



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Σχεδίαση και Παραγωγή Προϊόντων»

***Παραμετρική μοντελοποίηση πεδίων ροής με χρήση NURBS και
Διαφορικού Εξελικτικού αλγορίθμου***
ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Κούσκουρας Παναγιώτης

Επιβλέπων καθηγητής: Καθ. Ι.Κ. Νικολός
1^ο μέλος Εξεταστικής Επιτροπής: Καθ. Α.Ι. Δελής
2^ο μέλος Εξεταστικής Επιτροπής: Αν. Καθ. Γ. Αραμπατζής

ΧΑΝΙΑ

Νοέμβριος 2023

Αφιερώνεται στους γονείς μου

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Ιωάννη Νικολό, για το ενδιαφέρον θέμα που μου ανέθεσε, καθώς επίσης και για την εξαιρετική συνεργασία μας, καθ' όλη τη διάρκεια εκπόνησης της εργασίας. Η επιλογή του θέματος αυτού, συνέδεε αρκετές καινούριες έννοιες για μένα και έτσι αποτέλεσε μια πραγματική πρόκληση, δίνοντας μου όμως αρκετές γνώσεις. Θα ήθελα επίσης να ευχαριστήσω τον υποψήφιο διδάκτορα κ. Γεώργιο Μάνο, για την υποστήριξή του σε τεχνικά θέματα, αλλά και για την επεξήγηση σε θεωρητικά θέματα, που με βοήθησαν να κατανοήσω σημαντικές μαθηματικές διαδικασίες ώστε να οδηγηθώ στην επιτυχή περάτωση της παρούσας εργασίας. Δεν θα μπορούσα να παραλείψω από αυτή τη λίστα την οικογένειά μου, τους γονείς μου Κωνσταντίνο και Μαρία που με στήριζαν και πίστευαν στις ικανότητές μου, αλλά και τα αδέρφια μου Έλενα, Ιωάννη Συμεών και Δημήτρη.

Περίληψη

Η έννοια της βελτιστοποίησης αναφέρεται στην εύρεση της καλύτερης δυνατής λύσης ενός προβλήματος, το οποίο συνήθως υπόκειται και σε περιορισμούς (ρητούς και μη ρητούς). Αυτή η λύση μπορεί να βρεθεί μέσω διάφορων μεθόδων και αλγορίθμων, που έχουν αναπτυχθεί για την μεγιστοποίηση ή την ελαχιστοποίηση μιας συγκεκριμένης συνάρτησης (αντικειμενικής συνάρτησης ή συνάρτησης κόστους), που αντιπροσωπεύει το πρόβλημα προς επίλυση. Οι ευρετικοί αλγόριθμοι βελτιστοποίησης είναι συνήθως εμπνευσμένοι από διάφορους μηχανισμούς εξέλιξης και συμπεριφοράς ζωντανών οργανισμών.

Το αντικείμενο της παρούσας Διπλωματικής Εργασίας είναι η χρησιμοποίηση ενός Διαφορικού Εξελικτικού Αλγορίθμου (ΔΕΑ) (που αναπτύχθηκε σε γλώσσα προγραμματισμού Python), με σκοπό την εφαρμογή του για την εύρεση των βέλτιστων σημείων ελέγχου επιφανειών NURBS και B-Splines, οι οποίες θα προσεγγίζουν όσο το δυνατόν καλύτερα επιφάνειες-στόχους, που περιγράφουν πεδία ροής. Με τον τρόπο αυτό θα επιτευχθεί παραμετρική περιγραφή συγκεκριμένων πεδίων ροής, ώστε αυτά να περιγράφονται με μικρό αριθμό παραμέτρων και να μπορούν να μοντελοποιηθούν στη συνέχεια πιθανώς με χρήση Τεχνητών Νευρωνικών Δικτύων.

Στη συνέχεια της Διπλωματικής Εργασίας, αρχικά, γίνεται μια εισαγωγή στο βέλτιστο σχεδιασμό, καθώς και στα στάδια που πρέπει να επιτευχθούν για τη σωστή διατύπωση του προβλήματος βέλτιστου σχεδιασμού. Κατόπιν, αναλύεται η δομή των Γενετικών Αλγορίθμων (ΓΑ), ώστε ο αναγνώστης να κατανοήσει τις βασικές λειτουργίες των Εξελικτικών Αλγορίθμων (ΕΑ) και από εκεί να τον εισάγουμε στην ανάλυση της δομής του Διαφορικού Εξελικτικού Αλγορίθμου - ΔΕΑ. Στη συνέχεια, αναλύεται ο τρόπος με τον οποίο περιγράφονται μαθηματικά οι καμπύλες και οι επιφάνειες στα συστήματα CAD (Computer Aided Design), καθώς και ο τρόπος δημιουργίας τους, αρχίζοντας από τις καμπύλες Bezier και καταλήγοντας στις καμπύλες και επιφάνειες NURBS. Έπειτα, ορίζονται τα προβλήματα προς επίλυση, τα

οποία είναι οι επιφάνειες στόχοι που θέλουμε να προσεγγίσουμε μέσω του ΔΕΑ. Τέλος, παρουσιάζονται τα αποτελέσματα που προκύπτουν από διάφορες test-cases και αναπτύσσονται τα τελικά συμπεράσματα.

Λίστα Εικόνων

ΕΙΚΟΝΑ 1. ΤΟΠΟΛΟΓΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΚΑΘΙΣΜΑΤΟΣ. ΠΗΓΗ: GENERICO: A CASE STUDY ON PERFORMANCE-BASED DESIGN [2].	17
ΕΙΚΟΝΑ 2. ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ. ΠΗΓΗ: DEB, K (2012). OPTIMIZATION FOR ENGINEERING DESIGN. 2 ND ED. NEW .DELHI [3]. BOOKS. P.3. [FIGURE 1.1 A FLOWCHART OF THE OPTIMAL DESIGN PROCEDURE].	20
ΕΙΚΟΝΑ 3. ΕΞΕΛΙΚΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ.	23
ΕΙΚΟΝΑ 4. ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ.	25
ΕΙΚΟΝΑ 5. ΑΝΤΙΠΡΟΣΩΠΕΥΣΗ ΠΡΑΓΜΑΤΙΚΟΥ ΑΡΙΘΜΟΥ ΜΕΣΩ ΔΥΑΔΙΚΗΣ ΚΩΔΙΚΟΠΟΙΗΣΗΣ.	26
ΕΙΚΟΝΑ 6. ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΣΤΑΥΡΩΣΗΣ ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΓΟΝΩΝ.	28
ΕΙΚΟΝΑ 7. ΜΕΤΑΛΛΑΞΗ ΓΟΝΟΥ Α.	29
ΕΙΚΟΝΑ 8. ΕΥΡΕΣΗ ΟΛΙΚΟΥ ΕΛΑΧΙΣΤΟΥ ΜΕ ΧΡΗΣΗ ΔΙΑΦΟΡΙΚΟΥ ΕΞΕΛΙΚΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ.	31
ΕΙΚΟΝΑ 9. ΔΙΑΔΙΚΑΣΙΑ ΜΕΤΑΛΛΑΞΗΣ ΣΤΟΝ ΔΙΑΦΟΡΙΚΟ ΕΞΕΛΙΚΤΙΚΟ ΑΛΓΟΡΙΘΜΟ. ΠΗΓΗ: [SIMON EVOLUTIONARY OPTIMIZATION ALGORITHMS]. FIGURE 12.1. PAGE 294 [10].	33
ΕΙΚΟΝΑ 10. ΔΙΑΔΙΚΑΣΙΑ ΔΙΑΣΤΑΥΡΩΣΗΣ. ΠΗΓΗ: NEW OPTIMIZATION TECHNIQUES IN ENGINEERING. PAGE 129. [17]	35
ΕΙΚΟΝΑ 11. ΨΕΥΔΟΚΩΔΙΚΑΣ ΔΕ [18].	36
ΕΙΚΟΝΑ 12. ΚΑΜΠΥΛΗ BEZIER 3 ^{ΟΥ} ΒΑΘΜΟΥ ΜΑΖΙ ΜΕ ΤΟ ΠΟΛΥΓΩΝΟ ΕΛΕΓΧΟΥ.	39
ΕΙΚΟΝΑ 13. ΕΠΙΦΑΝΕΙΑ BEZIER, ΟΠΟΥ ΟΙ ΔΙΕΥΘΥΝΣΕΙΣ U ΚΑΙ V ΕΙΝΑΙ 3 ^{ΟΥ} ΚΑΙ 2 ^{ΟΥ} ΒΑΘΜΟΥ ΑΝΤΙΣΤΟΙΧΑ.	40
ΕΙΚΟΝΑ 14. ΚΑΜΠΥΛΗ B-SPLINE 2 ^{ΟΥ} ΒΑΘΜΟΥ ΜΕ ΠΕΝΤΕ ΣΗΜΕΙΑ ΕΛΕΓΧΟΥ.	42
ΕΙΚΟΝΑ 15. ΕΠΙΦΑΝΕΙΑ B-SPLINE, ΟΠΟΥ ΟΙ ΔΙΕΥΘΥΝΣΕΙΣ U ΚΑΙ V ΕΙΝΑΙ 2 ^{ΟΥ} ΒΑΘΜΟΥ.	42
ΕΙΚΟΝΑ 16. ΚΑΜΠΥΛΗ NURBS 2 ^{ΟΥ} ΒΑΘΜΟΥ ΜΕ ΠΕΝΤΕ ΣΗΜΕΙΑ ΕΛΕΓΧΟΥ, ΟΠΟΥ ΤΟ ΒΑΡΟΣ ΤΟΥ ΚΕΝΤΡΙΚΟΥ ΣΗΜΕΙΟΥ ΕΛΕΓΧΟΥ ΕΙΝΑΙ ΜΙΣΟ ΣΕ ΣΧΕΣΗ ΜΕ ΤΑ ΥΠΟΛΟΙΠΑ.	44
ΕΙΚΟΝΑ 17. ΕΠΙΦΑΝΕΙΑ NURBS, ΟΠΟΥ ΟΙ ΔΙΕΥΘΥΝΣΕΙΣ U ΚΑΙ V ΕΙΝΑΙ 2 ^{ΟΥ} ΒΑΘΜΟΥ, ΜΕ ΤΟ ΣΗΜΕΙΟ ΕΛΕΓΧΟΥ (2,0,6) ΝΑ ΕΧΕΙ ΜΕΓΑΛΥΤΕΡΟ ΒΑΡΟΣ.	45
ΕΙΚΟΝΑ 18. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΚΑΙ NURBS ΜΕ ΧΡΗΣΗ ΔΙΑΚΡΙΤΩΝ ΣΗΜΕΙΩΝ ΓΙΑ ΤΗΝ ΑΠΕΙΚΟΝΙΣΗ.	47
ΕΙΚΟΝΑ 19. ΕΠΙΦΑΝΕΙΑ ΜΕ ΑΝΑΛΥΣΗ 400 ΣΗΜΕΙΩΝ.	49
ΕΙΚΟΝΑ 20. ΕΠΙΦΑΝΕΙΑ ΜΕ ΑΝΑΛΥΣΗ 1600 ΣΗΜΕΙΩΝ.	49
ΕΙΚΟΝΑ 21. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΤΗ ΣΥΝΑΡΤΗΣΗ GIUNTA ΣΤΟ ΔΙΑΣΤΗΜΑ [-1, 1].	51
ΕΙΚΟΝΑ 22. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΤΗ ΣΥΝΑΡΤΗΣΗ ΚΕΑΝΕ ΣΤΟ ΔΙΑΣΤΗΜΑ [-4, 4].	52
ΕΙΚΟΝΑ 23. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΤΗΝ ΣΥΝΑΡΤΗΣΗ ΚΕΑΝΕ ΣΤΟ ΔΙΑΣΤΗΜΑ [-1, 1].	53
ΕΙΚΟΝΑ 24. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΤΗΝ ΣΥΝΑΡΤΗΣΗ PARSΟΡΟΥΛΟΣ ΣΤΟ ΔΙΑΣΤΗΜΑ [-5, 5].	54
ΕΙΚΟΝΑ 25. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΤΡΟΠΟΠΟΙΗΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ PARSΟΡΟΥΛΟΣ ΣΤΟ ΔΙΑΣΤΗΜΑ [-1, 1].	55
ΕΙΚΟΝΑ 26. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΤΗΝ ΣΥΝΑΡΤΗΣΗ PEAKS ΣΤΟ ΔΙΑΣΤΗΜΑ [-4, 4].	56
ΕΙΚΟΝΑ 27. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΤΡΟΠΟΠΟΙΗΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ PEAKS ΣΤΟ ΔΙΑΣΤΗΜΑ [-1, 1].	56
ΕΙΚΟΝΑ 28. ΕΠΙΦΑΝΕΙΑ ΣΤΟΧΟΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΜΙΑ ΤΥΧΑΙΑ ΣΥΝΑΡΤΗΣΗ ΣΤΟ ΔΙΑΣΤΗΜΑ [-1, 1].	57
ΕΙΚΟΝΑ 29. ΕΠΙΦΑΝΕΙΑ ΑΠΟ ΠΡΑΓΜΑΤΙΚΑ ΔΕΔΟΜΕΝΑ ΠΕΔΙΟΥ ΡΟΗΣ (CAVITY FLOW).	58
ΕΙΚΟΝΑ 30. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ ΑΝΑ ΓΕΝΙΑ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΜΕ ΒΑΣΗ ΤΗ ΣΥΝΑΡΤΗΣΗ GIUNTA.	60
ΕΙΚΟΝΑ 31. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑΣ GIUNTA ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 20.	61
ΕΙΚΟΝΑ 32. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑ GIUNTA ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 300.	61
ΕΙΚΟΝΑ 33. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ ΑΝΑ ΓΕΝΙΑ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΜΕ ΒΑΣΗ ΤΗ ΣΥΝΑΡΤΗΣΗ ΚΕΑΝΕ.	62
ΕΙΚΟΝΑ 34. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑΣ ΚΕΑΝΕ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 30.	63
ΕΙΚΟΝΑ 35. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑΣ ΚΕΑΝΕ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 300.	63
ΕΙΚΟΝΑ 36. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ ΑΝΑ ΓΕΝΙΑ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΜΕ ΒΑΣΗ ΤΗ ΣΥΝΑΡΤΗΣΗ PARSΟΡΟΥΛΟΣ.	64
ΕΙΚΟΝΑ 37. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑΣ PARSΟΡΟΥΛΟΣ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 30.	65
ΕΙΚΟΝΑ 38. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑΣ PARSΟΡΟΥΛΟΣ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 300.	65

ΕΙΚΟΝΑ 39. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ ΑΝΑ ΓΕΝΙΑ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΜΕ ΒΑΣΗ ΤΗ ΣΥΝΑΡΤΗΣΗ REAKS.	66
ΕΙΚΟΝΑ 40. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑΣ REAKS ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 30.	67
ΕΙΚΟΝΑ 41. ΠΡΟΣΕΓΓΙΣΗ ΕΠΙΦΑΝΕΙΑΣ REAKS ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 300.	67
ΕΙΚΟΝΑ 42. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ ΑΝΑ ΓΕΝΙΑ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΜΕ ΒΑΣΗ ΤΗΝ ΤΥΧΑΙΑ ΣΥΝΑΡΤΗΣΗ ΠΟΥ ΕΠΙΛΕΧΘΗΚΕ.	68
ΕΙΚΟΝΑ 43. ΠΡΟΣΕΓΓΙΣΗ ΤΥΧΑΙΑΣ ΕΠΙΦΑΝΕΙΑΣ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 30.	69
ΕΙΚΟΝΑ 44. ΠΡΟΣΕΓΓΙΣΗ ΤΥΧΑΙΑΣ ΕΠΙΦΑΝΕΙΑΣ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 300.	69
ΕΙΚΟΝΑ 45. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ ΚΑΘΕ ΕΠΙΦΑΝΕΙΑΣ, ΜΕ ΕΦΑΡΜΟΓΗ ΤΩΝ ΒΕΛΤΙΣΤΩΝ ΤΙΜΩΝ ΠΑΡΑΜΕΤΡΩΝ ΚΑΙ ΑΡΙΘΜΟ ΓΕΝΙΩΝ 3000.	70
ΕΙΚΟΝΑ 46. ΑΠΟΤΕΛΕΣΜΑΤΑ ΜΕΣΟΥ ΤΕΤΡΑΓΩΝΙΚΟΥ ΣΦΑΛΜΑΤΟΣ ΑΝΑ ΓΕΝΙΑ ΤΗΣ ΕΠΙΦΑΝΕΙΑΣ ΠΟΥ ΔΗΜΙΟΥΡΓΗΘΗΚΕ ΑΠΟ ΔΕΔΟΜΕΝΑ ΠΡΑΓΜΑΤΙΚΟΥ ΠΕΔΙΟΥ ΡΟΗΣ (CAVITY FLOW).	71
ΕΙΚΟΝΑ 47. ΠΡΟΣΕΓΓΙΣΗ ΠΕΔΙΟΥ ΡΟΗΣ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 1.	72
ΕΙΚΟΝΑ 48. ΠΡΟΣΕΓΓΙΣΗ ΠΕΔΙΟΥ ΡΟΗΣ ΜΕ ΧΡΗΣΗ NURBS ΚΑΙ ΔΕΑ ΣΤΗ ΓΕΝΙΑ 1000.	72

Λίστα Πινάκων

ΠΙΝΑΚΑΣ 1. ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΟΚΙΜΩΝ ΕΠΙΦΑΝΕΙΑΣ GIUNTA.	60
ΠΙΝΑΚΑΣ 2. ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΟΚΙΜΩΝ ΕΠΙΦΑΝΕΙΑΣ KEANE.	62
ΠΙΝΑΚΑΣ 3. ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΟΚΙΜΩΝ ΕΠΙΦΑΝΕΙΑΣ PARSΟΡΟΥΛΟΣ.	64
ΠΙΝΑΚΑΣ 4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΟΚΙΜΩΝ ΕΠΙΦΑΝΕΙΑΣ REAKS.....	66
ΠΙΝΑΚΑΣ 5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΟΚΙΜΩΝ ΤΥΧΑΙΑΣ ΕΠΙΦΑΝΕΙΑΣ.....	68

Περιεχόμενα

Κεφάλαιο 1 - Εισαγωγή.....	15
Κεφάλαιο 2 – Εξελικτικοί Αλγόριθμοι.....	22
2.1 Ιστορική Αναδρομή	22
2.2 Γενετικός Αλγόριθμος	24
2.2.1 Αντιπροσώπευση - παραμετροποίηση	26
2.2.2 Αρχικοποίηση	26
2.2.3 Αξιολόγηση.....	27
2.2.4 Επιλογή.....	27
2.2.5 Διασταύρωση	28
2.2.6 Μετάλλαξη	29
2.2.7 Αντικατάσταση	29
2.2.8 Κριτήριο Τερματισμού	30
2.3 Διαφορικός Εξελικτικός Αλγόριθμος.....	30
2.3.1 Πληθυσμός	32
2.3.2 Αρχικοποίηση	32
2.3.3 Μετάλλαξη	33
2.3.4 Διασταύρωση	34
2.3.5 Επιλογή.....	35
2.3.6 Ψευδοκώδικας ΔΕ	36
Κεφάλαιο 3 – Δημιουργία Παραμετρικών Καμπυλών και Επιφανειών	37
3.1 Καμπύλες και Επιφάνειες Bezier	38
3.2 Καμπύλες και Επιφάνειες B-Splines.....	41
3.3 Καμπύλες και Επιφάνειες NURBS	43
Κεφάλαιο 4 – Εφαρμογή	46
4.1 Μεθοδολογία	46
4.2 Ορισμός προβλημάτων	50
4.2.1 Συνάρτηση Giunta	50
4.2.2 Συνάρτηση Keane.....	52
4.2.3 Συνάρτηση Parsopoulos	53
4.2.3 Συνάρτηση Peaks.....	55
4.2.4 Τυχαία Συνάρτηση Επιφάνειας.....	57
4.2.5 Επιφάνεια πραγματικού πεδίου ροής	58
Κεφάλαιο 5 – Αποτελέσματα	59

5.1 Αποτελέσματα επιφάνειας Giunta.....	59
5.2 Αποτελέσματα επιφάνειας Keane	62
5.3 Αποτελέσματα επιφάνειας Parsopoulos	64
5.4 Αποτελέσματα επιφάνειας Peaks	66
5.5 Αποτελέσματα τυχαίας επιφάνειας.....	68
5.6 Αποτελέσματα επιφανειών με μεγάλο αριθμό γενιών	70
5.7 Αποτελέσματα επιφάνειας πραγματικού πεδίου ροής.....	71
Κεφάλαιο 6 – Συμπεράσματα	73
Βιβλιογραφία	75

Κεφάλαιο 1 - Εισαγωγή

Η διαδικασία και ο τρόπος σχεδίασης και κατασκευής ενός συστήματος, απασχολεί την ανθρωπότητα εκατοντάδες χρόνια [1]. Συστήματα, όπως κτήρια, δρόμοι, αυτοκίνητα, αεροπλάνα, υπολογιστές κ.α., αποτελούν ένα μεγάλο κομμάτι της κοινωνίας μας, τα οποία χρειάστηκαν πολλά χρόνια για να αναπτυχθούν και να φτάσουν στη μορφή που βρίσκονται σήμερα. Αυτή η εξέλιξη των συστημάτων, αλλά και η διαδικασία της σχεδίασης και κατασκευής του κάθε συστήματος είναι δαπανηρή και χρονοβόρα.

Η μελέτη, σχεδίαση και κατασκευή των παραπάνω συστημάτων, αλλά και άλλων συστημάτων μηχανικής είναι μια ιδιαίτερα απαιτητική διαδικασία. Πρέπει να ληφθούν υπόψη όλοι οι πιθανοί παράγοντες που επηρεάζουν το αποτέλεσμα του υπό μελέτη συστήματος. Για να ξεκινήσει η διαδικασία της μελέτης, αρχικά λαμβάνονται κάποιες υποθέσεις, οι οποίες μεταφράζονται ως δεδομένα ή σταθερές. Με βάση αυτά, αναπτύσσονται κάποια προσεγγιστικά μοντέλα του εν λόγω συστήματος, τα οποία υπόκεινται σε μαθηματικές αναλύσεις και στη συνέχεια επαληθεύονται οι αναλύσεις μέσω διαφόρων πειραμάτων.

Στη διαδικασία της μελέτης, πρέπει να συμπεριλάβουμε και την ανάλυση. Όταν σχεδιαστεί ένα σύστημα, ελέγχουμε αν εκπληρώνονται οι αρχικοί μας στόχοι ή προδιαγραφές. Εάν εκπληρώνονται, τότε η σχεδίαση είναι *αποδεκτή (acceptable)* ή *εφικτή (feasible)*. Στην περίπτωση που δεν εκπληρώνονται, τότε επανασχεδιάζεται το σύστημα και αναλύεται εκ νέου, μέχρις ότου να βρεθεί μία αποδεκτή λύση. Υπάρχει πιθανότητα να εκπληρώνονται οι αρχικοί στόχοι της σχεδίασης, αλλά να θέλουμε να τους βελτιώσουμε ή να προσθέσουμε κι άλλους αργότερα. Και στις δύο αυτές περιπτώσεις, διακρίνουμε ότι η διαδικασία της σχεδίασης του συστήματος, είναι στην ουσία μια επαναληπτική διαδικασία (iterative process). Αυτή η επαναληπτική διαδικασία, μπορεί όντως να βελτιώνει το σύστημα μας σε μεγάλο βαθμό, αλλά μπορεί και αρκετές φορές αυτή η βελτίωση να είναι μικρή, έως και αμελητέα. Η επαναληπτική διαδικασία από μόνη της δεν είναι αρκετή και δεν μας οδηγεί

απαραίτητα στην βέλτιστη λύση. Γι' αυτό το λόγο είναι απαραίτητη η χρήση μεθοδολογίας βέλτιστου σχεδιασμού.

Ο βέλτιστος σχεδιασμός είναι μια μεθοδολογία (ή καλύτερα σύνολο μεθόδων), η οποία χρησιμοποιεί μαθηματικά μοντέλα και αλγορίθμους για την εύρεση της βέλτιστης δυνατής λύσης, η οποία ικανοποιεί κάποιους περιορισμούς. Λύνοντας το πρόβλημα της σχεδίασης-μελέτης, ως ένα πρόβλημα βελτιστοποίησης, μπορούμε να βρούμε αποτελεσματικά την βέλτιστη λύση του συστήματος. Συνεπώς, η χρήση του βέλτιστου σχεδιασμού είναι σημαντική για να εξασφαλίσουμε ότι η βέλτιστη δυνατή σχεδίαση του συστήματος έχει επιτευχθεί.

Οι αλγόριθμοι βελτιστοποίησης έχουν γίνει πολύ δημοφιλείς σε διάφορους τομείς της μηχανικής, λόγω της διαθεσιμότητάς τους και της πιο εύκολης πρόσβασης σε υπολογιστές με υψηλή ταχύτητα. Χρησιμοποιούνται κυρίως σε προβλήματα, όπου θέλουμε να ελαχιστοποιήσουμε ή να μεγιστοποιήσουμε την αντικειμενική συνάρτηση-στόχο. Για παράδειγμα, οι μηχανικοί παραγωγής θέλουν να βελτιστοποιήσουν την παραγωγική διαδικασία. Με τη χρήση ενός τέτοιου αλγορίθμου μπορούν να αλλάξουν το πρόγραμμα παραγωγής, ώστε παράγοντας άλλου τύπου προϊόντα στην γραμμή παραγωγής, να μειώσουν το χρόνο και το κόστος παραγωγής. Οι μηχανολόγοι μηχανικοί μπορούν να βελτιστοποιήσουν το σχήμα ή την τοπολογία ενός αντικειμένου [2], ώστε να μειώσουν το βάρος του, κρατώντας παράλληλα τα μηχανολογικά του χαρακτηριστικά ([Εικόνα 1](#)). Οι ηλεκτρολόγοι μηχανικοί μπορούν βελτιστοποιήσουν τη δρομολόγηση της ισχύος ενός δικτύου διανομής ηλεκτρικής ενέργειας, για να ελαχιστοποιήσουν τις απώλειες και να αυξήσουν την απόδοση.



Εικόνα 1. Τοπολογική βελτιστοποίηση καθίσματος.
Πηγή: Generico: A case study on performance-based design [2].

Από τα παραπάνω διακρίνουμε ότι κάθε τομέας και κατ' επέκταση κάθε πρόβλημα, έχει τις δικές του ιδιαιτερότητες. Ως εκ τούτου η επιλογή του πιο κατάλληλου αλγορίθμου βελτιστοποίησης για την επίλυση του προβλήματος βελτιστοποίησης, απαιτεί και την σωστή διατύπωση (*formulation*) του εκάστοτε προβλήματος. «Στόχος της διατύπωσης του προβλήματος είναι η δημιουργία ενός μαθηματικού μοντέλου, το οποίο θα επιλυθεί, χρησιμοποιώντας τον αλγόριθμο βελτιστοποίησης» [3].

Κατά τη διατύπωση του προβλήματος, ορίζουμε τις μεταβλητές σχεδίασης (*design variables*), τους περιορισμούς (*constraints*), την αντικειμενική συνάρτηση (*objective function*) και τα όρια των μεταβλητών σχεδίασης (*variable bounds*).

Οι μεταβλητές σχεδίασης αναφέρονται στα χαρακτηριστικά του συστήματος. Ο αριθμός αυτών των μεταβλητών καθορίζεται από τον σχεδιαστή/μελετητή και θα πρέπει να τροποποιηθούν κατάλληλα για να βελτιωθεί η απόδοση του συστήματος. Βεβαίως, ο αριθμός αυτός δεν θα πρέπει να είναι πολύ μεγάλος, διότι όσο περισσότερες είναι οι μεταβλητές, τόσο δυσκολότερο γίνεται το πρόβλημα προς επίλυση (λόγω της μεγέθυνσης του χώρου των πιθανών λύσεων). Αυτό αντιστοιχεί σε μεγαλύτερο χρόνο επίλυσης του προβλήματος ή/και μεγαλύτερο υπολογιστικό κόστος. Οι μεταβλητές σχεδίασης μπορούν να είναι διάφορες ανάλογα με το

πρόβλημα βελτιστοποίησης που θέλουμε να επιλύσουμε. Για παράδειγμα, μπορεί να αναφέρονται σε χαρακτηριστικά ενός προϊόντος ή αντικειμένου, όπως το μέγεθος, το σχήμα ή το υλικό. Σε ένα ηλεκτρολογικό σύστημα μπορεί να είναι η τάση, η ισχύς ή οτιδήποτε άλλο επηρεάζει την απόδοση του συστήματος. Σε ένα σύστημα παραγωγής μπορεί να είναι διάφοροι παράμετροι που επηρεάζουν το κόστος ή και το χρόνο περάτωσης μιας παραγγελίας, όπως η διάταξη των μηχανημάτων, το κόστος των υλικών, η ταχύτητα του κάθε τμήματος παραγωγής, η ταχύτητα της κάθε μηχανής κ.ο.κ. Είναι στην ουσία παράμετροι που επηρεάζουν την απόδοση του συστήματος. Η σωστή επιλογή των μεταβλητών σχεδίασης είναι σημαντική στην επίλυση προβλημάτων βελτιστοποίησης, καθώς μπορεί να επηρεάσει σε μεγάλο βαθμό την ταχύτητα εύρεσης της βέλτιστης λύσης, αλλά και να οδηγήσει σε καλύτερα αποτελέσματα.

Οι περιορισμοί αναφέρονται στις συνθήκες που πρέπει να λαμβάνονται υπόψη κατά τη διαδικασία της βελτιστοποίησης ενός συστήματος. Όπως και στις μεταβλητές σχεδίασης, ο αριθμός τους καθορίζεται από τον σχεδιαστή/μελετητή και πρέπει να είναι όσο το δυνατόν μικρότερος, εφόσον η βέλτιστη λύση είναι εφικτή. Μπορούν να πάρουν διάφορες μορφές, ανάλογα με το είδος του προβλήματος. Οι τύποι των περιορισμών είναι είτε σε μορφή ανισότητας, είτε σε μορφή ισότητας. Σε προβλήματα βελτιστοποίησης, προτιμάται η μορφή της ανισότητας. Για παράδειγμα, σε ένα σύστημα παραγωγής, ένας περιορισμός ανισότητας μπορεί να είναι ο χρόνος λειτουργίας μίας μηχανής ή ο αριθμός των εργαζομένων που μπορούν να δουλέψουν ταυτόχρονα σε μία μηχανή κ.ο.κ. Αυτό μαθηματικά μπορεί να εκφραστεί για το προηγούμενο παράδειγμα ως,

$$T_m \leq T_{max} \quad (1)$$

Όπου T_m ο χρόνος λειτουργίας της μηχανής m και T_{max} ο μέγιστος χρόνος λειτουργίας της εκάστοτε μηχανής.

Η αντικειμενική συνάρτηση είναι μία μαθηματική σχέση που περιγράφει τον στόχο του προβλήματος βελτιστοποίησης. Περιγράφεται με μία ή περισσότερες μεταβλητές σχεδίασης. Σκοπός της βελτιστοποίησης είναι να βρεθούν οι κατάλληλες τιμές των παραπάνω μεταβλητών σχεδίασης, όπου αυτή η συνάρτηση θα ελαχιστοποιείται ή θα μεγιστοποιείται. Η αντικειμενική συνάρτηση μπορεί να αντιπροσωπεύει κόστος, χρόνο συνολικής παραγωγής ή ενός προϊόντος, κέρδος, παραγωγικότητα κ.α. Έτσι, για παράδειγμα ελαχιστοποιώντας την αντικειμενική συνάρτηση, βρίσκουμε τις μεταβλητές με τις οποίες το κόστος θα είναι το ελάχιστο δυνατό. Αντίστοιχα, μεγιστοποιώντας την αντικειμενική συνάρτηση, μπορούμε να βρούμε τις μεταβλητές, όπου το κέρδος ή η παραγωγικότητα θα είναι η μέγιστη. Η αντικειμενική συνάρτηση μπορεί να είναι είτε γραμμική, είτε μη γραμμική. Μαθηματικά η γραμμική συνάρτηση μπορεί να εκφραστεί ως,

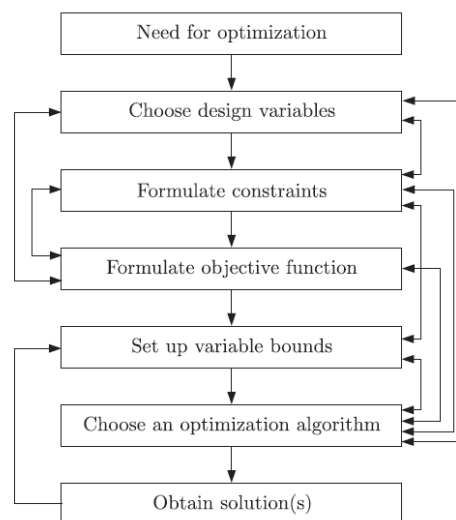
$$f(x) = a_1x_1 + a_2x_2 + \dots + a_nx_n = \sum_{i=1}^n a_ix_i \quad (2)$$

Όπου τα a_i είναι σταθερές και τα x_i οι μεταβλητές σχεδίασης, ενώ μια μη γραμμική συνάρτηση μπορεί να είναι της μορφής,

$$f(x) = ax^2 \quad (3)$$

Ένα άλλο σημαντικό κομμάτι που πρέπει να λάβουμε υπόψη κατά τη διατύπωση του προβλήματος, είναι ο ορισμός των ορίων των μεταβλητών σχεδίασης. Τα όρια των μεταβλητών σχεδίασης είναι οι (ρητοί) περιορισμοί που ορίζονται στις τιμές που μπορούν να πάρουν οι μεταβλητές σχεδίασης. Μπορούν να επηρεάσουν τη λειτουργία του αλγορίθμου βελτιστοποίησης, ως προς την ταχύτητα, αλλά και την απόδοσή του. Σε περίπτωση που η διαφορά μεταξύ της μικρότερης και της μεγαλύτερης επιτρεπτής τιμής είναι μεγάλη, τα όρια των μεταβλητών σχεδίασης έχουν οριστεί να κυμαίνονται σε μεγάλο εύρος τιμών και αντίστροφα. Η σωστή επιλογή των ορίων μεταβλητών σχεδίασης έχει καθοριστικό ρόλο για την αποδοτικότητα και την αποτελεσματικότητα του αλγορίθμου.

Εφόσον, έχουν οριστεί οι μεταβλητές σχεδίασης, οι περιορισμοί, η αντικειμενική συνάρτηση και τα όρια των μεταβλητών σχεδίασης, έχουμε ολοκληρώσει τη διαδικασία διατύπωσης του προβλήματος βελτιστοποίησης. Σε αυτή τη φάση, επιλέγεται ο αλγόριθμος βελτιστοποίησης, που θα μας δώσει τη βέλτιστη λύση για το πρόβλημα που έχει επιλεχθεί προς επίλυση. Στην [Εικόνα 2](#) παρουσιάζεται ένα διάγραμμα ροής, το οποίο ξεκινάει από την ανάγκη για χρήση βέλτιστου σχεδιασμού και καταλήγει στην εύρεση της βέλτιστης λύσης.



Εικόνα 2. Διάγραμμα ροής Βέλτιστου Σχεδιασμού.

Πηγή: DEB, K (2012). *Optimization for Engineering Design*. 2nd ed. New .Delhi [3]. Books. p.3. [Figure 1.1 A flowchart of the optimal design procedure].

Για την καλύτερη κατανόηση της διαδικασίας της σχεδίασης ακολουθεί ένα απλό παράδειγμα. Ας υποθέσουμε ότι βρισκόμαστε σε μία εταιρεία, η οποία παράγει δύο τύπους προϊόντων A και B. Συγκεκριμένα, χρειάζονται πέντε μονάδες πρώτης ύλης και τρεις ανθρωποώρες για την παραγωγή μιας μονάδας προϊόντος A, ενώ για το προϊόν B χρειάζονται τρεις μονάδες πρώτης ύλης και μια ανθρωποώρα. Η τιμή πώλησης είναι εικοσιπέντε και δέκα μονάδες χρήματος για τα προϊόντα A και B αντίστοιχα. Έχουμε στη διάθεση μας εκατό ανθρωποώρες και εκατό πενήντα μονάδες πρώτης ύλης. Θέλουμε να μεγιστοποιήσουμε το κέρδος. Αρχικά ακολουθούμε τη διαδικασία της διατύπωσης του προβλήματος. Οι μεταβλητές σχεδίασης είναι:

x : Μονάδες του προϊόντος A που πρέπει να παραχθούν

y : Μονάδες του προϊόντος B που πρέπει να παραχθούν

Εφόσον θέλουμε να μεγιστοποιήσουμε το κέρδος και λαμβάνοντας υπόψη μόνο τα παραπάνω δεδομένα, μπορούμε να γράψουμε την αντικειμενική συνάρτηση ως εξής:

$$f(x, y) = 25x + 10y \quad (4)$$

η οποία υπόκειται στους παρακάτω περιορισμούς:

- $5x + 3y \leq 150$
- $3x + y \leq 100$
- $x, y \geq 0$

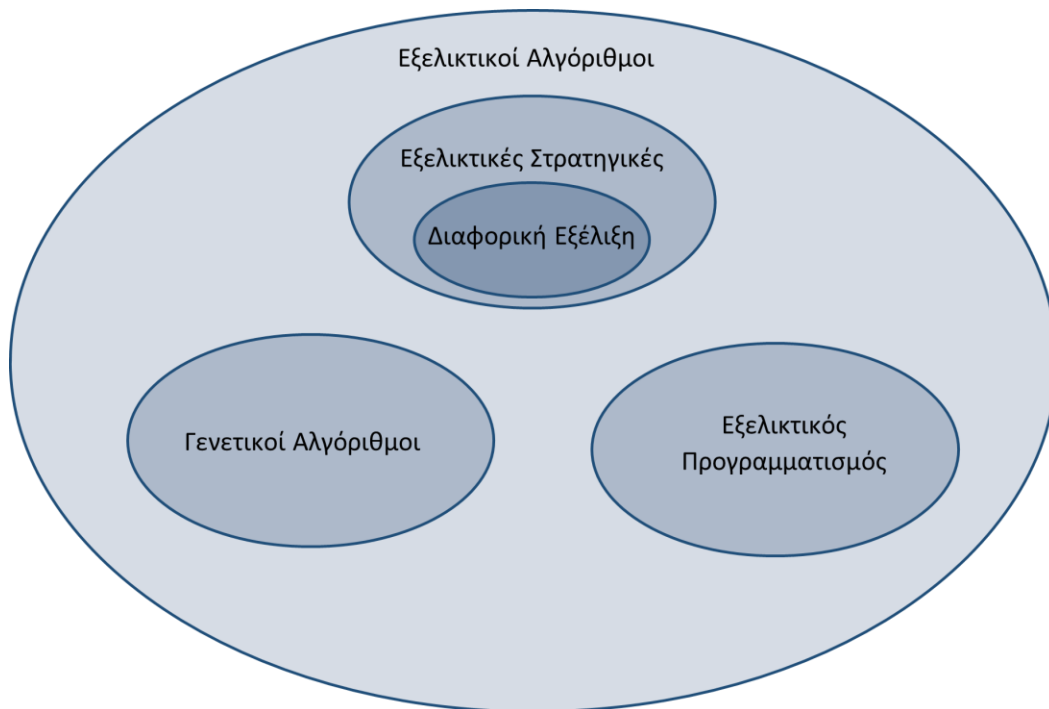
Το παραπάνω πρόβλημα είναι ένα γραμμικό πρόβλημα, το οποίο μπορεί επίσης να λυθεί χρησιμοποιώντας και τη μέθοδο του γραμμικού προγραμματισμού.

Κεφάλαιο 2 – Εξελικτικοί Αλγόριθμοι

2.1 Ιστορική Αναδρομή

Η ανάγκη για βελτιστοποίηση οδήγησε στην δημιουργία και ανάπτυξη των *Εξελικτικών Αλγορίθμων* (Evolutionary Algorithms, EA). Οι εξελικτικοί αλγόριθμοι βασίζονται στη θεωρία της εξέλιξης του Κάρολου Δαρβίνου “The Origin of Species” [4] και ανήκουν στους στοχαστικούς αλγορίθμους για την επίλυση προβλημάτων. Στη θεωρία της εξέλιξης, ο Δαρβίνος αναλύει ότι όλα τα είδη των οργανισμών εξελίσσονται με την πάροδο του χρόνου μέσω της *φυσικής επιλογής* (natural selection). Η φυσική επιλογή, καθοδηγείται από κάποιους παράγοντες. Ένας από αυτούς είναι η ποικιλομορφία των οργανισμών (variations), η οποία αναφέρεται στις διαφορές/παραλλαγές των φυσικών χαρακτηριστικών, τις συμπεριφορές ή τη γενετική σύνθεση των οργανισμών σε ένα πληθυσμό. Οι παραλλαγές αυτές μπορούν να μεταβιβαστούν από τους γονείς στους γόνους. Αυτό δημιουργεί τον δεύτερο παράγοντα, ο οποίος είναι η κληρονομικότητα (inheritance). Κατόπιν, μέσω της φυσικής επιλογής, τα άτομα που είναι καλύτερα προσαρμοσμένα στο περιβάλλον τους, είναι πιο πιθανό να επιβιώσουν, να αναπτυχθούν και να αναπαραχθούν. Ως αποτέλεσμα, τα φυσικά χαρακτηριστικά των οργανισμών ενός πληθυσμού, που προσδίδουν πλεονεκτήματα επιβίωσης, εμφανίζονται συχνότερα, άρα γίνονται πιο κοινά στον πληθυσμό με την πάροδο του χρόνου, οδηγώντας σε σταδιακές αλλαγές στα είδη του πληθυσμού. Αυτή η διαδικασία αλλαγής των ειδών του πληθυσμού, σε βάθος πολλών γενεών είναι γνωστή ως φυσική προσαρμογή (adaptation).

Οι Εξελικτικοί Αλγόριθμοι μιμούνται τη διαδικασία της φυσικής επιλογής εφαρμόζοντας στον πληθυσμό των υποψήφιων λύσεων του προς επίλυση προβλήματος κάποιες τεχνικές, οι οποίες ονομάζονται *τελεστές*. Αυτοί είναι ο τελεστής της επιλογής (selection), της μετάλλαξης (mutation) και του επιχiasμού (crossover) ή αλλιώς ανασυνδυασμού (recombination) [5].



Εικόνα 3. Εξελικτικοί Αλγόριθμοι.

Παρόλο που η ιδέα του Εξελικτικού Αλγορίθμου χρονολογείται από τις δεκαετίες του '50 και του '60 [6], ο ίδιος ο όρος προτάθηκε στις αρχές του '90, ως υπερσύνολο των ακόλουθων τριών ανεξάρτητων πεδίων, καθώς και όλων των παραλλαγών τους. Τα πεδία αυτά, είναι οι Γενετικοί Αλγόριθμοι (Genetic Algorithms, GA) [Holland], οι Εξελικτικές Στρατηγικές (Evolutionary Strategies, ES) [Rechenberg, Schwefel] και ο Εξελικτικός Προγραμματισμός (Evolutionary Programming, EP) [Fogel]. Οι τρεις παραπάνω μέθοδοι έχουν αρκετά κοινά σημεία. Όλες οι πιθανές λύσεις ανταγωνίζονται μεταξύ τους για την επιβίωση και την ευκαιρία για αναπαραγωγή [7]. Γενικότερα οι μέθοδοι αυτοί, χρησιμοποιούν μια προσέγγιση βάσει πληθυσμού (population-based), η οποία περιλαμβάνει τυχαίες παραλλαγές του αρχικού πληθυσμού (των υποψήφίων λύσεων) και τυχαίες επιλογές για την εξερεύνηση του χώρου των λύσεων.

Οι Γενετικοί Αλγόριθμοι εισήχθησαν για πρώτη φορά από τον John Holland, στις αρχές της δεκαετίας του '60, ο οποίος ήταν καθηγητής στο Πανεπιστήμιο του Μίσιγκαν.

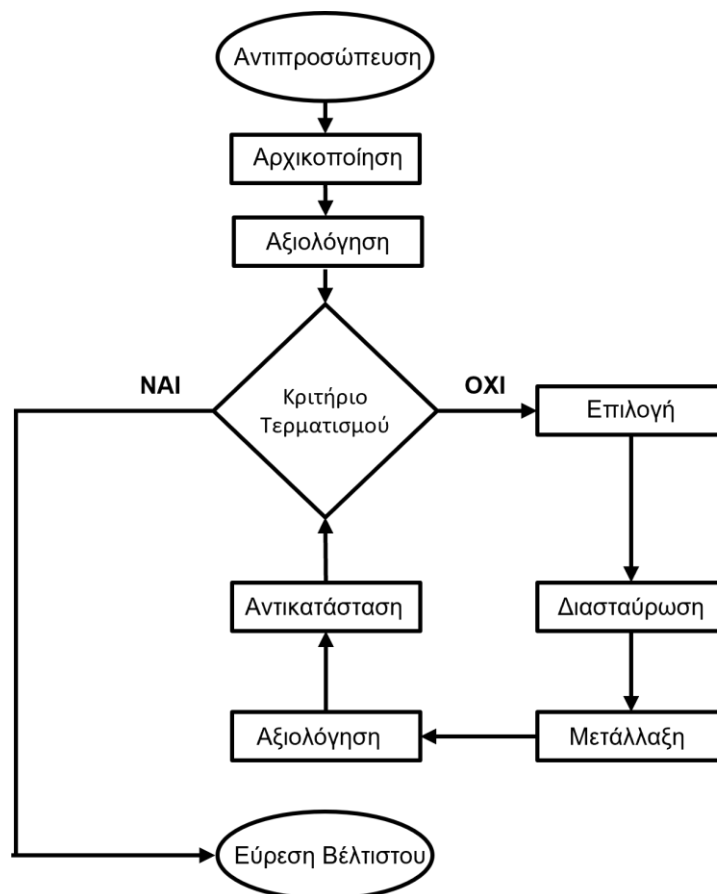
Κατόπιν, συνέχισε να βελτιώνει τον αλγόριθμο και το 1975 δημοσίευσε το βιβλίο "Adaptation in Natural and Artificial Systems" [8]. Παράλληλα, το 1975, ένας μαθητής του Holland, ο Kenneth De Jong δημοσίευσε την διδακτορική του διατριβή "An analysis of the behavior of a class of genetic adaptive systems" [9], στην οποία ανέλυε τις επιπτώσεις των διαφόρων παραμέτρων των Γενετικών Αλγορίθμων στην απόδοση της βελτιστοποίησης [10]. Ο Εξελικτικός Προγραμματισμός είχε προταθεί από τον Lawrence Fogel, μαζί με τους συνεργάτες του Al Owens και Jack Walsh τη δεκαετία του '60, μέσω του βιβλίου "Artificial Intelligence through Simulated Evolution" [11], ενώ οι Εξελικτικές Στρατηγικές ανακαλύφθηκαν από τους Ingo Rechenberg, Hans-Paul Schwefel και Peter Bientert, προσπαθώντας να ανακαλύψουν το βέλτιστο σχήμα μιας αεροδυναμικής σήραγγας, για να ελαχιστοποιήσουν την αντίσταση του αέρα. Το '73 ο Rechenberg δημοσίευσε το βιβλίο "Evolution Strategy: Optimization of Technical Systems Based on the Principles of Biological Evolution" [12] και κατόπιν ο Schwefel το '74 δημοσίευσε το βιβλίο "Numerical Optimization of Computer Models" [13]. Από τότε έχουν βγει πολλές δημοσιεύσεις και παραλλαγές αυτών των αλγορίθμων, ένας από αυτούς είναι ο Διαφορικός Εξελικτικός Αλγόριθμος, τον οποίο θα αναλύσουμε αργότερα. Προτού όμως γίνει η ανάλυση του Διαφορικού Εξελικτικού Αλγορίθμου, θα αναλύσουμε τη φιλοσοφία και τη λειτουργία του Γενετικού Αλγορίθμου, ώστε να κατανοηθούν οι βασικές λειτουργίες των ΕΑ.

2.2 Γενετικός Αλγόριθμος

Ο Γενετικός Αλγόριθμος βασίζεται σε ένα πλήθος *υποψήφιων λύσεων* (*candidate solutions*) ή αλλιώς *ατόμων* (*individuals*). Κάθε άτομο εκπροσωπεί μία εφικτή λύση στο πρόβλημα βελτιστοποίησης που θέλουμε να επιλύσουμε [7]. Επίσης, κάθε άτομο αποτελείται από τα *χρωμοσώματα* (*chromosomes*) και την *τιμή αξιολόγησης – συνάρτηση προσαρμογής* (*fitness value – fitness function*), η οποία αναφέρεται στο πόσο κοντά στην βέλτιστη λύση βρίσκεται το εκάστοτε άτομο. Τα χρωμοσώματα με τη σειρά τους, αποτελούνται από τα *γονίδια* (*genes*), τα οποία καθορίζουν την τιμή

αξιολόγησης. Οι ΓΑ αποτελούνται από διάφορους τελεστές που αναφέραμε και παραπάνω. Η δομή τους έχει την ακόλουθη μορφή (Εικόνα 4).

1. Αντιπροσώπηση - παραμετροποίηση (Representation)
2. Αρχικοποίηση (Initialization)
3. Αξιολόγηση (Evaluation)
4. Επιλογή (Selection)
5. Διασταύρωση - Επιχιασμός (Crossover)
6. Μετάλλαξη (Mutation)
7. Αντικατάσταση (Replacement)
8. Κριτήριο Τερματισμού (Termination)



Εικόνα 4. Διάγραμμα ροής Γενετικού Αλγορίθμου.

2.2.1 Αντιπροσώπευση - παραμετροποίηση

Το αρχικό βήμα που πρέπει να κάνουμε, είναι να κωδικοποιήσουμε τις πιθανές λύσεις με πραγματικούς αριθμούς ή δυαδικά ψηφία. Αν πάρουμε την περίπτωση της δυαδικής κωδικοποίησης, κάθε δυαδικό ψηφίο αντιστοιχεί και σε ένα γονίδιο του χρωμοσώματος. Κάθε γονίδιο έχει δύο ιδιότητες που το χαρακτηρίζουν. Αυτές είναι η τιμή του, η οποία καλείται αλληλόμορφο (allele) και η θέση του, η οποία καλείται τόπος (locus). Για παράδειγμα, η αναπαράσταση του αριθμού 9, σε δυαδική μορφή είναι 1001. Το κάθε «κουτάκι» στην παρακάτω εικόνα αναπαριστά το γονίδιο του χρωμοσώματος, οι τιμές εντός είναι η τιμή του κάθε γονιδίου, ενώ οι τιμές εκτός είναι η θέση στην οποία βρίσκεται το κάθε γονίδιο [14] ([Εικόνα 5](#)).

3	2	1	0
1	0	0	1

Εικόνα 5. Αντιπροσώπευση πραγματικού αριθμού μέσω δυαδικής κωδικοποίησης.

2.2.2 Αρχικοποίηση

Εφόσον ο ΓΑ κάνει διάφορες λειτουργίες στα άτομα ενός πληθυσμού, θα πρέπει πρώτα να ορίσουμε το πλήθος του (population size), το οποίο (συνήθως) παραμένει σταθερό καθ' όλη τη διάρκεια λειτουργίας του αλγορίθμου. Κατόπιν, δημιουργούνται τυχαίες τιμές στα γονίδια του κάθε ατόμου του πληθυσμού, ως αποτέλεσμα κάθε άτομο να έχει τυχαίες τιμές αξιολόγησης. Αυτή η διαδικασία είναι απαραίτητη, διότι έτσι υπάρχει μία βάση για την εκκίνηση της εξέλιξής του πληθυσμού. Η διαδικασία που περιγράφηκε παραπάνω ονομάζεται αρχικοποίηση [5].

2.2.3 Αξιολόγηση

Μετά την αρχικοποίηση (ή την αντικατάσταση) γίνεται αξιολόγηση κάθε ατόμου του πληθυσμού, μέσω της συνάρτησης αξιολόγησης (evaluation function), ώστε να συγκρίνουμε τις υποψήφιες λύσεις μεταξύ τους, ως προς την τιμή της αντικειμενικής συνάρτησης κάθε λύσης. Τα προβλήματα βελτιστοποίησης που θέλουμε να επιλύσουμε μπορεί να είναι, είτε σε μορφή ελαχιστοποίησης μιας συνάρτησης κόστους (cost function), είτε σε μορφή μεγιστοποίησης μια συνάρτησης κέρδους ή αντικειμενικής συνάρτησης (profit function ή objective function). Άρα ο στόχος της συνάρτησης αξιολόγησης, είναι η ελαχιστοποίηση ή η μεγιστοποίηση της τιμής της αντίστοιχης συνάρτησης και εξαρτάται από το εκάστοτε πρόβλημα που θέλουμε να επιλύσουμε. Αξίζει να σημειωθεί, ότι η αντικειμενική συνάρτηση που προαναφέραμε σε προηγούμενο κεφάλαιο ταυτίζεται με την συνάρτηση αξιολόγησης, εφόσον το πρόβλημα που επιλύεται, είναι πρόβλημα βελτιστοποίησης.

2.2.4 Επιλογή

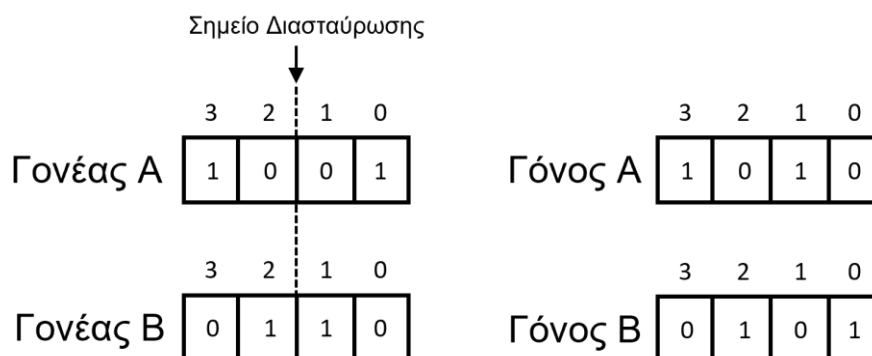
Η διαδικασία της επιλογής μιμείται την φυσική επιλογή, η οποία γίνεται με βάση την απόδοση, δηλαδή την τιμή αξιολόγησης του κάθε ατόμου του πληθυσμού. Όσο καλύτερη είναι η τιμή αξιολόγησης του ατόμου, τόσο μεγαλύτερη πιθανότητα έχει να επιλεγεί. Τα άτομα που επιλέγονται εισέρχονται σε μια «πισίνα ζευγαρώματος» (mating pool). Η πιο κοινή μέθοδος για την εισαγωγή των ατόμων σε αυτή την «πισίνα» είναι μέσω της διαδικασίας γνωστή ως αναλογική επιλογή (proportional selection) ή αλλιώς μέθοδος ρουλέτας (roulette wheel selection). Αν υποθέσουμε ότι η τιμή αξιολόγησης είναι f_i , τότε με αυτή τη μέθοδο υπολογίζεται η σχετική τιμή αξιολόγησης (relative fitness value) του κάθε ατόμου ως εξής:

$$relative\ fitness\ value = \frac{f_i}{\sum_{i=1}^{popsize} f_i} \quad (5)$$

Επειδή $\sum_{i=1}^{popsize} relative\ fitness\ value = 1$, μπορούμε να πούμε ότι η σχετική τιμή αξιολόγησης είναι η πιθανότητα να επιλεγθεί το εκάστοτε άτομο [14].

2.2.5 Διασταύρωση

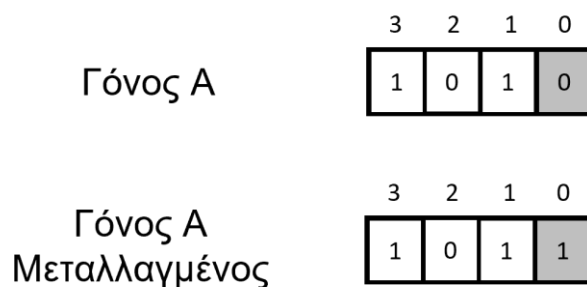
Η διαδικασία της διασταύρωσης (crossover) (ή επιχiasμού) ή αλλιώς του ανασυνδυασμού (recombination) δημιουργεί δύο ή περισσότερους γόνους από ένα ζευγάρι γονέων που έχουν επιλεγθεί από τον «πισίνα ζευγαρώματος». Εφόσον έχει επιλεγθεί το ζευγάρι, κατόπιν επιλέγεται τυχαία ένα ή περισσότερα σημεία του χρωμοσώματος από τα οποία θα γίνει ανταλλαγή των γονιδίων αυτού του ζευγαριού. Οι γόνοι που δημιουργούνται έχουν χαρακτηριστικά και από τους δύο γονείς, αυξάνοντας έτσι την εξερεύνηση (exploration) του αλγορίθμου. Αποτέλεσμα αυτού είναι ότι οι γόνοι μπορούν να ξεπεράσουν ακόμα και τους γονείς τους, όσον αφορά την εύρεση της βέλτιστης λύσης. Για παράδειγμα, αν ο ένας γονέας έχει χρωμόσωμα 1001 και ο άλλος 0110 και επιλεγθεί τυχαία μόνο ένα σημείο (single-point crossover), ας υποθέσουμε το σημείο μεταξύ της θέσης 1 και 2 της [Εικόνας 6](#). Από το σημείο αυτό, θα γίνει ανταλλαγή γονιδίων μεταξύ των χρωμοσωμάτων και θα δημιουργηθούν δύο γόνοι, ο ένας με χρωμόσωμα 1010 και ο άλλος με χρωμόσωμα 0101.



Εικόνα 6. Διαδικασία Διασταύρωσης και δημιουργία γόνων.

2.2.6 Μετάλλαξη

Ο μετάλλαξη εφαρμόζεται μετά την διασταύρωση, δηλαδή στους απογόνους και βασίζεται σε τυχαίες αλλαγές των γονιδίων του χρωμοσώματος. Η πιθανότητα να μεταλλαχθεί ένα γονίδιο του χρωμοσώματος ονομάζεται ρυθμός μετάλλαξης (mutation rate). Συνεχίζοντας το παράδειγμα, αν πάρουμε τους γόνους από την [Εικόνα 6](#) κάθε γονίδιο παίρνει τυχαία μια πιθανότητα και αν αυτή είναι μικρότερη του ρυθμού μετάλλαξης, τότε το γονίδιο αυτό μεταλλάσσεται. Εφόσον στο παράδειγμα, τα γονίδια παίρνουν τιμές 0,1, έχουμε δύο περιπτώσεις μετάλλαξης, από 0 σε 1 και από 1 σε 0. Σε αυτή την περίπτωση, η μετάλλαξη ονομάζεται bit-flip mutation [14] ([Εικόνα 7](#)).



Εικόνα 7. Μετάλλαξη γόνου Α.

2.2.7 Αντικατάσταση

Στη διαδικασία της αντικατάστασης οι απόγονοι που έχουν δημιουργηθεί αντικαθιστούν τους γονείς τους [15]. Μέσω του ελιτισμού (elitism) επιλέγονται άτομα του πληθυσμού των γονέων και των απογόνων με τις καλύτερες τιμές αξιολόγησης, ώστε να περάσουν στην επόμενη γενιά τα άτομα με τα καλύτερα χαρακτηριστικά. Υπάρχουν και άλλες τεχνικές επιλογής των ατόμων για αντικατάσταση, όπως για παράδειγμα η αντικατάσταση με βάση την ηλικία, όπου γίνεται επιλογή των νεότερων ατόμων για την μετάβαση στην επόμενη γενιά.

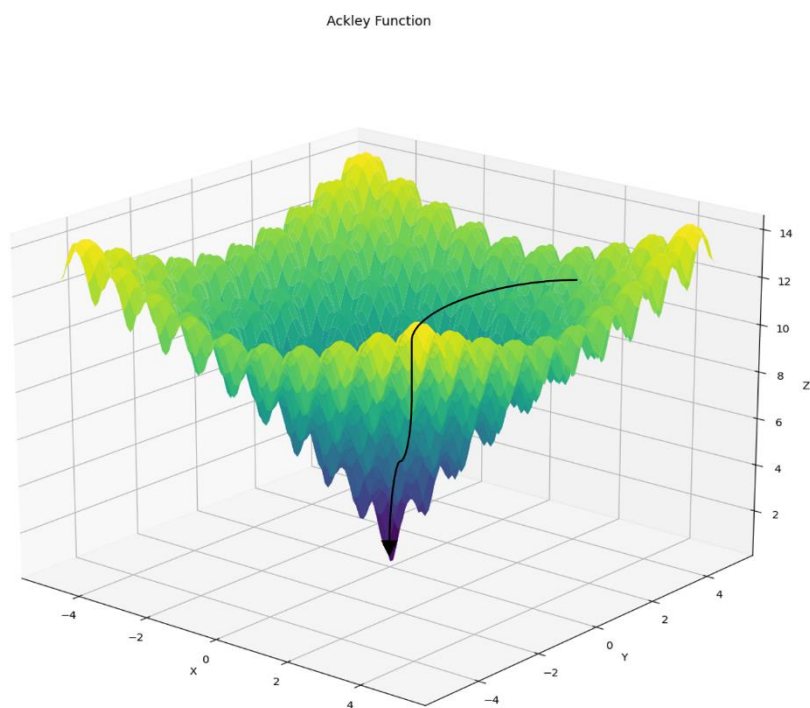
2.2.8 Κριτήριο Τερματισμού

Το κριτήριο τερματισμού αναφέρεται στο πότε θα σταματήσει να λειτουργεί ο αλγόριθμος. Ένας τρόπος που θα σκεφτόταν κανείς είναι να χρησιμοποιήσει μια τιμή σφάλματος, κάτω από το οποίο θεωρούμε ότι η λύση μας είναι βέλτιστη και έτσι να σταματάει ο αλγόριθμος. Αυτό όμως μπορεί να οδηγήσει σε έναν ατελείωτο βρόγχο, αν το σφάλμα δεν φτάσει ποτέ στα ικανοποιητικά επίπεδα. Γι' αυτόν το λόγο συνήθως δεν χρησιμοποιούνται τέτοιου είδους κριτήρια τερματισμού. Αντ' αυτού χρησιμοποιείται ο αριθμός των γενιών, δηλαδή οι επαναλήψεις που θα τρέξει ο βασικός αλγόριθμος, ο οποίος ορίζεται από την αρχή. Έτσι, είτε το σφάλμα πέσει κάτω από το όριο που θέλουμε, είτε όχι, ο αλγόριθμος θα σταματήσει κάποια στιγμή. Ένα άλλο κριτήριο τερματισμού, είναι μέσω του ρυθμού σύγκλισης (convergence rate), όπου αν η πρόοδος προς το βέλτιστο αποτέλεσμα είναι μηδαμινή, τότε ο αλγόριθμος σταματάει [7].

2.3 Διαφορικός Εξελικτικός Αλγόριθμος

Ο διαφορικός εξελικτικός αλγόριθμος (Differential Evolution, DE) είναι μία στοχαστική μέθοδος βελτιστοποίησης, η οποία προτάθηκε από τους R. Storn και K.V. Price το 1995 και χρησιμοποιείται για την επίλυση πολυδιάστατων προβλημάτων [16]. Διαφέρει από τους ΓΑ στο ότι ο πληθυσμός είναι N -διάστατα διανύσματα πραγματικών αριθμών και επίσης χρησιμοποιεί τους τελεστές της μετάλλαξης, της διασταύρωσης και της επιλογής, για την εύρεση της βέλτιστης λύσης, αλλά με παραλλαγές. Η ευκολία στη χρήση, λόγω του ότι διαθέτει λίγες ρυθμιστικές παραμέτρους, η απλότητα στον προγραμματισμό του, αλλά και η πολύ καλή απόδοσή του, λόγω της γρήγορης και ακριβούς σύγκλισης προς τη βέλτιστη λύση, τον καθιστούν ένα σημαντικό εργαλείο βελτιστοποίησης. Ο αλγόριθμος μπορεί να εύρει με επιτυχία τη βέλτιστη λύση σε πολλά προβλήματα, συμπεριλαμβανομένου και της συνάρτησης Ackley (Εικόνα 8), η οποία έχει πολλά τοπικά ελάχιστα, αλλά ένα μόνο ολικό ελάχιστο. Η γενική φιλοσοφία

του αλγορίθμου βασίζεται στην επιλογή ενός διανύσματος διαφοράς δύο τυχαίων ατόμων του πληθυσμού, πολλαπλασιάζοντας το με ένα σταθερό αριθμό και προσθέτοντας το σε ένα τρίτο τυχαίο άτομο, δημιουργώντας έτσι ένα νέο άτομο. Αυτό το νέο άτομο υπόκειται τη διαδικασία της διασταύρωσης μαζί με το αντίστοιχο αρχικό άτομο. Αυτό έχει ως αποτέλεσμα τη δημιουργία ενός υποψήφιου τελικού ατόμου, το οποίο συγκρίνεται με το αρχικό και περνάει στην επόμενη γενιά το καλύτερο από τα δύο.



Εικόνα 8. Εύρεση ολικού ελάχιστου με χρήση διαφορικού εξελικτικού αλγορίθμου.

Ειδικότερα, ο Διαφορικός Εξελικτικός αλγόριθμος έχει σχεδόν την ίδια δομή με τον ΓΑ, αλλά με ιδιαιτερότητες στην μορφή του πληθυσμού και των τελεστών της μετάλλαξης, της διασταύρωσης και της επιλογής. Παρακάτω γίνεται η ανάλυση αυτών των ιδιαιτεροτήτων.

2.3.1 Πληθυσμός

Στον ΔΕΑ δημιουργούνται τρεις διαφορετικοί πληθυσμοί σε κάθε γενιά, οι οποίοι είναι ο αρχικός, ο ενδιάμεσος και ο τελικός πληθυσμός. Ο αρχικός πληθυσμός (current population) συμβολίζεται με P_x και αποτελείται από τα άτομα ή αλλιώς τις υποψήφιες λύσεις:

$$\begin{aligned} x_{i,g}, \quad i &= 1, \dots, N_p, \\ g &= 1, \dots, g_{max}, \end{aligned} \quad (6)$$

όπου i είναι ένας δείκτης που δείχνει το εκάστοτε άτομο και g είναι η γενιά στην οποία ανήκει. Ο ενδιάμεσος πληθυσμός (intermediary population) συμβολίζεται με P_v και αποτελείται από τα ενδιάμεσα μεταλλαγμένα άτομα (mutant vectors), που συμβολίζονται ως $v_{i,g}$, ενώ ο τελικός πληθυσμός (trial population) συμβολίζεται με P_u και αποτελείται από τα άτομα δοκιμής (trial vectors) που συμβολίζονται ως $u_{i,g}$.

2.3.2 Αρχικοποίηση

Όπως και στον ΓΑ, έτσι και στον ΔΕΑ χρειάζεται να δώσουμε κάποιες αρχικές τιμές στα άτομα του πληθυσμού. Αυτές οι τιμές πρέπει να είναι τυχαίες, με ομοιόμορφα κατανεμημένη πιθανότητα δημιουργίας τους, αλλά και εντός του χώρου των εφικτών λύσεων. Για να βεβαιωθούμε ότι αυτοί οι τυχαίοι αριθμοί θα βρίσκονται πάντα εντός των εφικτών λύσεων, ορίζουμε τα άνω και κάτω όρια μέσω δύο N-διάστατων διανυσμάτων, b_U και b_L αντίστοιχα. Έτσι, το γονίδιο του κάθε ατόμου, ορίζεται αρχικά ως εξής:

$$x_{j,i,0} = \text{random uniform}(b_L, b_U), \quad (7)$$

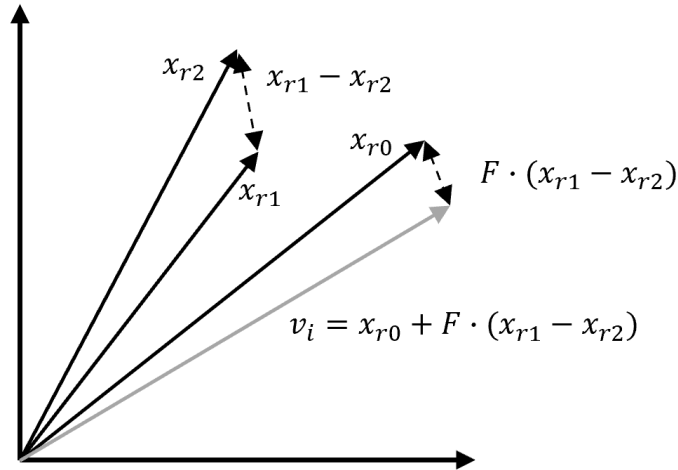
όπου i είναι το άτομο, j είναι το γονίδιο του ατόμου και το 0 συμβολίζει την αρχική γενιά.

2.3.3 Μετάλλαξη

Στη διαδικασία της μετάλλαξης, επιλέγονται για κάθε ένα μέλος του πληθυσμού τρία τυχαία άτομα, διαφορετικά μεταξύ τους. Κατόπιν, υπολογίζεται η διανυσματική διαφορά των δύο από αυτά και προστίθεται στο τρίτο άτομο (δότης – donor), εφόσον έχει πολλαπλασιαστεί πρώτα με μια ρυθμιστική παράμετρο F . Το αποτέλεσμα είναι η δημιουργία ενός νέου (ενδιάμεσου) μεταλλαγμένου ατόμου.

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}) \quad (8)$$

Το άτομο $x_{r0,g}$ καλείται διάνυσμα βάσης (base vector), ενώ η διαφορά των $x_{r1,g}$, $x_{r2,g}$, καλείται διάνυσμα διαφοράς (difference vector). Οι δείκτες r_0, r_1, r_2 εκτός από το ότι είναι διαφορετικοί μεταξύ τους, είναι διαφορετικοί και από τον δείκτη i , δηλαδή τον δείκτη διανύσματος στόχου (target vector index) (Εικόνα 9). Η ρυθμιστική παράμετρος (scale factor) F , ελέγχει το μέγεθος της διαταραχής που προστίθεται στο δότη, παίρνει θετικές τιμές και συνήθως το εύρος της είναι $F \in [0.4, 0.9]$.

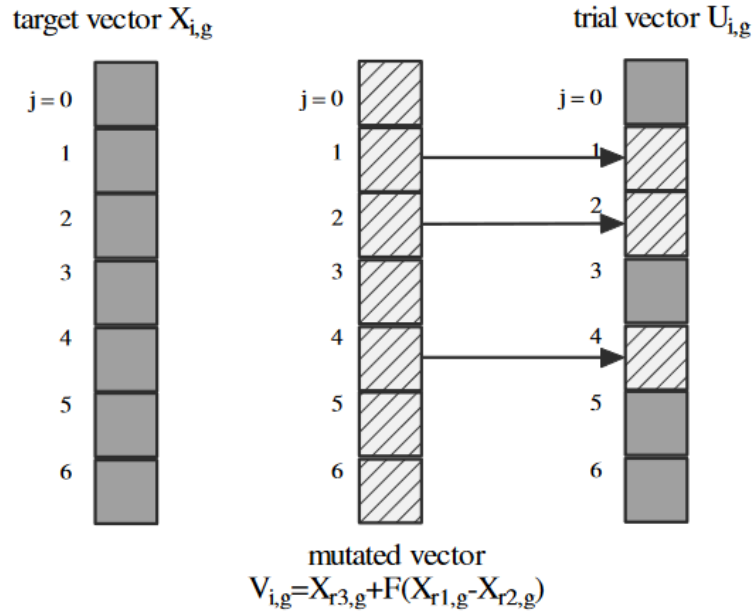


Εικόνα 9. Διαδικασία μετάλλαξης στον διαφορικό εξελικτικό αλγόριθμο.
Πηγή: [Simon Evolutionary Optimization Algorithms]. Figure 12.1. page 294 [10].

2.3.4 Διασταύρωση

Στη φάση της διασταύρωσης (Εικόνα 10) το εκάστοτε και το μεταλλαγμένο άτομο δημιουργούν το άτομο δοκιμής ή αλλιώς διάνυσμα δοκιμής (trial vector). Η πιθανότητα ώστε το κάθε γονίδιο ή αλλιώς παράμετρος του ατόμου δοκιμής να έχει χαρακτηριστικά από το εκάστοτε άτομο $x_{i,g}$ ή το μεταλλαγμένο άτομο $v_{i,g}$ ορίζεται από μια σταθερά, η οποία καλείται πιθανότητα διασταύρωσης (crossover rate) $C_r \in [0, 1]$. Έτσι, σε κάθε παράμετρο (γονίδιο) του ατόμου ορίζεται μια τυχαία τιμή $[0, 1]$ και αν η τιμή αυτή είναι μικρότερη ή ίση από την πιθανότητα διασταύρωσης, τότε το άτομο δοκιμής κληρονομεί την παράμετρο από το μεταλλαγμένο άτομο, αλλιώς την κληρονομεί από το εκάστοτε (αρχικό) άτομο. Σε περίπτωση που ο δείκτης (j_{rand}) της παραμέτρου δοκιμής (τυχαίος ακέραιος με μέγιστη τιμή το πλήθος των γονιδίων) είναι ίδιος με τον δείκτη της εκάστοτε παραμέτρου (θέση του γονιδίου), τότε κληρονομείται στο άτομο δοκιμής η συγκεκριμένη παράμετρος (γονίδιο) του μεταλλαγμένου ατόμου [17]. Αυτό γίνεται ώστε να εξασφαλιστεί ότι τουλάχιστον ένα γονίδιο στο νέο άτομο θα προέρχεται υποχρεωτικά από το μεταλλαγμένο.

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{αν } rand_j \leq C_r \text{ ή } j = j_{rand} \\ x_{j,i,g}, & \text{αλλιώς} \end{cases} \quad (9)$$



Εικόνα 10. Διαδικασία διασταύρωσης.

Πηγή: *New Optimization Techniques in Engineering*. Page 129. [17]

2.3.5 Επιλογή

Ο πληθυσμός της επόμενης γενιάς επιλέγεται από τον εκάστοτε (αρχικό) πληθυσμό P_x και τον τελικό πληθυσμό δοκιμής P_u . Αυτό γίνεται συγκρίνοντας την αντικειμενική συνάρτηση του κάθε ατόμου του αρχικού πληθυσμού, με το αντίστοιχο του στον τελικό πληθυσμό. Σε περίπτωση που το άτομο δοκιμής έχει καλύτερη αντικειμενική συνάρτηση από το αντίστοιχο αρχικό, τότε το αντικαθιστά, παίρνοντας τη θέση του στον πληθυσμό, αλλιώς, το αρχικό παραμένει για μία τουλάχιστον ακόμη γενιά. Επειδή σε κάθε περίπτωση η αντικειμενική συνάρτηση είναι καλύτερη ή ίση με την προηγούμενη γενιά, όλα τα άτομα του πληθυσμού έχουν και αυτά ίδια ή καλύτερα χαρακτηριστικά από τα προηγούμενα.

$$x_{i,g+1} = \begin{cases} u_{i,g}, & f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g}, & \text{αλλιώς} \end{cases} \quad (10)$$

2.3.6 Ψευδοκώδικας ΔΕ

Ο αλγόριθμος μπορεί εύκολα να αντιπροσωπευθεί από τον παρακάτω ψευδοκώδικα (Εικόνα 11), ενώνοντας τους τελεστές που αναλύθηκαν αλλά και προσθέτοντας τα στάδια της αρχικοποίησης και τερματισμού [18].

```
Generate initial population of size NP
Do while
  For each individual, j, in the population
    Generate three random integers,  $r_1, r_2, r_3 \in (1, NP)$ , with  $r_1 \neq r_2 \neq r_3 \neq j$ ,
    Generate a random integer  $i_{rand} \in (1, n)$ 
    For each parameter  $i$ 
      
$$x'_{i,j} = \begin{cases} x_{i,r3} + F * (x_{i,r1} - x_{i,r2}) & \text{if } rand(0,1) < Cr \text{ or } i = i_{rand} \\ x_{i,j} & \text{otherwise} \end{cases}$$

    End For
    Replace  $x_j$  with child  $x'_j$ , if  $x'_j$  is better
  End For
Until the termination condition is satisfied
```

Εικόνα 11. Ψευδοκώδικας ΔΕ [18].

Κεφάλαιο 3 – Δημιουργία Παραμετρικών Καμπυλών και Επιφανειών

Η γεωμετρική μοντελοποίηση είναι η μαθηματική αναπαράσταση της γεωμετρίας ενός αντικειμένου σε διδιάστατη ή/και τριδιάστατη μορφή. Τη σήμερον ημέρα, τα φυσικά αντικείμενα έχουν αντικατασταθεί από τα υπολογιστικά μοντέλα σε πολύ μεγάλο βαθμό [19]. Αυτό έχει ως αποτέλεσμα τα καινούρια προϊόντα που παράγονται να είναι καλύτερα και φθηνότερα, καθώς η ανάλυση και οι αλλαγές γίνονται στα τριδιάστατα μοντέλα, όντας διαδικασίες πιο απλές σε σχέση με τα φυσικά αντικείμενα.

Με την χρήση υπολογιστικών (ψηφιακών) μοντέλων, οι σχεδιαστές και οι μηχανικοί μπορούν να δημιουργήσουν ακριβείς ψηφιακές αναπαραστάσεις των προϊόντων. Αυτό δίνει τη δυνατότητα για εκτέλεση έγκυρων και προηγμένων αναλύσεων, όπως για παράδειγμα είναι η προσομοίωση της συμπεριφοράς του προϊόντος σε διάφορες τιμές φορτίσεων ή/και των διευθύνσεων τους, που μπορεί να δέχεται το αντικείμενο και εκτίμηση της απόδοσής του σε αυτές. Επίσης, η ευκολία τροποποίησης των ψηφιακών μοντέλων επιτρέπει τη γρήγορη εφαρμογή αλλαγών και έτσι την ανάπτυξη πολλών και παρόμοιων σχεδιαστικών επιλογών.

Η παραγωγή φυσικών πρωτοτύπων και προσωρινών φυσικών μοντέλων μπορεί να είναι ακριβή και χρονοβόρα. Με τη χρήση υπολογιστικών μοντέλων, ο σχεδιαστής μπορεί να τροποποιήσει και να κάνει δοκιμές στο ψηφιακό μοντέλο, πριν φτάσει στο στάδιο της ταχείας προτυποποίησης. Αυτό μειώνει δραματικά τον κίνδυνο σφαλμάτων και αποτυχιών, τον χρόνο ανάπτυξης νέου προϊόντος ή αλλαγής υπάρχοντος προϊόντος, αλλά και το κόστος ανάπτυξης και παραγωγής αυτού.

Η χρήση υπολογιστικών μοντέλων επιτρέπει επίσης την ανταλλαγή και την αποθήκευση πληροφοριών με ευκολία. Ψηφιακά μοντέλα μπορούν να μεταφερθούν ηλεκτρονικά και να αποθηκευτούν σε ψηφιακούς αποθηκευτικούς χώρους. Αυτό διευκολύνει την κοινή εργασία, την ανταλλαγή σχεδίων και πληροφοριών μεταξύ

ομάδων σχεδιαστών και μηχανικών, και τη διατήρηση ενημερωμένων εκδόσεων των μοντέλων.

Η γεωμετρική μοντελοποίηση χρησιμοποιείται στον κλάδο της βιομηχανίας για τον σχεδιασμό και την παραγωγή προϊόντων, επιτρέποντας στους μηχανικούς να αναλύσουν και να βελτιστοποιήσουν τα προϊόντα τους πριν την παραγωγή τους. Στον κλάδο της αρχιτεκτονικής δίνεται η δυνατότητα να δημιουργηθούν ακριβείς αναπαραστάσεις κτηρίων, επιτρέποντας την προσομοίωση διάφορων φυσικών φαινομένων, όπως η αντοχή σε σεισμούς. Στον κλάδο της ψυχαγωγίας, όπως τα βιντεοπαιχνίδια χρησιμοποιείται στη δημιουργία των ψηφιακών κόσμων, των χαρακτήρων και αντικειμένων τους, ενώ στη ιατρική μπορεί να χρησιμοποιηθεί για την ανάλυση και σχεδίαση ιατρικών εμφυτευμάτων.

Οι μαθηματικές αυτές αναπαραστάσεις των μοντέλων γίνονται με συστήματα CAD (Computer Aided Design), με τη χρήση συγκεκριμένων τεχνικών. Στη συνέχεια, γίνεται μια σύντομη περιγραφή αυτών των τεχνικών, που πλέον είναι ο κύριος τρόπος αναπαράστασης των καμπυλών και επιφανειών, τόσο των βασικών (τυποποιημένων) γεωμετρικών σχημάτων, όσο και των σχημάτων ελεύθερης μορφής.

3.1 Καμπύλες και Επιφάνειες Bezier

Οι καμπύλες Bezier δημιουργήθηκαν από τον Γάλλο μηχανικό και μαθηματικό Pierre Etienne Bezier (1910-1999), ο οποίος εργαζόταν τότε στη Renault. Με αυτές ο Bezier μπορούσε να δημιουργήσει ελεύθερης μορφής καμπύλες και να περιγράψει τα σχέδια των αυτοκινήτων. Οι καμπύλες αυτές εντάσσονται στην οικογένεια των παραμετρικών καμπυλών και έχουν τις εξής ιδιότητες [20]:

- Οι συναρτήσεις βάσεις είναι πραγματικοί αριθμοί.
- Ο βαθμός της καμπύλης είναι κατά έναν αριθμό μικρότερος από τον αριθμό των σημείων του πολυγώνου ελέγχου.

- Το πρώτο και το τελευταίο σημείο της καμπύλης ταυτίζεται με το πρώτο και το τελευταίο σημείο του πολυγώνου ελέγχου αντίστοιχα.

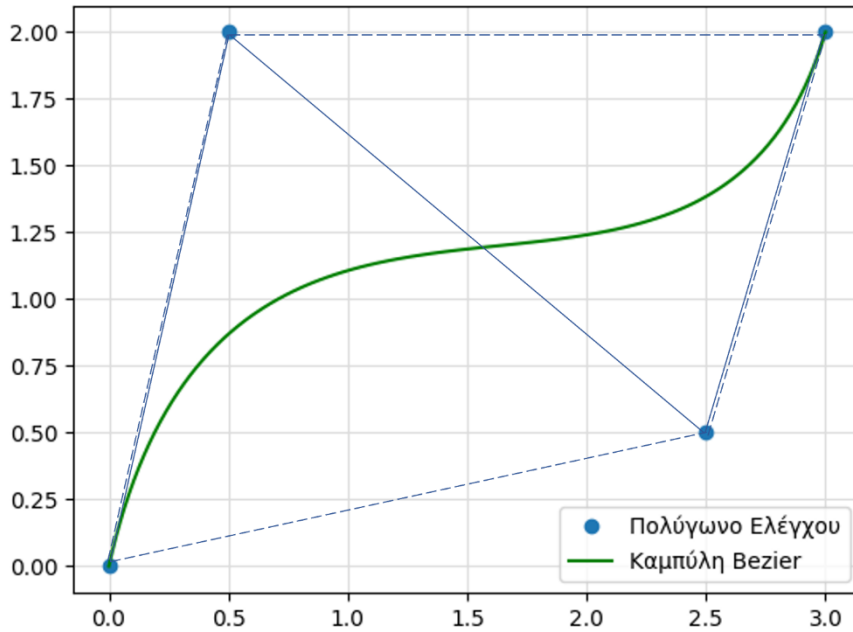
Ορίζονται ως εξής:

$$C(u) = \sum_{i=0}^n B_{i,n}(u) P_i, \quad 0 \leq u \leq 1 \quad (11)$$

όπου n είναι ο βαθμός της καμπύλης, $B_{i,n}(u)$ είναι το πολυώνυμο Bernstein που ορίζεται ως

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (12)$$

Οι μεταβλητές P_i είναι τα σημεία ελέγχου, ενώ το u είναι μια μεταβλητή που παίρνει τιμές από 0 έως 1 (Εικόνα 12).



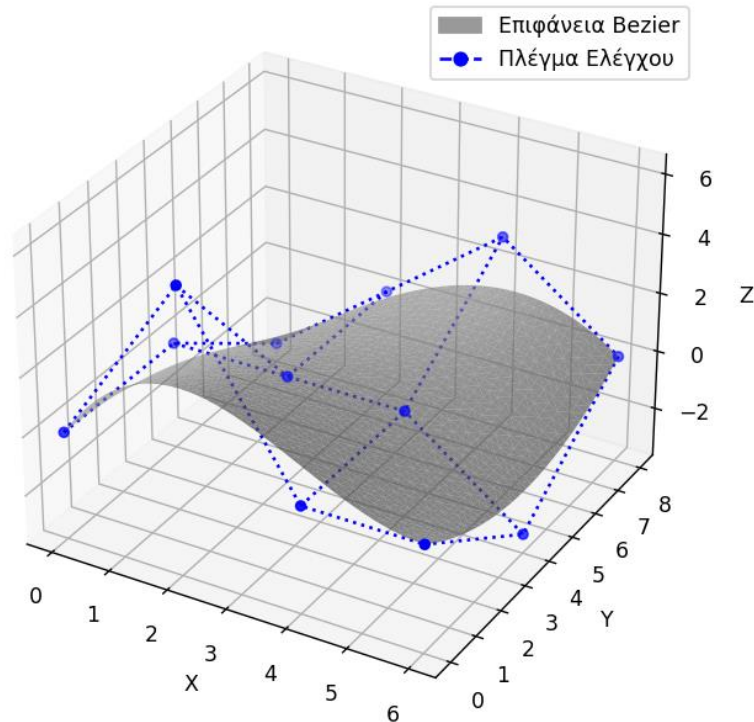
Εικόνα 12. Καμπύλη Bezier 3ου βαθμού μαζί με το πολύγωνο ελέγχου.

Με παρόμοιο τρόπο οι επιφάνειες Bezier ορίζονται ως

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) P_{i,j}, \quad 0 \leq u, v \leq 1 \quad (13)$$

όπου $B_{i,n}(u)$ και $B_{j,m}(v)$ είναι τα πολυώνυμα Bernstein για την u και v διεύθυνση αντίστοιχα (Εικόνα 13). Το πολυώνυμο $B_{i,n}(u)$ ορίζεται ακριβώς όπως και παραπάνω, ενώ το πολυώνυμο $B_{j,m}(v)$ ορίζεται ως

$$B_{j,m}(v) = \frac{m!}{j!(m-j)!} u^j (1-u)^{m-j} \quad (14)$$



Εικόνα 13. Επιφάνεια Bezier, όπου οι διευθύνσεις u και v είναι $3^{\text{ου}}$ και $2^{\text{ου}}$ βαθμού αντίστοιχα.

Επειδή οι επιφάνειες Bezier βασίζονται στο πολυώνυμο Bernstein, παρουσιάζουν τις εξής ιδιότητες [20]:

- Ο βαθμός της επιφάνειας σε κάθε διεύθυνση είναι κατά έναν αριθμό μικρότερος από τον αριθμό των σημείων ελέγχου στην αντίστοιχη διεύθυνση.
- Η επιφάνεια βρίσκεται εντός του πλέγματος ελέγχου που σχηματίζεται από τα σημεία ελέγχου στον τριδιάστατο χώρο.
- Τα σημεία της επιφάνειας που βρίσκονται στις γωνίες ταυτίζονται με τα σημεία ελέγχου.
- Η επιφάνεια προσεγγίζει τη μορφή του πλέγματος ελέγχου.

3.2 Καμπύλες και Επιφάνειες B-Splines

Οι καμπύλες Bezier περιορίζουν την ευελιξία του σχεδιασμού σε δύο βασικά χαρακτηριστικά. Πρώτον, ο βαθμός της καμπύλης περιορίζεται από τον αριθμό των σημείων ελέγχου. Έτσι, αν θέλουμε να αυξήσουμε ή να μειώσουμε τον βαθμό της καμπύλης, πρέπει να προσθέσουμε ή να αφαιρέσουμε ένα ή περισσότερα σημεία ελέγχου αντίστοιχα. Δεύτερον, οι αλλαγές που κάνουμε στον αριθμό ή στην θέση των σημείων ελέγχου, επηρεάζουν ολόκληρη την καμπύλη, με αποτέλεσμα να μην έχουμε τη δυνατότητα τοπικού ελέγχου της καμπύλης. Για να ξεπεραστεί αυτό, χρειάστηκε να εφαρμοστεί στις καμπύλες Bezier ενός άλλου τύπου συναρτήσεις βάσης, οι λεγόμενες B-Splines (Basis splines) που δημιουργήθηκαν από τους Cox [21] και deBoor [22] (Εικόνα 14). Αργότερα οι Riesenfeld [23] και Gorgon [24] εφάρμοσαν αυτές τις συναρτήσεις και όρισαν τις καμπύλες B-Splines ως

$$C(u) = \sum_{i=0}^n N_{i,p}(u)P_i, \quad a \leq u \leq b \quad (15)$$

όπου n είναι ο αριθμός των σημείων ελέγχου, P_i είναι τα διανύσματα θέσης των σημείων ελέγχου και $N_{i,p}(u)$ είναι οι συναρτήσεις βάσης B-Spline βαθμού p , ορισμένα στο διάνυσμα κόμβων

$$U = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\} \quad (16)$$

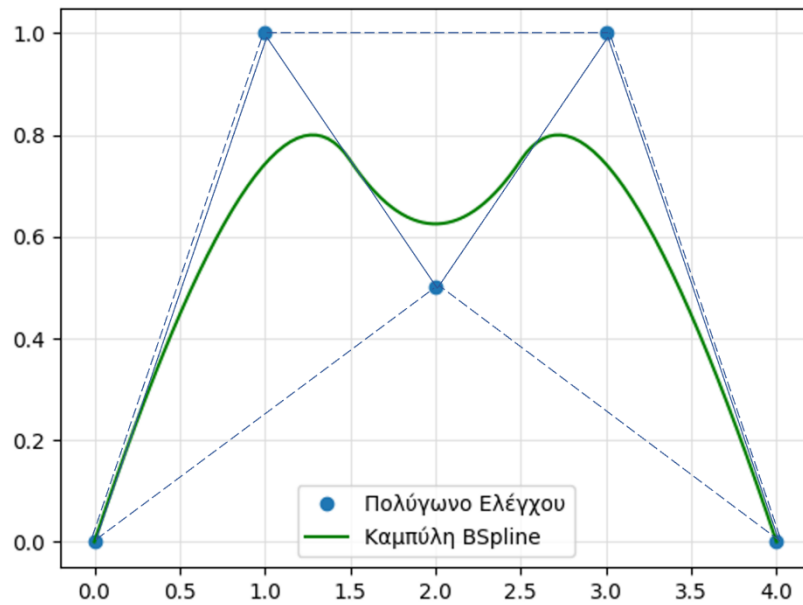
Το διάνυσμα κόμβων παίρνει πραγματικές τιμές κατά αύξουσα σειρά, δηλαδή $u_i \leq u_{i+1}$, και επηρεάζει σημαντικά τις συναρτήσεις βάσης B-Spline και κατ' επέκταση τη μορφή της καμπύλης.

Με παρόμοιο τρόπο οι επιφάνειες B-Splines ορίζονται ως

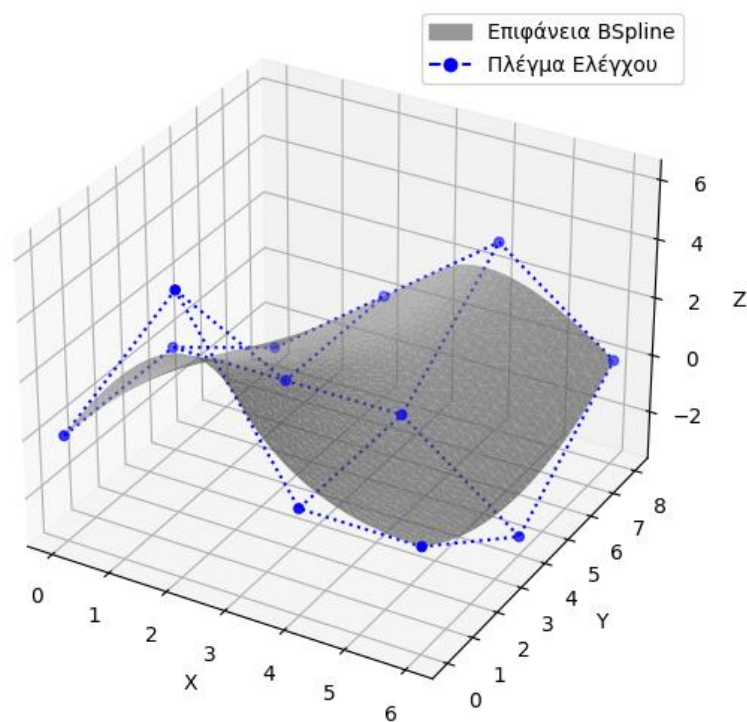
$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v)P_{i,j}, \quad a \leq u, v \leq b \quad (17)$$

όπου $N_{i,p}(u)$ και $N_{j,q}(v)$ είναι οι συναρτήσεις βάσης για την u και v διεύθυνση αντίστοιχα. Στην Εικόνα 15 φαίνεται μια επιφάνεια B-Spline, όπου οι διευθύνσεις u

και v είναι $2^{\text{ου}}$ βαθμού, αλλά ο αριθμός των σημείων ελέγχου είναι 4 προς την u διεύθυνση και 3 προς τη v διεύθυνση.



Εικόνα 14. Καμπύλη B-Spline $2^{\text{ου}}$ βαθμού με πέντε σημεία ελέγχου.



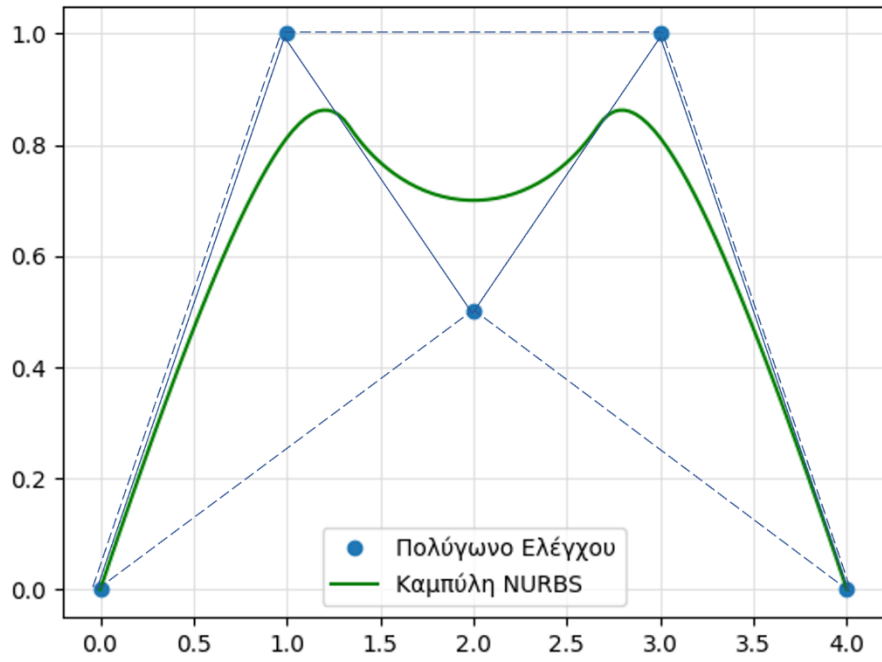
Εικόνα 15. Επιφάνεια B-Spline, όπου οι διευθύνσεις u και v είναι $2^{\text{ου}}$ βαθμού.

3.3 Καμπύλες και Επιφάνειες NURBS

Οι καμπύλες NURBS (Non Uniform Rational B-Splines) είναι «προέκταση» των καμπυλών B-Spline, οι οποίες προσφέρουν επιπλέον δυνατότητες, εκτός από τις υπάρχουσες που προσφέρουν οι καμπύλες B-Spline. Σε κάθε σημείο ελέγχου συνδέεται και ένα βάρος (weight), το οποίο στην ουσία καθορίζει το βαθμό «σημαντικότητάς» του. Αυτό δίνει μεγαλύτερο έλεγχο στο σχήμα της καμπύλης ή της επιφάνειας. Επίσης, μπορούν να είναι και ρητές (Rational), το οποίο επιτρέπει την ακριβή αναπαράσταση κωνικών τομών, όπως είναι ο κύκλος ή η έλλειψη, κάτι που δεν μπορούσε να γίνει με τις καμπύλες B-Splines. Οι καμπύλες NURBS (Εικόνα 16), ορίζονται ως

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i P_i}{\sum_{i=0}^n N_{i,p}(u)w_i}, \quad a \leq u \leq b \quad (18)$$

Όπου τώρα προστέθηκε ο όρος w_i , ο οποίος είναι το βάρος του εκάστοτε σημείου ελέγχου, ενώ επίσης διαιρείται με $\sum_{i=0}^n N_{i,p}(u)w_i$ για να υπολογιστεί το σημείο της καμπύλης. Όπως και στις B-Splines, οι συναρτήσεις βάσης είναι ορισμένες σε ένα διάστημα κόμβων (Εξ. 16).



Εικόνα 16. Καμπύλη NURBS 2^{ου} βαθμού με πέντε σημεία ελέγχου, όπου το βάρος του κεντρικού σημείου ελέγχου είναι μισό σε σχέση με τα υπόλοιπα.

Παρόμοια οι επιφάνειες NURBS ορίζονται ως

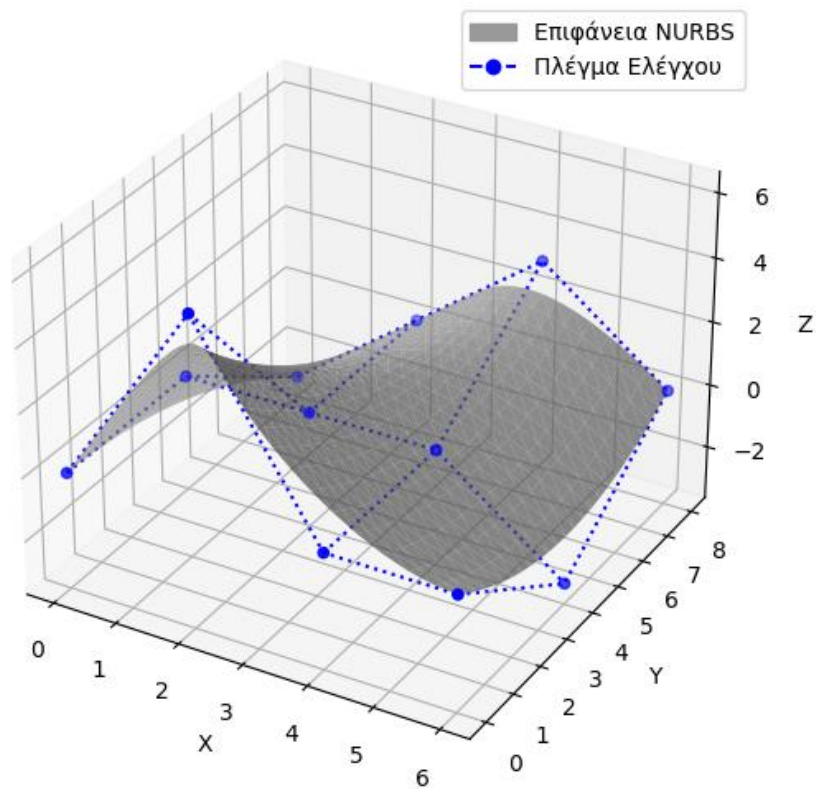
$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}}, \quad 0 \leq u, v \leq 1 \quad (19)$$

Ορισμένες στα διανύσματα κόμβων και των δύο διευθύνσεων u, v .

$$U = \{a, \dots, a, u_{p+1}, \dots, u_{r-p-1}, b, \dots, b\} \quad (20)$$

$$V = \{a, \dots, a, u_{q+1}, \dots, u_{s-q-1}, b, \dots, b\} \quad (21)$$

όπου $r = n + p + 1$ και $s = m + q + 1$. Στην παρακάτω [Εικόνα 17](#) φαίνεται μια επιφάνεια NURBS, όπου τα σημεία που δημιουργούν το πλέγμα ελέγχου είναι στις ίδιες θέσεις με αυτά στην [Εικόνα 15](#), με τη διαφορά ότι το σημείο ελέγχου με συντεταγμένες (2, 0, 6) έχει τριπλάσιο βάρος από τα υπόλοιπα. Αυτό έχει ως αποτέλεσμα, το σημείο αυτό να έχει μεγαλύτερο βαθμό «σημαντικότητας», και έτσι η επιφάνεια να πλησιάζει περισσότερο το σημείο συγκριτικά με την προηγούμενη.



Εικόνα 17. Επιφάνεια NURBS, όπου οι διευθύνσεις u και v είναι $2^{\text{ου}}$ βαθμού, με το σημείο ελέγχου $(2,0,6)$ να έχει μεγαλύτερο βάρος.

Κεφάλαιο 4 – Εφαρμογή

Στο παρόν κεφάλαιο περιγράφεται η μεθοδολογία που ακολουθήθηκε για την επίλυση των προβλημάτων, καθώς επίσης γίνεται και ο ορισμός των προβλημάτων αυτών, τα οποία ως επί το πλείστον αφορούν σε γνωστές μαθηματικές συναρτήσεις επιφανειών.

4.1 Μεθοδολογία

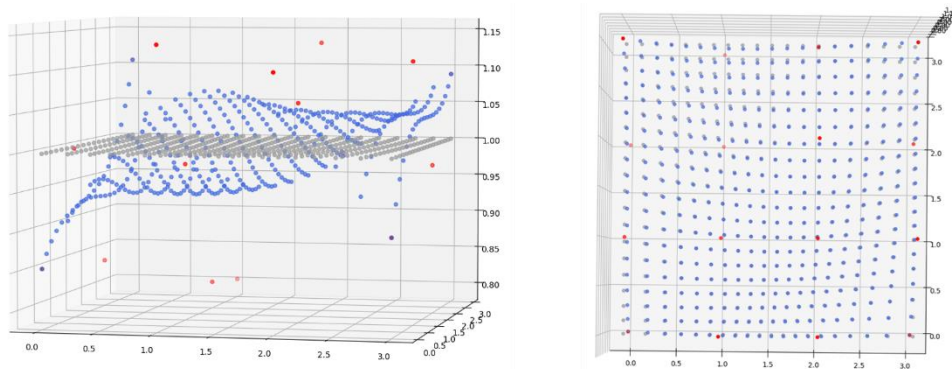
Ο τρόπος με τον οποίο θα προσεγγίσουμε τις επιφάνειες στόχους, που θα ορίσουμε στην επόμενη ενότητα, είναι ελαχιστοποιώντας το μέσο τετραγωνικό σφάλμα (Mean Squared Error, MSE), μεταξύ της επιφάνειας στόχου και της επιφάνειας NURBS, μέσω του Διαφορικού Εξελικτικού Αλγόριθμου, αλλά με μια τροποποίηση. Η τροποποίηση αυτή, αφορά ένα νέο σχήμα δότη, το οποίο προτάθηκε από τον Fan Hui-Yuan [25] και προσφέρει ταχύτερη σύγκλιση, χωρίς να θυσιάζεται η ακρίβεια της λύσης. Συγκεκριμένα, ο δότης θα ορίζεται ως

$$donor = v_{i,g} = \sum_{i=1}^3 \left\{ \frac{\lambda_i}{\sum_{j=1}^3 \lambda_j} \right\} \cdot x_{ri,g}, \quad \lambda_j = rand_j[0,1] \quad (22)$$

Ο αριθμός των σημείων ελέγχου των επιφανειών NURBS παραμένει σταθερός σε όλες τις δοκιμές και ίσος με 25, ενώ οι θέσεις αυτών θα παραμένουν και αυτές σταθερές στους άξονες x, y, με ίσες αποστάσεις μεταξύ τους στο διάστημα από -1 έως 1. Άρα, το μέσο τετραγωνικό σφάλμα θα υπολογίζεται μόνο από τη διαφορά που θα έχει το κάθε σημείο της επιφάνειας στόχου με το αντίστοιχο της επιφάνειας NURBS. Επίσης, ο βαθμός των διευθύνσεων u, v θα είναι σε όλες τις δοκιμές ίσος με τρία.

Στον υπολογιστή, οι επιφάνειες NURBS, καθώς και όλες οι επιφάνειες, είναι στην πραγματικότητα πολλά διακριτά σημεία, τα οποία ενώνονται με τα γειτονικά τους με ευθύγραμμα τμήματα για να δημιουργήσουν την επιφάνεια που απεικονίζεται. Αυτό

το χαρακτηριστικό δημιουργεί προβλήματα στον σωστό υπολογισμό της απόστασης στον άξονα z , μεταξύ ενός σημείου της επιφάνειας στόχου με το αντίστοιχο σημείο του στην επιφάνεια NURBS, διότι κατά πάσα πιθανότητα αυτά να μην έχουν τις ίδιες συντεταγμένες x , y και έτσι να υπολογίζεται η απόσταση στον άξονα z με δύο διαφορετικά σημεία. Το πρόβλημα αυτό, επιλύεται χρησιμοποιώντας τη μέθοδο της γραμμικής παρεμβολής (linear interpolation), ώστε για κάθε σημείο της επιφάνειας NURBS, να υπολογίζεται το αντίστοιχο του στην επιφάνεια στόχο. Δηλαδή, για τις x , y συντεταγμένες του εκάστοτε σημείου της επιφάνειας NURBS, υπολογίζονται οι συντεταγμένες x , y της επιφάνειας στόχου και έτσι πλέον μπορεί να γίνει σωστά η εύρεση της διαφοράς αυτών των σημείων στον άξονα z . Στην παρακάτω [Εικόνα 18](#), απεικονίζονται η επιφάνεια στόχος και NURBS με διακριτά σημεία. Από την κάτοψη, μπορούμε να παρατηρήσουμε ότι οι x , y συντεταγμένες είναι ίδιες για κάθε σημείο της επιφάνειας NURBS και το αντίστοιχο του στην επιφάνεια στόχο.



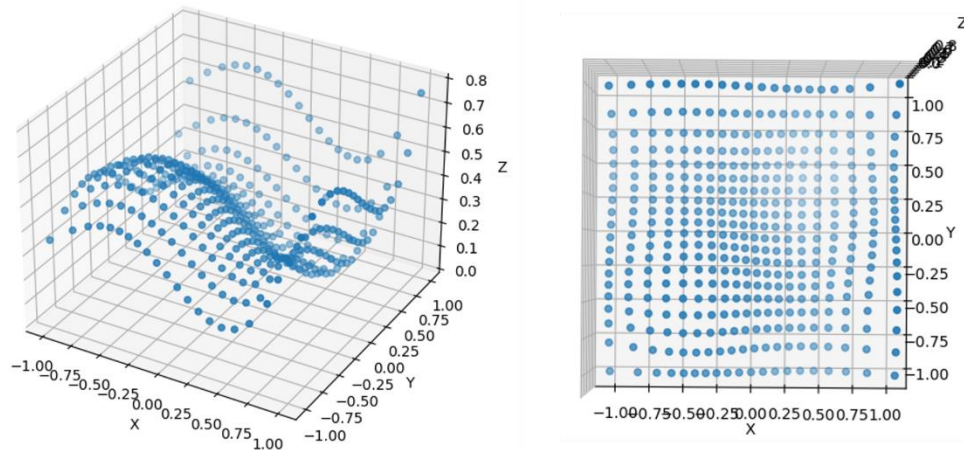
Εικόνα 18. Επιφάνεια στόχος και NURBS με χρήση διακριτών σημείων για την απεικόνιση.

Σε κάθε γενιά του ΔΕΑ εκτελούνται αρχικά οι τελεστές της μετάλλαξης και της διασταύρωσης, κατόπιν εκτελείται η συνάρτηση αξιολόγησης και η συνάρτηση επιλογής για τη δημιουργία του ενδιάμεσου πληθυσμού και τέλος εκτελείται η συνάρτηση ελιτισμού, ώστε να επιβιώσουν και να περάσουν στην επόμενη γενιά οι καλύτερες υποψήφίες λύσεις. Ο τελεστής της μετάλλαξης υπολογίζεται από την Εξίσωση 22, ενώ οι υπόλοιποι τελεστές υπολογίζονται ακριβώς όπως τους αναλύσαμε στην Ενότητα 2.3. Ο αλγόριθμος αλλάζει μόνο τις τιμές των συντεταγμένων z και των

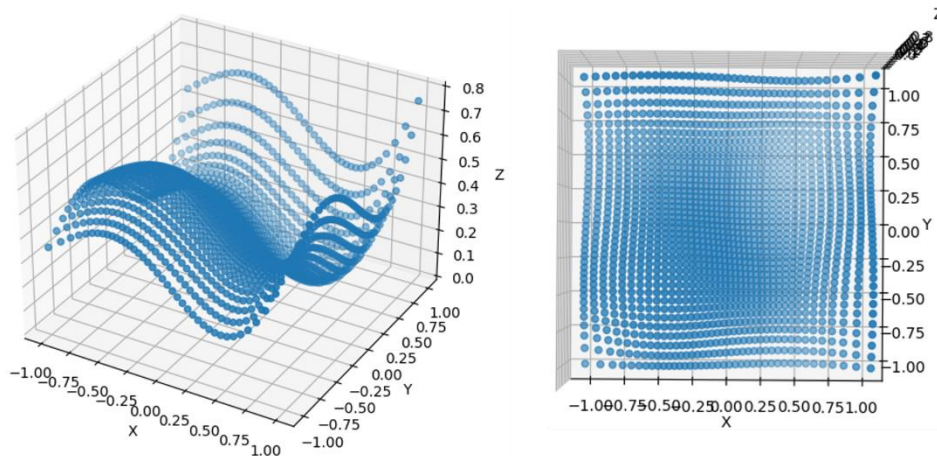
βαρών των σημείων ελέγχου, προκειμένου να ελαχιστοποιήσει το μέσο τετραγωνικό σφάλμα. Οι τιμές των συντεταγμένων z κυμαίνονται από -1 έως 1, ενώ τα βάρη από 1 έως 2. Η συνάρτηση αξιολόγησης σε κάθε γενιά διαβάζει τις τιμές των ανεξάρτητων μεταβλητών σχεδίασης, δηλαδή τις τιμές των συντεταγμένων z και των αντίστοιχων βαρών τους w , τα οποία είναι σύνολο πενήντα (εικοσιπέντε σημεία ελέγχου, τα οποία έχουν μία συντεταγμένη z και ένα βάρος w). Στη συνέχεια, με τη βοήθεια της βιβλιοθήκης *geomdl* [26], η οποία χρησιμοποιείται για την κατασκευή καμπυλών και επιφανειών, δημιουργεί μια επιφάνεια NURBS με βάση αυτές τις τιμές. Εφόσον έχει δημιουργηθεί η επιφάνεια, έχουμε πλέον όλα τα διακριτά σημεία της που την χαρακτηρίζουν. Κατόπιν, για κάθε ένα διακριτό σημείο της επιφάνειας NURBS, υπολογίζονται οι συντεταγμένες για το αντίστοιχο του με τη μέθοδο της γραμμικής παρεμβολής, με χρήση ειδικής συνάρτησης της βιβλιοθήκης *SciPy* [27]. Τέλος, υπολογίζεται το μέσο τετραγωνικό σφάλμα των διαφορών των αποστάσεων των συντεταγμένων z όλων των σημείων, το οποίο είναι η τιμή αξιολόγησης του τελικού πληθυσμού.

Για την ευκολότερη αλλαγή σημαντικών παραμέτρων που καθορίζουν το αποτέλεσμα του ΔΕΑ, δημιουργήθηκε το αρχείο 'parameters.txt', από το οποίο μπορούμε να ρυθμίσουμε τον αριθμό του πληθυσμού (*popsize*), τον μέγιστο αριθμό επαναλήψεων ή αλλιώς τον μέγιστο αριθμό των γενιών (*maxgens*), τον αριθμό των ανεξάρτητων μεταβλητών σχεδίασης (*nvars*), καθώς επίσης και της πιθανότητας μετάλλαξης (*Cr*) και της ρυθμιστικής παραμέτρου *F*. Αυτά διαβάζονται από το πρόγραμμα, το οποίο εκχωρεί τις τιμές αυτών στις κατάλληλες μεταβλητές. Επιπλέον, δημιουργήθηκαν τρία ακόμα αρχεία *text*. Το αρχείο 'initialize.txt', στο οποίο ορίζουμε τα όρια των μεταβλητών σχεδίασης, δηλαδή τα όρια μέσα στα οποία πρέπει να βρίσκονται οι τιμές των συντεταγμένων z και των αντίστοιχων βαρών του κάθε σημείου ελέγχου. Το αρχείο 'chromosome.txt', στο οποίο αποθηκεύονται όλες οι τελικές τιμές της κάθε μεταβλητής σχεδίασης και το αρχείο 'fitness.txt', στο οποίο αποθηκεύεται το μέσο τετραγωνικό σφάλμα της καλύτερης λύσης.

Για να μειώσουμε το χρόνο εκτέλεσης του αλγορίθμου, επιλέχθηκε ότι κάθε επιφάνεια που δημιουργείται θα περιγράφεται από είκοσι σημεία σε κάθε κατεύθυνση, δηλαδή σύνολο τετρακόσια σημεία (Εικόνες 19, 20). Με αυτό τον τρόπο, μειώνουμε την ανάλυση των επιφανειών, αλλά επίσης μειώνουμε και το χρόνο εκτέλεσης του αλγορίθμου.



Εικόνα 19. Επιφάνεια με ανάλυση 400 σημείων.



Εικόνα 20. Επιφάνεια με ανάλυση 1600 σημείων.

4.2 Ορισμός προβλημάτων

Τα προβλήματα προς επίλυση, δηλαδή οι επιφάνειες στόχοι που θα προσεγγίζουμε μέσω του ΔΕΑ με χρήση επιφανειών NURBS, είναι διδιάστατες συναρτήσεις επιφανειών. Αυτές οι συναρτήσεις είναι γνωστές μαθηματικά, περιγράφονται δηλαδή με μια συγκεκριμένη μαθηματική σχέση και ονομάζονται *test functions*, ή αλλιώς *benchmark functions*. Λόγω του ότι αυτές οι συναρτήσεις είναι γνωστές, τις καθιστούν πολύ καλά τυποποιημένα προβλήματα και ως εκ τούτου επιτρέπουν στους ερευνητές να αξιολογούν την απόδοση των αλγορίθμων βελτιστοποίησης συγκρίνοντας το αποτέλεσμα που τους δίνει ο αλγόριθμος, με το ολικό βέλτιστο της συνάρτησης. Επιπλέον, με τη χρήση αυτών των συναρτήσεων έχουμε τη δυνατότητα να συγκρίνουμε διάφορες μεθόδους και αλγορίθμους βελτιστοποίησης, αναλύοντας τα αποτελέσματά τους, όπως η ταχύτητα της σύγκλισης ή η ακρίβεια της λύσης σε διάφορους τύπους προβλημάτων.

Εμείς όμως, στην παρούσα εργασία δεν θα χρησιμοποιήσουμε αυτές τις συναρτήσεις για να βρούμε το ολικό ελάχιστο ή ολικό μέγιστο αυτών των συναρτήσεων, αλλά επιλέξαμε να τις χρησιμοποιήσουμε λόγω της ιδιαίτερης και δύσκολης μορφής που σχηματίζουν ως επιφάνειες, ώστε να αξιολογήσουμε την ποιότητα λύσης, με χρήση του ΔΕΑ και επιφανειών NURBS. Βέβαια, οι περισσότερες συναρτήσεις που θα αναλυθούν παρακάτω ορίζονται σε μεγαλύτερο διάστημα τιμών, όμως εμείς επικεντρωνόμαστε στο διάστημα $[-1, 1]$ για ευκολία στην τοποθέτηση των σημείων ελέγχου της επιφάνειας NURBS.

4.2.1 Συνάρτηση Giunta

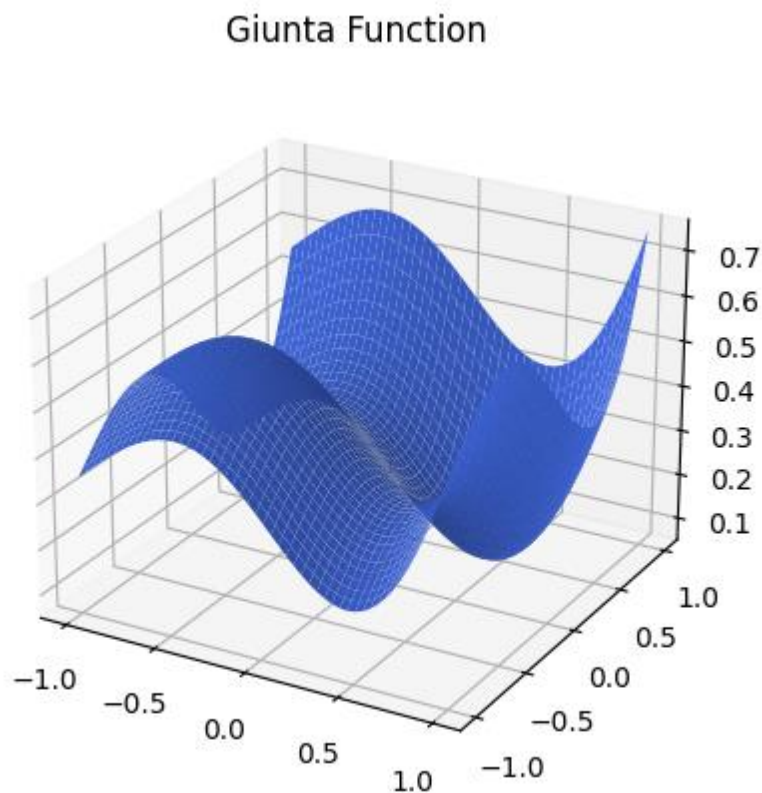
Η συνάρτηση Giunta χρησιμοποιείται με στόχο την εύρεση του ολικού ελάχιστου για την αξιολόγηση μεθόδων βελτιστοποίησης. Μαθηματικά, ορίζεται ως

$$f_{giunta} = 0.6 + \sum_{i=1}^n \left[\sin^2 \left(1 - \frac{16}{15} x_i \right) - \frac{1}{50} \sin \left(4 - \frac{64}{15} x_i \right) - \sin \left(1 - \frac{16}{15} x_i \right) \right] \quad (23)$$

στο διάστημα $-1 \leq x_i \leq 1$ με $i = 1, 2$. Ενώ αυτή περιγράφεται μέσω της Python, με τη βοήθεια της βιβλιοθήκης *numpy* [28], όπως φαίνεται παρακάτω.

```
def giunta(x, y):  
    term1 = 0.6 + np.sin(1-16/15*x)**2 - 1/50*np.sin(4-64/15*x)  
    - np.sin(1-16/15*x)  
    term2 = np.sin(1-16/15*y)**2 - 1/50*np.sin(4-64/15*y) -  
    np.sin(1-16/15*y)  
    f = term1 + term2  
    return f
```

Στην παρακάτω [Εικόνα 21](#) φαίνεται η συνάρτηση Giunta, όπως δημιουργείται με τη χρήση της Python και της βιβλιοθήκης *matplotlib* για την απεικόνιση της.



Εικόνα 21. Επιφάνεια στόχος που δημιουργήθηκε από τη συνάρτηση *Giunta* στο διάστημα $[-1, 1]$.

4.2.2 Συνάρτηση Keane

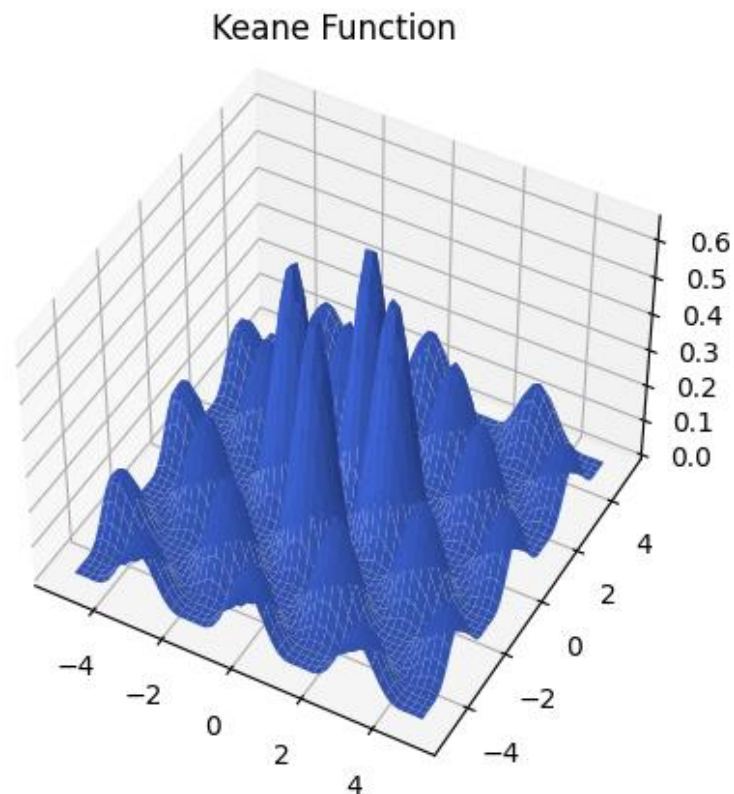
Η συνάρτηση Keane χρησιμοποιείται για εύρεση ολικού μέγιστου. Ορίζεται ως

$$f_{keane} = \frac{\sin^2(x_1 - x_2)\sin^2(x_1 + x_2)}{\sqrt{x_1^2 + x_2^2}} \quad (24)$$

στο διάστημα $0 \leq x_i \leq 10$ και $i = 1, 2$. Ο κώδικας στην Python είναι ο εξής

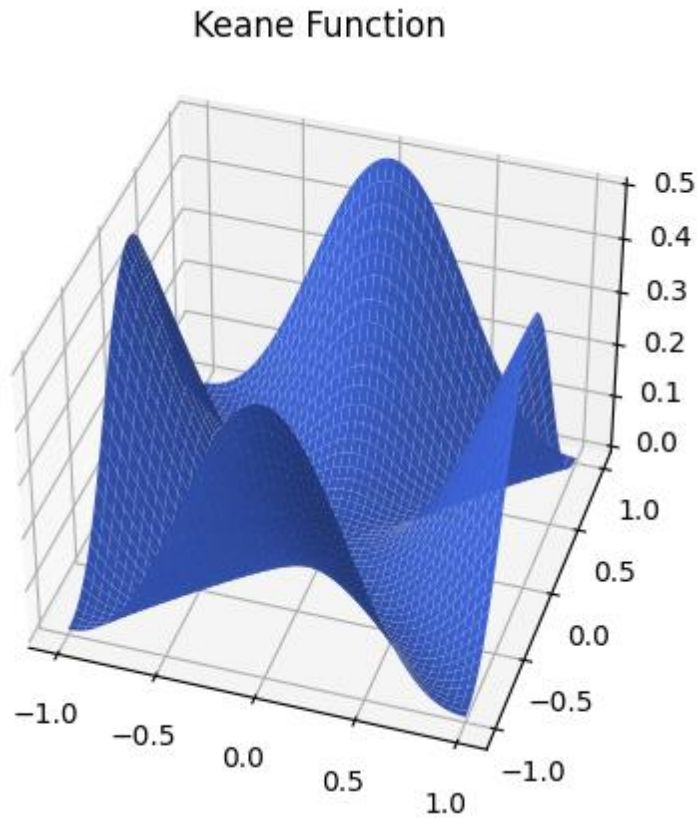
```
def keane(x, y):  
  
    term1 = np.sin(x-y)**2*np.sin(x+y)**2  
  
    term2 = np.sqrt(x**2+y**2)  
  
    f = term1 / term2  
  
    return f
```

Στην [Εικόνα 22](#) φαίνεται το σχήμα της επιφάνειας που δημιουργείται από τη συνάρτηση Keane στο διάστημα $[-4, 4]$. Επειδή, αυτή η επιφάνεια είναι πολύ



Εικόνα 22. Επιφάνεια στόχος που δημιουργήθηκε από τη συνάρτηση Keane στο διάστημα $[-4, 4]$.

περίπλοκη και τα σημεία ελέγχου που χρησιμοποιούμε είναι καθορισμένα να έχουν ένα σταθερό αριθμό, δεν μπορεί να προσεγγιστεί εύκολα, εκτός αν αυξήσουμε τα σημεία ελέγχου. Λόγω αυτού, επικεντρωνόμαστε στο διάστημα $[-1, 1]$, μετατρέποντας έτσι την επιφάνεια της [Εικόνας 22](#) στην επιφάνεια της [Εικόνας 23](#).



Εικόνα 23. Επιφάνεια στόχος που δημιουργήθηκε από την συνάρτηση Keane στο διάστημα $[-1, 1]$.

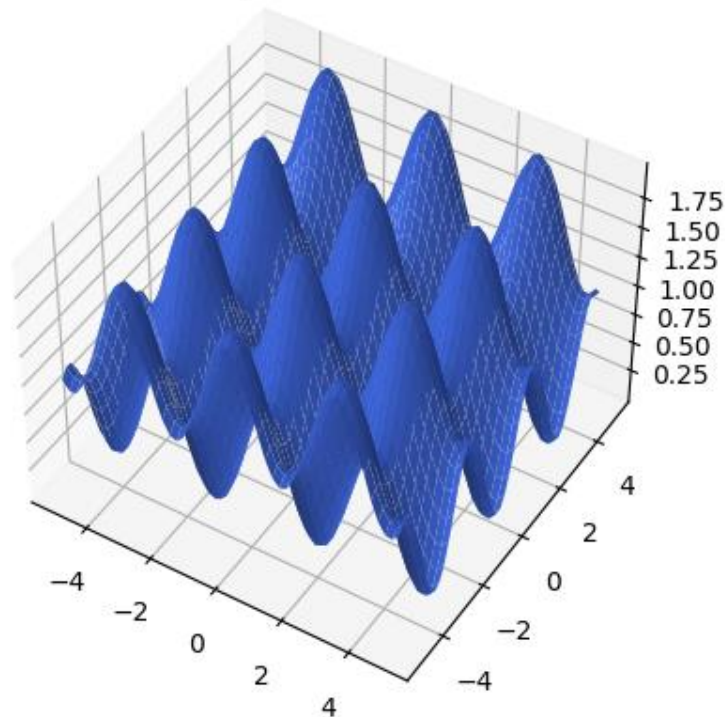
4.2.3 Συνάρτηση Parsopoulos

Η συνάρτηση Parsopoulos χρησιμοποιείται για εύρεση ολικού ελάχιστου, με την ιδιαιτερότητα ότι έχει δώδεκα ολικά ελάχιστα ([Εικόνα 24](#)). Ορίζεται ως

$$f_{\text{parsopoulos}} = \cos^2(x_1) + \sin^2(x_2) \quad (25)$$

στο διάστημα $-5 \leq x_i \leq 5$ και $i = 1, 2$.

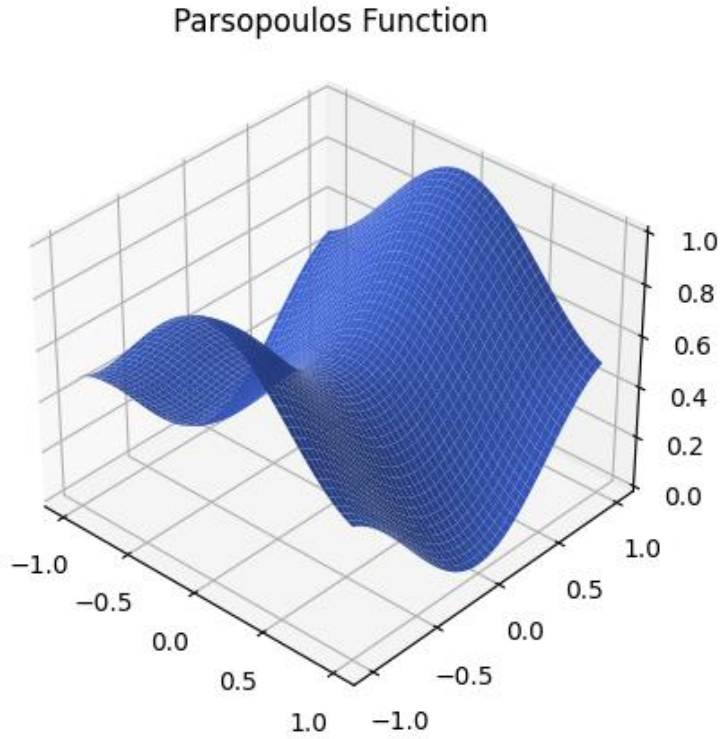
Parsopoulos Function



Εικόνα 24. Επιφάνεια στόχος που δημιουργήθηκε από την συνάρτηση Parsopoulos στο διάστημα $[-5, 5]$.

Στην [Εικόνα 24](#) παρατηρούμε ότι η συνάρτηση αυτή ξεπερνάει το διάστημα $[-1, 1]$ στον άξονα z , και έτσι την τροποποιούμε πολλαπλασιάζοντας τη επί 0.5, ώστε οι κορυφές να είναι εντός των διαστημάτων $[-1, 1]$. Επίσης, επειδή η επιφάνεια που σχηματίζεται στο διάστημα αυτό, στους άξονες x, y έχει απλή μορφή, πολλαπλασιάζουμε τα x_1, x_2 με μια ρυθμιστική παράμετρο που την ονομάσαμε *density*, ώστε να αυξήσουμε την καμπυλότητα στο διάστημα αυτό και να δημιουργήσουμε μια πιο δύσκολη μορφή στην επιφάνεια. Το αποτέλεσμα αυτής της τροποποίησης φαίνεται στην [Εικόνα 25](#).

```
def parsopoulos(x, y):  
    scale = 0.5  
  
    density = 1.4  
  
    f = scale * (np.cos(density*x)**2 + np.sin(density*y)**2)  
  
    return f
```

Εικόνα 25. Επιφάνεια στόχος που δημιουργήθηκε από τροποποίηση της συνάρτησης Parsopoulos στο διάστημα $[-1, 1]$.

4.2.3 Συνάρτηση Peaks

Η συνάρτηση Peaks (Εικόνα 26) χρησιμοποιείται για την εύρεση ολικού ελάχιστου και ορίζεται ως

$$f_{peaks}(x) = f_1(x) - f_2(x) - f_3(x) \quad (26)$$

Όπου

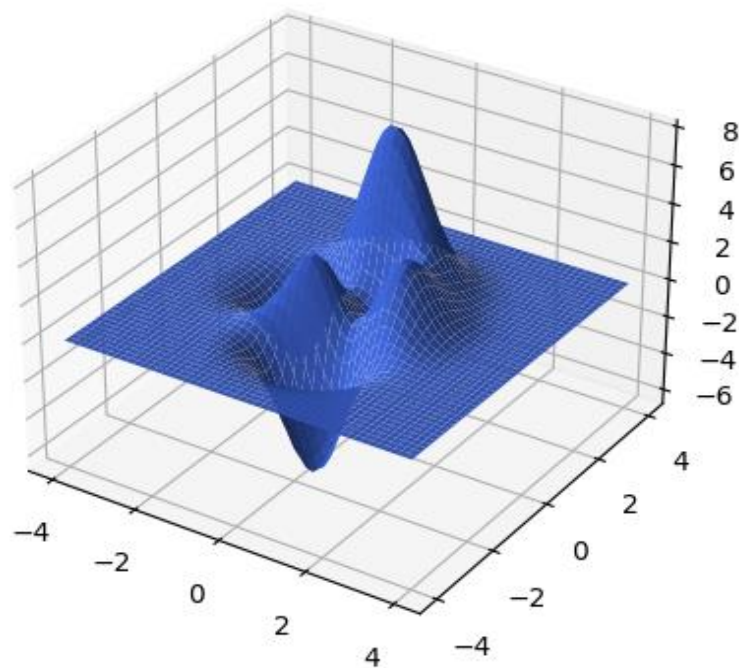
$$f_1(x) = 3(1 - x_1)e^{[-x_1^2 - (x_2+1)^2]} \quad (27)$$

$$f_2(x) = 10\left(\frac{x_1}{5} - x_1^3 - x_2^5\right)e^{(-x_1^2 - x_2^2)} \quad (28)$$

$$f_3(x) = \frac{1}{3}e^{[-(x_1+1)^2 - x_2^2]} \quad (29)$$

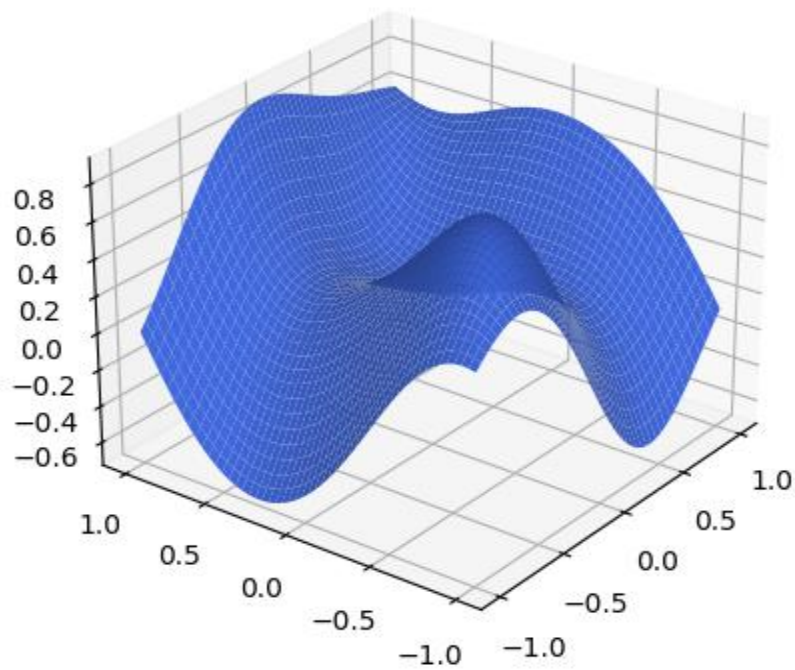
στο διάστημα $-4 \leq x_i \leq 4$ και $i = 1, 2$. Και σε αυτή την περίπτωση, επειδή οι κορυφές της συνάρτησης ξεπερνούν το διάστημα $[-1, 1]$, πολλαπλασιάζουμε με 0.25 για να βρίσκονται εντός των ορίων (Εικόνα 27).

Peaks Function



Εικόνα 26. Επιφάνεια στόχος που δημιουργήθηκε από την συνάρτηση Peaks στο διάστημα $[-4, 4]$.

Peaks Function



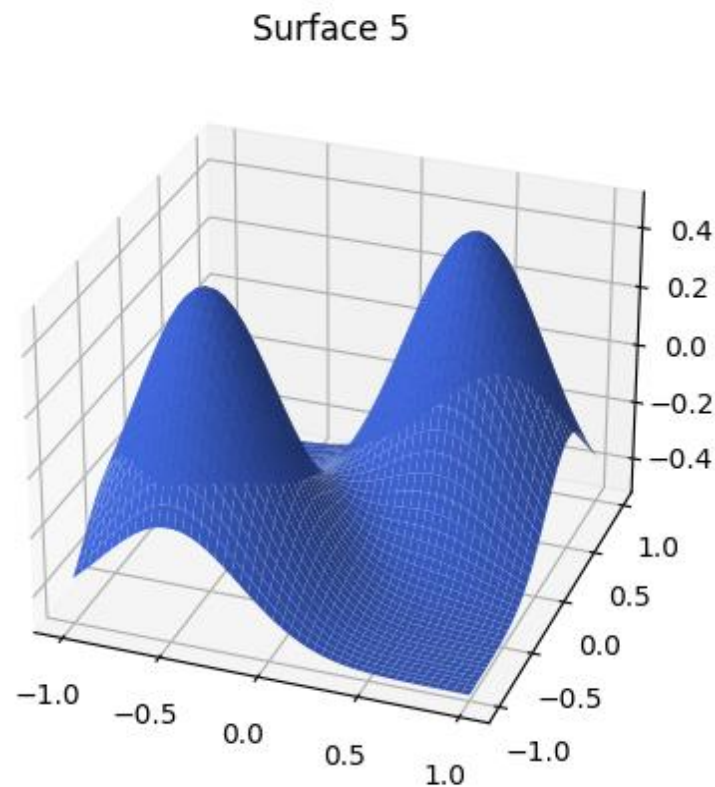
Εικόνα 27. Επιφάνεια στόχος που δημιουργήθηκε από τροποποίηση της συνάρτησης Peaks στο διάστημα $[-1, 1]$.


```
def peaks(x,y):
    scale = 0.25
    f = scale * (3*(1-x)**2 * np.exp(-(x**2) - (y+1)**2) \
        - 10*(x/5 - x**3 - y**5) * np.exp(-x**2-y**2) \
        - np.exp(-(x+1)**2 - y**2))
    return f
```

4.2.4 Τυχαία Συνάρτηση Επιφάνειας

Η τελευταία συνάρτηση που επιλέχθηκε δεν υπάγεται στις γνωστές συναρτήσεις για δοκιμή αλγορίθμων βελτιστοποίησης, αλλά επιλέχθηκε λόγω του ότι σχηματίζει μια ιδιαίτερη μορφή με δύο κορυφές (Εικόνα 28). Ορίζεται ως

$$f_{surface} = -0.5 + e^{-0.4[(x_1-0.5)^2+(x_2-0.5)^2]} + e^{-0.4[(x_1+0.5)^2+(x_2+0.5)^2]} \quad (30)$$

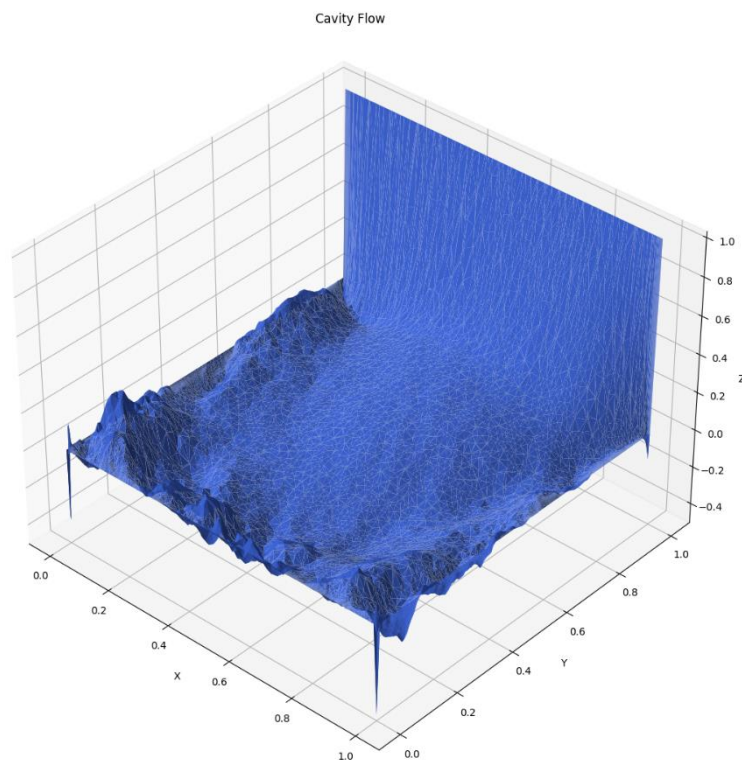


Εικόνα 28. Επιφάνεια στόχος που δημιουργήθηκε από μια τυχαία συνάρτηση στο διάστημα $[-1, 1]$.

```
def surface_5(x, y):
    f = -0.5 + np.exp(-4 * ((x - 0.5)**2 + (y - 0.5)**2)) +
        np.exp(-4 * ((x + 0.5)**2 + (y + 0.5)**2))
    return f
```

4.2.5 Επιφάνεια πραγματικού πεδίου ροής

Ως τελευταία επιφάνεια, επιλέχθηκε ένα πραγματικό πεδίο ροής (cavity flow) (Εικόνα 29), όπου τώρα οι κατευθύνσεις x, y, z είναι οι συντεταγμένες X, Y και η U_x συνιστώσα της ταχύτητας του ρευστού αντίστοιχα. Λόγω του ότι η μορφή της καμπύλης είναι πιο δύσκολη, οι παράμετροι που επιλέχθηκαν σε αυτή τη δοκιμή είναι οι εξής: αριθμός πληθυσμού 100, αριθμός γενιών 1000, αριθμός σημείων ελέγχου 256, ρυθμιστική παράμετρος $F=0.7$, πιθανότητα μετάλλαξης $Cr=0.4$, όρια άξονα Z -0.3 έως 1.0, όρια βαρών 0.5 έως 4, αριθμός σημείων που περιγράφουν την καμπύλη NURBS 19881.



Εικόνα 29. Επιφάνεια από πραγματικά δεδομένα πεδίου ροής (cavity flow).

Κεφάλαιο 5 – Αποτελέσματα

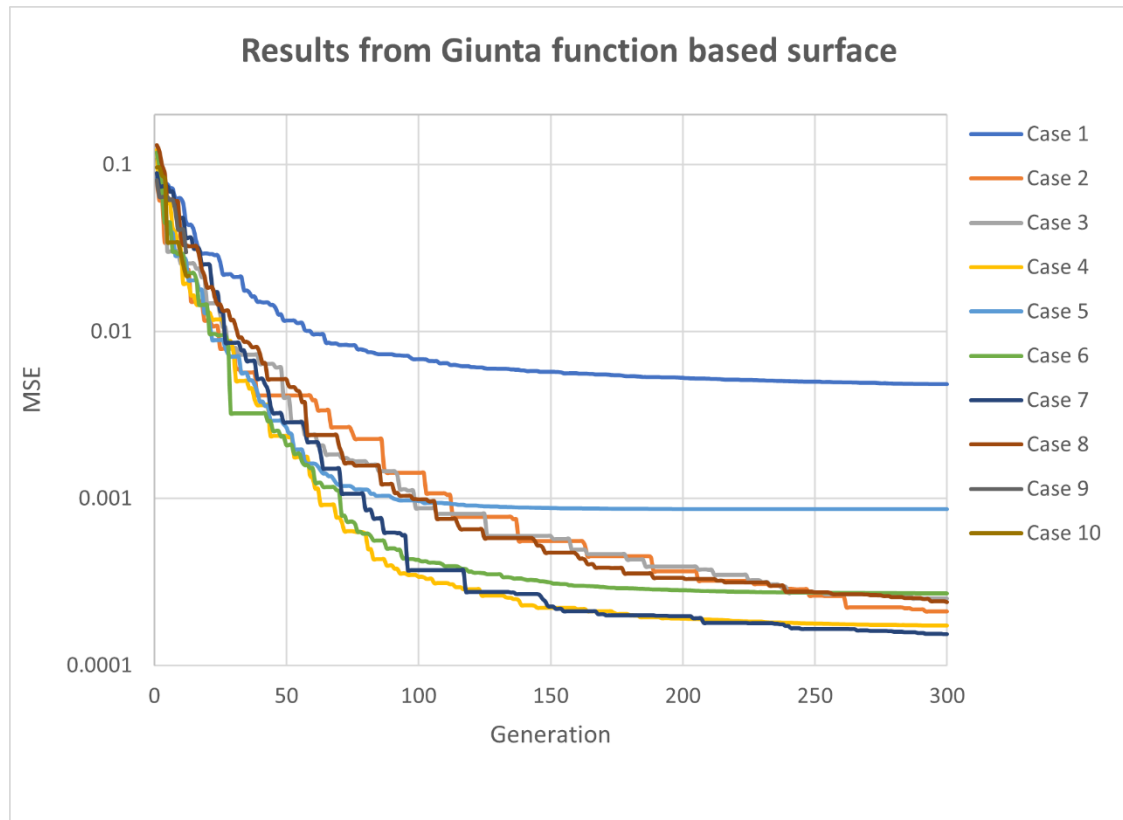
Στο κεφάλαιο αυτό παρουσιάζουμε τα αποτελέσματα των δοκιμών. Σε κάθε επιφάνεια που ορίσαμε παραπάνω, κάναμε δέκα δοκιμές. Οι παράμετροι που μπορούμε να αλλάξουμε είναι ο αριθμός των γενιών, το πλήθος του πληθυσμού, η ρυθμιστική παράμετρος F , καθώς και η πιθανότητα διασταύρωσης. Αυτά ορίζονται από το εξωτερικό αρχείο 'parameters.txt' που αναφέραμε και στο προηγούμενο κεφάλαιο. Ο αριθμός των γενιών ορίστηκε να είναι σταθερός σε όλες τις δοκιμές, διότι παρατηρήθηκε ότι ένας μεγάλος αριθμός γενιών δεν μειώνει σημαντικά το μέσο τετραγωνικό σφάλμα, αλλά αυξάνει αρκετά το χρόνο εκτέλεσης του αλγορίθμου. Ο αριθμός αυτός ορίστηκε ίσος με 300, ο οποίος αποδείχθηκε επαρκής. Οπότε, στις δοκιμές αλλάζαμε μόνο τις τρεις παραμέτρους, αρχίζοντας από ένα μικρό πλήθος και αυξάνοντας το μέχρι να παρατηρήσουμε ότι η προηγούμενη δοκιμή (δηλαδή με μικρότερο πλήθος) είχε μικρότερο μέσο τετραγωνικό σφάλμα, ενώ οι παράμετροι F και Cr παρέμεναν σταθεροί με τις αρχικές τιμές να είναι 0.7 και 0.4 αντίστοιχα. Κατόπιν, σχηματίσαμε διάφορους συνδυασμούς, αλλάζοντας τις παραμέτρους, για να παρατηρήσουμε πώς συμπεριφέρεται ο αλγόριθμος σε αυτές τις περιπτώσεις, καταγράφοντας τα αποτελέσματα.

Η δομή με την οποία παρουσιάζονται τα αποτελέσματα είναι σε μορφή πινάκων και γραφημάτων, για κάθε επιφάνεια και για κάθε δοκιμή. Εκτός από αυτά, δίνονται για κάθε επιφάνεια που προσεγγίστηκε, τα σχήματα της δοκιμής, η οποία προσέγγισε περισσότερο την επιφάνεια, δηλαδή αυτής που είχε το μικρότερο μέσο τετραγωνικό σφάλμα.

5.1 Αποτελέσματα επιφάνειας Giunta

Παρακάτω (Εικόνα 30) παρουσιάζονται τα αποτελέσματα της επιφάνειας που δημιουργήθηκε με βάση τη συνάρτηση Giunta. Παρατηρείται ότι η καλύτερη λύση είναι στη δοκιμή 7 του Πίνακα 1, όπου το πλήθος, η παράμετρος F , η παράμετρος Cr

και το μέσο τετραγωνικό σφάλμα είναι 20, 0.6, 0.3 και 0.00015384 αντίστοιχα. Στις [Εικόνες 31](#) και [32](#) παρουσιάζεται η προσέγγιση της επιφάνειας NURBS στην επιφάνεια στόχο σε δύο διαφορετικές γενιές.

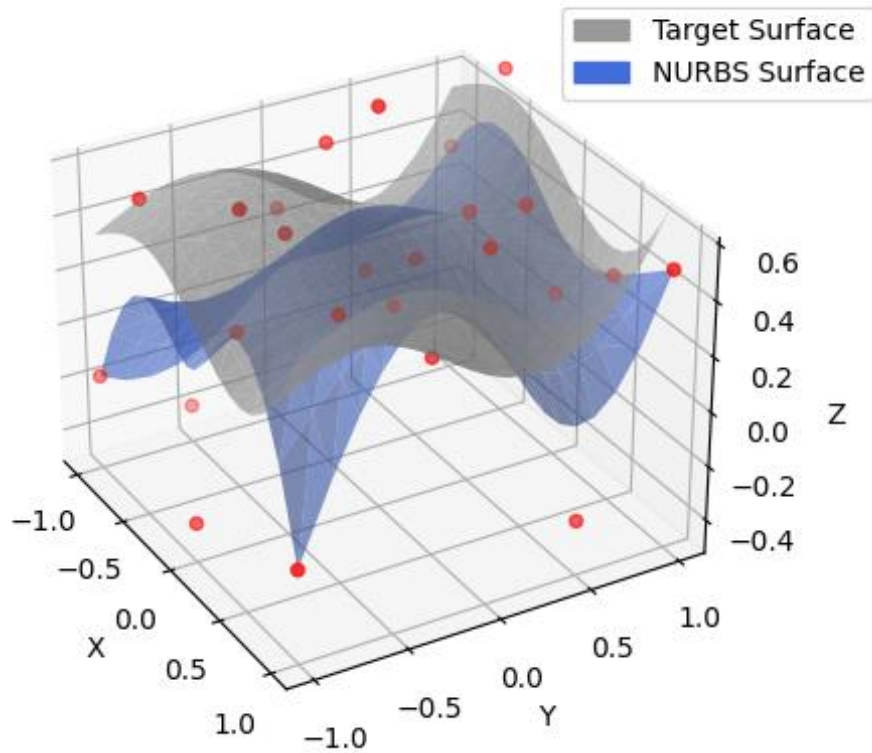


Εικόνα 30. Αποτελέσματα μέσου τετραγωνικού σφάλματος ανά γενιά της επιφάνειας που δημιουργήθηκε με βάση τη συνάρτηση Giunta.

Test Case	Maxgens	Popsiz	F	Cr	MSE	Time [min]
1	300	10	0.7	0.4	0.00482584	1.04
2	300	20	0.7	0.4	0.00020994	2.09
3	300	30	0.7	0.4	0.00025148	2.51
4	300	20	0.6	0.4	0.00017391	2.11
5	300	20	0.5	0.4	0.00086020	1.52
6	300	20	0.6	0.5	0.00026950	2.07
7	300	20	0.6	0.3	0.00015384	1.52
8	300	20	0.6	0.2	0.00023948	1.52
9	300	20	0.7	0.3	0.00033679	2.09
10	300	50	0.6	0.3	0.00016787	5.47

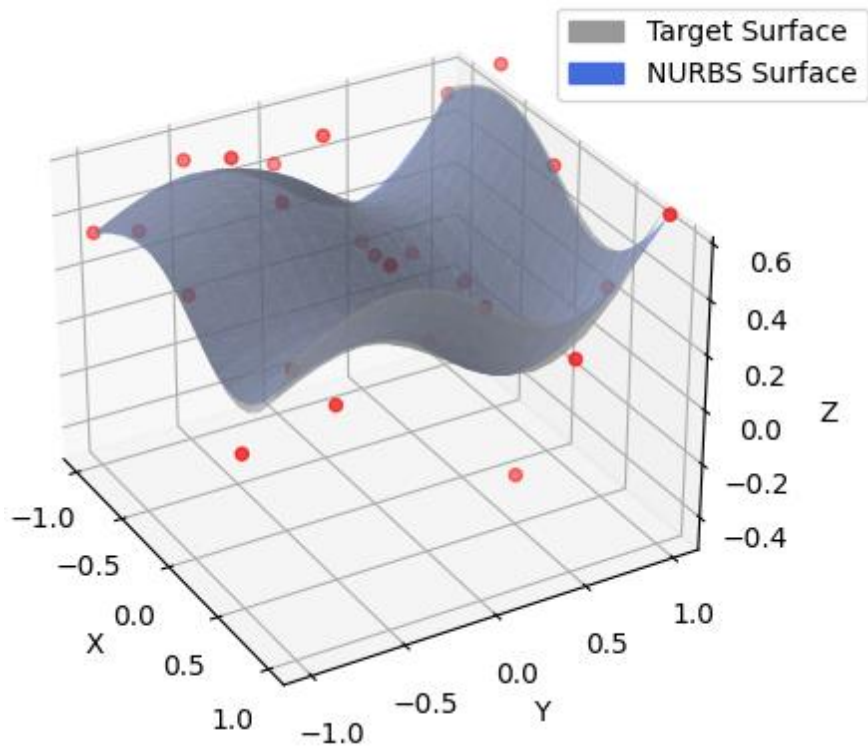
Πίνακας 1. Αποτελέσματα δοκιμών επιφάνειας Giunta.

Generation: 20



Εικόνα 31. Προσέγγιση επιφάνειας Giunta με χρήση NURBS και ΔΕΑ στη γενιά 20.

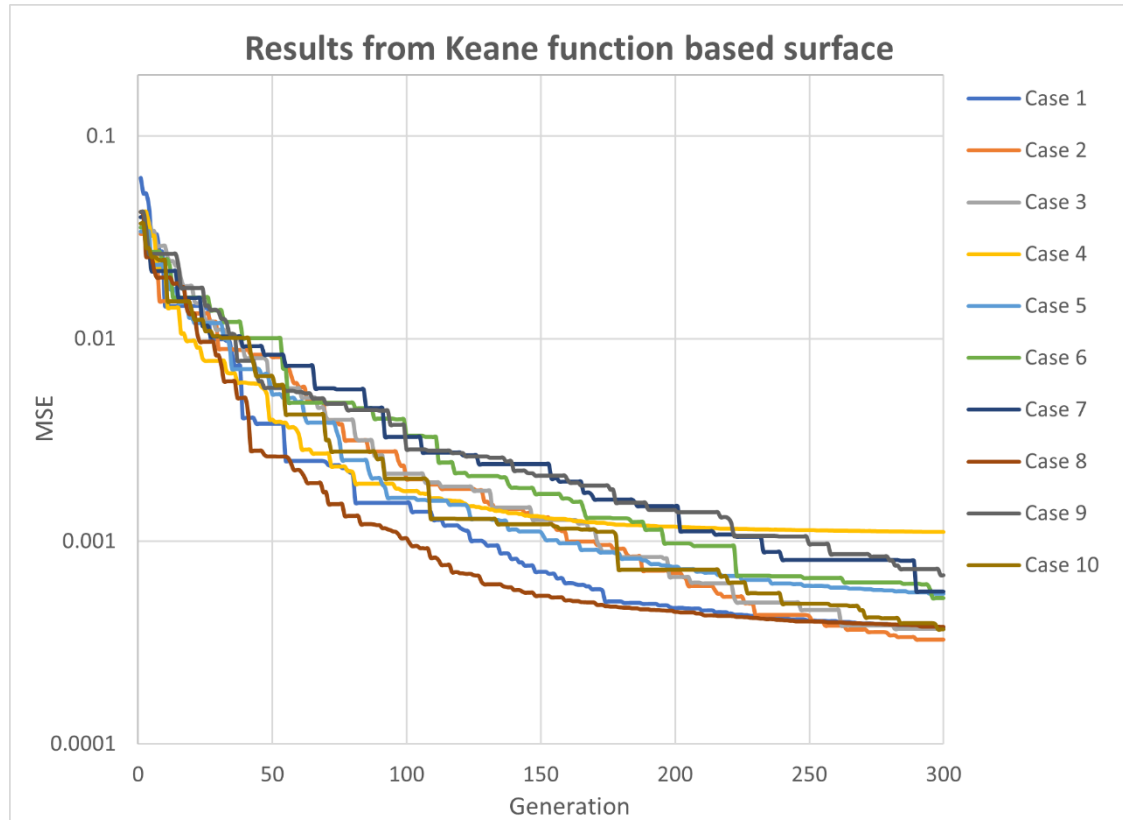
Generation: 300



Εικόνα 32. Προσέγγιση επιφάνεια Giunta με χρήση NURBS και ΔΕΑ στη γενιά 300.

5.2 Αποτελέσματα επιφάνειας Keane

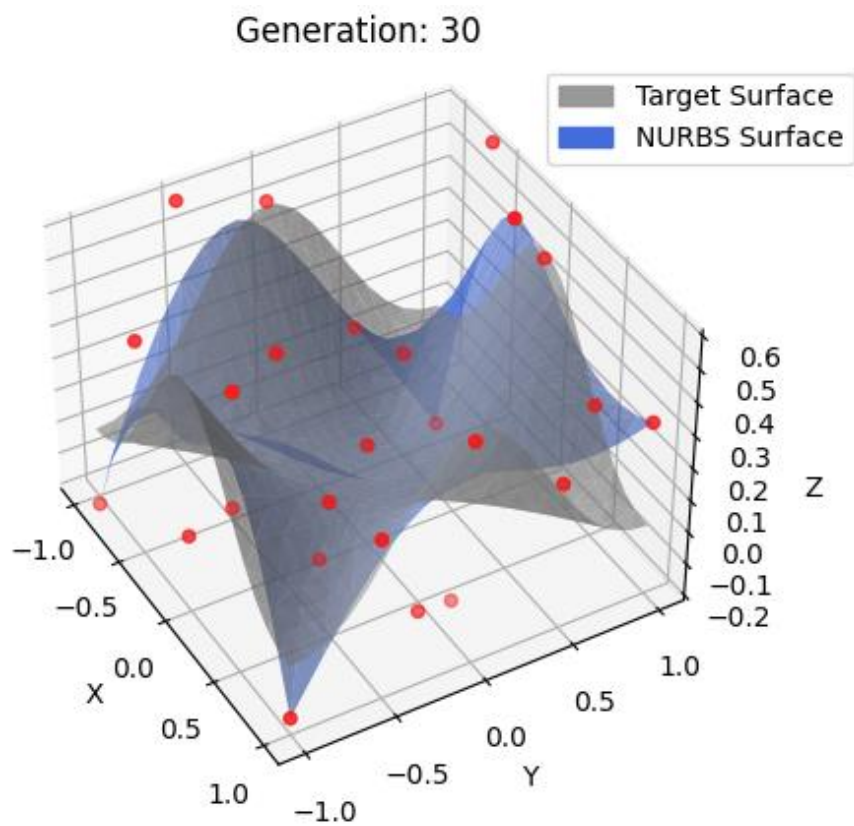
Παρακάτω (Εικόνα 33) παρουσιάζονται τα αποτελέσματα σύγκλισης του ΔΕΑ για την επιφάνεια που δημιουργήθηκε με βάση τη συνάρτηση Keane. Εδώ παρατηρείται ότι η καλύτερη λύση είναι στη δοκιμή 2 του Πίνακα 2, όπου το μέσο τετραγωνικό σφάλμα είναι 0.00032687. Τα αποτελέσματα παρουσιάζονται στις Εικόνες 34 και 35.



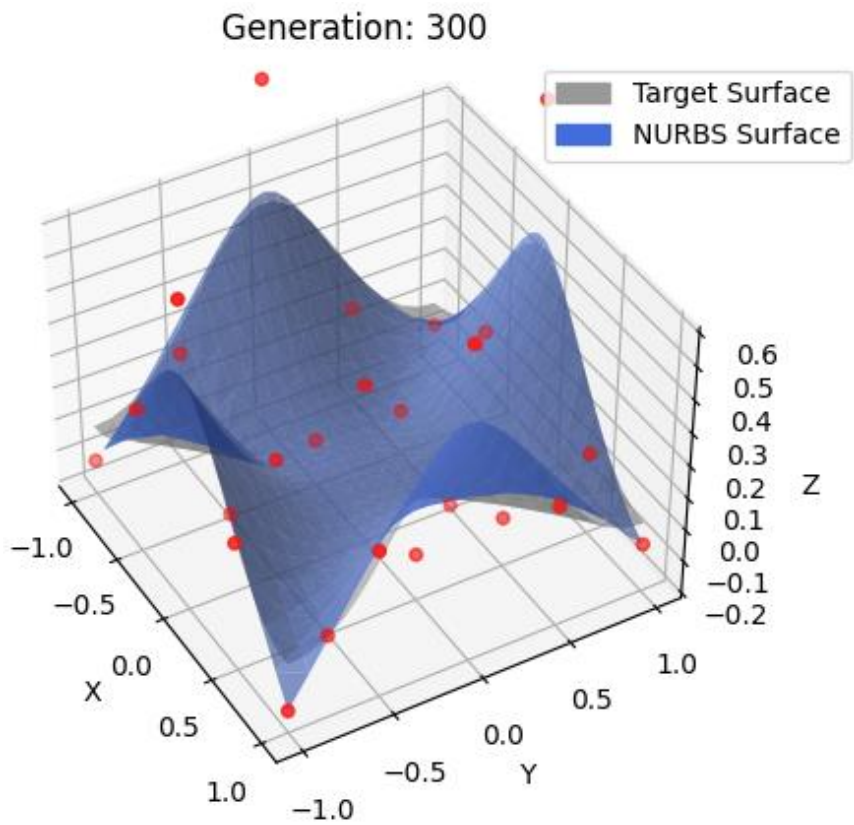
Εικόνα 33. Αποτελέσματα μέσου τετραγωνικού σφάλματος ανά γενιά της επιφάνειας που δημιουργήθηκε με βάση τη συνάρτηση Keane.

Test Case	Maxgens	Popsiz	F	Cr	MSE	Time [min]
1	300	10	0.7	0.4	0.00037371	1.03
2	300	20	0.7	0.4	0.00032687	2.14
3	300	30	0.7	0.4	0.00036877	3.04
4	300	20	0.6	0.4	0.00111131	1.55
5	300	20	0.7	0.5	0.00054980	1.53
6	300	20	0.8	0.5	0.00052633	2.03
7	300	30	0.8	0.5	0.00056414	3.04
8	300	20	0.7	0.6	0.00037831	2.08
9	300	20	0.7	0.3	0.00067837	1.51
10	300	50	0.7	0.4	0.00036842	5.25

Πίνακας 2. Αποτελέσματα δοκιμών επιφάνειας Keane.



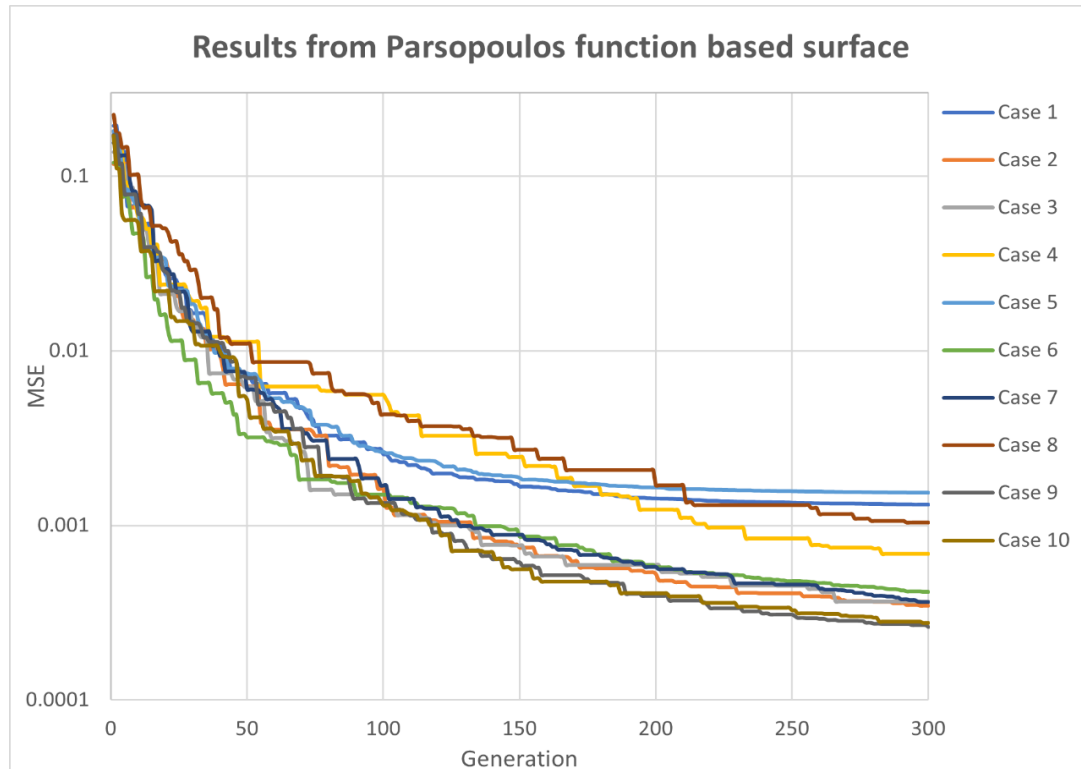
Εικόνα 34. Προσέγγιση επιφάνειας Keane με χρήση NURBS και ΔΕΑ στη γενιά 30.



Εικόνα 35. Προσέγγιση επιφάνειας Keane με χρήση NURBS και ΔΕΑ στη γενιά 300.

5.3 Αποτελέσματα επιφάνειας Parsopoulos

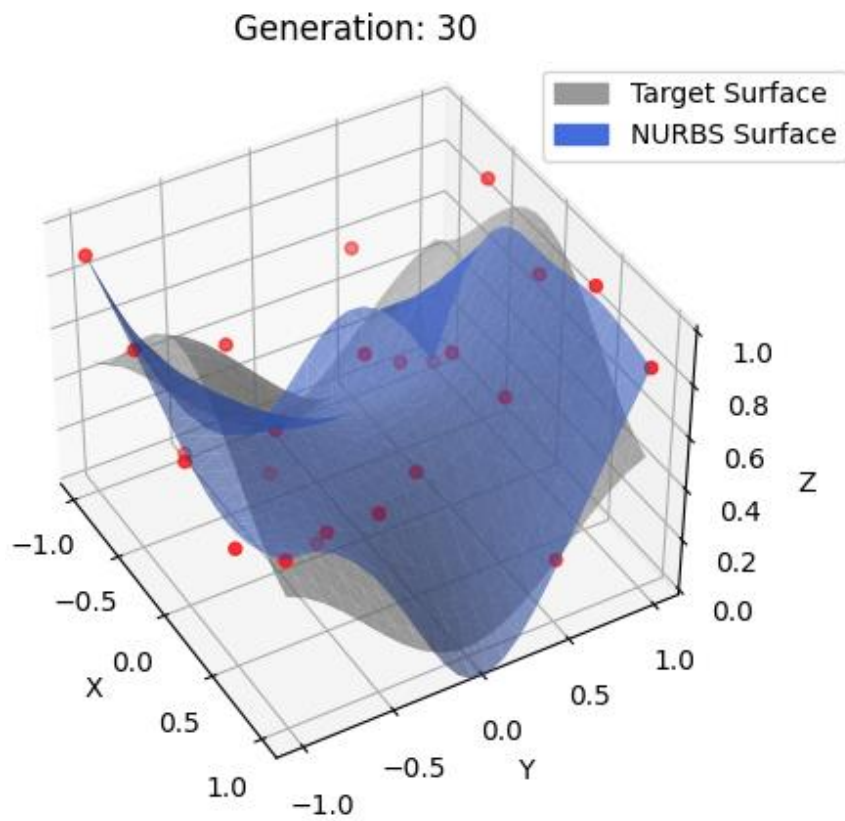
Στην [Εικόνα 36](#) παρουσιάζονται οι συγκλίσεις του ΔΕΑ για την επιφάνεια που δημιουργήθηκε με βάση τη συνάρτηση Parsopoulos. Η καλύτερη λύση παρατηρείται στη δοκιμή 9 του [Πίνακα 3](#), όπου το μέσο τετραγωνικό σφάλμα είναι 0.00026127. Οι συγκρίσεις των επιφανειών παρουσιάζονται στις [Εικόνες 37](#) και [38](#).



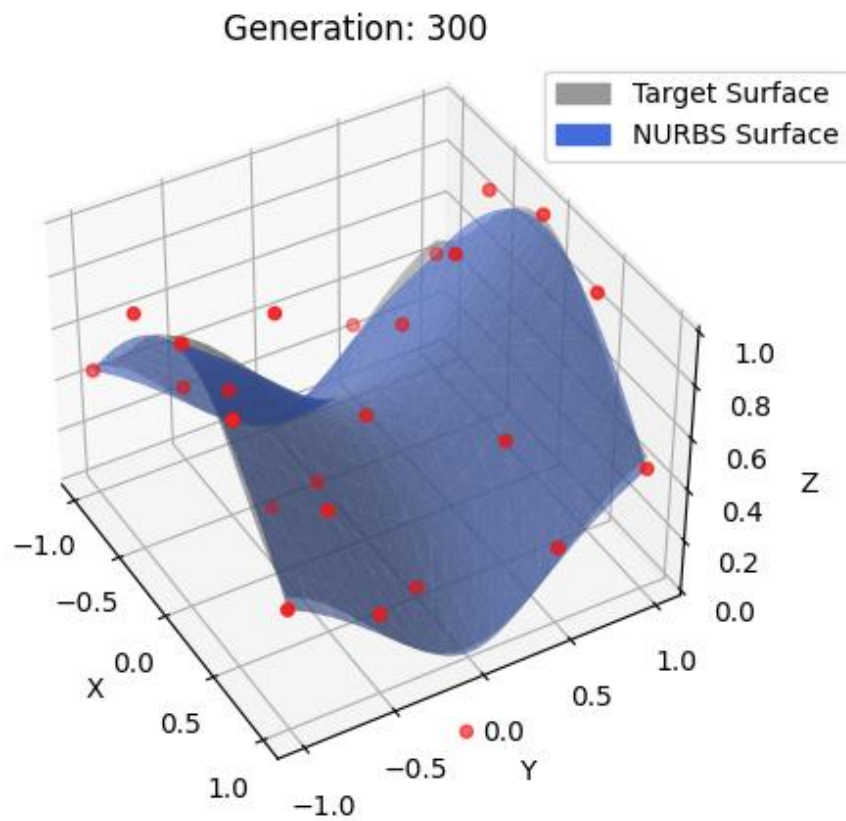
Εικόνα 36. Αποτελέσματα μέσου τετραγωνικού σφάλματος ανά γενιά της επιφάνειας που δημιουργήθηκε με βάση τη συνάρτηση Parsopoulos.

Test Case	Maxgens	Popsiz	F	Cr	MSE	Time [min]
1	300	10	0.7	0.4	0.00131264	0.58
2	300	20	0.7	0.4	0.00034706	2.07
3	300	30	0.7	0.4	0.00036580	3.01
4	300	20	0.8	0.4	0.00068837	2.03
5	300	20	0.6	0.4	0.00154116	2.09
6	300	20	0.7	0.5	0.00041744	2.1
7	300	20	0.7	0.3	0.00036330	1.56
8	300	20	0.8	0.3	0.00103322	2.11
9	300	40	0.7	0.5	0.00026127	4.22
10	300	100	0.7	0.5	0.00027655	10.53

Πίνακας 3. Αποτελέσματα δοκιμών επιφάνειας Parsopoulos.



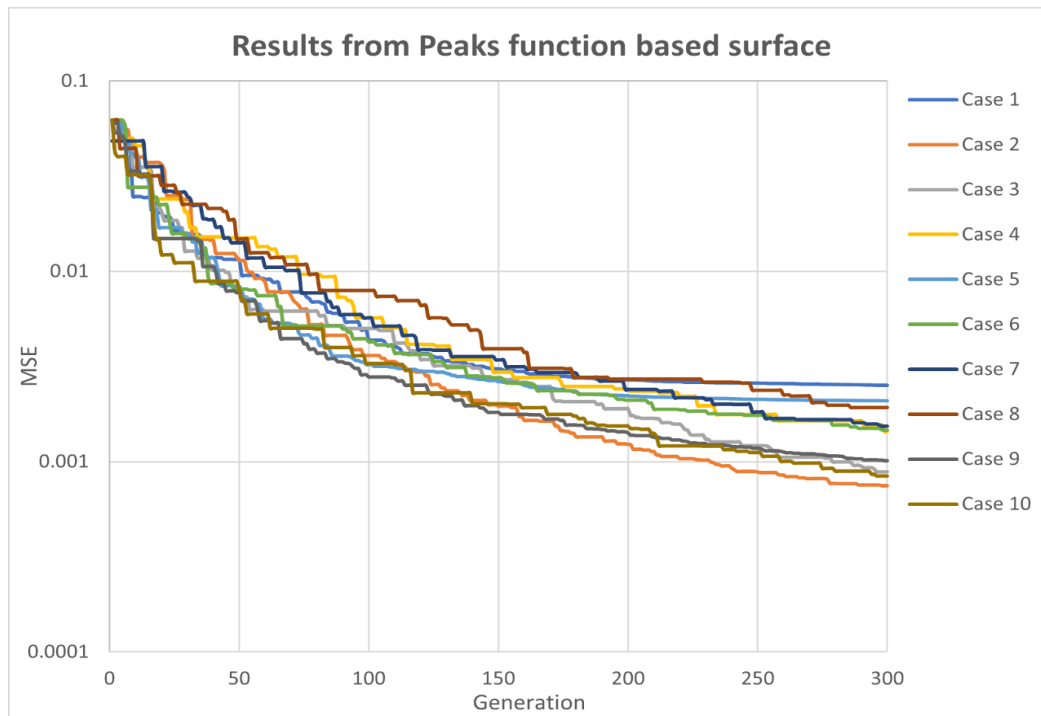
Εικόνα 37. Προσέγγιση επιφάνειας Parsopoulos με χρήση NURBS και ΔΕΑ στη γενιά 30.



Εικόνα 38. Προσέγγιση επιφάνειας Parsopoulos με χρήση NURBS και ΔΕΑ στη γενιά 300.

5.4 Αποτελέσματα επιφάνειας Peaks

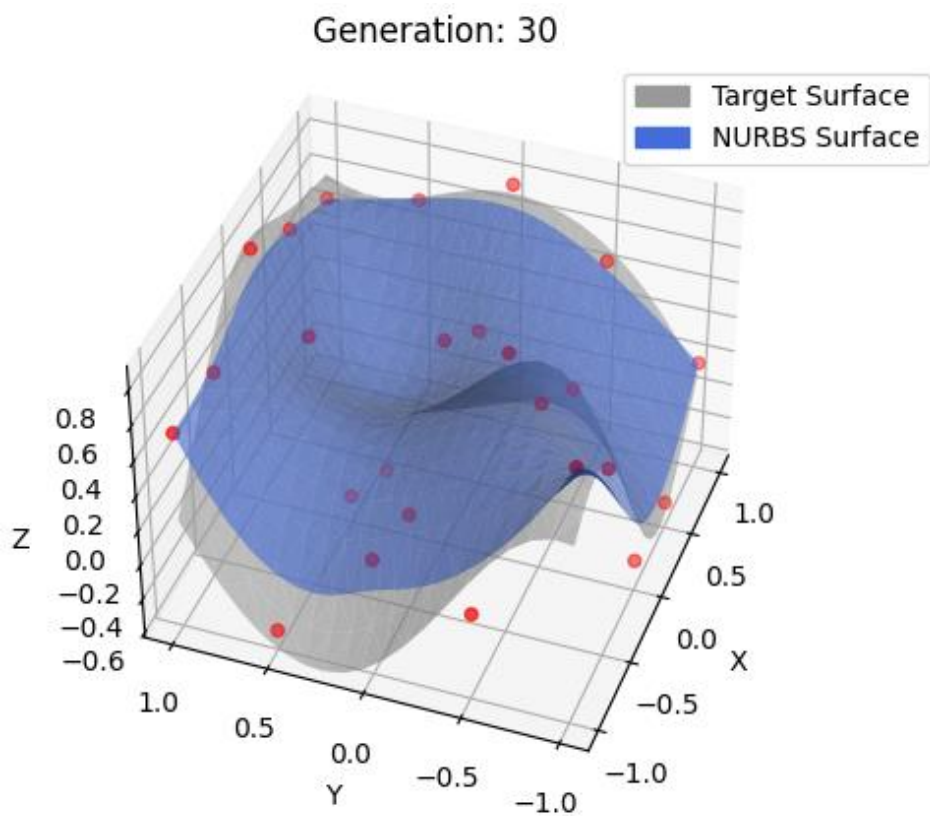
Στην [Εικόνα 39](#) παρουσιάζονται τα αποτελέσματα της σύγκλισης του ΔΕΑ για την επιφάνεια που δημιουργήθηκε με βάση τη συνάρτηση Peaks. Η καλύτερη λύση παρατηρείται στη δοκιμή 2 του [Πίνακα 4](#), όπου το μέσο τετραγωνικό σφάλμα είναι 0.00074924. Οι επιφάνειες σε δύο διαφορετικές γενιές παρουσιάζονται στις [Εικόνες 40](#) και [41](#).



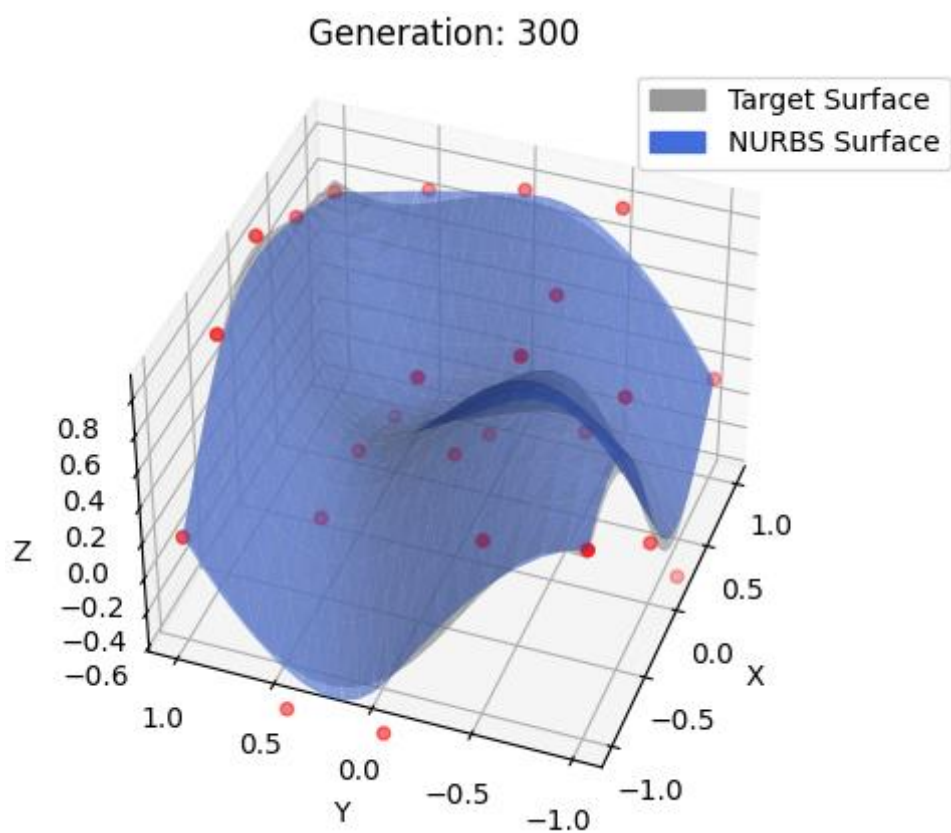
Εικόνα 39. Αποτελέσματα μέσου τετραγωνικού σφάλματος ανά γενιά της επιφάνειας που δημιουργήθηκε με βάση τη συνάρτηση Peaks.

Test Case	Maxgens	Popsiz	F	Cr	MSE	Time [min]
1	300	10	0.7	0.4	0.00252087	1.03
2	300	20	0.7	0.4	0.00074924	2.11
3	300	30	0.7	0.4	0.00088244	3.11
4	300	20	0.8	0.4	0.00144129	2.12
5	300	20	0.6	0.4	0.00209017	2.05
6	300	20	0.7	0.3	0.00146255	2.15
7	300	20	0.7	0.2	0.00153458	2.02
8	300	20	0.8	0.3	0.00192437	2.09
9	300	20	0.7	0.5	0.00101190	2.05
10	300	50	0.7	0.4	0.00083832	5.27

Πίνακας 4. Αποτελέσματα δοκιμών επιφάνειας Peaks.



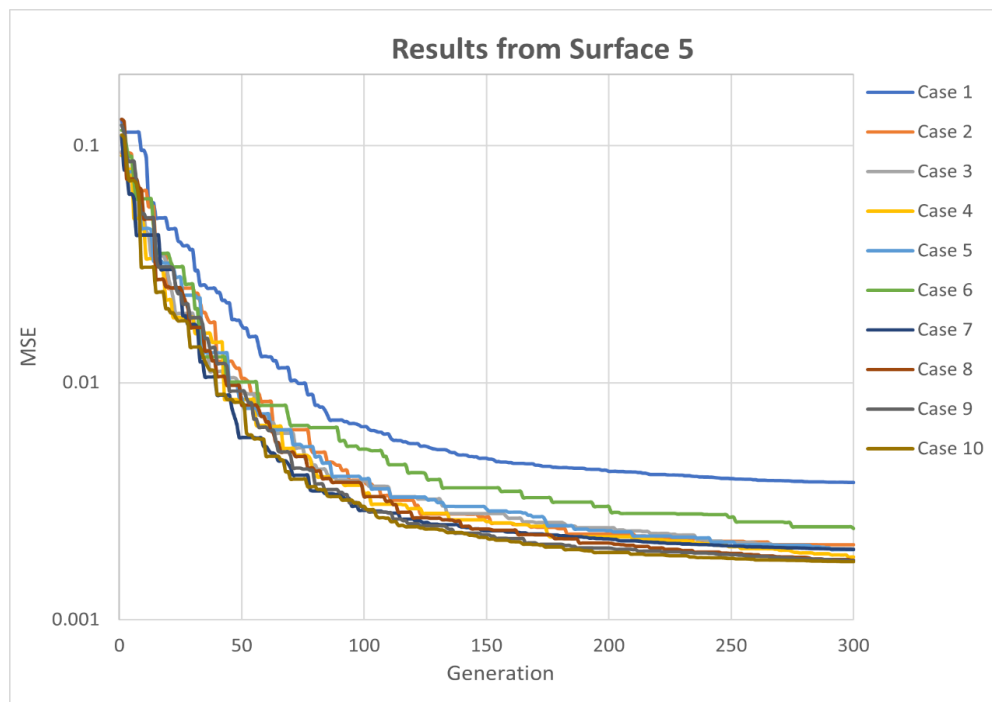
Εικόνα 40. Προσέγγιση επιφάνειας Peaks με χρήση NURBS και ΔΕΑ στη γενιά 30.



Εικόνα 41. Προσέγγιση επιφάνειας Peaks με χρήση NURBS και ΔΕΑ στη γενιά 300.

5.5 Αποτελέσματα τυχαίας επιφάνειας

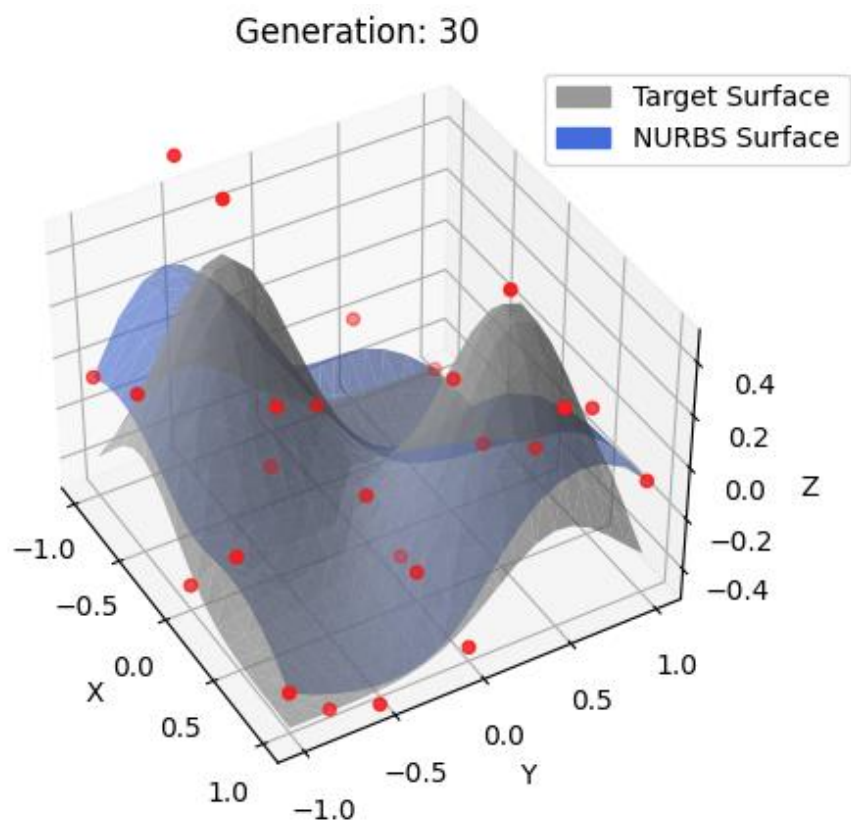
Στην [Εικόνα 42](#) παρουσιάζονται αντίστοιχα τα αποτελέσματα της σύγκλισης του ΔΕΑ για την επιφάνεια που δημιουργήθηκε με βάση την τυχαία συνάρτηση που δημιουργήθηκε. Η καλύτερη λύση παρατηρείται στη δοκιμή 10 του [Πίνακα 5](#), όπου το μέσο τετραγωνικό σφάλμα είναι 0.00175708. Στις [Εικόνες 43](#) και [44](#) παρουσιάζονται οι αντίστοιχες επιφάνειες για διαφορετικό αριθμό γενιών.



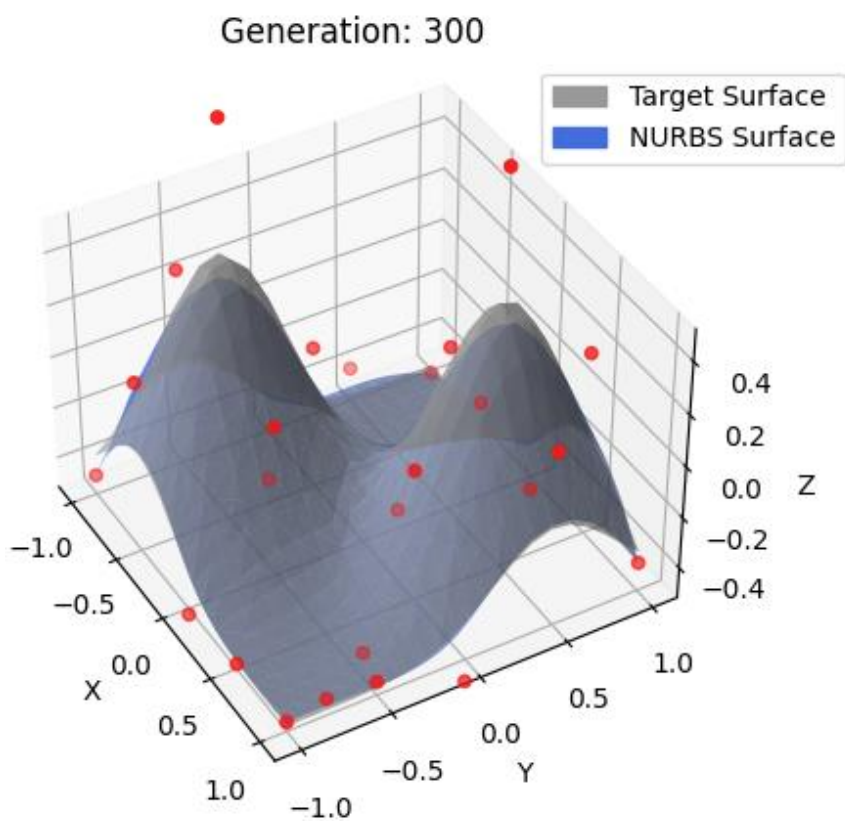
Εικόνα 42. Αποτελέσματα μέσου τετραγωνικού σφάλματος ανά γενιά της επιφάνειας που δημιουργήθηκε με βάση την τυχαία συνάρτηση που επιλέχθηκε.

Test Case	Maxgens	Popsiz	F	Cr	MSE	Time [min]
1	300	10	0.7	0.4	0.00379294	1.03
2	300	20	0.7	0.4	0.00207107	2.11
3	300	30	0.7	0.4	0.00198975	3.13
4	300	40	0.7	0.4	0.00184229	4.04
5	300	50	0.7	0.4	0.00197095	5.27
6	300	40	0.8	0.4	0.00242372	4.23
7	300	40	0.6	0.4	0.00197937	4.08
8	300	40	0.7	0.5	0.00178426	3.44
9	300	40	0.7	0.6	0.00177706	4.19
10	300	40	0.7	0.7	0.00175708	4.15

Πίνακας 5. Αποτελέσματα δοκιμών τυχαίας επιφάνειας.



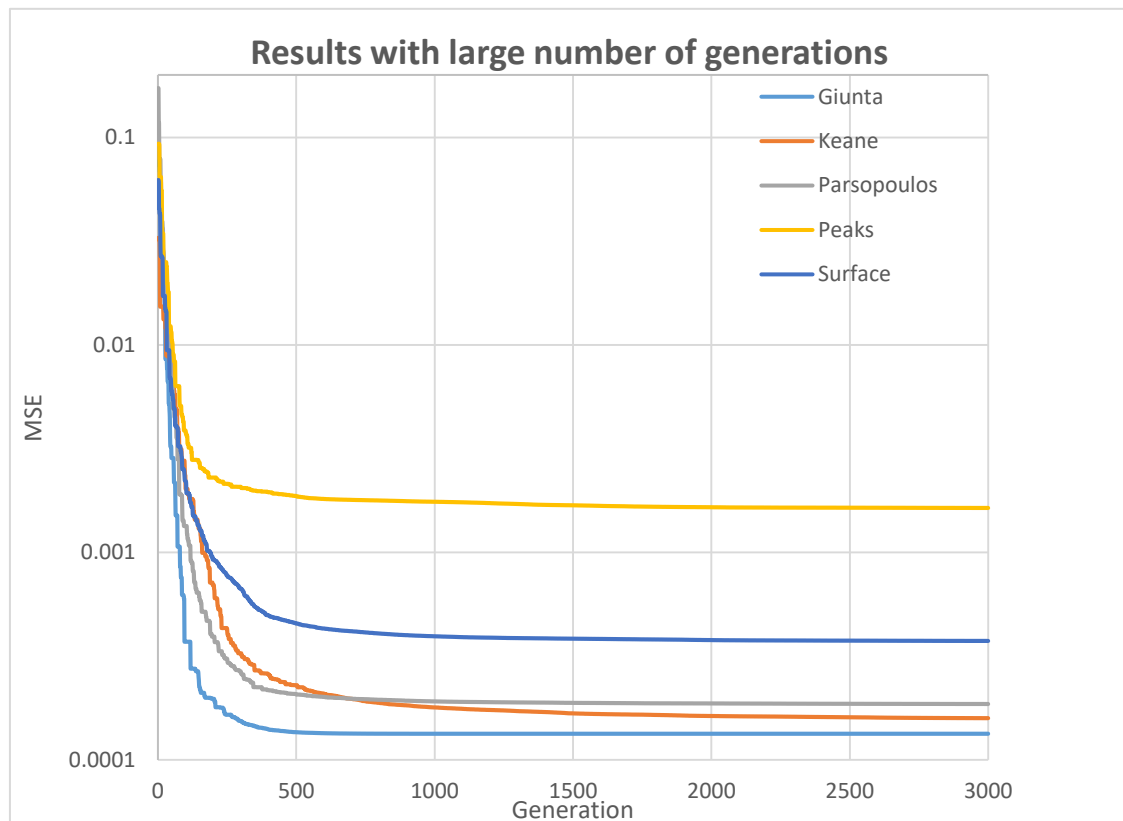
Εικόνα 43. Προσέγγιση τυχαίας επιφάνειας με χρήση NURBS και ΔΕΑ στη γενιά 30.



Εικόνα 44. Προσέγγιση τυχαίας επιφάνειας με χρήση NURBS και ΔΕΑ στη γενιά 300.

5.6 Αποτελέσματα επιφανειών με μεγάλο αριθμό γενιών

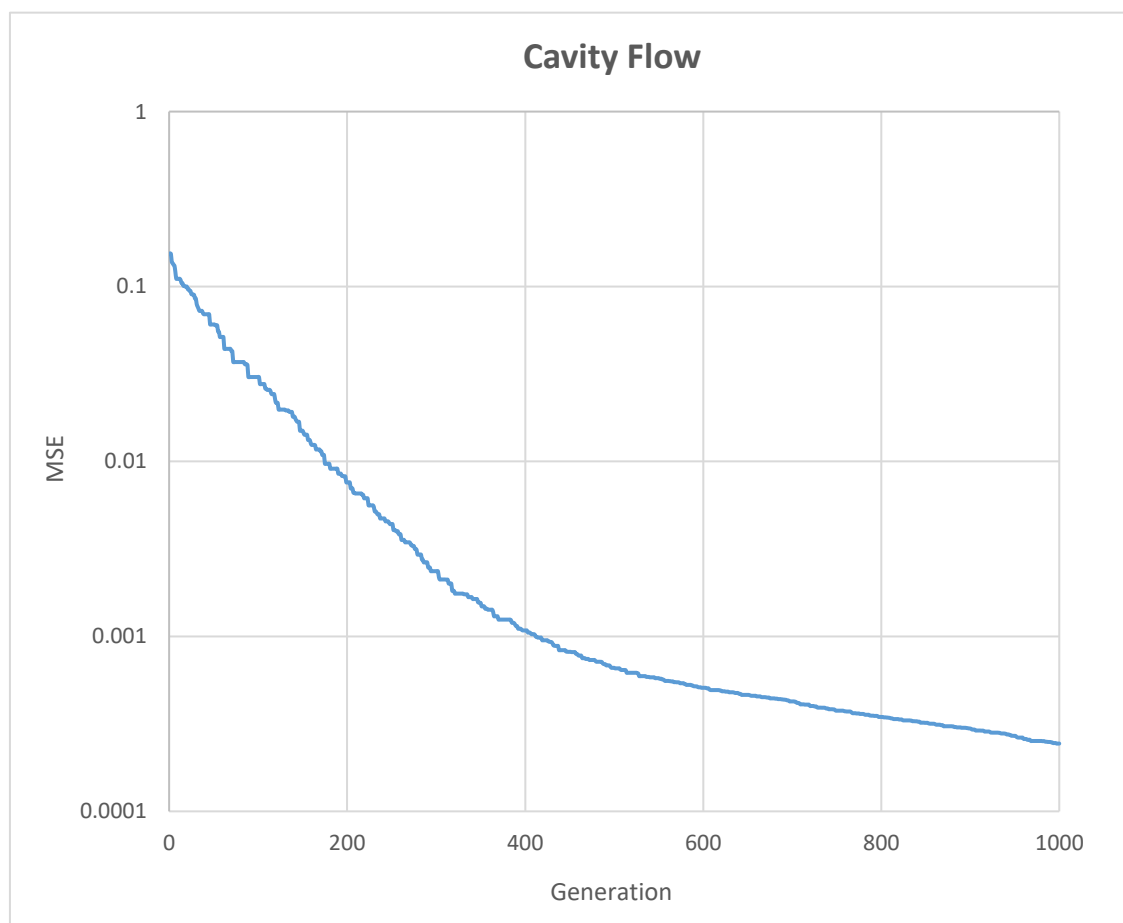
Στην [Εικόνα 45](#) παρουσιάζονται τα αποτελέσματα της σύγκλισης του ΔΕΑ για όλες τις επιφάνειες και με εφαρμογή των βέλτιστων παραμέτρων της κάθε επιφάνειας με αριθμό γενιών ίσο με 3000. Παρατηρείται ότι περίπου μετά τη γενιά 500, ο ρυθμός μείωσης του μέσου τετραγωνικού σφάλματος είναι αμελητέος.



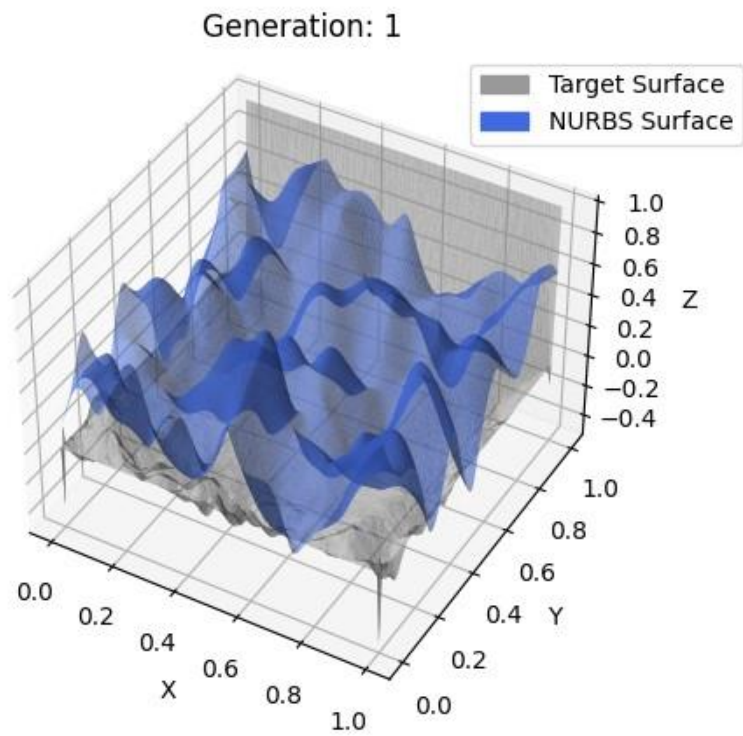
Εικόνα 45. Αποτελέσματα μέσου τετραγωνικού σφάλματος κάθε επιφάνειας, με εφαρμογή των βέλτιστων τιμών παραμέτρων και αριθμό γενιών 3000.

5.7 Αποτελέσματα επιφάνειας πραγματικού πεδίου ροής

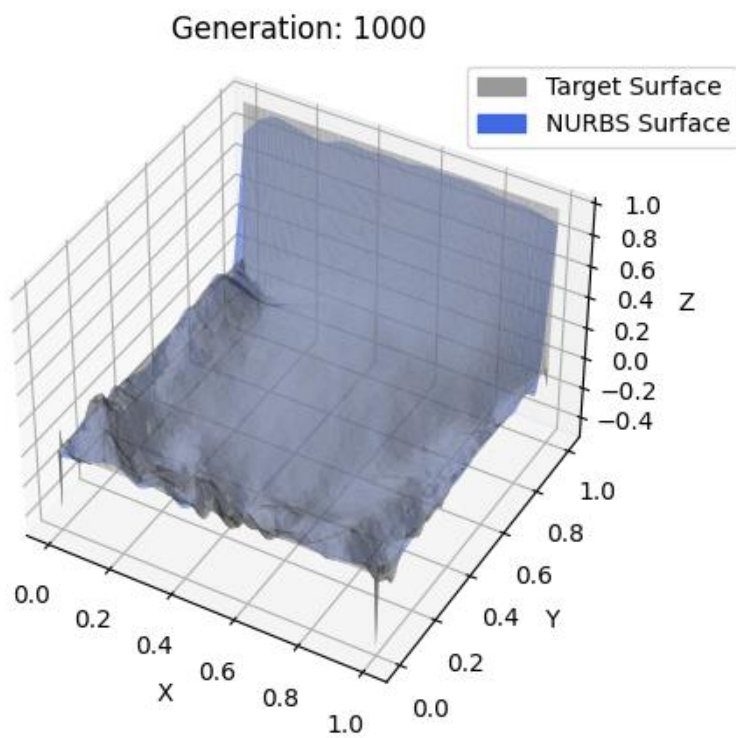
Στη συνέχεια, η προηγούμενη μεθοδολογία εφαρμόστηκε για ένα πραγματικό πεδίο ροής, που αφορά σε cavity flow. Αυτός ήταν και ο αρχικός στόχος της παρούσας εργασίας, δηλαδή να εξεταστεί η δυνατότητα εάν και κατά πόσον ένα πραγματικό πεδίο ροής (μία δηλαδή πολύπλοκη συνάρτηση 2 μεταβλητών) μπορεί να μοντελοποιηθεί ως παραμετρική συνάρτηση λίγων παραμέτρων. Στην [Εικόνα 46](#) παρουσιάζονται τα αποτελέσματα της σύγκλισης του ΔΕΑ για την επιφάνεια που έχει δημιουργηθεί από πραγματικά δεδομένα πεδίου ροής. Το μέσο τετραγωνικό σφάλμα σε αυτή την περίπτωση είναι 0.000244. Η επιφάνεια-στόχος, καθώς και η επιφάνεια NURBS σε δύο διαφορετικές γενιές παρουσιάζονται στις [Εικόνες 47](#) και [48](#).



Εικόνα 46. Αποτελέσματα μέσου τετραγωνικού σφάλματος ανά γενιά της επιφάνειας που δημιουργήθηκε από δεδομένα πραγματικού πεδίου ροής (cavity flow).



Εικόνα 47. Προσέγγιση πεδίου ροής με χρήση NURBS και ΔΕΑ στη γενιά 1.



Εικόνα 48. Προσέγγιση πεδίου ροής με χρήση NURBS και ΔΕΑ στη γενιά 1000.

Κεφάλαιο 6 – Συμπεράσματα

Ο στόχος της παρούσας Διπλωματικής Εργασίας ήταν πρωτίστως η δημιουργία ευέλικτων επιφανειών που περιγράφονται μαθηματικά, ώστε να μπορούμε να αλλάξουμε τη μορφή τους με λίγες παραμέτρους. Αυτή τη δυνατότητα μας τη δίνουν οι καμπύλες NURBS, που με τη χρήση σημείων ελέγχου και των αντίστοιχων βαρών τους μπορούμε να δημιουργήσουμε πολύπλοκες μορφές επιφανειών. Κατόπιν, με τη χρήση αυτών, αλλά και με τη χρησιμοποίηση Διαφορικού Εξελικτικού Αλγόριθμου (ως μεθόδου βελτιστοποίησης) προσπαθήσαμε να προσεγγίσουμε όσο το δυνατόν περισσότερο κάποιες επιφάνειες-στόχους, οι οποίες ως επί το πλείστον ήταν τροποποιήσεις γνωστών συναρτήσεων, που χρησιμοποιούνται για την αξιολόγηση, αλλά και τη σύγκριση αλγορίθμων βελτιστοποίησης.

Με βάση τα αποτελέσματα που παρουσιάζονται στο Κεφάλαιο 5, αλλά και των εικόνων που παρατίθενται σε αρχικό στάδιο του αλγορίθμου και μετά το κριτήριο τερματισμού του, μπορούμε να πούμε ότι ο αλγόριθμος σε συνδυασμό με τις επιφάνειες NURBS, είναι αρκετά ικανοποιητικός. Επίσης, μπορούμε να παρατηρήσουμε ότι η καλύτερη λύση που βρέθηκε από τον ΔΕΑ έχει σε κάθε πρόβλημα διαφορετικές ρυθμιστικές παραμέτρους, εκτός από το κριτήριο τερματισμού που έχει οριστεί το ίδιο σε κάθε περίπτωση. Οπότε, καταλήγουμε ότι ο ΔΕΑ για να μας δώσει τα βέλτιστα αποτελέσματα, πρέπει να ρυθμιστούν οι κατάλληλες παράμετροι στις πιο σωστές τιμές. Αυτό βέβαια δεν μπορούμε να το γνωρίζουμε εξ αρχής και χρειάζεται να γίνουν δοκιμές, ώστε να βρεθούν οι καταλληλότερες τιμές για το μέγεθος του πληθυσμού, της ρυθμιστικής παραμέτρου F και της πιθανότητας διασταύρωσης, που θα μας δίνουν το μικρότερο μέσο τετραγωνικό σφάλμα για δεδομένο αριθμό γενιών. Η μεθοδολογία εφαρμόστηκε και για την περίπτωση ενός πραγματικού πεδίου ροής, όπου φάνηκε η ικανότητα της μεθόδου να προσεγγίζει πολύ ικανοποιητικά και να περιγραφεί με ακρίβεια ένα πολύπλοκο πεδίο ροής με παραμετρικό τρόπο.

Κλείνοντας, με βάση τα πολύ ενθαρυντικά αποτελέσματα που επιτεύχθηκαν για την επιφάνεια που περιγράφει πραγματικό πεδίο ροής (cavity flow), ως επόμενος στόχος, θα είναι η εφαρμογή των αποτελεσμάτων που βρέθηκαν, δηλαδή των συντεταγμένων των βέλτιστων σημείων ελέγχου, καθώς και των αντίστοιχων βαρών που έχει υπολογίσει ο ΔΕΑ, σε Τεχνητά Νευρωνικά Δίκτυα τύπου PINN (Physics Informed Neural Networks) για την εκπαίδευσή τους, ώστε να είναι δυνατή η πρόβλεψη της μορφής των αντίστοιχων πεδίων ροής, μετά την εκπαίδευσή τους.

Βιβλιογραφία

- [1] J. S. Arora, *Introduction to optimum design*, 3rd ed. Boston, MA: Academic Press, 2011.
- [2] M. Hemmerling and U. Nether, "Generico: A case study on performance-based design," in *Proceedings of the XVIII Conference of the Iberoamerican Society of Digital Graphics - SIGraDi: Design in Freedom*, Montevideo, Uruguay: Editora Edgard Blücher, Dec. 2014, pp. 126–129. doi: 10.5151/despro-sigradi2014-0021.
- [3] K. Deb, *Optimization for engineering design: algorithms and examples*, Segunda edición, Decimocuarta impresión. New Delhi: PHI Learning Private Limited, 2012.
- [4] C. Darwin, *The origin of species: by means of natural selection of The preservation of favored races in the struggle for life*, 150th anniversary edition. New York: Signet classics, 2003.
- [5] Α. Α. Γκουντή, "Εξελικτικοί αλγόριθμοι: Ο αλγόριθμος της Διαφορικής Εξέλιξης," p. pages, 2014, doi: 10.26262/HEAL.AUTH.IR.135079.
- [6] D. B. Fogel, Ed., *Evolutionary computation: the fossil record*. New York: IEEE Press, 1998.
- [7] O. Kramer, *Genetic Algorithm Essentials*, 1st ed. 2017. in *Studies in Computational Intelligence*, no. 679. Cham: Springer International Publishing : Imprint: Springer, 2017. doi: 10.1007/978-3-319-52156-5.
- [8] J. H. Holland, *Adaptation in Natural and Artificial Systems - An Introductory Analysis with Applications to Biology, Control, and Artificial I.* Cambridge: The MIT Press, 2019.
- [9] D. Jong and K. Alan, "Analysis of the behavior of a class of genetic adaptive systems," 1975, Accessed: Jun. 05, 2023. [Online]. Available: <http://deepblue.lib.umich.edu/handle/2027.42/4507>
- [10] D. Simon, *Evolutionary optimization algorithms: biologically-Inspired and population-based approaches to computer intelligence*. Hoboken, New Jersey: John Wiley & Sons Inc, 2013.
- [11] "Artificial Intelligence through Simulated Evolution," in *Evolutionary Computation*, IEEE, 2009. doi: 10.1109/9780470544600.ch7.
- [12] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. in *Problemata*, 15. Stuttgart-Bad Cannstatt: Frommann-Holzboog, 1973.

- [13] A. Tripathy and H.-P. Schwefel, "Numerical Optimization of Computer Models," *J. Oper. Res. Soc.*, vol. 33, no. 12, p. 1166, Dec. 1982, doi: 10.2307/2581158.
- [14] X. Yu, "Introduction to evolutionary algorithms," in *The 40th International Conference on Computers & Industrial Engineering*, Awaji City, Japan: IEEE, Jul. 2010, pp. 1–1. doi: 10.1109/ICCIE.2010.5668407.
- [15] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann, "Evolutionary Algorithms," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 4, no. 3, pp. 178–195, May 2014, doi: 10.1002/widm.1124.
- [16] M. Georgioudakis and V. Plevris, "A Comparative Study of Differential Evolution Variants in Constrained Structural Optimization," *Front. Built Environ.*, vol. 6, p. 102, Jul. 2020, doi: 10.3389/fbuil.2020.00102.
- [17] G. C. Onwubolu and B. V. Babu, *New Optimization Techniques in Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.
- [18] M. J. Reddy and D. N. Kumar, "Multiobjective Differential Evolution with Application to Reservoir System Optimization," *J. Comput. Civ. Eng.*, vol. 21, no. 2, pp. 136–146, Mar. 2007, doi: 10.1061/(ASCE)0887-3801(2007)21:2(136).
- [19] E. Dimas and D. Briassoulis, "3D geometric modelling based on NURBS: a review," *Adv. Eng. Softw.*, vol. 30, no. 9–11, pp. 741–751, Sep. 1999, doi: 10.1016/S0965-9978(98)00110-0.
- [20] D. F. Rogers, *An introduction to NURBS: with historical perspective*. San Francisco: Morgan Kaufmann Publishers, 2001.
- [21] M. G. Cox, "The Numerical Evaluation of B-Splines," *IMA J. Appl. Math.*, vol. 10, no. 2, pp. 134–149, 1972, doi: 10.1093/imamat/10.2.134.
- [22] C. De Boor, "On calculating with B-splines," *J. Approx. Theory*, vol. 6, no. 1, pp. 50–62, Jul. 1972, doi: 10.1016/0021-9045(72)90080-9.
- [23] R. Riesenfeld, "Applications Of B-Spline Approximation To Geometric Problems Of Computer-Aided Design," *Electr. Eng. Comput. Sci. - Diss.*, Jan. 1973, [Online]. Available: https://surface.syr.edu/eecs_etd/258
- [24] W. J. Gordon and R. F. Riesenfeld, "Bernstein-Bézier Methods for the Computer-Aided Design of Free-Form Curves and Surfaces," *J. ACM*, vol. 21, no. 2, pp. 293–310, Apr. 1974, doi: 10.1145/321812.321824.
- [25] H. Fan, J. Liu, and J. Lampinen, "Some improvement to the mutation donor of differential evolution," *Eng. Comput.*, vol. 27, no. 2, pp. 225–242, Mar. 2010, doi: 10.1108/026444401011022382.

- [26] O. R. Bingol, “geomdl: Object-oriented B-Spline and NURBS evaluation library.” Accessed: Jun. 05, 2023. [OS Independent]. Available: <https://github.com/orbingol/NURBS-Python>
- [27] P. Virtanen *et al.*, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nat. Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/s41592-019-0686-2.
- [28] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.