιο παλιά, υπολογίζει p ή 2p διαφορετικές τιμές της συνάρτησης σε κάθε βήμα, όπου p είναι η διάσταση του διανύσματος μεταβλητών απόφασης. Η επιλογή των p ή 2p σημείων γίνεται μετακινώντας ένα μόνο στοιχείο του διανύσματος μεταβλητών απόφασης, ενώ τα υπόλοιπα παραμένουν ως έχουν. Στη μέθοδο SPSA απαιτείται ο υπολογισμός δύο μόλις τιμών της συνάρτησης. Η επιλογή των δύο αυτών τιμών γίνεται τυχαία.

Στη παρούσα εργασία θα συγκρίνουμε τις δύο παραπάνω μεθόδους με έναν νέο αλγόριθμο, τον προσαρμοστικό αλγόριθμο βελτιστοποίησης. Σε αυτόν τον αλγόριθμο η προς βελτιστοποίηση συνάρτηση μοντελοποιείται σαν μία συνάρτηση με άγνωστες αλλά σταθερές παραμέτρους. Η βασική ιδέα του αλγορίθμου είναι σε κάθε βήμα να εκτιμά τις άγνωστες παραμέτρους και να εξάγει την εκτίμηση του διανύσματος κλίσης κάνοντας χρήση των εκτιμήσεων αυτών.

Σκοπός μας είναι να συγκρίνουμε αυτούς τους τρεις αλγορίθμους αρχικά ως προς την ακρίβεια και την ταχύτητα της σύγκλισης τους, αλλά και ως προς την ευαισθησία των παραμέτρων τους.

Για να επιτύχουμε το παραπάνω, επιλέγουμε μία σειρά προβλημάτων, ειδικά για τη σύγκριση αλγορίθμων, από τη βιβλιοθήκη CUTE. Τα προβλήματα (συναρτήσεις) που επιλέξαμε είναι τα παρακάτω:

α)Rosenbrock function

β)Freudenstein and Roth function

γ) Extended Powell singular function

δ) Penalty function

Με τις παραπάνω τέσσερις συναρτήσεις θα ασχοληθούμε πιο αναλυτικά στο κεφάλαιο 4.

Γενικά ο προσαρμοστικός αλγόριθμος όπως θα δείξουμε και παρακάτω, υπερισχύει των άλλων δύο και ως προς την ακρίβεια και την ταχύτητα της σύγκλισης, αλλά και ως προς την ευαισθησία όσο αφορά την επιλογή των παραμέτρων.

1. Αλγόριθμοι στοχαστικής προσέγγισης

Ένας αλγόριθμος στοχαστικής προσέγγισης είναι μια στοχαστική διαφορική εξίσωση με πολύ μικρό βήμα. Το βασικό πρόβλημα που πρέπει να μελετήσουμε αφορά την ποιοτική του συμπεριφορά για μεγάλο αριθμό επαναλήψεων, δηλαδή τη σύγκλιση καθώς και την τάξη σύγκλισης. Το μεγάλο πεδίο εφαρμογών των στοχαστικών αλγορίθμων δημιουργεί πλήθος τέτοιων εξισώσεων.

Οι Robbins και Monro είναι αυτοί που ανάπτυξαν τη θεωρία της στοχαστικής προσέγγισης για την μελέτη του προβλήματος της διαδοχικής εκτίμησης της θέσης της ρίζας μιας συνάρτησης, όταν η ίδια η συνάρτηση είναι άγνωστη, αλλά μπορούμε να πάρουμε τομές της για διάφορες παραμέτρους της, όπου οι διορθώσεις σε κάθε βήμα είναι μικρές [1,2] Παρατηρούμε δηλαδή την τρέχουσα εκτίμηση της ρίζας, χρησιμοποιούμε αυτή τη παρατήρηση για να κάνουμε μια μικρή διόρθωση στην εκτίμηση, παρατηρούμε στη συνέχεια τη διορθωμένη εκτίμηση της ρίζας, και η διαδικασία συνεχίζεται ομοίως μέχρι να φτάσουμε σε ικανοποιητικά αποτελέσματα. Το γεγονός ότι το μέγεθος του βήματος πρέπει να είναι μικρό, είναι σημαντικό για τη σύγκλιση επειδή μας εξασφαλίζει το ότι στο τέλος θα έχουμε πάρει μια καλή προσέγγιση της ρίζας. Η βασική ιδέα των Robbins και Monro αναλύεται στην παράγραφο 1.1 . Τα παραδείγματα εφαρμογών είναι πολλά και κυρίως αναφέρονται στην εύρεση ριζών, στην επίλυση του προβλήματος ελαχίστων τετραγώνων, στην παραμετρική ελαχιστοποίηση στοχαστικών δυναμικών συστημάτων με τη βοήθεια της μεθόδου Monte Carlo κ.α.

Οι επαναληπτικοί αλγόριθμοι (π.χ. η μέθοδος Newton-Ramson) χρησιμοποιούνται ευρέως για το διαδοχικό υπολογισμό του ελαχίστου μιας γνωστής λείας συνάρτησης. Αν η συνάρτηση είναι άγνωστη, αλλά οι με θόρυβο παρατηρούμενες τιμές της μπορούν να ληφθούν σαν παραμετρικές τιμές, τότε μπορούμε να χρησιμοποιήσουμε μια ανάλογη επαναληπτική διαδικασία, στην οποία τα διανύσματα κλίσης μπορούμε να τα υπολογίσουμε μέσω πεπερασμένων διαφορών, χρησιμοποιώντας τις μετρήσεις με θόρυβο και πάλι, με βήμα μικρού μεγέθους. Αυτή η μέθοδος λέγεται διαδικασία Kiefer-Wolfowitz και θα τη μελετήσουμε παρακάτω, στη παράγραφο 1.2.

Το πρόβλημα της πρακτικής εφαρμογής του βασικού αλγόριθμου δημιουργεί διάφορες δυσκολίες, και στη προσπάθεια επίλυσής τους οδηγούμαστε σε πολλές διαφοροποιήσεις της βασικής μορφής του αλγορίθμου. Ένα βασικό πρόβλημα για την επιτυχή εφαρμογή του, σχετίζεται με τη «ποσότητα θορύβου» στις μετρήσεις, πρόβλημα που οδηγεί στην ανάπτυξη διαφόρων μεθόδων για τη μείωση της διασποράς του θορύβου. Με τη χρήση τέτοιων μεθόδων, ο αλγόριθμος γίνεται πιο αποτελεσματικός αλλά και πιο πολύπλοκος. Θέλουμε επίσης αλγορίθμους με ευστάθεια, που να μην είναι ιδιαίτερα ευαίσθητοι σε ασυνήθιστα μεγάλες τιμές του θορύβου. Ένα ακόμα πρόβλημα παρουσιάζεται όταν ζητείται η ικανοποίηση περιορισμών.

1.1 Ο αλγόριθμος των Robbins και Monro

Ο αλγόριθμος αυτός είναι μια επαναληπτική διαδικασία για την εύρεση της ρίζας μιας πραγματικής συνάρτησης $\bar{g}$ με πραγματική μεταβλητή θ. Αν η $\bar{g}$ ήταν γνωστή και συνεχώς διαφορίσιμη, τότε το πρόβλημα θα ήταν ένα κλασικό πρόβλημα αριθμητικής ανάλυσης και θα μπορούσαμε να χρησιμοποιήσουμε τη μέθοδο Newton, η οποία δημιουργεί μια ακολουθία εκτιμήσεων του θ με τη χρήση του αναδρομικού τύπου :

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \left(\frac{\partial \bar{g}(\theta)}{\partial \theta}\right)^{-1} \bar{g}(\theta)$$

Αν η αρχική τιμή θ_0 βρίσκεται στη γειτονιά της θ^* τότε το θ_k συγκλίνει σίγουρα στο θ^*. Μια εναλλακτική και πιο απλή, αλλά λιγότερο αποτελεσματική διαδικασία είναι να χρησιμοποιήσουμε τον τύπο :

$$\theta_{k+1} = \theta_k + \varepsilon \bar{g}(\vartheta)$$

με ε>0 και επιλεγμένο σχετικά μικρό.

Ο παραπάνω τύπος δεν απαιτεί από τη συνάρτηση να είναι διαφορίσιμη και συγκλίνει σίγουρα όταν $\theta_0 - \theta^*$ είναι πολύ μικρό.

Στη περίπτωση όμως που οι τιμές της $\bar{g}$ άγνωστες, αλλά μπορούμε να πάρουμε τιμές της για συγκεκριμένα θ, οι Robbins και Monro πρότειναν τον παρακάτω αναδρομικό τύπο:

$\theta_{\kappa+1} = \theta_\kappa + \alpha_\kappa Y_\kappa$

όπου Y_κ η εκτίμηση της τιμής της $\bar{g}(\theta_\kappa)$ και α_κ μια ακολουθία που ικανοποιεί τις συνθήκες:

$\alpha_\kappa > 0$, $\alpha_\kappa \to 0$ και $\sum_k a_k = \infty$

Ένα παράδειγμα για την ακολουθία α_κ θα ήταν απλά να θέσουμε $\alpha_\kappa = 1/\kappa$ ή και $\alpha_\kappa = \alpha/(\kappa + A)^{alpha}$ με α. > 0 , A > 0 , 0 < alpha ≤ 1 . Τα βήματα της προσέγγισης συνεχώς μικραίνουν, πράγμα που υποδηλώνει ότι τα στοιχεία του θ αλλάζουν πολύ λίγο καθώς το κ τείνει στο άπειρο ($\kappa \to \infty$). Η ιδέα είναι ότι καθώς μικραίνει το βήμα παίρνουμε ένα απλό μέσο όρο των παρατηρήσεων.

1.2 Η διαδικασία των Kiefer-Wolfowitz

Το 1952 οι Kiefer και Wolfowitz, ξεκινώντας από τη μέθοδο Robbins-Monro θεώρησαν το πρόβλημα του προσδιορισμού του ελαχίστου μιας άγνωστης συνάρτησης. Έστω λοιπόν ότι θέλουμε να ελαχιστοποιήσουμε τη συνάρτηση EF(θ , χ)=f(θ), όπου $\theta \in R^p$, η f είναι συνεχώς διαφορίσιμη και χ ένα τυχαίο διάνυσμα. Οι ακριβείς μορφές των f και F δεν είναι γνωστές. Η μορφή της στοχαστικής προσέγγισης έχει ως εξής: Έστω $c_k \to 0$, και e_i το μοναδιαίο διάνυσμα με 1 στην ί-οστή συνιστώσα του. Με θ_κ συμβολίζουμε την κ-οστή εκτίμηση του ελάχιστου. Υποθέτουμε ότι για κάθε ί και k μπορούμε να εκτιμήσουμε τη παράγωγο της F, χρησιμοποιώντας πεπερασμένες διαφορές:

$$Y_{k,i} = \frac{F(\theta_k + c_k e_i) - F(\theta_k - c_k e_i)}{2c_k}$$

όπου η Y_k είναι η στοχαστική εκτίμηση του $\frac{\partial f}{\partial \theta}$. Η νέα εκτίμηση για τη θ είναι $\theta_{\kappa+1}=\theta_\kappa - \alpha_\kappa Y_\kappa$

Χρησιμοποιώντας τον παραπάνω τύπο, δηλαδή αυτόν της κεντρικής διαφοράς, σε κάθε επανάληψη χρειαζόμαστε 2p μετρήσεις της συνάρτησης. Αν χρησιμοποιήσουμε τον τύπο της πρόσθιας διαφοράς, χρειαζόμαστε p+1 μετρήσεις της συνάρτησης, και τότε ο γενικός τύπος της μεθόδου θα έχει ως εξής:

$$\theta_{kp+i+1} = \theta_{kp+i} - a_k \tilde{Y}_{kp+i}$$

$$\tilde{Y}_{kp+i} = \frac{F(\theta_{kp+1} + c_k e_{i+1}) - F(\theta_{kp+1})}{2c_k}$$

δηλαδή για κάθε κ = 0,1,…, υπολογίζουμε τα θ_{kp+i+1} για κάθε $i = 1,2,...,p.$ Αυτή τη μορφή θα χρησιμοποιήσουμε στη συνέχεια της εργασίας για τη μελέτη του αλγορίθμου.

Το κάθε βήμα της διαδικασίας των Kiefer και Wolfowitz απαιτεί 2p ή p+1 μετρήσεις της συνάρτησης, ανάλογα με το ποιόν τύπο θα χρησιμοποιήσουμε για την προσέγγιση του διανύσματος κλίσης. Στη περίπτωση που το p είναι πολύ μεγάλο, ο αλγόριθμος δεν είναι πολύ αποτελεσματικός.

Μια εναλλακτική μέθοδος είναι κάθε φορά να αλλάζουμε το διάνυσμα μόνο κατά μια κατεύθυνση με τη χρήση πεπερασμένων διαφορών, και να επιλέγουμε αυτή τη κατεύθυνση τυχαία σε κάθε βήμα. Το κάθε βήμα χρειάζεται μόνο δύο παρατηρήσεις. Αυτήν ακριβώς τη περίπτωση μελέτησε ο Spall [4] και την παρουσιάζουμε στο κεφάλαιο 2.

Οι ακολουθίες $\{\alpha_\kappa\}$ και $\{c_\kappa\}$ μπορούν να έχουν τη πολύ απλή μορφή 1/κ. αποδεικνύεται όμως στο[5] ότι εμφανίζουν καλύτερη συμπεριφορά όταν είναι της μορφής: $\alpha_k = \alpha/(A+k)^a$ και $c_k = c/k^\gamma$.

1.3 Κωδικοποίηση σε Matlab της διαδικασίας πεπερασμένων διαφορών FDSA των Kiefer-Wolfowitz

Βήμα 1. (αρχικοποίηση και επιλογή των συντελεστών). Αρχίζουμε τις επαναλήψεις θέτοντας κ=1. Ορίζουμε τους μη αρνητικούς συντελεστές α,c,A ,a,γ για τις ακολουθίες $\alpha_k = \alpha/(A+k)^a$ και $c_k = c/k^\gamma$.

Βήμα 2. (μετρήσεις της προς ελαχιστοποίηση συνάρτησης). Παίρνουμε τιμές της συνάρτησης L οι οποίες βασίζονται στη διαταραχή γύρω από το τρέχων $\hat{\theta}_k$ δηλαδή την $F\left(\hat{\theta}_k + \delta\iota\alpha\tau\alpha\rho\alpha\chi\eta\right) = F\left(\hat{\theta}_k + c_k e_k\right) = y_k^{(+)}$, όπου το c_k ορίζεται στο βήμα 1. Οι συμβολισμοί στον κώδικα είναι : $\hat{\theta}_k \to$ theta , $\hat{\theta}_k + c_k e_k \to$ thetaplus , $F\left(\hat{\theta}_k + c_k e_k\right) \to$ yplus και $F\left(\hat{\theta}_k\right) \to$ yminus.

Βήμα 3. (προσέγγιση του διανύσματος κλίσης) Δημιουργούμε το άγνωστο

διάνυσμα κλίσης $\tilde{Y}_k(\theta_k) = \begin{bmatrix} \dfrac{y_{k1}^{(+)} - y_{k1}^{(-)}}{2c_k} \\ \vdots \\ \dfrac{y_{kp}^{(+)} - y_{kp}^{(-)}}{2c_k} \end{bmatrix}$ το οποίο συμβολίζεται με gkap.

Βήμα 4. (ενημέρωση της εκτίμησης της θ). Θα χρησιμοποιήσουμε τον τύπο $\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \tilde{Y}_k\left(\hat{\theta}_k\right)$ για να πάρει το $\hat{\theta}_k$ τη νέα τιμή.

Βήμα 5. (αποδοχή ή απόρριψη της νέας τιμής του θ_k). Κρατάμε τη νέα τιμή του $\hat{\theta}_k$ αν και μόνο εάν ικανοποιούνται οι δύο παρακάτω συνθήκες.

Συνθήκη 1

Η νόρμα της νέας τιμής του θ είναι μικρότερη από τη προηγούμενη

Συνθήκη 2

Η συνάρτηση μειώνεται σε κάθε επανάληψη, τουλάχιστον κατά μία θετική ποσότητα, που τη συμβολίζουμε ΄΄**tolerance**΄΄**.** Αν θέσουμε την τιμή αυτής της ποσότητας σαν το διπλάσιο της διασποράς του θορύβου που υπάρχει στις μετρήσεις της συνάρτησης, τότε έχουμε εξασφαλίσει μια ουσιαστική βελτίωση ανά βήμα.

Βήμα 7. (επανάληψη ή τερματισμός) Είτε γυρνάμε στο βήμα 1 με κ=κ+1, είτε σταματά ο αλγόριθμος. Το δικό μας κριτήριο τερματισμού του αλγορίθμου είναι η υπέρβαση του αριθμού των επαναλήψεων που έχει οριστεί αρχικά.

Στην επόμενη σελίδα δίνεται ο αλγόριθμος FDSA σε κωδικοποίηση στη Matlab.

1.4 Κώδικας σε Matlab του αλγορίθμου FDSA

```
clear all

p= #;
sigma=0.01;
n=1000;
b=eye(p);
a=0.02;
c=0.02;
A=50;
alpha=0.602;
gamma=0.101;

for i=1:p
   deye=b(:,i);
end

tolerance=2*sigma;
loss='function';
thetafmiden=(#,#)
thetafd=thetafmiden;
losspalia=feval(loss,thetafd);
thetafdplus=zeros(p,1);
thetafdminus=zeros(p,1);

for k=1:n
   a_k=a/(k+A)^alpha;
   c_k=c/(k^gamma);
   yminus(i,1)=feval(loss,thetafd);

   for i=1:p
      thetafdplus(i,1)=thetafd(i,1)+c_k*deye(i,1);
      yplus(i,1)=feval(loss,thetafdplus);
      gkap(i,1)=yplus(i,1)-yminus(i,1);
   end

   gkap=gkap/(2*c_k);
   thetafdtest=thetafd;
   thetafd=thetafd-a_k*gkap;
   lossnea=feval(loss,thetafd);
   losspalia=feval(loss,thetafdtest);

   if lossnea > losspalia-tolerance;
      thetafd=thetafdtest;
   else
      losspalia1=losspalia;
      losspalia=lossnea;
   end

   if norm(thetafd) > norm(thetafdtest)
      thetafd=thetafdtest;
      losspalia=losspalia1;
   end

   LLFD(k)=feval(loss,thetafd);
end
```

2. Πολύμεταβλητή στοχαστική προσέγγιση με τη μέθοδο ταυτόχρονων διαταραχών. (SPSA)

Η μέθοδος του Spall είναι ένας αλγόριθμος στοχαστικής προσέγγισης που βασίζεται στη ταυτόχρονη διαταραχή για την εκτίμηση του διανύσματος κλίσης, αντί της προσέγγισης μέσω πεπερασμένων διαφορών που αναλύσαμε στο προηγούμενο κεφάλαιο.

Έστω ότι έχουμε το πρόβλημα ελαχιστοποίησης της συνάρτησης L(θ), δηλαδή της εύρεσης της ρίζας θ^* της διαφορικής εξίσωσης:

$$g(\theta) \equiv \frac{\partial L(\theta)}{\partial \theta} = 0$$

για κάποια διαφορίσιμη συνάρτηση $L: R^p \rightarrow R^1$. Όταν τα L και g παρατηρούνται άμεσα υπάρχουν πολλές μέθοδοι για την εύρεση του θ^* και μάλιστα πολύ απλές όπως αυτή της μέγιστης κατάβασης ή η μέθοδος Newton-Ramson. Στη περίπτωση που το L παρατηρείται με παρουσία θορύβου, κατάλληλος είναι ένας στοχαστικός αλγόριθμος του τύπου FDSA.

Ενώ οι στοχαστικοί αλγόριθμοι πεπερασμένων διαφορών χρειάζονται όπως είδαμε προηγούμενα, 2p ή p+1 μετρήσεις της L, ο αλγόριθμος ταυτόχρονης διαταραχής απαιτεί μόνο 2q, ($q \geq 1$) τιμές της L. Για μεγάλο p τυπικά έχουμε $q \ll p$ για τον ίδιο αριθμό επαναλήψεων. Αυξάνεται έτσι η αποτελεσματικότητα και μειώνεται το υπολογιστικό κόστος.

2.1 Ο αλγόριθμος SPSA και η προσέγγιση του διανύσματος κλίσης

Έστω $\hat{\theta}_k$ η εκτίμηση του θ στο κ-οστό βήμα. Τότε ο αλγόριθμος SPSA έχει την τυπική σε αλγόριθμους στοχαστικής προσέγγισης μορφή:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}_k\left(\hat{\theta}_k\right) .$$

εξίσωση 1

<u>Η μέθοδος της ταυτόχρονης διαταραχής</u>

Έστω $\Delta_\kappa \in R^p$, ένα διάνυσμα από p αμοιβαίως ανεξάρτητες τυχαίες μεταβλητές με μέση τιμή μηδέν. Έστω επίσης μια $\{\Delta_\kappa\}$ μια αμοιβαίως ανεξάρτητη ακολουθία, με τα Δ_κ ανεξάρτητα από τα $\hat{\theta}_0, \hat{\theta}_1, \cdots, \hat{\theta}_k$. Από μελέτες που έχουν γίνει από τον Spall, προκύπτει ότι τα Δ_κ είναι προτιμότερο να ακολουθούν την συμμετρική κατανομή Bernoulli, αν και δεν υπάρχουν περιορισμοί στην επιλογή κατανομής. Ας υποθέσουμε ακόμα ότι έχουμε στη διάθεση μας τιμές της L με θόρυβο. Έστω $y_k = L\left(\hat{\theta}_k \pm c_k \Delta_k\right) + e_k$ όπου τα e_k αντιπροσωπεύουν τους όρους θορύβου και $\{c_k\}$ μια θετική ακολουθία. Ένας τρόπος εκτίμησης της g στο κ-οστό βήμα είναι:

$$\hat{g}_k\left(\hat{\theta}_k\right) = \begin{bmatrix} \dfrac{y_k^{(+)} - y_k^{(-)}}{2c_k \Delta_{k1}} \\ \vdots \\ \dfrac{y_k^{(+)} - y_k^{(-)}}{2c_k \Delta_{kp}} \end{bmatrix}$$

εξίσωση 2

Μία ενδιαφέρον παρατήρηση είναι ότι σε αυτή την εκτίμηση χρησιμοποιούνται μόνο δύο μετρήσεις της y, σε αντίθεση με τη προσέγγιση μέσω πεπερασμένων διαφορών που χρειάζονται 2p ή p+1 μετρήσεις ανά βήμα. Ο όρος ταυτόχρονη διαταραχή προκύπτει από το γεγονός ότι όλα τα στοιχεία του διανύσματος $\hat{\theta}_k$ αλλάζουν ταυτόχρονα. Εκτός από τον υπολογισμό της (1) από τη (2), μπορούμε επίσης να την υπολογίσουμε χρησιμοποιώντας διαφορετικές και ανεξάρτητες προσεγγίσεις με τη μέθοδο των ταυτόχρονων διαταραχών για τις οποίες παίρνουμε το μέσο όρο τους σε κάθε εκτίμηση. Σε αυτή τη περίπτωση το $\hat{g}_k\left(\hat{\theta}_k\right)$ της σχέσης 2 αντικαθίσταται από το

$$\hat{g}_k\left(\hat{\theta}_k\right) = q^{-1}\sum_{j=1}^{q}\hat{g}_k^{(j)}\left(\hat{\theta}_k\right)$$

εξίσωση 3

όπου το $\hat{g}_k^{(j)}$ δημιουργείται βάση της (2), και βασίζεται σε ένα νέο ζεύγος μετρήσεων που είναι ανεξάρτητο από τα άλλα ζεύγη τιμών και εξαρτάται μόνο από το $\Im_k$. Έτσι επιτυγχάνεται βελτίωση της απόδοσης της μεθόδου.

2.2 Κωδικοποίηση σε Matlab της στοχαστικής προσέγγισης με τη μέθοδο ταυτόχρονων διαταραχών. (SPSA)

Το δύσκολο σε αυτόν τον αλγόριθμο είναι η επιλογή των μεταβλητών, και πιο συγκεκριμένα, των μεταβλητών που σχετίζονται με τις ακολουθίες $\{\alpha_k\}$ και $\{c_k\}$, δηλαδή των α, A,alpha (a) για την πρώτη και c , gamma(γ) για τη δεύτερη.

Βήμα 1. (αρχικοποίηση και επιλογή των συντελεστών) Αρχίζουμε τις επαναλήψεις θέτοντας κ=1. ορίζουμε τους μη αρνητικούς συντελεστές α,c,A,a και γ, για τις ακολουθίες $\alpha_k = \alpha/(A+k)^a$ και $c_k = c/k^\gamma$.

Βήμα 2. («γεννάμε» το διάνυσμα ταυτόχρονης διαταραχής Δ_κ) Με τη βοήθεια της γεννήτρια τυχαίων αριθμών Monte Carlo δημιουργούμε το τυχαίο διάνυσμα διάστασης p, όπου τα p στοιχεία του είναι αμοιβαίως ανεξάρτητα και το κάθε ένα από αυτά ακολουθεί κατανομή με μέση τιμή μηδέν. Μια απλή και θεωρητικά έγκυρη κατανομή είναι η Bernoulli ±1 με πιθανότητα ½ για κάθε ένα από τα ±1 εξαγόμενα. Το διάνυσμα αυτό συμβολίζεται στον κώδικα ως «delta».

Βήμα 3. (μετρήσεις της προς ελαχιστοποίηση συνάρτησης). Παίρνουμε δύο τιμές της συνάρτησης L, οι οποίες βασίζονται στην ταυτόχρονη διαταραχή γύρω από το τρέχων $\hat{\theta}_k$, δηλαδή την $y\left(\hat{\theta}_k + \delta\iota\alpha\tau\alpha\rho\alpha\chi\eta\right) = y\left(\hat{\theta}_k + c_k\Delta_k\right) = y_k^{(+)}$ και την $y\left(\hat{\theta}_k - \delta\iota\alpha\tau\alpha\rho\alpha\chi\eta\right) = y\left(\hat{\theta}_k - c_k\Delta_k\right) = y_k^{(-)}$, όπου τα c_k και Δ_k τα βρήκαμε στα βήματα 1 και 2. Οι συμβολισμοί στον κώδικα είναι : $\hat{\theta}_k \rightarrow$ theta, $\hat{\theta}_k + c_k\Delta_k \rightarrow$ thetaplus, $\hat{\theta}_k - c_k\Delta_k \rightarrow$ thetaminus $y\left(\hat{\theta}_k + c_k\Delta_k\right) \rightarrow$ yplus και $y\left(\hat{\theta}_k - c_k\Delta_k\right) \rightarrow$ yminus.

Βήμα 4. (προσέγγιση του διανύσματος κλίσης) Δημιουργούμε με τη μέθοδο ταυτόχρονης διαταραχής το άγνωστο διάνυσμα κλίσης $g\left(\hat{\theta}_k\right)$ σύμφωνα με τον τύπο :

$$\hat{g}_k\left(\hat{\theta}_k\right) = \begin{bmatrix} \dfrac{y_k^{(+)} - y_k^{(-)}}{2c_k\Delta_{k1}} \\ \vdots \\ \dfrac{y_k^{(+)} - y_k^{(-)}}{2c_k\Delta_{kp}} \end{bmatrix} = \frac{y_k^{(+)} - y_k^{(-)}}{2c_k} \begin{bmatrix} (\Delta_{k1})^{-1} \\ (\Delta_{k2})^{-1} \\ \vdots \\ (\Delta_{kp})^{-1} \end{bmatrix}$$

όπου το Δ_{ki} είναι το i-οστό στοιχείο του διανύσματος Δ_k. Ο κοινός αριθμητής και στα p στοιχεία του $\hat{g}\left(\hat{\theta}_k\right)$ κάνει φανερή την ταυτόχρονη διαταραχή όλων των στοιχείων του διανύσματος $\hat{\theta}$ σε αντίθεση με την στοιχείο προς στοιχείο αλλαγή στην προσέγγιση πεπερασμένων διαφορών. Το $\hat{g}\left(\hat{\theta}_k\right)$ συμβολίζεται στον κώδικα με gkap.

Βήμα 5. (ενημέρωση της εκτίμησης της θ) Θα χρησιμοποιήσουμε τον τύπο $\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \tilde{g}_k\left(\hat{\theta}_k\right)$ για να πάρει το $\hat{\theta}_k$ τη νέα τιμή.

Βήμα 6. (αποδοχή ή απόρριψη της νέας τιμής του θ_k). Κρατάμε τη νέα τιμή του $\hat{\theta}_k$ αν και μόνο εάν ικανοποιούνται οι δύο παρακάτω συνθήκες.

Συνθήκη 1
Η νόρμα της νέας τιμής του θ είναι μικρότερη από τη προηγούμενη

Συνθήκη 2
Η συνάρτηση μειώνεται σε κάθε επανάληψη, τουλάχιστον κατά μία θετική ποσότητα, που τη συμβολίζουμε **΄΄tolerance΄΄.** Αν θέσουμε την τιμή αυτής της ποσότητας σαν το διπλάσιο της διασποράς του θορύβου που υπάρχει στις μετρήσεις της συνάρτησης, τότε έχουμε εξασφαλίσει μια ουσιαστική βελτίωση ανά βήμα.

Βήμα 7. (επανάληψη ή τερματισμός) Είτε γυρνάμε στο βήμα 1 με κ=κ+1, είτε σταματά ο αλγόριθμος. Το δικό μας κριτήριο τερματισμού του αλγορίθμου είναι η υπέρβαση του αριθμού των επαναλήψεων που έχει οριστεί αρχικά.

Παρακάτω παρατίθενται ο πίνακας 1 με τους ακριβείς συμβολισμούς των μεταβλητών στον κώδικα καθώς και ο αλγόριθμος SPSA σε κωδικοποίηση στη Matlab.

Συμβολισμοί	
Μαθηματική μορφή	Μεταβλητή στον κώδικα
α_κ	a_k
Δ_κ	delta
c_k	c_k
$\hat{\theta}_k$	theta
$\hat{\theta}_k + c_k\Delta_k$	thetaplus
$\hat{\theta}_k - c_k\Delta_k$	thetaminus
$y\left(\hat{\theta}_k + c_k\Delta_k\right)$	yplus
$y\left(\hat{\theta}_k - c_k\Delta_k\right)$	yminus
$\hat{g}\left(\hat{\theta}_k\right)$	gkap

Πίνακας 1

2.3 Κώδικας σε Matlab του αλγορίθμου SPSA

```
clear all
p=#;
sigma=0.01;
a=0.06;
A=50;
c=0.04;
alpha=0.602;
gamma=0.101;
n=1000;
gH_avg=1;
thetamiden=(#,#);
theta=thetamiden;
tolerance=2*sigma;

loss=('function');
losspalia=feval(loss,theta);

for k=1:n
   a_k=a/(k+A)^alpha;
   c_k=c/(k^gamma);
   gkapeisodou=zeros(1,p);
   for m=1:gH_avg
      delta=2*round(rand(p,1))-1;
      thetaplus=theta+c_k*delta;
      thetaminus=theta-c_k*delta;
      yplus=feval(loss,thetaplus);
      yminus=feval(loss,thetaminus);
      gkap=((yplus-yminus)/(2*c_k))*delta;
      gkapeisodou=((m-1)/m)*gkapeisodou+gkap'/m;
   end

   thetatest=theta;
   theta=theta-a_k*gkapeisodou';
   lossnea=feval(loss,theta);
   losspalia=feval(loss,thetatest);

   if lossnea > losspalia-tolerance
      theta=thetatest;
   else
      losspalia1=losspalia;
      losspalia=lossnea;
   end

   if norm(theta)>norm(thetatest)
      theta=thetatest;
      losspalia=losspalia1;
   end

   LL(k)=feval(loss,theta);

end
```

3. Προσαρμοστικός αλγόριθμος

3.1 Εκτίμηση παραμέτρων

Η εκτίμηση παραμέτρων αποτελεί έναν πολύ ενδιαφέρον τομέα της επιστήμης, διότι μια σειρά από προβλήματα κατηγοριοποίησης (classification), αναγνώρισης προτύπων (pattern recognition), πρόβλεψης (forecasting), και αυτομάτου ελέγχου (adaptive control) μπορούν να αναχθούν ουσιαστικά σε προβλήματα εκτίμησης παραμέτρων.

Θα δούμε πως είναι δυνατή η ανάπτυξη ενός εκτιμητή της άγνωστης παραμέτρου θ^* ενός συστήματος από τη στιγμή που αυτό το σύστημα μπορεί να γραφτεί στην εξής μορφή:

$$y(x) = \theta^* \phi(x) \qquad \text{εξίσωση 4}$$

όπου θ^* περιλαμβάνει όλες τις άγνωστες μεταβλητές του δυναμικού συστήματος, φ μπορεί να θεωρηθεί ότι έχει το νόημα της εισόδου του συστήματος και y αποτελεί την έξοδο.

3.2 Σύγκλιση προσαρμοστικών αλγορίθμων (μία άγνωστη παράμετρος)

Σε αυτή τη παράγραφο παρατίθεται ένα απλό παράδειγμα εκτίμησης παραμέτρων, το οποίο περιλαμβάνει όλη την απαιτούμενη θεωρία με βάση την οποία πραγματοποιείται η απόδειξη της σύγκλισης του προσαρμοστικού αλγορίθμου που χρησιμοποιείται για την εκτίμηση άγνωστων παραμέτρων. Ειδικότερα θεωρούμε το σύστημα του οποίου η δυναμική περιγράφεται από την απλή μαθηματική εξίσωση:

$$z(t) = \theta^* \phi(t) \qquad \text{εξίσωση 5}$$

όπου θ* αποτελεί την άγνωστη παράμετρο του συστήματος, $z(t)$ είναι η έξοδος του συστήματος και το μέγεθος $\phi(t)$ παίζει το ρόλο της εισόδου.

Η εξίσωση 5 αποτελεί ουσιαστικά το αποτέλεσμα της διαδικασίας παραμετροποίησης (parameterization). Μια μεγάλη γκάμα γραμμικών και μη γραμμικών εξισώσεων μπορεί να περιέλθει σε αυτή τη μορφή μετά τη διαδικασία της παραμετροποίησης. Είναι σημαντικό να τονισθεί ότι η εξίσωση 5 είναι γραμμική ως προς τις άγνωστες παραμέτρους, γεγονός που καθιστά εφικτή την απόδειξη σύγκλισης του προσαρμοστικού αλγορίθμου.

Έστω $\hat{\theta}$ η εκτίμηση της πραγματικής τιμής θ*, τότε θα ισχύει η παρακάτω εξίσωση:

$$\hat{z}(t) = \hat{\theta} \cdot \phi(t) \qquad \text{εξίσωση 6}$$

Η εξίσωση του σφάλματος μπορεί να οριστεί ως εξής:

$$\mathrm{E} = z - \hat{z} = z - \hat{\theta} \cdot \phi \qquad \text{εξίσωση 7}$$

$$E = -\bar{\theta} \cdot \phi \qquad \text{εξίσωση 8}$$

όπου $\bar{\theta} = \hat{\theta} - \theta^*$ εξίσωση 9

Ο προσαρμοστικός αλγόριθμος με τη χρήση του οποίου μπορούμε να υπολογίσουμε την εκτίμηση $\hat{\theta}$ της άγνωστης παραμέτρου θ* προκύπτει με την ελαχιστοποίηση κάποιου κριτηρίου κόστους του σφάλματος συναρτήσει του θ. Μια από τις πιο γνωστές μεθόδους βελτιστοποίησης, η οποία χρησιμοποιείται στη συγκεκριμένη περίπτωση, είναι η μέθοδος της μέγιστης κατάβασης. (Gradient Method)

Θεωρούμε ως κριτήριο κόστους την ενέργεια του σφάλματος. Άρα θα έχουμε:

$$J(\theta) = \frac{E^2}{2} = \frac{(z - \theta \cdot \phi)^2}{2}$$ εξίσωση 10

η παραπάνω συνάρτηση είναι κυρτή, γεγονός που σημαίνει ότι οποιοδήποτε τοπικό ελάχιστο αποτελεί ταυτόχρονα και ολικό ελάχιστο. Για να βρούμε το ελάχιστο της παραπάνω συνάρτησης θα πρέπει να ισχύει ότι:

$$\nabla J(\theta) = 0 \Rightarrow \nabla J(\theta) = -(z - \theta \cdot \phi) \cdot \phi = 0$$ εξίσωση 11

Σύμφωνα με τη μέθοδο της μέγιστης κατάβασης η επαναληπτική εξίσωση με βάση την οποία μπορούμε να υπολογίσουμε το θ είναι η εξής:

$$\dot{\theta} = -\gamma \cdot \nabla J(\theta) = \gamma \cdot (z - \theta \cdot \phi) \cdot \phi = \gamma \cdot E \cdot \phi$$ εξίσωση 12

$$\theta(0) = \theta_0$$

στην περίπτωσή μας που ο χρόνος είναι διακριτός θα χρησιμοποιήσουμε τη προσέγγιση: $$\dot{\theta} \approx \frac{\theta_{new} - \theta_{old}}{\Delta t}$$

Η επαναληπτική εξίσωση 12 ονομάζεται «προσαρμοστικός κανόνας» (Adaptive Law) ή εκτιμητής (Estimator). Το μέγεθος γ αποτελεί μια θετική ποσότητα (γ>0) και ονομάζεται προσαρμοστικό κέρδος (Adaptive gain).

Από τη θεωρία της εκτίμησης παραμέτρων [10] αποδεικνύεται ότι ισχύουν οι παρακάτω σχέσεις:

$$\lim_{t \to \infty} (E(t)) = 0$$

$$\lim_{t \to \infty} [\gamma \cdot E(t) \cdot \phi(t)] = 0$$

$$\lim_{t \to \infty} \left(\dot{\theta}(t) \right) = 0$$

Το συμπέρασμα από τις παραπάνω σχέσεις, είναι ότι για κάθε $\phi(t),\dot{\phi}(t)$ με πεπερασμένες τιμές, ο προσαρμοστικός κανόνας εγγυάται ότι η εκτιμώμενη έξοδος $\hat{z}(t)$ συγκλίνει στην πραγματική τιμή της εξόδου $z(t)$ ασυμπτωτικά. Ωστόσο το ερώτημα του κατά πόσο η εκτίμηση της άγνωστης παραμέτρου (θ) συγκλίνει στην πραγματική τιμή (θ^*) παραμένει ουσιαστικά αναπάντητο παρόλο που ισχύει $\underset{t\to\infty}{Lim}(E(t))=0$. Πράγματι υπάρχουν περιπτώσεις, που ενώ το σφάλμα συγκλίνει στο μηδέν καθώς ο χρόνος τείνει στο άπειρο, η εκτιμώμενη τιμή της παραμέτρου τελικά δε μπορεί να συγκλίνει προς την πραγματική της τιμή.

Από τη εξίσωση 6 φαίνεται ότι όταν $\phi=0$ τότε $z(t)=0$ για κάθε θετικό t. Αυτό σημαίνει πως όταν η είσοδος τείνει στο μηδέν δεν δίνεται καμιά χρήσιμη πληροφορία στον εκτιμητή, ώστε αυτός να εκτιμήσει την άγνωστη παράμετρο θ^*. Θα πρέπει λοιπόν να προστεθούν και άλλοι περιορισμοί, οι οποίοι θα εγγυούνται ότι το σήμα της εισόδου έχει την απαιτούμενη ενέργεια, καθώς ο χρόνος τείνει στο άπειρο, για να καταστεί εφικτή η εκτίμηση της άγνωστης παραμέτρου. Μια ικανή και αναγκαία συνθήκη για να συγκλίνει εκθετικά το $\bar{\theta}(t)$ στο μηδέν, είναι η είσοδος $\phi(t)$ να ικανοποιεί την παρακάτω σχέση:

$$\int_t^{t+T_0}\phi(t)^2\,dt \geq a_0\cdot T_0 \qquad \forall t\geq 0 \quad \text{και κάποια } \alpha_0, T_0>0$$

Η παραπάνω συνθήκη ονομάζεται συνθήκη «persistency of excitation» και σε ελεύθερη μετάφραση *"επιμονή της διέγερσης"*. Έτσι π.χ. για $\phi(t)=0$ και $\phi(t)=e^{-t}$ δεν ικανοποιείται η παραπάνω συνθήκη και άρα δεν μπορούμε να εκτιμήσουμε την άγνωστη παράμετρο, ενώ για $\phi(t)=1$ η συνθήκη ικανοποιείται. Η ιδιότητα αυτή είναι πάρα πολύ σημαντική για πάρα πολλούς προσαρμοστικούς αλγορίθμους.

3.3 Σύγκλιση προσαρμοστικών αλγορίθμων (περισσότερες από μία παράμετροι)

Η ίδια ακριβώς μεθοδολογία μπορεί να χρησιμοποιηθεί και για την εκτίμηση ενός n-διάστατου διανύσματος θ^* το οποίο ικανοποιεί την εξίσωση:

$z(t) = \theta^{*T} \cdot \phi(t)$ όπου $z \in \Re^1, \phi \in \Re^n$ είναι πεπερασμένα και είναι διαθέσιμα προς μέτρηση.

Οι εξισώσεις 6,7 και 8 μετατρέπονται σε:

$$\hat{z}(t) = \theta^T \cdot \phi(t)$$

$$\mathrm{E} = z - \hat{z} = z - \theta^T \cdot \phi$$

$$E = -\bar{\theta}^T \cdot \phi$$

όπου $\bar{\theta} = \theta - \theta^*$

Η εξίσωση 10, δηλαδή το κριτήριο κόστους, γίνεται

$$J(\theta) = \frac{E^2}{2} = \frac{\left(z - \theta^T \cdot \phi\right)^2}{2}$$

η παραπάνω συνάρτηση είναι και πάλι κυρτή. Για να βρούμε το ελάχιστο της θα πρέπει να ισχύει ότι:

$$\nabla J(\theta) = 0 \Rightarrow \nabla J(\theta) = -\left(z - \theta^T \cdot \phi\right) \cdot \phi = 0$$

Η επαναληπτική εξίσωση με την οποία θα υπολογίσουμε το θ είναι η εξής:

$$\dot{\theta} = -\gamma \cdot \nabla J(\theta) = \gamma \cdot \left(z - \theta^T \cdot \phi\right) \cdot \phi = \gamma \cdot E \cdot \phi$$

$$\theta(0) = \theta_0 \qquad \text{εξίσωση 13}$$

Και σε αυτή τη περίπτωση αποδεικνύεται ότι ισχύουν οι παρακάτω σχέσεις:

$$\lim_{t\to\infty}\left(E(t)\right)=0$$

$$\lim_{t\to\infty}[\gamma\cdot E(t)\cdot\phi(t)]=0$$

$$\lim_{t\to\infty}\left(\dot{\theta}(t)\right)=0$$

Στην περίπτωση των πολλαπλών μεταβλητών, αυτό που αλλάζει είναι η ικανή και αναγκαία συνθήκη για να συγκλίνει εκθετικά το $\bar{\theta}(t)$ στο μηδέν. Η συνθήκη «persistency of excitation» γίνεται :

$$\int_t^{t+T_0}\phi(t)\cdot\phi^T(t)dt\geq a_0\cdot T_0 \qquad \forall t\geq 0 \quad \text{και κάποια } \alpha_0, T_0>0$$

ή

$$\int_t^{t+T_0}\phi(t)^T\cdot q\cdot dt\geq a_0\cdot T_0 \qquad \forall t\geq 0\text{, κάποια } \alpha_0, T_0>0 \text{ και } \forall q\text{: } |q|=1, \quad q_i\geq 0$$

Η παραπάνω συνθήκη είναι αρκετά δύσκολο να εξηγηθεί. Με απλά λόγια για να ισχύει αυτή η συνθήκη, το διάνυσμα $\phi(t)$ θα πρέπει να έχει «αρκετή ενέργεια» προς όλες τις κατευθύνσεις και σε οποιοδήποτε χρονικό διάστημα. Συναρτήσεις για τις οποίες ισχύει η παραπάνω συνθήκη είναι π.χ. η διανυσματική συνάρτηση, το κάθε στοιχείο της οποίας, ακολουθεί την κανονική κατανομή (τυχαίος θόρυβος) και συναρτήσεις που είναι αθροίσματα ημιτονοειδών με πολλές διαφορετικές βασικές συχνότητες.

3.4 Προσαρμοστικός αλγόριθμος βελτιστοποίησης

Έστω ότι έχουμε να βελτιστοποιήσουμε μια συνάρτηση σαν αυτή που μελετήσαμε στο κεφάλαιο 3.3, δηλαδή μια συνάρτηση της μορφής:

$$F(x) = \theta^{*T} \cdot \phi(x)$$

ή πιο αναλυτικά

$$F(x) = \theta_1^* \cdot \phi_1(x) + \theta_2^* \cdot \phi_2(x) + \cdots + \theta_m^* \cdot \phi_m(x)$$

όπου θ_i^* είναι οι πραγματικές τιμές των άγνωστων παραμέτρων της συνάρτησης και x είναι το διάνυσμα μεγέθους n, των μεταβλητών απόφασης. Με m συμβολίζουμε τον αριθμό των όρων της αντικειμενικής συνάρτησης.

Σε κάθε βήμα ο αλγόριθμος μπορεί να μας δίνει μια εκτίμηση του διανύσματος των άγνωστων παραμέτρων θ^* κάνοντας χρήση της εξίσωσης 13 του κεφαλαίου 3.3, δηλαδή:

$\dot{\theta}_i = \bar{\gamma} \cdot E \cdot \phi_i$ με i=1,2,…,m

$\theta(0) = \theta_0$ (θ_0 μια τυχαία αρχική τιμή του διανύσματος θ)

όπου Ε είναι το σφάλμα (F - $\hat{F}$), $\hat{F}$ η εκτιμώμενη τιμή της συνάρτησης και F η πραγματική της τιμή.

Ή στη περίπτωσή διακριτού χρόνου:

$\theta_i^{new} = \theta_i^{old} + \gamma \cdot E \cdot \phi_i$ όπου γ = $\bar{\gamma} \cdot \Delta t$

Από την θ_0 μπορούμε να κάνουμε την αρχική μας εκτίμηση $\hat{F}$ από τη σχέση:

$$\hat{F}(x) = \theta_0^T \cdot \phi(x)$$

Η επαναληπτική σχέση για την εκτίμηση της τιμής της συνάρτησης θα είναι:

$$\hat{F}(x) = \hat{\theta}^T \cdot \phi(x).$$

Η εκτίμηση του διανύσματος κλίσης θα είναι:

$$\nabla \hat{F}(x) = \frac{\partial\left(\hat{\theta}^T \cdot \phi(x)\right)}{\partial x}$$

Βάση της εκτίμησης του διανύσματος κλίσης μεταβάλλουμε το διάνυσμα x με την επαναληπτική σχέση:

x_j new = x_j old - gamma $\dfrac{\partial\left(\hat{\theta}^T \cdot \phi(x_j)\right)}{\partial x_j}$ $\quad \forall j = 1,2,...,n$ εξίσωση 14

Ο προσαρμοστικός αλγόριθμος με αυτόν τον τρόπο θα μεταβάλλει συνεχώς το διάνυσμα x και θα μας δίνει κάθε φορά μια καλύτερη εκτίμηση της F. Σε περίπτωση που το εκτιμώμενο διάνυσμα κλίσης είναι σχεδόν μηδέν, για να αποφύγουμε την περίπτωση να τερματιστεί ο αλγόριθμος χωρίς να έχουμε φτάσει κοντά στο βέλτιστο, εισάγουμε με τυχαία διαδικασία κάποια $\phi(x)$ που να ικανοποιούν τη συνθήκη «persistency of excitation». Εισάγουμε π.χ. τυχαίο θόρυβο στα $\phi(x)$.

3.5 Ένα απλό παράδειγμα για κατανόηση

Έστω ότι έχουμε τη συνάρτηση $F(x)=(x_1+2x_2)^2$ προς ελαχιστοποίηση. Καταρχήν μοντελοποιούμε τη συνάρτηση στη μορφή $F(x)=\theta^{*T}\cdot\phi(x)$:

$F(x) = x_1^2 + 4x_1x_2 + 4x_2^2 = \theta_1x_1^2 + \theta_2x_1x_2 + \theta_3x_2^2$ με $\theta_1=1$ $\theta_2=4$ και $\theta_3=4$

Το διάνυσμα κλίσης δίνεται από τη σχέση:

$$\frac{\partial F}{\partial x}=\nabla F(x)=\begin{bmatrix}2\theta_1x_1+\theta_2\\ 2\theta_3x_2+\theta_2\end{bmatrix}$$

και αντίστοιχα η εκτίμηση του διανύσματος κλίσης θα δίνεται από

$$\frac{\partial \hat{F}}{\partial x}=\begin{bmatrix}2\hat{\theta}_1x_1+\hat{\theta}_2\\ 2\hat{\theta}_3x_2+\hat{\theta}_2\end{bmatrix}$$

Για τις πραγματικές τιμές των άγνωστων παραμέτρων θ, το διάνυσμα κλίσης μηδενίζεται για $x_1 = -2$ και $x_2 = -0.5$. Ο αλγόριθμος μας όμως χρησιμοποιεί εκτιμήσεις των άγνωστων παραμέτρων θ. Έτσι π.χ. για $\hat{\theta}_1=2$ $\hat{\theta}_2=3$ και $\hat{\theta}_3=5$ αν τα x εκτιμηθούν ως εξής: $x_1 = -0.75$ και $x_2 = -0.3$ θα έχουμε εκτιμήσει το διάνυσμα κλίσης μηδέν, έχοντας εκτιμήσει τα διανύσματα θ και x λάθος. Γι αυτό ακριβώς το λόγο επιβάλλεται, όταν μηδενίζεται το διάνυσμα κλίσης και σταματάμε να παίρνουμε καλύτερες εκτιμήσεις της συνάρτησης, να εισάγουμε $\phi(x)$ που να ικανοποιούν τη συνθήκη «persistency of excitation».

3.6 Κωδικοποίηση σε Matlab του προσαρμοστικού αλγορίθμου βελτιστοποίησης

Βήμα 1. (αρχικοποίηση και επιλογή των συντελεστών) Αρχίζουμε τις επαναλήψεις θέτοντας κ=1. ορίζουμε τους μη αρνητικούς συντελεστές γ (g1) και gamma (g2), ορίζουμε τα $\phi(x)$, παίρνουμε από γεννήτρια τυχαίων αριθμών το θ_0 και την αρχική τιμή της F.

Βήμα 2. (υπολογίζουμε τις τιμές των $\hat{F}$ και $\hat{\theta}$) Εκτιμούμε όπως δείξαμε στο 3.4 την τιμή της συνάρτησης και τις άγνωστες παραμέτρους θ.

Βήμα 3. (περιορισμοί για την εκτίμηση των άγνωστων παραμέτρων) Παρατηρείται το φαινόμενο της υπερβολικής αύξησης των τιμών των άγνωστων παραμέτρων, γι αυτό αν σε κάποια επανάληψη του αλγόριθμου παρατηρηθεί μια τέτοια αύξηση για μία άγνωστη παράμετρο, αυτή παίρνει την τιμή που είχε στην αμέσως προηγούμενη επανάληψη. Δηλαδή σε αυτό το βήμα κρατάμε τις τιμές των άγνωστων παραμέτρων μέσα σε ένα προκαθορισμένο όριο.

Βήμα 4. (εκτίμηση του διανύσματος κλίσης με βάση τα νέα $\hat{\theta}$ και υπολογισμός του νέου διανύσματος των x) Υπολογίζουμε τις νέες τιμές των μεταβλητών x με τη βοήθεια της εξίσωσης 14 του κεφαλαίου 3.4 .

Βήμα 5. (υπολογισμός της πραγματικής τιμής της F με χρήση των νέων θ και x) Υπολογίζουμε τη πραγματική τιμή της συνάρτησης F για τις νέες τιμές των x όπως αυτές υπολογίστηκαν στο βήμα 4.

Βήμα 6. (αποθήκευση του βέλτιστου). Όταν επιτύχουμε μια μικρότερη τιμή για τη συνάρτηση, (F<Fmin) «αποθηκεύουμε» τη νέα αυτή τιμή καθώς και τα αντίστοιχα x.

Βήμα 7. (εισάγουμε $\phi(x)$ που να ικανοποιούν τη συνθήκη «persistency of excitation») Είτε πηγαίνουμε στο βήμα 8, είτε σε περίπτωση που ο αλγόριθμος σταματά να μας δίνει μικρότερες τιμές της F *(F new-F old ≥ 0 που σημαίνει είτε οτι έχουμε εκτιμήσει το διάνυσμα κλίσης μηδέν, είτε ότι έχουμε φτάσει στην βέλτιστη δυνατή λύση)* εισάγουμε $\phi(x)$ με γεννήτρια τυχαίων αριθμών, έτσι ώστε να ικανοποιείται η συνθήκη «persistency of excitation». Στο τέλος της τυχαίας αυτής διαδικασίας (random διαδικασία) τα x ξαναπαίρνουν τις «αποθηκευμένες» τιμές του βήματος 6. Προχωράμε στο βήμα 8.

Βήμα 8. (επανάληψη ή τερματισμός) Είτε γυρνάμε στο βήμα 1 με κ=κ+1, είτε σταματά ο αλγόριθμος. Το κριτήριο τερματισμού του αλγορίθμου είναι η υπέρβαση του αριθμού των επαναλήψεων που έχει οριστεί αρχικά.

Παρακάτω δίνονται οι συμβολισμοί των μεταβλητών στον κώδικα καθώς και ο προσαρμοστικός αλγόριθμος βελτιστοποίησης σε κωδικοποίηση στη Matlab.

γ	→	g1	
Gamma	→	g2	
θ^*	→	theta	
$\hat{\theta}$	→	thetab	
$\phi(x)$	→	V(i)	
κατάσταση	→	sit	(1→κανονικό πρόγραμμα, 2→ random διαδικασία)
F	→	F	
$\hat{F}$	→	f	
(F-f)	→	E	

Το πλήθος των συνολικών επαναλήψεων που ο αλγόριθμος εκτελεί το βήμα 7 συμβολίζεται με r, με n η διάσταση του διανύσματος x και με m ο αριθμός των όρων της αντικειμενικής συνάρτησης.

3.7 Κώδικας σε Matlab του Προσαρμοστικού αλγόριθμου βελτιστοποίησης

```
clear all
m=#;
g1=0.001;
g2=0.001;
theta=[#;#;#,….#];
thetab=randn(m,1);
thetabarx=thetab
x1=#;
x2=#;
   ⋮
xn=#
Fmin=90000000000000;
v(1)=Φ1;
v(2)=Φ2;
   ⋮
v(m)=Φm;
r=0;
tolerence=0;
F_old=9000000000000;
sit=1;
F=0;

For i=1:m
F=theta(i)*v(i)+F;
end

for k=1:1000
   if sit==1
      f=0;
      for i=1:m
         f=f+thetab(i)*v(i);
      end

      for j=1:m
         thetab(j)=thetab(j)+g1*v(j)*(F-f);
      end

      thetab_old=thetab;

      for j=1:m
         thetab(j)=thetab(j)+g1*v(j)*(F-f)*(norm(v)+0.1)^-1;
      end
      if abs(thetab) > 150
        for j=1:m
           thetab(j)=abs(thetab_old(j))*sign(thetab(j));
        end
      end
```

```
    E(k)=F-f;
    x1=x1-g2*(∂f/∂x1);
    x2=x2-g2*(∂f/∂x2);
         ⋮
    xn=xn-g2*(∂f/∂xm)

    v(1)=Φ1;
    v(2)=Φ2;
     ⋮
    v(m)=Φm;

    F_old=F;
        F=0;
    for j=1:m,
       F=F+theta(j)*v(j);
    end

    FF(k)=F;
    FFF(k)=F;

    if F<Fmin
      Fmin=F;
      epanal=k;
      x1b=x1;
      x2b=x2;
         ⋮
      xnb=xn;
      thetabzitoymeno=thetab;
   end

   if F-F_old >= tolerence
     sit=2;
   end

else
    r=r+1;
    l(r)=k;

    for k1=1:10,
       f=0;
       for i=1:m
          f=f+thetab(i)*v(i);
       end
       for j=1:m
          thetab(j)=thetab(j)+g1*v(j)*(F-f);
       end
```

```
        thetab_old=thetab;
        for j=1:m
            thetab(j)=thetab(j)+g1*v(j)*(F-f)*(norm(v)+0.1)^-1;
        end
        if abs(thetab) > 150
          for j=1:m
              thetab(j)=abs(thetab_old(j))*sign(thetab(j));
          end
        end

        x1=randn(1);
        x2=randn(1);
         ⋮
        xn=randn(1);
       v(1)=Φ1;
       v(2)=Φ2;
         ⋮
       v(m)=Φm;

        F=0;
        for j=1:m,
            F=F+theta(j)*v(j);
        end

        FF(k)=F;
    end

    sit=1;
    x1=x1b;
    x2=x2b;
      ⋮
    xn=xnb;

    FFF(k)=FFF(k-1);
  end
end

r
Fmin
epanal
x1b
x2b
  ⋮
xnb
thetabzitoymeno
x1
x2
⋮
xn
thetab
plot(FFF)
```

4. Προβλήματα δοκιμής

Τα προβλήματα που θα χρησιμοποιήσουμε για τον έλεγχο των αλγορίθμων πάρθηκαν από τη βιβλιοθήκη CUTE, η οποία έχει αναπτυχθεί από τη συνεργασία διαφόρων πανεπιστημίων και περιλαμβάνει ένα πλήθος προβλημάτων για έλεγχο τόσο σε προβλήματα χωρίς, όσο και σε προβλήματα με περιορισμούς.

Επιλέξαμε 4 προβλήματα (συναρτήσεις) με τα εξής κριτήρια:

- Να μπορούν να γραφούν στη μορφή $y(x)=\theta^T \cdot \phi(x)$
- Να έχουμε ποικιλία στον αριθμό των μεταβλητών (x)
- Να έχουμε εν γένει «δύσκολες συναρτήσεις για βελτιστοποίηση», ακόμα και σε περιπτώσεις που η αναλυτική μορφή της συνάρτησης είναι γνωστή.

Αυτά τα 4 προβλήματα δίνονται παρακάτω, ενώ η μορφοποίηση για την περιγραφή του καθενός από αυτά είναι:

(a) .n ο αριθμός μεταβλητών του προβλήματος και m ο αριθμός των όρων της αντικειμενικής συνάρτησης.

(b) Οι όροι f_i της αντικειμενικής συνάρτησης, ώστε αυτή να δίνεται από το άθροισμα $f_{(x)} = \sum_{i=1}^{m} f_{i(x)}^2$

(c) το αρχικό σημείο εκκίνησης x_0

(d) το ή τα ελάχιστα καθώς και τα σημεία εμφάνισής τους όπου αυτά είναι γνωστά.

Αναλυτικά τα προβλήματα αυτά είναι:

1) Rosenbrock function

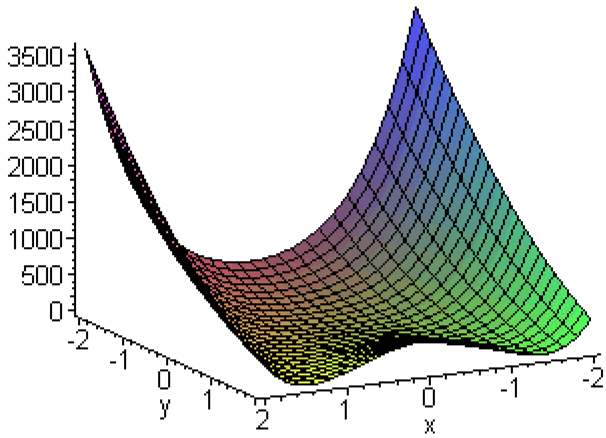
(a) $n = 2, m = 2$

(b) $f_1(x) = 10\ (x_2 - x_1^2)$

$f_2(x) = 1 - x_1$

(c) $x_0 = (-1.2\ ,\ 1)$

(d) $f = 0$ στο $(1\ ,\ 1)$



(το γράφημα της Rosenbrock function) εικόνα 1

2) Freudenstein and Roth function

(a) $n = 2, m = 2$

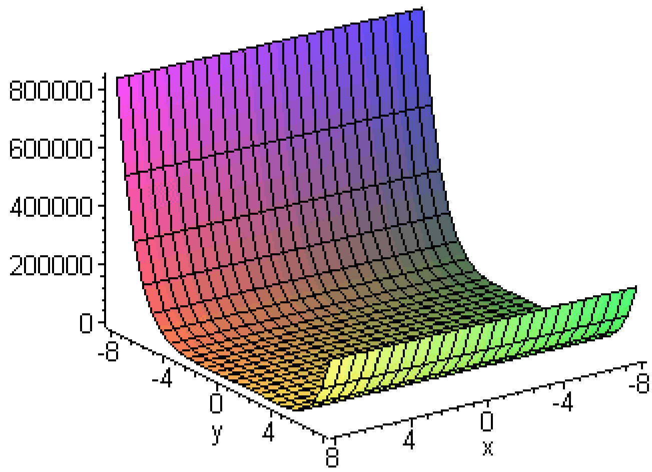
(b) $f_1(x) = -13 + x_1 + [\ (5 - x_2)\ x_2\ - 2\ ]\ x_2$

$f_2(x) = -29 + x_1 + [\ (x_2 + 1)\ x_2\ - 14\ ]\ x_2$

(c) $x_0 = (0.5\ ,\ -2)$

(d) $f = 0$ στο $(5\ ,\ 4)$

$f = 48.9842\ldots$ στο $(11.41\ldots\ ,\ -0.8968)$



(το γράφημα της Freudenstein and Roth function) εικόνα 2

Από το γράφημα της συνάρτησης Freudenstein and Roth καταλαβαίνουμε ότι θα έχουμε πρόβλημα στην εκτίμηση του διανύσματος κλίσης. Αυτός ο κίνδυνος είναι ορατός λόγω του σχεδόν επίπεδου γραφήματος της συνάρτησης.

3) Extended Powell singular function

(a) n ακέραιος αλλά πολλαπλάσιο του 4, m = n

(b) $f_{4i-3}(x) = x_{4i-3} - 10\, x_{4i-2}$

$f_{4i-2}(x) = 5^{\frac{1}{2}}\,(x_{4i-1} - x_{4i})$

$f_{4i-1}(x) = (x_{4i-2} - 2\, x_{4i-1})^2$

$f_{4i}(x) = 10^{\frac{1}{2}}\,(x_{4i-3} - x_{4i})^2$

(c) $x_0 = (\xi_j)$

όπου $\xi_{4j-3} = 3$, $\xi_{4j-2} = -1$, $\xi_{4j-1} = 0$ και $\xi_{4j} = 1$

(d) f = 0 στο (0 , 0)

4) Penalty function

(a) n ακέραιος , m = n + 1

(b) $f_i(x) = \alpha^{\frac{1}{2}} (x_1 - 1)$, $1 \leq í \leq n$

$f_{n+1}(x) = (\sum_{j=1}^{n} x_j^2) - \frac{1}{4}$

όπου $\alpha = 10^{-5}$

(c) $x_0 = (\xi_j)$

όπου $\xi_j = j$

(d) $f = 2.24997\ldots * 10^{-5}$ για n = 4

$f = 7.08765\ldots * 10^{-5}$ για n = 10

Για κάθε μία από τις παραπάνω συναρτήσεις (προβλήματα) θα προσπαθήσουμε να αναλύσουμε τη συμπεριφορά του κάθε αλγορίθμου.

Οι δύο εικόνες που αναπαριστούν τα γραφήματα των δύο πρώτων συναρτήσεων βρέθηκαν στο internet, στη διεύθυνση:

www.cs.duke.edu/~rodger/curious/pages/dolinsky/opt.html

5. Εκτίμηση των βέλτιστων τιμών των παραμέτρων

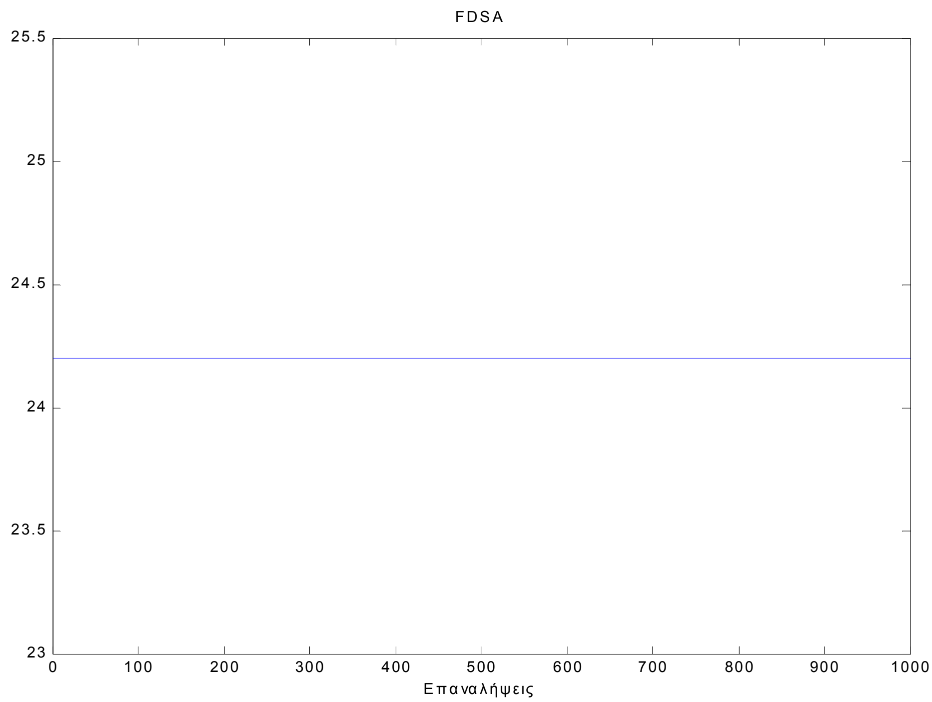
Για να προχωρήσουμε αργότερα στη σύγκριση των αλγορίθμων, πρέπει πρώτα να είμαστε σίγουροι ότι για κάθε αλγόριθμο και κάθε συνάρτηση, έχουμε βρει εκείνες τις τιμές των παραμέτρων για τις οποίες οι αλγόριθμοι έχουν τη βέλτιστη συμπεριφορά. Αλλιώς η οποιαδήποτε σύγκριση θα είναι άδικη.

5.1 Συνάρτηση 1 (Rosenbrock function)

Υπενθυμίζουμε ότι η πρώτη συνάρτηση παρουσιάζει ελάχιστο (fmin=0) στο (x1 , x2) = (1 , 1). Η αρχική τιμή των μεταβλητών θα είναι $x_0 = (-1.2 , 1)$.

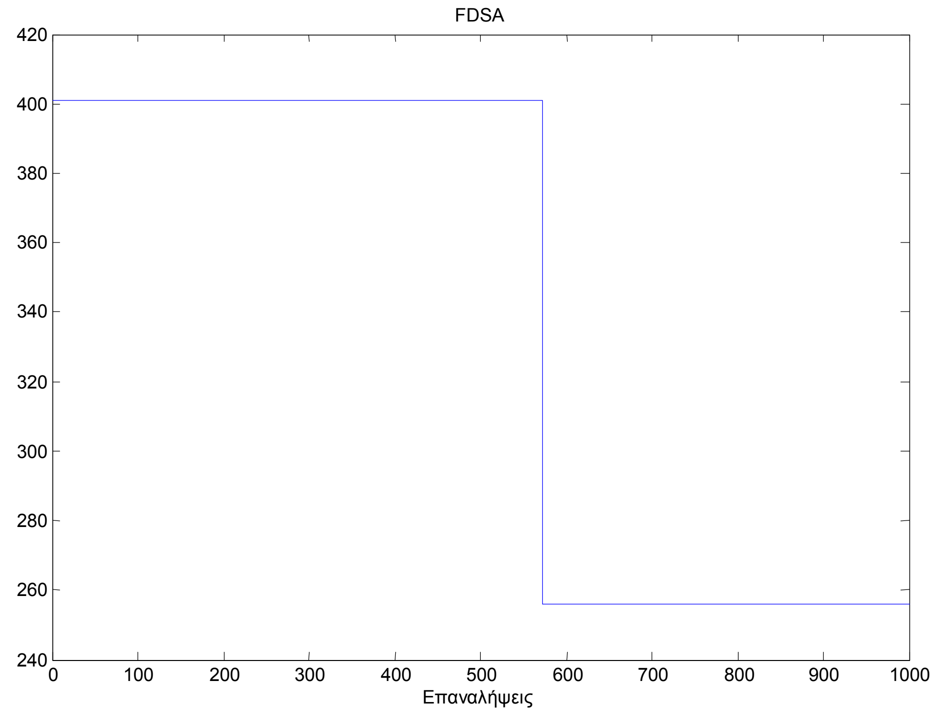
5.1.1 Μέθοδος FDSA

για τη συνάρτηση 1 η μέθοδος FDSA δεν μας δίνει επιθυμητά αποτελέσματα. Για την ακρίβεια δεν καταφέρνει να εκτιμήσει ούτε μία φορά καλύτερη τιμή για την f(θ). Αλλάξαμε τιμές στις παραμέτρους με όλους τους δυνατούς συνδυασμούς, μειώσαμε την tolerance της συνθήκης 2 στο ελάχιστο (tolerance=0), αφαιρέσαμε και την πρώτη συνθήκη, αλλά παρόλα αυτά ο αλγόριθμος δεν προσέγγισε στο ελάχιστο τη βέλτιστη τιμή της συνάρτησης. Παρέμεινε στο 24.2 που είναι η τιμή της συνάρτησης για τη συγκεκριμένη αρχική τιμή. Βλέπε γράφημα 1



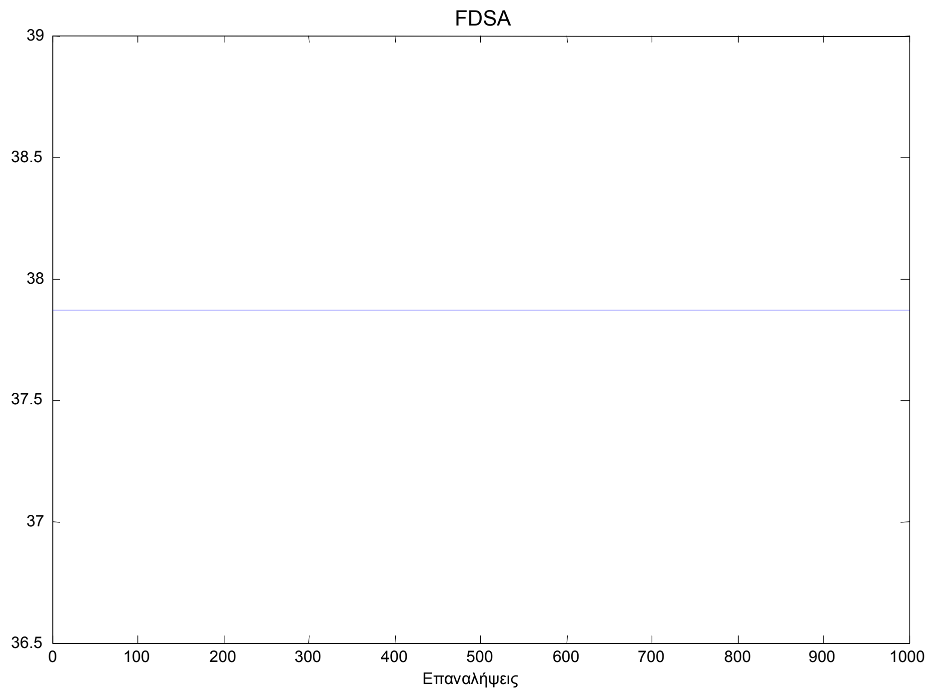
γράφημα 1

Αρχίσαμε να αλλάζουμε την αρχική τιμή της συνάρτησης και από τις πολλές προσπάθειες βρήκαμε δύο περιπτώσεις που ο αλγόριθμος «οδεύει προς τη λύση». Για x_0=(2,1) (γράφημα 2) στην $573^{η}$ επανάληψη αλλάζει την τιμή του $\hat{\theta}$ σε (-1.9123 , 2.0835) και στην f δίνει τη τιμή 256.0451.



γράφημα 2

και για x_0=(1,4) (γράφημα 3) στην 1η επανάληψη αλλάζει την τιμή του $\hat{\theta}$ σε (-1.6733 , 3.3541) και στην f δίνει τη τιμή 37.8687

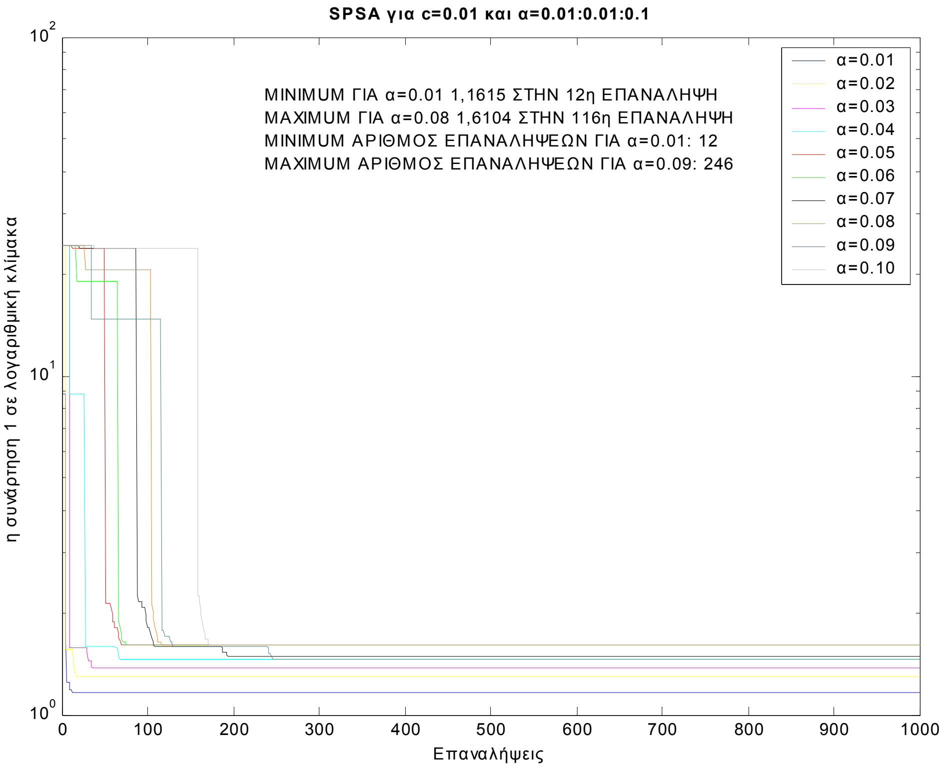


γράφημα 3

και στις 2 περιπτώσεις πάντως η τιμή της f δεν κατέβηκε κάτω από το 24,2 που είναι η τιμή της f για την αρχική τιμή x_0=(-1.2,1). Δεν έχει νόημα λοιπόν περαιτέρω έλεγχος για τη συγκεκριμένη μέθοδο, όσο αναφορά τουλάχιστον τη πρώτη συνάρτηση.

5.1.2 Μέθοδος SPSA

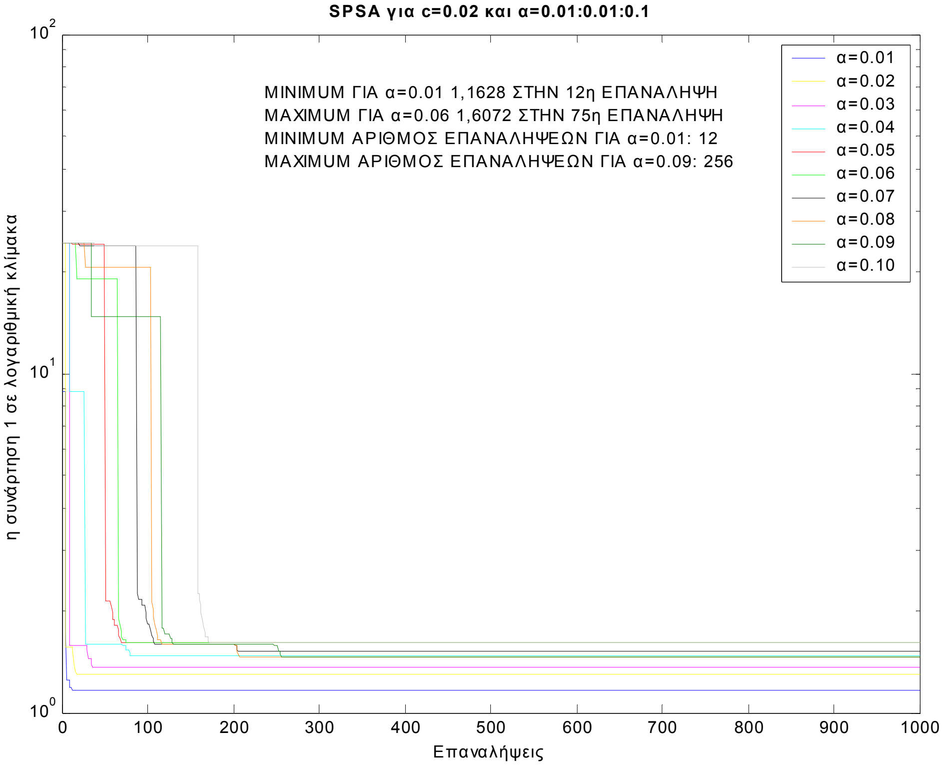
Θα προσπαθήσουμε τώρα να βρούμε τις κατάλληλες τιμές για τις παραμέτρους της μεθόδου SPSA, δηλαδή για τις α,c,A,alpha και gamma. Έχει αποδειχτεί από στα [8,9] ότι τα αποτελέσματα είναι βέλτιστα για gamma=0.101 και alpha=0.602. Επίσης το A παίρνει τιμές από το (5 έως το 10%) του αριθμού των επαναλήψεων. Στη περίπτωση μας οι επαναλήψεις είναι 1000, άρα η τιμή που μπορεί να πάρει το A κυμαίνεται από το 50 έως το 100. Η αρχική ιδέα για τον υπολογισμό των βέλτιστων τιμών για τις παραμέτρους α,c,A είναι να κρατήσουμε σταθερό το A=0 και επιλέγουμε οι παράμετροι α,c να κυμαίνονται στο διάστημα (0.01 , 0.1). Ορίζουμε αρχικά c=0.01 και αλλάζουμε την παράμετρο α. Καταλήγουμε έτσι στο γράφημα 4



γράφημα 4

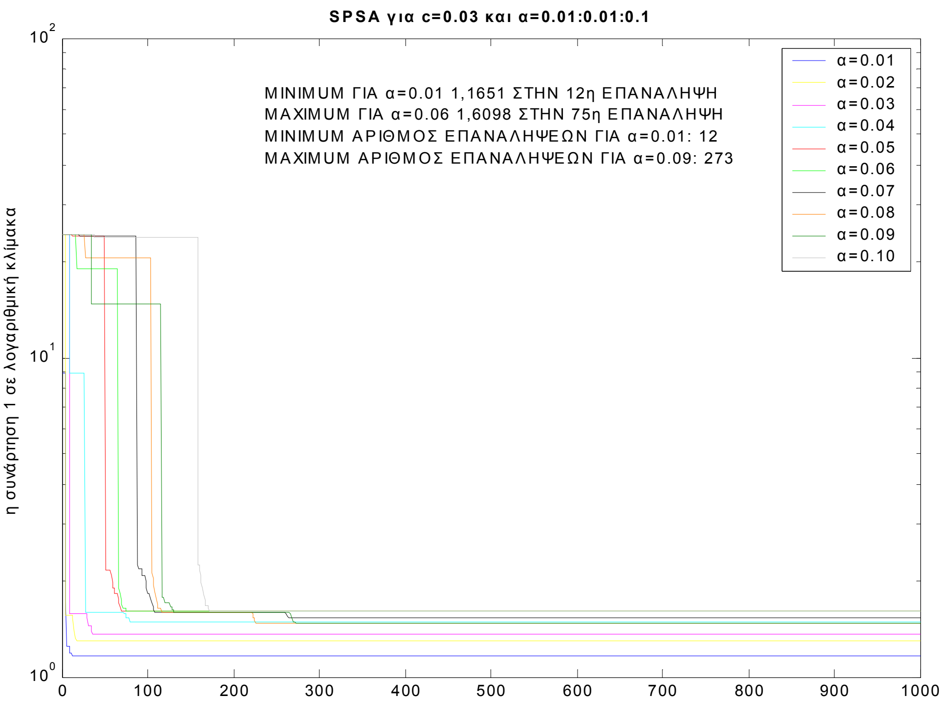
Η μέθοδος που ακολουθήθηκε ήταν η εξής: αφού ορίσαμε A=0 c=0.01 και α=0.01 «τρέξαμε» το πρόγραμμα στη Matlab. Σώσαμε τα αποτελέσματα και μετά τερματίσαμε τη λειτουργία της Matlab. Αρχίσαμε να τη λειτουργούμε πάλι ορίζοντας α=0.02 κ.ο.κ. Αυτό έγινε για τη δημιουργία όλων των διαγραμμάτων και ως στόχο είχε την μείωση της επιρροής των αποτελεσμάτων από τον τυχαίο παράγοντα. Ως γνωστόν η Matlab διαθέτει μία λίστα αριθμών και όχι μια γεννήτρια τυχαίων αριθμών, γι αυτό με αυτή τη διαδικασία , με την εντολή delta=2*round(rand(p,1))-1 είχαμε κάθε φορά το ίδιο delta.

Για c=0.02 και την ίδια διαδικασία καταλήγουμε στο γράφημα 5



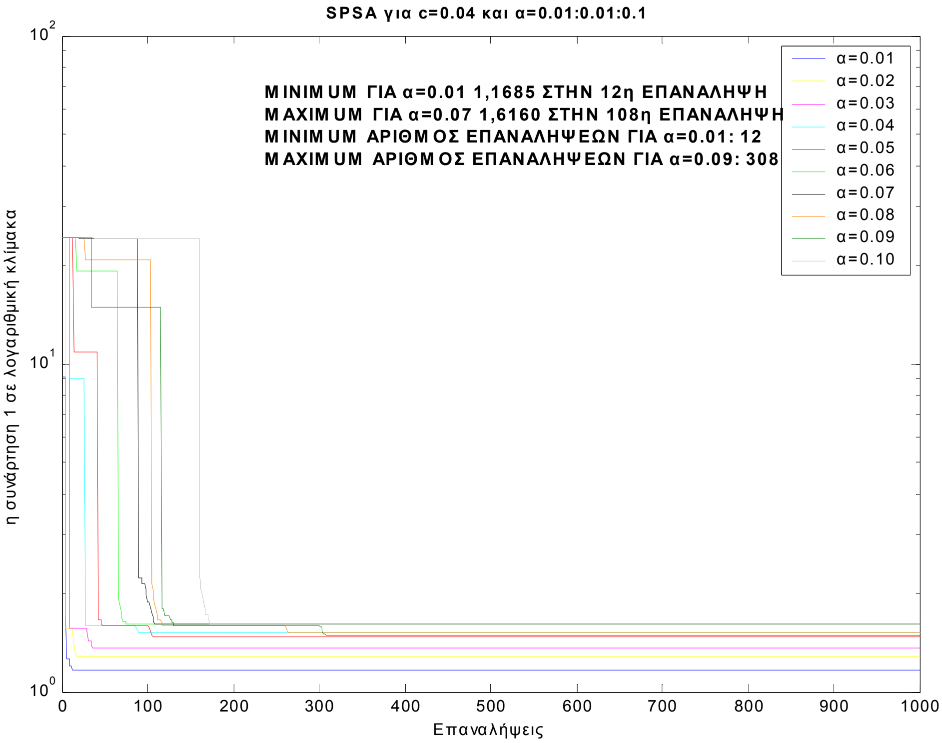
γράφημα 5

Για c=0.03 καταλήγουμε στο γράφημα 6



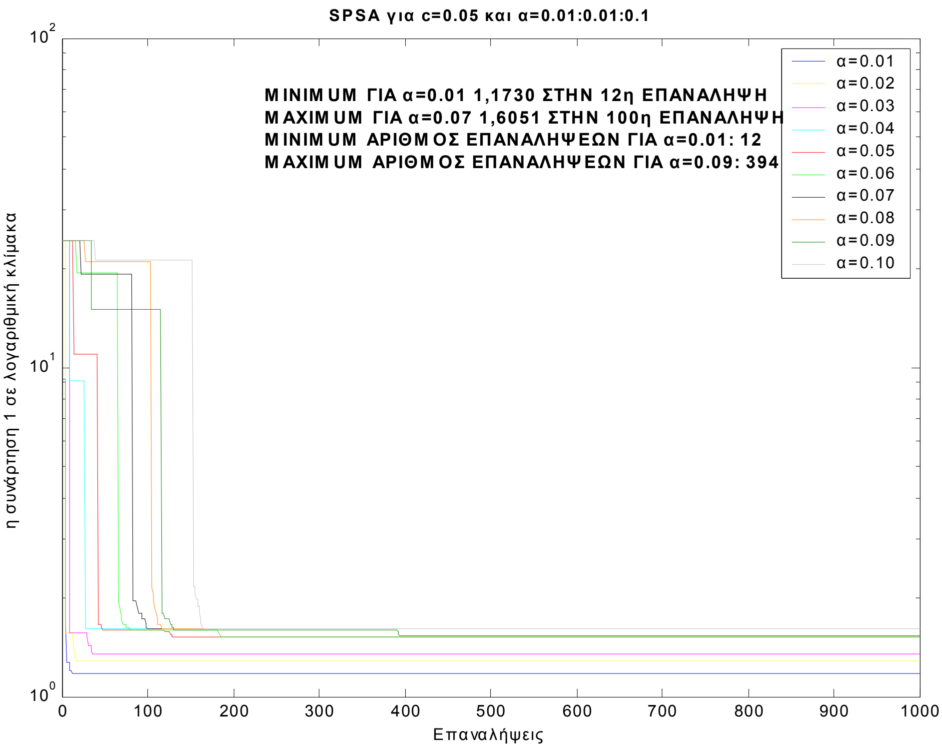
γράφημα 6

Για c=0.04 καταλήγουμε στο γράφημα 7



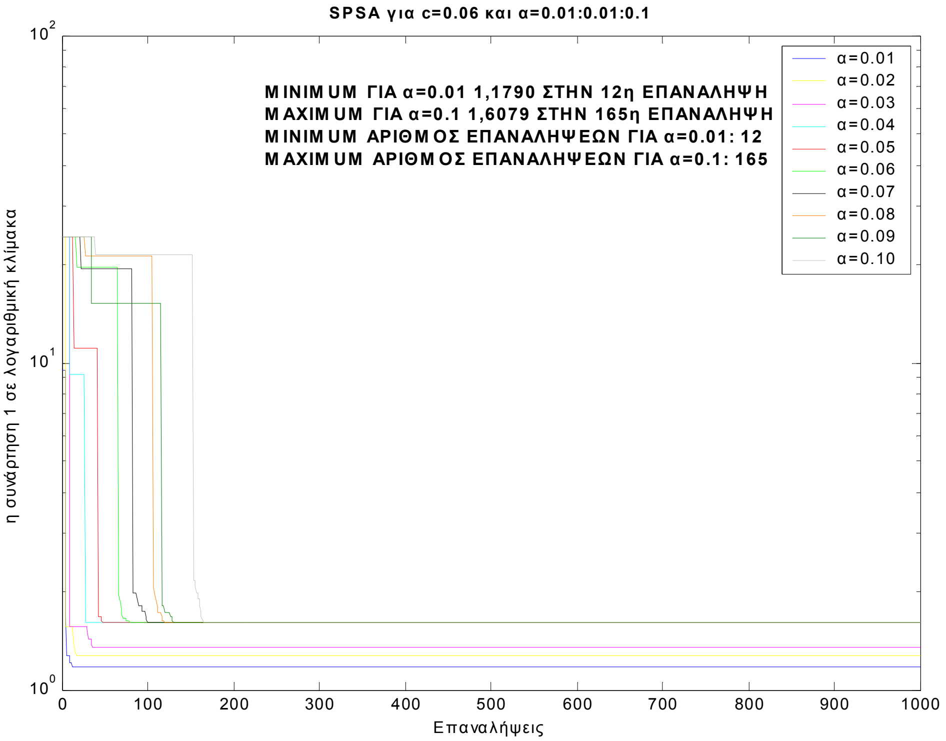
γράφημα 7

Για c=0.05 καταλήγουμε στο γράφημα 8



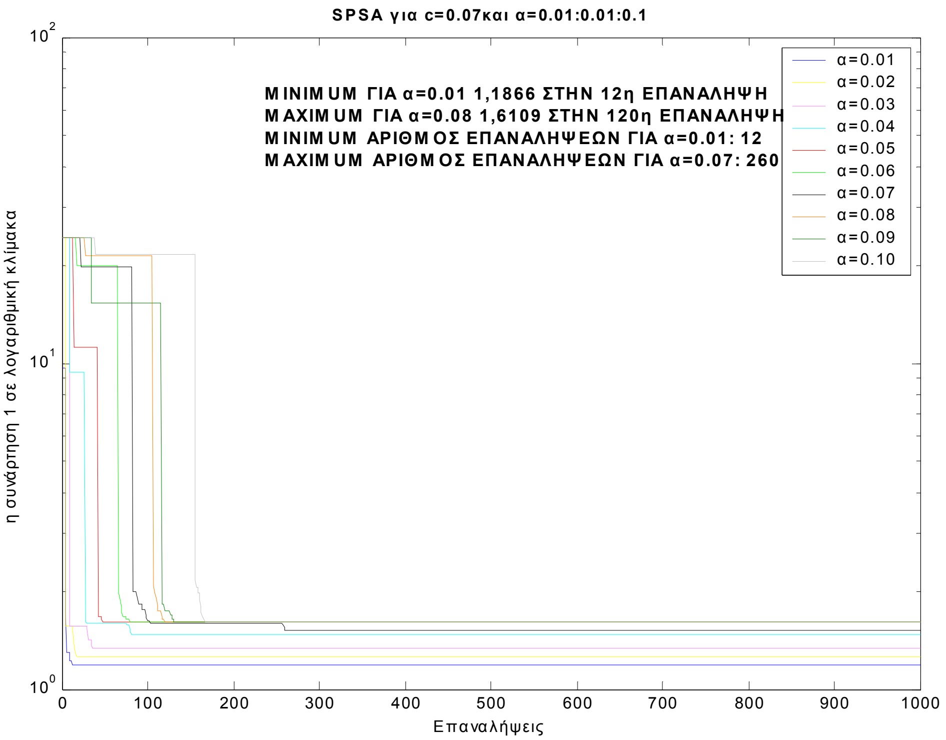
γράφημα 8

Για c=0.06 καταλήγουμε στο γράφημα 9



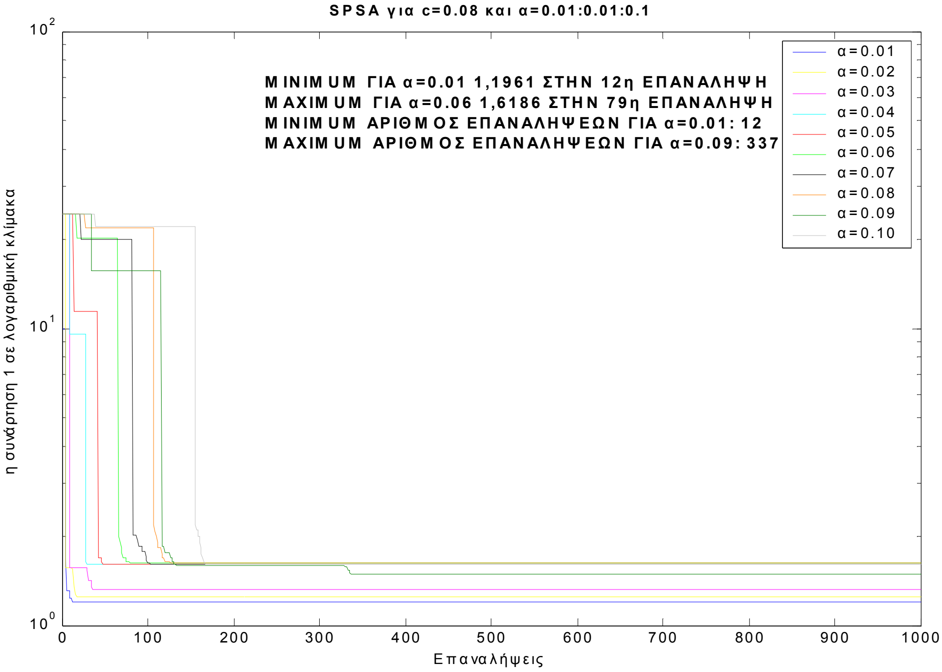
γράφημα 9

Για c=0.07 καταλήγουμε στο γράφημα 10



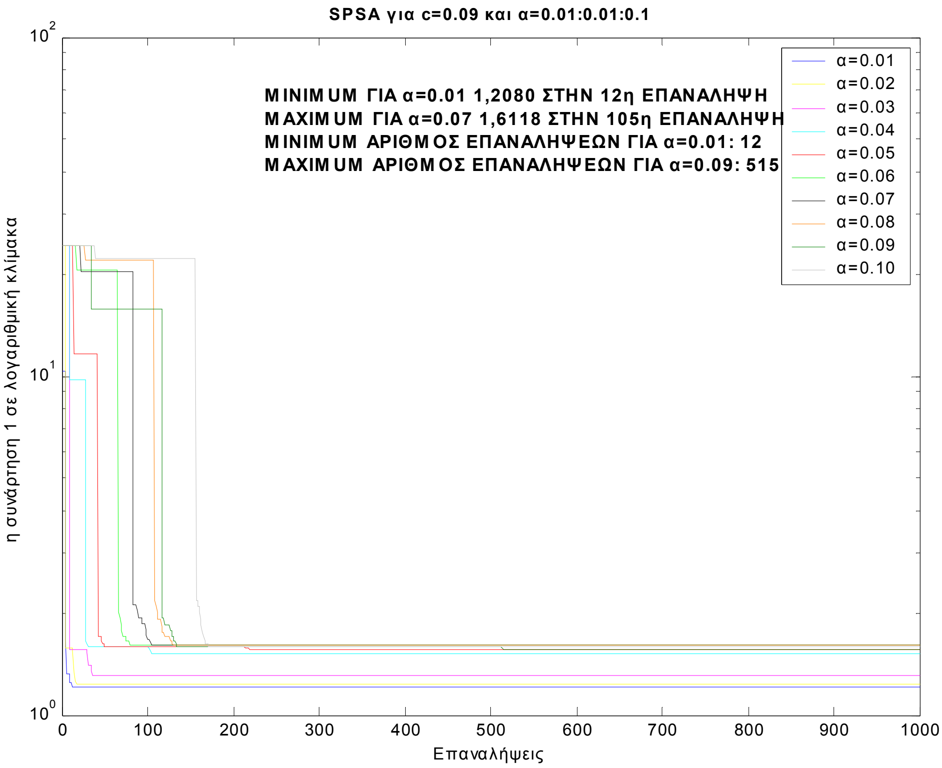
γράφημα 10

Για c=0.08 καταλήγουμε στο γράφημα 11



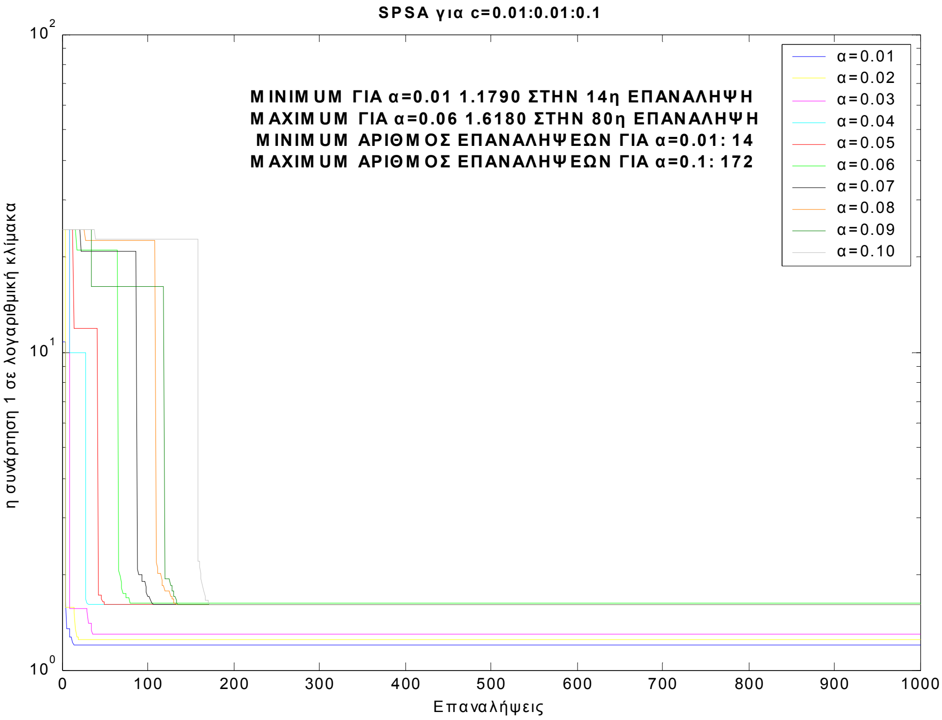
γράφημα 11

Για c=0.09 καταλήγουμε στο γράφημα 12



γράφημα 12

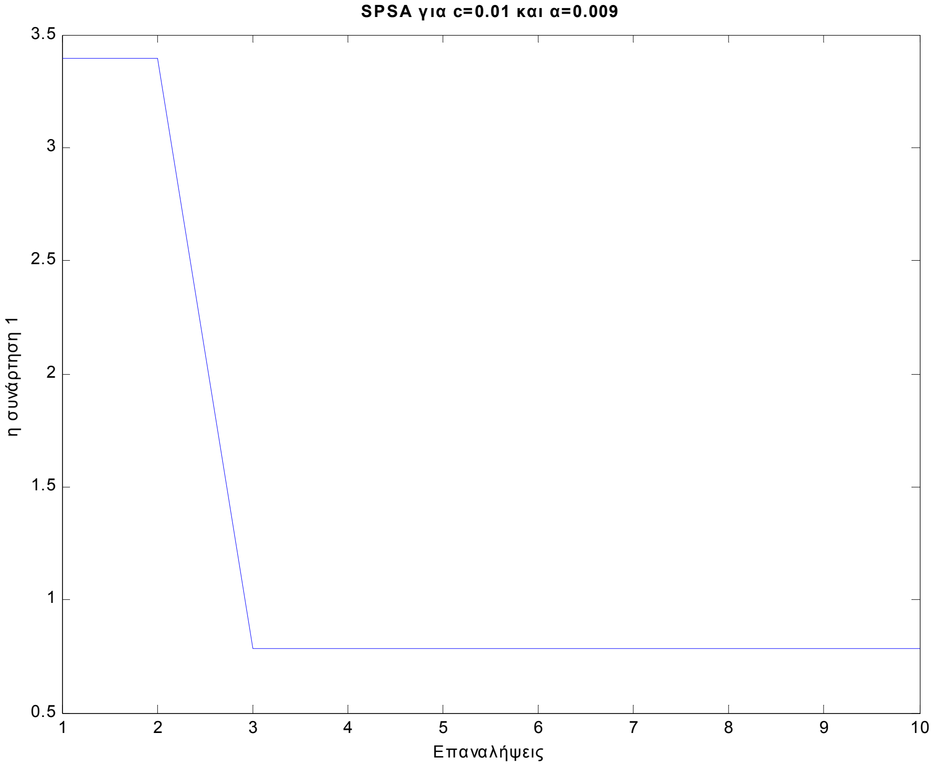
Για c=0.1 καταλήγουμε στο γράφημα 13



γράφημα 13

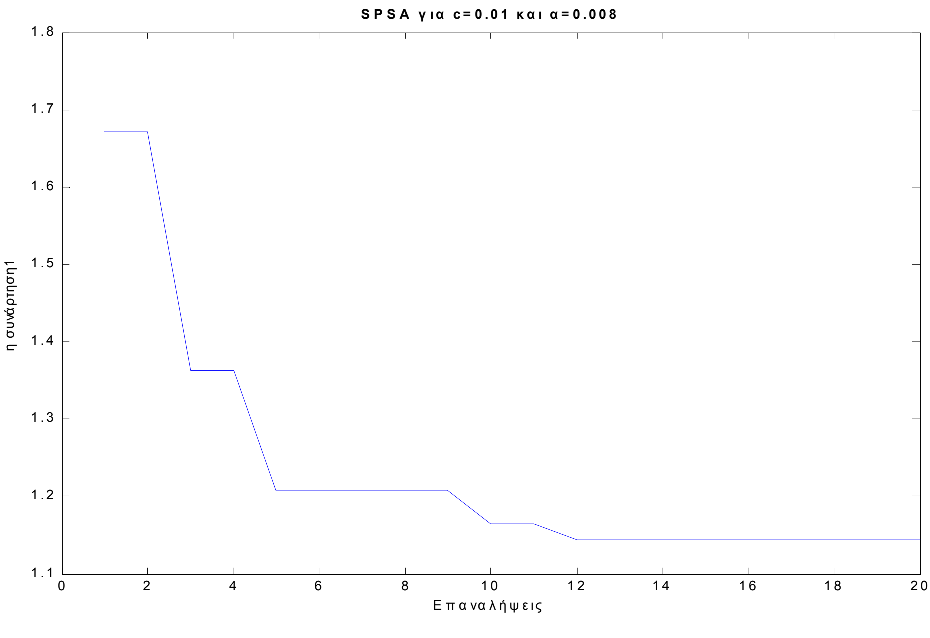
Η βέλτιστη τιμή που επιτυγχάνεται είναι 1.1615 για c=0.01 και για α=0.01 και η χείριστη 1.6186 για c=0.08 και για α=0.06. Σε όλα τα γραφήματα η καλύτερη τιμή της συνάρτησης και η καλύτερη ταχύτητα σύγκλισης δινόταν για α=0.01. Συμπεραίνουμε λοιπόν ότι τα αποτελέσματα του αλγορίθμου θα είναι καλύτερα για μικρά α. Η συνάρτηση είναι βέλτιστη για c=0.01. Μικραίνοντας το c περισσότερο από την τιμή της διασποράς διακινδυνεύουμε να μην ισχύει η δεύτερη συνθήκη του βήματος 6 του αλγορίθμου.

Θα εκτελέσουμε λοιπόν τώρα τον αλγόριθμο για τη βέλτιστη τιμή του c (c=0.01) και για μικρότερα α. Τα αποτελέσματα μπορούν να εξαχθούν από τα γραφήματα 14,15 και 16 που ακολουθούν.



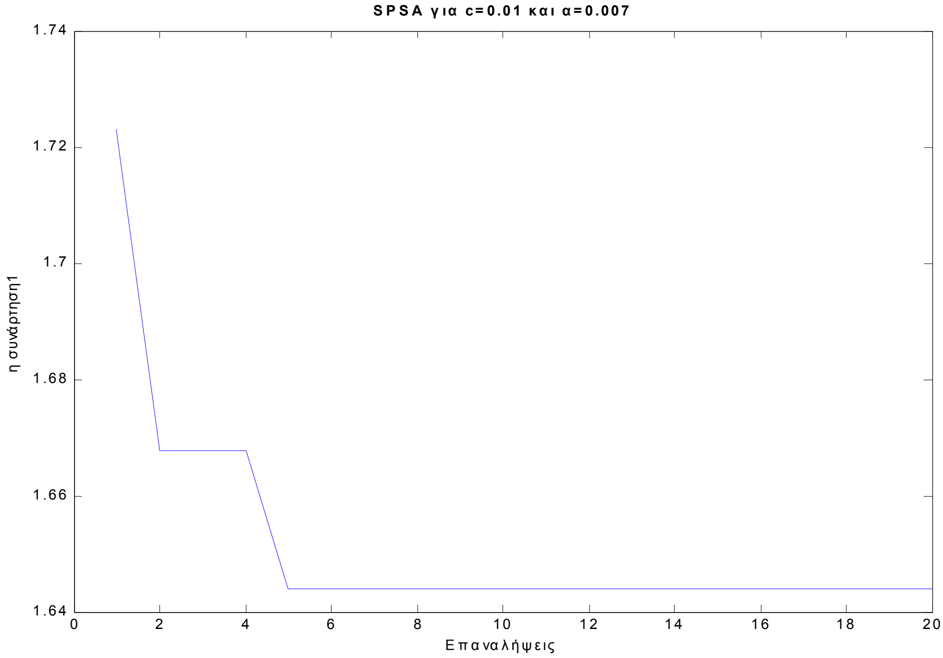
γράφημα 14

Στο γράφημα 14 βλέπουμε ότι για α=0.009 η βέλτιστη τιμή είναι 0.7841 στην $3^{η}$ μόλις επανάληψη.



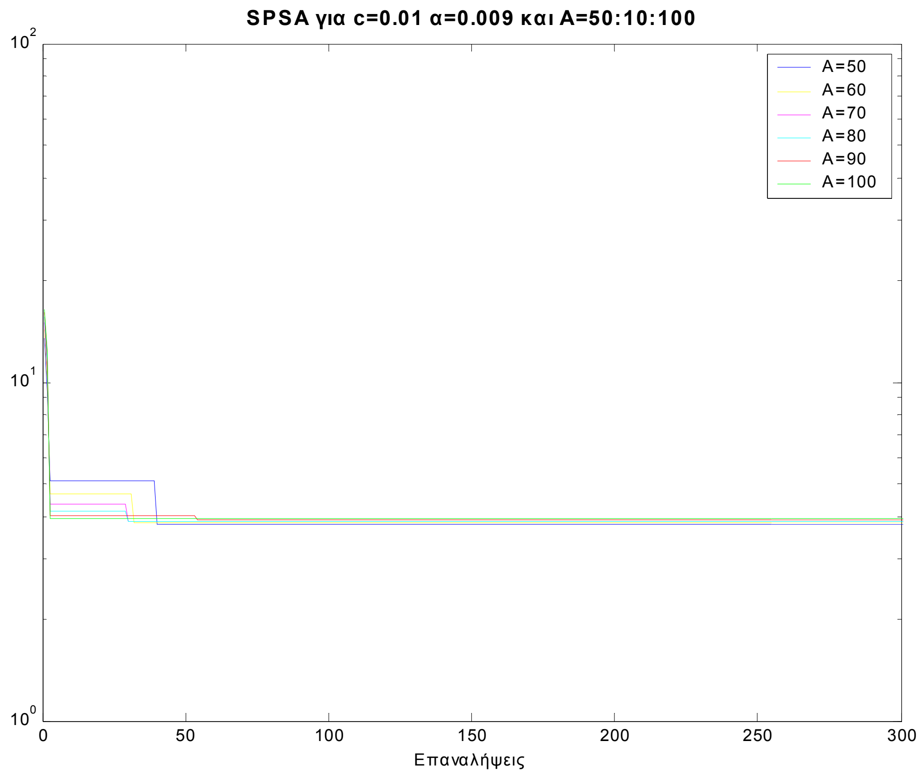
γράφημα 15

Στο γράφημα 15 βλέπουμε ότι η βέλτιστη τιμή της συνάρτησης γίνεται για α=0.008 ίση με 1.1444. Σαφώς καλύτερη από τις προηγούμενες αλλά όχι σαν αυτήν για α=0.009.



γράφημα 16

Στο γράφημα 16 παρατηρούμε ότι για α=0.007 η τιμή της συνάρτησης χειροτερεύει. Πιο συγκεκριμένα γίνεται 1.6439 στη $5^{η}$ επανάληψη. Από εδώ και εμπρός, όσο μικραίνουμε το α, τόσο χειροτερεύουν τα αποτελέσματα του αλγορίθμου. Τώρα κρατώντας σταθερές τις βέλτιστες τιμές που βρήκαμε θα διερευνήσουμε την μεταβλητή Α. Παρακάτω ακολουθεί το γράφημα 17



γράφημα 17

η καλύτερη τιμή της συνάρτησης επιτυγχάνεται για A=50 και είναι 3.8079. Παρατηρούμε ότι η τιμή της συνάρτησης είναι κατά πολύ μεγαλύτερη από όλες όσες έχουν προηγηθεί. Αυτό μας αναγκάζει να προχωρήσουμε σε ενός άλλου είδους διερεύνηση.

Θέλουμε να δούμε τη σταθερότητα του αλγορίθμου για τις τιμές των παραμέτρων α και c, και γι αυτό κάνουμε 10 προσομοιώσεις για κάθε πιθανό συνδυασμό τους. Θέλουμε έτσι να αποφύγουμε, οι βέλτιστες τιμές που έχουμε βρει να είναι τυχαίες. Μη ξεχνάμε ότι έχουμε χρησιμοποιήσει για την εύρεσή τους μόνο μια συγκεκριμένη τιμή της λίστας των ΄΄τυχαίων΄΄ αριθμών της Matlab. Τα αποτελέσματα δίνονται στον πίνακα 2 που ακολουθεί. Στις τελευταίες στήλες δίνονται τα αποτελέσματα του αλγορίθμου για A=50 και όχι A=0 όπως οι πρώτες 10 παρατηρήσεις.

α	c	Π1	Π2	Π3	Π4	Π5	Π6	Π7	Π8	Π9	Π10	**A=50**	**A=50**	**A=50**
0,01	0,01	1,6808	2,9367	1,4146	**2,9384**	1,4146	1,6655	1,2329	1,1539	2,5402	**0,9364**	***3.9361***	***4.4940***	***3.9627***
0,02	0,01	0,9674	**1,6608**	0,9725	1,6287	1,2988	0,9645	1,6174	1,6158	1,3683	**0,9612**	***2.9735***	***2.9820***	***3.1331***
0,03	0,01	1,5142	1,4992	0,9695	1,4683	**0,6888**	1,5074	0,9871	0,9717	**0,6888**	**1,6137**	***2.9370***	***2.9146***	***2.9403***
0,04	0,01	**0,6023**	1,4629	**0,9432**	0,9880	1,4665	1,3957	1,6095	0,8101	1,3197	1,4706	***2.8845***	***2.8830***	***2.8736***
0,05	0,01	1,6159	**0.9919**	1,6153	**1,6144**	1,6163	0,9973	1,6048	1,6138	1,6048	1,6064	***2.7226***	***2.7336***	***2.7336***
0,06	0,01	1,6011	1,4848	1,6022	1,6042	1,4409	**1.2604**	1,4699	**1,6050**	1,4549	1,2681	***1.6089***	***1.5061***	***1.6088***
0,07	0,01	1,5152	1,4758	1,5333	1,4539	**1,6124**	1,6085	1,6013	1,6116	1,6114	**1,4624**	***1.4376***	***1.4623***	***1.5209***
0,08	0,01	1,4647	1,4701	**1,6121**	1,4858	1,6066	**1,4624**	1,4694	1,5207	1,5074	1,5342	***1.4199***	***1.4479***	***1.3499***
0,09	0,01	1,6085	**1,6103**	**1,5273**	1,6043	1,6015	1,5305	1,6013	1,6025	1,6027	1,6072	***0.9765***	***0.9650***	***0.9701***
0,1	0,01	1,6083	1,5286	1,6042	1,6085	1,4727	1,6024	**1,6105**	**1,4424**	1,6048	1,6090	***0.9910***	***0.9963***	***0.9901***
0,01	0,02	1,6735	**0,9414**	1,2121	2,2102	1,6459	1,1551	**2,9153**	1,2000	1,6438	1,6959	***3.7738***	***3.7697***	***4.4979***
0,02	0,02	1,2896	1,6120	**0,9628**	1,6121	**1,6240**	0,9700	1,6148	1,5101	1,2870	0,9700	***2.9853***	***2.9905***	***2.9948***
0,03	0,02	1,5022	1,1193	1,2537	1,6092	0,9755	**0,7123**	**1,6102**	1,6100	1,5097	1,5100	***2.9138***	***2.9208***	***2.9227***
0,04	0,02	1,6021	1,5107	**1,6147**	1,1701	1,1638	1,1624	1,6040	**0,8225**	1,4695	1,5291	***2.8737***	***2.8826***	***2.8732***
0,05	0,02	1,6024	**0,7280**	1,6121	**1,6221**	1,6154	0,9243	1,3419	1,6062	1,5193	1,5186	***2.7220***	***2.7536***	***2.7220***
0,06	0,02	1,5240	1,6048	1,4723	1,4398	1,5173	1,6071	**1.2556**	**1,6097**	1,4648	1,5309	***1.6089***	***1.6006***	***1.6070***
0,07	0,02	1,6048	1,3965	1,4812	1,5179	1,6050	1,3983	1,5065	1,6075	**1,1783**	**1,6106**	***1.4322***	***1.4685***	***1.4834***
0,08	0,02	1,4779	**1,6105**	1,6091	1,5178	**1,4701**	1,4750	1,6095	1,6054	1,4852	1,4718	***1.4460***	***1.4452***	***1.4168***
0,09	0,02	1,6015	**1,6143**	1,5236	1,4800	1,6103	1,4699	**1,4463**	1,6030	1,6018	1,5216	***0.9930***	***1.1110***	***0.9839***
0,1	0,02	1,6021	**1,4466**	**1,6085**	1,6049	1,6065	1,5142	1,4769	**1,6085**	1,5147	1,4581	***0.9982***	***0.6907***	***0.6894***
0,01	0,03	1.6794	2.9345	1.4884	**2.9362**	1.4851	1.6644	**0.8951**	1.1571	2.5369	0.9481	***3.7853***	***3.5636***	***3.7745***
0,02	0,03	1.2913	0.9598	**1.6235**	1.6012	1.3761	**0.9557**	0.9586	1.5227	1.5291	0.9571	***2.9952***	***2.9668***	***2.9826***
0,03	0,03	1.4678	**0.9806**	1.4645	1.5201	1.0042	**1.6014**	0.9877	1.4997	1.3520	1.4599	***2.9190***	***2.9275***	***2.9293***
0,04	0,03	1.5122	1.6032	**0.8038**	1.5154	1.6009	1.6086	1.1560	0.9705	**1.6047**	1.5272	***2.8715***	***2.8820***	***2.8843***
0,05	0,03	1.6031	1.6033	1.4507	1.5184	1.6111	1.4623	1.6021	1.5171	1.5105	1.1334	***2.7210***	***2.7322***	***2.7210***
0,06	0,03	1.6025	1.4791	1.6132	**1.4667**	1.6021	1.4683	1.4820	1.6003	1.5158	1.6048	*1.5074*	*1.6039*	*1.5146*
0,07	0,03	1.5321	1.5281	1.6081	1.6027	**1.4459**	1.5218	**1.6164**	1.6029	1.4586	1.4791	*1.4298*	*1.4313*	*1.4711*
0,08	0,03	1.5087	1.5214	1.4682	**1.6074**	1.6039	1.5165	1.6016	1.5258	**1.4670**	1.4705	*1.4110*	*1.3990*	*1.4125*
0,09	0,03	1.4836	1.6015	1.6023	1.5238	**1.6057**	1.5071	1.5200	**1.4781**	1.6021	1.6020	*0.9984*	*0.9819*	*0.9676*
0,1	0,03	**3.1307**	1.6415	0.9522	1.4710	**0.8564**	2.8960	1.6672	1.4819	1.4426	1.1558	*0.9945*	*0.6890*	*0.7023*
0,01	0,04	**0.9590**	1.4970	**2.7894**	1.5309	1.4629	1.1611	1.2753	1.4537	1.4864	1.6457	*3.8809*	*4.5139*	*3.9478*
0,02	0,04	1.6073	**1.6135**	0.9626	0.9558	1.6038	**0.9525**	**1.6135**	1.6068	0.9546	1.6135	*2.9786*	*2.9875*	*2.9884*
0,03	0,04	1.5246	1.3454	1.5200	1.0907	1.0023	1.1114	**0.9880**	**1.6017**	1.5166	1.3418	*2.9251*	*2.9220*	*2.9091*
0,04	0,04	1.1107	1.5240	1.5162	**1.6095**	1.6052	1.4484	1.1459	1.6019	**0.8019**	1.5235	*2.8724*	*2.8688*	*2.8694*
0,05	0,04	1.5181	1.4601	1.6063	1.5557	0.7198	1.5074	**1.6057**	1.4607	**0.7194**	1.3567	*2.7519*	*2.7195*	*2.7195*
0,06	0,04	1.5107	1.5275	1.6034	1.5309	1.5222	1.5338	**1.6133**	1.6090	**1.5105**	1.6014	*1.6034*	*1.6009*	*1.4852*
0,07	0,04	1.6018	**1.4546**	1.6044	1.6004	**1.6165**	1.6103	1.6040	1.6016	1.6094	1.6018	*1.4394*	*1.4405*	*1.4823*
0,08	0,04	1.5111	1.6014	1.6032	1.5126	1.4785	**1.3656**	1.4687	**1.6105**	1.4822	1.6004	*1.4442*	*1.3333*	*1.3316*
0,09	0,04	1.6097	1.6098	1.6062	1.4872	**1.6108**	1.6060	1.4829	1.6105	**1.4737**	1.6005	*0.9970*	*1.0039*	*0.9112*
0,1	0,04	1.6073	1.6045	1.6033	1.5224	**1.6106**	1.4869	**1.4682**	1.6052	1.4765	1.4763	*1.0032*	*1.0032*	*1.0037*
0,01	0,05	2.1914	1.6444	**1.1639**	**2.9104**	1.2119	1.6424	1.7106	1.7025	2.7881	1.1651	*3.8695*	*3.7682*	*3.8788*
0,02	0,05	1.6187	1.6159	1.6101	1.6022	**0.9407**	1.2952	1.6054	1.6175	**1.6256**	0.9506	*2.9817*	*2.9757*	*2.9867*
0,03	0,05	**1.0922**	**1.6084**	1.3515	1.6022	1.5554	1.5189	**1.6084**	1.3405	**1.6084**	1.6006	*2.9250*	*2.9277*	*2.9258*
0,04	0,05	0.8123	1.5150	1.5199	1.5218	1.6007	**0.8035**	**1.6187**	1.6129	1.1275	1.6087	*2.8665*	*2.8661*	*2.8696*
0,05	0,05	1.6080	**0.7138**	1.4772	1.6082	1.4734	**1.6090**	1.6040	1.6043	1.5225	1.4026	*2.7176*	*2.7402*	*2.7176*
0,06	0,05	1.6038	1.5068	1.4747	**0.7131**	1.4807	1.6066	1.6114	1.6138	**1.6149**	1.5073	*1.6019*	*1.5216*	*1.6157*
0,07	0,05	1.4773	1.6035	1.6053	1.6044	1.4720	1.4849	1.5141	**1.6081**	1.6058	**1.4486**	*1.4369*	*1.4298*	*1.4405*

0,08	0,05	1.6011	1.6008	1.4793	1.5551	**1.6020**	**1.1306**	1.5125	1.6004	1.4805	1.4564	*1.3192*	*1.3191*	*1.3196*
0,09	0,05	1.5131	**1.4855**	1.6008	1.5082	1.5082	1.6095	1.6095	1.6144	1.5127	**1.6152**	*0.9449*	*0.9142*	*0.9951*
0,1	0,05	1.4761	1.4799	1.4794	1.4839	1.6024	1.4768	**1.6042**	1.6012	**1.4592**	1.5150	*0.8591*	*0.6856*	*0.7221*

Πίνακας 2

Έως εδώ έχουμε βγάλει το ασφαλές συμπέρασμα, ότι για α=0.1 έχουμε τη βέλτιστη συμπεριφορά του αλγορίθμου. Πράγματι, όπως τονίσαμε προηγουμένως, για α=0.09 και α=0.1 παρατηρείται μια σχετική σταθερότητα στα εξαγόμενα του αλγορίθμου που μας οδηγεί σε πολύ καλές προσεγγίσεις. ***(οι τελικές προσεγγίσεις στο ελάχιστο της f κυμαίνονται στο διάστημα (1.4,1.65) για A=0 και όταν δώσουμε στο A μια επιτρεπτή τιμή (50 έως 100) παρατηρείται βελτίωση των αποτελεσμάτων).*** Όσο το α μειώνεται, δεν υπάρχει σταθερότητα και τελικά οδηγούμαστε σε κακές προσεγγίσεις. ***(για α=0.01 το αντίστοιχο διάστημα είναι το (0.8,3) και όταν δώσουμε στο A μια επιτρεπτή τιμή (50 έως 100) παρατηρούνται τα χειρότερα αποτελέσματα που ξεπερνούν ακόμα και το 4)***

Για να βρούμε το βέλτιστο c θα εκτελέσουμε τον αλγόριθμο 5 φορές για κάθε τιμή του c στο διάστημα (0.02,0.1) χρησιμοποιώντας κάθε φορά την ίδια σειρά «τυχαίων αριθμών» της Matlab. Θα χρησιμοποιήσουμε φυσικά τα βέλτιστα α, A (0.1 και 50 αντίστοιχα).

Στατιστικά αποτελέσματα SPSA για A=50, α=0.1 και c=0.02:0.01:0.1										
	Π1		Π2		Π3		Π4		Π5	
c	$f\left(\hat{\theta}\right)$	Επ/ψεις	$f\left(\hat{\theta}\right)$	Επ/ψεις	$f\left(\hat{\theta}\right)$	Επ/ψεις	$f\left(\hat{\theta}\right)$	Επ/ψεις	$f\left(\hat{\theta}\right)$	Επ/ψεις
0.02	**0.9890**	144	0.9991	102	0.8746	**59**	**0.9888**	141	0.8705	**67**
0.03	0.9914	128	0.9999	102	0.8702	61	0.9913	125	0.8673	67
0.04	1.0014	**95**	1.0035	102	0.8659	61	1.0025	**96**	0.8628	67
0.05	1.1311	98	1.0020	99	0.8584	62	1.0020	139	0.8568	67
0.06	1.1428	99	**0.9853**	510	0.8507	64	1.1480	99	0.8494	67
0.07	1.1690	105	0.9943	176	0.8395	68	1.1706	111	0.8382	71
0.08	1.2002	117	1.1019	**96**	0.8259	69	1.1910	114	0.8238	73
0.09	1.2085	131	1.1212	100	0.8098	73	1.2079	127	0.8107	73
0.1	1.2364	145	1.1480	110	**0.7932**	77	1.2337	140	**0.8109**	75

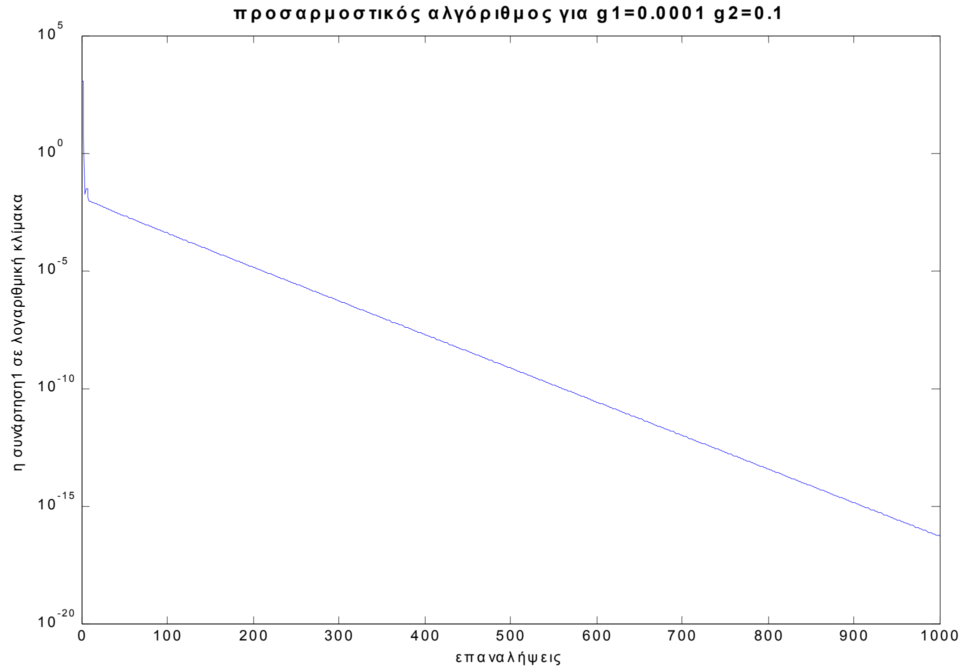
Πίνακας 3

Στον πίνακα 3 οι τιμές με έντονο φόντο είναι οι βέλτιστες και οι υπογραμμισμένες οι χείριστες. Παρατηρούμε ότι δε μπορούμε να βγάλουμε ασφαλή συμπεράσματα για τη βέλτιστη τιμή του c. Σε μερικές περιπτώσεις όσο αυξάνεται τόσο βελτιώνεται η εκτίμησή μας και σε άλλες το αντίθετο. Αυτό αποδεικνύεται όχι μόνο από τις πέντε παρατηρήσεις του πίνακα 3 αλλά και από άλλες παρατηρήσεις που δεν παρατίθενται στη συγκεκριμένη εργασία.

5.1.3 Προσαρμοστικός αλγόριθμος

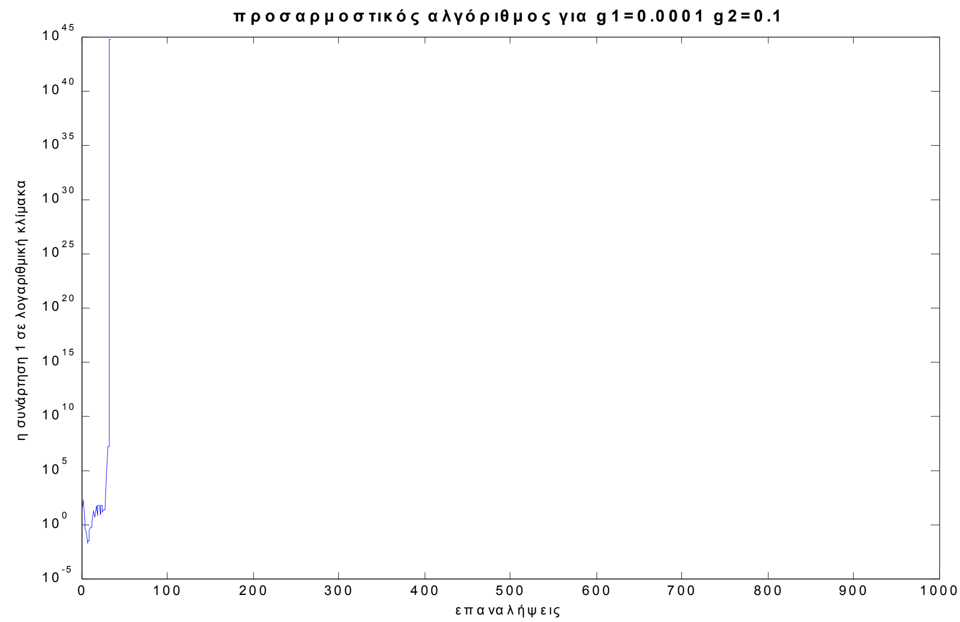
Στον προσαρμοστικό αλγόριθμο το αν θα καταφέρουμε να προσεγγίσουμε το ελάχιστο της συνάρτησης ή την άγνωστη παράμετρο, εκτός από την επιλογή των παραμέτρων g1 και g2, σημαντικό ρόλο παίζει και η αρχική τιμή του thetab (θ), η οποία είναι τυχαία. Όπως θα δούμε και παρακάτω, δεν υπάρχουν συγκεκριμένα g1,g2 για τα οποία ο αλγόριθμος να δίνει πάντα επιθυμητά αποτελέσματα, και επίσης δεν υπάρχουν συγκεκριμένες αρχικές τιμές του thetab για τις οποίες ο αλγόριθμος θα λειτουργεί ανεξαρτήτως των g1,g2. Παρόλα αυτά εμείς θα προσπαθήσουμε να βρούμε τα συγκεκριμένα g1,g2, για τα οποία ο αλγόριθμος, τις περισσότερες φορές (για τις περισσότερες αρχικές τιμές του thetab) καταλήγει σε ικανοποιητική λύση.

Για g1=0.0001 και g2=0.1 έχουμε τα παρακάτω γραφήματα (γραφήματα 18,19 και 20)



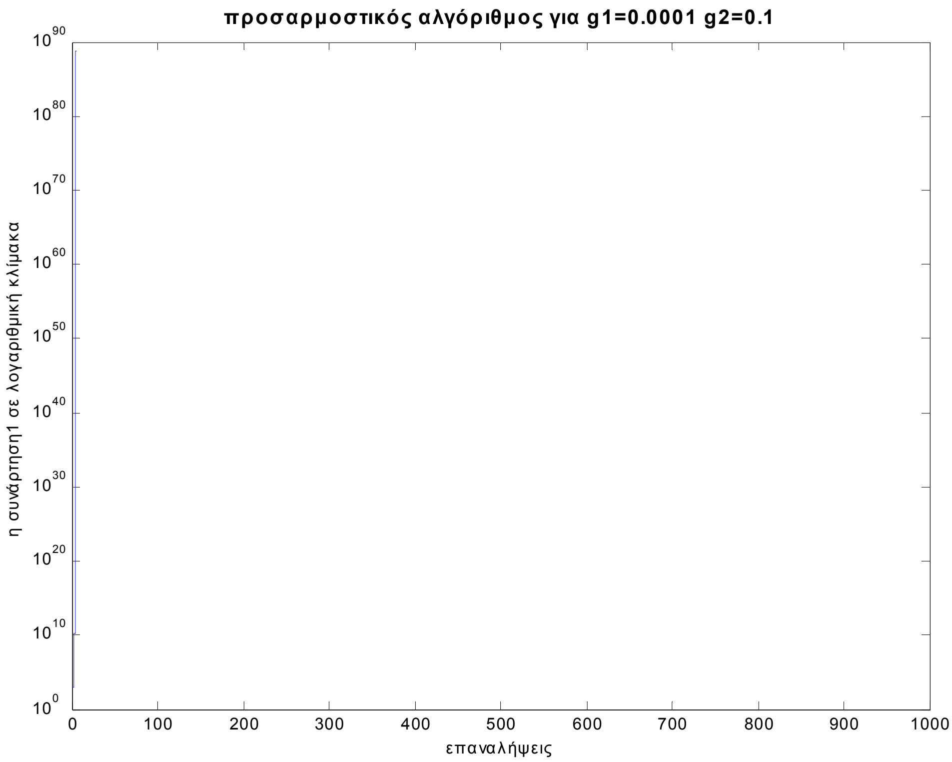
Fmin = 5.1203e-017
epanal = 1000
x1b = 1.0000
x2b = 1.0000
thetab = (3.2537, 0.1986)

γράφημα18



r = 12
Fmin = 0.0179
epanal = 7
x1b = 1.1146
x2b = 1.2492
thetabzitoymeno = (5.4772 , 3.8944)

γράφημα19



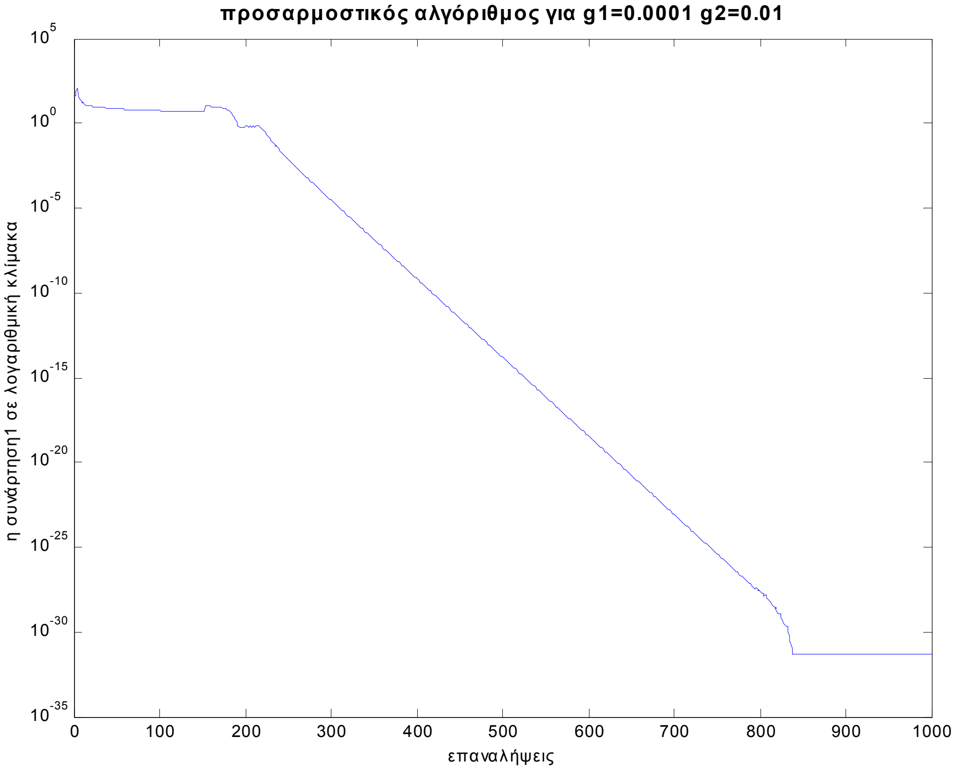
r = 4
Fmin = 956.6210
epanal = 1
x1b = -1.9136
x2b = 0.5826
thetabzitoymeno = (-0.7841, -1.2454)

γράφημα 20

για 50 επαναλήψεις είχαμε 7 γραφήματα σαν το γράφημα 18. Για τις υπόλοιπες περιπτώσεις ο αλγόριθμος δεν έδωσε ικανοποιητικά αποτελέσματα (γραφήματα 19 και 20)

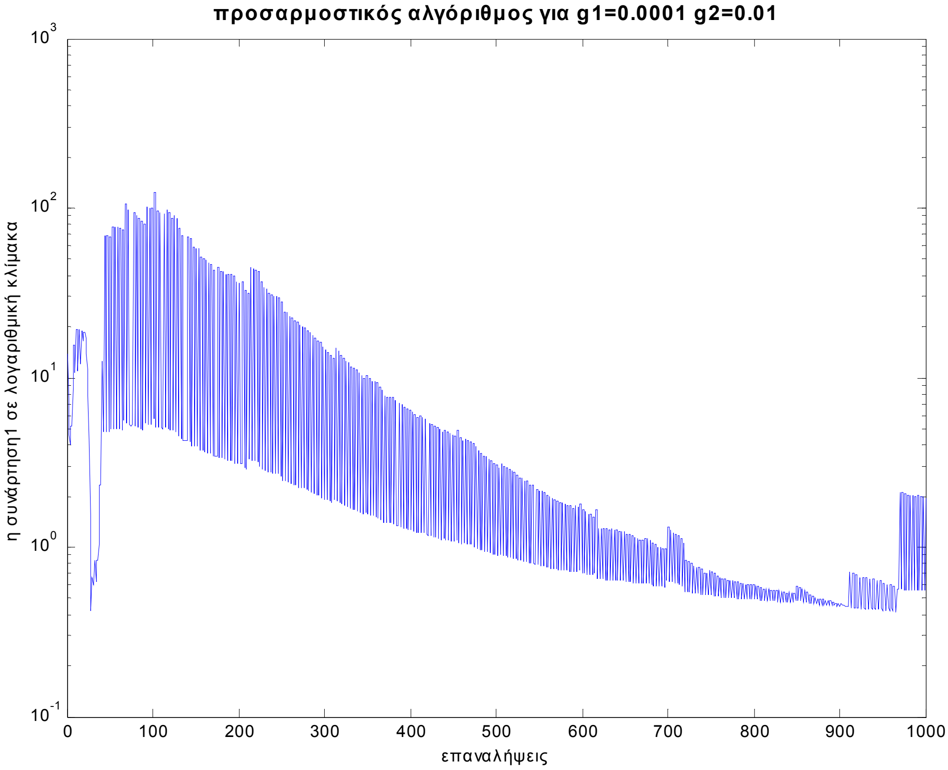
Λόγω του μεγάλου g1, όταν ο αλγόριθμος «έτρεξε» προσέγγισε το ελάχιστο της συνάρτησης με ακρίβεια, πολύ γρήγορα. Λόγω του μικρού g2 δεν καταφέραμε να προσεγγίσουμε την πραγματική τιμή της άγνωστης παραμέτρου θ^*. Για αυτές τις τιμές των παραμέτρων g1,g2 είχαμε 14% επιτυχία στην προσέγγιση στο ελάχιστο της συνάρτησης και 0% επιτυχία στην εκτίμηση της παραμέτρου θ .

Για g1=0.0001 και g2=0.01 έχουμε τα παρακάτω:



r = 19
Fmin = 4.9304e-032
epanal = 838
x1b =1.0000
x2b =1.0000
thetabzitoymeno = (45.2522, 12.2061)

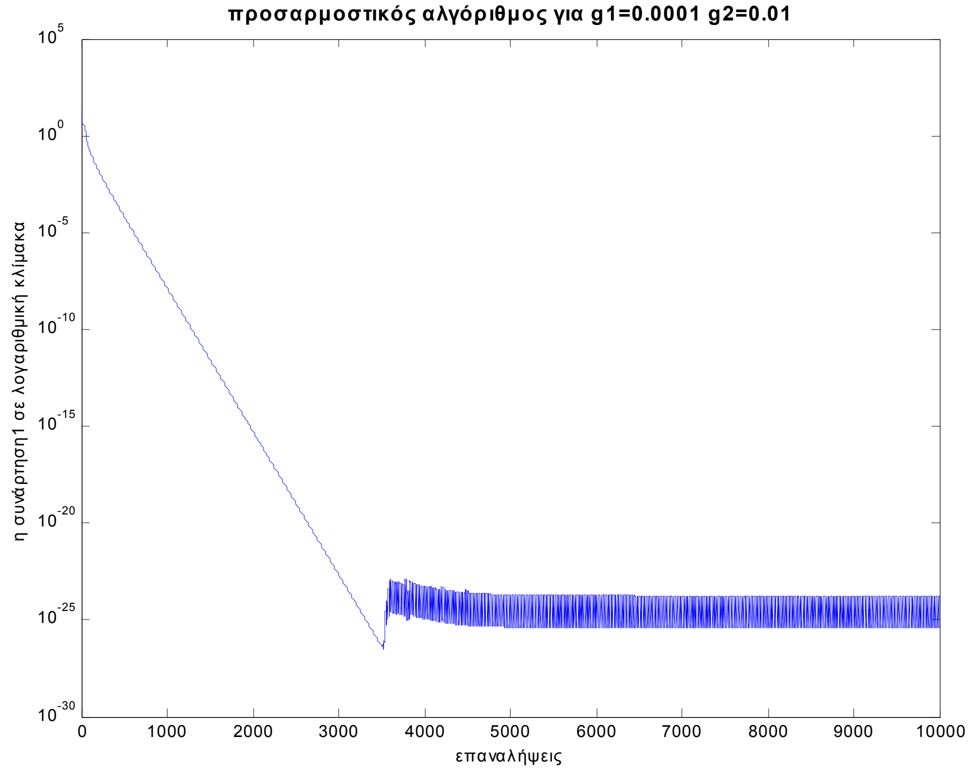
γράφημα 21



r = 328
Fmin = 0.4173
epanal = 966
x1b = 0.4652
x2b = 0.2526
thetabzitoymeno = (99.7704, 2.6446)

γράφημα 22

Για g1=0.0001 και g2=0.01 για 50 επαναλήψεις ο αλγόριθμος λειτουργεί ικανοποιητικά στις 48. Από αυτές τις 48, στις 39 έχουμε πολύ καλή προσέγγιση στο ελάχιστο της συνάρτησης και όχι τόσο καλή εκτίμηση της άγνωστης παραμέτρου θ (γράφημα 21) και στις υπόλοιπες 9 το αντίθετο (γράφημα 22). Παρακάτω δίνονται 2 γραφήματα για δέκα χιλιάδες επαναλήψεις, από τα οποία μπορούμε να βγάλουμε χρήσιμα συμπεράσματα. Το ποσοστό επιτυχίας για αυτές τις τιμές των παραμέτρων g1,g2 είναι 96% και για πολύ καλές προσεγγίσεις 78%.



r = 2167
Fmin = 2.7605e-027
epanal = 3521
x1b = 1.0000
x2b = 1.0000
thetabzitoymeno = (58.3201, 13.3646)
x1 = 1.0000
x2 = 1.0000
thetab = (100.0000 , 1.0000)

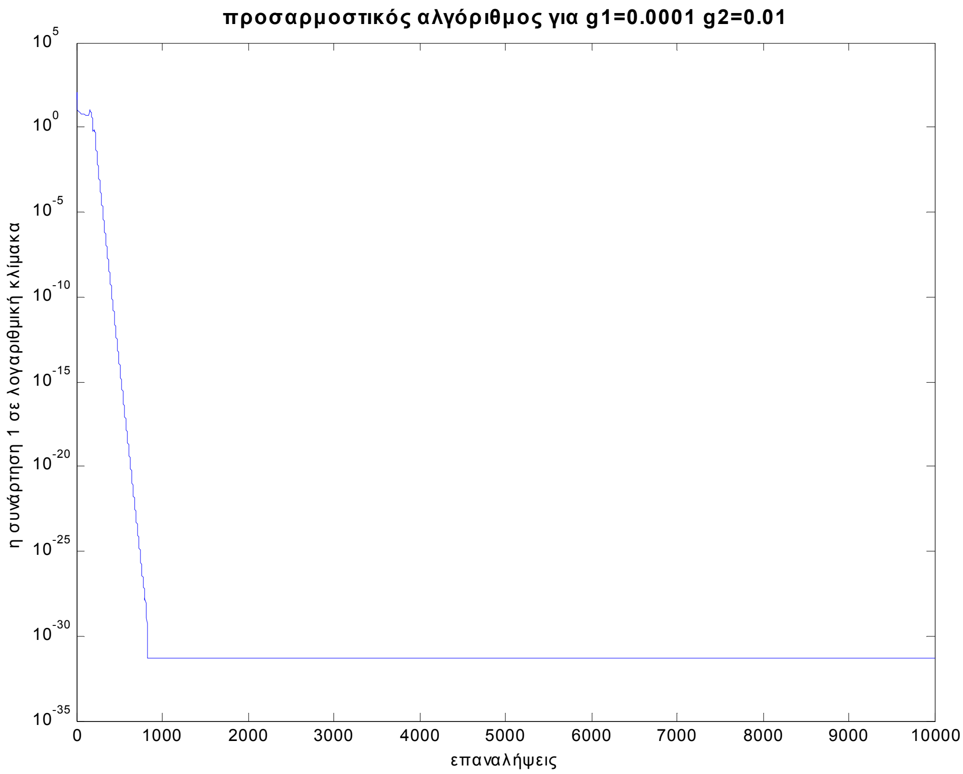
γράφημα 23

Από το γράφημα 23 συμπεραίνουμε ότι αν έχουμε πολλές επαναλήψεις και ο αλγόριθμος, αφού προσεγγίσει το ελάχιστο της f, καταφέρει να ικανοποιήσει την παρακάτω συνθήκη «F-F_old > tolerence», τότε μπορεί ο αλγόριθμος να εκτιμήσει σωστά και την άγνωστη παράμετρο θ. Βλέπουμε από το γράφημα ότι στην 3521η επανάληψη έχουμε την καλύτερη προσέγγιση στο ελάχιστο της συνάρτησης, αλλά όχι τόσο καλή εκτίμηση της άγνωστης παραμέτρου.

thetabzitoymeno = (58.3201, 13.3646) δηλαδή η εκτίμηση της άγνωστης παραμέτρου θ στην 3521η επανάληψη

Μετά από την 3521η επανάληψη, και αφού ο αλγόριθμος μπει στη «random διαδικασία», χωρίς να αλλάξει στο ελάχιστο τις μεταβλητές x1, x2, εκτιμά ακριβώς την άγνωστη παράμετρο θ.

<u>thetab = (100.0000 , 1.0000) δηλαδή η εκτίμηση της άγνωστης παραμέτρου θ στο τέλος του αλγορίθμου</u>

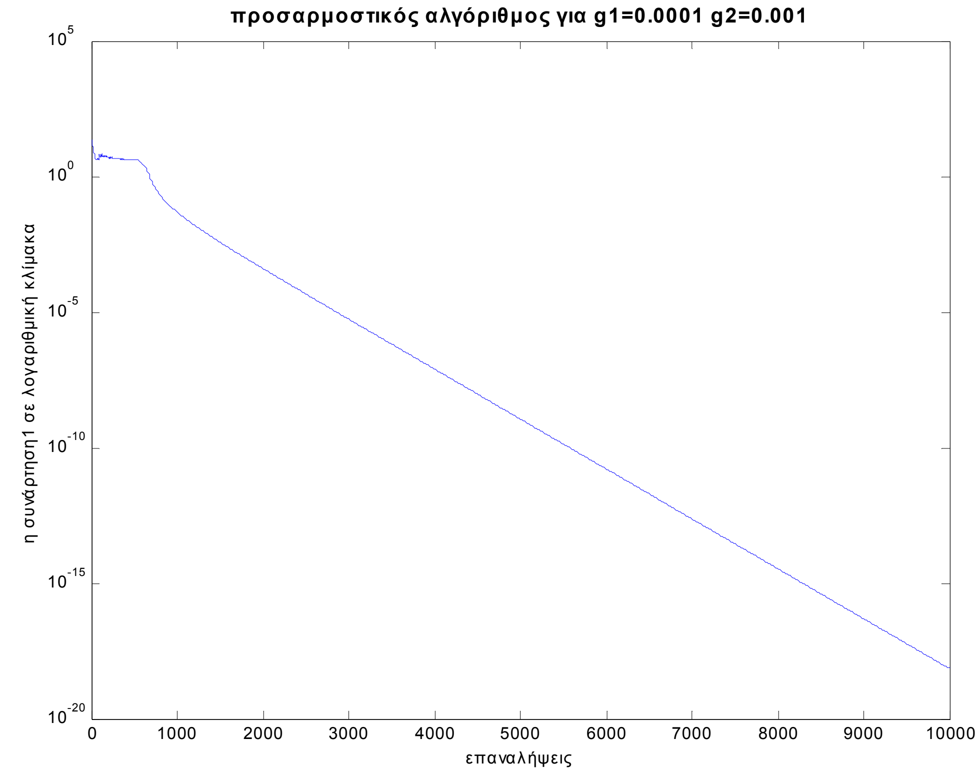


r = 19
Fmin = 4.9304e-032
epanal = 838
x1b = 1.0000
x2b = 1.0000
thetabzitoymeno = (45.2522 , 12.2061)
x1 = 1.0000
x2 = 1.0000
thetab = (45.2522 , 12.2061)

γράφημα 24

Στο γράφημα 24 βλέπουμε ότι μετά την 838η επανάληψη ο αλγόριθμος δεν μπορεί να μπει στη «random διαδικασία» και τελικά δεν έχουμε μια καλύτερη εκτίμηση της άγνωστης παραμέτρου θ. Αυτό γίνεται γιατί η συνθήκη F-F_old > tolerence δεν ισχύει. Αν η συνθήκη αυτή μετατραπεί σε F-F_old >= tolerence όπου tolerence=0, ο αλγόριθμος σε κάθε επανάληψη θα βελτιώνει την εκτίμηση της θ. Από εδώ και μπρος αυτή θα είναι η συνθήκη που θα οδηγεί τον αλγόριθμο στη «random διαδικασία».

Για g1=0.0001 και g2 =0.001 για 50 επαναλήψεις έχουμε 37 επαναλήψεις με καλή εκτίμηση του θ, αλλά όχι τόσο καλή προσέγγιση στο ελάχιστο της f, 10 επαναλήψεις με καλή προσέγγιση του ελαχίστου της f , αλλά όχι τόσο καλή εκτίμηση του θ, 2 επαναλήψεις που ο αλγόριθμος δεν ανταποκρίθηκε καθόλου και 1 επανάληψη που ο αλγόριθμος είχε και καλή εκτίμηση του θ και καλή προσέγγιση του ελαχίστου της f. Το ποσοστό επιτυχίας για αυτές τις τιμές των παραμέτρων g1,g2 είναι 96% και για πολύ καλές προσεγγίσεις 22%. Στο γράφημα 25 παρουσιάζονται τα αποτελέσματα του αλγορίθμου για 10000 επαναλήψεις

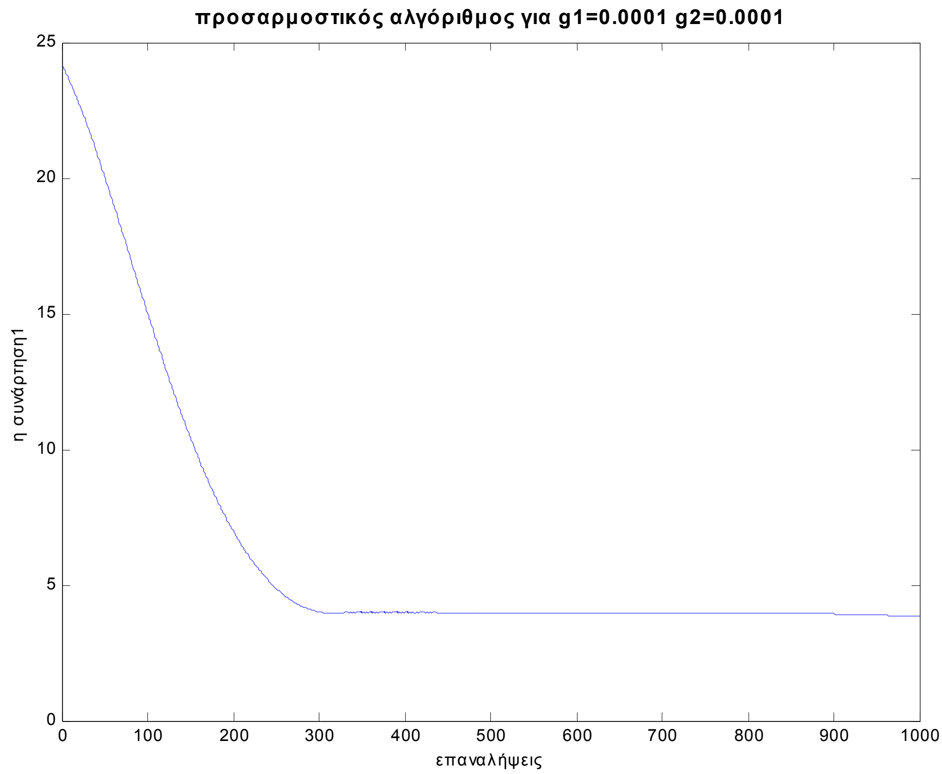


r = 164
Fmin = 7.3036e-019
epanal = 10000
x1b = 1.0000
x2b = 1.0000
thetabzitoymeno = (99.9655 , 4.4826)

γράφημα 25

παρατηρούμε ότι για g1=0.0001 και g2=0.001 σε κάθε επανάληψη του αλγορίθμου έχουμε βελτίωση και της προσέγγισης του ελαχίστου και της εκτίμησης της άγνωστης παραμέτρου. Αυτό γίνεται γιατί τα g1,g2 είναι μεγέθη σχεδόν της ίδιας τάξης. Για 10000 επαναλήψεις θα έχουμε σχεδόν πάντα παρόμοια αποτελέσματα. Υπενθυμίζουμε ότι το ελάχιστο της f είναι ίσο με 0 για x1=1 x2=1 και η άγνωστη παράμετρος θ είναι ίση με (100,1)

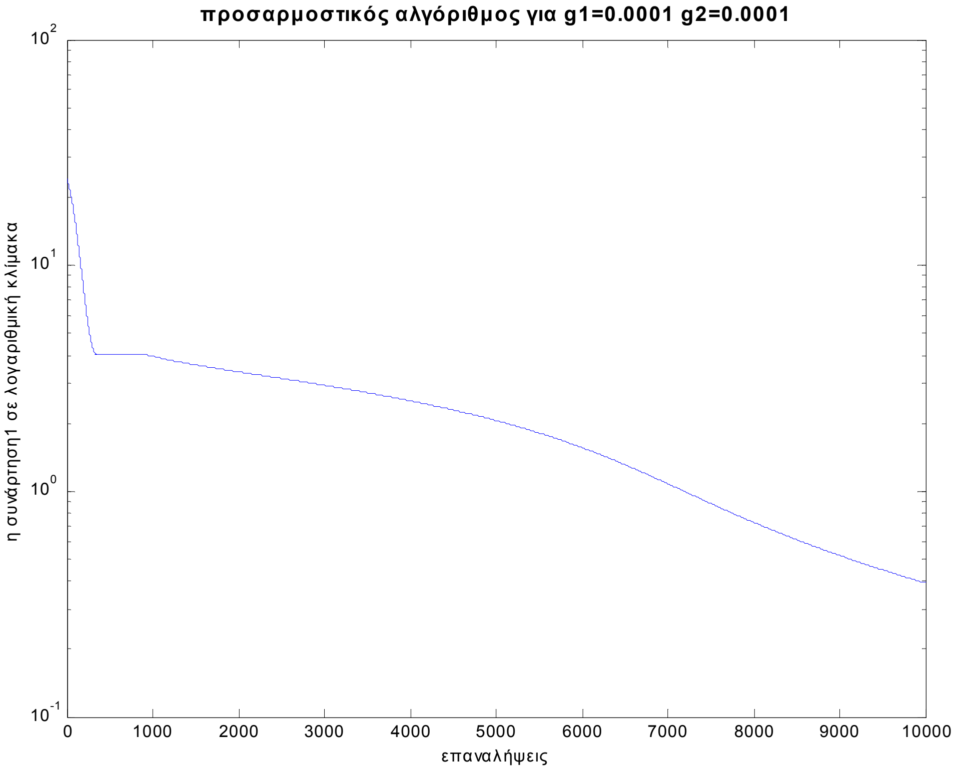
Για g1=0.0001 και g2=0.0001 ενώ ο αλγόριθμος κάνει πάντα πολύ καλή εκτίμηση της άγνωστης παραμέτρου θ δε μπορεί ποτέ να προσεγγίσει αρκετά το ελάχιστο της f. (π.χ. γράφημα 26)



r = 184
Fmin = 3.8536
epanal = 1000
x1b = -0.9546
x2b = 0.9294
thetabzitoymeno = (99.5550 , 2.2463)

γράφημα 26

Για 10000 χιλιάδες επαναλήψεις ο αλγόριθμος συνεχίζει να μας δίνει καλές εκτιμήσεις του θ, αλλά και βελτιωμένες προσεγγίσεις του ελαχίστου της f. (π.χ. γράφημα 27) Για περισσότερες επαναλήψεις θα έχουμε και πολύ καλά αποτελέσματα, αλλά κάτι τέτοιο θα είναι υπολογιστικά ασύμφορο.

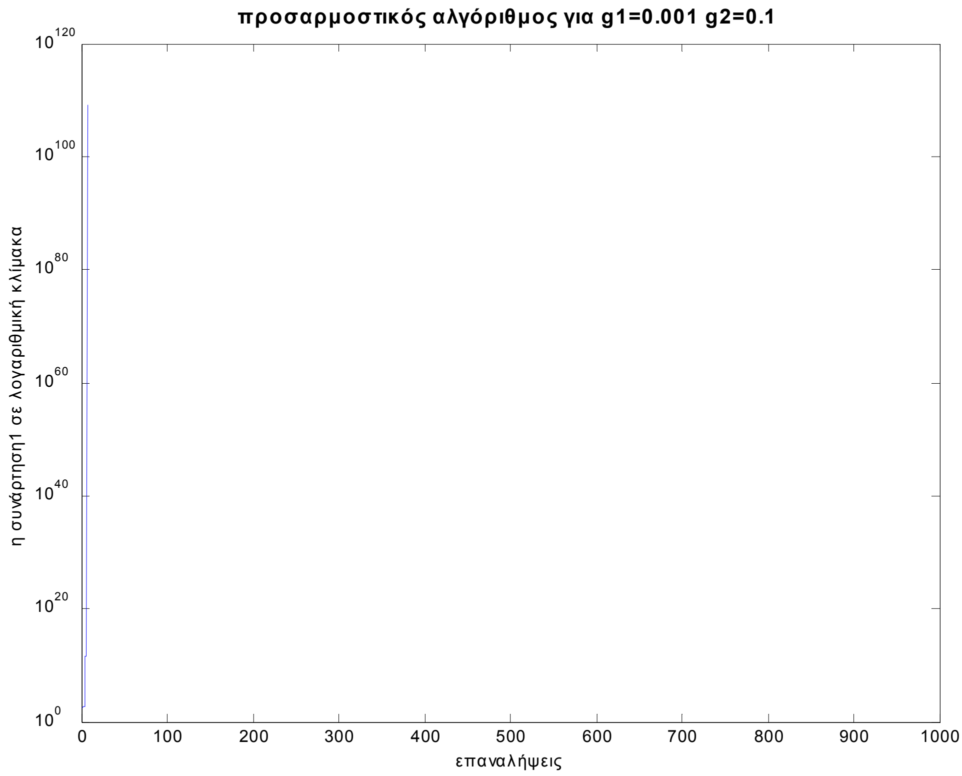


r = 202
Fmin = 0.3927
epanal = 10000
x1b = 0.3740
x2b = 0.1369
thetabzitoymeno = (99.5053 , 1.0042)

γράφημα 27

Εδώ βλέπουμε ότι και πάλι στις 1000 επαναλήψεις η προσέγγιση του ελαχίστου της f είναι της τάξης του (4) όπως και στο γράφημα 26. Μετά ακολουθεί μια πολύ σημαντική βελτίωση και στην 10000^{η} επανάληψη φτάνουμε στο fmin = 0.3927.

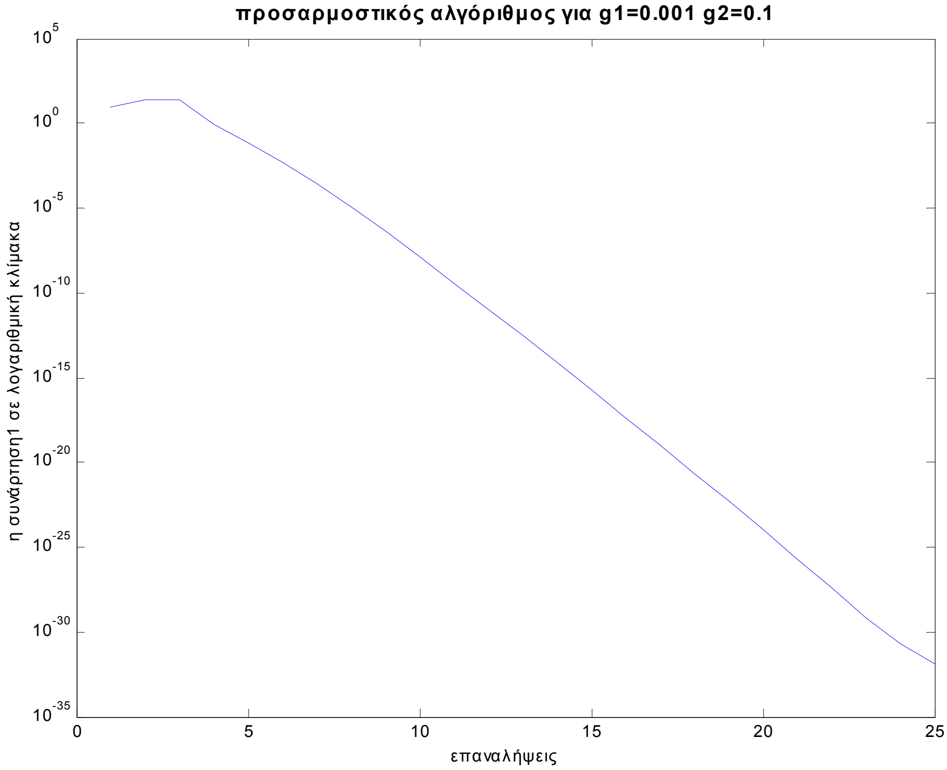
Για g1=0.001 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται στις 40 φορές που εκτελούμε το πρόγραμμα (π.χ. γράφημα 28) *(80% αποτυχία)* στις υπόλοιπες 10 έχουμε γραφήματα όπως το γράφημα 29.



r= 4
Fmin = 244.8347
epanal = 1
x1b = -1.5223
x2b = 0.7731
thetabzitoymeno = (-0.8612 , -0.3191)
x1 = NaN
x2 = NaN
thetab = (NaN , NaN)

γράφημα 28

Ο αλγόριθμος σε αυτή τη περίπτωση δεν ανταποκρίνεται καθόλου, αφού απειρίζονται τα x1,x2 από τη $2^{η}$ επανάληψη.



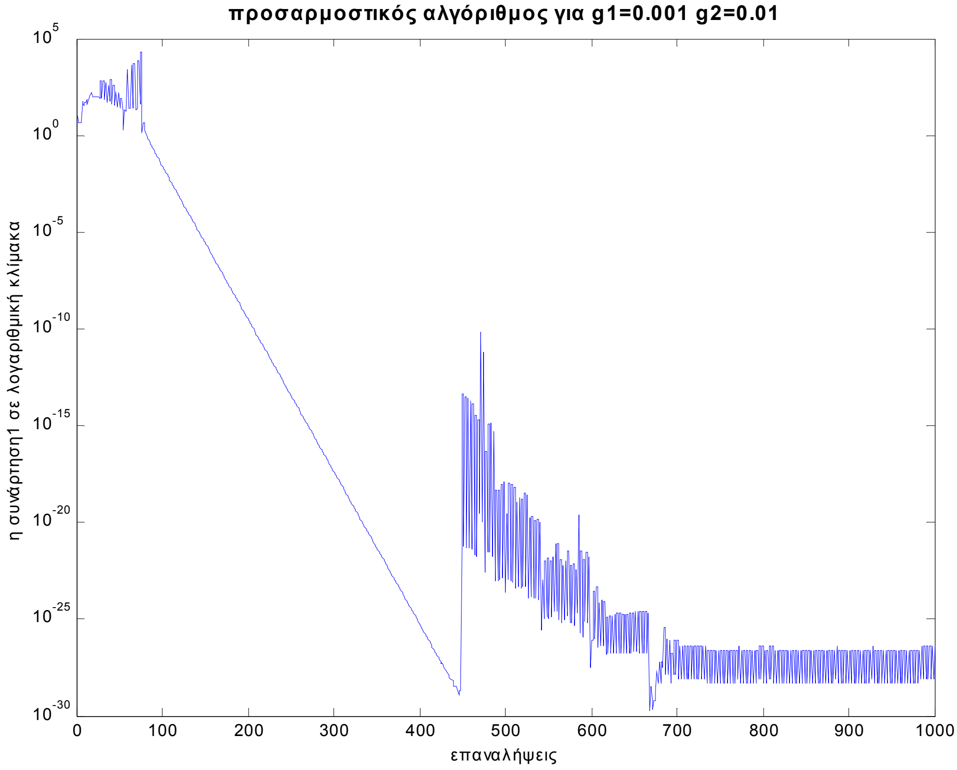
r = 326
Fmin = 0
epanal = 26
x1b = 1
x2b = 1
thetabzitoymeno = (5.0298 , 3.6085)
x1 = 1
x2 = 1
thetab = (100.0000 ,1.0000)

γράφημα 29

Στην $26^{η}$ επανάληψη έχουμε ταύτιση της εκτίμησης μας με το ελάχιστο της συνάρτησης, αλλά καθόλου καλή εκτίμηση της άγνωστης παραμέτρου θ. Στην $1000^{η}$ επανάληψη έχουμε πλήρη ταύτιση με την πραγματική τιμή $θ^{*}$.

Παρατηρούμε γενικά ότι για μεγάλα g2 έχουμε δυσκολία να προσεγγίσουμε το ελάχιστο της f, αλλά όταν αυτή η προσέγγιση γίνει εφικτή, είναι πολύ καλή και γίνεται σε πολύ λίγες επαναλήψεις.

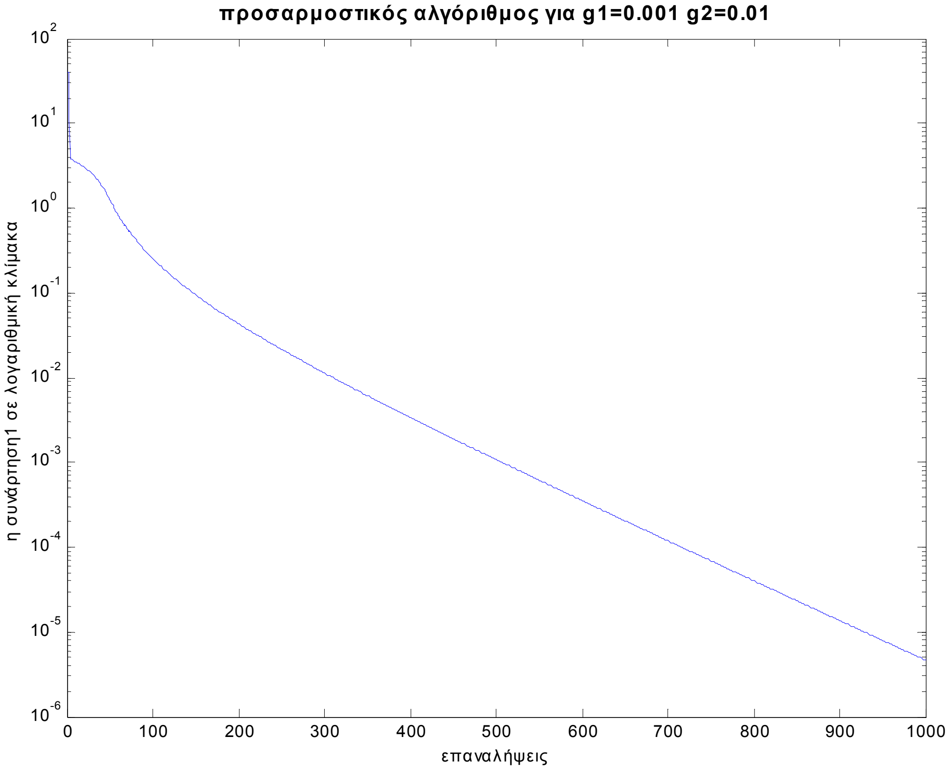
Για g1=0.001 και g2=0.01 δίνονται τα χαρακτηριστικά γραφήματα 30,31,32 και 33.



r = 210
Fmin = 1.8366e-030
epanal = 668
x1b = 1.0000
x2b = 1.0000
thetabzitoymeno = (4.3920, -14.3181)
x1 = 1.0000
x2 = 1.0000
thetab = (100.0002 , 1.0000)

γράφημα 30

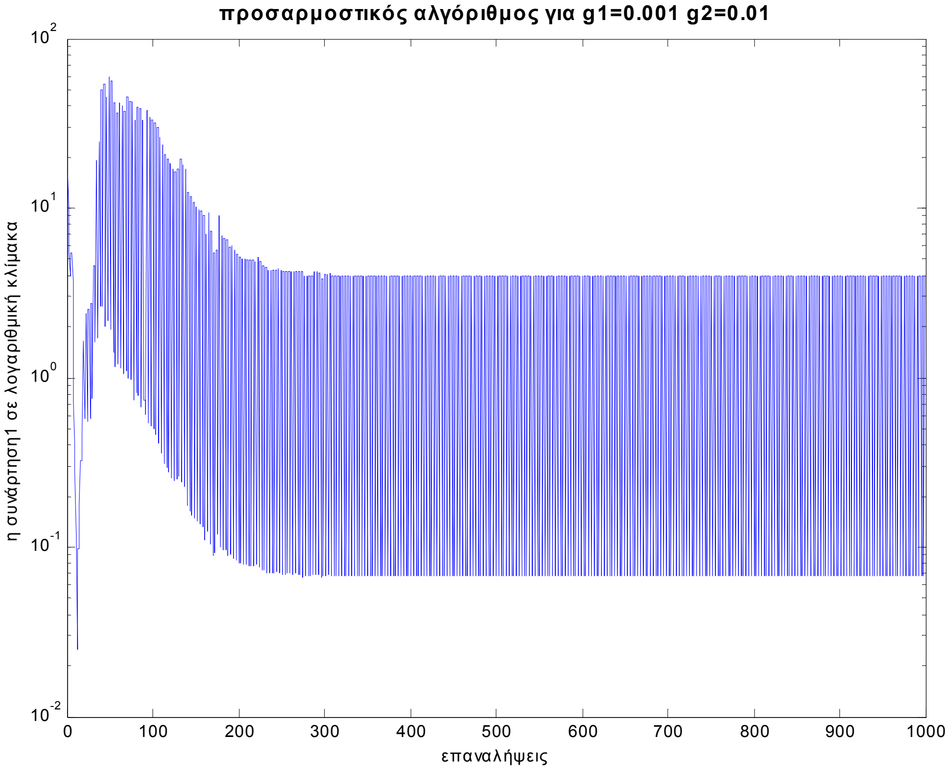
Εκτελέσαμε τον αλγόριθμο, και το 40% των εκτελέσεων αυτών κατέληξε σε γραφήματα σαν το γράφημα 30. Εδώ έχουμε σε λίγες επαναλήψεις καταπληκτική σύγκλιση στο ελάχιστο της f και ως τη χιλιοστή επανάληψη συνεχή βελτίωση της εκτίμησης της άγνωστης παραμέτρου έως ότου πετύχουμε την πραγματική της τιμή $\theta^*=(100\ ,\ 1)$.



r = 1
Fmin = 4.6801e-006
epanal = 1000
x1b = 0.9979
x2b = 0.9958
thetabzitoymeno = (14.6242 , 1.0374)

γράφημα 31

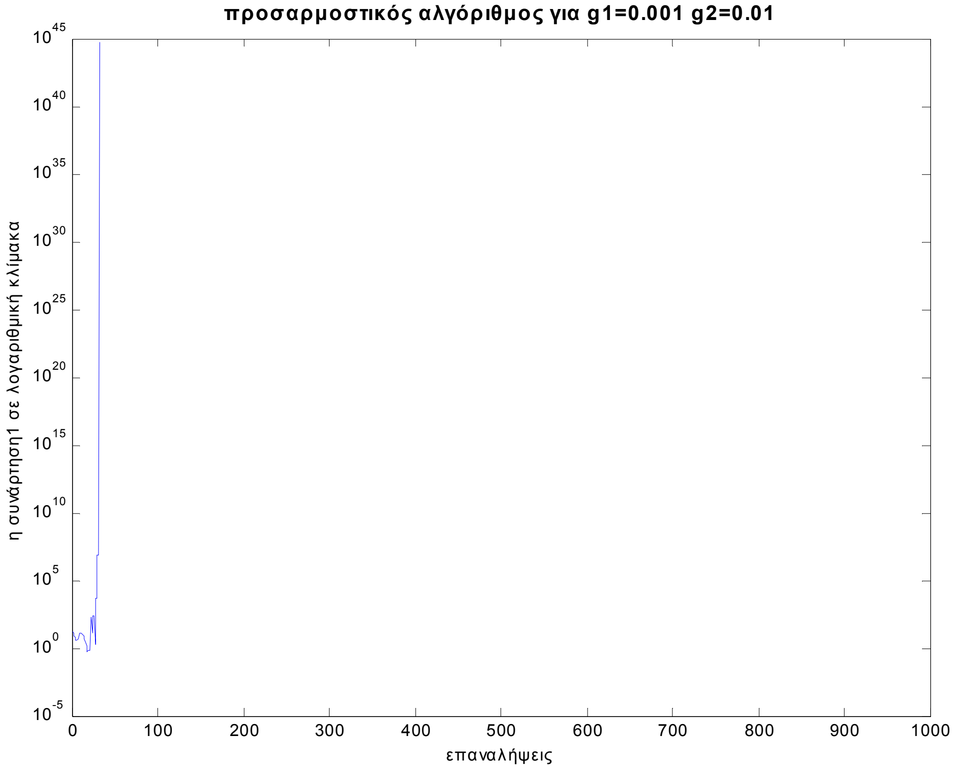
Σε αυτή την περίπτωση, που αντιστοιχεί στο 12% των εκτελέσεων του αλγορίθμου, έχουμε συνεχή βελτίωση της προσέγγισης στο ελάχιστο της f και καθόλου καλή εκτίμηση της θ^*. Αν οι επαναλήψεις ήταν περισσότερες θα καταφέρναμε να φτάσουμε σε βέλτιστα αποτελέσματα.



r = 331
Fmin = 0.0248
epanal = 12
x1b = 1.1567
x2b = 1.3365
thetabzitoymeno = (58.3140 , 25.0967)
x1 = 1.1567
x2 = 1.3365
thetab = (100.0000 , 1.0000)

γράφημα 32

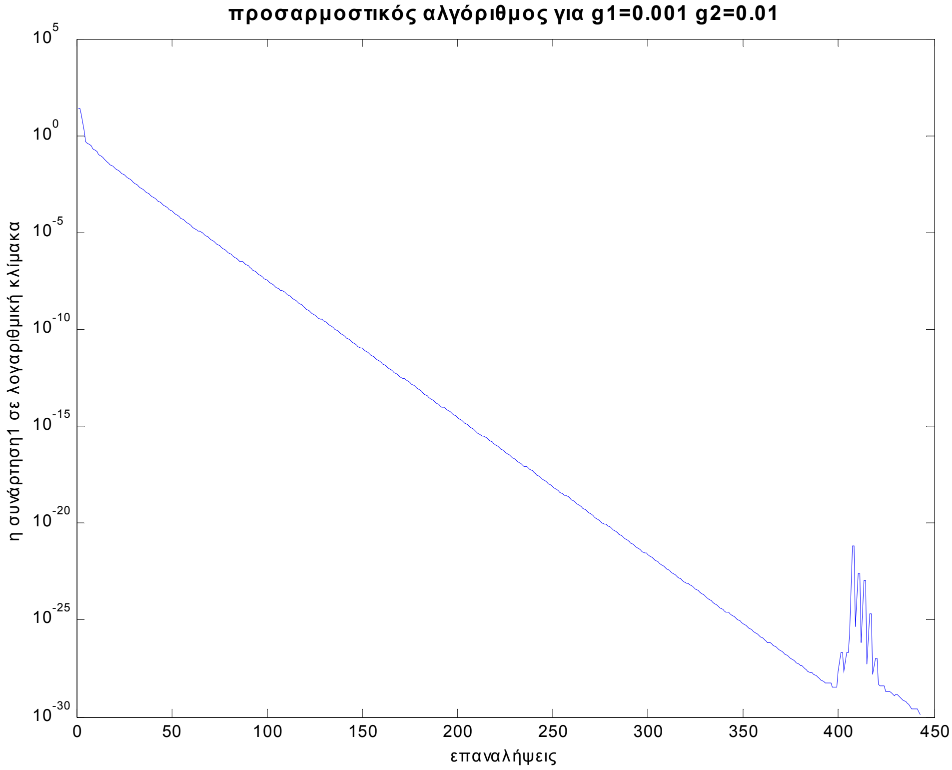
Σε αυτή την περίπτωση, που αντιστοιχεί στο 8% των εκτελέσεων του αλγορίθμου, έχουμε μια όχι πολύ καλή προσέγγιση στο ελάχιστο της f, αλλά πολύ καλή εκτίμηση του θ^*. Όσες επαναλήψεις και αν κάναμε ο αλγόριθμος δεν θα έδινε βελτιωμένα αποτελέσματα.



r = 8
Fmin = 0.4875
epanal = 19
x1b = 0.4027
x2b = 0.1260
thetabzitoymeno = (14.1260 , 6.5846)
x1 = NaN
x2 = NaN
thetab = (NaN , NaN)

γράφημα 33

Σε αυτή την περίπτωση, που αντιστοιχεί στο 40% των εκτελέσεων του αλγορίθμου, δεν είχαμε καλή συμπεριφορά. Από τις πρώτες επαναλήψεις τα x1,x2 μηδενίζονταν και ο προσαρμοστικός αλγόριθμος δεν μπορούσε να δώσει ικανοποιητικά αποτελέσματα.

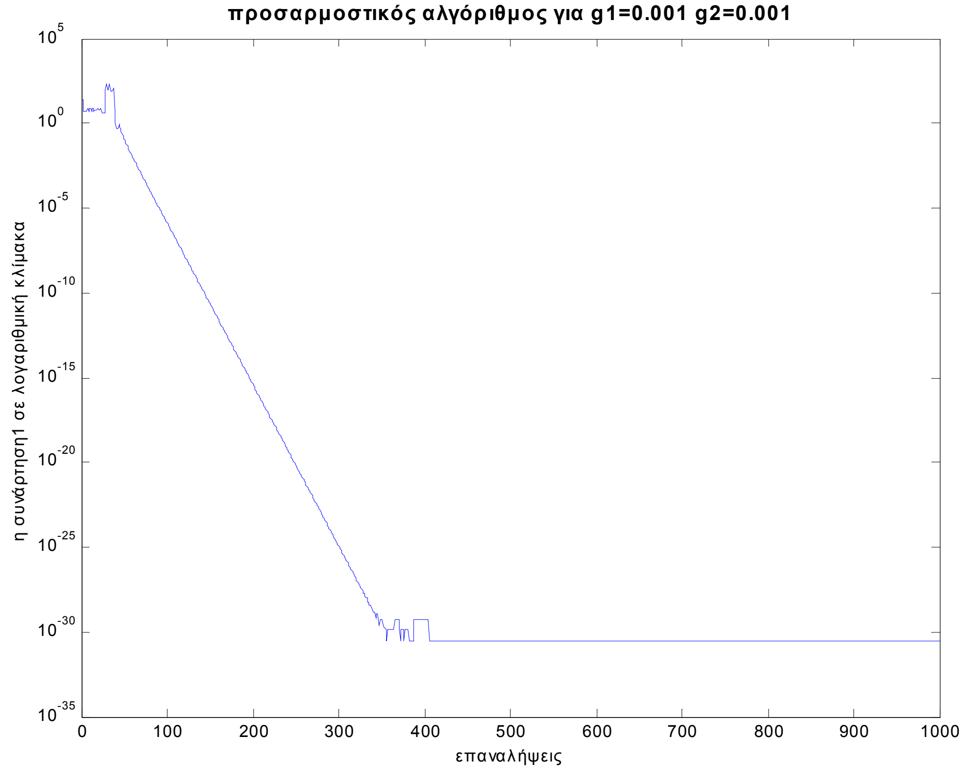


r = 198
Fmin = 0
epanal = 444
x1b = 1
x2b = 1
thetabzitoymeno = (56.6632 , 10.8039)
x1 = 1
x2 = 1
thetab = (100.0000 , 0.9999)

γράφημα 34

Το γράφημα 34 αντιστοιχεί σε μία σχεδόν βέλτιστη εφαρμογή του αλγορίθμου. Στην 444^{η} επανάληψη προσεγγίζουμε ακριβώς το ελάχιστο της f (fmin=0) και στη συνέχεια ο αλγόριθμος απλά βελτιώνει την εκτίμηση του θ^* έως ότου την πετυχαίνει ακριβώς.

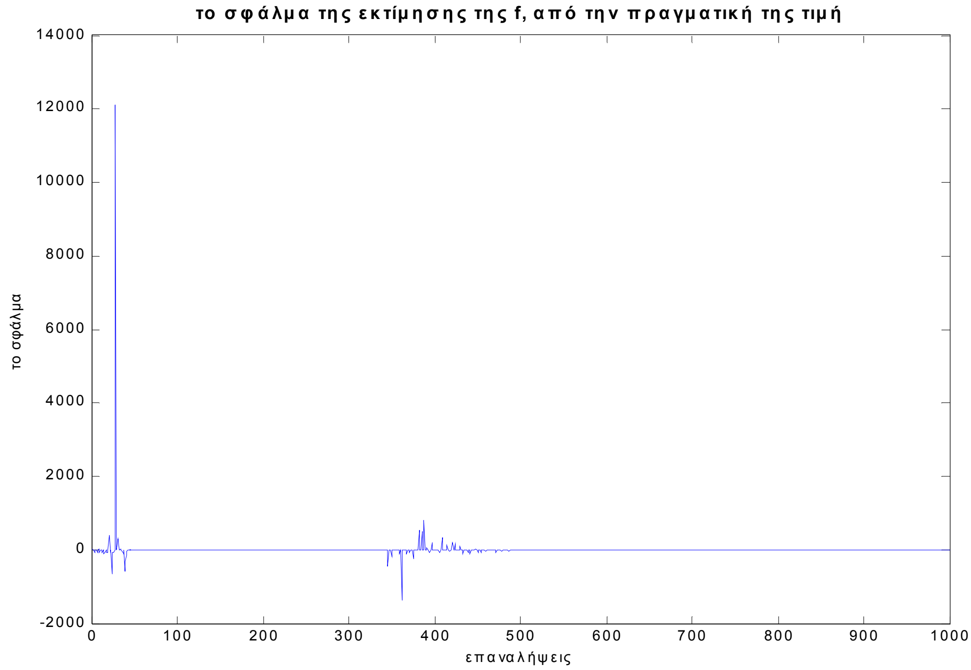
Για g1=0.001 και g2=0.001 δίνονται τα χαρακτηριστικά γραφήματα 35 και 36.

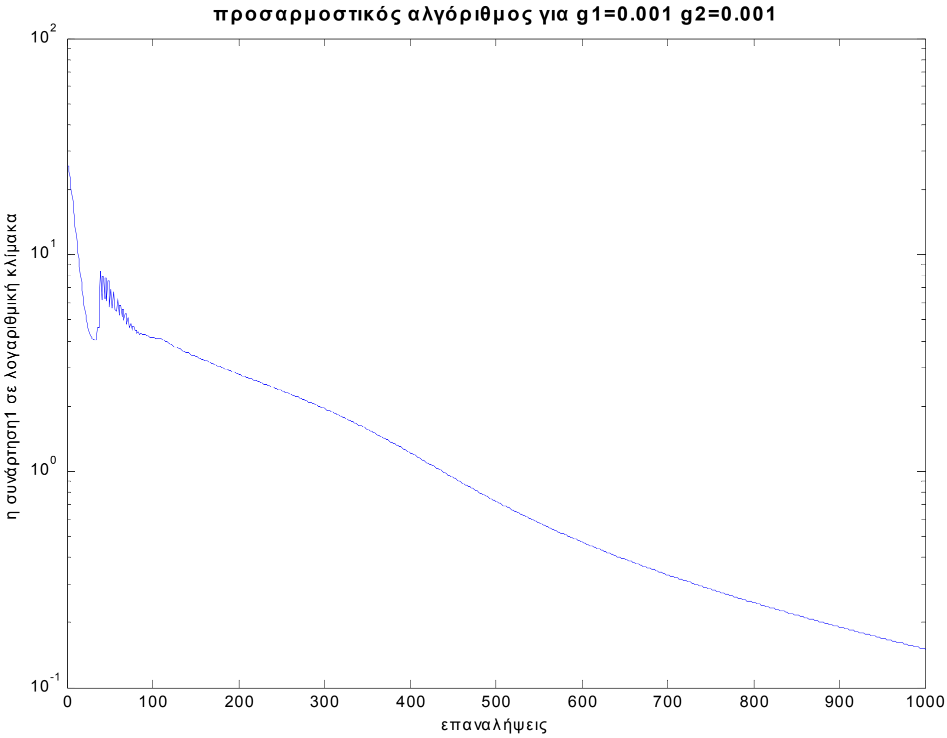


r = 230
Fmin = 3.0815e-031
epanal = 356
x1b = 1.0000
x2b = 1.0000
thetabzitoymeno = (254.9461 , 144.4145)
x1 = 1.0000
x2 = 1.0000
thetab = (100.0000 , 1.0000)

γράφημα 35

Σε αυτή την περίπτωση, που αντιστοιχεί στο 12% των εκτελέσεων του αλγορίθμου, έχουμε και καλή προσέγγιση στην πραγματική τιμή του fmin και καλή εκτίμηση του θ^*. Παρακάτω ακολουθεί το γράφημα του σφάλματος της εκτίμησης της τιμής της συνάρτησης σε σχέση με την πραγματική της τιμή.





r = 27
Fmin = 0.1509
epanal = 1000
x1b = 0.6122
x2b = 0.3726
thetabzitoymeno = (98.8188 , 1.2419)

γράφημα 36

Σε αυτή την περίπτωση, που αντιστοιχεί στο 28% των εκτελέσεων του αλγορίθμου, έχουμε καλή εκτίμηση της άγνωστης παραμέτρου θ, αλλά όχι τόσο καλή προσέγγιση στο fmin. Στο 48% των εκτελέσεων είχαμε πολύ καλό fmin, αλλά όχι καλό θ. Στο 4% όχι τόσο καλές εκτιμήσεις σε κανένα από τα 2. Σε όλες αυτές τις περιπτώσεις μεγαλύτερος αριθμός επαναλήψεων θα έδινε βέλτιστα αποτελέσματα. Στο 8% των εκτελέσεων, ο αλγόριθμος δεν ανταποκρίθηκε καθόλου.

Για g1=0.01 και g2=0.01 στο 90% των εκτελέσεων του αλγορίθμου δεν είχαμε ανταπόκριση ενώ για το 10% είχαμε καλές προσεγγίσεις στο fmin αλλά ποτέ δεν καταφέραμε να εκτιμήσουμε σωστά το θ^* παρά τον αριθμό των επαναλήψεων.

Για g1=0.01 και g2=0.001 στο 42% των εκτελέσεων δεν είχαμε ανταπόκριση. Στο 6% είχαμε καλά fmin, αλλά πολύ άσημες εκτιμήσεις του θ^*. Στο 52% των εκτελέσεων είχαμε fmin της τάξης του 0.3, δηλαδή όχι πολύ καλές, αλλά πολύ καλά θ. Σε αυτό το 52% αύξηση των επαναλήψεων του αλγορίθμου από 1000 σε 10000 θα έδινε βέλτιστα αποτελέσματα για το fmin.

Τα συγκριτικά αποτελέσματα παρατίθενται στον πίνακα 4

Συγκριτικός πίνακας					
g1-g2	$fmin^*$ για 1000 επαναλήψεις	θ^* για 1000 επαναλήψεις	Ασταθείς λύσεις για 1000 επαναλήψεις	$fmin^*$& θ^* για 1000 επαναλήψεις	$fmin^*$& θ^* για 10000 επαναλήψεις
0.0001-0.01	78%	18%	4%	0%	96%
0.0001-0.001	20%	74%	6%	0%	94%
0.001-0.01	12%	8%	40%	40%	60%
0.001-0.001	48%	28%	4%	12%	96%
0.01-0.01	10%	0%	90%	0%	0%
0.01-0.001	6%	52%	42%	0%	52%

Πίνακας 4

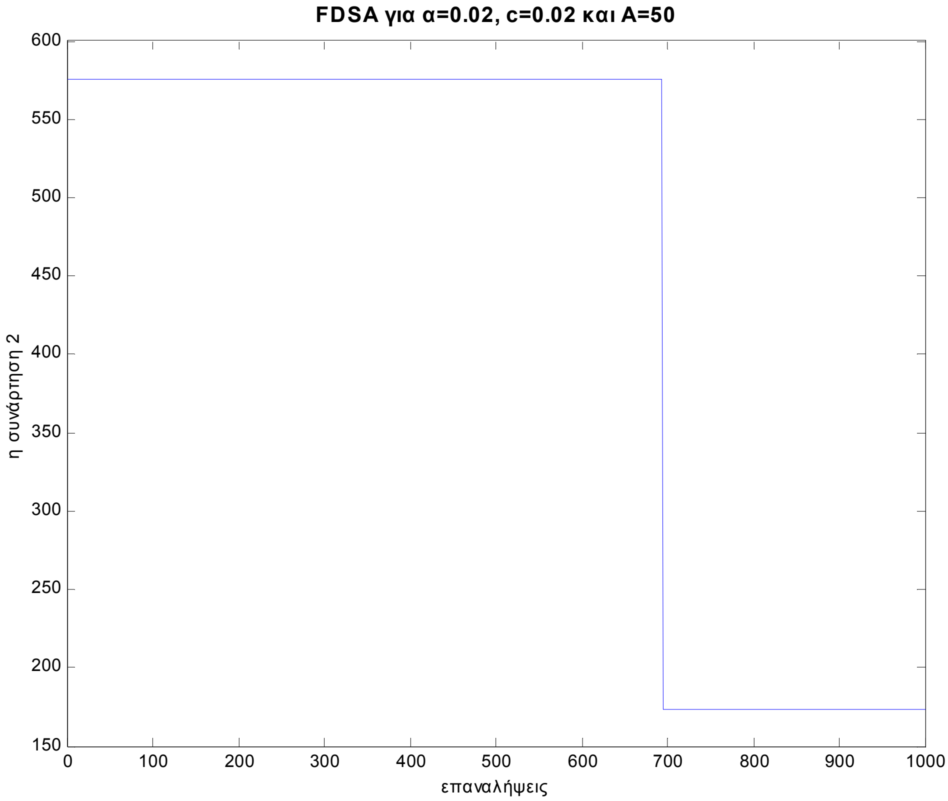
Βλέπουμε ότι για 1000 επαναλήψεις μόνο για (g1,g2) = (0.001,0.01) και (0.001,0.001) φτάνουμε σε βέλτιστες λύσεις. Για την πρώτη περίπτωση έχουμε ποσοστό 40% αλλά και 40% αποτυχία ενώ για τη δεύτερη 12% και 4% αποτυχία. Επιλέγουμε λοιπόν ως βέλτιστη τιμή των παραμέτρων g1,g2 το (0.001,0.001).

5.2 Συνάρτηση 2 (Freudenstein and Roth function)

Υπενθυμίζουμε ότι η δεύτερη συνάρτηση παρουσιάζει απόλυτο ελάχιστο ($fmin^*$=0) στο (x1,x2)=(5,1) και τοπικό ελάχιστο (fmin=48.9842…) στο (x1 , x2)= (11.41... , -0.8968). Η αρχική τιμή των μεταβλητών θα είναι x_0 = (0.5 , -2)

5.2.1 Μέθοδος FDSA

Παρατηρούμε πάλι ότι ανεξαρτήτως αλλαγής των συντελεστών, ο αλγόριθμος δεν βελτιώνει καθόλου την τιμή της συνάρτησης. Η τιμή της συνάρτησης για την αρχική τιμή x_0 =(0.5,-2) είναι 400.5 και τόσο παραμένει σε όλες τις επαναλήψεις. Αντίστοιχα το x=(0.5,-2) παραμένει ως έχει. Αλλάζοντας και πάλι την αρχική τιμή, βρίσκουμε ότι για τις περισσότερες δεν αλλάζει τίποτα στα αποτελέσματα του αλγορίθμου. Για αρχική τιμή x_0 =(5,-2) στην 695^η επανάληψη η συνάρτηση παίρνει την τιμή 173.6607 αντί 576 με καλύτερο x(theta) x=(-5.1074,-1.6987). (γράφημα 37)



γράφημα 37

5.2.2 Μέθοδος SPSA

Παρατηρούμε κάνοντας αρκετές προσομοιώσεις του αλγορίθμου ότι για τιμές του α ίσες με 0.8 ή μεγαλύτερες, ο αλγόριθμος δεν λειτουργεί. Όταν οι τιμές του α κυμαίνονται στο διάστημα (0.2,0.7) υπάρχει μια σταθερότητα στη σύγκλιση. Η εκτίμηση της συνάρτησης παίρνει τιμές ανάμεσα στο διάστημα (105,106). Μοναδικό μας κριτήριο για την επιλογή του συντελεστή α σε αυτή τη περιοχή είναι η ταχύτητα σύγκλισης. Για α=0.2 χρειαζόμαστε από 70 έως 80 επαναλήψεις για να συγκλίνει ο αλγόριθμος στην τελική τιμή, ενώ για α=0.7 χρειαζόμαστε από 915 έως 925 επαναλήψεις. Πιο συγκεκριμένα τα αποτελέσματα παρατίθενται στον πίνακα 5

A	c	α	επαναλήψεις	σύγκλιση
50	Αδιάφορο	0.02	70-80	105-106
50	»»	0.03	185-195	105-106
50	»»	0.04	330-340	105-106
50	»»	0.05	500-515	105-106
50	»»	0.06	695-705	105-106
50	»»	0.07	915-925	105-106

Πίνακας 5

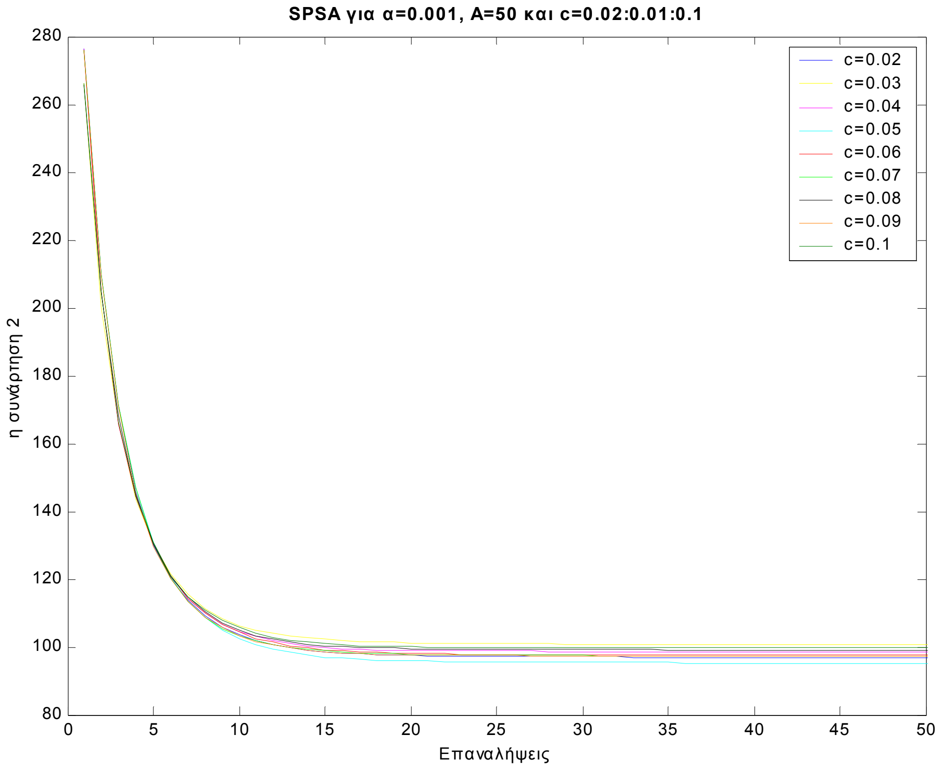
Προχωρώντας σε μικρότερες τιμές του α παίρνουμε τα αποτελέσματα του πίνακα 6 που ακολουθεί.

A	c	α	επαναλήψεις	σύγκλιση
50	Αδιάφορο	0.01	2-5	101-105*
50	»»	0.005	1	93.95… ή 112.45….
50	»»	0.001	30-45	96-102
50	»»	0.0005	62-76	97.5-101.2

Πίνακας 6

(*) για α=0.01 παρατηρούνται (σχετικά σπάνια) τιμές της $f\left(\hat{\theta}\right)$ της τάξεως του 393-394. Για α=0.005 δεν μπορούμε να εμπιστευτούμε την μία και μοναδική επανάληψη του αλγορίθμου. Για α=0.001 και α=0.0005 πετυχαίνουμε πολύ καλύτερες προσεγγίσεις και με λιγότερες επαναλήψεις. Προτιμότερη τιμή για τη μεταβλητή α είναι η (α=0.001). Οι προηγούμενοι πίνακες προέκυψαν από μεγάλο αριθμό εκτελέσεων του αλγορίθμου και δεν αφήνουν περιθώριο αμφιβολίας για τα εξαγόμενα συμπεράσματα. Το c που χρησιμοποιήθηκε ήταν το c=0.04 και θεωρήθηκε αδιάφορο γιατί με μια πρώτη ματιά δεν φάνηκε να επηρεάζει τα αποτελέσματα. Στη συνέχεια πάντως θα γίνει περαιτέρω διερεύνηση και για το c και για το A, που αυθαίρετα επιλέχθηκε ίσο με 50 (A=50), ίσο δηλαδή με τη μικρότερη δυνατή τιμή.

Για διάφορες τιμές του c, α=0.001 και A=50 δημιουργούμε το γράφημα 38



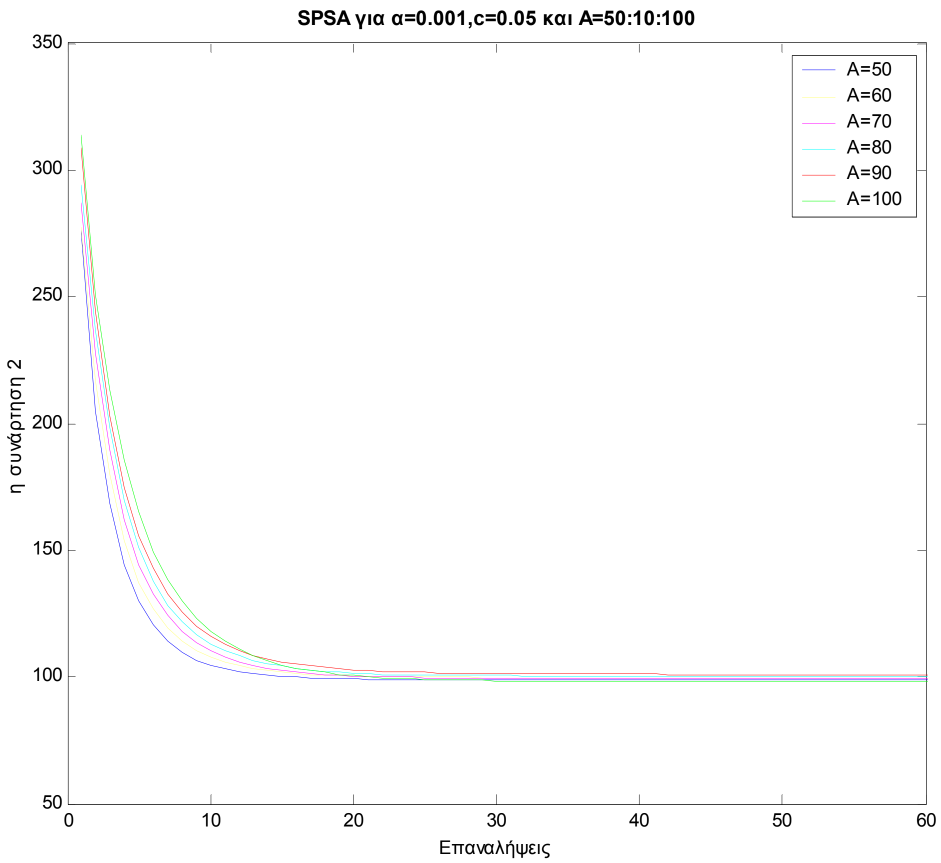
γράφημα 38

από το γράφημα βλέπουμε ότι για c=0.05 η τιμή της προσέγγισης της f ξεφεύγει από τα όρια που έχουμε θέσει. Η $f(\hat{\theta})$ παίρνει την τιμή 95.3167 στην 36^{η} επανάληψη. Για να επαληθεύσουμε την παραπάνω παρατήρηση εκτελέσαμε το πρόγραμμα αρκετές ακόμα φορές. Κάποιες χαρακτηριστικές τιμές δίνονται στον παρακάτω πίνακα (πίνακας 7)

Π1	Π2	Π3	Π4	Π5	Π6	Π7	Π8	Π9	Π10
98.0183	99.6896	101.3886	96.4169	100.8675	101.3513	95.9845	**95.3329***	97.6004	96.2957

Πίνακας 7

Παρατηρούμε ότι για c=0.05 ξαναβρήκαμε τιμή μικρότερη του 96, πράγμα που δεν συνέβη για καμιά άλλη τιμή του c, παρά τον μεγάλο αριθμό επαναλήψεων του αλγορίθμου. Έτσι λοιπόν καταλήγουμε στις παραμέτρους α=0.001 και c=0.05 και προσπαθούμε να διερευνήσουμε στο πώς επηρεάζει η παράμετρος Α τον αλγόριθμο.



γράφημα 39

Από το γράφημα 39 παρατηρούμε ότι για A=50 έχουμε τη γρηγορότερη σύγκλιση, ενώ για A=100 έχουμε την πιο αργή αλλά και τη πιο καλή προσέγγιση στην επιθυμητή τιμή. Δυστυχώς τα συμπεράσματα αυτά δεν είναι αξιόπιστα αφού όλες οι τιμές (και οι προσεγγίσεις τις f, και ο απαιτούμενος αριθμός επαναλήψεων) βρίσκονται στα όρια που τέθηκαν στον πίνακα 6.

Προχωράμε λοιπόν στον πίνακα 8 για σταθερές τιμές του α,c και διάφορες τιμές του Α. Στον πίνακα περιέχονται κάποιες από τις πιο χαρακτηριστικές τιμές της f καθώς και των απαραίτητων επαναλήψεων, από ένα μεγάλο νούμερο προσομοιώσεων του αλγορίθμου.

A=50		A=60		A=70	
$f\left(\hat{\theta}\right)$	επαναλήψεις	$f\left(\hat{\theta}\right)$	επαναλήψεις	$f\left(\hat{\theta}\right)$	επαναλήψεις
95.0179	39	100.2135	39	97.7069	39
98.5324	32	100.7599	38	102.0123	40
96.5938	40	99.2418	38	97.6903	38
100.6622	31	100.7135	49	101.9934	37
98.4515	31	101.9052	35	100.0824	35
102.0105	29	100.7758	37	101.2776	42
101.3319	36	96.6341	39	99.2665	37
96.1377	44	98.5873	40	100.2832	43
99.6072	37	96.9710	37	99.1729	43
97.9246	36	99.6376	36	96.7211	41
96.4726	34	96.4726	35	99.6626	51
A=80		A=90		A=100	
101.0187	47	100.5367	45	100.2520	46
99.1311	47	98.5668	49	98.3363	43
98.7630	40	99.3832	41	100.6177	50
99.5689	47	100.8189	48	99.2313	51
100.8082	45	101.7029	43	97.9718	44
99.6887	41	101.3402	42	100.5563	55
97.3289	43	99.3642	47	100.7015	48
99.3793	39	99.2856	51	99.2738	46
100.1697	44	99.4298	40	99.8569	48
99.1697	41	98.1732	46	96.5422	51
102.1681	46	97.1265	46	99.8359	49
96.5588	40	96.3002	54	101.3847	52

Πίνακας 8

Συμπεραίνουμε λοιπόν από τον πίνακα 8 ότι για A=50 πράγματι έχουμε τις γρηγορότερες συγκλίσεις (μικρότερο αριθμό επαναλήψεων του αλγορίθμου) και καμιά διαφορά στη τελική σύγκλιση (minimum $f\left(\hat{\theta}\right)$), εκτός του ότι για και μόνο για Α=50 (η μικρότερη δυνατή τιμή για το Α) παρουσιάζονται και κάποιες φορές τιμές της $f\left(\hat{\theta}\right)$ μικρότερες του 96. Άρα και στα δύο μας κριτήρια η τιμή της παραμέτρου Α (Α=50) είναι η βέλτιστη. Για αυτή τη συνάρτηση λοιπόν οι προτιμότερες τιμές των παραμέτρων είναι: α=0.001, c=0.05 και Α=50.

5.2.3 Προσαρμοστικός αλγόριθμος

Εδώ θα εξετάσουμε τον προσαρμοστικό αλγόριθμο για τη δεύτερη συνάρτηση. Θα προσπαθήσουμε να διερευνήσουμε την βέλτιστη τιμή των παραμέτρων g1,g2, δηλαδή την τιμή των παραμέτρων για την οποία ο αλγόριθμος έχει την καλύτερη συμπεριφορά ως προς την εκτίμηση της τιμής της άγνωστης συνάρτησης και την εκτίμηση της άγνωστης παραμέτρου θ.

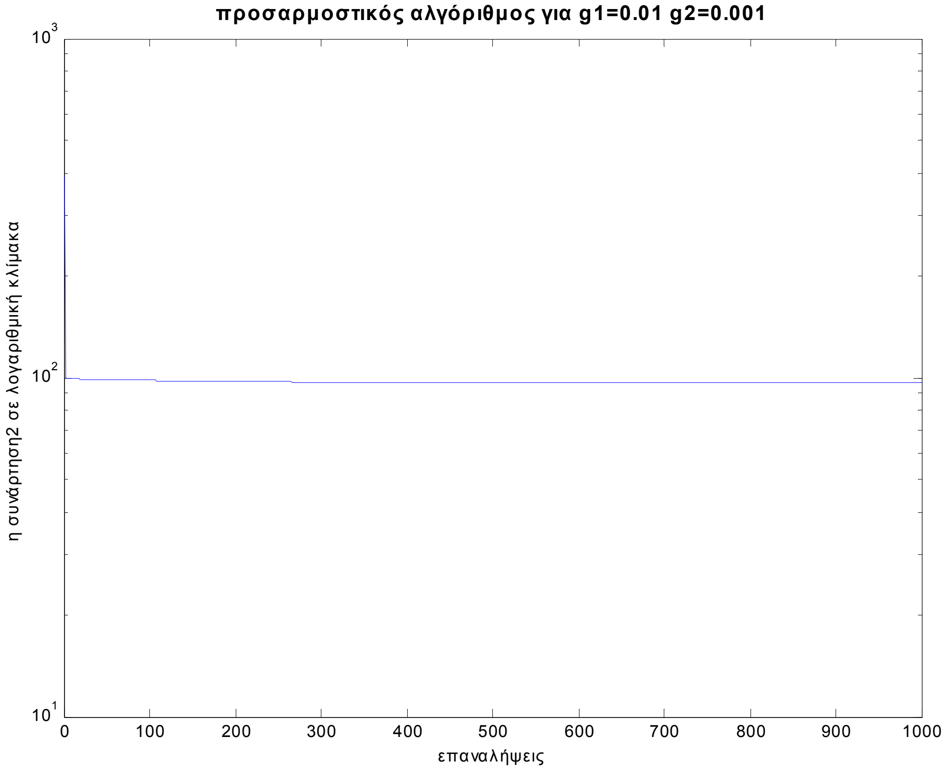
Για g1=0.1 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.1 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

Για g1=0.1 και g2=0.001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει:
fmin = 96.2407 και πολύ άσχημες εκτιμήσεις του θ^*. Υπενθυμίζουμε ότι $\theta^*=(1,1)$ και ότι fmin=0.

Για g1=0.1 και g2=0.0001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει:
fmin = 98.2937 και πολύ άσχημες εκτιμήσεις του θ^*.

Για g1=0.01 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.01 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

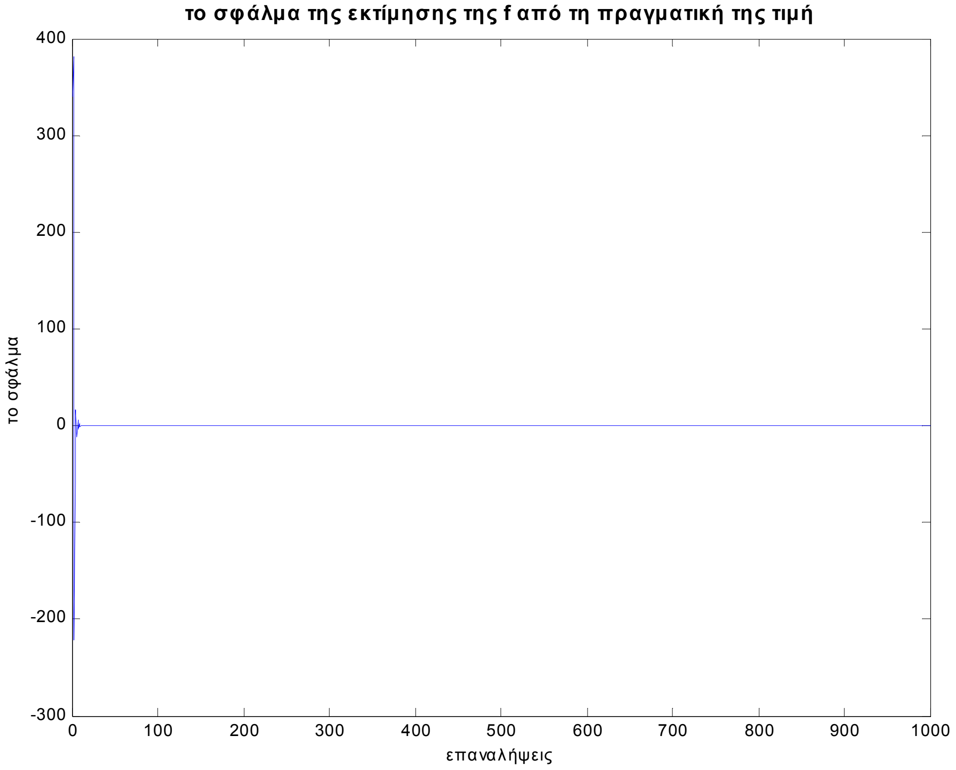
Για g1=0.01 και g2=0.001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει:
fmin = 96.2407 και θ αρκετά καλό (π.χ. γράφημα 40).



Fmin = 96.2407
epanal = 1000
x1b = 0.8163
x2b = -1.4448
thetabzitoymeno = (0.6625 , 1.0739)

γράφημα 40

Παρακάτω παρατίθεται το γράφημα του σφάλματος σε αυτή τη περίπτωση. (γράφημα 41)



γράφημα 41

Για g1=0.01 και g2=0.0001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει: fmin = 98.2937 και αρκετά καλές εκτιμήσεις του θ^*.

Για g1=0.001 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.001 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

Για g1=0.001 και g2=0.001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει: fmin = 96.2407 και αρκετά καλές εκτιμήσεις του θ^*.

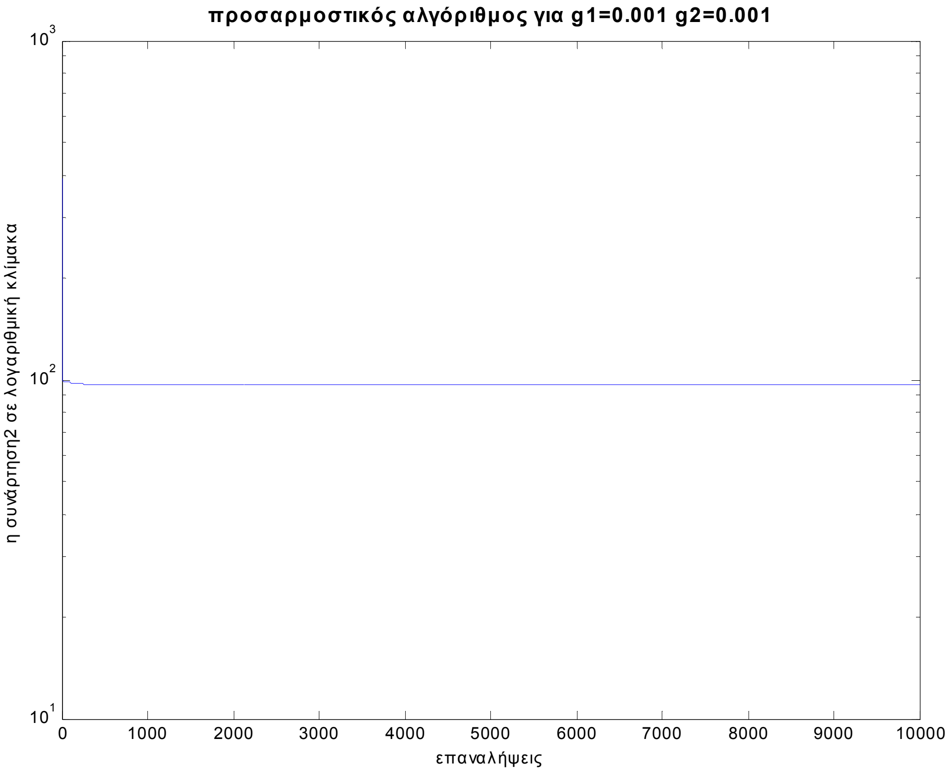
Για g1=0.001 και g2=0.0001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει: fmin = 98.2937 και αρκετά καλές εκτιμήσεις του θ^*.

Για g1=0.0001 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.0001 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

Για g1=0.0001 και g2=0.001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει: fmin = 96.2407 και αρκετά καλές εκτιμήσεις του θ^*.

Για g1=0.0001 και g2=0.0001 ο αλγόριθμος στη χιλιοστή επανάληψη μας δίνει: fmin = 98.2937 και αρκετά καλές εκτιμήσεις του θ^*.

Συμπεραίνουμε ότι το βέλτιστο g2 είναι το (g2=0.001) και για g1 μπορούμε να διαλέξουμε οποιαδήποτε τιμή στο διάστημα (0.1 , 0.0001) το g1=0.1 απορρίπτεται γιατί δεν μας δίνει καλές εκτιμήσεις του θ^*. Για g2=0.001 ο αλγόριθμος σταματάει στο 96.2407 και για g2=0.0001 στο 98.2937. Αυξάνοντας τις επαναλήψεις πετυχαίνουμε κάποια μικρή βελτίωση και στις δύο περιπτώσεις. Αυτή η βελτίωση δεν είναι καθόλου ουσιαστική (γράφημα 42).



r = 1383
Fmin = 96.2089
epanal = 5964
x1b = 0.8197
x2b = -1.4446
thetabzitoymeno = (1.0000 , 1.0000)

γράφημα 42

Όπως βλέπουμε, για 10000 επαναλήψεις η βελτίωση της εκτίμησης του fmin είναι αμελητέα σε σχέση με την αύξηση του υπολογιστικού φόρτου, αλλά η εκτίμηση του θ συμπίπτει με την πραγματική του τιμή (θ^*).

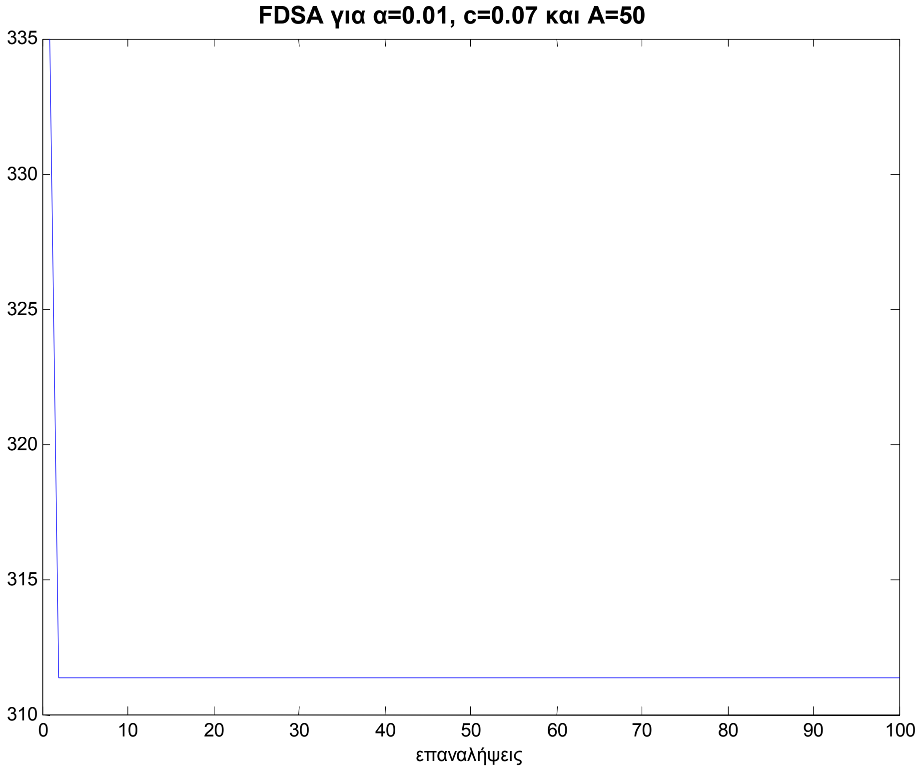
Για τη δεύτερη συνάρτηση επιλέγουμε (g1 , g2)=(0.001 , 0.001).

5.3 Συνάρτηση 3 (Extended Powell singular function)

Υπενθυμίζουμε ότι η τρίτη συνάρτηση παρουσιάζει ελάχιστο (fmin=0) στο xi= 0 όπου i=1,2,…,n και n είναι πολλαπλάσιο του 4. Εδώ θα μελετήσουμε δύο περιπτώσεις. α) n=m=4 και β) n=m=8 .

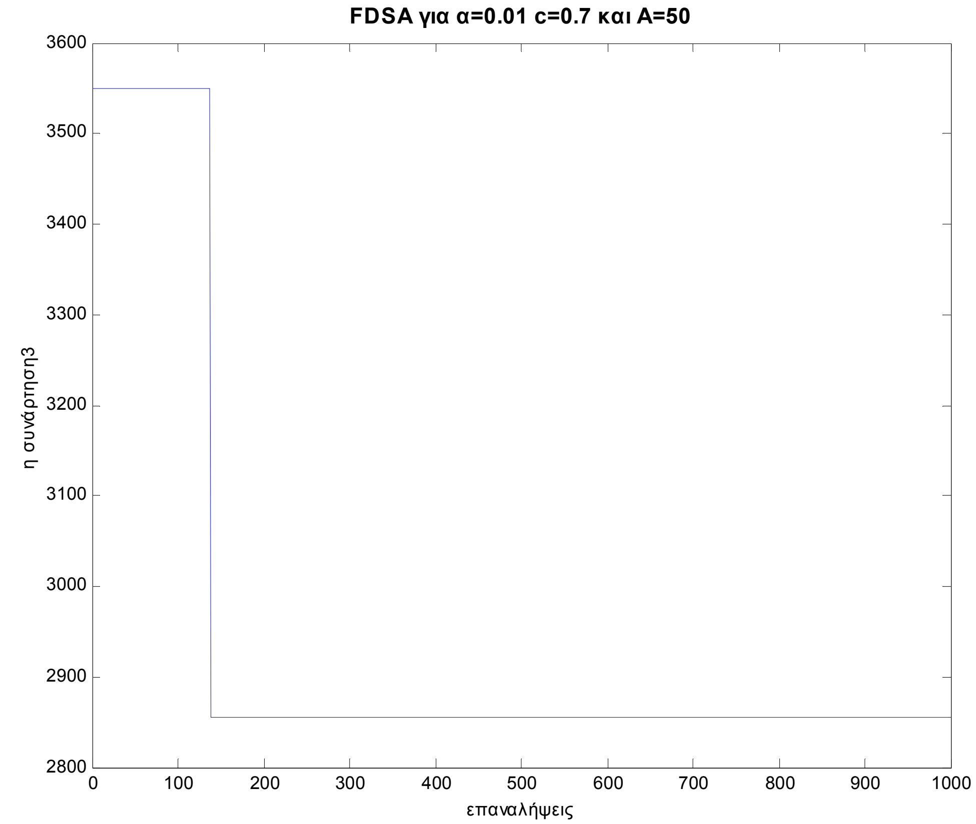
5.3.1 Μέθοδος FDSA

Για την τρίτη συνάρτηση παρατηρούμε ότι η χρήση της μεθόδου FDSA, έχει σχεδόν την ίδια συμπεριφορά με προηγουμένως. Ξεκινώντας με c και α να ανήκουν στο διάστημα (0.01,0.1) δεν καταφέρνουμε να βρούμε μια τιμή των παραμέτρων που να βελτιώνει την τιμή της $f_{(\theta)}$ (f=335). Από c≥0.4 αρχίζει ο αλγόριθμος να λειτουργεί και για c=0.7 α=0.01 και A=50 πετυχαίνουμε την καλύτερη τιμή για την f. (f=311.3729). Πάλι βλέπουμε ότι η απόκλιση από την ελάχιστη τιμή της συνάρτησης είναι πολύ μεγάλη και η βελτίωση της ελάχιστη ως μηδαμινή. (γράφημα 43)



γράφημα 43

Για τη τρίτη συνάρτηση πάλι, αλλά για n=m=8, δηλαδή για 8 μεταβλητές και 8 όρους στην αντικειμενική συνάρτηση, ο αλγόριθμος συμπεριφέρεται με τον ίδιο ακριβώς τρόπο. Για την προεπιλεγμένη αρχική τιμή x_0=(3,-1,0,1,3,-1,0,3) έχουμε την συνάρτηση μας να παίρνει την τιμή 3550. Ο αλγόριθμος αρχίζει να λειτουργεί για τιμές του c μεγαλύτερες του 0.4 και παίρνουμε πάλι τη βέλτιστη τιμή για c=0.7, α=0.01 και A=50 (γράφημα 44).



γράφημα 44

5.3.2 Μέθοδος SPSA

Για τιμές του α μεγαλύτερες του 0.1 ο αλγόριθμος δεν λειτουργεί. Για τιμές του α μικρότερες του 0.02 ο αλγόριθμος δεν προσεγγίζει καθόλου καλά την τελική τιμή της $f(\theta)$. Θα κάνουμε λοιπόν διερεύνηση για την εύρεση του βέλτιστου α στο διάστημα (0.02 , 0.1). Λειτουργούμε τον αλγόριθμο για A=50 και c=0.04 αλλάζοντας το α και κάνοντας αρκετές επαναλήψεις. Καταλήγουμε με αυτή τη διαδικασία στον πίνακα 9

Στατιστικά αποτελέσματα αλγορίθμου SPSA για A=50 c=0.04 και α=0.02:0.01:0.1						
α	$\max f\left(\hat{\theta}\right)$	$\min f\left(\hat{\theta}\right)$	Συχνότερη Διακύμανση $f\left(\hat{\theta}\right)$	Min αριθμός επαναλήψεων	Max αριθμός επαναλήψεων	Συχνότερη Διακύμανση επαναλήψεων
0.02	2.1056	0.2001	0.4-1.0	31	170	90-140
0.03	0.3858	0.0153	0.08-0.12	54	195	90-160
0.04	0.3799	0.0209	0.13-0.18	149	233	190-220
0.05	0.4193	0.0197	0.20-0.27	243	400	280-320
0.06	0.4171	0.0175	0.14-0.20	321	498	390-440
0.07	0.3180	0.0279	0.12-0.20	414	675	440-490
0.08	0.3246	0.0283	0.13-0.18	497	708	580-660
0.09	0.3608	0.0690	1.12-1.25	621	867	710-750
0.1	0.4360	0.0294	0.16-0.22	778	999	860-960

Πίνακας 9

Στον πίνακα 9 εμφανίζεται η μέγιστη και η ελάχιστη τιμή της $f\left(\hat{\theta}\right)$ καθώς και ο μέγιστος και ελάχιστος αριθμός απαραίτητων επαναλήψεων. Όταν μιλάμε για συχνότερη διακύμανση $f\left(\hat{\theta}\right)$ ή επαναλήψεων, εννοούμε το διάστημα εκείνο στο οποίο ανήκει το 70% των δειγμάτων. Ο πίνακας 9 προήλθε μετά από μεγάλο αριθμό προσομοιώσεων του αλγορίθμου. Από τα εξαγόμενα του πίνακα συμπεραίνουμε ότι για α=0.03 έχουμε και τις καλύτερες και τις ταχύτερες συγκλείσεις.

Για το Α παρατηρούμε ότι όσο μεγαλώνει βελτιώνεται η ταχύτητα σύγκλισης, αλλά χειροτερεύει αισθητά η προσέγγισή μας. Άρα η βέλτιστη τιμή για την μεταβλητή Α είναι (Α=50).

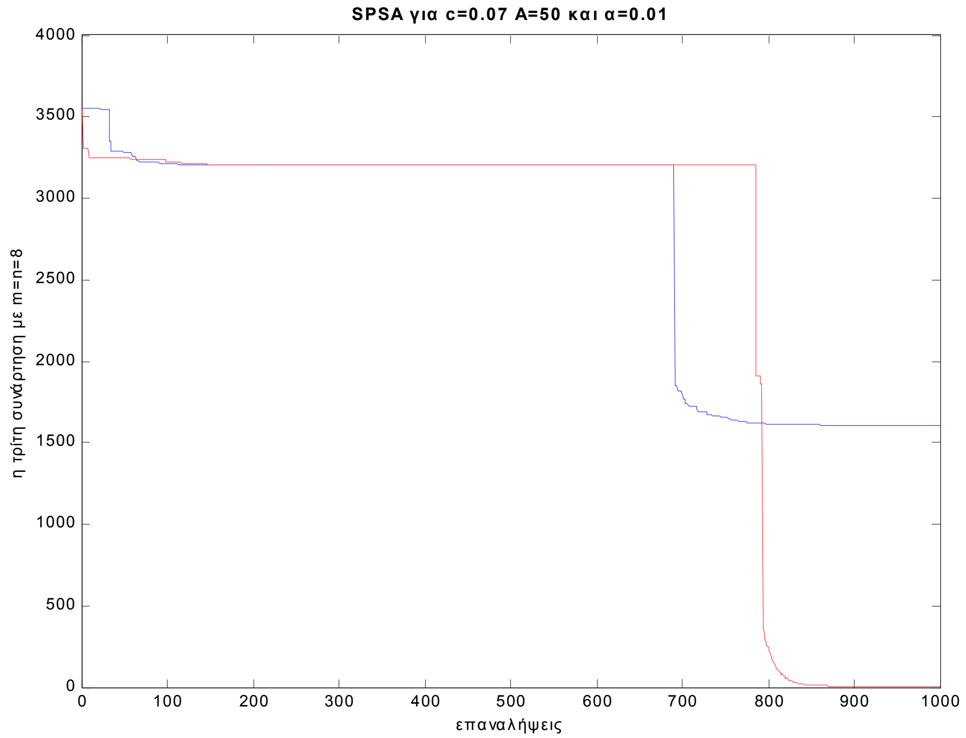
Για να βρούμε το βέλτιστο c, κρατάμε σταθερό το Α (Α=50) και το α (α=0.03) και αρχίζουμε να λειτουργούμε τον αλγόριθμο για c=0.02 έως c=0.1. Εκτελούμε τον αλγόριθμο 10 φορές για την πρώτη τιμή του c, τερματίζουμε τη Matlab και επαναλαμβάνουμε τη διαδικασία για την επόμενη τιμή του c. Έτσι καταλήγουμε στο πίνακα 10 που δίνεται παρακάτω.

Εκτέλεση της Matlab	c=0.02	c=0.03	c=0.04	c=0.05	c=0.06	c=0.07	c=0.08	c=0.09	c=0.10
1	0.1725	**0.1726**	0.1726	0.1496	<u>0.1495</u>*	0.1496	0.1498	0.1500	0.1503
2	**0.1597**	0.1560	0.1548	0.1533	0.1514	0.1493	0.1469	0.1442	<u>0.1412</u>*
3	**0.0841**	0.0838	0.0834	0.0828	0.0821	0.0814	0.0805	0.0795	<u>0.0360</u>*
4	0.1337	0.1336	0.1335	<u>0.1333</u>*	**0.1846**	0.1834	0.1821	0.1806	0.1791
5	<u>0.2339</u>*	0.2341	0.2345	0.2349	0.2355	0.2361	0.2369	0.2378	**0.2388**
6	0.1940	0.1929	0.1914	0.1896	0.1873	<u>0.1847</u>*	**0.2227**	0.2191	0.2159
7	**0.0693**	0.0686	0.0675	0.0282	0.0280	0.0277	0.0275	0.0272	<u>0.0270</u>*
8	0.2117	0.2108	0.2096	0.2080	0.2061	<u>0.2038</u>*	**0.2213**	0.2183	0.2151
9	0.1377	0.1372	<u>0.1365</u>*	**0.1414**	0.1405	0.1395	0.1408	0.1394	0.1378
10	0.1118	0.1106	0.1090	0.1069	0.1040	<u>0.1015</u>*	0.1079	0.1039	**0.1177**

Πίνακας 10

Στον πίνακα 10 οι υπογραμμισμένες τιμές είναι οι βέλτιστες και οι έντονες είναι οι χειρότερες που βρέθηκαν για την κάθε επανάληψη της Matlab. Συμπεραίνουμε από τα αποτελέσματα ότι για c=0.07 ο αλγόριθμος λειτουργεί καλύτερα, αφού για όλες τις επαναλήψεις της Matlab έχουμε πολύ ικανοποιητικές τιμές, εκτός από την 4^{η}, ενώ για την 6^{η},8^{η} και 10^{η} έχουμε τις βέλτιστες. Ανακεφαλαιώνοντας, έχουμε επιλέξει για την τρίτη συνάρτηση, για τον αλγόριθμο SPSA, οι μεταβλητές α,c και A να έχουν τις τιμές 0.03,0.07 και 50 αντίστοιχα.

Για τη τρίτη συνάρτηση πάλι, αλλά για n=m=8, δηλαδή για 8 μεταβλητές και 8 όρους στην αντικειμενική συνάρτηση, παρατηρούμε ότι για α μεγαλύτερο του 0.01 και μικρότερο του 0.003 ο αλγόριθμος δεν ανταποκρίνεται όπως θα θέλαμε. Ψάχνουμε λοιπόν να βρούμε το βέλτιστο α στο διάστημα [0.003,0.01]. Για α=0.01 ο αλγόριθμος δεν φαίνεται καθόλου σταθερός. Αυτό αποδεικνύεται και στο γράφημα 45, όπου για διαφορετικές επαναλήψεις του αλγορίθμου έχουμε τελείως διαφορετικές προσεγγίσεις. Υπενθυμίζουμε ότι η αναμενόμενη ελάχιστη τιμή της είναι μηδέν για x_i=0 i=1,2,…,8. Το γράφημα 45 προήλθε για A=50 και c=0.07



γράφημα 45

Άρα ψάχνουμε το βέλτιστο α στο διάστημα [0.003,0.009]. Για c=0.04,A=50 και α=0.009 εκτελούμε τον αλγόριθμο 6 φορές και καταγράφουμε τις αντίστοιχες παρατηρήσεις. Τερματίζουμε τη Matlab, και συνεχίζουμε τη διαδικασία για όλες τις πιθανές τιμές του α. Έτσι καταλήγουμε στον πίνακα 11

α	Π1	Π2	Π3	Π4	Π5	Π6
0.009	1.0366	1.3181	1.0817	0.6355	0.4085	0.5544
0.008	0.7575	1.0477	0.6775	0.3782	1.0044	0.2202
0.007	0.3306	0.5603	1.2645	0.2301	0.4113	0.8773
0.006	1.0398	1.4962	0.7711	0.9292	0.2399	1.1592
0.005	0.6162	1.0627	0.8603	0.7992	1.0406	1.3455
0.004	0.4492	0.3122	0.9675	1.1550	1.9539	1.8760
0.003	0.3085	1.0592	1.6672	1.9213	1.6006	1.7382

Πίνακας 11

Αντιλαμβανόμαστε ότι για α=0.007 έχουμε μεγαλύτερη ισορροπία στον αλγόριθμο και πολύ καλές προσεγγίσεις στο ελάχιστο της f. Για τους ίδιους λόγους επιλέγουμε και το Α=50, αφού αυξάνοντας το Α μπορούμε να πετύχουμε κάποιες λίγο καλύτερες τιμές, αλλά και κάποιες πολύ χειρότερες. Για να εκτιμήσουμε το βέλτιστο c κατασκευάζουμε τον πίνακα 12

Στατιστικά αποτελέσματα αλγορίθμου SPSA για Α=50 και α=0.007						
c	Π1	Π2	Π3	Π4	Π5	Π6
0.02	0.3307	0.5517	1.2516	0.2276	0.4169	0.8801
0.03	0.3307	0.5516	1.2517	0.2317	0.4145	0.8789
0.04	0.3306	0.5603	1.2645	0.2301	0.4113	0.8773
0.05	0.3305	0.5601	1.2649	0.2195	0.4071	0.8753
0.06	0.3287	0.5599	1.2809	0.2171	0.4022	0.8730
0.07	0.3287	0.5601	1.2818	0.2113	0.9256	0.8705
0.08	0.3288	0.5602	1.2999	0.2082	0.8848	0.6678
0.09	0.3297	0.5604	1.3010	0.2050	0.8931	0.8651
0.1	0.3467	0.5608	1.3227	0.2083	0.8841	0.8394

Πίνακας 12

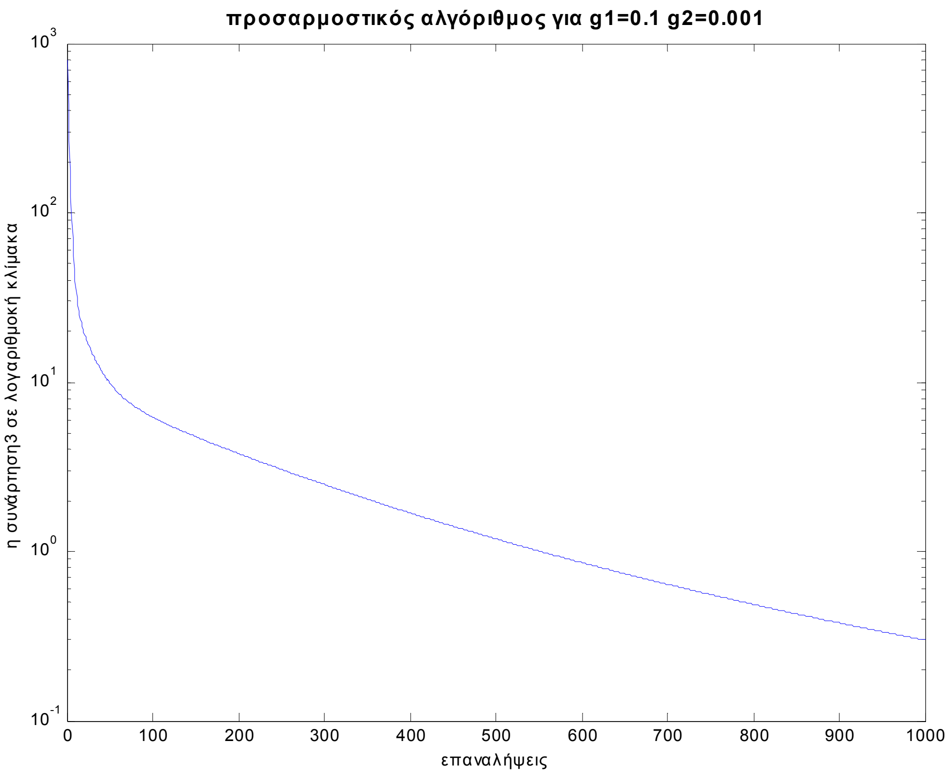
Από τις έξι παρατηρήσεις του πίνακα 12 μπορούμε να επιλέξουμε ως βέλτιστη τιμή της μεταβλητής c την (c=0.06), αφού για αυτήν την τιμή της c εμφανίζονται δύο βέλτιστες (σε σχέση με τις υπόλοιπές παρατηρήσεις) προσεγγίσεις του ελαχίστου της f και σε καμιά περίπτωση δεν εμφανίζονται «ανωμαλίες» στη ροή των τιμών, όπως π.χ. στην 5^{η} παρατήρηση για c=0.07 και στην 6^{η} παρατήρηση για c=0.08. Άρα οι βέλτιστες τιμές των παραμέτρων α,c και Α είναι σε αυτή την περίπτωση 0.007,0.06 και 50 αντίστοιχα.

5.3.3 Προσαρμοστικός αλγόριθμος

Για τον προσαρμοστικό αλγόριθμο, όσο αναφορά την τρίτη συνάρτηση, θα εργαστούμε ομοίως. Υπενθυμίζουμε ότι θ^*=(1,1,1,1) και fmin=0.

Για g1=0.1 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.1 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

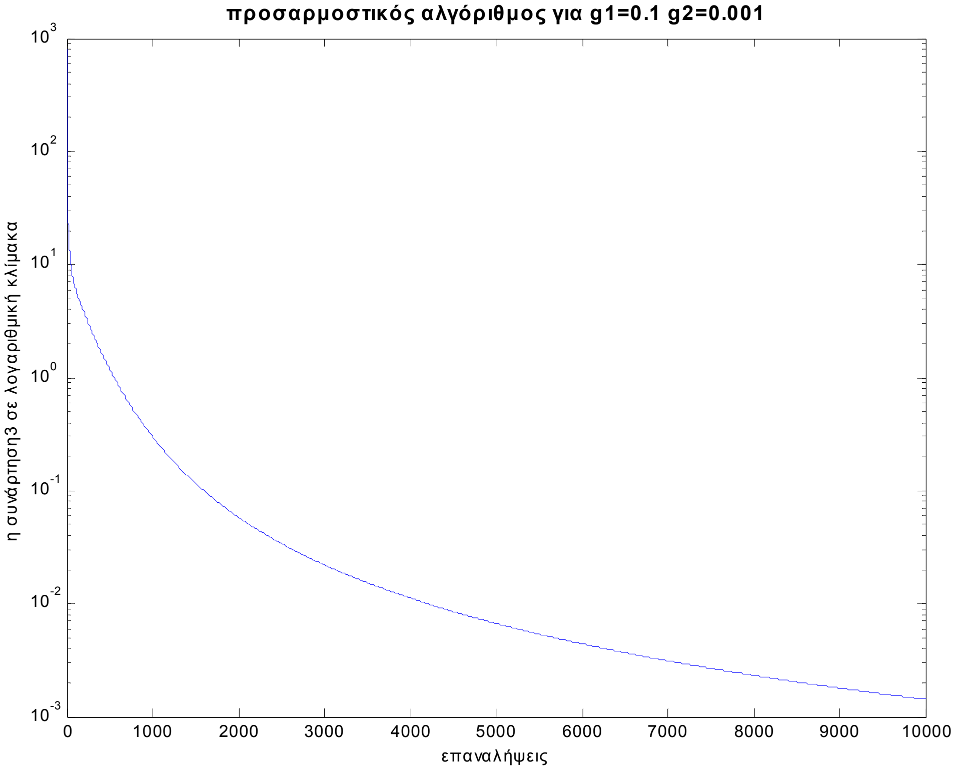
Για g1=0.1 και g2=0.001 ο αλγόριθμος καταλήγει πάντα στο ίδιο αποτέλεσμα (γράφημα 46)



r = 0
Fmin = 0.3006
epanal =1000
x1b = 0.5487
x2b = 0.0571
x3b = 0.2624
x4b = 0.3298
thetabzitoymeno = (4.5106 , 2.1496 , 3.9073, 0.2531)

γράφημα 46

Για 10000 επαναλήψεις καταλήγουμε πάντα στο ίδιο αποτέλεσμα (γράφημα 47)



r = 0
Fmin = 0.0014
epanal =10000
x1b = 0.1287
x2b = 0.0129
x3b = 0.0691
x4b = 0.0703
thetabzitoymeno = (3.4538 , 2.2672 , 3.7486 , 0.3260)

γράφημα 47

Για g1=0.1 και g2=0.0001 ο αλγόριθμος μας δίνει fmin = 7.0149 και κακές εκτιμήσεις του θ, ενώ για 10000 επαναλήψεις fmin = 0.3239 και πάλι κακές εκτιμήσεις του θ.

Για g1=0.01 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.01 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

Για g1=0.01 και g2=0.001 ο αλγόριθμος μας δίνει fmin =0.3006 και καλύτερα θ από προηγουμένως. {π.χ. thetabzitoymeno = (1.9767 , 2.3680 , 1.9502 , 0.5560)}

Για g1=0.01 και g2=0.0001 ο αλγόριθμος μας δίνει fmin = 7.0149 και κακές εκτιμήσεις του θ {π.χ. thetabzitoymeno= (4.9830 , 0.7121 , 4.9467 , 0.4081)}

Για g1=0.001 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.001 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

Για g1=0.001 και g2=0.001 ο αλγόριθμος μας δίνει fmin =0.3006 και καλύτερα θ από προηγουμένως. {π.χ. thetabzitoymeno = (1.6972 , 3.3019 , 1.4963 , 0.1432)}

Για g1=0.001 και g2=0.0001 ο αλγόριθμος μας δίνει fmin = 7.0149 και κακές εκτιμήσεις του θ {π.χ. thetabzitoymeno = (2.1073 , 2.7217 , 4.9587 , -0.5173)}

Για g1=0.0001 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται.
Για g1=0.0001 και g2=0.01 ο αλγόριθμος δεν ανταποκρίνεται.

Για g1=0.0001 και g2=0.001 ο αλγόριθμος μας δίνει fmin =0.3006 και σχετικά καλά θ. {π.χ. thetabzitoymeno = (1.0646 , 1.3491 , 0.1329 , 0.9317)}.

Για g1=0.0001 και g2=0.0001 ο αλγόριθμος μας δίνει fmin = 7.0149 και κακές εκτιμήσεις του θ {π.χ. thetabzitoymeno = (1.4977 , 3.6681 , 4.8305 , -0.7891)}.

Επιλέγουμε λοιπόν (g1 , g2)=(0.0001 , 0.001) για τα οποία ο αλγόριθμος δίνει την καλύτερη προσέγγιση του ελαχίστου της συνάρτησης και αρκετά καλές εκτιμήσεις του θ^*.

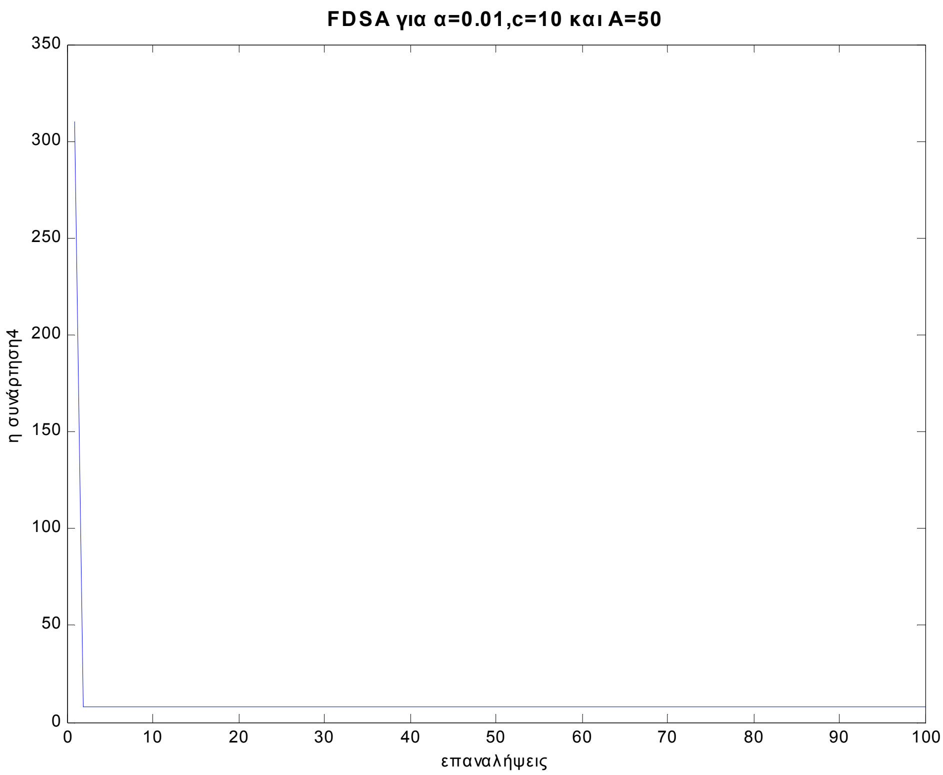
Για την τρίτη συνάρτηση πάλι, αλλά για n=m=8 παρατηρούμε ότι για τα g1,g2 έχουμε την ίδια συμπεριφορά με προηγουμένως. Οπότε πάλι επιλέγουμε (g1 , g2) = (0.0001 , 0.001) για τα οποία για 1000 επαναλήψεις έχουμε fmin = 3.4019e-004, ενώ για 10000 επαναλήψεις fmin = 5.7483e-005.

5.4 Συνάρτηση 4 (Penalty function)

Υπενθυμίζουμε ότι η τέταρτη συνάρτηση παρουσιάζει ελάχιστο (fmin= 2.24997…* 10^{-5}) για n = 4. Η αρχική τιμή θα είναι x_0=(1 , 2 , 3 , 4).

5.4.1 Μέθοδος FDSA

Παρατηρούμε πολύ εύκολα ότι ο αλγόριθμος έχει καλύτερη συμπεριφορά για α=0.01. Αρχίζουμε με c=0.01 και αυξάνοντας το συνεχώς βελτιώνονται οι προσεγγίσεις μας. Η καλύτερη προσέγγιση στο πραγματικό fmin της συνάρτησης 4, επιτυγχάνεται για c=10. Για A=50 (η ελάχιστη δυνατή τιμή) έχουμε την καλύτερη συμπεριφορά του αλγορίθμου. Στο γράφημα 48 εμφανίζεται το αποτέλεσμα της βέλτιστης δυνατής εκτέλεσης του αλγορίθμου FDSA για την $4^η$ συνάρτηση.



γράφημα 48

Οι πραγματικές επαναλήψεις ήταν 1000, αλλά στο γράφημα 48 παρουσιάζονται για περισσότερη ευκολία οι πρώτες 100. Στις υπόλοιπες ο αλγόριθμος απλά δεν μπορεί να προσεγγίσει καλύτερα το επιθυμητό αποτέλεσμα. Η προσέγγιση γίνεται στην 36^{η} επανάληψη του αλγορίθμου.

5.4.2 Μέθοδος SPSA

Για α μεγαλύτερα του 0.2 ο αλγόριθμος δεν ανταποκρίνεται καθόλου. Για α μικρότερα του 0.02 η προσέγγιση στην ελάχιστη τιμή της f δεν είναι καθόλου ικανοποιητική αφού παίρνει τιμές μεγαλύτερες της μονάδος και για α=0.005 φτάνουμε και στην τιμή 4. Το α λοιπόν βρίσκεται στο διάστημα (0.02 , 0.2)

Στατιστικά αποτελέσματα αλγορίθμου SPSA για A=50 και c=0.04										
	Π1		Π2		Π3		Π4		Π5	
α	fmin	Επ/ψη	fmin	Επ/ψη	fmin	Επ/ψη	fmin	Επ/ψη	fmin	Επ/ψη
0.02	<u>0.5165</u>	142	<u>0.5442</u>	102	<u>0.4697</u>	109	<u>0.4746</u>	**93**	<u>0.5133</u>	106
0.04	0.3679	**72**	0.2201	103	0.2329	**96**	0.2701	106	0.2377	**90**
0.06	0.2130	129	0.2121	**77**	0.2935	102	0.2057	102	0.2138	106
0.08	0.1784	178	0.1788	152	0.1766	144	0.1969	144	0.2021	159
0.1	0.1840	210	0.2253	201	0.1894	229	0.1563	244	0.1563	202
0.12	0.1665	277	0.1563	289	0.2174	289	0.1674	286	0.1536	327
0.14	0.1496	381	0.1526	359	0.1394	383	**0.1384**	383	**0.1327**	381
0.16	0.1485	431	**0.1475**	474	0.1747	445	0.1544	450	0.1445	468
0.18	0.1449	532	0.2280	524	0.1559	519	0.1851	520	0.1388	553
0.2	**0.1406**	<u>690</u>	0.1991	<u>624</u>	**0.1283**	<u>659</u>	0.1749	<u>630</u>	0.1606	<u>629</u>

Πίνακας 13

Στον πίνακα 13 με έντονους χαρακτήρες εμφανίζονται οι βέλτιστες τιμές για κάθε παρατήρηση και με υπογραμμισμένους χαρακτήρες οι χείριστες. Παρακολουθώντας τον πίνακα συμπεραίνουμε ότι όσο αυξάνεται το α, τόσο αυξάνεται ο απαραίτητος αριθμός επαναλήψεων, ώστε να γίνει η τελική προσέγγιση του αλγορίθμου στο ελάχιστο της f. Ο αλγόριθμος έχει καλύτερη συμπεριφορά για α=0.14, αφού στην 4^{η} και 5^{η} παρατήρηση έχουμε βέλτιστη συμπεριφορά του αλγορίθμου, στη 2^{η} και 3^{η} παρατήρηση έχουμε τη δεύτερη καλύτερη προσέγγιση στο ελάχιστο της f και στη 1^{η} παρατήρηση έχουμε πολύ μικρή διαφορά από την βέλτιστη προσέγγιση. Τώρα θα διερευνήσουμε τη παράμετρο α για τις τιμές 0.13 και 0.15

Στατιστικά αποτελέσματα αλγορίθμου SPSA για A=50 και c=0.04										
	Π1		Π2		Π3		Π4		Π5	
0.13	0.1497	318	0.1645	306	0.1616	311	0.2227	291	0.1446	364
0.14	0.1496	381	0.1526	359	0.1394	383	0.1384	383	0.1327	381
0.15	0.0061	340	0.1493	423	0.1656	398	0.1431	397	0.1483	395

Πίνακας 14

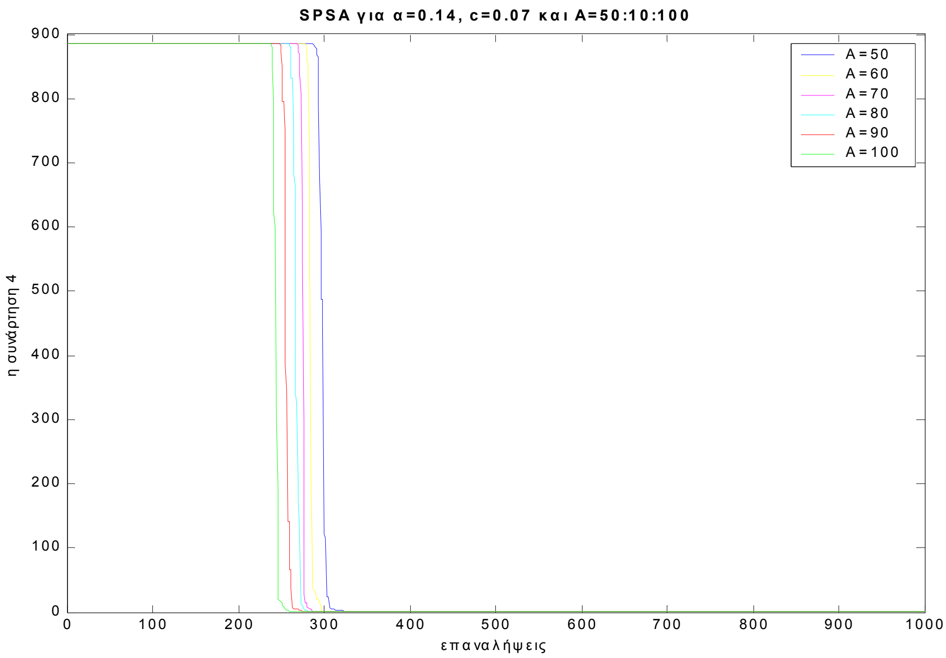
Αν εξαιρέσουμε την 1^{η} παρατήρηση για α=0.15 που μας δίνει την καλύτερη μέχρι τώρα τιμή, για α=0.14 0 αλγόριθμος συνεχίζει να έχει καλύτερη συμπεριφορά. Κάνοντας αρκετές επαναλήψεις του αλγορίθμου για α=0.15, παρατηρούμε ότι η τιμή 0.0061 στην πρώτη παρατήρηση είναι τυχαία, αφού σε καμιά άλλη παρατήρηση δεν έχουμε τόσο καλές τιμές.

Τώρα για α=0.14 και A=50 θα ψάξουμε για τη βέλτιστη τιμή της παραμέτρου c στο διάστημα (0.02 , 0.1). Τα αποτελέσματα της δουλειάς μας παρατίθενται στον πίνακα 15. Από τον πίνακα αυτόν βλέπουμε ότι για αύξηση του c από 0.02 έως 0.05 έχουμε μείωση των προσεγγίσεων μας σε όλες τις παρατηρήσεις. Από εκεί και πέρα έχουμε τυχαία, βελτίωση ή αύξηση των τιμών, ανάλογα με την παρατήρηση, δηλαδή ανάλογα με την αρχική τιμή του διανύσματος delta {delta=2*round(rand(p,1))-1}. Η αρχική τιμή του διανύσματος «delta» επηρεάζεται από τη «λίστα» των τυχαίων αριθμών της Matlab, και έχει να κάνει με το ποια μεταβλητή (x_i) θα αλλάξουμε για να βελτιώσουμε τη προσέγγισή μας. Στατιστικά, καλύτερη συμπεριφορά παρατηρείται για c=0.07 και αυτή τη τιμή της παραμέτρου c θα επιλέξουμε.

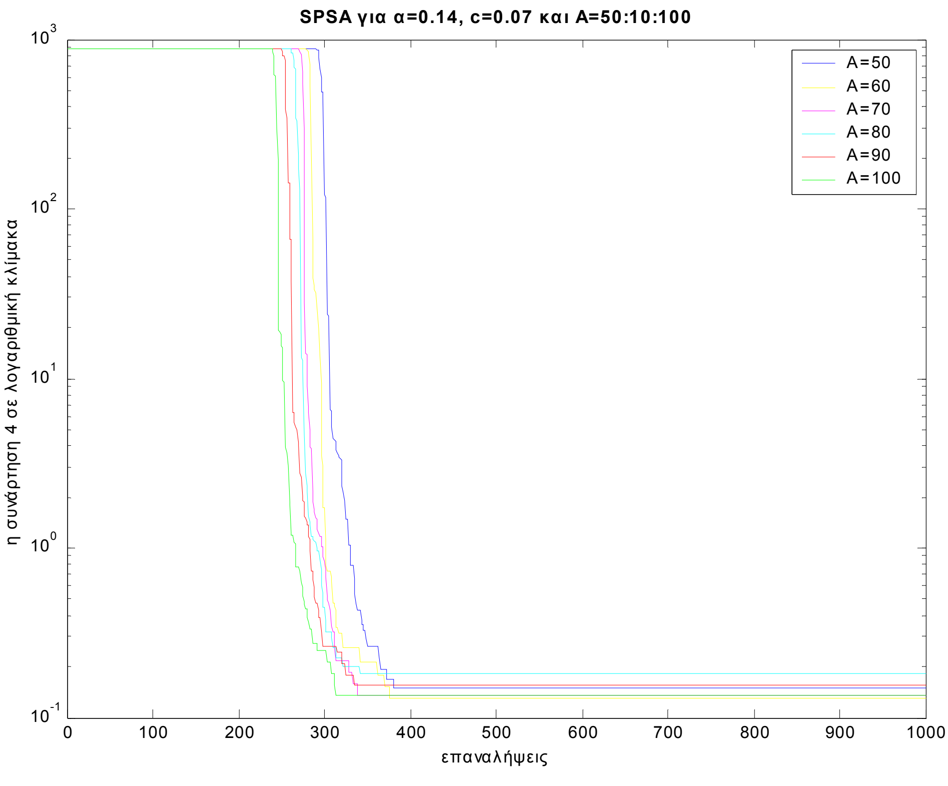
Στατιστικά αποτελέσματα αλγορίθμου SPSA για A=50 και α=0.14										
	Π1		Π2		Π3		Π4		Π5	
c	fmin	Επ/ψη	fmin	Επ/ψη	fmin	Επ/ψη	fmin	Επ/ψη	fmin	Επ/ψη
0.02	0.1501	381	0.1533	359	0.1397	383	0.1378	383	0.1338	381
0.03	0.1499	381	0.1530	359	0.1396	383	0.1381	383	0.1333	381
0.04	0.1496	381	0.1526	359	0.1394	383	0.1384	383	0.1327	381
0.05	0.1492	381	0.1521	359	0.1393	383	0.1371	383	0.1319	381
0.06	0.1488	381	0.1518	359	0.1391	383	0.1377	383	0.1310	381
0.07	0.1483	381	0.1518	359	0.1389	383	0.1386	383	0.1300	381
0.08	0.1477	381	0.1522	359	0.1380	383	0.1398	383	0.1289	381
0.09	0.1669	373	0.1532	359	0.1378	383	0.1415	383	0.1277	381
0.1	0.1660	373	0.1552	359	0.1377	383	0.1411	383	0.1266	381

Πίνακας 15

Για την επιλογή του Α παραθέτουμε δύο γραφήματα (γράφημα 49 και 50), το ένα σε κανονική και το δεύτερο σε λογαριθμική κλίμακα. Παρατηρούμε ότι για Α=60 ο αλγόριθμος αποκτά την καλύτερη συμπεριφορά όσο αναφορά την προσέγγιση στο ελάχιστο της συνάρτησης. Επίσης παρατηρούμε ότι όσο αυξάνεται το Α μειώνονται οι απαραίτητες επαναλήψεις του αλγορίθμου.(π.χ. για Α=60 0 αλγόριθμος καταλήγει στην βέλτιστη τιμή 0.1303 σε 339 επαναλήψεις, ενώ για Α=100, στη βέλτιστη τιμή 0.1352 σε 313 επαναλήψεις. Στις υπόλοιπες επαναλήψεις του αλγορίθμου δεν υπάρχει καμιά απολύτως αλλαγή των μεταβλητών ή μείωση της τιμής της συνάρτησης)



γράφημα 49



γράφημα 50

Οι βέλτιστες τιμές των παραμέτρων α,c και A για την 4^{η} συνάρτηση, στον αλγόριθμο SPSA είναι 0.14,0.07 και 60 αντίστοιχα.

5.4.3 Προσαρμοστικός αλγόριθμος

Υπενθυμίζουμε ότι για την τέταρτη συνάρτηση ισχύουν fmin = 2.24997…* 10^{-5} και θ^*=(1,1,1,1,1). Για την τέταρτη συνάρτηση ο προσαρμοστικός αλγόριθμος επηρεάζεται από τα g1,g2 ως εξής:

Για g1=0.1 και g2=0.1 ο αλγόριθμος δεν ανταποκρίνεται

Για g1=0.1 και g2=0.01 παίρνουμε fmin = 3.5942e-005 και όχι καλές εκτιμήσεις του θ^* {π.χ. thetabzitoymeno = (-0.6906 , -0.0109 , -0.8262 , 0.1461 , -2.7109)}

Για g1=0.1 και g2=0.001 έχουμε fmin = 0.0016 και όχι καλές εκτιμήσεις του θ^* {π.χ. thetabzitoymeno = (-0.2658 , 0.8040 , 2.4890 ,0.5306 , 1.0000)}

Για g1=0.1 και g2=0.0001 έχουμε fmin να ανήκει στο διάστημα (1 , 1.5) και όχι καλές εκτιμήσεις του θ^* .
{π.χ. thetabzitoymeno = (1.5645 , -0.1483 , -2.2775 , 0.2992 , 1.0000)}

Για g1=0.01 και g2=0.01 παίρνουμε fmin = 3.5942e-005 και καθόλου καλές εκτιμήσεις του θ^* .
{π.χ. thetabzitoymeno = (-0.4856 , -0.8731 , -0.9530 , 0.5420 , 47.7551)}

Για g1=0.01 και g2=0.001 έχουμε fmin = 0.0016 και όχι καλές εκτιμήσεις του θ^* {π.χ. thetabzitoymeno = (-0.1499 , -1.5698 , 0.0690 , 0.5426 , 1.0000)}.

Για g1=0.01 και g2=0.0001 έχουμε fmin να ανήκει στο διάστημα (1 , 1.5) και όχι καλές εκτιμήσεις του θ^* .
{π.χ. thetabzitoymeno = (0.1864 , 1.4427 , 0.2235 , 0.2551 , 1.0000)}.

Για g1=0.001 και g2=0.01 παίρνουμε fmin = 3.5942e-005 και καθόλου καλές εκτιμήσεις του θ^* .
{π.χ. thetabzitoymeno = (-2.1484 , 0.4666 , -1.8882 , -0.5833 , 0.6888)}.

Για g1=0.001 και g2=0.001 έχουμε fmin = 0.0016 και όχι καλές εκτιμήσεις του θ^* {π.χ. thetabzitoymeno = (-0.9296 , -0.7239 , 1.2942 , 0.2157 , 1.0000)}.

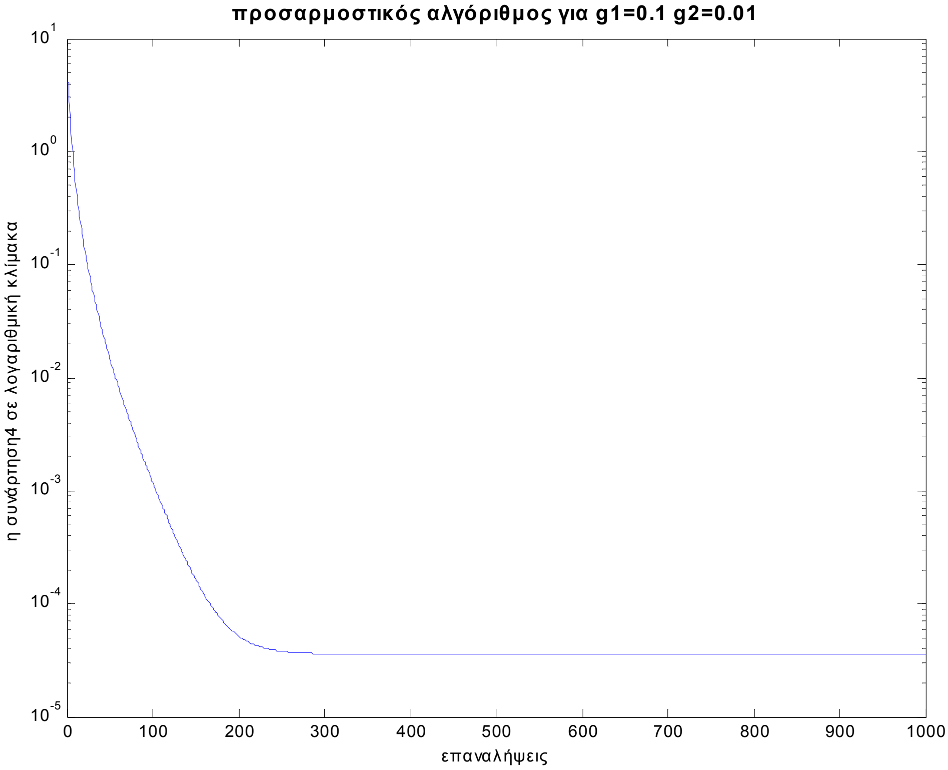
Για g1=0.001 και g2=0.0001 έχουμε fmin να ανήκει στο διάστημα (1 , 1.5) και όχι καλές εκτιμήσεις του θ^* .
{π.χ. thetabzitoymeno = (-0.6561 , -1.1809 , -0.5507 , 0.1927 , 1.0000)}.

Για g1=0.0001 και g2=0.01 παίρνουμε fmin = 3.5942e-005 και καθόλου καλές εκτιμήσεις του θ^* .
{π.χ. thetabzitoymeno = (0.0892 , 1.2373 , 0.5426 , -2.0772 , 6.8169)}.

Για g1=0.0001 και g2=0.001 έχουμε fmin = 0.0016 και όχι καλές εκτιμήσεις του θ^* {π.χ. thetabzitoymeno = (-0.4325 , -1.6656 , 0.1253 , 0.2875 , 1.0000)}.

Για g1=0.0001 και g2=0.0001 έχουμε fmin να ανήκει στο διάστημα (1 , 1.5) και όχι καλές εκτιμήσεις του θ^* .
{π.χ. thetabzitoymeno = (0.5982 , 0.1473 , -0.1015 , -2.6350 , 1.0000)}.

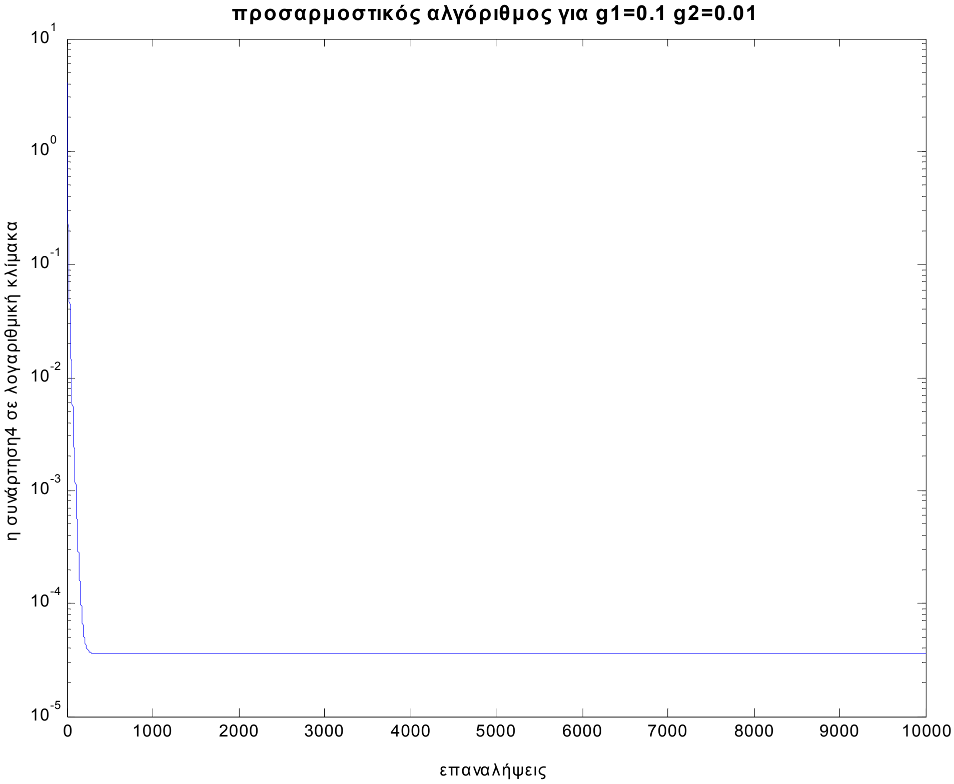
Βλέπουμε ότι τα βέλτιστα αποτελέσματα δίνονται για g2=0.01 ενώ όσο και να αλλάξουμε το g1 δεν υπάρχει καμιά διαφορά. Επιλέγουμε λοιπόν (g1 , g2) = (0.1 , 0.01). Για αυτά τα g1,g2 ακολουθούν τα γραφήματα 51 και 52 για 1000 και 10000 επαναλήψεις αντίστοιχα.



r = 1
Fmin = 3.5942e-005
epanal = 1000
x1b = -0.0627
x2b = -0.1002
x3b = 0.0051
x4b = 0.4858
thetabzitoymeno = (-1.6016 , -0.5723 , 0.2836 , 0.8164 , 3.6519)

γράφημα 51

Βλέπουμε ότι από την $200^{η}$ επανάληψη και μετά η βελτίωση των εκτιμήσεων μας είναι απειροελάχιστη αφού ήδη έχουμε φτάσει σε τιμές της τάξης του $α*10^{-5}$, όπου $0<α<10$.



r = 1
Fmin = 3.5814e-005
epanal = 10000
x1b = -0.0608
x2b = -0.0982
x3b = 0.0069
x4b = 0.4864
thetabzitoymeno = (-0.3835 , -1.7749 , -0.2694 , 2.1206 , 6.1438)

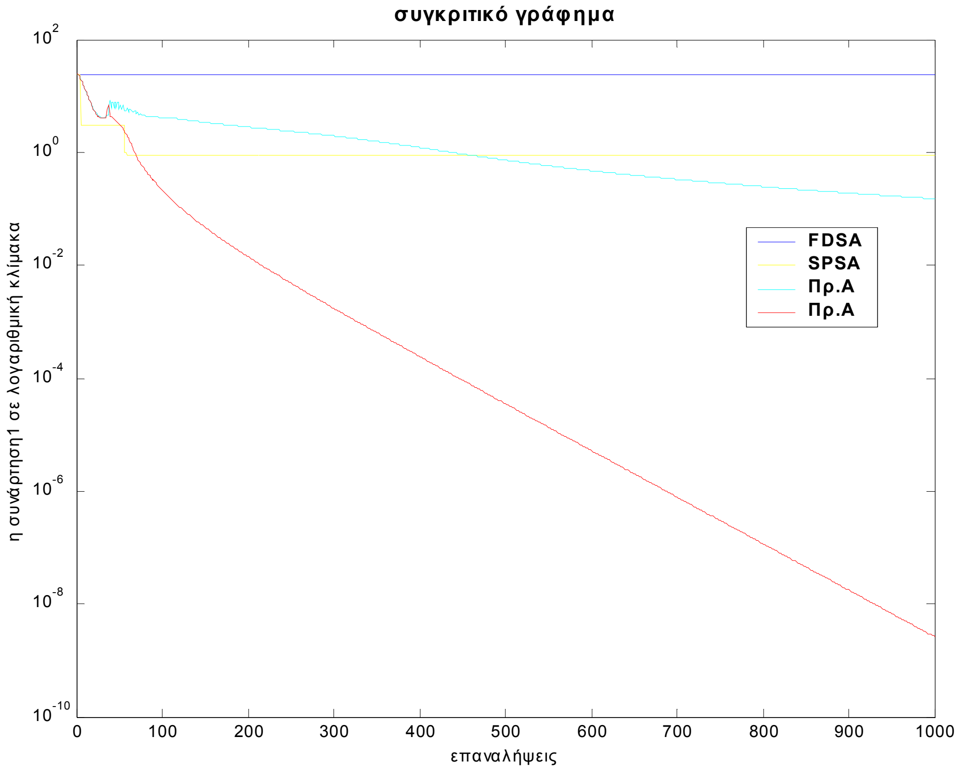
γράφημα 52

Όσο αυξάνουμε τις επαναλήψεις, τόσο με πιθανότητα 1 οδηγούμαστε στο πραγματικό ελάχιστο της συνάρτησης. Όταν αυτή η προσέγγιση επιτευχθεί, ο αλγόριθμος θα έχει εκτιμήσει σωστά και την άγνωστη παράμετρο θ.

6. Σύγκριση αλγορίθμων

Τώρα που βεβαιωθήκαμε ότι έχουμε βρει τις τιμές των παραμέτρων, για τις οποίες όλοι οι αλγόριθμοι παρουσιάζουν βέλτιστη συμπεριφορά, είμαστε έτοιμοι να συγκρίνουμε τους αλγορίθμους μεταξύ τους. Αυτή η σύγκριση θα γίνει για κάθε μία από τις τέσσερις (4) συναρτήσεις ξεχωριστά, και θα έχει να κάνει με την τελική εκτίμηση του ελαχίστου της κάθε συνάρτησης, καθώς και τον αριθμό που καλέστηκε η συνάρτηση ανά βήμα, από τον κάθε αλγόριθμο.

6.1 Συνάρτηση 1

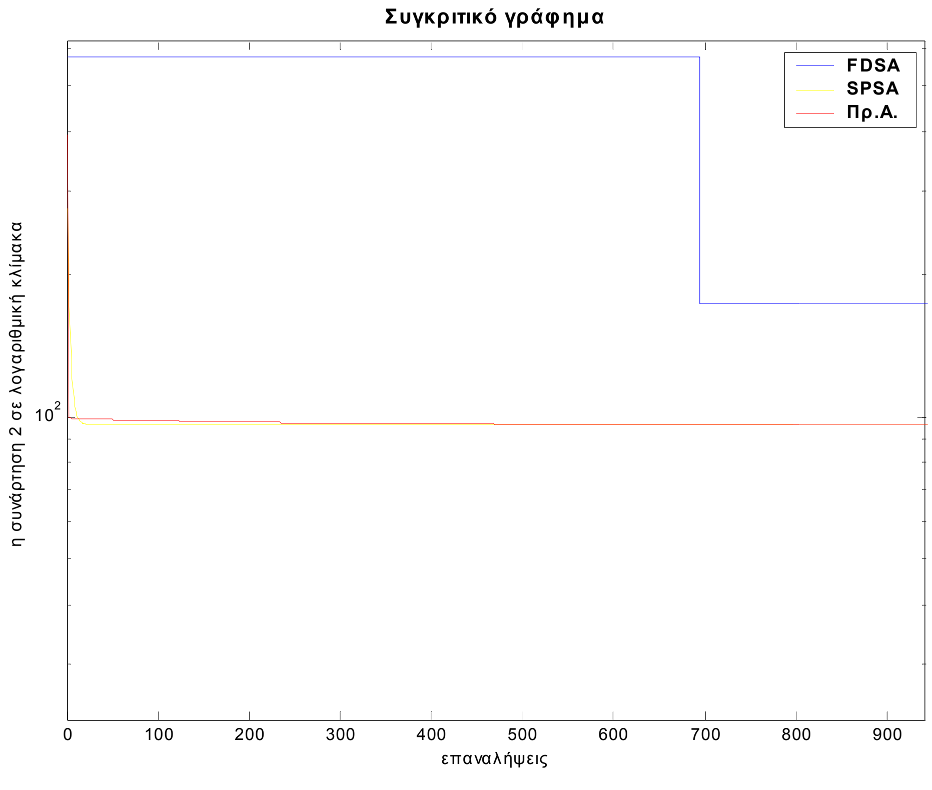


	FDSA	SPSA	Πρ.Α(1)	Πρ.Α(2)
ελάχιστο	LFDmin = 24.2000	LL1min = 0.8746	fmin = 0.1509	fmin = 2.6488e-009
επανάληψη	epanal = 1	epanal = 59	epanal = 1000	epanal =1000
r	-	-	r = 27	r = 4
thetab	-	-	thetab=(98.8188,1.2419)	thetab=(173.9143,17.3794)

γράφημα 53

Στο γράφημα 53 έχουμε τον αλγόριθμο FDSA (μπλε), τον SPSA (κίτρινο), τον προσαρμοστικό αλγόριθμο 1 (Πρ.Α γαλάζιο) και τον προσαρμοστικό αλγόριθμο 2 (Πρ.Α κόκκινο). Οι δύο προσαρμοστικοί αλγόριθμοι του γραφήματος είναι για τα ίδια (g1,g2)=(0.001,0.001) και αντιστοιχούν στις υποπεριπτώσεις του πίνακα 4 για 28% και 48% αντίστοιχα. Και οι δύο Πρ.Α δίνουν καλύτερες προσεγγίσεις με τον Πρ.Α(2) να ξεχωρίζει με διαφορά και ας μη καταφέρνει να εκτιμήσει απόλυτα την πραγματική τιμή της παραμέτρου θ (thetab).

6.2 Συνάρτηση 2

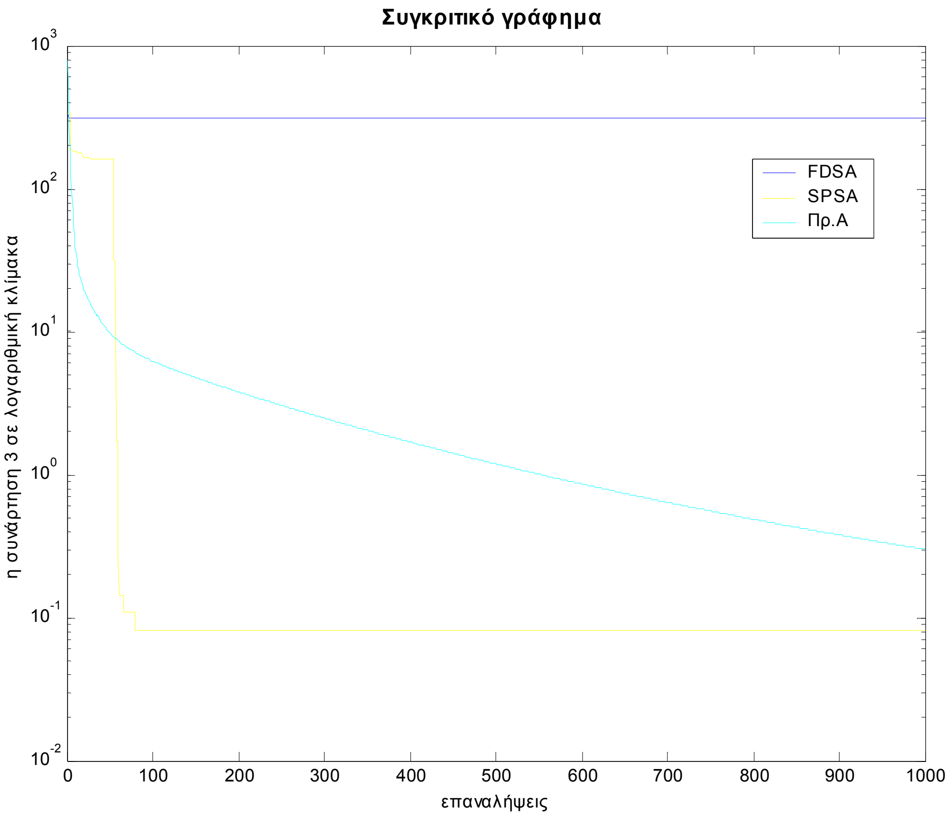


	FDSA	SPSA	Πρ.Α
ελάχιστο	LFDmin = 173.6607	LL1min = 96.1945	fmin = 96.2407
επανάληψη	epanal = 695	epanal = 30	epanal = 1000
r	-	-	r = 0
thetab	-	-	thetab=(-0.4729,13226)

γράφημα 54

Από το γράφημα 54 συμπεραίνουμε ότι σε αυτή τη συνάρτηση καλύτερη μέθοδος εμφανίζεται η SPSA. Για αυτή τη συνάρτηση όμως, όπως δείξαμε και στις παραγράφους 4.2.2 και 4.2.3, στον προσαρμοστικό αλγόριθμο εμφανίζεται πάντα η ίδια τιμή για το ελάχιστο της f, ενώ στην SPSA οι τιμές ποικίλουν ανάμεσα στο διάστημα (95-102). Αυτό είναι ένας λόγος να επιλέξουμε ως αλγόριθμο με την καλύτερη προσέγγιση τον προσαρμοστικό, παρά το γράφημα 54.

6.3 Συνάρτηση 3

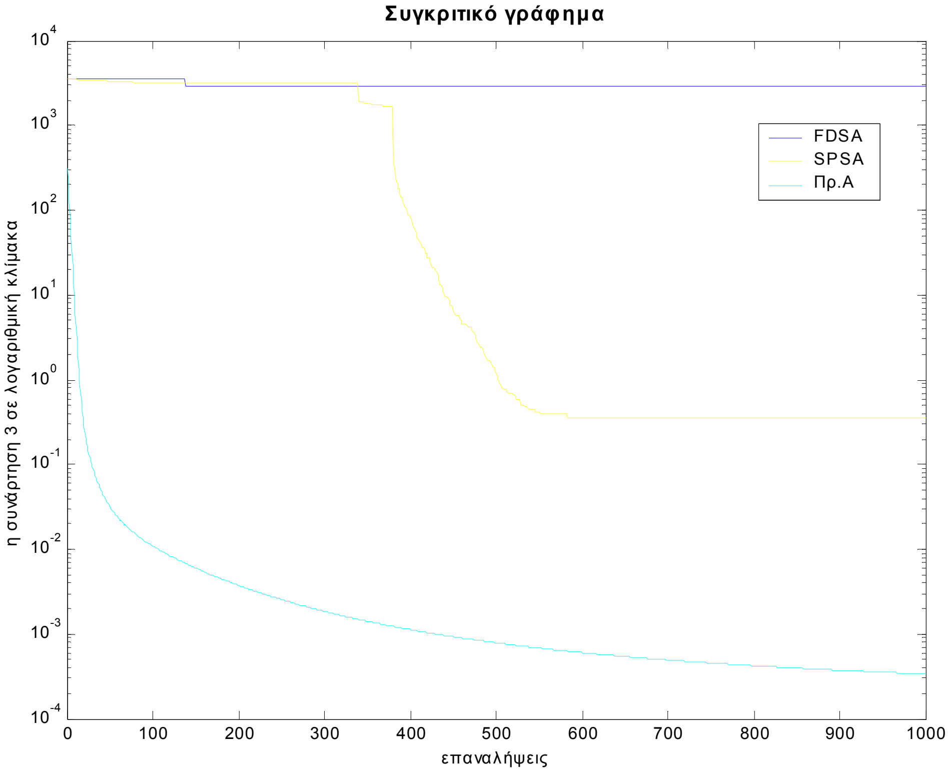


	FDSA	SPSA	Πρ.Α
ελάχιστο	LFDmin = 311.3729	LL1min = 0.0814	fmin =0.3006
επανάληψη	epanal = 2	epanal = 80	epanal = 1000
r	-	-	r = 0
thetab	-	-	thetab=(1.0557,-0.0161,3.4867,1.1669)

γράφημα 55

Από το γράφημα 55 συμπεραίνουμε ότι για την τρίτη συνάρτηση με τέσσερις μεταβλητές, λειτουργεί καλύτερα ο SPSA αλγόριθμος. Από τον πίνακα 10 φαίνεται ξεκάθαρα ότι οι προσεγγίσεις στο minimum της f είναι πάντα πολύ καλύτερες για τον SPSA από ότι στον προσαρμοστικό αλγόριθμο βελτιστοποίησης. Προτιμούμε σε αυτή τη περίπτωση τον αλγόριθμο SPSA, αλλά παραθέτουμε και μία πολύ σημαντική παρατήρηση. Ενώ ο SPSA σε λίγες επαναλήψεις καταλήγει σε ένα ελάχιστο στο διάστημα (0.02 , 0.21), ο προσαρμοστικός για 1000 επαναλήψεις καταλήγει στο 0.3006, για 10000 επαναλήψεις στο 0.0014 και γενικότερα όσο αυξάνονται οι επαναλήψεις τόσο ο αλγόριθμος βελτιώνει τις εκτιμήσεις μας.

Για την τρίτη συνάρτηση πάλι, αλλά αυτή τη φορά για οκτώ μεταβλητές το συγκριτικό γράφημα δίνεται παρακάτω.

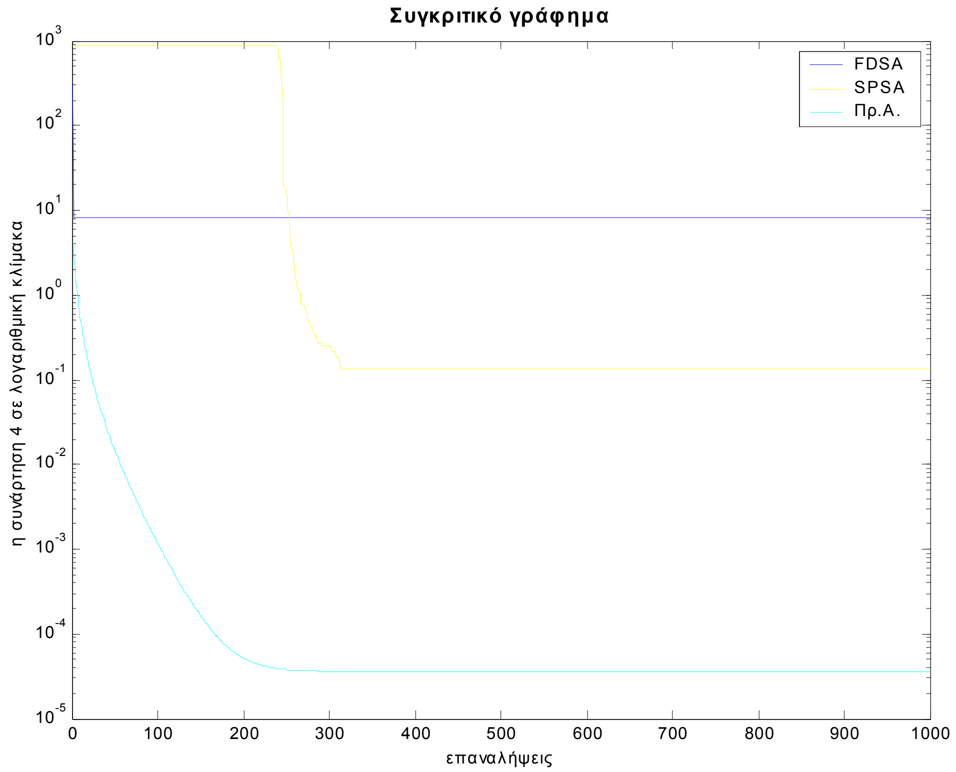


	FDSA	SPSA	Πρ.Α
ελάχιστο	LFDmin = 2855.7	LL1min = 0.3542	fmin = 3.4019 e-004
επανάληψη	epanal = 138	epanal = 584	epanal = 1000
r	-	-	r = 0
thetab	-	-	thetab=(1.1836,2.8657,6.5472,0.0374,0.3377,1.8494,0.1482,0.7250)

γράφημα 56

Από το γράφημα 56 και από τα αποτελέσματα του κεφαλαίου 5, συμπεραίνουμε ότι για αυτή τη περίπτωση ο αλγόριθμος που δίνει τα πιο ικανοποιητικά αποτελέσματα, είναι ο προσαρμοστικός αλγόριθμος βελτιστοποίησης.

6.4 Συνάρτηση 4



	FDSA	SPSA	Πρ.Α
ελάχιστο	LFDmin = 8.0786	LL1min = 0.1352	fmin = 3.5942 e-005
επανάληψη	epanal = 36	epanal = 313	epanal = 1000
r	-	-	r = 1
thetab	-	-	thetab=(-0.4284,-1.6636,0.1183,0.2888,2.5177)

γράφημα 57

Από το γράφημα 57 φαίνεται ξεκάθαρα ότι ο προσαρμοστικός αλγόριθμος δίνει καλύτερα αποτελέσματα και σε αυτό το πρόβλημα. Έχουμε και καλύτερη ταχύτητα και μεγαλύτερη ακρίβεια σύγκλισης. Αλλά και πάλι παρατηρούμε (όπως και στη παράγραφο 5.4.3) ότι στον προσαρμοστικό αλγόριθμο όσο αυξάνονται οι επαναλήψεις τόσο οδηγούμαστε σε καλύτερη ακρίβεια σύγκλισης, ενώ στους αλγορίθμους FDSA και SPSA μετά από κάποιο συγκεκριμένο αριθμό επαναλήψεων, φτάνουμε στη βέλτιστη δυνατή εκτίμηση χωρίς να υπάρχει πιθανότητα καλύτερων προσεγγίσεων.

7. Συμπεράσματα

Παρατηρούμε από όλα τα συγκριτικά γραφήματα, ότι ο προσαρμοστικός αλγόριθμος έχει καλύτερη συμπεριφορά από τους SPSA και FDSA. Ο FDSA δεν λειτουργεί καθόλου ικανοποιητικά. Οι προσεγγίσεις που μας δίνει είναι πολύ «μακριά» από τα πραγματικά ελάχιστα των συναρτήσεων που εξετάζουμε. Επίσης ο FDSA είναι πολύ ευαίσθητος για τις αλλαγές των παραμέτρων. Παρατηρούμε από την διερεύνηση του 5^{ου} κεφαλαίου ότι δίνει αποτελέσματα για πολύ συγκεκριμένες τιμές των παραμέτρων και ακόμα και για συγκεκριμένες αρχικές τιμές των μεταβλητών x. Σε κάθε βήμα ο FDSA υπολογίζει p+1 φορές τη συνάρτηση, όπου p η διάσταση του διανύσματος x. (αριθμός μεταβλητών)

Ο SPSA μας δίνει ικανοποιητικά αποτελέσματα και πολύ καλές προσεγγίσεις στα ελάχιστα των προς βελτιστοποίηση συναρτήσεων. Ο SPSA είναι αρκετά ευαίσθητος όσο αφορά την επιλογή των παραμέτρων του. Σε αυτή τη περίπτωση απαιτούνται 2 υπολογισμοί της συνάρτησης ανά επανάληψη του αλγορίθμου. Με λίγα λόγια ενώ για τον FDSA και για 1000 επαναλήψεις πραγματοποιούνται για ένα πρόβλημα διαστάσεως 4, 5000 υπολογισμοί, για τον SPSA πραγματοποιούνται 2000 ανεξάρτητα από τη διάσταση του προβλήματος.

Ο προσαρμοστικός αλγόριθμος βελτιστοποίησης μας δίνει πάρα πολύ ικανοποιητικά αποτελέσματα. Οι προσεγγίσεις που παίρνουμε αγγίζουν σε ορισμένες περιπτώσεις τα πραγματικά ελάχιστα των προς εξέταση συναρτήσεων. Είναι αρκετά ευαίσθητος αλλά όχι όσο οι αλγόριθμοι FDSA και SPSA. Για τον προσαρμοστικό αλγόριθμο βελτιστοποίησης δεν μπορούμε να υπολογίσουμε τον ακριβή αριθμό των υπολογισμών που απαιτούνται ανά βήμα. Για οποιοδήποτε πρόβλημα και ανεξάρτητα από τη διάσταση του, στο κυρίως πρόγραμμα απαιτείται μόλις μία εκτίμηση της F ανά επανάληψη. Δεν μπορούμε όμως να υπολογίσουμε με ακρίβεια τον αριθμό των εισόδων του αλγορίθμου στη random διαδικασία στην οποία πραγματοποιούνται άλλες δέκα εκτιμήσεις. Στις πρώτες δύο περιπτώσεις παρατηρήθηκε ότι ο αλγόριθμος εισέρχεται στη random διαδικασία από 27 έως 333 φορές. Στις συναρτήσεις 3 και 4, ο αριθμός αυτός για 1000 πάντα επαναλήψεις περιορίστηκε στο διάστημα (0 , 3). Παρόλα αυτά για 1000 επαναλήψεις, οι εκτιμήσεις της συνάρτησης που απαιτούνται για τον προσαρμοστικό αλγόριθμο βελτιστοποίησης είναι το λιγότερο 1000 και το περισσότερο 11000 αφού για κάθε είσοδο στη random διαδικασία χρειάζονται 10 εκτιμήσεις της συνάρτησης.

Πρέπει να τονισθεί επίσης ένα πλεονέκτημα του προσαρμοστικού αλγορίθμου σε σχέση με τους άλλους δύο που μελετήσαμε. Αυτό το πλεονέκτημα είναι το ότι για τα προβλήματα 3 και 4, όσο αυξάνουμε τις επαναλήψεις (τουλάχιστον ως 10000) τόσο προσεγγίζουμε καλύτερα τη συνάρτηση και άρα πετυχαίνουμε μεγαλύτερη ακρίβεια. Οι FDSA και SPSA από κάποιο σημείο και μετά παραμένουν στην ίδια εκτίμηση μη μπορώντας να υπολογίσουν μικρότερες τιμές της F.

Γενικά ο προσαρμοστικός αλγόριθμος υπερισχύει των άλλων δύο και ως προς την ακρίβεια και την ταχύτητα της σύγκλισης τους, αλλά και ως προς την ευαισθησία, όσο αφορά την επιλογή των παραμέτρων των αλγόριθμων.

8. Βιβλιογραφία – Αναφορές

[1] Harold J. Kushner, G. George Yin, "*Stochastic Approximation Algorithms and Applications",* New York: Springer – Verlag, 1997.

[2] M. B. Nevel'son, R. Z. Has'minskii, "*Translations of Mathematical Monographs,* τομ. 47, *Stochastic Approximation and Recursive Estimation",* Rhode Island: American Mathematical Society Providence, 1973.

[3] James Spall, "*Stochastic Optimization, Stochastic Approximation and Simulated Annealing",* Willey EEE, τομ. 20.

[4] "*Multivariate Stochastic Approximation using a Simultaneous Perturbation Gradient Approximation",* IEEE Trans. Automat. Control, τόμ. 37, 1992.

[5] "*Adaptive Stochastic Approximation by the Simultaneous Perturbation Method",* IEEE Trans. Automat. Control τόμ. 45, 2000

[6] Μάρκος Παπαγεωργίου, "*Σημειώσεις Μη Γραμμικού Προγραμματισμού*", Πολ. Κρήτης, Χανιά, 2000.

[7] James Spall, John A. Christon, *Model-Free Control of Nonlinear Stochastic Systems with Discrete-Time Measurements",* IEEE Trans. Automat. Control, τομ. 43, 1998.

[8] *"Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization"*, IEEE Trans. Aerosp. Electron. Sys., τομ. 34, 1997.

[9] P. Sadegh and James Spall, *"Optimal Random Perturbation using a Simultaneous Perturbation Gradient Approximation"*, IEEE Trans. Automat. Control, τομ. 43, 1998.

[10] P. Ioannou, J. Sun, "*Stable and Robust Adaptive Control Systems"*, Englewood Cliffs, Prentice Hall, 1996.

Παράρτημα 1: Ο Αλγόριθμος FDSA που χρησιμοποιήθηκε για την συνάρτηση 4

```
clear all
p=4;
sigma=0.01;
n=1000;
b=eye(p);
a=0.01;
c=10;
A=50;
alpha=0.602;
gamma=0.101;
LLFDmin=9999999999999999;

for i=1:p
   deye=b(:,i);
end

tolerance=2*sigma;
loss='F4';
thetafmiden=[1;2;3;4];
thetafd=thetafmiden;
losspalia=feval(loss,thetafd);
thetafdplus=zeros(p,1);
thetafdminus=zeros(p,1);

for k=1:n
   a_k=a/(k+A)^alpha;
   c_k=c/(k^gamma);
   yminus(i,1)=feval(loss,thetafd);
   for i=1:p
      thetafdplus(i,1)=thetafd(i,1)+c_k*deye(i,1);
      yplus(i,1)=feval(loss,thetafdplus);
      gkap(i,1)=yplus(i,1)-yminus(i,1);
   end

   gkap=gkap/(2*c_k);
   thetafdtest=thetafd;
   thetafd=thetafd-a_k*gkap;
   lossnea=feval(loss,thetafd);
   losspalia=feval(loss,thetafdtest);

   if lossnea > losspalia-tolerance;
      thetafd=thetafdtest;
   else
      losspalia1=losspalia;
      losspalia=lossnea;
   end

   if norm(thetafd) > norm(thetafdtest)
      thetafd=thetafdtest;
```

```
    losspalia=losspalia1;
  end

  LLFD(k)=feval(loss,thetafd);

    if LLFD(k)<LLFDmin
    LLFDmin=LLFD(k);
    epanal=k;
  end

end

LLFDmin
epanal
thetafd
plot(LLFD)
```

Η υπορουτίνα F4

```
function loss=F4(x)

p=4;

 F= ((10^(-5))^(1/2)*(x(1)-1))^2+((10^(-5))^(1/2)*(x(2)-1))^2+((10^(-5))^(1/2)*(x(3)-1))^2 +

    ((10^(-5))^(1/2)*(x(4)-1))^2+(x(1)^2+x(2)^2+x(3)^2+x(4)^2-1/4)^2;

loss=F;
```

Παράρτημα 2: Ο Αλγόριθμος SPSA που χρησιμοποιήθηκε για την συνάρτηση 4

```
clear all
p=4;
sigma=0.01;
a=0.14;
A=100;
c=0.07;
alpha=0.602;
gamma=0.101;
n=1000;
gH_avg=1;
thetamiden=[1;2;3;4];
theta=thetamiden;
tolerance=2*sigma;
LL1min=99999999999999999;
loss=('F4');
losspalia=feval(loss,theta1);
for k=1:n
   a_k=a/(k+A)^alpha;
   c_k=c/(k^gamma);
   gkapeisodou=zeros(1,p);
   for m=1:gH_avg
      delta=2*round(rand(p,1))-1;
      thetaplus=theta+c_k*delta;
      thetaminus=theta-c_k*delta;
      yplus=feval(loss,thetaplus);
      yminus=feval(loss,thetaminus);
      gkap=((yplus-yminus)/(2*c_k))*delta;
      gkapeisodou=((m-1)/m)*gkapeisodou+gkap'/m;
   end

   thetatest=theta;
   theta=theta-a_k*gkapeisodou';
   lossnea=feval(loss,theta);
   losspalia=feval(loss,thetatest);
   if lossnea > losspalia-tolerance
      theta=thetatest;
   else
      losspalia1=losspalia;
      losspalia=lossnea;
   end
   if norm(theta)>norm(thetatest)
      theta=thetatest;
      losspalia=losspalia1;
   end
   LL1(k)=feval(loss,theta);

   if LL1(k)<LL1min
      LL1min=LL1(k);
      epanal=k;
   end
end
theta
LL1min
epanal
plot(LL1)
```

Παράρτημα 3: Ο Προσαρμοστικός Αλγόριθμος Βελτιστοποίησης που χρησιμοποιήθηκε για την συνάρτηση 4

```
clear all
n=5;
g1=0.1;
g2=0.01;
theta=[1;1;1;1;1];
thetab=randn(n,1);
thetabarx=thetab
x1=1;
x2=2;
x3=3;
x4=4;
Fmin=900000000000000000000000000000000000000000000000000;

v(1)=((10^(-5))^(1/2)*(x1-1))^2;
v(2)=((10^(-5))^(1/2)*(x2-1))^2;
v(3)=((10^(-5))^(1/2)*(x3-1))^2;
v(4)=((10^(-5))^(1/2)*(x4-1))^2;
v(n)=(x1^2+x2^2+x3^2+x4^2-1/4)^2;

r=0;
tolerence=0;
F_old=900000000000000000000000000000000000000000000000000;
sit=1;

F=theta(1)*v(1)+theta(2)*v(2)+theta(3)*v(3)+theta(4)*v(4);

for k=1:1000,
   if sit==1
      f=0;
     for i=1:n
        f=f+thetab(i)*v(i);
     end

     thetab_old=thetab;

     for j=1:n
        thetab(j)=thetab(j)+g1*v(j)*(F-f);
     end

        for j=1:n
           if abs(thetab(j)) > 200
              thetab(j)=abs(thetab_old(j))*sign(thetab(j));
           end
        end

     E(k)=F-f;
```

```
    x1=x1-g2*(1/50000*x1-1/50000+4*(x1^2+x2^2+x3^2+x4^2-1/4)*x1);
    x2=x2-g2*(1/50000*x2-1/50000+4*(x1^2+x2^2+x3^2+x4^2-1/4)*x2);
    x3=x3-g2*(1/50000*x3-1/50000+4*(x1^2+x2^2+x3^2+x4^2-1/4)*x3);
    x4=x4-g2*(1/50000*x4-1/50000+4*(x1^2+x2^2+x3^2+x4^2-1/4)*x4);

    v(1)=((10^(-5))^(1/2)*(x1-1))^2;
    v(2)=((10^(-5))^(1/2)*(x2-1))^2;
    v(3)=((10^(-5))^(1/2)*(x3-1))^2;
    v(4)=((10^(-5))^(1/2)*(x4-1))^2;
    v(n)=(x1^2+x2^2+x3^2+x4^2-1/4)^2;

    F_old=F;

    F=0;
    for j=1:n,
       F=F+theta(j)*v(j);
    end

    FF(k)=F;
    FFF(k)=F;

    if F<Fmin
      Fmin=F;
      epanal=k;
      x1b=x1;
      x2b=x2;
      x3b=x3;
      x4b=x4;
      thetabzitoymeno=thetab;
   end

   if F-F_old >= tolerence
     sit=2;
   end


 else
    r=r+1;
    l(r)=k;
    for k1=1:10,
       f=0;
       for i=1:n
          f=f+thetab(i)*v(i);
       end

       thetab_old=thetab;

       for j=1:n
          thetab(j)=thetab(j)+g1*v(j)*(F-f);
       end


       for j=1:n
          if abs(thetab(j)) > 200
            thetab(j)=abs(thetab_old(j))*sign(thetab(j));
```

```
          end
        end

        x1=randn(1);
        x2=randn(1);
        x3=randn(1);
        x4=randn(1);

        v(1)=((10^(-5))^(1/2)*(x1-1))^2;
        v(2)=((10^(-5))^(1/2)*(x2-1))^2;
        v(3)=((10^(-5))^(1/2)*(x3-1))^2;
        v(4)=((10^(-5))^(1/2)*(x4-1))^2;
        v(n)=(x1^2+x2^2+x3^2+x4^2-1/4)^2;

        F=0;
        for j=1:n,
          F=F+theta(j)*v(j);
        end

        FF(k)=F;

     end
     sit=1;
     x1=x1b;
     x2=x2b;
     x3=x3b;
     x4=x4b;
     FFF(k)=FFF(k-1);

  end
end

r
Fmin
epanal
x1b
x2b
x3b
x4b
thetabzitoymeno
x1
x2
x3
x4
thetab
plot(FFF)
```

Παράρτημα 4: Οι Υπορουτίνες που χρησιμοποιήθηκαν από τους

FDSA και SPSA για την πρώτη, δεύτερη και τρίτη συνάρτηση αντίστοιχα.

Για τη πρώτη συνάρτηση

```
function loss=F1(theta)
p=2;
F=(10*(theta(2)-theta(1)^2))^2+(1-theta(1))^2;
loss=F;
```

Για τη δεύτερη συνάρτηση

```
function loss=F2(theta)
p=2;
F= (-13+theta(1)+((5-theta(2))*theta(2)-2)*theta(2))^2+
   (-29+theta(1)+((theta(2)+1)*theta(2)-14)*theta(2))^2;
loss=F;
```

Για τη τρίτη συνάρτηση (n=m=4)

```
function loss=F3(x)
p=4;
F= (x(1)-10*x(2))^2+(5^(1/2)*(x(3)-x(4)))^2+((x(2)-2*x(3))^2)^2+
   (10^(1/2)*(x(1)-x(4))^2)^2;
loss=F;
```

Για τη τρίτη συνάρτηση (n=m=8)

```
function loss=F3b(x)
p=8;
F= (x(1)-10*x(2))^2+(5^(1/2)*(x(3)-x(4)))^2+((x(2)-2*x(3))^2)^2+
   ((10^(1/2)*(x(1)-x(4)))^2)^2+(x(5)-10*x(6))^2+(5^(1/2)*(x(7)-x(8)))^2+
   ((x(6)-2*x(7))^2)^2+((10^(1/2)*(x(5)-x(8)))^2)^2;
loss=F;
```

Για τη τέταρτη συνάρτηση

```
function loss=F4(x)
p=4;
 F=((10^(-5))^(1/2)*(x(1)-1))^2+((10^(-5))^(1/2)*(x(2)-1))^2+
   ((10^(-5))^(1/2)*(x(3)-1))^2+((10^(-5))^(1/2)*(x(4)-1))^2+
   (x(1)^2+x(2)^2+x(3)^2+x(4)^2-1/4)^2;
loss=F;
```