



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΣΧΟΛΗ: ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ: ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Διδακτορική Διατριβή

Techniques Of Global Optimization

Τίτλος στα Ελληνικά
«**Τεχνικές Καθολικής Βελτιστοποίησης**»

Vasileios Charilogis

Βασίλειος Χαριλόγης

ΑΡΤΑ, 2024

ΕΥΧΑΡΙΣΤΙΕΣ

Πρώτα και πάνω απ' όλα, θα ήθελα να ευχαριστήσω τον Θεό για τη δύναμη, την καθοδήγηση και την ευλογία Του καθ' όλη τη διάρκεια της εκπόνησης αυτής της διατριβής. Η πίστη μου, υπήρξε πηγή έμπνευσης και αντοχής, ειδικά στις πιο δύσκολες στιγμές αυτής της πορείας.

Θα ήθελα επίσης να ευχαριστήσω τη σύζυγό μου, Άννα Μαρία Γιάννη. Η αδιάκοπη υποστήριξή της, η κατανόησή της και η αγάπη της ήταν αναντικατάστατες κατά τη διάρκεια αυτής της πορείας. Ανέλαβε μεγάλο μέρος του οικογενειακού βάρους, επιτρέποντάς μου να επικεντρωθώ στην εργασία μου, και ήταν πάντα δίπλα μου σε κάθε βήμα, προσφέροντάς μου δύναμη και κουράγιο. Χωρίς την αμέριστη στήριξή της, αυτή η διατριβή δεν θα είχε ολοκληρωθεί.

Θερμές ευχαριστίες οφείλω και στον καθηγητή μου, Ιωάννη Τσούλο, για την καθοδήγηση, την υποστήριξη και τις πολύτιμες συμβουλές του καθ' όλη τη διάρκεια της εκπόνησης της διατριβής. Η εμπειρία και η γνώση του, υπήρξαν ανεκτίμητες για την ανάπτυξη και την ολοκλήρωση αυτής της διατριβής. Η εμπιστοσύνη που μου έδειξε και οι προτροπές του για επιστημονική ακρίβεια και καινοτομία, αποτέλεσαν πηγή έμπνευσης και ενθάρρυνσης.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την συνεχή υποστήριξη και κατανόηση τους. Η παρουσία και οι ενθαρρύνσεις τους υπήρξαν καθοριστικοί παράγοντες για την επίτευξη αυτού του στόχου.

Με βαθιά εκτίμηση και ευγνωμοσύνη,

Βασίλειος Χαριλόγης.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<i>ΕΥΧΑΡΙΣΤΙΕΣ</i>	3
1. ΕΙΣΑΓΩΓΗ	13
1.1 ΟΡΙΣΜΟΙ	13
1.2 ΓΡΑΜΜΙΚΗ ΑΝΑΖΗΤΗΣΗ	28
1.3 ΤΟΠΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ	32
1.4 ΚΑΘΟΛΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ	53
2. ΜΕΘΟΔΟΙ ΠΟΛΛΑΠΛΩΝ ΕΚΚΙΝΗΣΕΩΝ	59
2.1 PURE RANDOM SEARCH	59
2.2 Η ΜΕΘΟΔΟΣ MULTISTART	61
2.3 ΤΕΧΝΙΚΕΣ ΤΕΡΜΑΤΙΣΜΟΥ	67
2.4 ΤΟΠΟΓΡΑΦΙΚΕΣ ΤΕΧΝΙΚΕΣ	73
2.5 Η ΜΕΘΟΔΟΣ MINCENTER	79
2.6 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	82
3. Η ΜΕΘΟΔΟΣ CONTROLLED RANDOM SEARCH	89
3.1 ΒΑΣΙΚΕΣ ΑΡΧΕΣ	89
3.2 Η ΜΕΘΟΔΟΣ CONTROLLED RANDOM SEARCH	89
3.3 ΠΑΡΑΛΛΑΓΕΣ ΤΗΣ ΜΕΘΟΔΟΥ CONTROLLED RANDOM SEARCH	92
3.4 Η ΤΡΟΠΟΠΟΙΗΜΕΝΗ ΜΕΘΟΔΟΣ CONTROLLED RANDOM SEARCH	93
4. Η ΔΙΑΦΟΡΙΚΗ ΕΞΕΛΙΚΤΙΚΗ ΜΕΘΟΔΟΣ	103
4.1 ΒΑΣΙΚΕΣ ΑΡΧΕΣ	103
4.2 Ο ΑΛΓΟΡΙΘΜΟΣ DE	105
4.3 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΙΑΦΟΡΟΠΟΙΗΣΕΙΣ	106
5. ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΜΗΝΟΥΣ ΣΩΜΑΤΙΔΙΩΝ	117
5.1 ΒΑΣΙΚΕΣ ΑΡΧΕΣ - ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ	117
5.2 Ο ΑΛΓΟΡΙΘΜΟΣ PSO	118
5.3 ΠΑΡΑΛΛΑΓΕΣ ΤΗΣ ΜΕΘΟΔΟΥ PSO	119
5.4 ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΔΙΑΦΟΡΟΠΟΙΗΣΕΙΣ	125
6. ΠΑΡΑΛΛΗΛΕΣ ΤΕΧΝΙΚΕΣ	143
6.1 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΝΗΜΑΤΑ – ΒΑΣΙΚΕΣ ΑΡΧΕΣ	143
6.2 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ OPENMPI	146
6.3 ΠΑΡΑΛΛΗΛΕΣ ΤΕΧΝΙΚΕΣ ΠΟΛΛΑΠΛΩΝ ΕΚΚΙΝΗΣΕΩΝ	147
6.4 ΠΑΡΑΛΛΗΛΗ ΤΕΧΝΙΚΗ DE	148
6.5 ΠΑΡΑΛΛΗΛΗ ΤΕΧΝΙΚΗ PSO	151
6.6 ΑΝΤΙΚΕΙΜΕΝΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ	153
6.7 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	154

ΠΕΡΙΕΧΟΜΕΝΑ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 1:Το αποτέλεσμα της χρήσης του νέου σχήματος για το διαφορετικό βάρος.	114
Διάγραμμα 2:Σχεδιάγραμμα κλήσεων συναρτήσεων χρησιμοποιώντας τις δύο τροποποιήσεις.....	115
Διάγραμμα 3:Συγκρίσεις χρόνου για μια ποικιλία συναρτήσεων δοκιμής	116
Διάγραμμα 4:Αποτελέσματα σύγκρισης μεθόδων	116
Διάγραμμα 5:Τυπική απόκλιση των κλήσεων συνάρτησης για τους κανόνες τερματισμού χωρίς έλεγχο της κλίσης	139
Διάγραμμα 6:Τυπική απόκλιση των κλήσεων συνάρτησης για τους κανόνες τερματισμού με έλεγχο της κλίσης	139
Διάγραμμα 7:Πειράματα με τη συνάρτηση SINU για μια σειρά από διαστάσεις προβλημάτων από $n = 2$ έως $n = 32$	140
Διάγραμμα 8:Πειράματα με τον προτεινόμενο κανόνα τερματισμού με έλεγχο της κλίσης και χωρίς έλεγχο της κλίσης.....	140
Διάγραμμα 9:Στατιστική σύγκριση για την προτεινόμενη μέθοδο και διαφορετικούς αριθμούς νημάτων.....	160
Διάγραμμα 10:Στατιστική σύγκριση για τα αποτελέσματα της προτεινόμενης μεθόδου έναντι διαφορετικών παραλλαγών της μεθόδου DE	160
Διάγραμμα 11:Διάγραμμα κουτιών για τη σύγκριση μεταξύ διαφορετικού αριθμού μονάδων επεξεργασίας. Η μέθοδος διάδοσης ορίστηκε σε 1 προς 1	163
Διάγραμμα 12:Διενεργήθηκε ένα τεστ Friedman για να καθοριστεί εάν υπήρχαν σημαντικές διαφορές στα αποτελέσματα των κλήσεων συνάρτησης σε πέντε επαναλαμβανόμενους αριθμούς μονάδων ($N_i = 1, 2, 4, 5$ και 10). Το Friedman Chisquare ήταν 94.38 με 4 βαθμούς ελευθερίας, υποδεικνύοντας μια σημαντική διαφορά μεταξύ των ομάδων ($p < 0.0001$).	163
Διάγραμμα 13:Μέσος χρόνος εκτέλεσης για τη συνάρτηση GKLS. Η διάσταση της συνάρτησης ποικίλλει από δύο έως πέντε.....	165
Διάγραμμα 14:Μέσος αριθμός κλήσεων συνάρτησης για διάφορες περιπτώσεις της συνάρτησης GKLS.....	165
Διάγραμμα 15:Ο μέσος όρος επιτυχίας στην εύρεση του καθολικού ελάχιστου για διαφορετικές περιπτώσεις της συνάρτησης Gkls.....	166

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 1 1: Σημεία τοπικών μεγίστων	14
Εικόνα 1 2: Σημεία τοπικού ελαχίστου	14
Εικόνα 1 3: Τοπικό μέγιστο - Ολικό μέγιστο	15
Εικόνα 1 4: Μέγιστη και ελάχιστη τιμή	15
Εικόνα 1 5: Οι διαφορετικοί τύποι ελαχίστων	18
Εικόνα 1 6: Διαφορετικοί τύποι στάσιμων σημείων για μια συνάρτηση δύο μεταβλητών.....	19
Εικόνα 1 7: Η επιλογή των εσωτερικών σημείων στην αναζήτηση διχοτόμησης.....	29
Εικόνα 1 8: Τοπική βελτιστοποίηση: $f(x^*) \leq f(x) \forall x \in S$	33
Εικόνα 1 9: Επανάληψη της μεθόδου Nelder-Mead σε δισδιάστατο χώρο	34
Εικόνα 1 10: Ανάκληση σε δύο διαστάσεις.....	35
Εικόνα 1 11: Διαστολή σε δύο διαστάσεις	36
Εικόνα 1 12: Σύστωση σε δύο διαστάσεις	36
Εικόνα 1 13: Συρρίκνωση σε δύο διαστάσεις	37
Εικόνα 1 14: Διάγραμμα αναζήτησης χρυσής τομής	40
Εικόνα 1 15: Σχηματική αναπαράσταση της αναζήτησης με διαμέριση	56
Εικόνα 1 16: Αναπαράσταση της συρρίκνωσης του διαστήματος αβεβαιότητας	57

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

Πίνακας 1: Συγκριτικά αποτελέσματα των μεθόδων Multistart, TMLSL και GTC με τη χρήση του κριτηρίου τερματισμού του διπλού κουτιού.....	82
Πίνακας 2: Αποτελέσματα της μεθόδου Minfinder με τη χρήση του κριτηρίου τερματισμού του διπλού κουτιού.....	84
Πίνακας 3: Αποτελέσματα της μεθόδου Multistart.....	85
Πίνακας 4: Τα αποτελέσματα της προτεινόμενης μεθόδου με τη $K=100$ κέντρα.....	86
Πίνακας 5: Συγκριτικά αποτελέσματα διαφορετικών εκδοχών της CRS σε μια σειρά από συναρτήσεις.....	96
Πίνακας 6: Χρονικές συγκρίσεις.....	99
Πίνακας 7: Πειράματα με τον κανόνα τερματισμού Ali.....	110
Πίνακας 8: Πειράματα με τον προτεινόμενο κανόνα τερματισμού.....	112
Πίνακας 9: Τιμές για τις παραμέτρους των πειραματικών παραμέτρων.....	127
Πίνακας 10: Πειράματα με τον κανόνα τερματισμού Ali χωρίς έλεγχο της κλίσης.....	130
Πίνακας 11: Πειράματα με τον κανόνα τερματισμού Ali με τον έλεγχο της κλίσης ενεργοποιημένο.....	131
Πίνακας 12: Πειράματα με τον διπλό κανόνα τερματισμού χωρίς έλεγχο της κλίσης.....	133
Πίνακας 13: Πειράματα με τον διπλό κανόνα τερματισμού με τον έλεγχο της κλίσης ενεργοποιημένο.....	134
Πίνακας 14: Πειράματα με τον προτεινόμενο κανόνα τερματισμού χωρίς τον έλεγχο της κλίσης.....	136
Πίνακας 15: Πειράματα με τον προτεινόμενο κανόνα τερματισμού με τον έλεγχο της κλίσης ενεργοποιημένο.....	137
Πίνακας 16: Αντικειμενικές συναρτήσεις που χρησιμοποιήθηκαν.....	153
Πίνακας 17: Τιμές παραμέτρων.....	155
Πίνακας 18: Πειράματα για την προτεινόμενη μέθοδο χρησιμοποιώντας διαφορετικές επιλογές για τη μέθοδο διάδοσης. Ο αριθμός των νημάτων επεξεργασίας ορίστηκε σε 10. Οι αριθμοί στα κελιά αντιπροσωπεύουν τον μέσο αριθμό κλήσεων συνάρτησης για κάθε δοκιμαστική συνάρτηση.....	156
Πίνακας 19: Σύγκριση της προτεινόμενης μεθόδου έναντι άλλων παραλλαγών της τεχνικής διαφορικής εξέλιξης.....	158
Πίνακας 20: Παράμετροι της προτεινόμενης μεθόδου.....	161
Πίνακας 21: Αποτελέσματα του πειράματος χρησιμοποιώντας την προτεινόμενη μέθοδο. Το σχήμα διάδοσης ορίστηκε σε 1 προς 1 και η τιμή του NP ορίστηκε σε 5. Στα διενεργηθέντα πειράματα, ο αριθμός των παράλληλων μονάδων επεξεργασίας ποικίλλονταν από 1 έως 10.....	162

1.ΕΙΣΑΓΩΓΗ

1.1 Ορισμοί

Η έννοια της συνάρτησης

Έστω A ένα υποσύνολο του \mathbb{R} , ονομάζουμε πραγματική συνάρτηση με πεδίο ορισμού το A μια διαδικασία f , με την οποία κάθε στοιχείο $x \in A$ αντιστοιχίζεται σε ένα μόνο πραγματικό αριθμό y . Το y ονομάζεται τιμή της f στο x και συμβολίζεται με $f(x)$. Προκειμένου να εκφραστεί η διαδικασία αυτή, γράφεται ως εξής:

$$f: A \rightarrow \mathbb{R}$$

$$x \in f(x)$$

Το γράμμα x , παριστάνει οποιοδήποτε στοιχείο του A και λέγεται ανεξάρτητη μεταβλητή, ενώ το γράμμα y , παριστάνει την τιμή της f στο x και λέγεται εξαρτημένη μεταβλητή. Το πεδίο ορισμού A της συνάρτησης f συμβολίζεται με D_f . Ενώ, το σύνολο που έχει για στοιχεία του τις τιμές της f σε όλα τα $x \in A$, λέγεται σύνολο τιμών της f και συμβολίζεται με $f(A)$. Δηλαδή [1]:

$$f(A) = \{ y \mid y = f(x) \text{ για κάποιο } x \in A \}.$$

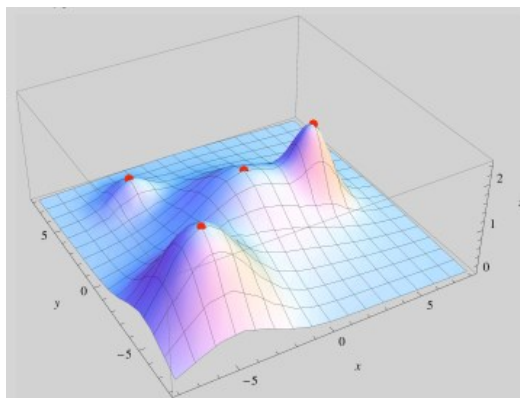
Ακρότατα συνάρτησης

Έστω μία συνάρτηση $f: \Delta \rightarrow \mathbb{R}$ και Δ διάστημα υποσύνολο του \mathbb{R} . Τότε ισχύει ότι:

- i. Η συνάρτηση f παρουσιάζει στο $x_0 \in \Delta$ (ολικό) μέγιστο, το $f(x_0)$ όταν για κάθε πραγματικό αριθμό x στο διάστημα Δ ισχύει:

$$f(x) \leq f(x_0)$$

Δηλαδή η $f(x_0)$ είναι η μεγαλύτερη τιμή, συγκρινόμενη με οποιαδήποτε άλλη τιμή της συνάρτησης στο διάστημα Δ και λέγεται η μέγιστη τιμή της f στο πεδίο ορισμού της Δ . [2, 3]

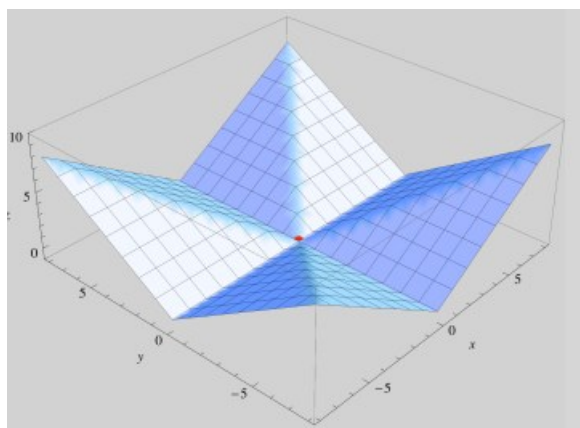


Εικόνα 1 1: Σημεία τοπικών μεγίστων

- ii. Η συνάρτηση f παρουσιάζει στο $x_0 \in \Delta$ (ολικό) ελάχιστο, το $f(x_0)$, όταν για κάθε πραγματικό αριθμό x στο Δ ισχύει:

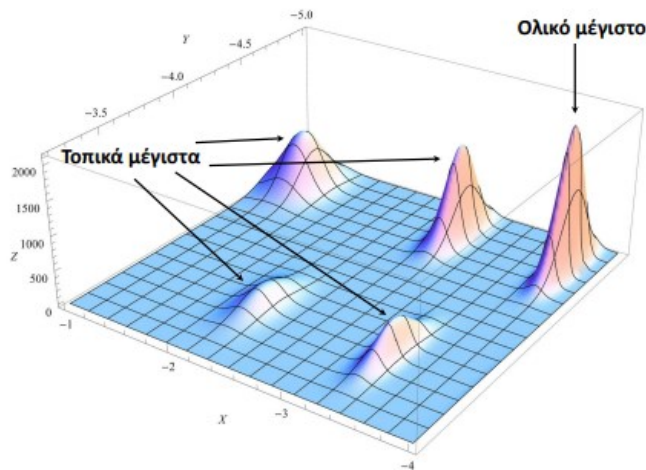
$$f(x) \geq f(x_0)$$

Δηλαδή η $f(x_0)$ είναι η μικρότερη τιμή, συγκρινόμενη με οποιαδήποτε άλλη τιμή της συνάρτησης στο διάστημα Δ και λέγεται η ελάχιστη τιμή της f στο πεδίο ορισμού της Δ . [2,3]



Εικόνα 1 2: Σημεία τοπικού ελαχίστου

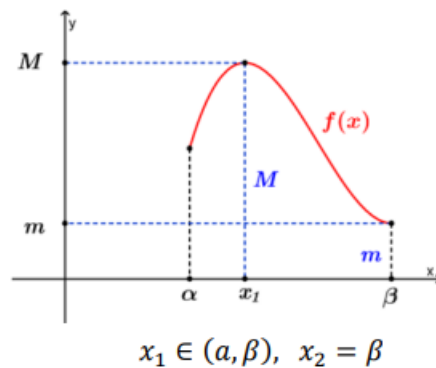
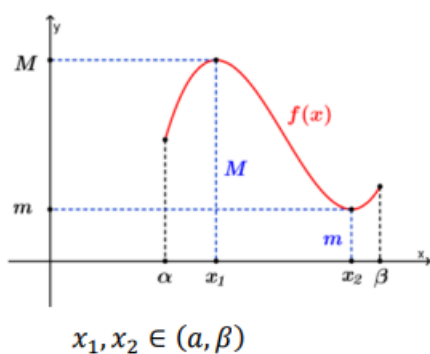
- iii. Το (ολικό) μέγιστο και το (ολικό) ελάχιστο της f είναι τα (ολικά) ακρότατα της f . [2,4]



Εικόνα 1 3: Τοπικό μέγιστο - Ολικό μέγιστο

Για παράδειγμα, έστω η πολυωνμική συνάρτηση $f: [\alpha, \beta] \rightarrow R$. Τότε, υπάρχει $x_1, x_2 \in [\alpha, \beta]$ τέτοια ώστε: $f(x) \leq f(x_1) = M$ και $f(x) \geq f(x_2) = m$, για κάθε $x \in [\alpha, \beta]$. Οπότε:

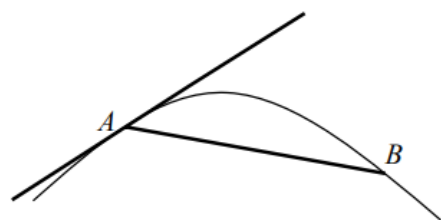
- Η τιμή $f(x_1) = M$ είναι η μέγιστη τιμή της f στο $[\alpha, \beta]$.
- Η τιμή της $f(x_2) = m$ είναι η ελάχιστη τιμή της f στο $[\alpha, \beta]$. [2]



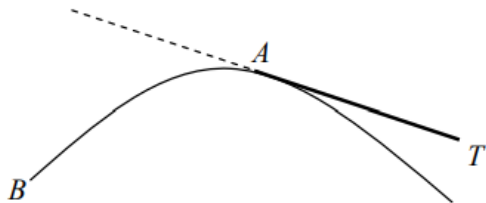
Εικόνα 1 4: Μέγιστη και ελάχιστη τιμή

Η έννοια της παραγώγου

Εάν A και B είναι δύο σημεία πάνω σε μία καμπύλη, τότε η γραμμή που ενώνει τα σημεία A και B ονομάζεται χορδή, ενώ η γραμμή που τέμνει την



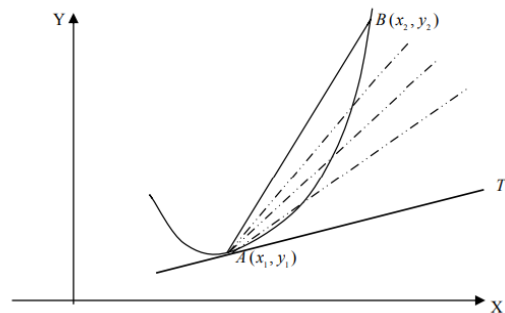
καμπύλη μόνο στο σημείο A ονομάζεται εφαπτομένη στο A. [5] Επειδή μία καμπύλη δεν έχει σε όλα τα σημεία της την ίδια κλίση, προήλθε η ανάγκη προσδιορισμού της εφαπτόμενης μιας καμπύλης σε ένα συγκεκριμένο σημείο με την έννοια της παραγώγου.



Ας υποθέσουμε ότι κινούμαστε πάνω σε μια καμπύλη από το σημείο B στο σημείο A και στο σημείο A η κλίση αλλάζει και παραμένει σταθερή. Εάν συμβεί αυτό τότε θα κινηθούμε πάνω στην ευθεία γραμμή AT, δηλαδή στην εφαπτομένη στο σημείο A. Οπότε η κλίση της καμπύλης στο σημείο A είναι η ίδια με την κλίση

της εφαπτομένης στο σημείο A. [5]

Έστω ότι παίρνουμε δύο σημεία $A(x_1, y_1)$ και $B(x_2, y_2)$ πάνω στη καμπύλη $y = f(x)$ και έστω ότι κινούμαστε πάνω στη καμπύλη αυτή από το σημείο B προς το σημείο A και επιθυμούμε να υπολογισθεί η παράγωγος στο σημείο A. [5]



Αυτό σημαίνει ότι το σημείο A παραμένει σταθερό, ενώ το σημείο B κινείται και πλησιάζει προς το σημείο A. Όσο πιο κοντά είναι το B στο A, τόσο καλύτερη είναι η προσέγγιση που θα πάρουμε.

Δηλαδή, $B \rightarrow A$ ή

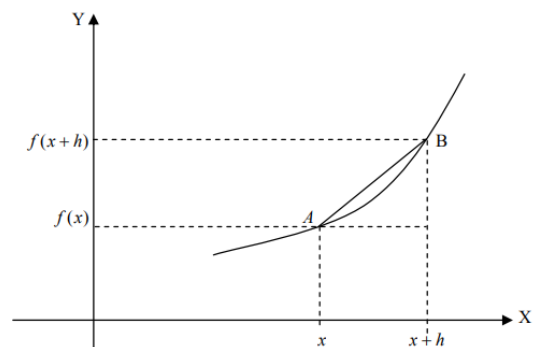
η κλίση της ευθείας (χορδής) AB \rightarrow κλίση της εφαπτομένης AT ή

$$\lim_{B \rightarrow A} (\text{κλίση χορδής } AB) = \text{κλίση της εφαπτομένης } AT.$$

Αν παρατηρήσουμε μεταξύ δύο πολύ κοντινών σημείων πάνω στη καμπύλη, όπου h είναι ένας αριθμός πολύ κοντά στο μηδέν, θα διαπιστώσουμε ότι η κλίση της ευθείας (χορδής) AB δίνεται από τον εξής τύπο:

$$\frac{\text{μεταβολή του } y}{\text{μεταβολή του } x} = \frac{f(x+h) - f(x)}{(x+h) - x} \quad \text{ή}$$

$$\frac{f(x+h) - f(x)}{h} = \frac{\Delta f}{\Delta x}$$



όπου: $\frac{\Delta f}{\Delta x}$ είναι ο λόγος μεταβολής της συνάρτησης $f(x)$ και τείνει σε ένα συγκεκριμένο όριο. Ο ρυθμός μεταβολής της συνάρτησης στο σημείο x ονομάζεται παράγωγος της συνάρτησης και ορίζεται ως εξής:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Επομένως, η παράγωγος μιας συνάρτησης $y = f(x)$ είναι η συνάρτηση $f'(x)$ που η τιμή της σε κάθε x ορίζεται από τον κανόνα :

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

όταν υπάρχει αυτό το όριο. Το πεδίο ορισμού της f' είναι το σύνολο των σημείων του πεδίου ορισμού της f όπου υπάρχει το εν λόγω όριο. [5]

Συνθήκες για την ύπαρξη ελαχίστου

Ένα σημείο x^* , σημείο στο οποίο η $f(x)$ παρουσιάζει τοπικό ελάχιστο, για το οποίο ισχύει:

$$f(x^*) \leq f(x)$$

για κάθε σημείο $x \neq x^*$, στη γειτονιά του x^* , ονομάζεται ασθενές τοπικό ελάχιστο.

Ένα σημείο x^* , σημείο στο οποίο η $f(x)$ παρουσιάζει τοπικό ελάχιστο, για το οποίο ισχύει

$$f(x^*) < f(x)$$

για κάθε σημείο $x \neq x^*$, στη γειτονιά του x^* , ονομάζεται ισχυρό τοπικό ελάχιστο.

Μια συνάρτηση μπορεί να έχει περισσότερα από ένα τοπικά ελάχιστα. Το τοπικό ελάχιστο που αντιστοιχεί στη μικρότερη τιμή της αντικειμενικής συνάρτησης ονομάζεται καθολικό ή γενικό ή απόλυτο ελάχιστο και για το αντίστοιχο σημείο x^* ισχύει $f(x^*) < f(x)$ για κάθε x στο πεδίο ορισμού της συνάρτησης. Οι διαφορετικοί τύποι ελαχίστου παρουσιάζονται παρακάτω [6]:



Εικόνα 1 5:Οι διαφορετικοί τύποι ελαχίστων

Αν υποθέσουμε ότι η $f(x)$ είναι δύο φορές συνεχώς διαφορίσιμη τότε μπορεί να αποδειχθούν συνθήκες οι οποίες μπορούν να ελεγχθούν πιο εύκολα συγκριτικά με τους αναφερόμενους ορισμούς. Σε αυτή την περίπτωση για να παρουσιάζει η $f(x)$ ισχυρό τοπικό ελάχιστο στο σημείο x^* ισχύουν οι ακόλουθες συνθήκες:

Αναγκαίες συνθήκες:

- $\nabla f(x) = 0$
- $s^T G^* s \geq \forall s$

Ικανές συνθήκες:

- $\nabla f(x) = 0$
- $s^T G^* s > 0, \forall s \neq 0$ [6]

Επειδή η συνθήκη $s^T G^* s > 0$ είναι δύσκολο να ελεγχθεί υπολογιστικά, γίνεται χρήση της ισοδύναμης διατύπωσης ότι οι ιδιοτιμές του G^* είναι θετικές.

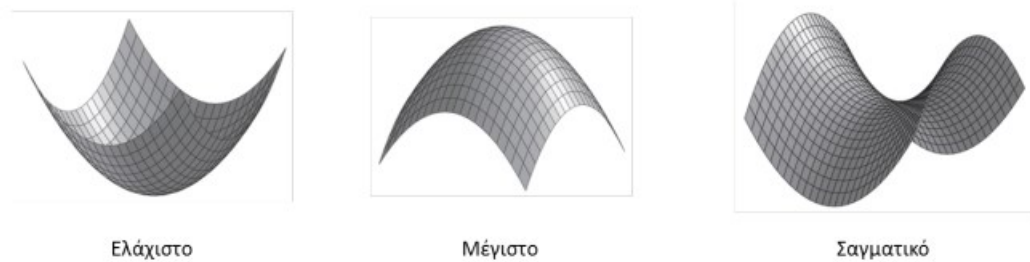
Η συνθήκη $\nabla f(x) = 0$, δεν εξασφαλίζει από μόνη της την ύπαρξη ελαχίστου. Τα σημεία που την ικανοποιούν ονομάζονται στάσιμα σημεία και μπορούν να ταξινομηθούν περαιτέρω, εξετάζοντας τον Εσσιανό πίνακα. Ο πίνακας αυτός μπορεί να είναι:

-Θετικά ορισμένος, όπου όλες οι τιμές του είναι θετικές: Τότε η $f(x)$, παρουσιάζει ελάχιστο στο x^* .

-Αρνητικά ορισμένος, όπου όλες οι τιμές του είναι αρνητικές: Τότε η $f(x)$, παρουσιάζει μέγιστο στο σημείο x^* .

-Έχει ιδιοτιμές αμφοτέρων προσήμων: Τότε το σημείο x^* λέγεται σαγματικό σημείο.

- Αόριστος όπου: έχει ιδιοτιμές αμφοτέρων προσήμων και μηδενικές ιδιοτιμές. [6]



Εικόνα 1 6: Διαφορετικοί τύποι στάσιμων σημείων για μια συνάρτηση δύο μεταβλητών

Ανάπτυγμα Taylor για συναρτήσεις μιας ή πολλών μεταβλητών

Για μια συνάρτηση μιας μεταβλητής $f(x)$, η τιμή της σε κάποιο γειτονικό σημείο x δίνεται από την σειρά Taylor:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)(\Delta x)^2 + \dots + \frac{1}{n!}f^{(n)}(k) + \dots \quad (1.1)$$

είτε από ένα πεπερασμένο ανάπτυγμα, με βάση τον τύπο του Taylor:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \dots + \frac{1}{n!}f^{(n)}(k)(\Delta x)^n \quad (1.2)$$

όπου k είναι κάποιο σημείο στο διάστημα μεταξύ x και Δx , δηλαδή είναι: $k=x+c\Delta x$,

όπου c είναι βαθμωτό μέγεθος τέτοιο ώστε $0 < c < 1$. [7]

Στη γενική περίπτωση όπου η συνάρτηση $f(x) = f(x_1, x_2, \dots, x_n)$ είναι συνάρτηση n -μεταβλητών, τότε στη γειτονιά του x η συνάρτηση $f(x + \Delta x)$ δίνεται σύμφωνα με τη σχέση:

$$g_{(t)} = f(x^* + t\Delta x) = f(x^*_1 + t\Delta_{x_1}, x^*_2 + t\Delta_{x_2}, \dots, x^*_n + t\Delta_{x_n}) \quad (1.3)$$

όπου:

Δx : θεωρείται η μεταβολή και $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ το αρχικό σημείο

από την ισοδύναμη έκφραση: $f(x + \Delta x) = g(t) \vee |_{t=1} = g(1)$ όπου $f(x) = g(0)$.

Τότε για την συνάρτηση μιας μεταβλητής $g(t)$ θα ισχύουν τα αναπτύγματα Taylor (1.1) και (1.2). Με αυτό το τρόπο ισχύει:

$$g(1) = g(0) + g'(0) + \frac{1}{2}g''(0) + \dots + \frac{1}{n!}g^{(n)}(0) + \dots \quad (1.4) \text{ ή}$$

$$g(1) = g(0) + g'(0) + \frac{1}{2}g''(0) + \dots + \frac{1}{n!}g^{(n)}(k) \quad (1.5)$$

όπου το k παίρνει τιμή στο διάστημα $0 < k < 1$ και $\Delta x = 1$. [7]

Για τον υπολογισμό των αναπτυγμάτων (1.4) και (1.5) είναι απαραίτητο να υπολογισθούν οι παράγωγοι της $g(t)$ όταν $t \square 0$, δηλαδή να υπολογιστούν η $g'(t)$, $g''(t)$, \dots , $g^{(n)}(t)$ για $t \square 0$.

$$\text{Οπότε για } g'(t) \vee |_{t=0} = \frac{d}{dt} f(x + t\Delta x) \vee |_{t=0} \quad (1.6)$$

Η ολική παράγωγος της f μπορεί να εκφραστεί ως προς τις μερικές παραγώγους της από τη σχέση:

$$\begin{aligned} \frac{d}{dt} f(x + t\Delta x) &= \frac{\partial f(x + t\Delta x)}{\partial(x_1 + t\Delta x_1)} \frac{d(x_1 + t\Delta x_1)}{dt} + \frac{\partial f(x + t\Delta x)}{\partial(x_2 + t\Delta x_2)} \frac{d(x_2 + t\Delta x_2)}{dt} + \dots + \\ &\frac{\partial f(x + t\Delta x)}{\partial(x_n + t\Delta x_n)} \frac{d(x_n + t\Delta x_n)}{dt} \quad (1.7) \end{aligned}$$

Όπου Δx θεωρείται ως παράμετρος και όχι μεταβλητή και ισχύει:

$$\frac{d(x_i + t\Delta x_i)}{dt}, \text{ για } i=1, 2, \dots, n \quad (1.8)$$

Αν αντικατασταθούν οι σχέσεις (1.7) και (1.8) στην (1.6) τότε για $t \square 0$

$$g'(0) = \frac{\partial f(x)}{\partial x_1} \Delta x_1 + \frac{\partial f(x)}{\partial x_2} \Delta x_2 + \dots + \frac{\partial f(x)}{\partial x_n} \Delta x_n \quad (1.9)$$

ή υπό μορφή διανυσμάτων $g'(0) = \nabla T_{f(x)} \Delta x$ (1.10), όπου ∇f η βάρθρωση της f :

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \dots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix} \text{ και } \Delta x = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \dots \\ \Delta x_n \end{bmatrix} \quad (1.11)$$

Με τον αντίστοιχο τρόπο βρίσκετε η δεύτερη παράγωγος:

$$\begin{aligned} g''(t) \Big|_{t=0} &= \frac{d}{dt} \Big|_t \\ &= 0 \left[\frac{\partial f(x + t\Delta x)}{\partial(x_1 + t\Delta x_1)} \frac{d(x_1 + t\Delta x_1)}{dt} + \dots + \frac{\partial f(x + t\Delta x)}{\partial(x_n + t\Delta x_n)} \frac{d(x_n + t\Delta x_n)}{dt} \right] \end{aligned}$$

Θεωρώντας την ολική παράγωγο σαν άθροισμα των μερικών παραγώγων της προηγούμενης έκφρασης και κάνοντας αριθμητικές πράξεις προκύπτει:

$$g''(0) = \Delta x^T \nabla^2 f(x) \Delta x \quad (1.12)$$

όπου:

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \dots & \frac{\partial^2 f(x)}{\partial x_n \partial x_1} \\ \dots & \dots & \dots \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix} \quad (1.13)$$

Η έκφραση (1.13) ονομάζεται Hessian μήτρα , ενώ με αντίστοιχο τρόπο ορίζεται και η γενική παράγωγος $g^{(n)}(0)$. [7]

Αναγκαίες συνθήκες για την ύπαρξη τοπικού ελαχίστου

Χρησιμοποιώντας το ανάπτυγμα Taylor με πεπερασμένο ανάπτυγμα δεύτερης παραγώγου, μπορεί να αποδειχθεί με άτοπο απαγωγή, ότι για να είναι το x^* τοπικό ελάχιστο πρέπει να ισχύει ότι:

$$f'(x^*) = 0$$

Η σχέση αυτή αποτελεί την 1^η αναγκαία συνθήκη για την ύπαρξη τοπικού ελαχίστου. Κάνοντας όμως χρήση του αναπτύγματος Taylor με πεπερασμένο ανάπτυγμα μέχρι παράγωγο τρίτης τάξεως, αποδεικνύεται ότι πρέπει να ισχύει:

$$f''(x^*) \geq 0$$

Η σχέση αυτή αποτελεί την δεύτερη αναγκαία συνθήκη για την ύπαρξη ελαχίστου. [8]

Ικανές συνθήκες για την ύπαρξη τοπικού ελαχίστου

Με παρόμοιο τρόπο ορίζονται οι ικανές συνθήκες για την ύπαρξη τοπικού ελαχίστου σε συναρτήσεις μιας μεταβλητής, όπως αναγράφονται ακολούθως:

- 1^η ικανή συνθήκη:

$$f'(x) = 0$$

- 2^η ικανή συνθήκη:

$$f''(x) \geq 0$$

Με ανάλογο τρόπο αποδεικνύεται και στην περίπτωση κατά την οποία ισχύει:

$$f''(x) = 0$$

Διαφορετικά, αν ισχύει ότι:

$$f'''(x) \neq 0$$

τότε έχουμε την ύπαρξη σημείου καμπής στο σημείο x .

Για την περίπτωση όπου $f'(x) = f''(x) = f'''(x) = 0$ και $f^{iv}(x) > 0$, τότε το x είναι τοπικό ελάχιστο. Στην περίπτωση όμως που $f^{iv}(x) = 0$, τότε θα πρέπει να διερευνηθούν οι παράγωγοι υψηλότερου βαθμού. [8]

Τότε η πρώτη μη μηδενική παράγωγος πρέπει να είναι άρτιας τάξης παράγωγος, η οποία θα πρέπει να είναι θετική ώστε να υπάρχει ελάχιστο. Οι συνθήκες αυτές, ισχύουν για τιμές του x στο διάστημα όπου η $f(x)$ είναι συνεχής και έχει συνεχείς παραγώγους. Με τον τρόπο αυτό οι ικανές συνθήκες μπορούν να διατυπωθούν ως εξής:

$$f'(x) = f''(x) = \dots = f^{(2k-1)}(x) = 0$$

$$f^{(2k)}(x) > 0, \text{ όπου } k \geq 1, \text{ ακέραιος}$$

Στην περίπτωση όπου ισχύει $f^{(2k)}(x) = 0$ και $f^{(2k+1)}(x) \neq 0$ τότε το σημείο x είναι σημείο καμπής της συνάρτησης. [8]

Πίνακες

Ένας πίνακας είναι μια ορθογώνια διάταξη στοιχείων της μορφής:

$$\begin{pmatrix} a_{11} & a_{12} \cdots & a_{1n} \\ a_{21} & a_{21} \cdots & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} \cdots & a_{mn} \end{pmatrix}$$

Ο παραπάνω πίνακας αποτελείται από m γραμμές και n στήλες, ενώ περιγράφεται ως ένας πίνακας με $m \times n$ διαστάσεις. Ο συμβολισμός των στοιχείων του πίνακα παριστάνεται ως a_{ij} όπου το i συμβολίζει τη γραμμή του πίνακα και το j τη στήλη του πίνακα. Οι πίνακες συμβολίζονται με κεφαλαία γράμματα: A, B κτλ., ενώ τα στοιχεία του πίνακα με μικρά γράμματα: a, b κτλ. [9]

Δύο πίνακες $A=(a_{ij})$ και $B=(b_{ij})$ θα είναι ίσοι και θα γράφεται ως $A=B$, εάν οι πίνακες είναι ίδιων διαστάσεων και τα αντίστοιχα στοιχεία τους είναι ίσα. Δηλαδή:

$$a_{ij} = b_{ij} \text{ , για όλα τα } i, j$$

Για παράδειγμα οι πίνακες που ακολουθούν είναι διαστάσεων 2×3 :

$$A = \begin{pmatrix} x & y & z \\ a & b & c \end{pmatrix} \quad B = \begin{pmatrix} 2 & 0 & -1 \\ 5 & 17 & \sqrt{2} \end{pmatrix}$$

Το στοιχείο a_{12} πίνακα A είναι το y , ενώ το στοιχείο a_{12} είναι το a . Η ισότητα $A=B$ των δύο πινάκων ανάγεται ουσιαστικά σε έξι εξισώσεις:

$$x=2 \quad y=0 \quad z=-1$$

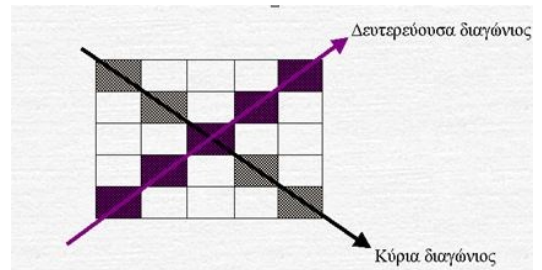
$$a=5 \quad b=17 \quad c=\sqrt{2}$$

Επίσης, ένας πίνακας $1 \times n$ έχει τη μορφή:

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

και ονομάζεται πίνακας-στήλη. Επιπρόσθετα, υπάρχει το ενδεχόμενο να έχουμε έναν πίνακα-στοιχείο, δηλαδή έναν 1x1 πίνακα, όπως για παράδειγμα οι πίνακες (8), (-6) κτλ. [9],[10].

Οι πίνακες που έχουν τον ίδιο αριθμό γραμμών και στηλών ονομάζονται τετραγωνικοί πίνακες, ενώ χαρακτηρίζονται από την κύρια και δευτερεύουσα διαγώνιο. Κύρια διαγώνιος είναι η διαγώνιος του πίνακα που ξεκινάει από επάνω αριστερά του πίνακα και καταλήγει κάτω δεξιά. Για να ανήκει ένα στοιχείο του πίνακα στην κύρια διαγώνιο θα πρέπει ο δείκτης της γραμμής και της στήλης να είναι ίδιος. Ενώ, η δευτερεύουσα διαγώνιος ξεκινάει από πάνω δεξιά προς τα κάτω αριστερά και ισχύει ότι το άθροισμα της γραμμής και της στήλης του πίνακα θα πρέπει να είναι ίσο με το $n+1$, όπου n η διάσταση του πίνακα.



Πράξεις πινάκων

- Πρόσθεση πινάκων

Δύο πίνακες A και B μπορούν να προστεθούν, αρκεί να έχουν τις ίδιες διαστάσεις. Δηλαδή, προσθέτουμε τα αντίστοιχα στοιχεία και το αποτέλεσμα που προκύπτει έχει εξίσου τις ίδιες διαστάσεις και συμβολίζονται ως $A+B$ [9]. Συνοπτικά:

Για τους $m \times n$ πίνακες $A=(a_{ij})$ και $B=(b_{ij})$, ορίζεται:

$$A+B = (c_{ij}) \text{ όπου } c_{ij} = a_{ij} + b_{ij}, \text{ για όλα τα } i,j.$$

Για παράδειγμα:

Έστω οι ακόλουθοι δύο πίνακες:

$$A = \begin{pmatrix} 1 & 2 \\ 0 & -5 \\ 3 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 3 \\ 2 & 5 \\ 0 & -4 \end{pmatrix}$$

Εφαρμόζοντας την πράξη της πρόσθεσης προκύπτει ότι:

$$\begin{pmatrix} 1 & 2 \\ 0 & -5 \\ 3 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 3 \\ 2 & 5 \\ 0 & -4 \end{pmatrix} = \begin{pmatrix} 2 & 5 \\ 2 & 0 \\ 3 & -3 \end{pmatrix} \quad [10,11]$$

Με τον ίδιο τρόπο πραγματοποιείται και η αφαίρεση πινάκων, με τη διαφορά ότι υλοποιείται η πράξη της αφαίρεσης.

- Βαθμωτό Γινόμενο: (Πολλαπλασιασμός ενός αριθμού με πίνακα)

Ένας αριθμός λ μπορεί να πολλαπλασιαστεί με έναν πίνακα A , πολλαπλασιάζοντας απλά τον αριθμό λ με κάθε στοιχείο του πίνακα. Το αποτέλεσμα έχει τις ίδιες διαστάσεις με τον A και συμβολίζεται ως: λA . Δηλαδή:

Εάν $\lambda \in \mathbb{R}$ και A είναι ένας $m \times n$ πίνακας $A = (a_{ij})$, τότε ορίζουμε $\lambda A = (c_{ij})$,

όπου: $c_{ij} = \lambda a_{ij}$, για όλα τα i, j [9].

Επίσης, συμβολίζουμε με $-A$ τον πίνακα $(-1)A$, οπότε μπορούμε να μιλάμε και για αφαίρεση πινάκων αν ορίσουμε:

$$A - B = A + (-B)$$

Για παράδειγμα, αν έχουμε τους προηγούμενους πίνακες A και B θα προκύψει:

$$5A = 5 \begin{pmatrix} 1 & 2 \\ 0 & -5 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 10 \\ 0 & -25 \\ 15 & 5 \end{pmatrix}$$

$$-B = -1B \begin{pmatrix} 1 & 3 \\ 2 & 5 \\ 0 & -4 \end{pmatrix} = \begin{pmatrix} -1 & -3 \\ -2 & -5 \\ 0 & 4 \end{pmatrix}$$

Ενώ:

$$5A - 2B \begin{pmatrix} 5 & 10 \\ 0 & -25 \\ 15 & 5 \end{pmatrix} - \begin{pmatrix} 2 & 6 \\ 4 & 10 \\ 0 & -8 \end{pmatrix} = \begin{pmatrix} 3 & 4 \\ -4 & -35 \\ 15 & 13 \end{pmatrix} \quad [9,10,11].$$

Μηδενικός πίνακας: οι πίνακες με $m \times n$ διαστάσεις οι οποίοι έχουν όλα τα στοιχεία τους μηδενικά. Ο μηδενικός πίνακας συμπεριφέρεται όπως και το μηδέν στους αριθμούς, δηλαδή:

$$A+O=A=O+A.$$

Η γενική μορφή $A=(a_{ij})$ βοηθάει να αποδεικνύουμε τέτοιου είδους ιδιότητες καθώς μας επιτρέπει να «περνάμε» από πράξεις πινάκων σε πράξεις αριθμών. Για το πρώτο σκέλος της παραπάνω ιδιότητας ισχύει [9]:

$$A+O = (a_{ij} + 0) \text{ [από τον ορισμό του αθροίσματος πινάκων]}$$

$$= (a_{ij}) \quad \text{[από την ιδιότητα του αριθμού 0]}$$

$$= A$$

Κατ' αυτό το τρόπο αποδεικνύονται και οι επόμενες ιδιότητες πράξεων, για οποιονδήποτε $m \times n$ πίνακα A, B, C και αριθμούς $\lambda, \mu \in \mathbb{R}$.

- i. $(A+B)+C = A+(B+C)$ [προσεταιριστική ιδιότητα]
- ii. $A+O = A$ [ύπαρξη ουδέτερου στοιχείου]
- iii. $A+(-A) = O$ [ύπαρξη αντίθετου στοιχείου]
- iv. $A+B = B+A$ [αντιμεταθετική ιδιότητα]
- v. $\lambda(A+B) = \lambda A + \lambda B$ [επιμεριστική ιδιότητα]
- vi. $(\lambda+\mu)A = \lambda A + \mu A$ [επιμεριστική ιδιότητα]
- vii. $(\lambda\mu)A = \lambda(\mu A)$
- viii. $1A = A$ [9].

Μοναδιαίος πίνακας: Όλα τα στοιχεία της κύριας διαγωνίου είναι ίσα με ένα και όλα τα υπόλοιπα είναι ίσα με μηδέν. Για παράδειγμα:

$$I_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

- Πολλαπλασιασμός Πινάκων

Το γινόμενο ενός πίνακα-γραμμή με έναν πίνακα-στήλη, ορίζεται όπως το εσωτερικό γινόμενο:

$$(a_1 \ a_2 \ \dots \ a_s) \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{pmatrix} = a_1 b_1 + a_2 b_2 + \dots + a_s b_s = \sum_{k=1}^s a_k b_k$$

Προκειμένου να γίνει ο παραπάνω πολλαπλασιασμός, θα πρέπει και οι δύο αρχικοί πίνακες να έχουν τον ίδιο αριθμό στοιχείων. Αλλιώς δεν γίνεται αυτός ο πολλαπλασιασμός.

Για να επιτευχθεί πολλαπλασιασμός μεταξύ δύο πινάκων A και B, θα πρέπει όσες στήλες έχει ο πίνακας A τόσες γραμμές να έχει και ο πίνακας B. Επίσης, για να βρεθεί το στοιχείο c_{ij} του γινομένου AB, πολλαπλασιάζεται η i -γραμμή του πίνακα A με την j -στήλη του πίνακα B [9,10,11].

Σύμφωνα με τα προαναφερόμενα ισχύει ότι:

Εάν $A=(a_{ij})$ είναι ένας $m \times s$ πίνακας και $B=(b_{ij})$ είναι ένας $s \times n$ πίνακας, τότε ορίζουμε ως γινόμενό τους τον $m \times n$ πίνακα :

$$AB = (c_{ij}), \text{ όπου } c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{is}b_{sj} = \sum_{k=1}^s a_{ik} b_{kj}, \text{ για όλα τα } i, j \text{ [9]}$$

Για παράδειγμα:

$$\begin{bmatrix} \alpha & \beta & \gamma \\ \delta & \varepsilon & \zeta \end{bmatrix} \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} =$$

$$\begin{bmatrix} \alpha_1 + \beta_2 + \gamma_3 & \alpha_4 + \beta_5 + \gamma_6 & \alpha_7 + \beta_8 + \gamma_9 & \alpha_{10} + \beta_{11} + \gamma_{12} \\ \delta_1 + \varepsilon_2 + \zeta_3 & \delta_4 + \varepsilon_5 + \zeta_6 & \delta_7 + \varepsilon_8 + \zeta_9 & \delta_{10} + \varepsilon_{11} + \zeta_{12} \end{bmatrix}$$

Στο παραπάνω παράδειγμα, πολλαπλασιάζεται η πρώτη γραμμή ($\alpha \beta \gamma$) του πίνακα A διαδοχικά και με τις τέσσερις στήλες του πίνακα B. Έτσι προκύπτει η πρώτη γραμμή του πίνακα AB. Στη συνέχεια επαναλαμβάνεται η ίδια διαδικασία με τη δεύτερη γραμμή ($\delta \varepsilon \zeta$). Οπότε, για να βρεθεί το στοιχείο (2,3) του AB πολλαπλασιάζεται η 2^η γραμμή του πίνακα A με την 3^η στήλη του πίνακα B [9].

Ανάστροφος πίνακας: Ο ανάστροφος ενός πίνακα A, ο οποίος συμβολίζεται με A^T , προκύπτει εάν διαταχθούν οι γραμμές του πίνακα A ως στήλες με την ίδια σειρά (οπότε αυτόματα οι στήλες γίνονται γραμμές). Έτσι το στοιχείο που βρίσκεται στη θέση ij, στον ανάστροφο θα πάρει τη θέση ji [9,10,11]. Για παράδειγμα:

$$A = \begin{bmatrix} 7 & 3 & 4 \\ 1 & 5 & 6 \end{bmatrix} \rightarrow B = \begin{bmatrix} 7 & 1 \\ 3 & 5 \\ 4 & 6 \end{bmatrix}$$

1.2 Γραμμική Αναζήτηση

Η γραμμική αναζήτηση αποτελεί στοιχειώδη στοιχείο στις μεθόδους ελαχιστοποίησης. Υπάρχει μεγάλο πλήθος μεθόδων ελαχιστοποίησης με γραμμική αναζήτηση, που χρησιμοποιούν όμως διαφορετικούς τρόπους ανεύρεσης της διεύθυνσης αναζήτησης. Πιο συγκεκριμένα, κατά την γραμμική αναζήτηση δίνεται ένα σημείο εκκίνησης $x^{(0)}$, ελέγχονται τα κριτήρια τερματισμού και ορίζεται μία διεύθυνση αναζήτησης $s^{(k)}$, ώστε να πραγματοποιηθεί γραμμική αναζήτηση στη διεύθυνση $s^{(k)}$. Πραγματοποιώντας ελαχιστοποίηση ως προς $a^{(k)}$ της $\varphi(a^{(k)}) = f(x^{(k)} + a^{(k)}s^{(k)})$, καθορίζεται το νέο σημείο ως: $x^{(k+1)} = x^{(k)} + a^{(k)}s^{(k)}$ [6]. Επομένως με βάση τη διεύθυνση αναζήτησης s, ζητείται

να βρεθεί a , τέτοιο ώστε η τιμή της συνάρτησης $f(x+as)$ να είναι η ελάχιστη. Εφόσον τα x και s είναι σταθερά, γράφουμε:

$$\varphi(a) = f(x + as)$$

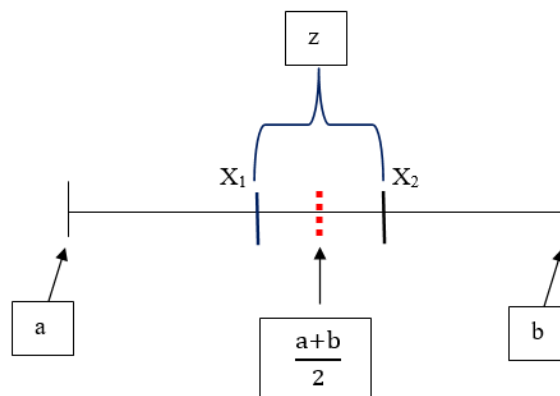
Όπου η παράγωγος $\varphi'(a) = \frac{d\varphi(a)}{da} = s^T g(x + as)$.

Υπάρχουν αρκετοί μέθοδοι αναζήτησης που χρησιμοποιούν είτε τιμές συνάρτησης και των παραγώγων τους είτε μέθοδοι που χρησιμοποιούν μόνο τις τιμές συναρτήσεων, όπως για παράδειγμα η μέθοδος της διχοτόμησης, η χρυσής τομής, η Fibonacci και η Armijo [6].

Η μέθοδος της διχοτόμησης

Με την αναζήτηση της διχοτόμησης, αν υποθέσουμε ότι z είναι μία θετική ποσότητα τότε επιλέγουμε τα εσωτερικά σημεία της z ως εξής:

$$X_1 = \frac{1}{2} - \frac{z}{2} \quad \text{και} \quad X_2 = \frac{1}{2} + \frac{z}{2}$$



Εικόνα 1 7: Η επιλογή των εσωτερικών σημείων στην αναζήτηση διχοτόμησης

Όπως απεικονίζεται και από το παραπάνω σχήμα, με τη μέθοδο διχοτόμησης γίνεται προσπάθεια να χωριστούν τα διαστήματα στη μέση.

Η μέθοδος της χρυσής τομής

Με τη μέθοδο της χρυσής τομής, τα σημεία x_1 και x_2 , με $x_1 < x_2$, ισαπέχουν από τα άκρα a και b , δηλαδή: $x_1 - a = b - x_2$. Προφανώς τα επόμενα διαστήματα θα είναι $[a, x_2]$ εάν $f_2 > f_1$

και $[x_1, b]$ εάν $f_1 > f_2$, με εύρος $d_1 = x_2 - a = b - x_1$. Αν το νέο διάστημα είναι το $[a, x_2]$, τότε το x_1 θα χρησιμοποιηθεί ως ενδιάμεσο σημείο ώστε να αποφευχθεί η εκ νέου αποτίμηση της συνάρτησης. Σε αυτή την περίπτωση τα x_1, x_3 θεωρούνται ενδιάμεσα στοιχεία με $a < x_3 < x_1 < x_2$. Με τον ίδιο τρόπο θεωρούμε τις ακόλουθες τέσσερις τετράδες διαδοχικά σημεία [12]. Αναλυτικά:

- $[a, x_1, x_2, b]$ με εύρος $d = b - a$ και $b - x_2 = x_1 - a \equiv w$
- $[a, x_3, x_1, x_2]$ με εύρος $d_1 = x_2 - a = d - w$ και με $x_3 - a = x_2 - x_1 = d - 2w$ ή $x_3 = b - 2w$
- $[a, x_4, x_3, x_1]$, με εύρος $d_2 = w$ και με $x_4 - a = x_1 - x_3 = 3w - d$ ή $x_4 = a - d + 3w$
- $[a, x_5, x_4, x_3]$, με εύρος $d_3 = d - 2w$ και με $x_5 - a = x_3 - x_4$ ή $x_5 = b + d - 5w$

Με βάση τα παραπάνω ισχύει ότι: $d_1 + d_2 = x_2 - a + w = b - w - a + w = d$ και $d_1 = d_2 + d_3$.

Επαγωγικά σημαίνει ότι: $d_k = d_{k+1} + d_{k+2}$. [12]

Στην περίπτωση που ο λόγος αναγωγής του διαστήματος αβεβαιότητας r πρέπει να είναι σταθερός, δηλαδή :

$$d_1 = rd,$$

$$d_2 = r d_1 = r^2 d,$$

$$d_3 = r d_2 = r^3 d, \dots$$

λόγω της σχέσης $d_k = d_{k+1} + d_{k+2}$, προκύπτει ότι: $r^2 + r - 1 = 0$ με $r > 0$.

Συνεπάγεται ότι: $r = \frac{\sqrt{5}-1}{2} \approx 0.618$. Ο λόγος του r είναι ο λόγος «Χρυσής τομής».

Η μέθοδος Fibonacci

Η μέθοδος Fibonacci διαφέρει από τη μέθοδο της «Χρυσής τομής» για δύο λόγους. Αρχικά, ο λόγος αναγωγής του διαστήματος αβεβαιότητας δεν είναι σταθερός, αλλά εξαρτάται από τον δείκτη της επανάληψης. Ενώ ο αριθμός των επαναλήψεων, πρέπει να είναι γνωστός από την αρχή. Η ακολουθία αριθμών στην οποία κάθε αριθμός είναι ίσος με το άθροισμα των δύο προηγούμενων είναι γνωστή ως ακολουθία Fibonacci [12,13]. Η ακολουθία αυτή ξεκινάει με τους αριθμούς: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181 και συνεχίζεται επ άοριστο. Για παράδειγμα:

1+1=2	13+21=34
1+2=3	21+34=55
2+3=5	34+55=89
3+5=8	55+89=144
5+8=13	89+144=233
8+13=21	144+233=377

Παρατηρούμε επομένως, ότι κάθε αριθμός είναι ίσος με το άθροισμα των δύο προηγούμενων. Με μαθηματικούς όρους, η ακολουθία F_n των αριθμών Fibonacci ορίζεται από τον αναδρομικό τύπο:

$$F_{(n)}=F_{(n-1)}+F_{(n-2)}, \text{ με } F_0 = 0 \text{ και } F_1 = 1$$

Ο τύπος αυτός αναλύεται και οι αριθμοί Fibonacci δίνονται από την κλειστή μορφή:

$$F_{(n)} = \frac{\varphi^n - \psi^n}{\varphi - \psi} = \frac{\varphi^n - \psi^n}{\sqrt{5}}, \text{ όπου } \varphi = \frac{1+\sqrt{5}}{2} \approx 1.6180 \dots$$

Παρατηρούμε ότι ο λόγος αναγωγής της Fibonacci προσεγγίζει τον σταθερό λόγο της «Χρυσής τομής». [12,13].

Η μέθοδος Armijo

Για τον τερματισμό της γραμμικής αναζήτησης, αναπτύχθηκε ένα σύνολο από συνθήκες, το οποίο διασφαλίζει τη μείωση της συνάρτησης f με όσο το δυνατό μικρό κόστος αλλά και την γενική σύγκλιση. Ο σκοπός της γραμμικής αναζήτησης είναι η εύρεση ενός βήματος a_k το οποίο δεν πλησιάζει την μικρότερη δυνατή θετική τιμή του a_k . Εάν υποθέσουμε ότι το \bar{a}_k είναι η μικρότερη θετική τιμή του a_k για την οποία ισχύει: $f(x_k + a_k p_k) = f(x_k)$ τότε υπάρχει μία μείωση της f αν $a_k \rightarrow \bar{a}_k$. Μία συνθήκη μη ακριβούς γραμμικής αναζήτησης δείχνει ότι το a_k θα πρέπει να οδηγεί σε μείωση της αντικειμενικής συνάρτησης f η οποία υπολογίζεται από την ανισότητα:

$$\varphi_{(a)} = f_{(x_k+a_k p_k)} < f_{(x_k)} + c_1 a_k \nabla f_k^T p_k, \text{ όπου } c_1 \in (0,1)$$

Επομένως, η μείωση της αντικειμενικής συνάρτησης f θα πρέπει να είναι ανάλογη του μήκους βήματος a_k όσο και με την κατεύθυνση παράγωγο $\nabla f_k^T p_k$.

Η παραπάνω ανισότητα ονομάζεται συνθήκη Armijo, όπου το δεξί της μέλος είναι μια γραμμική συνάρτηση του a και συμβολίζεται με $l_{(a)}$. Έχει αρνητική κλίση $c_1 \nabla f_k^T p_k$, αφού $c_1 \in (0,1)$ και εκτείνεται πάνω από το γράφημα της φ για μικρές θετικές τιμές του a . Η συνθήκη επαρκούς μειώσεως υποδηλώνει ότι το a γίνεται αποδεκτό μόνο εάν $\varphi_{(a)} \leq l_{(a)}$ [14].

1.3 Τοπική Βελτιστοποίηση

Σε πολλούς κλάδους της επιστήμης, τα προβλήματα που τίθενται προς επίλυση, προσπαθούν να διατυπωθούν βρίσκοντας λύση μέσω της εύρεσης του ελαχίστου μιας συνάρτησης πολλών μεταβλητών. Ο τομέας της βελτιστοποίησης έχει ως αντικείμενο την ανάπτυξη μεθόδων για τον εντοπισμό των ελαχίστων μιας συνάρτησης πολλών μεταβλητών. Υπάρχουν διάφορες κατηγορίες προβλημάτων βελτιστοποίησης όπως: συνεχή προβλήματα, διακριτά προβλήματα και μικτά προβλήματα, ενώ καθεμία από αυτές τις κατηγορίες επεκτείνεται σε υποκατηγορίες [15]. Η μαθηματική έκφραση ενός προβλήματος βελτιστοποίησης είναι:

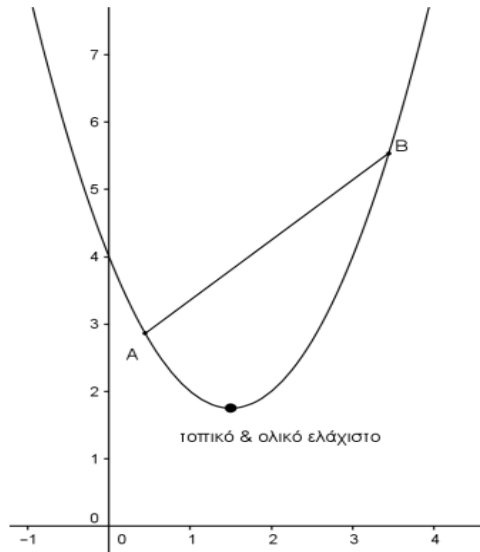
$$\min_x f(x), \text{ όπου } x \in R^n$$

υπό τους περιορισμούς:

- $g_i(x) \leq 0, i=1,..m$
- $h_j(x) = 0, j=1,..,n$ [16].

Το πρόβλημα της τοπικής βελτιστοποίησης είναι η εύρεση ενός σημείου τοπικού ελαχιστοποιητή ή τοπικού μεγιστοποιητή, τέτοιο ώστε για κάθε άλλο διάνυσμα x : έστω S (όπου $S \subset R^n$) που βρίσκεται εντός μίας περιοχής γύρω από το x^* διάνυσμα (όπου x^* παίρνει τιμές στον χώρο R^n), να ισχύει:

$$f(x^*) \leq f(x) \forall x \in S$$



Εικόνα 1 8: Τοπική βελτιστοποίηση: $f(x^*) \leq f(x) \forall x \in S$

Γενικότερα, μία μέθοδο τοπικής βελτιστοποίησης παράγει αποτελέσματα που εξαρτώνται από ένα αρχικό σημείο. Οι περισσότερες μέθοδοι τοπικής βελτιστοποίησης βασίζονται στη λογική της κλίσης (gradient-based), οι οποίες εκμεταλλεύονται τις πληροφορίες κλίσης για να βρουν τη βέλτιστη λύση. Οι μέθοδοι τοπικής βελτιστοποίησης, τοπική και καθολική, διακρίνονται τόσο από πλεονεκτήματα όσο και από μειονεκτήματα. Ακολούθως παρουσιάζονται τα βασικότερα πλεονεκτήματα και μειονεκτήματα της μεθόδου:

Πλεονεκτήματα:

- Αποδοτική μέθοδος για τον εντοπισμό εύρεσης ακροτάτου
- Αντιμετώπιση προβλημάτων μεγάλης κλίμακας
- Απαιτούν μικρή ρύθμιση παραμέτρων στο πρόβλημα
- Υψηλή ταχύτητα σύγκλισης

Μειονεκτήματα:

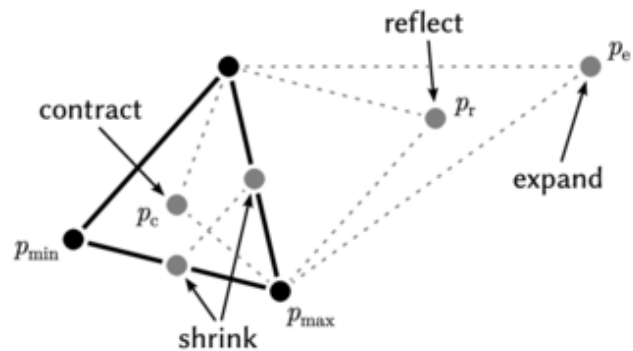
- Εντοπίζουν μόνο το τοπικό ακρότατο και όχι το ολικό βέλτιστο
- Δεν παρέχουν πληροφορία σχετικά με το πόσο κοντά στο ολικό βέλτιστο βρίσκεται η τοπικά βέλτιστη λύση
- Δυσκολεύονται να λύσουν διακριτά προβλήματα βελτιστοποίησης

- Περίπλοκοι αλγόριθμοι
- Ανεπαρκείς όταν υπάρχουν ασυνέχειες στην επιφάνεια στην οποία υπολογίζεται η κλίση [17].

Μέθοδοι χωρίς παραγώγους

❖ Nelder Mead Method

Ο αλγόριθμος Nelder-Mead έχει σχεδιαστεί για να επιλύει προβλήματα βελτιστοποίησης, χωρίς περιορισμούς της ελαχιστοποίησης μιας δεδομένης μη γραμμικής συνάρτησης $f: R^n \rightarrow R$. Η μέθοδος χρησιμοποιεί μόνο τιμές που βρίσκονται στο R και δεν προσπαθεί να σχηματίσει μια κατά προσέγγιση κλίση σε κάποιο σημείο από αυτά. Γι' αυτό άλλωστε ανήκει στη γενική κατηγορία μεθόδων άμεσης αναζήτησης [18, 19, 20].



Το γεωμετρικό πρότυπο στο οποίο βασίζεται είναι το simplex. Ο όρος simplex δηλώνει ένα σύνολο $n+1$ μη

Εικόνα 1 9: Επανάληψη της μεθόδου Nelder-Mead σε δισδιάστατο χώρο

συν-επιπέδων σημείων στο n -διάστατο πεδίο αναζήτησης. Επομένως στο δισδιάστατο χώρο απεικονίζεται με τη βοήθεια ενός τριγώνου, στο τριών διαστάσεων με τη βοήθεια ενός τετράεδρου, ενώ σε χώρο περισσότερων διαστάσεων δεν είναι δυνατή η οπτικοποίηση [21].

Ο αλγόριθμος Never-Mead όταν χρησιμοποιείται για πρόβλημα ελαχιστοποίησης σε n διαστάσεις περιγράφεται ως εξής:

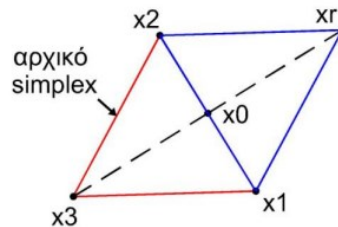
1. Υπολογισμός της αντικειμενικής συνάρτησης σε $n+1$ τυχαία σημεία
2. Ταξινόμηση των σημείων από το καλύτερο προς το χειρότερο ανάλογα με την τιμή της αντικειμενικής συνάρτησης του αντίστοιχου σημείου κάθε φορά:

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$$

3. Υπολογισμός του x_0 του κέντρου βάρους όλων των σημείων εκτός από το χειρότερο x_{n+1}
4. Ανάκληση: υπολογισμός του ανακλώμενου σημείου:

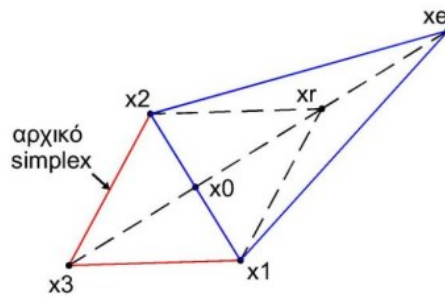
$$x_r = x_0 + \alpha(x_0 - x_{n+1})$$

Εάν το ανακλώμενο σημείο είναι καλύτερο από το δεύτερο χειρότερο αλλά δεν είναι καλύτερο από το πιο καλύτερο, δηλαδή: $f(x_1) \leq f(x_r) \leq f(x_n)$, τότε δημιουργείται ένα καινούριο simplex όπου αντικαθίσταται το χειρότερο σημείο x_{n+1} , με το ανακλώμενο σημείο x_r και το πρόβλημα επιστρέφει στο βήμα 2 [21].



Εικόνα 1 10: Ανάκληση σε δύο διαστάσεις

5. Διαστολή: εάν το ανακλώμενο σημείο είναι το καλύτερο έως τώρα δηλαδή: $f(x_r) < f(x_1)$, τότε υπολογίζεται το σημείο διαστολής $x_e = x_0 + \gamma(x_0 - x_{n-1})$. Στην περίπτωση που το σημείο διαστολής είναι καλύτερο από το ανακλώμενο σημείο, δηλαδή: $f(x_e) < f(x_r)$, δημιουργείται ένα νέο simplex αντικαθιστώντας το χειρότερο σημείο x_{n+1} με σημείο διαστολής x_e και το πρόβλημα επιστρέφει στο βήμα 2, διαφορετικά συνεχίζει στο βήμα 6 [21].

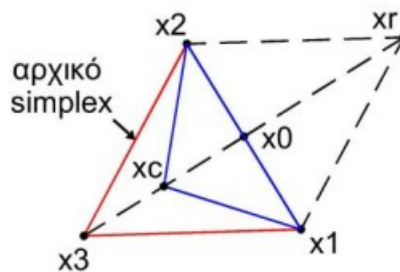


Εικόνα 1 11: Διαστολή σε δύο διαστάσεις

6. Συστολή: στο σημείο αυτό είναι βέβαιο ότι ισχύει $f(x_r) > f(x_n)$ και υπολογίζεται το σημείο συστολής ως εξής: $x_c = x_0 + \rho (x_0 - x_{n+1})$.

Εάν το σημείο συστολής είναι καλύτερο από το χειρότερο σημείο, δηλαδή ισχύει

$f(x_c) < f(x_{n+1})$, τότε δημιουργείται ένα νέο simplex αντικαθιστώντας το χειρότερο σημείο x_{n+1} με το σημείο συστολής x_c και το πρόβλημα επιστρέφει στο βήμα 2, διαφορετικά συνεχίζει στο βήμα 7 [21].

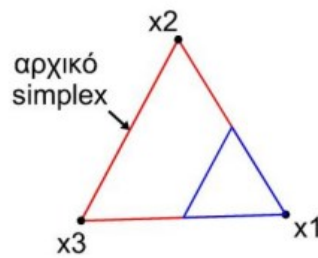


Εικόνα 1 12: Συστολή σε δύο διαστάσεις

7. Συρρίκνωση: Γίνεται αντικατάσταση όλων των σημείων στη σχέση:

$$x_i = x_1 + \sigma(x_i - x_1), \forall i \in \{2, \dots, n+1\}$$

εκτός από το καλύτερο σημείο και στη συνέχεια το πρόβλημα επιστρέφει στο βήμα 2.



Εικόνα 1 13: Συρρίκνωση σε δύο διαστάσεις

Οι συντελεστές α , γ , ρ και σ έχουν την εξής ερμηνεία:

α = συντελεστής ανάκλασης με τυπική τιμή 1,

γ = συντελεστής επέκτασης με τυπική τιμή 2,

ρ = συντελεστής συμπίεσης με τυπική τιμή $-1/2$ και

σ = συντελεστής συρρίκνωσης με τυπική τιμή $1/2$ [21].

Ο αλγόριθμος Nelder-Mead δεν είχε σχεδιαστεί για να μπορεί να χρησιμοποιηθεί σε προβλήματα με περιορισμούς. Για το λόγο αυτό προτάθηκαν μετέπειτα, διάφοροι τρόποι εισαγωγής περιορισμών στον αλγόριθμο. Οι περιορισμοί αυτοί παρουσιάζονται ακολούθως, όπου $c_{i,min}$ τα κατώτερα όρια του χώρου αναζήτησης και $c_{j,max}$ τα ανώτερα όρια αναζήτησης [21]:

- Εάν ένα σημείο βρίσκεται έξω από το χώρο αναζήτησης, τότε αντικαθίσταται από το σημείο $c_{i,min} + 10^{-6}$ στην περίπτωση που παραβιάζει το κάτω όριο ή $c_{j,max} - 10^{-6}$ στην περίπτωση που παραβιάζει το άνω όριο.
- Μία τροποποίηση του παραπάνω περιορισμού αναφέρει ότι εάν ένα σημείο είναι έξω από το χώρο αναζήτησης, αλλά το κέντρο βάρους του simplex x_0 είναι μέσα στο χώρο αναζήτησης, τότε το σημείο αντικαθίσταται από το $c_{i,min} + z(x_{0i} - c_{i,min})$ στην περίπτωση που παραβιάζεται το κάτω όριο. Αντίθετα, αν παραβιάζεται το άνω όριο τότε ισχύει $c_{i,min} - z(x_{0i} - c_{i,min})$. Ισχύει ότι το z είναι ένας τυχαίος αριθμός από το 10^{-6} μέχρι το 0.5. Στην περίπτωση που το κέντρο βάρους είναι έξω από το χώρο αναζήτησης τότε εφαρμόζεται η πρώτη μέθοδος.
- Εάν ένα σημείο είναι έξω από το χώρο αναζήτησης και το κέντρο βάρους του simplex x_0 είναι μέσα, τότε το σημείο αντικαθίσταται από τη μέση τιμή του σημείου και του κέντρου βάρους. Στην περίπτωση όπου και η μέση τιμή είναι έξω

από το χώρο αναζήτησης, τότε λαμβάνεται υπόψιν η «μέση τιμή» της μέσης τιμής και του κέντρου βάρους. Αυτή η διαδικασία επαναλαμβάνεται μέχρις ότου βρεθεί σημείο που δεν παραβιάζει τους περιορισμούς. Θα πρέπει να αναφερθεί ότι στην περίπτωση που το κέντρο βάρους είναι έξω από το χώρο αναζήτησης τότε εφαρμόζεται η πρώτη μέθοδος.

- Ένας άλλος περιορισμός που εφαρμόστηκε, είναι ο μετασχηματισμός της αντικειμενικής συνάρτησης σε μία νέα συνάρτηση εκθετικής μορφής, για τις περιπτώσεις όπου οι δοκιμές γίνονται έξω από τον χώρο αναζήτησης. Σε αυτή την περίπτωση, η νέα εκθετική μορφή της αντικειμενικής συνάρτησης, θα δώσει τις χειρότερες τιμές από την αντικειμενική συνάρτηση.
- Τέλος, η συνάρτηση ποινής, όπου η μορφή της γίνεται απλούστερη και χρησιμοποιείται για να παίρνει μεγάλες τιμές σε προβλήματα ελαχιστοποίησης, όταν ο αλγόριθμος κινείται εκτός του χώρου αναζήτησης.

❖ **Brent Method**

Η μέθοδος του Brent είναι ένας συνδυασμός της μεθόδου κανονικής ψευδούς ή ψευδούς θέσης και της μεθόδου διχοτόμησης. Αποτελεί μια οριοθετημένη μέθοδος, που συνήθως συγκλίνει ταχύτερα από τις μεθόδους διχοτόμησης και ψευδούς θέσης. Οι μέθοδοι διχοτόμησης και ψευδούς θέσης, δεν μπορούν να βρουν ρίζες που είναι τοπικά ελάχιστα ή μέγιστα, επειδή τα δεσμευμένα τελικά σημεία πρέπει να έχουν διαφορετικά πρόσημα. Κατ' αυτό το τρόπο η μέθοδος του Brent, ενσωματώνει μαζί με τις μεθόδους ψευδούς θέσης και διχοτόμησης, μία μέθοδο εύρεσης ρίζας αντίστροφης τετραγωνικής παρεμβολής. Η κύρια επιπλοκή του, προέρχεται από τους κανόνες που χρησιμοποιεί για να αποφασίσει ποια μέθοδο εύρεσης ρίζας θα χρησιμοποιήσει. Η μέθοδος λειτουργεί προσαρμόζοντας μια τετραγωνική συνάρτηση στο αντίστροφο της συνάρτησης, ενώ η ρίζα της συνάρτησης προσεγγίζεται ορίζοντας την είσοδο της αντίστροφης προσέγγισης στο μηδέν [22].

Αναφέρεται, ότι η μέθοδος Brent, συγκλίνει πάντα όσο οι τιμές της συνάρτησης είναι υπολογίσιμες σε μια δεδομένη περιοχή που περιέχει μια ρίζα. Δεδομένου των τριών σημείων x_1 , x_2 , x_3 η μέθοδος ταιριάζει με το x ως τετραγωνική συνάρτηση του y και χρησιμοποιώντας τον τύπο παρεμβολής [23]:

$$x = \frac{[y-f(x_1)][y-f(x_2)]x_3}{[f(x_3)-f(x_1)][f(x_3)-f(x_2)]} + \frac{[y-f(x_2)][y-f(x_3)]x_1}{[f(x_1)-f(x_2)][f(x_1)-f(x_3)]} + \frac{[y-f(x_3)][y-f(x_1)]x_2}{[f(x_2)-f(x_3)][f(x_2)-f(x_1)]}$$

Οι επόμενες εκτιμήσεις ρίζας, λαμβάνονται θέτοντας για $y=0$:

$$x = x_2 + \frac{P}{Q}$$

όπου: $P = S[T(R - T)(x_3 - x_2) - (1 - R)(x_2 - x_1)]$ και $Q = (T - 1)(R - 1)(S - 1)$

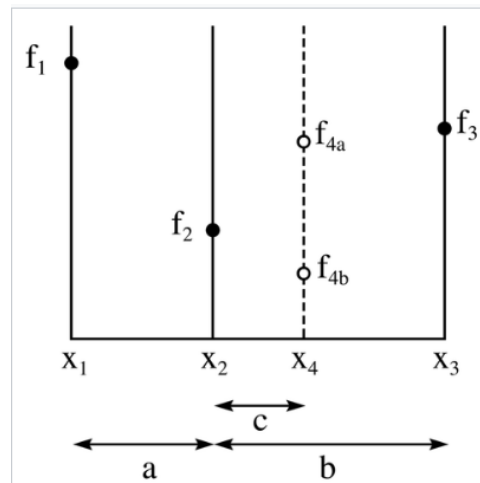
$$\text{με } R \equiv \frac{f(x_2)}{f(x_3)}, S \equiv \frac{f(x_2)}{f(x_1)} \text{ και } T \equiv \frac{f(x_1)}{f(x_3)} \text{ [23].}$$

❖ Golden-section Method

Η αναζήτηση χρυσής τομής είναι μια τεχνική για την εύρεση ενός άκρου (ελάχιστο ή μέγιστο) μιας συνάρτησης μέσα σε ένα καθορισμένο διάστημα. Για μια αυστηρά μονοτροπική συνάρτηση με ένα άκρο μέσα στο διάστημα, θα βρει αυτό το άκρο, ενώ για ένα διάστημα που περιέχει πολλαπλά άκρα, θα συγκλίνει σε ένα από αυτά. Εάν το μόνο άκρο στο διάστημα βρίσκεται σε ένα όριο του διαστήματος, θα συγκλίνει σε αυτό το οριακό σημείο. Η μέθοδος λειτουργεί περιορίζοντας διαδοχικά το εύρος τιμών στο καθορισμένο διάστημα, γεγονός που την καθιστά σχετικά αργή, αλλά πολύ ισχυρή. Η τεχνική αντλεί το όνομά της από το γεγονός ότι ο αλγόριθμος διατηρεί τις τιμές συνάρτησης για τέσσερα σημεία των οποίων τα τρία πλάτη διαστημάτων είναι στην αναλογία $\phi:1:\phi$ όπου ϕ είναι η χρυσή τομή. Αυτές οι αναλογίες διατηρούνται για κάθε επανάληψη και είναι μέγιστα αποδοτικές. Εκτός από τα οριακά σημεία, όταν αναζητάτε ένα ελάχιστο, το κεντρικό σημείο είναι πάντα μικρότερο ή ίσο με τα εξωτερικά σημεία, διασφαλίζοντας ότι ένα ελάχιστο περιέχεται μεταξύ των εξωτερικών σημείων. Όταν αναζητείται το μέγιστο, τότε ισχύει το αντίστροφο [24]. Στην περίπτωση της εύρεσης ενός μηδενός, αρκούν δύο αξιολογήσεις συναρτήσεων με αντίθετο πρόσημο για την τοποθέτηση ρίζας, σε αντίθεση με την αναζήτηση ελαχίστου που απαιτούνται τρεις τιμές.

Η αναζήτηση χρυσής τομής είναι ένας τρόπος για την σταδιακή μείωση των διαστημάτων εντοπισμού του ελαχίστου. Θα πρέπει να αναφερθεί στο σημείο αυτό,

ότι ανεξάρτητα από τον αριθμό των σημείων που έχουν αξιολογηθεί, το ελάχιστο βρίσκεται στο διάστημα που ορίζεται από τα δύο σημεία, δίπλα στο σημείο με την μικρότερη τιμή που έχει αξιολογηθεί μέχρι εκείνη τη χρονική στιγμή. Στο παρακάτω διάγραμμα απεικονίζεται μόνο η εύρεση του ελαχίστου, όπου τα $f(x)$ βρίσκονται στον κατακόρυφο άξονα και ο οριζόντιος άξονας είναι η παράμετρος x .



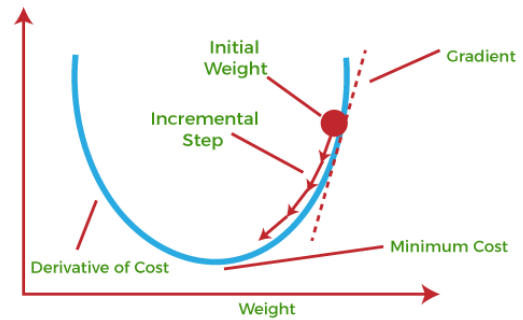
Εικόνα 1 14: Διάγραμμα αναζήτησης χρυσής τομής

Όπως διαπιστώνεται από το παραπάνω διάγραμμα, η $f(x)$ έχει αξιολογηθεί στα τρία σημεία: x_1 , x_2 και x_3 , όπου το $f(2)$ είναι μικρότερο από το $f(1)$ ή το $f(3)$. Οπότε, ένα ελάχιστο βρίσκεται ανάμεσα στο διάστημα από x_1 προς x_2 . Έπειτα, θα πρέπει να διερευνηθεί η συνάρτηση αξιολογώντας την σε μια νέα τιμή: x_4 . Επομένως εάν $f(x_4)=f_{4a}$, τότε ένα ελάχιστο εντοπίζεται μεταξύ x_1 και x_4 και η νέα τριάδα σημείων για την επόμενη αξιολόγηση, θα είναι: x_1 , x_2 και x_4 . Διαφορετικά, αν $f(x_4)=f_{4b}$, επιλέγεται η τριάδα x_2 , x_4 και x_3 . Με αυτό το τρόπο δημιουργείται ένα νέο στενότερο διάστημα αναζήτησης που περιέχει το ελάχιστο της συνάρτησης.

Μέθοδοι με παραγώγους

❖ Gradient Descent Method

Ο Gradient descent είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται κατά κύριο λόγο για την εκπαίδευση μοντέλων μηχανικής μάθησης και νευρωνικών δικτύων. Τα δεδομένα εκπαίδευσης βοηθούν τα μοντέλα μηχανικής μάθησης να εκπαιδευτούν, ενώ η συνάρτηση κόστους μετρά σε κάθε επανάληψη την ακρίβειά της για



να ενημερώνει τις παραμέτρους. Μέχρι η συνάρτηση να πλησιάσει ή να γίνει ίση με το μηδέν, το μοντέλο μηχανικής μάθησης θα συνεχίσει να προσαρμόζει τις παραμέτρους της μέχρι να αποφέρει το μικρότερο δυνατό σφάλμα. Ο καλύτερος τρόπος για να οριστεί το τοπικό ελάχιστο μιας συνάρτησης, είναι να κινηθούμε προς μία αρνητική κλίση ή να απομακρυνθούμε από τη διαβάθμιση της συνάρτησης στο τρέχον σημείο, ώστε να δώσει το τοπικό ελάχιστο της συνάρτησης. Ενώ, όταν κινούμαστε προς μια θετική κλίση ή προς τη διαβάθμιση της συνάρτησης στο τρέχον σημείο, θα λάβουμε το τοπικό μέγιστο αυτής της συνάρτησης. Προκειμένου να μάθουμε την κλίση μιας γραμμής από τη gradient descent, χρησιμοποιείται η εξίσωση: $Y=mX+c$, όπου το m αντιπροσωπεύει την κλίση της ευθείας και το c αντιπροσωπεύει τις τομές στον άξονα y . Από την παραπάνω εικόνα, παρατηρείται ένα σημείο εκκίνησης, το οποίο χρησιμοποιείται για την αξιολόγηση της απόδοσης. Από αυτό το σημείο εκκίνησης θα εξαχθεί η πρώτη παράγωγος ή η κλίση και έπειτα θα χρησιμοποιήσουμε μια εφαπτόμενη γραμμή για να υπολογιστεί η απότομη κλίση της. Βέβαια η κλίση γίνεται πιο απότομη στο σημείο εκκίνησης, αλλά όταν δημιουργούνται νέοι παράμετροι τότε η απότομη κλίση μειώνεται σταδιακά και στο χαμηλότερο σημείο πλησιάζει το χαμηλότερο σημείο, το οποίο και ονομάζεται σημείο σύγκλισης. Ο βασικός στόχος της κλίσης καθόδου είναι να ελαχιστοποιήσει τη συνάρτηση κόστους ή το σφάλμα μεταξύ αναμενόμενου και πραγματικού [25].

Υπάρχουν τρεις τύποι αλγορίθμων εκμάθησης: ο Batch gradient descent, ο Stochastic gradient descent και ο Mini-batch gradient descent. Ο αλγόριθμος Batch gradient descent, αθροίζει το σφάλμα για κάθε σημείο σ' ένα σετ εκπαίδευσης, ενημερώνοντας το μοντέλο μόνο αφού αξιολογηθούν όλα τα παραδείγματα εκπαίδευσης. Αυτή η ομαδοποίηση παρέχει υπολογιστική απόδοση, όμως

χρειάζεται χρόνο επεξεργασίας για τα σύνολα δεδομένων εκπαίδευσης ώστε να αποθηκεύσει όλα τα δεδομένα στη μνήμη. Επίσης, παράγει λιγότερο θόρυβο και σταθερή σύγκλιση [25,26].

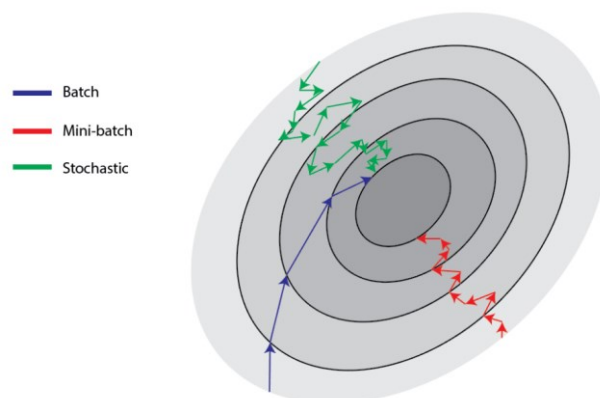
Ο αλγόριθμος Stochastic gradient descent, εκτελεί μια περίοδο εκπαίδευσης για κάθε περίπτωση μέσα στο σύνολο δεδομένων και ενημερώνει τις παραμέτρους κάθε περίπτωσης κάθε φορά. Αυτό προσφέρει περισσότερες λεπτομέρειες για το κάθε παράδειγμα εκπαίδευσης, όμως μπορεί να οδηγήσει σε απώλειες στην υπολογιστική απόδοση αλλά και σε θορυβώδεις κλίσεις. Είναι σχετικά γρήγορος στον υπολογισμό και πιο αποτελεσματικός για μεγάλα σύνολα δεδομένων [25,26].

Τέλος, ο αλγόριθμος Mini-batch gradient descent συνδυάζει στοιχεία και από τις δύο προηγούμενες μεθόδους. Συγκεκριμένα, διαχωρίζει το σύνολο των δεδομένων σε μικρά μεγέθη παρτίδων και εκτελεί ενημερώσεις σε κάθε μία από αυτές. Με αυτό το τρόπο επιτυγχάνεται μια ισορροπία μεταξύ της υπολογιστικής αποτελεσματικότητας της παρτίδας και της ταχύτητας της στοχαστικής κλίσης καθόδου. Γενικότερα είναι πιο αποδοτικός υπολογιστικά, ενώ παράγει μία σταθερή σύγκλιση κατάβασης [25,26].

❖ **Broyden Fletcher Goldfarb Shanno Method**

Ο αλγόριθμος Broyden Fletcher Goldfarb Shanno (BFGS), πήρε ο όνομά του από τα τέσσερα άτομα που τον ανακάλυψαν: Broyden, Fletcher, Goldfarb και Shanno. Είναι ένας από τους πιο ευρέως χρησιμοποιούμενους αλγορίθμους δεύτερης τάξης για αριθμητική βελτιστοποίηση και χρησιμοποιείται συνήθως για την προσαρμογή αλγορίθμων μηχανικής μάθησης [27]. Αποτελεί επέκταση του αλγορίθμου βελτιστοποίησης της μεθόδου Newton, η οποία χρησιμοποιεί πίνακα Hessian.

Το πρόβλημα βελτιστοποίησης είναι η ελαχιστοποίηση της $f(x)$, όπου x είναι ένα διάνυσμα στην R^n και η f είναι μια διαφορίσιμη βαθμωτή συνάρτηση. Δεν υπάρχουν περιορισμοί στις τιμές που μπορεί να πάρει το x . Ο



αλγόριθμος ξεκινά από μια αρχική εκτίμηση για τη βέλτιστη τιμή x_0 και προχωρά επαναληπτικά για να πάρει μια καλύτερη εκτίμηση σε κάθε στάδιο. Η κατεύθυνση αναζήτησης p_k στο στάδιο k δίνεται από τη λύση της ανάλογης εξίσωσης Newton:

$$B_k P_k = -\nabla f(x_k)$$

όπου:

- B_k : είναι μια προσέγγιση στον Εσσιανό πίνακα, ο οποίος ενημερώνεται επαναληπτικά σε κάθε στάδιο
- $\nabla f(x_k)$: είναι η παράγωγος της συνάρτησης που υπολογίζεται στο x_k .

Έπειτα χρησιμοποιείται μια αναζήτηση γραμμής προς την κατεύθυνση P_k για να βρεθεί το επόμενο σημείο x_{k+1} με ελαχιστοποίηση $f(x_k + \gamma P_k)$ με $\gamma > 0$.

Με βάση τη συνθήκη Newton, το B_k ενημερώνεται ως εξής:

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k)$$

Έχοντας $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ και $s_k = x_{k+1} - x_k$, τότε το B_{k+1} ικανοποιεί τη σχέση:

$$B_{k+1} s_k = y_k, \text{ όπου είναι και η εξίσωση τομής.}$$

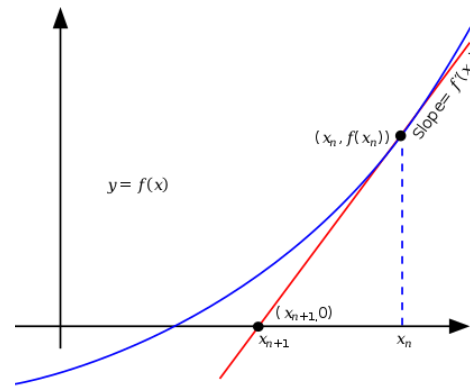
❖ **Newton Raphson Method**

Η μέθοδος Newton Raphson, είναι ένας αλγόριθμος εύρεσης ρίζας που παράγει διαδοχικά καλύτερες προσεγγίσεις στις ρίζες μιας συνάρτησης με πραγματική αξία. Ξεκινά με μια συνάρτηση μονής μεταβλητής f που ορίζεται για μια πραγματική μεταβλητή x , την παράγωγο της συνάρτησης f' και μια αρχική εικασία x_0 για μια ρίζα της f . Εάν η συνάρτηση ικανοποιεί επαρκείς υποθέσεις και η αρχική εικασία είναι κοντά, τότε:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

όπου είναι καλύτερη προσέγγιση της ρίζας από το x_0 .

Η κεντρική ιδέα είναι να ξεκινήσουμε με μια αρχική εικασία, στη συνέχεια να προσεγγίσουμε τη συνάρτηση με την εφαπτομένη της γραμμής και τέλος, να υπολογίσουμε την τέμνουσα x αυτής της εφαπτομένης γραμμής. Αυτή η τομή x θα είναι συνήθως μια καλύτερη προσέγγιση στη ρίζα της αρχικής συνάρτησης από την πρώτη εικασία και η μέθοδος μπορεί να επαναληφθεί. Εάν η εφαπτομένη στην καμπύλη $f(x) = x_n$, τέμνει τον άξονα x στο σημείο x_{n+1} , τότε η κλίση :



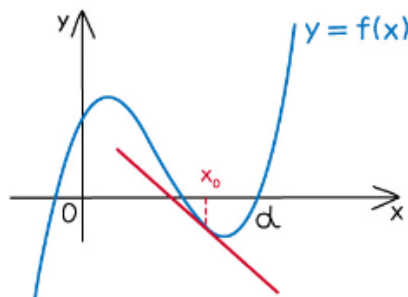
$$f'(x_n) = \frac{f(x_n) - 0}{x_n - x_{n+1}}, \text{ για } x_{n+1}$$

θα τροποποιηθεί ως εξής:

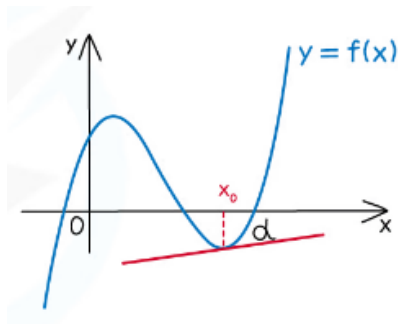
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Θα πρέπει να αναφερθεί, ότι η μέθοδος Newton-Raphson είναι δυνατό να αποτύχει σε ορισμένες περιπτώσεις. Αυτό μπορεί να συμβεί όταν:

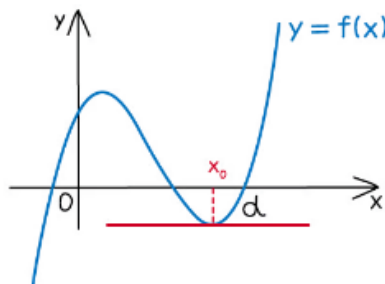
- Η αρχική τιμή x_0 είναι πολύ μακριά από τη ρίζα που οδηγεί σε μια αποκλίνουσα ακολουθία ή μια διαφορετική ρίζα [28].



- Η εφαπτόμενη κλίση είναι πολύ μικρή, όπου $f'(x)$ κοντά στο 0 οδηγεί σε μια αποκλίνουσα ακολουθία ή σε μια ακολουθία που συγκλίνει πολύ αργά [28].



- Η εφαπτόμενη είναι οριζόντια, όπου $f'(x)=0$ οπότε η εφαπτομένη δεν θα συναντήσει ποτέ τον άξονα x [28].



Γενικότερα, η μέθοδος Newton- Raphson τις περισσότερες φορές λειτουργεί εάν η αρχική εικασία είναι αρκετά καλή. Υπάρχουν όμως περιπτώσεις όπως αυτές που προαναφέρθηκαν, όπου μπορεί να αποτύχει. Βέβαια, αυτό μπορεί να αλλάξει αν απλώς αλλάξει η αρχική εικασία [28].

❖ Adam Method

Η μέθοδος Adam είναι ένας προσαρμοστικός αλγόριθμος βελτιστοποίησης ρυθμού εκμάθησης που χρησιμοποιεί την ορμή και την κλιμάκωση. Ο βελτιστοποιητής έχει σχεδιαστεί για να είναι κατάλληλος για μη σταθερούς στόχους και προβλήματα με πολύ θορυβώδεις ή αραιές κλίσεις. Πέρα από την αποθήκευση ενός εκθετικά μειούμενου μέσου όρου των προηγούμενων τετραγωνικών κλίσεων u_t , διατηρεί έναν εκθετικά μειούμενο μέσο όρο των προηγούμενων κλίσεων m_t . Για τον υπολογισμό αυτού του κινούμενου μέσου όρου, ισχύει ότι:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$u_t = \beta_2 u_{t-1} + (1 - \beta_2) g_t^2$$

όπου m_t και u_t είναι η μέση τιμή και η διακύμανση της κλίσης.

Όταν τα δύο αυτά στοιχεία αρχικοποιηθούν ως μηδενικά διανύσματα, τα β_1 και β_2 τείνουν στην μονάδα. Ισχύουν οι σχέσεις:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\widehat{u}_t = \frac{u_t}{1 - \beta_2^t}$$

Μέσα από τη χρήση αυτών των σχέσεων και την ενημέρωση των παραμέτρων, δίνεται ο κανόνας ενημέρωσης Adam ο οποίος αναφέρει ότι:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\widehat{u}_{t+\epsilon}}} * m_t$$

όπου: $\beta_1 = 0.9$, $\beta_2 = 0.999$ και το $\epsilon = 10^{-8}$ [29,30].

Η μέθοδος Adam, είναι ένας από τους καλύτερους βελτιστοποιητές σε σύγκριση με άλλους αλγόριθμους αλλά χαρακτηρίζεται τόσο από πλεονεκτήματα όσο και από μειονεκτήματα. Ορισμένα από τα πλεονεκτήματα είναι ότι ο αλγόριθμος μπορεί να χειριστεί αραιές διαβαθμίσεις σε θορυβώδη σύνολα δεδομένων, απαιτεί λίγη μνήμη οπότε είναι και πιο αποδοτικός, λειτουργεί πιο αποτελεσματικά σε μεγάλα σύνολα δεδομένων, ενώ παράλληλα οι προεπιλεγμένες τιμές παραμέτρων λειτουργούν πιο αποδοτικά. Σημαντικά μειονεκτήματα που έχουν διαπιστωθεί είναι ότι ο αλγόριθμος δεν συγκλίνει σε μία βέλτιστη λύση για κάποιους τομείς, υπάρχει το ενδεχόμενο μείωσης βάρους, ενώ αξίζει να σημειωθεί ότι πιο πρόσφατοι αλγόριθμοι έχουν αποδειχθεί πιο ταχύτεροι.

❖ RMSprop Method

Η μέθοδος RMSprop, λειτουργεί υπολογίζοντας την κλίση της συνάρτησης απώλειας σε σχέση με τις παραμέτρους του μοντέλου και ενημερώνει τις παραμέτρους στην αντίθετη κατεύθυνση της κλίσης για να ελαχιστοποιήσει την απώλεια. Ωστόσο, το RMSprop εισάγει μερικές πρόσθετες τεχνικές για τη βελτίωση της απόδοσης της διαδικασίας βελτιστοποίησης. Ένα βασικό

χαρακτηριστικό είναι η χρήση ενός κινούμενου μέσου όρου των τετραγωνικών κλίσεων για την κλίμακα του ρυθμού εκμάθησης για κάθε παράμετρο. Αυτό βοηθά στη σταθεροποίηση της διαδικασίας μάθησης και στην αποφυγή ταλαντώσεων στην τροχιά βελτιστοποίησης [31]. Ο RMSProp διατηρεί τον κινούμενο μέσο όρο των τετραγωνικών κλίσεων για κάθε βάρος και διαιρεί την κλίση με τετραγωνική ρίζα του μέσο τετράγωνου. Ισχύει η σχέση:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1 - \beta) \left(\frac{\delta C}{\delta W} \right)^2$$

$$W_{(t)} = W_{t-1} - \frac{\eta}{\sqrt{E[g^2]}} \frac{\delta C}{\delta W}$$

Όπου:

$E[g]$: ο μέσος όρος των τετραγωνικών κλίσεων

$\frac{\delta c}{\delta w}$: είναι η κλίση της συνάρτησης κόστους σε σχέση με το βάρος

η : ρυθμός μάθησης

Beta (β): μέσος όρος της κινούμενης παραμέτρου (προεπιλεγμένη τιμή το 0.9)

Η μέθοδος RMSProp είναι γνωστή για την γρήγορη ταχύτητα σύγκλισης και αυτό σημαίνει ότι μπορεί να βρει καλές λύσεις σε προβλήματα βελτιστοποίησης σε λιγότερες επαναλήψεις συγκριτικά με άλλους αλγόριθμους. Επίσης, η χρήση ενός κινητού μέσου όρου των τετραγωνικών κλίσεων στο RMSprop, βοηθά στη σταθεροποίηση της διαδικασίας εκμάθησης και στην αποφυγή ταλαντώσεων. Με αυτό το τρόπο η διαδικασία βελτιστοποίησης είναι λιγότερο επιρρεπής σε αποκλίσεις. Επιπλέον, έχει λιγότερες υπερ-παραμέτρους που διευκολύνουν τον συντονισμό και τη χρήση στην πράξη. Τέλος, το RMSprop αποδίδει καλά σε μη κυρτά προβλήματα που έχουν πολλαπλά τοπικά ελάχιστα και γρήγορη ταχύτητα σύγκλισης. Συμπερασματικά, το RMSprop είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος βελτιστοποίησης που μπορεί να είναι αποτελεσματικός για την εκπαίδευση μιας ποικιλίας μοντέλων [32].

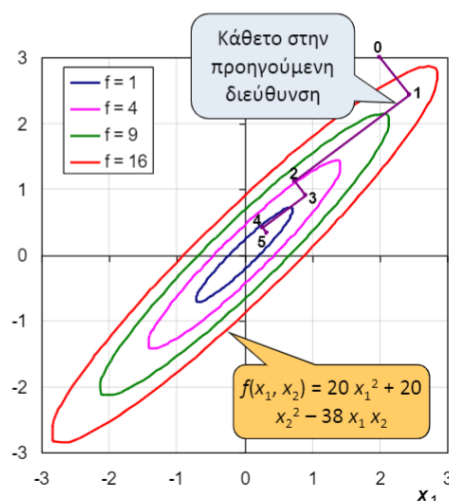
Τεχνικές τοπικής βελτιστοποίησης

Οι τεχνικές αναζήτησης τοπικών ακροτάτων, πρόκειται για επαναληπτικές μεθόδους που βελτιώνουν σταδιακά την τιμή της στοχαστικής συνάρτησης, παράγοντας μια βελτιωμένη λύση εφαρμόζοντας κανόνες μετάβασης που έχουν την μορφή:

$$x^{[k]} = x^{[k+1]} + \beta^{[k]} d^{[k]}$$

όπου το β είναι μια βαθμωτή παράμετρος κλίμακας και d μια διεύθυνση στο R^n τέτοιες ώστε: $f(x^{[k]}) < f(x^{[k-1]})$ για οποιαδήποτε μετατόπιση k .

Ανάλογα με τις εκτιμήσεις των παραμέτρων β και d οι τεχνικές αναζήτησης διαφοροποιούνται, ενώ οι μέθοδοι χωρίζονται σε δύο κατηγορίες ανάλογα με το αν χρησιμοποιούν ή όχι τις παραγώγους της συνάρτησης. Οι δύο βασικές μέθοδοι είναι: οι μέθοδοι κλίσης και οι μέθοδοι άμεσης αναζήτησης. [33]



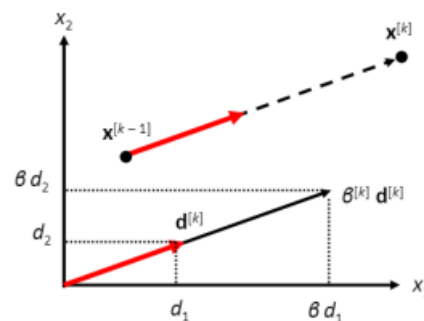
Μέθοδοι κλίσης:

- *Πλέον απότομη κατάβαση (Steepest descent):*

Είναι η απλούστερη τεχνική κλίσης, στην οποία η διεύθυνση d είναι αντίθετη στο διάνυσμα κλίσης ∇f της συνάρτησης f . Η διαδικασία βασίζεται στον κανόνα κίνησης του νερού για τον οποίο ισχύει ότι:

$$x^{[k]} = x^{[k-1]} - \beta^{[k]} \nabla f(x^{[k-1]})$$

Η μετακίνηση είναι κάθετη στη διεύθυνση της τρέχουσας λύσης $x^{[k-1]}$ και κάθε επόμενο σημείο $x^{[k]}$ είναι η θέση ελαχίστου της f , κατά μήκος της διεύθυνσης που ορίζει η κλίση $\nabla f(x^{[k-1]})$.



Το $\beta^{[k]}$ προσδιορίζεται έτσι ώστε να ελαχιστοποιείται η f στο ευθύγραμμο τμήμα $x^{[k-1]} - \beta^{[k]} \nabla f(x^{[k-1]})$, δηλαδή: $g(\beta) = [x^{[k-1]} - \beta \nabla f(x^{[k-1]})]$.

Μειονέκτημα αυτής της μεθόδου είναι η αργή σύγκλιση με μικρά κάθετα βήματα. [33]

- *Συζυγείς κλίσεις (conjugate gradients):*

Στη μέθοδο αυτή η πορεία αναζήτησης επιτυγχάνεται όταν η διεύθυνση (d) παράγεται ως γραμμικός συνδυασμός του διανύσματος κλίσης στο τρέχον σημείο: $(x^{[k-1]})$ και στο προηγούμενο σημείο: $(x^{[k-2]})$ ως εξής:

$$x^{[k]} = x^{[k-1]} - \beta^{[k]} [\nabla f(x^{[k-1]}) + \gamma^{[k]} \nabla f(x^{[k-2]})]$$

Το $\beta^{[k]}$ προσδιορίζεται με ελαχιστοποίηση της βοηθητικής έκφρασης $g(\beta)=f(x^{[k]})$.

Στη μέθοδο αυτή αν η συνάρτηση είναι τετραγωνικής μορφής, τότε το ακρότατο εντοπίζεται σε n βήματα. [33]

- *Μέθοδος της απότομης καθόδου:*

Η μέθοδος της απότομης καθόδου χαρακτηρίζεται από πρώτης τάξης ρυθμό σύγκλισης. Βασική προϋπόθεση αυτής της μεθόδου βελτιστοποίησης είναι η δυνατότητα αποθήκευσης και αντιστροφής του εσσιανού μητρώου $\nabla^2 f$ της συνάρτησης κόστους. Πιο συγκεκριμένα, η μέθοδος της απότομης καθόδου είναι μια επαναληπτική διαδικασία εύρεσης ενός τοπικού ελαχίστου μιας διαφορίσιμης συνάρτησης. Προκειμένου να προκύψει η αναδρομική σχέση της μεθόδου ορίζεται το σύστημα [34]:

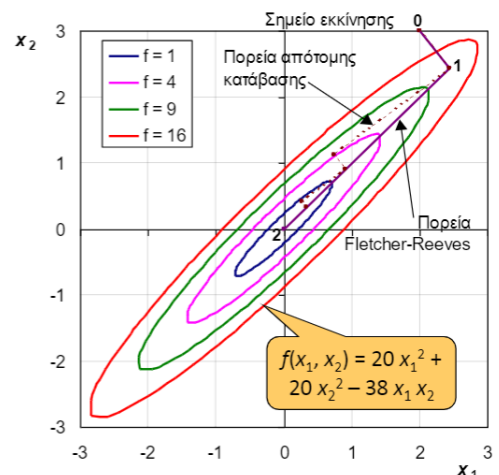
$\dot{z} = -\nabla f(z)$, για $z(t=0)=z_0$ και όπου ∇f ορίζεται από την εξίσωση:

$$\nabla f(z) = \sum_{i=1}^N (R_i - R_{max}) \nabla R_i$$

όπου:

$$\nabla = (\nabla_1, \dots, \nabla_k, \dots, \nabla_M)^T, \text{ με } \nabla_k = \left(\frac{\partial}{\partial x_k}, \frac{\partial}{\partial y_k} \right) \text{ και } 1 \leq k \leq M.$$

Η λύση του δυναμικού συστήματος συνιστά τροχιά $z(t) : \mathbb{R} \rightarrow \mathbb{R}^{2M}$ επί της οποίας η συνάρτηση f είναι φθίνουσα. Με αρχική κατάσταση Z_0 ισχύει ότι:



$$\begin{aligned} \dot{z} = -\nabla f(z) \implies \|\dot{z}\|^2 &= -\nabla f(z) * \dot{z} = -(f \circ z)'(t) \implies f(z) \\ &= f(z_0) - \int_{\tau=0}^{\tau=t} \|\dot{z}(\tau)\|^2 d\tau \leq f(z_0) \end{aligned}$$

Με τη μέθοδο Euler οδηγούμαστε στην αναδρομική σχέση:

$$z_{n+1} = z_n - \delta t \nabla f(z_n), z_{n=0} = z_0,$$

όπου n το όνομα της ακολουθίας και δt το βήμα της διακριτοποίησης.

Η παραπάνω έκφραση συνιστά τη διαδικασία βελτιστοποίησης με τη μέθοδο της απότομης καθόδου. Σε κάθε επανάληψη της μεθόδου, η κατεύθυνση ανίχνευσης καθορίζεται από το διάνυσμα της κλίσης της συνάρτησης κόστους. Αποδεικνύεται ότι για επαρκώς μικρό βήμα διακριτοποίησης η ακολουθία συγκλίνει ευσταθώς στη βέλτιστη κατάσταση: $\lim z_n = z^*$. Στη διατύπωση αυτή το βήμα ορίζεται ενιαία για κάθε κόμβο του πλέγματος και παραμένει σταθερό σε κάθε επανάληψη της μεθόδου [34].

- *Μέθοδος Limited-Memory BFGS (L-BFGS):*

Οι μέθοδοι βελτιστοποίησης Limited-Memory BFGS ανήκουν στην οικογένεια μεθόδων quasi – Newton. Στις μεθόδους αυτές χρησιμοποιείται μία προσέγγιση της κατεύθυνσης ανίχνευσης της Newton – Raphson προς αποφυγή της αντιστροφής του εσσιανού μητρώου της συνάρτησης κόστους f , διατηρώντας ταυτόχρονα υψηλό ρυθμό σύγκλισης. Το L-BFGS αποθηκεύει μόνο μερικά διανύσματα που αντιπροσωπεύουν την εκάστοτε προσέγγιση, τα οποία αντλούνται από αποθηκευμένα δεδομένα ή βιωματική γνώση. Πιο συγκεκριμένα, λόγω της προκύπτουσας απαίτησης γραμμικής μνήμης, η μέθοδος L-BFGS είναι ιδιαίτερα κατάλληλη για προβλήματα βελτιστοποίησης με πολλές μεταβλητές. Αντί για τον αντίστροφο Εσσιανό, διατηρεί ένα ιστορικό των προηγούμενων m ενημερώσεων της θέσης x και της κλίσης $\nabla f(x)$, όπου το μέγεθος ιστορικού m μπορεί να είναι μικρό, συνήθως $m < 10$ [34].

Δεδομένου ότι το L-BFGS έχει σχεδιαστεί για να ελαχιστοποιεί τις ομαλές συναρτήσεις χωρίς περιορισμούς, ο αλγόριθμος πρέπει να τροποποιηθεί για να χειρίζεται συναρτήσεις που περιλαμβάνουν μη διαφοροποιήσιμα στοιχεία ή περιορισμούς. Μια δημοφιλής κατηγορία τροποποιήσεων ονομάζονται μέθοδοι ενεργού συνόλου, με βάση την έννοια του

ενεργού συνόλου. Η ιδέα είναι ότι όταν περιορίζεται σε μια μικρή γειτονιά της τρέχουσας επανάληψης, η συνάρτηση και οι περιορισμοί μπορούν να απλοποιηθούν [34].

- Μέθοδος David-Fletcher-Powell formula (DFP)

Η μέθοδος DFP είναι υπεύθυνη για την εύρεση της εξίσωσης τομής που είναι πιο κοντά στην τρέχουσα εκτίμηση και ικανοποιεί τη συνθήκη καμπυλότητας. Αποτελεί την πρώτη μέθοδο που γενίκευσε τη μέθοδο της διατομής σε ένα πολυδιάστατο πρόβλημα. Αυτή η ενημέρωση διατηρεί τη συμμετρία και τη θετική βεβαιότητα του Εσσιανού πίνακα. Πιο αναλυτικά, με δεδομένο ένα σημείο $x_0 \in R^n$, η μέθοδος του Newton-Raphson δημιουργεί την κατεύθυνση αναζήτησης $H(f, x_0)^{-1}$ με βαθμό $f(x_0)$, όπου $H(f, x_0)$ είναι ο Εσσιανός της f στο x_0 . Με τη μέθοδο αυτή αποφεύγεται ο απευθείας υπολογισμός του Εσσιανού καθώς χρησιμοποιώντας μόνο την κλίση υπολογίζεται διαδοχικά ο αντίστροφος Εσσιανός: $\text{grad } f(x_0)$.

Η μέθοδος ακολουθεί τα παρακάτω βήματα:

- Αρχικοποίηση: Επιλέγεται ένα σημείο x_0 που προσεγγίζει τη θέση ενός τοπικού ελαχίστου
- Προσδιορισμός του διανύσματος μετατόπισης Δx_k : δίνονται x_k και d_k και έστω λ_k η τιμή του λ που ελαχιστοποιεί την $f(x_k - \lambda d_k)$. Στη συνέχεια ορίζεται $\Delta x_k = \lambda d_k$.
- Ενημέρωση για τη θέση ενός τοπικού ελαχίστου: Όρισε $x_{k+1} = x_k + \Delta x_k$
- Έλεγχος ικανοποίησης του κριτηρίου διακοπής
- Υπολογισμός της νέας κατεύθυνσης αναζήτησης [35]

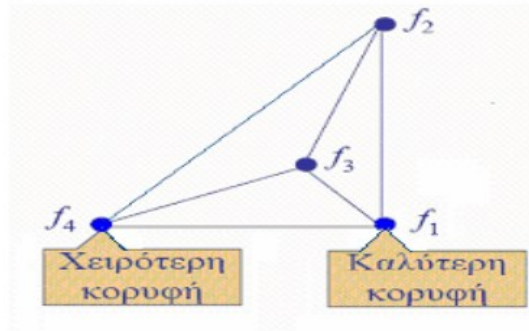
Μέθοδοι άμεσης αναζήτησης:

Οι τεχνικές άμεσης αναζήτησης είναι επαναληπτικές διαδικασίες που αντί να χρησιμοποιούν παραγώγους ή αριθμητικές προσεγγίσεις, εφαρμόζουν γεωμετρικά ανάλογα της κλίσης, εξερευνώντας τον Ευκλείδειο χώρο σε n γραμμικά ανεξάρτητες διευθύνσεις. Οι παράμετροι $\beta^{[k]}$ και $d^{[k]}$ της διαδικασίας μετάβασης επιλέγονται με βάση τη σχετική διάταξη των τιμών της συνάρτησης πάνω στα σημεία που ορίζεται κάθε φορά από το γεωμετρικό ανάλογο και όχι με βάση τις ίδιες τις τιμές συνάρτησης. [36]

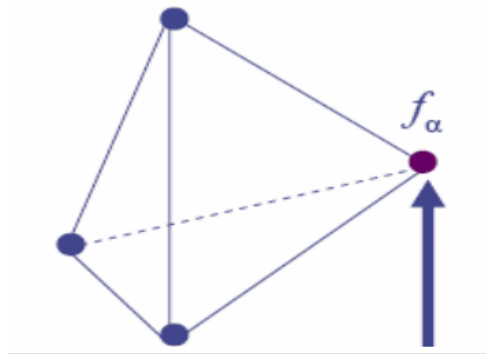
- Μέθοδος κατερχόμενου απλόκου:

Η μέθοδος downhill simplex είναι μια γεωμετρική μέθοδος για την επίτευξη ελαχιστοποίησης συναρτήσεων. Τα βήματα που ακολουθεί η μέθοδος αυτή παρουσιάζονται ακολούθως: [36]

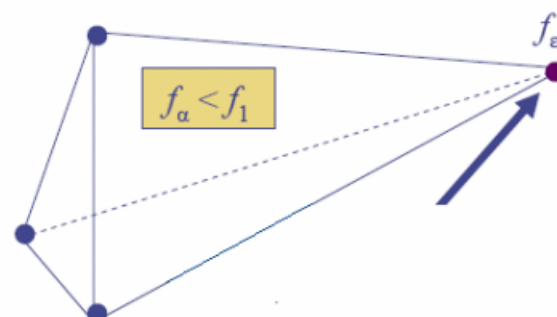
- Εντοπίζεται η καλύτερη και η χειρότερη κορυφή



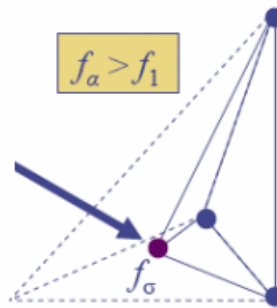
- Γίνεται ανάκλαση περί την χειρότερη κορυφή [36]



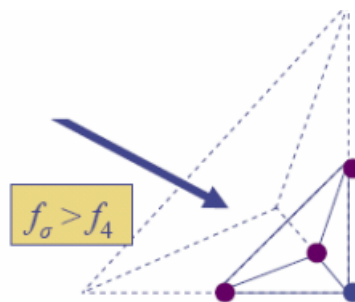
- Στην περίπτωση της αρνητικής κλίσης, τότε γίνεται επέκταση κατά μήκος της ανάκλασης, που γίνεται αποδεκτή εάν $f_\epsilon < f_\alpha$ [36]



- iv. Το απλόκο περικυκλώνει το ακρότατο και γίνεται συμπίεση αντίθετα στην διεύθυνση της ανάκλασης, εάν $f_{\sigma} < f_4$ [36]



- v. Πραγματοποιείται συρρίκνωση περί την καλύτερη κορυφή, εάν $f_{\sigma} < f_4$ [36]



1.4 Καθολική Βελτιστοποίηση

Στοχαστικές τεχνικές.

Σε ορισμένα προβλήματα η χρήση αναλυτικών μεθόδων βελτιστοποίησης δεν οδηγεί πάντα σε ικανοποιητικές λύσεις ή σε ορισμένες περιπτώσεις είναι αδύνατον να εφαρμοστούν, αφού θα πρέπει να είναι γνωστή η αντικειμενική συνάρτηση. Στις περιπτώσεις που οι αναλυτικοί μέθοδοι βελτιστοποίησης αδυνατούν να επιλύσουν προβλήματα, εφαρμόζονται οι στοχαστικοί αλγόριθμοι. Οι στοχαστικοί αλγόριθμοι δεν χρειάζονται περιγραφική ανάλυση των προβλημάτων αλλά εφαρμόζουν ορισμένες τυχαίες

διαδικασίες για να εντοπίσουν λύσεις κοντά στις βέλτιστες λύσεις. Οι βασικότεροι στοχαστικοί αλγόριθμοι βελτιστοποίησης παρουσιάζονται ακολούθως:

❖ *Τυχαία αναζήτηση*

Η τυχαία αναζήτηση είναι ο πιο απλός στοχαστικός αλγόριθμος, αφού ο στόχος της είναι να δοκιμάζει τυχαίους συνδυασμούς τιμών για τις παραμέτρους εισόδου. Η αντικειμενική συνάρτηση των προβλημάτων υπολογίζεται για κάθε συνδυασμό που προκύπτει και στη συνέχεια επιλέγεται ο συνδυασμός που δίνει τη βέλτιστη λύση στο πρόβλημα. Έχει διαπιστωθεί ότι η μέθοδος της τυχαίας αναζήτησης δεν προσδίδει τα επιθυμητά αποτελέσματα και γι' αυτό το λόγο χρησιμοποιείται κυρίως για συγκριτικούς λόγους. Υπάρχει μία παραλλαγή της τυχαίας αναζήτησης, η οποία ονομάζεται «τυχαίος περίπατος». Στην αναζήτηση αυτή επιλέγεται ένα αρχικό σημείο αναζήτησης και στη συνέχεια τυχαία αναζητείται ένα διπλανό σημείο. Η διαδικασία αυτή επαναλαμβάνεται για όλες τις διαστάσεις ενός προβλήματος, μεταβάλλοντας την τιμή της παραμέτρου εισόδου. Όμως, και αυτός ο τρόπος δεν οδηγεί στα βέλτιστα επιθυμητά αποτελέσματα [37].

❖ *Αναρρίχηση λόφου*

Στον αλγόριθμο αναρρίχησης λόφου, η αναζήτηση ξεκινάει από ένα τυχαίο σημείο, στη συνέχεια εξετάζονται τυχαία γειτονικά σημεία και καταλήγει προς την κατεύθυνση που εντοπίστηκε η βέλτιστη λύση. Οι υποψήφιες γειτονικές λύσεις, προκύπτουν με μία μικρή μεταβολή της τιμής μιας ή περισσότερων παραμέτρων εισόδου. Τα μειονεκτήματα αυτής της μεθόδου είναι ότι εξαρτάται άμεσα από την αρχική επιλογή της τυχαίας λύσης και ότι οδηγείται προς ένα τοπικό ακρότατο της συνάρτησης καταλληλότητας στο οποίο και εγκλωβίζεται [37].

❖ *Προσομοιωμένη ανόπτηση*

Ο αλγόριθμος προσομοιωμένης ανόπτησης, αποτελεί επέκταση του αλγορίθμου αναρρίχησης λόφου [38]. Σε αυτή τη μέθοδο, προσομοιώνεται η διαδικασία κατά την οποία ένα υλικό θερμαίνεται ξεπερνώντας τη θερμοκρασία τήξης του, μέχρις ότου δημιουργηθεί η κρυσταλλική δομή, η οποία ελαχιστοποιεί την ενέργειά του. Στην περίπτωση της απότομης ψύξης, το υλικό αποκτά μία άμορφη μάζα και η ενέργειά του γίνεται μεγαλύτερη. Για να επιτευχθεί επομένως η κρυσταλλική μορφή, θα πρέπει να δίνεται έμφαση στον έλεγχο της αλλαγής θερμοκρασίας. Στα προβλήματα της βελτιστοποίησης αυτό συσχετίζεται ως εξής: το επίπεδο ενέργειας αντιστοιχίζεται στην τιμή της συνάρτησης καταλληλότητας και η κρυσταλλική δομή στο ολικό ελάχιστο. Βασικό στοιχείο σ' αυτόν τον αλγόριθμο είναι η αποφυγή σύγκλισης σε ένα τοπικό

ελάχιστο. Πιο αναλυτικά, αν η γειτονική θέση οδηγεί σε καλύτερη λύση γίνεται αμέσως δεκτή και η αναζήτηση συνεχίζεται από εκείνο το σημείο. Το ίδιο θα συμβεί και στην περίπτωση που το γειτονικό σημείο οδηγεί στο χειρότερο σημείο, αλλά στην περίπτωση αυτή, η θέση θα γίνει δεκτή λαμβάνοντας υπόψιν μία πιθανότητα. Η τιμή αυτής της πιθανότητας εξαρτάται από το πόσο χειρότερη είναι η λύση από την τρέχουσα βέλτιστη καθώς και από την τιμή μιας παραμέτρου η οποία μειώνεται σταδιακά με την εξέλιξη του αλγορίθμου και προσομοιάζει την θερμοκρασία. Οπότε όσο μικρότερη είναι η τιμή της θερμοκρασίας, τόσο μικρότερη θα είναι και η πιθανότητα αποδοχής της χειρότερης λύσης. Με αυτό το τρόπο ο αλγόριθμος ξεκινά με μεγάλο πεδίο αναζήτησης και καταλήγει στην βέλτιστη περιοχή [37].

❖ *Αναζήτηση με απαγορευμένες θέσεις*

Η μέθοδος αυτή χαρακτηρίζεται από τη δημιουργία μιας λίστας με απαγορευμένες θέσεις, στις οποίες δεν επιτρέπεται η μετάβαση, αφού οι θέσεις αυτές έχουν ήδη εξεταστεί. Η λίστα αυτή έχει συγκεκριμένο μήκος, οπότε η εισαγωγή νέων θέσεων στη λίστα έχει ως αποτέλεσμα τη διαγραφή άλλων θέσεων. Για το λόγο αυτό θα ήταν ωφέλιμο η λίστα να περιέχει ολόκληρες οικογένειες σημείων με συγκεκριμένες ιδιότητες. Βέβαια με αυτό το τρόπο υπάρχει το ενδεχόμενο να μην υπάρξει μετάβαση σε καλές θέσεις που δεν έχει επισκεφτεί ακόμα ο αλγόριθμος. Τέλος, να αναφερθεί το γεγονός ότι ο αλγόριθμος είναι δυνατόν να τερματίσει σε μία θέση που πιθανόν να μην είναι η καλύτερη θέση που έχει επισκεφτεί [37].

❖ *Εξελικτικοί αλγόριθμοι*

Οι εξελικτικοί αλγόριθμοι διαφέρουν από τις υπόλοιπες στοχαστικές μεθόδους καθώς διατηρούν ένα σύνολο θέσεων αναζήτησης για την επίλυση ενός προβλήματος βελτιστοποίησης. Ο αλγόριθμος ξεκινά με τον υπολογισμό τυχαίων θέσεων και στη συνέχεια ακολουθούν ενέργειες που εξελίσσουν το υπάρχον σύνολο σε ένα καινούριο και αυτό επαναλαμβάνεται μέχρι να ικανοποιηθεί μία συνθήκη τερματισμού. Οι αλγόριθμοι αυτοί μπορούν να αποφύγουν την παγίδευση σε ένα τοπικό βέλτιστο και γι' αυτό το λόγο μπορούν να βρουν τις ολικά βέλτιστες λύσεις [37].

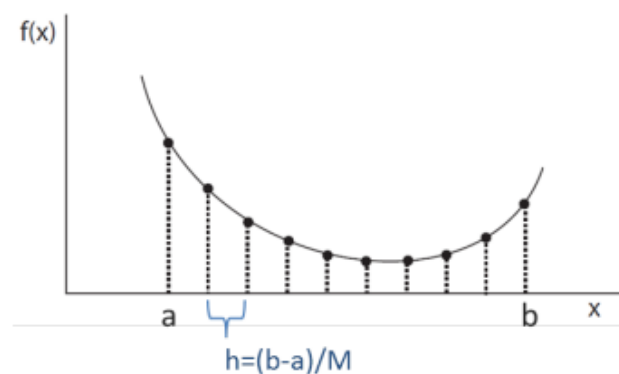
Δύο αλγόριθμοι αυτής της κατηγορίας είναι οι γενετικοί αλγόριθμοι και η βελτιστοποίηση σμήνους σωματιδίων. Στους γενετικούς αλγορίθμους οι παράμετροι εισόδου ονομάζονται γονίδια, ο συνδυασμός τους για τη δημιουργία ενός διανύσματος αναζήτησης ονομάζεται χρωμόσωμα, ενώ ο πληθυσμός που εξελίσσεται μέσα από τις επαναλήψεις ονομάζονται γενιές. Η διαδικασία αρχίζει με τη δημιουργία ενός τυχαίου συνόλου χρωμοσωμάτων που

είναι και ο αρχικός πληθυσμός και κατά τη διάρκεια των γενιών λαμβάνουν μέρος οι διαδικασίες της επιλογής χρωμοσωμάτων, της διασταύρωσης των σημείων για τη δημιουργία ζεύγους παιδιών και της μετάλλαξης, όπου δημιουργούνται καινούριες περιοχές αναζήτησης.

Ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων, βασίζεται σε συμπεριφορές που συναντιούνται στη φύση [39]. Συγκεκριμένα, κάθε διάνυσμα εισόδου στις παραμέτρους ονομάζεται σωματίδιο, ενώ το σύνολο των σωματιδίων ονομάζεται σμήνος. Η αρχή του αλγορίθμου γίνεται τυχαία, ενώ τα σωματίδια παραμένουν υποψήφιες λύσεις καθώς κινούνται στην επιφάνεια. Η νέα θέση ενός σωματιδίου σμήνους υπολογίζεται από την ταχύτητά του, καθώς εκείνη δίνει τόσο την κατεύθυνση κίνησης των σωματιδίων, όσο και την απόσταση μετακίνησής τους. Όλα τα σωματίδια αξιολογούνται με βάση την αντικειμενική συνάρτηση του προβλήματος και καθένα από αυτά γνωρίζει την καλύτερη θέση που έχει εντοπιστεί ως την τρέχουσα επανάληψη καθώς και την καλύτερη θέση που έχει βρεθεί από οποιοδήποτε σωματίδιο στη γειτονιά του.

Τεχνικές διαστημάτων.

Αν υποθέσουμε ότι σε ένα διάστημα $[a,b]$ γνωρίζουμε ότι βρίσκεται ο ελαχιστοποιητής x^* , τότε το διάστημα αυτό ονομάζεται διάστημα αβεβαιότητας.



Εικόνα 1 15: Σχηματική αναπαράσταση της αναζήτησης με διαμέριση

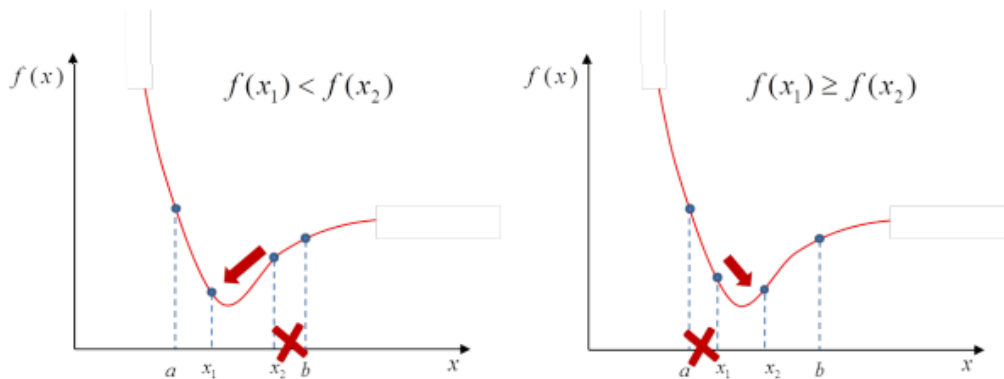
Στην περίπτωση που το μήκος είναι μικρό, τότε υπάρχει μία πιο ακριβή εκτίμηση για τη θέση του ελαχίστου, η οποία περιγράφεται από τη σχέση:

$$x^* = \frac{b+a}{2} \pm \frac{b-a}{2}$$

Οι μέθοδοι που χρησιμοποιούν διαστήματα αβεβαιότητας, απαιτούν μόνο τον υπολογισμό τιμών της συνάρτησης στοχεύοντας στη συρρίκνωση του διαστήματος αβεβαιότητας ώστε το ελάχιστο να εντοπιστεί με προκαθορισμένη ακρίβεια.

Έστω δύο νέα σημεία x_1 και x_2 τα οποία βρίσκονται εντός του διαστήματος $[a, b]$, τέτοια ώστε να ισχύει $a < x_1 < x_2 < b$. Τότε υπολογίζονται οι τιμές της συνάρτησης $f(x_1)$ και $f(x_2)$ ως εξής:

- Εάν $f(x_1) < f(x_2)$ τότε το νέο διάστημα αβεβαιότητας θα είναι το $[a, x_2]$
- Εάν $f(x_1) > f(x_2)$ τότε το νέο διάστημα αβεβαιότητας θα είναι το $[x_1, b]$
- Εάν $f(x_1) = f(x_2)$ τότε το νέο διάστημα αβεβαιότητας θα είναι το $[x_1, x_2]$ [6].



Εικόνα 1 16: Αναπαράσταση της συρρίκνωσης του διαστήματος αβεβαιότητας

Υπάρχουν μέθοδοι που χρησιμοποιούν τεχνικές διαστημάτων είναι: η απόδοση μεθόδου, η αναζήτηση ίσων διαστημάτων, η αναζήτηση διχοτόμησης, η αναζήτηση Fibonacci καθώς και η αναζήτηση χρυσής τομής.

2. ΜΕΘΟΔΟΙ ΠΟΛΛΑΠΛΩΝ ΕΚΚΙΝΗΣΕΩΝ

2.1 Pure random search

Ο αλγόριθμος «Pure Random Search» (PRS) είναι ο πρώτος αλγόριθμος στοχαστικής βελτιστοποίησης που εισήχθη πριν από οποιονδήποτε γενετικό αλγόριθμο από τον Brooks [44,45]. Ο αλγόριθμος αυτός, συνιστάται στη δειγματοληψία ανεξάρτητων σημείων, ομοιόμορφα κατανεμημένων στον χώρο αναζήτησης [46]. Συγκεκριμένα, ο αλγόριθμος PRS δημιουργεί ανεξάρτητα κατανεμημένα σημεία με πεδίο ορισμού S . Δημιουργούνται σημεία σύμφωνα με την ομοιόμορφη κατανομή, ωστόσο θα μπορούσε να χρησιμοποιηθεί οποιαδήποτε κατανομή πιθανότητας. Στην περίπτωση που πληρείται ένα κριτήριο διακοπής του αλγορίθμου, τότε το καλύτερο σημείο της ακολουθίας που έχει δημιουργηθεί μέχρι εκείνη τη χρονική στιγμή, χρησιμοποιείται ως προσέγγιση για τη βέλτιστη λύση [47].

Ακολουθία βημάτων αλγορίθμου:

Βήμα 1^ο:

Αρχικοποίηση το $x_0 \in S$ σύμφωνα με το μέτρο δ στο S .

Ορισμός δείκτης επανάληψης $k=0$.

Ορίζεται $Y_{BEST} = Y_0 = f(X_0)$.

Βήμα 2^ο:

Δημιουργία x_{k+1} .

Ορισμός $Y_{k+1} = f(X_{k+1})$.

Ενημέρωση του καλύτερου σημείου, μέχρι στιγμής, ως εξής: $Y_{BEST} = \min\{Y_{BEST}, Y_{k+1}\}$

Βήμα 3^ο:

Εάν πληρείται ένα κριτήριο διακοπής, τότε σταματά η εκτέλεση αλγορίθμου.

Διαφορετικά, αυξάνεται το k και επιστρέφει στο βήμα 1 [47,48].

Τα σημεία X_k , όπου $k=\{0, 1, \dots\}$ είναι σημεία ανεξάρτητα και κατανέμονται πανομοιότυπα στο S , με τιμές αντικειμενικής συνάρτησης που ικανοποιούν τη σχέση: $Y_k = f(X_k)$.

Πλεονεκτήματα και Μειονεκτήματα Αλγορίθμου PRS:

Ένα από τα σημαντικότερα πλεονεκτήματα των μεθόδων τυχαίας αναζήτησης, είναι ότι έχουν τη δυνατότητα να λύνουν προβλήματα μεγάλης κλίμακας [49] και ότι χαρακτηρίζονται από προσαρμοστικότητα [50]. Επίσης, τα όρια του χώρου αναζήτησης, δε χρειάζεται να προσδιοριστούν με ακρίβεια, ενώ μπορούν να προγραμματιστούν ώστε να επιλύσουν προβλήματα περιορισμένων ανισοτήτων [50]. Χαρακτηριστικό του αλγορίθμου PRS είναι ότι μπορεί να εφαρμοστεί σε σύνθετα προβλήματα, αφού βασίζεται σε αξιολογήσεις συναρτήσεων και όχι σε Εσσιανά κριτήρια αξιολόγησης [51]. Τα Εσσιανά κριτήρια, γνωστά και ως Εσσιανές παράγωγοι, είναι μαθηματικά μεγέθη που χρησιμοποιούνται συνήθως σε προβλήματα βελτιστοποίησης. Αυτά τα κριτήρια παρέχουν πληροφορίες σχετικά με τον ρυθμό μεταβολής της συνάρτησης από κάποιο σημείο στο χώρο αναζήτησης. Συγκεκριμένα, οι Εσσιανές παράγωγοι περιγράφουν την πλευρά της κλίσης της συνάρτησης και την ταχύτητα με την οποία αυτή αλλάζει. Κατά την επίλυση προβλημάτων βελτιστοποίησης, οι Εσσιανές παράγωγοι μπορούν να χρησιμοποιηθούν για να εντοπίσουν τα κρίσιμα σημεία, όπως τα τοπικά ελάχιστα ή τοπικά μέγιστα, και να καθοδηγήσουν τον αλγόριθμο βελτιστοποίησης προς την κατεύθυνση του βέλτιστου αποτελέσματος. Οι Εσσιανές παράγωγοι είναι σημαντικές στην προσέγγιση των βέλτιστων λύσεων σε προβλήματα βελτιστοποίησης, αλλά απαιτούν πληροφορίες για τις παραγωγούς της συνάρτησης στο χώρο αναζήτησης. Σε αντίθεση, ο αλγόριθμος PRS βασίζεται σε αξιολογήσεις συναρτήσεων, όχι σε Εσσιανά κριτήρια. Αυτό το καθιστά κατάλληλο για την επίλυση πολυκριτηριακών προβλημάτων βελτιστοποίησης όπου υπάρχουν πολλά αντικρουόμενα κριτήρια που πρέπει να ληφθούν υπόψη κατά την επιλογή των λύσεων. Επίσης, οι μέθοδοι τυχαίας αναζήτησης μπορούν να κωδικοποιηθούν και να εφαρμοστούν σε πολλών κατηγοριών προβλήματα βελτιστοποίησης [51].

Εκτός όμως από τα πλεονεκτήματα που χαρακτηρίζουν την PRS μέθοδο, υπάρχουν και ορισμένα μειονεκτήματα που θα πρέπει να αναφερθούν. Αρχικά, ένα βασικό μειονεκτήματα που την χαρακτηρίζει είναι ο μεγάλος αριθμός δοκιμών που υλοποιεί ώστε να προσαρμοστεί σε κάθε είδους πρόβλημα, το οποίο όμως επιφέρει πιθανά λάθη [50]. Επίσης, υπάρχουν πολλές διακυμάνσεις μεταξύ των εκτελέσεων, αφού οι αναζητήσεις είναι τυχαίες και γρήγορες [52]. Παράλληλα, εάν το πλαίσιο δειγματοληψίας είναι μεγάλο, τότε η μέθοδος της τυχαίας δειγματοληψίας μπορεί να μην είναι πρακτική [53]. Τέλος, υπάρχουν και άλλα δύο σημαντικά μειονεκτήματα της μεθόδου που θα πρέπει να ληφθούν

υπόψιν. Ο αλγόριθμος PRS απαιτεί τον έλεγχο της συνάρτησης αξιολόγησης σε πολλά σημεία του χώρου αναζήτησης. Αυτό μπορεί να είναι πολύ χρονοβόρο και δαπανηρό, ειδικά αν η συνάρτηση αξιολόγησης είναι πολύπλοκη ή ακριβή στον υπολογισμό της. Επομένως, η ανάγκη για πολλές κλήσεις μπορεί να αποτελέσει ένα σημαντικό μειονέκτημα, ειδικά σε περιπτώσεις όπου οι περιορισμοί στον αριθμό των αξιολογήσεων είναι σημαντικοί. Κλείνοντας, η μέθοδος PRS διακρίνεται από την τυχαία δειγματοληψία και τις τυχαίες αναζητήσεις. Αυτό σημαίνει ότι η σύγκλιση προς τον πραγματικό ελάχιστο της συνάρτησης αξιολόγησης είναι πιθανό να χρειάζεται πολλές επαναλήψεις και εξαιρετική τύχη. Υπάρχουν θεωρήματα σύγκλισης που εξετάζουν τη σύγκλιση της μεθόδου PRS, αλλά δεν υπάρχει εγγύηση ότι θα βρεθεί το πραγματικό ελάχιστο σε ένα λογικό χρονικό διάστημα. Αυτό μπορεί να είναι πρόβλημα σε πρακτικές εφαρμογές, όπου ο χρόνος είναι περιορισμένος και η ακρίβεια είναι κρίσιμη.

2.2 Η μέθοδος Multistart

Η μέθοδος Πολλαπλής Εκκίνησης (Multistart) είναι μία από τις απλούστερες μεθόδους καθολικής ελαχιστοποίησης. Σε κάθε επανάληψη αυτής της μεθόδου παράγεται μία λύση, η οποία είναι ένα τοπικό ελάχιστο. Η καλύτερη λύση ανάμεσα σε αυτές που έχουν προκύψει κατά τη διάρκεια των επαναλήψεων αποτελεί την τελική έξοδο του αλγορίθμου [54].

Τα βήματα που ακολουθεί η μέθοδος Multistart παρουσιάζονται παρακάτω:

Έστω ότι υπάρχει μια διαδικασία ελαχιστοποίησης που ξεκινά από ένα αρχικό σημείο x που ανήκει στον χώρο αναζήτησης S . Μετά από την εκτέλεση της διαδικασίας, φτάνουμε σε ένα σημείο L , το οποίο αποτελεί το ελάχιστο της συνάρτησης που προσπαθούμε να ελαχιστοποιήσουμε. Στο πλαίσιο αυτό, το αρχικό σημείο x συμβολίζεται ως $x \in S$ και το τελικό ελάχιστο σημείο L συμβολίζεται ως $L(x)$. Οπότε:

Βήμα 1^ο :

Θέσε $X^* = \emptyset$. Το σύνολο X^* αντιπροσωπεύει όλα τα τοπικά ελάχιστα που θα ανακαλυφθούν από τη μέθοδο.

Βήμα 2^ο :

Επιλογή με ομοιόμορφη δειγματοληψία ενός σημείου $x \in S$.

Βήμα 3^ο :

Αναζήτηση ενός τοπικού ελαχίστου x^* , δηλαδή : $x^*=L(x)$

Βήμα 4^ο :

Στην περίπτωση όπου $x \notin X^*$ τότε:

$$\Theta \acute{\epsilon}\sigma\epsilon X^* \leftarrow X^* \cup \{x^*\}$$

Βήμα 5^ο :

Αν ισχύει το κριτήριο τερματισμού τότε διακοπή του αλγορίθμου.

Διαφορετικά μεταβαίνει στο Βήμα 2.

Πιο συγκεκριμένα, η μέθοδος πολλαπλής εκκίνησης ξεκινάει από τυχαία σημεία του χώρου αναζήτησης, τα οποία ονομάζονται σημεία εκκίνησης. Ανάλογα με τον τύπο της αντικειμενικής συνάρτησης, η μέθοδος επαναλαμβάνεται εντοπίζοντας τοπικά ελάχιστα στο χώρο αναζήτησης, τα οποία αντικαθίσταται κάθε φορά από το επόμενο βέλτιστο τοπικά ελάχιστο. Η διαδικασία τερματίζει όταν ικανοποιηθεί κάποιο κριτήριο τερματισμού [54].

Πλεονεκτήματα και Μειονεκτήματα Αλγορίθμου Multistart

Η μέθοδος πολλαπλής εκκίνησης δεν θεωρείται αποτελεσματική, καθώς υπάρχει ενδεχόμενο ένα βέλτιστο τοπικό ελάχιστο, να εντοπιστεί παραπάνω από μία φορά στο χώρο αναζήτησης [54,55]. Η περίπτωση αυτή θα μπορούσε να αποφευχθεί εάν υπήρχε τρόπος αναγνώρισης των ήδη υπαρχόντων σημείων στην περιοχή προσέλευσης [56]. Η περιοχή προσέλευσης ενός τοπικού ελαχίστου είναι μια περιοχή του χώρου αναζητήσεως, τέτοια ώστε από οποιοδήποτε σημείο της και αν ξεκινήσει μια μέθοδος τοπικής βελτιστοποίησης, θα καταλήξει στο ίδιο τοπικό ελάχιστο. Η περιοχή προσέλευσης A_i του τοπικού ελαχίστου x_i^* ορίζεται ως εξής [57]:

$$A_i = \{x \in S, L(x) = x_i^*\}, \quad (2.1)$$

όπου:

S: ο χώρος αναζήτησης,

$L(x)$: το ελάχιστο σημείο μιας διαδικασίας τοπικής ελαχιστοποίησης L , για $x \in S$.

Αν ο χώρος S περιέχει z τοπικά ελάχιστα, τότε από την σχέση (2.1) προκύπτει ότι:

$$\bigcup_{i=1}^z A_i = S \quad (2.2)$$

Αν συμβολιστεί με $m(A)$ το μέτρο Lebesgue των $A \subset R^n$, τότε επειδή οι περιοχές προσελκύσεων είναι μεταξύ τους ξένες θα ισχύει:

$$m(S) = \sum_{i=1}^z m(A_i) \quad (2.3)$$

Αν ένα σημείο $x \in S$ λαμβάνεται δειγματοληπτικά με ομοιόμορφη κατανομή, τότε η πιθανότητα να ανήκει στην περιοχή A_i το σημείο x δίνεται από τη σχέση:

$$p_i = \frac{m(A_i)}{m(S)} \quad (2.4)$$

Αν στο S ληφθούν δειγματοληπτικά K σημεία, τότε η πιθανότητα ότι τουλάχιστον ένα σημείο από

αυτά βρίσκεται στην περιοχή προσέλκυσης δίνεται από τη σχέση:

$$p_x \in A_i = 1 - \left(1 - \frac{m(A_i)}{m(S)}\right)^K = 1 - (1 - p_i)^K \quad (2.5)$$

Από τη σχέση (2.5) γίνεται αντιληπτό ότι καθώς το K μεγαλώνει, η πιθανότητα να οδηγηθούμε σε ένα σημείο που να οδηγήσει στο ελάχιστο i τείνει στο 1. Άρα η μέθοδος πολλαπλής εκκίνησης είναι σε θέση να ανακαλύψει όλα τα ελάχιστα στον χώρο αναζήτησης [57]. Επίσης σημαντικό μειονέκτημα της μεθόδου είναι ο χρόνος που απαιτείται για την εύρεση του βέλτιστου τοπικού ελαχίστου, αφού θα πρέπει να ελεγχθεί ένα σύνολο από σημεία στο χώρο αναζήτησης. Θα πρέπει επίσης να αναφερθεί, ότι ανάλογα με την αρχική κατανομή των σημείων στο χώρο αναζήτησης, εξαρτάται αν θα υπάρξει επιτυχής αναζήτηση τοπικού ελαχίστου. Συγκεκριμένα, όταν η αρχική ποσότητα κατανομής των σημείων είναι μεγάλη, τότε είναι βέβαιο ότι θα εντοπιστεί το βέλτιστο τοπικό ελάχιστο. Αντίθετα, στην περίπτωση που δοθεί μικρό πλήθος σημείων, τότε δεν είναι βέβαιο ότι θα εντοπιστεί το τοπικό ελάχιστο. Τέλος, υπάρχει και το ενδεχόμενο του εγκλωβισμού σε τοπικά ελάχιστα, όπου η μέθοδος θεωρεί ότι το τοπικό ελάχιστο βρίσκεται σε ένα συγκεκριμένο σημείο του χώρου αναζήτησης, εγκλωβίζεται στην περιοχή αυτή και προσπαθεί επαναληπτικά να εντοπίσει το βέλτιστο τοπικό ελάχιστο σε εκείνη την περιοχή.

Εκτός όμως από τα μειονεκτήματα που προαναφέρθηκαν, η μέθοδος χαρακτηρίζεται και από πλεονεκτήματα. Συγκεκριμένα, όπως αποδείχθηκε και από τη σχέση (2.5) που προαναφέρθηκε, η μέθοδος πολλαπλής εκκίνησης είναι ικανή να εντοπίσει όλα τα ελάχιστα σ' ένα χώρο αναζήτησης S . Επίσης, σε μεγάλο αρχικό αριθμό κατανομής σημείων, η μέθοδος είναι βέβαιο ότι θα εντοπίσει το βέλτιστο τοπικό ελάχιστο.

Τεχνικές Αρχικοποίησης

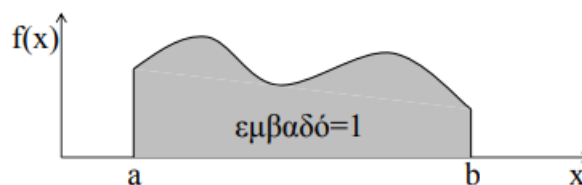
Στη βελτιστοποίηση, υπάρχουν διάφορες κατανομές που χρησιμοποιούνται για την περιγραφή και την εξερεύνηση του χώρου αναζήτησης. Η επιλογή της κατάλληλης κατανομής εξαρτάται από τους εξής παράγοντες: τις απαιτήσεις του προβλήματος, τον τύπο των παραμέτρων και τον τρόπο που ορίζονται οι παράμετροι στο χώρο αναζήτησης του προβλήματος. Οι πιο γνωστές κατανομές είναι οι εξής:

Ομοιόμορφη κατανομή: Στην ομοιόμορφη κατανομή, κάθε τιμή στο χώρο αναζήτησης έχει ίση πιθανότητα επιλογής. Η κατανομή συνήθως συμβολίζεται με $U(a,b)$, όπου a,b είναι οι τιμές στα δύο άκρα. Η συνάρτηση πυκνότητας πιθανότητας της συνεχούς ομοιόμορφης κατανομής είναι:

$$\begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & x < a \text{ or } x > b \end{cases}$$

Μία συνάρτηση $f(x)$ καλείται συνάρτηση πυκνότητας πιθανοτήτων στο διάστημα $a \leq x \leq b$, εάν πληροί δύο κριτήρια:

- $f(x) \geq 0$ για κάθε x μεταξύ a και b



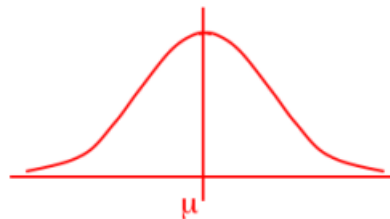
- Το συνολικό εμβαδό κάτω από την καμπύλη μεταξύ a και b είναι 1.0 [58].

Κανονική κατανομή: Η κανονική κατανομή είναι η πιο σημαντική από όλες τις κατανομές πιθανοτήτων. Θεωρείται σημαντική στους αλγορίθμους βελτιστοποίησης, καθώς βασίζεται στη στοχαστική αναζήτηση όπως για παράδειγμα ο γενετικός αλγόριθμος. Η συνάρτηση πυκνότητας πιθανοτήτων μιας κανονικής τυχαίας μεταβλητής δίνεται από την σχέση:

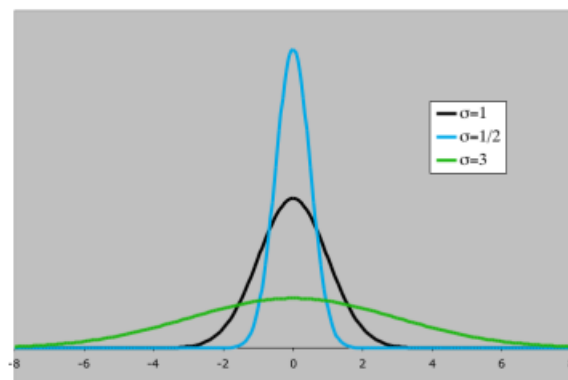
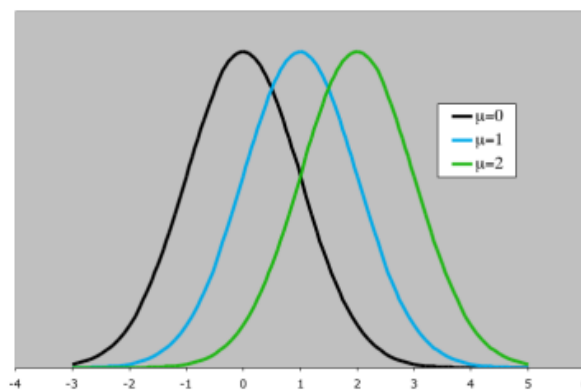
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad -\infty < x < \infty \quad (2.6)$$

όπου σ είναι η τυπική απόκλιση και μ το μέσο.

Η γραφική παράσταση έχει σχήμα καμπάνας και είναι συμμετρική ως προς το μ .



Αν αυξηθεί το μέσο (μ), τότε μετατοπίζεται η καμπύλη προς τα δεξιά. Ενώ, αν αυξηθεί η τυπική απόκλιση (σ), τότε «πλαταίνει» η καμπύλη [58].



Αποδοτική κατανομή: Η αποδοτική κατανομή παρουσιάζει υψηλή πιθανότητα για μικρές τιμές και μειώνεται εκθετικά καθώς οι τιμές αυξάνονται.

Βήτα-κατανομή: Η βήτα-κατανομή χρησιμοποιείται στη βελτιστοποίηση για τη μοντελοποίηση και την επιλογή παραμέτρων που περιορίζονται σε ένα συγκεκριμένο διάστημα [58].

Κατανομή αποτροπής: Είναι μια μέθοδος που επιχειρεί να βρει τα τοπικά ελάχιστα, ξεκινώντας αναζητήσεις από διάφορα σημεία στο χώρο. Ενώ άλλοι αλγόριθμοι χρησιμοποιούν μια συγκεκριμένη ομοιόμορφη κατανομή για να δημιουργήσουν τα αρχικά σημεία εκκίνησης, ο αλγόριθμος αποτροπής επιτρέπει την ευέλικτη αλλαγή της κατανομής μετά από κάθε τοπική αναζήτηση. Αυτό έχει σαν αποτέλεσμα να μειώνεται η πιθανότητα να δαπανώνται πολλές τοπικές αναζητήσεις σε περιοχές όπου είναι απίθανο να βρεθούν νέα τοπικά ελάχιστα. Πιο συγκεκριμένα, ο αλγόριθμος αποτροπής ακολουθεί τα παρακάτω βήματα: Αρχικά, δημιουργεί ένα σημείο y στον χώρο K , χρησιμοποιώντας μια ομοιόμορφη κατανομή και στη συνέχεια μετασχηματίζει το y σ' ένα νέο σημείο u που θα χρησιμοποιηθεί ως αρχικό σημείο για τις τοπικές αναζητήσεις. Ο μετασχηματισμός αυτός γίνεται για να προσπαθήσει ο αλγόριθμος να επισκεφθεί περιοχές έλξης που μπορεί να μην έχουν ανιχνευθεί από προηγούμενα δείγματα. Έπειτα, για κάθε προηγούμενο επισκεφθέν σημείο z , το σημείο y απομακρύνεται από το z . Η συνολική αποτροπή που ασκείται στο y , είναι η συνολική επίδραση από κάθε προηγούμενο σημείο αποτροπής [59].

Η απώθηση που το σημείο z ασκεί στο σημείο y δείχνει την κατεύθυνση από το σημείο y προς το σημείο z , και η έντασή της απώθησης, που συμβολίζεται ως $\Delta(r)$, εξαρτάται αποκλειστικά από την απόσταση μεταξύ του z και του y , που συμβολίζεται ως $r=d(z,y)$. Υπάρχουν πολλές επιλογές για τη συνάρτηση $\Delta(r)$, αλλά λαμβάνουμε υπόψη μόνο τις περιπτώσεις όπου η απώθηση είναι γραμμική.

$$\Delta(r) = \begin{cases} \alpha - \beta r, & \text{if } 0 \leq r < r_0 \\ 0, & \text{if } r \geq r_0 \end{cases} \quad (2.7)$$

όπου $\beta=\alpha/r_0$ και $0 \leq \alpha < r_0$.

Το σημείο y απωθείται στο σημείο u που δίνεται από τη σχέση:

$$u = u(y, z) = y + \Delta(r) \frac{y-z}{r} \quad (2.8)$$

Οι περιορισμοί στα α και β απενεργοποιούνται για δύο λόγους:

- Πρώτον εγγυώνται ότι η απόθεση δεν υπερβαίνει το r_0 και
- Δεύτερον γιατί αν το y' και το y'' είναι δύο σημεία τέτοια ώστε:

$$r' = d(z, y') < r'' = d(z, y'') \quad (2.9) \quad \text{τότε:} \quad \|u(y', z)\| < \|u(y'', z)\| \quad (2.10)$$

[59].

2.3 Τεχνικές τερματισμού

Οι αλγόριθμοι, έχουν σχεδιαστεί για να βρίσκουν το καθολικό ελάχιστο έπειτα από κάποιο αριθμό επαναλήψεων. Όμως, για να δοθεί μία ικανοποιητική λύση από τον αλγόριθμο, θα πρέπει ο αλγόριθμος να τερματίζει μετά από κάποιο χρονικό διάστημα με βάση κάποια κριτήρια. Τα γενικότερα κριτήρια τερματισμού είναι:

- **Καθορισμένο πλήθος αποτιμήσεων της συνάρτησης:**

Στο κριτήριο αυτό, ο αλγόριθμος τερματίζει όταν έχει πραγματοποιήσει ένα προκαθορισμένο αριθμό αποτιμήσεων της αντικειμενικής συνάρτησης. Αυτό μπορεί να γίνει είτε γιατί έχουμε περιορισμένους πόρους και θέλουμε να περιορίσουμε τον αριθμό των αποτιμήσεων, είτε γιατί ο αλγόριθμος έχει συγκλίνει σε μια ικανοποιητική λύση μετά από αρκετές αποτιμήσεις.

- **Προκαθορισμένος αριθμός βημάτων του αλγορίθμου:**

Στο κριτήριο αυτό ο αλγόριθμος τερματίζει αφού εκτελέσει έναν προκαθορισμένο αριθμό βημάτων. Κάθε βήμα αντιστοιχεί σε μια αποτίμηση της αντικειμενικής συνάρτησης ή σε μια μεταβολή της λύσης. Αυτό το κριτήριο είναι συνήθως χρήσιμο για να περιορίσει τον χρόνο εκτέλεσης του αλγορίθμου και να διασφαλίσει ότι δεν εκτελείται για πάρα πολλά βήματα.

Τα μειονεκτήματα των δύο προηγούμενων τεχνικών τερματισμού είναι ότι μπορεί να είναι δύσκολο να προβλέψουμε πότε θα βρεθεί η βέλτιστη λύση. Αν καθορίσουμε ένα πεπερασμένο πλήθος αποτιμήσεων ή βημάτων, μπορεί να μην είναι αρκετό για να φτάσουμε στην καλύτερη λύση, ή αντίθετα, να είναι υπερβολικά πολύ, προκαλώντας ανεπιθύμητη κατανάλωση πόρων. Επίσης, αυτές οι τεχνικές δεν είναι πάντα προσαρμοστικές, δηλαδή δεν μπορούν να προσαρμοστούν αυτόματα στην εξέλιξη του αλγορίθμου ή την πολυπλοκότητα του προβλήματος.

Ένα από τα κυριότερα θέματα των στοχαστικών μεθόδων είναι τα κριτήρια τερματισμού τους. Οι μέθοδοι αυτής της κατηγορίας προσπαθούν να εντοπίσουν το καθολικό ελάχιστο

μετά από έναν ορισμένο αριθμό επαναλήψεων. Προκειμένου να υπάρξει επιτυχής εκτέλεση του αλγορίθμου δίνοντας αποτελέσματα, έχουν αναπτυχθεί διάφοροι κανόνες τερματισμού, που στόχο έχουν τη διατήρηση της πιθανότητας του επιτυχούς αποτελέσματος του αλγορίθμου, αλλά και τη μείωση του υπολογιστικού κόστους. Με βάση τη βιβλιογραφία, έχουν αναπτυχθεί διάφοροι κανόνες όπως τον Zielinski [60], Betro et al [61], Rinnooy Kan [62], Hart [63].

Τα κριτήρια των Rinnooy Kan, Boender και Zielinski

Με βάση το κριτήριο τερματισμού του Kan, εάν για την εύρεση w ελαχίστων έχουν γίνει t τοπικές αναζητήσεις, τότε μια εκτίμηση για το πλήθος των τοπικών ελαχίστων της συνάρτησης περιγράφεται από την ακόλουθη σχέση:

$$\hat{w} = \frac{w(t-1)}{t-w-2} \quad (2.11)$$

Σε πολυδειγματικές μεθόδους καθολικής ελαχιστοποίησης, όπου πραγματοποιούνται επαναλαμβανόμενες τυχαίες διαδικασίες και τα αποτελέσματα από τα δείγματα συνθέτουν μια εκτίμηση του προβλήματος που μελετούμε, το t στην παραπάνω σχέση αντικαθίστανται από το σύνολο των δειγματοληψιών που έχει πραγματοποιήσει ο αλγόριθμος. Συγκεκριμένα διακόπτεται όταν ισχύει: $\hat{w} - w < \frac{1}{2}$ (2.12)

και κάνοντας ορισμένες πράξεις ανισότητας, ισχύει ότι: $t > 2w^2 + 3w + 2$ (2.13).

Από την ανίσωση προκύπτει ότι ο αριθμός των τοπικών ελαχιστοποιήσεων θα πρέπει να είναι ανάλογος του τετραγώνου των ελαχίστων που έχουν βρεθεί, ώστε να σταματήσει ο αλγόριθμος. Για παράδειγμα αν έχουν βρεθεί 1000 ελάχιστα, θα απαιτηθούν 10^6 επιπλέον σημεία κατά την διάρκεια των πρόσθετων τοπικών ελαχιστοποιήσεων.

Ο κανόνας του διπλάσιου κουτιού (double box)

Τα τμήματα του χώρου αναζήτησεως που δεν έχουν καλυφθεί από τις τοπικές ελαχιστοποιήσεις που έχουν ανακαλυφθεί είναι πολύ σημαντικό να τα γνωρίζουμε. Στην περίπτωση που το τμήμα αυτό είναι μικρό, τότε η μέθοδος καθολικής ελαχιστοποίησης μπορεί να τερματιστεί. Στο ενδεχόμενο που το ποσοστό της περιοχής αναζήτησεως δεν έχει καλυφθεί υπολογίζεται από την ποσότητα [64]:

$$U = 1 - \sum_{i=1}^w \frac{m(A_i)}{m(S)} \quad (2.14)$$

όπου:

- w , είναι ο αριθμός των τοπικών ελαχίστων που έχουν ανακαλυφθεί
- $\frac{m(A_i)}{m(S)}$, προσεγγίζεται ασυμπτωτικά από τον λόγο $\frac{t_i}{T}$
- $\frac{t_i}{T}$, όπου:
 - t_i , το πλήθος των εφαρμογών της μεθόδου τοπικής ελαχιστοποίησης που οδήγησαν στο τοπικό ελάχιστο x_i^* και
 - T , ο συνολικός αριθμός εφαρμογών της τοπικής ελαχιστοποίησης.

Συμπερασματικά, η ποσότητα U δίνεται από τον τύπο [64]:

$$\hat{U} = 1 - \sum_{i=1}^w \frac{t_i}{T} \quad (2.15)$$

Επειδή σε κάθε επανάληψη η ποσότητα \hat{U} έχει τιμή ίση με το μηδέν και $\sum_{i=1}^w t_i = T$, η εξίσωση (2.15) δεν μπορεί να χρησιμοποιηθεί για την προσέγγιση του τμήματος του χώρου αναζητήσεως που δεν έχει ακόμα καλυφθεί. Για το λόγο αυτό, χρειάζεται ένας μεγαλύτερος χώρος αναζητήσεως S_2 που περιέχει τον αρχικό χώρο S , έτσι ώστε να ισχύει:

$$(S_2) = 2 \times m(S) \quad (2.16)$$

και έναν μοναδικό (και ψεύτικο) τοπικό ελάχιστο που θεωρείται ότι υπάρχει στην περιοχή $A_0 = S_2 - S$. Στην περίπτωση αυτή, ο όγκος της περιοχής προσελκύσεως για το ελάχιστο είναι: $m(A_0) = m(S)$. Επομένως ισχύει ότι [64]:

$$U = 1 - \sum_{i=1}^w \frac{m(A_i)}{m(S)} \rightarrow$$

$$U = 1 - \sum_{i=1}^w \frac{m(A_i)}{\frac{1}{2}m(S_2)} \rightarrow$$

$$U = 1 - \frac{\sum_{i=1}^w 2m(A_i)}{m(S_2)} \rightarrow$$

$$U = 1 - 2 \frac{\sum_{i=1}^w m(A_i)}{m(S_2)} \quad (2.17)$$

Στην εξίσωση (2.17) η ποσότητα $\frac{m(A_i)}{m(S_2)}$, προσεγγίζεται από την ποσότητα $\frac{t_i}{T}$, όπου η ποσότητα T είναι ο συνολικός αριθμός εφαρμογών της μεθόδου τοπικής αναζήτησεως στον χώρο S_2 . Οπότε η ποσότητα \hat{U} δίνεται από την εξίσωση [64]:

$$\hat{U} = 1 - 2 \sum_{i=1}^w \frac{t_i}{T} \quad (2.18)$$

Σε κάθε επανάληψη, N σημεία συλλέγονται ομοιόμορφα τυχαία στο χώρο S_2 και απορρίπτονται σημεία που δεν ανήκουν στο χώρο S. Έστω ότι μετά από k επαναλήψεις ο συνολικός αριθμός των σημείων που έχουν ληφθεί στο χώρο S_2 είναι M_k και από αυτά τα kN βρίσκονται στο χώρο S. Η ποσότητα $\delta_k = \frac{kN}{M_k}$ έχει μια μέση τιμή $\langle \delta \rangle_k = \frac{1}{k} \sum_{i=1}^k \delta_k$, η οποία τείνει στην ποσότητα $\frac{m(S)}{m(S_2)} = \frac{1}{2}$ καθώς ο μετρητής επαναλήψεων k τείνει στο άπειρο. Η διακύμανση της ίδιας ποσότητας δίνεται από τον τύπο [64]:

$$\sigma_k^2(\delta) = \langle \delta^2 \rangle_k - \langle \delta \rangle_k^2 \quad (2.19)$$

η οποία τείνει στο μηδέν καθώς το k τείνει στο άπειρο και είναι περισσότερο ομαλά μεταβαλλόμενη από την μέση τιμή, οπότε και πιο κατάλληλη για να χρησιμοποιηθεί για τον σχηματισμό ενός κριτηρίου τερματισμού. Η καθολική μέθοδος ελαχιστοποίησης τερματίζεται όταν ισχύουν οι συνθήκες [64]:

- $\left| \langle \delta \rangle_k - \frac{1}{2} \right| \leq \sigma_k(\delta)$
- $\sigma_k(\delta) \leq p \sigma_{last}^2(\delta)$ (2.20)

όπου:

- $\sigma_{last}^2(\delta)$ η τιμή της διακυμάνσεως της ποσότητας δ_k στην επανάληψη που βρέθηκε το τελευταίο τοπικό ελάχιστο και
- p: είναι μία ποσότητα στο διάστημα (0,1) και χρησιμοποιείται σαν ένας συμβιβασμός ανάμεσα σε μια εξαντλητική αναζήτηση ($p \rightarrow 0$) και σε μια αναζήτηση βελτιστοποιημένη για ταχύτητα ($p \rightarrow 1$). Η προτεινόμενη λύση γι' αυτήν την παράμετρο είναι η τιμή 0.5.

Ο αλγόριθμος της μεθόδου του διπλάσιου κουτιού και με τη χρήση της μεθόδου Multistart είναι ο εξής [64]:

Βήμα 1^ο :

Αρχικά θέτουμε $a=0$

Βήμα 2^ο :

Δειγματοληψία N σημείων με την διαδικασία που διατυπώθηκε προηγουμένως.

Βήμα 3^ο :

Υπολογισμός της ποσότητας: $\sigma^2(\delta)$.

Βήμα 4^ο :

Εκτέλεση μιας επανάληψης της διαδικασίας καθολικής ελαχιστοποίησης χρησιμοποιώντας τα σημεία που ελήφθησαν προηγουμένως. Ανάλογα με τη χρησιμοποιούμενη μέθοδο (Multistart, TMLSL κτλ.), αυτά τα σημεία μπορούν να χρησιμοποιηθούν σαν σημεία εκκινήσεως τοπικής ελαχιστοποίησης ή να εισαχθούν σε κάποια ομάδα.

Βήμα 5^ο :

Αν βρεθεί τουλάχιστον ένα νέο ελάχιστο, θέσε $a = p\sigma^2(\delta)$ και μετάβαση στο βήμα 2.

Βήμα 6^ο :

Τερματισμός αν $\left| \langle \delta \rangle - \frac{1}{2} \right| \leq \sigma_k(\delta)$ και $\sigma^2(\delta) \leq a$, διαφορετικά μετάβαση στο βήμα 2.

Ο κανόνας της ομοιογένειας:

Αυτή η τεχνική τερματισμού φαίνεται να αποδίδει καλύτερα σε μια μεγάλη γκάμα μεθόδων καθολικής και τοπικής βελτιστοποίησης. Ο τερματισμός μπορεί να επιτευχθεί με πολλούς τρόπους [65, 66].

Ομοιογένεια Ποσοτήτων:

- Αποθήκευση των ελαχίστων σε ένα πίνακα καταλληλότητας
- Υπολογισμός του αθροίσματος S των τιμών του πίνακα σε κάθε επανάληψη
- Έλεγχος της διαφοράς των ποσοτήτων S σε n επαναλήψεις με κάποια πολύ μικρή τιμή

Ισχύει ότι:

$$\delta^{(t)} = \left| \sum_{i=1}^{NP} |f_i^{(t)}| - \sum_{i=1}^{NP} |f_i^{(t-1)}| \right| \quad (2.21)$$

όπου:

NP: ο πληθυσμός

t: οι γενιές

Ο κανόνας τερματισμού τερματίζεται εάν $\delta^{(t)} \leq \epsilon$ (όπου ϵ ένας πολύ μικρός αριθμός) για έναν προκαθορισμένο αριθμό επαναλήψεων [40, 68].

Ομοιογένεια Ποσοτήτων στο τετράγωνο:

Η ομοιογένεια ποσοτήτων στο τετράγωνο, διαφέρει από την ομοιογένεια ποσοτήτων στον υπολογισμό της διαφοράς των ποσοτήτων, όπου είναι εις το τετράγωνο. Ισχύει η σχέση:

$$\delta^{(t)} = \left| \left(\sum_{i=1}^{NP} |f_i^{(t)}| \right)^2 - \left(\sum_{i=1}^{NP} |f_i^{(t-1)}| \right)^2 \right| \quad (2.22)$$

Ο κανόνας τερματισμού τερματίζεται, εάν $\delta^{(t)} \leq \epsilon$ για έναν προκαθορισμένο αριθμό των M επαναλήψεων [40, 68].

Ομοιογένεια ελαχίστων:

Σε αυτόν των κανόνα τερματισμού υπολογίζεται το καλύτερο ελάχιστο σε κάθε επανάληψη και μετά ελέγχεται εάν παραμένει ίδιο για n επαναλήψεις. Ισχύει ότι:

$$\left| f_{min}^{(k)} - f_{min}^{(k-1)} \right| \quad (2.23) \text{ και}$$

$$\alpha \leq \epsilon \quad (2.24),$$

όπου ϵ μία πολύ μικρή θετική τιμή, όπως για παράδειγμα $\epsilon = 10^{-3}$

Αν αυτή η διαφορά είναι μηδέν για έναν προκαθορισμένο αριθμό επαναλήψεων kmax, τότε η μέθοδος τερματίζεται [40, 68].

Μεικτή ομοιογένεια:

Αυτός ο κανόνας τερματισμού είναι ο συνδυασμός των παραπάνω τύπων ομοιογένειας. Ο τερματισμός της βελτιστοποίησης επιτυγχάνεται όταν επαληθεύεται οποιαδήποτε συνθήκη ομοιογένειας με συνέπεια βελτίωση της ταχύτητας. Δεν προτείνεται για οποιαδήποτε μέθοδο βελτιστοποίησης καθώς μπορεί να γίνει πρόωρος τερματισμός. Επομένως ισχύουν οι ακόλουθες δύο σχέσεις [40, 68]:

- $|f_{min}^{(k)} - f_{min}^{(k-1)}|$ (2.25)

- $\delta^{(t)} = \left| \sum_{i=1}^{NP} |f_i^{(t)}| - \sum_{i=1}^{NP} |f_i^{(t-1)}| \right|$ (2.26)

με συνθήκη τερματισμού:

$$\alpha \leq \epsilon \quad (2.27)$$

όπου ϵ μία πολύ μικρή θετική τιμή.

2.4 Τοπογραφικές τεχνικές

Η μέθοδος MLSL

Η μέθοδος Multilevel Single Linkage Clustering (MLSL) προσπαθεί να εντοπίσει όλα τα τοπικά ελάχιστα στο χώρο αναζήτησης, ξεκινώντας τοπική βελτιστοποίηση από επιλεγμένα σημεία του χώρου. Σύμφωνα με τον αλγόριθμο, τα βήματα που υλοποιούνται περιγράφονται ακολούθως [57]:

Αρχικοποίηση:

-Ορίζεται το πλήθος των σημείων «M» που θα λαμβάνονται τυχαία αλλά ομοιόμορφα σε κάθε επανάληψη.

- Καθορίζεται ένας αριθμός $\gamma \in [0,1]$ ο οποίος ορίζει το ποσοστό των σημείων που θα διατηρούνται στο δείγμα σε κάθε επανάληψη.

- Δημιουργείται σύνολο X^* στο οποίο θα αποθηκεύονται τα τοπικά ελάχιστα της κάθε επανάληψης. Στην αρχή το σύνολο αυτό θα είναι κενό.

- Αρχικοποίηση μετρητή επαναλήψεων με $k=0$.

Βήμα 1^ο :

Γίνεται τυχαία επιλογή των σημείων M και προστίθενται στο σύνολο των δειγμάτων.

Βήμα 2^ο :

Δημιουργία ενός συνόλου δειγμάτων με τη χρήση των σημείων $\gamma\%$ από το αρχικό δείγμα, με την χαμηλότερη συναρτησιακή τιμή.

Βήμα 3^ο :

-Εκκίνηση της μεθόδου τοπικής ελαχιστοποίησης από κάθε σημείο x_i στο νέο δείγμα, εκτός και εάν υπάρχει σημείο x_j στο δείγμα τέτοιο ώστε:

$$f(x_i) < f(x_j) \text{ και } \|x_i - x_j\| < r_k \quad (2.28)$$

Η απόσταση r_k δίνεται από τη σχέση:

$$r_k = \pi^{1/2} (\sigma m(S) \Gamma(1 + \frac{n}{2}) \frac{\log k}{k})^{\frac{1}{n}} \quad (2.29)$$

όπου:

- n , είναι η διάσταση της αντικειμενικής συναρτήσεως
- σ , είναι μία σταθερά με συνήθεις τιμές 2,4 κτλ.
- $m(S)$, είναι το μέτρο Lebesgue του χώρου αναζήτησης S
- Γ , είναι η συνάρτηση $\Gamma(n)$ που ορίζεται από τον τύπο:

$$\Gamma(n) = \int_0^{\infty} t^{n-1} \exp(-t) dt \quad (2.30)$$

-Προσθήκη των νέων τοπικών ελαχίστων που θα βρεθούν στο X^* .

Βήμα 4^ο :

Το κριτήριο τερματισμού που χρησιμοποιείται στη μέθοδο MLSL είναι του Rinnooy Kan. Σύμφωνα με το κριτήριο αυτό, σε κάθε επανάληψη k εκτιμάται η ποσότητα:

$$\hat{w} = \frac{w(kN-1)}{kN-w-2} \quad (2.31)$$

όπου w ο αριθμός των τοπικών ελαχίστων που έχουν βρεθεί από τη μέθοδο, ενώ ο αλγόριθμος τερματίζει όταν:

$$\hat{w} \leq w + \frac{1}{2} \quad (2.32)$$

Μόλις η μέθοδος τερματίσει, επιστρέφει το σημείο y με την χαμηλότερη συναρτησιακή τιμή σαν το ολικό ελάχιστο της αντικειμενικής συνάρτησης, στο σύνολο X^* [57].

Η μέθοδος TMLSL

Η μέθοδος TMLSL είναι παρόμοια με την μέθοδο MLSL, με τη διαφορά ότι για να αποφασίσει αν ένα σημείο πρέπει να ενταχθεί σε κάποια ομάδα, κατασκευάζει έναν τοπολογικό γράφο και βρίσκει τα τοπολογικά ελάχιστα που υπάρχουν σε αυτόν. Σύμφωνα με τον αλγόριθμο, τα βήματα που υλοποιούνται περιγράφονται ακολούθως [57]:

Αρχικοποίηση:

- Καθορισμός του αριθμού N των δειγμάτων που θα λαμβάνονται σε κάθε επανάληψη.
- Καθορισμός της παραμέτρου g ($g < N$), του αριθμού των κοντινότερων γειτόνων για την εύρεση των τοπογραφικών ελαχίστων.
- $X^* = \emptyset$, το σύνολο με τα τοπικά ελάχιστα που ανακαλύπτει ο αλγόριθμος.
- $T = \emptyset$, όπου T είναι το σύνολο των δειγμάτων που θα λαμβάνονται τυχαία σε κάθε επανάληψη.
- $k=0$, ο μετρητής των επαναλήψεων [57].

Δειματοληψία:

- $T = \emptyset$
- $k=k+1$
- Ενημέρωση της κρίσιμης κατάστασης r_k
- Για $i=1, \dots, N$ επανέλαβε
 - τυχαία δειματοληψία ενός σημείου $x_i \in S$.
 - $f_i = f(x_i)$
 - $T = T \cup (x_i, f_i)$
- Τέλος Επαναλήψεων [57]

Δημιουργία γράφου:

- Δημιουργία ενός τοπολογικού γράφου G με τη χρήση των σημείων T και X^* από τα σύνολα.
 - Εύρεση των τοπογραφικών ελαχίστων στον γράφο G .
- Ο υπολογισμός τοπογραφικών ελαχίστων γίνεται ως εξής:

- Για κάθε σημείο x βρίσκονται οι κοντινότεροι γείτονες (g), με τη χρήση της Ευκλείδειας απόστασης.
- Το σημείο αυτό θεωρείται τοπογραφικό ελάχιστο αν έχει χαμηλότερη συναρτησιακή τιμή σε σχέση με όλους τους υπόλοιπους γείτονες. Στην περίπτωση που το σημείο x έχει θεωρηθεί τοπογραφικό ελάχιστο, θεωρείται σαν σημείο έναρξης της τοπικής μεθόδου ελαχιστοποίησης αν και μόνο αν δεν υπάρχει σε ακτίνα μικρότερη από r_k κάποιο άλλο σημείο y , τέτοιο ώστε $f(y) \leq f(x)$ [57].
- Η απόσταση r_k υπολογίζεται ως εξής:

$$r_k = \pi^{-1/2} (\Gamma(1 + \frac{n}{2}) m(S) \sigma \frac{\log kN}{kN})^{\frac{1}{n}} \quad (2.33)$$

(Οι παράμετροι της παραπάνω σχέσης είναι ίδιοι με της μεθόδου MLSL)

-Κάθε τοπογραφικό ελάχιστο θεωρείται σημείο έναρξης της μεθόδου τοπικής ελαχιστοποίησης, στην περίπτωση που δεν υπάρχει άλλο σημείο σε απόσταση μικρότερη του r_k που να είναι χαμηλότερη συναρτησιακή τιμή από αυτό.

-Εκτέλεση μεθόδου τοπικής βελτιστοποίησης από κάθε σημείο που θεωρείται σημείο έναρξης και προσθήκη των τοπικών ελαχίστων που θα βρεθούν στο σύνολο ελαχίστων X^* [57].

Έλεγχος:

Το κριτήριο τερματισμού που χρησιμοποιείται στη μέθοδο TMLSL είναι ίδιο με αυτό της μεθόδου MLSL που προαναφέρθηκε: κριτήριο του Rinnooy Kan.

Αν το κριτήριο τερματισμού ισχύει, η μέθοδος τερματίζεται. Διαφορετικά μεταβαίνει στο βήμα της δειγματοληψίας [57].

Η μέθοδος MinFinder

Η μέθοδος MinFinder αποτελεί ένα εργαλείο βελτιστοποίησης που βοηθά τους ερευνητές να βρίσκουν τις βέλτιστες λύσεις σε προβλήματα που απαιτούν τη μείωση της συνάρτησης σε ελάχιστη τιμή. Με τη χρήση της MinFinder, επιτυγχάνεται η εξοικονόμηση πόρων και η επίτευξη αποδοτικών λύσεων. Ο σκοπός της MinFinder είναι να βρίσκει όλα τα ελάχιστα

μιας συνάρτησης καθώς και το ελάχιστο (τοπικό ή καθολικό) μιας συνάρτησης βελτιστοποίησης.

Η μέθοδος αυτή για να αποφασίσει αν ένα σημείο μπορεί να είναι σημείο εκκινήσεως μεθόδου τοπικής ελαχιστοποίησης, χρησιμοποιεί δύο φάσεις. Στην πρώτη φάση ελέγχεται αν το σημείο βρίσκεται στην περιοχή προσελκύσεως κάποιου ήδη ευρεθέντος ελαχίστου. Στην περίπτωση που δεν ισχύει η υπόθεση αυτή, τότε το σημείο ελέγχεται αν βρίσκεται στην ίδια περιοχή προσελκύσεως με κάποιο από τα υπόλοιπα σημεία του δείγματος. Αν καμία από τις παραπάνω υποθέσεις δεν ισχύει, τότε το σημείο μπορεί να θεωρηθεί ως σημείο ενάρξεως μεθόδου τοπικής ελαχιστοποίησης. Θα πρέπει να αναφερθεί ότι και στις δύο περιπτώσεις χρησιμοποιούνται διαφορετικές τυπικές αποστάσεις. Σύμφωνα με τον αλγόριθμο, τα βήματα που υλοποιούνται περιγράφονται ακολούθως [57]:

Αρχικοποίηση:

- Καθορισμός της παραμέτρου N , αριθμός των δειγμάτων που θα λαμβάνονται σε κάθε επανάληψη.
- Δημιουργία του συνόλου X^* στο οποίο τοποθετούνται τα τοπικά ελάχιστα που ανακαλύπτει ο αλγόριθμος. Αρχικά ισχύει ότι: $X^* = \emptyset$.
- $T = \emptyset$, όπου T είναι ένα σύνολο στο οποίο τοποθετούνται δείγματα της αντικειμενικής συναρτήσεως, όπως αυτά λαμβάνονται σε κάθε επανάληψη του αλγορίθμου.
- $V = \emptyset$, όπου V περιέχει σε κάθε επανάληψη τα υποψήφια σημεία εκκινήσεως για τοπική ελαχιστοποίηση.

Βήμα 1^ο :

- **Δειγματοληψία** N σημείων από τον χώρο αναζήτησεως με τυχαίο τρόπο, χρησιμοποιώντας το σχήμα του διπλάσιου κουτού και προσθήκη των αυτών των σημείων στο σύνολο T .
- **Για κάθε $x \in T$ Επανέλαβε**
 - Αν το x δεν είναι απορριπτέο τότε
 - * Θέσε $V = V \cup x$
 - Τέλος A_n
- **Τέλος επανάληψης**

Βήμα 2^ο :

-Αν $\frac{|V|}{N} < \frac{1}{2}$ Τότε

- $N = \min(N + \frac{N}{10}, 100)$. Αυτή η αλλαγή στον αριθμό των δειγμάτων που λαμβάνει ο αλγόριθμος γίνεται με σκοπό να αναγκαστεί ο αλγόριθμος να κάνει καλύτερη έρευνα στο χώρο S, γιατί όπως παρατηρήθηκε πειραματικά ο αλγόριθμος τείνει να απορρίψει πολλά από τα δείγματα που λαμβάνει σε κάθε επανάληψη, στερώντας με αυτό το τρόπο να κάνει καλύτερη έρευνα.

- Τέλος Αν

Βήμα 3^ο:

-Για κάθε $x \in V$ επανέλαβε

- Αν x δεν θεωρείται απορριπτέο σημείο τότε
 - ***Εκκίνηση** μιας μεθόδου τοπικής αναζήτησεως $y = L(x)$, όπου y είναι το εντοπισμένο από τη μέθοδο τοπικό ελάχιστο.
 - ***Υπολογισμός** της τυπικής αποστάσεως r_t κάνοντας χρήση της εξισώσεως:

$$r_t = \frac{1}{M} \sum_{i=1}^M |x_i - L(x_i)|$$

Αν $y \notin X^$ τότε

$$\Thetaέσε X^* = X^* \cup y$$

*Τέλος Αν

- Τέλος Αν

- Τέλος επανάληψης

- Αν το κριτήριο τερματισμού του διπλάσιου κουτιού ικανοποιείται, τότε

- Διακοπή του αλγορίθμου

- Αλλιώς

- Μετάβαση στο Βήμα 1

Στον παραπάνω αλγόριθμο για να θεωρηθεί ένα σημείο x ως απορριπτέο, θα πρέπει να ισχύει τουλάχιστον μία από τις επόμενες συνθήκες:

1. Το σημείο x θα πρέπει να είναι αρκετά κοντά σε κάποιο τοπικό ελάχιστο που έχει ήδη βρεθεί, έστω το σημείο z , και το κριτήριο της παραγώγου να ισχύει ανάμεσα στα δύο αυτά σημεία. Δηλαδή:

$$(x - z)^T (\nabla f(z)) < 0$$

Θεωρούμε πως το σημείο x είναι κοντά στο τοπικό ελάχιστο z αν ισχύει η σχέση:

$$|x - z| < \min_{i,j \ i \neq j} |z_i - z_j|, z_i \in X^*, z_j \in X^*$$

2. Το σημείο x να είναι κοντά σε ένα άλλο σημείο y με $y \in V$ και το κριτήριο της παραγώγου ανάμεσα στα δύο αυτά σημεία να ισχύει. Θεωρούμε πως τα δύο αυτά σημεία θα είναι κοντά αν ισχύει:

$$|x - y| < r_t \quad [14].$$

2.5 Η μέθοδος MinCenter

Η μέθοδος MinCenter βασίζεται στη καθολική βελτιστοποίηση Multistart. Η μέθοδος πολλαπλών εκκινήσεων ξεκινάει έναν βελτιστοποιητή τοπικής αναζήτησης από διαφορετικά τυχαία σημεία και αποδίδει το χαμηλότερο ανακαλυφθέν καθολικό ελάχιστο. Τα κύρια βήματα της διαδικασίας πολλαπλών εκκινήσεων παρουσιάζονται παρακάτω [68]:

1. Βήμα Αρχικοποίησης

- i. Όρισε ως M τον συνολικό αριθμό δειγμάτων.
- ii. Όρισε (x^*, y^*) ως το τοπικό ελάχιστο. Αρχικοποίηση του y^* σε μια μεγάλη τιμή.

2. Βήμα Δειγματοληψίας

- i. Για $i = 1 \dots M$ επανάλαβε
 - Δείγμα σημείου $x_i \in S$
 - $y_i = LS(x_i)$, όπου το $LS(x)$ είναι μια τοπική διαδικασία αναζήτησης.
 - Εάν $y_i \leq y^*$ τότε $x^* = x_i, y^* = y_i$
- ii. Τέλος επανάληψης

Η νέα μέθοδος MinCenter, αντικαθιστά το βήμα της δειγματοληψίας της Multistart με τη χρήση κεντροειδών που κατασκευάζονται με ομαδοποίηση K-means. Τα βήματα του αλγορίθμου K-means παρουσιάζονται παρακάτω [68]:

1. Επανάλαβε

a. $S_j = \{ \}, j = 1..K$

b. Για κάθε δείγμα x_i επανάλαβε

- Όρισε $j^* = \min_{i=1}^K \{D(x_i, c_j)\}$, όπου j^* είναι το κοντινότερο κέντρο για το δείγμα x_i .
- Όρισε $S_{j^*} = S_{j^*} \cup \{x_i\}$

c. Τέλος επανάληψης

d. Για κάθε κέντρο c_j επανάλαβε

- Όρισε $M_j =$ ο αριθμός των στοιχείων στο S_j
- Ενημέρωσε c_j

$$c_j = \frac{1}{M_j} \sum_{i=1}^{M_j} x_i$$

e. Τέλος επανάληψης

2. Τερματισμός όταν c_j δεν αλλάζει πλέον.

Τα εκτιμώμενα κεντροειδή, ενισχύονται μέσω της K-means και έτσι προστίθενται νέα δείγματα αλλά για προκαθορισμένο αριθμό επαναλήψεων. Αφού δημιουργηθούν τα κεντροειδή, εφαρμόζεται μία διαδικασία απόρριψης με σκοπό τη μείωση των κεντροειδών. Κατά τη διαδικασία απόρριψης, αφαιρούνται από το σύνολο των κέντρων, τα σημεία που έχουν πολλούς γείτονες σε μια προκαθορισμένη ακτίνα. Ο αλγόριθμος, κατά τον οποίο απορρίπτονται τα κεντροειδή, παρουσιάζεται ακολούθως [68]:

1. Όρισε ως C το σύνολο των κέντρων.
2. Όρισε $R = \emptyset$ το αποτέλεσμα του αλγορίθμου απόρριψης.
3. Όρισε $D_{\min} = \min_{i \neq j} \|c_i - c_j\|$
4. Όρισε $F > 1$, μία διπλή τιμή.
5. Όρισε $N_{\min} > 1$, έναν ακέραιο αριθμό.

6. Για κάθε κέντρο c_i επανάλαβε
 - a. Όρισε $N = 0$
 - b. Για κάθε κέντρο $c_j, i \neq j$ επανάλαβε
 - Εάν $\|c_i - c_j\| \leq FD_{min}$ τότε $N = N + 1$
 - c. Τέλος επανάληψης
 - d. Εάν $N < N_{min}$ τότε $R = R \cup c_i$
7. Τέλος επανάληψης
8. Επέστρεψε R

Με βάση τα παραπάνω, δημιουργήθηκε η νέα μέθοδος MinCenter, της οποίας ο αλγόριθμος, παρουσιάζεται παρακάτω [67]:

1. Βήμα αρχικοποίησης.
 - a. Όρισε το M ως τον αριθμό των δειγμάτων.
 - b. Όρισε (x^*, y^*) ως το τοπικό ελάχιστο. Αρχικοποίηση του y^* με μία μεγάλη τιμή.
 - c. Όρισε ως K τον αριθμό των ομάδων, όπου $K < M$.
 - d. Όρισε ως K_{MAX} τον αριθμό των επαναλήψεων για τον αλγόριθμο K-means.
 - e. Όρισε $C = \{\}$ ως το σύνολο των κατασκευασμένων κέντρων.
2. Βήμα κατασκευής.
 - a. Για $i = 1..K_{MAX}$ επανάλαβε.
 - Δείγμα M σημείων από την αντικειμενική συνάρτηση S
 $= \{x_1, x_2, \dots, x_M\}$
 - Ενημέρωση των κέντρων C με το σετ S , χρησιμοποιώντας K-means.
 - b. Τέλος επανάληψης
3. Δημιουργία του συνόλου R από το C χρησιμοποιώντας τον αλγόριθμο απόρριψης.
4. Βήμα αξιολόγησης.
 - a. Για $i = 1 \dots |R|$ επανάλαβε
 - Όρισε $x_i = R_i$
 - $y_i = LS(x_i)$. Where $LS(x)$ is a local search procedure.
 - Εάν $y_i \leq y^*$ τότε $x^* = x_i, y^* = y_i$

b. Τέλος επανάληψης

Πλεονεκτήματα – Μειονεκτήματα αλγορίθμου MinCenter

Στη μέθοδο βελτιστοποίησης "MinCenter", η εκκίνηση των τοπικών αναζητήσεων διεξάγεται με βάση αντιπροσώπους δειγμάτων αντί για τα δείγματα της αρχικής κατανομής. Αυτό έχει ως αποτέλεσμα να μειώνονται οι κλήσεις της αντικειμενικής συνάρτησης, η οποία αποτελεί σημαντικό παράγοντα ταχύτητας. Ωστόσο, η μέθοδος παρουσιάζει δύο μειονεκτήματα. Πρώτον, απαιτείται χρονοβόρος εντοπισμός των αρχικών αντιπροσώπων πριν από την έναρξη των τοπικών βελτιστοποιήσεων. Δεύτερον, πολλαπλές τοπικές αναζητήσεις μπορεί να οδηγήσουν σε ίδιες συναρτησιακές τιμές όπου ενδεχομένως να προέρχονται από την ίδια γειτονιά του ευκλείδειου χώρου. Για να αντιμετωπιστούν αυτά τα προβλήματα, υιοθετείται μια τεχνική όπου υπολογίζεται η ευκλείδεια απόσταση μεταξύ των κοντινών αντιπροσώπων και επιλέγεται μόνο ένας από αυτούς, με τους υπόλοιπους να απορρίπτονται. Με αυτόν τον τρόπο, επιτυγχάνεται επιπλέον βελτίωση της ταχύτητας του αλγορίθμου.

2.6 Πειραματικά αποτελέσματα

Στον Πίνακα 1: «Συγκριτικά αποτελέσματα των μεθόδων Multistart, TMLSL και GTC με τη χρήση του κριτηρίου τερματισμού του διπλού κουτιού» που ακολουθεί, παρουσιάζονται συγκριτικά αποτελέσματα ανάμεσα στις μεθόδους Multistart, TMLSL και GTC. Στην πρώτη στήλη αναφέρονται οι συναρτήσεις για κάθε μία μέθοδο, ενώ σε κάθε μέθοδο γίνεται σύγκριση των τιμών MIN, FC και GC όπου:

- MIN: περιέχει τον μέσο όρο των ελαχίστων που βρέθηκαν
- FC: περιέχει τον μέσο αριθμό συναρτησιακών κλήσεων που χρειάστηκαν
- GC: αναπαριστά τον αριθμό των κλήσεων προς την παράγωγο της συναρτήσεως

Πίνακας 1: Συγκριτικά αποτελέσματα των μεθόδων Multistart, TMLSL και GTC με τη χρήση του κριτηρίου τερματισμού του διπλού κουτιού.

	MULTISTART			TMLSL			GTC		
ΣΥΝΑΡΤΗΣΗ	MIN	FC	GC	MIN	FC	GC	MIN	FCX	GC

CAMEL	6	5503	5503	6	3415	2583	6	844	1705
RASTRIGIN	49	19593	19593	49	24123	12380	49	4449	5090
SHUBERT	400	577738	577738	400	358623	252033	400	31674	59044
HANSEN	527	612015	612015	527	811679	592733	527	82572	10920
GRIEWANK2	529	1765175	1765175	528	1929165	1726632	529	103244 5	1140113
GKLS(3,30)	29	302853	302853	28	191867	102373	28	75363	206788
GKLS(3,100)	97	7492103	7492103	91	6683608	4583920	97	416089 7	5663373
GKLS(4,100)	95	8629052	8629052	78	1226934 2	8237461	96	256448 0	3878680
GUILIN(10,200)	200	3351391	3351391	200	3470206	3234820	200	208442 5	2538444
GUILIN(20,100)	100	1906288	1906288	100	1915478	1902730	100	862284	866943
Test2N(4)	16	19424	19424	16	3821	1684	16	2403	3165
Test2N(5)	32	30607	30607	32	8953	6288	32	7294	19390
Test2N(6)	64	34840	34840	64	54555	24366	64	18928	38555
Test2N(7)	128	117953	117953	128	88750	45229	128	39344	115102
GOLDSTEIN	4	5391	5391	4	4606	3836	4	3433	3770
BRANIN	3	1856	1856	3	1007	614	3	804	969
HARTMAN3	3	3509	3509	3	2014	1715	3	2738	2875
HARTMAN6	2	3903	3903	2	912	663	2	2949	3088
SHEKEL5	5	22128	22128	5	6434	5358	5	6558	6946
SHEKEL7	7	30702	30702	6	14617	14617	7	8589	9140
SHEKEL10	10	36838	36838	9	18447	15628	10	20226	21597

Από τα πειράματα που πραγματοποιήθηκαν διαπιστώνεται ότι η μέθοδος GTC δεν διαθέτει τις αδυναμίες των τεχνικών ομαδοποίησης σε ότι αφορά την απόρριψη σημείων. Οπότε, με την τυπική απόσταση που χρησιμοποιεί αλλά και με το κριτήριο της παραγώγου, εμφανίζεται να είναι πιο αποδοτική από αυτές, καθώς απαιτεί μικρότερο χρόνο και μικρότερο αριθμό συναρτησιακών αποτιμήσεων.

Για την υλοποίηση της μεθόδου Minfinder, δημιουργήθηκε ένα λογισμικό σε ANSI C++ στο οποίο ο χρήστης μπορεί να κωδικοποιήσει την αντικειμενική συνάρτηση. Σε αυτό το λογισμικό, ο χρήστης παρέχει την αντικειμενική συνάρτηση και την παράγωγό της σε ένα αρχείο ορίζοντας μια σειρά από συναρτήσεις και μέσω ενός βοηθητικού εργαλείου, δημιουργείται το τελικό εκτελέσιμο αρχείο. Το τελικό αρχείο διαθέτει μια σειρά από παραμέτρους, που αφήνει το χρήστη να πειραματιστεί. Στον Πίνακα 2 «Αποτελέσματα της μεθόδου Minfinder με τη χρήση του κριτηρίου τερματισμού του διπλού κουτιού», παρουσιάζονται τα αποτελέσματα της μεθόδου Minfinder σε μια σειρά από συναρτήσεις δοκιμής με την χρήση του κριτηρίου τερματισμού του διπλού κουτιού.

Πίνακας 2: Αποτελέσματα της μεθόδου Minfinder με τη χρήση του κριτηρίου τερματισμού του διπλού κουτιού

	MINFINDER		
ΣΥΝΑΡΤΗΣΗ	MIN	FC	GC
CAMEL	6	1598	2187
RASTRIGIN	49	1723	2975
SHUBERT	400	17204	41849
HANSEN	527	60916	94382
GRIEWANK2	529	1035094	1190595
GKLS(3,30)	29	67541	105006
GKLS(3,100)	96	1871681	2905475
GKLS(4,100)	98	3763131	7677290
GUILIN(10,200)	200	1467845	1512665
GUILIN(20,100)	100	646213	668183
Test2N(4)	16	953	1612
Test2N(5)	32	1850	4827
Test2N(6)	64	5133	29252
Test2N(7)	128	29908	160805
GOLDSTEIN	4	2197	2364
BRANIN	3	498	604
HARTMAN3	3	1581	1737
HARTMAN6	2	1090	1194
SHEKEL5	5	7144	7365

SHEKEL7	7	17125	17377
SHEKEL10	10	21551	21661

Στη συνέχεια ακολουθεί ένας πίνακας με τα αποτελέσματα μιας νέας μεθόδου ομαδοποίησης με σκοπό την αντιμετώπιση της βελτιστοποίησης (βλ.: Πίνακας 1: «Αποτελέσματα της μεθόδου Multistart»). Πιο συγκεκριμένα, για κάθε σύμπλεγμα δημιουργείται ένας αντιπρόσωπος χρησιμοποιώντας τη μέθοδο Kmeans και έπειτα ο αριθμός των συμπλεγμάτων μειώνεται με τη διαδικασία απόρριψης. Η μέθοδος που προτείνεται έχει ορισμένους περιορισμούς όπως το γεγονός ότι στηρίζεται σε μεγάλο βαθμό από τον αλγόριθμο Kmeans και ότι η διαδικασία μπορεί να διαρκέσει πολύ σε προβλήματα υψηλότερης κλίμακας.

Πίνακας 3: Αποτελέσματα της μεθόδου Multistart

ΣΥΝΑΡΤΗΣΗ	M=100	M=200	M=100, Rejection
B2	4518	8849	4472
Easom	943	1949	933
Bf1	4508	9300	4469
Bf2	3750	7621	3666
Branin	1948	3855	1938
Camel	2669	4983	2502
CM4	5714	11783	5644
CM8	7341(0.33)	14813(0.60)	7289(0.33)
DIFFPOWER10	123729	248924	121012
ELP4	1203	2474	1158
ELP8	1721	3395	1652
ELP16	2789	5485	2252
EXP4	3646	7063	3609
EXP8	3723	7447	3651
EXP16	3835	7486	3310
GKLS250	1486	2928	1426
GKLS350	1030(0.97)	2007	913(0.87)
GKLS3100	1020(0.77)	2005	1018(0.77)

GRIEWANK2	3131(0.70)	6197(0.97)	3048(0.70)
GREWANK10	10449	20763	10226
HANSEN	2482	4997	2422
HARTMAN3	2911	5753	2868
HARTMAN6	3825	7875	3787
POTENTIAL3	5237	10784	5178
POTENTIAL5	11594	22331	10127
POTENTIAL10	20361	40592	5089(0.70)
RASTRIGIN	2345	4731	2242(0.93)
SHEKEL5	3852	7841	3730
SHEKEL7	3951	7149	3885
SHEKEL10	3982	6987	3890
SINU4	3317	6624	3246
SINU8	4883	10015	4791
SINU16	8731	17005	8692
TEST2n4	3258	6608	3216
TEST2n5	3565	7128	3534
TEST2n6	3804(0.90)	7790	3850(0.90)
TEST2n7	4203(0.83)	8501(0.97)	4155(0.77)
TOTAL	281454(0.93)	562038(0.98)	259160(0.92)

Πίνακας 4: Τα αποτελέσματα της προτεινόμενης μεθόδου με τη $K=100$ κέντρα

ΣΥΝΑΡΤΗΣΗ	M=100	M=200
B2	4073	3886
Easom	830	782
Bf1	4046	3864
Bf2	3346	3153
Branin	1699	1623
Camel	2338	2237
CM4	4434	4043
CM8	3084(0.63)	1819(0.50)

DIFFPOWER10	26726	17980
ELP4	971	908
ELP8	601	338
ELP16	139	100
EXP4	2764	2522
EXP8	1564	943
EXP16	245	179
GKLS250	1337	1275
GKLS350	911(0.93)	777(0.83)
GKLS3100	939(0.97)	796(0.97)
GRIEWANK2	2812(0.77)	2684(0.70)
GREWANK10	1812	1152(0.80)
HANSEN	2210	2077
HARTMAN3	2400	1993
HARTMAN6	2707	2369
POTENTIAL3	1246	714
POTENTIAL5	752	664
POTENTIAL10	1621(0.23)	1045(0.10)
RASTRIGIN	2016	1917
SHEKEL5	3520	3116
SHEKEL7	3515	3113
SHEKEL10	3586	3237
SINU4	2548	2268
SINU8	2121	1624
SINU16	546	358
TEST2n4	2436	2198
TEST2n5	2186(0.97)	1840(0.97)
TEST2n6	2648(0.80)	2300(0.83)
TEST2n7	2469(0.77)	2173(0.73)
TOTAL	103198(0.95)	84067(0.93)

3. Η ΜΕΘΟΔΟΣ CONTROLLED RANDOM SEARCH

3.1 Βασικές αρχές

Η μέθοδος Controlled Random Search (CRS) είναι ένας αλγόριθμος που βασίζεται σε έναν πληθυσμό από λύσεις και έχει εφαρμοστεί σε πολλά προβλήματα [41]. Ο αλγόριθμος CRS δεν έχει απαραίτητη προϋπόθεση η συνάρτηση να είναι παραγωγίσιμη. Ομοίως υπάρχουν και άλλοι αλγόριθμοι που δεν απαιτούν την παραγωγισιμότητα της συνάρτησης όπως είναι ο Γενετικός Αλγόριθμος (Genetic Algorithm -GA) [69], η Διαφορική Εξέλιξη (Differential Evolution -DE) [40], ο αλγόριθμος Nelder-Mead [18], ο αλγόριθμος Brent Method [21] και η μέθοδος Golden-section Method [24].

Ο αλγόριθμος Controlled Random Search (CRS) αποτελεί μια μέθοδο βελτιστοποίησης που εκμεταλλεύεται έναν πληθυσμό σημείων στον χώρο αναζήτησης με στόχο την προσέγγιση του βέλτιστου αποτελέσματος σε ένα πρόβλημα. Κάθε σημείο αντιπροσωπεύει μια πιθανή λύση, και η εκτίμησή του βασίζεται στην αντικειμενική συνάρτηση. Κατά τη διάρκεια κάθε επανάληψης, υπολογίζονται οι συντεταγμένες ενός νέου σημείου σε σχέση με τα υπόλοιπα. Στη συνέχεια, γίνεται έλεγχος για το αν το νέο σημείο έχει μικρότερη συναρτησιακή τιμή από το σημείο με την υψηλότερη τιμή στο σύνολο των δειγμάτων. Σε περίπτωση θετικής επιβεβαίωσης, το σημείο με την υψηλότερη τιμή αντικαθίσταται από το νέο σημείο. Η επιλογή του νέου σημείου βασίζεται στη συναρτησιακή τιμή, καθώς ο στόχος είναι η εύρεση ελαχίστου της αντικειμενικής συνάρτησης. Το νέο σημείο, εάν προσφέρει μειωμένη τιμή, συμβάλλει στην πιθανότητα εύρεσης βέλτιστης λύσης. Κατά την εξέλιξη, ο αλγόριθμος συγκλίνει προς περιοχές χαμηλότερων συναρτησιακών τιμών, αναζητώντας το βέλτιστο σημείο. Η αποτελεσματικότητα του αλγορίθμου εξαρτάται από παραμέτρους, όπως ο αριθμός των αρχικών σημείων και η πολυπλοκότητα της αντικειμενικής συνάρτησης. Ο αλγόριθμος επιδιώκει να εξερευνήσει τον χώρο αναζήτησης και να εντοπίσει πιθανές βέλτιστες λύσεις μεσολαβώντας συνεχή προσαρμογή των σημείων [70].

3.2 Η μέθοδος Controlled random search

Η μέθοδος Controlled Random Search (CRS) αποτελεί μια διαδικασία βελτιστοποίησης που χρησιμοποιείται για την εύρεση του ελαχίστου (ή μεγίστου) μιας συνάρτησης, χωρίς

την ανάγκη γνώσης των παραγώγων της. Ο αλγόριθμος CRS ξεκινά με έναν αρχικό πληθυσμό N σημείων στον χώρο αναζήτησης, όπου κάθε σημείο αντιστοιχεί σε μια πιθανή λύση. Κατά τη διάρκεια κάθε επανάληψης, υπολογίζονται οι συντεταγμένες ενός νέου σημείου σχετικό προς τα υπόλοιπα. Έπειτα, γίνεται έλεγχος για το αν το νέο σημείο έχει μικρότερη συναρτησιακή τιμή από το σημείο με την υψηλότερη τιμή στο σύνολο των δειγμάτων. Αν ισχύει, το σημείο με την υψηλότερη τιμή αντικαθίσταται από το νέο σημείο. Ο λόγος που επιλέγεται το νέο σημείο με βάση τη συναρτησιακή τιμή είναι η αναζήτηση του ελαχίστου της αντικειμενικής συνάρτησης. Ένα νέο σημείο με μικρότερη τιμή συνεισφέρει στην εύρεση βέλτιστης λύσης. Κατά την εξέλιξη, ο αλγόριθμος συγκλίνει προς τις περιοχές με χαμηλότερες συναρτησιακές τιμές, αναζητώντας το βέλτιστο σημείο. Με τη μέθοδο CRS επιδιώκεται η δημιουργία ενός αλγορίθμου βελτιστοποίησης, που να είναι αποδοτικός και ακριβής σε διάφορα προβλήματα βελτιστοποίησης, χωρίς την ανάγκη για παραγώγους της αντικειμενικής συνάρτησης. Συγκεκριμένα, οι σχεδιαστές της μεθόδου CRS επιδίωκαν:

- **Εξερεύνηση του χώρου αναζήτησης:** Η CRS έχει σαν στόχο να εξερευνήσει αποτελεσματικά τον χώρο των πιθανών λύσεων σε ένα πρόβλημα και με αυτό το τρόπο επιτυγχάνεται η αξιολόγηση όλων των σημείων του πληθυσμού.
- **Εύρεση βέλτιστων λύσεων:** Ο στόχος είναι η σύγκλιση προς τις περιοχές του χώρου αναζήτησης που περιέχουν τις ελάχιστες τιμές της αντικειμενικής συνάρτησης. Με τη συνεχή δημιουργία και αξιολόγηση νέων σημείων, ο πληθυσμός στρέφεται προς τη βέλτιστη λύση.
- **Προσαρμογή των σημείων:** Μέσω της επαναλαμβανόμενης διαδικασίας αξιολόγησης και αντικατάστασης σημείων, η μέθοδος προσαρμόζει τα σημεία του πληθυσμού με σκοπό τη σύγκλιση προς τις βέλτιστες λύσεις.

Ο αλγόριθμος της μεθόδου CRS παρουσιάζεται παρακάτω [57]:

Αρχικοποίηση:

- **Θέσε** την παράμετρο N που καθορίζει τα δείγματα της αντικειμενικής συνάρτησης που θα ληφθούν με την εκκίνηση του αλγορίθμου. Συνήθως δίνεται η τιμή $N=25n$, διότι ένας μεγάλος αριθμός αρχικών σημείων N μπορεί να σημαίνει ευρύτερη εξερεύνηση του χώρου αναζήτησης. Η σχέση $N=25n$ είναι σχεδιασμένη για να εξερευνήσει αποτελεσματικά τον χώρο αναζήτησης σε προβλήματα μεγάλης

διάστασης n . Σε προβλήματα όπου η διάσταση είναι μεγάλη, μια τέτοια σχέση μπορεί να είναι πιο αποτελεσματική.

- **Θέσε** στην παράμετρο ϵ που χρησιμοποιείται στο κριτήριο τερματισμού της μεθόδου, μια μικρή θετική τιμή.
- **Δημιουργία** του συνόλου $T = \{x_1, x_2, \dots, x_N\}$, λαμβάνοντας N τυχαία δείγματα στο χώρο S .

Έλεγχος:

- **Υπολόγισε** τα σημεία $x_{max} = \operatorname{argmax} f(x)$ και $x_{min} = \operatorname{argmin} f(x)$ με τις αντίστοιχες συναρτησιακές τους τιμές:

$$f_{max} = \max_{x \in T} f(x), \text{ όπου } x \in T \text{ και } f_{min} = \min_{x \in T} f(x), \text{ όπου } x \in T$$

- **Αν** $|f_{max} - f_{min}| < \epsilon$, μετάβαση στο βήμα **Ελαχιστοποίηση**.

Νέο σημείο:

- **Επιλογή** με τυχαίο τρόπο του συνόλου $\mathcal{P} = \{x_{T_1}, x_{T_2}, \dots, x_{T_{n+1}}\}$ από το αρχικό σύνολο T .
- **Υπολογισμός** του κεντροειδούς G από τη σχέση:

$$G = \frac{1}{n} \sum_{i=1}^n x_{T_i}$$

- **Δημιουργία** δοκιμαστικού σημείου: $x = 2G - x_{T_{n+1}}$
- **Αν** $x \notin S$ $f(x) \geq f_{max}$ τότε εκτέλεση και πάλι του βήματος **Νέο σημείο**.

Ενημέρωση:

- $T = T \cup \{x\} - \{x_{max}\}$
- **Μετάβαση** στο βήμα **Έλεγχος**.

Ελαχιστοποίηση:

- $x^* = \operatorname{localSearch}(x)$. Η έκφραση " $\operatorname{localSearch}(x)$ ", αναφέρεται σε μια διαδικασία τοπικής αναζήτησης που εκτελείται για το σημείο x με σκοπό την εύρεση μιας καλύτερης λύσης στο πλαίσιο του προβλήματος βελτιστοποίησης. Η ιδέα της τοπικής αναζήτησης είναι να βελτιωθεί η τρέχουσα λύση, εξετάζοντας τις γειτονικές της τιμές στο χώρο αναζήτησης.

- Επιστροφή του σημείου x^* σαν το εντοπισμένο τοπικό ελάχιστο.

3.3 Παραλλαγές της μεθόδου Controlled Random Search

Η μέθοδος Controlled Random Search (CRS) έχει εξελιχθεί και τροποποιηθεί σε διάφορες παραλλαγές για να ανταποκριθεί σε διάφορες απαιτήσεις και προκλήσεις της βελτιστοποίησης. Κάθε παραλλαγή προσφέρει βελτιώσεις στο αρχικό CRS, προσαρμόζοντας τη στρατηγική αντικατάστασης, τη διαδικασία αξιολόγησης ή τους κανόνες τερματισμού για να επιτύχει καλύτερη απόδοση σε διάφορα προβλήματα βελτιστοποίησης.

Μία παραλλαγή της μεθόδου CRS πραγματοποιήθηκε από τους Kaelo και Ali (2006) [71]. Πιο συγκεκριμένα, τροποποίησαν την αρχική μέθοδο CRS με τέσσερις διαφορετικούς τρόπους ορίζοντας τέσσερις νέες εκδόσεις της μεθόδου. Στην πρώτη έκδοση τροποποιήθηκε η διαδικασία δημιουργίας των δοκιμαστικών σημείων του αλγορίθμου CRS, εισάγοντας μια νέα τεχνική τοπικής μετάλλαξης. Συγκεκριμένα, όταν ένα δοκιμαστικό σημείο που παράγεται από τη μέθοδο CRS αποτυγχάνει να αντικαταστήσει το τρέχον χειρότερο σημείο, η τοπική μετάλλαξη δημιουργεί ένα δεύτερο δοκιμαστικό σημείο που εξερευνά την περιοχή γύρω από το τρέχον καλύτερο σημείο, αντανακλώντας το πρώτο δοκιμαστικό σημείο μέσω συντεταγμένων [71]. Στη δεύτερη παραλλαγή, επεκτείνεται η αρχική μέθοδος προσθέτοντας ένα σύστημα πιθανοτικής προσαρμογής για τη δημιουργία δοκιμαστικών σημείων. Αντί να χρησιμοποιεί σταθερές πιθανότητες, η μέθοδος προσαρμόζει δυναμικά αυτές τις πιθανότητες βάσει της απόδοσης των σχημάτων. Αρχικά, ο αλγόριθμος αναθέτει ίσες πιθανότητες στα δύο σχήματα και κατά τη διάρκεια της εκτέλεσης, οι πιθανότητες ενημερώνονται με βάση την απόδοση των σχημάτων. Το σχήμα που θα επιφέρει καλύτερα αποτελέσματα, λαμβάνει υψηλότερη πιθανότητα για τη δημιουργία νέων δοκιμαστικών σημείων. Αυτή η προσαρμογή στοχεύει στη βελτίωση της απόδοσης του αλγορίθμου ενώ προσαρμόζεται δυναμικά σε διαφορετικά προβλήματα βελτιστοποίησης. Η επιλογή μεταξύ των σχημάτων γίνεται πλέον με κατευθυνόμενο τρόπο, βασιζόμενο στην απόδοση τους σε κάθε επανάληψη [71]. Η τρίτη τροποποίηση εισάγει μία τοπική τεχνική η οποία ονομάζεται τοπική μετάλλαξη, παρέχοντας έναν πιο ανθεκτικό και αποδοτικό αλγόριθμο. Τέλος, η τέταρτη έκδοση δεν περιλαμβάνει τοπική

τεχνική, αλλά δημιουργεί δοκιμαστικά σημεία χρησιμοποιώντας μια πιθανοτική κατανομή [71].

Μία άλλη τροποποίηση της μεθόδου CRS, παρουσιάζεται στο άρθρο των Τσούλου και Λαγαρή (2005) [72] όπου παρουσιάζεται ο αλγόριθμος Genetically Controlled Random Search (GCRS). Ο αλγόριθμος αυτός συνδυάζει τη μέθοδο ελεγχόμενης τυχαίας αναζήτησης (CRS) με τη γενετική εξέλιξη, χρησιμοποιώντας τη γραμματική εξέλιξη για τη δημιουργία κώδικα σε οποιαδήποτε γλώσσα προγραμματισμού. Ο αλγόριθμος περιλαμβάνει τα βήματα αρχικοποίησης, τη διαδικασία εξέλιξης με χρήση γενετικών λειτουργιών, τη μέθοδο του τουρνουά για την επιλογή χρωμοσωμάτων, τις τεχνικές μετάλλαξης και γραμματικής εξέλιξης για τη δημιουργία δοκιμαστικών σημείων. Η μέθοδος GCRS συνδυάζει τη γενετική εξέλιξη με την τυχαία αναζήτηση για την εύρεση του καθολικού ελάχιστου ενός πολυδιάστατου χώρου στο πλαίσιο του αλγορίθμου CRS. Η διαδικασία ξεκινά με την τυχαία δημιουργία μιας αρχικής γενεάς σημείων και στη συνέχεια, επικεντρώνεται στη βελτιστοποίηση των λύσεων με τη χρήση γενετικών τεχνικών. Κατά την διάρκεια του αλγορίθμου, επιλέγονται και ενημερώνονται τα σημεία που παρουσιάζουν τις ελάχιστες και μέγιστες τιμές της αντικειμενικής συνάρτησης. Στη συνέχεια, παράγεται ένα νέο δοκιμαστικό σημείο και εφαρμόζεται η διαδικασία GRS, για να εξερευνηθεί τον χώρο των λύσεων γύρω από αυτό. Η εξέλιξη των γενετικών χαρακτηριστικών με διασταύρωση και μετάλλαξη συνεισφέρει στη βελτίωση των λύσεων. Σε περίπτωση που η διαφορά μεταξύ των ελάχιστων και μέγιστων τιμών είναι πολύ μικρή, η μέθοδος μεταβαίνει σε ένα βήμα τοπικής αναζήτησης για να εξασφαλίσει την εύρεση ακριβούς τοπικής ελαχίστου. Στο άρθρο παρουσιάζονται συγκρίσεις απόδοσης με άλλες μεθόδους βελτιστοποίησης, επιδεικνύοντας την αποτελεσματικότητα του αλγορίθμου [72].

3.4 Η τροποποιημένη μέθοδος Controlled Random Search

Στις τεχνικές βελτιστοποίησης μέχρι στιγμής, συνηθίζεται να χρησιμοποιείται ένας προκαθορισμένος αριθμός επαναλήψεων ως κανόνας τερματισμού της μεθόδου. Κάποιες φορές όμως ο κανόνας αυτός, απαιτεί μεγάλο αριθμό κλήσεων των συναρτήσεων. Γι' αυτό το λόγο εμπνευσμένοι από το άρθρο του Τσούλου (2008) [73], δημιουργήθηκε ο κανόνας τερματισμού που παρουσιάζεται ακολούθως:

Σε κάθε επανάληψη k , υπολογίζεται η διακύμανση $s(k)$ της ποσότητας f_{min} .

Εάν η τεχνική βελτιστοποίησης δεν καταφέρει να βρει μια νέα εκτίμηση του γενικού ελαχίστου για κάποιες επαναλήψεις, τότε είναι πιθανό να έχει ανακαλυφθεί το γενικό ελάχιστο.

Στην περίπτωση αυτή ο αλγόριθμος πρέπει να τερματιστεί. Οπότε ο αλγόριθμος τερματίζει όταν:

$$\sigma^{(k)} \leq \frac{\sigma(k_{last})}{2}$$

Ο όρος k_{last} αντιπροσωπεύει την τελευταία επανάληψη όπου εντοπίστηκε ένα νέο γενικό ελάχιστο [70].

Καθώς η μέθοδος βρίσκει μία χαμηλότερη εκτίμηση για το γενικό ελάχιστο είτε στην περίπτωση που το γενικό ελάχιστο έχει ήδη βρεθεί, μειώνεται συνεχώς με το χρόνο. Επειδή η ποσότητα αυτή είναι πάντα θετική, καθώς αντιπροσωπεύει τη διαφορά των τιμών της αντικειμενικής συνάρτησης μεταξύ δύο διαδοχικών εκτιμήσεων, μπορεί να χρησιμοποιηθεί ως ένα κριτήριο για να αναγνωρίσουμε πότε η μέθοδος προσεγγίζει το ελάχιστο. Όσο μικραίνει η διαφορά αυτή, τόσο πιο κοντά βρισκόμαστε στο ελάχιστο της συνάρτησης. Οπότε, αποτελεί μια καλή υποψήφια λύση για χρήση στα κριτήρια τερματισμού. Εάν το γενικό ελάχιστο έχει ήδη βρεθεί, η μέθοδος δεν είναι πλέον σε θέση να βρει μια νέα εκτίμηση και η ποσότητα θα τείνει προς το μηδέν. Στην περίπτωση αυτή όταν η ποσότητα πέσει κάτω από ένα κατώφλι, μπορούμε να διακόψουμε την εκτέλεση του αλγορίθμου. Η τιμή που μπορεί να ορισθεί ως κατώφλι, μπορεί να είναι ένα κλάσμα της τιμής της $s(k)$ από την τελευταία φορά που βρέθηκε μια νέα εκτίμηση για το γενικό ελάχιστο. Στην περίπτωση όμως που επιθυμούμε να επιτρέψουμε στον αλγόριθμο να συνεχίσει για πολλές γενιές, αυτό το κλάσμα μπορεί να είναι μικρό (πχ. 0.25). Εάν όμως θέλουμε να σταματήσει πιο άμεσα, μία εκτίμηση του κλάσματος θα μπορούσε να είναι το 0.75. Φυσικά, η τιμή 0,5 αποτελεί μία συμβιβαστική λύση μεταξύ των δύο προηγούμενων τιμών [70].

Επίσης, πραγματοποιήθηκε τροποποίηση της μεθόδου για τη δημιουργία πιο αποτελεσματικών δοκιμαστικών σημείων για τη μέθοδο ελεγχόμενης τυχαίας αναζήτησης. Η τροποποίηση της μεθόδου παρουσιάζεται παρακάτω [70]:

1. **Υπολόγισε το κεντροϊδούς G :**

$$G = \frac{1}{n} \sum_{i=1}^n zT_i$$

2. **Θέσε:** $G = G + \frac{1}{n} z_{min}$

3. **Υπολόγισε** ένα δοκιμαστικό σημείο $z = G - \frac{1}{n} zT_{n+1}$

Η τροποποίηση που εισήχθη στη μέθοδο προσαρμόζει τη δημιουργία δοκιμαστικών σημείων με στόχο τη βελτίωση της απόδοσης της μεθόδου σε προβλήματα βελτιστοποίησης, Συγκεκριμένα, προστίθεται ένα διάνυσμα z_{min} που αντιστοιχεί στο ελάχιστο σημείο, ενισχύοντας έτσι την διαδικασία να εστιάσει σε περιοχές που πιθανόν περιέχουν τις πραγματικές ελάχιστες τιμές της συνάρτησης. Επιπλέον, η χρήση ενός δοκιμαστικού σημείου z , υπολογισμένο με βάση τον κεντροϊδό : G και το zT_{n+1} , βοηθά στη δημιουργία αποτελεσματικότερων δοκιμαστικών σημείων. Η τροποποίηση αυτή στοχεύει στη γρηγορότερη σύγκλιση του αλγορίθμου και στην εξερεύνηση του χώρου αναζήτησης με μεγαλύτερη ακρίβεια.

3.5 Πειραματικά αποτελέσματα

Τα πειράματα που διεξήχθησαν, πραγματοποιήθηκαν τριάντα φορές, όπου κάθε φορά χρησιμοποιήθηκαν διαφορετικά seeds για τον γεννήτορα τυχαίων αριθμών. Είναι σημαντικό να αναφερθεί ότι η μέθοδος τοπικής αναζήτησης «localsearch(x)» που χρησιμοποιήθηκε στα πειράματα, ήταν μια παραλλαγή του BFGS του Powell [40]. Στα πειράματα μετρήθηκαν δύο τιμές: ο ρυθμός απόρριψης στο βήμα «New_Point» και συγκεκριμένα τα σημεία που δημιουργήθηκαν εκτός του πεδίου τιμών της συνάρτησης και ο μέσος αριθμός κλήσεων συνάρτησης που απαιτήθηκαν.

Τα πειραματικά αποτελέσματα παρουσιάζονται στον πίνακα που ακολουθεί (βλ.: Πίνακας 5: «Συγκριτικά αποτελέσματα διαφορετικών εκδοχών της CRS σε μια σειρά από συναρτήσεις»). Πιο αναλυτικά [70]:

- Η στήλη FUNCTION αναφέρεται στο όνομα της συνάρτησης στόχου.
- Η στήλη CRS-R αντιστοιχεί στον ρυθμό απόρριψης για τη μέθοδο CRS.

- Η στήλη NEWCRS-R εμφανίζει την ίδια μέτρηση για την τρέχουσα μέθοδο.
- Η στήλη CRS-C αναπαριστά τον μέσο αριθμό κλήσεων συνάρτησης για τη μέθοδο CRS.
- Η στήλη NEWCRS-C αναφέρεται στον μέσο αριθμό κλήσεων συνάρτησης για την προτεινόμενη μέθοδο.

Πίνακας 5: Συγκριτικά αποτελέσματα διαφορετικών εκδοχών της CRS σε μια σειρά από συναρτήσεις.

FUNCTION	CRS-R	CRS-C	NEWCRS-R	NEWCRS-C
BF1	1.37%	2523	0.00%	1689
BF2	1.33%	2506	0.17%	1569
BRANIN	16.00%	2014	9.13%	851
CAMEL	1.67%	2235	0.20%	1487
EASOM	51.03%	591	11.43%	635
EXP2	3.03%	1290	0.70%	644
EXP4	2.67%	4688	0.00%	1302
EXP8	2.77%	16,453	0.00%	2601
EXP16	4.00%	47,400	0.00%	5207
EXP32	7.70%	93,520	0.00%	10,414
EXP64	18.80%	135,638	0.00%	13,602
EXP100	38.53%	129,327	0.00%	14,506
GKLS250	3.87%	1784	0.27%	1684
GKLS350	6.43%	3881	0.03%	2088
GOLDSTEIN	3.60%	2154	0.70%	1829
GRIEWANK2	1.20%	2503	0.03%	2742
GUILIN550	8.33%	9129	0.00%	25,333
GUILIN1050	9.63%	30,806	0.00%	10,561
HANSEN	47.60%	2643	4.03%	1736
HARTMAN3	9.97%	3009	6.13%	1331
HARTMAN6	13.37%	13,615	0.00%	6091
RASTRIGIN	9.17%	2130	1.33%	2986
ROSENBROCK	0.00%	59,024	0.00%	15,719

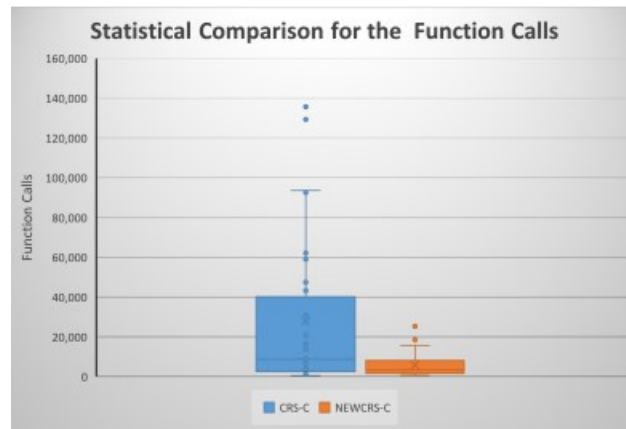
SHEKEL5	4.73%	8974	0.00%	2967
SHEKEL7	3.70%	8606	0.00%	3236
SHEKEL10	2.73%	9264	0.00%	3479
SINU4	3.90%	6525	0.00%	2889
SINU8	5.10%	21,561	0.00%	4946
SINU16	8.43%	62,194	0.00%	9539
SINU32	14.40%	135,986	0.00%	18,456
TEST2N4	24.57%	10,198	0.00%	3756
TEST2N5	34.17%	20,850	0.00%	4806
TEST2N6	42.50%	43,290	0.00%	6075
TEST2N7	50.37%	92,658	0.00%	7005
TEST30N3	24.10%	4011	0.00%	5691
TEST30N4	27.30%	7432	0.00%	8579
TOTAL	13.67%	1,000,412	0.86%	208,031

Σύμφωνα με τον παραπάνω πίνακα, νέα μέθοδος CRS έχει χαμηλότερο ρυθμό απόρριψης (0.86% έναντι 13.67%), υποδηλώνοντας ότι απορρίπτει λιγότερες λύσεις κατά τη διάρκεια της εκτέλεσής της. Παράλληλα, η νέα μέθοδος CRS επιτυγχάνει το ίδιο επίπεδο απόδοσης με μικρότερο συνολικό αριθμό κλήσεων συνάρτησης (208,031 έναντι 1,000,412). Αυτό υποδεικνύει ότι η νέα μέθοδος CRS είναι πιο αποδοτική από την προηγούμενη μέθοδο, καθώς παράγει λιγότερες απορρίψεις και απαιτεί λιγότερες κλήσεις συνάρτησης για την επίτευξη των αποτελεσμάτων. Πιο συγκεκριμένα:

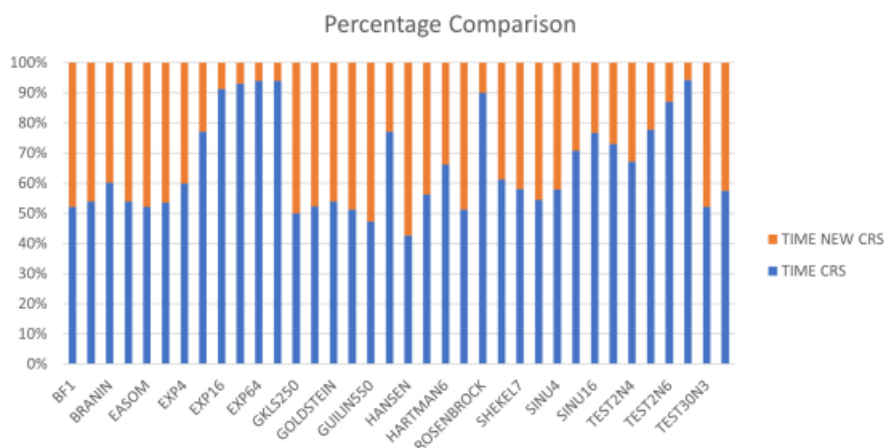
- Η μείωση του ρυθμού απόρριψης υποδεικνύει ότι η νέα μέθοδος είναι πιο αποτελεσματική στον τρόπο που εκτιμά τις λύσεις και αποφεύγει τις απορρίψεις που μπορεί να οδηγήσουν σε κακή σύγκλιση.
- Η μείωση του μέσου αριθμού κλήσεων συνάρτησης υποδεικνύει ότι η νέα μέθοδος επιτυγχάνει τη βέλτιστη λύση με λιγότερες απαιτούμενες εκτελέσεις της αντικειμενικής συνάρτησης, πράγμα που είναι επιθυμητό για οικονομία χρόνου και πόρων.

Γενικότερα, η νέα μέθοδος CRS φαίνεται να έχει βελτιωθεί σε σχέση με την προηγούμενη έκδοση, παρέχοντας καλύτερη απόδοση και αποτελέσματα με λιγότερη υπολογιστική πολυπλοκότητα.

Τέλος, παρουσιάζεται μια στατιστική σύγκριση μεταξύ της μεθόδου CRS και της προτεινόμενης μεθόδου, όπως φαίνεται στην ακόλουθη εικόνα:



Σύμφωνα με τα αποτελέσματα, η νέα μέθοδος σχεδόν εξαφανίζει τον ρυθμό απόρριψης σε κάθε συναρτησιακή δοκιμή. Αυτό είναι μία απόδειξη ότι ο νέος μηχανισμός που προτείνεται για τη δημιουργία νέου σημείου είναι πιο ακριβής από τον παραδοσιακό τρόπο. Επίσης, η νέα μέθοδος απαιτεί λιγότερες κλήσεις συνάρτησης από την CRS μέθοδο και αυτό απεικονίζεται από τις αντίστοιχες στήλες και τη στατιστική σύγκριση. Βέβαια, αυτό απεικονίζεται και στην ακόλουθη εικόνα [70]:



Σε πιο απαιτητικά προβλήματα, η προτεινόμενη μέθοδος φαίνεται να επιτυγχάνει τον στόχο με λιγότερες απαιτούμενες κλήσεις συνάρτησης σε σύγκριση με την αρχική. Ο συνδυασμός του κανόνα τερματισμού με τη βελτιωμένη τεχνική δημιουργίας νέου σημείου

οδηγεί σε πιο γρήγορο και ακριβές τερματισμό της μεθόδου, επιτρέποντάς της να επιτυγχάνει καλύτερη απόδοση.

Επιπρόσθετα, μετρήθηκε ο χρόνος εκτέλεσης για κάθε συναρτησιακή δοκιμή και καταχωρήθηκαν στον ακόλουθο πίνακα: Πίνακας 6: «Χρονικές συγκρίσεις» [70]:

Πίνακας 6:Χρονικές συγκρίσεις

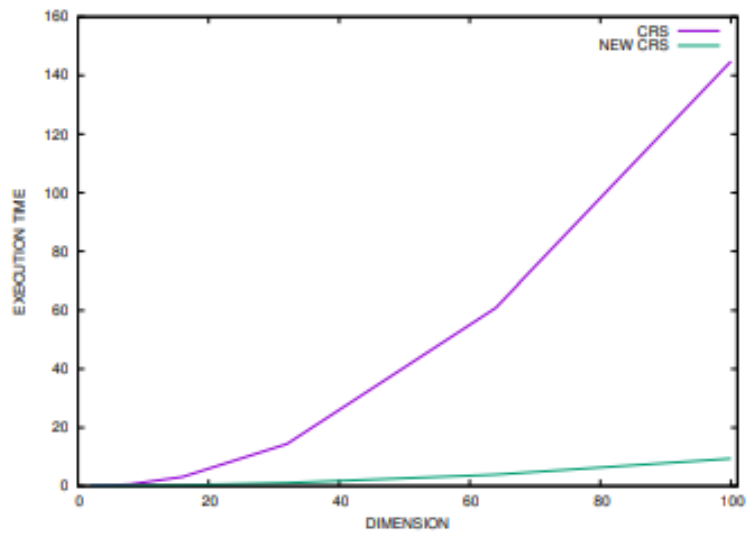
FUNCTION	CRS-TIME	NEWCRS-TIME	DIFF
BF1	0.168	0.154	8.33%
BF2	0.180	0.154	14.44%
BRANIN	0.209	0.138	33.97%
CAMEL	0.165	0.141	14.55%
EASOM	0.165	0.151	8.48%
EXP2	0.165	0.143	13.33%
EXP4	0.228	0.152	33.33%
EXP8	0.629	0.187	70.27%
EXP16	3.142	0.299	90.48%
EXP32	14.364	1.082	92.47%
EXP64	60.861	3.932	93.54%
EXP100	144.794	9.386	93.52%
GKLS250	0.592	0.593	-0.17%
GKLS350	0.658	0.599	8.97%
GOLDSTEIN	0.191	0.163	14.66%
GRIEWANK2	0.174	0.166	4.60%
GUILIN550	0.475	0.529	-11.37%
GUILIN1050	1.524	0.453	70.28%
HANSEN	0.217	0.292	-34.56%
HARTMAN3	0.21	0.163	22.38%
HARTMAN6	0.514	0.262	49.03%
RASTRIGIN	0.168	0.16	4.76%
ROSENBROCK	5.31	0.584	89.00%
SHEKEL5	0.321	0.203	36.76%
SHEKEL7	0.302	0.218	27.81%

SHEKEL10	0.325	0.271	16.62%
SINU4	0.283	0.206	27.21%
SINU8	0.897	0.369	58.86%
SINU16	4.775	1.448	69.68%
SINU32	24.413	8.999	63.14%
TEST2N4	0.389	0.19	51.16%
TEST2N5	0.733	0.209	71.49%
TEST2N6	1.714	0.256	85.06%
TEST2N7	4.326	0.264	93.90%
TEST30N3	0.222	0.203	8.56%
TEST30N4	0.324	0.239	26.23%
TOTAL	274.127	32.958	87.98%

Συγκεκριμένα:

- Η στήλη CRS-TIME αντιπροσωπεύει τον μέσο χρόνο εκτέλεσης της αρχικής μεθόδου CRS.
- Η στήλη NEWCRS-TIME αντιπροσωπεύει τον μέσο χρόνο εκτέλεσης της προτεινόμενης μεθόδου.
- Η στήλη DIFF είναι ο υπολογισμένος ποσοστιαίος διαφορά μεταξύ των προαναφερθέντων στηλών.

Όπως γίνεται αντιληπτό, η προτεινόμενη μέθοδος απαιτεί μικρότερο χρόνο εκτέλεσης σε σύγκριση με την αρχική μέθοδο, ενώ η μεγάλη διαφορά τους φαίνεται στα μεγάλα προβλήματα. Στην ακόλουθη εικόνα, παρουσιάζεται γραφικά ο μέσος χρόνος εκτέλεσης των δύο μεθόδων για το πρόβλημα EXP για διάφορο αριθμό διαστάσεων [70].



4. Η ΔΙΑΦΟΡΙΚΗ ΕΞΕΛΙΚΤΙΚΗ ΜΕΘΟΔΟΣ

4.1 Βασικές αρχές

Ο αλγόριθμος «Differential Evolution» (Διαφορικής Εξέλιξης) είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για την επίλυση προβλημάτων βελτιστοποίησης. Δημιουργήθηκε από τους Rainer Storn και Kenneth Price το 1995 [74]. Ο Kenneth V. Price, είναι καθηγητής στο Department of Electrical and Computer Engineering στο Montana State University. Είναι γνωστός για την έρευνά του στον τομέα της εξελικτικής βελτιστοποίησης και ένας από τους πρωτοπόρους που ανέπτυξαν τον αλγόριθμο DE. Συνεργάστηκε με τον Rainer Storn, ένας από τους σημαντικότερους ερευνητές στον τομέα της βελτιστοποίησης και του αλγορίθμου DE, ο οποίος συνέβαλε σημαντικά στη διάδοση και την εφαρμογή του αλγορίθμου σε διάφορους τομείς. Και οι δύο επιστήμονες, έχουν συγγράψει πολλές επιστημονικές εργασίες που ασχολούνται με τη βελτιστοποίηση και τους αλγορίθμους εξέλιξης [75,76,77,78].

Ο αλγόριθμος Differential Evolution (DE) ανήκει στην κατηγορία των εξερευνητικών αλγορίθμων βελτιστοποίησης. Λειτουργεί βασιζόμενος σε έναν πληθυσμό ατόμων (διανυσμάτων) και ο στόχος του είναι να βρει τον συνδυασμό παραμέτρων που ελαχιστοποιεί μια συνάρτηση κόστους ή μία αντικειμενική συνάρτηση. Ο αλγόριθμος DE, βασίζεται σε ορισμένες βασικές αρχές και διαδικασίες. Ακολούθως παρουσιάζονται οι βασικές αρχές της μεθόδου:

Δημιουργία Πληθυσμού:

- Ορίζεται ο αρχικός πληθυσμός των λύσεων που λέγεται P.
- Κάθε λύση αναπαρίσταται από ένα διάνυσμα αριθμών.

Υπολογισμός Κόστους:

- Για κάθε λύση στον πληθυσμό P, υπολογίζεται το κόστος, δηλαδή μια αριθμητική τιμή που περιγράφει την απόδοση της λύσης.

Διαφορική Μετάλλαξη (Mutation):

- Επιλέγονται τρία τυχαία άτομα από τον πληθυσμό P.

- Υπολογίζεται η διαφορά μεταξύ των προτύπων και των άλλων δύο ατόμων για κάθε παράμετρο του διανύσματος
- Η διαφορά για κάθε παράμετρο πολλαπλασιάζεται με έναν αριθμό που λέγεται F (συντελεστής κλιμάκωσης).
- Το αποτέλεσμα αυτών των πολλαπλασιασμών δημιουργεί ένα νέο διάνυσμα που λέγεται V (για νέα λύση).

Διασταυρωμένος Έλεγχος (Crossover):

- Συγκρίνεται κάθε στοιχείο του διανύσματος V, με το αντίστοιχο στοιχείο του αρχικού διανύσματος του στόχου.
- Αν το στοιχείο του V είναι καλύτερο από το αντίστοιχο στοιχείο του στόχου, τότε αποθηκεύετε, αλλιώς αντικαθίσταται με το στοιχείο του στόχου.

Επιλογή:

- Συγκρίνεται η νέα λύση V με την παλιά λύση στον πληθυσμό.
- Αν η V είναι καλύτερη από την παλιά λύση, τότε αντικαθίσταται η παλιά λύση με την V.
- Επαναλαμβάνεται αυτή η διαδικασία για πολλές επαναλήψεις ή για κάποιο κριτήριο σταθερού σημείου [79].

Αυτές οι αρχές επιτρέπουν στον αλγόριθμο DE να βελτιστοποιεί αριθμητικές συναρτήσεις κόστους εύρεσης της βέλτιστης λύσης.

Ο DE είναι ιδιαίτερα αποτελεσματικός σε προβλήματα βελτιστοποίησης όπου ο χώρος αναζήτησης είναι μη γραμμικός και δεν υπάρχουν κανόνες που να οδηγούν σε αναλυτικές λύσεις. Συνήθως χρησιμοποιείται σε προβλήματα βελτιστοποίησης όπου η συνάρτηση κόστους δεν είναι διαφορίσιμη ή διακριτική. Έχει εφαρμοστεί με επιτυχία σε πολλούς τομείς, συμπεριλαμβανομένης της βελτιστοποίησης παραμέτρων μηχανικής μάθησης και νευρωνικών δικτύων [76,70,81], καθώς και σε βιολογικές [82,83,84] και οικονομικές εφαρμογές [85, 86].

4.2 Ο αλγόριθμος DE

Ο βασικός αλγόριθμος της Διαφορικής Εξέλιξης (DE) ακολουθεί τα παρακάτω βήματα [40]:

1. **Όρισε** το μέγεθος του πληθυσμού $NP \geq 4$, συνήθως $NP = 10n$, όπου το n είναι η διάσταση του προβλήματος εισόδου.
2. **Όρισε** την πιθανότητα διασταύρωσης CR στο διάστημα $[0, 1]$. Μια τυπική τιμή για αυτήν την παράμετρο είναι 0,9.
3. **Όρισε** το βάρος διαφορικής μετάλλαξης F στο διάστημα $[0, 2]$. Μια τυπική τιμή για αυτήν την παράμετρο είναι 0,8.
4. **Αρχικοποίηση** όλων των μελών του πληθυσμού στον χώρο αναζήτησης. Τα μέλη του πληθυσμού ονομάζονται «πράκτορες».
5. **Επανάλαβε** τα ακόλουθα βήματα μέχρι να εκπληρωθεί ένα κριτήριο σταματήματος:
 - (a) **Για** $i = 1 \dots NP$, **κάνε** τα ακόλουθα:
 - i. **Όρισε** το x ως τον πράκτορα i .
 - ii. **Επέλεξε** τυχαία τρεις πράκτορες a, b, c .
 - iii. **Επέλεξε** ένα τυχαίο δείκτη R από το σύνολο $\{1, \dots, n\}$, όπου n είναι η διάσταση του προβλήματος.
 - iv. **Υπολόγισε** το διάνυσμα δοκιμής $y = [y_1, y_2, \dots, y_n]$ ως εξής:
 - A. **Όρισε** ένα τυχαίο αριθμό r_i στο διάστημα $[0, 1]$.
 - B. **Αν** $r_j < CR$ ή $j = R$, τότε $y_i = \alpha_j + F \times (b_j - c_j)$ αλλιώς $y_i = x_j$
 - v. **Για** $j = 1, \dots, n$ **κάνε** τα ακόλουθα:
 - A. **Όρισε** ένα τυχαίο αριθμό r_i στο διάστημα $[0, 1]$.
 - B. **Αν** $r_j < CR$ ή $j = R$, τότε $y_i = \alpha_j + F \times (b_j - c_j)$ αλλιώς $y_i = x_j$
 - vi. **Εάν** η τιμή της συνάρτησης $f(y) \leq f(x)$ τότε θέσε $x = y$.
 - vii. Τέλος επανάληψης
 - (b) **Τέλος επανάληψης**

6. **Επέστρεψε** τον πράκτορα που έχει το χαμηλότερο κόστος $f(x_{best})$ στον πληθυσμό, ως τη βέλτιστη λύση του αλγορίθμου.

4.3 Προτεινόμενες διαφοροποιήσεις

Ο αλγόριθμος DE τερματίζεται όταν επιτυγχάνεται ένα προκαθορισμένο αριθμό επαναλήψεων. Αυτό μπορεί να είναι εξαιρετικά ανεπαρκές σε ορισμένα προβλήματα, ενώ σε άλλα μπορεί να οδηγήσει σε πρόωρο τερματισμό. Δηλαδή, στον τερματισμό πριν βρεθεί το συνολικό ελάχιστο. Στο έργο του Ali και του Törn [87], προτείνεται ένα διαφορετικό κριτήριο τερματισμού, δηλαδή να τερματίζεται όταν ισχύει η σχέση:

$$f_{max} - f_{min} \leq \epsilon \quad (4.1)$$

όπου:

- f_{max} : αναφέρεται στην τιμή της συνάρτησης κόστους του χειρότερου ατόμου στον πληθυσμό
- f_{min} : αναφέρεται στην τιμή της συνάρτησης κόστους του καλύτερου ατόμου στον πληθυσμό.
- η παράμετρος ϵ : είναι ένας μικρός θετικός αριθμός που χρησιμοποιείται για την προσέγγιση της αλλαγής στη μέση τιμή της συνάρτησης κόστους που χρειάζεται για να αποφασίσουμε αν ο αλγόριθμος πρέπει να τερματιστεί.

Στο προτεινόμενο κριτήριο τερματισμού, υπολογίζεται η μέση τιμή της συνάρτησης κόστους του πληθυσμού σε κάθε επανάληψη. Εάν αυτή η τιμή δεν αλλάζει σημαντικά για έναν επαναλαμβανόμενο αριθμό επαναλήψεων, τότε ο αλγόριθμος ενδέχεται να μην ανακαλύψει ένα νέο γενικό ελάχιστο και να τερματίσει. Συνεπώς, σε κάθε γενιά t , μετριέται η ποσότητα από τη σχέση [40]:

$$\delta^{(t)} = \left| \sum_{i=1}^{NP} f_i^{(t)} \right| - \left| \sum_{i=1}^{NP} f_i^{(t-1)} \right| \quad (4.2)$$

με κανόνα τερματισμού που ορίζεται: $\delta^{(t)} \leq \epsilon$, για έναν προκαθορισμένο αριθμό γενεών M .

Το διαφορικό βάρος που προτάθηκε αρχικά στον αλγόριθμο DE ήταν μια στατική τιμή, πράγμα που σημαίνει ότι απαιτείται κάποια ρύθμιση για να βρει το γενικό ελάχιστο σε

κάθε συνάρτηση βελτιστοποίησης. Ο Ali κ.ά. [87] πρότειναν έναν μηχανισμό προσαρμογής γι' αυτήν την παράμετρο, έτσι ώστε ο αλγόριθμος να αναζητά σε μεγαλύτερους χώρους στις πρώτες γενιές και να επικεντρώνεται περισσότερο στις μεταγενέστερες γενιές. Ο μηχανισμός αυτός εκφράζεται ως εξής:

$$F = \begin{cases} \max \left(l_{min}, 1 - \left| \frac{f_{max}}{f_{min}} \right| \right), & \text{if } \left| \frac{f_{max}}{f_{min}} \right| \leq 1 \\ \max \left(l_{min}, 1 - \left| \frac{f_{min}}{f_{max}} \right| \right), & \text{otherwise} \end{cases} \quad (4.3)$$

Ο στοχαστικός μηχανισμός που προτείνεται εκφράζεται ως εξής [40]:

$$F = -\frac{1}{2} + 2 \times R \quad (4.4)$$

όπου το $R \in [0,1]$ είναι ένας τυχαίος αριθμός. Ο προτεινόμενος μηχανισμός, όπως και η τυχαιότητα που εισήγαγε η μέθοδος DE, θα επιτρέπει μια καλύτερη εξερεύνηση του χώρου αναζήτησης της αντικειμενικής συνάρτησης και θα βοηθά στην εύρεση με μεγαλύτερη ακρίβεια και ταχύτητα του γενικού ελάχιστου.

Προκειμένου να βελτιωθεί η απόδοση της μεθόδου Διαφοροποιημένης Εξέλιξης (DE) έχουν αναπτυχθεί διάφορες παραλλαγές που υιοθετούν διάφορες στρατηγικές. Οι αλλαγές που γίνονται διαφοροποιούν τον τρόπο επιλογής του διανύσματος του γονέα, τον αριθμό των διανυσματικών διαφορών που χρησιμοποιούνται και τον τρόπο καθορισμού των διανυσμάτων στη φάση της διασταύρωσης. Έχουν προταθεί αρκετές διαφορετικές στρατηγικές DE και κατευθυντήριες γραμμές για την εφαρμογή τους σε διάφορα προβλήματα από τους Price και Storn [88]. Μερικές από τις διαφοροποιήσεις αναφέρονται ακολούθως:

- Διαφοροποίηση μεταξύ των ατόμων: DE/rand/1/exp
- Διαφοροποίηση Μεταξύ Ατόμων και του Καλύτερου Ατόμου: DE/best/1/exp
- Συνδυασμός Διαφοροποίησης μεταξύ Ατόμων και Διαφοροποίησης μεταξύ Ατόμου και του Καλύτερου Ατόμου: DE/rand-to-best/1/exp
- Διαφοροποίηση μεταξύ του Τρέχοντος Ατόμου και Ενός Τυχαίου Ατόμου: DE/current-to-rand/1/exp

Η γενική σημειολογία DE/x/y/z, χρησιμοποιείται με σκοπό τον διαχωρισμό από το σύνολο των παραλλαγών. Το x αναφέρεται στη μέθοδο επιλογής του διανύσματος γονέα, δηλαδή τη μέθοδο μετάλλαξης, το y υποδεικνύει τον αριθμό των διανυσματικών διαφορών και το z υποδεικνύει τη στρατηγική διασταύρωσης των διανυσμάτων [89]. Παρακάτω ακολουθεί μία σύντομη περιγραφή για καθεμία από τις παραπάνω διαφοροποιήσεις:

Διαφοροποίηση μεταξύ των ατόμων: DE/rand/1/exp

Αυτή η στρατηγική χρησιμοποιείται για τη δημιουργία νέων λύσεων με τη χρήση τυχαίων ατόμων. Κάθε νέα λύση παράγεται από τη διαφορά μεταξύ δύο τυχαίων ατόμων του πληθυσμού, πολλαπλασιασμένη με ένα συγκεκριμένο σταθερό παράγοντα βήματος. Σύμφωνα με τη στρατηγική, αποδίδονται καλά αποτελέσματα όταν οι στρατηγικές που χρησιμοποιούν μεθόδους τύπου «best» για την επιλογή γονέων, αποτυγχάνουν.

Ο υπολογισμός του μεταλλαγμένου διανύσματος \vec{v}_i γίνεται με βάση την σχέση:

$$\vec{v}_i = \vec{x}_{i_0} + F (\vec{x}_{i_1} - \vec{x}_{i_2})$$

Ισχύει ότι $i \neq i_0 \neq i_1 \neq i_2 \in [0, \dots, NP - 1]$ και i_0, i_1, i_2 τυχαία.

Προτείνεται οι δοκιμές για τη ρύθμιση των παραμέτρων να ξεκινήσουν με τις τιμές: $F=0.7$, $CR=0.5$ [90].

Διαφοροποίηση Μεταξύ Ατόμων και του Καλύτερου Ατόμου: DE/best/1/exp

Σε αυτή τη στρατηγική το διάνυσμα με την καλύτερη τιμή της αντικειμενικής συνάρτησης στον τρέχοντα πληθυσμό (\vec{x}_{best}), επιλέγεται ως διάνυσμα γονέας. Αυτό συνήθως, οδηγεί σε μια πιο γρήγορη σύγκλιση προς τη βέλτιστη λύση. Στην περίπτωση αυτή το μεταλλαγμένο διάνυσμα υπολογίζεται από τη σχέση:

$$\vec{v}_i = (\vec{x}_{best} + F (\vec{x}_{i_1} - \vec{x}_{i_2}))$$

Ισχύει ότι $i \neq i_1 \neq i_2 \in [0, \dots, NP - 1]$ και i_1, i_2 τυχαία [18].

Συνδυασμός Διαφοροποίησης μεταξύ Ατόμων και Διαφοροποίησης μεταξύ Ατόμου και του Καλύτερου Ατόμου: DE/rand-to-best/1/exp

Σε αυτήν τη στρατηγική, συνδυάζονται οι δύο προηγούμενες προσεγγίσεις rand και best. Οι νέες λύσεις παράγονται με τη διαφορά μεταξύ ενός τυχαίου ατόμου και του καλύτερου ατόμου, πολλαπλασιασμένη με έναν παράγοντα βήματος. Η σχέση που υπολογίζεται το μεταλλαγμένο διάνυσμα είναι η ακόλουθη:

$$\vec{v}_i = \gamma \vec{x}_{best} + (1 - \gamma)\vec{x}_{i_1} + F (\vec{x}_{i_2} - \vec{x}_{i_3})$$

Ισχύει ότι: $i \neq i_1 \neq i_2 \neq i_3 \in [0, \dots, NP - 1]$ και i_1, i_2, i_3 τυχαία επιλεγμένα.

Το $\gamma \in [0,1]$ ελέγχει την απληστία του τελεστή μετάλλαξης.

Αν $\gamma=0$ τότε αυτή η στρατηγική ταυτίζεται με την προσέγγιση rand, ενώ όσο πιο κοντά στο 1 είναι η τιμή του γ , τόσο πιο άπληστη γίνεται η διαδικασία αναζήτησης.

Αν $\gamma=1$, τότε η στρατηγική ταυτίζεται με την best [90].

Διαφοροποίηση μεταξύ του Τρέχοντος Ατόμου και Ενός Τυχαίου Ατόμου: DE/current-to-rand/1/exp

Σε αυτήν τη στρατηγική, η διαφοροποίηση γίνεται μεταξύ του τρέχοντος ατόμου και ενός τυχαίου άλλου ατόμου. Πιο συγκεκριμένα, χρησιμοποιεί το τρέχον άτομο του συνόλου ως έναν από τους γονείς στη δημιουργία νέων ατόμων, ενώ το άλλο άτομο επιλέγεται τυχαία από το συνολικό πληθυσμό. Αυτή η στρατηγική αποσκοπεί στον συνδυασμό της εξερεύνησης του τρέχοντος ατόμου με την τυχειότητα που εισάγει η επιλογή ενός τυχαίου άλλου ατόμου, με σκοπό την αύξηση της ποικιλίας και της εξερεύνησης του χώρου αναζήτησης. Αυτό επιτρέπει στον αλγόριθμο να εξερευνήσει νέες περιοχές του χώρου αναζήτησης.

4.4 Πειραματικά αποτελέσματα

Τα πειράματα που διεξήχθησαν, πραγματοποιήθηκαν τριάντα φορές, όπου κάθε φορά χρησιμοποιήθηκαν διαφορετικά seeds για τον γεννήτορα τυχαίων αριθμών σε κάθε δοκιμαστική συνάρτηση. Οι παράμετροι που χρησιμοποιήθηκαν στα πειράματα φαίνονται στον ακόλουθο πίνακα [40]:

Παράμετροι	Τιμή
------------	------

NP	10n
F	0.8
CR	0.9
M	20
ε	10^{-4}

Τα πειραματικά αποτελέσματα παρουσιάζονται στους πίνακες που ακολουθούν (Πίνακας 7: «Πειράματα με τον κανόνα τερματισμού Ali», Πίνακας 8: «Πειράματα με τον προτεινόμενο κανόνα τερματισμού»). Η ερμηνεία των στηλών του πίνακα αναφέρεται παρακάτω [40]:

- Η στήλη Function: αναφέρεται στο όνομα της συνάρτησης στόχου.
- Η στήλη Static: αντιπροσωπεύει τη στατική τιμή του παράγοντα F ($F = 0.8$).
- Η στήλη Ali: αντιπροσωπεύει τον μηχανισμό που δίνεται από τη σχέση 4.3.
- Η στήλη Proposed: αντιπροσωπεύει την προτεινόμενη διαδικασία που δίνεται από την σχέση 4.4.

Από τα πειράματα που πραγματοποιήθηκαν, παρατηρήθηκε ότι οι δύο προτεινόμενες παραλλαγές μειώνουν σε μεγάλο βαθμό τον αριθμό κλήσεων των συναρτήσεων, ενώ δεν επηρεάζουν σημαντικά τη μέση απόδοση της μεθόδου, καθώς παραμένει υψηλή σε όλες τις περιπτώσεις (βλ.: Πίνακας 7, Πίνακας 8).

Πίνακας 7: Πειράματα με τον κανόνα τερματισμού Ali

FUNCTION	STATIC	ALI	PROPOSED
BF1	1142	1431	847
BF2	1164	1379	896
BRANIN	984	816	707
CM4	3590	7572	2079
CAMEL	1094	18849	685

EASOM	1707	2014	1327
EXP2	532	323	449
EXP4	2421	1019	1494
EXP8	15750	3670	5632
EXP16	160031	15150	21416
EXP32	320039	152548	77936
GKLS250	784	944	614
GKLS2100	772	1531	599(0.97)
GKLS350	1906(0.93)	3263	1275(0.93)
GKLS3100	1883	3539	1373
GOLDSTEIN	988	818	769
GRIEWANK2	1299(0.97)	1403	883(0.93)
HANSEN	2398	2968	1400
HARTMAN3	1448	836	1050
HARTMAN6	9489(0.97)	4015(0.97)	4667(0.80)
POTENTIAL3	90027	89776	21824
POTENTIAL4	120387(0.97)	120405(0.33)	45705(0.97)
POTENTIAL5	150073	150104	83342
RASTRIGIN	1246	1098(0.93)	871
ROSENBROCK4	6564	9695	4499
ROSENBROCK8	44240	72228	13959
ROSENBROCK16	160349(0.90)	160538(0.60)	53594
SHEKEL5	5524	3810	3057(0.83)
SHEKEL7	5266	3558	2992(0.87)

SHEKEL10	5319	3379	3076
TEST2N4	4200	1980	2592
TEST2N5	7357	2957	4055
TEST2N6	12074	4159	5836
TEST2N7	18872	5490	7904
SINU4	3270	1855	2216
SINU8	23108	6995	8135
SINU16	160092	36044	30943
SINU32	213757(0.70)	160536(0.53)	83369(0.80)
TEST30N3	1452	1732	959
TEST30N4	1917	2287	1378
TOTAL	1564515(0.97)	1062714(0.96)	506404(0.98)

Πίνακας 8: Πειράματα με τον προτεινόμενο κανόνα τερματισμού

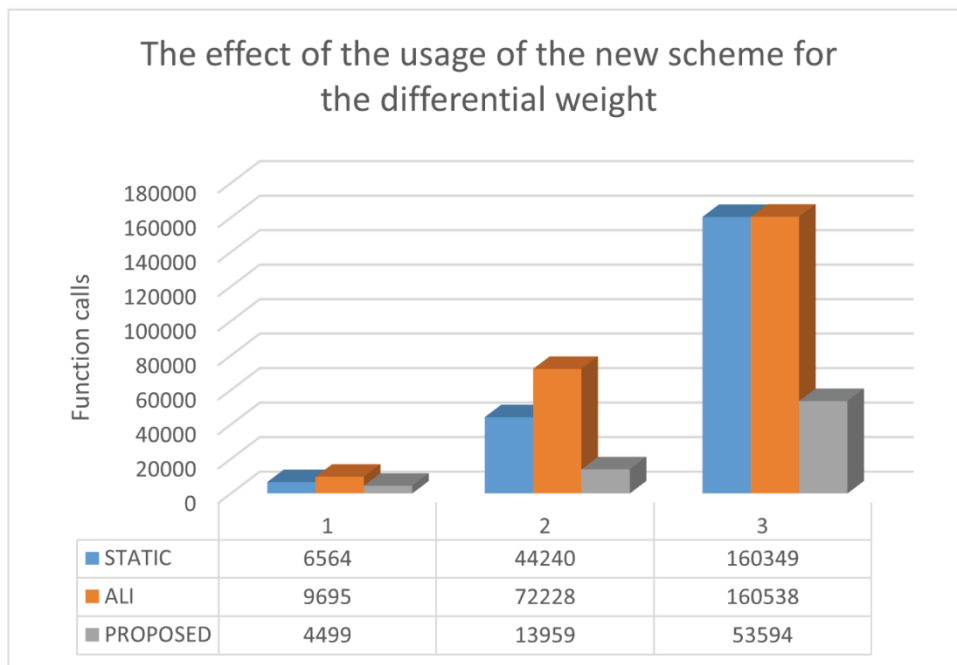
FUNCTION	STATIC	ALI	PROPOSED
BF1	996	1124	889
BF2	926	1026	816
BRANIN	878	900	730
CM4	1148(0.70)	1991	1103
CAMEL	1049	904(0.93)	846
EASOM	447	448	446
EXP2	470	461	467
EXP4	915	903	892
EXP8	1797	3558	1796
EXP16	3578	7082	3521

EXP32	7082	14125	7022
GKLS250	498	576	493
GKLS2100	533	884(0.97)	515
GKLS350	823	1130(0.93)	814(0.97)
GKLS3100	858	1495(0.97)	829(0.93)
GOLDSTEIN	945	993	915
GRIEWANK2	947	921	826
HANSEN	2104	1949	1479
HARTMAN3	1017	1005	952
HARTMAN6	4679(0.90)	3744(0.97)	3128(0.87)
POTENTIAL3	21473	2284	8197
POTENTIAL4	44191(0.43)	3098(0.33)	24659(0.97)
POTENTIAL5	75910	3443	52664
RASTRIGIN	841	994	777
ROSENBROCK4	4934	7192	3300
ROSENBROCK8	29583	49696	10907
ROSENBCROK16	160349	160538(0.60)	38315
SHEKEL5	4389(0.97)	4266	2839(0.83)
SHEKEL7	3905	3685	2668
SHEKEL10	4049	3548	2629
TEST2N4	2785	2275	2221
TEST2N5	4481	3170	3122
TEST2N6	6852	4286	4296
TEST2N7	11971	5701	6267

SINU4	2322	1987	1755
SINU8	9990	6156	5113
SINU16	6892	3628(0.97)	16905
SINU32	7235(0.80)	7438(0.83)	7218
TEST30N3	1033	1098	951
TEST30N4	1355	1444	1285
TOTAL	432610(0.98)	321166(0.96)	224567(0.99)

Στο διάγραμμα που ακολουθεί (βλ. Διάγραμμα 1: «Το αποτέλεσμα της χρήσης του νέου σχήματος για το διαφορετικό βάρος»), παρουσιάζεται γραφικά ο μέσος αριθμός κλήσεων των συναρτήσεων ROSENBROCK4, ROSENBROCK8 και ROSENBROCK16. Με βάση το σχήμα, παρατηρείται συγκριτικά η επίδραση της προτεινόμενης μεθόδου.

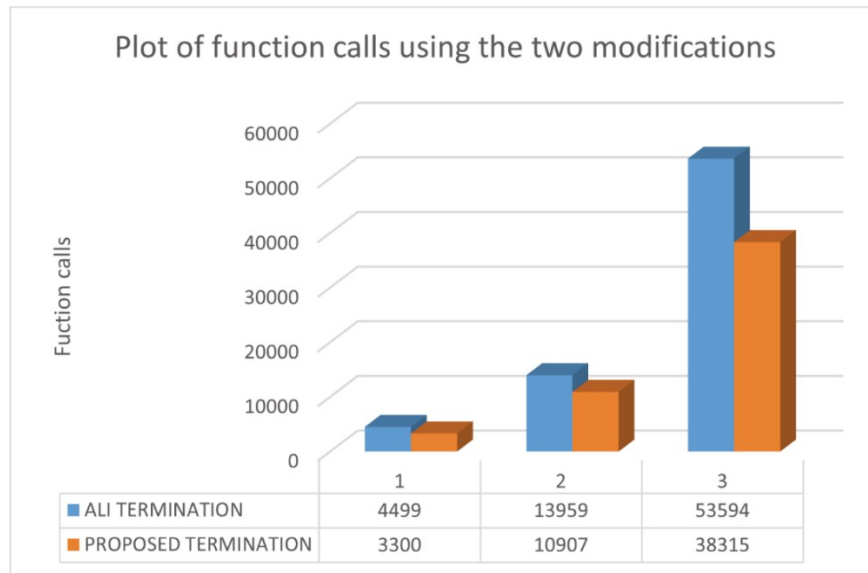
Διάγραμμα 1: Το αποτέλεσμα της χρήσης του νέου σχήματος για το διαφορετικό βάρος



Παράλληλα στο Διάγραμμα 2: «Σχεδιάγραμμα κλήσεων συναρτήσεων χρησιμοποιώντας τις δύο τροποποιήσεις», είναι εμφανές ότι ο συνδυασμός των δύο τροποποιήσεων που

προτάθηκαν, μείωσε ακόμη περισσότερο τον μέσο αριθμό κλήσεων των συναρτήσεων που απαιτούνται για τον εντοπισμό του καθολικού ελάχιστου.

Διάγραμμα 2: Σχεδιάγραμμα κλήσεων συναρτήσεων χρησιμοποιώντας τις δύο τροποποιήσεις



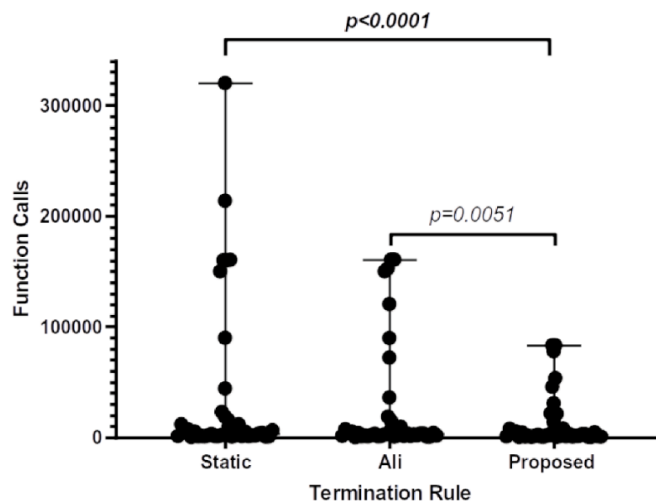
Για να αποδειχθεί πόσο αποτελεσματικές είναι οι τροποποιήσεις, στο Διάγραμμα 3: «Συγκρίσεις χρόνου για μια ποικιλία συναρτήσεων δοκιμής», παρουσιάζονται οι συνολικοί χρόνοι για 30 διαφορετικές εκτελέσεις σε έναν υπολογιστή I7 με LINUX DEBIAN και 16GB μνήμη. Έγινε σύγκριση της αρχικής μεθόδου με το κριτήριο τερματισμού ALI, τον προτεινόμενο κριτήριο τερματισμού και τον προτεινόμενο κριτήριο τερματισμού μαζί με το προτεινόμενο σχήμα βαρύτητας. Είναι φανερό ότι οι προτεινόμενες τροποποιήσεις μειώνουν σημαντικά όχι μόνο τον αριθμό των απαιτούμενων κλήσεων συναρτήσεων, αλλά και τον χρόνο που απαιτείται για την εκτέλεση.

Διάγραμμα 3: Συγκρίσεις χρόνου για μια ποικιλία συναρτήσεων δοκιμής



Για την σύγκριση του προτεινόμενου σχήματος βαρύτητας της διαφορικής εξέλιξης με τις άλλες δύο μεθόδους, χρησιμοποιήθηκε η στατιστική δοκιμή Wilcoxon signed-rank. Τα αποτελέσματα αυτής της στατιστικής δοκιμής φαίνονται στο Διάγραμμα 4: «Αποτελέσματα σύγκρισης μεθόδων».

Διάγραμμα 4: Αποτελέσματα σύγκρισης μεθόδων



5. ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΜΗΝΟΥΣ ΣΩΜΑΤΙΔΙΩΝ

5.1 Βασικές αρχές - Ιστορικά στοιχεία

Ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization-PSO) πρωτοεμφανίστηκε το 1995 από τους ερευνητές James Kennedy και Russell Eberhart [91]. Εμπνευσμένο από την κοινωνική συμπεριφορά των πτηνών και των ψαριών, ο PSO αναπτύχθηκε ως ένας αλγόριθμος βελτιστοποίησης βασισμένος στο συλλογικό έργο. Στον PSO, ένας αριθμός "σωματιδίων" αναπαριστούν δυνητικές λύσεις σε ένα πρόβλημα βελτιστοποίησης. Τα σωματίδια κινούνται στον χώρο αναζήτησης της βέλτιστης λύσης με βάση την εμπειρία τους και τη συνεργασία με άλλα σωματίδια. Η ιδέα πίσω από τον PSO, είναι ότι τα σωματίδια μπορούν να "μάθουν" από την εμπειρία τους και να συγκλίνουν προς την βέλτιστη λύση του προβλήματος βελτιστοποίησης. Κάθε σωματίδιο έχει μια θέση και μια ταχύτητα, οι οποίες προσαρμόζονται βάσει της απόδοσης τους. Οι επαναλήψεις του αλγορίθμου επιτρέπουν στα σωματίδια να εξερευνούν τον χώρο αναζήτησης και να βρίσκουν βέλτιστες λύσεις [92].

Οι βασικές αρχές του αλγορίθμου PSO είναι οι εξής:

- Σμήνος Σωματιδίων: Στον PSO, αναπαρίστανται λύσεις ως "σωματίδια" που ανήκουν σε ένα σμήνος. Κάθε σωματίδιο έχει μια θέση και μια ταχύτητα.
- Ενημέρωση Ταχύτητας: Κάθε σωματίδιο ανανεώνει τη θέση του βάσει της ταχύτητάς του και της καλύτερης θέσης που έχει βρει προσωπικά (τοπική) και η οποία έχει βρεθεί από όλο το σμήνος (καθολική).
- Βέλτιστο Στόχο: Σκοπός του PSO είναι να βρει την καλύτερη θέση που μπορεί να πάρει κάθε σωματίδιο, και κατά συνέπεια, το ολικό ελάχιστο της συνάρτησης.
- Ενημέρωση Ταχύτητας και Θέσης: Κατά την εκτέλεση του αλγορίθμου, οι ταχύτητες και οι θέσεις των σωματιδίων ενημερώνονται βάσει των παραπάνω αρχών [93].

Από τη δημιουργία του, ο PSO έχει εξελιχθεί και έχει εφαρμοστεί ευρέως σε πολλούς τομείς, συμπεριλαμβανομένων της φυσικής [94,95], της χημείας [96,97], της ιατρικής [98,99], της οικονομίας [100] κ.ά. Ο PSO συνεχίζει να αποτελεί ένα εργαλείο

βελτιστοποίησης με αξιοσημείωτη ισχύ και ευελιξία και η έρευνα συνεχίζει να τον εξελίσσει για να ανταποκριθεί σε νέες προκλήσεις και εφαρμογές.

5.2 Ο αλγόριθμος PSO

Ο βασικός τρόπος λειτουργίας του PSO είναι η χρήση ενός συνόλου από σωματίδια που κινούνται σε έναν χώρο αναζήτησης λύσεων. Κάθε σωματίδιο αναπαριστά μια πιθανή λύση στο πρόβλημα βελτιστοποίησης. Κατά την εκτέλεση του αλγορίθμου, κάθε σωματίδιο κινείται εντός του χώρου αναζήτησης με βάση δύο βασικές αρχές:

- Την προσωπική του εμπειρία: Κάθε σωματίδιο έχει ένα "ατομικό καλύτερο" που αντιπροσωπεύει την καλύτερη λύση που έχει βρει μέχρι στιγμής στον χώρο αναζήτησης. Το σωματίδιο κινείται προς αυτή την καλύτερη του λύση.
- Τη συλλογική εμπειρία: Τα σωματίδια αλληλοεπιδρούν μεταξύ τους και επηρεάζονται από τις καλύτερες λύσεις που έχουν βρεθεί από άλλα σωματίδια στον χώρο αναζήτησης. Αυτό βοηθά στη σύγκλιση του αλγορίθμου προς τη βέλτιστη λύση.

Καθώς ο PSO εκτελείται, τα σωματίδια συνεργάζονται και εξερευνούν τον χώρο αναζήτησης με σκοπό να βρουν την καλύτερη δυνατή λύση. Ο αλγόριθμος συνεχώς ενημερώνει τις θέσεις των σωματιδίων με βάση την εμπειρία τους και την εμπειρία του συνόλου.

Παρακάτω ακολουθεί αναλυτικά ο αλγόριθμος PSO [43]:

1. Αρχικοποίηση:

- Ορισμός $iter = 0$ (μετρητής επαναλήψεων).
- Ορισμός του αριθμού των σωματιδίων m .
- Ορισμός του μέγιστου επιτρεπόμενου αριθμού επαναλήψεων $iter_{max}$.
- Τυχαία αρχικοποίηση των θέσεων m σωματιδίων x_1, x_2, \dots, x_m με την προϋπόθεση ότι κάθε x_i ανήκει στο χώρο $S \subset R^n$.
- Τυχαία αρχικοποίηση των ταχυτήτων των m σωματιδίων u_1, u_2, \dots, u_m , με την προϋπόθεση ότι κάθε u_i ανήκει στο χώρο $S \subset R^n$.

- f. Για κάθε $i = 1 \dots m$, ορισμός $p_i = x_i$. Τα διανύσματα p_i είναι οι καλύτερες εντοπισμένες τιμές για κάθε σωματίδιο i .
- g. Ορισμός $p_{best} = \arg \min_{i \in 1..m} f(x_i)$.

2. Έλεγχος τερματισμού.

Έλεγχος αν τα κριτήρια τερματισμού έχουν ικανοποιηθεί. Αν ναι, τότε σταματά η διαδικασία.

3. Για $i=1..m$ επανάλαβε

- a. Ενημέρωση της ταχύτητας u_i ως μια συνάρτηση των u_i , p_i και p_{best}
- b. Ενημέρωση της θέσης $x_i = x_i + u_i$.
- c. Ορίζουμε ένα τυχαίο αριθμό $r \in [0, 1]$. Αν το $r \leq p_m$ τότε εκτελείται η διαδικασία τοπικής αναζήτησης για τη θέση $x_i = LS(x_i)$
- d. Αξιολόγηση της καταλληλότητας του σωματιδίου i , $f(x_i)$
- e. Εάν $f(x_i) \leq f(p_i)$ τότε $p_i = x_i$

4. Τέλος επανάληψης

5. Ορισμός $p_{best} = \arg \min_{i \in 1..m} f(x_i)$

6. Ορισμός $iter = iter + 1$.

7. Επιστροφή στο Βήμα 2.

5.3 Παραλλαγές της μεθόδου PSO

Ο αλγόριθμος Βελτιστοποίησης με Σμήνος Σωματιδίων (PSO) έχει εξελιχθεί και παραλλαγεί κατά την διάρκεια των χρόνων για την αντιμετώπιση διαφορετικών αναγκών και προβλημάτων βελτιστοποίησης. Ορισμένες από τις πιο δημοφιλείς παραλλαγές και επεκτάσεις του PSO περιλαμβάνουν [101]:

Directionally Driven Self-Regulating Particle Swarm Optimization (DDSRPSO)

Ο Tanweer και οι συνεργάτες του, το 2015, πρότειναν ένα νέο αλγόριθμο με το όνομα "Directionally Driven Self-Regulating Particle Swarm Optimization" (DDSRPSO) [102]. Αυτός ο αλγόριθμος συνέβαλλε στην ανάπτυξη δύο νέων προσεγγίσεων μάθησης, και συγκεκριμένα, στη στρατηγική ενημέρωσης κατεύθυνσης και στη στρατηγική ανεξάρτητη από περιστροφές. Ωστόσο, αποτελεί μια βελτιωμένη παραλλαγή της μεθόδου Self-Regulating Particle Swarm Optimization (SRPSO). Ο DDSRPSO χωρίζει το μέγεθος του πληθυσμού σε τρεις ομάδες βάσει την προσαρμογή των σωματιδίων. Οι ομάδες αυτές είναι: η ομάδα των "elite" (καλύτερα σωματίδια), η ομάδα των κακώς επιδιορθωμένων σωματιδίων (χειρότερο σωματίδιο) και τα υπόλοιπα σωματίδια. Τα σωματίδια της ομάδας των "elite" ακολουθούν την ίδια στρατηγική με το SRPSO, και η βελτίωση επιτυγχάνεται χρησιμοποιώντας μια ομάδα "elite" (καλύτερα σωματίδια) για την επίτευξη κατευθυντικών ενημερώσεων. Στην περίπτωση των χειρότερων σωματιδίων, οι συγγραφείς θεωρούν ότι ακολουθούν την καθολική κατεύθυνση αναζήτησης, και η ταχύτητα του SRPSO τροποποιείται ως εξής.

$$V_{kd}^{t+1} = \omega_k * V_{kd}^t + c_1 * r_1 * (lp_{gd}^t - x_{kd}^t) \quad (1)$$

όπου:

- ω_k , αντιπροσωπεύει το υποδειγματικό βάρος αδράνειας του SRPSO για να επιτευχθεί αργή κίνηση στις αλλαγές της ταχύτητας.
- lp_{gd}^k , αντιπροσωπεύει τη μέση τιμή των τριών καλύτερων επιδρώντων σωματιδίων από την ομάδα των "elite".
- p_{dt}^g , αντιπροσωπεύει το καλύτερο γενικό σωματίδιο που έχει επιτευχθεί έως στιγμής σε t γενιές.
- k, αντιπροσωπεύει τον αριθμό των σωματιδίων από την ομάδα των χειρότερων σωματιδίων.

Τα υπόλοιπα σωματίδια εκτελούν είτε τη στρατηγική SRPSO για την αυτο-αναγνώριση της γενικής κατεύθυνσης αναζήτησης είτε τη στρατηγική αναλλοίωτης περιστροφής για την έρευνα της ιδιότητας της περιστροφικής αντίστασης στο χώρο αναζήτησης. Η στρατηγική ανεξάρτητη από περιστροφές βασίζεται στον υπολογισμό του κέντρου βάρους, το οποίο προκύπτει από τρία σημεία: την τρέχουσα θέση των σωματιδίων, την καλύτερη προσωπική θέση τους και την καλύτερη γενική θέση. Στην SRPSO, το κέντρο βάρους

χρησιμοποιείται για τον υπολογισμό μιας νέας θέσης των σωματιδίων, η οποία αργότερα οδηγεί στην μη επίγνωση των καλύτερων γενικών θέσεων. Στη DD-SRPSO, η αυτό-αναγνώριση των σωματιδίων για τις γενικές κατευθύνσεις αναζήτησης παίζει κυρίαρχο ρόλο ώστε τα σωματίδια να είναι ενήμερα για τις γενικές κατευθύνσεις αναζήτησης.

$$\vec{p}_i^t = \vec{X}_i^t + c_1 * r_1 \otimes (\vec{p}_i^t - \vec{X}_i^t) \quad (2)$$

$$\vec{p}_g^t = \vec{X}_i^t + c_2 * r_2 \vec{p}_i^{s\sigma} (\vec{p}_g^t - \vec{X}_i^t) \quad (3)$$

$$\vec{G}_i^t = \begin{cases} \frac{\vec{p}_i^t + \vec{p}_g^t + \vec{x}_i^t}{3} \text{ if } \vec{p}_i^{s\sigma} = 1 \\ \frac{\vec{p}_i^t + \vec{x}_i^t}{2} \text{ if } \vec{p}_i^{s\sigma} = 0 \end{cases} \quad (4)$$

Όπου:

- G_i^t : αντιπροσωπεύει το κέντρο βάρους, το οποίο χρησιμοποιείται για την επιλογή του στοχαστικού σημείου των σωματιδίων \vec{X}_i^t στην υπερ-σφαίρα $H_i(\vec{G}_i^t, \|\vec{G}_i^t - \vec{X}_i^t\|)$,

- \vec{p}_i^t αντιπροσωπεύει την προσωπική καλύτερη θέση των σωματιδίων i στις t επαναλήψεις,

- \vec{p}_g^t ορίζει την παγκόσμια καλύτερη θέση των σωματιδίων i

- $\vec{p}_i^{s\sigma}$ αντιπροσωπεύει την αυτοαντίληψη των σωματιδίων με κοινωνική γνώση ($\vec{p}_i^{s\sigma} = 1$) και χωρίς κοινωνική γνώση ($\vec{p}_i^{s\sigma} = 0$), με r_1, r_2 τυχαίους αριθμούς μεταξύ (0,1)

- το σύμβολο \otimes : αντιπροσωπεύει την πρόσθεση μεταξύ των δύο επιλεγμένων χαρακτηριστικών. Επομένως, η εξίσωση (2) δηλώνει ότι η νέα ταχύτητα του σωματιδίου i στην επόμενη επανάληψη \vec{p}_i^{t+1} υπολογίζεται προσθέτοντας την ποσότητα $c_1 * r_1$ στη διαφορά μεταξύ της καλύτερης θέσης \vec{p}_i^t και της τρέχουσας θέσης \vec{X}_i^t .

Οπότε, η τροποποίηση της εξίσωσης έχει ως εξής:

$$\vec{V}_i^{t+1} = \omega * \vec{V}_i^t + H_i(\vec{G}_i^t, \|\vec{G}_i^t - \vec{X}_i^t\|) - \vec{X}_i^t \quad (5)$$

Όπου \vec{V}_i^{t+1} αντιπροσωπεύει την ενημέρωση της νέας θέσης και της ταχύτητας των σωματιδίων στην επόμενη επανάληψη ($t+1$) και την τρέχουσα ταχύτητα των σωματιδίων.

Bisection PSO (BPSO) – Παραλλαγή Σωματιδίου με Διχοτόμηση PSO:

Η τεχνική Bisection PSO (BPSO) εισήχθη από τους Kennedy και Eberhart [103]. Στην BPSO, ο χώρος αναζήτησης χωρίζεται για κάθε διάσταση με βάση το βέλτιστο σωματίδιο που βρίσκεται στο κέντρο. Μετά την αρχικοποίηση, η αξιολόγηση της προσαρμοστικότητας και το βέλτιστο σημείο αναζήτησης επεξεργάζονται βάσει της διαίρεσης του χώρου αναζήτησης [104]. Αν ο χώρος αναζήτησης διαιρείται στα δύο, τότε τα σωματίδια διαιρούνται σε πολλές γενιές με βάση τη διαίρεση του χώρου αναζήτησης. Διαφορετικά, ακολουθείται η συμβατική μέθοδος PSO χωρίς διαίρεση του χώρου αναζήτησης.

$$V_{id}^{t+1} = W_{LDW} * V_{id}^t + c_1 * r_1 * (p_{id}^t - X_{id}^t) + C_2 * r_2 * (p_{gd}^t - X_{id}^t) \quad (6)$$

$$x_{id}^{new} = x_{id}^{old} + V_{id}^{new} \quad (7)$$

Όπου:

- ω_{LDW} , αντιπροσωπεύει τη γραμμική φθίνουσα αδράνεια (LDW)
- p_{id}^t , αντιπροσωπεύει τον προσωπικό καλύτερο παράγοντα του σωματιδίου i της διάστασης d στις t επαναλήψεις
- p_{gd}^t , ορίζει την παγκόσμια καλύτερη θέση που έχει επιτευχθεί μέχρι στιγμής στις t επαναλήψεις
- X_{id}^t , αναπαριστά τη θέση του τρέχοντος σωματιδίου i

Dispersed PSO (DPSO) – Παραλλαγή με διασκορπισμένη ταύτιση:

Μία άλλη μέθοδος που προτάθηκε για να βελτιωθεί η ταχύτητα σύγκλισης της PSO με τη χρήση του συντελεστή διάχυσης στην παράμετρο, αντί να είναι η κεντρική όπως στην PSO είναι η μέθοδος DPSO [105]. Πιο συγκεκριμένα, στην PSO όλα τα σωματίδια κινούνται προς μια καλύτερη τιμή (Gbest) και εξαιτίας αυτού, το άτομο με την καλύτερη επίδοση που κρατά την καλύτερη τιμή γίνεται απαρατήρητο. Προκειμένου να βελτιωθεί αυτό, εισάγεται η διασκορπισμένη PSO με ένα δείκτη που κρατά τη διαφορά απόδοσης μεταξύ της καλύτερης, της χειρότερης και της τρέχουσας τιμής των σωματιδίων. Ο βαθμός των υπολογισμένων τιμών προσαρμοστικότητας αναπαρίσταται ως εξής:

$$Grade_u(t) = \frac{f_{worst}(t) - f(x_u(t))}{f_{worst}(t) - f_{best}(t)} \quad (8)$$

Όπου:

$Grade_u(t)$, προσδιορίζεται ως ο λόγος της διαφοράς μεταξύ του αριθμητή του χειρότερου $f_{worst}(t)$ και του τρέχοντος σωματιδίου $f(x_u(t))$ προς τη διαφορά μεταξύ του χειρότερου σωματιδίου $f_{worst}(t)$ και του καλύτερου σωματιδίου $f_{best}(t)$.

Με τη βοήθεια της σχέσης $Grade_u(t)$, η παράμετρος ορίζεται εκ νέου ως εξής:

$$C_2(t) = C_{low} + (C_{cup} - C_{low})Grade_u(t) \quad (99)$$

όπου:

C_{low} και C_{cup} , δύο σταθεροί αριθμοί

Αυτή η παράμετρος εισάγει την έννοια της «εκμετάλλευσης», για τα άτομα με την καλύτερη απόδοση, ενώ ταυτόχρονα περιγράφει το υψηλό ποσοστό εξερεύνησης, όταν εντοπίζονται τα άτομα με τις χειρότερες τιμές. Έπειτα οι λύσεις περνούν στη διαδικασία της μετάλλαξης, όπου τα άτομα που επιλέγονται είναι ομοιόμορφα τυχαία, προκειμένου να αποφευχθεί η πρόωρη σύγκλιση. Η στρατηγική μετάλλαξης ενσωματώνεται με την PSO για να βελτιώσει την τοπική ικανότητα αναζήτησης μεταξύ των σωματιδίων, με έναν συνθετικό παράγοντα τιμών βάσει ενός τυχαίου αριθμού που ικανοποιεί το κατώφλι (0,5). Η εξίσωση της στρατηγικής μετάλλαξης περιγράφεται ως εξής:

$$V_{jk}(t) = \begin{cases} 0.5Xx_{max}Xr_1if r_2 < 0.5 \\ 0.5Xx_{max}Xr_1 \text{ (σε διαφορετική περίπτωση)} \end{cases} \quad (100)$$

Όπου:

r_1, r_2 είναι τυχαίοι αριθμοί μεταξύ (0 ,1). Σε κάθε βήμα των επαναλήψεων συμμετέχει μόνο μία μεταβλητή διαδικασία μετάλλαξης. Η μεταβλητή αυτή, ελέγχει το πόσο συχνά συμβαίνει η μετάλλαξη, πόσο αποτελεσματική είναι και ποιο μέρος του πληθυσμού επηρεάζεται.

Compact PSO (CPSO)- Παραλλαγή με συμπυκνωμένη ταύτιση:

Η προσέγγιση αυτή λειτουργεί σε χώρους μνήμης, όπου οι απαιτήσεις της μνήμης είναι ελάχιστες, όπως συμβαίνει σε ενσωματωμένα συστήματα ή σε περιβάλλοντα με περιορισμένους πόρους. Στη μέθοδο cPSO, αρχικοποιείται ένα Διάνυσμα Διατάραξης (PV) αντί για τον πληθυσμό σμήνους PSO, και ο πληθυσμός του αποτελείται από δυαδικά διανύσματα [106]. Ισχύει η σχέση:

$$PV^t = \mu^t \sigma^t \quad (43)$$

Όπου:

μ , σ είναι ο μέσος όρος και η τυπική απόκλιση των μεταβλητών σχεδιασμού που θα κατασκευαστούν στις επαναλήψεις cPSO και t είναι ο αριθμός της τρέχουσας επανάληψης.

Μετά την αρχικοποίηση του PV, το cPSO αρχικοποιεί το $\mu[i]$, $\sigma[i]$ όπου το i είναι ο αριθμός της μεταβλητής. Στη συνέχεια, οι θέσεις του προσωπικού καλύτερου και του καθολικού καλύτερου καθορίζονται από το διάνυσμα PV. Τυχαίες γενεές θέσεων και ταχυτήτων σωματιδίων δημιουργούνται και ενημερώνονται ως εξής:

$$V_{id}^{t+1} = \phi_1 V^t + \phi_2 U_1 (P_{id}^t - x^t) + \phi U_2 (X_{gd}^t - X_{id}^t) \quad (101)$$

$$X_{id}^{t+1} = \gamma_1 X_{id}^t + \gamma_2 V_{id}^{t+1} \quad (102)$$

Όπου:

- ϕ_1, ϕ_2, ϕ_3 , είναι οι παράμετροι ρύθμισης της μετα-βελτιστοποίησης του αριθμού των σωματιδίων
- U_1, U_2 , είναι σταθερές
- X_{gd}^t , αντιπροσωπεύει την καλύτερη λύση που έχει βρεθεί μέχρι εκείνη τη στιγμή στις επαναλήψεις
- d , οι διαστάσεις
- X_{id}^t , αντιπροσωπεύει την τρέχουσα θέση σωματιδίου για d διάσταση σε t επαναλήψεις.

Μετά από αυτά τα βήματα, τα σωματίδια νικητή και χαμένου (καλύτερο, χειρότερο) έχουν υπολογιστεί από τα δημιουργούμενα X_{id}^{t+1} ενώ το P_{id}^t ενημερώνεται χρησιμοποιώντας το Διάνυσμα Διατάραξης (PV).

Η ενημέρωση των μεταβλητών σχεδιασμού της επόμενης γενιάς ως εξής:

$$\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_p} (\text{winner}[i] - \text{loser}[i]) \quad (103)$$

$$\sigma^{t+1}[i] = \sqrt{(e^t[i])^2 + (\mu^t[i])^2 - (\mu^{t+1}[i])^2 + \frac{1}{N_p} (\text{winner}^2[i] - \text{loser}^2[i])} \quad (104)$$

- N_p , αντιπροσωπεύει τον πληθυσμό εικονικού μεγέθους
- e^t , καθορίζει την εκθετική τιμή σε t επαναλήψεις
- $\text{winner}[i]$ και $\text{loser}[i]$, αντιπροσωπεύουν τα σωματίδια νικητή και χαμένου σε κάθε επανάληψη. Οι τιμές αυτές βοηθούν τα τρέχοντα σωματίδια να ενημερώνονται με τον νικητή της επανάληψης εάν έχει την καλύτερη αξία από την προηγούμενη καλύτερη τιμή.

5.4 Προτεινόμενες διαφοροποιήσεις

Σύμφωνα με τους Shi και Eberhat [107], οι υψηλές τιμές του συντελεστή αδράνειας οδηγούν σε καλύτερη εξερεύνηση της περιοχής αναζήτησης, ενώ οι μικρές τιμές χρησιμοποιούνται για την καλύτερη τοπική έρευνα γύρω από περιοχές με γενικό ελάχιστο. Η τιμή του συντελεστή αδράνειας ξεκινά με μεγάλες τιμές και μειώνεται με την επανάληψη. Στην μελέτη που πραγματοποιήθηκε, προτείνεται μια νέα τεχνική για την παράμετρο της αδράνειας μέσα από τρεις διαφοροποιήσεις. Παρακάτω ακολουθούν αναλυτικά οι τροποποιήσεις που υλοποιήθηκαν:

1^η τροποποίηση: Υπολογισμός της αδράνειας

$$\omega_{iter} = 0.5 + \frac{r}{2}$$

όπου r ένας τυχαίος αριθμός που ανήκει στο διάστημα $[0,1]$.

Θα πρέπει να αναφερθεί ότι αυτός ο υπολογισμός της αδράνειας στους πίνακες με τα πειραματικά αποτελέσματα θα ονομάζεται Π1.

2^η τροποποίηση: Γραμμικά μεταβαλλόμενη αδράνεια (ελάχιστη έκδοση)

Σε αυτή τη περίπτωση η σχέση που προτείνεται είναι η εξής:

$$\omega_{iter} = \frac{iter_{max} - iter}{iter_{max}}(\omega_{max} - \omega_{min}) + \omega_{min}$$

όπου ω_{min} είναι η ελάχιστη τιμή της αδράνειας και ω_{max} η μέγιστη τιμή για την αδράνεια. Αυτός ο υπολογισμός της αδράνειας θα ονομάζεται I2 στους πίνακες με τα πειραματικά αποτελέσματα.

3^η τροποποίηση: Προτεινόμενη τεχνική

Σε αυτή τη περίπτωση ο υπολογισμός της αδράνειας περιλαμβάνει τον αριθμό των επαναλήψεων όπου η μέθοδος καταφέρνει να βρει ένα νέο ελάχιστο. Στις πρώτες επαναλήψεις και όταν η μέθοδος πρέπει να εξερευνήσει περισσότερο την περιοχή έρευνας, η αδράνεια θα είναι μεγάλη. Όταν η μέθοδος πρέπει να επικεντρωθεί σε ένα ελάχιστο, τότε η αδράνεια θα μειωθεί. Για αυτό τον λόγο, σε κάθε επανάληψη iter, μετράται η ποσότητα:

$$\delta^{(iter)} = \left| \sum_{i=1}^m |f_i^{(iter)}| - \sum_{i=1}^m |f_i^{(iter-1)}| \right|$$

Στα πρώτα βήματα του αλγορίθμου, αυτή η ποσότητα θα αλλάζει από επανάληψη σε επανάληψη με γρήγορο ρυθμό, και σε κάποιο σημείο, δεν θα αλλάζει πλέον με τον ίδιο ρυθμό ή θα γίνει μηδενική. Για το λόγο αυτό, ορίζουμε στο $\delta^{(iter)}$ ως εξής:

$$\zeta^{(iter)} = \begin{cases} 1, & \delta^i = 0 \\ 0, & \text{διαφορετικά} \end{cases}$$

Χρησιμοποιώντας αυτήν την παρατήρηση, δημιουργούνται δύο επιπλέον μετρικές που δίνονται από τις ακόλουθες σχέσεις:

- $S_{\delta}^{(iter)} = \sum_{i=1}^{iter} \zeta^{(i)}$ και
- $C_{\delta}^{(iter)} = \frac{S_{\delta}^{(iter)}}{i=1} \zeta^{(i)}$

Οπότε, ο υπολογισμός της αδράνειας σύμφωνα με τα παραπάνω υπολογίζεται ως εξής:

$$\omega_{iter} = \omega_{max} - \frac{iter}{C_{\delta}^{(iter)}} (\omega_{max} - \omega_{min})$$

Αυτός ο μηχανισμός θα ονομάζεται IP στους πίνακες με τα πειραματικά αποτελέσματα.

5.5 Πειραματικά αποτελέσματα

Για να αξιολογήσουμε την επίδραση των προτεινόμενων τροποποιήσεων στην αρχική μέθοδο, πραγματοποιήθηκε μια σειρά πειραμάτων σε διάφορες συναρτήσεις δοκιμής. Όλα τα πειράματα διενεργήθηκαν 30 φορές χρησιμοποιώντας διαφορετική γεννήτρια τυχαίων αριθμών κάθε φορά. Ο κώδικας υλοποιήθηκε σε ANSI C++, ενώ χρησιμοποιήθηκε η συνάρτηση `drand48()` για την παραγωγή των τυχαίων αριθμών. Τέλος, η μέθοδος τοπικής αναζήτησης που χρησιμοποιήθηκε ήταν η BFGS. Όλες οι παράμετροι που χρησιμοποιήθηκαν στα διενεργηθέντα πειράματα περιλαμβάνονται στον ακόλουθο πίνακα (βλ. Πίνακας 9: «Τιμές για τις παραμέτρους των πειραματικών παραμέτρων»):

Πίνακας 9: Τιμές για τις παραμέτρους των πειραματικών παραμέτρων

Parameter	Value
m	100
$inter_{max}$	100
p_l	0.05
c_1	1.0
c_2	1.0
ω_{min}	0.4
ω_{max}	0.9
ϵ	0.001

k_{max}	15
-----------	----

Τα πειράματα αξιολόγησαν τόσο τον αντίκτυπο της νέας μεθόδου υπολογισμού της αδράνειας όσο και του κριτηρίου για την αποφυγή τοπικών ελαχίστων, καθώς και του νέου κανόνα τερματισμού. Τα αποτελέσματα αυτών των πειραμάτων καταγράφηκαν σε ξεχωριστούς πίνακες, προκειμένου να είναι πιο εύκολο να αναλυθεί η επίδραση κάθε μεταβολής ξεχωριστά.

Για κάθε κριτήριο τερματισμού, παρουσιάζονται δύο πίνακες: ο πρώτος περιέχει πειράματα με το σχετικό κριτήριο τερματισμού χωρίς τη διαδικασία απόρριψης της κλίσης, και ο δεύτερος πίνακας περιέχει πειράματα με τη διαδικασία απόρριψης της κλίσης ενεργοποιημένη. Οι αριθμοί στα κελιά του πίνακα αντιπροσωπεύουν τον μέσο αριθμό κλήσεων συνάρτησης. Το κλάσμα σε παρένθεση υποδηλώνει το ποσοστό των εκτελέσεων όπου βρέθηκε το καθολικό ελάχιστο. Η απουσία αυτών των κλασμάτων σημαίνει ότι το καθολικό ελάχιστο ανακαλύφθηκε σε κάθε εκτέλεση (επιτυχία 100%). Τα πειραματικά αποτελέσματα χρησιμοποιώντας το κριτήριο τερματισμού της Εξίσωσης: $a \leq \epsilon$, παρουσιάζονται στους πίνακες: Πίνακας 10: «Πειράματα με τον κανόνα τερματισμού Ali χωρίς έλεγχο της κλίσης» και Πίνακας 11: «Πειράματα με τον κανόνα τερματισμού Ali με τον έλεγχο της κλίσης ενεργοποιημένο». Τα πειραματικά αποτελέσματα για τον Διπλό Πίνακα τερματισμού της Εξίσωσης: $\sigma^{(iter)} \leq \sigma$, παρουσιάζονται στους πίνακες: Πίνακας 12: «Πειράματα με τον διπλό κανόνα τερματισμού χωρίς έλεγχο της κλίσης» και Πίνακας 13: «Πειράματα με τον διπλό κανόνα τερματισμού με τον έλεγχο της κλίσης ενεργοποιημένο». Τέλος, για το προτεινόμενο κριτήριο τερματισμού, τα αποτελέσματα παρουσιάζονται στους πίνακες: Πίνακας 14: «Πειράματα με τον προτεινόμενο κανόνα τερματισμού χωρίς τον έλεγχο της κλίσης» και Πίνακας 15: «Πειράματα με τον προτεινόμενο κανόνα τερματισμού με τον έλεγχο της κλίσης ενεργοποιημένο». Επιπλέον, τα boxplots για τα προτεινόμενα κριτήρια τερματισμού χωρίς και με τον έλεγχο της κλίσης της Εξίσωσης: $\|x - z\| \leq r_c \wedge$ εικονίζονται στα διαγράμματα: Διάγραμμα 5: «Τυπική απόκλιση των κλήσεων συνάρτησης για τους κανόνες τερματισμού χωρίς έλεγχο της κλίσης» και Διάγραμμα 6: «Τυπική απόκλιση των κλήσεων συνάρτησης για τους κανόνες τερματισμού με έλεγχο της κλίσης» αντίστοιχα. Τα παραπάνω αποτελέσματα οδηγούν σε μια σειρά παρατηρήσεων:

1. Η μέθοδος PSO είναι μια ανθεκτική μέθοδος, και αυτό είναι εμφανές από το υψηλό ποσοστό επιτυχίας στον εντοπισμό του καθολικού ελάχιστου, παρόλο που ο αριθμός των σωματιδίων που χρησιμοποιήθηκε ήταν σχετικά χαμηλός (100).
2. Η προτεινόμενη μέθοδος υπολογισμού της αδράνειας, όπως ορίζεται στην Εξίσωση: $\omega_{iter} = \omega_{max} - \frac{iter}{C_s(iter)}(\omega_{max} - \omega_{min})$, επιτυγχάνει μια σημαντική μείωση στον αριθμό των κλήσεων μεταξύ 11 και 25%, ανάλογα με το κριτήριο τερματισμού που χρησιμοποιείται. Ωστόσο, η παρουσία του μηχανισμού ελέγχου της κλίσης της Εξίσωσης: $\|x - z\| \leq r_c \wedge$, ακυρώνει οποιοδήποτε κέρδος της μεθόδου, καθώς το κριτήριο απόρριψης μειώνει σημαντικά τον αριθμό των κλήσεων ανεξάρτητα από το μηχανισμό υπολογισμού της αδράνειας που χρησιμοποιείται.
3. Ο μηχανισμός αποφυγής τοπικών βελτιστοποιήσεων του ελέγχου της κλίσης μειώνει δραστικά τον απαιτούμενο αριθμό κλήσεων για κάθε κριτήριο τερματισμού, διατηρώντας ταυτόχρονα το ποσοστό επιτυχίας της μεθόδου σε εξαιρετικά υψηλά επίπεδα.
4. Το προτεινόμενο κριτήριο τερματισμού είναι σημαντικά ανώτερο από τα άλλα δύο με τα οποία έγινε η σύγκριση. Επιπλέον, αν το κριτήριο τερματισμού συνδυαστεί με τον μηχανισμό αποφυγής τοπικών βελτιστοποιήσεων, το κέρδος στον αριθμό των κλήσεων αυξάνεται ακόμα περισσότερο.

Για να επιδείξουμε το αντίκτυπο της προτεινόμενης μεθόδου τερματισμού, πραγματοποιήθηκε ένα επιπρόσθετο πείραμα. Συγκεκριμένα, πραγματοποιήθηκαν πειράματα με τη συνάρτηση SINU (γνωστή και ως συνάρτηση sinusoidal) προκειμένου να μετρηθεί η αποδοτικότητά της, για τιμές από $n = 2$ έως $n = 32$. Για όλες αυτές τις τιμές, δοκιμάστηκαν και οι τρεις τεχνικές τερματισμού. Στην περίπτωση αυτή, το 'n' αναφέρεται στη διάσταση του προβλήματος, δηλαδή στο πλήθος των άγνωστων μεταβλητών. Το αποτέλεσμα αυτού του πειράματος απεικονίζεται γραφικά στο Διάγραμμα 7: «Πειράματα με τη συνάρτηση SINU για μια σειρά από διαστάσεις προβλημάτων από $n = 2$ έως $n = 32$ ». Αυτό το γράφημα δείχνει ότι η μέθοδος του διπλού κουτιού είναι σημαντικά ανώτερη από τη μέθοδο Ali, αλλά, φυσικά, η νέα προτεινόμενη μέθοδος επιπλέον μειώνει τον απαιτούμενο αριθμό κλήσεων συνάρτησης. Επιπλέον, ο αντίκτυπος της εφαρμογής του μηχανισμού απόρριψης βασισμένου στην κλίση της Εξίσωσης: $\|x - z\| \leq r_c \wedge$ εικονίζεται γραφικά στο Διάγραμμα 8: «Πειράματα με τον προτεινόμενο κανόνα τερματισμού με

έλεγχου της κλίσης και χωρίς έλεγχο της κλίσης», όπου το προτεινόμενο κριτήριο τερματισμού εφαρμόζεται σε μια σειρά από συναρτήσεις δοκιμής με τον έλεγχο της κλίσης και χωρίς τον έλεγχο της κλίσης.

Πίνακας 10: Πειράματα με τον κανόνα τερματισμού Ali χωρίς έλεγχο της κλίσης

FUNCTION	I1	I2	I3	IP
BF1	24929	22874	18739	22088
BF2	24043	22254	17172	20743
BRANIN	17691	16205	13397	12471
CM4	20117	22568	26867	14941
CAMEL	19474	17813	14461	13492
EASOM	13327	13106	9969	9212
EXP2	6339	8243	7853	3501
EXP4	7816	10066	10900	4458
EXP8	8667	10937	13126	4761
EXP16	8748	11402	15754	5098
EXP32	9567	12323	18189	5471
GKLS250	10907	12562	9673	8552
GKLS2100	12960	13403	9930	9541
GKLS350	15410(0.97)	14722	10542	9298
GKLS3100	16639	14495	10412	13075(0.97)
GOLDSTEIN	20437	22877	16410	8935
GRIEWANK2	27620	24230	18473	20133
HANSEN	21513	20279	16326	15046
HARTMAN3	16233	17152	12305	6511
HARTMAN6	47038	48947	46852	23431
POTENTIAL3	31684	32175	36930	24463
POTENTIAL4	184602	181231	168962	129267
POTENTIAL5	74508	70519	76890	54042
RASTRIGIN	23574	20865	15596	16198
ROSENBROCK4	145178	161136	160341	129891

ROSENBROCK8	95290	97035	96687	80408
ROSENBCROK16	118614	116454	115122	97004
SHEKEL5	27458	27088	25927	18036
SHEKEL7	27521	27271	25967	18805
SHEKEL10	29699(0.97)	28082	25511	20823
TEST2N4	26740	27050	22905	19495
TEST2N5	20243(0.97)	20290	17729	16024(0.97)
TEST2N6	33118	33366	30118	25235(0.93)
TEST2N7	23266(0.90)	22804	21294	18218(0.90)
SINU4	17035	20487	18971	11079
SINU8	22827	27176	27732	12379
SINU16	31055	35998	42984	15692
SINU32	44736(0.97)	51624	82114	25991
TEST30N3	18733	20119	17803	17543
TEST30N4	20348	22191	20679	20085
TOTAL	1365704(0.99)	1399419	1367612	1001436(0.99)

Πίνακας 11: Πειράματα με τον κανόνα τερματισμού Ali με τον έλεγχο της κλίσης ενεργοποιημένο

FUNCTION	I1	I2	I3	IP
BF1	9709	8918	9531	10932
BF2	10196	9588	9089	10730
BRANIN	10718	9597	9813	9501
CM4	6242	7503	12531	6985
CAMEL	10422	9306	8491	9624
EASOM	11565	11366	8497	8196
EXP2	3364	4443	4558	1926
EXP4	3558	4767	6023	2122
EXP8	3716	4787	7753	2186
EXP16	3784	5076	9696	2211
EXP32	4137	5698	11379	2323

GKLS250	5917	7080	7517	5273
GKLS2100	6843	8261	7449	7296
GKLS350	6845	8076	7833	5881(0.97)
GKLS3100	10290(0.93)	10187	7828	8066(0.97)
GOLDSTEIN	7977	9035	8505	4381
GRIEWANK2	12567	12222	12000	12037
HANSEN	13441	13360	11876	10818
HARTMAN3	9758	9548	8123	4114
HARTMAN6	12893(0.90)	12889(0.93)	22309	10126(0.93)
POTENTIAL3	17912	16420	21904	15969
POTENTIAL4	73629	64886	95707	67084
POTENTIAL5	40585	35239	47807	33661
RASTRIGIN	11305	10101	11141	10046
ROSENBROCK4	18115	21919	38407	43093
ROSENBROCK8	12869	14192	31923	25405
ROSENBCROK16	12096	13023	38486	23165
SHEKEL5	10347	11466	14446	11802
SHEKEL7	11511	10521	13944	10399
SHEKEL10	10834	10842	13785	12253
TEST2N4	11133	10869	12161	11546
TEST2N5	10923(0.97)	10315	10868	11072(0.97)
TEST2N6	12331(0.97)	12345	14123	15652
TEST2N7	11342(0.93)	11354	12118	12370(0.93)
SINU4	7724	9845	12294	6575
SINU8	8468	10969	18122	5382
SINU16	9334	13213	31589	9294
SINU32	13290	17502(0.97)	63111(0.97)	14959
TEST30N3	12675	12954	12472	12482
TEST30N4	13964	14903	14999	15389
TOTAL	494119(0.9)	504585(0.9)	720208(0.9)	502326(0.99)

	9)	9)	99))
--	----	----	-----	---

Πίνακας 12: Πειράματα με τον διπλό κανόνα τερματισμού χωρίς έλεγχο της κλίσης

FUNCTION	I1	I2	I3	IP
BF1	6807	6866	6712	6757
BF2	6102	6150	6057	6207
BRANIN	4551	4596	4470	4435
CM4	9814	10101	9580	9342
CAMEL	5055	5202	4897	5004
EASOM	2975	2788	3014	3000
EXP2	4436	4541	4377	4543
EXP4	5443	5562	5331	5290
EXP8	5682	5754	5614	5504
EXP16	5707	5799	5638	5526
EXP32	5871	5797	5769	5659
GKLS250	3973	3906	3971	3921
GKLS2100	4009	3862	4073	3958
GKLS350	4558	3965	4525(0.97)	4266
GKLS3100	4701(0.87)	4266	4361(0.90)	4465
GOLDSTEIN	10259	9145	7945	7625
GRIEWANK2	5932	6194	5700	5915
HANSEN	6386	6260	5688(0.97)	5874
HARTMAN3	4681	4694	4625	4675
HARTMAN6	14245	14091	13793	13825
POTENTIAL3	7219	7206	7532	7234
POTENTIAL4	38053	37924	38421	38897
POTENTIAL5	15196	14459	15708	15358
RASTRIGIN	5915	5797	5944(0.83)	5844
ROSENBROCK4	91574	101485	117512	76367
ROSENBROCK8	66648	61974	58831	41591
ROSENBCROK16	62029	54550	63406	55800

SHEKEL5	9119	10271(0.97)	8975	8538
SHEKEL7	9197	9831	9638	8732
SHEKEL10	10417	10449	9373	9721(0.90)
TEST2N4	8512	8272	8884	7992
TEST2N5	5793	5704	5511(0.90)	5515
TEST2N6	9797(0.93)	9731	9657(0.83)	9666(0.97)
TEST2N7	6435(0.80)	6659	6713(0.77)	5990(0.87)
SINU4	7567	7774	7334	7063
SINU8	9882	10083	9643	9331
SINU16	12750	12947	12569	12207
SINU32	20164	21112	19684(0.90)	19239
TEST30N3	6388	7942	5934	5855
TEST30N4	7611	9251	6385	8284
TOTAL	531453(0.99)	532690(0.99)	543794(0.98)	475015(0.99)

Πίνακας 13: Πειράματα με τον διπλό κανόνα τερματισμού με τον έλεγχο της κλίσης ενεργοποιημένο

FUNCTION	I1	I2	I3	IP
BF1	3296	3038	3063	3003
BF2	2922	2762	2863	2845
BRANIN	2562	2641	2538	2564
CM4	3569	4277	2944	3230
CAMEL	2646	2854	2467	2577
EASOM	2490	2390	2479	2464
EXP2	2377	2489	2261	2315
EXP4	2456	2669	2282	2389
EXP8	2429	2671	2268	2385
EXP16	2358	2569	2227	2326
EXP32	2337	2533	2248	2312
GKLS250	2394	2535	2274	2321
GKLS2100	2384	2511	2267	2333

GKLS350	2492	2410	2212	2339
GKLS3100	2800(0.90)	2708	2648(0.83)	2571
GOLDSTEIN	3161	3701	3166	2799
GRIEWANK2	3910	4520	3543	3641
HANSEN	4409	4268	3755	4325
HARTMAN3	2423	2518	2374	2425
HARTMAN6	3913	4390	4199(0.93)	3700
POTENTIAL3	3951	4093	4482	4021
POTENTIAL4	18555	19559	19506	18691
POTENTIAL5	8771	8397	9677	9154
RASTRIGIN	3111	3244(0.97)	3031	3146
ROSENBROCK4	9729	12980	11453	8587
ROSENBROCK8	4987	6738	4688	5512
ROSENBCROK16	4410	5939	4553	4002
SHEKEL5	3906	4095	3203	3495
SHEKEL7	3119	3965	2950	3528
SHEKEL10	3497(0.97)	4464	3142(0.97)	3353
TEST2N4	3468(0.97)	4059	4167	3881(0.93)
TEST2N5	3318(0.97)	3786	2926(0.90)	3157(0.97)
TEST2N6	4523(0.93)	5046(0.93)	5537(0.83)	4066(0.97)
TEST2N7	3364(0.80)	4191(0.90)	4183(0.80)	3315(0.87)
SINU4	3173	3807	2610	3004
SINU8	3055	3742	2592	2857
SINU16	3160	3746	3854	3290
SINU32	6613	7377	6327	6450
TEST30N3	5129	6367	5605	4451
TEST30N4	5649	6441	6074	5543
TOTAL	162816(0.99)	182490(0.99)	164638(0.98)	158367(0.99)

Πίνακας 14: Πειράματα με τον προτεινόμενο κανόνα τερματισμού χωρίς τον έλεγχο της κλίσης.

FUNCTION	I1	I2	I3	IP
BF1	5305	5326	5240	5209
BF2	4760	4841	4750	4856
BRANIN	3599	3703	3520	3443
CM4	7674	7835	7430	7057
CAMEL	3996	4131	3864	3825
EASOM	2370	2292	2425	2478
EXP2	3528	3613	3455	3675
EXP4	4292	4350	4178	4020
EXP8	4579	4632	4515	4278
EXP16	4576	4637	4505	4236
EXP32	4692	4771	4588	4296
GKLS250	3105	3065	3115	3024
GKLS2100	3193	3049	3193	3099
GKLS350	3308	3000	3560	3401
GKLS3100	2935(0.97)	2777	3158(0.83)	3088
GOLDSTEIN	5534	5595	5332	5265
GRIEWANK2	4225	4332	4413	4489
HANSEN	3865	3762	3824	3769
HARTMAN3	3724	3770	3714	3705
HARTMAN6	11901(0.97)	11829(0.97)	11386	10573
POTENTIAL3	5910	5850	6134	6501
POTENTIAL4	30880	30570	31180	30682
POTENTIAL5	12021	11643	12521	13475
RASTRIGIN	4583	4595	4625	4360
ROSENBROCK4	58299	61266	55759	35517
ROSENBROCK8	31778	30888	30989	22055
ROSENBROCK16	32719	30503	30957	24478
SHEKEL5	6806	7047(0.97)	6636	6233

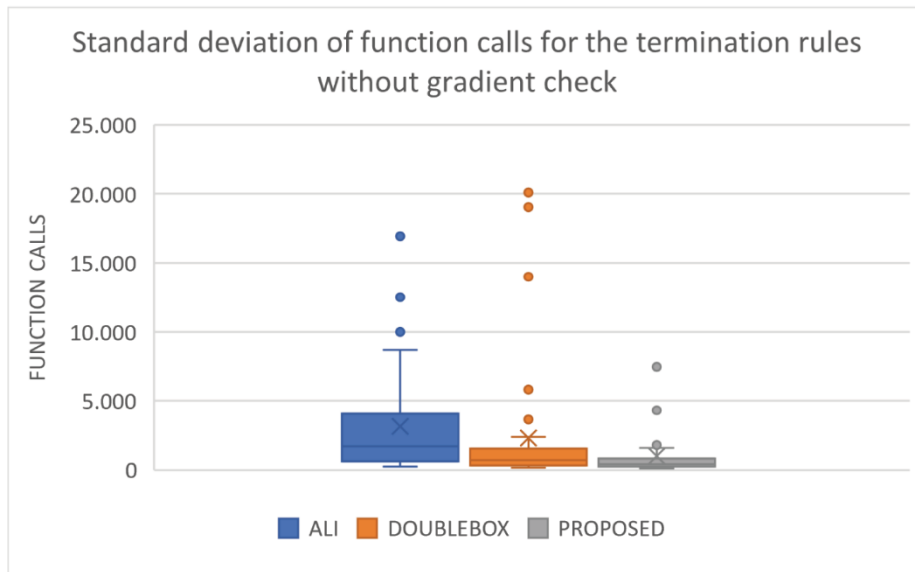
SHEKEL7	6807	7001	6626	6270
SHEKEL10	6774	6987	6583	6534
TEST2N4	6111	6127	5909	5893
TEST2N5	4455(0.97)	4558	4372(0.97)	4271(0.93)
TEST2N6	7446(0.97)	7419	7218(0.87)	7122(0.93)
TEST2N7	4992(0.90)	5057	4888(0.83)	4680(0.90)
SINU4	5948	6043	5750	5229
SINU8	7965	8095	7778	6963
SINU16	10121	10252	9968	9219
SINU32	16093	16509	15663	14478
TEST30N3	4331	4953	4230	3957
TEST30N4	6290	6341	4288	4717
TOTAL	361490(0.99)	363013(0.99)	352239(0.99)	310420(0.99)

Πίνακας 15: Πειράματα με τον προτεινόμενο κανόνα τερματισμού με τον έλεγχο της κλίσης ενεργοποιημένο

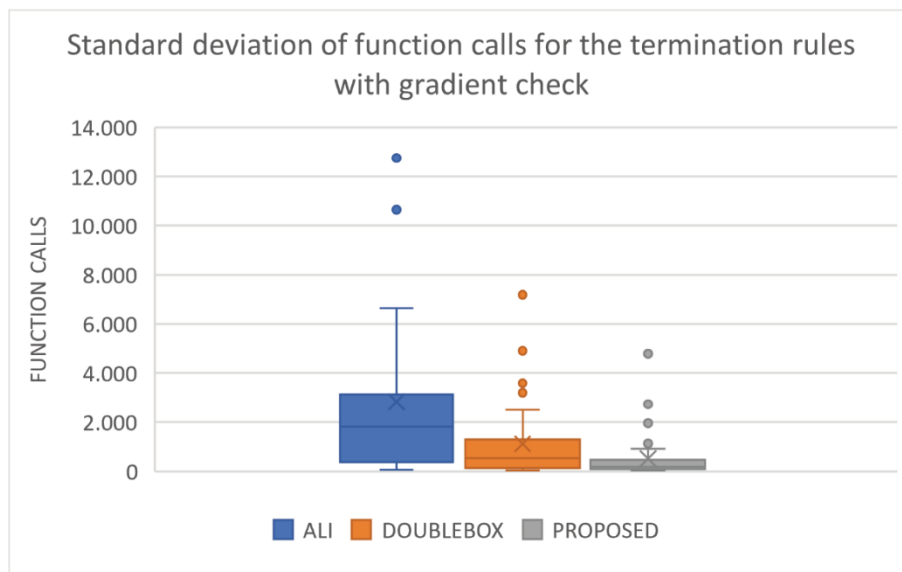
FUNCTION	I1	I2	I3	IP
BF1	2276	2379	2266	2250
BF2	2157	2274	2098	2191
BRANIN	2132	2178	2051	2170
CM4	3098	3717	2538	2791
CAMEL	2198	2335	1974	2058
EASOM	2007	2011	2031	2084
EXP2	1952	2030	1842	1861
EXP4	2046	2266	1877	1909
EXP8	1990	2240	1849	1879
EXP16	1944	2110	1828	1838
EXP32	1953	2126	1859	1867
GKLS250	1982	2079	1850	1900
GKLS2100	1983	2064	1859	1891

GKLS350	1882	1944	1793	1831
GKLS3100	1898	1909(0.97)	1850(0.83)	1833(0.83)
GOLDSTEIN	2523	2670	2110	2164
GRIEWANK2	2893	2885	2791	2681
HANSEN	2766	2879	2731	2804
HARTMAN3	1988	2093	1949	2015
HARTMAN6	3366	3871(0.97)	2767	3133
POTENTIAL3	3312	3487	3613	3892
POTENTIAL4	15392	16390	16223	17497
POTENTIAL5	7109	7104	7732	8477
RASTRIGIN	2591	2648	2474	2732
ROSENBROCK4	8023	12179	4433	6025
ROSENBROCK8	4376	6081	2721	3314
ROSENBCROK16	3643	4954	2746	2485
SHEKEL5	2849	3296	2274	2390
SHEKEL7	2696	3294	2262	2283
SHEKEL10	2624	3251	2338(0.93)	2359
TEST2N4	2536	2637	2427	2782
TEST2N5	2266(0.97)	2336(0.97)	2163(0.90)	2342(0.90)
TEST2N6	2724(0.93)	2832	2694(0.80)	3133(0.90)
TEST2N7	2283(0.80)	2370	2279(0.80)	2585(0.90)
SINU4	2789	3245	2228	2436
SINU8	2601	3151	2233	2348
SINU16	2721	3086	2443	2624
SINU32	4652	5135	4086	4089
TEST30N3	3031	3349	3007	2562
TEST30N4	3747	3797	3250	3237
TOTAL	126999(0.99)	142682(0.99)	115539(0.98)	122742(0.99)

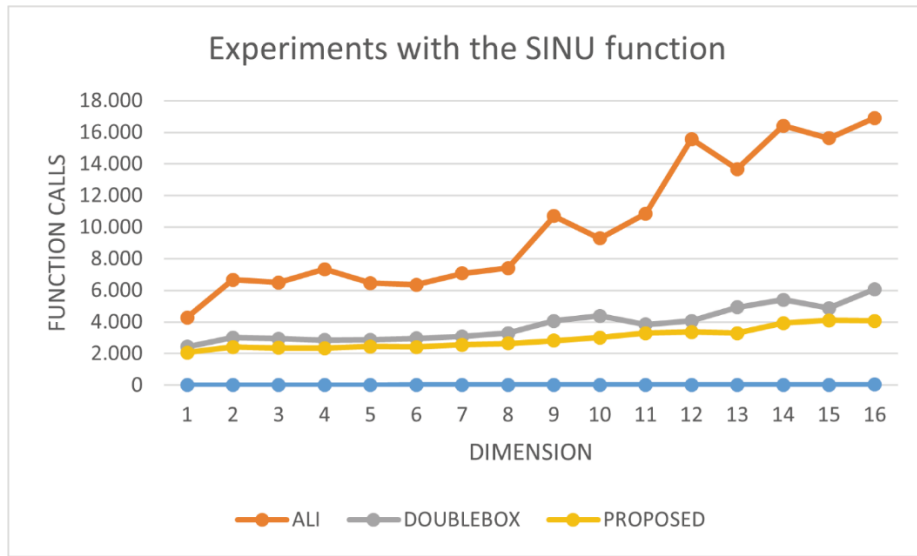
Διάγραμμα 5: Τυπική απόκλιση των κλήσεων συνάρτησης για τους κανόνες τερματισμού χωρίς έλεγχο της κλίσης



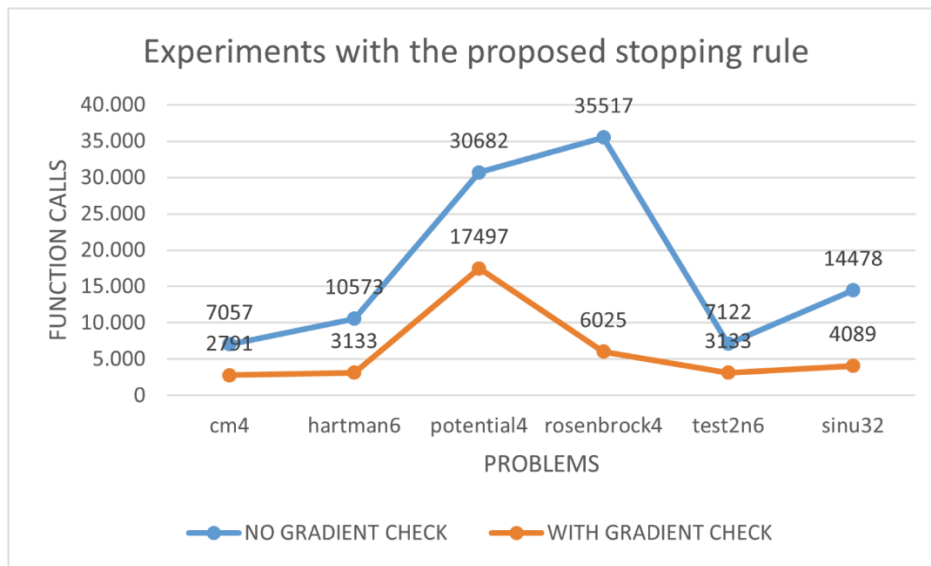
Διάγραμμα 6: Τυπική απόκλιση των κλήσεων συνάρτησης για τους κανόνες τερματισμού με έλεγχο της κλίσης



Διάγραμμα 7: Πειράματα με τη συνάρτηση SINU για μια σειρά από διαστάσεις προβλημάτων από $n = 2$ έως $n = 32$



Διάγραμμα 8: Πειράματα με τον προτεινόμενο κανόνα τερματισμού με έλεγχο της κλίσης και χωρίς έλεγχο της κλίσης



5.6 Συμπεράσματα

Συμπερασματικά στην ενότητα αυτή, παρουσιάστηκαν τρεις νέες τροποποιήσεις της μεθόδου PSO για τον εντοπισμό του γενικού ελάχιστου συνεχών και διαφορίσιμων συναρτήσεων. Η πρώτη τροποποίηση επιδιώκει την αλλαγή του υπολογισμού της

ταχύτητας του πληθυσμού για να προκαλέσει μεγάλες αλλαγές όταν η μέθοδος αναζητά συνεχώς νέα τοπικά ελάχιστα και μικρές αλλαγές όταν επικεντρώνεται σε μια ελπιδοφόρα περιοχή ενός γενικού ελάχιστου. Η δεύτερη τροποποίηση περιορίζει τον αριθμό των τοπικών αναζητήσεων μέσω ενός ασυμπτωτικού κριτηρίου που βασίζεται στην παράγωγο. Η τρίτη τροποποίηση εισάγει ένα νέο κριτήριο τερματισμού βασισμένο στην παρατήρηση ότι η μέθοδος δεν θα είναι σε θέση να ανιχνεύσει ένα νέο ελάχιστο από κάποια επανάληψη και έπειτα. Οι προτεινόμενες τροποποιήσεις εφαρμόστηκαν είτε μία-μία είτε όλες μαζί σε συνδυασμό. Τα πειραματικά αποτελέσματα έδειξαν ότι οι τροποποιήσεις μειώνουν σημαντικά τον αριθμό των κλήσεων συνάρτησης ακόμη και όταν δεν χρησιμοποιούνται σε συνδυασμό. Επιπλέον, η πρώτη τροποποίηση μειώνει τον αριθμό των απαιτούμενων κλήσεων αλλά μόνο όταν το κριτήριο για την αποφυγή τοπικών ελαχίστων δεν είναι παρόν. Τέλος, οι τροποποιήσεις δεν μειώνουν την ικανότητα της μεθόδου PSO να εντοπίσει το συνολικό ελάχιστο της συνάρτησης στόχου.

6. ΠΑΡΑΛΛΗΛΕΣ ΤΕΧΝΙΚΕΣ

6.1 Προγραμματισμός με νήματα – βασικές αρχές

Οι σύγχρονες απαιτήσεις στον τομέα της πληροφορικής διαμορφώνουν ένα περιβάλλον όπου η ανάγκη για αποτελεσματική χρήση των πόρων είναι ουσιαστική. Ο ταυτόχρονος προγραμματισμός επιτρέπει την εκτέλεση πολλών εργασιών ταυτόχρονα, εκμεταλλευόμενος τα πολυπύρνα και πολυνηματικά συστήματα. Αντί να περιμένουν οι εφαρμογές την ολοκλήρωση μιας εργασίας πριν αρχίσουν την εκτέλεση μιας άλλης, ο ταυτόχρονος προγραμματισμός επιτρέπει την αποτελεσματική αξιοποίηση των πόρων του συστήματος. Στον ταυτόχρονο προγραμματισμό, υπάρχουν δύο βασικά στοιχεία εκτέλεσης: οι διεργασίες και τα νήματα [108].

Μία διεργασία διαθέτει το δικό της περιβάλλον εκτέλεσης, αφού διαθέτει ένα πλήρες, ιδιωτικό σύνολο βασικών πόρων εκτέλεσης και κυρίως το δικό της χώρο μνήμης. Οι διεργασίες συχνά αντιμετωπίζονται ως συνώνυμο των προγραμμάτων ή των εφαρμογών. Όμως αυτό που βλέπουν οι χρήστες ως μία εφαρμογή συνήθως είναι ένα σύνολο συνεργαζόμενων διεργασιών. Για να διευκολυνθεί η επικοινωνία μεταξύ των διεργασιών, τα περισσότερα λειτουργικά συστήματα υποστηρίζουν την επικοινωνία μεταξύ των διεργασιών όπως είναι οι υποδοχές. Όπως οι διεργασίες, έτσι και τα νήματα παρέχουν ένα περιβάλλον εκτέλεσης, αλλά η δημιουργία ενός νήματος απαιτεί λιγότερους πόρους από ότι η δημιουργία μιας διεργασίας [108].

Τα νήματα υπάρχουν μέσα στις διεργασίες. Κάθε διεργασία έχει τουλάχιστον ένα νήμα. Τα νήματα διεργασιών διαθέτουν διάφορα χαρακτηριστικά που τα καθιστούν ικανά να εκτελούν πολλαπλές εργασίες παράλληλα και να μοιράζονται πόρους. Τα νήματα έχουν ένα νήμα κατάστασης της εκτέλεσης, που περιλαμβάνει την τρέχουσα κατάσταση του νήματος, όπως το αν βρίσκεται σε εκτέλεση, σε αναμονή ή μπλοκαρισμένο. Επίσης, κάθε νήμα έχει μια στοίβα αποθήκευσης, όπου αποθηκεύονται οι τοπικές μεταβλητές και τα δεδομένα των συναρτήσεων κατά την εκτέλεση. Επιπλέον, τα νήματα έχουν στατική αποθήκευση ανά νήμα για τις τοπικές μεταβλητές, επιτρέποντας τη διαχείριση των μεταβλητών που χρησιμοποιούνται από το κάθε νήμα. Παράλληλα, όλα τα νήματα μιας διεργασίας έχουν προσπέλαση στη μνήμη και τους πόρους της διεργασίας, όπως τα

ανοιχτά αρχεία και τους χειριστές συσκευών, όπου μοιράζονται αυτούς τους πόρους μεταξύ τους [109].

Τα νήματα προσφέρουν πολλά πλεονεκτήματα στον προγραμματισμό, τα οποία παρουσιάζονται παρακάτω:

- Μικρό Overhead Δημιουργίας:

Η δημιουργία μιας νέας διεργασίας απαιτεί πολλούς πόρους και χρόνο, καθώς πρέπει να δημιουργηθεί ένα νέο χώρο διευθύνσεων από την αρχή. Αντίθετα, τα νήματα χρησιμοποιούν το ίδιο χώρο διευθύνσεων της υπάρχουσας διεργασίας, κάτι που είναι πολύ πιο αποδοτικό και γρήγορο.

- Γρήγορη Μετάβαση Ελέγχου:

Η εναλλαγή εκτέλεσης μεταξύ διεργασιών είναι αργή και απαιτεί πολύ χρόνο. Αντίθετα, η εναλλαγή μεταξύ νημάτων της ίδιας διεργασίας είναι πολύ πιο γρήγορη, επειδή τα νήματα μοιράζονται τον ίδιο χώρο μνήμης.

- Αποδοτικός Διαμοιρασμός Πόρων:

Τα νήματα μιας διεργασίας μοιράζονται τους ίδιους πόρους (μνήμη, αρχεία κ.λπ.), κάτι που κάνει τον διαμοιρασμό των δεδομένων και των πόρων πολύ πιο αποδοτικό. Αυτό σημαίνει ότι όλα τα νήματα μπορούν να χρησιμοποιούν τις ίδιες μεταβλητές και δεδομένα, χωρίς την ανάγκη προστασίας μεταξύ τους.

- Ταχύτητα Δημιουργίας και Τερματισμού:

Είναι πιο γρήγορο να δημιουργήσεις ή να τερματίσεις ένα νήμα μέσα σε μια υπάρχουσα διεργασία, παρά να δημιουργήσεις ή να τερματίσεις μια νέα διεργασία.

- Αποδοτική Επικοινωνία:

Η επικοινωνία μεταξύ των νημάτων είναι πιο αποδοτική, καθώς όλα τα νήματα έχουν άμεση πρόσβαση στον ίδιο χώρο μνήμης.

- Απλότητα στον Προγραμματισμό:

Το μοντέλο προγραμματισμού με νήματα είναι απλούστερο, επειδή όλα τα νήματα έχουν πρόσβαση στον ίδιο χώρο μνήμης και δεν χρειάζεται να διαχειρίζεσαι ξεχωριστούς χώρους μνήμης.

- Συνδυασμός Παραλληλίας και Σειριακής Εκτέλεσης:

Τα νήματα προσφέρουν έναν καλό συνδυασμό παραλληλίας (για καλύτερη απόδοση) και σειριακής εκτέλεσης, επιτρέποντας εύκολο προγραμματισμό και αποδοτική εκτέλεση [109].

Τα POSIX Threads, γνωστά και ως Pthreads, είναι ένα πρότυπο API που δημιουργήθηκε για να παρέχει έναν ενοποιημένο και φορητό τρόπο διαχείρισης νημάτων σε λειτουργικά συστήματα συμβατά με το POSIX (Portable Operating System Interface). Τα Pthreads δημιουργήθηκαν για να αντιμετωπίσουν τις δυσκολίες φορητότητας που προέκυπταν από τις διαφορετικές υλοποιήσεις νημάτων από τους κατασκευαστές επεξεργαστών. Με την καθιέρωση ενός κοινόχρηστου προτύπου, τα Pthreads επιτρέπουν την ανάπτυξη λογισμικού που μπορεί να εκτελείται απρόσκοπτα σε διαφορετικές πλατφόρμες και λειτουργικά συστήματα [109].

Βασικό χαρακτηριστικό των POSIX Threads, είναι πως δεν απαιτούν ξεχωριστό χώρο διευθύνσεων, με αποτέλεσμα να έχουν ταχύτερη δημιουργία και εναλλαγή εκτέλεσης. Επίσης, βελτιώνουν την απόδοση, επιτρέποντας την παράλληλη εκτέλεση εργασιών σε πολυπύρνα συστήματα, γεγονός που μπορεί να αυξήσει σημαντικά την αποδοτικότητα των εφαρμογών. Παράλληλα, προσφέρουν απλότητα στον συγχρονισμό, που διευκολύνουν την ασφαλή διαχείριση των κοινών πόρων μεταξύ των νημάτων. Τα νήματα, μοιράζονται τον ίδιο χώρο διευθύνσεων, επιτρέποντας την άμεση πρόσβαση στις ίδιες μεταβλητές και δεδομένα, κάτι που βελτιώνει την αποδοτικότητα της επικοινωνίας μεταξύ τους. Τέλος, το πρότυπο POSIX εξασφαλίζει φορητότητα, επιτρέποντας στον κώδικα που γράφεται για ένα σύστημα να μεταφερθεί και να εκτελεστεί σε άλλα συστήματα που υποστηρίζουν το POSIX, χωρίς σημαντικές τροποποιήσεις [109,110].

Ωστόσο η ανάπτυξη πολυνηματικού κώδικα, μπορεί να είναι πιο σύνθετη, καθώς απαιτεί διαχείριση του συγχρονισμού και αποφυγή προβλημάτων. Επίσης, επειδή τα νήματα μοιράζονται τον ίδιο χώρο διευθύνσεων, ένα σφάλμα σε ένα νήμα μπορεί να επηρεάσει την κατάσταση όλων των νημάτων της διεργασίας, αυξάνοντας την πιθανότητα κρίσιμων σφαλμάτων. Τέλος, σε μονοπύρνα συστήματα, η χρήση πολυνηματικού προγραμματισμού μπορεί να μην προσφέρει σημαντική βελτίωση στην απόδοση, καθώς μόνο ένα νήμα μπορεί να εκτελείται κάθε φορά. Γενικότερα, τα POSIX Threads αποτελούν ένα ισχυρό εργαλείο για την υλοποίηση παράλληλων εφαρμογών, προσφέροντας αποδοτικότητα, βελτιωμένη απόδοση και φορητότητα, αν και απαιτούν προσεκτική διαχείριση για την αποφυγή προβλημάτων πολυνηματικού προγραμματισμού [109,110].

6.2 Προγραμματισμός με OpenMPI

Το Message Passing Interface (MPI) είναι μια διεπαφή προγράμματος εφαρμογής που ορίζει ένα μοντέλο παράλληλου υπολογισμού, όπου κάθε παράλληλη διεργασία έχει τη δική της τοπική μνήμη και τα δεδομένα πρέπει να μοιράζονται ρητά, περνώντας μηνύματα μεταξύ των διεργασιών. Η χρήση του MPI επιτρέπει στα προγράμματα να κλιμακώνονται πέρα από τους επεξεργαστές και την κοινόχρηστη μνήμη ενός μόνο υπολογιστικού διακομιστή, στην κατανεμημένη μνήμη και τους επεξεργαστές πολλών υπολογιστικών διακομιστών συνδυαστικά. Ένας παράλληλος κώδικας MPI απαιτεί ορισμένες αλλαγές από τον σειριακό κώδικα, καθώς προστίθενται κλήσεις συνάρτησης MPI για την επικοινωνία δεδομένων και τα δεδομένα πρέπει με κάποιο τρόπο να διαιρεθούν μεταξύ των διεργασιών [111].

Ένα από τα σημαντικότερα πλεονεκτήματα του MPI είναι η φορητότητα. Το MPI έχει υλοποιηθεί για πολλές αρχιτεκτονικές με κατανεμημένη μνήμη, πράγμα που σημαίνει ότι οι χρήστες δεν χρειάζεται να τροποποιούν τον πηγαίο κώδικα όταν μεταφέρουν εφαρμογές σε διαφορετικές πλατφόρμες που υποστηρίζουν το πρότυπο MPI. Επιπλέον, η υλοποίηση του MPI είναι συνήθως βελτιστοποιημένη για το υλικό στο οποίο τρέχει. Οι υλοποιήσεις από τους προμηθευτές μπορεί επίσης να είναι βελτιστοποιημένες για τις εγγενείς δυνατότητες του υλικού, επιτυγχάνοντας έτσι υψηλή απόδοση. Το MPI είναι σχεδιασμένο για υψηλή απόδοση σε μαζικά παράλληλες μηχανές και συστήματα. Η βασική υλοποίηση του MPI-1 περιλαμβάνει περισσότερες από 100 καθορισμένες ρουτίνες, προσφέροντας πλούσια λειτουργικότητα για την ανάπτυξη παράλληλων εφαρμογών. Συνολικά, το MPI παρέχει μια τυποποιημένη, φορητή και αποδοτική λύση για την ανάπτυξη παράλληλων εφαρμογών σε διάφορες αρχιτεκτονικές συστημάτων, καλύπτοντας τις ανάγκες τόσο των προγραμματιστών όσο και των χρηστών [112].

Το “Open Source High Performance Computing” ή αλλιώς «Υπολογισμός Υψηλής Απόδοσης Ανοιχτού Κώδικα» αναφέρεται σε υπολογιστικούς πόρους και αλγόριθμους που είναι διαθέσιμοι στο κοινό και μπορούν να τροποποιηθούν ελεύθερα από οποιονδήποτε. Η απόδοση είναι «υψηλή», επειδή χρησιμοποιούνται πολυπύρρηνοι και πολυεπεξεργαστικοί υπολογιστές, καθώς και προηγμένοι αλγόριθμοι για την επίλυση πολύπλοκων προβλημάτων. Το Open MPI, συνδυάζει την τεχνογνωσία, τις τεχνολογίες και τους πόρους από όλη την κοινότητά του, προσφέροντας πληθώρα πλεονεκτήματα όπως:

- **Ευελιξία:** το Open MPI είναι ευέλικτο και μπορεί να προσαρμοστεί σε διάφορες ανάγκες και περιβάλλοντα.
- **Απόδοση:** Η συνεχής ανάπτυξη και βελτίωση του Open MPI από μια ευρεία κοινότητα, σημαίνει ότι μπορεί να παρέχει υψηλή απόδοση σε ποικίλες εφαρμογές και περιβάλλοντα.
- **Κοινότητα:** Η συμμετοχή τόσο ακαδημαϊκών όσο και των βιομηχανικών εταιρών στην ανάπτυξη και συντήρηση του Open MPI, σημαίνει ότι υπάρχει μια ευρεία και ενεργή κοινότητα που μπορεί να παρέχει υποστήριξη και επίλυση προβλημάτων.
- **Διαθεσιμότητα:** Το Open MPI είναι δωρεάν για χρήση, καθιστώντας το προσιτό για οργανισμούς και ερευνητικά εργαστήρια με περιορισμένους πόρους.
- **Αξιοπιστία:** Η ενεργή συντήρηση και υποστήριξη από την κοινότητα διασφαλίζει ότι το Open MPI παρέχει αξιόπιστη λειτουργία σε ποικίλα περιβάλλοντα και εφαρμογές [113].

6.3 Παράλληλες τεχνικές πολλαπλών εκκινήσεων

Η παράλληλη βελτιστοποίηση αποτελεί μία σημαντική προσέγγιση στον τομέα της επίλυσης προβλημάτων βελτιστοποίησης και εφαρμόζεται σε ένα ευρύ φάσμα εφαρμογών, όπως για παράδειγμα στη βελτιστοποίηση των παραμέτρων μοντέλων μηχανικής μάθησης [114,115,116,117] στη σχεδίαση και στη βελτιστοποίηση συστημάτων ελέγχου [118,119,120], στη διαχείριση ενέργειας και πόρων [121,122,123,124] καθώς και σε προβλήματα που αφορούν την αιεφόρο ανάπτυξη και στην ενίσχυση της βιωσιμότητας [125,126,127].

Αξιοποιώντας παράλληλα πολλούς υπολογιστικούς πόρους, η παράλληλη βελτιστοποίηση επιτρέπει την ταυτόχρονη εκτέλεση πολλών αλγορίθμων, την επίτευξη ταχύτερης σύγκλισης και βελτιωμένης απόδοσης. Παράλληλα, η αξιοποίηση πολλαπλών πόρων επιτρέπει την αντιμετώπιση εξαιρέσεων και σφαλμάτων με μεγαλύτερη αποτελεσματικότητα, ενώ η μεγαλύτερη υπολογιστική ισχύ για την εκτέλεση περισσότερων δοκιμών ή τη χρήση πιο πολύπλοκων μοντέλων οδηγεί σε βελτιωμένη απόδοση [128]. Φυσικά, η διαδικασία αυτή απαιτεί την ανάπτυξη κατάλληλων αλγορίθμων και τεχνικών που να επιτρέπουν την αποτελεσματική διαχείριση και

εκμετάλλευση των διαθέσιμων πόρων. Κάθε αλγόριθμος παράλληλης βελτιστοποίησης απαιτεί μια συνεκτική στρατηγική για τη διανομή του φορτίου εργασίας μεταξύ των διαθέσιμων πόρων, καθώς και έναν αποδοτικό τρόπο συλλογής και αξιολόγησης των αποτελεσμάτων.

Βασικές μέθοδοι παράλληλης βελτιστοποίησης που χρησιμοποιούνται σε πληθώρα εφαρμογές λόγω των ιδιοτήτων τους και των οφελών που προσφέρουν είναι η Particle Swarm Optimization (PSO), η Multistart και η Differential Evolution (DE). Η PSO είναι μία ευέλικτη μέθοδος εμπνευσμένη από τη συμπεριφορά των σμήνων στη φύση, η οποία εφαρμόζει ένα σύνολο από ατομικά σωματίδια που αναζητούν τη βέλτιστη λύση μέσω συνεργασίας και αλληλεπίδρασης. Η Multistart αποτελεί μία απλή και αποτελεσματική μέθοδο εκκίνησης πολλαπλών τοπικών αλγορίθμων βελτιστοποίησης από διαφορετικά σημεία του χώρου αναζήτησης. Τέλος, η DE είναι μία πολύ αποτελεσματική εξελικτική μέθοδος που μιμείται τη φυσική εξέλιξη, εφαρμόζοντας λειτουργίες μετάλλαξης, διασταύρωσης και επιλογής σε ένα πληθυσμό λύσεων για την αναζήτηση της βέλτιστης. Με τη χρήση αυτών των μεθόδων, η παράλληλη βελτιστοποίηση μπορεί να επιτύχει ταχύτερη σύγκλιση και βελτιωμένη απόδοση σε πολύπλοκα προβλήματα βελτιστοποίησης.

6.4 Παράλληλη τεχνική DE

Η Μέθοδος Διαφορικής Εξέλιξης (Differential Evolution Method -DE) προτάθηκε αρχικά από τον Storn [129] και έπειτα τροποποιήθηκε σε διάφορες ερευνητικές εργασίες [130,131,132]. Τα βασικά βήματα του αλγορίθμου DE παρουσιάζονται παρακάτω:

1. Είσοδος:

- α. Το μέγεθος της πληθυσμιακής ομάδας $NP \geq 4$. Τα μέλη αυτής της ομάδας ονομάζονται πράκτορες.
- β. Η πιθανότητα διασταύρωσης $CR \in [0,1]$.
- γ. Το διαφορικό βάρος $F \in [0,2]$.

2. Έξοδος:

- α. Ο πράκτορας x_{best} με την χαμηλότερη τιμή της συνάρτησης $f(x_{best})$.
3. Αρχικοποίηση όλων των πρακτόρων στο S.

4. Όσο τα κριτήρια τερματισμού δεν έχουν επιτευχθεί **επανάλαβε**

α. Για $i=1 \dots NP$ επανάλαβε

- i. **Επέλεξε** ως x τον πράκτορα i
- ii. **Επέλεξε** τυχαία τρεις πράκτορες a, b, c με την ιδιότητα ότι $a \neq b, b \neq c, c \neq a$.
- iii. **Επέλεξε** μία τυχαία θέση R από το σύνολο $\{1, 2, \dots, n\}$.
- iv. **Δημιουργία** διανύσματος $y = [y_1, y_2, \dots, y_n]$ με την ακόλουθη διαδικασία
- v. **Για $j = 1, \dots, n$ επανάλαβε**
 - α. **Όρισε** $r_i \in [0, 1]$ έναν τυχαίο αριθμό
 - β. **Εάν $r_j < CR$ ή $j = R$, τότε $y_i = \alpha_j + F \times (b_j - c_j)$ διαφορετικά $y_j = x_j$**
- vi. Εάν $y \in S$ και $f(y) \leq f(x)$ τότε $x=y$
- vii. Τέλος επανάληψης

β. Τέλος επανάληψης

5. Τέλος επανάληψης

Στην προτεινόμενη διαδικασία, ο πληθυσμός των πρακτόρων διαιρείται σε N ανεξάρτητα συνεχή τμήματα, όπου κάθε τμήμα ονομάζεται νησί. Παραδείγματος χάρη, αν υπάρχουν 10 πράκτορες και 2 νησιά, τότε οι πράκτορες 1–5 ανατίθενται στο νησί 1 και οι πράκτορες 6–10 στο νησί 2. Σε κάθε νησί, η διαδικασία της διαφορικής εξέλιξης πραγματοποιείται ανεξάρτητα, ενώ θα πρέπει να υπάρχει μηχανισμός για επικοινωνία μεταξύ των νησιών, καθώς και κατάλληλος μηχανισμός για τον τερματισμό της συνολικής διαδικασίας. Ο προτεινόμενος αλγόριθμος παρουσιάζεται παρακάτω [133]:

1. **Είσοδος:**

- a. Οι παράμετροι NP, CR, F .
- b. Η ακέραια παράμετρος N , που αντιπροσωπεύει τον αριθμό των νησιών.
- c. Η ακέραια παράμετρος NR , που αντιπροσωπεύει το ρυθμό διασποράς.
- d. Η ακέραια παράμετρος NI , που αντιπροσωπεύει τον αριθμό των νησιών που πρέπει να τερματίσουν για να τερματιστεί η συνολική διαδικασία.

2. **Έξοδος:**

- a. Ο πράκτορας x_best με την χαμηλότερη τιμή της συνάρτησης $f(x_best)$.
3. **Αρχικοποίηση** όλων των πρακτόρων στο S .
4. **Όρισε** την μεταβλητή $iter=1$

5. Για $i=1, \dots, N$ επανάλαβε παράλληλα

a. Εκτέλεσε για κάθε νησί το βήμα 4α του βασικού αλγορίθμου DE

6. Τέλος επανάληψης

7. Εάν $iter \bmod N_R = 0$, εφαρμογή του μηχανισμού διάδοσης στα νησιά. Η προεπιλεγμένη τιμή που χρησιμοποιείται στα πειράματα είναι η περίπτωση "1 προς 1".

8. Όρισε $iter=iter+1$

9. Εάν ο κανόνας τερματισμού δεν είναι έγκυρος, τότε πήγαινε στο βήμα 5.

10. Εφάρμοσε τη διαδικασία της τοπικής αναζήτησης στο x_{best} . Η διαδικασία τοπικής αναζήτησης που χρησιμοποιείται στην προτεινόμενη μέθοδο είναι η παραλλαγή BFGS του Powell [134].

Μηχανισμός Διάδοσης:

Κατά τη διάρκεια του μηχανισμού διάδοσης, οι καλύτερες τιμές των νησιών διαδίδονται στα υπόλοιπα αντικαθιστώντας τις χειρότερες τιμές τους. Υπάρχουν οι ακόλουθες περιπτώσεις διάδοσης:

Ένα προς ένα. Σε αυτήν την περίπτωση, ένα τυχαίο νησί θα στείλει σε ένα άλλο τυχαία επιλεγμένο νησί την καλύτερη τιμή του.

Ένα προς N. Σε αυτήν την περίπτωση, ένα τυχαίο νησί θα στείλει την καλύτερή του τιμή σε όλα τα άλλα νησιά.

N προς ένα. Σε αυτήν την περίπτωση, όλα τα νησιά θα ενημερώσουν ένα τυχαία επιλεγμένο νησί για την καλύτερη τιμή τους.

N προς N. Όλα τα νησιά θα ενημερώσουν όλα τα άλλα νησιά για την καλύτερη τιμή τους.

Κανόνας τερματισμού:

Στον προτεινόμενο κριτήριο τερματισμού, ένα απλό κριτήριο ελέγχεται ξεχωριστά σε κάθε νησί. Για κάθε νησί i , η διαφορά:

$$\delta_i^{(k)} = \left| \delta_{i,min}^{(k)} - f_{i,min}^{(k-1)} \right|$$

μετριέται σε κάθε επανάληψη k όπου :

$f_{i,min}^{(k-1)}$, είναι η καλύτερη τιμή της συνάρτησης που εντοπίζεται στο νησί i στην επανάληψη

k . Εάν $\delta_i^{(k)} \leq \epsilon$ για τουλάχιστον M διαδοχικές επαναλήψεις, τότε πιθανότατα το νησί i θα πρέπει να τερματίσει την εξέλιξη του πληθυσμού. Στην προτεινόμενη τεχνική, εάν η

ποσότητα της παραπάνω εξίσωσης ισχύει για περισσότερα από ΝΙ νησιά, τότε ο συνολικός αλγόριθμος τερματίζει.

6.5 Παράλληλη τεχνική PSO

Στον τομέα της παγκόσμιας βελτιστοποίησης, έχουν αναπτυχθεί διάφορες τεχνικές για τον εντοπισμό του καθολικού ελάχιστου. Μία από τις πιο διαδεδομένες τεχνικές παγκόσμιας βελτιστοποίησης είναι η τεχνική βελτιστοποίησης σμήνους σωματιδίων ή αλλιώς Particle Swarm Optimization -PSO. Ο βασικός αλγόριθμος παρουσιάζεται παρακάτω:

1. Βήματα Αρχικοποίησης

- a. **Όρισε** $iter=0$
- b. **Όρισε** m το συνολικό αριθμό των σωματιδίων
- c. **Όρισε** $iter_{max}$ τον μέγιστο αριθμό των επιτρεπόμενων επαναλήψεων για τον σειριακό αλγόριθμο
- d. **Αρχικοποίηση** τυχαίων θέσεων των σωματιδίων x_1, x_2, \dots, x_m . Οι αρχικές θέσεις πρέπει να είναι εντός του πεδίου ορισμού της συνάρτησης στόχου.
- e. Τυχαία **αρχικοποίηση** των ταχυτήτων u_1, u_2, \dots, u_m .
- f. **Για** $i=1..m$ **επανάλαβε** $p_i = x_i$. Το διάνυσμα p_i αντιπροσωπεύει την καλύτερη θέση του σωματιδίου i .
- g. **Όρισε** $p_{best} = argmin_{i \in 1..m} f(x_i)$

2. **Βήματα Ελέγχου Τερματισμού.** Πραγματοποιείται ο έλεγχος του κριτηρίου τερματισμού για τους σειριακούς αλγορίθμους. Αν τα κριτήρια τερματισμού είναι αληθή, τότε η μέθοδος τερματίζει.

3. **Για** $i=1..m$ **επανάλαβε**

- a. **Υπολόγισε** την ταχύτητα u_i ως το συνδυασμό των διανυσμάτων u_i, p_i και p_{best}
- b. **Όρισε** τη νέα θέση για το σωματίδιο ως: $x_i = x_i + u_i$
- c. **Υπολόγισε** την $f(x|i)$ του σωματιδίου κάνοντας χρήση της αντικειμενικής συνάρτησης $f(x)$.

d. Εάν $f(x_i) \leq f(p|i)$, τότε $p_i = x_i$

4. Τέλος επανάληψης
5. Όρισε $p_{best} = \operatorname{argmin}_{i \in 1..m} f(x_i)$
6. Όρισε $\text{iter} = \text{iter} + 1$
7. Επέστρεψε στο βήμα 2

Η προτεινόμενη μέθοδος βασίζεται στην αναπαράσταση του προβλήματος ως ένα σμήνος σωματιδίων, όπου κάθε σωματίδιο αντιπροσωπεύει μια υποψήφια λύση στον Ευκλείδειο χώρο αναζήτησης. Κατά τη διάρκεια της διαδικασίας βελτιστοποίησης, τα σωματίδια κινούνται στον χώρο αναζήτησης με βάση την ατομική τους εμπειρία καθώς και με βάση τις κοινωνικές τους αλληλεπιδράσεις. Χρησιμοποιώντας αυτές τις δύο επιρροές, τα σωματίδια προσαρμόζουν τις ταχύτητές τους και τις θέσεις τους κατά τις επαναλήψεις του. Ο αλγόριθμος που προτείνεται παρουσιάζεται παρακάτω [135]:

1. Όρισε ως N_I τον μέγιστο αριθμό μονάδων παράλληλης επεξεργασίας
2. Όρισε ως N_R τον αριθμό των επαναλήψεων, μετά από τις οποίες κάθε μονάδα επεξεργασίας θα στείλει τα καλύτερά της σωματίδια στις υπόλοιπες μονάδες επεξεργασίας
3. Όρισε ως N_P τον αριθμό των μετακινηθέντων σωματιδίων μεταξύ των παράλληλων μονάδων επεξεργασίας.
4. Όρισε $K=0$ τον αριθμό της επανάληψης
5. Για $j=1, \dots, N$ επανάλαβε παράλληλα
 - a. Εκτέλεσε μια επανάληψη του αλγορίθμου PSO που περιγράφεται στον βασικό αλγόριθμο στη μονάδα επεξεργασίας j .
 - b. Εάν $K \bmod N_R = 0$, τότε:
 - i. Λάβε τα καλύτερα N_P σωματίδια από τον αλγόριθμο j .
 - ii. Μετάδωσε αυτά τα N_P σωματίδια στις υπόλοιπες μονάδες επεξεργασίας χρησιμοποιώντας κάποιο σχήμα μετάδοσης που θα αναλυθεί στη συνέχεια.
 - c. Τέλος αν.
6. Τέλος επανάληψης
7. Ενημέρωση $K=K+1$

8. Έλεγχος του προτεινόμενου κανόνα τερματισμού. Αν ο κανόνας τερματισμού είναι έγκυρος, τότε μεταβείτε στο βήμα 5.

- a. **Τερματισμός και αναφορά** της καλύτερης τιμής από όλες τις μονάδες επεξεργασίας. Εφαρμογή μιας διαδικασίας τοπικής αναζήτησης στην εντοπισμένη τιμή για να βελτιωθεί το εντοπισμένο καθολικό ελάχιστο. Στο προτεινόμενο αλγόριθμο χρησιμοποιήθηκε μια διαδικασία τοπικής αναζήτησης που είναι μια παραλλαγή της BFGS του Powell [134].

6.6 Αντικειμενικές Συναρτήσεις

Στα πειράματα που πραγματοποιήθηκαν, χρησιμοποιήθηκαν συγκεκριμένες αντικειμενικές συναρτήσεις που παρουσιάζονται παρακάτω (Πίνακας 16: «Αντικειμενικές συναρτήσεις που χρησιμοποιήθηκαν»):

Πίνακας 16: Αντικειμενικές συναρτήσεις που χρησιμοποιήθηκαν

Όνομα	Τύπος	Μέγεθος
Cigar	$f(x) = x_1^2 + 10^6 \sum_{i=2}^n x_i^2$	n=10
Bf1	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) \cos(4\pi x_2) + \frac{7}{10}$	n=2
Bf2	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) \cos(4\pi x_2) + \frac{3}{10}$	n=2
Branin	$f(x) = \sum_{i=1}^n x_i^2 - \frac{1}{10} \sum_{i=1}^n \cos(5\pi x_i)$	n=2
CM4	$f(x) = \sum_{i=1}^n x_i^2 - \frac{1}{10} \sum_{i=1}^n \cos(5\pi x_i)$	n=4
Camel	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	n=2
Discus10	$f(x) = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$	n=10
Eason	$f(x) = -\cos(x_1) \cos(x_2) \exp((x_2 - \pi)^2 - (x_1 - \pi)^2)$	n=2
Exp	$f(x) = -\exp\left(0.5 \sum_{i=1}^n x_i^2\right), -1 \leq x_i \leq 1$	n=4, 16, 64

Griewank2	$f(x) = 1 + \frac{1}{200} \sum_{i=1}^2 x_1^2 - \prod_{i=1}^2 \frac{\cos(x_i)}{\sqrt{(i)}}$	n=2
Gkls[94]	$f(x) = Gkls(x, n, w)$	n=2, 3
Hansen	$f(x) = \sum_{i=1}^5 i \cos[(i-1)x_i + i] \sum_{j=1}^5 j \cos[(j+1)x_2 + j]$	n=2
Hartman 3	$f(x) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	n=3
Hartman 6	$f(x) = - \sum_{i=1}^4 c_i \exp(- \sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	n=6
Potential[95]	$V_{LJ}(r) = 4 \epsilon [(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^6]$	n=3, 5
Rastrigin	$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$	n=2
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	n=4, 8
Shekel 5	$f(x) = - \sum_{i=1}^5 \frac{1}{(x - a_i)(x - a_i)^T + c_i}$	n=4
Shekel 7	$f(x) = - \sum_{i=1}^7 \frac{1}{(x - a_i)(x - a_i)^T + c_i}$	n=4
Shekel 10	$f(x) = - \sum_{i=1}^{10} \frac{1}{(x - a_i)(x - a_i)^T + c_i}$	n=4
Sinu	$f(x) = -(2.5 \prod_{i=1}^n \sin(x_i - z) + \prod_{i=1}^n \sin(5(x_i - z)))$	n=4, 8
Test2n	$f(x) = \frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$	n=4, 5, 6, 7
Test30n	$\frac{1}{10} \sin^2(3\pi x_1) \sum_{i=2}^{n-1} ((x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n))$	n=3, 4

6.7 Πειραματικά αποτελέσματα

Παράλληλη τεχνική DE

Για να αξιολογηθεί η απόδοση της τροποποιημένης έκδοσης της τεχνικής της διαφορικής εξέλιξης, διεξήχθησαν πειράματα στα οποία ο αριθμός των παράλληλων μονάδων υπολογισμού ποικίλλουν από 1 έως 10. Χρησιμοποιήθηκε η δωρεάν διαθέσιμη βιβλιοθήκη OpenMP για την παραλληλοποίηση, και η μέθοδος προγραμματίστηκε σε ANSI C++ μέσα στο πακέτο βελτιστοποίησης OPTIMUS που είναι διαθέσιμο από το <https://github.com/itsoulos/OPTIMUS> (προσπελάστηκε στις 4 Ιανουαρίου 2023). Όλα τα πειράματα πραγματοποιήθηκαν σε έναν επεξεργαστή AMD Ryzen 5950X με 128 GB μνήμη RAM και λειτουργικό σύστημα Debian Linux. Θα πρέπει να αναφερθεί επίσης, ότι όλα τα πειράματα διενεργήθηκαν 30 φορές χρησιμοποιώντας διαφορετικό σπόρο για τον τυχαίο γεννήτορα κάθε φορά και αναφέρθηκαν οι μέσες τιμές. Οι τιμές των παραμέτρων που χρησιμοποιήθηκαν στον αλγόριθμο DE φαίνονται στον Πίνακα 17: «Τιμές παραμέτρων», ενώ η παράμετρος F (διαφορικό βάρος) υπολογίστηκε ως:

$$F = \frac{-1}{2} + 2 \times R, \text{ όπου } R \in [0,1] \text{ τυχαίος αριθμός.}$$

Πίνακας 17: Τιμές παραμέτρων

Παράμετροι	Τιμή
NP	200 πράκτορες
Propagation	1- προς -1 μέθοδο
N_R	5 επαναλήψεις
N_l	2 νησιά
CR	0.9 για την πιθανότητα διασταύρωσης
M	15 επαναλήψεις
ϵ	10^{-4}

Επιπλέον, πραγματοποιήθηκαν επιπλέον πειράματα για να διαπιστωθεί εάν υπήρχε διαφοροποίηση χρησιμοποιώντας τις διαφορετικές τεχνικές διάδοσης. Χρησιμοποιήθηκαν 10 νήματα επεξεργασίας. Σε κάθε νήμα επεξεργασίας, όπως και πριν, ο πληθυσμός κάθε νησιού αποτελούνταν από 20 πράκτορες. Τα αποτελέσματα από αυτά τα πειράματα φαίνονται στον ακόλουθο πίνακα (Πίνακας 18: «Πειράματα για την προτεινόμενη μέθοδο χρησιμοποιώντας διαφορετικές επιλογές για τη μέθοδο διάδοσης. Ο αριθμός των νημάτων

επεξεργασίας ορίστηκε σε 10. Οι αριθμοί στα κελιά αντιπροσωπεύουν τον μέσο αριθμό κλήσεων συνάρτησης για κάθε δοκιμαστική συνάρτηση»):

Πίνακας 18: Πειράματα για την προτεινόμενη μέθοδο χρησιμοποιώντας διαφορετικές επιλογές για τη μέθοδο διάδοσης. Ο αριθμός των νημάτων επεξεργασίας ορίστηκε σε 10. Οι αριθμοί στα κελιά αντιπροσωπεύουν τον μέσο αριθμό κλήσεων συνάρτησης για κάθε δοκιμαστική συνάρτηση

ΣΥΝΑΡΤΗΣΗ	1 ΠΡΟΣ 1	1 ΠΡΟΣ N	N ΠΡΟΣ 1	N ΠΡΟΣ N
BF1	4887	4259	4209	2792
BF2	4577	4021	3917	2691
BRANIN	3895	3378	3307	2382
CIGAR10	1875	1874	1871	1873
CM4	2243 (0.97)	2173	2136 (0.97)	2030
DISCUS10	1807	1810	1808	1809
EASOM	1791	1790	1791	1789
ELP10	29944	42025	22876	19117
EXP4	1806	1807	1811	1806
EXP16	1830	1829	1828	1824
EXP64	1838	1838	1838	1836
GKLS250	1818	1812	1810	1802
GKLS350	2195	2163	2109	2011 (0.97)
GRIEWANK2	4208	3620	3514	2445 (0.80)
POTENTIAL3	5598	4445	4353	2521
POTENTIAL5	8620	7475	7025	3374
HANSEN	9242	6075	6181	3135
HARTMAN3	2883	2664	2593	2207
HARTMAN6	4877 (0.97)	4327 (0.83)	4362 (0.80)	2834 (0.90)
RASTRIGIN	3707	3213	2870	2220 (0.90)
ROSENBROCK4	9473	8294	7883	7084
ROSENBROCK8	20654	24470	15919	19272
SHEKEL5	5065 (0.80)	7556	5386 (0.93)	4456 (0.70)
SHEKEL7	6996	7207 (0.90)	6488 (0.93)	4493 (0.80)

SHEKEL10	7283	6812 (0.93)	6440	3916 (0.73)
SINU4	4434	4204	4020	2796 (0.97)
SINU8	4523	5386	4605	3341 (0.90)
TEST2N4	6834	5777	5625	3609 (0.97)
TEST2N5	8050	6695 (0.97)	6647	4179 (0.73)
TEST2N6	9175	7770 (0.93)	7660	4522 (0.53)
TEST2N7	9760	9259 (0.77)	9081	5200 (0.57)
TEST30N3	2994	2814	2653	2210
TEST30N4	3031	2797	2700	2107
ΣΥΝΟΛΟ	197913 (0.99)	201649 (0.98)	167316 (0.98)	129683 (0.91)

Από τα πειραματικά αποτελέσματα, φαίνεται ότι δεν υπήρχαν σημαντικές αλλαγές στον συνολικό αριθμό κλήσεων συνάρτησης εκτός από την περίπτωση μιας διάδοσης "N σε N". Υπήρξε μια σημαντική μείωση των κλήσεων συνάρτησης, αλλά και μια μείωση της αξιοπιστίας των τεχνικών από 99% σε 91%. Αυτό μπορεί να οφείλεται στο γεγονός ότι, λόγω της ανταλλαγής των καλύτερων κόστων μεταξύ όλων των νησιών, ο συνολικός πληθυσμός εγκλωβίστηκε σε τοπικά ελάχιστα.

Επίσης, η προτεινόμενη μέθοδος συγκρίθηκε με την αρχική μέθοδο διαφορικής εξέλιξης και δύο παραλλαγές από τη σχετική βιβλιογραφία, αναφερόμενες ως DERL και DELB. Οι μέθοδοι DERL και DELB είναι παραλλαγές της μεθόδου διαφορικής εξέλιξης και χρησιμοποιείται για την εύρεση της βέλτιστης λύσης σε προβλήματα με συνεχείς μεταβλητές. Συγκεκριμένα, η μέθοδος DERL (Differential Evolution with Random Localization) εισάγει την έννοια της τυχαίας τοπικοποίησης. Σε αυτή την παραλλαγή, χρησιμοποιείται ένας τυχαίος παράγοντας για τη δημιουργία νέων λύσεων, με στόχο τη βελτίωση της εξερεύνησης του χώρου αναζήτησης και την αποφυγή τοπικών βέλτιστων. Η μέθοδος DELB ((Differential Evolution with Local Best), είναι μια άλλη παραλλαγή της διαφορικής εξέλιξης, η οποία επικεντρώνεται στην εκμετάλλευση τοπικών βέλτιστων λύσεων. Αυτή η προσέγγιση χρησιμοποιεί πληροφορίες από τις καλύτερες λύσεις που έχουν βρεθεί τοπικά στον πληθυσμό, προκειμένου να καθοδηγήσει τη διαδικασία αναζήτησης προς βελτιωμένες λύσεις.

Τα αποτελέσματα από αυτήν τη σύγκριση φαίνονται στον Πίνακα 19: «Σύγκριση της προτεινόμενης μεθόδου έναντι άλλων παραλλαγών της τεχνικής διαφορικής εξέλιξης».

Πίνακας 19: Σύγκριση της προτεινόμενης μεθόδου έναντι άλλων παραλλαγών της τεχνικής διαφορικής εξέλιξης

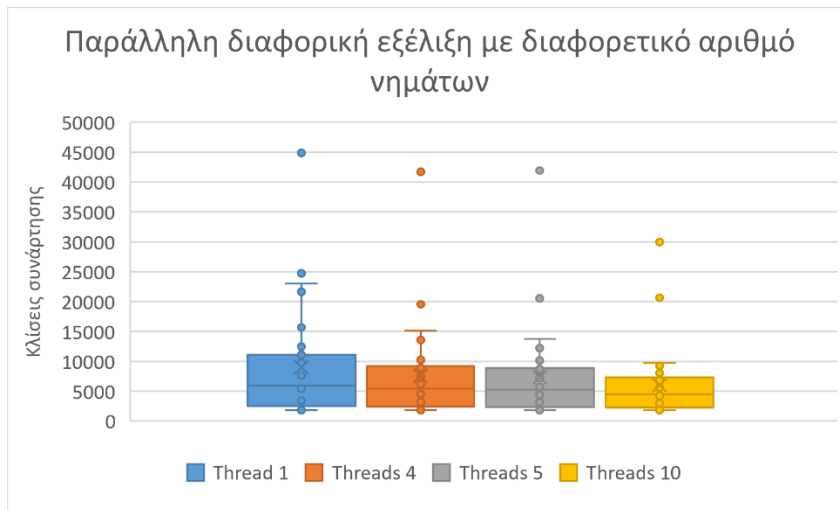
ΣΥΝΑΡΤΗΣΗ	ΠΡΟΤΕΙΝΕΤΑ I	ΑΡΧΙΚ Η DE	DERL	DELB
BF1	4887	5516	2881	5319
BF2	4577	5555	2895	5405
BRANIN	3895	5656	2857	4830
CIGAR10	1875	88396	66161	58460
CM4	2243 (0.97)	9107	3856	6014
DISCUS10	1807	87657	55722	49014
EASOM	1791	7879	7225	14934
ELP10	299444	33371	9345	39890
EXP4	1806	6027	2638	4142
EXP16	1830	26194	25117	11740
EXP64	1838	26497	27831	18346
GKLS250	1818	3800	1983	3706
GKLS350	2195	6206	2901	5027
GRIEWANK2	4208	6365	3325	6165
POTENTIAL3	5598	82933	111496	44592
POTENTIAL5	8620	24118	61694	46557
HANSEN	9242	18470	7123	12212
HARTMAN3	2883	4655	2205	4124
HARTMAN6	4877 (0.97)	15488	5343	7215
RASTRIGIN	3707	6362	3102	5704
ROSENBROCK4	9473	16857	6679	10411
ROSENBROCK8	20654	56445	17198	22939
SHEKEL5	5065 (0.80)	13079	5224	8167
SHEKEL7	6996	12409	4994	8093

SHEKEL10	7283	13238	5240	8822
SINU4	4434	8977	3828	6052
SINU8	4523	28871	9318	10157
TEST2N4	6834	10764	4529	7331
TEST2N5	8050	15568	5917	8969
TEST2N6	9175	21185	7613	10648
TEST2N7	9760	28411	9492	12252
TEST30N3	2994	4965	2758	4693
TEST30N4	3031	5123	2688	5153
ΜΕΣΗ ΤΙΜΗ	197913 (0.99)	706144	491178	477083

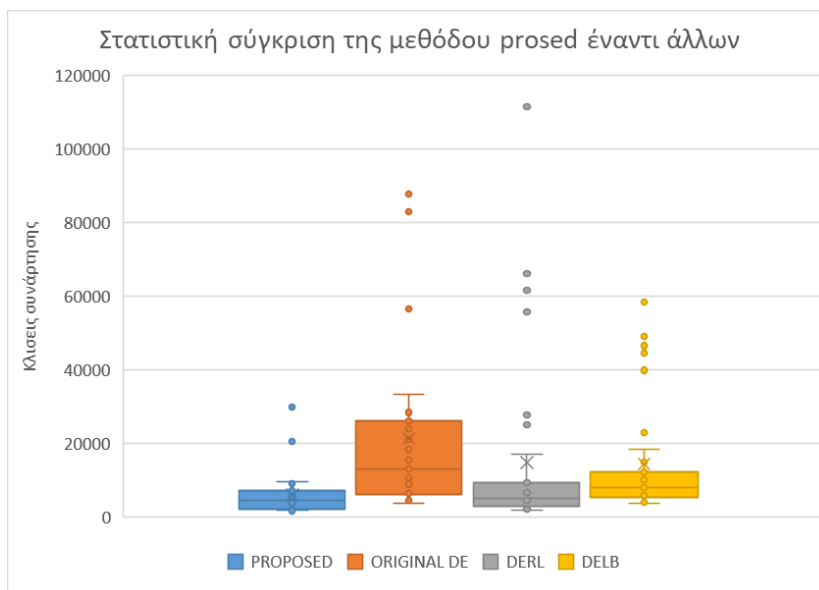
Όπως φαίνεται, η προτεινόμενη τεχνική υπερτερεί σημαντικά έναντι των άλλων τροποποιήσεων της μεθόδου της διαφορικής εξέλιξης. Αυτό οφείλεται κυρίως στη διαφορετική τεχνική υπολογισμού του διαφορικού βάρους, αλλά και στην προτεινόμενη μέθοδο τερματισμού. Η τεχνική υπολογισμού του διαφορικού βάρους κατά κύριο λόγο κατόρθωσε να παρέχει μια καλύτερη αναζήτηση του χώρου αναζήτησης, ενώ η νέα μέθοδος τερματισμού τερμάτισε τη μέθοδο βελτιστοποίησης στον κατάλληλο χρόνο. Επιπλέον, αυτή η νέα τεχνική τερματισμού τροποποιήθηκε για να λειτουργεί καλά και σε περιβάλλοντα παράλληλου υπολογισμού.

Επιπλέον, πραγματοποιήθηκε μια στατιστική σύγκριση για την προτεινόμενη μέθοδο και διαφορετικούς αριθμούς νημάτων επεξεργασίας, και τα αποτελέσματα παρουσιάζονται στο Διάγραμμα 9: «Στατιστική σύγκριση για την προτεινόμενη μέθοδο και διαφορετικούς αριθμούς νημάτων». Επιπλέον, συμπεριλήφθηκε μια στατιστική σύγκριση για τα αποτελέσματα της προτεινόμενης μεθόδου έναντι των άλλων παραλλαγών της μεθόδου DE και το αντίστοιχο γράφημα φαίνεται στο Διάγραμμα 10: «Στατιστική σύγκριση για τα αποτελέσματα της προτεινόμενης μεθόδου έναντι διαφορετικών παραλλαγών της μεθόδου DE».

Διάγραμμα 9: Στατιστική σύγκριση για την προτεινόμενη μέθοδο και διαφορετικούς αριθμούς νημάτων.



Διάγραμμα 10: Στατιστική σύγκριση για τα αποτελέσματα της προτεινόμενης μεθόδου έναντι διαφορετικών παραλλαγών της μεθόδου DE



Παράλληλη τεχνική PSO

Κάθε πείραμα εκτελέστηκε 30 φορές χρησιμοποιώντας διαφορετικούς τυχαίους αριθμούς κάθε φορά. Για την παραλληλοποίηση χρησιμοποιήθηκε η δωρεάν διαθέσιμη βιβλιοθήκη OpenMP [136], ενώ η γλώσσα προγραμματισμού που χρησιμοποιήθηκε ήταν ANSI C++ και το πακέτο βελτιστοποίησης OPTIMUS, το οποίο είναι διαθέσιμο δωρεάν στη διεύθυνση <https://github.com/itsoulos/OPTIMUS> (προσπελάστηκε στις 3 Μαρτίου 2023), χρησιμοποιήθηκε. Οι πιο κρίσιμες παράμετροι της προτεινόμενης μεθόδου αναφέρονται στον Πίνακα 20: «Παράμετροι της προτεινόμενης μεθόδου»:

Πίνακας 20: Παράμετροι της προτεινόμενης μεθόδου

Παράμετροι	Ερμηνεία	Τιμή
M	Αριθμός σωματιδίων	200
$iter_{max}$	Μέγιστος αριθμός γενεών	200
c_1	Γνωστική παράμετρος για PSO	1.0
c_2	Παράμετρος κοινωνικής μάθησης για την PSO	1.0
N_R	Συχνότητα ανταλλαγής σωματιδίων	15
ϵ	Μικρή τιμή που χρησιμοποιείται στον κανόνα τερματισμού	10^{-6}
N_M	Αριθμός επαναλήψεων χρησιμοποιώντας κανόνα τερματισμού	15

Για να αξιολογηθεί η επίδραση της εκτέλεσης της μεθόδου σε παράλληλες μονάδες επεξεργασίας, διενεργήθηκε ένα ακόμα πείραμα στο οποίο αυξήθηκε ο αριθμός των παράλληλων μονάδων επεξεργασίας από ένα έως δέκα και τα αποτελέσματα παρουσιάζονται στον Πίνακα 21: «Αποτελέσματα του πειράματος χρησιμοποιώντας την προτεινόμενη μέθοδο. Το σχήμα διάδοσης ορίστηκε σε 1 προς 1 και η τιμή του NP ορίστηκε σε 5. Στα διενεργηθέντα πειράματα, ο αριθμός των παράλληλων μονάδων επεξεργασίας ποικίλλονταν από 1 έως 10».

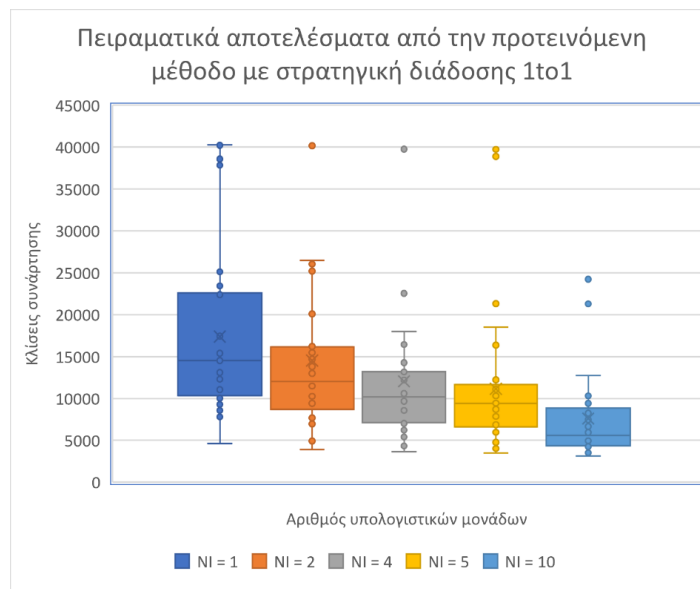
Πίνακας 21: Αποτελέσματα του πειράματος χρησιμοποιώντας την προτεινόμενη μέθοδο. Το σχήμα διάδοσης ορίστηκε σε 1 προς 1 και η τιμή του NP ορίστηκε σε 5. Στα διενεργηθέντα πειράματα, ο αριθμός των παράλληλων μονάδων επεξεργασίας ποικίλλονταν από 1 έως 10.

ΔΟΚΙΜΑΣΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ	N_I = 1	N_I = 2	N_I = 4	N_I = 5	N_I = 10
BF1	12625	11660	9984	10315	6666
BF2	13108	11600	10363	9403	6964
BRANIN	8574	6953	5412	5170	4141
CIGAR10	40274	40180	39763	38887	21291
CM4	11512	12019	12203	12339	9910
DISCUS10	37848	26044	13211	10989	4171
EASOM	4608	3927	3660	3513	3110
ELP10	23436	26469	14268	11100	7462
EXP4	9062	9691	9678	9596	9431
EXP16	22408	15608	18025	21307	21991
EXP64	40238	40177	39856	39731	24234
GKLS250	8070	7809	7225	6853	5591
GKLS350	10696	11488	10578	10095	7279
GRIEWANK2	11064	10681	9127	8926	5604
POTENTIAL3	12876	5568	5018	4756	4333
POTENTIAL5	38377	4905	4455	4221	4016
HANSEN	12467	5067	4340	4031	3518
HARTMAN3	10018	10263	10162	9711	8234
HARTMAN6	15082	9816	7212	7194	5935
RASTRIGIN	9286	9432	6227	5974	4254
ROSENBROCK4	25120	20084	16454	12244	7574
ROSENBROCK8	38577	25195	22531	18508	9587
SHEKEL5	15409	14112	8574	7898	4948
SHEKEL7	14989	13800	9227	8717	5050
SHEKEL10	15085	14662	10268	8229	4871
SINU4	12298	12997	13172	12842	10316
SINU8	15500	15475	16442	16375	12732

TEST2N4	14520	15043	12346	9769	4566
TEST2N5	14801	16097	12358	9440	4813
TEST2N6	17444	16224	9103	7855	4410
TEST2N7	22780	20330	12699	9773	4243
TEST30N3	7814	7967	7010	6382	5014
TEST30N4	8014	7683	6568	6317	5090
ΣΥΝΟΛΟ	573980	479026	397519	368460	251349

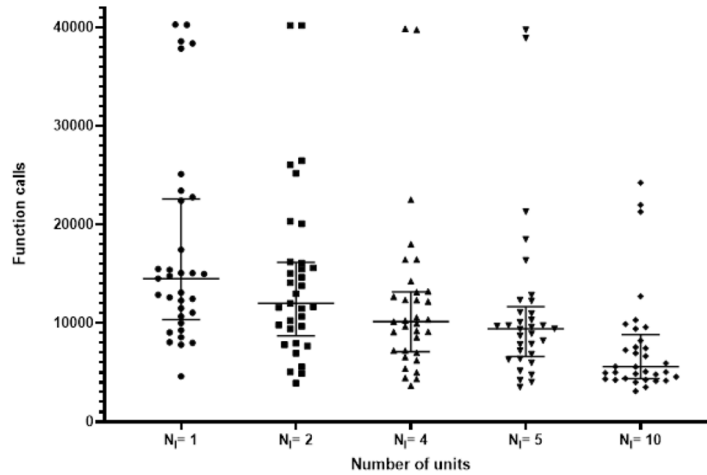
Στη συνέχεια, παρουσιάζεται ένα διάγραμμα box plot, για το πείραμα αυτό στο Διάγραμμα 11: «Διάγραμμα κουτιών για τη σύγκριση μεταξύ διαφορετικού αριθμού μονάδων επεξεργασίας. Η μέθοδος διάδοσης ορίστηκε σε 1 προς 1». Επίσης, διενεργήθηκε ένα τεστ Friedman για να προσδιοριστούν εάν υπήρχαν σημαντικές διαφορές στις βαθμολογίες των κλήσεων συνάρτησης σε πέντε επαναλαμβανόμενους αριθμούς μονάδων ($N_i = 1, 2, 4, 5$ και 10) και τα αποτελέσματα φαίνονται στο Διάγραμμα 12: «Διενεργήθηκε ένα τεστ Friedman για να καθοριστεί εάν υπήρχαν σημαντικές διαφορές στα αποτελέσματα των κλήσεων συνάρτησης σε πέντε επαναλαμβανόμενους αριθμούς μονάδων ($N_i = 1, 2, 4, 5$ και 10). Το Friedman Chisquare ήταν 94.38 με 4 βαθμούς ελευθερίας, υποδεικνύοντας μια σημαντική διαφορά μεταξύ των ομάδων ($p < 0.0001$)».

Διάγραμμα 11: Διάγραμμα κουτιών για τη σύγκριση μεταξύ διαφορετικού αριθμού μονάδων επεξεργασίας. Η μέθοδος διάδοσης ορίστηκε σε 1 προς 1



Διάγραμμα 12: Διενεργήθηκε ένα τεστ Friedman για να καθοριστεί εάν υπήρχαν σημαντικές διαφορές στα αποτελέσματα των κλήσεων συνάρτησης σε πέντε

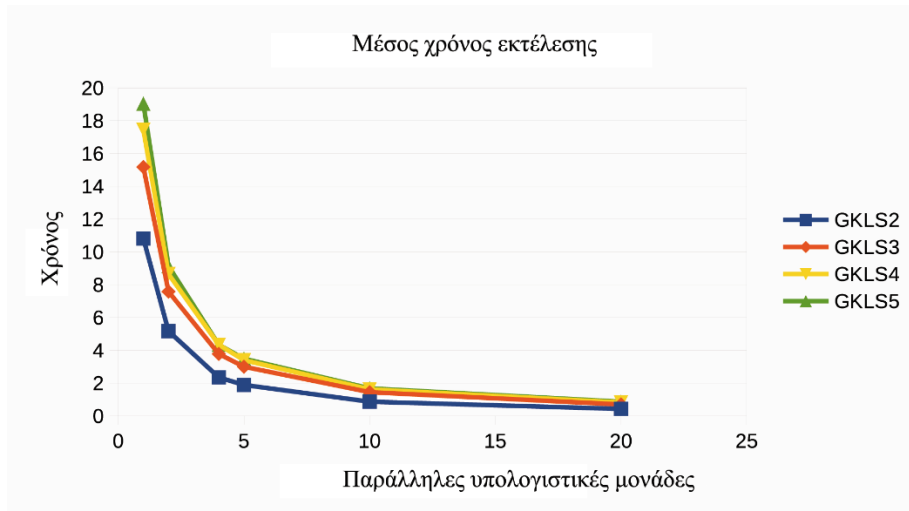
επαναλαμβανόμενους αριθμούς μονάδων ($N_i = 1, 2, 4, 5$ και 10). Το Friedman Chisquare ήταν 94.38 με 4 βαθμούς ελευθερίας, υποδεικνύοντας μια σημαντική διαφορά μεταξύ των ομάδων ($p < 0.0001$).



Από τα πειραματικά αποτελέσματα προκύπτει ότι ο διπλασιασμός των παράλληλων μονάδων επεξεργασίας σχεδόν μειώνει στο ήμισυ τον απαιτούμενο χρόνο εκτέλεσης σε περισσότερες περιπτώσεις.

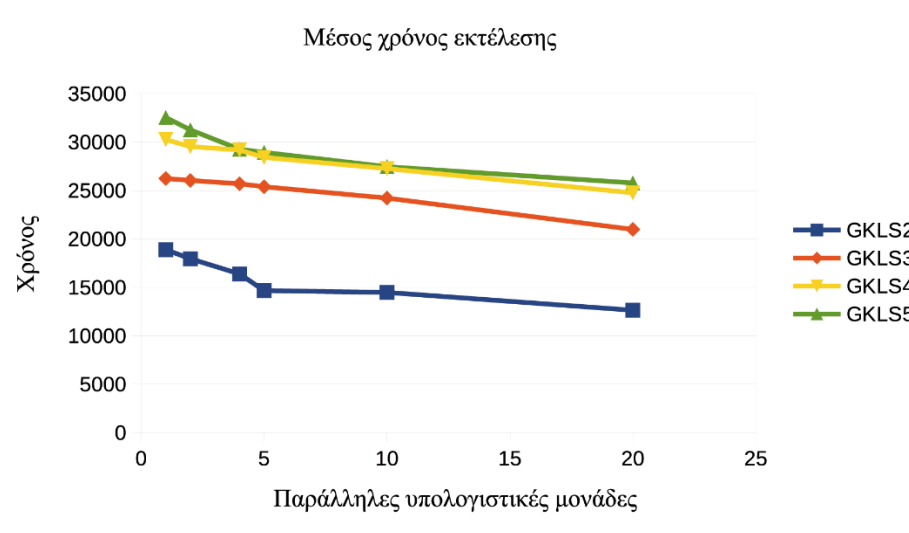
Κατά τη δεύτερη σειρά πειραμάτων χρησιμοποιήθηκαν διαφορετικές εκδόσεις της συνάρτησης GKLS για διαστάσεις από δύο έως πέντε. Σε κάθε διάσταση χρησιμοποιήθηκαν δέκα διαφορετικά παραδείγματα της συνάρτησης για να έχουμε μια ασφαλή εκτίμηση τόσο της απόδοσης της μεθόδου όσο και της πιθανής μείωσης του χρόνου εκτέλεσης καθώς αυξάνονται οι παράλληλες μονάδες επεξεργασίας. Ο μέσος χρόνος εκτέλεσης για διαφορετικές διαστάσεις της συνάρτησης GKLS και για διαφορετικούς αριθμούς μονάδων επεξεργασίας εμφανίζεται γραφικά στο Διάγραμμα 13: «Μέσος χρόνος εκτέλεσης για τη συνάρτηση GKLS. Η διάσταση της συνάρτησης ποικίλλει από δύο έως πέντε». Όπως φαίνεται, ο χρόνος εκτέλεσης μειώνεται γρήγορα με την αύξηση των παράλληλων μονάδων επεξεργασίας σε όλα τα παραδείγματα της συνάρτησης.

Διάγραμμα 13: Μέσος χρόνος εκτέλεσης για τη συνάρτηση GKLS. Η διάσταση της συνάρτησης ποικίλλει από δύο έως πέντε



Επιπλέον, μετρήθηκε ο μέσος αριθμός κλήσεων συνάρτησης για κάθε περίπτωση και τα αποτελέσματα αναπαρίστανται γραφικά στο Διάγραμμα 14: «Μέσος αριθμός κλήσεων συνάρτησης για διάφορες περιπτώσεις της συνάρτησης GKLS». Όπως φαίνεται, ο χρόνος εκτέλεσης μειώνεται γρήγορα με την αύξηση των παράλληλων μονάδων επεξεργασίας σε όλα τα παραδείγματα της συνάρτησης.

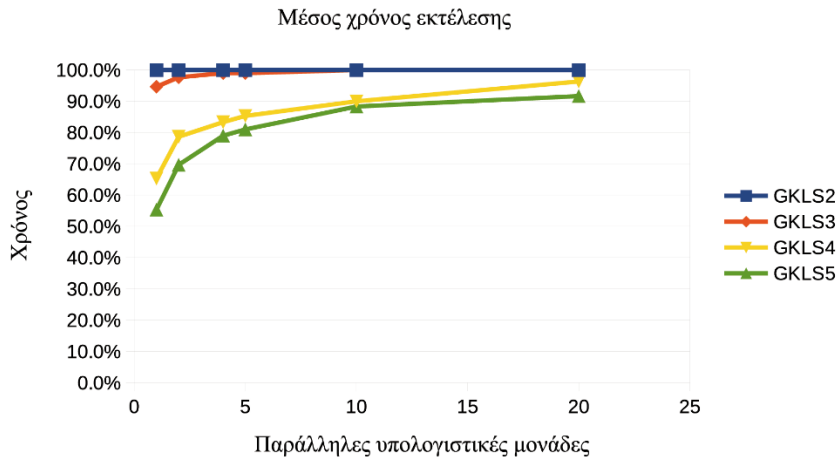
Διάγραμμα 14: Μέσος αριθμός κλήσεων συνάρτησης για διάφορες περιπτώσεις της συνάρτησης GKLS.



Επίσης, διαπιστώθηκε πως η αύξηση του αριθμού των παράλληλων μονάδων επεξεργασίας μειώνει τον αριθμό των απαιτούμενων κλήσεων συνάρτησης, ειδικά για τις χαμηλές διαστάσεις των περιπτώσεων: Gkls2 και Gkls3. Ωστόσο, το πιο ενδιαφέρον σημείο αυτής της σειράς πειραμάτων είναι η ικανότητα της προτεινόμενης τεχνικής να

εντοπίζει τον καθολικό ελάχιστο όσο αυξάνεται ο αριθμός των παράλληλων μονάδων υπολογισμού. Αυτή η ικανότητα απεικονίζεται γραφικά στο Διάγραμμα 15: «Ο μέσος όρος επιτυχίας στην εύρεση του καθολικού ελάχιστου για διαφορετικές περιπτώσεις της συνάρτησης Gkls».

Διάγραμμα 15: Ο μέσος όρος επιτυχίας στην εύρεση του καθολικού ελάχιστου για διαφορετικές περιπτώσεις της συνάρτησης Gkls



ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

[1]: Ανδρεαδάκης, Σ., Κατσαργύρης, Β., Μέτης, Σ., Μπρουχούτας, Κ., Παπασταυρίδης, Σ. & Πολύζος, Γ. (1999). *Μαθηματικά Β' μέρος*. Γ' Γενικού Λυκείου, Ομάδας Προσανατολισμού Θετικών Σπουδών και Σπουδών Οικονομίας & Πληροφορικής. Εκδόσεις: Διόφαντος.

[2]: Βολακάκη, Μ., Κοντοβούρκης, Μ., Κυριακού, Κ., Λοϊζιάς, Σ. & Ματαθίου, Κ. (2019). *Μαθηματικά. Γ' Λυκείου Κοινού Κορμού (Α' τεύχος)*. Εκδόσεις: Υπουργείο Παιδείας και Πολιτισμού, Παιδαγωγικό Ινστιτούτο Κύπρου.

[3]: Μπουμπούλης, Π. (2008). *Πολύωνυμο Taylor. Μέγιστα και Ελάχιστα*. Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής & Τηλεπικοινωνιών. Ανάλυση II. Ανακτήθηκε από: [Πολύωνυμο Taylor. Μέγιστα και Ελάχιστα](#) στις 11/3/2023.

[4] Ζυγκιρίδης, Θ. (2015). *Μαθηματική Ανάλυση II*. Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών. Ανακτήθηκε από: [Μαθηματική Ανάλυση II](#) στις 11/3/2023.

[5]: Αδάμ, Μ., Βραχάτης, Μ., Γαλάνης, Γ., Γράψα, Θ., Δρακόπουλος, Β., Καραμπετάκης, Ν., Κολέτσος, Ι., Κώτσιος, Σ., Μαλιάκας, Μ., Μελάς, Α., Μελετίου, Γ., Μπίσμπας, Α., Μπούνης, Α.,

Παπαγεωργίου, Β., Παπακώστας, Τ., Πάπιστας, Α., Πουλάκης, Δ., Ράγγος, Ο., Ρόθος, Β., Σίμος, Θ., Σπάρταλης, Σ., Φαμέλης, Ι. (2013). *Παράγωγοι*. Ελληνικό Ανοικτό Πανεπιστήμιο. Ανακτήθηκε από: [Παράγωγοι](#) στις 12/3/2023.

[6]: Παπαγεωργίου, Δ.Γ. (2016). *Σημειώσεις για το μάθημα «Υπολογιστικές μέθοδοι πολύπλοκων συστημάτων»*. Πανεπιστήμιο Ιωαννίνων, Τμήμα Μηχανικών Επιστήμης Υλικών. Ανακτήθηκε από: [Υπολογιστικές μέθοδοι πολύπλοκων συστημάτων](#) στις 11/3/2023.

[7]: Αλεξανδρίδης, Α.Θ (2008). *Εφαρμοσμένη βελτιστοποίηση*. Πανεπιστήμιο Πατρών. Πολυτεχνική Σχολή. Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών.

[8]: Αλεξανδρίδης, Α. (2015). *Εφαρμοσμένη βελτιστοποίηση*. Πανεπιστήμιο Πατρών. Πολυτεχνική Σχολή. Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών. Ανοικτά Ακαδημαϊκά μαθήματα.

- [9]: Νικολαΐδης, Χ.(2003). *Σημειώσεις Γραμμικής Άλγεβρας*. Πανεπιστήμιο Θεσσαλίας, Τμήμα Πολιτικών Μηχανικών. Ανακτήθηκε από: [Γραμμική Άλγεβρα](#) στις 13/3/2023.
- [10]: Ρασσιάς, Ι. (2003). *Γραμμική Άλγεβρα. Γραμμικός Προγραμματισμός*. Εκδόσεις Συμμετρία. Αθήνα, 2003.
- [11]: Παμφίλος, Π. (2001). *Γραμμική Άλγεβρα και Εφαρμογές (5^η Έκδοση)*. Πανεπιστημιακές εκδόσεις Κρήτης.
- [12]: Λαγαρής, Ι.Η. (2010). *Εισαγωγή στην Βελτιστοποίηση*. (Πανεπιστημιακές Σημειώσεις). Πανεπιστήμιο Ιωαννίνων, Τμήμα: Πληροφορικής. Διαθέσιμο στο διαδικτυακό τόπο: https://www.cs.uoi.gr/~lagaris/OPT_UNDER/haha.pdf
- [13]: Nicolai N.V (2002). *Fibonacci Numbers*. ISBN: 978-3-0348-8107-4 (eBook). Doi: 10.1007/978-3-0348-8107-4. Springer BaselAG. Διαθέσιμο στο διαδικτυακό τόπο: [Fibonacci Numbers](#) .
- [14] Κατηνιώτης, Ι.Μ. (2012). *Επαναληπτικός Τετραγωνικός Προγραμματισμός*. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Εφαρμοσμένων Μαθηματικών & Φυσικών Επιστημών.[Μεταπτυχιακή Εργασία]. Ανακτήθηκε από: [Επαναληπτικός Τετραγωνικός Προγραμματισμός](#) στις 25/03/2023.
- [15] Λαγαρής, Ι.Η. (2008). *Σημειώσεις διαλέξεων: Βελτιστοποίηση πολυδιάστατων συνεχών συναρτήσεων*. Πανεπιστήμιο Ιωαννίνων, Τμήμα Μηχανικών Η/Υ & Πληροφορικής.
- [16] Μαχαιράς, Β. (2017). *Θεωρία Βελτιστοποίησης*. Πανεπιστήμιο Θεσσαλίας, Σχολή Θετικών Επιστημών, Τμήμα Πληροφορικής. Ανακτήθηκε από : [Θεωρία Βελτιστοποίησης](#) στις 25/3/2023
- [17] Κουνής, Κ. (2019). *Μελέτη Αλγορίθμων Βελτιστοποίησης και Εφαρμογής σε Προβλήματα Σχεδίασης Ηλεκτρονικών Κυκλωμάτων*. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Πληροφορικής, Σχολή Θετικών Επιστημών. Ανακτήθηκε από: [Μελέτη Αλγορίθμων Βελτιστοποίησης και Εφαρμογής σε Προβλήματα Σχεδίασης Ηλεκτρονικών Κυκλωμάτων](#) στις 25/3/2023.
- [18] Singer, Sasa, Singer, Sanja (2004). *Efficient Implementation of the Nelder-Mead Search Algorithm*. Appl. Numer. Anal. Comput. Math. 1, No. 3, pp. 524–534.

- [19] Spendley, W., Hext, G.R., and Himsforth, F.R. (1962). *Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation*. Technometrics, Vol. 4, pp. 441–461.
- [20] Torczon, V. (1991). *On the Convergence of the Multidirectional Search Algorithm*. SIAM J. Optim. 1, pp. 123–145. Doi: <https://doi.org/10.1137/0801010>
- [21]: Αντώνης, Μ. (2013). *Υβριδοποίηση αλγορίθμων βελτιστοποίησης και εφαρμογές στη διαχείριση πόρων*. [Μεταπτυχιακή εργασία]. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Πολυτεχνική Σχολή. Ανακτήθηκε από: [Αλγόριθμοι βελτιστοποίησης](#) στις 26/03/2023.
- [22]: Welcome From Nothing. *Learn maths to solve real problems*. (χ.χ). Ανακτήθηκε από: [Learn maths to solve real problems](#) στις 26/3/2023.
- [23]: Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1992). *Van Wijngaarden-Dekker-Brent Method*. Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed. Cambridge, England: Cambridge University Press, pp. 352-355, 1992. Ανακτήθηκε από: [The Art of Scientific Computing](#)
- [24]: Kiefer, J. (1953). *Sequential minimax search for a maximum*. Proceedings of the American Mathematical Society, 4 (3): 502–506, doi:10.2307/2032161, JSTOR 2032161, MR 0055639.
- [25]: *Gradient Descent in Machine Learning* (2021). Java T point. Ανακτήθηκε από: [Gradient Descent in Machine Learning](#) στις 18/4/2023.
- [26]: *What is gradient descent?* (χ.χ). IBM. Ανακτήθηκε από: [Gradient-descent](#) στις 26/03/2023.
- [27]: Fletcher, Roger (1987). *Practical Methods of Optimization* (2nd ed.). New York: John Wiley & Sons, ISBN 978-0-471-91547-8
- [28] Kirkham, L. (2022). *Newton-Raphson*. Edexcel A Level Maths:Pure . Revision Notes. Ανακτήθηκε από: [Newton-raphson](#) στις 20/04/2023.
- [29]: Kingma, Diederik P., and Jimmy Lei Ba. 2015. *Adam: A Method for Stochastic Optimization*. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings: 1–15.
- [30]: Τζιώνας, Ε. (2021). *Ενισχυτική Μάθηση για Οικονομικές Χρονοσειρές* [Διπλωματική Εργασία]. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Τμήμα Ηλεκτρολόγων Μηχανικών

και Μηχανικών Υπολογιστών. Ανακτήθηκε από: [Ενισχυτική Μάθηση για Οικονομικές Χρονοσειρές](#) στις 26/03/2023.

[31]: Vitaly, B. (2018). *Understanding RMSprop — faster neural network learning*. Ανακτήθηκε από: [RMSprop](#) στις 27/03/2023.

[32]: RMSprop (2023). Deepchecks. Ανακτήθηκε από: [rmsprop](#) στις 22/01/2023.

[33]: Ευστρατιάδης, Α. & Μακρόπουλος, Χ. (χ.χ). Τοπικές και ολικές τεχνικές βελτιστοποίησης. Σημειώσεις στα πλαίσια του μαθήματος: Βελτιστοποίηση συστημάτων υδατικών πόρων -Υδροπληροφορική. Εθνικό Μετσόβιο Πολυτεχνείο. Ανακτήθηκε από: [Τοπικές και ολικές τεχνικές βελτιστοποίησης](#) στις 22/04/2023.

[34] Κυριακού, Δ. (2022). *Επαναδιατύπωση Μεθόδου Βελτιστοποίησης Μη-Δομημένων Πλεγμάτων με Κριτήριο τη Μετρική Σφαιρικότητας*, [Διπλωματική εργασία]. Σχολή Μηχανολόγων Μηχανικών, Εθνικό Μετσόβιο Πολυτεχνείο. Ανακτήθηκε από: [Επαναδιατύπωση Μεθόδου Βελτιστοποίησης Μη-Δομημένων Πλεγμάτων με Κριτήριο τη Μετρική Σφαιρικότητας](#) στις 23/04/2023

[35] Mathematics Source Library C & ASM (2004). *Fletcher-Powell-Davidon*. Ανακτήθηκε από: [Mathematics Source Library](#) στις 23/04/2023

[36] Σπανού, Π.Α. (2007). Βέλτιστα πλάνα φωτεινής σηματοδότησης σταθερού χρόνου. [Διπλωματική εργασία]. Τμήμα Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης.

[37]: Τσιλιμαντός, Δ.Χ.(2009). *Στοχαστικές Πολυδιάστατες Τεχνικές Βελτιστοποίησης για την Σχεδίαση Δικτύων Ασύρματων Επικοινωνιών*. [Διδακτορική Διατριβή]. Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών Υπολογιστών. Ανακτήθηκε από: [Στοχαστικές Πολυδιάστατες Τεχνικές Βελτιστοποίησης για την Σχεδίαση Δικτύων Ασύρματων Επικοινωνιών](#) στις 31/03/2023.

[38]: Laarhoven, P.J.M, Aarts, E.H.L (1987). *Simulated Annealing: Theory and Applications*. Mathematics and Its Applications (MAIA, volume 37). Springer. Ανακτήθηκε από: [Simulated Annealing: Theory and Applications](#) , eBook ISBN: 978-94-015-7744-1 στις 31/03/2023.

- [39]: Tam, A. (2021). *A Gentle Introduction to Particle Swarm Optimization*. Machine Learning Mastery. Ανακτήθηκε από: [A Gentle Introduction to Particle Swarm Optimization](#) στις 31/03/2023.
- [40]: Charilogis, V., Tsoulos, I.G., Tzallas, A. & Karvounis, E. (2022). *Modifications for the Differential Evolution Algorithm*. *Symmetry* 2022, 14, 447. <https://doi.org/10.3390/sym14030447>.
- [41]: Tsoulos, I.G., Tzallas, A. & Anastasopoulos, N. (2021). *An Improved Controlled Random Search Method*. *Symmetry* 2021, 13, 1981. <https://doi.org/10.3390/sym13111981>
- [42]: Charilogis, V., Tsoulos, I.G. (2022a). *MinCentre: using clustering in global optimization*. *Int. J. Computational Intelligence Studies*, Vol. 11, No. 1, 2022.
- [43]: Charilogis, V., Tsoulos, I.G. (2022b). *Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions*. *Information* 2022, 13, 217. <https://doi.org/10.3390/info13050217>
- [44]: S.H. Brooks: *A Discussion of random methods for locating surface maxima*. *Operations Research* 6 (1958), pp. 244-251. DOI: <https://doi.org/10.1287/opre.6.2.244>
- [45]: Zabinsky, Z.B. (2003α). *Stochastic Adaptive Search for Global Optimization*. University of Washington Seattle, Washington, USA, Chapter 1, p.11. Kluwer Academic Publishers. <http://courses.washington.edu/inde510/516/ZabinskyBookCh1.pdf>
- [46]: Auger, A., Brockhoff, D. (2018). *Exercise: Pure random search and (1+1)-EA on $\Omega = \{0,1\}^n$* . Advanced Optimization class. Université Paris-Saclay. Ανακτήθηκε από: <http://www.cmap.polytechnique.fr/~dimo.brockhoff/advancedOptSaclay/2018/exercises/01-exercise-PRSand1p1EA.pdf> στις 30/04/2023.
- [47]: Zabinsky, Z.B. (2003β). *Pure Random Search and Pure Adaptive Search*. In: *Stochastic Adaptive Search for Global Optimization. Nonconvex Optimization and Its Applications*, vol 72. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-9182-9_2
- [48]: Peng, J.P., Shi, D.H. (2000). *Improvement of Pure Random Search in Global Optimization*. *Journal of Shanghai University (English Edition)* ISSN 1007-6417, Vol.4, No.2(Jun.2000), pp 92—95.

- [49] Romeijn, H.E. (2008). *Random Search Methods* . In: Floudas, C., Pardalos, P. (eds) Encyclopedia of Optimization. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-74759-0_556
- [50] Borowski, N. (1971). *A comparison of three random search methods*. [Master of applied science]. Department of Electrical Engineering. The university of British Columbia.
- [51] Zabinsky , Z.B. (2009). *Random Search Algorithms*. University of Washington. Ανακτήθηκε από: [Random Search Algorithms](#)
- [52] Pfeiffer, A. (2019). *Comparing Grid and Randomized Search Methods in Python*. Published in Better Programming. Ανακτήθηκε από: [Comparing Grid and Randomized Search Methods in Python](#) .
- [53]: Types of Sampling – Numeracy, Maths and Statistics- Academic Skills Kit (χ.χ). *Random Sampling*. Ανακτήθηκε από: [Random Sampling](#)
- [54]: Martí, R., Aceves, R., León, M.T., Moreno-Vega, J.M., Duarte, A. (2019). *Intelligent Multi-Start Methods*. In: Gendreau, M., Potvin, JY. (eds) Handbook of Metaheuristics, Chapter 7, pp: 221-243. International Series in Operations Research & Management Science, vol 272. Springer, Cham. https://doi.org/10.1007/978-3-319-91086-4_7
- [55]: Θεός, Φ. (2001). *Μέθοδοι ολικής ελαχιστοποίησης* [Μεταπτυχιακή εργασία]. Πανεπιστήμιο Ιωαννίνων, Τμήμα Πληροφορικής. Ανακτήθηκε από : [Μέθοδοι ολικής ελαχιστοποίησης](#)
- [56]: Γούλα, Ε. (2020). *Ολική Βελτιστοποίηση: Μέθοδος της Διαφορικής Εξέλιξης και Παραλλαγές*. [Μεταπτυχιακή εργασία]. ΕΑΠ, Σχολή Θετικών Επιστημών και Τεχνολογίας. Ανακτήθηκε από: [Ολική Βελτιστοποίηση: Μέθοδος της Διαφορικής Εξέλιξης και Παραλλαγές](#)
- [57]: Τσούλος, Ι. (2005). *Καθολική Βελτιστοποίηση: Μέθοδοι, Λογισμικό και Εφαρμογές*. [Διδακτορική Διατριβή]. Πανεπιστήμιο Ιωαννίνων, Τμήμα: Πληροφορικής.
- [58]: Καλαπόδη, Α. (2009). *Εισαγωγή στην στατιστική των επιχειρήσεων. Συνεχείς Κατανομές Πιθανοτήτων*. ΤΕΙ Δυτικής Ελλάδας. Ανακτήθηκε από : [Εισαγωγή στην στατιστική των επιχειρήσεων. Συνεχείς Κατανομές Πιθανοτήτων](#).

- [59]: Sepulveda, A. & Epstein, L.D. (1996). The repulsion algorithm, a new multistart method for global optimization 11(3):145-152. DOI:10.1007/BF01197028.
- [60]: Zielinski, R. (1981). *A statistical estimate of the structure of multiextremal problems*. Mathematical Programming 21(1981), 348-356.
- [61]: Betro, B. & Schoen, F. (1992). *Optimal and sub-optimal stopping rules for the Multistart algorithm in global optimization*. Mathematical Programming 57(1992), 445-458.
- [62]: Kan, R. Boender, C.G.E. (2003). Bayesian stopping rules for Multistart global optimization methods. Mathematical Programming 37(1987), 59-80.
- [63]: Hart, W.E. (1998). Sequential stopping rules for random optimization methods with application to Multistart local search. Siam J. Optim 9 (1998), 270-290.
- [64]: Lagaris, I.E., Tsoulos, I.G. (2008). *Stopping rules for box-constrained stochastic global optimization*. Applied Mathematics and Computation. Volume 197, Issue 2, 1 April 2008, Pages 622-632. Doi: <https://doi.org/10.1016/j.amc.2007.08.001>
- [65]: Powell, M. J. D. "A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation." In *Advances in Optimization and Numerical Analysis*, eds. Susana Gomez and Jean-Pierre Hennart, 51-67. Kluwer Academic Publishers, 1994.
- [66]: Conn, Andrew R., Nicholas IM Gould, and Ph.L Toint. "On the use of more general algorithms for calculating trust regions steps." Numerical analysis report. Manchester: University of Manchester, 1996.
- [67]: Charilogis, V., Tsoulos, I.G. (2022c). *Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions*. Information 2022, 13, 217. <https://doi.org/10.3390/info13050217>
- [68]: Charilogis, V., Tsoulos, I. (2022d). *MinCentre: using clustering in global optimization*. Int. J. Computational Intelligence Studies, Vol. 11, No. 1, 2022.
- [69]: Tsoulos, I. (2008). *Modifications of real code genetic algorithm for global optimization*. Appl. Math. Comput. 2008, 203, 598–607. DOI:10.1016/j.amc.2008.05.005

- [70]: Charilogis, V., Tsoulos, I., Tzallas, A., Anastasopoulos, N. (2021). An Improved Controlled Random Search Method. *Symmetry* 2021, 13, 1981. <https://doi.org/10.3390/sym13111981>
- [71]: Kaelo, P., Ali, M. (2006). *Some variants of the controlled random search algorithm for global optimization*. ResearchGate, *Journal of Optimization Theory and Applications*, 130(2):253-264. DOI: 10.1007/s10957-006-9101-0
- [72]: Tsoulos, I.G., Lagaris, I.E. (2005). Genetically controlled random search: a global optimization method for continuous multidimensional functions. Elsevier: *Computer Physics Communications* 174 (2006) 152–159.
- [73]: Tsoulos, I.G. (2008). *Modifications of real code genetic algorithm for global optimization*. *Appl. Math. Comput.* 2008, 203, 598–607.
- [74]: Storn, R.M. & Price, K. (1995). *Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*. *Journal of Global Optimization* 23(1). [75]: Storn, R.M. & Price, K. (1997). *Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*. *Journal of Global Optimization*, Volume 11, pages341–359.
- [76]: Price, K., Storn, R.M. & Lampinen, J.A. (2006). *Differential Evolution: A Practical Approach to Global Optimization*. Springer.
- [77]: Storn, R.M. & Price, K. (1996). *Minimizing the real functions of the ICEC'96 contest by differential evolution*. Publisher: IEEE. DOI: 10.1109/ICEC.1996.542711.
- [78]: Lampinen, J.A., Price, K. & Storn, R.M. (2005). *New Optimization Techniques in Engineering*. Chapter 6: Differential Evolution, p.:123. Springer.
- [79]: Das, S. & Suganthan, P.N. (2010). *Differential Evolution: A Survey of the State-of-the-Art*. Publisher: IEEE. DOI: 10.1109/TEVC.2010.2059031.
- [80]: Eberhart, R. & Shi, Y. (2000). *Comparison between Genetic Algorithms and Particle Swarm Optimization*. *Evolutionary Programming VII*. Springer.
- [81]: Mallipeddi, R., Suganthan, P.N. & Pan, Q. (2011). *Differential Evolution Algorithm with Ensemble of Parameters and Mutation Strategies*. *Information Sciences*, DOI:10.1007/978-3-642-17563-3_9.

- [82]: Noghabi, H.S., Mashhadi,H.R. & Shojaei,K. (2015). *Differential Evolution with Generalized Mutation Operator for Parameters Optimization in Gene Selection for Cancer Classification*. Computer Science, Neural and Evolutionary Computing. Doi: <https://doi.org/10.48550/arXiv.1510.02516>
- [83]: Meenachi, M. & Ramakrishnan (2020). *Differential evolution and ACO based global optimal feature selection with fuzzy rough set for cancer data classification*. Soft Computing volume 24, pages18463–18475, Springer.
- [84]: Othman, M.S., Kumaran,S.R. & Yusuf,L.M. (2020). *Gene selection using hybrid multi-objective cuckoo search algorithm with evolutionary operators for cancer microarray data*. IEEE Access 8:186348-186361 , DOI:[10.1109/ACCESS.2020.3029890](https://doi.org/10.1109/ACCESS.2020.3029890)
- [85]:Krink,T. & Paterlini,S. (2008). *Differential Evolution for Multiobjective Portfolio Optimization*. ResearchGate.
- [86]: Fouad, A. (2014). *A novel hybrid genetic differential evolution algorithm for constrained optimization problems*. Academia.
- [87]: Ali, M.M & Törn, A. (2004). Population set-based global optimization algorithms: some modifications and numerical studies, Computers & Operations. Research 31, pp. 1703-1725, 2004.
- [88]: Price, K., & Storn, R. (2000, July). Differential Evolution (DE): for Continuous Function Optimization (an algorithm by Kenneth Price and Rainer Storn). Ανάκτηση από <http://www.ICSI.Berkeley.edu/~storn/code> στις 24/2/2024.
- [89]: Engelbrecht, A. P. (2007). Computational Intelligence: An Introduction (Second Edition εκδ.). Wiley Online Library. doi: <https://doi.org/10.1002/9780470512517.ch13>
- [90]: Onwubolu, G. C., & Babu, V. B. (2004). *New Optimization Techniques in Engineering*. Berlin: Springer.
- [91]: Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks (ICNN), Vol. 4, pp. 1942-1948.
- [92]: Clerc, M. (2006). Particle Swarm Optimization. ISTE Ltd. ISBN 10: 1-905209-04-5.
- [93]: Kennedy, J., Eberhart, R.C. & Shi, Y. (2001). *Swarm Intelligence*. ISBN: 978-1-55860-595-4. Doi: <https://doi.org/10.1016/B978-1-55860-595-4.X5000-1>.

- [94]: Meneses, A.A.D.; Dornellas, M.; Schirru, M.R. (2009). Particle Swarm Optimization applied to the nuclear reload problem of a Pressurized Water Reactor. *Progress Nucl. Energy* 2009, 51, 319–326.
- [95]: Shaw, R.; Srivastava, S. (2007). Particle swarm optimization: A new tool to invert geophysical data. *Geophysics* 2007, 72, F75–F83.
- [96]: Ourique, C.O.; Biscaia, E.C.; Pinto, J.C. (2002). The use of particle swarm optimization for dynamical analysis in chemical processes. *Comput. Chem. Eng.* 2002, 26, 1783–1793.
- [97]: Fang, H.; Zhou, J.; Wang, Z. (2022). Hybrid method integrating machine learning and particle swarm optimization for smart chemical process operations. *Front. Chem. Sci. Eng.* 2022, 16, 274–287.
- [98]: Wachowiak, M.P.; Smolikova, R.; Zheng, Y.J.M. Zurada, A.S.E. (2004). An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Trans. Evol. Comput.* 2004, 8, 289–301.
- [99]: Marinakis, Y. Marinaki, M.; Dounias, G. (2008). Particle swarm optimization for pap-smear diagnosis. *Expert Syst. Appl.* 2008, 35, 1645–1656.
- [100]: Park, J.; Jeong, Y.; Shin, J.; Lee, K. An Improved Particle Swarm Optimization for Nonconvex Economic Dispatch Problems. *IEEE Trans. Power Syst.* 2010, 25, 156–166.
- [101]: Poongavanam, R., Thirumal, V. & Thirugnanasambandam, K. (2019). A Comprehensive Survey On Particle Swarm Optimization (Pso) And Its Variants To Classical Benchmark Functions. *JETIR* June 2019, Vol. 6, Issue 6. ISSN-2349-5162.
- [102]: Tanweer, M.R., Auditya, R., Suresh, S., Sundararajan, N., Srikanth, N. (2016). *Directionally Driven Self-Regulating Particle Swarm Optimization algorithm*. *Swarm and Evolutionary Computation*, 2016; 28:98-116.
- [103]: Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, Orlando, FL, USA, 12–15 October 1997; Volume 4105, pp. 4104–4108.

- [104]: Jiang, B., Liu, A., Chen, W., Xing, M., Zhou, W. & Zheng, W. (2011). *The optimization of large gap–midgap ratio photonic crystal with improved Bisection-Particle Swarm Optimization*. Journal of Optics Communication, 2011; 284:226-230.
- [105]: Xingjuan, C., Zhihua, C., Jianchao, Z., Ying, T. (2008). *Dispersed Particle Swarm Optimization*. In Proceedings of Information Processing Letters, 2008; 105: 231-235.
- [106]: Neri, F., Mininno, E., Lacca, G. (2013). *Compact Particle Swarm Optimization*. Journal of Information Sciences, 2013; pp. 96-121.
- [107]: Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 1 December 1995; Volume 4, pp. 1942–1948.
- [108]: Αδαμίδης, Π. (2020). Νήματα και Συγχρονισμός. Τμήμα Μηχανικών Πληροφορικής, ΤΕΙ Θεσσαλονίκης. Διαθέσιμο στο διαδικτυακό τόπο: https://people.iee.ihu.gr/~adamidis/OOP/Threads_notes.pdf (Ανακτήθηκε στις 10/3/2024).
- [109] Δασυγενής, Μ. (χ.χ). Συστήματα Παράλληλης & Κατανεμημένης Επεξεργασίας. Ενότητα 8: Νήματα και Παραλληλισμός Διεργασιών. Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών. (Ανακτήθηκε στις 28/5/2024).
- [110]: Πρατικάκης, Π. (2024). Pthreads. Παράλληλος Προγραμματισμός. Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης. Διαθέσιμο στο διαδικτυακό τόπο: <https://www.csd.uoc.gr/~hy342/schedule.html>
- [111]: Introduction to MPI (2024). CARLETON UNIVERSITY. Διαθέσιμο στο διαδικτυακό τόπο: <https://carleton.ca/rcs/rcdc/introduction-to-mpi/> . (Ανακτήθηκε στις 2/6/2024).
- [112]: Gillis, A. S. & Bigelow, S.J. (2024). *Message passing interface (MPI)*. Διαθέσιμο στο διαδικτυακό τόπο: https://www.techtarget.com/searchenterprisedesktop/definition/message-passing-interface-MPI_ (Ανακτήθηκε στις 2/6/2024).
- [113]: Open MPI: Open Source High Performance Computing (2024). *A High Performance Message Passing Library*. Διαθέσιμο στο διαδικτυακό τόπο: <https://www.open-mpi.org> (Ανακτήθηκε στις 10/3/2024)

- [114]: Gu, R., Fan, S., Hu, Q., Yuan, C. & Huang, Y. (2018). Parallelizing Machine Learning Optimization Algorithms on Distributed Data-Parallel Platforms with Parameter Server. IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). DOI: 10.1109/PADSW.2018.8644533.
- [115]: Low, Y., Gonzalez, J., Kyrola, A., Bickson, D., Guestrin, C., Hellerstein, J. (2010). GraphLab: A New Framework For Parallel Machine Learning. Source: arXiv, Computer Science.
- [116]: Yangyang, L., Liu, G., Lu, G., Jiao, L., Marturi, N. & Shang, R. (2019). Hyper-Parameter Optimization Using MARS Surrogate for Machine-Learning Algorithms. IEEE Transactions on Emerging Topics in Computational Intelligence, pp(99):1-11. DOI: 10.1109/TETCI.2019.2918509.
- [117]: Yamashiro, H. & Nonaka, H. (2021). Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem. ScienceDirect: Operations Research Perspectives, Vol.:8, 2021, Doi: <https://doi.org/10.1016/j.orp.2021.100196> .
- [118]: Kim, H.S. & Tsai, L. (2003). Design Optimization of a Cartesian Parallel Manipulator. Journal of Mechanical Design, 125(1):43-51. Doi: <https://doi.org/10.1115/1.1543977> .
- [119]: Oh, S., Jong Jang, H. & Pedrycz, W. (2009). The design of a fuzzy cascade controller for ball and beam system: A study in optimization with the use of parallel genetic algorithms. ScienceDirect: Engineering Applications of Artificial Intelligence, 22(2):261-271. Doi: <https://doi.org/10.1016/j.engappai.2008.07.003> .
- [120]: Fatehi, M., Toloei, A., Zio, E., Niaki, S.T.A. & Kesh, B. (2023). Robust optimization of the design of monopropellant propulsion control systems using an advanced teaching-learning-based optimization method. ScienceDirect: Engineering Applications of Artificial Intelligence, Vol.:126, Part: A.
- [121]: Cai, J., Yang, H., Lai, T. & Xu, K. (2023). Parallel pump and chiller system optimization method for minimizing energy consumption based on a novel multi-objective gorilla troops optimizer. ScienceDirect: Journal Of Building Engineering, Vol.: 76, Doi: <https://doi.org/10.1016/j.jobee.2023.107366>

- [122]: Yu, Y. & Shahabi, L. (2023). Engineering Application of Artificial Intelligence: Optimal infrastructure in microgrids with diverse uncertainties based on demand response, renewable energy sources and two-stage parallel optimization algorithm. ScienceDirect. Vol.:123, Part b. Doi: <https://doi.org/10.1016/j.engappai.2023.106233>
- [123]: Ramirez-Gil, F.J., Pere-Madrid, C.M., Nelli Silva, E.C. & Montealerge-Rubio, W. (2021). Sustainable Computing: Informatics and Systems: Parallel computing for the topology optimization method: Performance metrics and energy consumption analysis in multiphysics problems. Vol.:30, Doi: <https://doi.org/10.1016/j.suscom.2020.100481>
- [124]: Tavakolan, M., Mostafazadeh, F., Eirdmoussa, S.J., Safari, A. & Mirzai, K. (2022). A parallel computing simulation-based multi-objective optimization framework for economic analysis of building energy retrofit: A case study in Iran. ScienceDirect: Journal of Building Engineering, Vol.: 45, Doi: <https://doi.org/10.1016/j.jobee.2021.103485>
- [125]: Lin, G. (2020). Parallel optimization n based operational planning to enhance the resilience of large-scale power systems. Mississippi State University, Scholars Junction. Online: <https://scholarsjunction.msstate.edu/cgi/viewcontent.cgi?article=4435&context=td>
- [126]: Pang, M. & Shoemaker, C.A. (2023). Comparison of parallel optimization algorithms on computationally expensive groundwater remediation designs. Science of the Total Environment: 857(3), Doi: <https://doi.org/10.1016/j.scitotenv.2022.159544>
- [127]: Ezugwu, A. (2023). A general Framework for Utilizing Metaheuristic Optimization for Sustainable Unrelated Parallel Machine Scheduling: A concise overview. Arxiv, Computer Science, Neural and Evolutionary Computing. Doi: <https://doi.org/10.48550/arXiv.2311.12802>
- [128]: Censor, Y. & Zenios, S. (1997). Parallel Optimization: Theory, Algorithms and Applications. Publisher: Oxford University Press, USA ISBN: ISBN-13: 978-0195100624. DOI: 10.1093/oso/9780195100624.001.0001
- [129]: Storn, R. On the usage of differential evolution for function optimization. In Proceedings of the North American Fuzzy Information Processing, Berkeley, CA, USA, 19–22 June 1996; pp. 519–523.
- [130]: F. Neri, E. Mininno, Memetic Compact Differential Evolution for Cartesian Robot Control. IEEE Comput. Intell. 2010, 5, 54–65.

- [131]: Mininno, E.; Neri, F.; Cupertino, F.; Naso, D. Compact Differential Evolution. *IEEE Trans. Evol.* 2011, 15, 32–54.
- [132]: Qin, A.K.; Huang, V.L.; Suganthan, P.N. Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization. *IEEE Trans. Evol. Comput.* 2009, 13, 398–417.
- [133]: Charilogis, V.; Tsoulos, I.G. (2023). A Parallel Implementation of the Differential evolution Method. *Analytics* **2023**, 2, 17–30. <https://doi.org/10.3390/analytics2010002>
- [134]: Powell, M.J.D. A Tolerant Algorithm for Linearly Constrained Optimization Calculations. *Math. Program.*
- [135]: Charilogis, V., Tsoulos, G.I., Tzallas, A. (2023). An Improved Parallel Particle Swarm Optimization. *SN Computer Science* (2023) 4:766. Doi: <https://doi.org/10.1007/s42979-023-02227-9>
- [136]: Chandra R, Dagum L, Kohr D, Maydan D, McDonald J, Menon R. (2001). *Parallel programming in open MP*. Morgan Kaufmann Publishers Inc.; 2001.

ΠΑΡΑΡΤΗΜΑ Α' – ΠΕΙΡΑΜΑΤΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

Οι περιγραφές των συναρτήσεων δοκιμής που χρησιμοποιήθηκαν στα πειράματα είναι οι εξής:

- **Bf1 (Bohachevsky 1) συνάρτηση.**

Η συνάρτηση ορίζεται ως εξής:

$$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) - \frac{4}{10} \cos(4\pi x_2) + \frac{7}{10},$$

όπου $x \in [-100,100]^2$. Η τιμή του καθολικού ελάχιστου είναι 0,0.

- **Bf2 (Bohachevsky 2) συνάρτηση.**

Η συνάρτηση ορίζεται ως εξής:

$$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) \cos(4\pi x_2) + \frac{3}{10},$$

όπου $x \in [-50,50]^2$. Η τιμή του καθολικού ελάχιστου είναι 0,0.

- **Branin Συνάρτηση.**

Η συνάρτηση ορίζεται από:

$$f(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10$$

με $-5 \leq x_1 \leq 10$, $0 \leq x_2 \leq 15$.

Το καθολικό ελάχιστο είναι 0.397887 όπου $x \in [-10,10]^2$. Η ελάχιστη τιμή της καθολικής είναι -0.397386.

- **CM συνάρτηση.**

Η Cosine Mixture συνάρτηση δίνεται από την εξίσωση:

$$f(x) = \sum_{i=1}^n x_i^2 - \frac{1}{10} \sum_{i=1}^n \cos(5\pi x_i)$$

όπου $x \in [-1,1]^n$. Για τα πειράματα χρησιμοποιήθηκε για $n=4$.

- **Camel συνάρτηση.**

Η συνάρτηση δίνεται από την εξίσωση:

$$f(x) = (4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4), \quad \text{όπου } x \in [-5,5]^2$$

- **Easom συνάρτηση.**

Η συνάρτηση δίνεται από την εξίσωση:

$$f(x) = -\cos(x_1) \cos(x_2) \exp((x_2 - \pi)^2 - (x_1 - \pi)^2), \quad \text{όπου } x \in [-100,100]^2.$$

- **Exponential συνάρτηση.**

Η συνάρτηση ορίζεται ως εξής:

$$f(x) = -\exp\left(-0.5 \sum_{i=1}^n x_i^2\right)$$

για $-1 \leq x_i \leq 1$.

Το καθολικό ελάχιστο εντοπίζεται στο $x^* = (0,0,\dots,0)$ με τιμή -1 . Στα πειράματα χρησιμοποιήθηκε η συνάρτηση με $n = 2,4,8,16,32$.

- **Goldstein and Price συνάρτηση.**

Η συνάρτηση δίνεται από την εξίσωση:

$$f(x) = [1 + (x_1 + x_2 + 1)^2$$

$$(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$$

$$[30 + (2x_1 - 3x_2)^2$$

$$(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

όπου $x \in [-2,2]^2$. Το καθολικό ελάχιστο εντοπίζεται στο $x^* = (0,-1)$ με τιμή 3.0 .

- **Griewank2 συνάρτηση.**

Η συνάρτηση δίνεται από:

$$f(x) = [1 + \frac{1}{200} \sum_{i=1}^2 x_i^2 - \prod_{i=1}^2 \frac{\cos(x_i)}{\sqrt{|i|}}], \quad x \in [-100,100]^2.$$

Το καθολικό ελάχιστο εντοπίζεται στο $x^* = (0,0,\dots,0)$ με τιμή 0 .

- **Gkls συνάρτηση.**

$$f(x) = Gkls(x, n, w),$$

είναι μια συνάρτηση με w τοπικά ελάχιστα, περιγράφεται με $x \in [-1,1]^n$ και το n είναι θετικός ακέραιος αριθμός ανάμεσα από το 2 και το 100. Το καθολικό ελάχιστο είναι -1 και στα πειράματα χρησιμοποιήθηκε για $n=2, 3$ και $w=50, 100$.

- **Hansen συνάρτηση.**

$$f(x) = \sum_{i=1}^5 i \cos[(i-1)x_1 + i] \sum_{j=1}^5 j \cos[(j+1)x_2 + j], \quad x \in [-10,10]^2.$$

- **Hartman 3 Συνάρτηση.**

Η συνάρτηση δίνεται από:

$$f(x) = \left(\sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right) \right)$$

με $x \in [0,1]^3$ και

$$a = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix} \quad \text{και} \quad p = \begin{pmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{pmatrix}$$

- **Hartman 6 συνάρτηση.**

$$f(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right)$$

με $x \in [0,1]^6$ και

$$a = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}, \quad c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix} \quad \text{και}$$

$$P = \begin{pmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{pmatrix}$$

- **Potential συνάρτηση.**

Η μοριακή διαμόρφωση που αντιστοιχεί στο καθολικό ελάχιστο της ενέργειας των ατόμων N που αλληλοεπιδρούν μέσω του δυναμικού Lennard-Jones χρησιμοποιείται ως α δοκιμαστική περίπτωση εδώ. Η συνάρτηση που πρέπει να ελαχιστοποιηθεί δίνεται από:

$$V_{LJ}(r) = 4 \epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]$$

Στα τρέχοντα πειράματα μελετήθηκαν τρεις διαφορετικές περιπτώσεις: N=3, 4, 5, 6, 10.

- **Rastrigin συνάρτηση.**

Η συνάρτηση δίνεται από:

$$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \quad x \in [-1,1]^2$$

Το καθολικό ελάχιστο βρίσκεται στο $x^* = (0,0)$ με τιμή -2.0.

- **Rosenbrock συνάρτηση.**

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

και $-30 \leq x_i \leq 30$.

Το καθολικό ελάχιστο βρίσκεται στο $x^*=(0,0,\dots,0)$ με $f(x^*)=0$. Στα πειράματα χρησιμοποιήθηκε το $n=4, 8, 16$.

- **Shekel 7 συνάρτηση.**

$$f(x) = - \sum_{i=1}^7 \frac{1}{(x - a_i)(x - a_i)^T + c_i}$$

όπου $x \in [0, 10]^4$ και

$$\alpha = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 3 & 5 & 3 \end{pmatrix}, \quad c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \end{pmatrix}.$$

Το καθολικό ελάχιστο είναι: -10.342378.

- **Shekel 5 συνάρτηση.**

$$f(x) = - \sum_{i=1}^5 \frac{1}{(x - a_i)(x - a_i)^T + c_i}$$

όπου $x \in [0, 10]^4$ και

$$\alpha = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \end{pmatrix}, \quad c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \end{pmatrix}.$$

Η ελάχιστη τιμή της καθολικής είναι -10.107749.

- **Shekel 10 συνάρτηση.**

$$f(x) = - \sum_{i=1}^{10} \frac{1}{(x - a_i)(x - a_i)^T + c_i}$$

όπου $x \in [0, 10]^4$ και

$$a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{pmatrix}, \quad c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.6 \end{pmatrix}.$$

Το καθολικό ελάχιστο είναι -10.536410.

- **Sinusoidal συνάρτηση.**

Η συνάρτηση δίνεται από:

$$f(x) = \left(2.5 \prod_{i=1}^n \sin(x_i - z) + \prod_{i=1}^n \sin(5(x_i - z)) \right)$$

όπου $0 \leq x_i \leq \pi$.

Το καθολικό ελάχιστο βρίσκεται στο $x^* = (2.09435, 2.09435, \dots, 2.09435)$ με $f(x^*) = -3.5$.

Στα πειράματα χρησιμοποιήθηκε για $n=4, 8, 16, 32$ και $z = \frac{\pi}{6}$ και εκφράζονται από τις παρακάτω συναρτήσεις SINU4, SINU8, SINU16 και SINU32.

- **Test2N συνάρτηση.**

Η συνάρτηση δίνεται από την εξίσωση:

$$f(x) = \frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$$

όπου $x_i \in [-5, 5]$.

Η συνάρτηση έχει 2^n στο καθορισμένο εύρος και στα πειράματα χρησιμοποιήθηκαν για $n=4, 5, 6, 7$. Οι αντίστοιχες τιμές του καθολικού ελαχίστου είναι -156.664663 για $n=4$, -195.830829 για $n=5$, -234.996994 για $n=6$ και -274.163160 για $n=7$.

- **Test30N συνάρτηση.**

Η συνάρτηση δίνεται από την σχέση:

$$f(x) = \frac{1}{10} \sin^2(3\pi x_1) \left(\sum_{i=2}^{n-1} ((x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n)) \right)$$

όπου $x \in [-10,10]$, με τοπικά ελάχιστα 30n στον χώρο αναζήτησης. Στα πειράματα χρησιμοποιήθηκε για n=3,4 [40, 41, 42, 43].