

# Δημιουργία Γενετικού Αλγορίθμου για προσδιορισμό αναλυτικής έκφρασης συνάρτησης

## Τεχνικές Βελτιστοποίησης Project

Φεβρουάριος 2022

### Περίληπτικά:

Στην εργασία αυτή ζητούμενο είναι ο προσδιορισμός μίας αναλυτικής έκφρασης χαμηλής πολυπλοκότητας για μια άγνωστη, αλλά συνεχή συνάρτηση για την οποία έχουμε διαθέσιμες μόνο μετρήσεις εισόδου-εξόδου. Το παραπάνω θα επιτευχθεί μέσω ενός γενετικού αλγορίθμου.

## Εισαγωγικά

Αρχικά, είναι χρήσιμο να σχεδιάσουμε τη γραφική παράσταση της δοσμένης συνάρτησης  $f(u_1, u_2)$  ή  $f(x, y)$ , με  $u_1 \in [-1, 2]$  και  $u_2 \in [-2, 1]$  προκειμένου να έχουμε μία εποπτεία του προβλήματος προσέγγισής της. Ο τύπος της συνάρτησης προς προσέγγιση θα χρησιμοποιηθεί μόνο για παραγωγή δεδομένων εισόδου-εξόδου κατά τη διαδικασία αξιολόγησης των αποτελεσμάτων. Η άγνωστη συνάρτηση θα προσδιοριστεί χρησιμοποιώντας κατάλληλα σχεδιασμένο γραμμικό συνδυασμό γκαουσιανών συναρτήσεων της

μορφής:  $G(u_1, u_2) = e^{-\left(\frac{(u_1 - c_1)^2}{2\sigma_1^2} + \frac{(u_2 - c_2)^2}{2\sigma_2^2}\right)}$

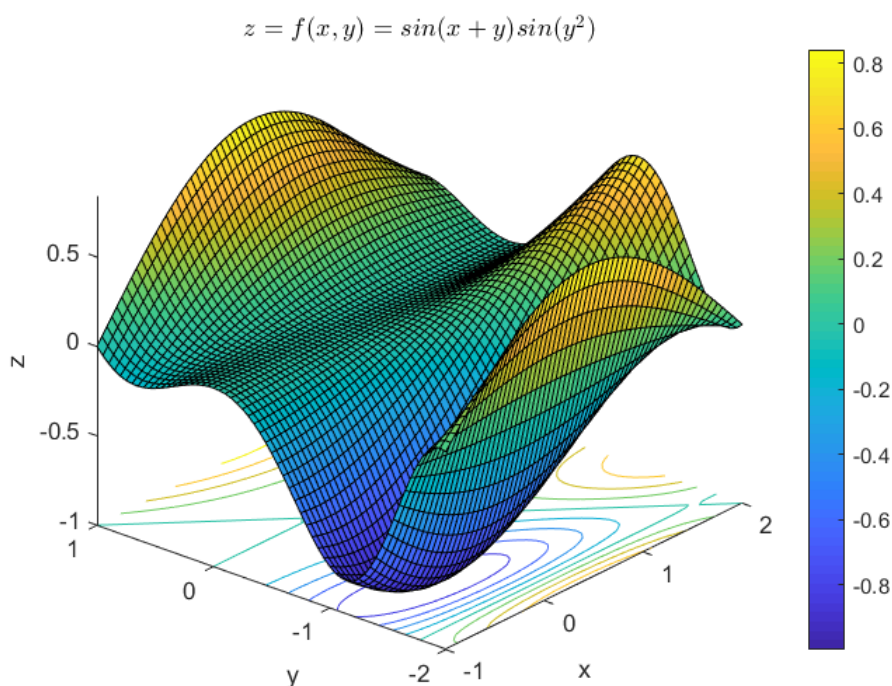


Figure 1: Γραφική παράσταση της  $f(u_1, u_2)$

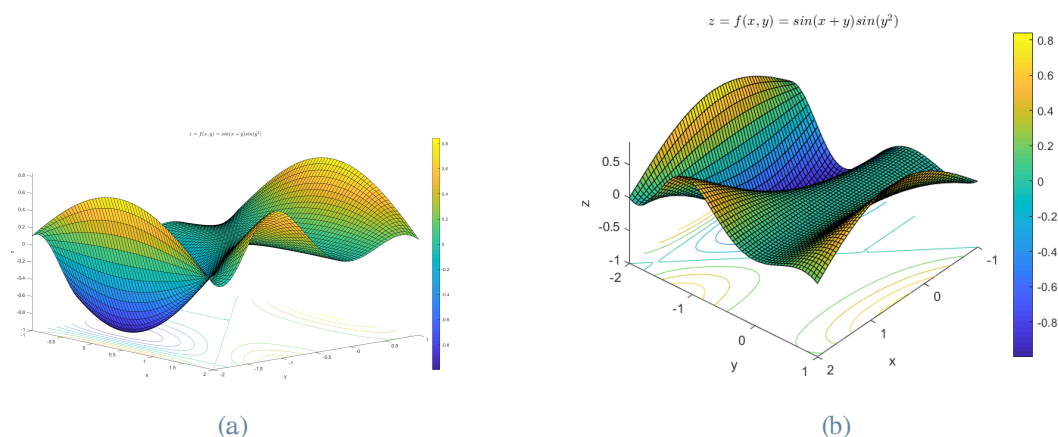


Figure 2: Επιπλέον γραφικές παραστάσεις

## 1. Περιγραφή Αλγορίθμου

Με βάση τις γκαουσιανές συναρτήσεις με τις οποίες πρέπει να προσεγγίσουμε την άγνωστη συνάρτηση, προκύπτει ότι υπάρχουν 5 παράμετροι σε κάθε γκαουσιανή συνάρτηση οι οποίες μπορούν να επηρεάσουν τη μορφή της τελικής λύσης. Αυτές είναι το κέντρο  $(c_1, c_2)$  (2 παράμετροι) της γκαουσιανής, οι τυπικές αποκλίσεις  $(\sigma_1, \sigma_2)$  (2 παράμετροι) και το πλάτος της γκαουσιανής (1 παράμετρος) σε ένα εύρος που καθορίσαμε ως  $(\min(f), \max(f))$ , μιας και δεν έχει νόημα να έχουμε πλάτη γκαουσιανών που να ξεφεύγουν από αυτό το εύρος. Το εύρος τιμών του  $c_1$  θα είναι  $[-1 - 0.5, 2 + 0.5] = [-1.5, 2.5]$ , καθώς  $u_1 \in [-1, 2]$ , δοσμένο από την εκφώνηση, και αντίστοιχα το εύρος τιμών του  $c_2$  θα είναι  $[-2 - 0.5, 1 + 0.5] = [-2.5, 1.5]$ , καθώς  $u_2 \in [-2, 1]$ , δοσμένο επίσης από την εκφώνηση. Η πρόσθεση επιπλέον περιοχής στο εύρος τιμών για τα  $c_1, c_2$  γίνεται προκειμένου να μην έχουμε μεγάλα σφάλματα στα άκρα των περιοχών. Τέλος, οι τυπικές αποκλίσεις παίρνουν τιμές στο διάστημα  $[0.01, 0.75]$ , για να δίνεται η δυνατότητα σε κάποια γκαουσιανή να είναι είτε απότομη είτε πιο διευρυμένη χωρικά (θα σχολιαστεί παρακάτω γιατί δεν επεκτείνουμε τη μέγιστη τιμή του διαστήματος περαιτέρω). Τις παραπάνω 5 παραμέτρους θα τις αποκαλούμε χρωμοσώματα.

Κάθε άτομο έχει καθορισμένα σε αριθμό γονίδια, τα οποία προκύπτουν άμεσα από τις γκαουσιανές που θα χρησιμοποιήσουμε. Για παράδειγμα, έστω 15 γκαουσιανές, τότε τα γονίδια ενός ατόμου θα είναι  $15 \times 5$ , όπου 5 είναι οι παράμετροι κάθε γκαουσιανής, που τις ονομάζουμε και χρωμοσώματα.

Αρχικά δημιουργούμε την πρώτη γενιά αρχικοποιώντας τυχαία (με ομοιόμορφη κατανομή) το γονίδια όλων των ατόμων του πληθυσμού, μέσα στα όρια που περιγράφηκαν παραπάνω. Για κάθε επόμενη γενιά, θα μεταφέρεται από το 10% των καλύτερων ατόμων της προηγούμενης, το 60% θα προέρχεται από δύο διαδικασίες μετάλλαξης (mutation 30%, mutation2 30%), οι οποίες θα περιγραφούν παρακάτω, το 20% από διαδικασία διασταύρωσης (crossover), και τέλος το 10% από άτομα που δημιουργούνται τυχαία εξ' αρχής.

Επιλέγουμε να έχουμε έναν πληθυσμό 100 ατόμων ανά γενιά, αφενός για να μην είναι πολύ μικρός, αφετέρου για να μην καθυστερήσει πολύ ο αλγόριθμος χρονικά.

Θα ξεκινήσουμε θεωρώντας αυτές τις παραμέτρους για τις ακόλουθες προσομοιώσεις, αφού αυτές ήταν οι τελικώς επιλεγμένες όταν τελείωσε η συγγραφή του κώδικα του αλγορίθμου και προέκυψαν ως καλοί συμβιβασμοί για να πάρουμε πολύ καλά αποτελέσματα. Θα προσπαθήσουμε να δικαιολογήσουμε αυτές τις επιλογές στο πλαίσιο του εφικτού, γνωρίζοντας πάντα ότι οι τυχαιότητες που εισάγονται κατά την εκτέλεση του αλγορίθμου μπορούν να επηρεάσουν καθοριστικά το αποτέλεσμα του. Για παράδειγμα, η πρώτη γενιά μπορεί εξ' αρχής να δημιουργηθεί με πολύ μικρό σφάλμα συγκρινόμενη με κάποιες άλλες υπό τις ίδιες συνθήκες δημιουργίας.

Με βάση τα παραπάνω, συνεχίζουμε να παράγουμε νέες γενιές, με μέγιστη για όλες τις παρακάτω προσομοιώσεις τη 10000n, ενώ έχουμε θέσει ως κατώφλι σφάλματος 0.0001, όποτε αυτό ξεπερνιέται

(προς τα κάτω), ο αλγόριθμος τερματίζει, και θεωρούμε ότι έχουμε πάρει μία πολύ καλή λύση. Παρατήρηση: θα μπορούσαμε να κωδικοποιήσουμε τα γονίδια (και κατά συνέπεια τα χρωμοσώματα) σε bits με τιμές 0 και 1, ωστόσο αυτό δε φαίνεται να είναι ιδιαίτερα χρήσιμο στη δική μας περίπτωση, μάλιστα μας προσθέτει και επιπλέον υπολογιστικό φόρτο. Οποιαδήποτε αλλαγή θα πρέπει να γίνει στα χρωμοσώματα γίνεται πολύ εύκολα χρησιμοποιώντας πραγματικούς αριθμούς, χωρίς την ανάγκη για κωδικοποίησή τους σε 0 και 1.

## 2. Συναρτήσεις

### 2.1. `calculateFitnessFunction`

Προκειμένου να καθορίσουμε πόσο καλό είναι ένα άτομο μίας γενιάς, και να μπορέσουμε να τη συγκρίνουμε με τα υπόλοιπα (και τις υπόλοιπες γενιές γενικά), χρειάζεται να ορίσουμε ένα σφάλμα που να μας δίνει το κατά πόσο προσεγγίζουμε την άγνωστη συνάρτηση καλά. Έτσι, για τη συνάρτηση σφάλματος θα χρησιμοποιήσουμε το μέσο τετραγωνικό σφάλμα ανάμεσα στην προτεινόμενη προσέγγιση και στην πραγματική συνάρτηση. Το επιθυμητό είναι να έχουμε όσο το δυνατόν μικρότερο σφάλμα. Η συνάρτηση `calculateFitnessFunction` υλοποιεί την παραπάνω λογική.

### 2.2. `createRandomGenes`

Η `calculateFitnessFunction` δημιουργεί ένα σετ τυχαίων γονιδίων, με ομοιόμορφη κατανομή σε ένα δοσμένο διάστημα. Χρησιμοποιείται τόσο στην αρχικοποίηση της πρώτης γενιάς, όσο και στην εισαγωγή νέων τυχαίων γονιδίων σε κάθε γενιά. Αυτή η πρακτική, δηλαδή η εισαγωγή εκ νέου τυχαίων γονιδίων, θεωρείται χρήσιμη προκειμένου να δίνεται η δυνατότητα στην κάθε γενιά για εύρεση ακόμα καλύτερης λύσης και να μην εγκλωβίζεται ο αλγόριθμος σε μία απλά καλή λύση, ενώ μπορεί να βρει καλύτερες.

### 2.3. `mutate`

Η `mutate` πραγματοποιεί μετάλλαξη σε κάποιο χρωμόσωμα του γονιδίου με κάποια δοσμένη (και μικρή) πιθανότητα. Τα νέα χρωμοσώματα θα είναι τυχαία, και θα ακολουθούν ομοιόμορφη κατανομή στα αρχικά προκαθορισμένα διαστήματα.

### 2.4. `mutate2`

Η `mutate2` πραγματοποιεί μετάλλαξη σε κάποιο χρωμόσωμα του γονιδίου με κάποια δοσμένη (και μικρή) πιθανότητα. Τα νέα χρωμοσώματα δε θα είναι τυχαία, αλλά θα είναι μία μικρή μετατόπιση των προηγούμενων χρωμοσωμάτων. Η τιμή της μετατόπισης θα καθορίζεται τυχαία με βάση την ομοιόμορφη κατανομή σε ένα διάστημα γύρω από την τιμή του αρχικού χρωμοσώματος. Αυτή η συνάρτηση θα εφαρμόζεται στα καλύτερα άτομα (πχ στα 30 καλύτερα άτομα αν ο πληθυσμός είναι 100 και το ποσοστό για αυτή τη μετάλλαξη το ορίσουμε ίσο με 30%) και έχει ως στόχο να ψάξει σε γειτονίες καλών τιμών ώστε να βρει ακόμα καλύτερες τιμές.

### 2.5. `crossover`

Η συνάρτηση `crossover` πραγματοποιεί διασταύρωση μεταξύ των καλύτερων γονιδίων της προηγούμενης γενιάς με δύο τρόπους. Ο πρώτος συνδυάζει ένα τμήμα των χρωμοσωμάτων του πρώτου γονέα μέχρι ένα σημείο (τυχαίο από 1 μέχρι το μήκος του γονιδίου - 1) και το υπόλοιπο αντίστοιχο τμήμα

το παίρνει από τον δεύτερο γονέα. Ο δεύτερος τρόπος υπολογίζει τον μέσο όρο των αντίστοιχων χρωμοσωμάτων των δύο γονέων.

### 3. Δοκιμές του αλγορίθμου

Κατά τη διάρκεια της υλοποίησης του αλγορίθμου έγιναν πολυάριθμα τεστ προκειμένου να προσδιοριστούν μέχρι έναν ικανοποιητικό βαθμό οι παράμετροι του αλγορίθμου και να βελτιστοποιηθούν ώστε να έχουμε ένα καλό τελικό αποτέλεσμα. Είναι σαφές ότι όσες περισσότερες δοκιμές τρέξουμε με διαφορετικές επιλογές στις παραμέτρους, είναι λογικό να καταλήγουμε σε όλα και καλύτερες επιλογές. Ωστόσο, δε μπορούμε να τρέξουμε για πάντα δοκιμές, και έτσι θα περιοριστούμε σε έναν πεπερασμένο αριθμό αυτών, που είναι όμως ικανός να μας δώσει μία πολύ καλή λύση. Περαιτέρω βελτιστοποίηση είναι σαφώς δυνατή, ωστόσο ο κόπος που θα καταβληθεί για αυτή δε θα αντικατοπτρίζεται στο ποσοστό βελτίωσης της λύσης, και έτσι δεν αξίζει να ασχοληθούμε παραπάνω.

Παρακάτω παρατίθενται κάποιες ενδεικτικές προσομοιώσεις με τους παράγοντες που θεωρήθηκαν ότι επηρεάζουν περισσότερο την καλή εξέλιξη του αλγορίθμου.

Ας σημειωθεί ότι κάθε προσομοίωση θα πραγματοποιείται 3 φορές και θα παίρνουμε τον μέσο όρο για κάθε σφάλμα. Αυτό γίνεται προκειμένου να εξαλείψουμε τις τυχαιότητες που υπεισέρχονται αναπόφευκτα κατά την εκτέλεση του αλγορίθμου.

Όλες οι παρακάτω προσομοιώσεις έγιναν δεδομένων των παραπάνω τιμών, όπως αναφέρθηκαν κατά την αρχική περιγραφή του αλγορίθμου. Όπου γίνονται αλλαγές, για τις ανάγκες της κάθε δοκιμής, αναφέρονται ρητά.

#### 3.1. Ποσοστό μετάλλαξης

Αρχικά θα ασχοληθούμε με τα ποσοστά μετάλλαξης, θα βάλουμε και για τη `mutation` και τη `mutation2` τις ίδιες παρακάτω τιμές, και θα πάρουμε τα σφάλματα που φαίνονται στον πίνακα:

Table 1: Επίδραση του ποσοστού μετάλλαξης στο ελάχιστο σφάλμα που μπορεί να επιτευχθεί

πιθανότητα μετάλλαξης	p = 0.001	p = 0.01	p = 0.1	p = 0.5
συνολικό σφάλμα	0.000529	0.000432	0.000633	0.00572

Με βάση των παραπάνω πίνακα, φαίνεται πως έχουμε τις καλύτερες επιδόσεις όταν η πιθανότητα μετάλλαξης είναι μεταξύ του 1% και 10%. Κατά συνέπεια, οι επιλογές μας θα κινηθούν ανάμεσα σε αυτές τις δύο τιμές, και συγκεκριμένα πιο κοντά στο 1%.

#### 3.2. Αριθμός Γκαουσιανών Συναρτήσεων

Στον παρακάτω πίνακα, φαίνεται η επίδραση του αριθμού των γκαουσιανών που χρησιμοποιούνται για την προσέγγιση της συνάρτησης.

αριθμός γκαουσιανών	σφάλμα	χρόνος εκτέλεσης
7	0.001790	5 min 13 sec
10	0.0010975	6 min 5 sec
12	0.000375	7 min 24 sec
15	0.000321	9 min 20 sec

Όπως θα περιμέναμε, όσο αυξάνεται ο αριθμός των γκαουσιανών τόσο μικρότερο γίνεται σφάλμα. Ωστόσο, παρατηρούμε πως ακόμα και με λιγότερες από τις 15 γκαουσιανές (12 εδώ), που είναι το



όριο που τέθηκε από την εκφώνηση, μπορούμε να πετύχουμε ένα πολύ μικρό σφάλμα, το οποίο δε φαίνεται να παρουσιάζει μεγάλη απόκλιση από τις 15 (δεν αποκλείεται όμως σε κάποιο άλλο σετ προσομοιώσεων να είχαμε ένα διαφορετικό, αλλά κοντινό, αποτέλεσμα). Έτσι, η επιλογή 12 μόνο γκαουσιανών τελικά, δε θα μας έδινε σημαντικά χειρότερη προσέγγιση, ενώ θα μείωνε και την πολυπλοκότητα του αλγορίθμου.

Παράλληλα, παρατηρούμε πως όσο αυξάνονται οι γκαουσιανές αυξάνεται και ο χρόνος που χρειάζεται ο αλγόριθμος για να εκτελεστεί, ως άμεση συνέπεια του μεγαλύτερου υπολογιστικού φόρτου.

### 3.3. Πληθυσμός

Σύμφωνα με τον παρακάτω πίνακα, αγνοώντας τις όποιες τυχαιότητες υπεισέρχονται κατά την εκτέλεση του αλγορίθμου, συμπεραίνουμε πως έχοντας έναν πληθυσμό μόλις 100 ατόμων μπορούμε να πάρουμε ικανοποιητικά αποτελέσματα, σε μικρό σχετικά χρόνο. Εξάλλου, με την αύξηση των ατόμων (κατά συνέπεια και του υπολογιστικού χρόνου και φόρτου) δε φαίνεται να παίρνουμε αντίστοιχα καλή βελτίωση στο σφάλμα - το αντίθετο μάλιστα συμβαίνει σύμφωνα με τις δοκιμές που έγιναν. Έτσι, είναι πολύ λογικό να επιλέξουμε έναν πληθυσμό 100 ατόμων.

πληθυσμός	σφάλμα	χρόνος εκτέλεσης
50	0.000488	4 min 42 sec
100	0.000368	9 min 33 sec
150	0.000554	13 min 28 sec
200	0.000454	18 min 20 sec

### 3.4. Ποσοστό καλύτερων γονιδίων

Δοκιμάζοντας διάφορα ποσοστά μεταφοράς των καλύτερων γονιδίων στην επόμενη γενιά, διαπιστώνουμε - όπως ίσως θα περιμέναμε - ότι η αύξηση αυτού του ποσοστού μεταφοράς στην πραγματικότητα χειροτερεύει την επίδοση της επόμενης γενιάς καθώς ο αλγόριθμος εγκλωβίζεται σε ορισμένες καλές λύσεις που έχει βρει μέχρι στιγμής με αποτέλεσμα να μην ψάχνει για νέες και ενδεχομένως καλύτερες. Φυσικά, κάποιο μικρό ποσοστό μεταφοράς των καλύτερων γονιδίων στην επόμενη γενιά είναι πολύ χρήσιμο προκειμένου να μη χάνονται κάποιες καλές λύσεις που έχουν ήδη βρεθεί. Συνεπώς, ένα καλό ποσοστό μεταφοράς φαίνεται να είναι το 10% - 30%, το οποίο σε συνδυασμό με κάποιες μεταλλάξεις και διασταυρώσεις, μπορεί και μας δίνει ακόμα καλύτερες λύσεις. Τελικά, θα επιλέξουμε το 10% δεδομένου πως τα καλύτερα άτομα μεταφέρονται στην επόμενη γενιά και μέσω διαδικασιών μετάλλαξης και διασταύρωσης.

Παρατήρηση: για κάθε ποσοστό που δοκιμάζεται, τα ποσοστά του πληθυσμού που προκύπτουν από τους άλλους τρόπους (μεταλλάξεις, διασταυρώσεις, νέα τυχαία άτομα) είναι ίσα μεταξύ τους, και μοιράζονται το υπόλοιπο της αφαίρεσης του ποσοστού μεταφοράς από τη μονάδα, ενώ το ποσοστό των τυχαίων νέων γονιδίων δεν ξεπερνά ποτέ το 10%.

ποσοστό μεταφοράς καλύτερων γονιδίων	70%	50%	30%	10%
σφάλμα	0.000496	0.000642	0.000469	0.000279

### 3.5. Διάστημα τιμών διασποράς

Γίνεται εύκολα αντιληπτό πως αν αλλάξουμε την περιοχή που μπορεί να πάρει τιμές η διασπορά, μπορούμε να επηρεάσουμε καθοριστικά και την επίδοση του αλγορίθμου. Για παράδειγμα, αν αφήσουμε το πάνω όριο του διαστήματος να πάρει μεγάλες τιμές (πχ 5) παρατηρούμε από τη γραφική απεικόνιση της συνάρτησης κιάλας ότι η λύση αδυνατεί να προσεγγίσει τη δοσμένη συνάρτηση σε

πλάτος, αν και η μορφή τους παραμένει κατά κύριο λόγο πολύ όμοια. Για αυτό το λόγο, παρακάτω παρατίθενται ορισμένες δοκιμές που έχουν ως σκοπό την εύρεση ενός διαστήματος για τη διασπορά, ικανό να μας δώσει ένα τελικά πολύ μικρό σφάλμα.

Έτσι, αρχικά κρατάμε το κάτω όριο του διαστήματος σταθερό και ίσο με 0.01 και μεταβάλλουμε το άνω διαδοχικά.

διαστήματα [0.01 x]	[0.01 5]	[0.01 2]	[0.01 1]	[0.01 0.75]	[0.01 0.5]	[0.01 0.3]
σφάλμα	0.0018815	0.001153	0.000502	0.000328	0.000373	0.001665

Από τον παραπάνω πίνακα, μπορούμε να δούμε πως το σφάλμα μειώνεται για μέγιστες τιμές της διασποράς κοντά στο 0.75. Άρα,  $\sigma_{\max} = 0.75$  αποτελεί μια καλή επιλογή.

Έπειτα, κρατάμε το άνω όριο του διαστήματος σταθερό και ίσο με 0.75 και μεταβάλλουμε το κάτω διαδοχικά.

διαστήματα [x 0.75]	[0.001 0.75]	[0.005 0.75]	[0.1 0.75]
σφάλμα	0.000245	0.000258	0.000569

Από τον παραπάνω πίνακα, μπορούμε να δούμε πως το σφάλμα μειώνεται για ελάχιστες τιμές της διασποράς κοντά στο 0.001. Άρα,  $\sigma_{\min} = 0.001$  αποτελεί μια καλή επιλογή.

### 3.6. Τελικές επιλογές παραμέτρων και εκτενή πειράματα

Με βάση όλες τις παραπάνω δοκιμές, μπορούμε να καταλήξουμε στις παρακάτω επιλογές παραμέτρων (που συμφωνούν με τις αρχικές παραμέτρους που προτάθηκαν):

1. πιθανότητα μετάλλαξης = 0.003
2. πιθανότητα μετάλλαξης2 = 0.005
3. αριθμός γκαουσιανών = 12
4. πληθυσμός = 100
5. ποσοστό μεταφοράς καλύτερων γονιδίων στην επόμενη γενιά = 10 %
6. διάστημα τιμών διασποράς = [0.001 0.75]

Παρακάτω φαίνονται όλες οι επιλεγμένοι παράμετροι για να τρέξουμε ένα εκτενές πείραμα:

```

numberOfGaussians = 12;
maxGenerations = 40000;
populationSize = 100;
mutationProbability = 0.03;
mutation2Probability = 0.05;
centerRangeX = [-1.5 2.5];
centerRangeY = [-2.5 1.5];
sigmaRange = [0.001 0.75];
percentageOfGenesToBeTranfered = 0.1;
percentageOfGenesToBeMutated = 0.3;
percentageOfGenesToBeMutated2 = 0.3;
percentageOfGenesToBeCrossovered = 0.2;
percentageOfRandomGenes = 0.1;
minErrorToAchieve = 0.0001;
numberOfRegions = 10;
numberOfTests = 1000;
wayToCreateTestingRegion = "uniform";

```

Μετά από 29 λεπτά και 16 δευτερόλεπτα έχουμε σφάλμα ίσο με 0.000276 στην 27097η γενιά. Τα αποτελέσματα φαίνονται οπτικά παρακάτω:

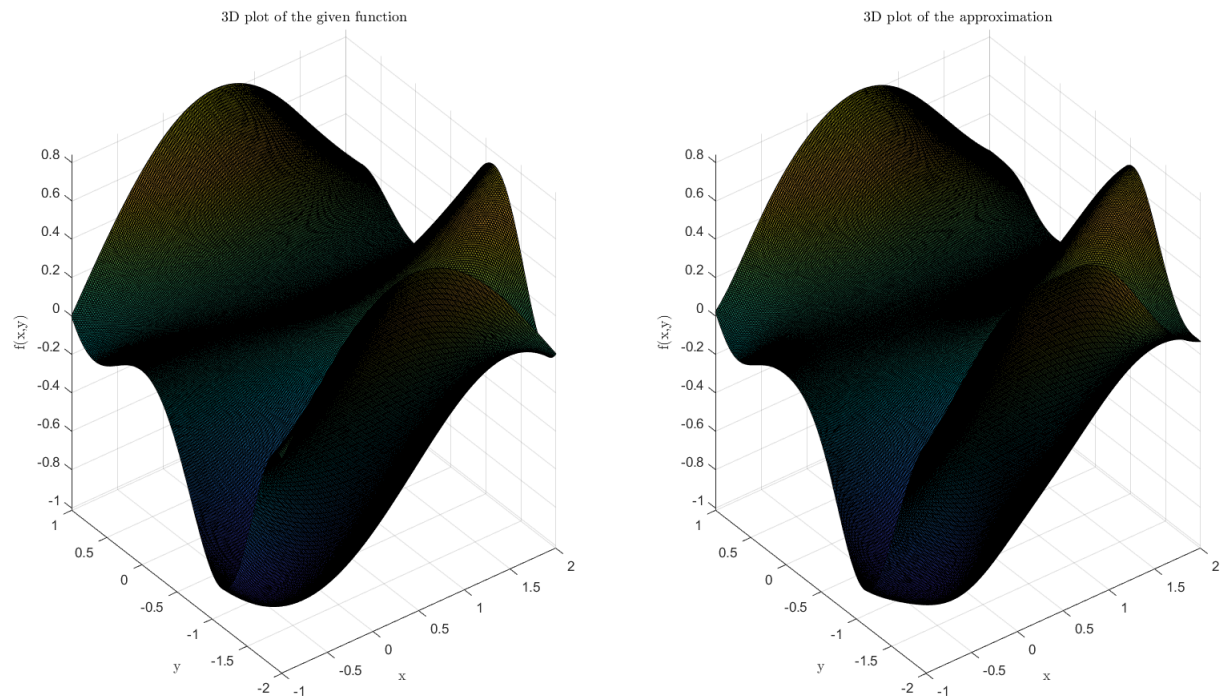


Figure 3: Σύγκριση της δοσμένης συνάρτησης και της προσέγγισής της

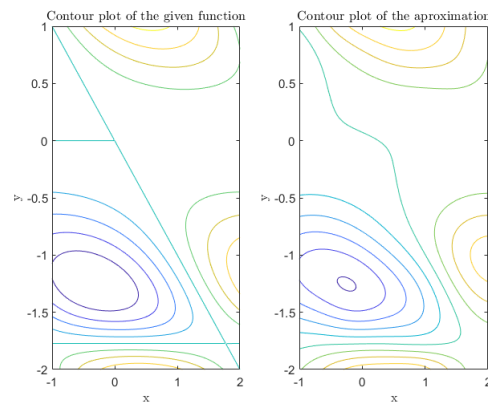


Figure 4: Contour plots της δοσμένης συνάρτησης και της προσέγγισής της

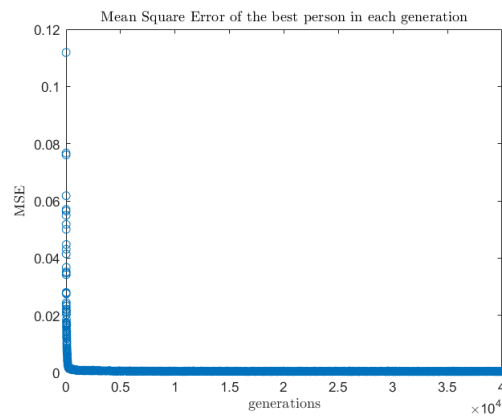


Figure 5: Μέσο τετραγωνικό σφάλμα του καλύτερου ατόμου ανά γενιά

Έτσι, η αναλυτική έκφραση θα είναι η εξής:

$$\hat{f}(u_1, u_2) = \sum_{i=1}^{12} A_i e^{-\left(\frac{(u_1 - c_{1i})^2}{2\sigma_{1i}^2} + \frac{(u_2 - c_{2i})^2}{2\sigma_{2i}^2}\right)}$$

όπου  $A_i$  το πλάτος κάθε γκαουσιανής.

Στον παρακάτω πίνακα φαίνονται όλες οι παράμετροι:

αριθμός γκαουσιανής	$c_1$	$c_2$	$\sigma_1$	$\sigma_2$	πλάτος
1	0.8061	1.0474	0.7352	0.4078	0.7581
2	0.6523	-1.3934	0.6963	0.3358	-0.6870
3	-0.0492	-2.4861	0.4293	0.5771	0.3793
4	-1.0864	0.7708	0.3334	0.3154	-0.1849
5	-0.8779	-1.1109	0.6754	0.3840	-0.7819
6	1.7059	0.5759	0.4091	0.2039	0.1092
7	-0.0662	-2.2090	0.6536	0.4348	0.7739
8	-0.1218	-1.4990	0.4345	0.4019	-0.4736
9	1.9821	-1.0687	0.7500	0.3733	0.7648
10	0.9484	-2.0447	0.4693	0.1538	0.5989
11	-0.2019	1.1714	0.5724	0.2189	0.4433
12	0.7370	0.4632	0.7500	0.2840	-0.0874

Μπορούμε να τρέξουμε ακόμα ένα πείραμα με 15 γκαουσιανές και μέγιστη γενιά την 100000.

Παρακάτω φαίνονται όλες οι επιλεγμένοι παράμετροι, οι αλλαγές σε σχέση με πριν είναι με **bold**:

```

numberOfGaussians = 15;
maxGenerations = 100000;
populationSize = 100;
mutationProbability = 0.03;
mutation2Probability = 0.05;
centerRangeX = [-1.5 2.5];
centerRangeY = [-2.5 1.5];
sigmaRange = [0.001 0.75];
percentageOfGenesToBeTranfered = 0.1;
percentageOfGenesToBeMutated = 0.3;
percentageOfGenesToBeMutated2 = 0.3;
percentageOfGenesToBeCrossovered = 0.2;
percentageOfRandomGenes = 0.1;
minErrorToAchieve = 0.0001;
numberOfRegions = 10;
numberOfTests = 1000;
wayToCreateTestingRegion = "uniform";

```

Μετά από 80 λεπτά και 9 δευτερόλεπτα έχουμε σφάλμα ίσο με 0.000144 στη 89995η γενιά. Τα αποτελέσματα φαίνονται οπτικά παρακάτω:



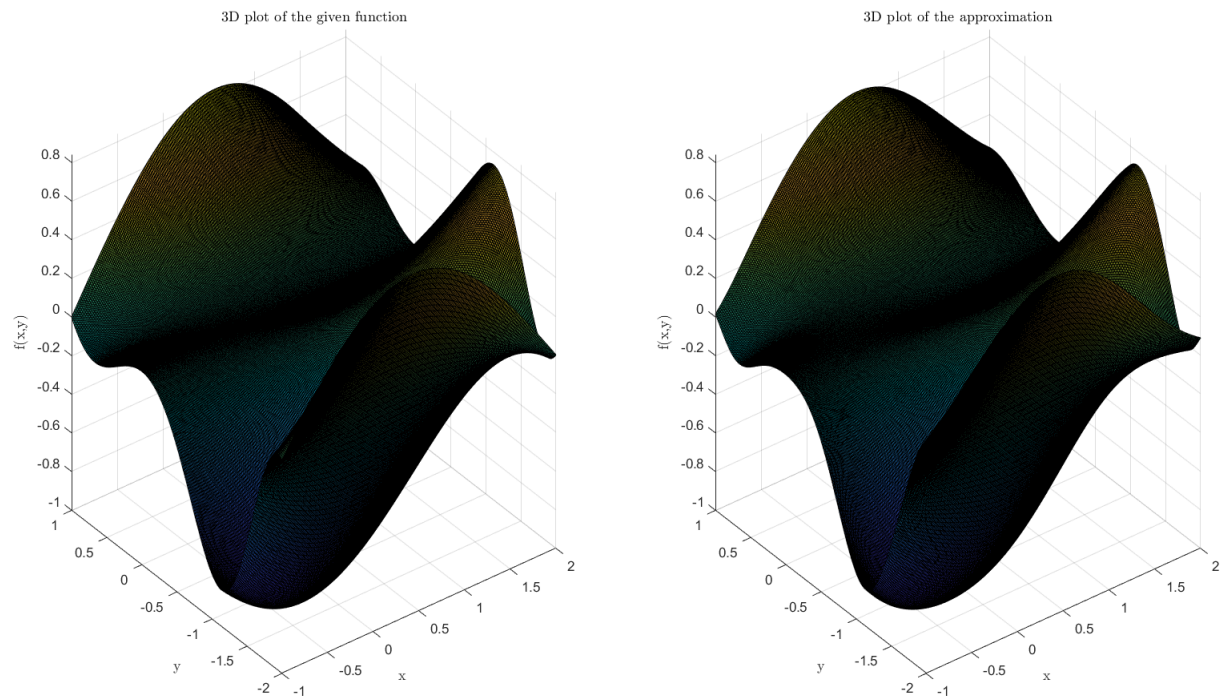


Figure 6: Σύγκριση της δοσμένης συνάρτησης και της προσέγγισής της

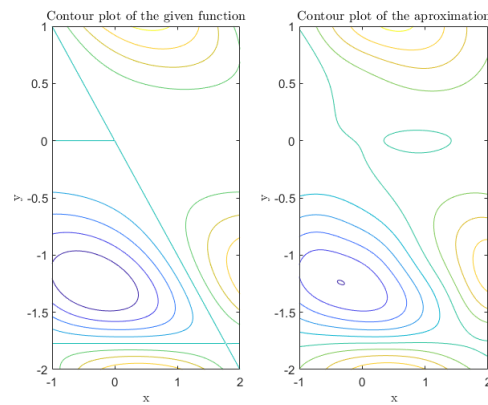


Figure 7: Contour plots της δοσμένης συνάρτησης και της προσέγγισής της

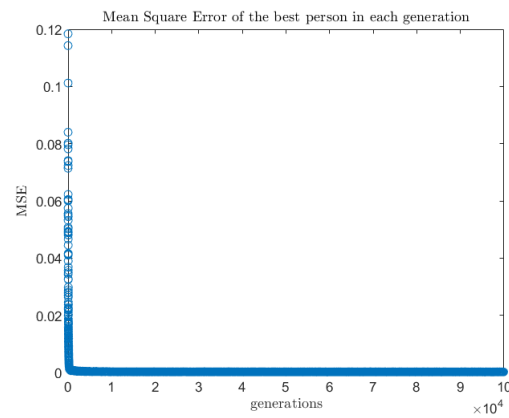


Figure 8: Μέσο τετραγωνικό σφάλμα του καλύτερου ατόμου ανά γενιά

Έτσι, η αναλυτική έκφραση θα είναι η εξής:

$$\hat{f}(u_1, u_2) = \sum_{i=1}^{15} A_i e^{-\left(\frac{(u_1 - c_{1i})^2}{2\sigma_{1i}^2} + \frac{(u_2 - c_{2i})^2}{2\sigma_{2i}^2}\right)}$$

Στον παρακάτω πίνακα φαίνονται όλες οι παράμετροι:

αριθμός γκαουσιανής	$c_1$	$c_2$	$\sigma_1$	$\sigma_2$	πλάτος
1	0.1836	-1.4518	0.5765	0.3554	-0.2746
2	0.7081	-1.5557	0.6370	0.2754	-0.5836
3	0.7513	-2.3324	0.7500	0.5946	0.8415
4	-0.0924	-1.0908	0.6753	0.4045	-0.4062
5	1.1156	0.8858	0.7126	0.4287	0.4457
6	-0.7503	-1.1016	0.6586	0.3733	-0.7136
7	-1.1722	0.7897	0.4150	0.2998	-0.2309
8	0.1684	1.0820	0.7500	0.2449	0.6261
9	0.2741	-0.5419	0.6547	0.3056	0.1139
10	-0.3466	-0.9296	0.4107	0.2240	0.1844
11	1.0336	0.1714	0.6326	0.3193	-0.0907
12	1.7167	-0.8331	0.6191	0.3315	0.3339
13	-0.0872	-2.0182	0.7500	0.2373	0.4624
14	2.1259	-1.2108	0.4112	0.3147	0.5681
15	1.8645	-1.7954	0.7500	0.1696	-0.2177

Από τα παραπάνω μπορούμε εύκολα να συμπεράνουμε πως το τελικό σφάλμα δεν έχει σημαντική απόκλιση όταν χρησιμοποιούμε 12 γκαουσιανές αντί για 15, και έτσι η χρήση 12 γκαουσιανών είναι μία αξιόπιστη επιλογή που μας προσφέρει παράλληλα πλεονεκτήματα λιγότερου υπολογιστικού φόρτου και χαμηλότερης πολυπλοκότητας συνάρτησης. Επιπλέον, είναι εμφανές από τη γραφική παράσταση του σφάλματος πως μετά από κάποιο σημείο η ελάττωση του σφάλματος είναι αμελητέα σε σχέση με το χρόνο που έχουμε να περιμένουμε. Επομένως, δε χρειάζεται να τρέξει κανείς στο συγκεκριμένο πρόβλημα πάνω από 10000 ή 20000 γενιές για να πάρει ένα καλό αποτέλεσμα. Η υπερβολή να τρέξουμε 40000 και 100000 γενιές και τα αποτελέσματά τους δικαιολογούν την παραπάνω πρόταση.

## 4. Συμπεράσματα και τελική πρόταση έκφρασης συνάρτησης

### 4.1. Τελική πρόταση έκφρασης συνάρτησης

Έχοντας όλα τα παραπάνω υπόψιν, μπορούμε να προτείνουμε με συνάρτηση με 12 γκαουσιανές και τις παρακάτω παραμέτρους:

```
numberOfGaussians = 12;  
maxGenerations = 10000;  
populationSize = 100;  
mutationProbability = 0.03;  
mutation2Probability = 0.05;  
centerRangeX = [-1.5 2.5];  
centerRangeY = [-2.5 1.5];  
sigmaRange = [0.001 0.75];  
percentageOfGenesToBeTranfered = 0.1;  
percentageOfGenesToBeMutated = 0.3;  
percentageOfGenesToBeMutated2 = 0.3;  
percentageOfGenesToBeCrossovered = 0.2;  
percentageOfRandomGenes = 0.1;  
minErrorToAchieve = 0.0001;  
numberOfRegions = 10;  
numberOfTests = 1000;  
wayToCreateTestingRegion = "uniform";
```

Μετά από 8 λεπτά και 18 δευτερόλεπτα έχουμε σφάλμα ίσο με 0.000305 στην 7018η γενιά. Τα αποτελέσματα φαίνονται οπτικά παρακάτω:

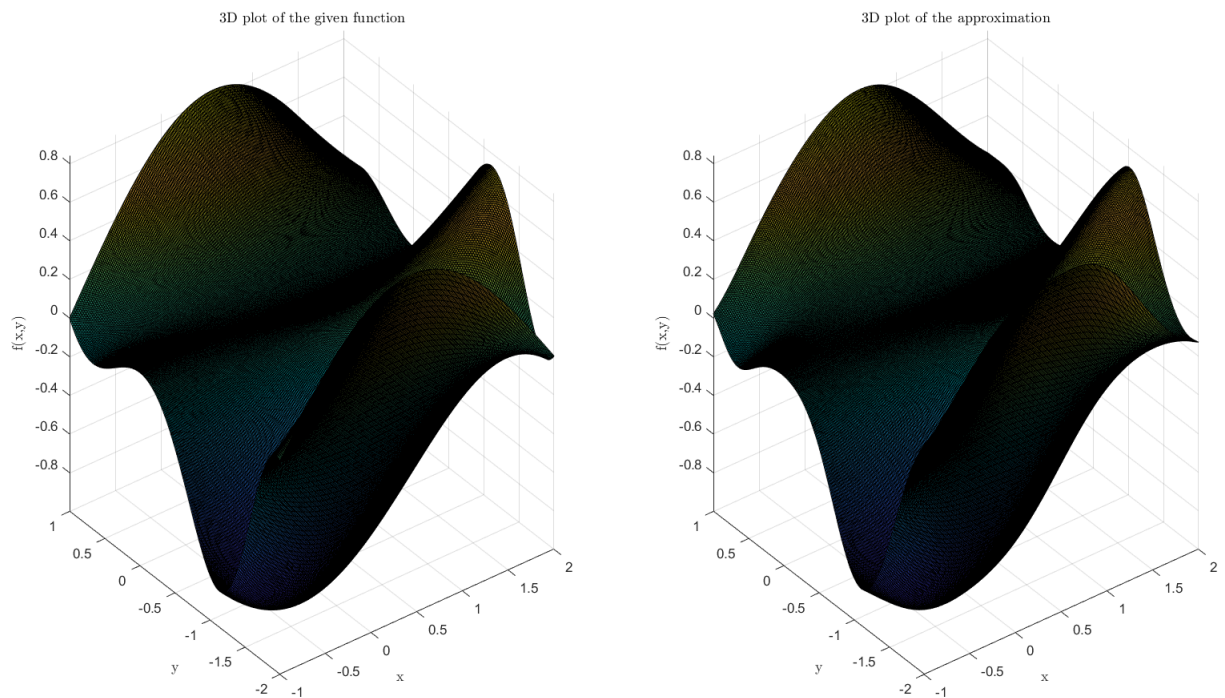


Figure 9: Σύγκριση της δοσμένης συνάρτησης και της προσέγγισής της

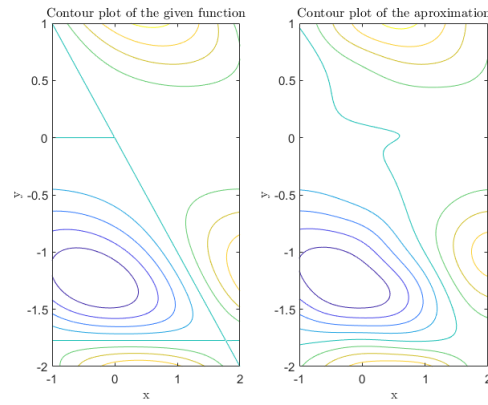


Figure 10: Contour plots της δοσμένης συνάρτησης και της προσέγγισής της

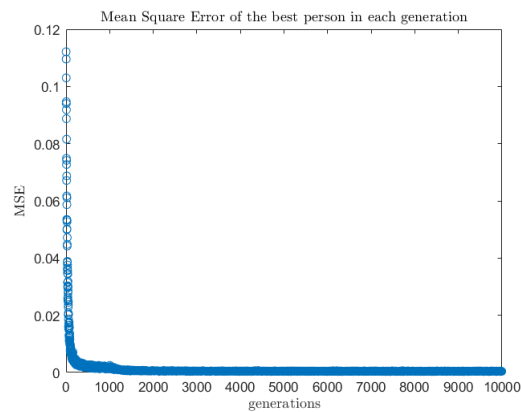


Figure 11: Μέσο τετραγωνικό σφάλμα του καλύτερου ατόμου ανά γενιά

Έτσι, η αναλυτική έκφραση θα είναι η εξής:

$$\hat{f}(u_1, u_2) = \sum_{i=1}^{12} A_i e^{-\left(\frac{(u_{1i}-c_{1i})^2}{2\sigma_{1i}^2} + \frac{(u_{2i}-c_{2i})^2}{2\sigma_{2i}^2}\right)}$$

όπου  $A_i$  το πλάτος κάθε γκαουσιανής.

Στον παρακάτω πίνακα φαίνονται όλες οι παράμετροι:

αριθμός γκαουσιανής	$c_1$	$c_2$	$\sigma_1$	$\sigma_2$	πλάτος
1	1.7599	0.6203	0.4089	0.2111	0.0964
2	0.9349	1.1151	0.6819	0.4344	0.6575
3	0.0352	-2.1546	0.6718	0.3744	0.8394
4	2.0606	-1.0384	0.6760	0.3646	0.7231
5	-1.2461	-0.9956	0.6354	0.3554	-0.5928
6	0.9930	-2.0832	0.4703	0.1400	0.5566
7	0.2149	-1.4238	0.7500	0.2963	-0.9231
8	-0.5318	-1.1249	0.4024	0.3529	-0.2860
9	0.0553	-0.9066	0.3838	0.2220	-0.2302
10	-1.1021	0.7463	0.3447	0.2784	-0.2079
11	0.8974	0.1697	0.7141	0.1698	-0.0404
12	-0.0284	1.0385	0.6478	0.2121	0.4471

Τα αποτελέσματά παραπάνω δείχνουν, ότι μπορούμε όντως με μια όχι υψηλής πολυπλοκότητας συνάρτηση να πάρουμε πολύ καλά αποτελέσματα, και μάλιστα σε πολύ μικρό χρόνο.

Παρατήρηση: Ο αλγόριθμος είναι γενικός και να δεν εξαρτάται από την εκάστοτε συνάρτηση και τις ιδιότητες της, αφού κάθε συνάρτηση έχει τα δικά της χαρακτηριστικά. Επιπλέον, η fitness function δεν εκτιμάει την ποιότητα της προσέγγισης της συνάρτησης με βάση τα ίδια σταθερά σημεία, αλλά επιλέγει κάθε φορά εξ' ολοκλήρου νέα σε όλα το πεδίο που θέλουμε να δούμε την ποιότητα της προσέγγισης (επειδή `wayToCreateTestingRegion = "uniform"` και όχι `"linear"`). Το τελευταίο είναι ιδιαίτερα χρήσιμο προκειμένου να μην κάνουμε overfit στη δοσμένη συνάρτηση.

Για άλλες συναρτήσεις θα χρειαστούν μικρές αλλαγές, όπως στα όρια των τιμών των γονιδίων, ενδεχομένως στις διασπορές κ.ο.κ.

## 4.2. Περαιτέρω βελτιστοποίηση

Ο αλγόριθμος που προτάθηκε φαίνεται πως βγάξει ικανοποιητικά αποτελέσματα, ωστόσο είναι σαφές ότι πάντα υπάρχουν περιθώρια βελτιστοποίησης. Έτσι, κάτι που θα μπορούσε να υλοποιηθεί επιπλέον, θα ήταν μια δυναμική συμπεριφορά του αλγορίθμου. Για παράδειγμα, θα μπορούσε ο αλγόριθμος σε κάποια στιγμή που θα ορίζαμε να αλλάζει ποσοστά ατόμων από κάθε κατηγορία για την επόμενη γενιά, ποσοστά μετάλλαξης κλπ. Έτσι, θα μπορούσε όταν έχει κάτω από ένα κατώφλι σφάλματος, να αυξάνει το ποσοστό μεταφοράς των καλύτερων γονιδίων στην επόμενη γενιά κ.ο.κ. Ακόμη, θα μπορούσαμε να χρησιμοποιούμε επιπλέον του ενός σφάλματα, όπως το NMSE ή το NDEI, για πιο εύρωστη αξιολόγηση.