

ΚΑΘΟΛΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ: ΜΕΘΟΔΟΙ,
ΛΟΓΙΣΜΙΚΟ ΚΑΙ ΕΦΑΡΜΟΓΕΣ

Η ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύγκλησης

του Τμήματος Πληροφορικής Εξεταστική Επιτροπή

από τον

ΙΩΑΝΝΗ ΤΣΟΥΛΟ

ως μέρος των Υποχρεώσεων για τη λήψη του

ΔΙΔΑΚΤΟΡΙΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ
ΠΛΗΡΟΦΟΡΙΚΗ

ΔΕΚΕΜΒΡΙΟΣ 2005

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ τον επιβλέποντα κ. Ισαάκ Λαγαρή και τα μέλη της τριμελούς επιτροπής κ. Αριστεΐδη Λύκα και Μιχαήλ Βραχάτη για την συμβολή τους στο επιστημονικό μέρος της διατριβής καθώς και τους γονείς μου Γρηγόρη και Καλλιόπη και τον αδερφό μου Γιώργο για την αμέριστη συμπαράστασή τους.

ΠΕΡΙΕΧΟΜΕΝΑ

1	Εισαγωγή	1
1.1	Βασικές έννοιες	2
1.2	Τοπική ελαχιστοποίηση	3
1.3	Κατηγορίες καθολικής ελαχιστοποίησης	3
1.4	Στόχοι της διατριβής	7
1.5	Περιεχόμενα της διατριβής	7
2	Στοχαστικές μέθοδοι	9
2.1	Γενικά	11
2.2	Η μέθοδος MULTISTART	11
2.3	Τεχνικές ομαδοποίησης	13
2.4	Η μέθοδος MLSL	13
2.5	Η μέθοδος TMSL	14
2.5.1	Ο αλγόριθμος	15
2.5.2	Το πλήθος των γειτόνων	16
2.5.3	Η κρίσιμη απόσταση	17
2.6	Η μέθοδος GTC (Ομαδοποίηση τυπικής αποστάσεως, ελεγχόμενη από την κλίση)	18
2.6.1	Η τυπική απόσταση	18
2.6.2	Το κριτήριο της κλίσεως	19
2.6.3	Ο προτεινόμενος αλγόριθμος	19
2.6.4	Πειραματικά αποτελέσματα	21
2.7	Η μέθοδος MinFinder	23
2.7.1	Περιγραφή του αλγορίθμου	23
2.7.2	Πειραματικά αποτελέσματα	25
2.8	Υλοποιημένο λογισμικό	25
2.9	Κανόνες τερματισμού	27
2.9.1	Τα κριτήρια των Rinnooy Kan, Boender και Zieliński	27
2.9.2	Ο κανόνας του διπλασίου κουτιού	28

2.9.3	Ο κανόνας των παρατηρησίμων μεγεθών	32
2.9.4	Ο κανόνας των αναμενόμενων ελαχίστων	33
2.9.5	Πειραματικά αποτελέσματα	35
3	Γενετικά βελτιωμένες στοχαστικές μέθοδοι	38
3.1	Γενικά	40
3.2	Γενετικοί αλγόριθμοι	40
3.2.1	Αναπαράσταση μελών πληθυσμού	41
3.2.2	Συνάρτηση καταλληλότητας	41
3.2.3	Επιλογή	42
3.2.4	Διασταύρωση	43
3.2.5	Ελιτισμός	44
3.2.6	Μετάλλαξη	45
3.2.7	Κριτήρια τερματισμού	45
3.3	Βασικές αρχές γενετικού προγραμματισμού	45
3.3.1	Χρησιμοποιούμενα σύνολα	46
3.3.2	Κλειστότητα	46
3.3.3	Επάρκεια	47
3.3.4	Γενετικοί τελεστές	48
3.4	Γραμματική Εξέλιξη	49
3.4.1	Βασικοί ορισμοί	49
3.4.2	Παράδειγμα παραγωγής	50
3.4.3	Γενετικοί τελεστές	51
3.5	Ο αλγόριθμος Genetic Random Search (GRS)	52
3.5.1	Προτεινόμενη γραμματική	53
3.5.2	Περιγραφή του αλγορίθμου	54
3.5.3	Παράδειγμα εφαρμογής	55
3.6	Η γενετικώς τροποποιημένη μέθοδος CRS (GenPrice)	55
3.7	Η γενετικώς τροποποιημένη μέθοδος Simulated Annealing (GenAnneal)	58
3.8	Πειραματικά αποτελέσματα	61
3.8.1	Αποτελέσματα CRS	62
3.8.2	Αποτελέσματα προσομοιούμενης ανοπτήσεως	62
3.9	Υλοποιημένο λογισμικό	64
4	Προσέγγιση συναρτήσεων με την χρήση Neural Splines	65
4.1	Γενικά	66
4.2	Neural Splines	66
4.2.1	Ορισμός του προβλήματος	66
4.2.2	Βασικοί ορισμοί	68
4.2.2.1	Πολύωνυμα Obreshkov	68

4.2.2.2	Neural Splines	69
4.2.3	Μονοδιάστατα προβλήματα	70
4.2.3.1	Ορισμός και εκπαίδευση	70
4.2.3.2	Πειραματικά αποτελέσματα	72
4.2.4	Προβλήματα δύο διαστάσεων	77
4.2.4.1	Προτεινόμενο μοντέλο	77
4.2.4.2	Πειραματικά αποτελέσματα	79
4.2.5	Συμπεράσματα	81
5	Επίλυση διαφορικών εξισώσεων	83
5.1	Γενικά	85
5.2	Προτεινόμενη τεχνική	85
5.3	Γραμματική	86
5.4	Αρχικοποίηση	86
5.5	Αποτίμηση καταλληλότητας	86
5.5.1	Συνήθεις διαφορικές εξισώσεις	88
5.5.2	Συστήματα διαφορικών εξισώσεων	88
5.5.3	Μερικές διαφορικές εξισώσεις	90
5.6	Υπολογισμός παραγώγων	91
5.7	Εφαρμογή γενετικών τελεστών	92
5.8	Κριτήρια τερματισμού	92
5.9	Πειραματικά αποτελέσματα	92
5.9.1	Παραδείγματα γραμμικών διαφορικών εξισώσεων	93
5.9.2	Παραδείγματα μη γραμμικών διαφορικών εξισώσεων	96
5.9.3	Παραδείγματα συστημάτων διαφορικών εξισώσεων	98
5.9.4	Συνήθεις διαφορικές εξισώσεις χωρίς αναλυτική λύση	100
5.9.5	Μια ειδική περίπτωση	101
5.9.6	Μερικές διαφορικές εξισώσεις	102
5.10	Συμπεράσματα	106

ΣΧΗΜΑΤΑ

2.1	Γράφημα της ποσότητας r_k ως προς k	17
2.2	Γράφημα της μέσης τιμής της ποσότητας δ_k και της διακυμάνσεως αυτής.	31
3.1	Διασταύρωση ενός σημείου.	43
3.2	Διασταύρωση δύο σημείων.	44
3.3	Συντακτικό δένδρο του $x + \sin(x + 2 * 5)$	46
3.4	Παράδειγμα διασταυρώσεως στον γενετικό προγραμματισμό.	48
3.5	Γραμματική μια γλώσσας με πράξεις ανάμεσα σε x και y	50
3.6	Παραγωγή έκφρασης.	51
3.7	Διασταύρωση ενός σημείου σε gramamatical evolution.	52
3.8	Η γραμματική της μεθόδου GRS.	53
3.9	Βήματα για το ολικό ελάχιστο της GKLS(2,30).	56
3.10	Εύρεση του ολικού ελαχίστου της GKLS(2,30) ξεκινώντας από διαφορετικά σημεία.	56
4.1	Γραφική παράσταση της $f(x) = x \sin(x^2)$ στο διάστημα $[-4, 4]$	72
4.2	Γραφική αναπαράσταση της προσεγγίσεως της $f(x) = x \sin(x^2)$ παρουσία θορύβου από διαφορετικές μεθόδους.	76
4.3	Λεπτομέρειες στην προσέγγιση της συναρτήσεως $f(x) = x \sin(x^2)$ γύρω από ένα σημείο τομής.	76
4.4	Σημεία για τον υπολογισμό της συναρτήσεως ϕ_{top}	79
4.5	Γραφική παράσταση της συναρτήσεως $f(x, y) = xy \sin(2y^2 + 3x)$ στο διάστημα $[-1, 1] \otimes [-1, 1]$	80
5.1	Η γραμματική της προτεινόμενης τεχνικής για επίλυση διαφορικών εξισώσεων.	87
5.2	Υποψήφιες λύσεις της $y' = -100y$ με οριακές συνθήκες στο αριστερό άκρο.	96
5.3	Υποψήφιες λύσεις της NLODE3.	98

5.4	Γράφημα των GP1(x) και $y(x) = \int_0^x \frac{\sin(t)}{t} dt$	100
5.5	Γράφημα των GP2(x) και $y(x) = \int_0^x \exp(-t^2) dt$	101
5.6	GP(x) και $y(x) = (x^2 + 1) \operatorname{atan}(x)$	102
5.7	Διαφορά μεταξύ των $\Psi(x, y) = \sin(x) \cos(y)$ και GP1(x, y).	105
5.8	Διαφορά μεταξύ των $\Psi(x, y) = \sin(x) \cos(y)$ και GP10(x, y).	105
5.9	Διαφορά μεταξύ των $\Psi(x, y) = \sin(x) \cos(y)$ και GP40(x, y).	106
B.1	Κωδικοποίηση αντικειμενικών συναρτήσεων στην γλώσσα προγραμματισμού C.	128
B.2	Κωδικοποίηση αντικειμενικών συναρτήσεων στην γλώσσα προγραμματισμού Fortran 77.	129
B.3	Υλοποίηση της συναρτήσεως CAMEL στην γλώσσα προγραμματισμού C++.	130
B.4	Υλοποίηση της συναρτήσεως CAMEL στην γλώσσα προγραμματισμού Fortran 77.	131
Γ.1	Κωδικοποίηση της συναρτήσεως RASTRIGIN σε ANSI C++.	136
Γ.2	Κωδικοποίηση της συναρτήσεως RASTRIGIN σε Fortran 77.	137

ΠΙΝΑΚΕΣ

2.1	Συγκριτικά αποτελέσματα MULTISTART, TMLSL και GTC σε μια σειρά από συναρτήσεις δοκιμής με την χρήση του κριτηρίου τερματισμού του διπλού κουτιού.	22
2.2	Αποτελέσματα της μεθόδου MINFINDER σε μια σειρά από συναρτήσεις δοκιμής με την χρήση του κριτηρίου τερματισμού του διπλού κουτιού.	26
2.3	Πειράματα με την ποσότητα p στην μέθοδο του διπλασίου κουτιού και με την χρήση της μεθόδου MULTISTART σε μια σειρά από συναρτήσεις δοκιμής.	31
2.4	Αποτελέσματα εφαρμογής διαφόρων κριτηρίων τερματισμού στην μέθοδο MULTISTART.	36
2.5	Αποτελέσματα εφαρμογής διαφόρων κριτηρίων τερματισμού στην μέθοδο TMLSL.	37
3.1	Παραδείγματα προστατευομένων συναρτήσεων.	47
3.2	Συγκριτικά αποτελέσματα διαφορετικών εκδοχών της Controlled Random Search σε μια σειρά από γνωστά προβλήματα καθολικής ελαχιστοποίησης.	63
3.3	Αποτελέσματα από την εφαρμογή των μεθόδων SA, ASA και GenAnneal σε μια σειρά από συναρτήσεις.	64
4.1	Ομοιόμορφα σημεία χωρίς θόρυβο - Απλό τεχνητό νευρωνικό δίκτυο.	73
4.2	Ομοιόμορφα σημεία χωρίς θόρυβο - τμηματικά νευρωνικά δίκτυα.	74
4.3	Ομοιόμορφα σημεία χωρίς θόρυβο - Προτεινόμενο μοντέλο με συνέχεια τιμής.	74
4.4	Ομοιόμορφα σημεία χωρίς θόρυβο - Προτεινόμενο μοντέλο με συνέχεια παραγώγου.	75
4.5	Ομοιόμορφα σημεία με θόρυβο - Απλό τεχνητό νευρωνικό δίκτυο.	75

4.6	Ομοιόμορφα σημεία με θόρυβο -τμηματικά νευρωνικά δίκτυα. . . .	77
4.7	Ομοιόμορφα σημεία με θόρυβο - Προτεινόμενο μοντέλο με συνέ- χεια τιμής.	77
4.8	Ομοιόμορφα σημεία με θόρυβο - Προτεινόμενο μοντέλο με συνέ- χεια παραγώγου.	78
4.9	Ομοιόμορφα δεδομένα στον διδιάστατο χώρο - Τμηματικά νευρω- νικά δίκτυα.	80
4.10	Ομοιόμορφα δεδομένα στον διδιάστατο χώρο - Προτεινόμενο μο- ντέλο.	81
4.11	Ομοιόμορφα δεδομένα στον διδιάστατο χώρο με θόρυβο - Τμημα- τικά νευρωνικά δίκτυα.	81
4.12	Ομοιόμορφα δεδομένα στον διδιάστατο χώρο με θόρυβο- Προτει- νόμενο μοντέλο.	82
5.1	Αποτελέσματα της προτεινόμενης μεθόδου για γραμμικές διαφορι- κές εξισώσεις.	95
5.2	Αποτελέσματα εφαρμογής της μεθόδου σε μη γραμμικές διαφορι- κές εξισώσεις.	97
5.3	Αποτελέσματα εφαρμογής της μεθόδου σε συστήματα διαφορικών εξισώσεων.	99
5.4	Αποτελέσματα εφαρμογής της μεθόδου σε μερικές διαφορικές εξι- σώσεις.	104

ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας διατριβής είναι η καθολική ελαχιστοποίηση, ένας τομέας με μεγάλη άνθηση τα τελευταία χρόνια και με πολλές πρακτικές εφαρμογές. Ωστόσο, καθώς αυτή η περιοχή είναι αρκετά ευρεία η παρούσα διατριβή περιορίζεται σε στοχαστικές μεθόδους και ιδιαίτερα σε τεχνικές ομαδοποιήσεως και σε υβριδικές στοχαστικές - γενετικές τεχνικές. Επίσης αναπτύσσονται μετά από ανάλυση νέα κριτήρια τερματισμού, ένα θέμα που έχει απασχολήσει σοβαρά την σχετική επιστημονική κοινότητα. Τέλος περιγράφονται δύο εφαρμογές, μία για προσέγγιση συναρτήσεων και μία για επίλυση διαφορικών εξισώσεων.

Τεχνικές ομαδοποιήσεως

Προτείνονται δύο νέες τεχνικές που αποσκοπούν στην εύρεση όλων των τοπικών ελαχίστων εντός ενός πεπερασμένου πολυδιάστατου χώρου και ανήκουν στην κατηγορία των μεθόδων ομαδοποιήσεως. Στοχεύουν στον περιορισμό των εκκινήσεων διαδικασιών τοπικής ελαχιστοποίησης διαλέγοντας ένα μόνο σημείο από αυτά που πιστεύεται πως ανήκουν στην ίδια περιοχή προσελκύσεως.

Κριτήρια τερματισμού

Παρουσιάζονται τρία νέα κριτήρια τερματισμού μεθόδων καθολικής ελαχιστοποίησης που έχουν σαν στόχο τον τερματισμό αυτών των μεθόδων, όταν με κάποια πιθανότητα θα έχουν βρεθεί όλα τα τοπικά ελάχιστα της συναρτήσεως. Το πρώτο κριτήριο εκτιμά το ποσοστό καλύψεως του χώρου αναζητήσεως με τις μέχρι τούδε εφαρμογές της τοπικής ελαχιστοποίησης. Το δεύτερο κριτήριο στηρίζεται στην σύγκριση ανάμεσα στις ασυμπτωτικές αναμενόμενες τιμές παρατηρήσιμων μεγεθών και στις τιμές που παρατηρούνται κατά την εκτέλεση της διαδικασίας. Το τρίτο κριτήριο επιδιώκει να εκτιμήσει το πλήθος των τοπικών ελαχίστων της αντικειμενικής συναρτήσεως.

Γενετικώς τροποποιημένες μέθοδοι

Με την χρήση της μεθοδολογίας του γενετικού προγραμματισμού επεκτείνονται δύο γνωστές τεχνικές καθολικής ελαχιστοποιήσεως: η μέθοδος της Ελεγχόμενης Τυχαίας Αναζήτησεως και η μέθοδος της Προσομοιούμενης Ανοπτήσεως. Αυτό που διαφοροποιεί τις προκύπτουσες τεχνικές είναι ο διαφορετικός τρόπος δειγματοληψίας, που χρησιμοποιεί μια ακολουθία σημείων που παράγονται με την χρήση γενετικού προγραμματισμού.

Προσέγγιση συναρτήσεων

Για την προσέγγιση συναρτήσεων εισάγεται η έννοια των Neural Splines, τα οποία είναι μαθηματικά μοντέλα και αποτελούν συνδυασμό τεχνητών νευρωνικών δικτύων και ειδικής μορφής πολυωνύμων και μπορούν να χρησιμοποιηθούν σαν δομική βάση για την δημιουργία τμηματικά συνεχών μοντέλων. Αυτά τα μοντέλα είναι κατάλληλα για προσέγγιση συναρτήσεων σε διαμερισμένο χώρο και μπορούν να παραλληλοποιηθούν. Για την αποτελεσματική εκπαίδευση αυτών των μοντέλων είναι απαραίτητες μέθοδοι καθολικής ελαχιστοποιήσεως.

Επίλυση διαφορικών εξισώσεων

Για την επίλυση συνήθων διαφορικών εξισώσεων καθώς και ελλειπτικών μερικών διαφορικών εξισώσεων χρησιμοποιήθηκε μια μεθοδολογία που βασίζεται σε γενετικό προγραμματισμό και συγκεκριμένα στην “Γραμματική Εξέλιξη”. Σε αυτήν την μεθοδολογία χρησιμοποιείται μια σειρά από υποψήφιες λύσεις, εκφρασμένες σε ένα υποσύνολο του συντακτικού της γλώσσας προγραμματισμού C, στις οποίες εφαρμόζονται οι γενετικοί τελεστές της διασταυρώσεως και της μεταλλάξεως.

EXTENDED ABSTRACT IN ENGLISH

The subject of this dissertation is in the area of global optimization, an important area with many applications. Nevertheless, as this area is very wide, this dissertation focuses on stochastic methods and specifically on clustering techniques and hybrid stochastic - genetic techniques. New stopping rules are developed and two applications are presented.

Clustering techniques

Two new clustering techniques, which attempt to discover all the existing local minima inside a bounded rectangular domain, are introduced. The methods aim to minimize the number of applications of the local search procedure by selecting only one point from a group that is believed to belong to the same region of attraction.

Stopping rules

Three new stopping rules for global optimization methods are presented. These rules aim to terminate the search process when all the local minima is believed that have been discovered. The first rule is based on an estimate of the covered portion of the search space. The second rule is based on the comparison of asymptotic expectation values of observable quantities to the actually measured ones. The third exploits an estimate of the number of expected local minima inside the search domain.

Genetically modified methods

Two well known global optimization methods are extended through genetic programming: the method of Controlled Random Search and the method of

Simulated Annealing. The main difference between the new and the old methods is the introduced sampling technique, that uses points produced by genetic programming.

Function approximation

The mathematical model of *Neural Splines* is introduced. This model is a combination of an artificial neural network and an associated Obreshkov polynomial and it is used for function approximation. The neural spline has finite support and can be used as the basic element in constructing continuous modular neural-based models that are suitable for function approximation in partitioned domains and ideal for parallel implementation. The training of these kind of models requires global optimization methods.

Solving differential equations

An evolutionary technique, based on genetic programming (more specifically on grammatical evolution), was implemented to solve ordinary as well as elliptic partial differential equations. The technique uses a population of candidate solutions, expressed in a subset of the C programming language's grammar, on which the genetic operators of mutation and crossover are applied.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Περιεχόμενα

1.1	Βασικές έννοιες	2
1.2	Τοπική ελαχιστοποίηση	3
1.3	Κατηγορίες καθολικής ελαχιστοποίησης	3
1.4	Στόχοι της διατριβής	7
1.5	Περιεχόμενα της διατριβής	7

1.1 Βασικές έννοιες

Η περιοχή της καθολικής βελτιστοποίησης είναι ένα πεδίο που παρουσίασε σημαντική ανάπτυξη από τα μέσα της δεκαετίας του '80 και μετά. Σημαντικό ρόλο διαδραμάτισε η ραγδαία ανάπτυξη των Η/Υ που είχε ως αποτέλεσμα την αύξηση της ταχύτητας των υπολογισμών με εκθετικούς ρυθμούς.

Η καθολική βελτιστοποίηση είναι χρήσιμη, γιατί βρίσκει εφαρμογή σε πολλά επιστημονικά και πρακτικά προβλήματα. Για παράδειγμα αναφέρουμε τα προβλήματα “εντοπισμού σταθερών μοριακών διαμορφώσεων”, της “πρωτεϊνικής αναδιπλώσεως (protein folding)”, της προσαρμογής δεδομένων σε θεωρητικά και παραμετρικά μοντέλα κλπ.

Οι υπολογισμοί που απαιτούνται είναι συνήθως μεγάλης εκτάσεως και για αυτόν τον λόγο η ανάπτυξη τεχνικών καθολικής βελτιστοποίησης καθυστέρησε, συγκρινόμενη με την ανάπτυξη τοπικών μεθόδων, που άρχισε στα μέσα της δεκαετίας του '40 και γνώρισε έξαρση στην δεκαετία του '60.

Η τοπική βελτιστοποίηση έχει βρει εφαρμογές σε πολλά επιστημονικά πεδία, όπως αυτά της Φυσικής, Χημείας, Μοριακής Βιολογίας, Οικονομίας, Επεξεργασίας Σήματος και Εικόνας, Μηχανικής κλπ. Δεδομένου ότι μια συνάρτηση μπορεί να έχει περισσότερα του ενός τοπικά ελάχιστα, η χρησιμότητα των μεθόδων καθολικής ελαχιστοποίησης γίνεται προφανής.

Το πρόβλημα που αντιμετωπίζει η καθολική ελαχιστοποίηση διατυπώνεται ως εξής: Δοθείσης μιας συναρτήσεως $f(x)$, $x \in S \subset R^n$, ζητείται να βρεθεί ένα σημείο $x^* \in S$ τέτοιο ώστε να ισχύει:

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

Η υπό ελαχιστοποίηση συνάρτηση αναφέρεται με πλήθος ονομάτων, όπως: αντικειμενική συνάρτηση, συνάρτηση κόστους, συνάρτηση σφάλματος κ.ά. Η θέση x^* του καθολικού ελαχίστου ονομάζεται “καθολικός ελαχιστοποιητής” και η τιμή $f(x^*)$ “καθολικό ελάχιστο”. Σε αντιδιαστολή το πρόβλημα της τοπικής ελαχιστοποίησης διατυπώνεται ως: Ζητείται να βρεθεί ένα σημείο $x^* \in S$ τέτοιο ώστε να ισχύει:

$$f(x^*) < f(x), \forall x \in S, |x^* - x| \leq \epsilon$$

όπου ϵ ένας μικρός θετικός αριθμός που ορίζει την ακτίνα μιας σφαιρικής γειτονιάς πέριξ του x^* . Το σημείο x^* ονομάζεται σε αυτήν την περίπτωση “τοπικός ελαχιστοποιητής” και η τιμή $f(x^*)$ “τοπικό ελάχιστο”. Από τους παραπάνω ορισμούς μπορεί κανείς να συμπεράνει πως το καθολικό ελάχιστο είναι και τοπικό ελάχιστο.

1.2 Τοπική ελαχιστοποίηση

Οι διάφορες μέθοδοι τοπικής ελαχιστοποίησης πολλές φορές αποτελούν την βάση για μεθόδους καθολικής ελαχιστοποίησης ή χρησιμοποιούνται από αυτές για την εύρεση του καθολικού ελαχίστου. Μάλιστα η κατάλληλη επιλογή μιας μεθόδου τοπικής ελαχιστοποίησης μπορεί να επιφέρει σημαντικές διαφοροποιήσεις στα παραγόμενα αποτελέσματα. Μια κατηγοριοποίηση των μεθόδων τοπικής ελαχιστοποίησης που βασίζεται στην συναρτησιακή πληροφορία που χρησιμοποιούν παρατίθεται:

1. Μέθοδοι που χρησιμοποιούν μόνον συναρτησιακές τιμές (Direct Search). Οι μέθοδοι αυτής της κατηγορίας είναι κατάλληλες για περιπτώσεις όπου η αντικειμενική συνάρτηση είτε δεν είναι ομαλή είτε περιέχει θόρυβο. Σε αυτήν την κατηγορία βρίσκονται οι μέθοδοι ιχνοαναζήτησης (pattern search) [15, 16, 14], οι μέθοδοι πολύτοπου (simplex) [17, 18] κτλ.
2. Μέθοδοι που χρησιμοποιούν πρώτη παράγωγο. Οι μέθοδοι αυτές είναι κατάλληλες για συνεχείς αντικειμενικές συναρτήσεις με συνεχή κλίση. Στην περίπτωση που η παράγωγος δεν διατίθεται αναλυτικά οι μέθοδοι αυτές καταφεύγουν σε χρήση πεπερασμένων διαφορών. Μέθοδοι αυτόματης παραγωγίσεως που βασίζονται στον κανόνα της αλυσίδας για παραγωγή επίσης έχουν αναπτυχθεί [109]. Σε αυτήν την κατηγορία υπάγονται οι μέθοδοι steepest descent, συζυγών κλίσεων (Fletcher - Reeves [12] και Polak - Ribière [13]), Quasi - Newton όπως η DFP [1, 2] και η BFGS [3, 4, 5, 6], η Limited memory BFGS (L-BFGS) [7] κλπ.
3. Μέθοδοι που χρησιμοποιούν παραγώγους υψηλότερης τάξεως. Οι πιο διαδεδομένες μέθοδοι σε αυτήν την περιοχή είναι παραλλαγές της μεθόδου του Newton, δηλαδή βασίζονται στην χρήση του Εσσιανού πίνακα. Σε αυτήν την περιοχή συναντά κανείς μεθόδους όπως αυτές των Gill - Murray [9], η μέθοδος Truncated Newton [10], οι τανυστικές (Tensor) μέθοδοι [8] κλπ.

1.3 Κατηγορίες καθολικής ελαχιστοποίησης

Το πεδίο της καθολικής ελαχιστοποίησης είναι ευρύ και μπορεί να χωριστεί σε κατηγορίες. Στην σχετική βιβλιογραφία αναφέρονται διαφορετικές κατηγοριοποιήσεις της περιοχής, όπως αυτές των Dixon και Szegö [20], Gormulka [21] και των Archetti και Schoen [22]. Βασιζόμενοι στις παραπάνω εργασίες μπορούμε να χρησιμοποιήσουμε τον ακόλουθο διαχωρισμό της περιοχής:

1. Αιτιοκρατικές (deterministic) μέθοδοι. Στην κατηγορία αυτήν ανήκουν μέθοδοι όπως:

- (α) Μέθοδοι της κατηγορίας **αναζήτησεως πλέγματος** (Grid Search), όπου δημιουργούνται λίστες με σημεία που θα χρησιμοποιηθούν σαν σημεία εκκινήσεως τοπικής ελαχιστοποίησης. Η εισαγωγή ενός σημείου σε μία λίστα αναζήτησεως δεν εξαρτάται από σημεία που έχουν ήδη εισαχθεί σε αυτήν. Εναλλακτικοί τρόποι σχηματισμού του πλέγματος παρουσιάζονται στις εργασίες των Leonov [23], Ganshin [24, 25] και Babij [26].
- (β) **Τροχιακές μέθοδοι** (Trajectory methods), όπου η αναζήτηση γίνεται κατά μήκος μιας τροχιάς, που ιδανικά διέρχεται από όλα τα τοπικά ελάχιστα. Στην κατηγορία αυτή έχουν προταθεί μέθοδοι από τους Fiodorova [27], Inomota και Cumada [28], Zhidkov και Shchedrin [29], Incerti [30], Branin [31].
- (γ) **Μέθοδοι καλύψεως**. Στην κατηγορία αυτή βρίσκονται μέθοδοι που ανιχνεύουν υποπεριοχές του χώρου αναζήτησεως στις οποίες δεν βρίσκεται το καθολικό ελάχιστο και τις αποκλείουν από περαιτέρω αναζήτηση. Κατά αυτόν τον τρόπο εντοπίζεται τελικά μια μικρή περιοχή η οποία περιέχει το καθολικό ελάχιστο. Στην περιοχή αυτή συναντάμε τους αλγόριθμους των Pijavskij-Shubert [32, 33], του Brent [34], του Wood [35], τεχνικές διαστημάτων [57, 58] κ.ά.
- (δ) **Μέθοδοι συμπληρωματικών συναρτήσεων** (Filled function methods). Σε αυτήν την κατηγορία μεθόδων η αντικειμενική συνάρτηση χρησιμοποιείται σαν βάση για την δημιουργία συναρτήσεων που διευκολύνουν την ελαχιστοποίηση. Τυπικό παράδειγμα αυτής της κατηγορίας μεθόδων αποτελούν οι μέθοδοι που βασίζονται σε συναρτήσεις τιμωρίας (penalty functions). Στις μεθόδους αυτής της κατηγορίας κάθε φορά που ανακαλύπτεται ένα τοπικό ελάχιστο τροποποιείται η αντικειμενική συνάρτηση έτσι ώστε να αποφευχθεί η εύρεση του ίδιου τοπικού ελαχίστου σε επόμενη επανάληψη του αλγορίθμου με τη προσθήκη ειδικών συναρτήσεων (penalties). Στην εργασία των Ge και Qin [36] εμφανίζονται διαφορετικές μορφές συμπληρωματικών συναρτήσεων.
- (ε) **Μέθοδοι tunneling**. Οι μέθοδοι αυτής της κατηγορίας βασίζονται σε δύο φάσεις: στην φάση της τοπικής ελαχιστοποίησης και στην φάση του tunneling. Στην πρώτη φάση βρίσκεται ένα τοπικό ελάχιστο και στην δεύτερη φάση η αντικειμενική συνάρτηση τροποποιείται με έναν παράγοντα τιμωρίας που εξαρτάται από το τοπικό ελάχιστο που βρέθηκε στην πρώτη φάση. Σκοπός της δεύτερης φάσεως είναι να βρεθεί ένα σημείο με χαμηλότερη συναρτησιακή τιμή από το τελευταίο ελάχιστο. Οι μέθοδοι tunneling παρουσιάστηκαν αρχικά από τον

Vilkov [37] για την περίπτωση μονοδιάστατων συναρτήσεων και στην συνέχεια επεκτάθηκε από τους Montalvo [38], Gomez και Levy [39], Levy [40], Levy και Montalvo [41] για την εφαρμογή των μεθόδων σε πολυδιάστατες συναρτήσεις.

2. Στοχαστικές μέθοδοι. Στην κατηγορία αυτή περιλαμβάνονται μέθοδοι όπως:

(α) Μέθοδοι τυχαίας αναζήτησης.

- i. **Αμιγώς τυχαία αναζήτηση** (pure random search), όπου η συνάρτηση αποτιμάται σε μια σειρά από σημεία επιλεγμένα με τυχαίο τρόπο και ελάχιστο θεωρείται το σημείο με την χαμηλότερη συναρτησιακή τιμή.
- ii. **Μοναδικής εκκινήσεως** (singlestart), όπου ύστερα από μια αμιγώς τυχαία αναζήτηση μία μέθοδος τοπικής ελαχιστοποίησης εφαρμόζεται στο σημείο με την χαμηλότερη συναρτησιακή τιμή.
- iii. **Πολλαπλής εκκινήσεως** (multistart), όπου μια μέθοδος τοπικής ελαχιστοποίησης εφαρμόζεται σε μία σειρά από σημεία επιλεγμένα με τυχαίο τρόπο και καθολικό ελάχιστο θεωρείται το ελάχιστο με την μικρότερη συναρτησιακή τιμή. Η μέθοδος αυτή αποτελεί την βάση για μια σειρά από αποτελεσματικότερες και αποδοτικότερες τεχνικές καθολικής ελαχιστοποίησης.
- iv. **Μέθοδοι αναζήτησεων με απαγορεύσεις** (tabu search). Στις μεθόδους αυτής της κατηγορίας δημιουργείται μία σειρά από κινήσεις. Μία κίνηση επιλέγεται να εκτελεστεί αν έχει διαπιστωθεί από τις προηγούμενες επαναλήψεις του αλγορίθμου ότι μπορεί να οδηγήσει σε χαμηλότερη τιμή της αντικειμενικής συναρτήσεως. Επιπλέον κατά την διάρκεια εκτελέσεως του αλγορίθμου διατηρείται ένα ιστορικό με μετακινήσεις που θεωρούνται απαγορευμένες, για την αποφυγή εγκλωβισμού της μεθόδου σε τοπικά ελάχιστα που είχαν ανακαλυφθεί σε προηγούμενα βήματα του αλγορίθμου. Περισσότερα για αυτές τις μεθόδους μπορεί κανείς να βρει στις σχετικές εργασίες [63, 64].
- v. **Μέθοδοι ελεγχόμενης τυχαίας αναζήτησεως** (Controlled Random Search). Στις μεθόδους αυτές [110, 111, 112, 113] η συναρτησιακή τιμή αποτιμάται σε μια σειρά από σημεία τα οποία χρησιμοποιούνται σαν βάση για μια σειρά από ενέργειες που εκτελούνται στην συνέχεια, όπως την δημιουργία του κεντροειδούς του δείγματος και την δημιουργία ομάδων από την αρχική στα οποία η αναζήτηση για το ολικό ελάχιστο ενδεχομένως να αποδοτικότερη.

vi. **Μέθοδοι Προσομοιούμενης Ανοπτήσεως (Simulated Annealing)** [59, 60, 61, 62, 117]. Η διαδικασία αυτή βασίζεται στην διαδικασία της ανοπτήσεως από την Φυσική, όπου ένα υλικό αφού θερμανθεί αρκετά στην συνέχεια ψύχεται σταδιακά ώστε να φτάσει σε κατάσταση χαμηλής ενέργειας. Αρχικά χρησιμοποιήθηκε με επιτυχία σε αλγόριθμους διακριτής ελαχιστοποίησης όπως στο πρόβλημα του πλανοδίου πωλητή, αλλά στην συνέχεια εφαρμόστηκε με επιτυχία και σε προβλήματα ελαχιστοποίησης συνεχών συναρτήσεων. Η φυσική διαδικασία της ανοπτήσεως στην Φυσική μπορεί να περιγραφεί ως εξής: καταρχήν το υλικό θερμαίνεται μέχρι να λιώσει και στην συνέχεια ψύχεται με σταδιακή και αργή μείωση της θερμοκρασίας. Στην υγρή κατάσταση τα σωματίδια του υλικού κατανέμονται τυχαία εξαιτίας της υψηλής θερμοκρασίας. Στην συνέχεια με την σταδιακή μείωση της θερμοκρασίας τα σωματίδια θα οδηγηθούν σε μια κατάσταση χαμηλής ενέργειας και θα αποκτήσουν μία συγκεκριμένη δομή, όπως αυτές που εμφανίζονται σε κρυσταλλικές μορφές. Αν η θερμοκρασία μειώνεται γρήγορα (κάτι που αποκαλείται quenching) το σύστημα ηρεμεί σε κατάσταση υψηλής ενέργειας. Στην διαδικασία καθολικής ελαχιστοποίησης η κατάσταση ελάχιστης ενέργειας είναι το ολικό ελάχιστο που αναζητείται από την μέθοδο.

(β) **Μέθοδοι ομαδοποιήσεως (Clustering)**. Η βασική ιδέα πίσω από τις τεχνικές αυτής της κατηγορίας είναι να δημιουργηθούν ομάδες σημείων τα οποία πιστεύεται πως ύστερα από τοπική ελαχιστοποίηση θα οδηγούσαν στο ίδιο τοπικό ελάχιστο. Στην περιοχή αυτή εμφανίζονται πολλές μέθοδοι όπως η μέθοδος των Timmer et al [11], Becker και Lago [42], η μέθοδος του Density Linkage των Wong και Lane με την χρήση των κ-κοντινότερων γειτόνων [43], ο αλγόριθμος του Törn [44], η μέθοδος του Timmer [87], η μέθοδος Topographical Multilvel Single Linkage των Ali και Storey [83] κ.ά.

(γ) **Εξελικτικές μέθοδοι**. Στην κατηγορία αυτή περιλαμβάνονται μέθοδοι που είναι εμπνευσμένες από τις φυσικές διαδικασίες της εξέλιξης, όπως οι γενετικοί αλγόριθμοι [19], ο γενετικός προγραμματισμός [88], βελτιστοποίηση βασισμένη σε σμήνος σωματιδίων [47, 48] (Particle Swarm Optimization), μέθοδοι διαφορικής εξέλιξης [46] (differential evolution) κ.ά.

1.4 Στόχοι της διατριβής

Η παρούσα διατριβή στοχεύει κυρίως στην ανάπτυξη νέων αλγορίθμων καθολικής ελαχιστοποίησης αλλά και στην εφαρμογή μεθόδων καθολικής ελαχιστοποίησης για την επίλυση πρακτικών προβλημάτων. Για την επίτευξη του πρώτου στόχου σχεδιάστηκαν νέες τεχνικές ομαδοποιήσεως που στοχεύουν στην εύρεση όλων των ελαχίστων της αντικειμενικής συναρτήσεως, νέες τεχνικές τερματισμού, καθώς και νέες τεχνικές που αποσκοπούν στην εύρεση μόνον του καθολικού ελαχίστου. Για την επίτευξη του δεύτερου στόχου διαλέγουμε προβλήματα από την κατηγορία της προσαρμογής δεδομένων, που αντιμετωπίστηκαν με καθολική ελαχιστοποίηση και την επίλυση διαφορικών εξισώσεων με γενετικές τεχνικές καθολικής ελαχιστοποίησης.

1.5 Περιεχόμενα της διατριβής

Η διατριβή αποτελείται από 6 κεφάλαια και τρία παραρτήματα ως ακολούθως:

1. Το πρώτο (παρόν) κεφάλαιο της εισαγωγής.
2. Στο δεύτερο κεφάλαιο παρουσιάζονται τρεις νέες τεχνικές τερματισμού κατάλληλες για στοχαστικές μεθόδους τύπου Multistart, και δύο νέες τεχνικές καθολικής ελαχιστοποίησης που βασίζονται στην τεχνική της ομαδοποίησης [45, 97].
3. Στο τρίτο κεφάλαιο, γίνεται αρχικά μια ανασκόπηση του γενετικού προγραμματισμού και της “Γραμματικής Εξελίξεως” (Grammatical Evolution). Στην συνέχεια τροποποιούνται οι τεχνικές της “ελεγχόμενης τυχαίας αναζήτησεως” [52] και της “Προσομοιούμενης Ανοπτήσεως” [53] (Simulated Annealing) με ιδέες προερχόμενες από τον γενετικό προγραμματισμό και την “Γραμματική Εξέλιξη”.
4. Στο τέταρτο κεφάλαιο εισάγεται μια νέα τεχνική [54] για την προσέγγιση συναρτήσεων, που βασίζεται στα τεχνητά νευρωνικά δίκτυα και στα πολυώλυμα Obreshkov. Η μέθοδος απαιτεί την ελαχιστοποίηση συναρτήσεων με πολλά τοπικά ελάχιστα και συνεπώς οι τεχνικές καθολικής ελαχιστοποίησης είναι απαραίτητες.
5. Στο πέμπτο κεφάλαιο προτείνεται μια τεχνική για επίλυση συνήθων και μερικών διαφορικών εξισώσεων [55] βασισμένη στην μεθοδολογία της “Γραμματικής Εξελίξεως”.
6. Στο πρώτο παράρτημα παρουσιάζονται μια σειρά συναρτήσεων που χρησιμοποιούνται για την αξιολόγηση τεχνικών καθολικής ελαχιστοποίησης,

πολλές εκ των οποίων συνιστώνται από τους Dixon και Szegö [20], Törn και Žilinskas [56] και Pardalos et al [49].

7. Στο δεύτερο παράρτημα παρουσιάζεται το υλοποιημένο λογισμικό για την μέθοδο ομαδοποίησης MinFinder, συνοδευόμενο από τις σχετικές οδηγίες εγκατάστασής και χρήσής.
8. Στο τρίτο παράρτημα παρουσιάζεται το υλοποιημένο λογισμικό για τις νέες μεθόδους GenPrice και GenAnneal του τρίτου κεφαλαίου. Τα πακέτα αυτά καθώς και το πακέτο MinFinder είναι διαθέσιμα από την βιβλιοθήκη προγραμμάτων του περιοδικού Computer Physics Communications, καθώς και από την ηλεκτρονική διεύθυνση

<http://www.cs.uoi.gr/~lagaris/PREPRINTS/>

ΚΕΦΑΛΑΙΟ 2

ΣΤΟΧΑΣΤΙΚΕΣ ΜΕΘΟΔΟΙ

Περιεχόμενα

2.1	Γενικά	11
2.2	Η μέθοδος MULTISTART	11
2.3	Τεχνικές ομαδοποιήσεως	13
2.4	Η μέθοδος MLSL	13
2.5	Η μέθοδος TMLSL	14
2.5.1	Ο αλγόριθμος	15
2.5.2	Το πλήθος των γειτόνων	16
2.5.3	Η κρίσιμη απόσταση	17
2.6	Η μέθοδος GTC (Ομαδοποίηση τυπικής αποστάσεως, ελεγχό- μενη από την κλίση)	18
2.6.1	Η τυπική απόσταση	18
2.6.2	Το κριτήριο της κλίσεως	19
2.6.3	Ο προτεινόμενος αλγόριθμος	19
2.6.4	Πειραματικά αποτελέσματα	21
2.7	Η μέθοδος MinFinder	23
2.7.1	Περιγραφή του αλγορίθμου	23
2.7.2	Πειραματικά αποτελέσματα	25
2.8	Υλοποιημένο λογισμικό	25

2.9	Κανόνες τερματισμού	27
2.9.1	Τα κριτήρια των Rinnooy Kan, Boender και Zieliński. . .	27
2.9.2	Ο κανόνας του διπλασίου κουτιού	28
2.9.3	Ο κανόνας των παρατηρησίμων μεγεθών	32
2.9.4	Ο κανόνας των αναμενόμενων ελαχίστων	33
2.9.5	Πειραματικά αποτελέσματα	35

2.1 Γενικά

Οι στοχαστικές μέθοδοι βασίζονται στην χρήση ψευδοτυχαίων αριθμών. Η σύγκλισή τους στο καθολικό ελάχιστο είναι ασυμπτωτική, δηλαδή το καθολικό ελάχιστο ανακαλύπτεται στην k -οστή επανάληψη με πιθανότητα p_k για την οποία ισχύει

$$\lim_{k \rightarrow \infty} p_k = 1$$

Από τις απλούστερες στοχαστικές μεθόδους και ίσως η πλέον χρησιμοποιημένη, είναι η μέθοδος πολλαπλών ενάρξεων γνωστή ως MULTISTART. Άλλες, επίσης μέθοδοι που χρησιμοποιούνται αρκετά είναι η “Προσομοίωση Ανοπτύσεως” και η “Ελεγχόμενη Τυχαία Αναζήτηση”. Στην συνέχεια γίνεται μία ανασκόπηση της MULTISTART που αποτελεί την βάση για δύο νέες μεθόδους που αναπτύχθηκαν στα πλαίσια της παρούσας διατριβής.

2.2 Η μέθοδος MULTISTART

Μία από τις απλούστερες μεθόδους καθολικής ελαχιστοποίησης είναι η MULTISTART, η οποία αποτελεί την βάση για την ανάπτυξη νέων στοχαστικών μεθόδων. Η Multistart εκκινεί τοπική ελαχιστοποίηση από διαφορετικά αρχικά σημεία στον χώρο αναζήτησεως, συλλέγει τα τοπικά ελάχιστα που προκύπτουν και επιστρέφει το ελάχιστο με την χαμηλότερη συναρτησιακή τιμή ως το καθολικό ελάχιστο. Το σημείο (ελάχιστο) στο οποίο καταλήγει μια διαδικασία τοπικής ελαχιστοποίησης L , εκκινώντας από ένα σημείο $x \in S$, αναφέρεται ως $L(x)$ με το S να συμβολίζει τον χώρο αναζήτησεως. Ο παραπάνω συμβολισμός διευκολύνει την περιγραφή των αλγορίθμων και χρησιμοποιείται συχνά στην παρούσα διατριβή. Πρέπει να σημειωθεί ότι σε όλες τις περιπτώσεις όπου χρειάστηκε να εφαρμοσθεί μια μέθοδος τοπικής ελαχιστοποίησης, χρησιμοποιήσαμε μια έκδοση της μεθόδου BFGS (TOLMIN) όπως προτάθηκε από τον M.J.D. Powell [98]. Τα βήματα που ακολουθούνται από την Multistart έχουν ως εξής:

Βήμα-0: Θέσε $X^* = \emptyset$. Το σύνολο X^* τελικά θα περιέχει όλα τα τοπικά ελάχιστα που θα ανακαλυφθούν από την μέθοδο.

Βήμα-1: Επιλογή με ομοιόμορφη δειγματοληψία ενός σημείου $x \in S$.

Βήμα-2: $x^* = L(x)$. (Εύρεση ενός τοπικού ελαχίστου x^*).

Βήμα-3: Αν $x \notin X^*$ Τότε

- Θέσε $X^* \leftarrow X^* \cup \{x^*\}$.

- Τέλος Αν

Βήμα-4: Αν το κριτήριο τερματισμού ισχύει διακοπή, αλλιώς μετάβαση στο **Βήμα-1**.

Η παραπάνω μέθοδος προσπαθεί από τυχαία αρχικά σημεία να βρει όλα τα τοπικά ελάχιστα που διαθέτει η συνάρτηση. Κάθε φορά που ανακαλύπτεται ένα νέο ελάχιστο, προστίθεται στο σύνολο X^* . Η μέθοδος δεν είναι αποδοτική, καθώς πολλές από τις τυχαίες εκκινήσεις που γίνονται οδηγούν σε ελάχιστα που έχουν ήδη βρεθεί. Οι άσκοπες αυτές εκκινήσεις θα μπορούσαν να είχαν αποφευχθεί αν υπήρχε κάποιος τρόπος για την αναγνώριση σημείων που ανήκουν στην ίδια περιοχή προσελκύσεως. **Περιοχή προσελκύσεως** (region of attraction) ενός τοπικού ελαχίστου είναι μία περιοχή του χώρου αναζητήσεως, τέτοια ώστε από οποιοδήποτε σημείο της και αν ξεκινήσει μια αιτιοκρατική μέθοδος τοπικής ελαχιστοποιήσεως, έστω L , θα καταλήξει στο ίδιο τοπικό ελάχιστο. Η περιοχή προσελκύσεως A_i του τοπικού ελαχίστου x_i^* ορίζεται ως:

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

Αν ο χώρος S περιέχει w τοπικά ελάχιστα, τότε από την εξίσωση (2.1) έπεται ότι:

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

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

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

Αν ένα σημείο $x \in S$ λαμβάνεται με δειγματοληψία βασισμένη στην ομοιόμορφη κατανομή, τότε η εκ των προτέρων πιθανότητα να ανήκει στην περιοχή A_i δίνεται από την σχέση:

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

Αν ληφθούν με δειγματοληψία K σημεία στον χώρο S , τότε η πιθανότητα ότι τουλάχιστον ένα από αυτά βρίσκεται στην περιοχή προσελκύσεως A_i , δίδεται από την σχέση:

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

Αυτό σημαίνει ότι καθώς το K μεγαλώνει η πιθανότητα να πάρουμε ένα σημείο που να οδηγήσει στο ελάχιστο i τείνει στο 1. Άρα η μέθοδος MULTISTART είναι σε θέση να ανακαλύψει όλα τα ελάχιστα στον χώρο S .

2.3 Τεχνικές ομαδοποιήσεως

Η τεχνική της ομαδοποιήσεως προσπαθεί να δημιουργήσει τις περιοχές προσελκύσεως εντάσσοντας σημεία τα οποία πιστεύεται ότι ανήκουν στην ίδια περιοχή προσελκύσεως σε ένα υποσύνολο. Το κέντρο αυτών των ομάδων ονομάζεται πυρήνας (seed point) και είναι το σημείο από το οποίο μπορεί να ξεκινήσει μία μέθοδος τοπικής ελαχιστοποίησης. Σε πρώτη φάση στην ομάδα μπαίνει μόνον ο πυρήνας και στην συνέχεια με διάφορα κριτήρια η ομάδα εμπλουτίζεται με περισσότερα σημεία μέχρι να ισχύσει κάποιο στοχαστικό (συνήθως) κριτήριο. Οι μέθοδοι ομαδοποιήσεως γενικά διαθέτουν δύο φάσεις: την ολική φάση και την τοπική φάση. Στην ολική φάση γίνεται η δειγματοληψία των σημείων και δημιουργούνται οι ομάδες και στην τοπική φάση γίνεται η εκκίνηση της μεθόδου τοπικής ελαχιστοποίησης από επιλεγμένα σημεία σύμφωνα με κάποια κριτήρια που θέτει η μέθοδος. Για να δημιουργηθούν οι ομάδες κάποιος μπορεί να ξεκινήσει με μια ομοιόμορφη δειγματοληψία στον χώρο αναζήτησεως και στην συνέχεια να εφαρμόσει είτε μια διαδικασία μείωσης (reduction) [42] είτε μια διαδικασία συγκεντρώσεως (concentration) [50]. Στην πρώτη τα σημεία με την υψηλότερη συναρτησιακή τιμή απομακρύνονται από το σύνολο και στην δεύτερη εφαρμόζονται μερικά βήματα μιας τεχνικής τοπικής ελαχιστοποίησης σε κάθε δείγμα με σκοπό αυτά τα δείγματα να μετακινήθούν προς το τοπικό ελάχιστο στο οποίο θα οδηγούσε μια πλήρης εφαρμογή της μεθόδου τοπικής ελαχιστοποίησης. Τέτοιες τεχνικές χρησιμοποιήθηκαν αρκετά σε μεθόδους ομαδοποιήσεως, όπως η Multilevel-Single Linkage (MLSL) [86] και η Topographical MLSL [83, 84]. Στην συνέχεια θα περιγραφούν οι τεχνικές MLSL και TMLSL καθώς και τα προβλήματα τους, μιας και αυτές οι μέθοδοι θα αποτελέσουν την βάση για την ανάπτυξη των δύο προτεινομένων τεχνικών ομαδοποιήσεως: της μεθόδου GTC και της μεθόδου MinFinder.

2.4 Η μέθοδος MLSL

Η μέθοδος Multilevel - Single Linkage Clustering (MLSL) [87] αποτέλεσε την βάση για την ανάπτυξη βελτιωμένων τεχνικών ομαδοποιήσεως. Ο αλγόριθμος προσπαθεί να βρει όλα τα τοπικά ελάχιστα στον χώρο αναζήτησεως εκκινώντας τοπική ελαχιστοποίηση από επιλεγμένα σημεία του χώρου. Μια περιγραφή του αλγορίθμου παρατίθεται στην συνέχεια:

Αρχικοποίηση Καθορίζεται το πλήθος M των σημείων που θα λαμβάνονται ομοιόμορφα τυχαία σε κάθε επανάληψη. Επίσης καθορίζεται ένας αριθμός $\gamma \in [0, 1]$, που ορίζει το πλήθος των σημείων που θα διατηρούνται στο δείγμα σε κάθε επανάληψη. Επίσης δημιουργείται το σύνολο X^* στο οποίο

θα αποθηκεύονται τα τοπικά ελάχιστα που θα ανακαλύπτονται σε κάθε επανάληψη. Αρχικά αυτό το σύνολο είναι κενό. Αρχικοποίηση του μετρητή επαναλήψεων: $k = 0$.

Βήμα-1 Επιλογή M σημείων με τυχαίο τρόπο και προσθήκη στο σύνολο δειγμάτων.

Βήμα-2 Δημιουργία ενός τροποποιημένου συνόλου δειγμάτων με την χρήση των $\gamma\%$ σημείων από το αρχικό δείγμα με την χαμηλότερη συναρτησιακή τιμή.

Βήμα-3 Εκκίνηση μια μεθόδου τοπικής ελαχιστοποίησης από κάθε σημείο x_i στο νέο δείγμα, εκτός και αν υπάρχει κάποιο άλλο σημείο στο δείγμα x_j τέτοιο ώστε $f(x_i) < f(x_j)$ και $\|x_i - x_j\| < r_k$. Προσθήκη των νέων τοπικών ελαχίστων που θα βρεθούν στο σύνολο X^* . Η απόσταση r_k δίνεται από:

$$r_k = \pi^{\frac{1}{2}} \left(\sigma m(S) \Gamma \left(1 + \frac{n}{2} \right) \frac{\log k}{k} \right)^{\frac{1}{n}} \quad (2.6)$$

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 και Γ είναι η συνάρτηση $\Gamma(n)$, που ορίζεται από τον τύπο:

$$\Gamma(n) = \int_0^{\infty} t^{n-1} \exp(-t) dt \quad (2.7)$$

Βήμα-4 . Η μέθοδος χρησιμοποιεί σαν μέθοδο τερματισμού το κριτήριο του Rinnooy Kan. Αυτό σημαίνει πως σε κάθε επανάληψη k , υπολογίζεται η ποσότητα

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

η οποία είναι μία εκτίμηση του συνολικού πλήθους των ελαχίστων. Η ποσότητα w είναι ο αριθμός των τοπικών ελαχίστων που μέχρι στιγμής έχουν βρεθεί. Ο αλγόριθμος τερματίζει όταν:

$$\hat{w} \leq w + \frac{1}{2}$$

Το σημείο y του συνόλου X^* με την χαμηλότερη συναρτησιακή τιμή θεωρείται ως το ολικό ελάχιστο της αντικειμενικής συναρτήσεως.

2.5 Η μέθοδος TMLSL

Ο αλγόριθμος TMLSL [83] βασίζεται στις ίδιες αρχές με τον αλγόριθμο MSL, αλλά προκειμένου να αποφασιστεί για το αν ένα σημείο πρέπει να ενταχθεί σε κάποια ομάδα κατασκευάζει έναν τοπολογικό γράφο και βρίσκει τα υπάρχοντα τοπογραφικά ελάχιστα (graph minima).

2.5.1. Ο αλγόριθμος

Τα βήματα του αλγορίθμου TMLSL έχουν ως ακολούθως:

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.

Δειγματοληψία

- $T = \emptyset$.
- $k = k + 1$.
- Ενημέρωση της κρίσιμης αποστάσεως r_k .
- **Για $i = 1, \dots, N$ Επανάλαβε**
 - Τυχαία δειγματοληψία ενός σημείου $x_i \in S$.
 - $f_i = f(x_i)$.
 - $T = T \cup (x_i, f_i)$
- **Τέλος Επαναλήψεων**

Δημιουργία_Γράφου

- Δημιουργία ενός τοπολογικού γράφου G με χρήση των σημείων από τα σύνολα T και X^* .
- Εύρεση των τοπογραφικών ελαχίστων στον γράφο G .
- Κάθε τοπογραφικό ελάχιστο θεωρείται σημείο ενάρξεως μεθόδου τοπικής ελαχιστοποίησης αν δεν υπάρχει άλλο σημείο σε απόσταση μικρότερη του r_k που να έχει χαμηλότερη συναρτησιακή τιμή από αυτό.

- Εκτέλεση μίας μεθόδου τοπικής ελαχιστοποίησης από κάθε σημείο που θεωρήθηκε σημείο ενάρξεως από το προηγούμενο βήμα. Προσθήκη των τοπικών ελαχίστων που θα βρεθούν στο σύνολο ελαχίστων X^* .

Έλεγχος

- Αν το κριτήριο τερματισμού ισχύει η διαδικασία τερματίζεται, αλλιώς επαναλαμβάνεται από το βήμα της δειγματοληψίας.

Ο υπολογισμός των τοπογραφικών ελαχίστων γίνεται ως ακολούθως: Για κάθε σημείο x βρίσκονται οι g κοντινότεροι γείτονές του με χρήση Ευκλείδειας απόστασης. Το σημείο θα θεωρηθεί τοπογραφικό ελάχιστο, αν και μόνον αν, έχει χαμηλότερη συναρτησιακή τιμή από κάθε γείτονα. Προφανώς η πολυπλοκότητα της παραπάνω διαδικασίας μεγαλώνει όσο το g μεγαλώνει και για αυτό στην πράξη δεν χρησιμοποιούνται περισσότεροι από 3 κοντινότεροι γείτονες. Αν το σημείο x έχει θεωρηθεί τοπογραφικό ελάχιστο θα θεωρηθεί σαν σημείο ενάρξεως της τοπικής μεθόδου ελαχιστοποίησης αν και μόνον αν δεν υπάρχει σε ακτίνα μικρότερη από r_k κάποιο άλλο σημείο y τέτοιο ώστε $f(y) \leq f(x)$. Η απόσταση r_k υπολογίζεται από την εξίσωση:

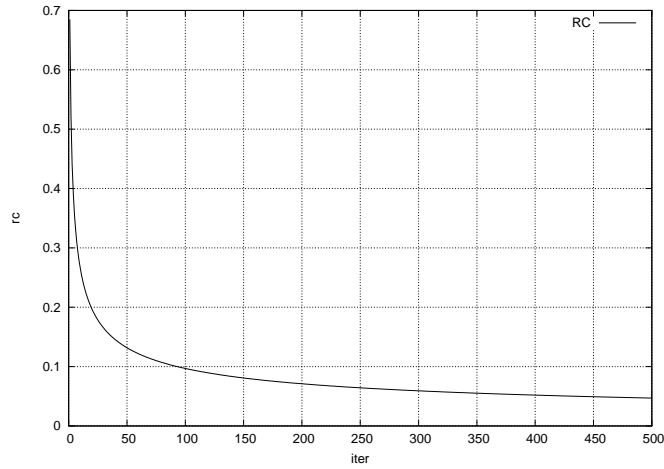
$$r_k = \pi^{-\frac{1}{2}} \left(\Gamma \left(1 + \frac{n}{2} \right) m(S) \sigma \frac{\log kN}{kN} \right)^{\frac{1}{n}} \quad (2.8)$$

Στην παραπάνω απόσταση χρησιμοποιούνται οι ίδιοι παράμετροι με την απόσταση της μεθόδου MLSL. Σαν κριτήριο τερματισμού η μέθοδος χρησιμοποιεί το ίδιο κριτήριο με την μέθοδο MLSL.

2.5.2. Το πλήθος των γειτόνων

Στην μέθοδο MLSL σε κάθε επανάληψη του αλγορίθμου εκτελείται μία διαδικασία συγκεντρώσεως του δείγματος προκειμένου να έρθουν τα σημεία πιο κοντά προς το τοπικό ελάχιστο στην περιοχή προσελκύσεώς τους. Αυτή η διαδικασία επιτυγχάνεται με δύο τεχνικές: είτε με την εκτέλεση μερικών βημάτων μίας μεθόδου τοπικής ελαχιστοποίησης είτε με την απόρριψη ενός ποσοστού από τα σημεία τα οποία έχουν μεγαλύτερη συναρτησιακή τιμή από τα υπόλοιπα. Στην TMLSL η διαδικασία της συγκεντρώσεως αντικαθίσταται από την διαδικασία της εύρεσης των τοπογραφικών ελαχίστων, καθώς έτσι από ομάδες g σημείων μόνον το καλύτερο θεωρείται σημείο εκκινήσεως. Ωστόσο η εύρεση των g γειτόνων σε ένα σύνολο N σημείων έχει πολυπλοκότητα gN^2 , ενώ η εύρεση των g σημείων με την χαμηλότερη τιμή από ένα σύνολο σημείων έχει πολυπλοκότητα $N \log N$. Ωστόσο αυτό δεν είναι τόσο μεγάλο κόστος διότι στον αλγόριθμο TMLSL διατηρούνται μόνον τα N δείγματα της τελευταίας επανάληψεως και τα ελάχιστα που

Σχήμα 2.1: Γράφημα της ποσότητας r_k ως προς k .



έχουν ήδη βρεθεί. Επιπλέον, σύμφωνα με τους Ali και Storey που ανέπτυξαν την TMLSL δεν υπάρχει κάποια στενή σύνδεση ανάμεσα στο g και στην διάσταση της συναρτήσεως ή τον αριθμό των ελαχίστων που διαθέτει η συνάρτηση.

2.5.3. Η κρίσιμη απόσταση

Η κρίσιμη απόσταση δίνεται από την έκφραση (2.8). Διατηρώντας τις παραμέτρους $N = 20$, $\sigma = 4$ σταθερές, το γράφημα που προκύπτει για την ποσότητα r_k σε σχέση με τον αριθμό των επαναλήψεων k αποτυπώνεται στο σχήμα 2.1. Ο Rinnooy Kan στην εργασία του για την MLSL [87] απέδειξε πως αν $\sigma > 2$, τότε ασυμπτωτικά θα βρεθούν όλα τα ελάχιστα της αντικειμενικής συναρτήσεως. Ωστόσο η ποσότητα r_k πολύ γρήγορα τείνει στο 0 καθώς το πλήθος k των επαναλήψεων αυξάνει. Έτσι από ένα σημείο και μετά η χρήση της αποστάσεως αυτής δεν θα έχει κάποια επίδραση στην απόρριψη σημείων από τα οποία δεν πρέπει να ξεκινήσει τοπική μέθοδος αναζήτησεως. Αυτό είναι περισσότερο έκδηλο στην MLSL επειδή η απόφαση για το αν ένα σημείο είναι σημείο εκκινήσεως αναθεωρείται σε κάθε βήμα. Έτσι σημεία που αρχικά θεωρήθηκαν ότι δεν αποτελούν σημεία εκκινήσεως σε επόμενο βήμα ίσως θεωρηθούν σημεία εκκινήσεως, καθώς η κρίσιμη απόσταση θα έχει μειωθεί αρκετά.

2.6 Η μέθοδος GTC (Ομαδοποίηση τυπικής αποστάσεως, ελεγχόμενη από την κλίση)

Η μέθοδος Gradient-controlled, Typical-Distance Clustering (GTC) αποτελεί μία εξέλιξη της μεθόδου Topographical Multilevel Single Linkage Clustering (TMLSL) που διατυπώθηκε από τους Ali και Storey [83] και βελτιώνει πολλές από τις αδυναμίες της. Η προτεινόμενη μέθοδος GTC (Ομαδοποίηση βασισμένη στην τυπική απόσταση και στην κλίση) αποφεύγει πολλά από τα προβλήματα της TMLSL θεωρώντας έναν νέο τύπο για απόσταση και ένα νέο κριτήριο ευρέσεως γειτόνων. Πριν παρουσιαστεί αναλυτικά ο προτεινόμενος αλγόριθμος, παρουσιάζονται δύο στοιχεία τα οποία διαδραματίζουν σπουδαίο ρόλο στην αποδοτικότητα του αλγορίθμου, η τυπική απόσταση και το κριτήριο της παραγώγου.

2.6.1. Η τυπική απόσταση

Η τυπική απόσταση υπολογίζεται από τον τύπο:

$$r_t = \frac{1}{M} \sum_{i=1}^M |x_i - L(x_i)| \quad (2.9)$$

όπου x_i είναι σημεία εκκινήσεως της μεθόδου τοπικής ελαχιστοποίησης L και M είναι ο αριθμός των τοπικών ελαχίστων που έχουν βρεθεί. Αυτή η απόσταση εξαρτάται από το πρόβλημα (σε αντίθεση με την απόσταση του τύπου (2.8) που διατυπώθηκε στις εργασίες [85, 86]) και ουσιαστικά είναι η μέση απόσταση των σημείων εκκινήσεως από το ελάχιστο που κατέληξαν. Δίνει μια εκτίμηση του μέσου όρου των ακτίνων των περιοχών προσελκύσεως και είναι εξαιρετικά απλή στην υλοποίησή της. Για να φανεί αυτό πιο καθαρά, ας θεωρήσουμε την περίπτωση ότι έχουν βρεθεί μέχρι στιγμής w ελάχιστα και ας συμβολίσουμε με M_i το πλήθος των φορών που έχει βρεθεί το ελάχιστο x_i^* και έστω R_i η ακτίνα:

$$R_i = \frac{1}{M_i} \sum_{j=1}^{M_i} |x_i^{(j)} - x_i^*| \quad (2.10)$$

όπου $\{x_i^{(j)}, j = 1, \dots, M_i\}$ τα σημεία εκκινήσεως που οδήγησαν στο ελάχιστο x_i^* . Επειδή $M = \sum_{i=1}^w M_i$, έπεται πως η μέση ακτίνα για όλες τις περιοχές προσελκύσεως δίνεται από:

$$\langle R \rangle \equiv \sum_{i=1}^w \frac{M_i}{M} R_i = \frac{1}{M} \sum_{i=1}^w \sum_{j=1}^{M_i} |x_i^{(j)} - x_i^*| \quad (2.11)$$

Συγκρίνοντας τις εξισώσεις (2.9), (2.10) και (2.11) έπεται ότι $r_t = \langle R \rangle$.

Ίσως να ήταν προτιμότερο να μην χρησιμοποιείται μόνον μια τυπική απόσταση, αλλά πολλές τυπικές αποστάσεις μία για κάθε ελάχιστο. Με αυτόν τον τρόπο θα παρέχεται μια λεπτομερέστερη εικόνα των περιοχών προσελκύσεως, καθώς κάθε μια από αυτές είναι διαφορετική ως προς το εύρος της. Το σημείο αυτό χρήζει περαιτέρω μελέτης και αναζητήσεως.

2.6.2. Το κριτήριο της κλίσεως

Ας υποθέσουμε πως υπάρχει έναν τοπικό ελάχιστο x^* . Τότε χρησιμοποιώντας ανάπτυγμα Taylor σε μια μικρή ακτίνα γύρω από το τοπικό ελάχιστο η αντικειμενική συνάρτηση μπορεί να προσεγγιστεί ως:

$$f(x) \simeq f(x^*) + \frac{1}{2} (x - x^*)^T B^* (x - x^*) \quad (2.12)$$

όπου B^* είναι ο Εσσιανός πίνακας στο τοπικό ελάχιστο. Παραγωγίζοντας την εξίσωση (2.12) έπεται ότι

$$\nabla f(x) \simeq B^* (x - x^*) \quad (2.13)$$

Ομοίως για οποιοδήποτε άλλο σημείο y κοντά στο τοπικό ελάχιστο x^* θα ισχύει:

$$\nabla f(y) \simeq B^* (y - x^*) \quad (2.14)$$

Αφαιρώντας τις σχέσεις (2.13) και (2.14) κατά μέλη και πολλαπλασιάζοντας με $(x - y)^T$ συνάγεται ότι:

$$(x - y)^T (\nabla f(x) - \nabla f(y)) \simeq (x - y)^T B^* (x - y) > 0 \quad (2.15)$$

μιας και ο πίνακας B^* είναι θετικά ορισμένος. Άρα εάν δύο γειτονικά σημεία x και y είναι στην περιοχή προσελκύσεως του ίδιου ελαχίστου, τότε ισχύει η σχέση (2.15). Η παραπάνω συνθήκη ονομάζεται και κριτήριο της κλίσεως (gradient test).

2.6.3. Ο προτεινόμενος αλγόριθμος

Ο προτεινόμενος αλγόριθμος εντάσσεται στην κατηγορία των τεχνικών ομαδοποίησης, αφού προσπαθεί να εντοπίσει τις περιοχές προσελκύσεως. Έχει ως στόχο να εντοπίσει όλα τα τοπικά ελάχιστα της αντικειμενικής συναρτήσεως, δαπανώντας τον μικρότερο αριθμό συναρτησιακών αποτιμήσεων. Η μέθοδος διαθέτει μια σειρά από παραμέτρους και δομές που περιγράφονται αναλυτικά στην συνέχεια.

Παράμετροι

1. M . Είναι το πλήθος των σημείων που δειγματοληπτούνται σε κάθε επανάληψη αλγορίθμου.
2. K . Είναι ένας ακέραιος με $K \leq M - 1$. Καθορίζει το πλήθος των πλησιέστερων γειτόνων που πρέπει να εντοπίζονται για κάθε σημείο.

Χρήσιμες δομές

1. X^* : Είναι το σύνολο όλων των τοπικών ελαχίστων που εντοπίστηκαν από τον αλγόριθμο.
2. RC. Είναι η τυπική απόσταση όπως υπολογίζεται από την σχέση (2.9).
3. MAXRC. Είναι η μέγιστη που λαμβάνει κατά την διάρκεια του αλγορίθμου η τιμή της τυπικής αποστάσεως.
4. $\text{grad}(x, y)$ Είναι μια συνάρτηση που δέχεται σαν ορίσματα δύο σημεία στον χώρο S και επιστρέφει αληθές αν για αυτά ισχύει το κριτήριο της παραγωγού και ψευδές σε άλλη περίπτωση.
5. LS. Είναι ένας μετρητής που καταγράφει πόσες φορές κλήθηκε η τοπική μέθοδος ελαχιστοποίησης.
6. TD. Είναι ένας μετρητής ο οποίος καταγράφει το άθροισμα αποστάσεων ανάμεσα στα σημεία εκκινήσεως της τοπικής μεθόδου ελαχιστοποίησης και στα σημεία καταλήξεως της μεθόδου τοπικής ελαχιστοποίησης.

Βήματα

1. Αρχικά $LS=0$, $TD=0$, $RC=0$, $MAXRC=0$, $X^* = \emptyset$
2. Δειγματοληπτούνται M νέα σημεία και αυτά αποτελούν το σύνολο Z .
3. $Q = Z \cup X^*$
4. $\forall x \in Z$ **Επανέλαβε**
 - (α) Έστω P το σύνολο των K κοντινότερων γειτόνων του x από το σύνολο Q .
 - (β) $\forall y \in P$ **Επανέλαβε**

- i. Αν $\|x - y\| \leq RC$ και $\text{grad}(x, y)$ και $\exists w \in X^*$ τέτοιο ώστε $\|x - w\| \leq \text{MAXRC}$ και $\|y - w\| \leq \text{MAXRC}$ και $\text{grad}(x, w)$ και $\text{grad}(y, w)$, τότε το σημείο x δεν θεωρείται σημείο εκκινήσεως τοπικής μεθόδου ελαχιστοποίησης. Δηλαδή αν τα σημεία x και y υποτίθεται πως βρίσκονται στην περιοχή προσελκύσεως κάποιου τοπικού ελαχίστου, τότε το σημείο x δεν μπορεί να θεωρηθεί σημείο εκκινήσεως μεθόδου τοπικής ελαχιστοποίησης.
- (γ) Αν x είναι σημείο εκκινήσεως τότε
- i. Έστω $(x', f(x'))$ το σημείο που προκύπτει από την εφαρμογή της τοπικής μεθόδου στο x .
 - ii. $LS = LS + 1$.
 - iii. $TD = TD + \|x - x'\|$.
 - iv. $RC = TD / LS$.
 - v. $\text{MAXRC} = \max(RC, \text{MAXRC})$.
 - vi. Αν $x' \notin X^*$ τότε $X^* = X^* \cup x'$.

5. Αν δεν ικανοποιείται το **κριτήριο τερματισμού**, μετάβαση στο βήμα 2. Το κριτήριο τερματισμού που χρησιμοποιεί η μέθοδος είναι αυτό του διπλασίου κουτιού, που θα παρουσιαστεί σε επόμενη ενότητα αυτού του κεφαλαίου.

2.6.4. Πειραματικά αποτελέσματα

Στον πίνακα 2.1 παρουσιάζονται συγκριτικά αποτελέσματα ανάμεσα στις μεθόδους MULTISTART, TMLSL και GTC για την σειρά συναρτήσεων που παρουσιάστηκε και στους προηγούμενους πίνακες. Στον πίνακα έχει προστεθεί η στήλη GC που αναπαριστά τον αριθμό των κλήσεων προς την παράγωγο της συναρτήσεως. Αυτή η πληροφορία είναι χρήσιμη για την εξαγωγή συμπερασμάτων, αφού η μέθοδος GTC χρησιμοποιεί επιπλέον κλήσεις προς την παράγωγο προκειμένου να αποφύγει να εκτελέσει άσκοπες κλήσεις προς την τοπική μέθοδο ελαχιστοποίησης.

Από τα πειράματα φαίνεται πως η μέθοδος GTC δεν διαθέτει τις αδυναμίες αυτών των άλλων τεχνικών ομαδοποιήσεως σε ότι αφορά την απόρριψη σημείων. Έτσι με την τυπική απόσταση που χρησιμοποιεί αλλά και με το κριτήριο της παραγώγου εμφανίζεται να είναι πιο αποδοτική από αυτές και επιπλέον να απαιτεί μικρότερο χρόνο αλλά και μικρότερο αριθμό συναρτησιακών αποτιμήσεων.

Πίνακας 2.1: Συγκριτικά αποτελέσματα MULTISTART, TMLSL και GTC σε μια σειρά από συναρτήσεις δοκιμής με την χρήση του κριτηρίου τερματισμού του διπλού κουτιού.

ΣΥΝΑΡΤΗΣΗ	MULTISTART			TMLSL			GTC		
	MIN	FC	GC	MIN	FC	GC	MIN	FC	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	1032445	1140113
GKLS(3,30)	29	302853	302853	28	191867	102373	28	75363	206788
GKLS(3,100)	97	7492103	7492103	91	6683608	4583920	97	4160897	5663373
GKLS(4,100)	95	8629052	8629052	78	12269342	8237461	96	2564480	3878680
GUILIN(10,200)	200	3351391	3351391	200	3470206	3234820	200	2084425	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

2.7 Η μέθοδος MinFinder

Σε αυτήν την ενότητα περιγράφεται η μέθοδος MinFinder, που αποτελεί μια εξέλιξη του αλγορίθμου GTC. Η νέα μέθοδος δεν χρησιμοποιεί την έννοια των γειτόνων όπως παρουσιάστηκε στην GTC γεγονός που την καθιστά ταχύτερη. Επιπλέον η νέα μέθοδος χρησιμοποιεί δύο φάσεις προκειμένου να αποφασίσει αν ένα σημείο μπορεί να είναι σημείο εκκινήσεως τοπικής ελαχιστοποίησης. Στην πρώτη φάση ελέγχεται αν το σημείο βρίσκεται (με κάποιο προσεγγιστικό κριτήριο) στην περιοχή προσελκύσεως κάποιου ήδη ευρεθέντος ελαχίστου. Αν δεν ισχύει η παραπάνω υπόθεση το σημείο ελέγχεται αν βρίσκεται στην ίδια περιοχή προσελκύσεως με κάποιο από τα άλλα σημεία του δείγματος. Αν καμία από τις παραπάνω υποθέσεις δεν ισχύει, το σημείο μπορεί να θεωρηθεί ως σημείο ενάρξεως τοπικής ελαχιστοποίησης. Σε κάθε μια από τις δύο φάσεις ελέγχου χρησιμοποιούνται διαφορετικές τυπικές αποστάσεις αντί της μιας που χρησιμοποιήθηκε στην μέθοδο GTC. Η πρώτη τυπική απόσταση εξαρτάται από τα ελάχιστα που έχουν βρεθεί και η δεύτερη είναι η τυπική απόσταση που χρησιμοποιήθηκε στην μέθοδο GTC. Και στην νέα μέθοδο κυρίαρχο ρόλο παίζει το κριτήριο της παραγώγου (gradient test), όπως αυτό παρουσιάστηκε στην μέθοδο GTC.

2.7.1. Περιγραφή του αλγορίθμου

Τα βήματα του προτεινόμενου αλγορίθμου έχουν ως ακολούθως:

Αρχικοποίηση:

- Καθορισμός της παραμέτρου N , αριθμός των σημείων που δειγματοληπτούνται 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 σημεία που δειγματοληπτούνται στην αρχή κάθε επαναλήψεως.
- $V = \emptyset$, όπου V περιέχει σε κάθε επανάληψη τα υποψήφια σημεία εκκινήσεως τοπικής αναζήτησεως.

Βήμα 1:

- $T = \emptyset, V = \emptyset$.
- **Δειγματοληψία** N σημείων από τον χώρο αναζήτησεως με τυχαίο τρόπο και με την χρήση του σχήματος του διπλασίου κουτιού που θα παρουσιασθεί σε επόμενη ενότητα. Προσθήκη αυτών των σημείων στο σύνολο T .

- Για Κάθε $x \in T$ Επανάλαβε
 - Αν x δεν είναι απορριπτό **Τότε**
 - * Θέσε $V = V \cup x$.
 - Τέλος Αν
- Τέλος Επαναλήψεως

Βήμα 2:

- 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 προκειμένου να αναγκαστεί ο αλγόριθμος να κάνει καλύτερη έρευνα του χώρου S , γιατί όπως παρατηρήθηκε πειραματικά ο αλγόριθμος τείνει να απορρίψει πολλά από τα δείγματα που λαμβάνει σε κάθε επανάληψη, στερώντας έτσι την δυνατότητα να κάνει καλύτερη έρευνα.
- Τέλος Αν

Βήμα 3:

- Για Κάθε $x \in V$ Επανάλαβε
 - Αν x δεν θεωρείται απορριπτό σημείο **Τότε**
 - * **Εκκίνηση** μιας μεθόδου τοπικής αναζήτησεως και έστω $y = L(x)$ το ευρεθέν ελάχιστο.
 - * **Υπολογισμός** της τυπικής αποστάσεως r_i κάνοντας χρήση της εξισώσεως (2.9).
 - * Αν $y \notin X^*$ **Τότε**
 - Θέσε $X^* = X^* \cup y$.
 - * Τέλος Αν
 - Τέλος Αν
- Τέλος Επαναλήψεως
- Αν το κριτήριο τερματισμού του διπλασίου κουτιού ικανοποιείται **τότε**
 - Διακοπή του αλγορίθμου.
- Αλλιώς

- **Μετάβαση στο Βήμα 1.**

Στον παραπάνω αλγόριθμο για να θεωρηθεί ένα σημείο x σαν απορριπτό θα πρέπει τουλάχιστον μια από τις επόμενες συνθήκες να ισχύει:

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

$$(x - z)^T (\nabla f(x) - \nabla f(z)) > 0 \quad (2.16)$$

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

$$|x - z| < \min_{i,j, i \neq j} |z_i - z_j|, \quad z_i \in X^*, z_j \in X^* \quad (2.17)$$

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

$$|x - y| < r_t \quad (2.18)$$

2.7.2. Πειραματικά αποτελέσματα

Στον πίνακα 2.2 παρουσιάζονται τα αποτελέσματα από την εφαρμογή της μεθόδου MinFinder στα πειράματα που έγινε και προηγουμένως η εφαρμογή των άλλων μεθόδων αυτού του κεφαλαίου. Σχεδόν σε όλες τις περιπτώσεις η μέθοδος MinFinder είναι περισσότερο αποτελεσματική από την TMLSL και σε αρκετές περιπτώσεις βελτιώνει την απόδοση της μεθόδου GTC. Η νέα τεχνική είναι ταχύτερη και περισσότερο αποτελεσματική από τις άλλες μεθόδους και είναι περισσότερο απλή στην υλοποίηση από ότι η GTC. Η διαφορά στα αποτελέσματα που παρουσιάζονται ανάμεσα στην GTC και την νέα μέθοδο MinFinder, αποδεικνύουν πως στο πεδίο της μεθόδων καθολικής ελαχιστοποίησης υπάρχει αρκετός χώρος για περαιτέρω έρευνα και ανάπτυξη νέων και αποτελεσματικότερων τεχνικών στο μέλλον.

2.8 Υλοποιημένο λογισμικό

Για την μέθοδο ομαδοποίησης MinFinder δημιουργήθηκε ένα συνοδευτικό λογισμικό σε ANSI C++, στο οποίο ο χρήστης μπορεί να κωδικοποιήσει την αντικειμενική συνάρτηση είτε σε C++ είτε σε FORTRAN77. Σε αυτό το λογισμικό ο χρήστης παρέχει την αντικειμενική συνάρτηση και την παράγωγό της σε ένα

Πίνακας 2.2: Αποτελέσματα της μεθόδου MINFINDER σε μια σειρά από συναρτήσεις δοκιμής με την 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

αρχείο ορίζοντας μια σειρά από συναρτήσεις και μέσα από ένα βοηθητικό προγραμματιστικό εργαλείο που παρέχεται από το λογισμικό, δημιουργείται το τελικό εκτελέσιμο αρχείο. Το τελικό εκτελέσιμο αρχείο διαθέτει μια σειρά από παραμέτρους, που επιτρέπουν στον χρήστη να πειραματιστεί. Στο παράρτημα Β περιγράφονται οι διαδικασίες εγκατάστασης και χρήσεως του προγραμματιστικού εργαλείου και δίνεται ένα παράδειγμα χρήσεως.

2.9 Κανόνες τερματισμού

Ένα από τα κυριότερα θέματα των στοχαστικών μεθόδων είναι τα κριτήρια τερματισμού τους. Επειδή οι μέθοδοι αυτής της κατηγορίας ανακαλύπτουν το καθολικό ελάχιστο με κάποια πιθανότητα, μετά από έναν πεπερασμένο αριθμό επαναλήψεων δεν είναι εγγυημένη η επιτυχία της εφαρμογής τους. Αυτό σημαίνει ότι εφόσον σε πρακτικές εφαρμογές το ασυμπτωτικό όριο (άπειρες επαναλήψεις) δεν είναι εφικτό, θα πρέπει να τερματιστεί η διαδικασία πρώιμα, διακινδυνεύοντας ωστόσο την πλήρη επιτυχία της.

Έχουν αναπτυχθεί διάφοροι “κανόνες τερματισμού” που στοχεύουν στο να διατηρούν μεγάλη πιθανότητα επιτυχούς αποτελέσματος και ταυτόχρονα να μην απαιτούν υψηλό υπολογιστικό κόστος. Στην βιβλιογραφία έχουν εμφανιστεί αρκετοί κανόνες όπως των Zieliński [77], Betro et al [78], Rinnooy Kan [80], Hart [82]. Μια βιβλιογραφική επισκόπηση των κανόνων τερματισμού δίνεται στο [81].

Στην επόμενη παράγραφο γίνεται μια σύντομη αναφορά στα κριτήρια των Boender, Rinnooy Kan και Zieliński και στην συνέχεια περιγράφουμε τρεις νέους κανόνες που αναπτύχθηκαν στα πλαίσια της παρούσας διατριβής, ήτοι τον κανόνα του “διπλασίου κουτιού”, τον κανόνα “των παρατηρήσιμων μεγεθών” και τέλος “των αναμενόμενων ελαχίστων”.

2.9.1. Τα κριτήρια των Rinnooy Kan, Boender και Zieliński.

Σύμφωνα με το κριτήριο τερματισμού που διατυπώθηκε από τον Rinnooy Kan [80], αν για την εύρεση w ελαχίστων έχουν γίνει t τοπικές αναζητήσεις, τότε μια εκτίμηση για το πλήθος των τοπικών ελαχίστων της συναρτήσεως δίνεται από τον τύπο:

$$\hat{w} = \frac{w(t-1)}{t-w-2} \quad (2.19)$$

Σε πολυδειγματικές μεθόδους καθολικής ελαχιστοποίησης όπως η Topographical Multilevel Single Linkage Clustering [83], το t στην εξίσωση (2.19) αντικαθίσταται από το σύνολο των δειγματοληψιών που έχει πραγματοποιήσει ο αλγό-

ριθμός. Ο αλγόριθμος καθολικής ελαχιστοποίησης διακόπτεται όταν:

$$\hat{w} - w < \frac{1}{2} \quad (2.20)$$

Ωστόσο κάνοντας πράξεις στην ανισότητα (2.20) διαπιστώνεται πως πρέπει να ισχύει ότι:

$$t > 2w^2 + 3w + 2 \quad (2.21)$$

Με άλλα λόγια θα πρέπει ο αριθμός των τοπικών ελαχιστοποιήσεων (ή δειγμάτων σε άλλες περιπτώσεις) να είναι ανάλογος του τετραγώνου των ελαχίστων που έχουν βρεθεί, προκειμένου να σταματήσει ο αλγόριθμος. Αν έχουν βρεθεί πχ. 1000 ελάχιστα θα απαιτηθούν 10^6 και πλέον ελαχιστοποιήσεις προκειμένου να τερματιστεί ο αλγόριθμος, ακόμα και αν δεν βρεθούν επιπλέον ελάχιστα κατά την διάρκεια των πρόσθετων τοπικών ελαχιστοποιήσεων. Ο κανόνας αυτός δεν είναι ιδιαίτερα αποδοτικός για συναρτήσεις με πολλά ελάχιστα και επίσης δεν εκμεταλλεύεται την γνώση που αποκτάται από την δειγματοληψία. Ο Zieliński [77] εκτιμά το ποσοστό του χώρου αναζητήσεως που δεν έχει καλυφτεί με την δειγματοληψία ως:

$$P(w) = \frac{w(w+1)}{t(t-1)} \quad (2.22)$$

Σε αυτήν την περίπτωση η μέθοδος καθολικής ελαχιστοποίησης διακόπτεται όταν:

$$P(w) \leq \epsilon \quad (2.23)$$

όπου ϵ ένας μικρός θετικός αριθμός, πχ. $\epsilon = 10^{-4}$.

2.9.2. Ο κανόνας του διπλασίου κουτιού

Στα πειράματα που πραγματοποιούνται είναι ανάγκη να γνωρίζουμε το τμήμα του χώρου αναζητήσεως που δεν έχει καλυφτεί από τις τοπικές ελαχιστοποιήσεις που έχουν πραγματοποιηθεί. Αν το τμήμα αυτό είναι μικρό, τότε η μέθοδος καθολικής ελαχιστοποίησης μπορεί να τερματιστεί. Το ποσοστό της περιοχής αναζητήσεως που δεν έχει καλυφθεί δίνεται από την ποσότητα:

$$U = 1 - \sum_{i=1}^w \frac{m(A_i)}{m(S)} \quad (2.24)$$

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

της τοπικής ελαχιστοποιήσεως. Κατά συνέπεια μια εκτίμηση της ποσότητας U δίνεται από τον τύπο:

$$\hat{U} = 1 - \sum_{i=1}^w \frac{t_i}{T} \quad (2.25)$$

Δυστυχώς σε κάθε επανάληψη η ποσότητα \hat{U} έχει την τιμή 0 μιας και $\sum_{i=1}^w t_i = T$ και σαν συνέπεια η εξίσωση (2.25) δεν μπορεί να χρησιμοποιηθεί για την προσέγγιση του τμήματος του χώρου αναζητήσεως που δεν έχει ακόμα καλυφθεί. Για αυτόν τον σκοπό κατασκευάζεται ένας μεγαλύτερος χώρος αναζητήσεως S_2 που περιέχει τον αρχικό χώρο S , έτσι ώστε να ισχύει $m(S_2) = 2 \times m(S)$. Επίσης ένα μοναδικό (και ψεύτικο) τοπικό ελάχιστο θεωρείται πως υπάρχει στην περιοχή $A_0 = S_2 - S$. Ο όγκος της περιοχής προσελκύσεως για αυτό το ελάχιστο είναι $m(A_0) = m(S)$.

$$\begin{aligned} U &= 1 - \sum_{i=1}^w \frac{m(A_i)}{m(S)} \Rightarrow \\ U &= 1 - \sum_{i=1}^w \frac{m(A_i)}{\frac{1}{2}m(S_2)} \Rightarrow \\ U &= 1 - \frac{\sum_{i=1}^w 2m(A_i)}{m(S_2)} \Rightarrow \\ U &= 1 - 2 \frac{\sum_{i=1}^w m(A_i)}{m(S_2)} \end{aligned} \quad (2.26)$$

Και στην εξίσωση (2.26) η ποσότητα $\frac{m(A_i)}{m(S_2)}$ προσεγγίζεται από την ποσότητα $\frac{t_i}{T}$, όπου η ποσότητα T ο συνολικός αριθμός εφαρμογών της μεθόδου τοπικής αναζητήσεως στον χώρο S_2 . Έτσι η ποσότητα \hat{U} δίνεται από την ακόλουθη εξίσωση:

$$\hat{U} = 1 - 2 \sum_{i=1}^w \frac{t_i}{T} \quad (2.27)$$

Σε κάθε επανάληψη N σημεία συλλέγονται ομοιόμορφα τυχαία στον χώρο S_2 και απορρίπτονται σημεία που δεν ανήκουν στον χώρο S . Έστω ότι μετά από k επαναλήψεις ο συνολικός αριθμός των σημείων που έχουν ληφθεί στον χώρο S_2 είναι M_k και από αυτά τα kN βρίσκονται στον 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

$$\sigma_k^2(\delta) = \langle \delta^2 \rangle_k - \langle \delta \rangle_k^2 \quad (2.28)$$

η οποία τείνει στο 0 καθώς το k τείνει στο άπειρο. Όπως μπορούμε να δούμε και στο σχήμα 2.2 αυτή η ποσότητα είναι περισσότερο ομαλά μεταβαλλόμενη

από την μέση τιμή και επομένως πιο κατάλληλη για να χρησιμοποιηθεί για τον σχηματισμό ενός κριτηρίου τερματισμού. Η καθολική μέθοδος ελαχιστοποίησης τερματίζεται όταν ισχύσουν οι συνθήκες:

$$\left| \langle \delta \rangle_k - \frac{1}{2} \right| \leq \sigma_k(\delta), \quad \sigma_k(\delta) \leq p\sigma_{\text{last}}^2(\delta) \quad (2.29)$$

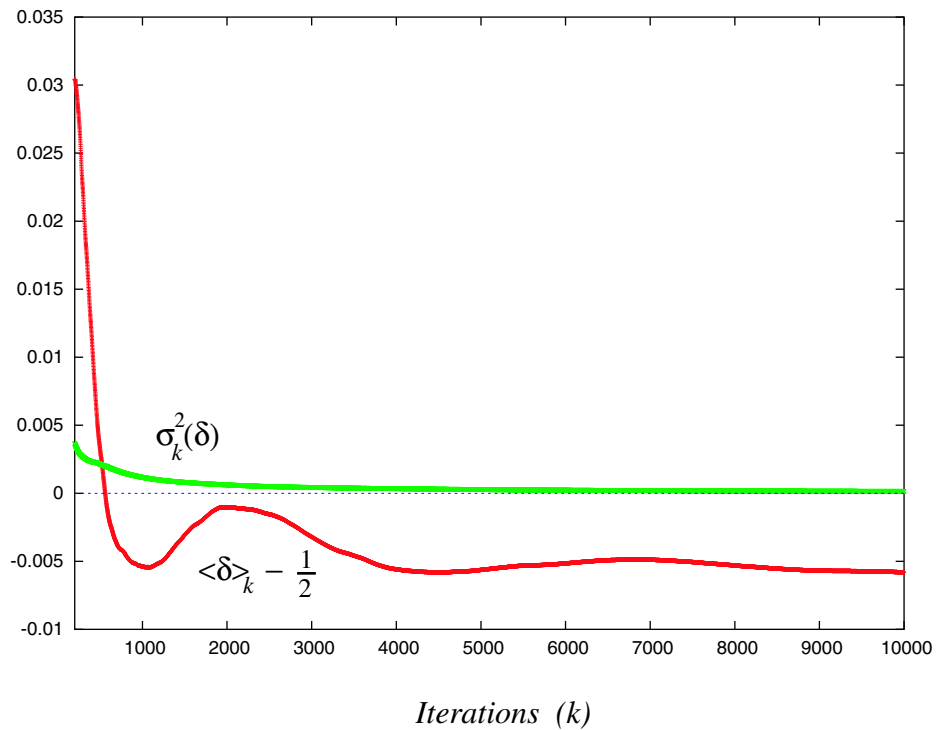
όπου $\sigma_{\text{last}}^2(\delta)$ η τιμή της διακυμάνσεως της ποσότητας δ_k στην επανάληψη που βρέθηκε το τελευταίο τοπικό ελάχιστο και p είναι μια ποσότητα στο διάστημα $(0, 1)$ και χρησιμοποιείται σαν ένας συμβιβασμός ανάμεσα σε μια εξαντλητική αναζήτηση ($p \rightarrow 0$) και σε μια αναζήτηση βελτιστοποιημένη για ταχύτητα ($p \rightarrow 1$). Στον πίνακα 2.3 μπορούμε να δούμε πειράματα που έγιναν από την εφαρμογή του κριτηρίου του διπλασίου κουτιού με την χρήση της μεθόδου MULTISTART σε μια σειρά από συναρτήσεις δοκιμής και για διαφορετικές τιμές της παραμέτρου p . Εκτελέστηκαν 100 πειράματα για κάθε συνάρτηση με διαφορετική αρχικοποίηση της γεννήτριας τυχαίων αριθμών σε κάθε εκτέλεση. Η στήλη MIN περιέχει τον μέσο όρο των ελαχίστων που βρέθηκαν και η στήλη FC περιέχει τον μέσο αριθμό συναρτησιακών κλήσεων που χρειάστηκαν για αυτόν τον σκοπό. Όπως βλέπουμε καθώς η ποσότητα p αυξάνει η μέθοδος τερματίζει ταχύτερα αλλά ενίοτε μερικά τοπικά ελάχιστα (και ίσως και το καθολικό) μπορεί να χαθούν. Η προτεινόμενη τιμή για αυτήν την παράμετρο (σαν συμβιβασμός ανάμεσα στις δύο ακραίες περιπτώσεις) είναι η τιμή 0.5. Ο αλγόριθμος έχει ως εξής:

1. Αρχικά θέτουμε $a = 0$.
2. Δειγματοληψία N σημείων με την διαδικασία που διατυπώθηκε προηγουμένως.
3. Υπολογισμός της ποσότητας $\sigma^2(\delta)$.
4. Εκτέλεση μιας επαναλήψεως της διαδικασίας καθολικής ελαχιστοποίησης χρησιμοποιώντας τα σημεία που ελήφθησαν προηγουμένως. Ανάλογα με την χρησιμοποιούμενη μέθοδο (Multistart, TMLSL κτλ.) αυτά τα σημεία μπορούν να χρησιμοποιηθούν σαν σημεία εκκινήσεως μεθόδου τοπικής ελαχιστοποίησης ή να εισαχθούν σε κάποια ομάδα.
5. Αν βρεθεί τουλάχιστον ένα νέο ελάχιστο, θέσε: $a = p\sigma^2(\delta)$ και μετάβαση στο βήμα 2.
6. Τερματισμός αν $\left| \langle \delta \rangle_k - \frac{1}{2} \right| \leq \sigma_k(\delta)$ και $\sigma^2(\delta) \leq a$, διαφορετικά μετάβαση στο βήμα 2.

Πίνακας 2.3: Πειράματα με την ποσότητα p στην μέθοδο του διπλασίου κουτιού και με την χρήση της μεθόδου MULTISTART σε μια σειρά από συναρτήσεις δοκιμής.

ΣΥΝΑΡΤΗΣΗ	$p = 0.3$		$p = 0.5$		$p = 0.7$		$p = 0.9$	
	MIN	FC	MIN	FC	MIN	FC	MIN	FC
SHUBERT	400	1150243	400	577738	400	322447	395	139768
GKLS(3,30)	30	961269	29	302583	23	41026	15	3920
RASTRIGIN	49	50384	49	19593	49	13581	49	10034
Test2N(5)	32	78090	32	30607	32	20870	32	13462
Test2N(6)	64	85380	64	34840	64	22535	64	15393
Guilin(20,100)	100	3405112	100	1906288	100	854511	71	79331
Shekel10	10	93666	10	36838	10	23870	10	15976

Σχήμα 2.2: Γράφημα της μέσης τιμής της ποσότητας δ_k και της διακυμάνσεως αυτής.



2.9.3. Ο κανόνας των παρατηρησίμων μεγεθών

Ο δεύτερος κανόνας τερματισμού είναι ο κανόνας των παρατηρήσιμων μεγεθών. Ο κανόνας αυτός βασίζεται στην σύγκλιση των αναμενόμενων τιμών ορισμένων μεγεθών που υπολογίζονται κατά την διάρκεια της διαδικασίας, στις ασυμπτωτικές τους τιμές. Γίνεται μια σύγκριση ανάμεσα στην ποσότητα t_i (το πλήθος των ελαχιστοποιήσεων που οδήγησαν στο ελάχιστο x_i^*) και στην αναμενόμενη τιμή για αυτή την ποσότητα.

Έστω πως έχουν βρεθεί w τοπικά ελάχιστα και από τις συνολικά T επαναλήψεις t_1 φορές βρέθηκε το ελάχιστο 1, t_2 το ελάχιστο 2 κ.ό.κ. Προφανώς ισχύει ότι $t_i \geq 1$, $\forall i = 1 \dots w$. Επίσης συμβολίζουμε με n_J το πλήθος των επαναλήψεων που μεσολάβησαν για την εύρεση του J ελαχίστου από το προηγούμενο. Με βάση τα παραπάνω και λαμβάνοντας υπόψη ότι χρησιμοποιείται ομοιόμορφη δειγματοληψία ο αναμενόμενος αριθμός $L_J^{(w)}$ των τοπικών αναζητήσεων που οδηγήθηκαν στο ελάχιστο J την στιγμή που εντοπίζεται για πρώτη φορά το ελάχιστο w είναι:

$$L_J^{(w)} = L_J^{(w-1)} + (n_w - 1) \frac{m(A_J)}{m(S)} \quad (2.30)$$

όπου $m(A)$ είναι το μέτρο Lebesgue του συνόλου A . Είναι προφανές πως μια τέτοια ποσότητα δεν είναι δυνατόν να εκτιμηθεί αναλυτικά, αλλά μπορεί να εκτιμηθεί ασυμπτωτικά με την χρήση των ποσοτήτων L_i . Καθώς το πλήθος των τοπικών αναζητήσεων αυξάνει θα ισχύει ότι:

$$\frac{t_i}{\sum_{j=1}^w t_j} \simeq \frac{m(A_i)}{m(S)} \quad (2.31)$$

Με απλή αντικατάσταση στον τύπο 2.30 προκύπτει ότι:

$$\begin{aligned} L_J^{(w)} &= L_J^{(w-1)} + (n_w - 1) \frac{t_J}{\sum_{i=1}^{w-1} t_i} \\ &= L_J^{(w-1)} + (n_w - 1) \frac{t_J}{\sum_{i=1}^{w-1} n_i} \end{aligned} \quad (2.32)$$

και ισχύει ότι $n_1 = 1$, $J \leq w - 1$ και $L_w^{(w)} = 1$. Ο τύπος 2.32 ισχύει όταν έχει βρεθεί το w ελάχιστο για πρώτη φορά. Ωστόσο μετά την εύρεση αυτού του ελαχίστου είναι πιθανόν να έχουν γίνει και άλλες τοπικές αναζητήσεις που να είναι άκαρπες, έστω K . Σε αυτήν την περίπτωση ο αναμενόμενος αριθμός εμφανίσεων του ελαχίστου J δίνεται από:

$$\mathcal{L}_J^{(w)}(K) = \mathcal{L}_J^{(w)}(K - 1) + \frac{t_J}{K + \sum_{i=1}^w n_i} \quad (2.33)$$

με την ιδιότητα ότι $\mathcal{L}_J^{(w)}(0) = L_J^{(w)}$. Όπως και στην πρώτη τεχνική τερματισμού που παρουσιάστηκε σε αυτό το κεφάλαιο, έτσι και εδώ το κριτήριο τερματισμού

που χρησιμοποιήθηκε βασίστηκε στην διακύμανση της ποσότητας:

$$E_2(W, K) = \frac{1}{W} \sum_{J=1}^W \left(\frac{\mathcal{L}_J^{(W)}(K) - t_J}{\sum_{i=1}^W t_i} \right)^2 \quad (2.34)$$

Η μέθοδος καθολικής ελαχιστοποίησης διακόπτεται όταν:

$$E_2(w, k) < \sigma_k(E_2), \quad \sigma^2(E_2) < p\sigma_{\text{last}}^2(E_2) \quad (2.35)$$

όπου $\sigma_{\text{last}}^2(E_2)$ είναι η διακύμανση της ποσότητας E_2 όπως αυτή υπολογίστηκε την τελευταία φορά που βρέθηκε καινούργιο ελάχιστο και η ποσότητα p έχει την ίδια λειτουργία με την περίπτωση της μεθόδου του διπλασίου κουτιού.

2.9.4. Ο κανόνας των αναμενόμενων ελαχίστων

Η τεχνική αυτή βασίζεται στην προσπάθεια υπολογισμού του πλήθους των ελαχίστων μίας συναρτήσεως. Η μέθοδος διακόπτεται όταν θεωρήσει πως δεν υπάρχουν πλέον αλλά ελάχιστα να ανακαλύψει.

Έστω ότι με P_m^l συμβολίζεται η πιθανότητα ύστερα από m εφαρμογές της μεθόδου τοπικής ελαχιστοποίησης να έχουν βρεθεί l τοπικά ελάχιστα. Επίσης έστω ότι με π_k συμβολίζεται η πιθανότητα σε μία εφαρμογή της μεθόδου τοπικής ελαχιστοποίησης να βρεθεί το ελάχιστο x_k^* . Αυτή η πιθανότητα είναι γνωστή εκ των προτέρων και δίνεται από τον τύπο:

$$\pi_k = \frac{m(A_k)}{m(S)} \quad (2.36)$$

Η πιθανότητα P_m^l μπορεί να υπολογιστεί αναδρομικά από τον τύπο:

$$P_m^l = \left(1 - \sum_{i=1}^{l-1} \pi_i \right) P_{m-1}^{l-1} + \left(\sum_{i=1}^l \pi_i \right) P_{m-1}^l \quad (2.37)$$

Παρατηρήστε ότι $P_1^0 = 0$, $P_1^1 = 1$. Στην γενική περίπτωση θα ισχύει ότι $P_m^l = 0$ όταν $l > m$ και $P_m^0 = 0, \forall m \geq 1$. Η λογική πίσω από την εξίσωση (2.37) είναι ότι η πιθανότητα μετά από m εφαρμογές τοπικής ελαχιστοποίησης να βρεθούν l ελάχιστα, συνδέεται με την πιθανότητα στις $(m-1)$ εφαρμογές είτε να έχουν βρεθεί $(l-1)$ τοπικά ελάχιστα (και το l τοπικό ελάχιστο ανακαλύπτεται στην επόμενη εφαρμογή) είτε να έχουν βρεθεί ακριβώς l τοπικά ελάχιστα και να μην βρεθεί κανένα στην επόμενη προσπάθεια. Η ποσότητα $\sum_{i=1}^l \pi_i$ είναι η πιθανότητα ότι ένα από τα l ελάχιστα βρίσκεται σε μία προσπάθεια και ομοίως η ποσότητα $1 - \sum_{i=1}^{l-1} \pi_i$ είναι η πιθανότητα να μην βρεθεί κανένα από τα $l-1$ ελάχιστα σε μια προσπάθεια. Επιπλέον, αφού οι ποσότητες P_m^l ορίζει πιθανότητες θα πρέπει να ισχύει ο τύπος:

$$\sum_{l=1}^m P_m^l = 1 \quad (2.38)$$

Για να αποδειχτεί ο παραπάνω ισχυρισμός ας θεωρήσουμε την ποσότητα $s_l = \sum_{i=1}^l \pi_i$. Εφαρμόζοντας άθροισμα και στα δύο μέλη της εξίσωσης (2.37) λαμβάνουμε:

$$\sum_{l=1}^m P_m^l = \sum_{l=1}^m P_{m-1}^{l-1} - \sum_{l=1}^m s_{l-1} P_{m-1}^{l-1} + \sum_{l=1}^m s_l P_{m-1}^l \quad (2.39)$$

Μιας και $P_{m-1}^0 = 0$, $P_{m-1}^m = 0$, από την εξίσωση (2.39) παρατηρούμε ότι $\sum_{l=1}^m P_m^l = \sum_{l=1}^{m-1} P_{m-1}^l$. Από αυτό φαίνεται ότι το άθροισμα $\sum_{l=1}^m P_m^l$ δεν εξαρτάται από την ποσότητα m και σαν συνέπεια θέτοντας $m = 1$ στην εξίσωση (2.37) επαληθεύουμε την εξίσωση (2.38).

Ο αναμενόμενος αριθμός ελαχίστων που έχουν βρεθεί ύστερα από m εφαρμογές της μεθόδου τοπικής ελαχιστοποίησης δίνεται από:

$$\langle w \rangle = \sum_{l=1}^m l P_m^l \quad (2.40)$$

και η διακύμανση από:

$$\sigma^2(w)_m = \sum_{l=1}^m l^2 P_m^l - \left(\sum_{l=1}^m l P_m^l \right)^2 \quad (2.41)$$

Η ποσότητα στην εξίσωση 2.36 δεν μπορεί να υπολογιστεί ακριβώς παρά μόνον προσεγγιστικά. Μια προσέγγιση που μπορεί να χρησιμοποιηθεί, η οποία εξαρτάται από το πλήθος εφαρμογών της μεθόδου τοπικής ελαχιστοποίησης (m) δίνεται από τον τύπο:

$$\pi_i^{(m)} = \frac{L_i^{(m)}}{m} \quad (2.42)$$

Όπου $L_i^{(m)}$ είναι το πλήθος των εφαρμογών της μεθόδου τοπικής ελαχιστοποίησης που οδήγησαν στο ελάχιστο x_i^* . Έτσι η νέα μορφή της εξίσωσης (2.37) είναι:

$$P_m^l = \left(1 - \sum_{i=1}^{l-1} \pi_i^{(m-1)} \right) P_{m-1}^{l-1} + \left(\sum_{i=1}^l \pi_i^{(m-1)} \right) P_{m-1}^l \quad (2.43)$$

Και στον νέο κανόνα τερματισμού, η καθολική μέθοδος τερματίζει με ένα κριτήριο που βασίζεται στην διακύμανση μιας ποσότητας και πιο συγκεκριμένα όταν:

$$|\langle w \rangle - w| \leq \sigma(w)_m, \quad \sigma^2(w)_m < p \sigma^2(w)_{\text{last}} \quad (2.44)$$

Όπου $\sigma^2(w)_{\text{last}}$ είναι η διακύμανση της ποσότητας l την τελευταία φορά που βρέθηκε κάποιο καινούργιο ελάχιστο και η ποσότητα p παίζει τον ίδιο ρόλο με τις προηγούμενες μεθόδους.

2.9.5. Πειραματικά αποτελέσματα

Στην παρούσα ενότητα παρουσιάζονται τα αποτελέσματα από την εφαρμογή των μεθόδων τερματισμού που παρουσιάστηκαν σε αυτό το κεφάλαιο. Στους πίνακες που θα ακολουθήσουν η επικεφαλίδα PCOV αναφέρεται στο κριτήριο τερματισμού της ανισότητας (2.23) χρησιμοποιώντας την παράμετρο $\epsilon = 0.001$, η επικεφαλίδα KAN αναφέρεται στο κριτήριο τερματισμού της ανισότητας (2.20), η επικεφαλίδα DOUBLE αναφέρεται στην μέθοδο τερματισμού του διπλασίου κουτιού, η επικεφαλίδα OBS αναφέρεται στην μέθοδο τερματισμού των παρατηρήσιμων μεγεθών και η επικεφαλίδα EXPM στην μέθοδο των αναμενόμενων ελαχίστων. Για κάθε συνάρτηση εκτελέστηκαν 100 πειράματα, χρησιμοποιώντας διαφορετική ακολουθία τυχαίων αριθμών σε κάθε επανάληψη. Κάτω από την επικεφαλίδα MIN βρίσκεται ο μέσος αριθμός των ελαχίστων που εντοπίστηκαν και κάτω από την στήλη FC ο μέσος αριθμός των απαιτούμενων κλήσεων στην αντικειμενική συνάρτηση.

Στους πίνακες 2.4 και 2.5 παρουσιάζονται τα αποτελέσματα από την εφαρμογή των πέντε κριτηρίων τερματισμού για τις μεθόδους MULTISTART και TMLSL (η οποία θα περιγραφεί αναλυτικά στην συνέχεια του κεφαλαίου). Από τα αποτελέσματα προκύπτει μια αισθητή υπεροχή των νέων κριτηρίων τερματισμού ειδικά σε συναρτήσεις με μεγάλο αριθμό ελαχίστων ή/και διαστάσεων. Ωστόσο πρέπει να επισημανθεί πως η μέθοδος του διπλού κουτιού έχει μικρότερο χρόνο εκτέλεσεως αφού απαιτεί λιγότερους υπολογισμούς από τις άλλες μεθόδους.

Πίνακας 2.4: Αποτελέσματα εφαρμογής διαφόρων κριτηρίων τερματισμού στην μέθοδο MULTISTART.

ΣΥΝΑΡΤΗΣΗ	PCOV		KAN		DOUBLE		OBJS		EXPM	
	MIN	FC	MIN	FC	MIN	FC	MIN	FC	MIN	FC
CAMEL	6	5642	6	2549	6	5503	6	2720	6	2916
RASTRIGIN	49	38104	49	121182	49	19593	49	13342	49	9007
SHUBERT	400	316640	400	8034563	400	577738	400	369958	400	212353
HANSEN	527	426056	527	14220225	527	612015	527	391597	527	240092
GRIEWANK2	528	565932	529	18941546	529	1765175	528	996188	527	449090
GKLS(3,30)	16	5286	13	4249	29	302853	23	84291	25	96260
GKLS(3,100)	34	11464	61	97124	97	7492103	94	5658721	92	3416276
GKLS(4,100)	20	6010	12	7816	95	8629052	73	5290564	93	6358587
GUILIN(10,200)	191	354650	200	4736609	200	3351391	200	2178890	199	1136783
GUILIN(20,100)	96	263869	100	1760826	100	1906288	100	973307	99	655374
Test2N(4)	16	17373	16	18716	16	19424	16	5296	16	3970
Test2N(5)	32	37639	32	78931	32	30607	32	10700	32	7707
Test2N(6)	64	81893	64	336353	64	34840	64	27679	64	18367
Test2N(7)	128	175850	128	1435579	128	117953	128	70370	128	41981
GOLDSTEIN	4	5906	4	3812	4	5391	4	3842	4	3850
BRANIN	3	2173	3	1782	3	1856	3	1782	3	1782
HARTMAN3	3	3348	3	2750	3	3509	3	2778	3	2772
HARTMAN6	2	3919	2	3851	2	3903	2	3907	2	3851
SHEKEL5	5	8720	5	4733	5	22128	5	6430	5	8850
SHEKEL7	7	11742	6	5485	7	30702	7	7581	7	10914
SHEKEL10	10	16020	10	10611	10	36838	9	9812	10	12751

Πίνακας 2.5: Αποτελέσματα εφαρμογής διαφόρων κριτηρίων τερματισμού στην μέθοδο TMLSL.

ΣΥΝΑΡΤΗΣΗ	PCOV		KAN		DOUBLE		OBS		EXPM	
	MIN	FC	MIN	FC	MIN	FC	MIN	FC	MIN	FC
CAMEL	4	406	4	404	6	3415	6	7325	6	1815
RASTRIGIN	31	1821	15	1614	49	24123	49	12815	49	9199
SHUBERT	52	2971	236	226048	400	358623	400	321031	400	275773
HANSEN	66	3837	361	451662	527	811679	527	781209	527	501844
GRIEWANK2	295	56348	481	298174	528	1929165	527	1602467	526	591899
GKLS(3,30)	8	908	7	624	28	191867	28	364849	29	483227
GKLS(3,100)	10	1171	7	727	91	6683608	90	2535356	98	32034155
GKLS(4,100)	2	310	2	310	78	12269342	72	11199668	93	32034155
GUILIN(10,200)	139	76928	181	1600657	200	3470206	200	2650971	199	1752886
GUILIN(20,100)	74	72974	99	734645	100	1915478	100	1333144	99	867312
Test2N(4)	12	874	8	508	16	3821	16	2807	16	4162
Test2N(5)	32	3798	32	7750	32	8953	32	6634	32	6092
Test2N(6)	20	1513	11	1184	64	54555	64	34649	64	27500
Test2N(7)	104	10776	128	73947	128	88750	128	69371	128	52248
GOLDSTEIN	4	583	3	587	4	4606	4	11709	4	2522
BRANIN	3	354	3	359	3	1007	3	182919	3	1227
HARTMAN3	3	1567	3	1567	3	2014	3	18271	3	3857
HARTMAN6	2	685	2	677	2	912	2	87124	2	2094
SHEKEL5	4	446	4	446	5	6434	5	90356	5	3310
SHEKEL7	4	436	4	434	6	14617	7	191969	6	4363
SHEKEL10	5	412	5	406	9	18447	10	78645	9	10288

ΚΕΦΑΛΑΙΟ 3

ΓΕΝΕΤΙΚΑ ΒΕΛΤΙΩΜΕΝΕΣ ΣΤΟΧΑΣΤΙΚΕΣ ΜΕΘΟΔΟΙ

Περιεχόμενα

3.1	Γενικά	40
3.2	Γενετικοί αλγόριθμοι	40
3.2.1	Αναπαράσταση μελών πληθυσμού	41
3.2.2	Συνάρτηση καταλληλότητας	41
3.2.3	Επιλογή	42
3.2.4	Διασταύρωση	43
3.2.5	Ελιτισμός	44
3.2.6	Μετάλλαξη	45
3.2.7	Κριτήρια τερματισμού	45
3.3	Βασικές αρχές γενετικού προγραμματισμού	45
3.3.1	Χρησιμοποιούμενα σύνολα	46
3.3.2	Κλειστότητα	46
3.3.3	Επάρκεια	47
3.3.4	Γενετικοί τελεστές	48
3.4	Γραμματική Εξέλιξη	49

3.4.1	Βασικοί ορισμοί	49
3.4.2	Παράδειγμα παραγωγής	50
3.4.3	Γενετικοί τελεστές	51
3.5	Ο αλγόριθμος Genetic Random Search (GRS)	52
3.5.1	Προτεινόμενη γραμματική	53
3.5.2	Περιγραφή του αλγορίθμου	54
3.5.3	Παράδειγμα εφαρμογής	55
3.6	Η γενετικώς τροποποιημένη μέθοδος CRS (GenPrice)	55
3.7	Η γενετικώς τροποποιημένη μέθοδος Simulated Annealing (GenAnneal)	58
3.8	Πειραματικά αποτελέσματα	61
3.8.1	Αποτελέσματα CRS	62
3.8.2	Αποτελέσματα προσομοιούμενης ανοπτήσεως	62
3.9	Υλοποιημένο λογισμικό	64

3.1 Γενικά

Το κεφάλαιο ξεκινά με μια ανασκόπηση του γενετικού προγραμματισμού και της μεθοδολογίας της Γραμματικής Εξελίξεως, η οποία θα αποτελέσει την βάση για την υλοποίηση των δύο νέων στοχαστικών τεχνικών που θα παρουσιαστούν στην συνέχεια. Οι μέθοδοι αυτές αποτελούν βελτιωμένες εκδοχές των καθολικών μεθόδων ελαχιστοποίησης Controlled Random Search και Simulated Annealing. Οι τροποποιημένες εκδοχές χρησιμοποιούν έναν διαφορετικό τρόπο δειγματοληψίας που δεν βασίζεται σε ομοιόμορφα τυχαία κατανομημένα σημεία αλλά σε μονοπάτια στα οποία η τιμή της αντικειμενικής συναρτήσεως μειώνεται σταδιακά που δημιουργούνται με την χρήση μιας τεχνικής βασισμένης σε Γραμματική Εξέλιξη, με το όνομα Genetic Random Search (GRS).

3.2 Γενετικοί αλγόριθμοι

Οι γενετικοί αλγόριθμοι αποτελούν μια ειδική τεχνική παραμετρικής βελτιστοποίησης, με την οποία μια ομάδα από υποψήφιες λύσεις¹ του αντικειμενικού προβλήματος υπόκεινται σε μία επαναληπτική διαδικασία για την ανάδειξη του καλύτερου μέλους της ομάδας. Αυτή η προσέγγιση έχει τις ρίζες της στα μοντέλα βιολογικής εξέλιξεως, όπου από διαφορετικούς πληθυσμούς μόνον τα καταλληλότερα ως προς το περιβάλλον μέλη επιβιώνουν και αυτά με την σειρά τους παράγουν νέες γενιές πληθυσμών. Μια γενική υλοποίηση ενός γενετικού αλγορίθμου είναι και αυτή που δίνεται στην συνέχεια:

1. Αρχικοποίηση πληθυσμού.
2. Αποτίμηση πληθυσμού.
3. Επιλογή μελών για διασταύρωση.
4. Διαδικασία διασταυρώσεως.
5. Μετάλλαξη μελών του πληθυσμού.
6. Αν ισχύει το κριτήριο τερματισμού επιστροφή του καλύτερου μέλους, διαφορετικά επανάληψη από το βήμα 2.

Δεν είναι απαραίτητο τα βήματα του παραπάνω αλγορίθμου να εφαρμοστούν με αυτήν την σειρά ούτε και είναι απαραίτητο πως θα υπάρχουν όλα σε έναν αλγόριθμο. Ωστόσο στις περισσότερες υλοποιήσεις παρουσιάζεται ένα σχήμα σαν το παραπάνω. Μια αρκετά καλή περιγραφή των γενετικών αλγορίθμων γίνεται και στο βιβλίο του D. Goldberg [19].

¹Συχνά αποκαλούνται και χρωμοσώματα.

3.2.1. Αναπαράσταση μελών πληθυσμού

Έχει αποδειχθεί θεωρητικά πως οι γενετικοί αλγόριθμοι συγκλίνουν στην βέλτιστη λύση όταν τα χρωμοσώματά τους είναι εκπεφρασμένα σε δυαδική μορφή. Ωστόσο στην πράξη δεν είναι δυνατόν να χρησιμοποιούνται πάντοτε δυαδικές αναπαραστάσεις. Για παράδειγμα έστω το πρόβλημα της ευρέσεως των βαρών ενός τεχνητού νευρωνικού δικτύου με την χρήση γενετικών αλγορίθμων. Τα βάρη μπορούν να λάβουν άπειρες τιμές μέσα στο σύνολο των δεκαδικών αριθμών και σε πολλές περιπτώσεις, όπως στα επαναληπτικά (Recurrent) νευρωνικά δίκτυα, μία πολύ μικρή μεταβολή σε κάποιο βάρος του δικτύου έχει σαν αποτέλεσμα την δραματική αλλαγή στην συμπεριφορά του συνολικού μοντέλου. Τέτοια παραδείγματα απαιτούν η αναπαράσταση των χρωμοσωμάτων να είναι σε μορφή κινητής υποδιαστολής ή τουλάχιστον να παραμείνουν σε δυαδική μορφή και να λαμβάνει χώρα κάθε φορά που αποτιμάται η καταλληλότητα μια απεικόνιση από τον γενότυπο, που είναι τα δυαδικά χρωμοσώματα, στον φαινότυπο που είναι η παράμετρος ως προς την οποία γίνεται ελαχιστοποίηση, στο συγκεκριμένο παράδειγμα το βάρος του τεχνητού νευρωνικού δικτύου. Ωστόσο αυτές οι μετατροπές από γενότυπο σε φαινότυπο δεν έχουν την μέγιστη δυνατή ακρίβεια επειδή χρησιμοποιούν έναν περιορισμένο αριθμό από δυαδικά ψηφία (bits) για την αναπαράσταση. Στην πράξη στα χρωμοσώματα χρησιμοποιούνται απευθείας αριθμοί κινητής υποδιαστολής χωρίς κάποια ενδιάμεση απεικόνιση και κάτι τέτοιο έχει βρεθεί [76] ότι:

1. Επιτυγχάνει καλύτερη προσαρμογή σε προβλήματα που σχετίζονται με αριθμούς κινητής υποδιαστολής.
2. Αυξάνει την ταχύτητα του αλγορίθμου.
3. Επιτρέπει τον συνδυασμό των γενετικών τελεστών με άλλες μεθοδολογίες (π.χ. τοπική ελαχιστοποίηση) για την επίτευξη καλύτερων αποτελεσμάτων.

Ωστόσο η απευθείας χρήση αριθμών κινητής υποδιαστολής στα χρωμοσώματα έχει και μειονεκτήματα, όπως:

1. Πρέπει να βρεθούν διαφορετικοί γενετικοί τελεστές από αυτούς που υπάρχουν ήδη.
2. Δεν υπάρχει θεωρητική απόδειξη πως ο γενετικός αλγόριθμος θα συγκλίνει στην βέλτιστη λύση.

3.2.2. Συνάρτηση καταλληλότητας

Στενά συνδεδεμένο ζήτημα με αυτό των γενετικών αλγορίθμων είναι επιλογή της συναρτήσεως καταλληλότητας (fitness function). Η συνάρτηση αυτή αποτελεί και

ένα κριτήριο για το πόσο κατάλληλο είναι ένα μέλος του πληθυσμού. Δεν υπάρχει κάποιος σαφής τρόπος για την επιλογή αυτής της συναρτήσεως και εξαρτάται άμεσα από το πρόβλημα που καλείται ο αλγόριθμος να επιλύσει. Επίσης δεν είναι απαραίτητο αυτή η συνάρτηση να είναι θετική ή αρνητική (αν και συνήθως είναι θετική), αλλά πρέπει να είναι τέτοια ώστε να διακρίνονται τα μέλη του πληθυσμού μεταξύ τους. Για παράδειγμα συνάρτηση καταλληλότητας σε μία διαδικασία επιλύσεως στο πρόβλημα του πλανοδίου πωλητή (TSP) μπορεί να είναι το μήκος του μονοπατιού που υποδεικνύεται κάθε φορά από τον αλγόριθμο. Όσο μικρότερο είναι το μήκος του μονοπατιού τόσο πιο κατάλληλο είναι το αντίστοιχο μέλος του πληθυσμού.

3.2.3. Επιλογή

Η διαδικασία της επιλογής εφαρμόζεται για την επιλογή των ατόμων από τον πληθυσμό τα οποία θα δημιουργήσουν νέα μέλη του πληθυσμού. Η διαδικασία αυτή είναι δανεισμένη από την φύση, όπου η επιλογή κατάλληλων γονέων για αναπαραγωγή τείνει να δημιουργήσει τον ίδιο ή πιο κατάλληλους απογόνους. Γενικά δύο είναι η διαδικασίες που χρησιμοποιούνται περισσότερο σήμερα για επιλογή γονέων: η επιλογή ρουλέτας και η επιλογή tournament.

Η επιλογή ρουλέτας πολλές φορές ονομάζεται και επιλογή ανάλογη της καταλληλότητας. Σε αυτήν την περίπτωση κάθε άτομο i ταξινομείται ως προς την καταλληλότητά του f_i (υποθέτοντας ότι $f_i \geq 0$) και στην συνέχεια επιλέγουμε έναν αριθμό r τέτοιο ώστε:

$$r \leq \sum_{i=1}^N f_i \quad (3.1)$$

όπου N είναι το πλήθος των ατόμων στον πληθυσμό. Το άτομο k που επιλέγεται για διασταύρωση είναι εκείνο για το οποίο ισχύει:

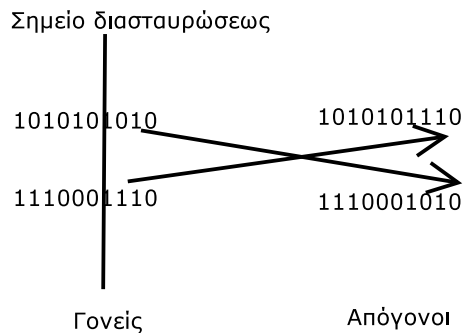
$$\sum_{i=1}^{k-1} f_i \leq r \leq \sum_{i=1}^k f_i \quad (3.2)$$

Η παραπάνω τεχνική έχει δύο βασικά μειονεκτήματα:

1. Έχει αργή υλοποίηση.
2. Οδηγεί σε διασταυρώσεις στις οποίες κυριαρχεί η ομάδα του πληθυσμού με την μεγαλύτερη καταλληλότητα. Αυτό το πρόβλημα οδηγεί τον πληθυσμό σε τοπικά ελάχιστα (ή μέγιστα) της συναρτήσεως καταλληλότητας και δεν του δίνει την δυνατότητα να οδηγηθεί προς το ολικό ελάχιστο.

Για τους λόγους αυτούς στην πράξη πολλές φορές χρησιμοποιείται η τεχνική της επιλογής tournament, που προκαλεί μεγαλύτερη ποικιλομορφία στον πληθυσμό.

Σχήμα 3.1: Διασταύρωση ενός σημείου.



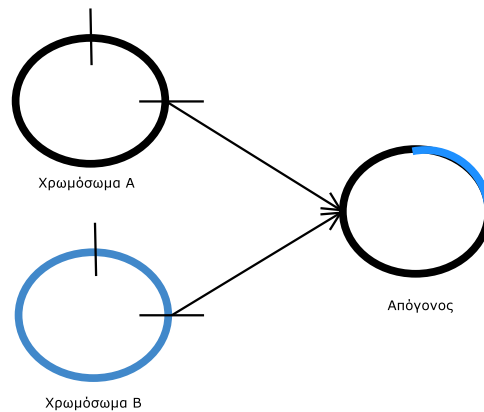
Σύμφωνα με αυτήν την τεχνική επιλογής για να επιλέξουμε ένα μέλος για διασταύρωση επιλέγουμε ομοιόμορφα περισσότερα από δύο άτομα. Στην συνέχεια το άτομο με την καλύτερη τιμή καταλληλότητας επιλέγεται για διασταύρωση.

3.2.4. Διασταύρωση

Στην διαδικασία της διασταυρώσεως μέλη που έχουν επιλεγεί στην φάση της επιλογής ανταλλάσσουν γενετικό υλικό για την δημιουργία απογόνων. Οι απόγονοι είτε θα αντικαταστήσουν κάποια μέλη του πληθυσμού (συνήθως τα χειρότερα) είτε θα εισαχθούν και αυτοί στον πληθυσμό. Δεν υπάρχει κάποιος γενικά αποδεκτός τρόπος διασταυρώσεως, ωστόσο στην βιβλιογραφία συναντάμε τους ακόλουθους τρόπους:

1. **Διασταύρωση ενός σημείου.** Η ανταλλαγή γενετικού υλικού γίνεται με την επιλογή ενός σημείου στα χρωμοσώματα και την ανταλλαγή του γενετικού υλικού που υπάρχει πριν και μετά από αυτό. Στην πράξη πολύ συχνά χρησιμοποιούμε την διασταύρωση ενός σημείου και την διασταύρωση δύο σημείων. Στο σχήμα 3.1 βλέπουμε την εφαρμογή της διασταυρώσεως ενός σημείου σε δυαδικά χρωμοσώματα.
2. **Διασταύρωση δύο σημείων.** Σε αυτήν την μορφή διασταυρώσεως τα χρωμοσώματα θεωρούνται πως είναι κυκλικά με το πρώτο και το τελευταίο στοιχείο τους να είναι συνδεδεμένα. Επιλέγονται τυχαία δύο σημεία πάνω στα δύο κυκλικά χρωμοσώματα και τα αντίστοιχα τμήματα, ανταλλάσσονται όπως φαίνεται και στο σχήμα 3.2.
3. **Ομοιόμορφη διασταύρωση.** Σε αυτήν την εκδοχή διασταυρώσεως για κάθε στοιχείο των δύο χρωμοσωμάτων επιλέγεται ένας τυχαίος αριθμός στο

Σχήμα 3.2: Διασταύρωση δύο σημείων.



διάστημα $(0,1)$. Αν αυτός είναι μικρότερος του 0.5, τότε το στοιχείο του πρώτου χρωμοσώματος πηγαίνει στον απόγονο, διαφορετικά το στοιχείο του δεύτερου χρωμοσώματος.

Συνήθως δεν διασταυρώνονται όλα τα μέλη ενός πληθυσμού. Το ποσοστό των μελών που θα διασταυρωθούν συνήθως καθορίζεται από ένα συντελεστή επιλογής, ο οποίος είναι στο διάστημα $[0,1]$. Όσο μεγαλύτερη είναι η τιμή αυτού του συντελεστή, τόσο περισσότερες διασταυρώσεις θα πραγματοποιηθούν.

3.2.5. Ελιτισμός

Ο ελιτισμός είναι και αυτός μια μορφή επιλογής, καθώς σε κάθε γενιά το καλύτερο ή τα K καλύτερα στοιχεία της προηγούμενης γενιάς περνούν στην επομένη γενιά χωρίς να αλλοιωθούν. Αυτή η διαδικασία έχει το πλεονέκτημα πως καλά στοιχεία του πληθυσμού που έχουν βρεθεί σε κάποια επανάληψη δεν θα χαθούν στις επόμενες γενιές από τις διασταυρώσεις και τις μεταλλάξεις που θα υποστούν. Ωστόσο η παρουσία των καλύτερων στοιχείων της προηγούμενης γενιάς στην επόμενη γενιά οδηγεί τον πληθυσμό σε πρόωρη σύγκλιση σε τοπικά ελάχιστα και δεν επιτρέπει στον γενετικό αλγόριθμο να κάνει καλύτερη εξερεύνηση του χώρου αναζήτησης. Εναλλακτικά μπορεί να φυλάσσεται το καλύτερο μέχρι τώρα στοιχείο του πληθυσμού και μετά το τέλος του γενετικού αλγορίθμου να χρησιμοποιείται αυτό ως το αποτέλεσμα του αλγορίθμου.

3.2.6. Μετάλλαξη

Η μετάλλαξη είναι η τυχαία αλλαγή ενός ή περισσότερων σημείων ενός χρωμοσώματος. Η μετάλλαξη βασίζεται στην χρήση ενός συντελεστή μεταλλάξεως ο οποίος είναι ένας αριθμός στο $[0,1]$. Όσο μεγαλύτερος είναι αυτός ο αριθμός, τόσο μεγαλύτερη είναι η πιθανότητα να συμβεί μία μετάλλαξη σε ένα στοιχείο ενός χρωμοσώματος. Ο αριθμός αυτός γενικά είναι μικρός σε τιμή και πολλές φορές χρησιμοποιείται ο ακόλουθος κανόνας για τον καθορισμό του:

$$p_m = \frac{1}{M} \quad (3.3)$$

όπου M είναι το μέγεθος του χρωμοσώματος. Ο παραπάνω κανόνας μπορεί να χρησιμοποιηθεί προκειμένου κατά μέσο όρο να αλλάζει ένα στοιχείο σε κάθε χρωμόσωμα.

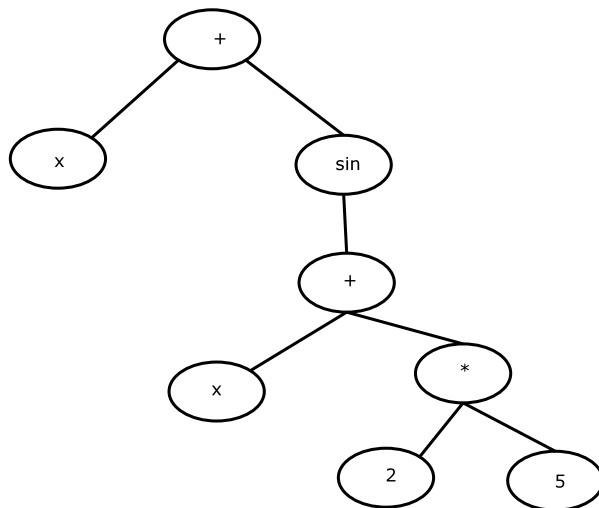
3.2.7. Κριτήρια τερματισμού

Τα κριτήρια τερματισμού ενός γενετικού αλγορίθμου δεν είναι πάντοτε καθορισμένα με σαφήνεια. Σε άλλες περιπτώσεις ο γενετικός αλγόριθμος παράγει γενιές μέχρι να βρει μια ικανοποιητική καταλληλότητα, πχ. το τετραγωνικό σφάλμα εκπαιδύσεως ενός τεχνητού νευρωνικού δικτύου να είναι κάτω από κάποιο όριο. Σε άλλες περιπτώσεις από την αρχή έχει τεθεί ένας μέγιστος αριθμός γενιών και ο αλγόριθμος τερματίζει όταν φτάσει σε αυτόν τον αριθμό. Σε άλλες υλοποιήσεις κριτήριο τερματισμού μπορεί να είναι και η ποικιλομορφία του πληθυσμού, δηλαδή κατά πόσο είναι ίδια τα στοιχεία του πληθυσμού. Βέβαια κανείς μπορεί να συνδυάσει τους παραπάνω τρόπους τερματισμού ή ακόμα και να επινοήσει κάποιον που να ανταποκρίνεται καλύτερα στο πρόβλημα που καλείται να λύσει.

3.3 Βασικές αρχές γενετικού προγραμματισμού

Ο γενετικός προγραμματισμός όπως επινοήθηκε από τον Koza στην εργασία του [88] είναι μία ειδική περίπτωση γενετικών αλγορίθμων, όπου για χρωμοσώματα δεν χρησιμοποιούνται αλφαριθμητικά ή δυαδικοί αριθμοί, αλλά συντακτικά δέντρα προγραμμάτων. Για παράδειγμα η έκφραση $x + \sin(x + 2 * 5)$ μπορεί να εκφραστεί με μορφή δένδρου όπως φαίνεται στο σχήμα 3.3. Στον γενετικό προγραμματισμό με δένδρα, η διασταύρωση γίνεται με ανταλλαγή υποδένδρων μεταξύ προγραμμάτων και η μετάλλαξη με την τυχαία παραγωγή υποδένδρων και αντικατάσταση παλαιότερων.

Σχήμα 3.3: Συντακτικό δένδρο του $x + \sin(x + 2 * 5)$.



3.3.1. Χρησιμοποιούμενα σύνολα

Τα χρησιμοποιούμενα σύνολα στον γενετικό προγραμματισμό είναι το σύνολο των τερματικών συμβόλων που συμβολίζεται συνήθως με T και το σύνολο των μη τερματικών συμβόλων που συνήθως συμβολίζεται με F . Τα δύο σύνολα μαζί είναι το σύνολο $C = T \cup F$ όλων των συμβόλων της γραμματικής που χρησιμοποιείται για την παραγωγή προγραμμάτων. Στο πρώτο σύνολο εμπεριέχονται τα στοιχεία που βρίσκονται στα φύλλα των συντακτικών δένδρων παραγωγής. Παραδείγματα τερματικών συμβόλων μπορούν να είναι οι ακέραιοι αριθμοί $\{0, 1, 2, \dots\}$, μεταβλητές όπως x, y κτλ. Από την άλλη στο σύνολο των μη τερματικών συμβόλων βρίσκονται τα στοιχεία εκείνα που είναι εσωτερικά στο δένδρο παραγωγής. Παραδείγματα τέτοιων συμβόλων μπορούν να είναι οι αριθμητικοί τελεστές $\{+, -, *, /\}$, συναρτήσεις όπως $\sin(x), \cos(x)$ και οποιοδήποτε σύμβολο στην γραμματική εφαρμόζεται σε περισσότερα από ένα σύμβολα. Για παράδειγμα η συνάρτηση $\min(x, y)$ εφαρμόζεται σε δύο στοιχεία, ενώ η συνάρτηση $\log(x)$ εφαρμόζεται σε ένα στοιχείο. Αν θέλουμε να εννοήσουμε τα δύο παραπάνω σύνολα μπορούμε να πούμε πως τα τερματικά σύμβολα είναι αυτά που δεν εφαρμόζονται σε κανένα στοιχείο και τα μη τερματικά σύμβολα είναι όλα τα υπόλοιπα.

3.3.2. Κλειστότητα

Η κλειστότητα στον γενετικό προγραμματισμό απαιτεί πως η εφαρμογή οποιασδήποτε συναρτήσεως του συνόλου F σε ένα αριθμό από σύμβολα θα επιστρέψει

Πίνακας 3.1: Παραδείγματα προστατευόμενων συναρτήσεων.

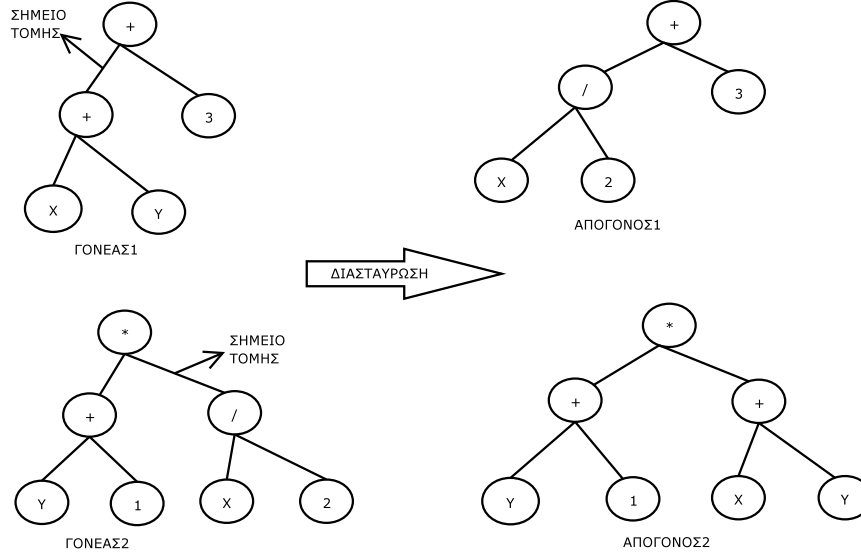
Συνάρτηση	Προστατευόμενη έκδοση
x/y	Αν $y = 0$ 1, αλλιώς $\frac{x}{y}$
$\log(x)$	Αν $x = 0$ 0, Αλλιώς $\log(x)$
\sqrt{x}	$\sqrt{ x }$

έγκυρη έκφραση. Για παράδειγμα ας θεωρήσουμε πως διαθέτουμε το σύνολο λογικών πράξεων $C = \{\text{AND}, \text{OR}, \text{NOT}, x, y, \text{TRUE}\}$. Το σύνολο των τερματικών συμβόλων εμπεριέχει τις μεταβλητές x, y και την σταθερά TRUE. Το σύνολο των μη τερματικών συμβόλων είναι οι λογικές συναρτήσεις AND, OR και NOT. Οποιαδήποτε εφαρμογή των συναρτήσεων με οποιαδήποτε διάταξη θα έχει σαν αποτέλεσμα μια έγκυρη λογική έκφραση. Από την άλλη αν θεωρήσουμε το σύνολο $C = \{+, -, *, /, x, y, 1, 0\}$ θα διαπιστώσουμε πως δεν έχει την ιδιότητα της κλειστότητας, γιατί για παράδειγμα οι εκφράσεις $x/0$ και $\frac{x+y}{x-x}$ δεν δίνουν έγκυρο αποτέλεσμα (διαίρεση με μηδέν). Το ίδιο πρόβλημα μπορεί να παρατηρηθεί και σε άλλες περιπτώσεις όπως με την χρήση της τετραγωνικής ρίζας \sqrt{x} που απαιτεί το όρισμα x να είναι θετικός αριθμός, προκειμένου να επιστρέψει έγκυρο αποτέλεσμα. Οι παραπάνω “δυσάρεστες” καταστάσεις μπορούν να αποφευχθούν με την χρήση προστατευόμενων συναρτήσεων, δηλαδή συναρτήσεων οι οποίες έχουν επεκταθεί για να επιστρέφουν έγκυρα αποτελέσματα ακόμα και όταν τα ορίσματά τους δεν είναι έγκυρα. Παραδείγματα προστατευόμενων συναρτήσεων φαίνονται στον πίνακα 3.1. Προφανώς κάποιος μπορεί να επεκτείνει τις συναρτήσεις του πίνακα με διαφορετικούς τρόπους ανάλογα με τις ανάγκες του προβλήματός του. Ωστόσο η επιβολή κλειστότητας με χρήση προστατευόμενων συναρτήσεων δεν εμφανίζεται πάντα στα προβλήματα γενετικού προγραμματισμού, καθώς οι προστατευόμενες συναρτήσεις διαστρεβλώνουν το αποτέλεσμα που παράγει ο αλγόριθμος. Έτσι, σε πολλές εφαρμογές αυτό που γίνεται είναι όταν εμφανίζεται παραβίαση της κλειστότητας (πχ. διαίρεση με μηδέν) η καταλληλότητα του συγκεκριμένου χρωμοσώματος τίθεται σε μια πολύ μικρή τιμή.

3.3.3. Επάρκεια

Η επάρκεια ενός συνόλου C έγκειται στην δυνατότητα του συνόλου να εκφράσει με συνδυασμούς τερματικών και μη συμβόλων την λύση του προβλήματος. Για παράδειγμα αν διαθέτουμε το σύνολο $C = \{x, +, -, *, /, 2, 1, 0\}$ και η λύση σε ένα πρόβλημα προσεγγίσεως συναρτήσεων είναι η $f(x) = x^3 + x^2 + x$, τότε το σύνολο θεωρείται επαρκές ως προς την συνάρτηση $f(x)$, αφού η έκφραση $g(x) = x*x*x+x*x+x$ είναι ίδια με την ζητούμενη συνάρτηση. Από την άλλη αν

Σχήμα 3.4: Παράδειγμα διασταυρώσεως στον γενετικό προγραμματισμό.



Ζητείται η προσέγγιση της συναρτήσεως $f(x) = \exp(x)$, τότε το σύνολο δεν είναι επαρκές. Ωστόσο η ζητούμενη συνάρτηση μπορεί να υπολογιστεί προσεγγιστικά από αναπτύγματα της σειράς Taylor ως ακολούθως:

$$\exp(x) \simeq 1$$

$$\exp(x) \simeq 1 + x$$

$$\exp(x) \simeq 1 + x + \frac{1}{2}x * x$$

$$\exp(x) \simeq 1 + x + \frac{1}{2}x * x + \frac{1}{1 + 2 + 2 + 1}x * x * x$$

3.3.4. Γενετικοί τελεστές

Οι γενετικοί τελεστές της διασταυρώσεως και της μεταλλάξεως βρίσκουν εφαρμογή και στην περίπτωση του γενετικού προγραμματισμού. Ωστόσο είναι περισσότερο πολύπλοκοι σε αυτήν την περίπτωση επειδή τα χρωμοσώματα είναι συντακτικά δένδρα παραγωγής. Για παράδειγμα η διασταύρωση ανάμεσα σε δύο χρωμοσώματα πραγματοποιείται συνήθως με την ανταλλαγή υποδένδρων των δύο εμπλεκομένων χρωμοσωμάτων, όπως μπορεί να φανεί και στο σχήμα 3.4. Για την περίπτωση του γενετικού τελεστή της μεταλλάξεως μπορούν να χρησιμοποιηθούν διάφορες τεχνικές. Σε άλλες περιπτώσεις ένα σύμβολο του δένδρου παραγωγής μπορεί να αλλάξει σε κάποιο άλλο με τον ίδιο αριθμό παραμέτρων, πχ. ένα σύμβολο \sin να γίνει σύμβολο \cos ή ένα x να γίνει y . Σε άλλες περιπτώσεις ένα

υπόδενδρο παραγωγής αφαιρείται από το δένδρο και παράγεται ένα καινούργιο το οποίο το αντικαθιστά.

3.4 Γραμματική Εξέλιξη

Η τεχνική της Γραμματικής Εξελίξεως είναι μία νέα εξελικτική τεχνική που αναπτύχθηκε από τους O'Neil και Ryan στην εργασία [89] και βασίζεται στην περιγραφή με BNF των γραμματικών για την παραγωγή προγραμμάτων σε οποιαδήποτε γλώσσα προγραμματισμού. Η τεχνική αυτή έχει εφαρμοστεί με επιτυχία σε μια σειρά από προβλήματα όπως εύρεση τριγωνομετρικών ισοτήτων [99], ρομποτική [100], αυτόματη παραγωγή αλγορίθμων για caching [101], οικονομικά [102] κτλ. Περισσότερα για αυτήν την τεχνική μπορεί να βρει κανείς στα σχετικά άρθρα [90, 91, 92, 93] όπως και στον σχετικό ιστοχώρο

<http://www.grammatical-evolution.org>

3.4.1. Βασικοί ορισμοί

Η τεχνική της Γραμματικής Εξελίξεως είναι ένας εξελικτικός αλγόριθμος, ο οποίος μπορεί να παράγει προγράμματα σε οποιαδήποτε γλώσσα προγραμματισμού. Ο αλγόριθμος απαιτεί την γραμματική της αντικειμενικής γλώσσας εκπεφρασμένη σε BNF μορφή και μια ανίστοιχη συνάρτηση καταλληλότητας. Τα χρωμοσώματα στην διαδικασία αυτή δεν εκφράζονται με συντακτικά δένδρα παραγωγής, όπως είδαμε στην προηγούμενη ενότητα, αλλά σαν διανύσματα ακέραιων αριθμών. Κάθε ακέραιος συμβολίζει και έναν κανόνα παραγωγής από την αντίστοιχη BNF γραμματική. Ο αλγόριθμος παραγωγής εκφράσεων ξεκινά από το σύμβολο εκκινήσεως της γραμματικής και βαθμιαία δημιουργεί το αντίστοιχο πρόγραμμα αντικαθιστώντας μη τερματικά σύμβολα με το δεξιό τμήμα του επιλεγέντος κανόνα παραγωγής. Η επιλογή του κανόνα παραγωγής πραγματοποιείται σε δύο βήματα:

- Διαβάζουμε ένα στοιχείο από το χρωμόσωμα, έστω με τιμή V .
- Διαλέγουμε τον αντίστοιχο κανόνα παραγωγής σύμφωνα με το σχήμα

$$\text{Rule} = V \bmod \text{NR} \quad (3.4)$$

όπου NR είναι ο αριθμός των κανόνων που διαθέτει το μη - τερματικό σύμβολο το οποίο θα αντικατασταθεί. Η διαδικασία της αντικαταστάσεως μη τερματικών συμβόλων με το δεξιό τμήμα κανόνων παραγωγής συνεχίζεται μέχρι να συμβεί μία από τις δύο επόμενες καταστάσεις:

Σχήμα 3.5: Γραμματική μια γλώσσας με πράξεις ανάμεσα σε x και y.
 $S ::= \langle \text{expr} \rangle$

$$\begin{aligned} \langle \text{expr} \rangle &::= \langle \text{expr} \rangle + \langle \text{expr} \rangle & (0) \\ &| \langle \text{expr} \rangle - \langle \text{expr} \rangle & (1) \\ &| x & (2) \\ &| y & (3) \end{aligned}$$

1. Να έχει παραχθεί ένα έγκυρο πρόγραμμα στην αντικειμενική γλώσσα προγραμματισμού. Αυτό σημαίνει πως δεν υπάρχουν πλέον μη τερματικά σύμβολα στην έκφραση για να αντικατασταθούν και η διαδικασία τερματίζει με επιτυχία.
2. Να έχουν διαβαστεί όλα τα στοιχεία από το χρωμόσωμα και να μην έχει παραχθεί έκφραση που δεν περιέχει μη - τερματικά σύμβολα. Σε αυτήν την περίπτωση είτε το χρωμόσωμα απορρίπτεται και του ανατίθεται μια τιμή καταλληλότητας τέτοια που να μην του επιτρέπει να συμμετάσχει στην διαδικασία αναπαραγωγής ή η ανάγνωση των στοιχείων του χρωμοσώματος ξεκινά και πάλι από την αρχή. Η δεύτερη περίπτωση ονομάζεται **αναδιπλωση** και συνήθως 1-2 αναδιπλώσεις είναι αρκετές είτε για να παράγουν μια έγκυρη έκφραση είτε για να δείξουν πως το συγκεκριμένο χρωμόσωμα δεν είναι σε θέση να παράγει έγκυρες εκφράσεις. Στις υλοποιημένους αλγορίθμους αυτού του κεφαλαίου επιτρέπονται 2 αναδιπλώσεις.

3.4.2. Παράδειγμα παραγωγής

Για να γίνει περισσότερο κατανοητή η διαδικασία παραγωγής προγραμμάτων έστω πως διατίθεται η γραμματική του σχήματος 3.5 που παριστά πράξεις προσθέσεως και αφαιρέσεως ανάμεσα στο x και το y. Το σύμβολο S είναι το αρχικό σύμβολο της γραμματικής και στους υλοποιημένους αλγορίθμους αυτού του κειμένου γίνεται η υπόθεση πως το αρχικό σύμβολο θα έχει πάντα έναν κανόνα παραγωγής. Οι αριθμοί στο τέλος των κανόνων παραγωγής διακριτοποιούν τους κανόνες μεταξύ τους. Έστω το χρωμόσωμα [8,2,5,3,3]. Σταδιακά θα γίνουν οι επόμενες παραγωγές του σχήματος 3.6. Ας δούμε αναλυτικά τα βήματα για την παραγωγή της εκφράσεως $x+y-y$. Το αρχικό σύμβολο της γραμματικής αντικαθίσταται από το σύμβολο $\langle \text{expr} \rangle$, αφού ο μόνος γραμματικός κανόνας που υπάρχει από το αρχικό σύμβολο είναι αυτός που οδηγεί στο $\langle \text{expr} \rangle$. Στην συνέχεια επειδή το $\langle \text{expr} \rangle$ είναι μη τερματικό σύμβολο διαβάζεται ένα στοιχείο από το χρωμόσωμα προκειμένου να παραχθεί συμβολοσειρά με τερματικά σύμβολα (που είναι και ο σκοπός κάθε γραμματικής). Το επόμενο σύμβολο στο χρωμόσωμα είναι το 8.

Σχήμα 3.6: Παραγωγή έκφρασης.

```
S::=<expr>
      <expr>+<expr>
      x+<expr>
      x+<expr>-<expr>
      x+y-<expr>
      x+y-y
```

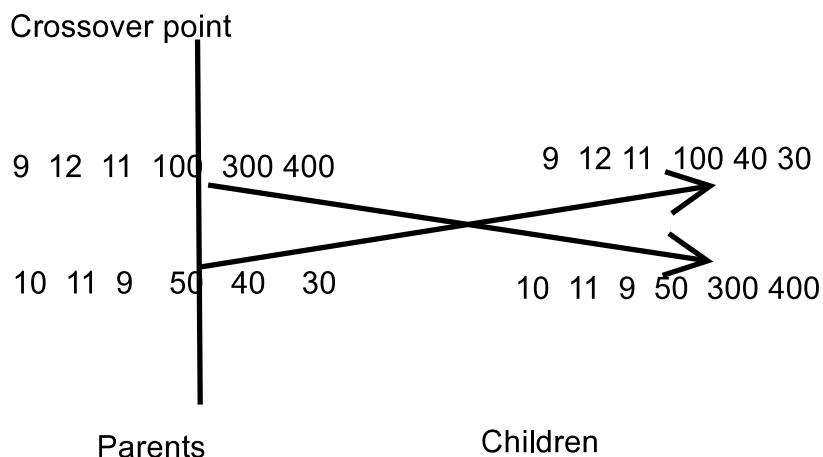
Το πλήθος των κανόνων του συμβόλου $\langle \text{expr} \rangle$ είναι 4 και $8 \% 4 = 0$. Αυτό έχει σαν αποτέλεσμα να επιλεγεί ο πρώτος κανόνας που θα δώσει $\langle \text{expr} \rangle + \langle \text{expr} \rangle$. Στην συνέχεια καθένα από τα μη τερματικά $\langle \text{expr} \rangle$ που παρήχθησαν πρέπει να αναχθεί σε σειρά τερματικών συμβόλων. Με την ανάγνωση ακόμα ενός συμβόλου από το χρωμόσωμα λαμβάνεται ο κανόνας 2 που θα μετατρέψει το πρώτο $\langle \text{expr} \rangle$ σε x . Στην συνέχεια με την ίδια διαδικασία το δεύτερο $\langle \text{expr} \rangle$ θα γίνει $\langle \text{expr} \rangle - \langle \text{expr} \rangle$ και τελικά $y - y$.

3.4.3. Γενετικοί τελεστές

Η διασταύρωση και η μετάλλαξη στην περίπτωση της Γραμματικής Εξελίξεως γίνεται με την χρήση των απλών τελεστών διασταυρώσεως και μεταλλάξεως που παρουσιάστηκαν στην περίπτωση των γενετικών αλγορίθμων, αφού ουσιαστικά τα χρωμοσώματα είναι ακέραιοι αριθμοί και δεν απαιτείται κάποια ειδική μεταχείριση πέραν αυτής που εμφανίζεται στους γενετικούς αλγόριθμους. Ένα παράδειγμα από την εφαρμογή της διαδικασίας της διασταυρώσεως ενός σημείου σε Γραμματική Εξέλιξη παρουσιάζεται στο σχήμα 3.7. Όμοια η διαδικασία της μεταλλάξεως στο Grammatical Evolution δεν είναι τίποτα παραπάνω από την τυχαία αλλαγή κάποιου στοιχείου ενός χρωμοσώματος σε άλλη τιμή.

Έχει παρατηρηθεί γενικά στις διασταυρώσεις της Γραμματικής Εξελίξεως και ειδικά όταν το μήκος των χρωμοσωμάτων παραμένει σταθερό, πως κάποια μέρη των χρωμοσωμάτων που δεν χρησιμοποιούνται για την παραγωγή εκφράσεων στην αντικειμενική γλώσσα προγραμματισμού. Αυτό μπορεί να συμβεί διότι μετά από μία διασταύρωση αυτά τα τμήματα του χρωμοσώματος να μπουν στις πρώτες θέσεις κάποιου χρωμοσώματος και έτσι να είναι άμεσα διαθέσιμα για παραγωγή εκφράσεων είτε να μεταφερθούν σε τέτοια περιοχή που δεν θα μπορούν να χρησιμοποιηθούν. Για παράδειγμα έστω πως διατίθενται χρωμοσώματα των 10 στοιχείων και έχει παρατηρηθεί πειραματικά πως δεν χρησιμοποιούνται ποτέ περισσότερο από τα 5 πρώτα στοιχεία των χρωμοσωμάτων για την παραγωγή εκφράσεων. Έστω το χρωμόσωμα [1,3,2,11,100,20,10,15,20,25]. Με την χρήση

Σχήμα 3.7: Διασταύρωση ενός σημείου σε gramamatical evolution.



της διαδικασίας της διασταυρώσεως ακόμα και τα στοιχεία 20,10,15,20,25 που δεν θα είχαν καμία πιθανότητα να χρησιμοποιηθούν μπορούν σε κάποια διασταύρωση να βρεθούν στις πρώτες θέσεις του χρωμοσώματος. Επίσης πεντάδες που είναι στην αρχή ενός χρωμοσώματος μπορούν να βρεθούν στο τέλος. Από την άλλη η διαδικασία της μεταλλάξεως παρά τις μικρές τυχαίες αλλαγές της που προκαλεί μπορεί να αλλάξει το πλήρως, καθώς το αποτέλεσμα της εφαρμογής ενός γονιδίου δεν εξαρτάται μόνον από τα στοιχεία που αυτό περιέχει αλλά και από την σειρά με την οποία εφαρμόζονται. Τέλος επειδή η μετάλλαξη είναι τυχαία αλλαγή ενός ακεραίου πολλές φορές μπορεί να παρουσιαστεί το φαινόμενο μία μετάλλαξη να μην προκαλεί αλλαγές στο αποτέλεσμα ενός χρωμοσώματος. Έστω για παράδειγμα πως σε μια γλώσσα με 5 κανόνες αλλάζουμε το στοιχείο με τιμή 10 σε ένα γονίδιο, δίνοντάς του την τιμή 100. Αν και η αλλαγή που προκλήθηκε ήταν εξαιρετικά σημαντική δεν θα προκληθεί καμία αλλαγή στην συμπεριφορά του χρωμοσώματος, καθώς το ακέραιο υπόλοιπο της διαιρέσεως του 10 με το 5 είναι 0, όπως και του 100 με το 5.

3.5 Ο αλγόριθμος Genetic Random Search (GRS)

Ο αλγόριθμος Genetic Random Search (GRS) είναι μια στοχαστική διαδικασία που μέσα από γενετικό προγραμματισμό και ειδικότερα μέσα από Γραμματική Εξέλιξη, προσπαθεί να εντοπίσει το ολικό ελάχιστο μιας συναρτήσεως. Ο αλγόριθμος ξεκινά από ένα αρχικό σημείο στον χώρο αναζήτησεως. Στον αλγό-

Σχήμα 3.8: Η γραμματική της μεθόδου GRS.

```
S::=<expr>

<expr>:=(<expr>
  |<expr><binary_op><expr>
  |<func_op>(<expr>)
  |<terminal>)
<binary_op>::=+|-|*|/
<func_op>::=sin | cos | exp | log
<terminal>::=<digitlist>.<digitlist>
  |x
<digitlist>::=<digit>
  |<digit><digit>
  |<digit><digit><digit>
<digit>::=0|1|2|3|4|5|6|7|8|9
```

ριθμο χρησιμοποιούνται χρωμοσώματα που αντιπροσωπεύουν μετατοπίσεις από το αρχικό σημείο. Το χρωμόσωμα που απεικονίζει μετατόπιση που οδηγεί στην χαμηλότερη τιμή της αντικειμενικής συναρτήσεως θεωρείται πως έχει και την μεγαλύτερη τιμή καταλληλότητας. Τα χρωμοσώματα αυτά υπόκεινται σε διασταυρώσεις και μεταλλάξεις δημιουργώντας γενιές από μετατοπίσεις που μετακινούν το αρχικό σημείο σε σημεία με πιο χαμηλή τιμή της αντικειμενικής συναρτήσεως. Όταν πλέον η διαδικασία αυτή δεν μπορεί να επιφέρει άλλη αποδοτική μετακίνηση η μέθοδος τερματίζει. Η διαδικασία αυτή από μόνη της δεν μπορεί να δώσει ικανοποιητικά αποτελέσματα σε πολύπλοκες συναρτήσεις, αλλά όπως θα δούμε όταν χρησιμοποιηθεί μέσα από περισσότερο εξελιγμένες τεχνικές μπορεί να βελτιώσει την απόδοσή τους.

3.5.1. Προτεινόμενη γραμματική

Η γραμματική που θα χρησιμοποιηθεί από την διαδικασία της Γραμματικής Εξελίξεως για την μέθοδο GRS είναι ένα μικρό τμήμα της γραμματικής της γλώσσας προγραμματισμού C και εκφραζόμενη σε μορφή BNF παρουσιάζεται στο σχήμα 3.8. Όπως παρατηρούμε η χρησιμοποιούμενη γραμματική χρησιμοποιεί τέσσερις συναρτήσεις και αριθμητικές σταθερές μέχρι και 3 ψηφία, ωστόσο είναι σχετικά απλό να επεκταθεί ώστε να συμπεριλάβει περισσότερες συναρτήσεις και μεγαλύτερους σε πλήθος ψηφίων αριθμούς.

3.5.2. Περιγραφή του αλγορίθμου

Τα βήματα του αλγορίθμου Genetic Random Search (GRS) έχουν ως εξής:

Είσοδοι:

- Ένα σημείο $x = (x_1, x_2, \dots, x_n) \in S \subset R^n$.
- ϵ , ένας μικρός θετικός αριθμός. Τυπικές τιμές για αυτήν την παράμετρο είναι 10^{-4} , 10^{-5} κτλ.
- k , ένας ακέραιος αριθμός που συνήθως είναι στο διάστημα $[10, 20]$.

Αρχικοποίηση:

- Αρχικοποίηση του γενετικού πληθυσμού. Η αρχικοποίηση πραγματοποιείται επιλέγοντας έναν τυχαίο αριθμό στο διάστημα $[0, 255]$ για κάθε στοιχείου κάθε χρωμοσώματος στον γενετικό πληθυσμό.

Επανάληψη:

- Για $i = 1, \dots, k$ Επανάλαβε
 - Θέσε $x_{\text{old}} = x$.
 - **Δημιουργία** μίας νέας γενιάς χρησιμοποιώντας τους γενετικούς τελεστές της επιλογής, της διασταυρώσεως και της μεταλλάξεως.
 - Για κάθε χρωμόσωμα στον γενετικό πληθυσμό **Κάνε**
 - * **Διαχωρισμός** του χρωμοσώματος ομοιόμορφα σε n τμήματα, ένα για κάθε διάσταση του ορίσματος της αντικειμενικής συναρτήσεως. Κάθε τμήμα αντιστοιχεί σε μία τυχαία κίνηση και δηλώνεται ως p_i , $i = 1, \dots, n$. Σε κάθε τμήμα p_i εφαρμόζεται η διαδικασία του μετασχηματισμού με την χρήση της μεθόδου της Γραμματικής Εξελίξεως και με την γραμματική που παρουσιάστηκε προηγούμενες. Το αποτέλεσμα αυτής της εφαρμογής είναι η μονοδιάστατη συνάρτηση f_i .
 - * **Υπολογίζεται** το διάνυσμα $d \in R^n$ ως $d_i = f_i(x_i)$, $i = 1, \dots, n$
 - * **Θέσε** $x_+ = x + d$.
 - * **Θέσε** $y = f(x)$.
 - * **Αν** $x_+ \notin S$ ή $f(x_+) > y$ τότε
 - **Θέσε** $x_- = x - d$.
 - **Αν** $x_- \notin S$ ή $f(x_-) > y$, τότε
 - Θέσε την καταλληλότητα σε μια πολύ μεγάλη τιμή.

- Διαφορετικά
 - Θέσε την καταλληλότητα σε $f(x_-)$.
 - Τέλος - Αν
 - * Διαφορετικά
 - Θέσε την καταλληλότητα σε $f(x_+)$.
 - * Τέλος - Αν
 - Τέλος Επαναλήψεως
 - Θέσε $x = x + d_{\text{best}}$, όπου d_{best} η μετακίνηση που αντιστοιχεί στο χρωμόσωμα με την καλύτερη τιμή καταλληλότητας (μικρότερη συναρτησιακή τιμή).
 - Αν $|x - x_{\text{old}}| \leq \epsilon$, τερματισμός του αλγορίθμου και επέστρεψε το x .
- Τέλος Επαναλήψεως

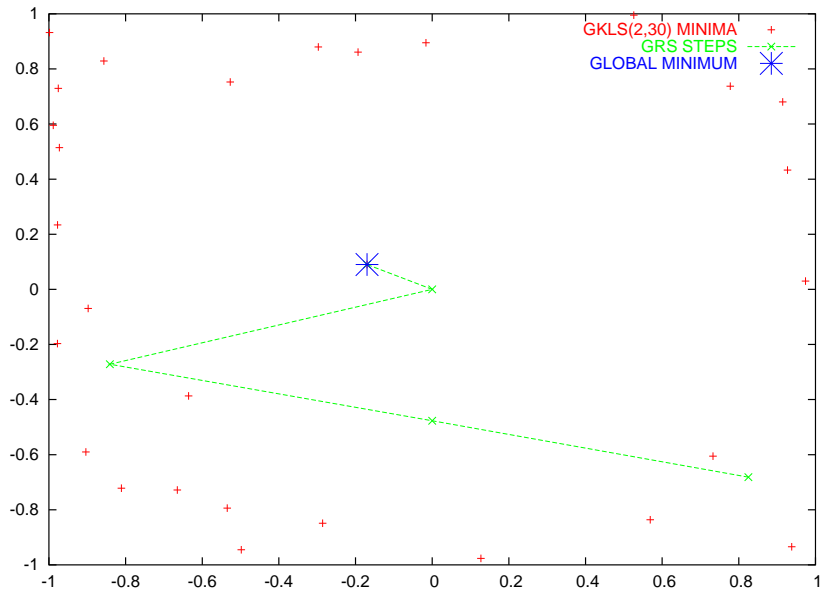
3.5.3. Παράδειγμα εφαρμογής

Στο σχήμα 3.9 παρουσιάζονται τα βήματα που κάνει η μέθοδος προκειμένου να βρεθεί στο ολικό ελάχιστο της συναρτήσεως GKLS(2,30), δηλαδή στην συνάρτηση GKLS σε διάσταση 2 και με 30 συνολικά ελάχιστα. Το πλήθος των χρωμοσωμάτων είναι 50. Το σημείο που επισημαίνεται με αστερίσκο στο γράφημα είναι το ολικό ελάχιστο της συναρτήσεως. Ξεκινώντας την έρευνα από διάφορα σημεία του πεδίου ορισμού της συναρτήσεως, (ακόμα και από τοπικά ελάχιστα της συναρτήσεως) παίρνουμε ένα γράφημα σαν και αυτό του σχήματος 3.10. Στο γράφημα οι ετικέτες RUN1, RUN2 και RUN3 αναφέρονται στις διαφορετικές εκτελέσεις της προτεινομένης μεθόδου, ξεκινώντας κάθε φορά από διαφορετικά σημεία. Όπως μπορούμε να παρατηρήσουμε η προτεινόμενη μέθοδος κατάφερε να αποφύγει πολλά από τα τοπικά ελάχιστα προκειμένου να οδηγηθεί σε αυτό με την χαμηλότερη τιμή.

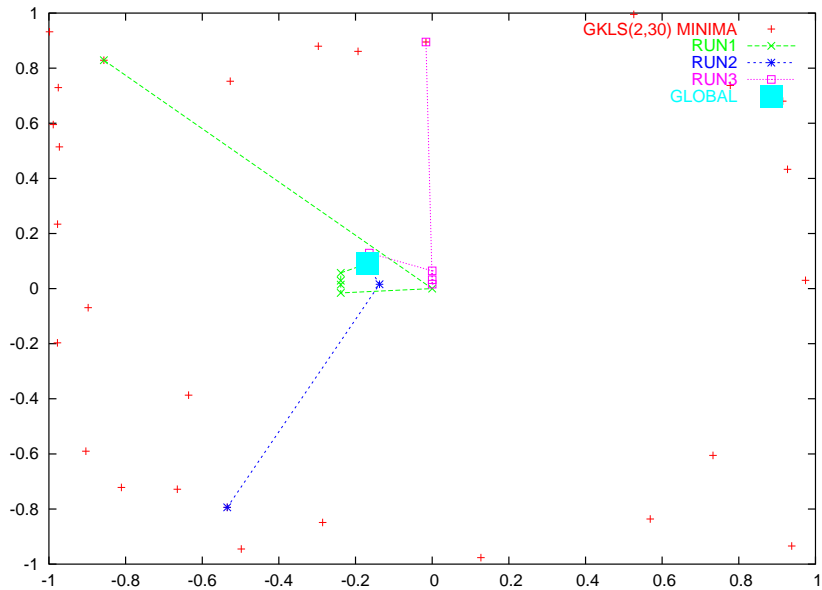
3.6 Η γενετικώς τροποποιημένη μέθοδος CRS (GenPrice)

Η μέθοδος Controlled Random Search [110, 111, 112, 113] είναι ένας αλγόριθμος που βασίζεται σε έναν πληθυσμό από λύσεις και έχει εφαρμοστεί με ιδιαίτερη επιτυχία σε μια σειρά από προβλήματα [114]. Ο αλγόριθμος CRS δεν απαιτεί η συνάρτηση να είναι παραγωγίσιμη. Ο αλγόριθμος ξεκινά λαμβάνοντας N σημεία στον χώρο αναζήτησεως και αποτιμώντας την τιμή της αντικειμενικής συναρτήσεως σε καθένα από αυτά τα σημεία. Σε κάθε επανάληψη ένα νέο σημείο λαμβάνεται του οποίου οι συντεταγμένες είναι σχετικές ως προς αυτές των υπολοίπων

Σχήμα 3.9: Βήματα για το ολικό ελάχιστο της GKLS(2,30).



Σχήμα 3.10: Εύρεση του ολικού ελαχίστου της GKLS(2,30) ξεκινώντας από διαφορετικά σημεία.



σημείων. Στην συνέχεια ελέγχεται αν αυτό το σημείο έχει μικρότερη συναρτησιακή τιμή από το σημείο με την υψηλότερη τιμή στο σύνολο δειγμάτων. Αν αυτό ισχύει το σημείο με την υψηλότερη συναρτησιακή τιμή αντικαθίσταται από το νέο σημείο και ο αλγόριθμος συνεχίζει. Με την πάροδο του χρόνου τα σημεία τείνουν να επικεντρωθούν γύρω από το καθολικό ελάχιστο, αν και αυτό εξαρτάται σε μεγάλο βαθμό από το πλήθος των αρχικών σημείων και την πολυπλοκότητα της αντικειμενικής συναρτήσεως. Τα βήματα του αλγορίθμου CRS έχουν ως εξής:

Αρχικοποίηση:

- **Θέσε** την παράμετρο N που καθορίζει τα δείγματα της αντικειμενικής συναρτήσεως που θα ληφθούν με την εκκίνηση του αλγορίθμου. Μια πολύ κοινά χρησιμοποιούμενη τιμή είναι $N = 25n$.
- **Θέσε** σε μια μικρή θετική τιμή την παράμετρο ϵ , που χρησιμοποιείται στο κριτήριο τερματισμού της μεθόδου.
- **Δημιουργία** του συνόλου $T = \{x_1, x_2, \dots, x_N\}$, παίρνοντας N τυχαία δείγματα στον 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

$$f_{\max} = \max_{x \in T} f(x)$$

και

$$f_{\min} = \min_{x \in T} f(x)$$

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

Νέο_Σημείο:

- **Επιλογή** με τυχαίο τρόπο του συνόλου $\tilde{T} = \{x_{T_1}, x_{T_2}, \dots, x_{T_{n+1}}\}$ από το αρχικό σύνολο T .
- **Υπολογισμός** του κεντροϊδούς G σύμφωνα με την σχέση

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

- **Δημιουργία** ενός δοκιμαστικού σημείου $\tilde{x} = 2G - x_{T_{n+1}}$.
- **Αν** $\tilde{x} \notin S$ **ή** $f(\tilde{x}) \geq f_{\max}$ **Τότε** εκτέλεση και πάλι του βήματος **Νέο_Σημείο**.

Ενημέρωση:

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

Ελαχιστοποίηση:

- $x^* = \text{localSearch}(x)$.
- Επιστροφή του σημείου x^* σαν το εντοπισμένο τοπικό ελάχιστο.

Ο παραπάνω αλγόριθμος είναι ο αρχικός αλγόριθμος Controlled Random Search με την προσθήκη μιας τοπικής μεθόδου ελαχιστοποίησης μετά το τέλος των απαιτούμενων επαναλήψεων, προκειμένου να διασφαλισθεί πως το ελάχιστο που θα βρει ο αλγόριθμος θα είναι πραγματικά ένα τοπικό ελάχιστο. Ο προτεινόμενος αλγόριθμος GenPrice προσθέτει ένα ακόμα βήμα στον παραπάνω αλγόριθμο, στο τέλος του βήματος που τιτλοφορείται NEO_ΣΗΜΕΙΟ: Μετά την εύρεση του δοκιμαστικού σημείου \tilde{x} , εκτελείται μια κλήση στον αλγόριθμο GRS με είσοδο αυτό το σημείο. Αυτό έχει σαν αποτέλεσμα το δοκιμαστικό σημείο να τροποποιηθεί έτσι ώστε να είναι πιο κοντά στο ολικό ελάχιστο και σαν συνέπεια ο αλγόριθμος να συγκλίνει ταχύτερα.

3.7 Η γενετικώς τροποποιημένη μέθοδος Simulated Annealing (GenAnneal)

Στην βιβλιογραφία υπάρχουν πολλές εκδοχές της μεθόδου της προσομοιούμενης ανοπτήσεως. Η γενετικά τροποποιημένη τεχνική που αναπτύχθηκε βασίζεται στην εκδοχή για προσομοιούμενη ανόπτηση από τους Goffe et al [115]. Ο κώδικας τους είναι γραμμένος στην γλώσσα προγραμματισμού Fortran 77 (simann.f) και διατίθεται ελεύθερα από τον ιστοχώρο

www.netlib.org

Τα βήματα του αρχικού αλγορίθμου έχουν ως ακολούθως:

Αρχικοποίηση

- **Θέσε** τις τιμές των ακεραίων παραμέτρων N_T , N_S , TLAST. Η τιμή της παραμέτρου N_T καθορίζει τον αριθμό των επαναλήψεων που θα εκτελέσει ο αλγόριθμος πριν να μειώσει την “θερμοκρασία” T . Η τιμή της παραμέτρου N_S καθορίζει τον αριθμό των επαναλήψεων που θα εκτελέσει ο αλγόριθμος πριν από την ενημέρωση του διανύσματος προσαρμογής βήματος \vec{h} . Η τιμή

της παραμέτρου TLAST ελέγχει τον τερματισμό του αλγορίθμου ως ακολούθως: αν η καλύτερη τιμή για το ολικό ελάχιστο δεν έχει βελτιωθεί στις προηγούμενες TLAST μειώσεις της “θερμοκρασίας”, τότε ο αλγόριθμος τερματίζει.

- **Καθορισμός** των αρχικών τιμών για τις τιμές του διανύσματος προσαρμογής βήματος $\{h_1, h_2, \dots, h_n\}$.
- **Θέσε** την παράμετρο ϵ σε μια πολύ μικρή θετική τιμή.
- **Αρχικοποίηση** της τιμής της “θερμοκρασίας” T .
- **Θέσε** την τιμή του παράγοντα μειώσεως θερμοκρασίας a , $a \in (0, 1)$.
- **Δειγματοληψία** με τυχαίο τρόπο ενός σημείου $x \in S$.
- **Αρχικοποίηση** $f_c = f(x)$, $x_{\text{best}} = x$, $f_{\text{best}} = f_c$.

Επανάληψη:

- **Για** $i = 1, \dots, N_T$ **Επανάλαβε**
 - **Για** $j = 1, \dots, N_S$ **Επανάλαβε**
 - * **Για** $k = 1, \dots, n$ **Επανάλαβε**
 - **Θέσε** $x^t = x$.
 - **Θέσε** $x_k^t = x_k + (2\xi - 1)h_k$, όπου ξ ένας τυχαίος αριθμός στο διάστημα $(0,1)$.
 - **Αν** $x^t \notin [a_k, b_k]$ **Τότε** **Θέσε** $x_k^t = a_k + (b_k - a_k)\xi$, όπου ξ ένας τυχαίος αριθμός στο διάστημα $(0,1)$.
 - **Θέσε** $f_t = f(x^t)$.
 - **Αν** $f_t \leq f_{\text{best}}$ **Τότε** **Θέσε** $x_{\text{best}} = x^t$, $f_{\text{best}} = f_t$.
 - **Αν** $f_t < f_c$ **Τότε** **Θέσε** $x = x^t$, $f_c = f_t$ **Αλλιώς** **Αν** $e^{-\frac{f_t - f_c}{T}} > \xi$ **Τότε** **Θέσε** $x = x^t$, $f_c = f_t$. ξ ένας τυχαίος αριθμός στο διάστημα $(0,1)$.
 - * **Τέλος** **Επαναλήψεων**
 - **Τέλος** **Επαναλήψεων**
 - **Υπολογισμός** $r_i = \frac{M_i}{N_S}$, όπου M_i είναι ο αριθμός των αποδεκτών κινήσεων στην κατεύθυνση x_i στις τελευταίες N_S προσπάθειες. Μια κίνηση θεωρείται αποδεκτή αν βελτιώνει το μέχρι τώρα καθολικό ελάχιστο.
 - **Αν** $r_i > \frac{3}{5}$ **θέσε** : $h_i = h_i \left(1 + 2\frac{r_i - 3/5}{2/5}\right)$

- Αν $r_i < \frac{2}{5}$ θέσει : $h_i = \frac{h_i}{1+2^{\frac{2/\delta-r_i}{2/\delta}}}$
- Αν $h_i > b_i - a_i$ θέσει: $h_i = b_i - a_i$.

- **Τέλος Επαναλήψεων**

Ενημέρωση:

- **Ενημέρωση** της “θερμοκρασίας” σύμφωνα με το σχήμα $T = aT$.
- **Ενημέρωση** $x = x_{\text{best}}$, $f_c = f_{\text{best}}$.
- Αν δεν υπήρξε βελτίωση στην τιμή του ελαχίστου στις τελευταίες TLAST επαναλήψεις ή η “θερμοκρασία” T έπεσε κάτω από το όριο ϵ , **Μετάβαση** στο βήμα **Ελαχιστοποίηση**.
- **Μετάβαση** στο βήμα **Επανάληψη**.

Ελαχιστοποίηση:

- $x^* = \text{localSearch}(x)$.
- Επιστροφή του ελαχίστου x^* .

Ο αλγόριθμος εκτελεί μια τοπική ελαχιστοποίηση στο τέλος προκειμένου να βρεθεί με βεβαιότητα ένα τοπικό ελάχιστο της αντικειμενικής συναρτήσεως. Ο προτεινόμενος αλγόριθμος (GenAnneal) είναι αισθητά πιο απλός από τον παραπάνω. Δεν διαθέτει την πληθώρα επαναλήψεων του προηγούμενου αλγορίθμου και χρησιμοποιεί την μέθοδο GRS προκειμένου να κινηθεί ταχύτερα προς το ολικό ελάχιστο. Τα βασικά βήματα του αλγορίθμου έχουν ως ακολούθως:

Αρχικοποίηση

- **Θέσει** τις αρχικές τιμές των παραμέτρων TLAST, T και ϵ με την ίδια σημασία όπως προηγουμένως.
- **Θέσει** την τιμή του παράγοντα μείωσης θερμοκρασίας a , $a \in (0, 1)$.
- **Δειγματοληψία** με τυχαίο τρόπο ενός σημείου $x \in S$.
- **Αρχικοποίηση** $f_c = f(x)$, $x_{\text{best}} = x$, $f_{\text{best}} = f_c$.

Επανάληψη

- Εφαρμογή της μεθόδου GRS στο σημείο x . Έστω \tilde{x} το σημείο που προκύπτει.

- Αν $f(\tilde{x}) < f_c$ τότε
 - Θέσε $x = \tilde{x}$
 - Θέσε $f_c = f(\tilde{x})$
 - Αν $f_c < f_{\text{best}}$ τότε
 - * Θέσε $x_{\text{best}} = x$
 - * Θέσε $f_{\text{best}} = f_c$
 - Τέλος - Αν
- Αλλιώς
 - Αν $e^{\frac{f(\tilde{x})-f_c}{T}} > \xi$, ξ τυχαίος στο διάστημα $(0,1)$ τότε
 - * Θέσε $x = \tilde{x}$
 - * Θέσε $f_c = f(\tilde{x})$
 - Τέλος - Αν
- Τέλος - Αν

Ενημέρωση:

- Ενημέρωση της “θερμοκρασίας” σύμφωνα με το σχήμα $T = aT$.
- Ενημέρωση $x = x_{\text{best}}$, $f_c = f_{\text{best}}$.
- Αν δεν υπήρξε βελτίωση στην τιμή του ελαχίστου στις τελευταίες TLAST επαναλήψεις ή η “θερμοκρασία” T έπεσε κάτω από το όριο ϵ , **Μετάβαση** στο βήμα **Ελαχιστοποίηση**.
- **Μετάβαση** στο βήμα **Επανάληψη**.

Ελαχιστοποίηση:

- $x^* = \text{localSearch}(x)$.
- Επιστροφή του ελαχίστου x^* .

3.8 Πειραματικά αποτελέσματα

Οι προτεινόμενες τροποποιήσεις των στοχαστικών μεθόδων συγκρίθηκαν με τις αρχικές εκδόσεις των αλγορίθμων σε μια σειρά από γνωστές συναρτήσεις. Κάθε μέθοδος δοκιμάστηκε σε κάθε συνάρτηση 50 φορές με διαφορετική γεννήτρια τυχαίων αριθμών κάθε φορά. Μετρήθηκε τόσο η ικανότητα κάθε μεθόδου να βρಿಸκει

το ολικό ελάχιστο όσο και ο αριθμός των συναρτησιακών κλήσεων που απαιτείται για την συγκεκριμένη εργασία. Σε όλες τις περιπτώσεις ο ρυθμός επιλογής ήταν στο 90% και ο ρυθμός μεταλλάξεως στο 5%. Το μήκος των χρωμοσωμάτων για την μέθοδο GRS ήταν $10 \times n$. Τα βήματα που παρήχθησαν από την διαδικασία της Γραμματικής Εξελίξεως αποτιμήθηκαν με την χρήση της βιβλιοθήκης προγραμματισμού FunctionParser [105].

3.8.1. Αποτελέσματα CRS

Για την σύγκριση της αρχικής μεθόδου CRS και της βελτιωμένης μεθόδου CRS χρησιμοποιήθηκε η προτεινόμενη [111] τιμή $N = 25n$ για τον αρχικό πληθυσμό των μεθόδων. Η GRS χρησιμοποιήθηκε στην βελτιωμένη εκδοχή με 10 χρωμοσώματα σαν πληθυσμό και μέγιστο αριθμό γενιών 10. Στον πίνακα 3.2 παρατίθενται τα αποτελέσματα για την απλή μέθοδο CRS, την διαφοροποίηση της CRS από το πακέτο PANMIN [116] (PCRS) και την προτεινόμενη διαφοροποίηση της γενετικά ελεγχόμενης CRS (GCRS). Οι αριθμοί εντός των παρενθέσεων δηλώνουν το ποσοστό των πειραμάτων για τα οποία η μέθοδος κατόρθωσε να βρει το ολικό ελάχιστο και δεν εγκλωβίστηκε σε κάποιο τοπικό ελάχιστο. Η απουσία ενός τέτοιου ποσοστού δηλώνει επιτυχία 100%. Η προτεινόμενη τεχνική παρουσιάζει μεγαλύτερο ποσοστό επιτυχίας και μικρότερες απαιτήσεις σε συναρτησιακές κλήσεις ειδικά στα προβλήματα πολλών διαστάσεων όπως είναι το τεχνητό νευρωνικό δίκτυο ή οι περιπτώσεις POTENTIAL.

3.8.2. Αποτελέσματα προσομοιούμενης ανοπτήσεως

Για την αξιολόγηση του αλγορίθμου GenAnneal εκτελέστηκαν μια σειρά από συγκριτικά πειράματα ανάμεσα στο προτεινόμενο αλγόριθμο και τους αλγορίθμους του Goffe et al (SA) [115] και του Ingber(ASA) [117]. Για την περίπτωση του αλγορίθμου SA χρησιμοποιήθηκαν οι προτεινόμενοι παράμετροι στον σχετικό ιστοχώρο <http://www.netlib.org>:

1. $N_S = 20$
2. $N_T = 5$
3. $T = 5.0$
4. $a = 0.5$
5. TLAST = 4

Για την περίπτωση του αλγορίθμου ASA χρησιμοποιήθηκαν οι προτεινόμενοι παράμετροι όπως αυτές εμφανίζονται στο λογισμικό που διατίθεται στον ιστοχώρο

Πίνακας 3.2: Συγκριτικά αποτελέσματα διαφορετικών εκδοχών της Controlled Random Search σε μια σειρά από γνωστά προβλήματα καθολικής ελαχιστοποίησης.

FUNCTION	CRS	PCRS	GCRS
CAMEL	1852	1409	1504
RASTRIGIN	1903	1982	428
GRIEWANK2	2105	2004	977
GKLS(2,50)	1627	1495	1220
GKLS(3,50)	3349	3059	2056
GOLDSTEIN	1923	1456	961
TEST2N(4)	6835(0.97)	4831	4280(0.97)
TEST2N(5)	25270(0.97)	12342	7958
TEST2N(6)	32801(0.70)	8840(0.87)	9914
TEST2N(7)	38057(0.40)	11751(0.63)	9740
TEST30N(3)	3703	2124	1519
TEST30N(4)	5135	4058	1416
POTENTIAL(3)	198046	34985	9265
POTENTIAL(5)	188646	39305	9096
NEURAL	122617	94016	14559

Πίνακας 3.3: Αποτελέσματα από την εφαρμογή των μεθόδων SA, ASA και GenAnneal σε μια σειρά από συναρτήσεις.

FUNCTION	SA	ASA	GenAnneal
CAMEL	4820	3125	1791
RASTRIGIN	4843	3534	488
GRIEWANK2	4832(0.27)	3271(0.43)	580
GKLS(2,50)	4820	3354	1641
GKLS(3,50)	7228	5269(0.17)	2004
GOLDSTEIN	4842	3385(0.93)	1281
TEST2N(4)	9631	6460	2923
TEST2N(5)	12034(0.87)	10763	3456
TEST2N(6)	14438(0.66)	18466	3633
TEST2N(7)	16840(0.37)	29972	3840
TEST30N(3)	7930(0.23)	3220	1425
TEST30N(4)	9858(0.23)	6002	1001
POTENTIAL(3)	21404	50202	3075
POTENTIAL(5)	36212	80527	2770
NEURAL	76667(0.93)	67167	6241(0.93)

<http://www.ingber.com> Το πλήθος των χρωμοσωμάτων για την μέθοδο GRS ήταν 100 και ο μέγιστος αριθμός γενιών επίσης 10. Και σε αυτήν την περίπτωση όπως μπορούμε να παρατηρήσουμε στον πίνακα 3.3 η προτεινόμενη μέθοδος (GSA) εμφανίζεται να είναι περισσότερο αποτελεσματική και να απαιτεί λιγότερες συναρτησιακές κλήσεις από τις άλλες μεθόδους.

3.9 Υλοποιημένο λογισμικό

Για τις δύο μεθόδους GenPrice και GenAnneal υλοποιήθηκαν δύο αντίστοιχα πακέτα λογισμικού, τα οποία διαθέτουν την ίδια λειτουργικότητα, τρόπο εγκατάστασης και χειρισμό. Στα πακέτα αυτά η κωδικοποίηση των συναρτήσεων γίνεται με τα πρότυπα του πακέτου MinFinder και ο χρήστης μέσω παραμέτρων της γραμμής εντολών μπορεί να αλλάξει παραμέτρους του αλγόριθμου ή/και της διαδικασίας GRS, προσαρμόζοντας έτσι τον αλγόριθμο στο πρόβλημα που αντιμετωπίζεται κάθε φορά. Στο παράρτημα Γ παρατίθενται τα βασικά στοιχεία αυτών των πακέτων.

ΚΕΦΑΛΑΙΟ 4

ΠΡΟΣΕΓΓΙΣΗ ΣΥΝΑΡΤΗΣΕΩΝ ΜΕ ΤΗΝ ΧΡΗΣΗ NEURAL SPLINES

Περιεχόμενα

4.1	Γενικά	66
4.2	Neural Splines	66
4.2.1	Ορισμός του προβλήματος	66
4.2.2	Βασικοί ορισμοί	68
4.2.3	Μονοδιάστατα προβλήματα	70
4.2.4	Προβλήματα δύο διαστάσεων	77
4.2.5	Συμπεράσματα	81

4.1 Γενικά

Στο κεφάλαιο αυτό παρουσιάζεται μια νέα τεχνική για την αντιμετώπιση του προβλήματος της προσεγγίσεως δεδομένων. Η νέα τεχνική βασίζεται σε τμηματικά νευρωνικά δίκτυα σε μια μορφή που μπορεί να παραλληλοποιηθεί χωρίς το συνδυασμένο μοντέλο να χάνει τις ιδιότητες συνέχειας στα σημεία συγκολήσεως των επιμέρους τμηματικών νευρωνικών δικτύων. Για την εκπαίδευση των τεχνητών νευρωνικών δικτύων είναι απαραίτητες οι τεχνικές καθολικής ελαχιστοποίησης.

4.2 Neural Splines

Στην ενότητα εισάγεται η έννοια των Neural Splines, που είναι ένα νέο μαθηματικό μοντέλο που συνδυάζει ένα τεχνητό νευρωνικό δίκτυο και ένα πολυώνυμο Obreshkov. Το νέο μοντέλο μπορεί να χρησιμοποιηθεί σαν βάση για τον σχεδιασμό τμηματικών μοντέλων, τα οποία διατηρούν την συνέχειά τους ακόμα και στα όρια μεταξύ των τμημάτων. Το συνολικό παράλληλο μοντέλο που προκύπτει από τον συνδυασμό των επιμέρους Neural Splines μπορεί να χρησιμοποιηθεί για την προσέγγιση συναρτήσεων και στο κεφάλαιο αυτό δίνουμε παραδείγματα από την δημιουργία τέτοιων μοντέλων για την περίπτωση μονοδιάστατων και διδιάστατων συναρτήσεων.

4.2.1. Ορισμός του προβλήματος

Η κατηγορία των τμηματικά συνεχών μοντέλων έχει ευρέως χρησιμοποιηθεί κατά το παρελθόν για την προσέγγιση συναρτήσεων και για την συναρτησιακή παρεμβολή σε τμηματικά συνεχή πεδία ορισμού. Σαν παράδειγμα μπορεί κάποιος να χρησιμοποιήσει τα Φυσικά Splines, τα B-Splines και τα Hermite Splines [65]. Αυτές οι μορφές τμηματικά συνεχών μοντέλων χρησιμοποιούν σαν βάση τους πολυώνυμα. Σε αυτό το κεφάλαιο προτείνεται μια νέα μορφή τμηματικά συνεχών μοντέλων που διαθέτει σαν βάση της τεχνητά νευρωνικά δίκτυα. Δεν είναι η πρώτη φορά που προτείνεται στην βιβλιογραφία τμηματικά συνεχές μοντέλο που δεν χρησιμοποιεί πολυώνυμο σαν βάση. Στο παρελθόν έχουν κάνει την εμφάνισή τους τα λεγόμενα Tension Splines [66], τα οποία χρησιμοποιούν σαν βάση τους εκθετικές συναρτήσεις. Τα τεχνητά νευρωνικά δίκτυα είναι γνωστά για τις δυνατότητές τους να προσεγγίζουν κάθε συνάρτηση [67, 68] και έχουν χρησιμοποιηθεί με επιτυχία σε πολύπλοκα προβλήματα, όπως συναρτησιακή παρεμβολή, προσέγγιση συναρτήσεων, αναγνώριση προτύπων [69], επεξεργασία σήματος και την επίλυση διαφορικών εξισώσεων [70, 71, 72]. Ένα τεχνητό νευρωνικό δίκτυο μπορεί να διατυπωθεί με πολλούς τρόπους ανάλογα με τον αριθμό των επιπέδων επεξεργασίας που έχει και τον αριθμό των εξόδων του, αλλά σε αυτό το κεφά-

λαιο θα ασχοληθούμε μόνον με την περίπτωση του τεχνητού νευρωνικού δικτύου πρόσθιας τροφοδοτήσεως με ένα επίπεδο επεξεργασίας και μια έξοδο. Η έξοδος ενός τέτοιου νευρωνικού δικτύου δίνεται από την σχέση:

$$N(x, p) = \sum_{i=1}^H p_{(d+2)*i-(d+1)} \text{sig} \left(\sum_{j=1}^d (p_{(d+2)*i-(d+1)+j} * x_j) + p_{(d+2)*i} \right) \quad (4.1)$$

όπου H είναι ο αριθμός των κρυμμένων νευρώνων και d η διάσταση του προτύπου εισόδου x . Το διάνυσμα p είναι το διάνυσμα των βαρών και πολώσεων του τεχνητού νευρωνικού δικτύου. Η συνάρτηση $\text{sig}(x)$ είναι η σιγμοειδής λογιστική συνάρτηση και ορίζεται ως ακολούθως:

$$\text{sig}(x) = \frac{1}{1 + \exp(-x)}$$

Τα τμηματικά συνεχή μοντέλα που θα παρουσιαστούν σε αυτό το κεφάλαιο θα χρησιμοποιηθούν για την προσέγγιση συναρτήσεων. Το πρόβλημα της προσεγγίσεως συναρτήσεων μπορεί να διατυπωθεί ως ακολούθως: *δοθέντων M σημείων και των συσχετισμένων με αυτά τιμών (x_i, y_i) , $i = 1, 2, \dots, M$ $x_i \in \mathbb{R}^N$, $y_i \in \mathbb{R}$, να κατασκευαστεί ένα μοντέλο που να προσεγγίζει αυτά τα δεδομένα.* Ο συνηθισμένος τρόπος για να επιλυθεί αυτό το πρόβλημα είναι να ορισθεί ένα παραμετρικό μοντέλο $\Psi(x, p)$ με ένα σύνολο παραμέτρων p και στην συνέχεια με την βοήθεια κάποιου αλγορίθμου βελτιστοποίησης να ελαχιστοποιηθεί το τετραγωνικό σφάλμα:

$$E_T[p] = \sum_{i=1}^M (\Psi(x_i, p) - y_i)^2 \quad (4.2)$$

ως προς το σύνολο των παραμέτρων p . Αν το μοντέλο $\Psi(x, p)$ είναι ένα τεχνητό νευρωνικό δίκτυο, τότε το σύνολο των παραμέτρων συμπίπτει με το διάνυσμα των βαρών και πολώσεων του τεχνητού νευρωνικού δικτύου. Αν διαμερίσουμε το πεδίο ορισμού D σε N τμήματα και τοποθετήσουμε ένα επιμέρους μοντέλο $\psi_i(x, p_i, q_i)$ σε κάθε τμήμα D_i , τότε το συνολικό τετραγωνικό σφάλμα μπορεί να γραφεί σαν

$$E_T = \sum_{i=1}^N \left(\sum_{x \in D_i} (\psi_i(x, p_i, q_i) - y)^2 \right) \quad (4.3)$$

Οι παράμετροι q_i χρησιμοποιούνται από κάθε επιμέρους μοντέλο, προκειμένου να εξασφαλίσουν την συνέχεια του συνολικού μοντέλου στα όρια των επιμέρους μοντέλων. Αν στην θέση των μοντέλων $\Psi(x, p_i, q_i)$ χρησιμοποιηθεί ένα τεχνητό νευρωνικό δίκτυο σαν αυτό της εξισώσεως 4.1, τότε μένει να βρεθεί ένας τρόπος καθορισμού των παραμέτρων q_i , ώστε να εξασφαλίζεται η συνέχεια του συνολικού μοντέλου. Αυτό μπορεί να επιτευχθεί με την χρήση των πολυωνύμων Obreshkon που παρουσιάζονται στην συνέχεια.

4.2.2. Βασικοί ορισμοί

4.2.2.1. Πολυώνυμα Obreshkov

Θεωρούμε μια μονοδιάστατη συνεχώς παραγωγίσιμη συνάρτηση $f(x)$, $x \in [a, b]$ και ένα πολυώνυμο $P_{a,b}^{k,m}(f, x)$ με τις ακόλουθες ιδιότητες:

$$\frac{d^j}{dx^j} P_{a,b}^{k,m}(f, a) = \frac{d^j}{dx^j} f(x)|_{x=a} \equiv f^{(j)}(a), \quad \forall j = 0, 1, \dots, k \quad (4.4)$$

$$\frac{d^j}{dx^j} P_{a,b}^{k,m}(f, b) = \frac{d^j}{dx^j} f(x)|_{x=b} \equiv f^{(j)}(b), \quad \forall j = 0, 1, \dots, m \quad (4.5)$$

Ο Obreshkov στην εργασία [73] έδειξε πως υπάρχει ένα μοναδικό πολυώνυμο βαθμού $k + m + 1$, το οποίο ικανοποιεί τις ιδιότητες των εξισώσεων 4.4 και 4.5 και αυτό μπορεί να διατυπωθεί ως ακολούθως:

$$\begin{aligned} P_{a,b}^{k,m}(f, x) = & \sum_{j=0}^k f^{(j)}(a) \frac{(x-b)^{m+1}(x-a)^j}{j!(a-b)^{m+1}} \sum_{i=0}^{k-j} \binom{m+i}{i} \frac{(x-a)^i}{(b-a)^i} + \\ & \sum_{j=0}^m f^{(j)}(b) \frac{(x-a)^{k+1}(x-b)^j}{j!(b-a)^{k+1}} \sum_{i=0}^{m-j} \binom{k+i}{i} \frac{(x-b)^i}{(a-b)^i} \end{aligned} \quad (4.6)$$

Έχοντας ορίσει τα πολυώνυμα Obreshkov μπορούμε να ορίσουμε έναν νέο τελεστή $L_{x \in [a,b]}^{k,m}$ ο οποίος εφαρμόζεται σε συνεχώς παραγωγίσιμες συναρτήσεις $f(x)$, $x \in [a, b]$ ως εξής:

$$L_{x \in [a,b]}^{k,m} f(x) = P_{a,b}^{k,m}(f, x) \quad (4.7)$$

Με βάση τον παραπάνω τελεστή ορίζουμε την συνάρτηση $S_{a,b}^{k,m}(f, x)$ ως:

$$S_{a,b}^{k,m}(f, x) \equiv f(x) - P_{a,b}^{k,m}(f, x) = \left(1 - L_{x \in [a,b]}^{k,m}\right) f(x) \quad (4.8)$$

Η συνάρτηση $S_{a,b}^{k,m}(f, x)$ έχει τις εξής ιδιότητες:

1. Στο σημείο $x = a$ μηδενίζεται η τιμή της συναρτήσεως αλλά και οι τιμές όλων των παραγώγων της μέχρι και τάξεως k .
2. Στο σημείο $x = b$ μηδενίζεται η τιμή της συναρτήσεως αλλά και οι τιμές όλων των παραγώγων της μέχρι και τάξεως m .

Οι ιδιότητες αυτές προσδίδουν στην συνάρτηση $S_{a,b}^{k,m}(f, x)$ συμπεριφορά παρόμοια των B-splines. Ορίζουμε την συνάρτηση $B_{a,b}^{k,m}(f, x)$ σύμφωνα με την σχέση:

$$\begin{aligned} B_{a,b}^{k,m}(f, x) & \equiv f(x) - S_{a,b}^{k,m}(f, x) = \left(1 - \left(1 - L_{x \in [a,b]}^{k,m}\right)\right) f(x) \\ & = L_{x \in [a,b]}^{k,m} f(x) = P_{a,b}^{k,m}(f, x) \end{aligned} \quad (4.9)$$

Στα σύνορα $[a, b]$ η συνάρτηση $B_{a,b}^{k,m}(f, x)$ έχει την ίδια τιμή με την συνάρτηση $f(x)$ και για αυτόν τον λόγο μπορούμε να την αποκαλέσουμε συνοριακή συνάρτηση. Στις δύο διαστάσεις, όπου το διάστημα εφαρμογής των τελεστών γίνεται το $[a_1, b_1] \otimes [a_2, b_2]$, οι ορισμοί των εξισώσεων 4.8 και 4.9 μπορούν να γενικευτούν σε:

$$S(f, x_1, x_2) = \left(1 - L_{x_1 \in [a_1, b_1]}^{k_1, m_1}\right) \left(1 - L_{x_2 \in [a_2, b_2]}^{k_2, m_2}\right) f(x_1, x_2) \quad (4.10)$$

και αντίστοιχα

$$\begin{aligned} B(f, x_1, x_2) &= f(x_1, x_2) - S(f, x_1, x_2) \\ &= \left(L_{x_1 \in [a_1, b_1]}^{k_1, m_1} + L_{x_2 \in [a_2, b_2]}^{k_2, m_2} - L_{x_1 \in [a_1, b_1]}^{k_1, m_1} L_{x_2 \in [a_2, b_2]}^{k_2, m_2}\right) f(x_1, x_2) \end{aligned} \quad (4.11)$$

Η επέκταση των συναρτήσεων σε περισσότερες από δύο διαστάσεις μπορεί να γίνει ακολουθώντας την λογική των εξισώσεων (4.10) και (4.11). Αν η συνάρτηση $N(x, p)$ ορίζει ένα νευρωνικό δίκτυο με μια είσοδο x και διάνυσμα παραμέτρων p , τότε η συνάρτηση $S(N, x)$ ορίζει αυτό που στην συνέχεια θα αποκαλούμε **Neural Spline**.

4.2.2.2. Neural Splines

Σε κάθε υποδιάστημα $D_i = [a_i, b_i]$ του αρχικού διαστήματος $D = [a, b]$ το προτεινόμενο μοντέλο θα έχει την μορφή:

$$\psi_i(x, p_i, q_i) = B_{x \in [a_i, b_i]}^{k, m}(f, x) + S_{x \in [a_i, b_i]}^{l, n}(N, x) \quad (4.12)$$

όπου με f ορίζουμε την συνάρτηση που θέλουμε να προσεγγίσουμε. Το μοντέλο της σχέσεως 4.12 ικανοποιεί εκ κατασκευής τις επόμενες συνθήκες:

$$\frac{d^j}{dx^j} \psi_i(x, p_i, q_i) \Big|_{x=a} = f^{(j)}(a), \quad j = 0, \dots, \min(k, l) \quad (4.13)$$

$$\frac{d^j}{dx^j} \psi_i(x, p_i, q_i) \Big|_{x=b} = f^{(j)}(b), \quad j = 0, \dots, \min(m, n) \quad (4.14)$$

όπου οι τιμές $f^{(j)}(a)$ και $f^{(j)}(b)$ αποτελούν τα στοιχεία του διανύσματος παραμέτρων q_i και αντιστοιχούν στις παραγώγους j τάξεως της προσεγγιζομένης συναρτήσεως. Αν για παράδειγμα θέσουμε $k = l = m = n = 0$ στην εξίσωση 4.12, τότε προκύπτει το επόμενο μοντέλο με συνέχεια τιμής στα άκρα του διαστήματος $[a_i, b_i]$:

$$\begin{aligned} \psi_i(x, p_i, q_i) &= f(a_i) \frac{x - b_i}{a_i - b_i} + f(b_i) \frac{x - a_i}{b_i - a_i} + \\ &\quad \left\{ N(x, p_i) - \left(N(a_i, p_i) \frac{x - b_i}{a_i - b_i} + N(b_i, p_i) \frac{x - a_i}{b_i - a_i} \right) \right\} \end{aligned} \quad (4.15)$$

Σε αυτήν την περίπτωση το διάνυσμα παραμέτρων q_i αποτελείται από τις τιμές $f(a_i)$ και $f(b_i)$. Αν υπάρχει η απαίτηση το προτεινόμενο μοντέλο να έχει και συνέχεια ως προς την πρώτη του παράγωγο, τότε χρησιμοποιώντας τις τιμές $k = l = m = n = 1$ στην εξίσωση 4.12 προκύπτει το επόμενο μοντέλο:

$$\begin{aligned} \psi_i(x, p_i, q_i) &= f^{(0)}(a_i) \pi_{3,0}(x, a_i, b_i) + f^{(1)}(a_i) \pi_{3,1}(x, a_i, b_i) \\ &+ f^{(0)}(b_i) \tau_{3,0}(x, a_i, b_i) + f^{(1)}(b_i) \tau_{3,1}(x, a_i, b_i) \\ &+ \{N(x, p_i) - (N(a_i, p_i) \pi_{3,0}(x, a_i, b_i) + N^{(1)}(a_i, p_i) \pi_{3,1}(x, a_i, b_i) \\ &+ N(b_i, p_i) \tau_{3,0}(x, a_i, b_i) + N^{(1)}(b_i, p_i) \tau_{3,1}(x, a_i, b_i))\} \quad (4.16) \end{aligned}$$

όπου έχουν χρησιμοποιηθεί οι επόμενες συναρτήσεις:

$$\begin{aligned} \pi_{1,0}(x, a, b) &= \frac{x-b}{a-b} \\ \pi_{3,0}(x, a, b) &= \frac{(x-b)^2}{(a-b)^2} \left(1 + 2 \frac{x-a}{b-a} \right) \\ \pi_{3,1}(x, a, b) &= (x-a) \frac{(x-b)^2}{(a-b)^2} \\ \tau_{2k+1,j}(x, a, b) &= \pi_{2k+1,j}(x, b, a) \end{aligned}$$

Το σύνολο παραμέτρων q_i στην εξίσωση (4.16) έχει ως στοιχεία τα $q_i = \{f^{(0)}(a_i), f^{(1)}(a_i), f^{(0)}(b_i), f^{(1)}(b_i)\}$.

4.2.3. Μονοδιάστατα προβλήματα

4.2.3.1. Ορισμός και εκπαίδευση

Ο αλγόριθμος εκπαίδευσης στην μονοδιάστατη περίπτωση μπορεί να διατυπωθεί ως ακολούθως:

Αρχικοποίηση

- Διαμερισμός του πεδίου ορισμού D σε N μη επικαλυπτόμενα τμήματα που ορίζονται ως $D_i = [t_i, t_{i+1}]$, $i = 1, \dots, N$.
- Θέσε $q_i = \{f_i^{(0)}, f_i^{(1)}, \dots, f_i^{(k)}\}$ που θα αποτελέσουν προσεγγίσεις της τιμής της συναρτήσεως και των k παραγώγων της σε κάθε σημείο διαχωρισμού t_i .
- Δημιουργία ενός ξεχωριστού μοντέλου $\psi_i(x, p_i, q_i)$ σε κάθε διάστημα D_i σύμφωνα με την εξίσωση 4.12. Αρχικοποίηση των τιμών των βαρών p_i κάθε επιμέρους τεχνητού νευρωνικού δικτύου.

Εκπαίδευση

- Για $i = 1, 2, \dots, N$ Επανέλαβε

– **Ελαχιστοποίηση** του τετραγωνικού σφάλματος

$$E_i = \sum_{x \in [t_i, t_{i+1}]} (\psi_i(x, p_i, q_i) - f(x))^2$$

κάθε επιμέρους μοντέλου $\psi_i(x, p_i, q_i)$ ως προς το σύνολο των παραμέτρων p_i . Οι παράμετροι q_i δεν μεταβάλλονται σε αυτό το βήμα.

- **Τέλος Επαναλήψεων**
- Υπολογισμός του συνολικού σφάλματος

$$E_T = \sum_{i=1}^N E_i$$

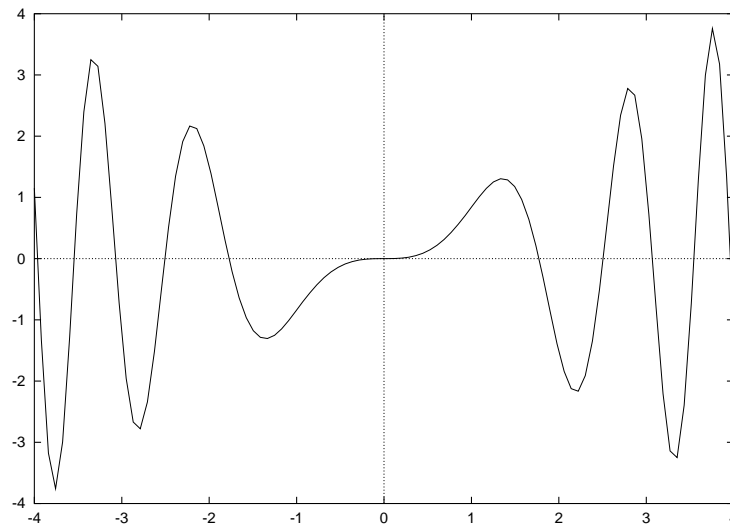
- **Εάν $E_T < \epsilon$, όπου ϵ ένας μικρός θετικός αριθμός, Τότε**
 - **Διακοπή** του αλγορίθμου.

- **Τέλος Εάν**
- **Ελαχιστοποίηση** του συνολικού σφάλματος E_T ως προς τις παραμέτρους q_i . Σε αυτήν την φάση οι παράμετροι p_i των επιμέρους τεχνητών νευρωνικών δικτύων παραμένουν αμετάβλητες.
- **Επανάληψη** από το βήμα της **Εκπαιδεύσεως**.

Όπως μπορεί να παρατηρήσει κανείς, η εκπαίδευση των βαρών των επιμέρους τεχνητών νευρωνικών δικτύων μπορεί να γίνει παράλληλα, αφού τα δίκτυα είναι ανεξάρτητα μεταξύ τους και είναι εξασφαλισμένη η συνέχεια στα σημεία τομής των επιμέρους μοντέλων. Ωστόσο η εκπαίδευση των παραμέτρων q_i δεν μπορεί να γίνει παράλληλα, αφού αυτές εξασφαλίζουν την συνέχεια των επιμέρους μοντέλων. Η αρχική εκτίμηση των παραμέτρων q_i έγινε με την χρήση του εξής αλγορίθμου:

- **Για $i = 1, \dots, N$ Επανάλαβε**
 - **Δημιουργία** ενός τεχνητού νευρωνικού δικτύου $NT_i(x, p)$ με σημεία εκπαίδευσής κοντά στο σημείο διαχωρισμού t_i .
 - **Εκπαίδευση** του δικτύου $NT_i(x, p)$.
 - **Για $j = 0, \dots, k$ Επανάλαβε**
 - $f_i^{(j)} = NT_i^{(j)}(x, p)$.
 - **Τέλος Επαναλήψεων**
- **Τέλος Επαναλήψεων**

Σχήμα 4.1: Γραφική παράσταση της $f(x) = x \sin(x^2)$ στο διάστημα $[-4, 4]$.



4.2.3.2. Πειραματικά αποτελέσματα

Στα πειράματα που έγιναν μελετήθηκε η δυνατότητα του προτεινόμενου μοντέλου να μάθει την συνάρτηση $f(x) = x \sin(x^2)$ στο διάστημα $[-4, 4]$. Το γράφημα της συναρτήσεως σε αυτό το διάστημα φαίνεται στην εικόνα 4.1. Σε όλες τις περιπτώσεις η ελαχιστοποίηση των αντιστοιχών μοντέλων έγινε με την μέθοδο καθολικής ελαχιστοποίησης Multistart. Η μέθοδος τοπικής ελαχιστοποίησης που χρησιμοποιήθηκε από την Multistart ήταν η μέθοδος TOLMIN [98], που αποτελεί μια βελτιωμένη εκδοχή της γνωστής μεθόδου ελαχιστοποίησης quasi-Newton BFGS [74]. Η εκπαίδευση των επιμέρους μοντέλων έγινε παράλληλα με την χρήση της βιβλιοθήκης παράλληλου προγραμματισμού Message Passing Interface (MPI) και πιο συγκεκριμένα της εκδοχής LAM. Τα δεδομένα που χρησιμοποιήθηκαν χωρίστηκαν σε δύο κατηγορίες: ομοιόμορφα και τυχαία. Επίσης κάθε κατηγορία περιελάμβανε δύο μικρότερες κατηγορίες: σημεία χωρίς θόρυβο και σημεία στα οποία έχει προστεθεί 15% “λευκός” θόρυβος. Η προσθήκη του θορύβου έγινε για να εξακριβωθεί αν το προτεινόμενο μοντέλο εξακολουθεί να συμπεριφέρεται ομαλά στα όρια των τμηματικών μοντέλων ακόμα και με την παρουσία θορύβου. Στους πίνακες που θα ακολουθήσουν παρουσιάζονται τα αποτελέσματα στα ομοιόμορφα σημεία, καθώς οι παρατηρήσεις που θα γίνουν σε αυτά ισχύουν και για την περίπτωση των τυχαία καταναμημένων σημείων. Στα πειράματα έγινε

σύγκριση των αποτελεσμάτων μεταξύ ενός απλού τεχνητού νευρωνικού δικτύου, τμηματικών νευρωνικών δικτύων που δεν είναι συνεχή στα σημεία τομής και του προτεινομένου μοντέλου. Στους πίνακες που θα ακολουθήσουν η στήλη “NP” καθορίζει τον αριθμό των τμημάτων που χρησιμοποιήθηκαν τόσο από το προτεινόμενο μοντέλο όσο και από τα τμηματικά νευρωνικά δίκτυα, η στήλη “NODES” καθορίζει τον αριθμό των νευρώνων του κρυμμένου επιπέδου, η στήλη “TRAIN” είναι το τετραγωνικό σφάλμα ανά σημείο στο σύνολο εκπαίδευσης, η στήλη “TEST” είναι το τετραγωνικό σφάλμα ανά σημείο στο σύνολο ελέγχου και η στήλη “TIME” είναι ο χρόνος που απαιτήθηκε σε δευτερόλεπτα για την ολοκλήρωση του προγράμματος. Σε όλες τις περιπτώσεις χρησιμοποιήθηκαν 200 σημεία για εκπαίδευση και 1000 για έλεγχο.

Στον πίνακα 4.1 παρατηρούμε την συμπεριφορά ενός απλού τεχνητού νευρωνικού δικτύου για ομοιόμορφα σημεία χωρίς θόρυβο. Όπως μπορούμε να παρατη-

Πίνακας 4.1: Ομοιόμορφα σημεία χωρίς θόρυβο - Απλό τεχνητό νευρωνικό δίκτυο.

NODES	TRAIN	TEST	TIME
10	2.1E-5	2.1E-5	14844
15	1.1E-7	1.0E-7	21953
20	5.0E-9	5.1E-9	16724

ρήσουμε ο χρόνος που απαιτείται για την εκπαίδευση του συστήματος μειώνεται όταν αυξάνονται οι νευρώνες από 15 σε 20. Αυτό συμβαίνει όπως θα παρατηρήσουμε και σε άλλα μοντέλα, καθώς το σύστημα διακόπτει την εκπαίδευση του όταν πετύχει σφάλμα μικρότερο από μια μικρή προκαθορισμένη τιμή. Σε όλα τα πειράματα που έγιναν αυτή η τιμή ήταν ίση με 10^{-8} . Εφαρμόζοντας τμηματικά νευρωνικά δίκτυα στα ίδια δεδομένα και χωρίς απαιτήσεις συνέχειας παίρνουμε τα αποτελέσματα που εμφανίζονται στον πίνακα 4.2. Τα τμηματικά νευρωνικά δίκτυα εκπαιδεύτηκαν και αυτά παράλληλα προκειμένου να υπάρξει σύγκριση του χρόνου εκπαίδευσης τους με τον αντίστοιχο χρόνο του προτεινομένου μοντέλου. Στα αποτελέσματα δεν εμφανίζονται όλοι οι δυνατοί συνδυασμοί διαμερίσεων και αριθμού κρυμμένων νευρώνων, αλλά μόνον κάποιοι από τους συνδυασμούς που διατηρούν το γινόμενο NP*NODES σχετικά σταθερό. Οι ίδιοι συνδυασμοί αριθμού διαμερίσεων και αριθμού κρυμμένων νευρώνων παρουσιάζονται για το προτεινόμενο μοντέλο στους πίνακες 4.3 και 4.4. Όπως παρατηρούμε το προτεινόμενο μοντέλο ειδικά με συνέχεια τιμής επιτυγχάνει καλύτερα αποτελέσματα στο σύνολο ελέγχου από ότι τα τμηματικά νευρωνικά δίκτυα. Επίσης, αν και αρχικά ο χρόνος εκπαίδευσης του προτεινομένου μοντέλου είναι μεγαλύτερος από τον χρόνο εκπαίδευσης των τμηματικών νευρωνικών δικτύων σε μεγάλες

Πίνακας 4.2: Ομοιόμορφα σημεία χωρίς θόρυβο - τμηματικά νευρωνικά δίκτυα.

NP	NODES	TRAIN	TEST	TIME
2	10	1.6E-9	2.9E-8	7750
2	15	1.5E-10	5.6E-8	5537
5	5	5.7E-9	1.6E-7	3119
5	10	5.0E-11	2.7E-9	550
10	3	5.4E-8	1.6E-7	760
10	5	5.0E-11	6.0E-8	76

διαμερίσεις αυτόν παύει να ισχύει, καθώς το προτεινόμενο μοντέλο επιτυγχάνει να εκπαιδευτεί καλύτερα και έτσι η εκπαίδευσή του διακόπτεται.

Πίνακας 4.3: Ομοιόμορφα σημεία χωρίς θόρυβο - Προτεινόμενο μοντέλο με συνέχεια τιμής.

NP	NODES	TRAIN	TEST	TIME
2	10	9.5E-10	1.4E-9	13945
2	15	2.0E-10	2.6E-10	20215
5	5	4.0E-10	4.5E-10	1974
5	10	5.0E-11	7.0E-11	270
10	3	3.0E-10	4.1E-10	445
10	5	5.0E-11	1.0E-10	83

Εφαρμόζοντας το απλό τεχνητό νευρωνικό δίκτυο στα ίδια σημεία αλλά με τυχαίο θόρυβο 15% λαμβάνουμε τα αποτελέσματα του πίνακα 4.5. Όπως παρατηρούμε αν και κάτω από κανονικές συνθήκες η αύξηση των νευρώνων θα έπρεπε να φέρει και βελτίωση των επιδόσεων του τεχνητού νευρωνικού δικτύου στο σύνολο ελέγχου, αυτό δεν επιτυγχάνεται καθώς στα δεδομένα έχει προστεθεί θόρυβος που αποπροσανατολίζει το τεχνητό νευρωνικό δίκτυο και έτσι ενώ αυτό βελτιώνει τις επιδόσεις του στο σύνολο εκπαίδευσεως αποτυγχάνει να κάνει το ίδιο και στο σύνολο ελέγχου. Το ίδιο φαινόμενο παρατηρείται και με την εφαρμογή τμηματικών νευρωνικών δικτύων όπως μπορούμε να δούμε και στον πίνακα 4.6. Μάλιστα όπως μπορούμε να παρατηρήσουμε το σφάλμα στο σύνολο ελέγχου δεν μειώνεται καθώς αυξάνεται το μέγεθος της διαμερίσεως και μάλιστα σε κάποιες περιπτώσεις η αναλογία του σφάλματος ελέγχου προς το σφάλμα εκπαίδευσεως γίνεται 100/1 ή και 1000/1. Αυτό το φαινόμενο δεν παρατηρείται στο προτεινόμενο μοντέλο, όπως μπορούμε άλλωστε να δούμε και στους πίνακες 4.7 και

Πίνακας 4.4: Ομοιόμορφα σημεία χωρίς θόρυβο - Προτεινόμενο μοντέλο με συνέχεια παραγώγου

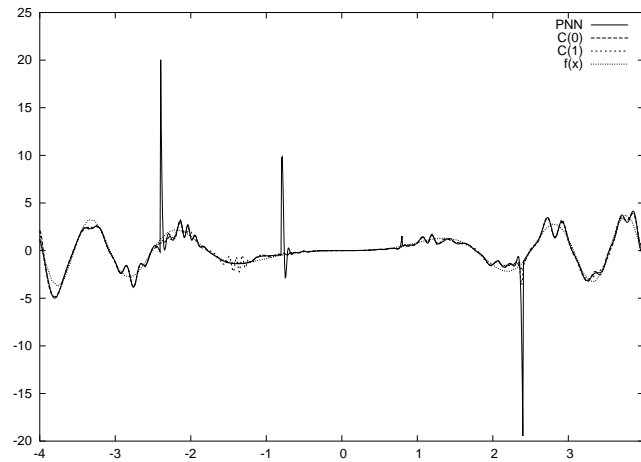
NP	NODES	TRAIN	TEST	TIME
2	10	2.9E-9	4.3E-9	14106
2	15	1.1E-9	2.9E-9	13254
5	5	4.9E-9	1.8E-8	2590
5	10	5.0E-11	1.1E-8	3313
10	3	7.5E-10	2.1E-9	670
10	5	1.5E-10	1.3E-9	144

Πίνακας 4.5: Ομοιόμορφα σημεία με θόρυβο - Απλό τεχνητό νευρωνικό δίκτυο.

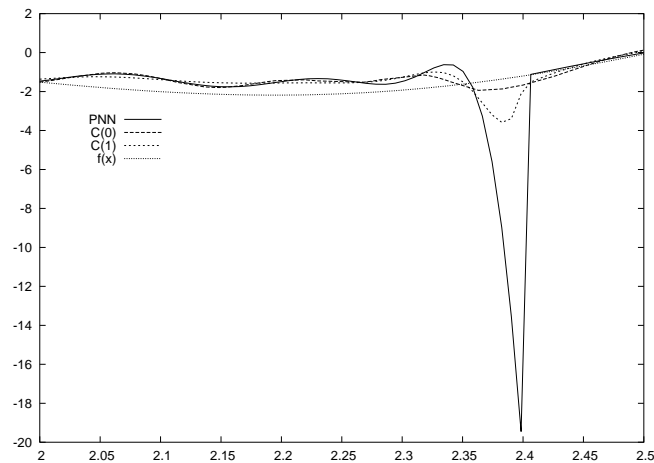
NODES	TRAIN	TEST	TIME
10	1.8E-1	8.3E-2	16319
15	1.6E-1	1.0E-1	17746
20	1.4E-1	1.3E-1	16425

4.8. Το σφάλμα ελέγχου προς το σφάλμα εκπαίδευσης σχεδόν πάντα έχουν μια αναλογία 1/1, κάτι που σημαίνει πως το προτεινόμενο σύστημα δεν μαθαίνει τον θόρυβο που έχει προστεθεί στα αρχικά σημεία. Η ευρωστία του προτεινομένου μοντέλου υπό την παρουσία θορύβου γίνεται περισσότερο ξεκάθαρη αν σχεδιάσουμε την προσέγγιση που κάνουν τα διάφορα είδη μοντέλων υπό την παρουσία θορύβου. Στο σχήμα 4.2 παρατηρούμε την προσέγγιση της αντικειμενικής συναρτήσεως που επιτυγχάνεται από τα διάφορα μοντέλα. Με $f(x)$ συμβολίζεται η αντικειμενική συνάρτηση, με την ετικέτα PNN συμβολίζονται τα τμηματικά νευρωνικά δίκτυα, με $C(0)$ συμβολίζεται το προτεινόμενο μοντέλο με συνέχεια τιμής και με $C(1)$ το προτεινόμενο μοντέλο με συνέχεια παραγώγου. Σε όλες τις περιπτώσεις χρησιμοποιήθηκε διαμέριση μεγέθους 10. Το προτεινόμενο μοντέλο με συνέχεια παραγώγου εμφανίζεται να είναι περισσότερο ομαλό στην προσέγγιση που κάνει και μάλιστα αν εστιάσουμε γύρω από ένα σημείο τομής, όπως φαίνεται και στο σχήμα 4.3, θα παρατηρήσουμε πως το προτεινόμενο μοντέλο με συνέχεια παραγώγου είναι περισσότερο ομαλό και κάνει το μικρότερο σφάλμα γύρω από το σημείο διαχωρισμού.

Σχήμα 4.2: Γραφική αναπαράσταση της προσεγγίσεως της $f(x) = x \sin(x^2)$ παρουσία θορύβου από διαφορετικές μεθόδους.



Σχήμα 4.3: Λεπτομέρειες στην προσέγγιση της συναρτήσεως $f(x) = x \sin(x^2)$ γύρω από ένα σημείο τομής.



Πίνακας 4.6: Ομοιόμορφα σημεία με θόρυβο -τμηματικά νευρωνικά δίκτυα.

NP	NODES	TRAIN	TEST	TIME
2	10	1.3E-1	2.1E-1	6874
2	15	1.1E-1	1.8E-1	10401
5	5	1.4E-1	7.5E+2	2438
5	10	9.2E-2	1.7E+0	6559
10	3	1.3E-1	4.5E+1	1254
10	5	6.7E-2	2.1E+0	2060

Πίνακας 4.7: Ομοιόμορφα σημεία με θόρυβο - Προτεινόμενο μοντέλο με συνέχεια τιμής.

NP	NODES	TRAIN	TEST	TIME
2	10	1.3E-1	1.5E-1	18700
2	15	1.2E-1	1.7E-1	20656
5	5	1.5E-1	1.8E-1	4951
5	10	9.7E-2	2.1E-1	8302
10	3	1.4E-1	1.6E-1	2448
10	5	6.7E-2	3.0E-1	3013

4.2.4. Προβλήματα δύο διαστάσεων

4.2.4.1. Προτεινόμενο μοντέλο

Στις δύο διαστάσεις το πεδίο ορισμού του προβλήματος διαμερίζεται σε $K_x \times K_y$ τετράγωνα. Το προτεινόμενο μοντέλο σε κάθε τετράγωνο θα έχει την μορφή:

$$\psi(x, y, p, q) = B(f, x, y) + S(N, x, y) \quad (4.17)$$

όπου οι συναρτήσεις $B(f, x, y)$ και $S(N, x, y)$ δίνονται από τις εξισώσεις 4.10 και 4.11 θέτοντας $k_1 = k_2 = m_1 = m_2 = 0$. Το τεχνητό νευρωνικό δίκτυο $N(x, y, p)$ είναι ένα Perceptron με ένα κρυμμένο επίπεδο επεξεργασίας, με δύο εισόδους x και y και με διάνυσμα βαρών p . Αν κάνουμε τις πράξεις για τον υπολογισμό της συναρτήσεως $B(f, x, y)$ στο διάστημα $[a_1, b_1] \otimes [a_2, b_2]$ θα καταλήξουμε στην εξίσωση:

$$\begin{aligned} B(f, x_1, x_2) &= f(a_1, x_2) \frac{x_1 - b_1}{a_1 - b_1} + f(b_1, x_2) \frac{x_1 - a_1}{b_1 - a_1} \\ &+ f(x_1, a_2) \frac{x_2 - b_2}{a_2 - b_2} + f(x_1, b_2) \frac{x_2 - a_2}{b_2 - a_2} \end{aligned} \quad (4.18)$$

Πίνακας 4.8: Ομοιόμορφα σημεία με θόρυβο - Προτεινόμενο μοντέλο με συνέχεια παραγώγου.

NP	NODES	TRAIN	TEST	TIME
2	10	1.3E-1	1.9E-1	47572
2	15	1.3E-1	1.8E-1	131208
5	5	1.5E-1	1.8E-1	16252
5	10	1.0E-1	2.1E-1	45861
10	3	1.3E-1	1.8E-1	7167
10	5	7.2E-2	2.4E-1	10358

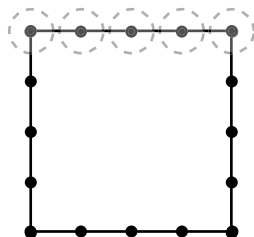
$$- \left\{ \left(f(a_1, a_2) \frac{x_1 - b_1}{a_1 - b_1} + f(b_1, a_2) \frac{x_1 - a_1}{b_1 - a_1} \right) \frac{x_2 - b_2}{a_2 - b_2} \right. \\ \left. \left(f(a_1, b_2) \frac{x_1 - b_1}{a_1 - b_1} + f(b_1, b_2) \frac{x_1 - a_1}{b_1 - a_1} \right) \frac{x_2 - a_2}{b_2 - a_2} \right\}$$

Αν και οι τιμές $f(a_1, a_2)$, $f(b_1, a_2)$, $f(a_1, b_2)$, $f(b_1, b_2)$ αποτελούν στοιχεία του διανύσματος παραμέτρων g και μπορούν να εκτιμηθούν όπως και στην μονοδιάστατη περίπτωση, δεν μπορούμε να πούμε πως ισχύει το ίδιο και για τις παραμέτρους $f(a_1, x_2)$, $f(b_1, x_2)$, $f(x_1, a_2)$ και $f(x_1, b_2)$. Αυτές οι παράμετροι αποτελούν μονοδιάστατες συναρτήσεις και ένας διαφορετικός μηχανισμός απαιτείται για την εκτίμησή τους. Θεωρώντας πως σε κάθε πλευρά του τετραγώνου $[a_1, b_1] \otimes [a_2, b_2]$ ορίζεται και μια διαφορετική μονοδιάστατη συνάρτηση ϕ_{left} , ϕ_{right} , ϕ_{top} και ϕ_{bottom} , η εξίσωση 4.18 αναδιατυπώνεται ως εξής:

$$B(f, x_1, x_2) = \phi_{\text{left}}(a_1, x_2) \frac{x_1 - b_1}{a_1 - b_1} + \phi_{\text{right}}(b_1, x_2) \frac{x_1 - a_1}{b_1 - a_1} \\ + \phi_{\text{bottom}}(x_1, a_2) \frac{x_2 - b_2}{a_2 - b_2} + \phi_{\text{top}}(x_1, b_2) \frac{x_2 - a_2}{b_2 - a_2} \quad (4.19) \\ - \left\{ \left(\phi_{\text{left}}(a_1, a_2) \frac{x_1 - b_1}{a_1 - b_1} + \phi_{\text{right}}(b_1, a_2) \frac{x_1 - a_1}{b_1 - a_1} \right) \frac{x_2 - b_2}{a_2 - b_2} \right. \\ \left. \left(\phi_{\text{left}}(a_1, b_2) \frac{x_1 - b_1}{a_1 - b_1} + \phi_{\text{right}}(b_1, b_2) \frac{x_1 - a_1}{b_1 - a_1} \right) \frac{x_2 - a_2}{b_2 - a_2} \right\}$$

Στην συνέχεια παρατίθενται τα βήματα για την εκτίμηση της συναρτήσεως ϕ_{top} . Ανάλογα βήματα πρέπει να γίνουν για τον υπολογισμό των υπολοίπων συναρτήσεων. Καταρχήν θεωρούμε πως έχουμε μια σειρά από σημεία κατά μήκος της πάνω πλευράς του τετραγώνου όπως φαίνεται και στο σχήμα 4.4. Τα σημεία στα οποία υπάρχουν εστιγμένοι κύκλοι θα αποτελέσουν την βάση για τον υπολογισμό της συναρτήσεως ϕ_{top} .

Σχήμα 4.4: Σημεία για τον υπολογισμό της συναρτήσεως ϕ_{top} .



1. Επιλογή $k + 1$ ισαπέχοντων σημείων P_0, P_1, \dots, P_k (τα σημεία στο κέντρο των εστιγμένων κύκλων) στο διάστημα $[a_1, b_1]$.
2. Εκπαίδευση ενός διδιάστατου τεχνητού νευρωνικού δικτύου $NN_i(x, y, p)$ με σημεία εκπαίδευσεως γύρω από το σημείο P_i .
3. Έστω $N_i = NN_i(P_i)$
4. Δημιουργία ενός κυβικού φυσικού spline $s(x)$ [75] στο διάστημα $[a_1, b_1]$ με χρήση των ζευγαριών (P_i, N_i) .
5. $\phi_{\text{top}}(x_1) = s(x_1), x_1 \in [a_1, b_1]$.

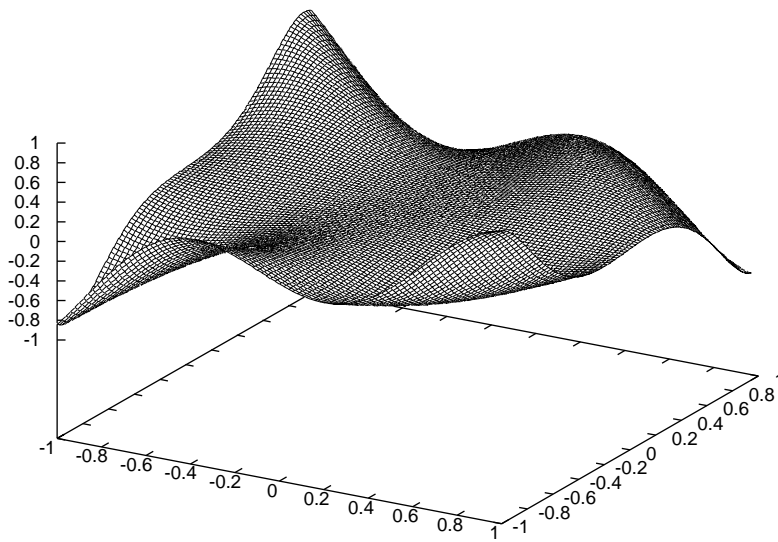
Μετά από την δημιουργία των πλευρικών συναρτήσεων $\phi_{\text{left}}, \phi_{\text{right}}, \phi_{\text{top}}$ και ϕ_{bottom} γίνεται η εκπαίδευση των επιμέρους μοντέλων ως προς τα βάρη των τεχνητών νευρωνικών δικτύων όμοια με την μονοδιάστατη περίπτωση.

4.2.4.2. Πειραματικά αποτελέσματα

Για να αξιολογηθεί η ικανότητα του μοντέλου να μαθαίνει συναρτήσεις στον διδιάστατο χώρο χρησιμοποιήθηκε η συνάρτηση $f(x, y) = xy \sin(2y^2 + 3x)$ στο διάστημα $[-1, 1] \otimes [-1, 1]$. Η γραφική παράσταση αυτής της συναρτήσεως στο συγκεκριμένο διάστημα παρουσιάζεται στο σχήμα 4.5. Χρησιμοποιήθηκαν δύο σύνολα εκπαίδευσεως: ένα με 1600 ομοιόμορφα κατανομημένα σημεία χωρίς θόρυβο και ένα με τα ίδια σημεία στα οποία είχε προστεθεί θόρυβος 10%. Όπως και στην μονοδιάστατη περίπτωση έτσι και στην διδιάστατη η εκπαίδευση των επιμέρους μοντέλων έγινε στο ίδιο παράλληλο περιβάλλον.

Στον πίνακα 4.9 παρουσιάζονται τα αποτελέσματα από την εφαρμογή τμηματικών νευρωνικών δικτύων για διάφορες διαμερίσεις για δεδομένα χωρίς θόρυβο. Τα αντίστοιχα αποτελέσματα για το προτεινόμενο μοντέλο παρουσιάζονται στον πίνακα 4.10. Το προτεινόμενο μοντέλο επιτυγχάνει ίδια ή και ελαφρώς καλύτερα

Σχήμα 4.5: Γραφική παράσταση της συναρτήσεως $f(x, y) = xy \sin(2y^2 + 3x)$ στο διάστημα $[-1, 1] \otimes [-1, 1]$.



Πίνακας 4.9: Ομοιόμορφα δεδομένα στον διδιάστατο χώρο - Τμηματικά νευρωνικά δίκτυα.

NP	NODES	TRAIN	TEST	TIME
2×2	5	2.1E-5	1.8E-5	122
2×2	8	4.7E-7	4.0E-7	684
3×3	4	1.8E-6	2.0E-6	88
3×3	6	3.7E-8	5.0E-8	309
4×4	4	2.0E-7	2.8E-7	59
4×4	6	2.6E-9	4.5E-9	262

Πίνακας 4.10: Ομοιόμορφα δεδομένα στον διδιάστατο χώρο - Προτεινόμενο μοντέλο.

NP	NODES	TRAIN	TEST	TIME
2 × 2	5	3.9E-7	3.9E-7	835
2 × 2	8	2.0E-7	1.8E-7	2823
3 × 3	4	2.8E-8	3.3E-8	267
3 × 3	6	1.8E-8	3.3E-8	925
4 × 4	4	5.6E-9	1.3E-8	283
4 × 4	6	2.6E-9	1.2E-8	781

Πίνακας 4.11: Ομοιόμορφα δεδομένα στον διδιάστατο χώρο με θόρυβο - Τμηματικά νευρωνικά δίκτυα.

NP	NODES	TRAIN	TEST	TIME
3 × 3	6	6.8E-4	1.5E-3	237
3 × 3	8	6.2E-4	1.3E-1	613
4 × 4	6	6.1E-4	1.4E-1	141
4 × 4	8	4.9E-4	7.4E-3	224

αποτελέσματα σε σχέση με τα τμηματικά νευρωνικά δίκτυα, αλλά απαιτεί περισσότερο χρόνο για να το πετύχει αυτό. Κάτι τέτοιο είναι δικαιολογημένο από την στιγμή που το προτεινόμενο μοντέλο διαθέτει αρκετές παραμέτρους που δεν ανήκουν στα τεχνητά νευρωνικά δίκτυα και οι οποίες δεν μπορούν να εκπαιδευτούν παράλληλα. Ωστόσο, το προτεινόμενο μοντέλο επιτυγχάνει καλύτερα αποτελέσματα παρουσία θορύβου. Εφαρμόζοντας τα τμηματικά νευρωνικά δίκτυα στα δεδομένα που διαθέτουν θόρυβο παίρνουμε τα αποτελέσματα του πίνακα 4.11. Η αναλογία του σφάλματος ελέγχου προς το σφάλμα εκπαίδευσης φτάνει ακόμα και το 1000/1, κάτι που σημαίνει πως τα τμηματικά νευρωνικά δίκτυα μαθαίνουν τον θόρυβο και όχι την πραγματική συνάρτηση. Από την άλλη εφαρμόζοντας το προτεινόμενο μοντέλο στα ίδια δεδομένα παίρνουμε τα αποτελέσματα του πίνακα 4.12. Από ότι μπορούμε να δούμε το προτεινόμενο μοντέλο δεν παρουσιάζει το ίδιο πρόβλημα παρουσία θορύβου και η αναλογία του σφάλματος ελέγχου προς το σφάλμα εκπαίδευσης διατηρείται περίπου στο 1/1.

4.2.5. Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάστηκε μια νέα τεχνική τμηματικής προσεγγίσεως συναρτήσεων με το όνομα Neural Splines. Η νέα τεχνική είναι μια υβριδική

Πίνακας 4.12: Ομοιόμορφα δεδομένα στον διδιάστατο χώρο με θόρυβο- Προτεινόμενο μοντέλο.

NP	NODES	TRAIN	TEST	TIME
3 × 3	6	7.5E-4	3.7E-4	474
3 × 3	8	7.1E-4	4.2E-4	862
4 × 4	6	6.7E-4	5.0E-4	202
4 × 4	8	5.9E-4	5.9E-4	423

προσέγγιση με την χρήση τεχνητών νευρωνικών δικτύων και κατάλληλων πολυωνύμων που εγγυώνται την συνέχεια της τιμής του μοντέλου ή και παραγώγων αυτού στα σύνορα μεταξύ των τμημάτων. Το νέο μοντέλο έχει το πλεονέκτημα πως είναι κατάλληλα διαμορφωμένο, ώστε να μπορεί να χρησιμοποιηθεί με ευκολία σε παράλληλα προγραμματιστικά περιβάλλοντα. Μετά τα πειράματα που πραγματοποιήθηκαν για την προσέγγιση μονοδιάστατων και διδιάστατων συναρτήσεων διαπιστώθηκε η ευρωστία της νέας μεθοδολογίας και η ικανότητά της να διατηρεί τις προσεγγιστικές της ιδιότητες ακόμα και υπό την παρουσία θορύβου στα δεδομένα. Από την άλλη η μέθοδος βασίζει την λειτουργία της και τις ικανοποιητικές προσεγγιστικές ιδιότητες σε μια σειρά από παραμέτρους (διάνυσμα q) οι οποίες δεν μπορούν να εκπαιδευτούν παράλληλα όπως οι παράμετροι των νευρωνικών δικτύων. Αυτό το γεγονός αν και δεν προκαλεί σημαντική καθυστέρηση σε μία και δύο διαστάσεις ενδέχεται να το κάνει σε περισσότερες διαστάσεις, όπου το πλήθος των παραμέτρων αυτών θα ξεπεράσει σημαντικά το πλήθος των παραμέτρων των τεχνητών νευρωνικών δικτύων.

ΚΕΦΑΛΑΙΟ 5

ΕΠΙΛΥΣΗ ΔΙΑΦΟΡΙΚΩΝ ΕΞΙΣΩΣΕΩΝ

Περιεχόμενα

5.1	Γενικά	85
5.2	Προτεινόμενη τεχνική	85
5.3	Γραμματική	86
5.4	Αρχικοποίηση	86
5.5	Αποτίμηση καταλληλότητας	86
5.5.1	Συνήθεις διαφορικές εξισώσεις	88
5.5.2	Συστήματα διαφορικών εξισώσεων	88
5.5.3	Μερικές διαφορικές εξισώσεις	90
5.6	Υπολογισμός παραγώγων	91
5.7	Εφαρμογή γενετικών τελεστών	92
5.8	Κριτήρια τερματισμού	92
5.9	Πειραματικά αποτελέσματα	92
5.9.1	Παραδείγματα γραμμικών διαφορικών εξισώσεων	93
5.9.2	Παραδείγματα μη γραμμικών διαφορικών εξισώσεων	96
5.9.3	Παραδείγματα συστημάτων διαφορικών εξισώσεων	98

5.9.4	Συνήθεις διαφορικές εξισώσεις χωρίς αναλυτική λύση . . .	100
5.9.5	Μια ειδική περίπτωση	101
5.9.6	Μερικές διαφορικές εξισώσεις	102
5.10	Συμπεράσματα	106

5.1 Γενικά

Πάρα πολλά προβλήματα από τον χώρο της φυσικής, της χημείας, της βιολογίας κτλ. μπορούν να εκφραστούν με διαφορικές εξισώσεις ή και με συστήματά τους. Τέτοια προβλήματα μπορεί να είναι η πρόβλεψη της θερμοκρασίας σε μια σειρά από χημικές αντιδράσεις, η εκτίμηση κέρδους σε αγοραπωλησίες μετοχών, κβαντική μηχανική, κινηματική κ.ά. Για αυτό πάρα πολλές τεχνικές έχουν προταθεί και αναπτυχθεί για την επίλυση τέτοιων εξισώσεων, όπως οι τεχνικές Runge Kutta, τεχνικές προβλέψεως - διορθώσεως [103], τεχνικές που βασίζονται σε ακτινικές συναρτήσεις βάσεως [104] και μέθοδοι που βασίζονται σε τεχνητά νευρωνικά δίκτυα [94]. Με χρήση γενετικού προγραμματισμού έχουν εμφανιστεί προσπάθειες [95, 96] για την ακριβή επίλυση διαφορικών εξισώσεων. Στο παρόν κεφάλαιο προτείνεται μια τεχνική για την συμβολική επίλυση διαφορικών εξισώσεων με την χρήση της τεχνικής της Γραμματικής Εξελίξεως. Η προτεινόμενη μέθοδος δημιουργεί δοκιμαστικές λύσεις για την διαφορική εξίσωση και μέσα από την εφαρμογή γενετικών τελεστών προσπαθεί να προσεγγίσει την ακριβή λύση της. Προκειμένου να βρεθεί η ακριβής λύση η μέθοδος χρησιμοποιεί συναρτήσεις τιμωρίας (penalty functions) που προστίθενται στην τιμή καταλληλότητας. Η μέθοδος είναι γενική και όπως θα δούμε στην συνέχεια μπορεί να εφαρμοστεί σε συνήθεις διαφορικές εξισώσεις, συστήματα διαφορικών εξισώσεων αλλά και σε μερικές διαφορικές εξισώσεις.

5.2 Προτεινόμενη τεχνική

Η προτεινόμενη μέθοδος δημιουργεί συναρτήσεις χρησιμοποιώντας ένα υποσύνολο της γλώσσας προγραμματισμού C και στην συνέχεια, αφού υπολογιστούν αναλυτικά οι απαιτούμενες παράγωγοι με την τεχνική της στοίβας που θα παρουσιαστεί παρακάτω, υπολογίζεται το σφάλμα προσεγγίσεως σε ένα σύνολο από σημεία στο πεδίο ορισμού της εξισώσεως. Σε μορφή αλγορίθμου η προτεινόμενη διαδικασία έχει ως ακολούθως:

1. Αρχικοποίηση του πληθυσμού.
2. Αποτίμηση της καταλληλότητας κάθε χρωμοσώματος. Κάθε χρωμόσωμα είναι ένα διάνυσμα ακεραίων αριθμών, το οποίο μέσα από την απεικόνιση της Γραμματικής Εξελίξεως θα παράγει μια υποψήφια λύση για την διαφορική εξίσωση.
3. Εφαρμογή γενετικών τελεστών της διασταυρώσεως, μεταλλάξεως και επιλογής.

4. Αν ισχύουν τα κριτήρια τερματισμού τότε τερματισμός, διαφορετικά μετάβαση στο 2.

Στην συνέχεια παρουσιάζονται αναλυτικά τα τμήματα του παραπάνω αλγορίθμου.

5.3 Γραμματική

Η γραμματική που παρουσιάζεται στο σχήμα 5.1 χρησιμοποιείται για την παραγωγή υποψηφίων λύσεων για την διαφορική εξίσωση. Μέσα από την διαδικασία της Γραμματικής Εξελίξεως και με την χρήση αυτής της γραμματικής τα χρωμοσώματα του γενετικού αλγορίθμου θα παράγουν υποψήφιας λύσεις της διαφορικής εξίσωσης σε αναλυτική μορφή. Δεξιά από τους κανόνες παραγωγής υπάρχουν διακριτικοί αριθμοί για την εφαρμογή της Γραμματικής Εξελίξεως, όπως είδαμε σε προηγούμενο κεφάλαιο. Δεν χρησιμοποιήθηκαν όλες οι συναρτήσεις από την βιβλιοθήκη της γλώσσας C, ωστόσο κάτι τέτοιο είναι σχετικά εύκολο να γίνει και δεν επηρεάζει καθόλου την λειτουργία της προτεινόμενης τεχνικής. Επίσης όπως φαίνεται στην γραμματική της προτεινόμενης τεχνικής δεν υπάρχουν ακέραιες σταθερές με μήκος μεγαλύτερο του 1. Αυτό επιλέχθηκε προκειμένου να μην παράγονται μεγάλοι ακέραιοι από την Γραμματική Εξέλιξη που θα δυσκόλευαν την εύρεση των λύσεων διαφορικών εξισώσεων.

5.4 Αρχικοποίηση

Στην φάση της αρχικοποίησης επιλέγονται οι τιμές για τον ρυθμό επιλογής και για τον ρυθμό μεταλλάξεως και αρχικοποιείται ο γενετικός πληθυσμός. Η αρχικοποίηση εκτελείται διαλέγοντας τυχαία έναν ακέραιο αριθμό στο διάστημα $[0,255]$ για κάθε στοιχείο κάθε χρωμοσώματος. Ο περιορισμός της αρχικοποίησης σε αυτό το διάστημα έγινε για να επιταχυνθεί η διαδικασία αναζήτησεως και για να είναι περισσότερο αποδοτικοί οι γενετικοί τελεστές.

5.5 Αποτίμηση καταλληλότητας

Η αποτίμηση της καταλληλότητας κάθε χρωμοσώματος αποτελεί την πλέον βασική εργασία του αλγορίθμου και αυτή η διαδικασία είναι διαφορετική ανάλογα με το αν το πρόβλημα που καλείται να λύσει η προτεινόμενη μέθοδος είναι για συνήθη διαφορική εξίσωση ή για μερική διαφορική εξίσωση.

Σχήμα 5.1: Η γραμματική της προτεινόμενης τεχνικής για επίλυση διαφορικών εξισώσεων.

```
S ::= <expr> (0)
<expr> ::= <expr> <op> <expr> (0)
          | ( <expr> ) (1)
          | <func> ( <expr> ) (2)
          | <digit> (3)
          | x (4)
          | y (5)
          | z (6)
<op> ::= + (0)
          | - (1)
          | * (2)
          | / (3)
<func> ::= sin (0)
          | cos (1)
          | exp (2)
          | log (3)
<digit> ::= 0 (0)
          | 1 (1)
          | 2 (2)
          | 3 (3)
          | 4 (4)
          | 5 (5)
          | 6 (6)
          | 7 (7)
          | 8 (8)
          | 9 (9)
```

5.5.1. Συνήθειες διαφορικές εξισώσεις

Οι συνήθειες διαφορικές εξισώσεις μπορούν να εκφραστούν ως:

$$f(x, y, y^{(1)}, \dots, y^{(n-1)}, y^{(n)}) = 0, \quad x \in [a, b] \quad (5.1)$$

όπου $y^{(n)}$ είναι η n -οστή παράγωγος της y . Έστω ότι οι οριακές συνθήκες δίνονται από:

$$g_i(x, y, y^{(1)}, \dots, y^{(n-1)})|_{x=t_i} = 0, \quad i = 1, \dots, n$$

όπου t_i είναι κάποιο από τα δύο άκρα a ή b . Τα βήματα για τον υπολογισμό της καταλληλότητας ενός μέλους του πληθυσμού είναι τα επόμενα:

1. Επιλογή N ισαπέχοντων σημείων $(x_0, x_1, \dots, x_{N-1})$ στο πεδίο ορισμού της διαφορικής.
2. Για κάθε χρωμόσωμα i
 - (α) Κατασκευή του αντίστοιχου μοντέλου $M_i(x)$, που εκφράζεται με την χρήση της γραμματικής του σχήματος 5.1.
 - (β) Υπολογισμός της ποσότητας

$$E(M_i) = \sum_{j=0}^{N-1} \left(f(x_j, M_i^0(x_j), \dots, M_i^{(n)}(x_j)) \right)^2 \quad (5.2)$$

- (γ) Υπολογισμός της τιμωρίας $P(M_i)$ όπως περιγράφεται παρακάτω.
- (δ) Υπολογισμός της τιμής καταλληλότητας του χρωμοσώματος ως:

$$v_i = E(M_i) + P(M_i) \quad (5.3)$$

Η συνάρτηση τιμωρίας P εξαρτάται από τις οριακές συνθήκες και έχει την μορφή:

$$P(M_i) = \lambda \sum_{k=1}^n g_k^2(x, M_i, M_i^{(1)}, \dots, M_i^{(n-1)})|_{x=t_k} \quad (5.4)$$

όπου λ είναι ένας θετικός αριθμός.

5.5.2. Συστήματα διαφορικών εξισώσεων

Η προτεινόμενη τεχνική μπορεί να επιλύσει συστήματα διαφορικών εξισώσεων που εκφράζονται στην μορφή:

$$\begin{pmatrix} f_1(x, y_1, y_1^{(1)}, y_2, y_2^{(1)}, \dots, y_k, y_k^{(1)}) \\ f_2(x, y_1, y_1^{(1)}, y_2, y_2^{(1)}, \dots, y_k, y_k^{(1)}) \\ \vdots \\ f_k(x, y_1, y_1^{(1)}, y_2, y_2^{(1)}, \dots, y_k, y_k^{(1)}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (5.5)$$

όπου k είναι ο αριθμός των διαφορικών εξισώσεων του συστήματος. Οι αρχικές συνθήκες του συστήματος εκφράζονται από τον επόμενο πίνακα:

$$\begin{pmatrix} y_1(a) = y_{1a} \\ y_2(a) = y_{2a} \\ \vdots \\ y_k(a) = y_{ka} \end{pmatrix} \quad (5.6)$$

Τα βήματα για την αποτίμηση της τιμής καταλληλότητας είναι τα επόμενα:

1. Επιλογή N ισαπέχοντων σημείων $(x_0, x_1, \dots, x_{N-1})$ στο αντίστοιχο πεδίο.
2. Για κάθε δοθέν χρωμόσωμα i κάνε
 - (α) Ομοιόμορφος διαχωρισμός του χρωμοσώματος σε k τμήματα, όπου k είναι ο αριθμός των διαφορικών εξισώσεων του συστήματος.
 - (β) Δημιουργία με την χρήση της Γραμματικής Εξελιξέως k μοντέλων, έστω τα M_{ij} , $j = 1, \dots, k$.
 - (γ) Υπολογισμός των ποσοτήτων $\forall j = 1, 2, \dots, k$

$$E(M_{ij}) = \sum_{l=0}^{N-1} (x_l, M_{i1}(x_l), M_{i1}^{(1)}(x_l), \dots, M_{ik}(x_l), M_{ik}^{(1)}(x_l)) \quad (5.7)$$

- (δ) Υπολογισμός της ποσότητας

$$E(M_i) = \sum_{j=1}^k (E(M_{ij})) \quad (5.8)$$

- (ε) Υπολογισμός των συσχετιζομένων τιμών τιμωρίας

$$P(M_{ij}) = \lambda (M_{ij}(x_0) - y_{ja})^2, \quad \forall j = 1, 2, \dots, k \quad (5.9)$$

όπου λ ένας θετικός αριθμός.

- (στ) Υπολογισμός της συνολικής τιμής τιμωρίας

$$P(M_i) = \sum_{j=1}^k P(M_{ij}) \quad (5.10)$$

- (ζ) Τελικά, η τιμή καταλληλότητας για το χρωμόσωμα i δίνεται από την σχέση:

$$u_i = E(M_i) + P(M_i) \quad (5.11)$$

5.5.3. Μερικές διαφορικές εξισώσεις

Η μέθοδος χρησιμοποιήθηκε μόνον σε ελλειπτικές μερικές διαφορικές εξισώσεις δύο και τριών μεταβλητών με Dirichlet οριακές συνθήκες. Η γενίκευση της διαδικασίας για άλλους τύπους διαφορικών και για διαφορετικές οριακές συνθήκες είναι σχετικά εύκολο να γίνει. Κάθε μερική διαφορική εξίσωση που αντιμετωπίζεται στην παρούσα διατριβή εκφράζεται στην μορφή:

$$f\left(x, y, \Psi(x, y), \frac{\partial}{\partial x}\Psi(x, y), \frac{\partial}{\partial y}\Psi(x, y), \frac{\partial^2}{\partial x^2}\Psi(x, y), \frac{\partial^2}{\partial y^2}\Psi(x, y)\right) = 0 \quad (5.12)$$

με $x \in [x_0, x_1]$ και $y \in [y_0, y_1]$. Από τις διαφορικές εξισώσεις της παραπάνω γενικής μορφής επικεντρωθήκαμε στο υποσύνολο των ελλειπτικών διαφορικών εξισώσεων. Οι σχετιζόμενες Dirichlet οριακές συνθήκες εκφράζονται ως: $\Psi(x_0, y) = f_0(y)$, $\Psi(x_1, y) = f_1(y)$, $\Psi(x, y_0) = g_0(x)$, $\Psi(x, y_1) = g_1(x)$.

Τα βήματα για τον υπολογισμό της τιμής καταλληλότητας είναι τα επόμενα:

1. Επιλογή N^2 ομοιόμορφα κατανομημένων σημείων στο κουτί $[x_0, x_1] \times [y_0, y_1]$, N_x ισαπεχόντων σημείων στο διάστημα από $x = x_0$ έως $x = x_1$, N_y διάστημα από $y = y_0$ έως $y = y_1$.

2. Για κάθε χρωμόσωμα i

- Δημιουργία μιας δοκιμαστικής λύσεως $M_i(x, y)$ που εκφράζεται στην γραμματική που περιγράφηκε παραπάνω.
- Υπολογισμός της ποσότητας

$$E(M_i) = \sum_{j=1}^{N^2} f(x_j, y_j, M_i(x_j, y_j), \frac{\partial}{\partial x}M_i(x_j, y_j), \frac{\partial}{\partial y}M_i(x_j, y_j), \frac{\partial^2}{\partial x^2}M_i(x_j, y_j), \frac{\partial^2}{\partial y^2}M_i(x_j, y_j))^2 \quad (5.13)$$

- Υπολογισμός των συναρτήσεων τιμωρίας:

$$\begin{aligned} P_1(M_i) &= \sum_{j=1}^{N_x} (M_i(x_0, y_j) - f_0(y_j))^2 \\ P_2(M_i) &= \sum_{j=1}^{N_x} (M_i(x_1, y_j) - f_1(y_j))^2 \\ P_3(M_i) &= \sum_{j=1}^{N_y} (M_i(x_j, y_0) - g_0(x_j))^2 \\ P_4(M_i) &= \sum_{j=1}^{N_y} (M_i(x_j, y_1) - g_1(x_j))^2 \end{aligned} \quad (5.14)$$

- Υπολογισμός της καταλληλότητας του χρωμοσώματος ως:

$$v_i = E(M_i) + P_1(M_i) + P_2(M_i) + P_3(M_i) + P_4(M_i) \quad (5.15)$$

5.6 Υπολογισμός παραγώγων

Οι παράγωγοι υπολογίζονται μαζί με τις αντίστοιχες συναρτησιακές τιμές χρησιμοποιώντας μια πρόσθετη στοίβα και τους ακόλουθους στοιχειώδεις κανόνες παραγωγίσεως, που χρησιμοποιήθηκαν από τις μεθόδους αυτόματης παραγωγίσεως [109] και που χρησιμοποιήθηκαν στα αντίστοιχα εργαλεία [106, 107, 108]:

1. $(f(x) + g(x))' = f'(x) + g'(x)$
2. $(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$
3. $\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - g'(x)f(x)}{g^2(x)}$
4. $f(g(x))' = g'(x)f'(g(x))$

Για την εύρεση της πρώτης παραγώγου χρησιμοποιούνται δύο διαφορετικές στοίβες, η πρώτη χρησιμοποιείται για την τιμή της συναρτήσεως και η δεύτερη για την τιμή της πρώτης παραγώγου. Για παράδειγμα έστω ότι ζητείται η αποτίμηση της πρώτης παραγώγου της $f(x) = \sin(x) + \log(x + 1)$. Έστω ότι S_0 είναι η στοίβα για τις συναρτησιακές τιμές και S_1 είναι η στοίβα για την τιμή της πρώτης παραγώγου. Η συνάρτηση $f(x)$ σε επιθεματική μορφή μπορεί να γραφεί ως “ $x \sin x 1 + \log +$ ”. Τα στοιχεία της συναρτήσεως διαβάζονται από αριστερά προς δεξιά μέχρι να βρεθεί το τέλος του αλφαριθμητικού. Οι επόμενοι υπολογισμοί πραγματοποιούνται στις στοίβες S_0 και S_1 . Συμβολίζουμε με (a_0, a_1, \dots, a_n) τα στοιχεία μίας στοίβας και a_n το στοιχείο στην κορυφή της στοίβας.

1. $S_0 = (x), S_1 = (1)$
2. $S_0 = (\sin(x)), S_1 = (1 \cos(x))$
3. $S_0 = (\sin(x), x), S_1 = (1 \cos(x), 1)$
4. $S_0 = (\sin(x), x, 1), S_1 = (1 \cos(x), 1, 0)$
5. $S_0 = (\sin(x), x + 1), S_1 = (1 \cos(x), 1 + 0)$
6. $S_0 = (\sin(x), \log(x + 1)), S_1 = (1 \cos(x), \frac{1+0}{x+1})$
7. $S_0 = (\sin(x) + \log(x + 1)), S_1 = (1 \cos(x) + \frac{1+0}{x+1})$

Στο τέλος των παραπάνω υπολογισμών η στοίβα S_1 περιέχει την πρώτη παράγωγο της συναρτήσεως $f(x)$. Για την επέκταση των παραπάνω υπολογισμών για δεύτερες παραγώγους χρειάζεται και τρίτη στοίβα, κάτι που είναι σχετικά εύκολο να επιτευχθεί.

5.7 Εφαρμογή γενετικών τελεστών

Καταρχήν, εκτελείται μία ταξινόμηση των χρωμοσωμάτων ως προς την τιμή καταλληλότητάς τους, με τέτοιο τρόπο ώστε τα καλύτερα χρωμοσώματα να μετακινηθούν στην αρχή του πληθυσμού και τα χειρότερα στο τέλος. Μετά από την ταξινόμηση, παράγονται $c = (1 - s) * g$ νέα χρωμοσώματα, όπου s είναι ο ρυθμός επιλογής της προτεινομένης μεθόδου και g είναι ο συνολικός αριθμός των στοιχείων του πληθυσμού. Τα νέα στοιχεία θα αντικαταστήσουν τα παλαιότερα στο τέλος της διαδικασίας διασταυρώσεως. Για κάθε ζευγάρι απογόνων επιλέγονται δυο μέλη από τον τρέχοντα πληθυσμό με την μέθοδο της επιλογής tournament. Έχοντας επιλέξει δύο στοιχεία για διασταύρωση, εκτελείται διασταύρωση ενός σημείου, όπως παρουσιάστηκε στην ενότητα των γενετικών αλγορίθμων. Όταν τελειώσει η διαδικασία της διασταυρώσεως εφαρμόζεται η διαδικασία της μεταλλάξεως όπως περιγράφηκε στο προηγούμενο κεφάλαιο.

5.8 Κριτήρια τερματισμού

Οι γενετικοί τελεστές εφαρμόζονται στον πληθυσμό και παράγονται έτσι νέες γενιές από λύσεις. Αυτό συνεχίζεται είτε μέχρι να φτάσει ο αριθμός των γενιών σε ένα προκαθορισμένο πάνω όριο είτε το καλύτερο χρωμόσωμα στον πληθυσμό να έχει τιμή καταλληλότητας καλύτερη ή ίση από ένα προκαθορισμένο όριο.

5.9 Πειραματικά αποτελέσματα

Σε αυτήν την ενότητα περιγράφονται διάφορα πειράματα που εκτελέστηκαν με την προτεινόμενη μέθοδο σε γραμμικές και μη συνήθεις διαφορικές εξισώσεις, σε συστήματα συνήθων διαφορικών εξισώσεων αλλά και σε μερικές διαφορικές εξισώσεις. Επιπλέον, η προτεινόμενη μέθοδος εφαρμόστηκε και σε διαφορικές εξισώσεις που δεν έχουν κλειστή αναλυτική λύση και επομένως αυτή η λύση δεν μπορεί να αναπαρασταθεί ακριβώς από την γραμματική της προτεινόμενης μεθόδου. Για την περίπτωση των συστημάτων συνήθων διαφορικών εξισώσεων, κάθε χρωμόσωμα διαιρείται σε M τμήματα, όπου M είναι το πλήθος των διαφορικών εξισώσεων στο σύστημα και καθένα από αυτά τα τμήματα αναπαριστά μία υποψή-

φια λύση για την αντίστοιχη συνήθη διαφορική εξίσωση. Χρησιμοποιήθηκε η τιμή του 10% για τον ρυθμό επιλογής και η τιμή του 5% για τον ρυθμό μεταλλάξεως. Το πλήθος των χρωμοσωμάτων ήταν 1000 και το μήκος κάθε χρωμοσώματος 50. Η παράμετρος τιμωρίας λ στην συνάρτηση τιμωρίας ήταν 100 σε όλα τα πειράματα. Ο μέγιστος επιτρεπόμενος αριθμός γενιών ήταν 2000 και το προκαθορισμένο όριο της τιμής καταλληλότητας για τα κριτήρια τερματισμού ήταν 10^{-7} . Η τιμή του N για την περίπτωση των συνήθων διαφορικών εξισώσεων ήταν μεταξύ 10 και 20, κάτι που εξαρτώταν άμεσα από το πρόβλημα προς λύση. Για την περίπτωση των μερικών διαφορικών εξισώσεων το N είχε την τιμή 5 και $N_x = N_y = 50$. Σε όλες τις περιπτώσεις ο αλγόριθμος εκτελέστηκε 30 φορές με διαφορετική τυχαία γεννήτρια κάθε φορά. Στους πίνακες που θα ακολουθήσουν η στήλη MIN επισημαίνει τον ελάχιστο αριθμό γενιών που χρειάστηκαν για την επίλυση της διαφορικής, η στήλη MAX επισημαίνει τον μέγιστο αριθμό γενιών που χρειάστηκαν και η στήλη AVG τον μέσο αριθμό γενιών. Επίσης, για μερικά από τα προβλήματα προς επίλυση παρουσιάζονται γραφήματα με τις ενδιάμεσες λύσεις. Η αποτίμηση των συναρτήσεων που παράγονται από την Γραμματική Εξέλιξη έγινε με την χρήση της προγραμματιστικής βιβλιοθήκης FunctionParser [105].

5.9.1. Παραδείγματα γραμμικών διαφορικών εξισώσεων

ODE1

$$y' = \frac{2x - y}{x}$$

με $y(0.1) = 20.1$ και $x \in [0.1, 1.0]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = x + \frac{2}{x}$.

ODE2

$$y' = \frac{1 - y \cos(x)}{\sin(x)}$$

με $y(0.1) = \frac{2.1}{\sin(0.1)}$ και $x \in [0.1, 1.0]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = \frac{x+2}{\sin(x)}$.

ODE3

$$y' = -\frac{1}{5}y + \exp\left(-\frac{x}{5}\right) \cos(x)$$

με $y(0) = 0$ και $x \in [0, 1]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = \exp\left(-\frac{x}{5}\right) \sin(x)$.

ODE4

$$y'' = -100y$$

με $y(0) = 0$, $y'(0) = 10$ και $x \in [0, 1]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = \sin(10x)$.

ODE5

$$y'' = 6y' - 9y$$

με $y(0) = 0$, $y'(0) = 2$ και $x \in [0, 1]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = 2x \exp(3x)$.

ODE6

$$y'' = -\frac{1}{5}y' - y - \frac{1}{5} \exp\left(-\frac{x}{5}\right) \cos(x)$$

με $y(0) = 0$, $y'(0) = 1$ και $x \in [0, 2]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = \exp\left(-\frac{x}{5}\right) \sin(x)$.

ODE7

$$y'' = -100y$$

με $y(0) = 0$, $y(1) = \sin(10)$ και $x \in [0, 1]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = \sin(10x)$.

ODE8

$$xy'' + (1-x)y' + y = 0$$

με $y(0) = 1$, $y(1) = 0$ και $x \in [0, 1]$. Η αναλυτική λύση είναι η $y(x) = 1 - x$.

ODE9

$$y'' = -\frac{1}{5}y' - y - \frac{1}{5} \exp\left(-\frac{x}{5}\right) \cos(x)$$

με $y(0) = 0$, $y(1) = \frac{\sin(1)}{\exp(0.2)}$ και $x \in [0, 1]$. Η αναλυτική λύση της διαφορικής εξισώσεως είναι $y(x) = \exp\left(-\frac{x}{5}\right) \sin(x)$.

Πίνακας 5.1: Αποτελέσματα της προτεινόμενης μεθόδου για γραμμικές διαφορικές εξισώσεις.

ODE	MIN	MAX	AVG
ODE1	8	1453	653
ODE2	52	1816	742
ODE3	23	1598	705
ODE4	14	1158	714
ODE5	89	1189	441
ODE6	37	1806	451
ODE7	42	1242	444
ODE8	3	702	66
ODE9	59	1050	411

Αποτελέσματα

Στον πίνακα 5.1 βλέπουμε τα αποτελέσματα από την εφαρμογή της προτεινόμενης μεθοδολογίας για την επίλυση των παραπάνω διαφορικών εξισώσεων. Οι ίδιες συναρτήσεις δόθηκαν σαν είσοδος στην συνάρτηση επίλυσεως διαφορικών εξισώσεων DSolve του γνωστού πακέτου μαθηματικών MATHEMATICA. Σε όλες τις περιπτώσεις η DSolve κατόρθωσε να βρει την αναλυτική λύση εκτός από τις περιπτώσεις ODE6 και ODE9.

Ένα παράδειγμα

Στο σχήμα 5.2 παρουσιάζονται γραφήματα των δοκιμαστικών λύσεων για την διαφορική εξίσωση **ODE4**. Στην γενιά 22 η τιμή καταλληλότητας ήταν 4200.5 και η ενδιάμεση λύση ήταν:

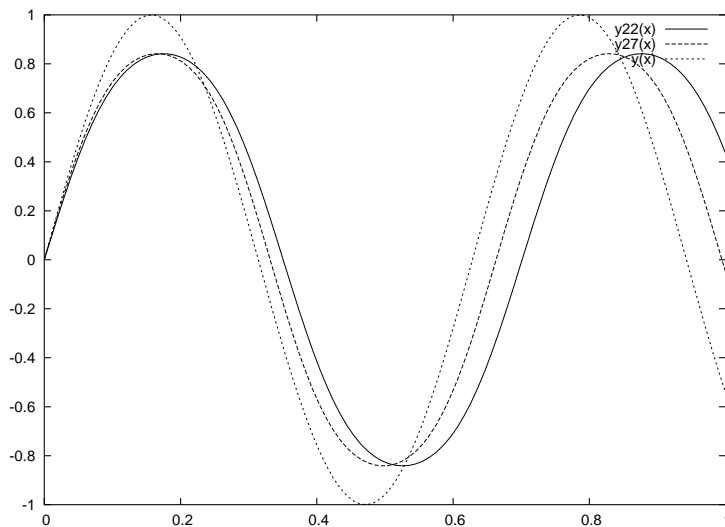
$$y_{22}(x) = \sin((\sin(-\log(4)x((- \cos(\cos(\exp(7))) \exp(\cos(6))) - 5))))$$

Στην γενιά 27 η τιμή καταλληλότητας ήταν 517.17 και η αντίστοιχη συναρτησιακή μορφή:

$$y_{27}(x) = \sin((\sin(-\log(4)x((- \cos(\cos(\sin(7))) \exp(\cos(6))) - 5))))$$

Τελικά, στην γενιά 59 βρέθηκε η ακριβής λύση του προβλήματος.

Σχήμα 5.2: Υποψήφιες λύσεις της $y' = -100y$ με οριακές συνθήκες στο αριστερό άκρο.



5.9.2. Παραδείγματα μη γραμμικών διαφορικών εξισώσεων

NLODE1

$$y' = \frac{1}{2y}$$

με $y(1) = 1$ και $x \in [1, 4]$. Η ακριβής λύση είναι η $y = \sqrt{x}$. Άξιο προσοχής είναι το γεγονός ότι η συνάρτηση \sqrt{x} δεν ανήκει στην βάση συναρτήσεων της μεθόδου. Ωστόσο η προκύπτουσα λύση $y = \exp\left(\frac{\log(x)}{2}\right)$ είναι ακριβώς η ίδια με την \sqrt{x} .

NLODE2

$$(y')^2 + \log(y) - \cos^2(x) - 2 \cos(x) - 1 - \log(x + \sin(x)) = 0$$

με $y(1) = 1 + \sin(1)$ και $x \in [1, 2]$. Η ακριβής λύση είναι η $y = x + \sin(x)$.

NLODE3

$$y''y' = -\frac{4}{x^3}$$

με $y(1) = 0$ και $x \in [1, 2]$. Η ακριβής λύση είναι η $y = \log(x^2)$.

Πίνακας 5.2: Αποτελέσματα εφαρμογής της μεθόδου σε μη γραμμικές διαφορικές εξισώσεις.

NLODE	MIN	MAX	AVG
NLODE1	6	945	182
NLODE2	3	692	86
NLODE3	4	1564	191
NLODE4	6	954	161

NLODE4

$$x^2 y'' + (xy')^2 + \frac{1}{\log(x)} = 0$$

με $y(e) = 0$, $y'(e) = \frac{1}{e}$ και $x \in [e, 2e]$. Η ακριβής λύση είναι η $y(x) = \log(\log(x))$.

Αποτελέσματα

Στον πίνακα 5.2 παρουσιάζονται τα αποτελέσματα από την εφαρμογή της μεθόδου στις παραπάνω μη γραμμικές διαφορικές εξισώσεις. Αξίζει να σημειωθεί πως η συνάρτηση DSolve του πακέτου MATHEMATICA κατόρθωσε να βρει την αναλυτική λύση μόνον για την διαφορική NLODE1.

Ένα παράδειγμα

Στο σχήμα 5.3 παρουσιάζονται τρεις ενδιαμέσες δοκιμαστικές λύσεις για την διαφορική **NLODE3**. Στην δεύτερη γενιά η δοκιμαστική λύση είχε τιμή καταλληλότητας 73.512 και την μορφή:

$$y_2(x) = \log(x - \exp(-x - 1)) - \cos(5)$$

ενώ στην τέταρτη γενιά είχε τιμή καταλληλότητας 48.96 και τύπο:

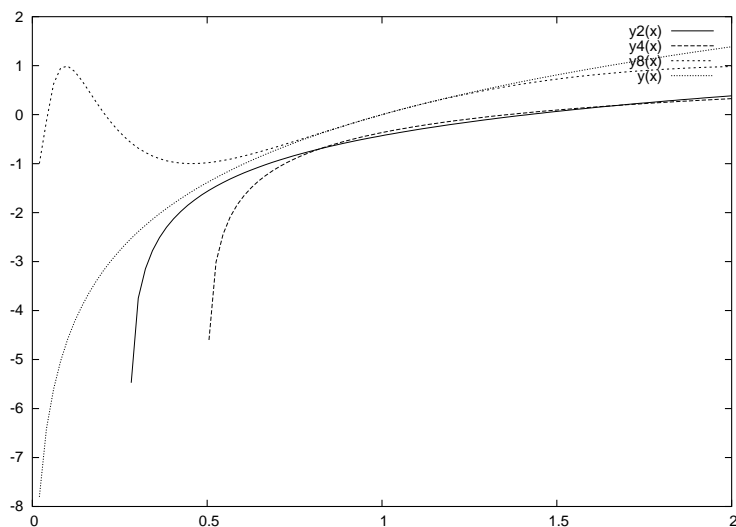
$$y_4(x) = \log(\log(x + x))$$

Ομοίως στην όγδοη η τιμή καταλληλότητας της ενδιαμέσης λύσεως ήταν 4.61 και η συναρτησιακή της μορφή ήταν:

$$y_8(x) = \sin(\log(x * x))$$

Η ακριβής λύση βρέθηκε στην ένατη γενιά.

Σχήμα 5.3: Υποψήφιες λύσεις της NLODE3.



5.9.3. Παραδείγματα συστημάτων διαφορικών εξισώσεων

SODE1

$$\begin{aligned} y_1' &= \cos(x) + y_1^2 + y_2 - (x^2 + \sin^2(x)) \\ y_2' &= 2x - x^2 \sin(x) + y_1 y_2 \end{aligned}$$

με $y_1(0) = 0$, $y_2(0) = 0$ και $x \in [0, 1]$. Η ακριβής λύση δίνεται από: $y_1 = \sin(x)$, $y_2 = x^2$.

SODE2

$$\begin{aligned} y_1' &= \frac{\cos(x) - \sin(x)}{y_2} \\ y_2' &= y_1 y_2 + \exp(x) - \sin(x) \end{aligned}$$

με $y_1(0) = 0$, $y_2(0) = 1$ και $x \in [0, 1]$. Η ακριβής λύση δίνεται από $y_1 = \frac{\sin(x)}{\exp(x)}$, $y_2 = \exp(x)$.

Πίνακας 5.3: Αποτελέσματα εφαρμογής της μεθόδου σε συστήματα διαφορικών εξισώσεων.

SODE	MIN	MAX	AVG
SODE1	6	1211	201
SODE2	15	1108	234
SODE3	30	1205	244
SODE4	5	630	75

SODE3

$$\begin{aligned}
 y_1' &= \cos(x) \\
 y_2' &= -y_1 \\
 y_3' &= y_2 \\
 y_4' &= -y_3 \\
 y_5' &= y_4
 \end{aligned}$$

με $y_1(0) = 0$, $y_2(0) = 1$, $y_3(0) = 0$, $y_4(0) = 1$, $y_5(0) = 0$ και $x \in [0, 1]$.
 Η ακριβής λύση είναι $y_1 = \sin(x)$, $y_2 = \cos(x)$, $y_3 = \sin(x)$, $y_4 = \cos(x)$,
 $y_5 = \sin(x)$.

SODE4

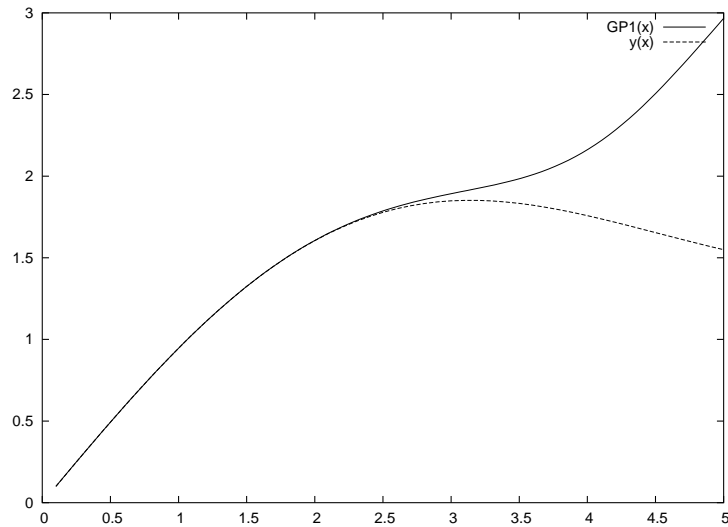
$$\begin{aligned}
 y_1' &= -\frac{1}{y_2} \sin(\exp(x)) \\
 y_2' &= -y_2
 \end{aligned}$$

με $y_1(0) = \cos(1.0)$, $y_2(0) = 1.0$ και $x \in [0, 1]$. Η ακριβής λύση είναι η
 $y_1 = \cos(\exp(x))$, $y_2 = \exp(-x)$.

Αποτελέσματα

Στον πίνακα 5.3 παρουσιάζονται τα αποτελέσματα από την εφαρμογή της μεθόδου στα παραπάνω παραδείγματα. Η συνάρτηση DSolve του πακέτου MATHEMATICA κατόρθωσε να βρει την αναλυτική λύση μόνον για το σύστημα διαφορικών εξισώσεων **SODE3**.

Σχήμα 5.4: Γράφημα των $GP1(x)$ και $y(x) = \int_0^x \frac{\sin(t)}{t} dt$.



5.9.4. Συνήθειες διαφορικές εξισώσεις χωρίς αναλυτική λύση

Παράδειγμα 1

$$y'' + \frac{1}{x}y' - \frac{1}{x} \cos(x) = 0$$

με $x \in [0, 1]$ και $y(0) = 0$ και $y'(0) = 1$. Με 20 σημεία στο πεδίο ορισμού βρέθηκε η ακόλουθη προσεγγιστική λύση:

$$GP1(x) = x(\cos(-\sin(x/3 + \exp(-5 + x - \exp(\cos(x)))))))$$

με τιμή καταλληλότητας $2.1 * 10^{-6}$. Η ακριβής λύση είναι η:

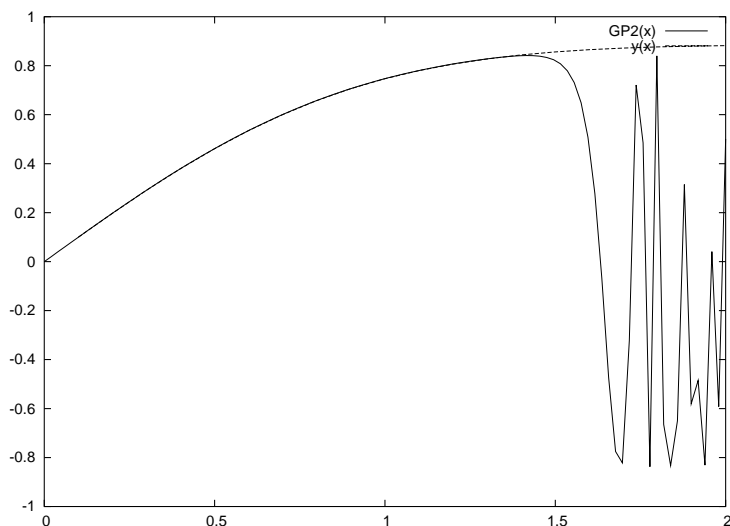
$$y(x) = \int_0^x \frac{\sin(t)}{t} dt$$

Στο σχήμα 5.4, παρουσιάζονται και οι δύο συναρτήσεις στο διάστημα $[0,5]$

Παράδειγμα 2

$$y'' + 2xy = 0$$

Σχήμα 5.5: Γράφημα των $GP2(x)$ και $y(x) = \int_0^x \exp(-t^2) dt$.



με $x \in [0, 1]$ και $y(0) = 0$ και $y'(0) = 1$. Η ακριβής λύση είναι η:

$$y(x) = \int_0^x \exp(-t^2) dt$$

Με 20 σημεία στο πεδίο ορισμού βρέθηκε η ακόλουθη προσεγγιστική λύση:

$$GP2(x) = \sin(\sin(x + \exp(\exp(x) \log(9)) / \exp(8 + \cos(1))) / (\exp(7 / \exp(x)) + 6))$$

με τιμή καταλληλότητας $1.7 * 10^{-5}$. Στο σχήμα 5.5 παρουσιάζονται οι δύο συναρτήσεις στο διάστημα $[0, 5]$.

Αυτό που είναι άξιο παρατήρησης είναι ότι αν και οι διαφορικές εξισώσεις επιλύθηκαν σχεδόν ακριβώς για $x \in [0, 1]$, η προσέγγιση διατηρεί τη ποιότητά της ακόμα και έξω από το διάστημα εκπαίδευσης, ένα γεγονός που δείχνει την γενικευτική δυνατότητα της προτεινόμενης μεθόδου.

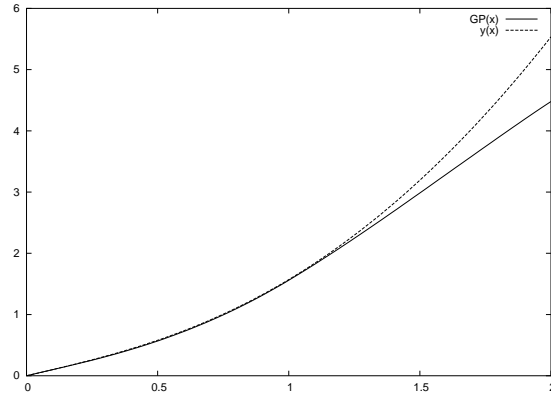
5.9.5. Μια ειδική περίπτωση

Έστω η συνήθης διαφορική εξίσωση

$$y''(x^2 + 1) - 2xy - x^2 - 1 = 0$$

στο διάστημα $[0, 1]$ και με οριακές συνθήκες $y(0) = 0$ and $y'(0) = 1$. Η αναλυτική λύση είναι η $y(x) = (x^2 + 1)\text{atan}(x)$. Η συνάρτηση $\text{atan}(x)$ δεν ανήκει στις

Σχήμα 5.6: $GP(x)$ και $y(x) = (x^2 + 1)\text{atan}(x)$.



γνωστές συναρτήσεις της γραμματικής της προτεινομένης μεθόδου και αυτό το γεγονός καθιστά αυτήν την περίπτωση ειδική. Η λύση που βρέθηκε δεν είναι ακριβής αλλά προσεγγιστική και έχει τον τύπο:

$$GP(x) = \frac{x / \sin(\exp(\cos(5/4 / \exp(x)) - \exp((- \exp(((- ((- \exp(\cos(\sin(2x))))))))))))}{\exp((- \exp(((- ((- \exp(\cos(\sin(2x))))))))))))}$$

Αυτή η προσεγγιστική λύση που παρουσιάζεται στο σχήμα 5.6 έχει τιμή καταλληλότητας 0.0059168.

5.9.6. Μερικές διαφορικές εξισώσεις

PDE1

$$\nabla^2 \Psi(x, y) = \exp(-x)(x - 2 + y^3 + 6y)$$

με $x \in [0, 1]$ και $y \in [0, 1]$ και οριακές συνθήκες: $\Psi(0, y) = y^3$, $\Psi(1, y) = (1 + y^3) \exp(-1)$, $\Psi(x, 0) = x \exp(-x)$, $\Psi(x, 1) = (x + 1) \exp(-x)$. Η ακριβής λύση είναι $\Psi(x, y) = (x + y^3) \exp(-x)$.

PDE2

$$\nabla^2 \Psi(x, y) = -2\Psi(x, y)$$

με $x \in [0, 1]$ και $y \in [0, 1]$ και οριακές συνθήκες: $\Psi(0, y) = 0$, $\Psi(1, y) = \sin(1) \cos(y)$, $\Psi(x, 0) = \sin(x)$, $\Psi(x, 1) = \sin(x) \cos(1)$. Η ακριβής λύση είναι $\Psi(x, y) = \sin(x) \cos(y)$.

PDE3

$$\nabla^2 \Psi(x, y) = 4$$

με $x \in [0, 1]$ και $y \in [0, 1]$ και οριακές συνθήκες: $\Psi(0, y) = y^2 + y + 1$, $\Psi(1, y) = y^2 + y + 3$, $\Psi(x, 0) = x^2 + x + 1$, $\Psi(x, 1) = x^2 + x + 3$. Η ακριβής λύση είναι $\Psi(x, y) = x^2 + y^2 + x + y + 1$.

PDE4

$$\nabla^2 \Psi(x, y) = -(x^2 + y^2)\Psi(x, y)$$

με $x \in [0, 1]$ και $y \in [0, 1]$ και οριακές συνθήκες: $\Psi(x, 0) = 0$, $\Psi(x, 1) = \sin(x)$, $\Psi(0, y) = 0$, $\Psi(1, y) = \sin(y)$. Η ακριβής λύση είναι η $\Psi(x, y) = \sin(xy)$.

PDE5

$$\nabla^2 \Psi(x, y) = (x - 2)\exp(-x) + x\exp(-y)$$

με $x \in [0, 1]$ και $y \in [0, 1]$ και οριακές συνθήκες: $\Psi(x, 0) = x(\exp(-x) + 1)$, $\Psi(x, 1) = x(\exp(-x) + \exp(-1))$, $\Psi(0, y) = 0$, $\Psi(1, y) = \exp(-y) + \exp(-1)$. Η ακριβής λύση είναι η $\Psi(x, y) = x(\exp(-x) + \exp(-y))$.

PDE6

Η ακόλουθη είναι μία εξαιρετικά μη γραμμική μερική διαφορική εξίσωση:

$$\nabla^2 \Psi(x, y) + \exp(\Psi(x, y)) = 1 + x^2 + y^2 + \frac{4}{(1 + x^2 + y^2)^2}$$

με $x \in [-1, 1]$ και $y \in [-1, 1]$ και οριακές συνθήκες: $f_0(y) = \log(1 + y^2)$, $f_1(y) = \log(2 + y^2)$, $g_0(x) = \log(1 + x^2)$ και $g_1(x) = \log(2 + x^2)$. Η ακριβής λύση είναι η $\Psi(x, y) = \log(1 + x^2 + y^2)$.

PDE7

$$\nabla^2 \Psi(x, y, z) = 6$$

με $x \in [0, 1]$ και $y \in [0, 1]$ και $z \in [0, 1]$ και οριακές συνθήκες: $\Psi(0, y, z) = y^2 + z^2$, $\Psi(1, y, z) = y^2 + z^2 + 1$, $\Psi(x, 0, z) = x^2 + z^2$, $\Psi(x, 1, z) = x^2 + z^2 + 1$, $\Psi(x, y, 0) = x^2 + y^2$, $\Psi(x, y, 1) = x^2 + y^2 + 1$. Η ακριβής λύση είναι η $\Psi(x, y, z) = x^2 + y^2 + z^2 + 1$.

Πίνακας 5.4: Αποτελέσματα εφαρμογής της μεθόδου σε μερικές διαφορικές εξισώσεις.

PDE	MIN	MAX	AVG
PDE1	159	1772	966
PDE2	5	1395	203
PDE3	18	311	154
PDE4	4	1698	207
PDE5	195	945	444
PDE6	185	1579	797
PDE7	10	1122	325

Αποτελέσματα

Στον πίνακα 5.4 παρατηρούμε αποτελέσματα από την εφαρμογή της μεθόδου στις παραπάνω μερικές διαφορικές εξισώσεις. Η συνάρτηση DSolve του πακέτου MATHEMATICA δεν κατόρθωσε να βρει την αναλυτική λύση για καμία από τις παραπάνω εξισώσεις.

Ένα παράδειγμα

Στην συνέχεια ακολουθεί ένα παράδειγμα από μια εκτέλεση για την μερική διαφορική εξίσωση **PDE2**. Στην πρώτη γενιά η δοκιμαστική λύση ήταν η:

$$GP1(x, y) = \frac{x}{7}$$

με τιμή καταλληλότητας 8.14. Η διαφορά μεταξύ της δοκιμαστικής λύσεως $GP1(x, y)$ και της ακριβούς λύσεως $\Psi(x, y)$ φαίνεται στο σχήμα 5.7. Στην δέκατη γενιά η δοκιμαστική λύση ήταν:

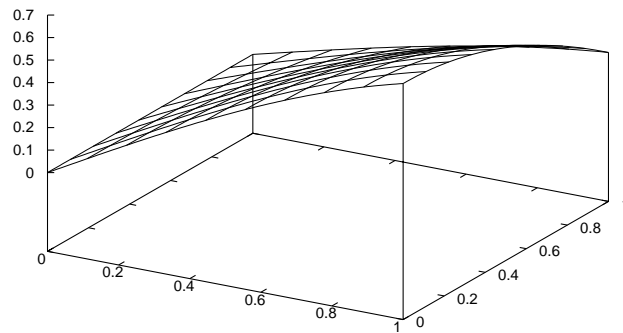
$$GP10(x, y) = \sin(x/3 + x)$$

με τιμή καταλληλότητας 3.56. Η διαφορά μεταξύ της δοκιμαστικής λύσης $GP10(x, y)$ και της πραγματικής λύσεως $\Psi(x, y)$ φαίνεται στο σχήμα 5.8. Στην τεσσαρακοστή γενιά η δοκιμαστική λύση ήταν η:

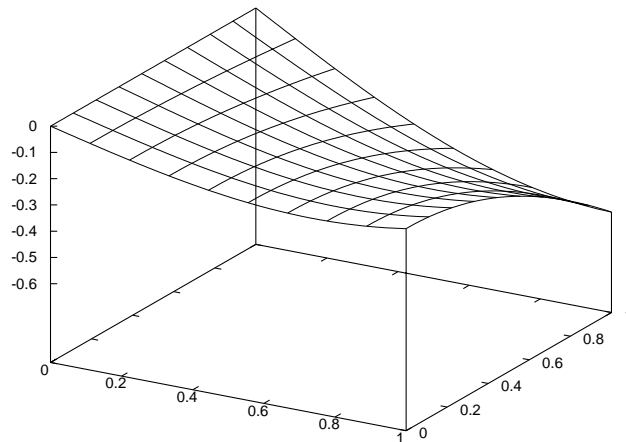
$$GP40(x) = \sin(\cos(y)x)$$

με τιμή καταλληλότητας 0.59. Η διαφορά μεταξύ της δοκιμαστικής λύσεως $GP40(x, y)$ και της πραγματικής λύσεως $\Psi(x, y)$ φαίνεται στο σχήμα 5.9.

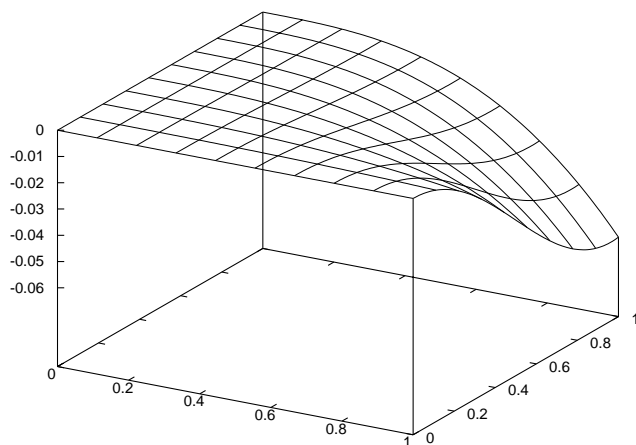
Σχήμα 5.7: Διαφορά μεταξύ των $\Psi(x, y) = \sin(x) \cos(y)$ και $GP1(x, y)$.



Σχήμα 5.8: Διαφορά μεταξύ των $\Psi(x, y) = \sin(x) \cos(y)$ και $GP10(x, y)$.



Σχήμα 5.9: Διαφορά μεταξύ των $\Psi(x, y) = \sin(x) \cos(y)$ και $GP40(x, y)$.



5.10 Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάστηκε μια πρωτότυπη μέθοδος για την επίλυση συνήθων και μερικών διαφορικών εξισώσεων. Η μέθοδος βασίζεται στην τεχνική της Γραμματικής Εξελίξεως, δημιουργεί δοκιμαστικές λύσεις σε κλειστή μορφή προσπαθώντας να ελαχιστοποιήσει ένα σχετιζόμενο σφάλμα. Το πλεονέκτημα είναι ότι η μεθοδολογία της Γραμματικής Εξελίξεως μπορεί να παράγει δοκιμαστικές λύσεις μεγάλης πολυπλοκότητας οι οποίες όμως είναι εύκολο να γίνουν κατανοητές. Αν η γραμματική της μεθόδου έχει αρκετές ενσωματωμένες συναρτήσεις και η εξίσωση έχει λύση που μπορεί να εκφραστεί σε κλειστή αναλυτική μορφή, τότε είναι εξαιρετικά πιθανό η προτεινόμενη μέθοδος να βρει την ακριβή λύση του προβλήματος. Όταν η λύση της διαφορικής εξίσωσης δεν μπορεί να εκφραστεί σε κλειστή αναλυτική μορφή, τότε η προτεινόμενη μέθοδος παράγει μια προσεγγιστική λύση σε κλειστή αναλυτική μορφή. Η γραμματική που παρουσιάστηκε μπορεί σαφώς να βελτιωθεί και να επεκταθεί, με την προσθήκη περισσότερων συναρτήσεων και τελεστών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] R.E. Fletcher and M.J..D. Powell, A rapidly convergent descent algorithm for minimization, *Computer Journal* **6** (1963), 163-168.
- [2] W.C. Davidon, Variable metric method for minimization. *SIAM Journal on Optimization* **1** (1991), 1-17.
- [3] C.G. Broyden, The convergence of a class of double-rank minimization algorithms, part I and II, *Journal of the Institute of Mathematics and its Applications* **6** (1970), 76-90 222-231.
- [4] R.E. Fletcher, A new approach to variable metric algorithms, *Computer Journal* **13** (1970), 317-322.
- [5] D. Goldfarb, A family of variable metric methods derived by variational means, *Mathematics of Computation* **24** (1970), 23-26.
- [6] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, *SIAM Journal on Scientific and Statistical Computing* **19** (1970), 647-656.
- [7] D.C Liu . and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Mathematical Programming* **45** (1989), pp. 503-528.
- [8] D. Feng, P.D. Frank, R.B. Schnabel: Local convergence analysis of tensor methods for nonlinear equations. *Mathematical Programming* **62**(1993), 427-459.
- [9] P.E. Gill., W. Murray and M.A. Saunders, SNOPT: An SQP algorithm for large-scale constrained optimization, Numerical Analysis Report 97-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1997.
- [10] R.S. Dembo and T. Steihaug, Truncated-Newton Algorithms for Large-Scale Unconstrained Optimization, *Mathematical Programming* **26** (1983), 190-212.

- [11] C.G.E. Boender, A.H.G. Rinnooy Kan, G.T. Timmer and L. Stougie, A stochastic method for global optimization, *Mathematical Programming* **22**(1982), 125-140.
- [12] R. E. Fletcher. and C.M. Reeves, Function minimization by conjugate gradients, *Computer Journal* **7** (1964), 149-154.
- [13] E. Polak. and Ribière, Note sur la convergence de méthodes de directions conjuguées, *Revue Française d'Informatique et de Recherche Opérationnelle* **16**, pp. 35-43, 1969.
- [14] R. Hooke R. and T.A. Jeeves, Direct Search solution of numerical and statistical problems, *J. Ass. Comput. Mach.* **8** (1961), 212-229.
- [15] V. Torczon, Pattern search methods for nonlinear optimization, *SIAM Journal on Optimization* **7**(1997), 1-25.
- [16] R.M. Lewis, V. Torczon and M.W. Trosset, Why pattern search works, *Optima: Mathematical Programming Newsletter* **59**(1998), 1-7.
- [17] W. Spendley, G.R. Hext and F.R. Himsworth, Sequential application of simplex desin in optimization and evolutionary operation, *Technometrics* **4** (1992), 441.
- [18] J.A. Nelder J.A. and R. Mead, A Simplex method for function minimization, *Computer Journal* **7**(1965), 308.
- [19] D. E. Goldberg, Genetic Algorithms in search optimization and Machine Learning, New York, Addison - Wesley, 1989.
- [20] L.C.W. Dixon and G.P. Szegö , The global optimization problem: an introduction, in: Dixon, L.C.W. and G.P. Szegö (eds.), *Towards Global Optimization 2, (North - Holland, Amsterdam)* (1978), 1-15 .
- [21] J. Gomulka, Deterministic vs Probabilistic approaches to Global Optimization, in: Dixon, L.C.. and G.P. Szegö (eds.) *Towards Global Optimization 2, (North - Holland, Amsterdam)* (1978), 19-29.
- [22] F. Archetti and F. Schoen, A survey on the global optimization problem: general theory and computational approaches, *Annals of Operations Research* **1**(1984), 87-110.
- [23] V.V. Leonov, A method for covering for the determination of the global maximum of functions of several variables, *Investigations in Cybergenetics, Sov. Radio* (1970), 41-52 (in Russian).

- [24] G.S. Ganshin, Calculation of the maximum of functions, *J. of Comput. Math. and Math. Phys.* **16**(1976), 30-39 (in Russian).
- [25] G.S. Ganshin G.S., Calculation of the maximum of functions of several variables, *Cybergenetics* **83.2** (1983), 61-63 (in Russian).
- [26] A.N. Babij, An algorithm for the search of the global extremum of a function of several variables with a given accuracy, *Cybergenetics* **78.5**(1978), 52-55.
- [27] I. Fiodorova, Search for the global optimum of multiextremal problems, *Optimal Decision Theory* **4**(1978), 93-100.
- [28] S. Inomata and M. Cumada, On the golf method, *Bulletin of the Electrotechnical Laboratory* **25** (1964), 495-512.
- [29] N. Zhidkov and B. Shchedrin, On the search of minimum of a function of several variables, *Computing Methods and Programming* **10** (1978), 203-210.
- [30] S. Incerti, V. Parisi V. and F. Zirilli, A new method for solving nonlinear simultaneous equations, *SIAM J. Num. Anal.* **16**(1979), 779-789.
- [31] F.H. Branin, Widely convergent methods for finding mutiple solutions of simultaneous nonlinear equations, *IBM Journal of Research Developments*(1972), 504-522.
- [32] S.A. Pijavskij, An algorithm for finding the absolute extremum of a function, *Optimal Decision Theory* **2**(1967), 13-24 (in Russian).
- [33] B.O. Shubert, A sequential method seeking the global maximum of a function, *Siam Journal on Numerical Analysis* **9**(1972), 379-388.
- [34] R.P. Brent, *Algorithms for minimization without derivatives*, Prentice-Hall, New Jersey , 1973.
- [35] G.R. Wood, Multidimensional bisection and global minimization, Technical report, University of Canterbury (2), 1985.
- [36] R.P. Ge and Y.F. Qin, A class of filled functions for finding global minimizers of a function fo several variables, *JOTA* **54**(1987), 241-252.
- [37] A.V. Vilkov, N.P. Zhidkov and B.M. Shchedrin, A method for search of the global minimum of a function of one variable, *Journal of Comput. Math. and Math. Phys.* **15**(1975), 1040-1042.

- [38] A. Montalvo, Development of a new algorithm for the global minimization of functions, Ph. D. Thesis in Theoretical and Applied Mechanics, Universidad Nacional Autonoma de Mexico, 1979.
- [39] S. Gomez and A.V. Levy, The tunneling method for solving the constrained global optimization problem with several non - connected feasible regions, *Springer - Verlag, Lecture Notes in Math.* **909**(1982), 34-47.
- [40] A.V. Levy, A. Montalvo, S. Gomez and A. Calderon, Topics in global optimization, *Lecture Notes in Mathematics* **909**(1982), 18-33.
- [41] A.V. Levy and A. Montalvo, The tunneling algorithm for global optimization of functions, *SIAM Journal Sci. Stat. Comp.* **6**(1985), 15-29.
- [42] R.W. Becker and G.V. Lago, A global optimization algorithm, in: *Proceedings of the 8th Allerton Conference on Circuits and Systems Theory* (Monticello, Illinois) (1970), 3-12.
- [43] M.A. Wong and T. Lane, A kth nearest neighbour clustering procedure, *J. R. Statist. Soc. B* **45**(1983), 362-368.
- [44] A. Törn, Global optimization as a combination of global and local search, *Gothenburg Business Adm. Studies* **17**(1973), 191-206.
- [45] I.G. Tsoulos and I.E. Lagaris, MinFinder: Locating all the local minima of a function, *accepted for publication in Computer Physics Communications*.
- [46] R. Storn and K. Price, Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization* **11**(1997), 341-359.
- [47] J. Kennedy and R.C. Eberhart, Particle Swarm Optimization in *Proceedings IEEE International Conference on Neural Networks, IV* (1995) , IEEE Service Center, Piscataway, NJ, 1942-1948.
- [48] J. Kennedy J. and R.C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001.
- [49] C.A. Floudas, P.M. Pardalos, C. Adjiman, W. Esposito, Z. Gümüs, S. Harding, J. Klepeis, C. Meyer and C. Schweiger, *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1999.
- [50] A. Törn, A search clustering approach to global optimization, in Dixon L.C.W, and Szegö (eds.), *Towards Global Optimization 2*, North - Holland, Amsterdam, (1978).

- [51] I.G. Tsoulos and I.E. Lagaris, Stopping rules for Box-Constrained Stochastic Global Optimization, University of Ioannina, Dept. of Computer Science, Technical Report 3-2004.
- [52] I.G. Tsoulos and I.E. Lagaris, Genetically Controlled Random Search: A global optimization method for continuous multidimensional functions, accepted for publication in Computer Physics Communications.
- [53] I.G. Tsoulos and I.E. Lagaris, GenAnneal: Genetically modified Simulated Annealing, accepted for publication in Computer Physics Communications.
- [54] I.G. Tsoulos and I.E. Lagaris, Neural Splines: exploiting Parallelism for Function Approximation Using Modular Neural Networks, accepted for publication in Neural, Parallel and Scientific Computations.
- [55] I.G. Tsoulos and I.E. Lagaris, Solving differential equations with genetic programming, accepted for publication in Genetic Programming and Evolvable Machines.
- [56] A. Törn and A. Žilinskas, *Global Optimization, volume 350 of Lecture Notes in Computer Science*, Springer, Heidelberg, 1987.
- [57] E. Hansen, *Global Optimization Using Interval Analysis, Pure and Applied Mathematics, A series of monographs and textbooks*, 1992.
- [58] H. Ratschek, E. C. Stavropoulos and M. N. Vrahatis, A new hybrid genetic algorithms for Global Optimization, *Nonlinear Analysis, Theory, methods and applications*, **30** (1997), 4529-4538.
- [59] S. Kirkpatrick, C. D. Gellat and M. P. Vecchi, Optimization by simulated annealing, *Science* **220** (1983), 671-680.
- [60] P. J. M van Laarhoven and E. H. L. Aarts, *Simulated Annealing and Boltzman Machines*, John Wiley and Sons, 1980.
- [61] W. L. Goffe, G. D. Ferrier and J. Rogers, Global Optimization of Statistical Functions with Simulated Annealing, *Journal of Econometrics* **60**(1994), 65-100.
- [62] A. W. Jones and G. W. Forbes, Adaptive Simulated Annealing, *Journal of Global Optimization* **6** (1995),1-37.
- [63] F. Glover, Tabu Search - Part I, *ORSA Journal of Computing* **2** (1989), 4-32.

- [64] F. Glover, Tabu Search - Part II, *ORSA Journal of Computing* **2**(1990),:4-32.
- [65] C. De Boor, *A practical guide to Splines*, Springer - Verlag, New York, 1978.
- [66] D. Kincaid and W. Cheney, *Numerical Analysis*, Brooks/Cole Publishing Company, 1991.
- [67] K. Hornik, M. Stinchcombe and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* **2** (1989), 359-366.
- [68] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control Signals and Systems* **2** (1989) 303-314.
- [69] C. Bishop, *Neural Networks for Pattern recognition*, Oxford University Press, 1995.
- [70] I.E. Lagaris, A. Likas and D.I. Fotiadis, Artificial Neural Networks for solving ordinary and partial differential equations, *IEEE Trans. on Neural Networks* **9** (1998), 987-1000.
- [71] I.E. Lagaris, A. Likas and D.I. Fotiadis, Artificial Neural Network methods in Quantum Mechanics, *Computer Physics Communications* **104** (1997), 1-14.
- [72] I.E. Lagaris, A. Likas and D.G. Papageorgiou, Neural Network methods for boundary value problems with irregular boundaries, *IEEE Trans. on Neural Networks* **11** (2000), 1041-1049.
- [73] N. Obreshkov, On the Mechanical Quadratures, *J. Bulgar. Acad. Sci. and Arts* **8**(1942), 191-289.
- [74] R. Fletcher, A new approach to variable metric algorithms, *Computer Journal* **13**(1970), 317-322.
- [75] G. Engeln-Mullges and F. Uhlig, *Numerical Algorithms with Fortran*, Springer-Verlag, Berlin Heidelberg 1996.
- [76] A.P. Leclerc, *Efficient and global optimization*, Dissertation, The Ohio State University, 1994.
- [77] R. Zieliński, A statistical estimate of the structure of multiextremal problems, *Mathematical Programming* **21**(1981), 348-356.

- [78] B. Betrò and F. Schoen, Optimal and sub-optimal stopping rules for the Multistart algorithm in global optimization, *Mathematical Programming* **57**(1992),445-458.
- [79] M. Gaviano, D.E. Ksasov, D. Lera and Y.D. Sergeyev, Software for generation of classes of test functions with known local and global minimum for global optimization, *ACM Trans. Math. Softw.* **29**(2003), 469-480.
- [80] C.G.E Boender, A.H.G. Rinnooy Kan, Bayesian stopping rules for multistart global optimization methods, *Mathematical Programming* **37**(1987), 59-80.
- [81] C.G.E. Boender and H. Romeijn, Stochastic methods, in *Handbook of Global Optimization* (Horst R. and Pardalos P.M. eds.), Kluwer, Dordrecht (1995), 829-871.
- [82] Hart W.E., Sequential stopping rules for random optimization methods with application to multistart local search, *Siam J. Optim* **9**(1998), 270-290.
- [83] M.M. Ali and C. Storey, Topographical Multilevel Single Linkage, *Journal of Global Optimization* **5**(1994), 349-358.
- [84] A. Törn and S. Vitanen, Topographical Global Optimization Using Pre-Sampled Points, *Journal of Global Optimization* **5**(1994), 267-276.
- [85] A.H.G. Rinnooy Kan and G.T. Timmer, Stochastic global optimization methods. Part I: Clustering methods, *Mathematical Programming* **39**(1987), 27-56.
- [86] A.H.G. Rinnooy Kan and G.T. Timmer, Stochastic global optimization methods. Part II: Multi level methods, *Mathematical Programming* **39**(1987), 57-78.
- [87] C.G. Boender, A.H.G. Rinnooy Kan and G.T. Timmer, A stochastic method for global optimization, *Mathematical Progrmming* **22**(1982), 125-140.
- [88] J.R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Pres, Cambridge, Massachusetts, 1994.
- [89] C. Ryan and M. O'Neil, Grammatical Evolution: A Steady State Approach. In *Late Breaking Papers, Genetic Programming* 1998, pp. 180-185.

- [90] M. O'Neill M. and C. Ryan, Genetic Code Degeneracy: Implications for Grammatical Evolution and Beyond. *In Proceedings of the Fifth European Conference in Artificial Life*, 1999, 149-154.
- [91] M. O'Neill and C. Ryan, Under the Hood of Grammatical Evolution. *In Proceedings of the Genetic and Evolutionary Computation Conference 1999*, 1143-1149.
- [92] M. O'Neill and C. Ryan, Evolving Multi-Line Compilable C Programs. *Lecture Notes in Computer Science 1598, Proceedings of the Second European Workshop on Genetic Programming*, Springer - Verlag(1999), 83-92.
- [93] Ryan C., Collins J.J. and O'Neill M. Grammatical Evolution: Evolving Programs for an Arbitrary Language. *Lecture Notes in Computer Science 1391, Proceedings of the First European Workshop on Genetic Programming*, pp. 83-95. Springer - Verlag. 1998.
- [94] I.E. Lagaris, A. Likas, A. and D.I. Fotiadis, Artificial Neural Networks for Solving Ordinary and Partial Differential Equations, *IEEE Transactions on Neural Networks* **9**(1998), 987-1000.
- [95] G. Burgess, Finding Approximate Analytic Solutions to Differential Equations Using Genetic Programming, Surveillance Systems Division, Electronics and Surveillance Research Laboratory, Department of Defense, Australia.
- [96] C. Hongqing, K. Lishan and C. Yuping., Evolutionary Modeling of Systems of Ordinary Differential Equations with Genetic Programming, *Genetic Programming and Evolvable Machines* **1** (2000), 309-337.
- [97] I.G. Tsoulos and I.E. Lagaris, Gradient - Controlled, Typical - Distance Clustering for Global Optimization, University of Ioannina, Dept. of Computer Science, Technical Report 4-2004.
- [98] M.J.D. Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, *Mathematical Programming* **45**(1989), 547-566.
- [99] Ryan C., O'Neill M. and Collins J.J., "Grammatical Evolution: Solving Trigonometric Identities," In proceedings of Mendel 1998: 4th International Mendel Conference on Genetic Algorithms, Optimization Problems, Fuzzy Logic, Neural Networks, Rough Sets., Brno, Czech Republic, June 24-26 1998. Technical University of Brno, Faculty of Mechanical Engineering, pp. 111-119.

- [100] J.J. Collins and C. Ryan, Automatic Generation of Robot Behaviors using Grammatical Evolution, In *Proc. of AROB 2000, the Fifth International Symposium on Artificial Life and Robotics*.
- [101] M. O'Neill and C. Ryan, Automatic generation of caching algorithms, In *Kaisa Miettinen, Marko M. Mkel, Pekka Neittaanmki, and Jacques Periaux (eds.), Evolutionary Algorithms in Engineering and Computer Science*, Jyväskylä, Finland(1999), 127-134.
- [102] A. Brabazon and M. O'Neill, A grammar model for foreign-exchange trading, In *H. R. Arabnia et al., editor, Proceedings of the International conference on Artificial Intelligence, volume II, CSREA Press*(2003) 492-498.
- [103] J.D. Lambert, *Numerical methods for Ordinary Differential Systems: The initial value problem*, John Wiley & Sons: Chichester, England, 1991.
- [104] G.E. Fasshauer, Solving Differential Equations with Radial Basis Functions: Multilevel Methods and Smoothing, *Advances in Computational Mathematics* **11**(1999), 139-159.
- [105] J. Nieminen and J. Yliluoma, Function Parser for C++ v2.7, available from <http://www.students.tut.fi/~warp/FunctionParser/>.
- [106] P. Cusdin P. and J.D. Muller, *Automatic Differentiation and Sensitivity Analysis Methods for CFD*, QUB School of Aeronautical Engineering, 2003.
- [107] O. Stauning, "Flexible Automatic Differentiation using Templates and Operator Overloading in C++," Talk presented at the Automatic Differentiation Workshop at Shrivenham Campus, Cranfield University, June 6, 2003.
- [108] C. Bischof, A. Carle, G. Corliss and A. Griewank, ADIFOR - Generating Derivative Codes from Fortran Programs, *Scientific Programming* **1**(1992), 1-29.
- [109] A. Griewank, On Automatic Differentiation, in M. Iri and K. Tanabe (eds.), *Mathematical Programming: Recent Developments and Applications*, Kluwer Academic Publishers, Amsterdam(1989), 83-108.
- [110] W.L. Price, Global Optimization by Controlled Random Search, *Computer Journal* **20**(1977), 367-370.
- [111] W.L. Price, A Controlled Random Search Procedure for Global Optimization, in *Towards Global Optimization 2*(eds. L. C. W. Dixon and C. P. Szegö) , North-Holland, Amsterdam (1978), 71-84.

- [112] W.L. Price, Global Optimization by Controlled Random Search, *Journal of Optimization, Theory and Applications* **40**(1983), 333-348.
- [113] M.M. Ali and C. Storey, Modified Controlled Random Search Algorithms, *International Journal of Computer Mathematics* **54**(1995), 229-235.
- [114] M.M. Ali, C. Storey and A. Törn, Application of some stochastic global optimization algorithms to practical problems, *Journal of Optimization Theory and Applications* **95**(1997), 545-563.
- [115] W.L. Goffe, G.D. Ferrier and J. Rogers, Global Optimization of Statistical Functions with Simulated Annealing, *J. Econometrics* **60**(1994), 65-100.
- [116] F.V. Theos, I.E. Lagaris and D.G. Papageorgiou, PANMIN: sequential and parallel global optimization procedures with a variety of options for the local search strategy, *Computer Physics Communications* **159** (2004), 63-69.
- [117] L. Ingber, Simulated Annealing: Practice versus theory, *J. Math. Comput. Modelling* **18**(1983), 29-57.

ΠΑΡΑΡΤΗΜΑ Α.

ΣΥΝΑΡΤΗΣΕΙΣ ΔΟΚΙΜΗΣ

1. **Rastrigin.** Ο τύπος της συναρτήσεως είναι

$$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$

και διαθέτει 49 τοπικά ελάχιστα στο διάστημα $[-1, 1]^2$.

2. **Shubert.** Ο τύπος της συναρτήσεως είναι:

$$f(x) = - \sum_{i=1}^2 \sum_{j=1}^5 j (\sin((j+1)x_i) + 1)$$

και διαθέτει 400 ελάχιστα στο διάστημα $[-10, 10]^2$.

3. **Gkls.** Η συνάρτηση $f(x) = \text{Gkls}(x, n, w)$ με $x \in [-1, 1]^n$ διαθέτει w ελάχιστα και η κατασκευή της αναλύεται στην εργασία [79].

4. **Guilin Hills.** Ο τύπος της συναρτήσεως είναι

$$f(x) = 3 + \sum_{i=1}^n c_i \frac{x_i + 9}{x_i + 10} \sin\left(\frac{\pi}{1 - x_i + \frac{1}{2k_i}}\right)$$

όπου $x \in [0, 1]^n$, $c_i > 0$ και k_i είναι θετικοί ακέραιοι. Αυτή η συνάρτηση διαθέτει $\prod_{i=1}^n k_i$ τοπικά ελάχιστα και στα πειράματά μας χρησιμοποιήθηκε σαν n η τιμή 20 και οι παράμετροι k_i ορίστηκαν έτσι ώστε το συνολικό πλήθος των τοπικών ελαχίστων να είναι 100.

5. **Griewank #2.** Ο τύπος της συναρτήσεως είναι

$$f(x) = 1 + \frac{1}{200} \sum_{i=1}^2 x_i^2 - \prod_{i=1}^2 \frac{\cos(x_i)}{\sqrt{i}}$$

η συνάρτηση διαθέτει 529¹ ελάχιστα στο διάστημα $[-100, 100]^2$.

6. **Hansen.** Ο τύπος της συναρτήσεως είναι:

$$f(x) = \sum_{i=1}^5 i \cos((i-1)x_i + i) \sum_{j=1}^5 j \cos((j+1)x_2 + j)$$

και η συνάρτηση διαθέτει 527 τοπικά ελάχιστα στο διάστημα $[-10, 10]^2$.

7. **Camel.** Ο τύπος της συναρτήσεως είναι:

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

και η συνάρτηση διαθέτει 6 ελάχιστα στο διάστημα $[-5, 5]^2$.

8. **Shekel 5**

$$f(x) = - \sum_{i=1}^5 \frac{1}{(x - a_i)(x - a_i)^T + c_i}$$

με $x \in [0, 10]^4$ και

$$a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \end{pmatrix}$$

και

$$c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \end{pmatrix}$$

Η συνάρτηση έχει 5 τοπικά ελάχιστα.

9. **Shekel 7**

$$f(x) = - \sum_{i=1}^7 \frac{1}{(x - a_i)(x - a_i)^T + c_i}$$

¹Ο πραγματικός αριθμός των ελαχίστων της είναι άγνωστος. Ωστόσο καμία μέθοδος που δοκιμάστηκε δεν βρήκε ποτέ περισσότερα από 529 τοπικά ελάχιστα.

με $x \in [0, 10]^4$ και

$$a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 3 & 5 & 3 \end{pmatrix}$$

και

$$c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \end{pmatrix}$$

Η συνάρτηση έχει 7 τοπικά ελάχιστα

10. **Shekel 10.** Ο τύπος της συναρτήσεως είναι:

$$f(x) = - \sum_{i=1}^4 \left(\frac{1}{(x - A_i)(x - A_i)^T + c_i} \right)$$

όπου:

$$A = \begin{pmatrix} 4 & 4 & 4 & 1 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{pmatrix}$$

και

$$c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.5 \end{pmatrix}$$

Η συνάρτηση διαθέτει 10 ελάχιστα στο διάστημα $[0, 10]^4$.

11. **Potential**, το ευρέως γνωστό δυναμικό Lennard-Jones για $N = 3$ άτομα και για $N = 5$ άτομα. Η πρώτη περίπτωση στους πίνακες αποτελεσμάτων εμφανίζεται σαν **Potential(3)** και έχει 9 μεταβλητές και η δεύτερη περίπτωση εμφανίζεται σαν **Potential(5)** και έχει 15 μεταβλητές. Η κλειστή μορφή για το δυναμικό που χρησιμοποιήθηκε στα πειράματα της παρούσας διατριβής ήταν:

$$E = 4 \sum_{j=1}^N \sum_{i=1}^{j-1} \left(\frac{1}{r_{ij}^{12}} - \frac{1}{r_{ij}^6} \right)$$

όπου r_{ij} είναι η ευκλείδεια απόσταση ανάμεσα στα άτομα i και j . Η θέση κάθε ατόμου περιγράφεται από τρεις μεταβλητές (x, y, z) και αυτό έχει σαν συνέπεια το συνολικό πλήθος των μεταβλητών να είναι τριπλάσιο του αριθμού των ατόμων.

12. **Neural**, ένα τεχνητό νευρωνικό δίκτυο με ένα κρυφό επίπεδο και 10 νευρώνες επεξεργασίας ($N = 30$ μεταβλητές) χρησιμοποιήθηκε για την προσέγγιση της συναρτήσεως $g(x) = x \sin(x^2)$, $x \in [-2, 2]$. Το ολικό ελάχιστο της συναρτήσεως τετραγωνικού σφάλματος του τεχνητού νευρωνικού δικτύου είναι $f^* = 0.0$. Η συνάρτηση που ελαχιστοποιείται δίνεται από τον τύπο:

$$E(p) = \sum_{i=1}^M (N(x_i, p) - y_i)^2$$

όπου διατίθενται M ζευγάρια (x_i, y_i) από την αντικειμενική συνάρτηση ως δεδομένα εκπαίδευσης. Η ελαχιστοποίηση γίνεται ως προς τα βάρη p του τεχνητού νευρωνικού δικτύου $N(x, p)$ με τύπο:

$$N(x, p) = \sum_{i=1}^{N/3} p_{3i-2} \text{sig}(p_{3i-1}x + p_{3i})$$

Η συνάρτηση $\text{sig}(x)$ είναι η σιγμοειδής συνάρτηση με τύπο:

$$\text{sig}(x) = \frac{1}{1 + \exp(-x)}$$

13. **Test2N**, είναι μια τεχνητή συνάρτηση με τύπο:

$$f(x) = \frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$$

με $x \in [-5, 5]^n$. Η συνάρτηση έχει 2^n τοπικά ελάχιστα στο συγκεκριμένο διάστημα.

14. **Test30N**, με τύπο

$$f(x) = \frac{1}{10} \sin^2(3\pi x_1) \sum_{i=2}^n \left((x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) \right) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n))$$

και $x \in [-10, 10]^n$. Η συνάρτηση έχει 30^n τοπικά ελάχιστα στο συγκεκριμένο διάστημα.

15. **Goldstein & Price**

$$\begin{aligned} f(x) = & [1 + (x_1 + x_2 + 1)^2 \\ & (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times \\ & [30 + (2x_1 - 3x_2)^2 \\ & (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \end{aligned}$$

Η συνάρτηση έχει 4 τοπικά ελάχιστα στο διάστημα $[-2, 2]^2$ και το ολικό ελάχιστο είναι $f^* = 3.0$.

16. **Branin** με τύπο $f(x) = (x_2 - \frac{5.1}{\sqrt{\pi}} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10$ με $-5 \leq x_1 \leq 10$, $0 \leq x_2 \leq 15$. Η συνάρτηση έχει 3 τοπικά ελάχιστα στο καθοριζόμενο διάστημα.

17. **Hartman 3**

$$f(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$$

με $x \in [0, 1]^3$ και

$$a = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}$$

και

$$c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$$

και

$$p = \begin{pmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{pmatrix}$$

Η συνάρτηση έχει τρία τοπικά ελάχιστα στο συγκεκριμένο διάστημα

18. Hartman 6

$$f(x) = - \sum_{i=1}^4 c_i \exp \left(- \sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right)$$

με $x \in [0, 1]^6$ και

$$a = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}$$

και

$$c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$$

και

$$p = \begin{pmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{pmatrix}$$

Η συνάρτηση διαθέτει 2 τοπικά ελάχιστα στο συγκεκριμένο διάστημα.

ΠΑΡΑΡΤΗΜΑ Β. ΤΟ ΠΑΚΕΤΟ ΛΟΓΙΣΜΙΚΟΥ MINFINDER

Διανομή

Το πακέτο διανέμεται σε μορφή ενός tar.gz αρχείου με το όνομα `MinFinder.tar.gz`. Σε UNIX συστήματα ο χρήστης πρέπει να εκτελέσει τις επόμενες εντολές (σε έναν φάκελο με τα κατάλληλα δικαιώματα) για να αποσυμπιέσει τα αρχεία του πακέτου:

1. `gunzip MinFinder.tar.gz`
2. `tar xfv MinFinder.tar`

Τα παραπάνω βήματα δημιουργούν έναν φάκελο με το όνομα `MinFinder` με τα ακόλουθα περιεχόμενα:

1. **bin**: Αυτός ο κατάλογος είναι αρχικά κενός. Μετά την μεταγλώττιση του πακέτου θα περιέχει το εκτελέσιμο **make_program**, το οποίο χρησιμοποιείται για την δημιουργία των τελικών εκτελέσιμων αρχείων.
2. **doc**: Αυτός ο κατάλογος περιέχει το συνοδευτικό εγχειρίδιο του λογισμικού.
3. **examples**: Σε αυτόν τον κατάλογο περιέχονται διάφορες συναρτήσεις δοκιμής γραμμένες κυρίως σε ANSI C++ αλλά μερικές είναι γραμμένες και σε Fortran 77.
4. **include**: Ο κατάλογος αυτός περιέχει τα αρχεία συμπεριλήψεως για όλες τις κατηγορίες του πακέτου.
5. **src**: Στον κατάλογο αυτόν βρίσκονται τα αρχεία πηγαίου κώδικα του πακέτου.

6. **Makefile**: Είναι το αρχείο εισόδου για το εργαλείο `make`, που θα χρησιμοποιηθεί για την μεταγλώττιση του πακέτου. Ο χρήστης δεν χρειάζεται να αλλάξει κάτι σε αυτό το αρχείο.
7. **Makefile.inc**: Είναι το αρχείο που περιέχει διάφορες παραμέτρους που εξαρτώνται από το σύστημα εγκαταστάσεως, όπως το όνομα του χρησιμοποιούμενου μεταγλωττιστή, ο κατάλογος εγκαταστάσεως κτλ. Ο χρήστης πρέπει να διαβάσει και να αλλάξει αυτό το αρχείο ανάλογα με το σύστημά του πριν την εκκίνηση της εγκαταστάσεως.

Εγκατάσταση

Τα επόμενα βήματα είναι απαραίτητα για την μεταγλώττιση του πακέτου:

1. Αποσυμπίεση του πακέτου με την διαδικασία που διατυπώθηκε προηγουμένως.
2. `cd MinFinder`
3. Τροποποίηση του αρχείου `Makefile.inc` ανάλογα με το σύστημα που θα γίνει η εγκατάσταση.
4. Εκτέλεση της εντολής `make`.

Οι πέντε παράμετροι του αρχείου `Makefile.inc` είναι οι επόμενες:

1. **CXX**: Είναι η παράμετρος που διατηρεί το όνομα του χρησιμοποιούμενου μεταγλωττιστή C++. Στα περισσότερα συστήματα που χρησιμοποιούν την εργαλειοθήκη GNU αυτή η παράμετρος θα έχει την τιμή `g++`.
2. **CC**: Είναι το όνομα του χρησιμοποιούμενου μεταγλωττιστή C, αν ο χρήστης θέλει να γράψει τις αντικειμενικές του συναρτήσεις σε C. Σε GNU συστήματα αυτή η παράμετρος έχει την τιμή `gcc`.
3. **F77**: Είναι το όνομα του χρησιμοποιούμενου μεταγλωττιστή Fortran 77. Για GNU συστήματα αυτή η παράμετρος συνήθως θα έχει την τιμή `g77`.
4. **F77FLAGS**: Ο μεταγλωττιστής GNU FORTRAN 77 (`g77`) προσαρτά μία ή περισσότερες κάτω παύλες στο όνομα των υπορουτινών μετά την μεταγλώττιση. Για να μην συμβεί αυτό μπορούμε να περάσουμε ένα διακόπτη στον μεταγλωττιστή και αυτό γίνεται με αυτή την παράμετρο. Η προκαθορισμένη της τιμή είναι `-fno-underscoring`.
5. **ROOTDIR**: Είναι ο κατάλογος όπου θα εγκατασταθεί το πακέτο `MinFinder`. Αυτή η παράμετρος ουσιαστικά είναι και η μόνη που πρέπει να αλλάξει στα περισσότερα συστήματα.

Κωδικοποίηση συναρτήσεων

Στο σχήμα B.1 βλέπουμε το πρότυπο για τις αντικειμενικές συναρτήσεις που είναι γραμμένες στην γλώσσα προγραμματισμού C. Το ίδιο σχήμα πρέπει να ακολουθηθεί σε C++, αλλά στον κώδικα πρέπει να προστεθεί η γραμμή

```
extern ‘‘C’’ {
```

πριν από τις συναρτήσεις και η γραμμή

```
}
```

μετά από αυτές, προκειμένου να αποτραπεί η δημιουργία συμβόλων από τον μεταγλωττιστή, που θα προκαλούσε προβλήματα στην διαδικασία της συνδέσεως του τελικού εκτελέσιμου. Το πρότυπο για την κωδικοποίηση των αντικειμενικών συναρτήσεων σε Fortran 77 δίνεται στο σχήμα B.2. Το σύμβολο d συμβολίζει την διάσταση της αντικειμενικής συναρτήσεως. Η σημασία των συναρτήσεων στα παραπάνω πρότυπα έχει ως ακολούθως:

1. **getdimension()**: Επιστρέφει την διάσταση της αντικειμενικής συναρτήσεως.
2. **getleftmargin(left)**: Τοποθετεί στον δεκαδικό πίνακα left τα αριστερά άκρα του πεδίου ορισμού της αντικειμενικής συναρτήσεως.
3. **getrightmargin(right)**: Τοποθετεί στον δεκαδικό πίνακα right τα δεξιά περιθώρια της αντικειμενικής συναρτήσεως.
4. **funmin(x)**: Επιστρέφει την τιμή της αντικειμενικής συναρτήσεως, αποτιμώμενη στο σημείο x.
5. **granal(x,g)**: Αποτιμά την παράγωγο της αντικειμενικής συναρτήσεως στο σημείο x και τοποθετεί το αποτέλεσμα στον δεκαδικό πίνακα g.

Το πρόγραμμα make_program

Όταν μεταγλωττίζεται το πακέτο, το εκτελέσιμο make_program θα τοποθετηθεί στον κατάλογο bin. Αυτό το πρόγραμμα δημιουργεί το τελικό εκτελέσιμο της εφαρμογής με το όνομα MinFinder. Δέχεται σαν μοναδικό του όρισμα το όνομα του αρχείου που περιέχει την αντικειμενική συνάρτηση. Αυτό το αρχείο μπορεί να είναι γραμμένο είτε σε ANSI C, είτε σε ANSI C++ ή ακόμα και σε Fortran 77. Το εργαλείο καλεί τον απαραίτητο μεταγλωττιστή για να δημιουργήσει το τελικό εκτελέσιμο MinFinder.

Το πρόγραμμα MinFinder

Το εργαλείο MinFinder δημιουργείται μετά την μεταγλώττιση του αρχείου που περιέχει την αντικειμενική συνάρτηση με την χρήση του `make_program`. Δέχεται μια σειρά από προαιρετικές παραμέτρους στην γραμμή εντολών:

1. **-h**: Το πρόγραμμα εμφανίζει μια βοηθητική οθόνη και τερματίζει.
2. **-s size**: Η ακέραια παράμετρος `size` καθορίζει το μέγεθος του δείγματος. Η προκαθορισμένη τιμή για αυτήν την παράμετρο είναι 20.
3. **-o filename**: Η αλφαριθμητική παράμετρος `filename` καθορίζει ένα αρχείο στο οποίο θα τοποθετηθούν τα τοπικά ελάχιστα που θα βρει η μέθοδος μετά τον τερματισμό του προγράμματος.
4. **-p level**: Η ακέραια παράμετρος `level` δέχεται μόνον δύο τιμές 0 και 1. Αν έχει την τιμή 0, τότε στην καθιερωμένη έξοδο δεν εμφανίζεται τίποτα όσο εκτελείται το πρόγραμμα. Αν έχει την τιμή 1, τότε σε κάθε επανάληψη του αλγορίθμου εμφανίζεται στην καθιερωμένη έξοδο μια γραμμή που περιέχει τον αύξοντα αριθμό της επαναλήψεως, τον αριθμό των ελαχίστων που έχουν βρεθεί, τον αριθμό των συναρτησιακών κλήσεων που έχουν γίνει, τον αριθμό των κλήσεων προς την παράγωγο που έχουν γίνει, την τιμή της ποσότητας $\sigma^2(\delta)$ και την τιμή της μεταβλητής a που χρησιμοποιείται στο κριτήριο τερματισμού του διπλού κουτιού. Η προκαθορισμένη τιμή για την παράμετρο `level` είναι 0.
5. **-r seed**: Η ακέραια παράμετρος `seed` καθορίζει τον σπόρο για την γεννήτρια τυχαίων αριθμών.

Παράδειγμα εφαρμογής

Θεωρήστε την συνάρτηση Six Hump Camel που δίνεται από τον τύπο

$$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, \quad x \in [-5, 5]^2$$

με 6 τοπικά ελάχιστα. Η υλοποίηση σε ANSI C++ και FORTRAN 77 δίνεται στα σχήματα B.3 και B.4. Έστω ότι το αρχείο που περιέχει την συνάρτηση σε C++ ονομάζεται `camel.cc` και αυτό που περιέχει την υλοποίηση σε Fortran ονομάζεται `camel.f`. στω επίσης πως αυτά τα δύο αρχείο βρίσκονται στον υποκατάλογο `examples` της διανομής. Σε αυτόν τον υποκατάλογο μπορούμε να δώσουμε την εντολή:

```
../bin/make_program camel.cc
```

ή για την έκδοση της συναρτήσεως σε Fortran 77

```
../bin/make_program camel.f
```

Και στις δύο περιπτώσεις το πρόγραμμα `make_program` απαντά:

```
RUN ./MinFinder IN ORDER TO RUN THE PROBLEM
```

Εκτελούμε το πρόγραμμα `MinFinder` δίνοντας την εντολή:

```
./MinFinder -o camel.out -p 1 -r 7
```

Στην οθόνη εμφανίζεται η ακόλουθη έξοδος:

```
iters= 1 minimum= 1 fevals= 24 gevals= 33 delta= 0 stopat=1.9763e-323
iters= 2 minimum= 2 fevals= 89 gevals= 106 delta= 0 stopat= 0
iters= 3 minimum= 3 fevals= 151 gevals= 178 delta=6.7063e-05 stopat=3.3531e-05
iters= 4 minimum= 4 fevals= 236 gevals= 272 delta=5.4213e-05 stopat=2.7106e-05
iters= 5 minimum= 6 fevals= 282 gevals= 331 delta=4.1245e-05 stopat=2.0622e-05
iters= 6 minimum= 6 fevals= 386 gevals= 446 delta=3.5875e-05 stopat=1.7938e-05
iters= 7 minimum= 6 fevals= 520 gevals= 591 delta=3.0131e-05 stopat=1.7938e-05
iters= 8 minimum= 6 fevals= 604 gevals= 688 delta=2.5983e-05 stopat=1.7938e-05
iters= 9 minimum= 6 fevals= 680 gevals= 778 delta=2.2972e-05 stopat=1.7938e-05
iters= 10 minimum= 6 fevals= 804 gevals= 916 delta=2.0464e-05 stopat=1.7938e-05
iters= 11 minimum= 6 fevals= 888 gevals= 1015 delta=1.8794e-05 stopat=1.7938e-05
```

Μέχρι την επανάληψη 5 είχαν βρεθεί όλα τα ελάχιστα, αλλά το πρόγραμμα συνέχισε την εκτέλεσή του μέχρι την επανάληψη 11, προκειμένου να ικανοποιηθούν τα κριτήρια τερματισμού. Τα τοπικά ελάχιστα που βρέθηκαν τοποθετήθηκαν στο αρχείο `camel.out`, τα περιεχόμενα του οποίου παρουσιάζονται στην συνέχεια:

```
2
6
-1.703606715 0.7960835687 -0.2154638244
0.0898420131 -0.712656403 -1.031628453
-0.0898420131 0.712656403 -1.031628453
-1.607104753 -0.5686514549 2.10425031
1.703606715 -0.7960835687 -0.2154638244
1.607104753 0.5686514549 2.10425031
```

Στην πρώτη γραμμή του αρχείου ο αριθμός 2 καθορίζει τις διαστάσεις της αντικειμενικής συναρτήσεως και ο αριθμός 6 στην δεύτερη γραμμή το πλήθος των ελαχίστων που βρέθηκαν. Σε κάθε μία από τις επόμενες γραμμές υπάρχουν τρεις καταχωρήσεις. Οι δύο πρώτες είναι οι συντεταγμένες του τοπικού ελαχίστου που βρέθηκε και η τρίτη είναι η τιμή της αντικειμενικής συναρτήσεως στις δεδομένες συντεταγμένες.

Σχήμα Β.1: Κωδικοποίηση αντικειμενικών συναρτήσεων στην γλώσσα προγραμματισμού C.

```
int getdimension()  
{  
}
```

```
void getleftmargin(double *left)  
{  
}
```

```
void getrightmargin(double *right)  
{  
}
```

```
double funmin(double *x)  
{  
}
```

```
void granal(double *x,double *g)  
{  
}
```

Σχήμα Β.2: Κωδικοποίηση αντικειμενικών συναρτήσεων στην γλώσσα προγραμματισμού Fortran 77.

```
integer function getdimension()  
getdimension = d  
end
```

```
subroutine getleftmargin(left)  
double precision left(d)  
end
```

```
subroutine getrightmargin(right)  
double precision right(d)  
end
```

```
double precision function funmin(x)  
double precision x(d)  
end
```

```
subroutine granal(x,g)  
double precision x(d)  
double precision g(d)  
end
```

Σχήμα Β.3: Υλοποίηση της συναρτήσεως CAMEL στην γλώσσα προγραμματισμού C++.

```
extern "C" {
    int getdimension()
    {
        return 2;
    }

    void getleftmargin(double *left)
    {
        left[0]=-5.0;
        left[1]=-5.0;
    }

    void getrightmargin(double *right)
    {
        right[0]=5.0;
        right[1]=5.0;
    }

    double funmin(double *x)
    {
        double x1=x[0],x2=x[1];
        return 4*x1*x1-2.1*x1*x1*x1*x1+
            x1*x1*x1*x1*x1*x1/3.0+x1*x2-4*x2*x2+4*x2*x2*x2*x2;
    }

    void granal(double *x,double *g)
    {
        double x1=x[0],x2=x[1];
        g[0]=8*x1-8.4*x1*x1*x1+2*x1*x1*x1*x1*x1+x2;
        g[1]=x1-8*x2+16*x2*x2*x2;
    }
}
```

Σχήμα Β.4: Υλοποίηση της συναρτήσεως CAMEL στην γλώσσα προγραμματισμού Fortran 77.

```
integer function getdimension()  
getdimension = 2  
end
```

```
subroutine getleftmargin(left)  
double precision left(2)  
left(1)=-5.0  
left(2)=-5.0  
end
```

```
subroutine getrightmargin(right)  
double precision right(2)  
right(1)= 5.0  
right(2)= 5.0  
end
```

```
double precision function funmin(x)  
double precision x(2)  
double precision x1,x2  
x1=x(1)  
x2=x(2)  
funmin=4*x1**2-2.1*x1**4+x1**6/3.0+x1*x2-4*x2**2+4*x2**4  
end
```

```
subroutine granal(x,g)  
double precision x(2)  
double precision g(2)  
double precision x1,x2  
x1=x(1)  
x2=x(2)  
g(1)=8.0*x1-8.4*x1**3+2*x1**5+x2;  
g(2)=x1-8.0*x2+16.0*x2**3;  
end
```


ΠΑΡΑΡΤΗΜΑ Γ. ΤΑ ΠΑΚΕΤΑ GENPRICE ΚΑΙ GENANNEAL

Διανομή

Τα πακέτα που υλοποιήθηκαν ονομάζονται `GenPrice.tar.gz` και `GenAnneal.tar.gz`. Μετά από την αποσυμπίεση και εξαγωγή των περιεχομένων των δύο αρχείων δημιουργείται μια σειρά από υποκαταλόγους και αρχεία:

1. **bin**: Αυτός ο κατάλογος είναι αρχικά κενός. Μετά την μεταγλώττιση των πακέτων θα περιέχει ένα βοηθητικό εργαλείο το οποίο θα χρησιμοποιηθεί για την δημιουργία των τελικών εκτελεσίμων. Στην περίπτωση της `GenPrice` το εργαλείο αυτό ονομάζεται `make_genprice` και για την περίπτωση της `GenAnneal` ονομάζεται `make_genanneal`.
2. **doc**: Αυτός ο κατάλογος περιέχει το συνοδευτικό εγχειρίδιο του λογισμικού.
3. **examples**: Σε αυτόν τον κατάλογο περιέχονται διάφορες συναρτήσεις δοκιμής γραμμένες κυρίως σε ANSI C++ αλλά μερικές είναι γραμμένες και σε Fortran 77.
4. **include**: Ο κατάλογος αυτός περιέχει τα αρχεία συμπεριλήψεως για όλες τις κατηγορίες του πακέτου.
5. **src**: Στον κατάλογο αυτόν βρίσκονται τα αρχεία πηγαίου κώδικα του πακέτου.
6. **Makefile**: Είναι το αρχείο εισόδου για το εργαλείο `make`, που θα χρησιμοποιηθεί για την μεταγλώττιση του πακέτου. Ο χρήστης δεν χρειάζεται να αλλάξει κάτι σε αυτό το αρχείο.
7. **Makefile.inc**: Είναι το αρχείο που περιέχει διάφορες παραμέτρους που εξαρτώνται από το σύστημα εγκαταστάσεως, όπως το όνομα του χρησιμοποιούμενου μεταγλωττιστή, ο κατάλογος εγκαταστάσεως κτλ. Ο χρήστης

πρέπει να διαβάσει και να αλλάξει αυτό το αρχείο ανάλογα με το σύστημά του πριν την εκκίνηση της εγκατάστασής.

Εγκατάσταση

Τα βήματα για την εγκατάσταση και μεταγλώττιση των πακέτων είναι ίδια με αυτά που παρουσιάστηκαν για την περίπτωση του πακέτου MinFinder. Και στα δύο νέα πακέτα ο χρήστης πρέπει πρώτα να αλλάξει το αρχείο Makefile.inc ανάλογα με το σύστημα στο οποίο θα γίνει η εγκατάσταση και να στην συνέχεια να μεταγλωττίσει τα πακέτα με την εντολή make.

Κωδικοποίηση συναρτήσεων

Η κωδικοποίηση των συναρτήσεων ακολουθεί τα ίδια ακριβώς πρότυπα με αυτά του πακέτου MinFinder. Στα σχήματα B.1 και B.2 παρουσιάστηκε η κωδικοποίηση των συναρτήσεων σε ANSI C και Fortran 77.

Βοηθητικά εργαλεία

Μετά την μεταγλώττιση των πακέτων στον κατάλογο bin θα βρεθεί από ένα βοηθητικό πρόγραμμα για την δημιουργία της τελικής εφαρμογής, το οποίο θα ονομάζεται make_genprice για την περίπτωση της GenPrice και make_genanneal για την περίπτωση της GenAnneal. Και τα δύο εργαλεία έχουν τις ίδιες παραμέτρους γραμμής εντολών, όπως παρουσιάζονται στην συνέχεια:

1. **-h:** Εμφανίζει μια βοηθητική οθόνη και το πρόγραμμα τερματίζει.
2. **-p filename:** Η παράμετρος **filename** καθορίζει το όνομα του αρχείου που περιέχει την αντικειμενική συνάρτηση. Ανάλογα με την κατάληξη του αρχείου το καλείται ο αντίστοιχος μεταφραστής για να δημιουργήσει το τελικό εκτελέσιμο.
3. **-o filename:** Η παράμετρος **filename** καθορίζει το όνομα του τελικού εκτελέσιμου αρχείου. Η προκαθορισμένη παράμετρος είναι GenPrice για την περίπτωση της GenPrice και GenAnneal για την περίπτωση της μεθόδου GenAnneal.

Το τελικό εκτελέσιμο

Το τελικό εκτελέσιμο (GenPrice ή GenAnneal) έχει μια σειρά από παραμέτρους στην γραμμική εντολών:

1. `-h`: Το πρόγραμμα εμφανίζει μια βοηθητική οθόνη και τερματίζει.
2. `-c count`: Η ακέραια παράμετρος `count` καθορίζει τον αριθμό των χρωμοσωμάτων για την διαδικασία Genetic Random Search. Η προκαθορισμένη τιμή είναι 20.
3. `-s srate`: Η δεκαδική παράμετρος `srate` καθορίζει τον ρυθμό επιλογής που θα χρησιμοποιηθεί στην διαδικασία Genetic Random Search. Η προκαθορισμένη τιμή για αυτήν την παράμετρο είναι 0.10 (10%).
4. `-m mrate`: Η δεκαδική παράμετρος `mrate` καθορίζει τον ρυθμό μεταλλάξεως για την διαδικασία Genetic Random Search. Η προκαθορισμένη τιμή για αυτήν την παράμετρο είναι 0.05 (5%).
5. `-r seed`: Η ακέραια παράμετρος `seed` καθορίζει τον σπόρο για την γεννήτρια τυχαίων αριθμών.
6. `-o filename`: Η παράμετρος `filename` καθορίζει ένα αρχείο, όπου θα τοποθετηθεί η έξοδος του προγράμματος (ολικό ελάχιστο και κλήσεις). Αν δεν δοθεί κάποια παράμετρος η προκαθορισμένη τιμή είναι η καθιερωμένη έξοδος.

Παράδειγμα εφαρμογής

Έστω η συνάρτηση Rastrigin, με τύπο:

$$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \quad x \in [-1, 1]^2$$

Η συνάρτηση διαθέτει 49 τοπικά ελάχιστα. Η υλοποίηση της παραπάνω συναρτήσεως σε ANSI C++ δίνεται στο σχήμα Γ.1 και σε Fortran 77 στο σχήμα Γ.2. Έστω πως τα αντίστοιχα αρχεία ονομάζονται `rastrigin.cc` και `rastrigin.f` και βρίσκονται στον υποκατάλογο `examples`. Για την δημιουργία των τελικών εκτελεσίμων μπορούμε να δώσουμε στην γραμμική εντολών:

```
../bin/make_ganneal -p rastrigin.cc
```

ή για την έκδοση σε Fortran 77

```
../bin/make_ganneal -p rastrigin.f
```

Ανάλογα για την GenPrice δίνουμε τις εντολές

```
../bin/make_genprice -p rastrigin.cc
```

ή για την έκδοση σε Fortran 77

```
../bin/make_genprice -p rastrigin.f
```

Εκτελώντας την GenAnneal με την εντολή

```
./GenAnneal -c 100 -r 1
```

θα πάρουμε την έξοδο

```
FUNCTION EVALUATIONS = 468  
GRADIENT EVALUATIONS = 1  
MINIMUM = 0.000000 0.000000 -2.000000
```

Όμοια εκτελώντας την GenPrice θα λάβουμε την παρόμοια έξοδο:

```
FUNCTION EVALUATIONS = 513  
GRADIENT EVALUATIONS = 1  
MINIMUM = 0.000000 0.000000 -2.000000
```

Σχήμα Γ.1: Κωδικοποίηση της συναρτήσεως RASTRIGIN σε ANSI C++.

```
extern "C" {
int getdimension()
{
    return 2;
}

void getleftmargin(double *left)
{
    left[0]=-1.0;
    left[1]=-1.0;
}

void getrightmargin(double *right)
{
    right[0]=1.0;
    right[1]=1.0;
}

double funmin(double *x)
{
    double x1=x[0],x2=x[1];
    return x1*x1+x2*x2-cos(18.0*x1)-cos(18.0*x2);
}

void granal(double *x,double *g)
{
    double x1=x[0],x2=x[1];
    g[0]=2.0*x1+18.0*sin(18.0*x1);
    g[1]=2.0*x2+18.0*sin(18.0*x2);
}
}
```

Σχήμα Γ.2: Κωδικοποίηση της συναρτήσεως RASTRIGIN σε Fortran 77.

```
integer function getdimension()  
getdimension = 2  
end  
  
subroutine getleftmargin(left)  
double precision left(2)  
left(1)=-1.0  
left(2)=-1.0  
end  
  
subroutine getrightmargin(right)  
double precision right(2)  
right(1)= 1.0  
right(2)= 1.0  
end  
  
double precision function funmin(x)  
double precision x(2)  
double precision x1,x2  
x1=x(1)  
x2=x(2)  
funmin=x1**2+x2**2-cos(18.0*x1)-cos(18.0*x2)  
end  
  
subroutine granal(x,g)  
double precision x(2)  
double precision g(2)  
double precision x1,x2  
x1=x(1)  
x2=x(2)  
g(1)=2.0*x1+18.0*sin(18.0*x1)  
g(2)=2.0*x2+18.0*sin(18.0*x2)  
end
```

ΔΗΜΟΣΙΕΥΣΕΙΣ

Περιοδικά

1. I.G. Tsoulos and I.E. Lagaris, Solving differential equations with genetic programming, accepted for publication in Genetic Programming and Evolvable Machines.
2. I.G. Tsoulos and I.E. Lagaris, Genetically Controlled Random Search: A global optimization method for continuous multidimensional functions, accepted for publication in Computer Physics Communications.
3. I.G. Tsoulos and I.E. Lagaris, GenAnneal: Genetically modified Simulated Annealing, accepted for publication in Computer Physics Communications.
4. I.G. Tsoulos and I.E. Lagaris, Neural Splines: exploiting Parallelism for Function Approximation Using Modular Neural Networks, accepted for publication in Neural, Parallel and Scientific Computations.
5. I.G. Tsoulos and I.E. Lagaris, MinFinder: Locating all the local minima of a function, accepted for publication in Computer Physics Communications.
6. I.G. Tsoulos, D. Gavrilis and E. Dermatas, GDF: A tool for function estimation through grammatical evolution, accepted for publication in Computer Physics Communications.

Συνέδρια

1. V. V. Dimakopoulos, A. Kinalis, E. Pitoura, I. Tsoulos, On Deploying and Executing Data-Intensive Code on SMart Autonomous Storage (SmAS) Disks. *ADBIS-DASF*AA(2000), 323-330.

2. I.G. Tsoulos, I.E. Lagaris, A.Likas. Piecewise Neural Networks for Function Approximation, Cast in a Form Suitable for Parallel Computation. *SETN 2002*, 314-324.
3. D. Gavrilis, I.G. Tsoulos, E. Dermatas, Feature Selection for Robust Detection of Distributed Denial-of-Service Attacks Using Genetic Algorithms. *SETN 2004*, 276-281.
4. I.G. Tsoulos , D. Gavrilis, E. Glavas, Neural Network Construction using Grammatical Evolution, accepted in ISSPIT 2005.
5. D. Gavrilis, I. Tsoulos, E. Dermatas, Stochastic Classification of Scientific Abstracts, accepted in Specom 2005.
6. D. Gavrilis, I. Tsoulos, E. Dermatas, Evolutionary Grammar Induction for Protein Relation Extraction, accepted in Specom 2005.
7. D. Gavrilis, G. Georgoulas, I. G. Tsoulos, E. Glavas, Classification of Fetal Heart Rate using Grammatical Evolution, accepted in SIPS 2005.

ΒΙΟΓΡΑΦΙΚΟ

Γεννήθηκα στις 8/1/1977 στο Λαγανά Ηλείας και τελείωσα το Γενικό Λύκειο Σιμόπουλου το 1994 με βαθμό 19 και 5/11 πρώτος μεταξύ των συμμαθητών μου. Στην συνέχεια φοίτησα στο τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων από το οποίο αποφοίτησα το 1998 με βαθμό 7.73, πρώτος μεταξύ των συμφοιτητών μου. Τον Νοέμβριο του 2001 έλαβα το Μεταπτυχιακό Δίπλωμα Εξειδίκευσης από το ίδιο τμήμα με θέμα “Παράλληλα Τμηματικά Τεχνητά Νευρωνικά Δίκτυα” και έκτοτε εργάζομαι πάνω στο διδακτορικό μου με θέμα “Καθολική βελτιστοποίηση: μέθοδοι, λογισμικό και εφαρμογές”.