



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

**ΤΟΜΕΑΣ ΥΔΑΤΙΚΩΝ ΠΟΡΩΝ &
ΠΕΡΙΒΑΛΛΟΝΤΟΣ**

Διπλωματική Εργασία:

AMSEEAS: Ένα προσαρμοστικό υβριδικό σχήμα μηχανικής μάθησης, εξελικτικών αλγορίθμων και ανόπτησης – απλόκου για βελτιστοποίηση στοχικών συναρτήσεων

Σπυρίδων Μ. Τσατταλιός

**Επιβλέπων: Ανδρέας Ευστρατιάδης,
Επίκουρος Καθηγητής Ε.Μ.Π.**

Αθήνα, Ιούλιος 2021

Ευχαριστίες

Καθώς με την ολοκλήρωση της παρούσας διπλωματικής εργασίας, η προπτυχιακή μου φοίτηση στο Εθνικό Μετσόβιο Πολυτεχνείο φτάνει στο τέλος της, θα ήθελα και γω με τη σειρά μου να ευχαριστήσω από καρδιάς τα άτομα που συνέβαλαν, ο καθένας με το δικό του τρόπο, σε αυτή την προσπάθεια.

Θα ήθελα, αρχικά, να ευχαριστήσω θερμά τον Επίκουρο Καθηγητή του Τομέα Υδατικών Πόρων και Περιβάλλοντος της Σχολής Πολιτικών Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, κ. Ανδρέα Ευστρατιάδη, για την ανάθεση και επίβλεψη αυτής της εργασίας. Δίνοντας μου την ευκαιρία να συνεργαστώ μαζί του, ήρθα σε επαφή με τον κόσμο του προγραμματισμού και ειδικότερα της βελτιστοποίησης, έναν κόσμο που λάτρεψα από την πρώτη στιγμή. Παρά τις δυσκολίες και τις αντίξοες συνθήκες που επικράτησαν καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής, λόγω της πανδημίας του κορωνοϊού, κατορθώσαμε να έχουμε μια άψογη συνεργασία και να πετύχουμε ένα πολύ όμορφο αποτέλεσμα.

Επιπρόσθετα, θα ήθελα να ευχαριστήσω τον Καθηγητή και Κοσμήτορα της Σχολής Πολιτικών Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, κ. Νικόλαο Λαγαρό, όπως επίσης και τον Καθηγητή της Σχολής Πολιτικών Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, κ. Χρήστο Μακρόπουλο, για τη συμμετοχή τους στην εξεταστική επιτροπή της παρούσας εργασίας.

Δεν θα μπορούσα να παραλείψω, όμως, από αυτές τις ευχαριστίες και τους Διδάκτορες της Σχολής Πολιτικών Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, κ. Ιωάννη Τσουκαλά και κ. Παναγιώτη Κοσιέρη, οι οποίοι στάθηκαν δίπλα μου σε όλα τα στάδια της διπλωματικής και βοήθησαν στην παραγωγή του τελικού αποτελέσματος.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου εντός και εκτός Πολυτεχνείου.

Σπυρίδων Τσατταλιός,

Αθήνα, Ιούλιος 2021

Περιεχόμενα

Περίληψη	10
Extended Abstract	12
1 Εισαγωγή	27
1.1. Αντικείμενο της εργασίας	27
1.2. Πρωτότυπα σημεία.....	28
1.3. Διάρθρωση της εργασίας.....	28
2 Θεωρητικό υπόβαθρο της βελτιστοποίησης και επισκόπηση συναφών δημοφιλών μεθόδων και τεχνικών.....	30
2.1. Θεωρητικό υπόβαθρο της βελτιστοποίησης και θεμελιώδεις έννοιες	30
2.1.1. Η έννοια της βελτιστοποίησης από τη μαθηματική της σκοπιά	31
2.1.2. Συναρτήσεις πολλών μεταβλητών και υπολογισμός των ακροτάτων ...	32
2.2. Επισκόπηση δημοφιλών μεθόδων και τεχνικών βελτιστοποίησης	33
2.2.1. Η μέθοδος του κατερχόμενου απλόκου	33
2.2.2. Η μέθοδος της προσομοιωμένης ανόπτησης	36
2.2.3. Εξελκτικοί αλγόριθμοι.....	38
2.2.3.1. Γενετικοί αλγόριθμοι.....	38
3 Ο εξελκτικός αλγόριθμος ανόπτησης - απλόκου.....	42
3.1. Θεμελιώδεις αρχές του αλγορίθμου	42
3.2. Υπολογιστική διαδικασία.....	43
3.3. Υλοποίηση του εξελκτικού αλγορίθμου ανόπτησης - απλόκου σε γλώσσα προγραμματισμού Python	47
4 Εμπλουτισμός αλγορίθμων βελτιστοποίησης με τεχνικές υποκατάστατων μοντέλων	49
4.1. Η έννοια των υποκατάστατων μοντέλων και η χρησιμότητά τους στις διαδικασίες βελτιστοποίησης	49
4.2. Ο εξελκτικός αλγόριθμος ανόπτησης – απλόκου εμπλουτισμένος με υποκατάστατα μοντέλα	50

4.2.1.	Επισκόπηση του αλγορίθμου	51
4.2.2.	Acquisition Function.....	52
4.2.3.	Αναλυτική περιγραφή της υπολογιστικής διαδικασίας του SEEAS.....	52
4.2.4.	Υλοποίηση του SEEAS σε γλώσσα προγραμματισμού Python	58
5	Θεμελιώδεις αρχές μηχανικής μάθησης και επισκόπηση συναφών δημοφιλών μοντέλων	59
5.1.	Η έννοια της αυτοεκμάθησης μηχανών	59
5.2.	Περιγραφή δημοφιλών μοντέλων μηχανικής μάθησης.....	60
5.2.1.	Cubic Radial Basis Function with Linear Polynomial Tail	61
5.2.2.	Polynomial Regression	62
5.2.3.	Support Vector Machines (SVMs).....	63
5.2.4.	k – Nearest Neighbors.....	64
5.2.5.	Decision Trees	65
5.2.6.	Gaussian Processes	66
6	Ο αλγόριθμος AMSEEAS	67
6.1.	Γενική επισκόπηση του αλγορίθμου και πρωτότυπα σημεία.....	67
6.1.1.	Περιγραφή των πρωτότυπων σημείων του αλγορίθμου	70
6.2.	Αναλυτική περιγραφή της υπολογιστικής διαδικασίας του AMSEEAS	72
7	Συγκρίσεις και αξιολογήσεις των επιδόσεων των αλγορίθμων πάνω σε συναρτήσεις ελέγχου	77
7.1.	Συναρτήσεις ελέγχου.....	77
7.2.	Διερεύνηση της καλύτερης εκδοχής του AMSEEAS	79
7.3.	Σύγκριση της επίδοσης του AMSEEAS με άλλους δημοφιλείς αλγορίθμους βελτιστοποίησης.....	82
7.3.1.	Σύντομη περιγραφή των αλγορίθμων DDS, MLMSRBF, DYCORS....	83
7.3.2.	Συγκριτικοί πίνακες και σχολιασμός των αποτελεσμάτων.....	84
8	Συγκρίσεις και αξιολογήσεις των επιδόσεων των αλγορίθμων πάνω σε πρόβλημα βαθμονόμησης παραμέτρων υδρολογικού μοντέλου	88
8.1.	Βασικές έννοιες και ορισμοί	89
8.2.	Περιγραφή του μοντέλου υδατικού ισοζυγίου	90
8.3.	Χρήση συνθετικών χρονοσειρών για αποδοτικότερη βαθμονόμηση των παραμέτρων.....	93

8.4. Επίλυση του προβλήματος βαθμονόμησης και σύγκριση των αποτελεσμάτων των αλγορίθμων βελτιστοποίησης	94
9 Συμπεράσματα και προτάσεις για μελλοντική έρευνα	97
9.1. Συμπεράσματα	97
9.2. Προτάσεις για μελλοντική έρευνα	98
Αναφορές.....	101
Παράρτημα Α: Συναρτήσεις Ελέγχου	105
Παράρτημα Β: Καμπύλες σύγκλισης των 2 εκδοχών του AMSEEAS στις 6 συναρτήσεις ελέγχου	108

Ευρετήριο Σχημάτων

Σχήμα 2-1: Απεικόνιση των κινήσεων του κατερχόμενου απλόκου στις 2 διαστάσεις (Πηγή: Aloise, D., et. al., 2018. Less it more: Simplified Nelder-Mead method for large unconstrained optimization.).....	34
Σχήμα 2-2: Δυαδική κωδικοποίηση χρωμοσώματος (Πηγή: www.geeksforgeeks.org)	39
Σχήμα 2-3: Δεκαδική κωδικοποίηση χρωμοσώματος (Πηγή: www.geeksforgeeks.org)	39
Σχήμα 2-4: Διασταύρωση ενός σημείου (Πηγή: www.geeksforgeeks.org)	40
Σχήμα 2-5: Διασταύρωση δύο σημείων (Πηγή: www.geeksforgeeks.org)	41
Σχήμα 2-6: Γενετικός τελεστής μετάλλαξης (Πηγή: www.geeksforgeeks.org)	41
Σχήμα 3-1: Παραγωγή τυχαίων λύσεων μέσω μεθόδου LHS (Πηγή: www.wikipedia.org)	48
Σχήμα 3-2: Παραγωγή τυχαίων λύσεων μέσω ομοιόμορφης κατανομής (Πηγή: www.wikipedia.org)	48
Σχήμα 4-1: Διάγραμμα ροής της υπολογιστικής διαδικασίας του SEEAS (Πηγή: Tsoukalas, I., et. al., 2016. Surrogate-enhanced evolutionary annealing simplex algorithm for effective and efficient optimization of water resources problems on a budget.)	54
Σχήμα 5-1: Cubic RBF (Πηγή: www.stackoverflow.com).....	62
Σχήμα 5-2: Polynomial Regression (Πηγή: www.developpaper.com).....	62
Σχήμα 5-3: Support Vector Machine στον R^2 (Πηγή: www.javatpoint.com)	63
Σχήμα 5-4: k-Nearest Neighbors (Πηγή: www.researchgate.net)	64
Σχήμα 5-5: Τυπική δομή δέντρου απόφασης (Πηγή: www.researchgate.net)	65
Σχήμα 5-6: Gaussian Process (Πηγή: https://juanitorduz.github.io/)	66
Σχήμα 8-1: Εννοιολογική απεικόνιση των διεργασιών και παραμέτρων του υδρολογικού μοντέλου (Πηγή: Efstratiadis, A., et. al., 2021. Improving model identifiability by embedding stochastic simulation within hydrological calibration.)	91

Ευρετήριο Πινάκων

Πίνακας 7-1: Μέση τιμή και τυπική απόκλιση του ολικού ελαχίστου των 2 εκδοχών του AMSEEAS στα 24 προβλήματα	80
Πίνακας 7-2: Διάμεσος του ολικού ελαχίστου των 2 εκδοχών του AMSEEAS στα 24 προβλήματα	81
Πίνακας 7-3: Μέση τιμή και τυπική απόκλιση των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 15-D.....	84
Πίνακας 7-4: Μέση τιμή και τυπική απόκλιση των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 30-D.....	84
Πίνακας 7-5: Κατάταξη της απόδοσης των αλγορίθμων σύμφωνα με τη μέση τιμή των καλύτερων λύσεων που βρέθηκαν στα προβλήματα 15-D και 30-D	85
Πίνακας 7-6: Διάμεσος των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 15-D	86
Πίνακας 7-7: Διάμεσος των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 30-D	86
Πίνακας 7-8: Κατάταξη της απόδοσης των αλγορίθμων σύμφωνα με τη διάμεσο των καλύτερων λύσεων που βρέθηκαν στα προβλήματα 15-D και 30-D	86
Πίνακας 8-1: Μέση τιμή, τυπική απόκλιση και διάμεσος των καλύτερων τιμών του δείκτη NSE στο πρόβλημα βαθμονόμησης των παραμέτρων υδρολογικού μοντέλου για τους αλγορίθμους EAS, SEEAS και AMSEEAS	96

Περίληψη

Ο τομέας της βελτιστοποίησης αποτελεί αντικείμενο ευρείας μελέτης για τον υδραυλικό μηχανικό. Υπάρχει πληθώρα προβλημάτων που καλείται να επιλύσει αυτός σε καθημερινή βάση, τα οποία προκειμένου να αντιμετωπιστούν αποτελεσματικά, απαιτείται να γίνει χρήση μεθόδων και τεχνικών βελτιστοποίησης. Τέτοια προβλήματα συναντώνται συχνά στον τομέα των υδατικών πόρων. Δύο πολύ συνήθη ζητήματα του τομέα αυτού είναι το πρόβλημα λήψης αποφάσεων και το «αντίστροφο» πρόβλημα, δηλαδή το πρόβλημα «καλιμπραρίσματος» ενός μοντέλου. Και τα δύο αυτά είναι αδύνατο να επιλυθούν αποδοτικά, χωρίς τη χρήση της έννοιας της βελτιστοποίησης.

Με το πέρασμα των χρόνων, η επιστημονική κοινότητα έχει αναπτύξει πάρα πολλούς αλγορίθμους και τεχνικές βελτιστοποίησης, οι οποίες παρά τις διαφορές που εμφανίζουν στον πυρήνα τους, έχουν κοινό τελικό στόχο. Να αντιμετωπιστεί αποτελεσματικά το ζήτημα της βελτιστοποίησης συστημάτων, υπό την έννοια εύρεσης αλγορίθμων, που κατορθώνουν την αποτελεσματική σύγκλιση στο ολικό ελάχιστο των στοχικών συναρτήσεων, με τον ελάχιστο δυνατό υπολογιστικό φόρτο.

Ένα ζήτημα, το οποίο δυσχεραίνει σε μεγάλο βαθμό τη διαδικασία της βελτιστοποίησης στον τομέα των υδατικών πόρων, είναι η στοχική συνάρτηση. Αυτή, στην πλειονότητα τέτοιων εφαρμογών, δεν ορίζεται μέσω μιας απλής μαθηματικής έκφρασης, αλλά αποτελεί στην ουσία ένα μοντέλο προσομοίωσης, το οποίο πρέπει να τρέχει κάθε φορά που θέλουμε να αποκτήσουμε μια τιμή της. Το μοντέλο προσομοίωσης αυτό, όμως, απαιτεί ένα συγκεκριμένο χρονικό διάστημα για να τρέξει κάθε μεμονωμένη φορά, που δεν είναι σε καμία περίπτωση αμελητέο. Ιδίως σε περιπτώσεις που η στοχική συνάρτηση δημιουργεί διαταραγμένες και πολύπλοκες μη κυρτές επιφάνειες απόκρισης, ένας τυπικός αλγόριθμος βελτιστοποίησης, απαιτείται να κάνει χιλιάδες φορές κλήση της αντικειμενικής συνάρτησης για να κατορθώσει να εντοπίσει αποτελεσματικά το ολικό της ακρότατο. Αυτό, όμως δημιουργεί πολύ σημαντικά θέματα χρόνου, καθώς σε πολλές περιπτώσεις, μπορεί να απαιτηθούν ολόκληρες μέρες ή και εβδομάδες για να ολοκληρωθεί η διαδικασία.

Μια πολύ δημοφιλής τεχνική για την αντιμετώπιση αυτού του ζητήματος, είναι η χρήση των υποκατάστατων μοντέλων ή αλλιώς μεταμοντέλων. Αυτά, προσπαθούν να προσεγγίσουν σε κάποιο βαθμό την επιφάνεια απόκρισης του αρχικού μοντέλου προσομοίωσης, με τον χρόνο εκτέλεσης τους, όμως, να είναι ασύλληπτα μειωμένος. Το υποκατάστατο μοντέλο έχει ουσιαστικά την αρμοδιότητα να εντοπίζει, γρήγορα και επαρκώς, πιθανά «καλά» σημεία, τα οποία θα βοηθήσουν στην αποτελεσματική σύγκλιση του αλγορίθμου.

Στα πλαίσια της εργασίας αυτής, γίνεται, αρχικά, μια επισκόπηση της θεωρίας της βελτιστοποίησης και ορισμένων δημοφιλών μεθόδων και τεχνικών. Ακολουθεί η αναλυτική περιγραφή δύο συγκεκριμένων αλγορίθμων, του εξελικτικού αλγορίθμου ανόπτησης – απλόκου (EAS) και της επέκτασης αυτού, του λεγόμενου SEEAS, ο οποίος ενσωματώνει στον πυρήνα του EAS ένα υποκατάστατο μοντέλο για την επιτάχυνση της διαδικασίας. Οι δύο αυτοί αλγόριθμοι παραλαμβάνονται σε προγραμματιστικό περιβάλλον MATLAB και μεταφράζονται σε γλώσσα

προγραμματισμού Python. Εν συνεχεία, αναλύονται ορισμένα μοντέλα αυτοεκμάθησης μηχανών, τα οποία είναι εφικτό να ενσωματωθούν σε αλγορίθμους βελτιστοποίησης με τη μορφή υποκατάστατων μοντέλων, για επιτάχυνση της διαδικασίας. Τέλος, γίνεται η παρουσίαση ενός νέου πρωτότυπου κώδικα που αναπτύχθηκε στα πλαίσια της διπλωματικής. Αυτός ονομάζεται Adaptive Multi-Surrogate Enhanced Evolutionary Annealing Simplex algorithm (AMSEAS) και αποτελεί επέκταση του SEEAS. Η βασική ιδέα του αλγορίθμου έγκειται στην ταυτόχρονη χρήση πολλαπλών μεταμοντέλων, ενσωματωμένων στον ίδιο αλγόριθμο, τα οποία θα κατορθώνουν να συνυπάρχουν αρμονικά και να αλληλοβοηθούνται στα πλαίσια μιας ομάδας, επιτυγχάνοντας με τον τρόπο αυτό, την περαιτέρω βελτίωση της επίδοσης του SEEAS. Ο νέος αλγόριθμος υποβάλλεται σε συγκρίσεις με άλλους δημοφιλείς αλγορίθμους βελτιστοποίησης της διεθνούς κοινότητας, συμπεριλαμβανομένου και του SEEAS, μέσω χρήσης πολλών απαιτητικών συναρτήσεων ελέγχου, που δυσκολεύουν έντονα τη διαδικασία εύρεσης του κάθε ολικού ακροτάτου. Τα αποτελέσματα της σύγκρισης αυτής είναι άκρως ικανοποιητικά. Ο AMSEAS κυριαρχεί έναντι όλων των υπολοίπων, κατορθώνοντας, στην συντριπτική πλειονότητα των προβλημάτων που εξετάζονται, την καλύτερη σύγκλιση στο εκάστοτε ολικό ακρότατο. Επίσης, υλοποιούνται και οι καμπύλες σύγκλισης του αλγορίθμου σε όλα τα εξεταζόμενα προβλήματα, για να υπάρχει μια πλήρης εικόνα της ταχύτητας σύγκλισής του στο κάθε ολικό ακρότατο, υπό την έννοια εύρεσης του πλήθους υπολογισμών της στοχικής συνάρτησης, που απαιτούνται για να το προσεγγίσει. Πέραν αυτών, αξιολογείται και η επίδοση του AMSEAS πάνω σε ένα τυπικό πρόβλημα βαθμονόμησης παραμέτρων υδρολογικού μοντέλου.

Extended Abstract

Introduction

The field of optimization is a subject of extensive study for the hydraulic engineer. There is a variety of problems that have to be solved on a daily basis, which in order to be dealt with effectively, the use of optimization methods and techniques is required. Such problems are often encountered in the field of water resources. Two very common issues in this area are: 1) the decision-making problem and 2) the “reverse” problem, i.e. the model calibration problem. In the first problem, all system properties and related operations are known a priori and the goal is finding the optimal system management policy, whereas in the second problem some of the system properties are considered unknown and the goal is finding the optimal values for them. Both of these problems are impossible to be solved efficiently without the use of the concept of optimization.

To achieve optimization in a problem, it is necessary to consider a number of alternative choices and evaluate the impact of each option. Each individual option, that meets the constraints of the problem, is called feasible. The set of all possible options is called feasible space or else decision space or search space. Assuming that, each of these feasible options can be described by a set of variables, which are called control variables or decision variables or parameters $x = [x_1, x_2, \dots, x_n]$ and given that in each such description, a real performance measure can be assigned, then we consider as best, the option that maximizes this performance measure. The mathematical expression of the performance measure is called objective function and is usually expressed by $f(x)$.

Optimization, from a purely mathematical point of view, is the process by which, the optima of a function are located. A typical optimization problem, in which a representative objective function f has been defined, has the following form:

$$f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R} :$$
$$\min f(x), \quad g_i(x) \leq \alpha_i, \quad i = 1, 2, 3, \dots, m \quad (1)$$

where $x = [x_1, x_2, \dots, x_n]$ is the vector of control variables, $g_i(x)$ are the algebraic expressions of the restrictions of the problem and α_i , for $i=1, 2, \dots, m$ are the constants. Of course, in an optimization problem, there are not necessarily restrictions. It is therefore understandable, that a basic classification of optimization problems is to problems with control variables constraints and those without. These constraints, in general, may relate either to the variability limits of the decision variables, which is the simplest case, or they may be more complex and relate to operational requirements of the system.

It is also understandable, through equation (1), that an optimization problem is usually treated as a minimization problem of the objective function. This simplification does not affect the problem outcome, since even in the case where the maximum of an objective function is sought, the problem can be converted to the respective minimization problem, through the expression:

$$\max f(x) \equiv \min [-f(x)] \quad (2)$$

Overview of some popular optimization methods

Over the years, the scientific community has developed many algorithms and optimization techniques, which, despite their core differences, have a common end goal. That is, to effectively address the system optimization issue, in the sense of finding algorithms that achieve effective convergence to the overall minimum of the objective function, with the least possible computational burden. Some popular optimization techniques, that were vital in the completion of this thesis, are the following:

- The Nelder-Mead method
- The simulated annealing method
- Evolutionary algorithms

The Nelder-Mead algorithm: A simplex is defined as the geometric convex shape, of non-zero volume, defined by a set of $n + 1$ points, non-coplanar, in the n -dimensional search space \mathbb{R}^n . A simplex practically represents the minimum number of points required, in order to be able to numerically calculate the slope of the objective function. The descending simplex method is based on this geometric shape and the movements it can perform. Over the years, more and more algorithms are developed using a simplex, but the original idea is attributed to Spendley, Hext and Himsworth (1962). In the first algorithm of Spendley et al. (1962), the simplex is formed by $n + 1$ peaks, for each of which, the value of the objective function is calculated. The peak with the worst value (i.e. the largest value for minimization problems) is located and replaced by its anti-diametric, relatively to the centroid of the other peaks. This movement performed by the simplex is called reflection and leads to the formation of a new simplex with a new direction and, at the same time, maintaining the volume of the original. If the new peak, resulting from the reflection, is the worst in the new simplex that is formed, then the peak with the second worst value is selected for reflection, in order to disallow the return to the previous state and therefore the reproduction of the same solutions .

This idea, of using a simplex and executing the reflection motion, was very innovative, but not enough to produce an efficient optimization algorithm. That's why Nelder and Mead come forth, developing the basic algorithm in 1965, by inserting three new movements to the simplex, which are expansion, shrinkage and contraction. The combination of these four moves created a new algorithm, significantly improving the performance of the original.

The simulated annealing method : Let f be an objective function of an optimization problem, where $f : S \rightarrow \mathbb{R}$, with S being a set of distinct solutions and let x be an initial problem solution. In the proximity of this x solution, an optimization method is applied (e.g. the Nelder-Mead algorithm) and a new solution y emerges. At this new point, the value of the objective function $f(y)$ is calculated. If $f(y) < f(x)$, then solution y is definitely accepted. If $f(y) > f(x)$, i.e. the solution y does not improve the value of the objective function, the solution y is accepted, provided that some probabilistic criterion $p(x, y, T)$ is satisfied. The probabilistic criterion of Metropolis et al. (1983), where $T > 0$ is the system temperature, $p(x, y, T)$ is given by the following equation:

$$p(x, y, T) = \begin{cases} 1, & f(y) < f(x) \\ e^{-\frac{f(y)-f(x)}{T}}, & f(y) > f(x) \end{cases} \quad (3)$$

Evolutionary Algorithms: Evolutionary algorithms are computational methods, to solve optimization problems, modeled on the natural evolutionary process and are inspired by the principles of biological evolution (C. Darwin's Theory of Evolution, 1858). Evolutionary algorithms are based on the simulation of the evolution of a population of feasible points, P, through computational processes derived from natural selection, reproduction and mutation. The concept of natural selection refers to the idea that the strongest members of a species are more likely to survive, while the concept of reproduction refers to the ability of children to inherit the genetic characteristics of their parents. Finally, the concept of mutation refers to the fact that random changes in genetic material can lead to species differentiation.

The most popular and important subclass of evolutionary algorithms are the so-called genetic algorithms. In genetic algorithms, each solution of the problem is a member of the population and is called an individual. Each individual is composed of specific genetic characteristics, referred to as chromosomes. Chromosomes regard the position of the solution in the search space and consist of the so-called genes. In order to finally find a solution to the optimization problem, the population evolves in stages and the solutions population in one step is called a generation. The production of the first population is performed randomly, and then, population evolution from generation to generation is performed by the genetic operators, which are selection, crossover and mutation. Each new solution of the problem produced through genetic operators is called offspring and solutions, which participated in the creation of this offspring, are called its parents.

Within the operation of a typical genetic algorithm, the population of each generation undergoes a simulated genetic evolution, through which, the next solutions generation will consist mainly of the most powerful solutions and the least powerful will be removed from it. To make the separation of solutions into strong and weak possible, a fitness function is required, which practically is the objective function of the optimization problem. To create a fully developed and functional genetic algorithm, it is required for the following features to be included in its structure:

- an appropriate way to genetically represent the problem solutions
- suitable means for creating the initial population
- finding a fitness function, suitable for the evaluation of the individuals in the population
- utilization of the genetic operators, to create new solutions to the problem
- determining the critical parameters of the algorithm, such as the size of the population and the probability of performing each genetic operator

The Evolutionary Annealing - Simplex algorithm

The evolutionary annealing-simplex (EAS) algorithm is a heuristic probabilistic global optimization method, developed by A. Efstratiadis (2001, 2008), that attempts the

coupling of a range of different methodologies, with the simultaneous addition of new original elements. EAS attempts to combine the Nelder-Mead algorithm with the simulated annealing method, incorporating, simultaneously, fundamentals and concepts of evolutionary algorithms. The basic idea of the algorithm is to combine, the ability of the simulated annealing to escape from local optima, with the ability of the simplex to locate an area of attraction quickly and accurately. At the same time, the algorithm uses basic principles of evolutionary algorithms, such as the concept of population, solution generations and the genetic operator of mutation.

The basic principles of the algorithm are the following:

- An initial population is generated randomly and it is gradually improved from generation to generation, approaching the total optima.
- The production of new solutions is done through the simplex. The simplex performs the well-known movements of the Nelder-Mead method i.e. reflection, expansion, contraction and shrinkage, but in EAS, the element of randomness is added to the movement step, thus making the process of population evolution more stochastic.
- Randomness, in the process of searching for new solutions, is adjusted through a self-adjusting annealing schedule, which depending on the case, can allow the acceptance of solutions, that worsen the value of the objective function, attempting, in these cases, to free the algorithm from any local optima. At the start of the algorithm, the system temperature is set equal to the difference between the minimum and maximum value of the objective function in the initial population. However, as the population evolves, the system begins a cooling process i.e. the system temperature, and thus its randomness, begins to decrease. The user of the algorithm, in fact, for the best execution in each specialized case, is required to enter two parameters, β and λ , which relate to the temperature regulation. Parameter β determines when the cooling function of the system is activated, while parameter λ determines how much the temperature will decrease, each time a traction area is detected by the algorithm.
- To evaluate the peaks of the simplex formed in each iteration of the algorithm and to find its worst peak, the values of the objective function of each peak are not being compared alone, but a random term relative to the current system temperature is added. Of course, the best peak of the simplex does not participate in this evaluation, so that there is no possibility of it being replaced.
- If in some iteration of the algorithm, the movements performed by the simplex do not succeed in improving the solutions, then a mutation mechanism is activated, creating, this way, diversity in the solutions of the population.

As part of this thesis, an attempt was made to translate EAS into Python programming language. The algorithm was initially received in MATLAB programming environment. That was followed by the understanding of its basic principles and computational procedures, so that its successful translation is enabled. The following link presents its implementation:

<https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/EAS>

Fundamental principles of surrogate models

One issue, that greatly complicates the process of optimizing water resources problems, is the objective function. This, in most such applications, is not defined by a simple mathematical expression, but is essentially a simulation model, that must be run every time we want to obtain a value of it. This simulation model, however, requires a certain amount of time to run each time, which is by no means negligible. Especially in cases where the objective function creates complex, non-convex response surfaces, a typical optimization algorithm is required to call the objective function thousands of times in order to effectively locate its total optima. This, unfortunately, creates very important time issues, as in many cases, it may take days or even weeks to complete the procedure.

Among the many tactics proposed by the international community to address this issue, a very popular one, with proven effectiveness, is the use of surrogate models. Those, attempt with various ideas, to approach, to some extent, the actual response surface of the original simulation model, with their execution time, being significantly reduced. Surrogate models are essentially machine learning models, that can be integrated into any optimization algorithm, to help speed up its convergence. That is, a surrogate model has the ability to identify quickly and adequately, possible “good” points, that help the converging process, and only in these "good" points is the value of the "time-consuming" real objective function calculated.

The Surrogate-Enhanced Evolutionary Annealing Simplex algorithm

The Surrogate-Enhanced Evolutionary Annealing Simplex algorithm (SEEAS) is a global optimization algorithm developed by I. Tsoukalas, P. Kossieris, A. Efstratiadis and C. Makropoulos (2016) and it is specially designed for "time consuming" objective functions, which represent the majority of functions in water resources problems. The goal of SEEAS is the accurate approach of the global optima, with the minimum possible number of objective function calculations.

SEEAS is coupling the evolutionary annealing-simplex algorithm (EAS) with a surrogate model. The surrogate model used, is called Cubic RBF with Linear Polynomial Tail and it is a subcategory of a broader class of metamodels, called Radial Basis Functions (RBFs).

During the iterative process of the code, an external file is maintained, that includes all the points, in which, the real "time-consuming" objective function has been calculated. The external file exists, so that the surrogate model can make the most accurate fit on all known points and thus the metamodel can simulate the response of the actual objective function the best way possible. If this is achieved, it will be able to make accurate predictions of possible "good" points, which will help to converge to the global optima and in which points, it is worthwhile calculating the actual objective function. In addition to the above, the surrogate model in SEEAS helps the descending simplex to explore in the right directions and perform its known movements. More specifically, the predictions made by the metamodel are imported into a function, called the Acquisition Function (AF). AF makes the final predictions, indicating the points, at which the calculation of the actual objective function is made, afterwards.

A typical iteration of the algorithm, begins with adapting the surrogate model to the external file. The original version of EAS is then called in to minimize the Acquisition Function, and once this process is completed, a new point is generated (i.e. the global minimum of AF), which is a candidate, to enter the population. At the point resulted from the minimization, the value of the real objective function is calculated and if it is less (and thus better) than the value of the worst point in the population, it replaces it. Then follows the production process of the simplex, which performs the known movements of the Nelder-Mead method in combination with the self-adjusting annealing schedule, to add randomness to the algorithm and help escape from local optima. The only thing that changes during this known process, is that all the effort is supported by the metamodel, which helps to explore directions for performing the simplex movements. At the end of each iterative cycle of the algorithm, at least one new point is generated and enters the population, replacing one of its preexisting members.

As part of this thesis, an attempt was made to translate SEEAS into Python programming language. SEEAS was initially received in MATLAB programming environment. That was followed by an understanding of its basic principles and computational procedures, so that its successful translation can be enabled. The following link presents its implementation:

<https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/SEEAS>

The concept of machine learning

Machine learning is a branch of computer science, that uses statistical techniques, to give computer systems the ability to "learn" from data and make predictions, without following a deterministic algorithm. The scientific field of machine learning is part of the broader branch of artificial intelligence. Optimization, in the context of water resources, belongs to the section of supervised learning and any regression model (namely the surrogate model) can be used for addressing the problem.

The Adaptive Multi-Surrogate Enhanced Evolutionary Annealing Simplex algorithm (AMSEEAS)

An important part of the present study was devoted to the effort of developing an original algorithm, based on SEEAS, that aims to further improve its performance. That being said, during the translation of SEEAS into Python programming language, a lot of thought was put into finding a way to make a drastic change to the core of the code, that would make a significant difference to the results. During this search process, the following two questions arose:

- 1) Why is EAS specifically supported by a Cubic RBF with Linear Polynomial Tail, as a metamodel and not by one of the other popular ones in the international community?
- 2) Why is EAS enhanced using a single surrogate model and not more?

These two issues led to the search for incorporation of different metamodels on EAS in an attempt to develop some kind of hybrid system, which consists of many surrogate models. Over the years, a wide variety of machine learning (ML) models has been developed, using different ideas to produce "smart" computer systems. However, it is well known that there is no model that outperforms the rest in every single problem category. It is, therefore, necessary for the programmer, to try many different ML models in each individual problem and to ultimately select the one/ones, who seem to have the best response to the specific data. For the production of the final version of AMSEEAS, many different combinations were examined, in which the following machine learning models were used: 1) Decision Trees, 2) Random Forests, 3) Extra Trees, 4) Gradient Boosting, 5) Ada Boosting, 6) Gaussian Processes, 7) Cubic RBF with Linear Polynomial Tail, 8) Support Vector Machines, 9) Polynomial Regression, 10) Artificial Neural Networks, 11) Naïve Bayes, 12) k-Nearest Neighbors.

The computational procedure of AMSEEAS is identical to SEEAS. In this case, as well, the descending simplex algorithm is used, combined with the simulated annealing method, as it was introduced in EAS and the whole effort is supported by surrogate models. However, there is a key difference in the new code. Instead of using only the Cubic RBF with Polynomial Tail as a metamodel, a hybrid system is now being developed, in which many different surrogate models coexist, functioning cooperatively as a group. More specifically, since there is no machine learning model clearly superior to the others and any of them can respond better or worse depending on the optimization problem, the incorporation of many different ML models in the same algorithm is likely to result in more favorable outcomes. In order for this idea to actually be effective, it is necessary to find an efficient way for the many models to coexist in the same algorithm. The aim is to significantly improve the convergence speed of the algorithm (i.e. to achieve convergence with the minimum possible number of objective function calculations) and at the same time not to burden the original SEEAS in terms of computer workload.

Out of the many different versions of algorithms created, searching for which machine learning models will be integrated into the final algorithm and how they will coexist and collaborate effectively, the final version that seemed to be superior to all of the rest, was the one in which, the following 6 different machine learning models were used:

- Polynomial Regression
- Support Vector Machine
- Cubic RBF with Linear Polynomial Tail combined with the Acquisition Function, as it was introduced in SEEAS
- k-Nearest Neighbors
- Decision Tree
- Gaussian Process

For the coexistence of these 6 different models in the same algorithm, the idea of imposing some kind of penalty on models, that do not have satisfactory behavior, is introduced.

In more detail, the following rationale is implemented:

- The Polynomial Regression metamodel, in the absence of penalty, will be activated in the 1st iteration of the algorithm, then the 7th, the 13th, the 19th etc. The Support Vector Machine model, in the absence of penalty, will be activated in the

2nd algorithm iteration, then the 8th, the 14th, the 20th etc. Similarly, the RBF will be activated in the 3rd, 9th, 15th, 21st etc., the k-NN in the 4th, 10th, 16th, 22nd etc., the Decision Tree in the 5th, 11th, 17th, 23rd etc. and, finally, the Gaussian Process surrogate in the 6th, 12th, 18th, 24th etc. algorithm iteration. This, of course, is not absolute, as there are penalties, that can be imposed on the models, as explained below.

Since in each iteration, only one metamodel of the above is activated, it is understandable, that the algorithm is hardly burdened in terms of computational workload. However, the idea of this hybrid system is that, it is desirable for the insufficient models, that do not make good adjustments to some optimization problems, to "step aside", so that iterations and calculations of the objective function are not wasted without improving the convergence. Thus, the models that make the best predictions are allowed to take control and guide the convergence process. In order to make this reasoning work, the idea of imposing penalty on models, that do not make good predictions, is introduced into the algorithm. The definition of this penalty idea is explained through the following example:

The Polynomial Regression metamodel will definitely be activated on the 1st algorithm iteration. It will adapt to the points, that currently exist in the external file and then, an optimization algorithm (specifically a genetic algorithm embedded from a Python library) will minimize the, now trained, Polynomial Regression surrogate. From this minimization process, one new candidate point will emerge (i.e. the total minimum of the curve). This point is the Polynomial Regression predicted value and therefore a new solution to the problem. What follows, is the calculation of the actual "time consuming" objective function at this candidate point and depending on whether this solution is good or not, it enters the population or is discarded, respectively. Two different paths are created here:

- 1st case : If the solution resulting from the prediction is good and inserted in the population, then the Polynomial Regression metamodel is not penalized and will be reactivated, as planned, on the 7th algorithm iteration.
- 2nd case : If the solution resulting from the prediction is rejected by the population, then the specific surrogate is penalized. The penalty imposed, is the absence of activation in the 7th, 13th etc. iterations., contradicting the initial algorithm course. To specify, the metamodel remains excluded from the algorithm for a specific number of iterations, defined by the programmer (in the final version, the penalty imposed on each surrogate is exclusion from the code for its 10 next iterations).

If it happens, that all the metamodels are in penalty state in a specific iteration, then one by one, the penalty rounds of each metamodel are sequentially reduced, until the first one without remaining penalty is found and, thus, activated.

The following link introduces the AMSEEAS source code, coded in Python programming language:

<https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/AMSEEAS>

Comparison of AMSEAS to other popular optimization algorithms within a benchmarking "suite" composed of six known mathematical functions

For the following comparisons, a popular benchmarking "suite" composed of six known mathematical functions is used, each of which has its own unique complexity, rendering the approximation process of the total optima, quite demanding. This "suite" consists of the following functions:

- 1) Sphere Function (Objective Function 1 - OF1)
- 2) Ackley Function (Objective Function 2 - OF2)
- 3) Griewank Function (Objective Function 3 - OF3)
- 4) Zakharov Function (Objective Function 4 - OF4)
- 5) Rastrigin Function (Objective Function 5 - OF5)
- 6) Levy Function (Objective Function 6 - OF6)

The mathematical formula for each of the above is given in the [Appendix A](#), however here it is mentioned that, each of these 6 functions has a total minimum value $f_{\min} = 0$.

In order to compare the algorithms more sufficiently and accurately, for each of these 6 functions, 4 different problems were studied:

- 1) The problem consists of $n = 15$ control variables and the maximum allowed number of objective function evaluations is $MFE = 500$.
- 2) The problem consists of $n = 15$ control variables and the maximum allowed number of objective function evaluations is $MFE = 1000$.
- 3) The problem consists of $n = 30$ control variables and the maximum allowed number of objective function evaluations is $MFE = 500$.
- 4) The problem consists of $n = 30$ control variables and the maximum allowed number of objective function evaluations is $MFE = 1000$.

In this way, a total of $6 \times 2 \times 2 = 24$ different problems arise for comparison. To collect a sample of sufficient size, each algorithm will "run" each of these 24 problems 30 independent times. Obviously, the common goal of all algorithms in each "run" performed, is locating the best approximation of the total optima of each function, i.e. the closest convergence of them to the 0 value. Thus, in each algorithm run, the best approximation of the total minimum is recorded.

To evaluate the performance of each algorithm in these 24 problems, from the total sample of the 30 "runs" performed, the statistical parameters of each sample are calculated, namely the mean value, the standard deviation and the median. Then, two different comparative tables are created. In the first table, the comparisons are made according to the mean value of the minimum that has been estimated from the 30 "runs", while in the second according to the median of the sample.

The goal of each algorithm is its dominance over the rest, in as many of the 24 total problems as possible, i.e. achieving the lowest mean value and the lowest median of the total minimum in these 30 "runs".

In order to make the comparisons fair, all optimization algorithms will use the same population size in each generation, equal to $m = 32$ in problems with $n = 15$ control variables and $m = 62$ in problems with $n = 30$ control variables. In addition, in all algorithms, the initial random solutions are produced using the same statistical method, namely the Latin Hypercube Sampling (LHS) method. Finally, in all algorithms, the default values for all parameters / hyperparameters are set, as suggested in their user manuals.

The 5 different optimization algorithms, that AMSEEAS is compared to, in order to evaluate its performance, are the following:

- 1) The Evolutionary Annealing Simplex (EAS) algorithm
- 2) The DDS algorithm
- 3) The Surrogate-Enhanced Evolutionary Annealing Simplex algorithm
- 4) The DYCORS algorithm
- 5) The MLMSRBF algorithm

Of the above, SEEAS, DYCORS and MLMSRBF incorporate a surrogate model in their core, namely the Cubic RBF with Linear Polynomial Tail, while EAS and DDS don't.

First, the resulting comparative tables are presented, using the **mean value** of each sample as a performance metric:

Mean and standard deviation of best solutions in 15-D test problems (optimal results are highlighted)													
MFE	Test Function	EAS		DDS		SEEAS		DYCORS		MLMSRBF		AMSEEAS (GA)	
		Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
500	OF1	1.938	0.978	0.852	0.479	0.002	0.001	0.002	0.001	0.019	0.014	0.0002	0.0002
	OF2	7.159	1.723	6.025	1.314	0.812	0.233	0.809	0.372	2.231	0.658	0.002	0.002
	OF3	7.682	2.997	2.626	1.269	0.538	0.118	0.885	0.084	1.085	0.052	0.853	0.118
	OF4	39.434	14.894	137.447	52.366	59.144	28.023	158.669	47.788	150.411	49.875	50.708	24.828
	OF5	86.245	14.148	24.887	7.081	46.268	15.359	38.958	12.340	45.920	18.803	30.651	18.990
	OF6	1.905	0.877	0.681	0.314	0.203	0.105	1.208	1.406	1.344	2.129	0.231	0.441
1000	OF1	0.378	0.177	0.150	0.079	0.001	0.001	0.001	0.000	0.011	0.007	0.0003	0.0002
	OF2	3.523	0.936	3.847	0.528	0.437	0.208	0.607	0.092	1.862	0.556	0.0002	0.0003
	OF3	2.444	1.061	1.505	0.299	0.368	0.140	0.809	0.082	1.040	0.037	0.833	0.118
	OF4	26.828	17.895	97.541	38.226	41.290	26.639	121.266	36.925	121.359	37.730	29.956	13.854
	OF5	59.735	17.012	11.233	3.136	29.733	12.838	33.585	13.490	35.784	11.031	19.689	9.266
	OF6	0.767	0.292	0.234	0.104	0.124	0.060	0.536	0.860	0.524	0.863	0.241	0.388

Table 1: Mean value and standard deviation of the best solutions found by the algorithms in 15-D problems

Mean and standard deviation of best solutions in 30-D test problems (optimal results are highlighted)													
MFE	Test Function	EAS		DDS		SEEAS		DYCORS		MLMSRBF		AMSEEAS (GA)	
		Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
500	OF1	4.305	1.163	9.516	2.737	0.019	0.006	0.083	0.034	0.739	0.708	0.003	0.002
	OF2	9.923	1.160	12.872	1.329	1.878	0.301	4.297	3.721	6.193	4.362	0.032	0.046
	OF3	17.866	3.455	38.398	12.050	0.782	0.118	1.265	0.079	3.459	1.927	0.983	0.038
	OF4	117.821	28.757	562.145	113.230	173.240	44.185	472.815	90.897	575.424	174.073	298.921	116.243
	OF5	228.693	18.442	132.149	24.567	122.658	19.427	112.046	23.076	165.437	46.846	177.644	20.262
	OF6	6.338	2.652	15.823	5.481	0.659	0.184	3.407	2.540	7.326	10.944	0.660	0.714
1000	OF1	2.529	0.933	2.112	0.791	0.006	0.004	0.011	0.004	0.358	0.177	0.003	0.003
	OF2	6.516	0.845	7.670	0.924	1.206	0.297	1.085	0.168	3.643	1.103	0.002	0.003
	OF3	8.836	2.617	8.273	2.679	0.549	0.093	1.020	0.026	2.420	0.713	1.005	0.106
	OF4	94.598	20.317	412.238	118.573	151.472	54.097	403.812	93.081	491.425	146.097	175.583	59.006
	OF5	198.335	16.587	71.598	15.028	98.371	19.505	85.267	22.956	134.864	39.193	54.790	42.263
	OF6	2.683	0.736	3.921	2.215	0.443	0.126	4.213	5.440	2.865	4.583	0.239	0.184

Table 2: Mean value and standard deviation of the best solutions found by the algorithms in 30-D problems

	EAS	DDS	SEEAS	DYCORS	MLMSRBF	AMSEEAS (GA)
1ος (φορές)	4	2	7	1	0	11
2ος (φορές)	0	2	10	5	0	8
3ος (φορές)	1	2	5	10	0	4
4ος (φορές)	1	2	1	4	16	0
5ος (φορές)	6	9	1	2	5	1
6ος (φορές)	12	7	0	2	3	0

Table 3 : Algorithms performance rankings according to the mean value of the best solutions found in 15-D and 30-D problems

As shown in Table 3, if the mean value is used as a performance metric, AMSEEAS ranks first in 11/24 optimization problems. The original SEEAS version comes second, as expected, occupying the 1st place in 7/24 optimization problems. SEEAS is followed by EAS, managing to get a win in 4/24 total cases. DDS prevails in 2/24 problems and finally DYCORS wins in 1/24 problems. The MLMSRBF algorithm is not superior to the others in any of the 24 total problems. It is noteworthy, that AMSEEAS occupies the 1st or 2nd place in 19/24 problems, while 1st or 2nd or 3rd in 23/24.

More specifically, in 15-D optimization problems, AMSEEAS achieves the best performance (i.e. the lowest mean value of the total of 30 "runs") in 2 of the 6 objective functions (OF1, OF2) for MFE = 500 and in the same 2 out of the 6 (OF1, OF2) for MFE = 1000. Moreover, in 30-D problems, in which, the increase in dimensions obviously increases the difficulty of approaching the optima, AMSEEAS dominates in 3 of the 6 functions (OF1, OF2, OF6) for MFE = 500 and in 4 out of 6 (OF1, OF2, OF5, OF6) for MFE = 1000. From the comparison results, it is also understandable, that none of the 6 optimization algorithms manages to accurately approach the total minimum of the Zakharov (OF4) and Rastrigin (OF5) objective functions, which is not surprising, as the very complex response surface of these two functions, makes it extremely difficult to adapt any metamodel to them.

As expected, the increase of the maximum allowable number of objective function evaluations from MFE = 500 to MFE = 1000, improves the performance of all algorithms. Greater improvement, of course, is seen in EAS and DDS algorithms, which makes sense, since both of them are not supported by a surrogate model and, thus, the convergence proceeds at a slower rate than in the other algorithms.

Now, the resulting comparative tables are presented, using the **median** of each sample as a performance metric:

<i>Median of best solutions in 15-D test problems (optimal results are highlighted)</i>							
MFE	Test Function	EAS	DDS	SEEAS	DYCORS	MLMSRBF	AMSEAS (GA)
		Median	Median	Median	Median	Median	Median
500	OF1	1.457	0.684	0.002	0.002	0.012	0.0001
	OF2	7.367	5.942	0.838	0.745	2.353	0.001
	OF3	7.446	2.312	0.513	0.921	1.088	0.875
	OF4	34.205	133.574	53.874	154.151	147.998	44.470
	OF5	85.223	24.714	45.061	37.912	37.696	29.880
	OF6	1.592	0.616	0.198	0.681	0.488	0.090
1000	OF1	0.380	0.131	0.001	0.001	0.008	0.0002
	OF2	3.519	3.877	0.410	0.574	1.629	0.0001
	OF3	2.211	1.400	0.360	0.819	1.027	0.865
	OF4	25.224	98.089	34.413	127.557	110.313	27.715
	OF5	58.926	10.813	31.808	32.644	34.522	18.465
	OF6	0.765	0.216	0.114	0.069	0.191	0.130

Table 4: Median of the best solutions found by the algorithms in 15-D problems

<i>Median of best solutions in 30-D test problems (optimal results are highlighted)</i>							
MFE	Test Function	EAS	DDS	SEEAS	DYCORS	MLMSRBF	AMSEAS (GA)
		Median	Median	Median	Median	Median	Median
500	OF1	4.391	9.828	0.018	0.073	0.590	0.002
	OF2	9.844	13.110	1.918	3.144	4.725	0.020
	OF3	17.758	36.453	0.807	1.249	2.974	0.990
	OF4	114.878	540.070	168.695	456.956	570.266	276.105
	OF5	232.766	130.090	121.973	112.009	156.834	179.590
	OF6	5.264	15.496	0.630	2.075	2.302	0.475
1000	OF1	2.516	1.860	0.005	0.009	0.270	0.003
	OF2	6.579	7.831	1.170	1.108	3.438	0.001
	OF3	8.741	7.920	0.554	1.025	2.507	0.990
	OF4	95.274	386.140	147.120	409.986	465.057	170.215
	OF5	200.952	71.160	97.994	85.728	127.299	39.465
	OF6	2.458	2.918	0.431	2.762	1.412	0.190

Table 5: Median of the best solutions found by the algorithms in 30-D problems

	EAS	DDS	SEEAS	DYCORS	MLMSRBF	AMSEAS (GA)
1ος (φορές)	4	2	4	2	0	12
2ος (φορές)	0	1	13	5	0	7
3ος (φορές)	0	1	5	9	3	4
4ος (φορές)	1	4	1	4	15	0
5ος (φορές)	6	9	1	2	4	1
6ος (φορές)	13	7	0	2	2	0

Table 6: Algorithms performance rankings according to the median of the best solutions found in 15-D and 30-D problems

As shown in Table 6, the results in this case are even more impressive. If the median of each sample is used as a performance metric of the algorithms, AMSEAS dominates in 12/24 optimization problems. Tied at 2nd place are SEEAS and EAS, prevailing in 4/24 problems. DYCORS and DDS follow with dominance in 2/24 cases each, and finally the MLMSRBF algorithm is not superior to the others in any of the 24 total problems. It is noteworthy, that AMSEAS occupies the 1st or 2nd place in 19/24 problems, while 1st or 2nd or 3rd in 23/24.

More specifically, in 15-D optimization problems, AMSEAS achieves the best performance (i.e. the lowest median in the total of 30 "runs") in 3 of the 6 objective functions (OF1, OF2, OF6) for MFE = 500 and in 2 of the 6 (OF1, OF2) for MFE = 1000. Moreover, in the 30-D problems, AMSEAS dominates in 3 of the 6 functions

(OF1, OF2, OF6) for MFE = 500 and in 4 of the 6 (OF1, OF2, OF5, OF6) for MFE = 1000. That is, it outperforms the rest of the algorithms in the exact same cases, as those found using the mean value as a performance metric, with an additional dominance in the Levy function (OF6) for n = 15 control variables and MFE = 500 evaluations.

Indeed, the results of comparing AMSEEAS to these popular optimization algorithms, confirm the potential and success of the idea.

Comparison of AMSEEAS to other optimization algorithms within a stochastic calibration problem

In order to further investigate the effectiveness of AMSEEAS, it is considered necessary, in addition to comparing it to other optimization algorithms on mathematical functions with known algebraic expressions, an analysis on a typical problem of the water resources field to be performed. The problem examined is the known stochastic calibration problem and its goal is adjusting the optimal values for the control variables of a hydrologic model, which are considered unknown a priori. These parameters, although having some physical meaning, are impossible to be calculated through the system properties of the model and their assessment requires applying an optimization process. The aim of the problem is finding the appropriate values for the parameters, in order to minimize the discrepancy between the simulated response of the model and the actual measurements at the outlet of the system.

In order to evaluate the deviation of the simulated response of the model from the measured one, a performance metric is required. This metric will be in essence the objective function of the problem. A very widespread and popular metric is the so-called coefficient of determination or else efficiency or Nash-Sutcliffe efficiency (NSE) and is given by the following formula:

$$NSE = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \mu_y)^2} \quad (4)$$

where y_i and \hat{y}_i represent the measured and the simulated runoff time-series, respectively, n is the total length of the model calibration period and μ_y is the mean value of the measured runoff. The NSE metric is dimensionless and its values range between $-\infty$ and 1. If the value $NSE = 1$ is reached, this means that there is an absolute match between the simulated and the observed time series, i.e. the error is zero. Negative NSE values indicate a very poor fit of the model and as it is understandable, the goal is achieving NSE values, that approach the value $NSE = 1$.

A schematic view of the conceptual model, the associated fluxes as well as the equations representing the calculation procedure, is given in the following Figure:

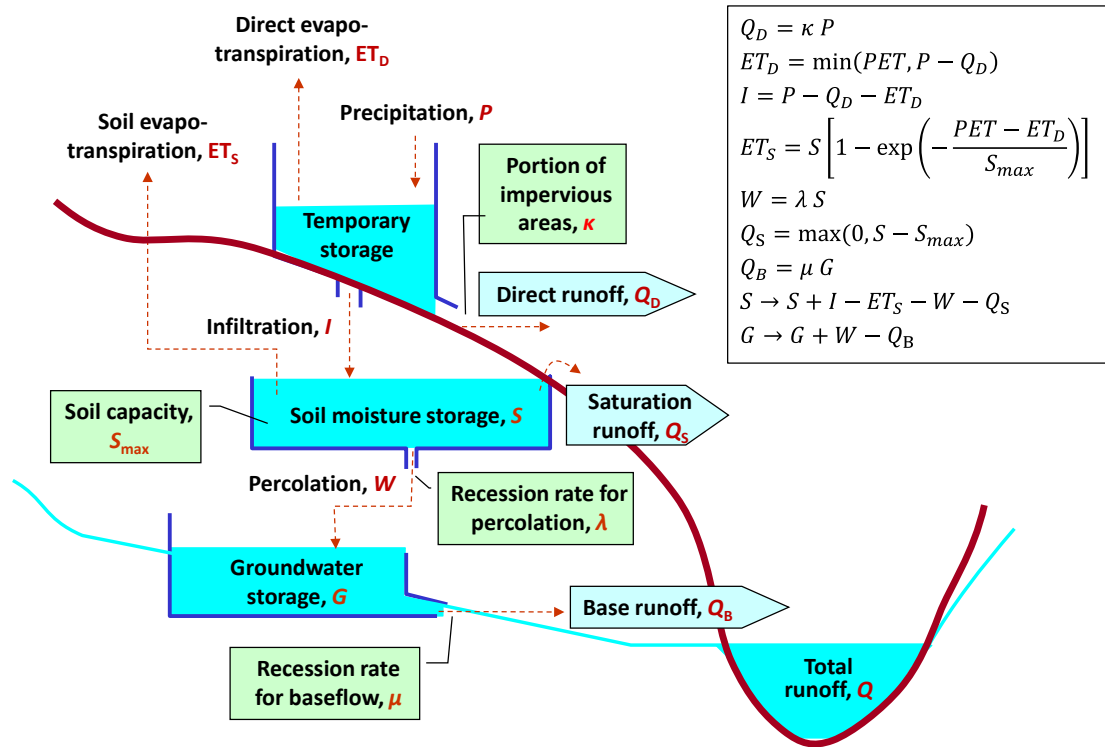


Figure 1: Schematic view of the conceptual model and the associated fluxes and parameters; governing equations are shown in the top right panel. (Source: Efstratiadis, A., et. al., 2021. Improving model identifiability by embedding stochastic simulation within hydrological calibration.)

The problem studied, refers to the area upstream of Evinos dam, Western Greece, which is a key component of the water supply system of Athens. For a total period of 50 years, the monthly time series of rainfall, potential evapotranspiration and runoff are available. The problem is described by a typical hydrological model, as shown in the figure above, and consists of 4 total control variables: 1) the impermeable surface percentage ν , 2) the storage capacity of the soil moisture tank K , 3) the soil moisture sterilization coefficient κ and 4) the groundwater sterilization coefficient λ . The ν , κ , λ parameters range between 0 and 1, while K between 0 and 700mm.

The 50-year historical data, along with the help of a stochastic model, called anySim (<https://www.itia.ntua.gr/en/softinfo/33/>), produced synthetic time series for the rainfall, the potential evapotranspiration and the runoff, 2048 years long each. The total length of the synthetic time series is utilized during the calibration process, i.e. the process of finding the optimal values for the model parameters, while the total length of the historical time series is used for validation. Both of these time series are available in the following links:

- https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/historical_time_series.csv
- https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/synthetic_time_series.csv

As a criterion of good fit, the NSE metric is chosen, which will therefore be the objective function of the problem, i.e. $f(x) \equiv \text{NSE}$. More specifically, the optimization algorithm used by the model, will optimize the NSE coefficient, that results from the total synthetic time series of data, 2048 years long. Upon completion of this process,

the optimal values of the control variables will be generated. Then, one final model execution will follow, using the now known values of the model parameters, on the whole historical time series, 50 years long, and the value of the corresponding NSE, from this last execution, will be recorded. This value will represent the performance of the optimization algorithm on the calibration problem.

The problem will be solved using 3 different optimization algorithms, in order to compare the effectiveness of each. These are:

- 1) The Evolutionary Annealing-Simplex algorithm (EAS)
- 2) The Surrogate-Enhanced Evolutionary Annealing-Simplex algorithm (SEEAS)
- 3) The Adaptive Multi-Surrogate Enhanced Evolutionary Annealing-Simplex algorithm (AMSEEAS)

In order to better investigate the problem, 3 different cases are studied, in which the maximum allowable number of function evaluations is changed between MFE = 100, MFE = 500 and MFE = 1000 evaluations. In each of these cases, 30 individual executions are performed and the best value of the NSE coefficient in the total length of the historical time series is recorded, after the completion of every execution.

All algorithms use the same population size ($m = 10$) and the same method to generate the initial random solutions, namely the LHS method. In addition, the default values for all parameters / hyperparameters are set, as suggested in the user manuals of the algorithms.

After the completion of all executions, the statistical parameters of all samples are calculated (i.e. the mean, standard deviation and median value) and a comparative table is created, so that the performance of all algorithms can be evaluated.

The final comparative table, emerged from the analysis, mentioned above, is as follows:

Mean, standard deviation and median value of best Nash-Sutcliffe efficiency (NSE) metrics found by the algorithms on the calibration problem									
MFE	EAS			SEEAS			AMSEEAS		
	MEAN	STDEV	MEDIAN	MEAN	STDEV	MEDIAN	MEAN	STDEV	MEDIAN
100	0.764	0.007	0.762	0.762	0.007	0.761	0.761	0.006	0.760
500	0.770	0.007	0.775	0.771	0.006	0.774	0.775	0.003	0.776
1000	0.773	0.006	0.776	0.775	0.003	0.776	0.776	0.000	0.776

Table 7: Mean, standard deviation and median value of best Nash-Sutcliffe efficiency (NSE) metrics found by the algorithms on the calibration problem

As shown in Table 7, if the mean value of each sample is used as a performance metric, AMSEEAS dominates over the rest in 2 / 3 cases, while in the 3rd case, the 1st place is occupied by EAS. SEEAS does not prevail in any of the 3 total problems. On the contrary, if the median value of each sample is used as a performance metric, both AMSEEAS and EAS dominate in 1 / 3 cases, while in the 3rd case, all 3 algorithms achieve the same performance. Here, as well, SEEAS does not dominate in any of the 3 problems. However, all 3 algorithms find very similar results in all cases.

Κεφάλαιο 1^ο

1 Εισαγωγή

1.1. Αντικείμενο της εργασίας

Αντικείμενο της παρούσας εργασίας αποτελεί η μελέτη μεθόδων και τεχνικών βελτιστοποίησης με στόχο την αξιοποίησή τους για την επίλυση προβλημάτων του τομέα υδατικών πόρων. Στα συνήθη προβλήματα του τομέα αυτού, η στοχική συνάρτηση δεν ορίζεται μέσω μια απλής μαθηματικής έκφρασης, αλλά προκύπτει από την εφαρμογή ενός μοντέλου προσομοίωσης. Το μοντέλο προσομοίωσης αυτό, δημιουργεί περίπλοκες, διαταραγμένες και μη κυρτές επιφάνειες απόκρισης, με αποτέλεσμα, οι αλγόριθμοι βελτιστοποίησης να απαιτείται να καλέσουν χιλιάδες φορές την αντικειμενική συνάρτηση, προκειμένου να καταφέρουν να εντοπίσουν με ικανοποιητική ακρίβεια το ολικό ακρότατο. Εκτός αυτού, όμως, το μοντέλο προσομοίωσης απαιτεί και ένα χρονικό διάστημα για να τρέχει κάθε φορά, το οποίο δεν είναι αμελητέο. Ο συνδυασμός, αυτός, της πολυπλοκότητας της επιφάνειας απόκρισης και του χρόνου που απαιτείται για την εκτέλεση του μοντέλου προσομοίωσης, καθιστά τους αλγορίθμους ανίκανους να καταφέρουν να ολοκληρώσουν τη συγκλητική διαδικασία εντός λογικών χρονικών πλαισίων. Ανάμεσα στις αρκετές τακτικές που προτείνονται από τη διεθνή κοινότητα για την αντιμετώπιση του προβλήματος αυτού, μια πολύ δημοφιλής με αποδεδειγμένη αποτελεσματικότητα, είναι η χρήση των υποκατάστατων μοντέλων ή αλλιώς μεταμοντέλων. Αυτά, προσπαθούν, πρακτικά, με διάφορες «πατέντες», να προσεγγίσουν σε κάποιο βαθμό την πραγματική επιφάνεια απόκρισης του αρχικού μοντέλου προσομοίωσης, με τον χρόνο εκτέλεσης τους, όμως, να είναι ασύλληπτα μειωμένος. Τα υποκατάστατα μοντέλα, αποτελούν στην ουσία μοντέλα αυτοεκμάθησης μηχανών, τα οποία ενσωματώνονται στον οποιοδήποτε αλγόριθμο βελτιστοποίησης, για να βοηθήσουν και να επιταχύνουν τη σύγκλισή του. Στα πλαίσια της εργασίας μελετώνται διεξοδικά δύο ήδη υλοποιημένοι αλγόριθμοι, ο εξελικτικός αλγόριθμος ανόπτησης – απλόκου (EAS) και η επέκταση αυτού, ο λεγόμενος εξελικτικός αλγόριθμος ανόπτησης – απλόκου εμπλουτισμένος με υποκατάστατα μοντέλα (SEEAS). Οι δύο αυτοί αλγόριθμοι παραλαμβάνονται, αρχικά, σε προγραμματιστικό περιβάλλον MATLAB και μεταφράζονται σε γλώσσα Python. Πέραν αυτών, επιχειρείται η δημιουργία ενός νέου πρωτότυπου αλγορίθμου, βασισμένο πάνω στον SEEAS, που έχει ως στόχο την περαιτέρω βελτίωση της επίδοσης του. Ο κώδικας αυτός ονομάζεται AMSEEAS και υλοποιείται, επίσης, σε γλώσσα Python. Ο νέος αλγόριθμος υποβάλλεται σε συγκρίσεις με άλλους δημοφιλείς αλγορίθμους βελτιστοποίησης της διεθνούς κοινότητας για την αξιολόγηση και την εκτίμηση της αποδοτικότητάς του.

1.2. Πρωτότυπα σημεία

Στα πλαίσια της εργασίας, υλοποιείται ένας πρωτότυπος αλγόριθμος βελτιστοποίησης, ο λεγόμενος Adaptive Multi-Surrogate Enhanced Evolutionary Annealing Simplex algorithm (AMSEES). Ο αλγόριθμος αυτός αποτελεί επέκταση του SEES, ο οποίος αποτελεί με τη σειρά του επέκταση του EAS. Ο EAS είναι μια ευρετική πιθανοτική μέθοδος ολικής βελτιστοποίησης, στην οποία επιχειρείται η σύζευξη ενός φάσματος διαφορετικών μεθοδολογιών, με την ταυτόχρονη προσθήκη νέων πρωτότυπων στοιχείων. Ο αλγόριθμος προσπαθεί να συνδυάσει, ουσιαστικά, μια εδραιωμένη τεχνική, τη μέθοδο του κατερχόμενου απλόκου, με μια άλλη που ονομάζεται μέθοδος προσομοιωμένης ανόπτωσης. Ταυτόχρονα, ενσωματώνονται θεμελιώδη στοιχεία και έννοιες από μια τρίτη κατηγορία αλγορίθμων, τους λεγόμενους εξελικτικούς αλγορίθμους. Οι θεμελιώδεις αρχές των τριών αυτών κατηγοριών αλγορίθμων αναλύονται διεξοδικά στα πλαίσια της εργασίας. Ο SEES, από την άλλη, αποτελεί επέκταση του EAS, καθώς στον πυρήνα του αρχικού κώδικα, ενσωματώνεται ένα υποκατάστατο μοντέλο, προκειμένου να επιταχυνθεί η συγκλητική διαδικασία. Αυτό που διαφοροποιεί, τώρα, τον SEES από τον AMSEES, είναι ότι στον τελευταίο, επιχειρείται η ταυτόχρονη χρήση πολλαπλών μεταμοντέλων, ενσωματωμένων στον ίδιο αλγόριθμο, τα οποία θα κατορθώνουν να συνυπάρχουν αποδοτικά και να αλληλοβοηθούνται στα πλαίσια μιας ομάδας, επιτυγχάνοντας, έτσι, περαιτέρω βελτίωση της επίδοσης. Ο νέος αλγόριθμος αποδεικνύει την κυριαρχία του έναντι άλλων δημοφιλών αλγορίθμων της βιβλιογραφίας, συμπεριλαμβανομένου και του SEES, μέσω συγκρίσεων που πραγματοποιούνται σε πολλές συναρτήσεις ελέγχου με γνωστή αλγεβρική έκφραση, όπως επίσης και σε ένα τυπικό πρόβλημα βαθμονόμησης παραμέτρων υδρολογικού μοντέλου.

1.3. Διάρθρωση της εργασίας

Η διπλωματική εργασία, συμπεριλαμβανομένου αυτού του 1^{ου} κεφαλαίου της εισαγωγής, αποτελείται συνολικά από **9 Κεφάλαια**, τις **Αναφορές** και **2 Παραρτήματα**.

Στο **2^ο Κεφάλαιο** γίνεται μια εισαγωγή στη θεωρία της βελτιστοποίησης και αναλύονται ορισμένες δημοφιλείς μέθοδοι και τεχνικές, που έχουν καθοριστική σημασία για την υλοποίηση της υπόλοιπης εργασίας.

Στο **3^ο Κεφάλαιο** παρουσιάζεται ο εξελικτικός αλγόριθμος ανόπτωσης – απλόκου (EAS) και περιγράφεται διεξοδικά η υπολογιστική του διαδικασία.

Στο **4^ο Κεφάλαιο** επιδεικνύεται η συμβολή που έχουν τα υποκατάστατα μοντέλα στην επιτάχυνση της διαδικασίας βελτιστοποίησης και ακολούθως παρουσιάζεται διεξοδικά ο εξελικτικός αλγόριθμος ανόπτωσης – απλόκου εμπλουτισμένος με υποκατάστατα μοντέλα (SEES).

Στο **5^ο Κεφάλαιο** γίνεται μια εισαγωγή στη θεωρία της αυτοεκμάθησης μηχανών και περιγράφονται οι θεμελιώδεις αρχές ορισμένων μοντέλων μηχανικής μάθησης, χρήσιμων για τη διεκπεραίωση του υπολοίπου της εργασίας.

Στο **6^ο Κεφάλαιο** παρουσιάζεται ο νέος πρωτότυπος κώδικας (AMSEEAS), που υλοποιήθηκε στα πλαίσια αυτής της εργασίας. Περιγράφονται αναλυτικά οι θεμελιώδεις αρχές του αλγορίθμου, τα πρωτότυπα σημεία του και η υπολογιστική του διαδικασία.

Στο **7^ο Κεφάλαιο** πραγματοποιείται σύγκριση του AMSEEAS με άλλους δημοφιλείς αλγορίθμους βελτιστοποίησης της διεθνούς κοινότητας πάνω σε πολλές συναρτήσεις ελέγχου με γνωστή αλγεβρική έκφραση και σχολιάζονται τα αποτελέσματα που προέκυψαν.

Στο **8^ο Κεφάλαιο** αναλύεται διεξοδικά ένα τυπικό πρόβλημα του τομέα των υδατικών πόρων, αυτό της βαθμονόμησης παραμέτρων υδρολογικού μοντέλου και αξιολογείται η επίδοση του AMSEEAS πάνω σε αυτό.

Στο **9^ο Κεφάλαιο** πραγματοποιείται μια επισκόπηση των συμπερασμάτων που διαπιστώθηκαν από την εκπόνηση αυτής της διπλωματικής και προτείνονται ιδέες, οι οποίες αξίζει να μελετηθούν σε μελλοντικές έρευνες.

Στις **Αναφορές** παρουσιάζεται η ελληνική και διεθνής βιβλιογραφία που χρησιμοποιήθηκε για την εκπόνηση της εργασίας.

Στο **Παράρτημα Α** υπάρχουν οι μαθηματικές εκφράσεις όλων των συναρτήσεων ελέγχου που χρησιμοποιήθηκαν για την πραγματοποίηση συγκρίσεων μεταξύ των αλγορίθμων.

Στο **Παράρτημα Β** υπάρχουν οι καμπύλες σύγκλισης, πάνω σε 6 συναρτήσεις ελέγχου, των δύο εκδοχών του AMSEEAS, που αναπτύχθηκαν στα πλαίσια της εργασίας.

Κεφάλαιο 2^ο

2 Θεωρητικό υπόβαθρο της βελτιστοποίησης και επισκόπηση συναφών δημοφιλών μεθόδων και τεχνικών

2.1. Θεωρητικό υπόβαθρο της βελτιστοποίησης και θεμελιώδεις έννοιες

Η βελτιστοποίηση (optimization) είναι ένα μέσο, το οποίο παίζει καθοριστικό ρόλο στη λήψη αποφάσεων (decision making). Η έννοια της βελτιστοποίησης βρίσκει χρησιμότητα σε ένα ευρύ φάσμα εφαρμογών και μπορεί να αξιοποιηθεί για την επίλυση πάρα πολλών διαφορετικών κατηγοριών προβλημάτων. Στα προβλήματα αυτά εντάσσονται και τα προβλήματα υδατικών πόρων (water resources problems). Πιο συγκεκριμένα, υπάρχει πληθώρα ζητημάτων, τα οποία καλείται να επιλύσει σε καθημερινή βάση ένας υδραυλικός μηχανικός και των οποίων η επίλυση απαιτεί τη χρήση της έννοιας της βελτιστοποίησης. Τέτοια ζητήματα είναι για παράδειγμα ο σχεδιασμός δικτύων ύδρευσης ή δικτύων ακαθάρτων, η κατασκευή αντιπλημμυρικών έργων, ο σχεδιασμός υδροσυστημάτων και η διαχείριση των υδατικών πόρων.

Για να επιτευχθεί βελτιστοποίηση σε ένα πρόβλημα λήψης κάποιας απόφασης, είναι αναγκαίο να εξεταστούν μια σειρά από εναλλακτικές επιλογές (alternatives) και να γίνει αξιολόγηση των επιπτώσεων της κάθε επιλογής. Κάθε μεμονωμένη επιλογή, η οποία ικανοποιεί τους περιορισμούς του προβλήματος καλείται εφικτή (feasible) επιλογή. Το σύνολο των εφικτών επιλογών ονομάζεται εφικτός χώρος (feasible space) ή χώρος αποφάσεων (decision space) ή χώρος αναζήτησης (search space). Αν υποθέσουμε ότι κάθε μια από αυτές τις εφικτές επιλογές μπορεί να περιγραφεί από ένα σύνολο μεταβλητών, οι οποίες αποκαλούνται μεταβλητές ελέγχου (control variables) ή μεταβλητές απόφασης (decision variables) ή παράμετροι (parameters) $x = [x_1, x_2, \dots, x_n]$ και δεδομένου ότι σε κάθε τέτοια περιγραφή μπορεί να αντιστοιχιστεί ένα πραγματικό μέτρο επίδοσης, τότε θεωρούμε ως βέλτιστη απόφαση, την επιλογή εκείνη που μεγιστοποιεί αυτό το μέτρο επίδοσης. Η μαθηματική έκφραση του μέτρου επίδοσης ονομάζεται στοχική ή αντικειμενική συνάρτηση (objective function) και συμβολίζεται, συνήθως, με $f(x)$. Η στοχική συνάρτηση μπορεί να είναι βαθμωτή ή διανυσματική ανάλογα με το αν το μέτρο επίδοσης περιλαμβάνει ένα ή περισσότερα κριτήρια, αντίστοιχα.

2.1.1. Η έννοια της βελτιστοποίησης από τη μαθηματική της σκοπιά

Βελτιστοποίηση, από καθαρά μαθηματική σκοπιά, είναι η διαδικασία μέσω της οποίας επιτυγχάνεται ο εντοπισμός των ακροτάτων μιας συνάρτησης. Πιο αναλυτικά, αν εντοπίζεται το ολικό ακρότατο (μέγιστο ή ελάχιστο) της συνάρτησης στο συνολικό πεδίο ορισμού, τότε η διαδικασία αυτή αναφέρεται ως ολική βελτιστοποίηση (global optimization), ενώ αν εντοπίζεται ένα τοπικό ακρότατο σε μια υποπεριοχή του πεδίου, τότε αναφέρεται ως τοπική βελτιστοποίηση (local optimization).

Ένα τυπικό πρόβλημα βελτιστοποίησης, στο οποίο έχει γίνει ο ορισμός κάποιας αντιπροσωπευτικής αντικειμενικής συνάρτησης f , έχει την ακόλουθη μορφή:

$$f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R} :$$
$$\min f(x), \quad g_i(x) \leq \alpha_i, \quad i = 1, 2, 3, \dots, m \quad (2.1)$$

όπου $x = [x_1, x_2, \dots, x_n]$ είναι το διάνυσμα των μεταβλητών ελέγχου πλήθους n , $g_i(x)$ είναι οι αλγεβρικές συναρτήσεις των περιορισμών του προβλήματος και $\alpha_i, i=1,2,3,\dots,m$ είναι οι σταθερές. Βέβαια σε ένα πρόβλημα βελτιστοποίησης, δεν είναι υποχρεωτικό να υπάρχουν περιορισμοί. Γίνεται λοιπόν κατανοητό, ότι μια βασική κατηγοριοποίηση των προβλημάτων βελτιστοποίησης είναι αυτή σε προβλήματα βελτιστοποίησης με περιορισμούς και σε προβλήματα βελτιστοποίησης χωρίς περιορισμούς, ανάλογα με το εάν οι μεταβλητές ελέγχου πρέπει να ικανοποιούν κάποιους περιορισμούς ή όχι. Οι περιορισμοί αυτοί, γενικά, μπορούν να αφορούν είτε στα όρια διακύμανσης των μεταβλητών απόφασης, που είναι και η απλούστερη περίπτωση, είτε να είναι πιο περίπλοκοι και να αφορούν σε λειτουργικές απαιτήσεις του συστήματος.

Μια άλλη κατηγοριοποίηση των προβλημάτων βελτιστοποίησης είναι σε προβλήματα γραμμικής (linear programming) και μη γραμμικής (non - linear programming) βελτιστοποίησης, ανάλογα με το αν οι συναρτήσεις $f(x)$ και $g_i(x)$ είναι γραμμικές ή μη γραμμικές αντίστοιχα.

Γίνεται επίσης αντιληπτό, με τη βοήθεια της εξίσωσης (2.1), ότι συνηθίζεται ένα πρόβλημα βελτιστοποίησης να αντιμετωπίζεται ως πρόβλημα ελαχιστοποίησης (minimization) της αντικειμενικής συνάρτησης, που το περιγράφει. Αυτή η συνήθεια δεν επηρεάζει την γενικότητα του προβλήματος, αφού και στην περίπτωση, όπου αναζητείται το μέγιστο κάποιας αντικειμενικής συνάρτησης, το πρόβλημα μπορεί να αναχθεί στο αντίστοιχο πρόβλημα ελαχιστοποίησης, μέσω της σχέσης:

$$\max f(x) \equiv \min [-f(x)] \quad (2.2)$$

Επομένως, για το υπόλοιπο της παρούσας εργασίας, το πρόβλημα βελτιστοποίησης θα αντιμετωπίζεται ως πρόβλημα ελαχιστοποίησης της αντικειμενικής συνάρτησης.

2.1.2. Συναρτήσεις πολλών μεταβλητών και υπολογισμός των ακροτάτων

Έστω μια συνάρτηση $f(x) : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, όπου $x = [x_1, x_2, \dots, x_n]$ διάνυσμα πραγματικών μεταβλητών, οι οποίες λαμβάνουν τιμές από το πεδίο X . Η f θεωρούμε ότι είναι ομοιόμορφα συνεχής, και έχει συνεχείς μερικές παραγώγους πρώτης και δευτέρας τάξης στο X . Ορίζεται ως κλίση ή βαθμίδα της $f(x)$ το διάνυσμα:

$$\nabla f(x) = \text{grad}f(x) = \left[\frac{df}{dx_1}, \frac{df}{dx_2}, \dots, \frac{df}{dx_n} \right]^T \quad (2.3)$$

το οποίο περιλαμβάνει όλες τις μερικές παραγώγους 1^{ης} τάξης της $f(x)$.

Αντίστοιχα ορίζεται και το λεγόμενο Εσσιανό Μητρώο (Hessian Matrix), το οποίο περιλαμβάνει όλες τις μερικές παραγώγους 2^{ης} τάξης της $f(x)$:

$$H = \begin{bmatrix} \frac{d^2f}{dx_1^2} & \frac{d^2f}{dx_1 dx_2} & \dots & \frac{d^2f}{dx_1 dx_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{d^2f}{dx_n dx_1} & \frac{d^2f}{dx_n dx_2} & \dots & \frac{d^2f}{dx_n^2} \end{bmatrix} \quad (2.4)$$

Λέμε ότι η $f(x)$ παρουσιάζει τοπικό ελάχιστο σε ένα σημείο $x^* \in X$ όταν υπάρχει περιοχή $X_0 \subseteq X$ του x^* για την οποία να ισχύει για κάθε $x \in X_0$:

$$f(x^*) < f(x) \quad (2.5)$$

Με αντίστοιχο τρόπο ορίζεται και το τοπικό μέγιστο. Τα τοπικά ελάχιστα και τα τοπικά μέγιστα μιας συνάρτησης ονομάζονται γενικότερα τοπικά ακρότατα της συνάρτησης. Στην ειδική περίπτωση, όπου το $X_0 \equiv X$ τότε τα ακρότατα ονομάζονται ολικά ακρότατα.

Για να είναι εφικτός ο αναλυτικός υπολογισμός των ακροτάτων της συνάρτησης, είναι προαπαιτούμενο να υπάρχουν αναλυτικές μαθηματικές εξισώσεις και για την f αλλά και για τις παραγώγους 1^{ης} και 2^{ης} τάξης αυτής. Στην περίπτωση αυτή, κάθε σημείο $x^* \in X$ για το οποίο ισχύει ότι:

$$\nabla f(x^*) = \text{grad}f(x^*) = 0 \quad (2.6)$$

καλείται στάσιμο σημείο και στην πράξη είναι ένα πιθανό τοπικό ακρότατο. Για να εξαχθεί όμως το τελικό συμπέρασμα σχετικά με την ιδιότητα του σημείου x^* πρέπει να γίνει εύρεση του Εσσιανού Μητρώου H και εν συνεχεία να γίνει η ακόλουθη διερεύνηση:

Αν υποθέσουμε ότι $H_i(x)$ είναι η i υπο-ορίζουσα του Εσσιανού Μητρώου, όπου προκύπτει μετά από αφαίρεση των $n - i$ τελευταίων γραμμών και στηλών του, τότε:

- Στην περίπτωση όπου $H_i(x^*) > 0$ για το κάθε i , τότε το x^* είναι τοπικό ελάχιστο.
- Στην περίπτωση όπου $H_i(x^*) \neq 0$ για το κάθε i , και $\text{sign}(H_i) = \text{sign}(-1)^i$, τότε το x^* είναι τοπικό ελάχιστο.

- Στην περίπτωση όπου $H_n(x^*) \neq 0$ για το κάθε i και δεν ισχύει καμία από τις παραπάνω συνθήκες, τότε το x^* ονομάζεται σημείο σέλλας, είναι δηλαδή η τομή επιπέδων, των οποίων το ένα παρουσιάζει ελάχιστο και το άλλο μέγιστο.
- Στην περίπτωση όπου $H_n(x^*) = 0$ για το κάθε i , τότε δεν μπορεί να εξαχθεί κάποιο συμπέρασμα.

2.2. Επισκόπηση δημοφιλών μεθόδων και τεχνικών βελτιστοποίησης

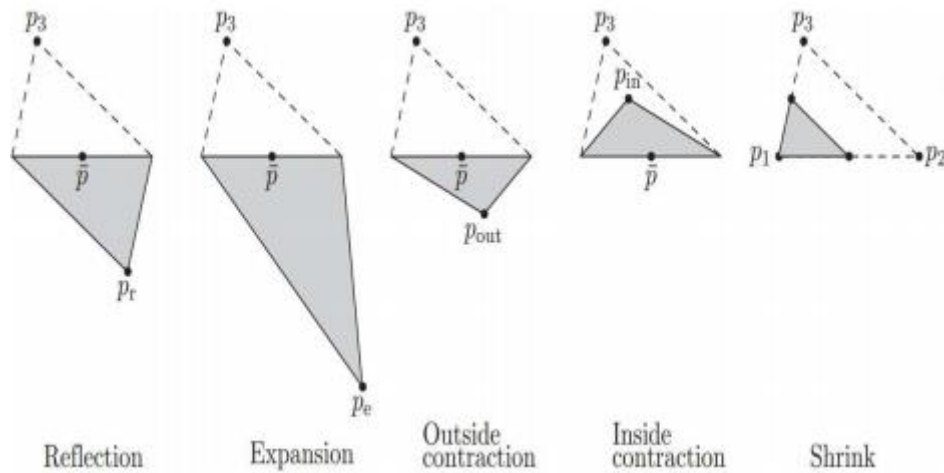
Με το πέρασμα των χρόνων και τη συνεχή διεθνή έρευνα πάνω στο αντικείμενο, έχουν αναπτυχθεί αμέτρητοι αλγόριθμοι και τεχνικές βελτιστοποίησης. Στα πλαίσια της παρούσας εργασίας θα γίνει αναφορά σε συγκεκριμένες μεθοδολογίες βελτιστοποίησης, οι οποίες έπαιξαν σημαντικό ρόλο στην ανάπτυξη του πρωτότυπου κώδικα που θα παρουσιαστεί στα επόμενα κεφάλαια. Οι μεθοδολογίες αυτές είναι :

- Η μέθοδος του κατερχόμενου απλόκου
- Η μέθοδος προσομοιωμένης απόπτωσης
- Οι εξελικτικοί αλγόριθμοι

2.2.1. Η μέθοδος του κατερχόμενου απλόκου

Άπλοκο (simplex) ονομάζεται το γεωμετρικό κυρτό σχήμα, μη μηδενικού όγκου, που ορίζεται από ένα σύνολο $n+1$ σημείων, μη συνεπίεδων, στο n -διάστατο πεδίο αναζήτησης R^n . Το άπλοκο πρακτικά αντιπροσωπεύει το ελάχιστο απαιτούμενο πλήθος σημείων, για να είναι εφικτός ο αριθμητικός υπολογισμός της κλίσης της συνάρτησης. Στο γεωμετρικό, λοιπόν, αυτό σχήμα και στις κινήσεις τις οποίες αυτό μπορεί να εκτελέσει, στηρίζεται ο αλγόριθμος του κατερχόμενου απλόκου. Με το πέρασμα των χρόνων αναπτύσσονται ολοένα και περισσότεροι αλγόριθμοι που κάνουν χρήση του απλόκου, όμως η αρχική ιδέα αποδίδεται στους Spendley, Hext και Himsworth (1962). Στον αρχικό, λοιπόν, αλγόριθμο των Spendley et al. (1962) το άπλοκο διαμορφώνεται από $n + 1$ κορυφές, για κάθε μια από τις οποίες υπολογίζεται η τιμή της αντικειμενικής συνάρτησης. Η κορυφή που έχει τη χειρότερη τιμή (δηλαδή τη μεγαλύτερη για προβλήματα ελαχιστοποίησης) εντοπίζεται και την θέση της παίρνει η αντιδιαμετρική της, ως προς το κεντροειδές των υπολοίπων κορυφών. Αύτη η κίνηση που εκτελεί το άπλοκο ονομάζεται ανάκλαση και οδηγεί στον σχηματισμό ενός νέου απλόκου που έχει καινούργια διεύθυνση και ταυτόχρονα διατηρεί τον όγκο του αρχικού. Αν το καινούργιο σημείο, που προκύπτει από την ανάκλαση, είναι το χειρότερο στο νέο άπλοκο που σχηματίζεται, τότε επιλέγεται για ανάκλαση το σημείο με την αμέσως χειρότερη τιμή, με σκοπό να μην επιτραπεί η επιστροφή στην προηγούμενη κατάσταση και άρα η αναπαραγωγή των ίδιων λύσεων.

Η ιδέα του απλόκου και της εκτέλεσης της κίνησης της ανάκλασης ήταν πολύ πρωτοποριακή, αλλά όχι αρκετή για να παράξει έναν αποδοτικό αλγόριθμο βελτιστοποίησης. Για το λόγο αυτό, έρχονται στο προσκήνιο οι Nelder και Mead, οι οποίοι το 1965 εξέλιξαν τον βασικό αλγόριθμο, εισάγοντας στο άπλοκο τρεις καινούριες κινήσεις, την επέκταση, τη συμπίεση και τη συρρίκνωση. Ο συνδυασμός των τεσσάρων αυτών κινήσεων δημιούργησε έναν νέο αλγόριθμο, ο οποίος βελτίωσε καθοριστικά την επίδοση του αρχικού. Οι κινήσεις, τις οποίες εκτελεί το άπλοκο στον αλγόριθμο των Nelder και Mead, παρουσιάζονται γραφικά ακολούθως:



Σχήμα 2-1: Απεικόνιση των κινήσεων του κατερχόμενου απλόκου στις 2 διαστάσεις (Πηγή: Aloise, D., et. al., 2018. *Less it more: Simplified Nelder-Mead method for large unconstrained optimization.*)

Εν συνεχεία, ακολουθεί μια αναλυτική περιγραφή του αλγορίθμου των Nelder και Mead, ή αλλιώς του αλγορίθμου του κατερχόμενου απλόκου:

Σε μια τυχαία επανάληψη του αλγορίθμου, έστω k , είναι γνωστές όλες οι $n+1$ κορυφές του απλόκου $\{x_1, x_2, \dots, x_{n+1}\}$. Γνωστές είναι, επίσης, και οι αντίστοιχες τιμές της αντικειμενικής συνάρτησης για την κάθε μια κορυφή, $\{f(x_1), f(x_2), \dots, f(x_{n+1})\}$. Αρχικά, τοποθετούνται οι κορυφές σε αύξουσα σειρά, με κριτήριο την τιμή της στοχικής συνάρτησης και βρίσκεται η καλύτερη λύση x_1 , η χειρότερη λύση x_{n+1} και η δεύτερη χειρότερη λύση x_n . Δηλαδή:

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_n) \leq f(x_{n+1}) \quad (2.7)$$

Έπειτα, ακολουθεί ο υπολογισμός του κεντροειδούς των n καλύτερων κορυφών του απλόκου:

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j \quad (2.8)$$

Γίνεται εκτέλεση ενός βήματος ανάκλασης περί της χειρότερης κορυφής x_{n+1} ως προς το κεντροειδές, μέσω της σχέσης:

$$x_r = 2\bar{x} - x_{n+1} \quad (2.9)$$

Υπολογίζεται η αντικειμενική συνάρτηση στο σημείο που προκύπτει από την ανάκλαση $f_r = f(x_r)$.

Η τιμή της f_r καθορίζει τα επόμενα βήματα του αλγορίθμου:

1^η περίπτωση: Αν $f_r < f(x_1)$, δηλαδή αν η λύση που προέκυψε από την ανάκλαση είναι καλύτερη από την αρχική καλύτερη λύση του απλόκου, εκτελείται επέκταση προς την κατεύθυνση βελτίωσης που έχει εντοπιστεί, από την οποία επέκταση προκύπτει το σημείο x_e ως εξής:

$$x_e = 2x_r - \bar{x} \quad (2.10)$$

Στο νέο αυτό σημείο x_e υπολογίζεται η τιμή της αντικειμενικής συνάρτησης $f_e = f(x_e)$. Τώρα προκύπτουν δύο υποπεριπτώσεις, δηλαδή αν το νέο σημείο x_e είναι καλύτερο από το x_r , δηλαδή αν $f_e < f_r$, τότε γίνεται αποδεκτό το x_e και αντικαθιστά την χειρότερη κορυφή x_{n+1} , ενώ διαφορετικά η χειρότερη κορυφή αντικαθίσταται από το σημείο ανάκλασης x_r .

2^η περίπτωση: Αν $f_n < f_r < f_{n+1}$, δηλαδή αν η λύση που προκύπτει από την ανάκλαση είναι χειρότερη από το 2^ο χειρότερο σημείο, αλλά καλύτερη από το χειρότερο, τότε πραγματοποιείται μια διαφορετική κίνηση, η λεγόμενη εξωτερική συμπίεση του απλόκου, από την οποία προκύπτει το νέο σημείο x_c ως εξής:

$$x_c = \bar{x} + 0.5(x_r - \bar{x}) \quad (2.11)$$

Στο σημείο αυτό x_c υπολογίζεται η στοχική συνάρτηση $f_c = f(x_c)$ και εφόσον $f_c < f_r$, τότε το x_c αντικαθιστά την χειρότερη κορυφή x_{n+1} .

3^η περίπτωση: Αν $f_r > f_{n+1}$, δηλαδή αν το σημείο που προκύπτει από την ανάκλαση είναι χειρότερο από την χειρότερη κορυφή του απλόκου, τότε το άπλοκο εκτελεί την λεγόμενη εσωτερική συμπίεση, μέσω της οποίας προκύπτει μια νέα λύση, το x_c ως εξής:

$$x_c = \bar{x} - 0.5(x_r - \bar{x}) \quad (2.12)$$

Στο νέο αυτό σημείο x_c υπολογίζεται η αντικειμενική συνάρτηση $f_c = f(x_c)$ και αν $f_c < f_{n+1}$ τότε το νέο αυτό σημείο x_c αντικαθιστά την κορυφή x_{n+1} .

4^η περίπτωση: Αν ο αλγόριθμος οδηγηθεί είτε στην 2^η περίπτωση είτε στην 3^η περίπτωση και δεν βελτιωθεί η τιμή της στοχικής συνάρτησης, τότε το άπλοκο εκτελεί την λεγόμενη συρρίκνωση προς την κατεύθυνση της καλύτερης κορυφής x_1 . Οι υπόλοιπες, λοιπόν, n κορυφές του απλόκου υφίστανται συρρίκνωση σύμφωνα με τη σχέση:

$$x_s = 0.5(x_j + x_1) \quad (2.13)$$

Ο αλγόριθμος των Nelder και Mead γνώρισε πολύ μεγάλη επιτυχία και αποδοχή από την επιστημονική κοινότητα. Παρόλα αυτά, μελετητές εξέτασαν τη δυνατότητα σύγκλισης του αλγορίθμου σε απλές μαθηματικές συναρτήσεις και σε πολλές περιπτώσεις παρατηρήθηκε η αδυναμία του να εντοπίσει το ολικό ακρότατο της συνάρτησης, καθώς ο αλγόριθμος εγκλωβιζόταν είτε σε κάποιο τοπικό ακρότατο είτε και σε άλλο μη στάσιμο σημείο. Έτσι με το πέρασμα των χρόνων, αρκετοί μελετητές έχουν προσπαθήσει με τις δικές τους τροποποιήσεις να εξελίξουν περαιτέρω τον

αλγόριθμο Nelder-Mead, για να εξαλείψουν στο μέγιστο δυνατό τις αδυναμίες που αυτός παρουσιάζει.

2.2.2. Η μέθοδος της προσομοιωμένης ανόπτωσης

Η μέθοδος της προσομοιωμένης ανόπτωσης είναι μια δημοφιλής μέθοδος ολικής βελτιστοποίησης, η οποία προσπαθεί με πιθανοτικούς κανόνες να εντοπίσει το ολικό ακρότατο της στοχικής συνάρτησης. Το όνομα αλλά και η λογική της μεθόδου βρίσκουν ρίζα στην έννοια της, γνωστής από τη φυσική, διαδικασίας ανόπτωσης, η οποία είναι μια τεχνική ψύξης ενός υλικού (για παράδειγμα μετάλλου), κάτω από ελεγχόμενες συνθήκες, κάτι το οποίο στοχεύει στην εύρεση της ολικά ελάχιστης ενέργειας του συστήματος. Ενώ αρχικά η κινητικότητα των μορίων του υλικού είναι μεγάλη, λόγω της ανόπτωσης αυτή η κινητικότητα σιγά σιγά περιορίζεται και ξεκινάει η δημιουργία κρυστάλλων. Στόχος της όλης προσπάθειας είναι να επιτευχθεί ελάχιστη ενέργεια στο υλικό, δηλαδή να επέλθει το σύστημα σε κατάσταση τέλει κρυσταλλοποίησης, όπως λέγεται. Για να γίνει αυτό, απαραίτητος είναι ο αργός ρυθμός ψύξης. Και αυτό γιατί διαφορετικά, το υλικό μεταβαίνει σε ενδιάμεσες καταστάσεις, υψηλότερης ενέργειας, με τα μόριά του να αποκτούν πολυκρυσταλλική ή άμορφη δομή, κάτι το οποίο δεν είναι επιθυμητό.

Αυτή τη διαδικασία προσπάθησαν οι Metropolis et al. (1953) να προσομοιώσουν για να βρουν την ολικά ελάχιστη ενέργεια του συστήματος. Σύμφωνα με τη θεωρία τους, ένα θερμοδυναμικό σύστημα με δεδομένη θερμοκρασία T , θα έχει πιθανότητα p να μεταβεί από μία ενεργειακή κατάσταση E_1 σε μια κατάσταση E_2 . Η πιθανότητα αυτή p θα δίνεται από τη σχέση:

$$p = e^{\left(\frac{-E_2 - E_1}{kT}\right)} \quad (2.14)$$

όπου k είναι η λεγόμενη σταθερά Boltzmann. Με λίγη ανάλυση πάνω στην εξίσωση (2.14) διαπιστώνεται ότι αν $E_2 < E_1$ προκύπτει $p > 1$ και στην περίπτωση αυτή τίθεται $p = 1$, ενώ αν $E_2 > E_1$ προκύπτει $0 < p < 1$. Έτσι, γίνεται αντιληπτή η πρωτοπορία της μεθόδου, ότι δηλαδή να μην θα γίνονται αποδεκτές με πιθανότητα $p = 1$ (βέβαιο ενδεχόμενο) οι μεταβάσεις σε καταστάσεις χαμηλότερης ενέργειας, μιας και αυτός είναι ο στόχος, αλλά θα γίνονται επίσης αποδεκτές και οι μεταβάσεις σε καταστάσεις υψηλότερης ενέργειας με κάποια πιθανότητα p . Αυτή η ιδέα έχει απόλυτη συνάφεια με την προσπάθεια απεγκλωβισμού από πιθανά τοπικά ακρότατα και την εύρεση της πραγματικά ολικά ελάχιστης ενέργειας του συστήματος.

Με γνώμονα τα παραπάνω, οι Kirkpatrick et al. (1983) ανέπτυξαν τη μέθοδο της προσομοιωμένης ανόπτωσης, αξιοποιώντας την αντιστοιχία που μπορεί να επιτευχθεί ανάμεσα στην εύρεση της ολικά ελάχιστης ενέργειας ενός θερμοδυναμικού συστήματος και στην εύρεση του ολικού ελαχίστου μιας στοχικής συνάρτησης. Στα πλαίσια της αντιστοιχίας αυτής, η ενέργεια θεωρείται το αντίστοιχο της στοχικής συνάρτησης και η θερμοκρασία του θερμοδυναμικού συστήματος θεωρείται το αντίστοιχο κάποιας παραμέτρου που θα ελέγχει τη διαδικασία της βελτιστοποίησης.

Η μέθοδος της προσομοιωμένης απόπτωσης θεωρήθηκε απόλυτα πρωτοποριακή, καθώς, στα πλαίσια αυτής, εισήχθη η ιδέα του να γίνονται δεκτές, εκτός από λύσεις που βελτιώνουν την τιμή της αντικειμενικής συνάρτησης, και ορισμένες λύσεις οι οποίες δεν βελτιώνουν την τιμή της, με κάποια συγκεκριμένη πιθανότητα $p(T)$. Αρχικά η μέθοδος εφαρμόστηκε σε προβλήματα, στα οποία ο χώρος αναζήτησης είναι διακριτός, παρόλα αυτά έχουν γίνει κατά καιρούς τροποποιήσεις για τη γενίκευση της μεθόδου και την εφαρμογή της σε προβλήματα συνεχών μεταβλητών. Μια επισκόπηση του τρόπου με τον οποίο λειτουργεί η μέθοδος είναι η παρακάτω:

Έστω μια στοχαστική συνάρτηση ενός προβλήματος βελτιστοποίησης $f : S \rightarrow R$, με S ένα σύνολο διακριτών λύσεων και έστω μια αρχική λύση του προβλήματος x . Στη γειτονιά αυτής της λύσης x εφαρμόζεται κάποια διαφορετική μέθοδος βελτιστοποίησης (πχ. ο αλγόριθμος του κατερχόμενου απλόκου) και προκύπτει μια νέα λύση y . Στο νέο αυτό σημείο y υπολογίζεται η τιμή της αντικειμενικής συνάρτησης $f(y)$. Αν $f(y) < f(x)$ τότε η λύση y γίνεται σίγουρα αποδεκτή. Αν $f(y) > f(x)$, δηλαδή η λύση y δεν βελτιώνει την τιμή της στοχαστικής συνάρτησης, η λύση y γίνεται αποδεκτή, εφόσον ικανοποιείται κάποιο πιθανοτικό κριτήριο $p(x, y, T)$. Ως πιθανοτικό κριτήριο συνηθίζεται να λαμβάνεται το πιθανοτικό κριτήριο των Metropolis et al. (1983), σύμφωνα με το οποίο για $T > 0$, το $p(x, y, T)$ δίνεται από την ακόλουθη σχέση:

$$p(x, y, T) = \begin{cases} 1, & f(y) < f(x) \\ e^{-\frac{f(y)-f(x)}{T}}, & f(y) > f(x) \end{cases} \quad (2.15)$$

Η απόλυτη συνάφεια της σχέσης (2.14) και της σχέσης (2.15) γίνεται εύκολα αντιληπτή. Με το πέρασμα των χρόνων και καθώς επόμενοι ερευνητές έρχονταν σε επαφή με τη μέθοδο, παρουσιάστηκαν και διαφορετικά πιθανοτικά κριτήρια.

Όπως γίνεται κατανοητό από την εξίσωση (2.15), ένας κρίσιμος παράγοντας που καθορίζει την αποδοχή μιας λύσης, είναι η παράμετρος T . Η παράμετρος T είναι επιθυμητό στα αρχικά στάδια του αλγορίθμου να έχει υψηλές τιμές, με στόχο η πιθανότητα αποδοχής μιας λύσης στο σύστημα να είναι σχετικά υψηλή και έτσι να μπορεί να γίνει μια ευρεία διερεύνηση του χώρου των εφικτών λύσεων και να αποφεύγεται ο εγκλωβισμός σε κάποιο τοπικό ακρότατο. Αυτό όμως έχει και ένα αρνητικό αντίκτυπο, το γεγονός ότι η ταχύτητα σύγκλισης του αλγορίθμου θα είναι αρκετά αργή. Για αυτό και σε επόμενα στάδια, είναι εύλογο η παράμετρος T να μειώνεται, αποδέχοντας στο σύστημα, κυρίως, λύσεις που βελτιώνουν την τιμή της στοχαστικής συνάρτησης, κάνοντας έτσι τον κώδικα περισσότερο ντετερμινιστικό και επιταχύνοντας τη σύγκλιση του.

Γίνεται, επομένως, κατανοητό ότι, αφενός η αρχική θερμοκρασία T του συστήματος και αφετέρου ο ρυθμός μείωσης αυτής, είναι δύο βασικοί παράγοντες που μπορούν να καθορίσουν την επιτυχία ή αποτυχία του αλγορίθμου. Η πιο συνήθης συνάρτηση που εφαρμόζεται για τη μείωση της θερμοκρασίας T του συστήματος είναι η ακόλουθη:

$$T^{[k+1]} = \lambda \times T^{[k]} \quad (2.16)$$

με τον συντελεστή λ να παίρνει τιμές στο εύρος $(0, 1)$. Οι τιμές του λ συνηθίζεται να κυμαίνονται στο εύρος $[0.80, 0.99]$. Βέβαια, έχουν αναπτυχθεί και πολλές ακόμα

θεωρίες – εξισώσεις για τη συνάρτηση μείωσης της θερμοκρασίας από διάφορους ερευνητές.

2.2.3. Εξελικτικοί αλγόριθμοι

Οι εξελικτικοί αλγόριθμοι (evolutionary algorithms) είναι υπολογιστικές μέθοδοι που, για να επιλύσουν προβλήματα βελτιστοποίησης, στηρίζονται στο πρότυπο των φυσικών εξελικτικών διεργασιών και είναι εμπνευσμένοι από τις αρχές της βιολογικής εξέλιξης (θεωρία της εξέλιξης C. Darwin, 1858). Σύμφωνα λοιπόν με τη θεωρία της εξέλιξης του Δαρβίνου, η εξέλιξη των ζωντανών οργανισμών είναι μια φυσική διαδικασία που οδηγεί στη βελτιστοποίηση του είδους, σύμφωνα με την ικανότητα επιβίωσής τους στο περιβάλλον του προβλήματος. Σε αντιστοιχία λοιπόν με αυτή τη θεωρία, οι εξελικτικοί αλγόριθμοι στηρίζονται στην προσομοίωση της εξέλιξης ενός πληθυσμού (population) εφικτών σημείων, P, μέσω υπολογιστικών διαδικασιών που προέρχονται από την φυσική διαλογή, την αναπαραγωγή και την μετάλλαξη. Η έννοια της φυσικής διαλογής αφορά στο σκεπτικό ότι μεγαλύτερη πιθανότητα επιβίωσης έχουν τα ισχυρότερα μέλη ενός είδους, ενώ η έννοια της αναπαραγωγής αφορά στην ιδιότητα των παιδιών να κληρονομούν τα γενετικά χαρακτηριστικά των γονέων τους. Τέλος, η έννοια της μετάλλαξης αφορά στο γεγονός ότι τυχαίες αλλαγές στο γονιδιακό υλικό μπορούν να επιφέρουν διαφοροποίηση των ειδών.

2.2.3.1. Γενετικοί αλγόριθμοι

Η δημοφιλέστερη και σπουδαιότερη υποκατηγορία των εξελικτικών αλγορίθμων είναι οι γενετικοί αλγόριθμοι (genetic algorithms). Στους γενετικούς αλγορίθμους, κάθε μια λύση του προβλήματος είναι ένα μέλος του πληθυσμού και αποκαλείται άτομο (individual). Κάθε άτομο απαρτίζεται από συγκεκριμένα γενετικά χαρακτηριστικά, τα οποία αποκαλούνται χρωμοσώματα (chromosomes). Τα χρωμοσώματα αφορούν τη θέση της λύσης στο χώρο αναζήτησης και αποτελούνται από τα λεγόμενα γονίδια (genes). Για να μπορέσει να βρεθεί τελικά μια λύση στο πρόβλημα βελτιστοποίησης, ο πληθυσμός εξελίσσεται σε στάδια και ο πληθυσμός των λύσεων σε κάποιο στάδιο ονομάζεται γενιά (generation) λύσεων. Η παραγωγή του αρχικού πληθυσμού (της 1^{ης} γενιάς δηλαδή) πραγματοποιείται τυχαία, ενώ στη συνέχεια για να εξελίσσεται ο πληθυσμός από γενιά σε γενιά χρησιμοποιούνται οι λεγόμενοι γενετικοί τελεστές (genetic operators), που είναι η επιλογή (selection), η διασταύρωση (crossover) και η μετάλλαξη (mutation). Κάθε νέα λύση του προβλήματος που παράγεται μέσω των γενετικών τελεστών ονομάζεται απόγονος (offspring) και οι λύσεις, οι οποίες συμμετείχαν στην δημιουργία του συγκεκριμένου απογόνου, είναι οι γονείς (parents) του.

Στα πλαίσια λειτουργίας ενός τυπικού γενετικού αλγορίθμου, ο πληθυσμός της κάθε γενιάς υφίσταται μια προσομοιωμένη γενετική εξέλιξη, μέσω της οποίας στον πληθυσμό της επόμενης γενιάς θα παραμείνουν κυρίως οι πιο ισχυρές λύσεις και οι λιγότερο ισχυρές θα απομακρυνθούν (δεν θα επιβιώσουν). Για να γίνει εφικτός ο

διαχωρισμός των λύσεων σε ισχυρές και αδύναμες, χρησιμοποιείται μια συνάρτηση καταλληλότητας (fitness function), η οποία πρακτικά είναι η στοχική συνάρτηση του προβλήματος βελτιστοποίησης. Για να δημιουργηθεί ένας πλήρως αναπτυγμένος και λειτουργικός γενετικός αλγόριθμος, απαιτείται να εμπεριέχει στη δομή του τα ακόλουθα χαρακτηριστικά:

- κατάλληλο τρόπο για τη γενετική αναπαράσταση των λύσεων του προβλήματος
- κατάλληλο τρόπο για τη δημιουργία του αρχικού πληθυσμού λύσεων, δηλαδή της 1^{ης} γενιάς
- εύρεση συνάρτησης καταλληλότητας για την αξιολόγηση των ατόμων του πληθυσμού
- αξιοποίηση των γενετικών τελεστών για την δημιουργία νέων λύσεων του προβλήματος
- καθορισμό των κρίσιμων παραμέτρων του αλγόριθμου, όπως είναι για παράδειγμα το μέγεθος του πληθυσμού και η πιθανότητα πραγματοποίησης του κάθε γενετικού τελεστή

Γενετική αναπαράσταση χρωμοσώματος: Το χρωμόσωμα ενός ατόμου περιέχει την πληροφορία για την λύση που αναπαριστά. Για αυτό η κωδικοποίηση (encoding) του χρωμοσώματος είναι το πρώτο βήμα κατά την κατασκευή ενός γενετικού αλγορίθμου. Υπάρχουν αρκετοί τρόποι με τους οποίους μπορεί να γίνει η κωδικοποίηση και ο τρόπος κωδικοποίησης που επιλέγεται κάθε φορά εξαρτάται από τον τύπο του εκάστοτε προβλήματος. Ωστόσο, οι συνηθέστεροι τρόποι κωδικοποίησης είναι η δυαδική και η δεκαδική κωδικοποίηση. Γραφικά, οι δύο αυτοί τρόποι κωδικοποίησης ακολουθούν την εξής μορφή:

Chromosome A	10110010110011100101
Chromosome B	11111110000000011111

Σχήμα 2-2: Δυαδική κωδικοποίηση χρωμοσώματος (Πηγή: www.geeksforgeeks.org)

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

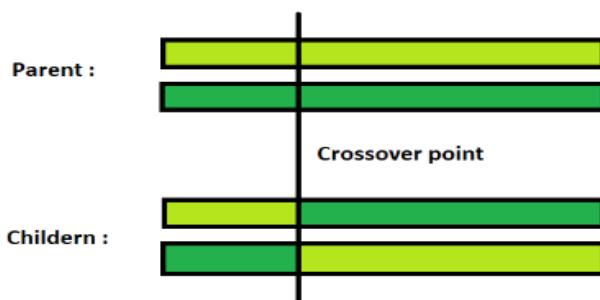
Σχήμα 2-3: Δεκαδική κωδικοποίηση χρωμοσώματος (Πηγή: www.geeksforgeeks.org)

Αρχικός πληθυσμός λύσεων: Πολλή μεγάλη σημασία για την επιτυχή λειτουργία του γενετικού αλγορίθμου διακατέχει η δημιουργία ενός ικανοποιητικού αρχικού πληθυσμού λύσεων. Η δημιουργία του γίνεται τυχαία, όμως με το πέρασμα των χρόνων έχουν αναπτυχθεί αρκετές μεθοδολογίες, έτσι ώστε οι αρχικές τυχαίες λύσεις του προβλήματος να είναι διασκορπισμένες αποδοτικά σε μεγάλο τμήμα του συνολικού χώρου αναζήτησης, ώστε να διευκολυνθεί και να επιταχυνθεί σημαντικά η σύγκλιση του αλγορίθμου.

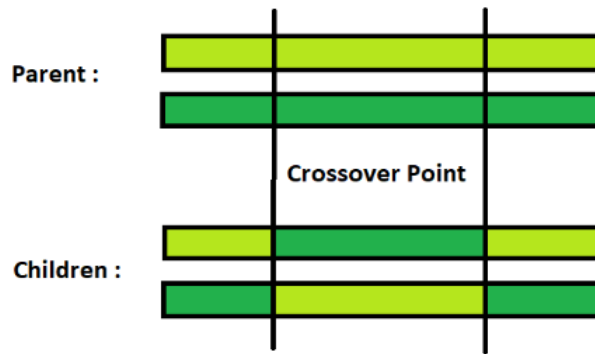
Συνάρτηση καταλληλότητας: Όπως αναφέρθηκε παραπάνω, η συνάρτηση καταλληλότητας είναι ο τρόπος αξιολόγησης των λύσεων του προβλήματος για να γίνει αποδοτικά ο διαχωρισμός αυτών σε ισχυρές και αδύναμες. Πρακτικά η συνάρτηση καταλληλότητας είναι η αντικειμενική συνάρτηση του προβλήματος βελτιστοποίησης.

Γενετικοί τελεστές: Οι 3 βασικοί γενετικοί τελεστές είναι η επιλογή (selection), η διασταύρωση (crossover) και η μετάλλαξη (mutation).

- **Επιλογή:** Η επιλογή αφορά στη μέθοδο που εφαρμόζεται για να επιλεγθούν τα χρωμοσώματα που θα χρησιμοποιηθούν για αναπαραγωγή. Η πιο δημοφιλής μέθοδος επιλογής είναι η μέθοδος του τροχού της ρουλέτας (roulette wheel selection), κατά την οποία αυτή μέθοδο δημιουργείται ένας εικονικός τροχός, με τόσες εγκοπές όσο και το μέγεθος του πληθυσμού, με το πλάτος της κάθε «εγκοπής» να είναι ανάλογο του βαθμού καταλληλότητας κάθε ατόμου για αναπαραγωγή. Με τον τρόπο αυτό, ακόμα και το πιο αδύναμο άτομο στον πληθυσμό θα έχει μη μηδενική πιθανότητα να επιλεγεί.
- **Διασταύρωση:** Ο γενετικός τελεστής της διασταύρωσης είναι ο ένας από τους δύο γενετικούς τελεστές με τους οποίους δημιουργούνται οι νέες λύσεις του προβλήματος (ο άλλος τελεστής είναι η μετάλλαξη). Στη διασταύρωση επιλέγονται ανά δύο τα χρωμοσώματα - γονείς και μετά από συνδυασμό των γονιδίων τους προκύπτουν δύο καινούργια χρωμοσώματα, οι λεγόμενοι απόγονοι, με τελείως διαφορετικό γενετικό υλικό από τους γονείς τους. Οι τεχνικές με τις οποίες μπορεί να επιτευχθεί διασταύρωση ποικίλουν. Όμως, δύο από τις πιο συνηθισμένες τεχνικές είναι η διασταύρωση ενός σημείου (single point crossover) και η διασταύρωση δύο σημείων (two point crossover). Κατά τη διασταύρωση ενός σημείου, τα δύο χρωμοσώματα - γονείς χωρίζονται σε ένα τυχαία καθορισμένο σημείο. Στη συνέχεια, δημιουργούνται δύο καινούργια χρωμοσώματα, προσαρτώντας το πρώτο μέρος του πρώτου γονέα με το δεύτερο μέρος του δεύτερου γονέα και αντίστοιχα το πρώτο μέρος του δεύτερου γονέα με το δεύτερο μέρος του πρώτου γονέα. Με αντίστοιχο τρόπο ορίζεται και η διασταύρωση δύο σημείων.



Σχήμα 2-4: Διασταύρωση ενός σημείου (Πηγή: www.geeksforgeeks.org)



Σχήμα 2-5: Διασταύρωση δύο σημείων (Πηγή: www.geeksforgeeks.org)

- **Μετάλλαξη:** Η μετάλλαξη είναι μια διαδικασία, η οποία είναι απαραίτητη για να διατηρείται η γενετική ποικιλομορφία από γενιά σε γενιά. Η μετάλλαξη αλλάζει μία ή περισσότερες γονιδιακές τιμές ενός χρωμοσώματος. Μετά τη διαδικασία της μετάλλαξης, το χρωμόσωμα που προκύπτει μπορεί να είναι εντελώς διαφορετικό από το αρχικό. Το κάθε γονίδιο ενός χρωμοσώματος επιλέγεται για μετάλλαξη ανάλογα με την πιθανότητα μετάλλαξης. Η πιθανότητα αυτή είναι μια παράμετρος που καθορίζεται από το χρήστη και πρέπει να παίρνει γενικά χαμηλές τιμές, καθώς υψηλές τιμές μπορεί να μετατρέψουν τον γενετικό αλγόριθμο σε έναν αλγόριθμο τυχαίας αναζήτησης. Ο γενετικός τελεστής της μετάλλαξης χρησιμοποιείται πρακτικά για να αποφευχθεί ο εγκλωβισμός του αλγορίθμου σε τοπικά ακρότατα και για να υπάρχει ποικιλομορφία στον πληθυσμό της κάθε γενιάς.



Σχήμα 2-6: Γενετικός τελεστής μετάλλαξης (Πηγή: www.geeksforgeeks.org)

Κρίσιμες παράμετροι: Ο αρχικός πληθυσμός και η πιθανότητα μετάλλαξης είναι δύο βασικές παράμετροι που μπορούν να καθορίσουν πλήρως την επιτυχία ή αποτυχία του γενετικού αλγορίθμου. Άλλες κρίσιμες παράμετροι είναι αυτές που καθορίζουν τα κριτήρια τερματισμού του αλγορίθμου. Στο σύνηθες σχήμα ενός γενετικού αλγορίθμου εφαρμόζονται δύο κριτήρια τερματισμού. Το πρώτο κριτήριο τερματίζει τον κώδικα μόλις συμπληρωθεί ένας προκαθορισμένος συνολικός αριθμός επαναλήψεων. Το δεύτερο τον τερματίζει, όταν δεν παρατηρείται βελτίωση της στοχικής συνάρτησης για έναν συγκεκριμένο διαδοχικό αριθμό γενιών λύσεων.

Κεφάλαιο 3^ο

3 Ο εξελικτικός αλγόριθμος ανόπτησης - απλόκου

3.1. Θεμελιώδεις αρχές του αλγορίθμου

Ο εξελικτικός αλγόριθμος ανόπτησης – απλόκου (evolutionary annealing simplex – EAS) αναπτύχθηκε από τον Α. Ευστρατιάδη (2001, 2008) και αποτελεί μια ευρετική πιθανοτική μέθοδο ολικής βελτιστοποίησης, στην οποία επιχειρείται η σύζευξη ενός φάσματος διαφορετικών μεθοδολογιών, με την ταυτόχρονη προσθήκη νέων πρωτότυπων στοιχείων. Ο EAS προσπαθεί να συνδυάσει τη μέθοδο του κατερχόμενου απλόκου των Nelder και Mead με την μέθοδο προσομοιωμένης ανόπτησης, ενσωματώνοντας, ταυτόχρονα, θεμελιώδη στοιχεία και έννοιες από τους εξελικτικούς αλγορίθμους. Η βασική ιδέα του αλγορίθμου είναι να συνδυάσει την ικανότητα της προσομοιωμένης ανόπτησης να απεγκλωβίζεται από τοπικά ακρότατα, με την ικανότητα του απλόκου να εντοπίζει μια περιοχή έλξης γρήγορα και με ακρίβεια. Παράλληλα, ο αλγόριθμος χρησιμοποιεί βασικές αρχές των εξελικτικών αλγορίθμων, όπως είναι η έννοια του πληθυσμού, των γενιών λύσεων και του γενετικού τελεστή της μετάλλαξης.

Οι βασικές αρχές του αλγορίθμου είναι οι εξής:

- Παράγεται ένας αρχικός πληθυσμός λύσεων με τυχαίο τρόπο, ο οποίος σταδιακά από γενιά σε γενιά βελτιώνεται προσεγγίζοντας το ολικό ακρότατο. Επιτρέπεται να οριστούν δύο διαφορετικά πεδία τιμών, αναφορικά με τις μεταβλητές ελέγχου του προβλήματος, ένα εσωτερικό και ένα εξωτερικό. Το εσωτερικό πεδίο καθορίζει τα όρια εντός των οποίων γίνεται η γέννηση του αρχικού πληθυσμού, ενώ το εξωτερικό πεδίο καθορίζει τα όρια εντός των οποίων γίνεται η παραγωγή των λύσεων. Συνήθως το εσωτερικό πεδίο και το εξωτερικό ταυτίζονται, αλλά αν ο χρήστης θέλει να παράξει τον αρχικό πληθυσμό λύσεων σε κάποιο συγκεκριμένο υποπεδίο του συνολικού πεδίου (του εξωτερικού πεδίου δηλαδή), αυτό είναι εφικτό.
- Η παραγωγή των νέων λύσεων για την εξέλιξη του πληθυσμού γίνονται μέσω του απλόκου. Το άπλοκο πραγματοποιεί τις γνωστές κινήσεις της μεθόδου Nelder-Mead, δηλαδή ανάκλαση, επέκταση, συρρίκνωση και εξωτερική/εσωτερική συμπίεση, αλλά στον EAS γίνεται και η προσθήκη τυχαιότητας στο βήμα της μετακίνησης, κάνοντας έτσι την διαδικασία εξέλιξης του πληθυσμού των λύσεων περισσότερο στοχαστική.

- Η τυχειότητα κατά τη διαδικασία αναζήτησης νέων λύσεων προσαρμόζεται μέσω ενός αυτορρυθμιζόμενου χρονοδιαγράμματος ανόπτησης, το οποίο ανάλογα την περίπτωση, μπορεί να επιτρέπει και την αποδοχή λύσεων, οι οποίες χειροτερεύουν την τιμή της στοχικής συνάρτησης, προσπαθώντας στις περιπτώσεις αυτές να απεγκλωβίσει τον αλγόριθμο από τυχόν τοπικά ακρότατα. Κατά την εκκίνηση του αλγόριθμου, η θερμοκρασία του συστήματος τίθεται ίση με τη διαφορά της ελάχιστης τιμής της αντικειμενικής συνάρτησης στον αρχικό πληθυσμό από τη μέγιστη. Όσο, όμως προχωράει η εξέλιξη του πληθυσμού, το σύστημα ξεκινάει μια διαδικασία ψύξης, δηλαδή η θερμοκρασία του συστήματος, και άρα η τυχειότητα του αρχίζει να μειώνεται. Ο χρήστης του αλγορίθμου, μάλιστα, για την καλύτερη εκτέλεση του σε κάθε εξειδικευμένη περίπτωση, πρέπει να εισάγει δύο παραμέτρους β και λ , οι οποίες αφορούν στην ρύθμιση της θερμοκρασίας. Η παράμετρος β καθορίζει το πότε θα ενεργοποιείται η λειτουργία ψύξης του συστήματος, ενώ η παράμετρος λ καθορίζει το πόσο θα μειώνεται η θερμοκρασία, κάθε φορά που εντοπίζεται από τον αλγόριθμο μια περιοχή έλξης.
- Για την αξιολόγηση των κορυφών του απλόκου που σχηματίζεται σε κάθε επανάληψη του αλγορίθμου και την εύρεση της χειρότερης κορυφής του, δεν συγκρίνονται μόνο οι τιμές της στοχικής συνάρτησης της κάθε κορυφής, αλλά σε αυτές προστίθεται και ένας όρος τυχειότητας, που είναι ανάλογος της τρέχουσας θερμοκρασίας του συστήματος. Βέβαια, η καλύτερη κορυφή του εκάστοτε απλόκου δεν συμμετέχει σε αυτή την αξιολόγηση, για να μην υπάρχει ενδεχόμενο να αντικατασταθεί.
- Αν σε κάποια επανάληψη του αλγορίθμου, οι κινήσεις που εκτελεί το άπλοκο δεν καταφέρουν να βελτιώσουν τις λύσεις, τότε ενεργοποιείται ένας μηχανισμός μετάλλαξης, ο οποίος συμβάλλει στην ύπαρξη ποικιλομορφίας στις λύσεις του πληθυσμού.

3.2. Υπολογιστική διαδικασία

Έστω ένα πρόβλημα βελτιστοποίησης, το οποίο περιγράφεται από n συνεχείς μεταβλητές ελέγχου. Όπως είναι στημένος ο αλγόριθμος, το πρόβλημα αντιμετωπίζεται σαν πρόβλημα ελαχιστοποίησης της στοχικής συνάρτησης που το περιγράφει. Η στοχική συνάρτηση είναι βαθμωτή. Για να ξεκινήσει ο αλγόριθμος, ο χρήστης απαιτείται να εισάγει τα ακόλουθα στοιχεία, σαν παραμέτρους εισόδου:

- το πλήθος των μεταβλητών ελέγχου του προβλήματος n
- το σταθερό μέγεθος του πληθυσμού m , με $m \geq n+1$
- δύο πεδία τιμών $[x_{\min}, x_{\max}]$, $[x_{\text{low}}, x_{\text{up}}]$, που αντιστοιχούν στα όρια τιμών του εσωτερικού και του εξωτερικού πεδίου
- την αντικειμενική συνάρτηση που περιγράφει το πρόβλημα

- το μέγιστο επιτρεπτό πλήθος υπολογισμών της αντικειμενικής συνάρτησης
- μια τιμή για το όριο σύγκλισης, το οποίο αν επιτευχθεί, ο αλγόριθμος οδηγείται σε πρόωρο τερματισμό
- τις παραμέτρους του αυτορρυθμιζόμενου χρονοδιαγράμματος ανόπτωσης β και λ
- την πιθανότητα πραγματοποίησης μετάλλαξης p_m
- το μέγιστο επιτρεπτό πλήθος διαδοχικών αναρριχήσεων

Κατά την εκκίνηση του αλγορίθμου δημιουργείται ο αρχικός πληθυσμός ($1^{\text{η}}$ γενιά λύσεων) P_0 . Ο πληθυσμός αυτός αποτελείται από m τυχαίες λύσεις ομοιόμορφα καταναμημένες στο διάστημα $[x_{\min}, x_{\max}]$. Για κάθε μια λύση υπολογίζεται η τιμή της στοχικής συνάρτησης. Έπειτα, προσδιορίζεται η αρχική θερμοκρασία του συστήματος T_0 , η οποία θα είναι ίση με τη διαφορά της ελάχιστης από τη μέγιστη τιμή της αντικειμενικής συνάρτησης, δηλαδή:

$$T_0 = \max\{f_1, f_2, \dots, f_m\} - \min\{f_1, f_2, \dots, f_m\} \quad (3.1)$$

Στο σημείο αυτό, γίνεται μια αναλυτική περιγραφή των βημάτων που εκτελεί ο EAS σε μια τυπική επανάληψη του:

Βήμα 1^ο: Υπολογίζονται η μέση τιμή μ_x και η τυπική απόκλιση σ_x των συντεταγμένων του πληθυσμού.

Βήμα 2^ο: Από τον συνολικό πληθυσμό, επιλέγονται τυχαία $n+1$ σημεία, τα οποία σχηματίζουν το απλόκο $S = \{s_1, s_2, \dots, s_{n+1}\}$. Οι κορυφές του απλόκου τοποθετούνται σε αύξουσα σειρά με την κορυφή s_1 να αντιστοιχεί στο σημείο που έχει τη χαμηλότερη (άρα καλύτερη) τιμή στοχικής συνάρτησης, ενώ το s_{n+1} να έχει τη μεγαλύτερη (άρα χειρότερη).

Βήμα 3^ο: Από το υποσύνολο του απλόκου $\{s_2, \dots, s_{n+1}\}$, το οποίο θα περιλαμβάνει όλες τις κορυφές του απλόκου εκτός της καλύτερης s_1 , βρίσκεται η κορυφή w που θα οδηγηθεί προς αντικατάσταση. Η κορυφή αυτή βρίσκεται σύμφωνα με το κριτήριο Metropolis και θα είναι η κορυφή εκείνη που μεγιστοποιεί τη συνάρτηση:

$$g(s) = f(s) + uT \quad (3.2)$$

όπου u ένας τυχαίος αριθμός που ακολουθεί την ομοιόμορφη κατανομή στο $[0, 1]$ και T η τρέχουσα θερμοκρασία του συστήματος. Στις πρώτες επαναλήψεις του αλγορίθμου η θερμοκρασία του συστήματος είναι αρκετά υψηλή και η επίδραση της τυχαιότητας uT καθορίζει την κορυφή που θα οδηγηθεί προς αντικατάσταση σε μεγάλο βαθμό. Αντίθετα, στις επόμενες επαναλήψεις, που η διαδικασία σύγκλισης προχωρά, το σύστημα ψύχεται και έτσι η θερμοκρασία του συστήματος μειώνεται. Αυτό οδηγεί σε μείωση της επίδρασης του όρου τυχαιότητας uT και έτσι ο αλγόριθμος γίνεται περισσότερο ντετερμινιστικός, ακολουθώντας τη λογική του αλγορίθμου Nelder – Mead.

Βήμα 4^ο : Το άπλοκο εκτελεί ανάκλαση ως προς την κορυφή w και έτσι προκύπτει μια νέα λύση r_0 , η οποία δίνεται από τη σχέση:

$$r_0 = g + (0.5 + u)(g - w) \quad (3.3)$$

όπου u είναι ένας τυχαίος αριθμός που ακολουθεί ομοιόμορφη κατανομή στο $[0, 1]$ και g είναι το κεντροειδές του απλόκου χωρίς την χειρότερη κορυφή w , το οποίο δίνεται από τη σχέση:

$$g = \frac{1}{n} (\sum_{i=1}^{n+1} s_i - w) \quad (3.4)$$

Βήμα 5^ο : Αν το σημείο που προέκυψε από την ανάκλαση έχει χαμηλότερη τιμή αντικειμενικής συνάρτησης και από την χειρότερη κορυφή w , αλλά και από την καλύτερη κορυφή του απλόκου s_1 , δηλαδή αν $f(r_0) < f(w)$ και $f(r_0) < f(s_1)$, τότε εκτελούνται διαδοχικά βήματα επέκτασης κατά την κατεύθυνση της ανάκλασης $r_0 - g$, σύμφωνα με την αναδρομική σχέση:

$$r_s = g + \eta_s (r_0 - g) \quad (3.5)$$

όπου $\eta_s = \eta_{s-1} + u$ με $\eta_0 = 1$. Η επέκταση συνεχίζεται μέχρι να βρεθεί λύση που δεν βελτιώνει άλλο την τιμή της στοχικής συνάρτησης. Αντίθετα, σε περίπτωση που ναί μεν $f(r_0) < f(w)$ αλλά $f(r_0) > f(s_1)$, το άπλοκο εκτελεί εξωτερική συμπίεση, σύμφωνα με τη σχέση:

$$r_1 = g + (0.25 + 0.5u)(r_0 - g) \quad (3.6)$$

Αν το r_1 προκύψει καλύτερο από το r_0 , δηλαδή αν $f(r_1) < f(r_0)$, τότε το αντικαθιστά.

Βήμα 6^ο : Αντίθετα, αν το σημείο που προκύπτει από την ανάκλαση είναι χειρότερο από την χειρότερη κορυφή του απλόκου w , δηλαδή αν $f(r_0) > f(w)$, τότε ο αλγόριθμος ακολουθεί διαφορετικές πορείες που μπορεί να περιλαμβάνουν είτε συμπίεση του απλόκου, είτε συρρίκνωσή του, είτε διαδοχικά βήματα επέκτασης προς την κατεύθυνση που χειροτερεύει τη στοχική συνάρτηση. Η βασική συνθήκη που καθορίζει το ποια θα είναι η πορεία που θα ακολουθήσει ο αλγόριθμος είναι η ακόλουθη:

$$f(r_0) - uT > f(w) + uT \quad (3.7)$$

Αν η συνθήκη (3.7) προκύψει αληθής, το σημείο της ανάκλασης απορρίπτεται και το άπλοκο εκτελεί εσωτερική συμπίεση, σύμφωνα με τη σχέση:

$$r_1 = g - (0.25 + 0.5u)(g - w) \quad (3.8)$$

Αν το r_1 της σχέσης (3.8) προκύψει καλύτερο από την κορυφή w , τότε την αντικαθιστά. Διαφορετικά, αν δηλαδή $f(r_1) > f(w)$, τότε το άπλοκο εκτελεί συρρίκνωση γύρω από την καλύτερη κορυφή του s_1 , σύμφωνα με τη σχέση:

$$s'_i = 0.5(s_1 + s_i), \text{ για } i = 2, 3, \dots, n + 1 \quad (3.9)$$

και υπολογίζονται τα $f(s'_i)$, με τα s'_i να αντικαθιστούν τα s_i . Έπειτα ο αλγόριθμος προχωράει στο βήμα 9.

Οι δύο τελευταίες κινήσεις οδηγούν σε σύγκλιση στην περιοχή κάποιου τοπικού ακροτάτου, κάτι το οποίο οδηγεί και σε ενεργοποίηση του μηχανισμού μείωσης της θερμοκρασίας του συστήματος.

Βήμα 7^ο : Αν η συνθήκη (3.7) προκύψει ψευδής, τότε το σημείο που προκύπτει από την ανάκλαση γίνεται αποδεκτό και ενεργοποιείται ένας μηχανισμός διαφυγής από τοπικό ακρότατο. Ο μηχανισμός αυτός περιλαμβάνει βήματα αναρρίχησης όπως ονομάζονται, δηλαδή βήματα επέκτασης του απλόκου προς την κατεύθυνση όπου αυξάνεται η τιμή της αντικειμενικής συνάρτησης. Τα βήματα αυτά εκτελούνται σύμφωνα με τη σχέση (3.5) και προκύπτουν νέα σημεία. Αν κάποιο από τα σημεία που προκύψουν από την αναρρίχηση είναι καλύτερο από το r_0 ή αν βρεθεί ένα ζεύγος σημείων για τα οποία να ισχύει ότι $f(x_{s+1}) > f(x_s)$, τότε το r_0 αντικαθίσταται.

Βήμα 8^ο : Αν κατά την εκτέλεση του βήματος 7, δεν αντικατασταθεί το σημείο ανάκλασης r_0 , τότε ενεργοποιείται ο μηχανισμός της μετάλλαξης. Στόχος του μηχανισμού αυτού είναι να δημιουργηθεί ένα νέο σημείο εκτός του εύρους της τυπικής απόκλισης γύρω από τη μέση τιμή του τρέχοντος πληθυσμού, δηλαδή το σημείο αυτό να είναι εκτός του εύρους $[\mu_x - \sigma_x, \mu_x + \sigma_x]$. Οι συντεταγμένες του νέου αυτού σημείου προκύπτουν ως εξής:

$$x_i = \mu_i \pm (\sigma_i + u d_i) \quad (3.10)$$

όπου μ_i και σ_i είναι η μέση τιμή και η τυπική απόκλιση αντίστοιχα του πληθυσμού για την i συντεταγμένη και u ένας όρος τυχαιότητας που ακολουθεί την ομοιόμορφη κατανομή στο $[0, 1]$, ενώ d_i είναι η υπολειπόμενη απόσταση μέχρι τα όρια του εξωτερικού πεδίου τιμών του χώρου αναζήτησης για την i συντεταγμένη. Η σχέση (3.10) μπορεί να εκτελεστεί είτε με πρόσημο (+) είτε με πρόσημο (-). Τα δύο πρόσημα έχουν ίση πιθανότητα να επιλεγθούν. Αν επιλεγεί το πρόσημο (+), τότε η αντίστοιχη συντεταγμένη του σημείου προκύπτει από ομοιόμορφη κατανομή στο εύρος $[\mu_i + \sigma_i, x_{i \max}]$ ενώ για το πρόσημο (-) το αντίστοιχο εύρος είναι το $[x_{i \min}, \mu_i - \sigma_i]$. Αυτός ο μηχανισμός ενεργοποιείται για την επίτευξη ποικιλομορφίας στον πληθυσμό και την αποτελεσματικότερη εξερεύνηση, συνεπώς, του χώρου αναζήτησης. Έτσι, προκύπτει το νέο σημείο x_m , το οποίο εισέρχεται στο εξωτερικό αρχείο και γίνεται αποδεκτό και στον πληθυσμό, αν $f(x_m) < f(r_0)$. Διαφορετικά, ανάλογα με μια πιθανότητα μετάλλαξης, η οποία εισάγεται από το χρήστη, επιλέγεται είτε το x_m είτε το r_0 για να αντικαταστήσει το w στον πληθυσμό.

Βήμα 9^ο : Μόλις ολοκληρώνεται μια επανάληψη του αλγορίθμου, ελέγχεται αν η θερμοκρασία του συστήματος υπερβαίνει την τιμή $\beta(f_{\max} - f_{\min})$, όπου β η παράμετρος εισόδου που εισάγει ο χρήστης, με $\beta \geq 1$ και εφόσον την υπερβαίνει τότε η θερμοκρασία επαναπροσδιορίζεται σύμφωνα με τη σχέση:

$$T = \beta[f(s_{n+1}) - f(s_1)] \quad (3.11)$$

Ο EAS μπορεί να τερματιστεί με δύο διαφορετικούς τρόπους:

- 1) όταν ο αριθμός των υπολογισμών της στοχαστικής συνάρτησης υπερβεί το μέγιστο επιτρεπτό όριο, το οποίο εισάγεται από τον χρήστη

- 2) όταν επιτευχθεί το κριτήριο σύγκλισης του αλγορίθμου, δηλαδή όταν η σύγκλισή του είναι καλύτερη από το όριο σύγκλισης, το οποίο εισάγεται από το χρήστη

Αναλυτικότερες πληροφορίες σχετικά τις θεμελιώδεις αρχές και τις υπολογιστικές διαδικασίες του αλγορίθμου, όπως επίσης και σχετικά με την επίδοση του σε πραγματικά προβλήματα βελτιστοποίησης, παρουσιάζονται στις σχετικές εργασίες του Α. Ευστρατιάδη (2001, 2008):

- <https://www.itia.ntua.gr/el/docinfo/446/>
- <https://www.itia.ntua.gr/el/docinfo/838/>

3.3. Υλοποίηση του εξελικτικού αλγορίθμου ανόπτησης - απλόκου σε γλώσσα προγραμματισμού Python

Στα πλαίσια της παρούσας διπλωματικής εργασίας επιχειρήθηκε η μεταφορά του εξελικτικού αλγορίθμου ανόπτησης – απλόκου σε γλώσσα προγραμματισμού Python. Ο αλγόριθμος παραλήφθηκε, αρχικά, σε προγραμματιστικό περιβάλλον MATLAB. Έπειτα, ακολούθησε η κατανόηση των θεμελιωδών αρχών και υπολογιστικών διαδικασιών του, για να καταστεί εφικτή η επιτυχής μετάφραση του. Ταυτόχρονα, εξετάστηκαν πιθανές αλλαγές που θα μπορούσαν να πραγματοποιηθούν στον αρχικό κώδικα για να βελτιωθεί περαιτέρω η επίδοση του.

Τελικά, κατά τη μεταφορά του EAS σε Python, ο αρχικός αλγόριθμος υπέστη μια διαφοροποίηση:

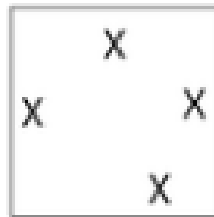
Στον πρωτότυπο κώδικα, ο αρχικός πληθυσμός λύσεων προκύπτει από τυχαίες λύσεις, οι οποίες ακολουθούν την ομοιόμορφη κατανομή στο διάστημα $[x_{\min}, x_{\max}]$. Κατά τη μεταφορά του αλγορίθμου, η παραγωγή του αρχικού πληθυσμού πραγματοποιείται σύμφωνα με τη στατιστική μέθοδο Latin Hypercube Sampling (LHS) στο διάστημα $[x_{\min}, x_{\max}]$, η οποία αναπτύχθηκε από τον McKay (1979). Η μέθοδος LHS χρησιμοποιείται για να παραχθούν «τυχαίες» λύσεις σε ένα πρόβλημα, οι οποίες θα είναι όσο το δυνατόν «καλύτερα» διασκορπισμένες στο πεδίο αναζήτησης, για να γίνει μια αποτελεσματικότερη διερεύνηση του συνολικού χώρου. Η λογική της μεθόδου εξηγείται απλοϊκά μέσω του ακόλουθου παραδείγματος:

Ας υποθέσουμε ότι θέλουμε να παράξουμε τις αρχικές τυχαίες λύσεις του πληθυσμού σε ένα πρόβλημα βελτιστοποίησης δύο διαστάσεων, δηλαδή με δύο μεταβλητές ελέγχου ($n=2$). Ο χώρος αναζήτησης, λοιπόν, θα είναι ένα υποσύνολο του \mathbb{R}^2 . Αν θεωρήσουμε ότι θέλουμε ο πληθυσμός των λύσεων να απαρτίζεται από $m=4$ άτομα, τότε το αποτέλεσμα εφαρμογής της στατιστικής μεθόδου LHS στο πρόβλημα αυτό, απεικονίζεται παρακάτω:

X			
	X		
			X
		X	

Σχήμα 3-1: Παραγωγή τυχαίων λύσεων μέσω μεθόδου LHS (Πηγή: www.wikipedia.org)

Δεδομένου ότι οι τυχαίες λύσεις που θα παραχθούν είναι 4, το συνολικό πεδίο αναζήτησης, μέσω παράλληλων ευθειών στις 2 διαστάσεις, θα διαχωριστεί σε 16 τμήματα ίσης επιφάνειας και οι λύσεις που θα δημιουργηθούν (εμφανίζονται με το σύμβολο «X»), διασκορπίζονται στον συνολικό χώρο, με τον τρόπο που φαίνεται στην εικόνα. Η ιδέα αυτή γενικεύεται ανάλογα τη διάσταση του προβλήματος και το πλήθος των ατόμων που απαρτίζουν τον πληθυσμό. Αντιθέτως, με την παραγωγή των τυχαίων λύσεων μέσω ομοιόμορφης κατανομής στο παράδειγμα αυτό, ένα πιθανό αντιπροσωπευτικό αποτέλεσμα της εφαρμογής απεικονίζεται παρακάτω:



Σχήμα 3-2: Παραγωγή τυχαίων λύσεων μέσω ομοιόμορφης κατανομής (Πηγή: www.wikipedia.org)

Η μέθοδος LHS, επομένως, θεωρήθηκε προτιμότερη από την παραγωγή των τυχαίων λύσεων μέσω ομοιόμορφης κατανομής, καθώς στη δεύτερη περίπτωση η παραγωγή γίνεται τελείως τυχαία και είναι πιθανό το δείγμα των λύσεων να μην κάνει αποδοτική διερεύνηση του συνολικού χώρου.

Στον ακόλουθο σύνδεσμο γίνεται η παρουσίαση υλοποίησης του εξελικτικού αλγορίθμου ανόπτησης – απλόκου (EAS) σε γλώσσα προγραμματισμού Python:

<https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/EAS>

Κεφάλαιο 4^ο

4 Εμπλουτισμός αλγορίθμων βελτιστοποίησης με τεχνικές υποκατάστατων μοντέλων

4.1. Η έννοια των υποκατάστατων μοντέλων και η χρησιμότητά τους στις διαδικασίες βελτιστοποίησης

Όπως αναφέρθηκε στο [Κεφάλαιο 2](#), η βελτιστοποίηση είναι ένας τομέας, ο οποίος ενδιαφέρει σε πολύ μεγάλο βαθμό τον υδραυλικό μηχανικό. Και αυτό, γιατί υπάρχει πληθώρα προβλημάτων που καλείται να επιλύσει αυτός σε καθημερινή βάση, τα οποία προκειμένου να αντιμετωπιστούν αποτελεσματικά, απαιτείται να γίνει χρήση μεθόδων και τεχνικών βελτιστοποίησης. Ειδικά στα προβλήματα βελτιστοποίησης υδατικών πόρων, ένα ζήτημα, το οποίο αποτελεί συχνά τον περιοριστικό παράγοντα για την επιτυχή ολοκλήρωση της διαδικασίας, είναι η αντικειμενική συνάρτηση. Αυτό, γιατί στα προβλήματα αυτά, η στοχική συνάρτηση δεν ορίζεται μέσω ενός απλού μαθηματικού τύπου, αλλά μέσω ενός μοντέλου προσομοίωσης. Με άλλα λόγια, προκειμένου να αποκτηθεί μια μεμονωμένη τιμή της αντικειμενικής συνάρτησης, πρέπει να προηγηθεί η εκτέλεση ενός μοντέλου (π.χ. ενός υδραυλικού μοντέλου), του οποίου η ολοκλήρωση απαιτεί ένα συγκεκριμένο χρονικό διάστημα, που δεν είναι αμελητέο. Είναι επίσης γνωστό, ότι προκειμένου να καταφέρει ένας αλγόριθμος να προσεγγίσει με ικανοποιητική ακρίβεια το ολικό ακρότατο μιας συνάρτησης, μπορεί να χρειαστεί να κάνει κλήση αυτής πολλές χιλιάδες φορές. Ο ακριβής αριθμός των κλήσεων ποικίλει ανάλογα με τη δυσκολία του προβλήματος και την μορφή της επιφάνειας απόκρισης της συνάρτησης.

Με βάση τα παραπάνω, γίνεται αντιληπτό ότι ο συνολικός χρόνος που μπορεί να απαιτηθεί για την επιτυχή ολοκλήρωση ενός προβλήματος βελτιστοποίησης υδατικών πόρων μέσω ενός συμβατικού αλγορίθμου, είναι πιθανό να υπερβαίνει τα λογικά πλαίσια, κάτι το οποίο οδηγεί συχνά τον προγραμματιστή στην ανάγκη να περιορίσει τον συνολικό επιτρεπόμενο αριθμό κλήσεων της αντικειμενικής συνάρτησης, προκειμένου να παραλάβει ένα αποτέλεσμα σε ένα εύλογο χρονικό διάστημα. Ο πρόωρος, όμως, αυτός τερματισμός του αλγορίθμου έχει σαν συνέπεια την μη αποτελεσματική σύγκλισή του.

Το ζήτημα αυτό προβλημάτισε σημαντικά την επιστημονική κοινότητα και αυτή, όπως είναι λογικό, ξεκίνησε να σκέφτεται τρόπους για να το επιλύσει. Ανάμεσα στις αρκετές τακτικές που προτάθηκαν μέσα στα χρόνια, μάλλον, η πιο δημοφιλής και εδραιωμένη είναι η χρήση τεχνικών υποκατάστατων μοντέλων (surrogate models). Τα υποκατάστατα μοντέλα ή αλλιώς μεταμοντέλα (meta-models) προσπαθούν να προσεγγίσουν την επιφάνεια απόκρισης του αρχικού μοντέλου προσομοίωσης (της

αντικειμενικής συνάρτησης δηλαδή) και ταυτόχρονα ο χρόνος εκτέλεσης τους να είναι ασύλληπτα μειωμένος σε σύγκριση με το χρόνο εκτέλεσης του αρχικού μοντέλου. Πρακτικά, το υποκατάστατο μοντέλο έχει την αρμοδιότητα να εντοπίζει, γρήγορα και επαρκώς, πιθανά «καλά» σημεία, τα οποία θα βοηθήσουν στην αποτελεσματική σύγκλιση του αλγορίθμου και μόνο σε αυτά τα «καλά» σημεία θα υπολογίζεται η τιμή της «χρονοβόρας» πραγματικής αντικειμενικής συνάρτησης. Αν κάτι τέτοιο καταστεί εφικτό, το πρόβλημα υπέρβασης των λογικών χρονικών πλαισίων για την ολοκλήρωση της διαδικασίας βελτιστοποίησης θα είναι πλέον παρελθόν και εκτός από αυτό, πλήρως ικανοποιητική θα είναι και η ακρίβεια σύγκλισης του αλγορίθμου στο πραγματικό ακρότατο της συνάρτησης.

Ο πρώτος γνωστός επιτυχημένος αλγόριθμος βελτιστοποίησης, που κάνει χρήση υποκατάστατου μοντέλου για την επιτάχυνση της διαδικασίας, αναπτύχθηκε από τον Jones et al. (1998) και είναι ο λεγόμενος Efficient Global Optimization (EGO) algorithm, ο οποίος χρησιμοποιεί σαν μεταμοντέλο το Kriging. Από τότε μέχρι σήμερα αναπτύσσονται συνεχώς καινούριες ιδέες χρήσης μεταμοντέλων, οι οποίες μπορεί να διαφέρουν αρκετά μεταξύ τους, αλλά έχουν παρόλα αυτά τον ίδιο πάντοτε στόχο, να παράξουν γρήγορους και αποτελεσματικούς αλγορίθμους.

4.2. Ο εξελικτικός αλγόριθμος ανόπτωσης – απλόκου εμπλουτισμένος με υποκατάστατα μοντέλα

Ο εξελικτικός αλγόριθμος ανόπτωσης - απλόκου εμπλουτισμένος με υποκατάστατα μοντέλα (Surrogate Enhanced Evolutionary Annealing Simplex algorithm – SEEAS) αναπτύχθηκε από τους I.Tsoukalas, P.Kossieris, A.Efstratiadis και C.Makropoulos (2016) και είναι ένας αλγόριθμος ολικής βελτιστοποίησης, ειδικά σχεδιασμένος για «χρονοβόρες» στοχαστικές συναρτήσεις, οι οποίες αντιπροσωπεύουν την πλειονότητα των συναρτήσεων στα προβλήματα υδατικών πόρων. Στόχος του SEEAS είναι η ικανοποιητική προσέγγιση του ολικού ακροτάτου, με τον ελάχιστο δυνατό αριθμό υπολογισμών της αντικειμενικής συνάρτησης.

Ο αλγόριθμος δημιουργήθηκε με σκοπό να μπορεί να ανταποκριθεί εξίσου ικανοποιητικά στις δύο βασικές κατηγορίες προβλημάτων υδατικών πόρων, που είναι 1) το πρόβλημα λήψης απόφασης και 2) το «αντίστροφο» πρόβλημα ή αλλιώς το πρόβλημα «καλιμπραρίσματος» του μοντέλου. Στο 1^ο πρόβλημα είναι γνωστές όλες οι ιδιότητες του συστήματος και οι σχετικές διαδικασίες και αναζητείται η βέλτιστη πολιτική διαχείρισης του συστήματος (πχ. ποσότητα σχεδιασμού κάποιου προϊόντος), ενώ στο 2^ο πρόβλημα κάποιες από τις ιδιότητες του συστήματος θεωρούνται άγνωστες και επιδιώκεται η εύρεση της βέλτιστης τιμής τους.

Ο SEEAS αποτελεί σύζευξη του εξελικτικού αλγορίθμου ανόπτωσης – απλόκου, ο οποίος αναλύθηκε στο [Κεφάλαιο 3](#), με ένα υποκατάστατο μοντέλο. Το υποκατάστατο μοντέλο που χρησιμοποιεί ο αλγόριθμος είναι υποκατηγορία μιας ευρύτερης κατηγορίας μεταμοντέλων, που είναι τα λεγόμενα Radial Basis Functions (RBFs).

4.2.1. Επισκόπηση του αλγορίθμου

Ο SEEAS είναι μια επέκταση του εξελικτικού αλγορίθμου ανόπτησης – απλόκου (EAS), καθώς ενσωματώνει στον πυρήνα του τεχνικές υποκατάστατων μοντέλων για επιτάχυνση της διαδικασίας βελτιστοποίησης. Ο αλγόριθμος χρησιμοποιεί μια ευρέως διαδεδομένη κατηγορία μεταμοντέλων, που έχει πειραματικά αποδεδειγμένη επιτυχία, τα RBFs, και πιο συγκεκριμένα το Cubic RBF with Linear Polynomial Tail.

Κατά την επαναληπτική διαδικασία του κώδικα, διατηρείται ένα εξωτερικό αρχείο, το οποίο περιλαμβάνει όλα τα σημεία, στα οποία έχει γίνει υπολογισμός της πραγματικής «χρονοβόρας» αντικειμενικής συνάρτησης. Το εξωτερικό αρχείο υπάρχει, για να μπορεί το υποκατάστατο μοντέλο να κάνει όσο το δυνατόν ακριβέστερη προσαρμογή πάνω σε όλα τα γνωστά σημεία και να μπορεί έτσι το μεταμοντέλο να προσομοιάσει στο καλύτερο δυνατό βαθμό την απόκριση της πραγματικής στοχικής συνάρτησης. Αν αυτό επιτευχθεί, τότε θα μπορεί πράγματι να κάνει σωστές προβλέψεις των πιθανών «καλών» σημείων, τα οποία θα βοηθήσουν στη σύγκλιση προς το ολικό ακρότατο και στα οποία αξίζει να γίνει και υπολογισμός της πραγματικής αντικειμενικής συνάρτησης, συμβάλλοντας με τον τρόπο αυτό στην επιτάχυνση ολοκλήρωσης της διαδικασίας. Πέρα από τα παραπάνω, το υποκατάστατο μοντέλο στον SEEAS βοηθάει και το κατερχόμενο άπλοκο στην διερεύνηση σωστών κατευθύνσεων για πραγματοποίηση των γνωστών κινήσεων του. Πιο αναλυτικά, οι προβλέψεις τις οποίες κάνει το μεταμοντέλο διοχετεύονται σε μια συνάρτηση, την λεγόμενη Acquisition Function (AF), η οποία μετά από μια επεξεργασία των ορισμάτων που δέχεται, κάνει τις τελικές προβλέψεις, υποδεικνύοντας τα σημεία στα οποία θα γίνει, αμέσως μετά, ο υπολογισμός της πραγματικής στοχικής συνάρτησης.

Μια τυπική επανάληψη του αλγορίθμου ξεκινάει με την προσαρμογή του υποκατάστατου μοντέλου στο εξωτερικό αρχείο. Έπειτα, καλείται η αρχική έκδοση του EAS για να κάνει ελαχιστοποίηση στην Acquisition Function και μετά την ολοκλήρωση αυτής της διαδικασίας παράγεται ένα σημείο (το ολικό ελάχιστο της AF), το οποίο αποτελεί έναν υποψήφιο για να εισέλθει στον πληθυσμό λύσεων της συγκεκριμένης γενιάς. Στο σημείο αυτό, λοιπόν, που προέκυψε από την ελαχιστοποίηση, υπολογίζεται η τιμή της πραγματικής αντικειμενικής συνάρτησης και εφόσον είναι μικρότερη (και άρα καλύτερη) από την τιμή του χειρότερου σημείου στον πληθυσμό, τότε το αντικαθιστά. Έπειτα, ακολουθεί η γνωστή διαδικασία παραγωγής του απλόκου, το οποίο εκτελεί τις γνωστές κινήσεις της μεθόδου Nelder-Mead σε συνδυασμό με το αυτορρυθμιζόμενο χρονοδιάγραμμα ανόπτησης για προσθήκη στοχαστικότητας και διαφυγή από τοπικά ακρότατα. Το μόνο που αλλάζει κατά τη γνωστή αυτή διαδικασία, είναι ότι όλη η προσπάθεια ενισχύεται από το μεταμοντέλο, το οποίο βοηθάει στη διερεύνηση κατευθύνσεων για την εκτέλεση των κινήσεων του απλόκου. Κατά την ολοκλήρωση κάθε επαναληπτικού κύκλου του αλγορίθμου, παράγεται τουλάχιστον ένα καινούριο σημείο, το οποίο εισέρχεται στον πληθυσμό, αντικαθιστώντας κάποιο από τα μέλη του.

4.2.2. Acquisition Function

Έστω ένα σύνολο από, N_s στο πλήθος, σημεία x_s , που βρίσκονται στο εξωτερικό αρχείο, με πραγματική τιμή αντικειμενικής συνάρτησης $f(x_s)$ και ένα άλλο σύνολο από, N_c στο πλήθος, σημεία x_c με προβλεπόμενη τιμή από κάποιο υποκατάστατο μοντέλο $s(x_c)$. Τα σημεία x_c είναι τα υποψήφια σημεία. Η διαδικασία υπολογισμού της τιμής της Acquisition Function για τα σημεία x_c , δηλαδή της $AF(x_c)$ είναι η εξής:

Βήμα 1^ο : Γίνεται κανονικοποίηση των τιμών $s(x_c)$ σύμφωνα με τη σχέση:

$$s^*(x_c) = \frac{s(x_c) - s_{min}}{s_{max} - s_{min}} \quad (4.1)$$

όπου s_{min} και s_{max} είναι η ελάχιστη και η μέγιστη, αντίστοιχα, προβλεπόμενη τιμή του υποκατάστατου μοντέλου που προέκυψε από όλα τα υποψήφια σημεία

Βήμα 2^ο : Για κάθε υποψήφιο σημείο x_c υπολογίζεται η Ευκλείδεια απόστασή του από όλα τα σημεία x_s του εξωτερικού αρχείου και βρίσκεται η ελάχιστη από αυτές, έστω $d_i(x_c)$. Έπειτα, οι αποστάσεις $d_i(x_c)$ κανονικοποιούνται σύμφωνα με τη σχέση:

$$d_i^*(x_c) = \frac{d_{i_{max}} - d_i(x_c)}{d_{i_{max}} - d_{i_{min}}} \quad (4.2)$$

όπου $d_{i_{min}}$ και $d_{i_{max}}$ είναι η ελάχιστη και η μέγιστη, αντίστοιχα, απόσταση που προέκυψε από όλα τα υποψήφια σημεία.

Βήμα 3^ο : Για κάθε υποψήφιο σημείο υπολογίζεται η $AF(x_c)$ σύμφωνα με τη σχέση:

$$AF(x_c) = w \times s^*(x_c) + (1 - w) \times d_i^*(x_c) \quad (4.3)$$

όπου w ο συντελεστής βάρους που ορίζεται στη σχέση (4.5).

Από τα συνολικά υποψήφια σημεία x_c , εκείνη με τη χαμηλότερη τιμή της AF , θα επιλεγεί και στο σημείο αυτό θα γίνει υπολογισμός της πραγματικής «χρονοβόρας» αντικειμενικής συνάρτησης.

4.2.3. Αναλυτική περιγραφή της υπολογιστικής διαδικασίας του SEEAS

Έστω ένα πρόβλημα βελτιστοποίησης, το οποίο περιγράφεται από n συνεχείς μεταβλητές ελέγχου. Όπως είναι στημένος ο αλγόριθμος, το πρόβλημα αντιμετωπίζεται σαν πρόβλημα ελαχιστοποίησης της στοχικής συνάρτησης που το περιγράφει. Η στοχική συνάρτηση είναι βαθμωτή, μη γραμμική και αντιστοιχεί πρακτικά σε κάποιο μέτρο επίδοσης του μοντέλου προσομοίωσης που περιγράφει το πρόβλημα. Για να ξεκινήσει ο αλγόριθμος, ο χρήστης απαιτείται να εισάγει τα ακόλουθα στοιχεία, σαν παραμέτρους εισόδου:

- το πλήθος των μεταβλητών ελέγχου του προβλήματος n

- το σταθερό μέγεθος του πληθυσμού m , με $m \geq n+1$
- ένα διάνυσμα x_{\min} που αντιστοιχεί στα κάτω όρια τιμών των μεταβλητών ελέγχου
- ένα διάνυσμα x_{\max} που αντιστοιχεί στα άνω όρια τιμών των μεταβλητών ελέγχου
- την αντικειμενική συνάρτηση που περιγράφει το πρόβλημα
- το μέγιστο επιτρεπτό πλήθος υπολογισμών της αντικειμενικής συνάρτησης
- μια τιμή για το όριο σύγκλισης, το οποίο αν επιτευχθεί, ο αλγόριθμος οδηγείται σε πρόωρο τερματισμό
- την παράμετρο του αυτορρυθμιζόμενου χρονοδιαγράμματος ανόπτησης β
- την πιθανότητα πραγματοποίησης μετάλλαξης p_m
- το μέγιστο επιτρεπτό πλήθος διαδοχικών αναρριχήσεων

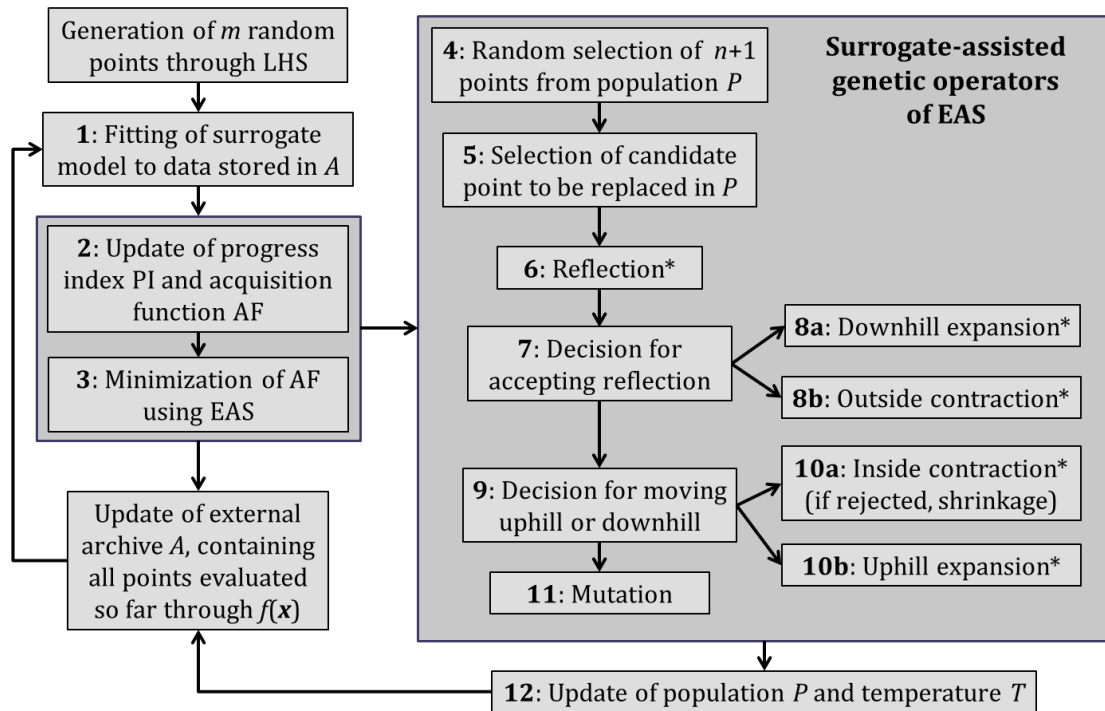
Ο SEEAS χρησιμοποιεί δύο αρχεία, το P και το A. Το P αντιστοιχεί στον πληθυσμό λύσεων, ο οποίος εξελίσσεται από γενιά σε γενιά με την συνέχιση της υπολογιστικής διαδικασίας. Το A αντιστοιχεί στο λεγόμενο εξωτερικό αρχείο, που όπως αναφέρθηκε παραπάνω, περιλαμβάνει όλα τα σημεία, στα οποία έχει γίνει υπολογισμός της πραγματικής στοχικής συνάρτησης. Ο πληθυσμός P παραμένει σταθερός στο μέγεθός του καθ' όλη την εκτέλεση του αλγορίθμου με m σημεία μέσα, ενώ το μέγεθος του εξωτερικού αρχείου A αυξάνει διαρκώς, κάθε φορά που γίνεται ένας νέος υπολογισμός της στοχικής συνάρτησης. Ο αρχικός πληθυσμός λύσεων της 1^{ης} γενιάς παράγεται με χρήση της στατιστικής μεθόδου Latin Hypercube Sampling (LHS), για την οποία έχει γίνει αναφορά στο [Υποκεφάλαιο 3.3](#). Όπως γίνεται αντιληπτό, στην πρώτη επανάληψη του αλγορίθμου το εξωτερικό αρχείο A ταυτίζεται με τον πληθυσμό P.

Για την εξισορρόπηση στοχαστικότητας και ντετερμινιστικότητας του αλγορίθμου, σε πλήρη αντιστοιχία με τον EAS, έτσι και ο SEEAS χρησιμοποιεί ένα αυτορρυθμιζόμενο χρονοδιάγραμμα ανόπτησης, το οποίο εφαρμόζεται μέσω της μεταβλητής T. Αυτή αντιστοιχεί στην εκάστοτε θερμοκρασία του συστήματος και ρυθμίζεται μέσω της καλύτερης και της χειρότερης κορυφής του απλόκου που σχηματίζεται, του πληθυσμού P και μέσω ενός δείκτη PI, ο οποίος δίνεται από τη σχέση:

$$PI = \frac{\log(FE)}{\log(MFE)} \quad (4.4)$$

όπου το FE (Function Evaluations) αντιστοιχεί στον εκάστοτε αριθμό υπολογισμών της αντικειμενικής συνάρτησης που έχουν πραγματοποιηθεί και το MFE (Maximum Function Evaluations) στον αντίστοιχο μέγιστο επιτρεπτό.

Ένα συνοπτικό διάγραμμα ροής της υπολογιστικής διαδικασίας του SEEAS είναι το ακόλουθο:



Σχήμα 4-1: Διάγραμμα ροής της υπολογιστικής διαδικασίας του SEEAS (Πηγή: Tsoukalas, I., et. al., 2016. *Surrogate-enhanced evolutionary annealing simplex algorithm for effective and efficient optimization of water resources problems on a budget.*)

Στο σημείο αυτό, γίνεται μια αναλυτική περιγραφή των βημάτων που εκτελεί ο SEEAS σε μια τυπική επανάληψη του:

Βήμα 1^ο: Το μεταμοντέλο κάνει προσαρμογή πάνω σε όλα τα σημεία που υπάρχουν τη δεδομένη στιγμή στο εξωτερικό αρχείο A.

Βήμα 2^ο: Ένας συντελεστής βάρους w που χρησιμοποιεί η Acquisition Function υπολογίζεται, σύμφωνα με τη σχέση:

$$w = \max[0.75, \min(PI, 0.95)] \quad (4.5)$$

Βήμα 3^ο: Η αρχική εκδοχή του EAS εκτελεί ελαχιστοποίηση πάνω στην Acquisition Function, μέσω της οποίας προκύπτει μια νέα λύση του προβλήματος, το σημείο x_p . Στο σημείο αυτό γίνεται υπολογισμός της πραγματικής αντικειμενικής συνάρτησης, δηλαδή υπολογίζεται το $f(x_p)$ και εφόσον $f(x_p) < f(x_{\text{worst}})$, όπου x_{worst} το σημείο του πληθυσμού με τη μεγαλύτερη τιμή στοχικής συνάρτησης, τότε το x_p αντικαθιστά το x_{worst} και εισέρχεται στον πληθυσμό.

Βήμα 4^ο: Από τον συνολικό πληθυσμό επιλέγονται τυχαία $n+1$ σημεία, τα οποία σχηματίζουν το άπλοκο $S = \{s_1, s_2, \dots, s_{n+1}\}$. Οι κορυφές του απλόκου τοποθετούνται σε αύξουσα σειρά με την κορυφή s_1 να αντιστοιχεί στο σημείο που έχει τη χαμηλότερη (άρα καλύτερη) τιμή στοχικής συνάρτησης, ενώ το s_{n+1} να έχει τη μεγαλύτερη (άρα χειρότερη).

Βήμα 5^ο : Από το υποσύνολο του απλόκου $\{ s_2, \dots, s_{n+1} \}$, το οποίο θα περιλαμβάνει όλες τις κορυφές του απλόκου εκτός της καλύτερης s_1 , βρίσκεται η κορυφή x_w που θα οδηγηθεί προς αντικατάσταση. Η κορυφή αυτή βρίσκεται σύμφωνα με το κριτήριο Metropolis και θα είναι η κορυφή εκείνη που μεγιστοποιεί τη συνάρτηση:

$$g(s) = f(s) + uT \quad (4.6)$$

όπου u ένας τυχαίος αριθμός που ακολουθεί την ομοιόμορφη κατανομή στο $[0, 1]$ και T η τρέχουσα θερμοκρασία του συστήματος. Στις πρώτες επαναλήψεις του αλγορίθμου η θερμοκρασία του συστήματος είναι αρκετά υψηλή και η επίδραση της τυχαιότητας uT καθορίζει την κορυφή που θα οδηγηθεί προς αντικατάσταση σε μεγάλο βαθμό. Αντίθετα, στις επόμενες επαναλήψεις, που η διαδικασία σύγκλισης προχωρά, το σύστημα ψύχεται και έτσι η θερμοκρασία του συστήματος μειώνεται. Αυτό οδηγεί σε μείωση της επίδρασης του όρου τυχαιότητας uT και έτσι ο αλγόριθμος γίνεται περισσότερο ντετερμινιστικός, ακολουθώντας τη λογική του αλγορίθμου Nelder – Mead.

Βήμα 6^ο : Το άπλοκο εκτελεί ανάκλαση ως προς την κορυφή x_w και έτσι προκύπτουν νέες υποψήφιες λύσεις σύμφωνα με τη σχέση:

$$x_{cr}^k = g + (0.5 + \delta_k)(g - x_w) \quad (4.7)$$

όπου g είναι το κεντροειδές του απλόκου χωρίς την χειρότερη κορυφή w , το οποίο δίνεται από τη σχέση:

$$g = \frac{1}{n} (\sum_{i=1}^{n+1} s_i - w) \quad (4.8)$$

ενώ δ_k είναι ένας συντελεστής κλίμακας με τιμές που «σπάνε» το διάστημα $[0, 1]$ σε N_r τμήματα ίσου μεγέθους και δίνεται από τη σχέση:

$$\delta_k = \frac{(k - 1)}{(N_r - 1)} \quad \text{για } k = 1, 2, \dots, N_r \quad (4.9)$$

Από όλα τα υποψηφία σημεία που προκύπτουν από τις ανακλάσεις, επιλέγεται εκείνο που αντιστοιχεί σε χαμηλότερη τιμή της Acquisition Function ως το καλύτερο και αυτό θεωρείται η λύση που προκύπτει από την ανάκλαση του απλόκου. Στο σημείο αυτό x_r γίνεται και υπολογισμός, επομένως, της πραγματικής στοχικής συνάρτησης, υπολογίζεται δηλαδή το $f(x_r)$ και ταυτόχρονα το x_r εισέρχεται και στο εξωτερικό αρχείο.

Βήμα 7^ο : Αν $f(x_r) < f(x_w)$, τότε το x_w αντικαθίσταται από το x_r στον πληθυσμό και ο αλγόριθμος συνεχίζει στα βήματα $\delta\alpha$ ή $\delta\beta$, ανάλογα με το αποτέλεσμα της σύγκρισης $f(x_r) < f(x_1)$. Διαφορετικά, ο αλγόριθμος συνεχίζει στο βήμα 9, όπου θα καθοριστεί αν το σημείο x_r θα γίνει τελικά αποδεκτό ή θα απορριφθεί και θα αναζητηθεί ένας άλλος υποψήφιος.

Βήμα 8α : Αν το σημείο x_r που προέκυψε από την ανάκλαση είναι καλύτερο και από το καλύτερο σημείο του απλόκου x_1 , αν δηλαδή $f(x_r) < f(x_1)$, τότε εκτελούνται βήματα επέκτασης προς την κατεύθυνση $x_r - g$. Από την επέκταση αυτή προκύπτουν συνολικά N_e δοκιμαστικά σημεία, σύμφωνα με τη σχέση:

$$x_{ce}^k = g + \delta_k(x_r - g) \quad (4.10)$$

όπου δ_k είναι ένας συντελεστής κλίμακας που δίνεται από τη σχέση:

$$\delta_k = \delta_{k-1} + \frac{k-1}{N_e-1} \text{ για } k = 1, 2, \dots, N_e \quad (4.11)$$

Η επέκταση συνεχίζεται για όσο παρατηρείται βελτίωση της Acquisition Function. Το καλύτερο σημείο x_e που προέκυψε από τα συνολικά βήματα επέκτασης εισέρχεται στο εξωτερικό αρχείο και στον πληθυσμό, εφόσον $f(x_e) < f(x_r)$. Στην περίπτωση αυτή ο αλγόριθμος προχωράει στο βήμα 12.

Βήμα 8β : Αν αντίθετα, το σημείο x_r που προέκυψε από την ανάκλαση δεν είναι καλύτερο από το καλύτερο σημείο του απλόκου x_1 , αν δηλαδή $f(x_r) > f(x_1)$, τότε επιχειρείται η εύρεση μιας λύσης στη γειτονιά του x_1 . Έτσι, πραγματοποιούνται βήματα συμπίεσης στη γειτονιά ανάμεσα στο κεντροειδές του απλόκου και στο σημείο της ανάκλασης και παράγονται N_c υποψήφιες λύσεις, σύμφωνα με τη σχέση:

$$x_{cc}^k = g + (0.25 + 0.5\delta_k)(x_r - g) \text{ όπου } \delta_k = \frac{k-1}{N_c-1} \text{ για } k = 1, 2, \dots, N_c \quad (4.12)$$

Ο υποψήφιος εκείνος που δίνει τη χαμηλότερη τιμή της Acquisition Function, θα είναι το σημείο x_c , το οποίο εισέρχεται σίγουρα στο εξωτερικό αρχείο και στον πληθυσμό, μόνο αν $f(x_c) < f(x_r)$. Στην περίπτωση αυτή ο αλγόριθμος προχωράει στο βήμα 12.

Βήμα 9^ο : Αν το σημείο x_r που προέκυψε από την ανάκλαση είναι χειρότερο από το χειρότερο σημείο του απλόκου x_w , αν δηλαδή $f(x_r) > f(x_w)$ τότε χρησιμοποιείται η συνθήκη:

$$f(x_r) - uT > f(x_w) + uT \quad (4.13)$$

για να καθοριστεί αν το άπλοκο θα πραγματοποιήσει εσωτερική συμπίεση για τον εντοπισμό κάποιου τοπικού ακροτάτου ή αν θα εκτελεστούν βήματα αναρρίχησης, δηλαδή βήματα επέκτασης του απλόκου προς την κατεύθυνση όπου αυξάνεται η τιμή της αντικειμενικής συνάρτησης. Αν η συνθήκη προκύψει αληθής, τότε ο αλγόριθμος προχωράει στο βήμα 10α, αλλιώς προχωράει στο βήμα 10β.

Βήμα 10α : Πραγματοποιείται μείωση της θερμοκρασίας του συστήματος, σύμφωνα με μια παράμετρο ψ , η οποία δίνεται από τη σχέση:

$$\psi = \max\left(1 - \frac{\log(FE-m)}{\log(MFE)}, 0.50\right) \quad (4.14)$$

Το σημείο της ανάκλασης x_r απορρίπτεται και εκτελούνται βήματα εσωτερικής συμπίεσης του απλόκου στη γειτονιά ανάμεσα στο κεντροειδές και στην χειρότερη κορυφή του, σύμφωνα με τη σχέση:

$$x_{cc}^k = g - (0.25 + 0.5\delta_k)(g - x_w), \text{ όπου } \delta_k = \frac{k-1}{N_c-1} \text{ για } k = 1, 2, \dots, N_c \quad (4.15)$$

Ο υποψήφιος εκείνος που δίνει τη χαμηλότερη τιμή της Acquisition Function, θα είναι το σημείο x_c , το οποίο εισέρχεται σίγουρα στο εξωτερικό αρχείο και στον πληθυσμό μόνο αν $f(x_c) < f(x_w)$, αντικαθιστώντας το x_w . Διαφορετικά, το άπλοκο εκτελεί βήματα συρρίκνωσης προς την κατεύθυνση της καλύτερης κορυφής του απλόκου x_1 , σύμφωνα με τη σχέση:

$$s'_i = 0.5(x_1 + s_i) \text{ για } i = 2, 3, \dots, n + 1 \quad (4.16)$$

Εφόσον ενεργοποιηθεί ο μηχανισμός αυτός, παράγονται n καινούριες λύσεις του προβλήματος και αντικαθιστούνται στον πληθυσμό όλες οι κορυφές του απλόκου, εκτός της x_1 , από τα n καινούρια αυτά σημεία.

Βήμα 10β : Το σημείο της ανάκλασης x_r γίνεται αποδεκτό, παρόλο που είναι χειρότερο από το x_w και ενεργοποιείται ένας μηχανισμός διαφυγής από τοπικό ακρότατο. Ο μηχανισμός αυτός περιλαμβάνει βήματα αναρρίχησης όπως ονομάζονται, δηλαδή βήματα επέκτασης του απλόκου προς την κατεύθυνση όπου αυξάνεται η τιμή της αντικειμενικής συνάρτησης. Τα βήματα αυτά εκτελούνται σύμφωνα με τη σχέση (4.12) και προκύπτουν νέα σημεία. Αντίστοιχα με τις υπόλοιπες περιπτώσεις, μέσω της Acquisition Function προκύπτει το σημείο x_u . Αν $f(x_u) < f(x_r)$, τότε το x_u εισάγεται στο εξωτερικό αρχείο και αντικαθιστά το x_r και ο αλγόριθμος προχωράει στο βήμα 12. Διαφορετικά, μιας και καμία από τις προηγούμενες ενέργειες δεν είχε αποτέλεσμα, ο αλγόριθμος προχωράει στο βήμα 11 όπου ενεργοποιείται ένας μηχανισμός μετάλλαξης.

Βήμα 11° : Ενεργοποιείται ο μηχανισμός της μετάλλαξης. Στόχος του μηχανισμού αυτού είναι να δημιουργηθεί ένα νέο σημείο εκτός του εύρους της τυπικής απόκλισης γύρω από τη μέση τιμή του τρέχοντος πληθυσμού, δηλαδή το σημείο αυτό να είναι εκτός του εύρους $[\mu_\chi - \sigma_\chi, \mu_\chi + \sigma_\chi]$. Οι συντεταγμένες του νέου αυτού σημείου προκύπτουν ως εξής:

$$x_i = \mu_i \pm (\sigma_i + u d_i) \quad (4.17)$$

όπου μ_i και σ_i είναι η μέση τιμή και η τυπική απόκλιση αντίστοιχα του πληθυσμού για την i συντεταγμένη και u ένας όρος τυχαιότητας που ακολουθεί την ομοιόμορφη κατανομή στο $[0, 1]$, ενώ d_i είναι η υπολειπόμενη απόσταση μέχρι τα όρια του εξωτερικού πεδίου τιμών του χώρου αναζήτησης για την i συντεταγμένη. Η σχέση (4.17) μπορεί να εκτελεστεί είτε με πρόσημο (+) είτε με πρόσημο (-). Τα δύο πρόσημα έχουν ίση πιθανότητα να επιλεγθούν. Αν επιλεγεί το πρόσημο (+), τότε η αντίστοιχη συντεταγμένη του σημείου προκύπτει από ομοιόμορφη κατανομή στο εύρος $[\mu_i + \sigma_i, x_{i \max}]$ ενώ για το πρόσημο (-) το αντίστοιχο εύρος είναι το $[x_{i \min}, \mu_i - \sigma_i]$. Αυτός ο μηχανισμός ενεργοποιείται για την επίτευξη ποικιλομορφίας στον πληθυσμό και την αποτελεσματικότερη εξερεύνηση, συνεπώς, του χώρου αναζήτησης. Έτσι, προκύπτει το νέο σημείο x_m , το οποίο εισέρχεται στο εξωτερικό αρχείο και γίνεται αποδεκτό και στον πληθυσμό, αν $f(x_m) < f(x_r)$. Διαφορετικά, ανάλογα με μια πιθανότητα μετάλλαξης, η οποία εισάγεται από το χρήστη, επιλέγεται είτε το x_m είτε το x_r για να αντικαταστήσει το x_w στον πληθυσμό.

Βήμα 12° : Γίνεται εύρεση της καλύτερης λύσης x_b και της χειρότερης λύσης x_{worst} που υπάρχει στον πληθυσμό και επιχειρείται να βρεθεί μια ακόμα λύση του προβλήματος στη γειτονιά του x_b . Το σημείο αυτό βρίσκεται με χρήση της μεθόδου DDS, μέσω της οποίας παράγονται N_t υποψηφίες λύσεις. Από τους N_t συνολικούς υποψηφίους επιλέγεται, ως γνωστόν, εκείνος που δίνει τη χαμηλότερη τιμή στην Acquisition Function. Έτσι, παράγεται μια νέα λύση x_{ref} , η οποία εισέρχεται σίγουρα στο εξωτερικό αρχείο και αντικαθιστά το x_{worst} στον πληθυσμό, εφόσον $f(x_{\text{ref}}) < f(x_{\text{worst}})$.

Βήμα 13° : Ολοκληρώνεται ο συγκεκριμένος επαναληπτικός κύκλος. Γίνεται έλεγχος αν ικανοποιείται κάποιος από τους τρόπους τερματισμού του αλγορίθμου, διαφορετικά

ξεκινάει ο επόμενος επαναληπτικός κύκλος. Οι δύο τρόποι με τους οποίους μπορεί να ολοκληρωθεί ο αλγόριθμος είναι:

- 1) ο αριθμός των υπολογισμών της στοχικής συνάρτησης να υπερβεί το μέγιστο επιτρεπτό όριο, το οποίο εισάγεται από τον χρήστη
- 2) όταν επιτευχθεί το κριτήριο σύγκλισης του αλγορίθμου, δηλαδή όταν η σύγκλισή του είναι καλύτερη από το όριο σύγκλισης, το οποίο εισάγεται από το χρήστη

Αναλυτικότερες πληροφορίες σχετικά τις θεμελιώδεις αρχές και τις υπολογιστικές διαδικασίες του αλγορίθμου, όπως επίσης και σχετικά με την επίδοση του σε πραγματικά προβλήματα βελτιστοποίησης, συγκριτικά με άλλους δημοφιλείς αλγορίθμους, παρουσιάζονται στην σχετική εργασία των I.Tsoukalas, P.Kossieris, A.Efstratiadis και C.Makropoulos (2016):

➤ <https://www.itia.ntua.gr/el/docinfo/1587/>

4.2.4. Υλοποίηση του SEEAS σε γλώσσα προγραμματισμού Python

Ένα επόμενο εγχείρημα της παρούσας διπλωματικής εργασίας ήταν η μεταφορά του εξελικτικού αλγορίθμου ανόπτησης – απλόκου εμπλουτισμένου με υποκατάστατα μοντέλα σε γλώσσα προγραμματισμού Python. Όπως ο EAS, έτσι και ο SEEAS παραλήφθηκε, αρχικά, σε προγραμματιστικό περιβάλλον MATLAB. Ακολούθησε η κατανόηση των θεμελιωδών αρχών και υπολογιστικών διαδικασιών του, για να καταστεί εφικτή η επιτυχής μετάφραση του.

Στον ακόλουθο σύνδεσμο γίνεται η παρουσίαση υλοποίησης του SEEAS σε γλώσσα προγραμματισμού Python:

<https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/SEEAS>

Κεφάλαιο 5^ο

5 Θεμελιώδεις αρχές μηχανικής μάθησης και επισκόπηση συναφών δημοφιλών μοντέλων

5.1. Η έννοια της αυτοεκμάθησης μηχανών

Στο επόμενο κεφάλαιο ([Κεφάλαιο 6^ο](#)) πρόκειται να γίνει η παρουσίαση του νέου πρωτότυπου κώδικα που κατασκευάστηκε στα πλαίσια της παρούσας διπλωματικής εργασίας. Ο κώδικας αυτός αποτελεί επέκταση του εξελικτικού αλγορίθμου ανόπτησης – απλόκου εμπλουτισμένου με υποκατάστατα μοντέλα (SEEAS) και έχει ως απώτερο σκοπό την περαιτέρω βελτίωση της επίδοσης του. Ο νέος αλγόριθμος ενσωματώνει στον πυρήνα του πολλά μοντέλα αυτοεκμάθησης μηχανών. Ως εκ τούτου, κρίνεται σκόπιμο στο κεφάλαιο αυτό να γίνει μια εισαγωγή στη θεωρία μηχανικής μάθησης (machine learning) και να παρουσιαστούν ορισμένα συναφή μοντέλα (machine learning models), τα οποία χρησιμοποιήθηκαν για την εξαγωγή του τελικού αποτελέσματος.

Η μηχανική μάθηση είναι ένας κλάδος της επιστήμης των υπολογιστών, στον οποίο χρησιμοποιούνται τεχνικές στατιστικής, με στόχο να αποκτήσουν τα υπολογιστικά συστήματα την ικανότητα να «μαθαίνουν» από δεδομένα και να κάνουν προβλέψεις, χωρίς να ακολουθούν κάποιον ντετερμινιστικό αλγόριθμο. Όπως αναφέρει, χαρακτηριστικά ο A. Samuel (1959), η μηχανική μάθηση αποτελεί το πεδίο μελέτης, που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί. Το επιστημονικό πεδίο της μηχανικής μάθησης εντάσσεται στον ευρύτερο κλάδο της τεχνητής νοημοσύνης (artificial intelligence). Η μηχανική μάθηση διακρίνεται σε διάφορες κατηγορίες, ανάλογα με τις βασικές λειτουργίες που εκτελεί αυτή σε κάθε περίπτωση, αλλά οι κυριότερες εξ αυτών είναι αναμφίβολα οι εξής:

- 1) Επιβλεπόμενη Μάθηση (Supervised Learning), στην οποία εντάσσονται οι αλγόριθμοι, που προσπαθούν να μάθουν μια συνάρτηση μέσω ενός συνόλου δεδομένων, από τα οποία για κάθε μια είσοδο είναι γνωστή η αντίστοιχη επιθυμητή έξοδος.
- 2) Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning), στην οποία εντάσσονται οι αλγόριθμοι που χρησιμοποιούν ένα σύνολο δεδομένων, για τα οποία η επιθυμητή έξοδος δεν είναι γνωστή.
- 3) Ενισχυτική Μάθηση (Reinforcement Learning), στην οποία εντάσσονται οι αλγόριθμοι που βρίσκονται στα πλαίσια ενός περιβάλλοντος, από το οποίο λαμβάνουν ως είσοδο συγκεκριμένες καταστάσεις του και τους παρέχεται κάποιου είδους επιβράβευση (reward).

Μια τυπική διαδικασία μηχανικής μάθησης αποτελείται από τα εξής βήματα:

Βήμα 1^ο : Γίνεται η συλλογή των δεδομένων, τα οποία θα χρησιμοποιηθούν για να εκπαιδευτεί το μοντέλο και έπειτα να δοκιμαστεί η επίδοσή του.

Βήμα 2^ο : Γίνεται η προετοιμασία των δεδομένων (data preprocessing), μια διαδικασία κατά την οποία, τα δεδομένα «καθαρίζονται» από την «άχρηστη» πληροφορία που υπάρχει ανάμεσα τους και τροποποιούνται κατάλληλα, για να έχουν όλα μια ενιαία μορφή.

Βήμα 3^ο : Γίνεται η επιλογή του μοντέλου μηχανικής μάθησης που θα χρησιμοποιηθεί. Δεν υπάρχει κάποιος γενικός κανόνας για το ποιο μοντέλο είναι το καταλληλότερο σε κάθε κατηγορία προβλήματος. Απαιτείται να γίνεται πάντοτε δοκιμή πολλών μοντέλων, ούτως ώστε να εξακριβωθεί από τον προγραμματιστή ποιο από αυτά φαίνεται να ανταποκρίνεται καλύτερα στο συγκεκριμένο πρόβλημα και αξίζει να προτιμηθεί έναντι των υπολοίπων.

Βήμα 4^ο : Γίνεται η εκπαίδευση (training) του μοντέλου που επιλέχθηκε στο προηγούμενο βήμα. Η εκπαίδευση πραγματοποιείται με χρήση των δεδομένων που έχουν προετοιμαστεί στο βήμα 2. Συνηθίζεται το συνολικό δείγμα των δεδομένων (dataset) να διασπάται σε δύο μικρότερα, ένα το οποίο θα χρησιμοποιηθεί κατά την εκπαίδευση του μοντέλου (training set) και ένα το οποίο θα χρησιμοποιηθεί κατά την δοκιμή του (test set).

Βήμα 5^ο : Το εκπαιδευμένο, πλέον, μοντέλο είναι έτοιμο να κάνει τις πρώτες του εκτιμήσεις (predictions). Συνηθίζεται οι πρώτες προβλέψεις του μοντέλου να γίνονται πάνω στο δείγμα δοκιμής (test set), για να μπορεί εύκολα να προσδιοριστεί η επίδοσή του.

Βήμα 6^ο : Εφόσον το μοντέλο κατά την εκτέλεση του βήματος 5 δεν ανταποκρίθηκε όπως αναμενόταν, ακολουθεί η βελτίωση του μοντέλου, κατά την οποία πραγματοποιούνται αλλαγές στην αρχιτεκτονική του και στις υπερπαραμέτρους (hyperparameters) του. Ακολούθως, το νέο βελτιωμένο μοντέλο επαναεκπαιδεύεται και δοκιμάζεται μέχρις ότου επιτευχθεί το επιθυμητό αποτέλεσμα.

5.2. Περιγραφή δημοφιλών μοντέλων μηχανικής μάθησης

Το πρόβλημα της βελτιστοποίησης στοχικών συναρτήσεων στα πλαίσια των υδατικών πόρων εντάσσεται στην κατηγορία της επιβλεπόμενης μάθησης και για την αντιμετώπιση του μπορεί να χρησιμοποιηθεί οποιοδήποτε μοντέλο παλινδρόμησης (regression model), το οποίο θα έχει πρακτικά τον ρόλο του υποκατάστατου μοντέλου. Ακολουθεί μια περιγραφή των εξής μοντέλων μηχανικής μάθησης, τα οποία μπορούν να χρησιμοποιηθούν σε προβλήματα παλινδρόμησης:

- Cubic Radial Basis Function with Linear Polynomial Tail
- Polynomial Regression

- Support Vector Machines
- k – Nearest Neighbors
- Decision Trees
- Gaussian Processes

5.2.1. Cubic Radial Basis Function with Linear Polynomial Tail

Τα Radial Basis Functions (RBFs) είναι πολύ δημοφιλείς τεχνικές που έχουν εδραιωθεί στην επιστημονική κοινότητα σαν μοντέλα μηχανικής μάθησης, λόγω των ικανοποιητικών επιδόσεων που εξασφαλίζουν. Πληθώρα δημοσιεύσεων έχουν πραγματοποιηθεί μέσα στα χρόνια, στις οποίες γίνεται χρήση κάποιου RBF είτε σαν υποκατάστατο μοντέλο σε ένα πρόβλημα βελτιστοποίησης είτε γενικότερα σαν τεχνική αυτοεκμάθησης μηχανών και αποδεικνύεται σε αυτές η αξία της συγκεκριμένης τεχνικής. Εδώ γίνεται ανάλυση του Cubic RBF with Linear Polynomial Tail, το οποίο αποτελεί το μεταμοντέλο που χρησιμοποιεί και ο SEEAS, όπως αναφέρθηκε στο προηγούμενο κεφάλαιο.

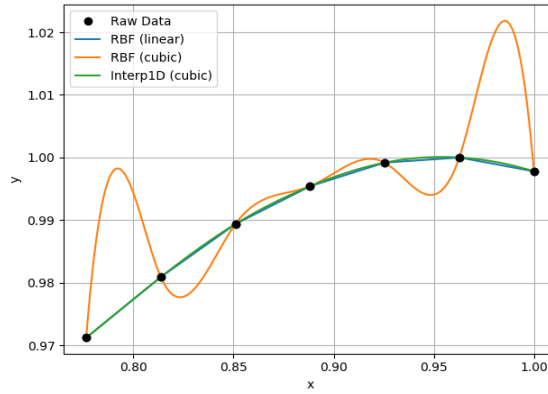
Έστω ένα σύνολο από, N_s στο πλήθος, σημεία $x \in \mathbb{R}^n$ με πραγματική τιμή αντικειμενικής συνάρτησης y . Είναι, λοιπόν, γνωστά τα ζεύγη τιμών (x_i, y_i) με $i = 1, 2, \dots, N_s$. Η προβλεπόμενη τιμή που επιστρέφει το συγκεκριμένο μεταμοντέλο ως απόκριση για ένα τυχαίο σημείο x δίνεται από τη σχέση:

$$s(x) = \sum_{i=1}^{N_s} \lambda_i \varphi(\|x - x_i\|) + p(x) \quad (5.1)$$

όπου τα $\lambda_i \in \mathbb{R}$ και φ είναι μια συνάρτηση της μορφής $\varphi(r) = r^3$. Ο συμβολισμός $\| \cdot \|$ υποδηλώνει την Ευκλείδεια απόσταση και το $p(x)$ είναι ένα πολυώνυμο της μορφής $p(x) = b^T x + a$ με $b = [b_1, b_2, \dots, b_n]^T$ και $a \in \mathbb{R}$. Για να θεωρηθεί γνωστή η σχέση (5.1) και να μπορεί το μοντέλο να κάνει προβλέψεις πρέπει να βρεθούν οι συντελεστές λ_i , το a και το διάνυσμα b . Αυτά προκύπτουν από επίλυση του γραμμικού συστήματος:

$$\begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ c \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \quad (5.2)$$

όπου Φ ένα $N_s \times N_s$ μητρώο με στοιχεία $\varphi_{ij} = \varphi(\|x_i - x_j\|)$, το P είναι ένα μητρώο $N_s \times (n+1)$ με την i -οστή γραμμή του να είναι η $(1, x_i)$, το $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_{N_s}]$, το $c = [b_1, b_2, \dots, b_n, a]^T$ και το $y = [y_1, y_2, \dots, y_{N_s}]^T$.



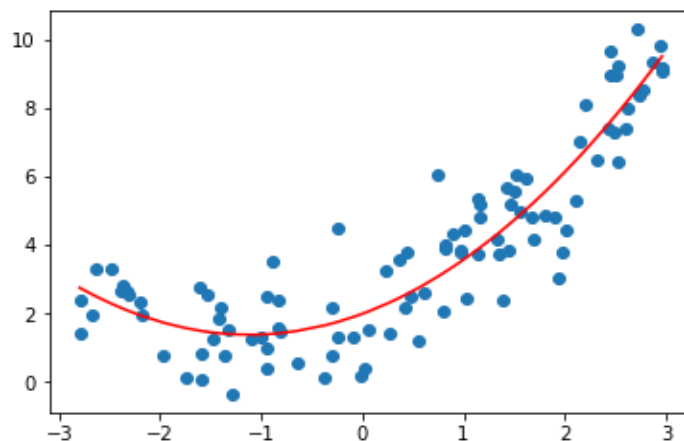
Σχήμα 5-1: Cubic RBF (Πηγή: www.stackoverflow.com)

5.2.2. Polynomial Regression

Η πολυωνυμική παλινδρόμηση είναι ένα μοντέλο μηχανικής μάθησης, το οποίο μπορεί να θεωρηθεί επέκταση της μεθόδου γραμμικής παλινδρόμησης (linear regression), καθώς υπάρχει γραμμική συσχέτιση μεταξύ των συντελεστών του πολυωνύμου. Αποτελεί μια πολύ συχνή και απλή τακτική για την παραγωγή μοντέλου, όταν δεν υπάρχει γραμμική συσχέτιση μεταξύ των ανεξάρτητων μεταβλητών και της εξαρτημένης. Σε ένα τυπικό πρόβλημα, το οποίο αναπαρίσταται από μια ανεξάρτητη μεταβλητή x και την αντίστοιχη εξαρτημένη μεταβλητή y , με ένα γνωστό πλήθος δεδομένων N με ζεύγη τιμών (x_i, y_i) για $i = 1, 2, \dots, N$, η προβλεπόμενη τιμή που επιστρέφει η πολυωνυμική παλινδρόμηση με πολυώνυμο n βαθμού ως απόκριση για το τυχαίο σημείο x δίνεται από τη σχέση:

$$s(x) = b_0 + b_1x + b_2x^2 + \dots + b_nx^n \quad (5.3)$$

όπου τα b_i για $i = 0, 1, 2, \dots, n$ πρέπει να βρεθούν με τη βοήθεια του συνόλου των δεδομένων για να γίνει η καλύτερη δυνατή προσαρμογή του μεταμοντέλου πάνω σε αυτά. Η ίδια θεωρία επεκτείνεται και για περισσότερες από μια ανεξάρτητες μεταβλητές.



Σχήμα 5-2: Polynomial Regression (Πηγή: www.developpaper.com)

5.2.3. Support Vector Machines (SVMs)

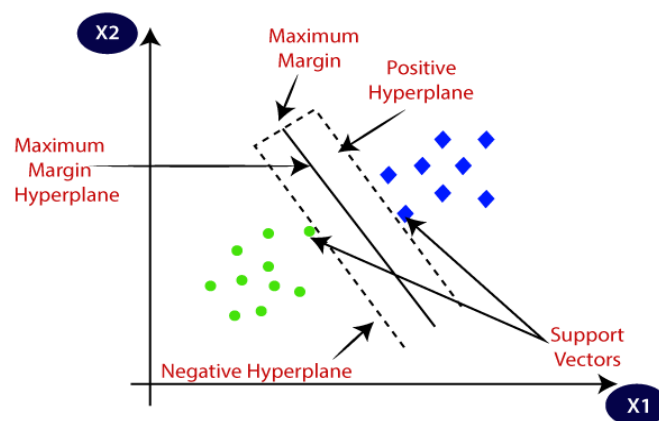
Όπως είναι γνωστό από τη θεωρία γραμμικής παλινδρόμησης, για δεδομένα ζεύγη τιμών (x_i, y_i) πλήθους N , η μέθοδος αυτή προσπαθεί να δημιουργήσει μια γραμμική τάση στην οποία για κάθε x_i με πραγματική τιμή y_i , υπολογίζεται ένα y'_i . Αυτή η γραμμική τάση, που αποτελείται από τα ζεύγη (x_i, y'_i) , δημιουργείται με στόχο την ελαχιστοποίηση του σφάλματος, σύμφωνα με τη μέθοδο των ελαχίστων τετραγώνων, η οποία δίνεται από τη σχέση:

$$\sum_{i=1}^N (y_i - y'_i)^2 \rightarrow \min \quad (5.4)$$

Σε αντίθεση με τη μέθοδο γραμμικής παλινδρόμησης, στην οποία δεν γίνεται επιτρεπτό κανένα σφάλμα, στην τεχνική των Support Vector Machines επιτρέπονται τα σφάλματα, εφόσον αυτά βρίσκονται εντός μιας επιφάνειας (σωλήνα) που περικλείει σε κάποια απόσταση την αντίστοιχη γραμμική τάση που θα υπολογίσει η μέθοδος. Αν υποθέσουμε ένα σύνολο N σημείων (x_i, y_i) στον R^2 και την τελική γραμμική τάση, την οποία θα υπολογίσει το SVM, παράλληλα στην γραμμική αυτή και από τις δύο μεριές σε απόσταση ϵ , δημιουργούνται δύο ακόμα γραμμές, οι οποίες σχηματίζουν τον σωλήνα. Ο σωλήνας στη βιβλιογραφία αναφέρεται ως ϵ -Insensitive Tube και όλα τα σφάλματα εντός αυτού γίνονται αποδεκτά από το μοντέλο. Έτσι, σημεία τα οποία να μην δεν βρίσκονται πάνω στη γραμμική τάση, αλλά βρίσκονται εντός του σωλήνα δεν δημιουργούν σφάλμα και αγνοούνται από το μοντέλο. Αντιθέτως, σημεία (x_i, y_i) , τα οποία βρίσκονται εκτός του σωλήνα είναι καίριας σημασίας για το SVM και για το λόγο αυτό υπολογίζονται οι κατακόρυφες αποστάσεις από τα σημεία αυτά μέχρι τα όρια του σωλήνα. Οι αποστάσεις αυτές συμβολίζονται με ξ_i^* αν τα σημεία είναι πάνω από τον σωλήνα ή με ξ_i , διαφορετικά. Τα ξ_i, ξ_i^* αποκαλούνται slack variables. Το μοντέλο, λοιπόν, προσπαθεί να προσδιορίσει αυτή την κατάλληλη γραμμική τάση, μέσω μιας σχέσης, η οποία προσπαθεί να ελαχιστοποιήσει τον ακόλουθο όρο:

$$\frac{1}{2} \times \| w \|^2 + c \times \sum_{i=1}^m (\xi_i + \xi_i^*) \rightarrow \min \quad (5.5)$$

Η μέθοδος ονομάζεται Support Vector Machine, γιατί τα σημεία (x_i, y_i) που ορίζουν τις αποστάσεις ξ_i και ξ_i^* είναι διανύσματα που έχουν ως αρχή την αρχή του συστήματος των αξόνων O και στην ουσία «αποστηρίζουν» τη δομή και τη μορφή του σωλήνα.



Σχήμα 5-3: Support Vector Machine στον R^2 (Πηγή: www.javatpoint.com)

Η θεωρία που αναλύθηκε αφορά μια κατηγορία των SVM, που είναι τα Linear Support Vector Regression Models και επεκτείνεται και σε μεγαλύτερες διαστάσεις του προβλήματος. Πέρα από αυτή την κατηγορία, υπάρχουν και τα Non - Linear Support Vector Regression Models στα οποία, όπως γίνεται αντιληπτό, η γραμμική τάση που υπολογίζει το μοντέλο δεν είναι μια ευθεία γραμμή. Επίσης, αξίζει να επισημανθεί ότι η θεωρία των SVM μπορεί να χρησιμοποιηθεί για την επίλυση τόσο προβλημάτων παλινδρόμησης, όσο και προβλημάτων κατηγοριοποίησης (classification problems).

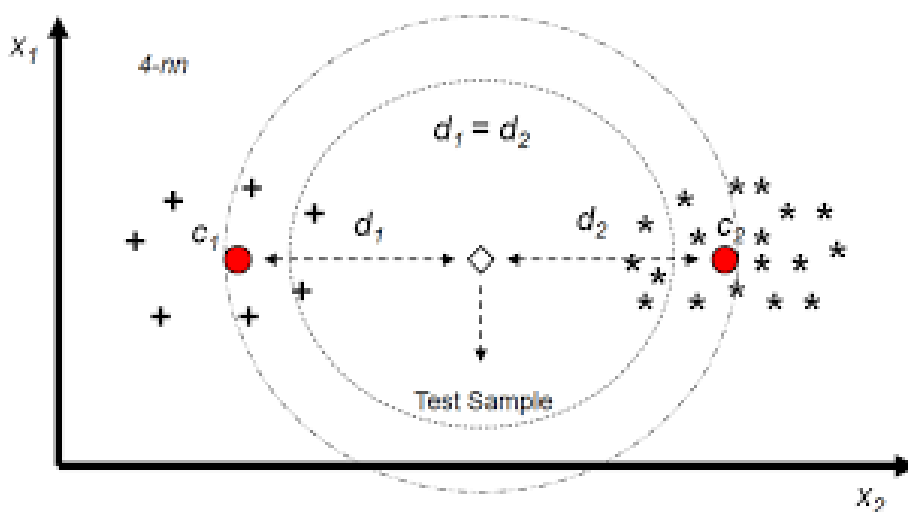
5.2.4. k – Nearest Neighbors

Το k – Nearest Neighbors (k-NN) είναι μια τεχνική αυτοεκμάθησης μηχανών, που μπορεί να εφαρμοστεί όχι μόνο σε προβλήματα παλινδρόμησης, αλλά και σε προβλήματα κατηγοριοποίησης. Έστω ένα πρόβλημα παλινδρόμησης στο οποίο υπάρχουν δύο ανεξάρτητες μεταβλητές x_1 , x_2 και μία εξαρτημένη συνεχής μεταβλητή y . Από το σύνολο των δεδομένων, πλήθους N , είναι γνωστά τα σημεία (x_1, x_2, y) . Για να κάνει τις προβλέψεις του ο αλγόριθμος, λειτουργεί ως εξής:

Βήμα 1^ο : Επιλέγεται ο αριθμός των γειτόνων k που θα χρησιμοποιηθούν. Συνηθισμένη τιμή για το k είναι το $k = 5$.

Βήμα 2^ο : Για το νέο σημείο που εισάγεται και για το οποίο ο αλγόριθμος θέλουμε να κάνει την πρόβλεψη, βρίσκονται οι αποστάσεις του σημείου αυτού από τους k πιο κοντινούς γείτονες του. Για την εύρεση της απόστασης μπορεί να χρησιμοποιηθεί οποιοσδήποτε γνωστός τύπος απόστασης, αλλά συνήθως επιλέγεται η Ευκλείδεια απόσταση.

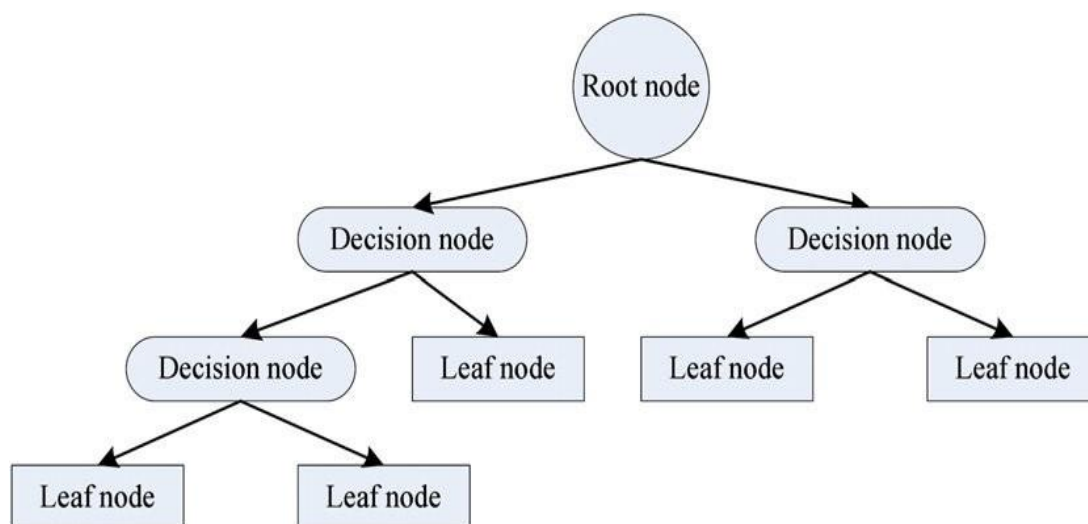
Βήμα 3^ο : Ο αλγόριθμος ελέγχει την τιμή που έχουν οι k πιο κοντινοί γείτονες στο σημείο και μέσω αυτών υπολογίζεται η τελική προβλεπόμενη τιμή για το νέο σημείο.



Σχήμα 5-4: k-Nearest Neighbors (Πηγή: www.researchgate.net)

5.2.5. Decision Trees

Τα δέντρα απόφασης (Decision Trees) είναι αναμφίβολα μια από τις πιο δημοφιλείς μεθόδους μηχανικής μάθησης, καθώς η θεωρία τους βρίσκει χρήση σε πολλούς διαφορετικούς κλάδους της επιστήμης. Μπορούν να χρησιμοποιηθούν τόσο σε προβλήματα παλινδρόμησης, όσο και σε προβλήματα κατηγοριοποίησης. Τα δέντρα απόφασης είναι εργαλεία που λειτουργούν με διαδοχικές εντολές «if...then» και προκαλούν μια αναδρομική διχοτόμηση του χώρου γεγονότων. Σε κάθε δέντρο υπάρχει ένας κόμβος, που ονομάζεται ρίζα (root node) και ο οποίος δεν έχει καμιά εισερχόμενη ακμή. Αυτός υποδεικνύει την κατεύθυνση από την οποία ξεκινάει το δέντρο. Οι υπόλοιποι κόμβοι του δέντρου έχουν μια εισερχόμενη ακμή. Κάποιοι κόμβοι του δέντρου έχουν και εξερχόμενη ακμή, ενώ άλλοι όχι. Οι πρώτοι ονομάζονται «εσωτερικοί κόμβοι» ή αλλιώς «κόμβοι απόφασης» (decision nodes), ενώ οι δεύτεροι ονομάζονται «φύλλα» (leaf nodes). Κάθε εσωτερικός κόμβος του δέντρου χρησιμοποιεί μια διακριτή συνάρτηση των αξιών των χαρακτηριστικών εισόδου και με αυτήν χωρίζει το συνολικό χώρο γεγονότων σε υποχώρους. Τα φύλλα, τα οποία αποτελούν, πρακτικά, τα τελικά παρακλάδια του δέντρου, αντιστοιχίζονται με μια κατηγορία, η οποία αναπαριστά την κατάλληλη αξία της μεταβλητής στόχου. Ένα δεδομένο, το οποίο εισέρχεται στο δέντρο, ξεκινάει την πορεία του από τη ρίζα του δέντρου και διασπάται από κόμβο σε κόμβο με βάση τη βέλτιστη ιδιότητα του κάθε κόμβου, η οποία υπολογίζεται σύμφωνα με κάποιο συγκεκριμένο κριτήριο. Καθώς τα δεδομένα προχωρούν σε επόμενους εσωτερικούς κόμβους υπάρχει ολοένα και μεγαλύτερη ομοιογένεια μεταξύ τους και μόλις φτάνουν στα φύλλα του δέντρου, προκύπτουν οι τελικές κλάσεις του προβλήματος. Μια από τις πιο γνωστές μεθόδους κατασκευής δέντρων αποφάσεων, είναι το Classification And Regression Tree (CART).

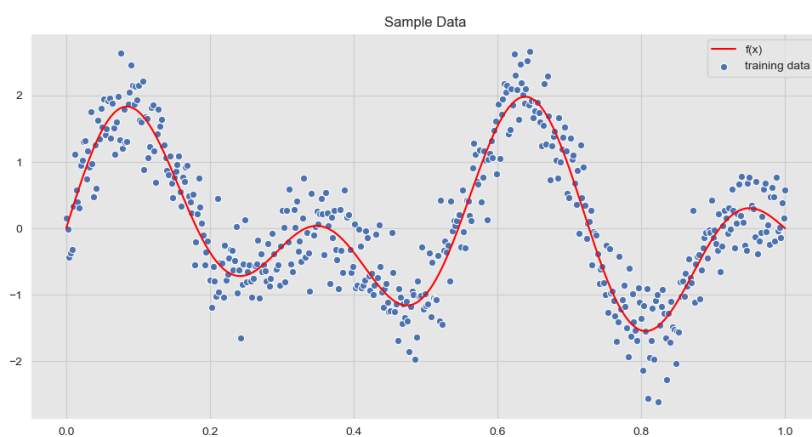


Σχήμα 5-5: Τυπική δομή δέντρου απόφασης (Πηγή: www.researchgate.net)

5.2.6. Gaussian Processes

Το Gaussian Process ή Kriging όπως αλλιώς συναντάται στη βιβλιογραφία, είναι ένα μοντέλο μηχανικής μάθησης το οποίο χρησιμοποιείται πολύ συχνά ως υποκατάστατο μοντέλο σε προβλήματα βελτιστοποίησης, καθώς έχει αποδεδειγμένη ικανοποιητική συμπεριφορά στην πλειονότητα των περιπτώσεων. Παρουσιάζει πολλές ομοιότητες με τα Radial Basis Functions και ειδικότερα με μια διαφορετική κατηγορία από τα Cubic RBFs που αναλύθηκαν παραπάνω, τα λεγόμενα Gaussian RBFs. Η basis function του Kriging δίνεται από τη σχέση:

$$\psi^{(i)} = e^{-\sum_{j=1}^k \theta_j |x_j^{(i)} - x_j|^{\rho_j}} \quad (5.6)$$



Σχήμα 5-6: Gaussian Process (Πηγή: <https://juanitorduz.github.io/>)

Κεφάλαιο 6^ο

6 Ο αλγόριθμος AMSEEAS

Ένα σημαντικό τμήμα της παρούσας εργασίας αφιερώθηκε στην προσπάθεια για ανάπτυξη ενός νέου πρωτότυπου αλγορίθμου, βασισμένο πάνω στον εξελικτικό αλγόριθμο ανόπτωσης – απλόκου εμπλουτισμένο με υποκατάστατα μοντέλα (SEEAS), που θα έχει ως στόχο την περαιτέρω βελτίωση της επίδοσής του. Ο SEEAS έχει αδιαμφισβήτητη μεγάλη επιτυχία, καθώς, όπως φαίνεται και στην αντίστοιχη εργασία των δημιουργών του, ο αλγόριθμος συγκρίθηκε με πολλούς άλλους δημοφιλείς και εδραιωμένους κώδικες της διεθνούς κοινότητας και τα αποτελέσματα της σύγκρισης αυτής ήταν πολύ ευχάριστα. Η σύγκριση πραγματοποιήθηκε όχι μόνο πάνω σε πολύπλοκες μαθηματικές συναρτήσεις με πολλαπλά τοπικά ακρότατα και ανώμαλη επιφάνεια απόκρισης, αλλά και σε πραγματικά συχνά προβλήματα του τομέα των υδατικών πόρων. Τα αποτελέσματα της σύγκρισης επιβεβαίωσαν την ανωτερότητα του SEEAS και τις μεγάλες προοπτικές του. Ο ήδη υψηλά ανεβασμένος «πήχης» που έθεσε ο SEEAS, αποτέλεσε στην ουσία και το κίνητρο να πραγματοποιηθεί στην εργασία αυτή μια προσπάθεια για την περαιτέρω εξέλιξη του. Και πράγματι αυτό κατέστη εφικτό, όπως αποδεικνύουν και τα αποτελέσματα των συγκρίσεων, που θα παρουσιαστούν στη συνέχεια. Στο κεφάλαιο αυτό γίνεται η παρουσίαση ενός νέου πρωτότυπου κώδικα, βασισμένου πάνω στον SEEAS, του λεγόμενου Adaptive Multi-Surrogate Enhanced Evolutionary Annealing Simplex algorithm (AMSEEAS).

6.1. Γενική επισκόπηση του αλγορίθμου και πρωτότυπα σημεία

Όπως αναφέρθηκε στο [Υποκεφάλαιο 4.2.4](#), στα πλαίσια της εργασίας πραγματοποιήθηκε μεταφορά του SEEAS από γλώσσα προγραμματισμού MATLAB σε Python. Κατά τη μεταφορά αυτή, εξετάζονταν διαρκώς σενάρια για το πώς θα μπορούσε να πραγματοποιηθεί στη συνέχεια κάποια δραστική αλλαγή στον πυρήνα του κώδικα, η οποία να επιφέρει ουσιώδη διαφορά στα αποτελέσματα. Κατά τη διαδικασία αυτή της αναζήτησης, προέκυψαν οι εξής δύο απορίες:

- 1) Για ποιο λόγο ο EAS ενισχύεται συγκεκριμένα από το Cubic RBF with Linear Polynomial Tail σαν μεταμοντέλο και όχι από κάποιο άλλο από τα δημοφιλή της διεθνούς κοινότητας;
- 2) Για ποιο λόγο ο EAS ενισχύεται μέσω ενός μονάχα υποκατάστατου μοντέλου και όχι περισσότερων;

Τα δύο αυτά ζητήματα οδήγησαν στην αναζήτηση για προσαρμογή διαφορετικών μεταμοντέλων πάνω στον EAS και στην προσπάθεια για ανάπτυξη κάποιου είδους υβριδικού συστήματος, το οποίο θα αποτελείται από πολλά υποκατάστατα μοντέλα.

Όπως αναλύθηκε και στο προηγούμενο κεφάλαιο, με το πέρασμα των χρόνων έχουν αναπτυχθεί πάρα πολλά διαφορετικά μοντέλα μηχανικής μάθησης, κάθε ένα από τα οποία προσπαθεί με μια διαφορετική πατέντα να παράξει «έξυπνα» υπολογιστικά συστήματα, τα οποία θα μπορούν να διδάσκονται αποδοτικά από τα δεδομένα που τους εισάγονται, με στόχο να μπορούν να κάνουν στη συνέχεια ακριβείς προβλέψεις για τις νέες τιμές που δέχεται το σύστημα. Παρόλα αυτά, είναι γνωστό ότι δεν υπάρχει κάποιο μοντέλο που να υπερτερεί των υπολοίπων σε κάθε κατηγορία προβλήματος. Είναι, λοιπόν, απαραίτητο σε κάθε μεμονωμένο πρόβλημα, να εξετάζονται πολλά διαφορετικά μοντέλα αυτοεκμάθησης μηχανών και να επιλέγεται τελικά από τον προγραμματιστή αυτό ή αυτά που φαίνεται να έχουν την καλύτερη ανταπόκριση πάνω στα συγκεκριμένα δεδομένα. Για την παραγωγή της τελικής έκδοσης του AMSEEAS, εξετάστηκαν περιπτώσεις, στις οποίες ενσωματώθηκαν τα εξής μοντέλα μηχανικής μάθησης:

- Decision Trees
- Random Forests
- Extra Trees
- Gradient Boosting
- Ada Boosting
- Gaussian Processes
- Cubic RBF with Linear Polynomial Tail
- Support Vector Machines
- Polynomial Regression
- Artificial Neural Networks
- Naive Bayes
- k – Nearest Neighbors

Για την αξιολόγηση της επίδοσης των διαφορετικών αλγορίθμων που παράγονταν, χρησιμοποιήθηκαν συγκεκριμένες δημοφιλείς συναρτήσεις ελέγχου (test functions), με πολύπλοκες επιφάνειες απόκρισης. Αυτές ονομαστικά είναι οι εξής:

- Sphere Function
- Ackley Function
- Griewank Function
- Zakharov Function
- Rastrigin Function
- Levy Function
- Hozaki Function
- Goldstein-Price Function
- Styblinski-Tang Function
- Michalewisc Function
- Rozenbrock Function

Περισσότερες πληροφορίες σχετικά με τις συναρτήσεις αυτές, δίνονται στο [Παράρτημα Α](#).

Ο AMSEEAS έχει πανομοιότυπη υπολογιστική διαδικασία με αυτή του SEEAS. Γίνεται και εδώ χρήση της τεχνικής του κατερχόμενου απλόκου σε συνδυασμό με τη μέθοδο προσομοιωμένης απόκτησης, με τη ίδια λογική που εισήχθη στον EAS και η όλη προσπάθεια ενισχύεται από υποκατάστατα μοντέλα. Παρόλα αυτά, στον νέο κώδικα υπάρχει μια βασική διαφοροποίηση. Αντί να χρησιμοποιείται αποκλειστικά το Cubic RBF with Polynomial Tail σαν μεταμοντέλο, πλέον αναπτύσσεται ένα υβριδικό σύστημα, στο οποίο συνυπάρχουν πολλά διαφορετικά υποκατάστατα μοντέλα, τα οποία λειτουργούν σαν μια κανονική ομάδα. Αναλυτικότερα, η ιδέα που εισάγεται είναι η εξής:

- Από τη στιγμή που δεν υπάρχει κάποιο μοντέλο μηχανικής μάθησης σαφώς ανώτερο των υπολοίπων και όλα μπορεί να ανταποκρίνονται καλύτερα ή χειρότερα από τα άλλα ανάλογα με το εκάστοτε πρόβλημα βελτιστοποίησης, εισάγοντας πολλά διαφορετικά μοντέλα αυτοεκμάθησης στον ίδιο αλγόριθμο, είναι πιθανό ο αλγόριθμος που θα προκύψει να ανταποκρίνεται καλύτερα στο σύνολο των προβλημάτων, τα οποία καλείται να αντιμετωπίσει. Για να μπορέσει αυτή η ιδέα να αποδειχτεί όντως αποτελεσματική, απαιτείται να βρεθεί ένας αποδοτικός τρόπος συνύπαρξης των πολλών μοντέλων στον ίδιο αλγόριθμο. Αυτό που τίθεται ως στόχος είναι να υπάρξει αισθητή βελτίωση στην ταχύτητα σύγκλισης του αλγορίθμου (δηλαδή η σύγκλιση να επιτυγχάνεται με τον ελάχιστο δυνατό αριθμό υπολογισμών της «ακριβής» από άποψη χρόνου και υπολογισμών στοχικής

συνάρτησης) και ταυτόχρονα να μην συμβεί επιβάρυνση του αρχικού SEEAS από άποψη υπολογιστικού φόρτου.

Από τις πολλές διαφορετικές εκδοχές αλγορίθμων που κατασκευάστηκαν στα πλαίσια αναζήτησης του ποια/πόσα μοντέλα μηχανικής μάθησης θα ενσωματωθούν στον τελικό αλγόριθμο και ποιος θα είναι ο τρόπος με τον οποίο αυτά θα συνυπάρχουν και θα συνεργάζονται αποτελεσματικά, η τελική εκδοχή που φάνηκε να είναι ανώτερη όλων των υπολοίπων παρουσιάζεται ακολούθως.

6.1.1. Περιγραφή των πρωτότυπων σημείων του αλγορίθμου

Στον αλγόριθμο ενσωματώνονται συνολικά 6 διαφορετικά μοντέλα μηχανικής μάθησης. Αυτά είναι τα εξής:

- Polynomial Regression
- Support Vector Machine
- Cubic RBF with Linear Polynomial Tail σε συνδυασμό με την Acquisition Function, όπως αναλύθηκε στον SEEAS
- k – Nearest Neighbors
- Decision Tree
- Gaussian Process

Για την αποτελεσματική συνεργασία των μοντέλων, εισάγεται η ιδέα επιβολής ενός είδους ποινής (penalty) σε αυτά, εφόσον δεν έχουν την επιθυμητή συμπεριφορά. Υπενθυμίζεται ότι, όπως είναι στημένος ο SEEAS, η προσέγγιση του ολικού ελαχίστου της αντικειμενικής συνάρτησης γίνεται με χρήση της έννοιας των γενιών λύσεων, η οποία εφαρμόστηκε αρχικά στους εξελικτικούς αλγορίθμους. Κατά την ολοκλήρωση της κάθε μίας επανάληψης του αλγορίθμου, προκύπτει η επόμενη γενιά, που αποτελείται από έναν συγκεκριμένο πληθυσμό λύσεων. Κάθε επόμενη γενιά είναι κατά μέσο όρο καλύτερη της προηγούμενης της και έτσι προχωράει η διαδικασία της σύγκλισης.

Για την συνύπαρξη, λοιπόν, των 6 αυτών διαφορετικών μοντέλων μηχανικής μάθησης ακολουθήθηκε η εξής λογική:

- Το μοντέλο Polynomial Regression, αν δεν δεχτεί καμία τιμωρία, θα ενεργοποιηθεί στην 1^η επανάληψη του αλγορίθμου, έπειτα στην 7^η, στην 13^η, στην 19^η κοκ. Το μοντέλο Support Vector Machine, αν δεν δεχτεί καμία τιμωρία, θα ενεργοποιηθεί στην 2^η, 8^η, 14^η, 20^η κοκ. επανάληψη του κώδικα. Με αντίστοιχο τρόπο, λοιπόν, το RBF θα ενεργοποιηθεί στην 3^η, 9^η, 15^η, 21^η κοκ. επανάληψη, το k-NN στην 4^η,

10^η, 16^η, 22^η κοκ., το Decision Tree στην 5^η, 11^η, 17^η, 23^η κοκ. και τέλος, το Gaussian Process στην 6^η, 12^η, 18^η, 24^η κοκ. επανάληψη του αλγορίθμου. Αυτό βέβαια, δεν είναι απόλυτο, καθώς υπάρχουν και οι ποινές που μπορεί να επιβληθούν στα μοντέλα, όπως εξηγείται παρακάτω.

Από τη στιγμή που σε κάθε επανάληψη του αλγορίθμου ενεργοποιείται ένα μόνο μοντέλο από τα παραπάνω και όχι πολλά μαζί, γίνεται αντιληπτό ότι ο κώδικας δεν επιβαρύνεται σχεδόν καθόλου από άποψη υπολογιστικού φόρτου. Παρόλα αυτά, η ιδέα του υβριδικού αυτού συστήματος έγκειται στο ότι είναι επιθυμητό τα μοντέλα, τα οποία δεν κάνουν καλές προσαρμογές σε κάποια προβλήματα βελτιστοποίησης, να «κάνουν πέρα» για να μην ξοδεύονται άσκοπα επαναλήψεις και υπολογισμοί της αντικειμενικής συνάρτησης χωρίς βελτίωση στη σύγκλιση, και να επιτρέπουν στα μοντέλα τα οποία κάνουν καλύτερες προβλέψεις να αναλάβουν τα ηνία και να καθοδηγήσουν εκείνα την διαδικασία της σύγκλισης. Για να υλοποιηθεί, λοιπόν, το σκεπτικό αυτό, εισάγεται στον αλγόριθμο η ιδέα επιβολής τιμωρίας στα μοντέλα που δεν κάνουν καλές προβλέψεις. Η ακριβής έννοια της τιμωρίας που επιβάλλεται στα μοντέλα εξηγείται στο ακόλουθο παράδειγμα:

Το μοντέλο Polynomial Regression θα ενεργοποιηθεί σίγουρα στην 1^η επανάληψη του αλγορίθμου. Θα κάνει προσαρμογή πάνω στα σημεία που υπάρχουν εκείνη τη στιγμή στο εξωτερικό αρχείο και ένας αλγόριθμος βελτιστοποίησης (ο EAS στην έκδοση του SEEAS) θα κάνει ελαχιστοποίηση πάνω στο εκπαιδευμένο, πλέον, Polynomial Regression. Από την διαδικασία της ελαχιστοποίησης αυτής, θα προκύψει ένα σημείο (το ολικό ελάχιστο της καμπύλης του Polynomial Regression, πρακτικά). Το σημείο αυτό λοιπόν, είναι η προβλεπόμενη τιμή του Polynomial Regression και άρα μια νέα λύση του προβλήματος. Ακολουθεί, έτσι ο υπολογισμός της πραγματικής «χρονοβόρας» στοχικής συνάρτησης στο σημείο αυτό και ανάλογα με το αν η λύση αυτή είναι καλή ή όχι, εισέρχεται στον πληθυσμό ή απορρίπτεται, αντίστοιχα. Εδώ δημιουργούνται δύο διαφορετικά μονοπάτια:

- 1^η περίπτωση: Αν το σημείο που προέκυψε από την πρόβλεψη είναι πράγματι καλό και εισέλθει στον πληθυσμό των λύσεων, τότε το μοντέλο Polynomial Regression δεν δέχεται τιμωρία και θα ενεργοποιηθεί ξανά, όπως είναι προγραμματισμένο στην 7^η επανάληψη του αλγορίθμου.
- 2^η περίπτωση: Αν το σημείο που προέκυψε από την πρόβλεψη δεν είναι καλό και απορριφθεί από τον πληθυσμό, τότε το συγκεκριμένο μοντέλο δέχεται τιμωρία. Η τιμωρία που του επιβάλλεται είναι να μην ενεργοποιηθεί στην 7^η επανάληψη, όπως είναι κανονικά προγραμματισμένο, ούτε στην 13^η κτλ., δηλαδή στην ουσία το μοντέλο παραμένει εκτός του αλγορίθμου για συγκεκριμένο αριθμό επαναληπτικών κύκλων που ορίζονται από τον προγραμματιστή (στην τελική έκδοση, η τιμωρία που επιβάλλεται στο κάθε μοντέλο είναι αποχή για 10 γύρους ενεργοποίησής του από την επαναληπτική διαδικασία).

Σε περίπτωση όπου, όλα τα μεταμοντέλα βρίσκονται σε κατάσταση τιμωρίας σε κάποια επανάληψη του αλγορίθμου, τότε αφαιρείται διαδοχικά ένας-ένας γύρος τιμωρίας από το κάθε μοντέλο, μέχρι να βρεθεί το πρώτο χωρίς τιμωρία, προκειμένου να ενεργοποιηθεί.

6.2. Αναλυτική περιγραφή της υπολογιστικής διαδικασίας του AMSEEAS

Έστω ένα πρόβλημα βελτιστοποίησης, το οποίο περιγράφεται από n συνεχείς μεταβλητές ελέγχου. Όπως είναι στημένος ο αλγόριθμος, το πρόβλημα αντιμετωπίζεται σαν πρόβλημα ελαχιστοποίησης της στοχικής συνάρτησης που το περιγράφει. Η στοχική συνάρτηση είναι βαθμωτή, μη γραμμική και αντιστοιχεί πρακτικά σε κάποιο μέτρο επίδοσης του μοντέλου προσομοίωσης που περιγράφει το πρόβλημα. Για να ξεκινήσει ο αλγόριθμος, ο χρήστης απαιτείται να εισάγει τα ακόλουθα στοιχεία, σαν παραμέτρους εισόδου:

- το πλήθος των μεταβλητών ελέγχου του προβλήματος n
- το σταθερό μέγεθος του πληθυσμού m , με $m \geq n+1$
- ένα διάνυσμα x_{\min} που αντιστοιχεί στα κάτω όρια τιμών των μεταβλητών ελέγχου
- ένα διάνυσμα x_{\max} που αντιστοιχεί στα άνω όρια τιμών των μεταβλητών ελέγχου
- την αντικειμενική συνάρτηση που περιγράφει το πρόβλημα
- το μέγιστο επιτρεπτό πλήθος υπολογισμών της αντικειμενικής συνάρτησης
- μια τιμή για το όριο σύγκλισης, το οποίο αν επιτευχθεί, ο αλγόριθμος οδηγείται σε πρόωρο τερματισμό
- την παράμετρο του αυτορρυθμιζόμενου χρονοδιαγράμματος ανόπτησης β
- την πιθανότητα πραγματοποίησης μετάλλαξης p_m
- το μέγιστο επιτρεπτό πλήθος διαδοχικών αναρριχήσεων

Ο AMSEEAS χρησιμοποιεί δύο αρχεία, το P και το A. Το P αντιστοιχεί στον πληθυσμό λύσεων και το A αντιστοιχεί στο εξωτερικό αρχείο, κατά τα γνωστά. Ο αρχικός πληθυσμός λύσεων της 1^{ης} γενιάς παράγεται με χρήση της στατιστικής μεθόδου Latin Hypercube Sampling (LHS).

Ο AMSEEAS χρησιμοποιεί ένα αυτορρυθμιζόμενο χρονοδιάγραμμα ανόπτησης, το οποίο εφαρμόζεται μέσω της μεταβλητής T, που αντιστοιχεί στην εκάστοτε θερμοκρασία του συστήματος και η οποία ρυθμίζεται μέσω της καλύτερης και της χειρότερης κορυφής του απλόκου που σχηματίζεται, μέσω του πληθυσμού P και μέσω ενός PI, ο οποίος δίνεται από τη σχέση (4.4).

Στο σημείο αυτό, γίνεται μια αναλυτική περιγραφή των βημάτων που εκτελεί ο AMSEEAS σε μια τυπική επανάληψη του:

Βήμα 1^ο : Ένα μεταμοντέλο από τα 6 συνολικά που υπάρχουν ενσωματωμένα στον αλγόριθμο, εφόσον δεν βρίσκεται σε κατάσταση τιμωρίας, κάνει προσαρμογή πάνω σε όλα τα σημεία που υπάρχουν τη δεδομένη στιγμή στο εξωτερικό αρχείο A.

Βήμα 2^ο : Ένας αλγόριθμος βελτιστοποίησης (θα γίνει στη συνέχεια ανάλυση για το ποιος θα είναι αυτός ο αλγόριθμος) εκτελεί ελαχιστοποίηση πάνω στο προσαρμοσμένο μεταμοντέλο του βήματος 1, μέσω της οποίας προκύπτει μια νέα λύση του προβλήματος, το σημείο x_p . Στο σημείο αυτό γίνεται υπολογισμός της πραγματικής αντικειμενικής συνάρτησης, δηλαδή υπολογίζεται το $f(x_p)$ και εφόσον $f(x_p) < f(x_{\text{worst}})$, όπου x_{worst} το σημείο του πληθυσμού με τη μεγαλύτερη τιμή στοχικής συνάρτησης, τότε το x_p αντικαθιστά το x_{worst} και εισέρχεται στον πληθυσμό. Αν το x_p εισέλθει στον πληθυσμό, τότε στο συγκεκριμένο μεταμοντέλο δεν επιβάλλεται καμία τιμωρία, διαφορετικά αν $f(x_p) > f(x_{\text{worst}})$, το μεταμοντέλο παραμένει εκτός υπολογιστικής διαδικασίας για ένα διάστημα 10 δικών του επαναληπτικών κύκλων, όπως αναλύθηκε νωρίτερα. Στο υπόλοιπο τμήμα του επαναληπτικού κύκλου, κατά την εκτέλεση των κινήσεων του κατερχόμενου απλόκου, ο AMSEAS ακολουθεί την λογική του SEEAS και χρησιμοποιεί σαν υποκατάστατο μοντέλο κανονικά το Cubic RBF with Linear Polynomial Tail σε συνδυασμό με την Acquisition Function.

Βήμα 3^ο : Από τον συνολικό πληθυσμό επιλέγονται τυχαία $n+1$ σημεία, τα οποία σχηματίζουν το άπλοκο $S = \{ s_1, s_2, \dots, s_{n+1} \}$. Οι κορυφές του απλόκου τοποθετούνται σε αύξουσα σειρά με την κορυφή s_1 να αντιστοιχεί στο σημείο που έχει τη χαμηλότερη (άρα καλύτερη) τιμή στοχικής συνάρτησης, ενώ το s_{n+1} να έχει τη μεγαλύτερη (άρα χειρότερη).

Βήμα 4^ο : Από το υποσύνολο του απλόκου $\{ s_2, \dots, s_{n+1} \}$, το οποίο θα περιλαμβάνει όλες τις κορυφές του απλόκου εκτός της καλύτερης s_1 , βρίσκεται η κορυφή x_w που θα οδηγηθεί προς αντικατάσταση. Η κορυφή αυτή βρίσκεται σύμφωνα με το κριτήριο Metropolis και θα είναι η κορυφή εκείνη που μεγιστοποιεί τη συνάρτηση της σχέσης (4.6).

Βήμα 5^ο : Το άπλοκο εκτελεί ανάκλαση ως προς την κορυφή x_w και έτσι προκύπτουν νέες υποψήφιες λύσεις, σύμφωνα με τη σχέση (4.7). Από όλα τα υποψηφία σημεία που προκύπτουν από τις ανακλάσεις, επιλέγεται εκείνο που αντιστοιχεί σε χαμηλότερη τιμή της Acquisition Function ως το καλύτερο και αυτό θεωρείται η λύση που προκύπτει από την ανάκλαση του απλόκου. Στο σημείο αυτό x_r γίνεται και υπολογισμός, επομένως, της πραγματικής στοχικής συνάρτησης, υπολογίζεται δηλαδή το $f(x_r)$ και ταυτόχρονα το x_r εισέρχεται και στο εξωτερικό αρχείο.

Βήμα 6^ο : Αν $f(x_r) < f(x_w)$, τότε το x_w αντικαθίσταται από το x_r στον πληθυσμό και ο αλγόριθμος συνεχίζει στα βήματα 7α ή 7β, ανάλογα με το αποτέλεσμα της σύγκρισης $f(x_r) < f(x_1)$. Διαφορετικά, ο αλγόριθμος συνεχίζει στο βήμα 8, όπου θα καθοριστεί αν το σημείο x_r θα γίνει τελικά αποδεκτό ή θα απορριφθεί και θα αναζητηθεί ένας άλλος υποψήφιος.

Βήμα 7α : Αν το σημείο x_r που προέκυψε από την ανάκλαση είναι καλύτερο και από το καλύτερο σημείο του απλόκου x_1 , αν δηλαδή $f(x_r) < f(x_1)$, τότε εκτελούνται βήματα επέκτασης προς την κατεύθυνση $x_r - g$. Από την επέκταση αυτή προκύπτουν συνολικά N_e δοκιμαστικά σημεία, σύμφωνα με τη σχέση (4.10). Η επέκταση συνεχίζεται για

όσο παρατηρείται βελτίωση της Acquisition Function. Το καλύτερο σημείο x_e που προέκυψε από τα συνολικά βήματα επέκτασης εισέρχεται στο εξωτερικό αρχείο και στον πληθυσμό, εφόσον $f(x_e) < f(x_f)$. Στην περίπτωση αυτή ο αλγόριθμος προχωράει στο βήμα 11.

Βήμα 7β : Αν αντίθετα, το σημείο x_f που προέκυψε από την ανάκλαση δεν είναι καλύτερο από το καλύτερο σημείο του απλόκου x_1 , αν δηλαδή $f(x_f) > f(x_1)$, τότε επιχειρείται η εύρεση μιας λύσης στη γειτονιά του x_1 . Έτσι, πραγματοποιούνται βήματα συμπίεσης στη γειτονιά ανάμεσα στο κεντροειδές του απλόκου και στο σημείο της ανάκλασης και παράγονται N_c υποψήφιος λύσεις, σύμφωνα με τη σχέση (4.12). Ο υποψήφιος εκείνος που δίνει τη χαμηλότερη τιμή της Acquisition Function, θα είναι το σημείο x_c , το οποίο εισέρχεται στο εξωτερικό αρχείο σίγουρα και στον πληθυσμό αν $f(x_c) < f(x_f)$. Στην περίπτωση αυτή ο αλγόριθμος προχωράει στο βήμα 11.

Βήμα 8° : Αν το σημείο x_f που προέκυψε από την ανάκλαση είναι χειρότερο από το χειρότερο σημείο του απλόκου x_w , αν δηλαδή $f(x_f) > f(x_w)$ τότε χρησιμοποιείται η συνθήκη (4.13) για να καθοριστεί αν το άπλοκο θα πραγματοποιήσει εσωτερική συμπίεση για τον εντοπισμού κάποιου τοπικού ακροτάτου ή αν θα εκτελεστούν βήματα αναρρίχησης, δηλαδή βήματα επέκτασης του απλόκου προς την κατεύθυνση όπου αυξάνεται η τιμή της αντικειμενικής συνάρτησης. Αν η συνθήκη προκύψει αληθής, τότε ο αλγόριθμος προχωράει στο βήμα 9α, αλλιώς προχωράει στο βήμα 9β.

Βήμα 9α : Πραγματοποιείται μείωση της θερμοκρασίας του συστήματος, σύμφωνα με μια παράμετρο ψ , η οποία δίνεται από τη σχέση (4.14). Το σημείο της ανάκλασης x_f απορρίπτεται και εκτελούνται βήματα εσωτερικής συμπίεσης του απλόκου στη γειτονιά ανάμεσα στο κεντροειδές και στην χειρότερη κορυφή του, σύμφωνα με τη σχέση (4.15). Ο υποψήφιος εκείνος που δίνει τη χαμηλότερη τιμή της Acquisition Function, θα είναι το σημείο x_c , το οποίο εισέρχεται στο εξωτερικό αρχείο σίγουρα και στον πληθυσμό αν $f(x_c) < f(x_w)$, αντικαθιστώντας το x_w . Διαφορετικά, το άπλοκο εκτελεί βήματα συρρίκνωσης προς την κατεύθυνση της καλύτερης κορυφής του απλόκου x_1 , σύμφωνα με τη σχέση (4.16). Εφόσον ενεργοποιηθεί ο μηχανισμός αυτός, παράγονται n καινούριες λύσεις του προβλήματος και αντικαθιστούνται όλες οι κορυφές του απλόκου, εκτός της x_1 , από τα n καινούρια αυτά σημεία στον πληθυσμό.

Βήμα 9β : Το σημείο της ανάκλασης x_f γίνεται αποδεκτό παρόλο που είναι χειρότερο από το x_w και ενεργοποιείται ένας μηχανισμός διαφυγής από τοπικό ακρότατο. Ο μηχανισμός αυτός περιλαμβάνει βήματα αναρρίχησης όπως ονομάζονται, δηλαδή βήματα επέκτασης του απλόκου προς την κατεύθυνση όπου αυξάνεται η τιμή της αντικειμενικής συνάρτησης. Τα βήματα αυτά εκτελούνται σύμφωνα με τη σχέση (4.12) και προκύπτουν νέα σημεία. Αντίστοιχα με τις υπόλοιπες περιπτώσεις, μέσω της Acquisition Function προκύπτει το σημείο x_u . Αν $f(x_u) < f(x_f)$, τότε το x_u εισάγεται στο εξωτερικό αρχείο και αντικαθιστά το x_f και ο αλγόριθμος προχωράει στο βήμα 11. Διαφορετικά, μιας και καμία από τις προηγούμενες ενέργειες δεν είχε αποτέλεσμα, ο αλγόριθμος προχωράει στο βήμα 10 όπου ενεργοποιείται ένας μηχανισμός μετάλλαξης.

Βήμα 10° : Ενεργοποιείται ο μηχανισμός της μετάλλαξης. Στόχος του μηχανισμού αυτού είναι να δημιουργηθεί ένα νέο σημείο εκτός του εύρους της τυπικής απόκλισης γύρω από τη μέση τιμή του τρέχοντος πληθυσμού, δηλαδή το σημείο αυτό να είναι εκτός του εύρους $[\mu_\chi - \sigma_\chi, \mu_\chi + \sigma_\chi]$. Οι συντεταγμένες του νέου αυτού σημείου

προκύπτουν σύμφωνα με τη σχέση (4.17). Έτσι, προκύπτει το νέο σημείο x_m , το οποίο εισέρχεται στο εξωτερικό αρχείο και γίνεται αποδεκτό και στον πληθυσμό, αν $f(x_m) < f(x_r)$. Διαφορετικά, ανάλογα με μια πιθανότητα μετάλλαξης, η οποία εισάγεται από το χρήστη, επιλέγεται είτε το x_m είτε το x_r για να αντικαταστήσει το x_w στον πληθυσμό.

Βήμα 11^ο : Γίνεται εύρεση της καλύτερης λύσης x_b και της χειρότερης λύσης x_{worst} που υπάρχει στον πληθυσμό και επιχειρείται να βρεθεί μια ακόμα λύση του προβλήματος στη γειτονιά του x_b . Το σημείο αυτό βρίσκεται με χρήση της μεθόδου DDS, μέσω της οποίας παράγονται N_t υποψηφίες λύσεις. Από τους N_t συνολικούς υποψηφίους επιλέγεται ως γνωστόν εκείνος που δίνει τη χαμηλότερη τιμή στην Acquisition Function. Έτσι, παράγεται μια νέα λύση x_{ref} , η οποία εισέρχεται σίγουρα στο εξωτερικό αρχείο, ενώ αντικαθιστά το x_{worst} στον πληθυσμό, εφόσον $f(x_{ref}) < f(x_{worst})$.

Βήμα 12^ο : Ολοκληρώνεται ο συγκεκριμένος επαναληπτικός κύκλος. Γίνεται έλεγχος αν ικανοποιείται κάποιος από τους τρόπους τερματισμού του αλγορίθμου, διαφορετικά ξεκινάει ο επόμενος επαναληπτικός κύκλος. Οι δύο τρόποι με τους οποίους μπορεί να ολοκληρωθεί ο αλγόριθμος είναι:

- 1) ο αριθμός των υπολογισμών της στοχικής συνάρτησης να υπερβεί το μέγιστο επιτρεπτό όριο, το οποίο εισάγεται από τον χρήστη
- 2) όταν επιτευχθεί το κριτήριο σύγκλισης του αλγορίθμου, δηλαδή όταν η σύγκλιση του είναι καλύτερη από το όριο σύγκλισης, το οποίο εισάγεται από το χρήστη

Στον ακόλουθο σύνδεσμο γίνεται η παρουσίαση υλοποίησης του AMSEEAS σε γλώσσα προγραμματισμού Python:

<https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/AMSEEAS>

Όπως ειπώθηκε και στο βήμα 2, ένας αλγόριθμος βελτιστοποίησης εκτελεί ελαχιστοποίηση πάνω στο προσαρμοσμένο μεταμοντέλο του βήματος 1. Το ερώτημα που τίθεται και διερευνήθηκε στα πλαίσια της εργασίας είναι το ποιος θα είναι αυτός. Ο SEEAS, όπως έχει αναφερθεί, χρησιμοποιεί σαν αλγόριθμο βελτιστοποίησης, την αρχική έκδοση του EAS. Για την επίτευξη του καλύτερου δυνατού αποτελέσματος, αναπτύχθηκαν δύο διαφορετικές εκδοχές του AMSEEAS, πανομοιότυπες μεταξύ τους, με μοναδική διαφορά τον αλγόριθμο βελτιστοποίησης που χρησιμοποιούν.

- Στην 1^η εκδοχή χρησιμοποιήθηκε ως αλγόριθμος βελτιστοποίησης η αρχική έκδοση του EAS.
- Στην 2^η εκδοχή χρησιμοποιήθηκε ως αλγόριθμος βελτιστοποίησης ένας τυπικός γενετικός αλγόριθμος που παραλήφθηκε υλοποιημένος από μια βιβλιοθήκη της Python.

Μετά την υλοποίηση των 2 αυτών εκδοχών του AMSEEAS, ακολούθησε διεξοδική σύγκριση των αποτελεσμάτων τους πάνω σε πολλές συναρτήσεις ελέγχου και σε διαφορετικούς τύπους προβλήματος για να εξακριβωθεί ποια εκδοχή από τις δύο ανταποκρίνεται καλύτερα.

Η σύγκριση της επίδοσης των 2 αυτών εκδοχών του AMSEEAS, όπως επίσης και η σύγκριση της επίδοσης της τελικής εκδοχής του AMSEEAS με άλλους δημοφιλείς αλγορίθμους βελτιστοποίησης, συμπεριλαμβανομένου και του SEEAS, πάνω σε πολλές συναρτήσεις ελέγχου, παρουσιάζονται στο επόμενο κεφάλαιο.

Κεφάλαιο 7^ο

7 Συγκρίσεις και αξιολογήσεις των επιδόσεων των αλγορίθμων πάνω σε συναρτήσεις ελέγχου

7.1. Συναρτήσεις ελέγχου

Για την πραγματοποίηση όλων των συγκρίσεων που θα παρουσιαστούν σε αυτό το κεφάλαιο, χρησιμοποιήθηκε μια δημοφιλής «σουίτα συναρτήσεων», αποτελούμενη από 6 γνωστές μαθηματικές συναρτήσεις, κάθε μια εκ των οποίων έχει τα δικά της ξεχωριστά στοιχεία πολυπλοκότητας, που καθιστούν αρκετά απαιτητική τη διαδικασία προσέγγισης του ολικού ακροτάτου. Η «σουίτα συναρτήσεων» αποτελείται από τις εξής συναρτήσεις:

- 1) Sphere Function (Objective Function 1 - OF1)
- 2) Ackley Function (Objective Function 2 – OF2)
- 3) Griewank Function (Objective Function 3 – OF3)
- 4) Zakharov Function (Objective Function 4 – OF4)
- 5) Rastrigin Function (Objective Function 5 - OF5)
- 6) Levy Function (Objective Function 6 – OF6)

Ο μαθηματικός τύπος κάθε μιας εξ αυτών υπάρχει στο [Παράρτημα Α](#), ωστόσο εδώ αναφέρεται ότι κάθε μια από τις 6 αυτές συναρτήσεις έχει ολικό ελάχιστο την τιμή $f_{\min} = 0$.

Με στόχο την καλύτερη και πιο αναλυτική σύγκριση των αλγορίθμων, για κάθε μια από τις 6 αυτές συναρτήσεις, μελετήθηκαν 4 διαφορετικά προβλήματα:

- 1) Το πρόβλημα αποτελείται από $n = 15$ μεταβλητές ελέγχου και ο μέγιστος επιτρεπόμενος αριθμός υπολογισμών της στοχικής συνάρτησης είναι $MFE = 500$ υπολογισμοί.

- 2) Το πρόβλημα αποτελείται από $n = 15$ μεταβλητές ελέγχου και ο μέγιστος επιτρεπόμενος αριθμός υπολογισμών της στοχικής συνάρτησης είναι $MFE = 1000$ υπολογισμοί.
- 3) Το πρόβλημα αποτελείται από $n = 30$ μεταβλητές ελέγχου και ο μέγιστος επιτρεπόμενος αριθμός υπολογισμών της στοχικής συνάρτησης είναι $MFE = 500$ υπολογισμοί.
- 4) Το πρόβλημα αποτελείται από $n = 30$ μεταβλητές ελέγχου και ο μέγιστος επιτρεπόμενος αριθμός υπολογισμών της στοχικής συνάρτησης είναι $MFE = 1000$ υπολογισμοί.

Με τον τρόπο αυτό, προκύπτουν συνολικά $6 \times 2 \times 2 = 24$ διαφορετικά προβλήματα προς σύγκριση. Για τη συλλογή δείγματος ικανοποιητικού μεγέθους, κάθε αλγόριθμος θα «τρέχει» το καθένα από αυτά τα 24 προβλήματα συνολικά 30 φορές. Προφανώς, κοινός στόχος του κάθε αλγορίθμου σε κάθε «τρέξιμο» που πραγματοποιεί, είναι η καλύτερη προσέγγιση του ολικού ακρότατου της κάθε συνάρτησης, δηλαδή η πιο κοντινή σύγκλιση του καθενός στην τιμή 0. Έτσι, σε κάθε τρέξιμο του κάθε αλγορίθμου σημειώνεται η καλύτερη προσέγγιση του ολικού ακροτάτου, δηλαδή η ελάχιστη τιμή που βρήκε ο καθένας κατά την ολοκλήρωση όλων των επαναληπτικών κύκλων. Πέρα από αυτό, ανά 50 υπολογισμούς της αντικειμενικής συνάρτησης σημειώνεται και η τρέχουσα βέλτιστη τιμή που έχει προσεγγιστεί μέχρι τη δεδομένη στιγμή, για να γίνει έπειτα και σύγκριση σχετικά με το πόσο γρήγορα προσεγγίζει ο κάθε αλγόριθμος το ολικό ακρότατο.

Για την αξιολόγηση της επίδοσης του κάθε αλγορίθμου στα 24 αυτά προβλήματα, από το συνολικό δείγμα των 30 «τρέξιμάτων» που πραγματοποιεί ο καθένας, υπολογίζονται τα στατιστικά χαρακτηριστικά του δείγματος και συγκεκριμένα η μέση τιμή, η τυπική απόκλιση και η διάμεσος. Έπειτα δημιουργούνται δύο διαφορετικοί συγκριτικοί πίνακες. Στον πρώτο πίνακα οι συγκρίσεις γίνονται σύμφωνα με την μέση τιμή του ελαχίστου που έχει προσεγγιστεί από τα 30 «τρέξιματα», ενώ στον δεύτερο σύμφωνα με τη διάμεσο του δείγματος.

Στόχος του κάθε αλγορίθμου είναι η κυριαρχία του έναντι των υπολοίπων σε όσο το δυνατόν περισσότερα από τα 24 προβλήματα, δηλαδή η επίτευξη του χαμηλότερου μέσου όρου ολικού ελαχίστου από τα 30 «τρέξιματα» ή αντίστοιχα της χαμηλότερης διαμέσου ολικού ελαχίστου.

Για να είναι δίκαια η σύγκριση, όλοι οι αλγόριθμοι βελτιστοποίησης θα χρησιμοποιούν το ίδιο μέγεθος πληθυσμού σε κάθε γενιά λύσεων, το οποίο τίθεται ίσο με $m = 32$ στα προβλήματα που υπάρχουν $n = 15$ μεταβλητές ελέγχου και $m = 62$ στα προβλήματα με $n = 30$ μεταβλητές ελέγχου. Επιπλέον, σε όλους τους αλγορίθμους η παραγωγή των αρχικών τυχαίων λύσεων γίνεται με χρήση της μεθόδου Latin Hypercube Sampling (LHS).

7.2. Διερεύνηση της καλύτερης εκδοχής του AMSEEAS

Όπως ειπώθηκε στο τέλος του προηγούμενου κεφαλαίου, στα πλαίσια της εργασίας υλοποιήθηκαν δύο διαφορετικές εκδοχές του AMSEEAS. Στην 1^η εκδοχή, ο αλγόριθμος βελτιστοποίησης που καλείται για να κάνει ελαχιστοποίηση στο εκάστοτε προσαρμοσμένο μεταμοντέλο πάνω στα δεδομένα του εξωτερικού αρχείου, είναι η αρχική έκδοση του EAS, ενώ στην 2^η εκδοχή χρησιμοποιείται ένας τυπικός γενετικός αλγόριθμος (GA), που ενσωματώθηκε στον κώδικα από μια δημοφιλή βιβλιοθήκη συναρτήσεων της Python. Πέρα από αυτή τη διαφοροποίηση, οι δύο εκδοχές είναι πανομοιότυπες. Για την εύρεση της καλύτερης εκδοχής, που θα αναπαριστά και την τελική εκδοχή του AMSEEAS, οι 2 εκδοχές υποβλήθηκαν σε αξιολόγηση με χρήση της «σουίτας» των 6 συναρτήσεων ελέγχου, οι οποίες δημιούργησαν συνολικά 24 ξεχωριστά προβλήματα βελτιστοποίησης. Η κάθε εκδοχή έτρεξε καθένα από τα 24 αυτά προβλήματα 30 φορές, με στόχο τη συλλογή επαρκούς δείγματος, και πραγματοποιήθηκαν δύο διαφορετικές συγκρίσεις, μια σύμφωνα με την μέση τιμή του ελαχίστου που προσεγγίστηκε από τα 30 «τρεξίματα», και μια σύμφωνα με τη διάμεσο του ελαχίστου. Έπειτα, ακολούθησε η σύγκριση της ταχύτητας σύγκλισης των 2 εκδοχών στο ολικό ακρότατο, δηλαδή σύγκριση σχετικά με πόσο γρήγορα προχωράει η διαδικασία εύρεσης του ολικού ακροτάτου ανάλογα με τον αριθμό υπολογισμών της στοιχικής συνάρτησης μέχρι τη δεδομένη στιγμή. Ακολουθούν τα αποτελέσματα όλων αυτών των συγκρίσεων:

	MFE	Test Function	AMSEEAS (EAS)		AMSEEAS (GA)	
			Mean	StDev	Mean	StDev
15-D	500	OF1	0.0002	0.0002	0.0002	0.0002
		OF2	0.003	0.003	0.002	0.002
		OF3	0.858	0.085	0.853	0.118
		OF4	50.531	27.267	50.708	24.828
		OF5	30.036	19.452	30.651	18.990
		OF6	0.261	0.452	0.231	0.441
	1000	OF1	0.0003	0.0004	0.0003	0.0002
		OF2	1.43E-05	0.0000	0.0002	0.0003
		OF3	0.780	0.121	0.833	0.118
		OF4	33.285	17.686	29.956	13.854
		OF5	16.545	7.682	19.689	9.266
		OF6	0.381	0.590	0.241	0.388
	MFE	Test Function	AMSEEAS (EAS)		AMSEEAS (GA)	
			Mean	StDev	Mean	StDev
30-D	500	OF1	0.035	0.121	0.003	0.002
		OF2	0.108	0.249	0.032	0.046
		OF3	0.998	0.034	0.983	0.038
		OF4	252.313	70.928	298.921	116.243
		OF5	163.939	38.410	177.644	20.262
		OF6	0.404	0.221	0.660	0.714
	1000	OF1	0.023	0.055	0.003	0.003
		OF2	0.0002	0.000	0.002	0.003
		OF3	1.057	0.268	1.005	0.106
		OF4	177.260	61.401	175.583	59.006
		OF5	77.695	48.043	54.790	42.263
		OF6	0.263	0.229	0.239	0.184

Πίνακας 7-1: Μέση τιμή και τυπική απόκλιση του ολικού ελαχίστου των 2 εκδοχών του AMSEEAS στα 24 προβλήματα

Αριστερά, βρίσκεται η εκδοχή του AMSEEAS που χρησιμοποιεί τον EAS σαν αλγόριθμο βελτιστοποίησης, ενώ δεξιά η εκδοχή που χρησιμοποιεί τον GA. Η εκδοχή που επιτυγχάνει καλύτερη (δηλαδή χαμηλότερη) μέση τιμή (mean) ολικού ελαχίστου σε καθένα από τα 24 προβλήματα επισκιάζεται με κόκκινο χρώμα, αν είναι η

AMSEEAS (EAS), ή με πράσινο, αν είναι η AMSEEAS (GA). Όπως γίνεται αντιληπτό από τον πίνακα (7-1), η εκδοχή AMSEEAS (GA) προσεγγίζει καλύτερα το ολικό ακρότατο σε 13/24 περιπτώσεις, η εκδοχή AMSEEAS (EAS) σε 9/24 περιπτώσεις, ενώ σε 2/24 περιπτώσεις παρατηρείται ισοπαλία. Επομένως, στον πρώτο συγκριτικό πίνακα νικητής είναι η εκδοχή AMSEEAS (GA).

Συνεχίζουμε με το δεύτερο συγκριτικό πίνακα, που αφορά τη διάμεσο του δείγματος:

	MFE	Test Function	AMSEEAS (EAS)	AMSEEAS (GA)
			Median	Median
15-D	500	OF1	0.0001	0.0001
		OF2	0.002	0.001
		OF3	0.865	0.875
		OF4	44.255	44.470
		OF5	25.795	29.880
		OF6	0.148	0.090
	1000	OF1	0.0002	0.0002
		OF2	0.0000004	0.0001
		OF3	0.818	0.865
		OF4	30.621	27.715
		OF5	15.525	18.465
		OF6	0.137	0.130
30-D	MFE	Test Function	AMSEEAS (EAS)	AMSEEAS (GA)
			Median	Median
	500	OF1	0.002	0.0020
		OF2	0.043	0.020
		OF3	1.002	0.990
		OF4	251.690	276.105
		OF5	171.856	179.590
		OF6	0.330	0.475
	1000	OF1	0.003	0.003
		OF2	0.0001	0.001
		OF3	0.984	0.990
		OF4	160.145	170.215
OF5		62.386	39.465	
OF6		0.190	0.190	

Πίνακας 7-2: Διάμεσος του ολικού ελαχίστου των 2 εκδοχών του AMSEEAS στα 24 προβλήματα

Αριστερά, βρίσκεται η εκδοχή του AMSEEAS που χρησιμοποιεί τον EAS σαν αλγόριθμο βελτιστοποίησης, ενώ δεξιά η εκδοχή που χρησιμοποιεί τον GA. Η εκδοχή που επιτυγχάνει καλύτερη διάμεσο (median) ολικού ελαχίστου σε καθένα από τα 24 προβλήματα επισκιάζεται με κόκκινο χρώμα, αν είναι η AMSEEAS (EAS), ή με πράσινο, αν είναι η AMSEEAS (GA). Όπως γίνεται αντιληπτό από τον πίνακα (7-2), η εκδοχή AMSEEAS (GA) προσεγγίζει καλύτερα το ολικό ακρότατο σε 7 / 24 περιπτώσεις, η εκδοχή AMSEEAS (EAS) σε 12 / 24 περιπτώσεις, ενώ σε 5 / 24 περιπτώσεις παρατηρείται ισοπαλία. Επομένως, στον δεύτερο συγκριτικό πίνακα νικητής είναι η εκδοχή AMSEEAS (EAS).

Τέλος, στο [Παράρτημα Β](#) παρουσιάζονται οι καμπύλες σύγκλισης των 2 εκδοχών του AMSEEAS στο ολικό ελάχιστο της κάθε στοχικής συνάρτησης. Τα διαγράμματα ομαδοποιούνται σε 6 4αδες, όπου κάθε 4αδα αφορά τις 4 διαφορετικές περιπτώσεις προβλήματος (ανάλογα με το αν οι μεταβλητές ελέγχου είναι 15 ή 30 και οι μέγιστοι υπολογισμοί 500 ή 1000) για την κάθε μια από τις 6 αντικειμενικές συναρτήσεις.

Από τα αποτελέσματα όλων των ανωτέρω συγκρίσεων, γίνεται αντιληπτό ότι και οι δύο εκδοχές του AMSEEAS επιτυγχάνουν πολύ ικανοποιητικές συγκλίσεις στα ολικά ακρότατα, στην πλειονότητα των περιπτώσεων. Η εκδοχή AMSEEAS (GA) συμπεριφέρεται καλύτερα όταν χρησιμοποιείται ως μέτρο επίδοσης η μέση τιμή του ελαχίστου από τα 30 «τρεξίματα», ενώ η εκδοχή AMSEEAS (EAS) όταν χρησιμοποιείται η διάμεσος. Ωστόσο, σπουδαιότερη σημαντικότητα για την απόδοση της αξιοπιστίας του μοντέλου ανάμεσα στα δύο μέτρα επίδοσης θεωρείται ότι έχει η μέση τιμή. Για το λόγο αυτό, ως τελική εκδοχή του AMSEEAS, λαμβάνεται η εκδοχή AMSEEAS (GA). Επομένως, σε όλες τις επόμενες συγκρίσεις αλγορίθμων, αυτή η εκδοχή θα αντιπροσωπεύει τον αλγόριθμο AMSEEAS.

7.3. Σύγκριση της επίδοσης του AMSEEAS με άλλους δημοφιλείς αλγορίθμους βελτιστοποίησης

Για την αξιολόγηση της επίδοσης του AMSEEAS στα 24 αυτά προβλήματα βελτιστοποίησης, πραγματοποιήθηκε σύγκριση του με 5 διαφορετικούς αλγορίθμους βελτιστοποίησης:

- τον εξελικτικό αλγόριθμο ανόπτησης – απλόκου (EAS)
- τον αλγόριθμο DDS
- τον εξελικτικό αλγόριθμο ανόπτησης–απλόκου εμπλουτισμένο με υποκατάστατα μοντέλα (SEEAS)
- τον αλγόριθμο DYCORS
- τον αλγόριθμο MLMSRBF

Από αυτούς, οι αλγόριθμοι SEEAS, DYCORS και MLMSRBF ενσωματώνουν στον πυρήνα τους ένα υποκατάστατο μοντέλο και συγκεκριμένα το Cubic RBF with Linear Polynomial Tail, ενώ οι αλγόριθμοι EAS και DDS δεν υποβοηθούνται από κάποιο μεταμοντέλο.

Όλοι οι αλγόριθμοι υποβλήθηκαν στα ίδια 24 προβλήματα βελτιστοποίησης. Για να είναι δίκαια η σύγκριση, όλοι οι αλγόριθμοι χρησιμοποιούν το ίδιο μέγεθος πληθυσμού και χρησιμοποιείται η ίδια μέθοδος για την παραγωγή των αρχικών τυχαίων λύσεων της 1^{ης} γενιάς. Επιπλέον, σε όλους τους αλγορίθμους, τοποθετούνται οι προκαθορισμένες τιμές για όλες τις παραμέτρους/υπερπαραμέτρους, όπως αυτές προτείνονται στα εγχειρίδια χρήσης τους.

7.3.1. Σύντομη περιγραφή των αλγορίθμων DDS, MLMSRBF, DYCORS

Ο αλγόριθμος DDS: Η πλήρης ονομασία του είναι Dynamically Dimension Search. Αναπτύχθηκε από τους Tolson και Shoemaker (2007) και είναι ένας στοχαστικός αλγόριθμος σχεδιασμένος για να βρίσκει κοντινές προσεγγίσεις στα ολικά ακρότατα με λίγους υπολογισμούς της στοχαστικής συνάρτησης. Όπως είναι στημένος ο κώδικας, στα αρχικά στάδια γίνεται εξερεύνηση του συνολικού χώρου αναζήτησης των λύσεων και μόλις συμπληρωθεί ένας συγκεκριμένος αριθμός υπολογισμών της στοχαστικής συνάρτησης, η αναζήτηση μετατρέπεται σε περισσότερο τοπική. Η σύγκλιση προχωρά, χρησιμοποιώντας τη εκάστοτε βέλτιστη λύση του πληθυσμού και μέσω μιας εξελικτικής διαδικασίας, βασισμένη στην κανονική κατανομή. Ο DDS έχει γνωρίσει μεγάλη επιτυχία σε πολλές διαφορετικές εφαρμογές, όπως επιβεβαιώνει και η διεθνής βιβλιογραφία.

Ο αλγόριθμος MLMSRBF: Η πλήρης ονομασία του είναι Multistart Local Metric Stochastic Radial Basis Function. Αναπτύχθηκε από τους Regis και Shoemaker (2007) και αποτελεί πρακτικά επέκταση του DDS, στον οποίο ενσωματώνεται ένα υποκατάστατο μοντέλο και συγκεκριμένα, σε αυτή την έκδοση, το Cubic RBF with Linear Polynomial Tail. Η διαδικασία σύγκλισης προχωράει παρόμοια με τον DDS, όμως αξίζει να επισημανθούν δύο σημαντικές διαφοροποιημένες λειτουργίες:

- Ο αλγόριθμος δεν χρησιμοποιεί σαν μεταμοντέλο αυτούσιο το RBF, αλλά αυτό περνάει σαν όρισμα σε μια συνάρτηση πολύ παρόμοια με την Acquisition Function, που αναλύθηκε στο [Υποκεφάλαιο 4.2.2](#), η οποία κάνει τις τελικές προβλέψεις του μεταμοντέλου.
- Όταν ο αλγόριθμος παγιδεύεται σε κάποιο τοπικό ακρότατο, ενεργοποιείται μια λειτουργία, κατά την οποία για την διερεύνηση του χώρου αναζήτησης χρησιμοποιούνται περισσότεροι από ένας πληθυσμοί. Ο μηχανισμός αυτός ενεργοποιείται με στόχο την προσπάθεια αποφυγής από το τοπικό ακρότατο.

Και αυτός ο αλγόριθμος γνώρισε πολύ μεγάλη επιτυχία, καθώς εφαρμόστηκε αποτελεσματικά σε πληθώρα διαφορετικών εφαρμογών, συμπεριλαμβανομένων και προβλημάτων υδατικών πόρων.

Ο αλγόριθμος DYCORS: Η πλήρης ονομασία του είναι DYNamic COordinate Search-Multistart Local Metric Stochastic RBF (DYCORS – LMSRBF). Αναπτύχθηκε από τους Regis και Shoemaker (2013) και πρόκειται επίσης για έναν αλγόριθμο, ο οποίος υποβοηθείται από ένα υποκατάστατο μοντέλο και συγκεκριμένα, σε αυτή την έκδοση, από το Cubic RBF with Linear Polynomial Tail. Αποτελεί επέκταση του LMSRBF, στον οποίο μεταξύ άλλων αλλαγών, έχουν εισαχθεί και συντελεστές βάρους για την εξισορρόπηση «exploration» και «exploitation» του μεταμοντέλου.

7.3.2. Συγκριτικοί πίνακες και σχολιασμός των αποτελεσμάτων

Αρχικά, παρουσιάζονται οι συγκριτικοί πίνακες που προέκυψαν, χρησιμοποιώντας ως μέτρο επίδοσης τη μέση τιμή του κάθε δείγματος:

Mean and standard deviation of best solutions in 15-D test problems (optimal results are highlighted)													
MFE	Test Function	EAS		DDS		SEEAS		DYCORS		MLMSRBF		AMSEAS (GA)	
		Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
500	OF1	1.938	0.978	0.852	0.479	0.002	0.001	0.002	0.001	0.019	0.014	0.0002	0.0002
	OF2	7.159	1.723	6.025	1.314	0.812	0.233	0.809	0.372	2.231	0.658	0.002	0.002
	OF3	7.682	2.997	2.626	1.269	0.538	0.118	0.885	0.084	1.085	0.052	0.853	0.118
	OF4	39.434	14.894	137.447	52.366	59.144	28.023	158.669	47.788	150.411	49.875	50.708	24.828
	OF5	86.245	14.148	24.887	7.081	46.268	15.359	38.958	12.340	45.920	18.803	30.651	18.990
	OF6	1.905	0.877	0.681	0.314	0.203	0.105	1.208	1.406	1.344	2.129	0.231	0.441
1000	OF1	0.378	0.177	0.150	0.079	0.001	0.001	0.001	0.000	0.011	0.007	0.0003	0.0002
	OF2	3.523	0.936	3.847	0.528	0.437	0.208	0.607	0.092	1.862	0.556	0.0002	0.0003
	OF3	2.444	1.061	1.505	0.299	0.368	0.140	0.809	0.082	1.040	0.037	0.833	0.118
	OF4	26.828	17.895	97.541	38.226	41.290	26.639	121.266	36.925	121.359	37.730	29.956	13.854
	OF5	59.735	17.012	11.233	3.136	29.733	12.838	33.585	13.490	35.784	11.031	19.689	9.266
	OF6	0.767	0.292	0.234	0.104	0.124	0.060	0.536	0.860	0.524	0.863	0.241	0.388

Πίνακας 7-3: Μέση τιμή και τυπική απόκλιση των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 15-D

Mean and standard deviation of best solutions in 30-D test problems (optimal results are highlighted)													
MFE	Test Function	EAS		DDS		SEEAS		DYCORS		MLMSRBF		AMSEAS (GA)	
		Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev	Mean	StDev
500	OF1	4.305	1.163	9.516	2.737	0.019	0.006	0.083	0.034	0.739	0.708	0.003	0.002
	OF2	9.923	1.160	12.872	1.329	1.878	0.301	4.297	3.721	6.193	4.362	0.032	0.046
	OF3	17.866	3.455	38.398	12.050	0.782	0.118	1.265	0.079	3.459	1.927	0.983	0.038
	OF4	117.821	28.757	562.145	113.230	173.240	44.185	472.815	90.897	575.424	174.073	298.921	116.243
	OF5	228.693	18.442	132.149	24.567	122.658	19.427	112.046	23.076	165.437	46.846	177.644	20.262
	OF6	6.338	2.652	15.823	5.481	0.659	0.184	3.407	2.540	7.326	10.944	0.660	0.714
1000	OF1	2.529	0.933	2.112	0.791	0.006	0.004	0.011	0.004	0.358	0.177	0.003	0.003
	OF2	6.516	0.845	7.670	0.924	1.206	0.297	1.085	0.168	3.643	1.103	0.002	0.003
	OF3	8.836	2.617	8.273	2.679	0.549	0.093	1.020	0.026	2.420	0.713	1.005	0.106
	OF4	94.598	20.317	412.238	118.573	151.472	54.097	403.812	93.081	491.425	146.097	175.583	59.006
	OF5	198.335	16.587	71.598	15.028	98.371	19.505	85.267	22.956	134.864	39.193	54.790	42.263
	OF6	2.683	0.736	3.921	2.215	0.443	0.126	4.213	5.440	2.865	4.583	0.239	0.184

Πίνακας 7-4: Μέση τιμή και τυπική απόκλιση των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 30-D

	EAS	DDS	SEEAS	DYCORDS	MLMSRBF	AMSEEAS (GA)
1ος (φορές)	4	2	7	1	0	11
2ος (φορές)	0	2	10	5	0	8
3ος (φορές)	1	2	5	10	0	4
4ος (φορές)	1	2	1	4	16	0
5ος (φορές)	6	9	1	2	5	1
6ος (φορές)	12	7	0	2	3	0

Πίνακας 7-5: Κατάταξη της απόδοσης των αλγορίθμων σύμφωνα με τη μέση τιμή των καλύτερων λύσεων που βρέθηκαν στα προβλήματα 15-D και 30-D

Όπως γίνεται αντιληπτό και μέσω του πίνακα 7-5, εφόσον χρησιμοποιηθεί ως μέτρο επίδοσης των αλγορίθμων η μέση τιμή του δείγματος, ο AMSEEAS (GA) κυριαρχεί σε 11 / 24 προβλήματα βελτιστοποίησης, στα οποία επιτυγχάνει καλύτερη σύγκλιση στο ολικό ελάχιστο της κάθε συνάρτησης από όλους τους υπόλοιπους αλγορίθμους. Αμέσως μετά, όπως ήταν και αναμενόμενο, βρίσκεται η αρχική έκδοση του SEEAS, η οποία καταλαμβάνει την 1^η θέση σε 7 / 24 προβλήματα βελτιστοποίησης. Ακολουθεί ο EAS με κυριαρχία σε 4 / 24 περιπτώσεις, ο DDS με κυριαρχία σε 2 / 24 και τέλος ο DYCORDS, ο οποίος κερδίζει σε 1 / 24 προβλήματα. Ο αλγόριθμος MLMSRBF δεν υπερτερεί των υπολοίπων σε καμία από τις 24 περιπτώσεις. Αξίζει να επισημανθεί, ότι ο AMSEEAS (GA) καταλαμβάνει την 1^η ή 2^η θέση σε 19 / 24 προβλήματα, ενώ την 1^η ή 2^η ή 3^η σε 23 / 24.

Λίγο πιο αναλυτικά, στα προβλήματα βελτιστοποίησης με 15 μεταβλητές ελέγχου (15-D), ο AMSEEAS (GA) επιτυγχάνει την καλύτερη επίδοση (δηλαδή τη χαμηλότερη μέση τιμή ολικού ελαχίστου από τα 30 «τρεξίματα») σε 2 από τις 6 στοχαστικές συναρτήσεις (OF1, OF2) για MFE=500 και σε άλλες 2 από τις 6 συναρτήσεις (OF1, OF2) για MFE=1000. Αντιστοίχως, στα προβλήματα με 30 μεταβλητές ελέγχου (30 – D), στα οποία προφανώς η αύξηση των διαστάσεων αυξάνει και τη δυσκολία προσέγγισης των ολικών ελαχίστων, ο AMSEEAS (GA) κυριαρχεί σε 3 από τις 6 συναρτήσεις (OF1, OF2, OF6) για MFE=500 και σε 4 από τις 6 (OF1, OF2, OF5, OF6) για MFE=1000.

Από τα αποτελέσματα των συγκρίσεων, γίνεται επίσης αντιληπτό, ότι κανένας από τους 6 αλγορίθμους βελτιστοποίησης δεν κατορθώνει να προσεγγίσει με ακρίβεια το ολικό ελάχιστο των στοχαστικών συναρτήσεων Zakharov (OF4) και Rastrigin (OF5), κάτι το οποίο δεν δημιουργεί έκπληξη, καθώς η πολύ περίπλοκη επιφάνεια απόκρισης των δύο αυτών συναρτήσεων κάνει εξαιρετικά δύσκολη την προσαρμογή οποιουδήποτε μεταμοντέλου σε αυτές.

Όπως ήταν αναμενόμενο, η αύξηση του μέγιστου επιτρεπτού αριθμού υπολογισμών της στοχαστικής συνάρτησης από MFE=500 σε MFE=1000 βελτιώνει την επίδοση όλων των αλγορίθμων. Μεγαλύτερη βελτίωση, βέβαια, παρατηρείται στους αλγορίθμους EAS και DDS, το οποίο είναι λογικό, μιας και αυτοί οι δύο δεν υποβοηθούνται από κάποιο υποκατάστατο μοντέλο και έτσι η διαδικασία της σύγκλισης προχωράει με αργότερους ρυθμούς από ότι στους υπόλοιπους αλγορίθμους.

Έπειτα, ακολουθούν οι συγκριτικοί πίνακες που προέκυψαν, χρησιμοποιώντας ως μέτρο επίδοσης τη **διάμεσο** του κάθε δείγματος:

<i>Median of best solutions in 15-D test problems (optimal results are highlighted)</i>							
MFE	Test Function	EAS	DDS	SEEAS	DYCORS	MLMSRBF	AMSEAS (GA)
		Median	Median	Median	Median	Median	Median
500	OF1	1.457	0.684	0.002	0.002	0.012	0.0001
	OF2	7.367	5.942	0.838	0.745	2.353	0.001
	OF3	7.446	2.312	0.513	0.921	1.088	0.875
	OF4	34.205	133.574	53.874	154.151	147.998	44.470
	OF5	85.223	24.714	45.061	37.912	37.696	29.880
	OF6	1.592	0.616	0.198	0.681	0.488	0.090
1000	OF1	0.380	0.131	0.001	0.001	0.008	0.0002
	OF2	3.519	3.877	0.410	0.574	1.629	0.0001
	OF3	2.211	1.400	0.360	0.819	1.027	0.865
	OF4	25.224	98.089	34.413	127.557	110.313	27.715
	OF5	58.926	10.813	31.808	32.644	34.522	18.465
	OF6	0.765	0.216	0.114	0.069	0.191	0.130

Πίνακας 7-6: Διάμεσος των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 15-D

<i>Median of best solutions in 30-D test problems (optimal results are highlighted)</i>							
MFE	Test Function	EAS	DDS	SEEAS	DYCORS	MLMSRBF	AMSEAS (GA)
		Median	Median	Median	Median	Median	Median
500	OF1	4.391	9.828	0.018	0.073	0.590	0.002
	OF2	9.844	13.110	1.918	3.144	4.725	0.020
	OF3	17.758	36.453	0.807	1.249	2.974	0.990
	OF4	114.878	540.070	168.695	456.956	570.266	276.105
	OF5	232.766	130.090	121.973	112.009	156.834	179.590
	OF6	5.264	15.496	0.630	2.075	2.302	0.475
1000	OF1	2.516	1.860	0.005	0.009	0.270	0.003
	OF2	6.579	7.831	1.170	1.108	3.438	0.001
	OF3	8.741	7.920	0.554	1.025	2.507	0.990
	OF4	95.274	386.140	147.120	409.986	465.057	170.215
	OF5	200.952	71.160	97.994	85.728	127.299	39.465
	OF6	2.458	2.918	0.431	2.762	1.412	0.190

Πίνακας 7-7: Διάμεσος των καλύτερων λύσεων που βρέθηκαν από τους αλγορίθμους στα προβλήματα 30-D

	EAS	DDS	SEEAS	DYCORS	MLMSRBF	AMSEAS (GA)
1ος (φορές)	4	2	4	2	0	12
2ος (φορές)	0	1	13	5	0	7
3ος (φορές)	0	1	5	9	3	4
4ος (φορές)	1	4	1	4	15	0
5ος (φορές)	6	9	1	2	4	1
6ος (φορές)	13	7	0	2	2	0

Πίνακας 7-8: Κατάταξη της απόδοσης των αλγορίθμων σύμφωνα με τη διάμεσο των καλύτερων λύσεων που βρέθηκαν στα προβλήματα 15-D και 30-D

Όπως γίνεται αντιληπτό και μέσω του πίνακα 7-8, τα αποτελέσματα σε αυτή την περίπτωση είναι ακόμα πιο εντυπωσιακά. Εφόσον χρησιμοποιηθεί ως μέτρο επίδοσης των αλγορίθμων η διάμεσος του κάθε δείγματος, ο AMSEAS (GA) κυριαρχεί σε 12 / 24 προβλήματα βελτιστοποίησης. Αμέσως μετά βρίσκονται ο SEEAS και ο EAS, οι οποίοι καταλαμβάνουν την 1^η θέση σε 4 / 24 προβλήματα έκαστος. Ακολουθούν ο DYCORS και ο DDS με κυριαρχία σε 2 / 24 περιπτώσεις ο καθένας, και τέλος ο

αλγόριθμος MLMSRBF δεν υπερτερεί, πάλι, πουθενά. Αξίζει να επισημανθεί, ότι ο AMSEEAS (GA) και εδώ καταλαμβάνει την 1^η ή 2^η θέση σε 19 / 24 προβλήματα, ενώ την 1^η ή 2^η ή 3^η σε 23 / 24.

Αναλυτικότερα, στα προβλήματα βελτιστοποίησης με 15 μεταβλητές ελέγχου, ο AMSEEAS (GA) επιτυγχάνει την καλύτερη επίδοση (δηλαδή τη χαμηλότερη διάμεσο ολικού ελαχίστου από τα 30 «τρεξίματα») σε 3 από τις 6 στοχαστικές συναρτήσεις (OF1, OF2, OF6) για MFE=500 και σε άλλες 2 από τις 6 συναρτήσεις (OF1, OF2) για MFE=1000. Αντιστοίχως, στα προβλήματα με 30 μεταβλητές ελέγχου, ο AMSEEAS (GA) κυριαρχεί σε 3 από τις 6 συναρτήσεις (OF1, OF2, OF6) για MFE=500 και σε 4 από τις 6 (OF1, OF2, OF5, OF6) για MFE=1000. Δηλαδή, υπερτερεί των υπολοίπων αλγορίθμων στις ίδιες ακριβώς περιπτώσεις με αυτές που βρέθηκαν χρησιμοποιώντας τη μέση τιμή ως μέτρο σύγκρισης, με μία επιπλέον κυριαρχία στη συνάρτηση Levy (OF6) για $n=15$ και MFE=500.

Κεφάλαιο 8^ο

8 Συγκρίσεις και αξιολογήσεις των επιδόσεων των αλγορίθμων πάνω σε πρόβλημα βαθμονόμησης παραμέτρων υδρολογικού μοντέλου

Για την περαιτέρω διερεύνηση της αποτελεσματικότητας του AMSEAS, θεωρήθηκε απαραίτητο, πέραν της σύγκρισής του με άλλους αλγορίθμους βελτιστοποίησης πάνω σε μαθηματικές συναρτήσεις με γνωστές αλγεβρικές εκφράσεις, να πραγματοποιηθεί και μια ανάλυση πάνω σε ένα τυπικό πρόβλημα του τομέα των υδατικών πόρων. Στις πραγματικές, αυτές, εφαρμογές των υδατικών πόρων παρουσιάζονται ορισμένες διαφοροποιήσεις σε σχέση με τις θεωρητικές πάνω σε συναρτήσεις ελέγχου, οι οποίες καθιστούν ακόμα πιο απαιτητική τη διαδικασία προσέγγισης του ολικού ακροτάτου. Πιο συγκεκριμένα, στις πραγματικές εφαρμογές, οι ιδιότητες της στοχαστικής συνάρτησης και η θέση του ολικού ακροτάτου δεν είναι γνωστά εκ των προτέρων. Πέραν αυτού, στα προβλήματα του τομέα, η αντικειμενική συνάρτηση δεν προκύπτει, πλέον, από την εφαρμογή κάποιας γνωστής αλγεβρικής έκφρασης. Για να προσδιοριστεί μια μεμονωμένη τιμή της στοχαστικής συνάρτησης, είναι απαραίτητο να τρέχει κάθε φορά ένα μοντέλο προσομοίωσης. Το μοντέλο προσομοίωσης αυτό, χρησιμοποιείται είτε για να περιγράψει κάποιο μεμονωμένο γεγονός, όπως την αιχμή κάποιας πλημμύρας είτε για να περιγράψει τη διαχρονική εξέλιξη ενός υδρολογικού ισοζυγίου σε μια πεπερασμένη χωρική ενότητα, όπως για παράδειγμα σε μια λεκάνη απορροής. Κατά την εφαρμογή του μοντέλου, όμως, εμφανίζονται πολλά σφάλματα, τα οποία οδηγούν στην αδυναμία προσέγγισης του ολικού ακροτάτου με απόλυτη ακρίβεια. Τα σφάλματα αυτά μπορεί να αφορούν στους ακόλουθους τομείς:

- σφάλματα στη μαθηματική δομή του μοντέλου, λόγω προσέγγισης των φυσικών διεργασιών με χρήση απλοποιημένων αναλυτικών σχέσεων
- σφάλματα στις μετρήσεις των δεδομένων εισόδου και εξόδου του συστήματος
- σφάλματα λόγω χρήσης ακατάλληλης και μη αντιπροσωπευτικής στοχαστικής συνάρτησης για το πρόβλημα

Βέβαια, η αδυναμία απόλυτης προσέγγισης του ακροτάτου ενισχύεται και από τον περιοριστικό παράγοντα του χρόνου που χαρακτηρίζει τα προβλήματα αυτά. Αναλυτικότερα, το μοντέλο προσομοίωσης έχει πολύ μεγάλο υπολογιστικό φόρτο (πολύ μεγαλύτερο από τον αντίστοιχο μιας τυπικής μαθηματικής συνάρτησης ελέγχου), γεγονός που οδηγεί συχνά τον αναλυτή στην ανάγκη να συμβιβαστεί με την

εύρεση μιας απλής προσέγγισης του ακροτάτου, προκειμένου να κατορθώσει να παραλάβει μια λύση εντός λογικών χρονικών πλαισίων.

8.1. Βασικές έννοιες και ορισμοί

Ο όρος υδρολογικό μοντέλο αφορά σε ένα σύνολο μαθηματικών μετασχηματισμών, που χρησιμοποιούν δεδομένα πεδίου και κάνουν λογικές υποθέσεις, σχετικές με φυσικούς μηχανισμούς, με απώτερο σκοπό την ακριβή ποσοτική εκτίμηση υδρολογικών μεταβλητών, ιδιαίτερα σημαντικών σε πραγματικές εφαρμογές. Οι υδρολογικές αυτές μεταβλητές, παρά το γεγονός ότι έχουν κάποια φυσική σημασία, είναι αδύνατο να υπολογιστούν μέσω των ιδιοτήτων της λεκάνης απορροής και για την εκτίμησή τους είναι απαραίτητο να εφαρμοστεί μια διαδικασία βελτιστοποίησης. Η διαδικασία βελτιστοποίησης, αυτή, επιλύει το λεγόμενο πρόβλημα βαθμονόμησης (calibration) ή αλλιώς ρύθμισης του υδρολογικού μοντέλου και αποτελεί μια από τις συνηθέστερες εφαρμογές υδραυλικού μηχανικού στα πλαίσια των υδατικών πόρων. Στόχος του προβλήματος είναι η εύρεση των κατάλληλων τιμών για τις ζητούμενες παραμέτρους του μοντέλου, ώστε να ελαχιστοποιηθεί η απόκλιση ανάμεσα στην προσομοιωμένη απόκριση του μοντέλου και στις πραγματικές μετρήσεις στην έξοδο του συστήματος.

Για την αξιολόγηση της απόκλισης της προσομοιωμένης απόκρισης του μοντέλου από την μετρημένη, απαιτείται κάποιο μέτρο επίδοσης που θα διακατέχει το ρόλο του κριτηρίου καλής προσαρμογής και θα αποτελεί στην ουσία τη στοχική συνάρτηση του προβλήματος. Ένα πολύ συνηθισμένο τέτοιο κριτήριο καλής προσαρμογής είναι το λεγόμενο μέσο τετραγωνικό σφάλμα (mean squared error – MSE), το οποίο δίνεται από τη σχέση:

$$MSE = \frac{1}{n} \sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8.1)$$

όπου y_i και \hat{y}_i η μετρημένη και η προσομοιωμένη χρονοσειρά εξόδου, αντίστοιχα, ενώ το n αποτελεί το συνολικό μήκος της περιόδου βαθμονόμησης του μοντέλου.

Ένα άλλο, εξίσου διαδεδομένο και δημοφιλές, κριτήριο καλής προσαρμογής, είναι ο λεγόμενος συντελεστής προσδιορισμού (coefficient of determination) ή αλλιώς αποτελεσματικότητα (efficiency) ή συντελεστής Nash-Sutcliffe (NSE), που δίνεται από τη σχέση:

$$NSE = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \mu_y)^2} \quad (8.2)$$

όπου y_i και \hat{y}_i η μετρημένη και η προσομοιωμένη χρονοσειρά εξόδου, αντίστοιχα, n το συνολικό μήκος της περιόδου βαθμονόμησης του μοντέλου και μ_y η μέση τιμή της μετρημένης χρονοσειράς εξόδου. Ο συντελεστής Nash-Sutcliffe είναι αδιάστατος και έχει σύνολο τιμών το $(-\infty, 1]$. Αν επιτευχθεί η τιμή $NSE=1$, αυτό σημαίνει ότι υπάρχει απόλυτη ταύτιση ανάμεσα στην προσομοιωμένη και την παρατηρημένη χρονοσειρά, δηλαδή το σφάλμα είναι μηδενικό. Αρνητικές τιμές του NSE ($NSE < 0$) υποδηλώνουν

πολύ κακή προσαρμογή του μοντέλου και όπως γίνεται κατανοητό η προσπάθεια έγκειται στο να επιτευχθούν τιμές του NSE που προσεγγίζουν τη μονάδα.

Στα πρόβλημα που εξετάζεται στο κεφάλαιο αυτό, επιλέχθηκε ως κριτήριο καλής προσαρμογής, ο συντελεστής Nash-Sutcliffe, ο οποίος θα αποτελεί επομένως και την στοχική συνάρτηση του προβλήματος, δηλαδή $f(x) \equiv \text{NSE}$.

Όπως αναλύθηκε και μέσω της εξίσωσης (2.2), κάθε πρόβλημα βελτιστοποίησης μπορεί να αναχθεί στο αντίστοιχο πρόβλημα ελαχιστοποίησης, μέσω της σχέσης $\max f(x) \equiv \min [-f(x)]$. Έτσι το πρόβλημα της βαθμονόμησης μεταπίπτει στο ακόλουθο πρόβλημα ελαχιστοποίησης:

$$\text{minimize } f(x), x_{\min} \leq x \leq x_{\max} \quad (8.3)$$

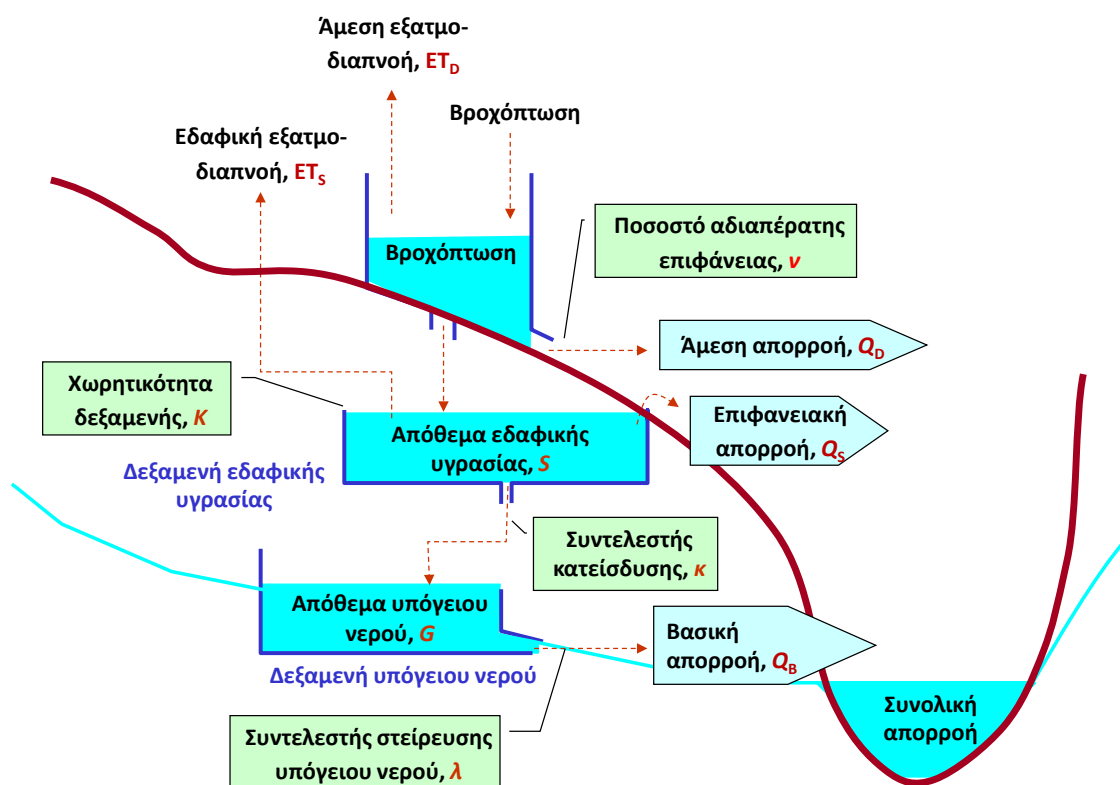
όπου $f(x)$ είναι κάποιο κριτήριο καλής προσαρμογής (στην συγκεκριμένη περίπτωση ο δείκτης NSE) και x το διάνυσμα των παραμέτρων του μοντέλου, με άνω και κάτω όρια διακύμανσης, τα x_{\min} και x_{\max} , αντίστοιχα. Επομένως, στόχος του προβλήματος θα είναι η καλύτερη προσέγγιση της χαμηλότερης δυνατής τιμής για την $-f(x) \equiv -\text{NSE}$, δηλαδή η κοντινότερη προσέγγιση στην τιμή -1.

8.2. Περιγραφή του μοντέλου υδατικού ισοζυγίου

Τα μοντέλα υδατικού ισοζυγίου χρησιμοποιούνται εδώ και πάρα πολλές δεκαετίες από την επιστημονική κοινότητα, καθώς παρέχουν λύση σε μια πολύ μεγάλη ποικιλία υδρολογικών προβλημάτων. Ως είσοδοι στα μοντέλα αυτά, λαμβάνονται, συνήθως, βροχομετρικά και μετεωρολογικά δεδομένα, που είναι διαθέσιμα στη λεκάνη απορροής (πχ. κατακρήμνιση, απολήψεις υπόγειου υδροφορέα, θερμοκρασία), ενώ μια πολύ συνηθισμένη έξοδος του κάθε μοντέλου είναι η εκτίμηση της επιφανειακής απορροής στην έξοδο της λεκάνης. Στα πλαίσια του προβλήματος που μελετάται στην παρούσα εργασία, επιδιώκεται η εφαρμογή ενός απλού μοντέλου υδατικού ισοζυγίου πάνω σε μια πραγματική λεκάνη απορροής, με στόχο την απόκτηση της χρονοσειράς επιφανειακής απορροής σε ένα συγκεκριμένο χρονικό διάστημα μελέτης. Έχοντας δεδομένες τις παρατηρημένες τιμές επιφανειακής απορροής που έχουν προκύψει από μετρήσεις για το υπό μελέτη χρονικό διάστημα στην έξοδο της συγκεκριμένης λεκάνης, επιδιώκεται η βαθμονόμηση (καλιμπράρισμα) του μοντέλου, δηλαδή η εύρεση των βέλτιστων τιμών για τις μεταβλητές ελέγχου του προβλήματος, με απώτερο σκοπό την καλύτερη δυνατή προσαρμογή των προσομοιωμένων τιμών απορροής πάνω στις παρατηρημένες και την ελαχιστοποίηση, έτσι, του αντίστοιχου σφάλματος.

Πιο συγκεκριμένα, μελετάται η λεκάνη απορροής του Ευήνου, ανάντη του ομώνυμου φράγματος. Η λεκάνη έχει συνολική έκταση 351.9 km² και είναι πολύτιμη για το υδροδοτικό σύστημα της Αττικής. Επιδιώκεται η κατάστρωση και η βαθμονόμηση ενός αδιαμέριστου μοντέλου μηνιαίου υδατικού ισοζυγίου λαμβάνοντας ως δεδομένες τις μηνιαίες χρονοσειρές βροχόπτωσης, δυνητικής εξατμισοδιαπνοής και παρατηρημένης απορροής, μήκους 50 ετών (από τον Οκτώβριο του 1970 μέχρι το Σεπτέμβριο του 2020), που έχουν προκύψει από μετρήσεις.

Όπως φαίνεται και στο ακόλουθο Σχήμα 8-1, το υδρολογικό μοντέλο εκτιμά την απορροή της λεκάνης με εφαρμογή διαδοχικών μετασχηματισμών της βροχόπτωσης, μέσω ενός συστήματος διασυνδεδεμένων δεξαμενών. Όλες οι μεταβλητές εκφράζονται σε ισοδύναμα ύψη νερού, αναφέρονται στη συνολική έκταση της λεκάνης και δίνονται σε μηνιαίο χρονικό βήμα.



Σχήμα 8-1: Εννοιολογική απεικόνιση των διεργασιών και παραμέτρων του υδρολογικού μοντέλου (Πηγή: Efstratiadis, A., et. al., 2021. Improving model identifiability by embedding stochastic simulation within hydrological calibration.)

Μεταβλητές εισόδου του μοντέλου είναι η επιφανειακή βροχόπτωση P και η δυνητική εξαποδιαπνοή PET . Έξοδοι του μοντέλου είναι:

- το απόθεμα εδαφικής υγρασίας S
- το απόθεμα της δεξαμενής υπόγειου νερού G
- η πραγματική εξαποδιαπνοή E , που επιμερίζεται σε άμεση ET_D , και εδαφική ET_S , δηλαδή $E = ET_D + ET_S$
- η συνολική απορροή Q , που επιμερίζεται σε άμεση Q_D , επιφανειακή Q_S και βασική Q_B , δηλαδή $Q = Q_D + Q_S + Q_B$

Οι παράμετροι του μοντέλου είναι τέσσερις:

- Το ποσοστό αδιαπέρατης επιφάνειας ν , που εκφράζει το τμήμα της βροχόπτωσης που απορρέει επιφανειακά, χωρίς να διεισδύσει στο έδαφος.

- Η χωρητικότητα της δεξαμενής εδαφικής υγρασίας K , που εκφράζει τη μέγιστη αποθηκευτική ικανότητα της ακόρεστης ζώνης του εδάφους.
- Ο συντελεστής στείρευσης εδαφικής υγρασίας κ , που εκφράζει το ποσοστό του νερού που διακινείται από τη δεξαμενή εδαφικής υγρασίας προς τη δεξαμενή υπόγειου νερού, δηλαδή το νερό που ρέει από την ακόρεστη προς την κορεσμένη ζώνη του εδάφους (κατείσδυση).
- Ο συντελεστής στείρευσης υπόγειου νερού λ , που εκφράζει το ποσοστό του υπόγειου νερού που απορρέει ως βασική απορροή στο υδρογραφικό δίκτυο.

Αρχικές συνθήκες του μοντέλου είναι οι τιμές αποθήκευσης στις δύο δεξαμενές. Για απλούστευση της διαδικασίας, οι τιμές αυτές λαμβάνονται μηδενικές, δηλαδή $S(0) = G(0) = 0$.

Σε κάθε μήνα t θεωρούνται γνωστά η βροχόπτωση $P(t)$, η δυνητική εξατμισοδιαπνοή $PET(t)$, καθώς και η τρέχουσα αποθήκευση εδαφικής υγρασίας $S(t - 1)$ και υπόγειου νερού $G(t - 1)$.

Αρχικά, ένα ποσοστό ν της βροχόπτωσης μετατρέπεται σε άμεση απορροή, $Q_D(t)$, ενώ η περίσσεια διαφεύγει ως άμεση εξατμισοδιαπνοή $ET_D(t)$.

Η ποσότητα νερού που απομένει, διηθείται στη δεξαμενή εδαφικής υγρασίας, η οποία αντιπροσωπεύει την ακόρεστη ζώνη του εδάφους, και προστίθεται στην τρέχουσα αποθήκευση $S(t - 1)$. Ακολούθως, μέρος της αποθηκευμένης υγρασίας διαφεύγει ως εδαφική εξατμισοδιαπνοή $E_S(t)$, ενώ ένα άλλο μέρος διαφεύγει ως κατείσδυση προς τη δεξαμενή συγκέντρωσης υπόγειου νερού, που αντιπροσωπεύει τον υδροφορέα. Η εδαφική εξατμισοδιαπνοή θεωρείται συνάρτηση του ποσοστού πλήρωσης της δεξαμενής υγρασίας, και εκτιμάται μέσω της εμπειρικής σχέσης του Thornthwaite:

$$E_S(t) = S(t) \left[1 - e^{-\frac{\Delta PET(t)}{K}} \right] \quad (8.4)$$

όπου $\Delta PET(t) = PET(t) - ET_D(t)$ είναι το έλλειμμα εξατμισοδιαπνοής, που καλύπτεται μέσω της εδαφικής υγρασίας.

Η κατείσδυση προς τη δεξαμενή υπόγειου νερού δίνεται ως ποσοστό κ (συντελεστής στείρευσης) της αποθήκευσης, δηλαδή $PERC(t) = \kappa S(t)$. Τέλος, εφόσον έχει κορεστεί το έδαφος, η περίσσεια $Q_S(t)$, δηλαδή η υπερχείλιση της δεξαμενής εδαφικής υγρασίας, απορρέει επιφανειακά.

Η δεξαμενή υπόγειου νερού τροφοδοτείται από την κατείσδυση $PERC(t)$, η οποία προστίθεται στην τρέχουσα αποθήκευση $G(t - 1)$. Από την υπόψη δεξαμενή, που δεν έχει όριο χωρητικότητας, ποσοστό λ του αποθηκευμένου υπόγειου νερού απορρέει ως βασική απορροή $Q_B(t)$.

8.3. Χρήση συνθετικών χρονοσειρών για αποδοτικότερη βαθμονόμηση των παραμέτρων

Όπως ειπώθηκε στο προηγούμενο υποκεφάλαιο, για τη λεκάνη απορροής του Ευήνου λαμβάνονται ως δεδομένες οι μηνιαίες χρονοσειρές βροχόπτωσης, δυνητικής εξατμισοδιαπνοής και παρατηρημένης απορροής συνολικού μήκους 50 ετών, οι οποίες έχουν προκύψει από μετρήσεις. Για να βρεθούν οι τιμές των μεταβλητών ελέγχου του συστήματος και να επιλυθεί το πρόβλημα της βαθμονόμησης, η συνήθης τακτική, που εφαρμοζόταν εξ ολοκλήρου παλιότερα, περιλαμβάνει τον διαχωρισμό του συνολικού μήκους της κάθε διαθέσιμης ιστορικής χρονοσειράς σε δύο υποσύνολα. Το πρώτο υποσύνολο χρησιμοποιείται για το καλιμπράρισμα του μοντέλου και το δεύτερο υποσύνολο για έλεγχο και επαλήθευση. Πιο αναλυτικά, για να ολοκληρωθεί η διαδικασία της βαθμονόμησης του μοντέλου, ο αλγόριθμος βελτιστοποίησης που θα χρησιμοποιηθεί, έχει ως στοχική συνάρτηση κάποιο κριτήριο καλής προσαρμογής, όπως το NSE, το οποίο προκύπτει από το πρώτο τμήμα του συνολικού μήκους της περιόδου βαθμονόμησης του μοντέλου. Έτσι, οι τελικές τιμές των παραμέτρων του συστήματος, θα προκύπτουν έπειτα από την ολοκλήρωση της βελτιστοποίησης στο πρώτο αυτό υποσύνολο. Ακολούθως, για να ελεγχθεί η εγκυρότητα των παραμέτρων που προέκυψαν, χρησιμοποιείται το δεύτερο υποσύνολο. Δηλαδή, αφού ολοκληρωθεί η διαδικασία της βελτιστοποίησης, με γνωστές, πλέον, τις τιμές των παραμέτρων, εκτελείται ακόμα μια φορά το μοντέλο υδατικού ισοζυγίου πάνω στο τμήμα του δείγματος που δεν έχει χρησιμοποιηθεί κατά την βελτιστοποίηση και υπολογίζεται η τελική τιμή του δείκτη NSE. Αυτή η τελική τιμή του δείκτη, καθορίζει και το πόσο επιτυχές ήταν το καλιμπράρισμα του μοντέλου που προηγήθηκε.

Παρόλα αυτά, η παραπάνω τακτική παρουσιάζει ορισμένα μειονεκτήματα. Πιο συγκεκριμένα, η αποτελεσματικότητα τέτοιων προσεγγίσεων καθορίζεται πλήρως από το μήκος της ιστορικής χρονοσειράς και το πόσο καλά αυτή αναπαριστά τη συνολική εικόνα του προβλήματος. Το γεγονός ότι απαιτείται ένα τμήμα των δεδομένων να παραμείνει εκτός διαδικασίας βελτιστοποίησης, προκειμένου να αξιολογηθεί στο τμήμα του ελέγχου, οδηγεί συχνά στην απώλεια σημαντικής πληροφορίας, η οποία θα έπρεπε να συμπεριληφθεί κανονικά για την καλή βαθμονόμηση των παραμέτρων και έτσι δεν επιτυγχάνονται υψηλές επιδόσεις κατά τη διαδικασία ελέγχου της εγκυρότητας των αποτελεσμάτων. Πέραν αυτών, ο τελικός υπολογισμός των παραμέτρων του συστήματος και ο έλεγχος τους, πραγματοποιείται εξ ολοκλήρου σε δεδομένα που έχουν παρατηρηθεί στο παρελθόν. Συνεπώς, ακόμα και αν το μοντέλο εμφανίζει πολύ υψηλές επιδόσεις κατά τη διαδικασία του ελέγχου, είναι άγνωστο το αν θα συμπεριφέρεται εξίσου καλά στις μελλοντικές χρονοσειρές, οι οποίες όπως είναι φυσικό, θα παρουσιάζουν διακυμάνσεις συγκριτικά με τις παρελθοντικές. Ιδίως σε περιπτώσεις, όπου δεν υπάρχουν επαρκή διαθέσιμα δεδομένα ιστορικής χρονοσειράς πολλών δεκαετιών, αυτές οι προσεγγίσεις εμφανίζουν αδυναμία να περιγράψουν ικανοποιητικά το πρόβλημα και να παρέχουν έγκυρα αποτελέσματα. Τέλος, συχνά εμφανίζονται και περιπτώσεις, στις οποίες δεν ταυτίζονται τα συνολικά διαθέσιμα ιστορικά δεδομένα εισόδου (πχ. βροχόπτωση, εξατμισοδιαπνοή) με τα αντίστοιχα

εξόδου (πχ. απορροή) και αυτό οδηγεί στην απαίτηση να αγνοείται το μη κοινό τμήμα των δεδομένων, χάνοντας έτσι χρήσιμη πληροφορία.

Λύση σε όλα τα παραπάνω ζητήματα, έρχονται να δώσουν τα στοχαστικά μοντέλα και η παραγωγή συνθετικών χρονοσειρών. Σε αυτές τις προσεγγίσεις, η συνολική διαθέσιμη ιστορική πληροφορία χρησιμοποιείται για να παραχθούν συνθετικές χρονοσειρές τεράστιου μήκους (πχ. χιλιετίες) με τη βοήθεια στοχαστικών μοντέλων. Οι συνθετικές χρονοσειρές χρησιμοποιούνται κατά το τμήμα της βαθμονόμησης του μοντέλου και υπολογισμού των παραμέτρων, ενώ η συνολική ιστορική χρονοσειρά αξιοποιείται για τον έλεγχο εγκυρότητας των αποτελεσμάτων. Με τον τρόπο αυτό, παύει, πλέον, να υφίσταται το πρόβλημα διαχωρισμού των συνολικών δεδομένων σε δύο υποσύνολα και μη αξιοποίησης του ενός υποσυνόλου κατά το στάδιο της βαθμονόμησης. Πέραν αυτού, σε περιπτώσεις, όπου δεν ταυτίζονται τα συνολικά διαθέσιμα ιστορικά δεδομένα εισόδου με τα αντίστοιχα εξόδου δεν απαιτείται να αγνοηθεί το μη κοινό τμήμα των δεδομένων και έτσι αξιοποιείται όλη η διαθέσιμη ιστορική πληροφορία για την παραγωγή των καλύτερων δυνατών αποτελεσμάτων. Επιπλέον, αν για την παραγωγή των συνθετικών χρονοσειρών χρησιμοποιηθεί ένα καλά σχεδιασμένο στοχαστικό μοντέλο, τότε τα συνθετικά παραγόμενα δεδομένα, εκτός του ότι θα έχουν πλήρη στατιστική ομοιότητα με τα ιστορικά δεδομένα και πολύ μεγαλύτερο μήκος από αυτά, θα μπορούν να ενσωματώνουν στον πυρήνα τους και το στοιχείο της τυχαιότητας. Αυτό εξασφαλίζει ότι θα γίνεται μια πολύ έγκυρη αποτύπωση της πραγματικής κατάστασης και θα λαμβάνονται υπόψη τα πιθανά ακραία σενάρια, που πιθανότατα δεν θα μπορούσαν να αναπαρασταθούν μέσω της ιστορικής χρονοσειράς.

Με γνώμονα τα παραπάνω, για την επίλυση του προβλήματος βαθμονόμησης των παραμέτρων, χρησιμοποιείται η τακτική παραγωγής συνθετικών χρονοσειρών. Αυτή η παραγωγή γίνεται μέσω ενός δημοφιλούς στοχαστικού μοντέλου με αποδεδειγμένη αποτελεσματικότητα, του λεγόμενου anySim πακέτου (Tsoukalas, I., et al., 2020). Πληροφορίες σχετικές με το πακέτο **anySim** μπορούν να βρεθούν στις σχετικές εργασίες των δημιουργών του:

➤ <https://www.itia.ntua.gr/en/softinfo/33/>

8.4. Επίλυση του προβλήματος βαθμονόμησης και σύγκριση των αποτελεσμάτων των αλγορίθμων βελτιστοποίησης

Το υπό μελέτη πρόβλημα αναφέρεται στη λεκάνη απορροής του Ευήνου. Για ένα συνολικό χρονικό διάστημα 50 ετών είναι γνωστές οι μηνιαίες χρονοσειρές βροχόπτωσης, δυνητικής εξατμισοδιαπνοής και παρατηρημένης απορροής, οι οποίες έχουν προκύψει από μετρήσεις. Το πρόβλημα περιγράφεται από ένα τυπικό μοντέλο υδατικού ισοζυγίου, όπως αναλύθηκε στο [Υποκεφάλαιο 8.2](#) και αποτελείται από ένα σύνολο 4 μεταβλητών ελέγχου: 1) του ποσοστού αδιαπέρατης επιφάνειας v , 2) της χωρητικότητας της δεξαμενής εδαφικής υγρασίας K , 3) του συντελεστή στείρευσης

εδαφικής υγρασίας κ και 4) του συντελεστή στείρευσης υπόγειου νερού λ . Οι παράμετροι ν , κ , λ παίρνουν τιμές στο εύρος $[0, 1]$, ενώ το K στο $[0, 700 \text{ mm}]$.

Από το σύνολο των διαθέσιμων ιστορικών δεδομένων μήκους 50 ετών, με τη βοήθεια του στοχαστικού μοντέλου anySim, παράγονται συνθετικές χρονοσειρές για την βροχόπτωση, τη δυνητική εξατμισοδιαπνοή και την παρατηρημένη απορροή, μήκους 2048 ετών. Το συνολικό μήκος των συνθετικών χρονοσειρών αξιοποιείται κατά το στάδιο της βαθμονόμησης, δηλαδή για την εύρεση των τελικών τιμών των παραμέτρων του μοντέλου, ενώ το συνολικό μήκος των ιστορικών χρονοσειρών για τον έλεγχο εγκυρότητας των αποτελεσμάτων. Οι ιστορικές και συνθετικές χρονοσειρές που χρησιμοποιούνται, είναι διαθέσιμες στους ακόλουθους συνδέσμους:

- https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/historical_time_series.csv
- https://github.com/spyrostsat/Global-Optimization-Algorithms/blob/main/synthetic_time_series.csv

Ως κριτήριο καλής προσαρμογής, επιλέγεται ο συντελεστής Nash-Sutcliffe, ο οποίος θα αποτελεί επομένως και την στοχική συνάρτηση του προβλήματος, δηλαδή $f(x) \equiv \text{NSE}$. Πιο συγκεκριμένα, ο αλγόριθμος βελτιστοποίησης που θα χρησιμοποιηθεί από το μοντέλο, θα βελτιστοποιεί τον δείκτη NSE, που προκύπτει από τη συνολική συνθετική χρονοσειρά των δεδομένων, μήκους 2048 ετών. Κατά την ολοκλήρωση αυτής της διαδικασίας, θα προκύπτουν οι τελικές τιμές για τις μεταβλητές ελέγχου του προβλήματος. Έπειτα, θα εκτελείται ένα ακόμα υδατικό ισοζύγιο με τις γνωστές πλέον τιμές των παραμέτρων του μοντέλου πάνω στο σύνολο της ιστορικής χρονοσειράς, μήκους 50 ετών, και θα καταγράφεται η τιμή του δείκτη NSE, που προκύπτει από αυτό το τελευταίο υδατικό ισοζύγιο. Αυτή η τιμή θα αναπαριστά και την επίδοση του αλγορίθμου πάνω στο πρόβλημα βαθμονόμησης.

Το πρόβλημα θα επιλυθεί μέσω 3 διαφορετικών αλγορίθμων βελτιστοποίησης, προκειμένου να γίνει σύγκριση της αποτελεσματικότητας του καθενός. Αυτοί είναι οι εξής:

- 1) Ο εξελικτικός αλγόριθμος ανόπτησης – απλόκου (EAS)
- 2) Ο εξελικτικός αλγόριθμος ανόπτησης – απλόκου εμπλουτισμένος με υποκατάστατα μοντέλα (SEEAS)
- 3) Ο νέος πρωτότυπος αλγόριθμος, AMSEEAS

Για την καλύτερη διερεύνηση του προβλήματος, μελετώνται 3 διαφορετικές περιπτώσεις, στις οποίες μεταβάλλεται ο μέγιστος επιτρεπτός αριθμός υπολογισμών της στοχικής συνάρτησης, σε $\text{MFE}=100$ ή $\text{MFE}=500$ ή $\text{MFE}=1000$ υπολογισμοί. Σε κάθε περίπτωση, για τη συλλογή αντιπροσωπευτικού δείγματος, πραγματοποιούνται

30 μεμονωμένες εκτελέσεις και σε κάθε μια από αυτές, καταγράφεται η καλύτερη τιμή του δείκτη NSE στο συνολικό μήκος της ιστορικής χρονοσειράς.

Για να είναι δίκαια η σύγκριση, σε όλους τους αλγορίθμους χρησιμοποιείται το ίδιο μέγεθος πληθυσμού ($m=10$) και η ίδια μέθοδος για την παραγωγή των αρχικών τυχαίων λύσεων της 1^{ης} γενιάς (LHS). Επιπλέον, τοποθετούνται οι προκαθορισμένες τιμές για όλες τις παραμέτρους/υπερπαραμέτρους των αλγορίθμων, όπως αυτές προτείνονται στα αντίστοιχα εγχειρίδια χρήσης.

Μετά την ολοκλήρωση όλων των εκτελέσεων, υπολογίζονται τα στατιστικά χαρακτηριστικά των δειγμάτων (μέση τιμή, τυπική απόκλιση και διάμεσος) και δημιουργείται ένας συγκριτικός πίνακας για την αξιολόγηση των επιδόσεων των αλγορίθμων.

Ο τελικός συγκριτικός πίνακας, που προέκυψε από αυτή την ανάλυση, είναι ο ακόλουθος:

Mean, standard deviation and median value of best Nash-Sutcliffe efficiency (NSE) metrics found by the algorithms in the calibration problem									
MFE	EAS			SEEAS			AMSEEAS		
	MEAN	STDEV	MEDIAN	MEAN	STDEV	MEDIAN	MEAN	STDEV	MEDIAN
100	0.764	0.007	0.762	0.762	0.007	0.761	0.761	0.006	0.760
500	0.770	0.007	0.775	0.771	0.006	0.774	0.775	0.003	0.776
1000	0.773	0.006	0.776	0.775	0.003	0.776	0.776	0.000	0.776

Πίνακας 8-1: Μέση τιμή, τυπική απόκλιση και διάμεσος των καλύτερων τιμών του δείκτη NSE στο πρόβλημα βαθμονόμησης των παραμέτρων υδρολογικού μοντέλου για τους αλγορίθμους EAS, SEEAS και AMSEEAS

Όπως φαίνεται και μέσω του Πίνακα 8-1, αν χρησιμοποιηθεί ως μέτρο επίδοσης η μέση τιμή του κάθε δείγματος, ο AMSEEAS κυριαρχεί έναντι των υπολοίπων σε 2 / 3 περιπτώσεις, ενώ στην 3^η περίπτωση την 1^η θέση καταλαμβάνει ο EAS. Ο SEEAS δεν κερδίζει σε κανένα πρόβλημα από τα 3. Αντίθετα, αν χρησιμοποιηθεί ως μέτρο επίδοσης η διάμεσος του κάθε δείγματος, ο AMSEEAS και ο EAS κυριαρχούν σε 1 / 3 περιπτώσεις, ενώ στην 3^η περίπτωση παρατηρείται η ίδια τιμή και στους 3 αλγορίθμους. Και εδώ ο SEEAS δεν κυριαρχεί σε κανένα πρόβλημα από τα 3. Παρόλα αυτά, είναι εμφανές ότι και οι 3 αλγόριθμοι βρίσκουν πολύ κοντινά αποτελέσματα σε όλα τα εξεταζόμενα προβλήματα.

Κεφάλαιο 9^ο

9 Συμπεράσματα και προτάσεις για μελλοντική έρευνα

9.1. Συμπεράσματα

Τα κύρια συμπεράσματα που προκύπτουν κατά την ολοκλήρωση αυτής της εργασίας είναι τα ακόλουθα:

- Τα προβλήματα του τομέα των υδατικών πόρων περιλαμβάνουν πολλών πηγών αβεβαιότητες, οι οποίες δημιουργούν με τη σειρά τους περίπλοκες, διαταραγμένες και μη κυρτές επιφάνειες απόκρισης των στοχικών συναρτήσεων. Αυτό έχει ως συνέπεια, την αδυναμία των αλγορίθμων βελτιστοποίησης να εντοπίσουν με ευκολία τα ολικά ακρότατα και να απαιτείται να καλέσουν την εκάστοτε αντικειμενική συνάρτηση πολλές χιλιάδες φορές, προτού ολοκληρωθεί με επιτυχία η συγκλητική διαδικασία. Ταυτόχρονα, στα προβλήματα υδατικών πόρων, οι στοχικές συναρτήσεις δεν ορίζονται μέσω ενός απλού μαθηματικού τύπου, αλλά προκύπτουν μετά την εφαρμογή κάποιου μοντέλου προσομοίωσης. Για να ολοκληρωθεί μια μονάχα εκτέλεση αυτού του μοντέλου προσομοίωσης, απαιτείται να μεσολαβήσει κάποιος χρόνος, ο οποίος δεν είναι αμελητέος (πχ. 1 λεπτό). Οι δύο αυτοί λόγοι σε συνδυασμό, καθιστούν ανήμπορους τους τυπικούς αλγορίθμους βελτιστοποίησης να εντοπίσουν με ακρίβεια το ολικό ακρότατο εντός λογικών χρονικών πλαισίων.
- Μια πολύ συνήθης τακτική που εφαρμόζεται στην πράξη για την αντιμετώπιση του ζητήματος αυτού, είναι η χρήση των υποκατάστατων μοντέλων. Τα υποκατάστατα μοντέλα, αποτελούν στην ουσία μοντέλα αυτοεκμάθησης μηχανών, τα οποία ενσωματώνονται στον οποιοδήποτε αλγόριθμο βελτιστοποίησης, για να βοηθήσουν και να επιταχύνουν τη σύγκλιση του.
- Ο εξελικτικός αλγόριθμος ανόπτησης – απλόκου εμπλουτισμένος με υποκατάστατα μοντέλα (SEEAS) είναι ένα παράδειγμα αλγορίθμου βελτιστοποίησης που ενσωματώνει στον πυρήνα του ένα υποκατάστατο μοντέλο και συγκεκριμένα το Cubic RBF with Linear Polynomial Tail. Ο αλγόριθμος αυτός είναι πάρα πολύ αποδοτικός και έχει πολύ μεγάλη επιτυχία, όπως επιβεβαιώνουν και οι σχετικές εργασίες των δημιουργών του.
- Παρόλα αυτά, όπως αποδεικνύεται και στην παρούσα εργασία, δεν είναι ανάγκη να περιορίζεται ο αλγόριθμος στην ενσωμάτωση ενός, μονάχα,

μεταμοντέλου. Είναι εφικτό, πολλά μεταμοντέλα να συνυπάρχουν μαζί στον ίδιο αλγόριθμο και να συνεργάζονται στα πλαίσια μιας ομάδας. Σε περιπτώσεις, όπου κάποιο υποκατάστατο μοντέλο δεν συνεισφέρει αποδοτικά στην ομάδα, κάνει πίσω και επιτρέπει σε άλλα μεταμοντέλα περισσότερο αποτελεσματικά, να αναλάβουν τα ηνία και να καθοδηγήσουν εκείνα την συγκλητική διαδικασία. Με τον τρόπο αυτό, ο αρχικός αλγόριθμος μπορεί να αποκομίζει τα οφέλη πολλών διαφορετικών μοντέλων αυτοεκμάθησης μηχανών και να συγκλίνει όσο το δυνατόν ταχύτερα, από άποψη πλήθους υπολογισμών της στοχικής συνάρτησης, στο ολικό ακρότατο.

- Αυτή ακριβώς είναι και η ιδέα πίσω από τον πρωτότυπο κώδικα που παρουσιάζεται στην εργασία, τον λεγόμενο Adaptive Multi-Surrogate Enhanced Evolutionary Annealing Simplex algorithm (AMSEEAS). Ο αλγόριθμος αυτός, συγκεκριμένα, ενσωματώνει στον πυρήνα του 6 διαφορετικά μοντέλα μηχανικής μάθησης, τα οποία λειτουργούν ως μια ομάδα υποκατάστατων μοντέλων.
- Πράγματι, τα αποτελέσματα της σύγκρισης του AMSEEAS με άλλους δημοφιλείς αλγορίθμους βελτιστοποίησης της διεθνούς βιβλιογραφίας πάνω σε πολλές συναρτήσεις ελέγχου και σε ένα τυπικό πρόβλημα βαθμονόμησης παραμέτρων υδρολογικού μοντέλου, επιβεβαιώνουν τις προοπτικές και την επιτυχία της ιδέας.

9.2. Προτάσεις για μελλοντική έρευνα

Παρόλη την επιτυχία του AMSEEAS, εννοείται ότι υπάρχουν ακόμα πολλά περιθώρια για περαιτέρω βελτίωση της επίδοσης του. Στο υποκεφάλαιο αυτό συνοψίζονται μερικές βασικές ιδέες που θα είχε πολύ ενδιαφέρον να ερευνηθούν σε βάθος και να υλοποιηθούν:

- Στη συγκεκριμένη εκδοχή του AMSEEAS που παρουσιάζεται σε αυτή την εργασία, προκειμένου τα μεταμοντέλα που ενσωματώνονται στον αλγόριθμο, να συνεργάζονται αποδοτικά και αποτελεσματικά στα πλαίσια της ομάδας, χρησιμοποιείται η έννοια της επιβολής ποινών. Έτσι, τα υποκατάστατα μοντέλα που αδυνατούν να κάνουν καλή προσαρμογή στα σημεία του εξωτερικού αρχείου, δέχονται ποινές από τον αλγόριθμο και παραμένουν εκτός υπολογιστικής διαδικασίας για συγκεκριμένο αριθμό επαναληπτικών κύκλων. Αυτό έχει ως στόχο να μην ξοδεύονται άσκοπα υπολογισμοί της στοχικής συνάρτησης και να επιτρέπεται στα πιο αποδοτικά μεταμοντέλα να αναλαμβάνουν εκείνα τα ηνία και να καθοδηγούν τη συγκλητική διαδικασία. Βέβαια, η ιδέα αυτή της χρήσης τιμωριών είναι κάπως ντετερμινιστική, καθώς άπαξ και ένα μεταμοντέλο σε κάποιον επαναληπτικό κύκλο δεν δώσει καλή πρόβλεψη, τότε σίγουρα θα παραμείνει εκτός υπολογιστικής διαδικασίας για μεγάλο αριθμό επαναλήψεων, παρόλο που είναι πιθανό να είναι σε θέση να κάνει και αυτό καλές προσαρμογές στα μελλοντικά σημεία του εξωτερικού αρχείου αρκετά νωρίτερα από την

ολοκλήρωση της τιμωρίας που του έχει επιβληθεί. Μια διαφορετική ιδέα, λοιπόν, για τη αποδοτική συνεργασία των υποκατάστατων μοντέλων στον αλγόριθμο, που δεν περιλαμβάνει τη έννοια των ποινών και καθιστά πολύ πιο στοχαστική τη διαδικασία επιλογής κάποιου μεταμοντέλου σε κάθε επανάληψη του αλγορίθμου είναι η ακόλουθη:

Στον πρώτο επαναληπτικό κύκλο του αλγορίθμου, όλα τα υποκατάστατα μοντέλα που ενσωματώνονται στον κώδικα, έχουν την ίδια πιθανότητα να επιλεγθούν για να κάνουν την 1^η προσαρμογή, δηλαδή στην περίπτωση που υπάρχουν 6 μεταμοντέλα, όπως σε αυτή την εκδοχή του AMSEAS, το καθένα από αυτά έχει πιθανότητα $P = 1 / 6 = 16.67 \%$ να επιλεγθεί. Αν, λοιπόν, το υποκατάστατο μοντέλο που επιλέγεται τελικά, κάνει καλή προσαρμογή στα σημεία του εξωτερικού αρχείου, τότε η πιθανότητα το ίδιο μεταμοντέλο να επιλεγθεί ξανά στην επόμενη επανάληψη, αυξάνεται ανάλογα με το πόσο καλή είναι η πρόβλεψη που κάνει, συγκριτικά με τις υπόλοιπες λύσεις που υπάρχουν στον πληθυσμό της 1^{ης} γενιάς. Επομένως, οι πιθανότητες να επιλεγθούν τα υπόλοιπα 5 μεταμοντέλα στην επόμενη επανάληψη μειώνονται κατάλληλα, ούτως ώστε η συνολική πιθανότητα επιλογής όλων των υποκατάστατων μοντέλων να αθροίζει στη μονάδα. Αντίθετα, αν η πρόβλεψη δεν είναι καλή (αν για παράδειγμα στο σημείο της πρόβλεψης προκύπτει μεγαλύτερη τιμή στοχαστικής συνάρτησης από την αντίστοιχη της χειρότερης λύσης στον πληθυσμό τη δεδομένη στιγμή), τότε η πιθανότητα του συγκεκριμένου μεταμοντέλου να επιλεγθεί στον επόμενο επαναληπτικό κύκλο μειώνεται (ανάλογα με το πόσο κακή ήταν η πρόβλεψη) και των υπολοίπων αυξάνεται, ώστε πάλι η συνολική πιθανότητα να αθροίζει στη μονάδα. Αν η πρόβλεψη που κάνει δεν είναι ούτε καλή ούτε κακή, σύμφωνα με κάποιο συγκεκριμένο όριο, τότε οι πιθανότητες επιλογής του κάθε μεταμοντέλου στην επόμενη επανάληψη παραμένουν οι ίδιες, δηλαδή στο 16.67 %. Με τον τρόπο αυτό, ο αλγόριθμος αποκτάει πολύ περισσότερο στοχαστικό χαρακτήρα και κανένα μεταμοντέλο δεν αποκλείεται από τη συγκλητική διαδικασία, όπως συμβαίνει στο σενάριο χρήσης των ποινών. Η ιδέα αυτή θεωρείται πως μπορεί να εξελίξει σημαντικά τον AMSEAS και θα είχε πολύ ενδιαφέρον να υλοποιηθεί μελλοντικά.

- Επιπλέον ιδέες που θα μπορούσαν να μελετηθούν σε βάθος, αφορούν στη μεταβολή είτε των μεταμοντέλων που ενσωματώνονται στον αλγόριθμο είτε του πλήθους αυτών είτε και της σειράς επιλογής τους. Πιο αναλυτικά, θα μπορούσε να εξεταστεί το ενδεχόμενο να εισαχθούν διαφορετικά υποκατάστατα μοντέλα στον κώδικα αντί για τα 6 συγκεκριμένα που παρουσιάζονται στην εκδοχή αυτή, μιας και υπάρχει πραγματικά πελώρια γκάμα διαφορετικών μοντέλων μηχανικής μάθησης στη διεθνή βιβλιογραφία. Πέρα από αυτό, η όλη ιδέα πίσω από τον AMSEAS, έγκειται στο ότι δεν πρέπει να υπάρχει περιορισμός στο πόσα μεταμοντέλα μπορούν να συνυπάρχουν στον ίδιο αλγόριθμο. Επομένως, η επιλογή του πλήθους των 6 μεταμοντέλων που έγινε, δεν πρέπει να θεωρείται απόλυτη σε καμία περίπτωση και θα μπορούσε να εξεταστεί το ενδεχόμενο εισαγωγής ακόμα περισσότερων υποκατάστατων μοντέλων αυτοεκμάθησης μηχανών. Τέλος, θα μπορούσε να εξεταστεί το αν η σειρά με την οποία επιλέγονται τα υποκατάστατα μοντέλα στον αλγόριθμο παίζει κάποιο ρόλο στη συγκλητική διαδικασία (δηλαδή το γεγονός ότι

στην εκδοχή που παρουσιάζεται επιλέγεται το Polynomial Regression στον 1^ο επαναληπτικό κύκλο, το Support Vector Machine στον 2^ο κοκ.) και εφόσον ισχύει αυτό, να βρεθεί η βέλτιστη σειρά επιλογής για μεγιστοποίηση της επίδοσης.

- Μια τελευταία πρόταση, που θεωρείται ότι μπορεί να έχει καθοριστική σημασία για την περαιτέρω βελτίωση της επίδοσης του AMSEEAS, είναι η μόνιμη απενεργοποίηση του κάθε μεταμοντέλου από τον αλγόριθμο, εφόσον αυτό δεχθεί ένα συγκεκριμένο πλήθος ποινών. Πιο συγκεκριμένα, θα μπορούσαν να ενσωματωθούν στον κώδικα 6 μετρητές, οι οποίοι θα μετρούν το πόσες φορές μεταβαίνει το κάθε υποκατάστατο μοντέλο σε κατάσταση τιμωρίας. Όταν κάποιος μετρητής φτάσει σε ένα συγκεκριμένο όριο, που καθορίζεται από τον προγραμματιστή (πχ. κάποιο μοντέλο να επέλθει 5 φορές συνολικά σε κατάσταση τιμωρίας), τότε αυτό το μοντέλο θα αποκλείεται μόνιμα από τη συγκλητική διαδικασία, μιας και φαίνεται ότι δεν μπορεί να προσφέρει, πλέον, κάποια βοήθεια στη προσέγγιση του ολικού ακροτάτου και, αντιθέτως, σπαταλάει άσκοπα υπολογισμούς της στοχικής συνάρτησης.

Αναφορές

- Andradóttir, S., 1998. *A Review of Simulation Optimization Techniques*.
- Angra, S., Ahuja, S., 2017. *Machine Learning and its Applications: A Review*. IEEE.
- April, J., Glover, F., Kelly, J.P., Laguna, M., 2003. *Simulation-based optimization: practical introduction to simulation optimization*.
- Bors, A.G., 2001. *Introduction of the Radial Basis Function (RBF) Networks*. University of York
- Bozkurt, S., Elibol, G., Gunal, S., Yayan, U., 2015. *A Comparative Study on Machine Learning Algorithms for Indoor Positioning*. IEEE.
- Breiman, L., 2001. *Random Forests*. Kluwer Academic Publishers
- Chengsheng, T., Huacheng, L., Bing, X., 2017. *AdaBoost typical Algorithm and its application research*. Matec Web of Conferences
- Cortes, C., Vapnik, V., 1995. *Support-Vector Networks*. Machine Learning, 20(3) 273-297, Springer
- Dibike, Y.B., Velickov, S., Solomatine, D., Abbott, M.B., 2001. *Model Induction with Support Vector Machines : Introduction and Applications*. Journal of Computing in Civil Engineering 208-216
- Efstratiadis, A., Koutsoyiannis, D., 2001. *Διερεύνηση μεθόδων αναζήτησης ολικού βέλτιστου σε προβλήματα υδατικών πόρων*.
- Efstratiadis, A., Koutsoyiannis, D., 2002. *An evolutionary annealing-simplex algorithm for global optimisation of water resource systems*. Proceedings of the Fifth International Conference on Hydroinformatics, Cardiff, UK, 1423-1428, doi:10.13140/RG.2.1.1038.6162, International Water Association
- Efstratiadis, A., Koutsoyiannis, D., Rozos, E., Nalbantis, I., 2004. *Calibration of a semi-distributed model for conjunctive simulation of surface and groundwater flows*. Hydrological Sciences Journal.
- Efstratiadis, A., Koutsoyiannis, D., 2008. *Fitting hydrological models on multiple responses using the multiobjective evolutionary annealing simplex approach*. Practical hydroinformatics: Computational intelligence and technological developments in water applications, edited by Abraham, R.J., See, L.M., Solomatine, D.P., 259-273, doi:10.1007/978-3-540-79881-1_19, Springer
- Efstratiadis, A., 2008. *Μη γραμμικές μέθοδοι σε πολυκριτηριακά προβλήματα βελτιστοποίησης υδατικών πόρων, με έμφαση στη βαθμονόμηση υδρολογικών μοντέλων*.
- Efstratiadis, A., Koutsoyiannis, D., 2010. *One decade of multiobjective calibration approaches in hydrological modelling: a review*. Hydrol. Sci. J., 55(1), 58–78, doi:10.1080/02626660903526292

- Efstratiadis, A., 2011. *New insights on model evaluation inspired by the stochastic simulation paradigm*. EGU General Assembly 2011, Geophysical Research Abstracts, Vol. 13, Vienna, 1852, European Geosciences Union
- Efstratiadis, A., Nalbantis, I., Koutsoyiannis, D., 2015. *Hydrological modelling of temporally-varying catchments: Facets of change and the value of information*. Hydrol. Sci. J., 60(7-8), 1438–1461, doi:10.1080/02626667.2014.982123
- Efstratiadis, A., Tsoukalas, I., Kossieris, P., 2021. *Improving model identifiability by embedding stochastic simulation within hydrological calibration*.
- Forrester, A.I.J., Sobester, A., Keane, A.J., 2008. *Engineering Design via Surrogate modelling*. John Wiley & Sons Ltd.
- Forrester, A., Keane, A., 2009. *Recent advances in surrogate-based optimization*. Progress in Aerospace Sciences.
- Goel, T., Haftka, R.T., Shyy, W., Queipo, N.V., 2006. *Ensemble of surrogates*. Springer
- Guo, G., Wang, H., Bell, D.A., Bi, Y., Greer, K., 2004. *KNN Model-Based Approach in Classification*.
- Halde, R.R., 2016. *Application of Machine Learning algorithms for betterment in Education system*. International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT).
- Harris, C.R., Millman, K.J., Gommers, R., et.al. 2020. *Array programming with NumPy*
- Henderson, D., Jacobson, S.H., Johnson, A.W., 2006. *The Theory and Practice of Simulated Annealing*.
- Hensman, J., Fusi, N., Lawrence, N.D., 2013. *Gaussian processes for big data*. The University of Sheffield
- Iman, R.L., 1999. *Latin Hypercube Sampling*. John Wiley & Sons, Ltd
- James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. *An Introduction to Statistical Learning*. Springer
- Jones, D.R., Schonlau, M., Welch, W.J., 1998. *Efficient Global Optimization of Expensive Black-Box Functions*. Journal of Global Optimization 13: 455-492
- Kossieris, P., Koutsoyiannis, D., 2013. *Προσαρμογή εξελικτικού αλγορίθμου ανόπτησης-απλόκου για βελτιστοποίηση στοχαστικών στοιχείων συναρτήσεων σε προβλήματα υδατικών πόρων*.
- Kossieris, P., Tsoukalas, I., Makropoulos, C., Savic, D., 2019. *Simulating marginal and dependence behaviour of water demand processes at any fine time scale*. Water, 11(5), 885, doi:10.3390/w11050885
- Kourakos, V., Efstratiadis, A., Tsoukalas, I., 2021. *Can hydrological model identifiability be improved? Stress-testing the concept of stochastic calibration*. EGU General Assembly 2021, online, EGU21-11704, doi:10.5194/egusphere-egu21-11704, European Geosciences Union

- Koutsoyiannis, D., Karavokiros, G., Efstratiadis, A., Mamassis, N., Koukouvinos, A., Christofides, A., 2003. *A decision support system for the management of the water resource system of Athens*. Phys. Chem. Earth, 28(14-15), 599-609, doi:10.1016/S1474-7065(03)00106-2
- Koutsoyiannis, D., 2013. *Hydrology and Change*. Hydrol. Sci. J., 58(6), 1177–1197, doi:10.1080/02626667.2013.804626
- Koutsoyiannis, D., 2020. *Simple stochastic simulation of time irreversible and reversible processes*. Hydrological Sciences Journal, 65 (4), 536–551, doi:10.1080/02626667.2019.1705302
- Ky, V.K., D' Ambrosio, C., Hamadi, Y., Liberti, L., 2016. *Surrogate-based methods for black-box optimization*. Ecole Polytechnique
- Liang, J., Meyerson, E., Hodjat, B., Fink, D., Mutch, K., Miikkulainen, R., 2019. *Evolutionary Neural AutoML for Deep Learning*. Gecco, Prague, Czech Republic
- Nalbantis, I., Efstratiadis, A., Rozos, E., Kopsiafti, M., Koutsoyiannis, D., 2011. *Holistic versus monomeric strategies for hydrological modelling of human-modified hydrosystems*. Hydrol. Earth Syst. Sci., 15, 743-758, doi:10.5194/hess-15-743-2011
- Ostertagova, E., 2012. *Modelling Using Polynomial Regression*. Elsevier Ltd. Selection
- Pan, L., Wu, L., 1998. *A hybrid global optimization method for inverse estimation of hydraulic parameters: annealing-simplex method*. Water Resources Research.
- Pedregosa, F., Varoquaux, G., Michel, V., Gramfort, A., et.al. 2012. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research
- Rios, L.M., Sahinidis, N.V., 2012. *Derivative-free optimization: a review of algorithms and comparison of software implementations*. Springer Science + Business Media, LLC
- Rish, I., 2001. *An Empirical Study of the Naïve Bayes Classifier*. T.J. Watson Research Center
- Rokach, L., Maimon, O., 2005. *Decision Trees*
- Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P., 1989. *Design and Analysis of Computer Experiments*. Institute of Mathematical Statistics 409-423
- Schmidhuber, J., 2014. *Deep Learning in neural networks: An overview*. Elsevier Ltd.
- Shinde, P.P., Shah, S., 2018. *A review of Machine Learning and Deep Learning Applications*. IEEE.
- Shoemaker, C., Müller, J., 2014. *Influence of Ensemble Surrogate Models and Sampling Strategy on the Solution Quality of Algorithms for Computationally Expensive Black-Box Global Optimization Problems*. Journal of Global Optimization.
- Simon, D., 2013. *Evolutionary Optimization Algorithms : Biologically Inspired and Population-Based Approaches to Computer Intelligence*. Wiley

Singer, S., Singer, S., 2004. *Efficient Implementation of the Nelder-Mead Search Algorithm*. WILEY-VCH Verlag GmbH & Co. KGaA

Sloss, A.N., Gustafson, S., 2019. *2019 Evolutionary Algorithms Review*.

Sun, S., Cao, Z., Zhu, H., Zhao, J., 2019. *A Survey of Optimization Methods from a Machine Learning Perspective*.

Tekin, E., Sabuncuoglu, I., 2004. *Simulation optimization: A comprehensive review on theory and applications*. IIE

Tsoukalas, I., Kossieris, P., Efstratiadis, A., Makropoulos, C., 2016. *Surrogate-enhanced evolutionary annealing simplex algorithm for effective and efficient optimization of water resources problems on a budget*. Environmental Modelling and Software, 77, 122–142, doi:10.1016/j.envsoft.2015.12.008

Tsoukalas, I., Papalexiou, S.M., Efstratiadis, A., Makropoulos, C., 2018. *A cautionary note on the reproduction of dependencies through linear stochastic models with non-Gaussian white noise*. Water, 10(6), 771, doi:10.3390/w10060771

Tsoukalas, I., Efstratiadis, A., Makropoulos, C., 2018. *Stochastic periodic autoregressive to anything (SPARTA): Modelling and simulation of cyclostationary processes with arbitrary marginal distributions*. Water Resour. Res., 54 (1), 161–185, WRCR23047, doi:10.1002/2017WR021394

Tsoukalas, I., Efstratiadis, A., Makropoulos, C., 2019. *Building a puzzle to solve a riddle: A multi-scale disaggregation approach for multivariate stochastic processes with any marginal distribution and correlation structure*. J. Hydrol., 575, 354–380, doi:10.1016/j.jhydrol.2019.05.017

Tsoukalas, I., Kossieris, P., Makropoulos, C., 2020. *Simulation of non-Gaussian correlated random variables, stochastic processes and random fields: Introducing the anySim R-Package for environmental applications and beyond*. Water, 12(6), 1645, doi:10.3390/w12061645

Uyanik, G.K., Güler, N., 2013. *A study on multiple linear regression analysis*. Elsevier Ltd.

Παράρτημα Α: Συναρτήσεις Ελέγχου

- OF1 - Sphere Function:

$$f(x) = \sum_{i=1}^n x_i^2, \text{ με } x_i \in [-5.12, 5.12] \text{ για } i = 1, 2, \dots, n$$

με ελάχιστο το $f(x^*) = 0$, όπου $x^* = (0, 0, \dots, 0)$

- OF2 – Ackley Function:

$$f(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)} + 20 + e,$$

με $x_i \in [-32.768, 32.768]$ για $i = 1, 2, \dots, n$

με ελάχιστο το $f(x^*) = 0$, όπου $x^* = (0, 0, \dots, 0)$

- OF3 – Griewank Function:

$$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \left(\prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) + 1$$

με $x_i \in [-600, 600]$ για $i = 1, 2, \dots, n$

με ελάχιστο το $f(x^*) = 0$, όπου $x^* = (0, 0, \dots, 0)$

- OF4 – Zakharov Function:

$$f(x) = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^4$$

με $x_i \in [-5, 10]$ για $i = 1, 2, \dots, n$

με ελάχιστο το $f(x^*) = 0$, όπου $x^* = (0, 0, \dots, 0)$

- OF5 – Rastrigin Function:

$$f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$$

$$\text{με } x_i \in [-5.12, 5.12] \text{ για } i = 1, 2, \dots, n$$

$$\text{με ελάχιστο το } f(x^*) = 0, \text{ όπου } x^* = (0, 0, \dots, 0)$$

- OF6 – Levy Function:

$$f(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{n-1} \{ (\omega_i - 1)^2 [1 + 10 \sin^2(\pi\omega_i + 1)] \} + (\omega_n - 1)^2 [1 + \sin^2(2\pi\omega_n)]$$

$$\text{όπου } \omega_i = 1 + \frac{x_i - 1}{4}, \text{ για } i = 1, 2, \dots, n$$

$$\text{με } x_i \in [-10, 10] \text{ για } i = 1, 2, \dots, n$$

$$\text{με ελάχιστο το } f(x^*) = 0, \text{ όπου } x^* = (1, 1, \dots, 1)$$

Πέρα, όμως, από αυτή τη σουίτα των 6 συναρτήσεων, σύμφωνα με τις οποίες πραγματοποιήθηκαν όλες οι συγκρίσεις του [Κεφαλαίου 7](#), ανεπίσημα, για περαιτέρω διερεύνηση, χρησιμοποιήθηκαν μερικές ακόμα συναρτήσεις ελέγχου οι οποίες παρουσιάζονται εδώ:

- Rosenbrock Function:

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$$

- Goldstein-Price Function:

$$f(x, y) = [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)][30 + (2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$$

- Styblinski-Tang Function:

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

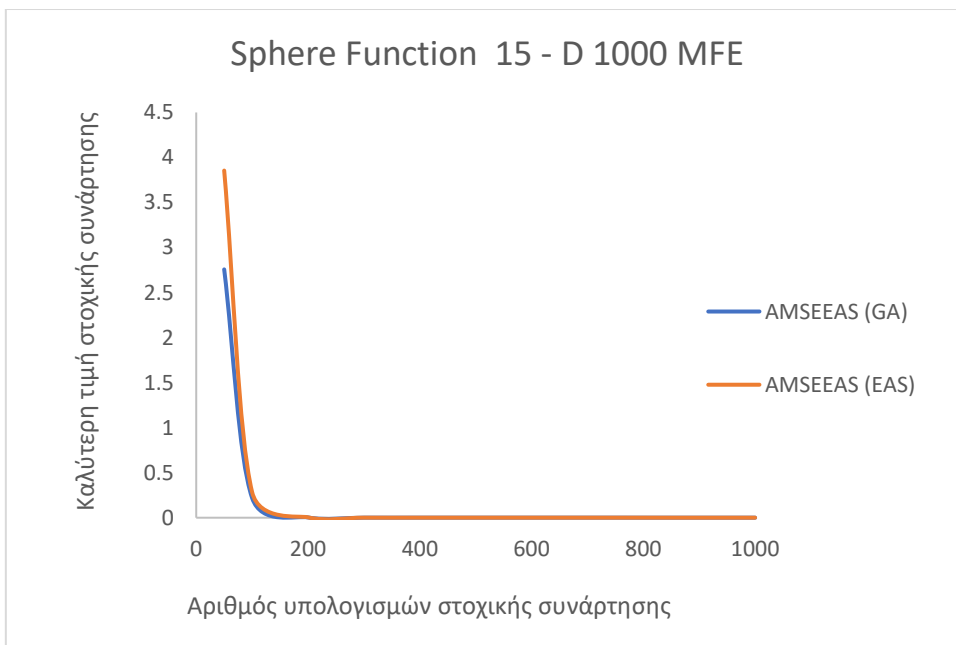
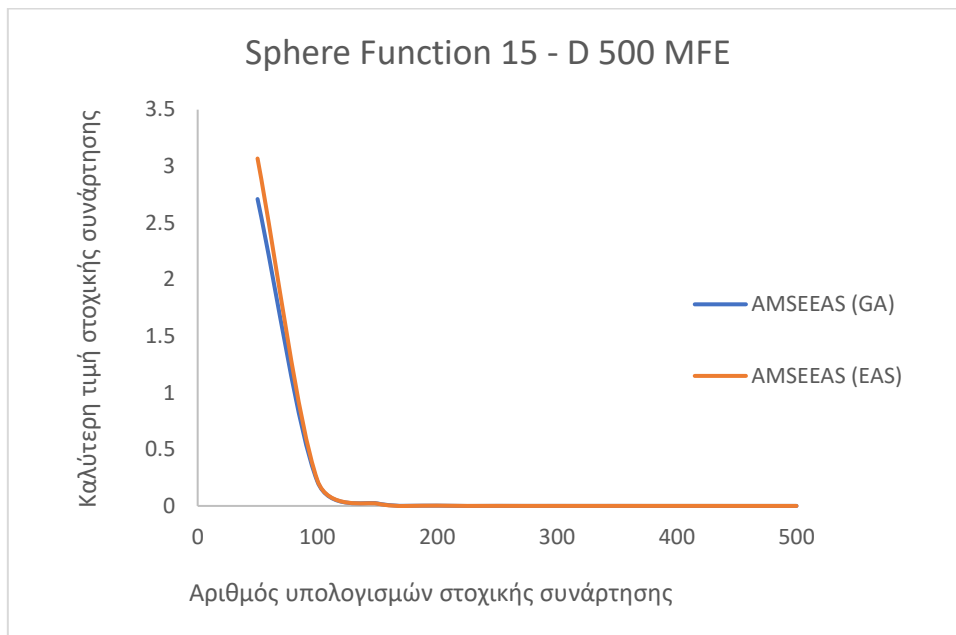
- Hozaki Function:

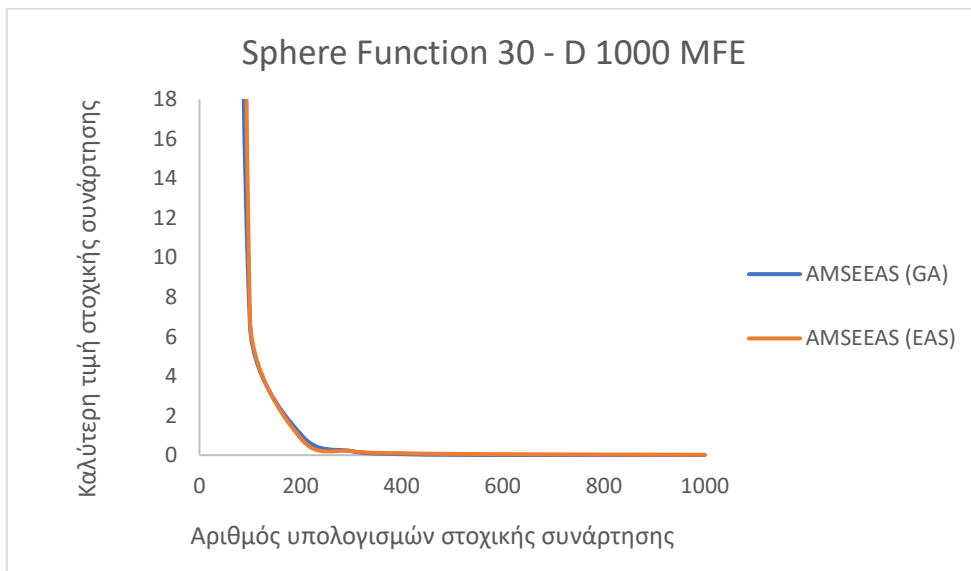
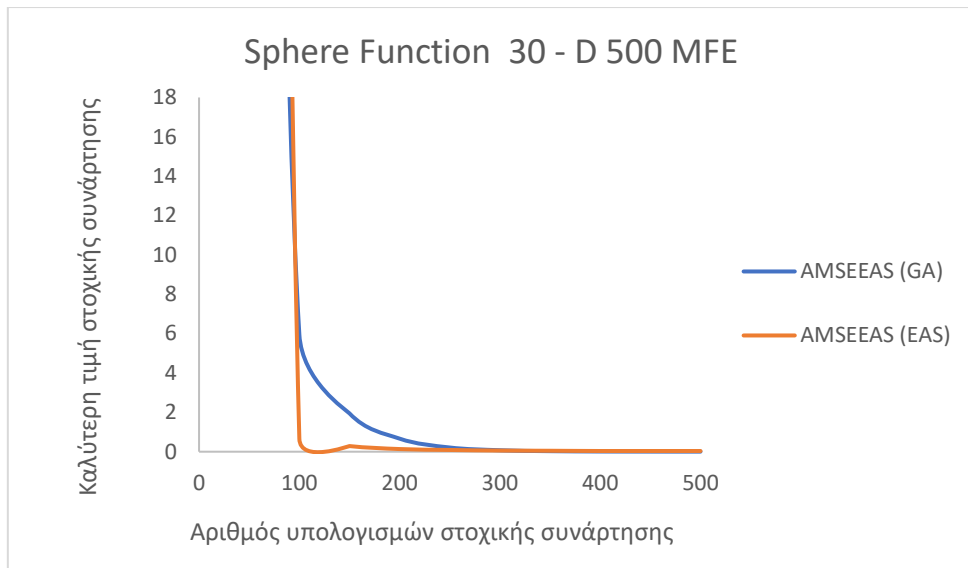
$$f(x, y) = (1 - 8x + 7x^2 - 2.33x^3 + 0.25x^4)y^2e^{-y}$$

- Michalewisc Function:

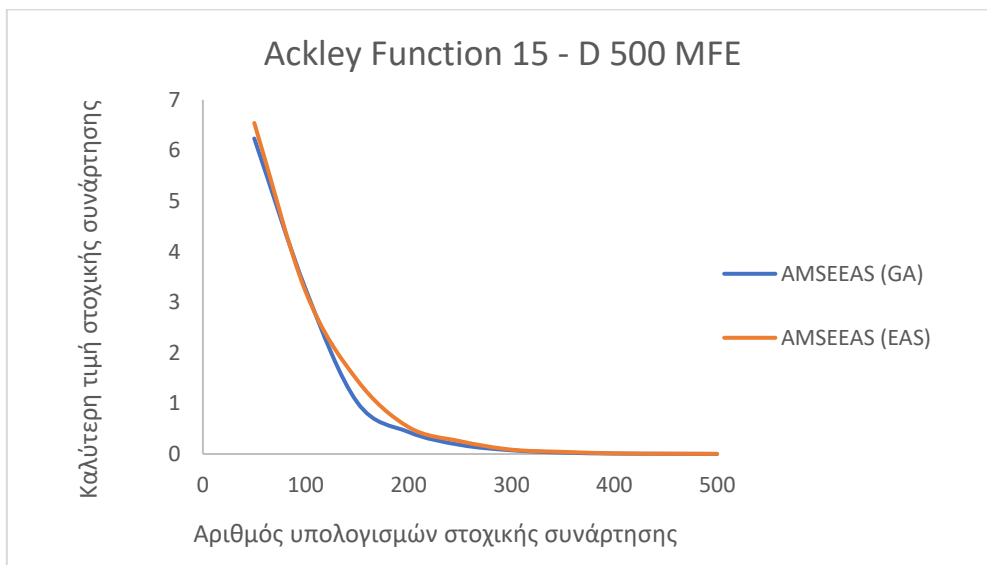
$$f(x, y) = -21.5 + x\sin(4\pi x) + y\sin(20\pi y)$$

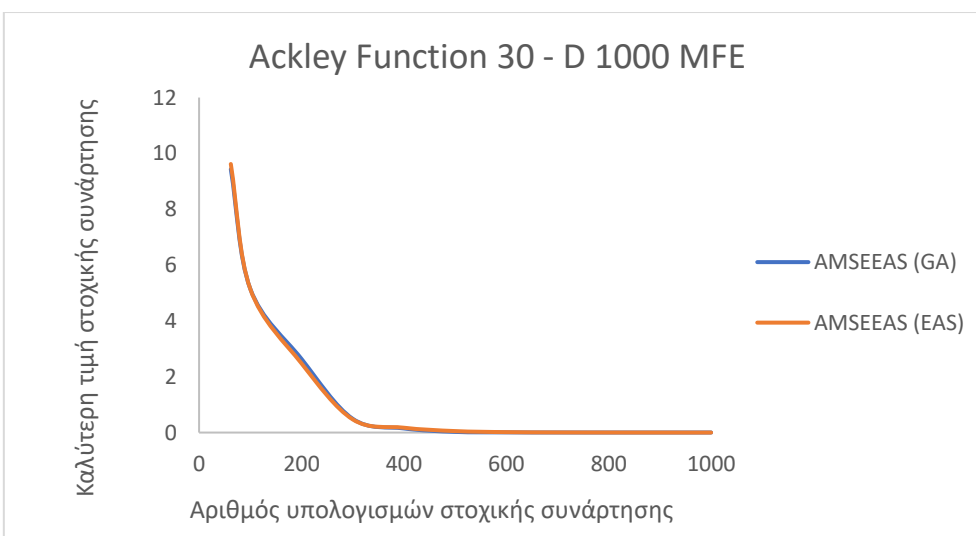
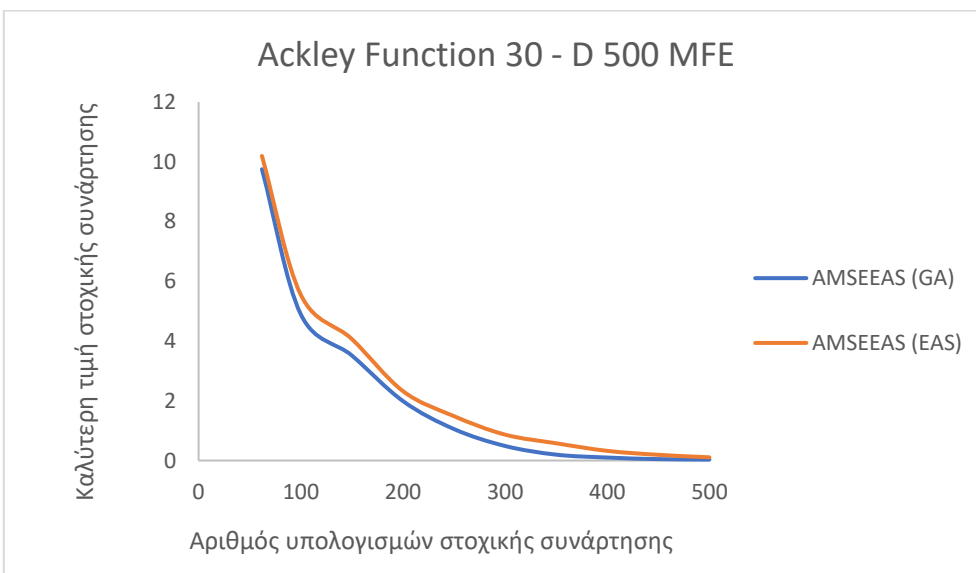
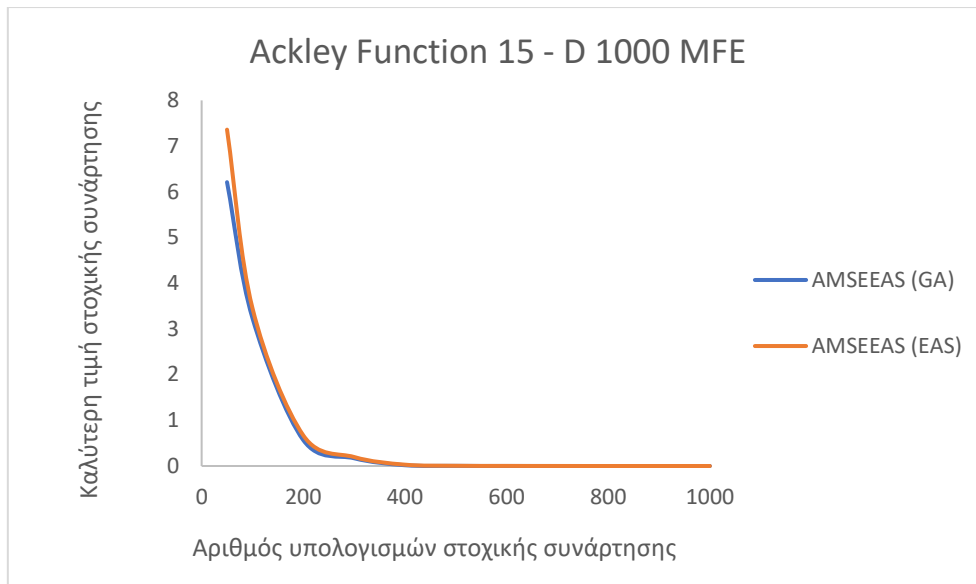
Παράρτημα Β: Καμπύλες σύγκλισης των 2 εκδοχών του AMSEEEAS στις 6 συναρτήσεις ελέγχου



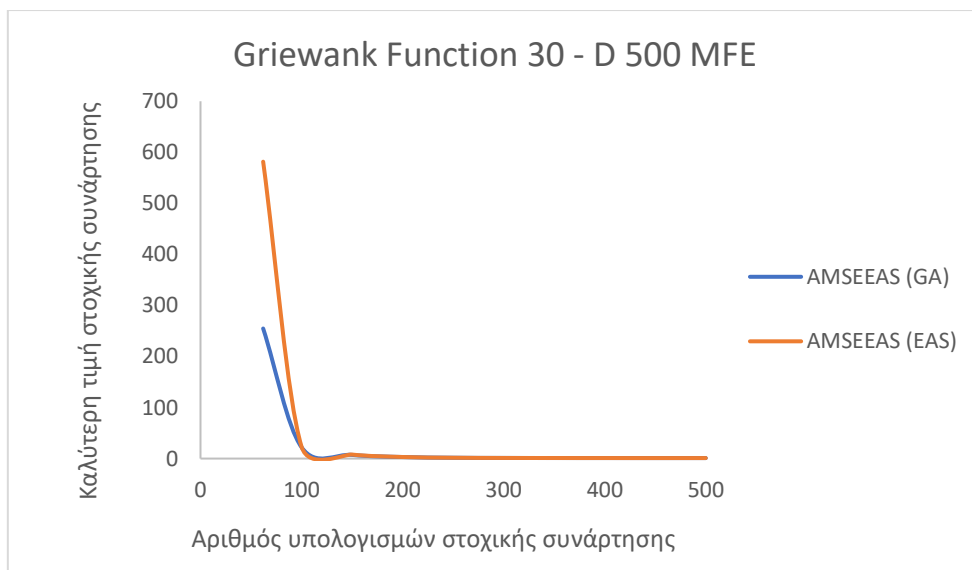
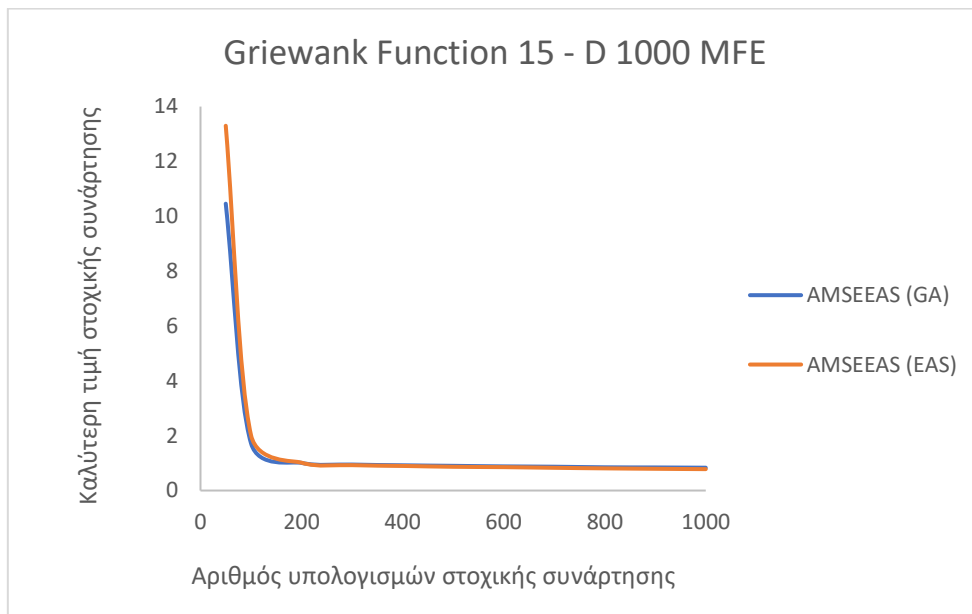
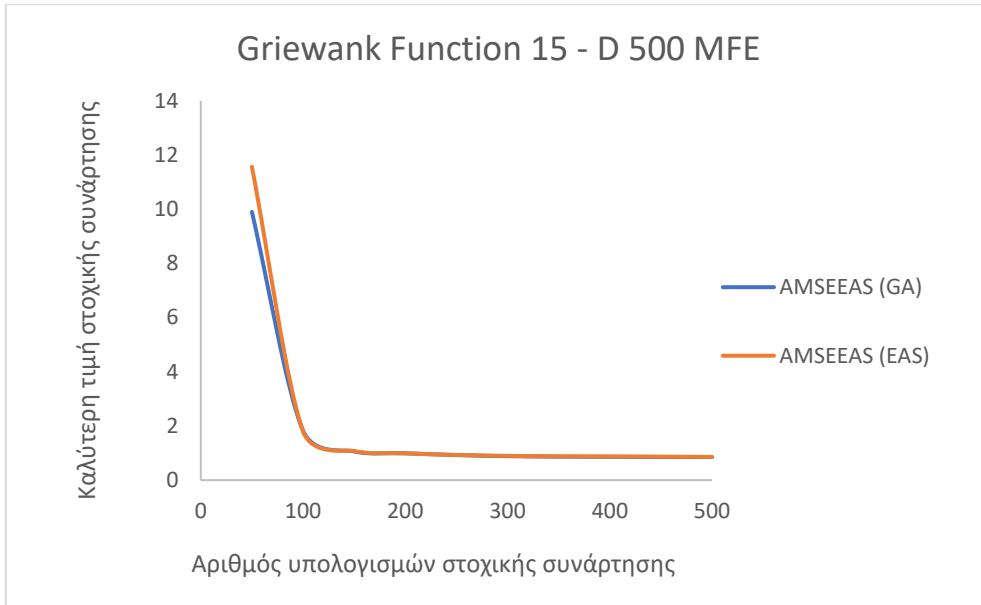


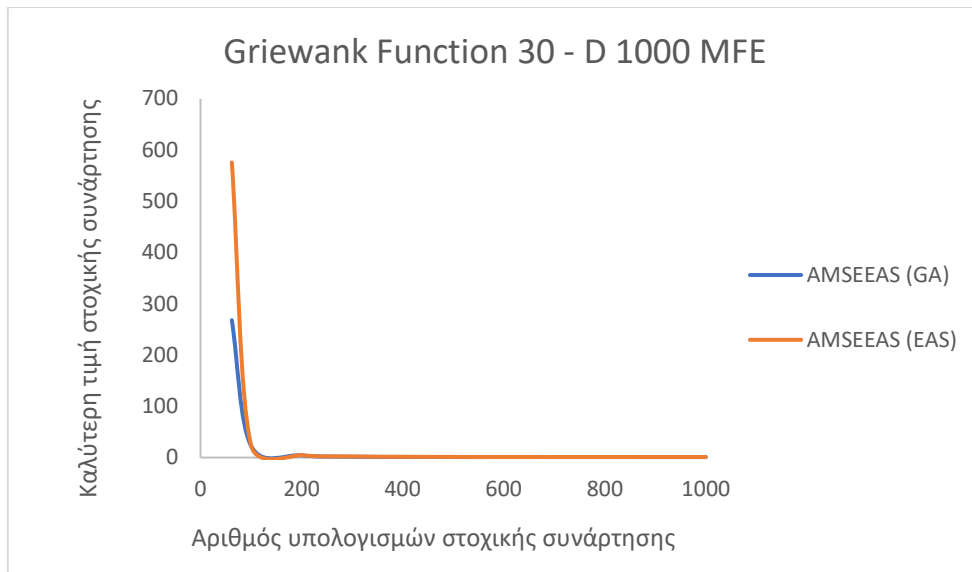
Σχήμα 1: Καμπύλες σύγκλισης των 4 περιπτώσεων της Sphere Function (OF1)



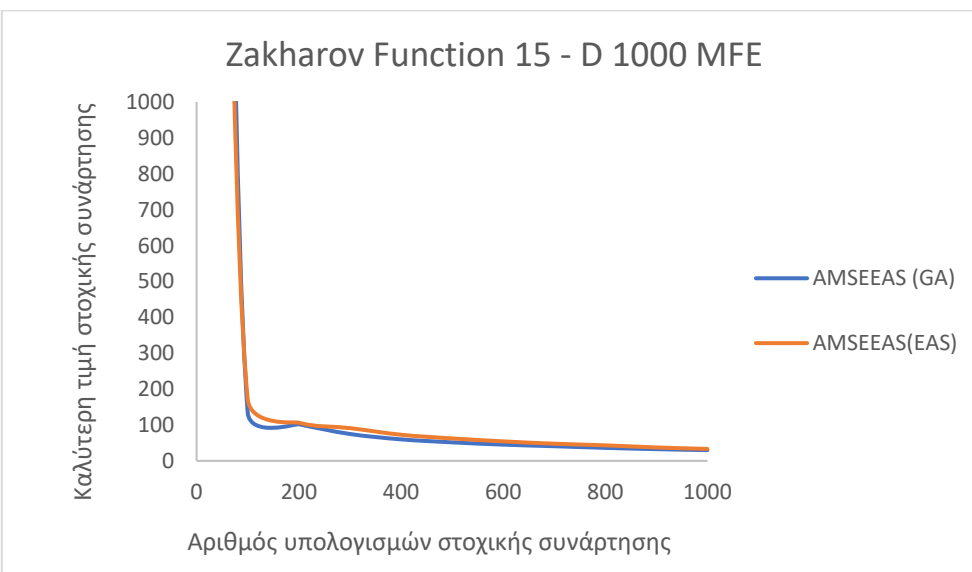
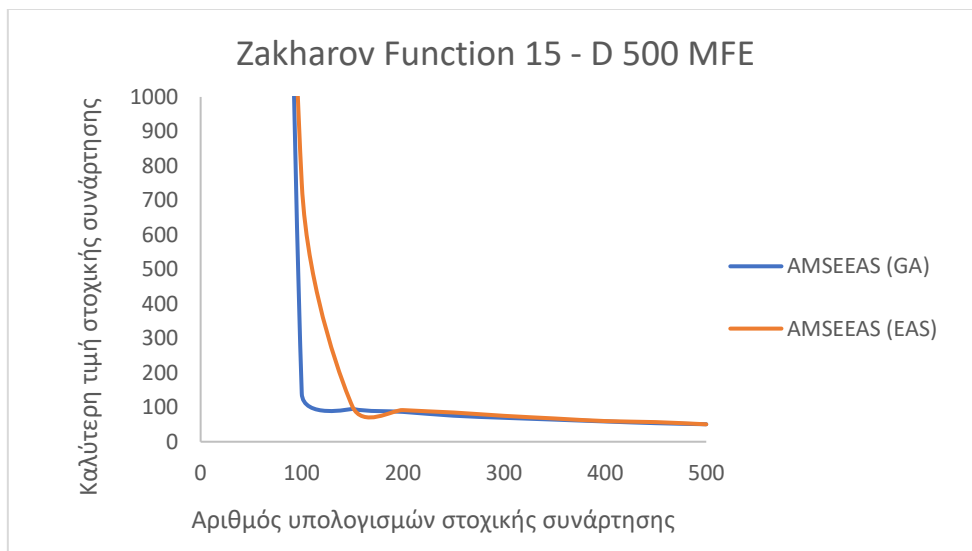


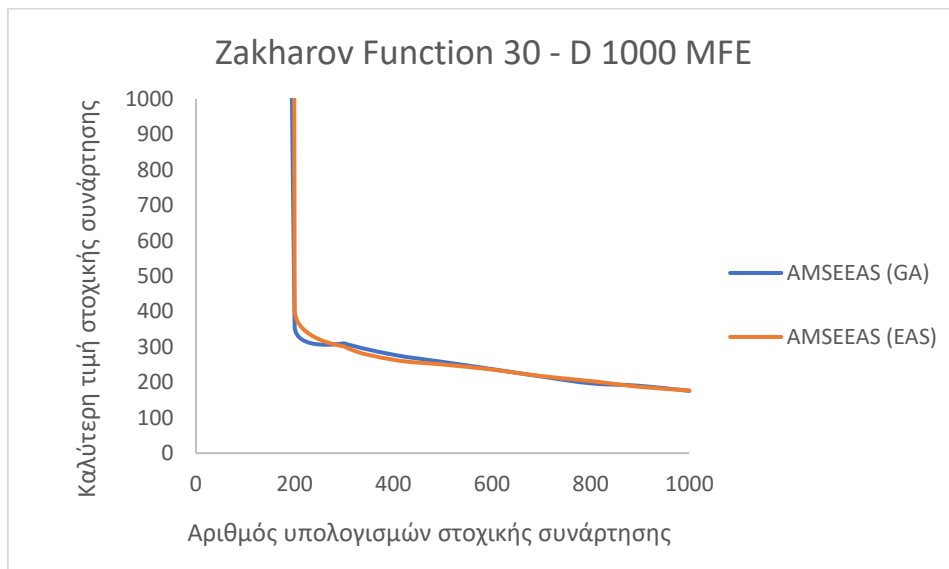
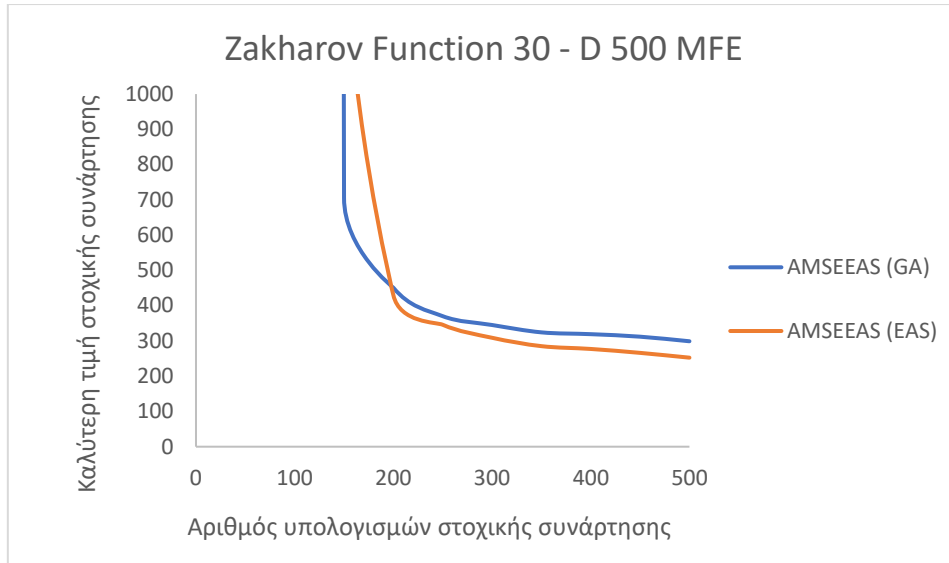
Σχήμα 2: Καμπύλες σύγκλισης των 4 περιπτώσεων της Ackley Function (OF2)



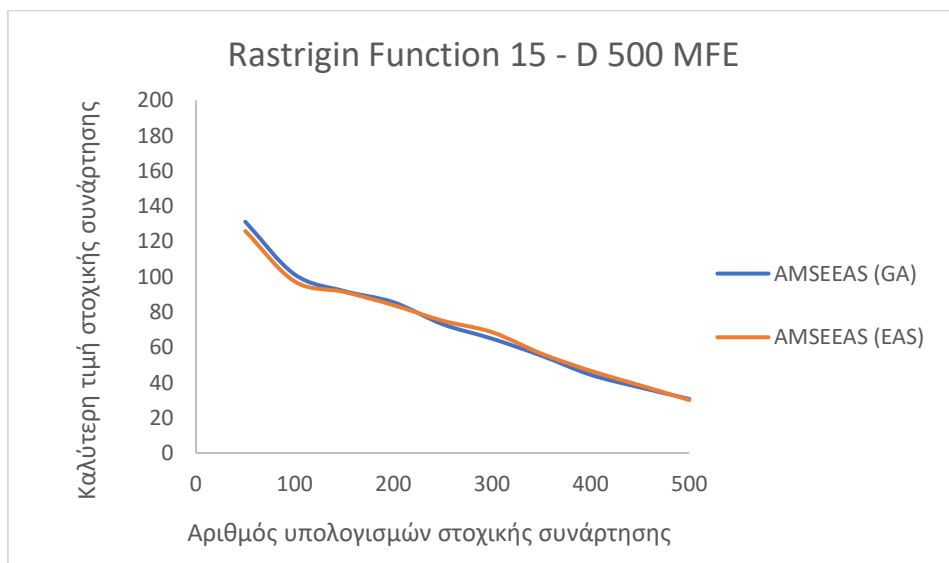


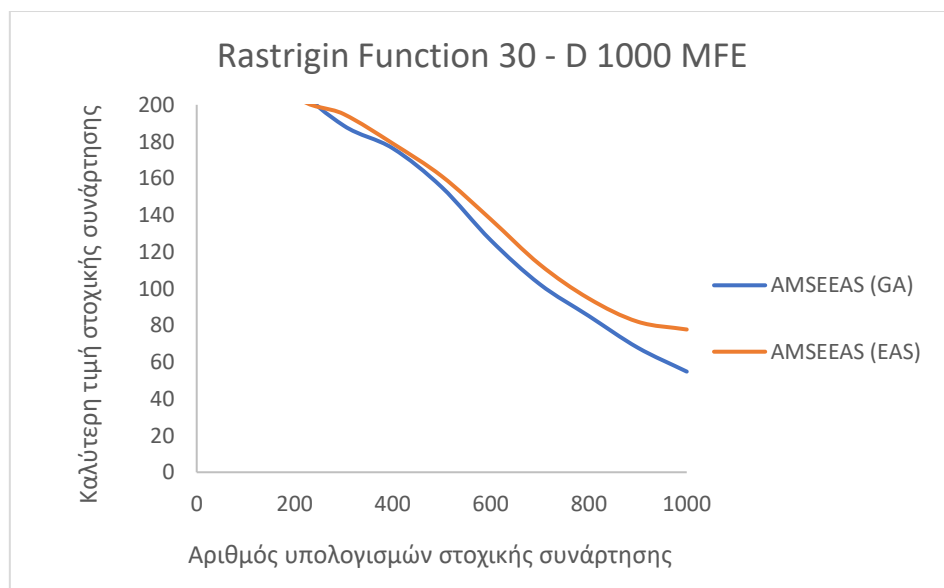
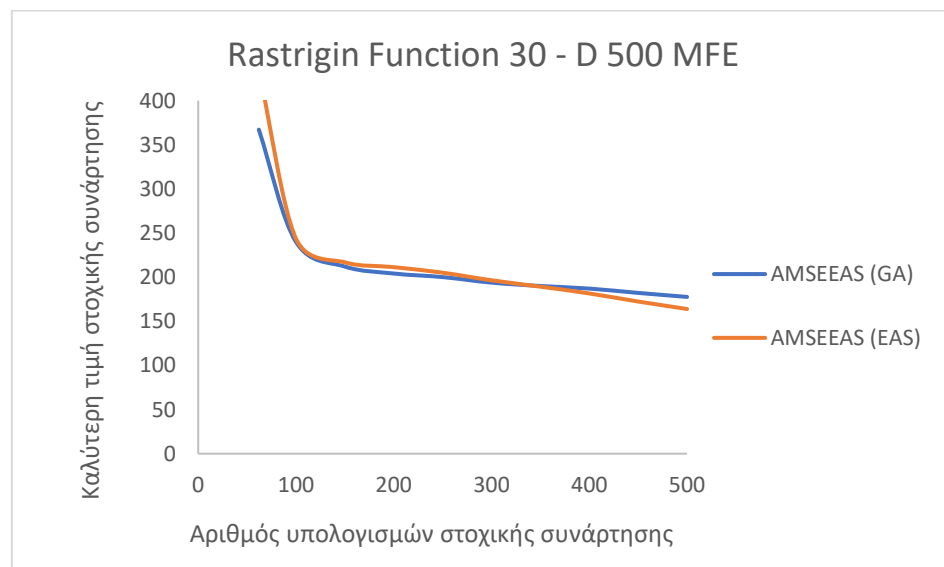
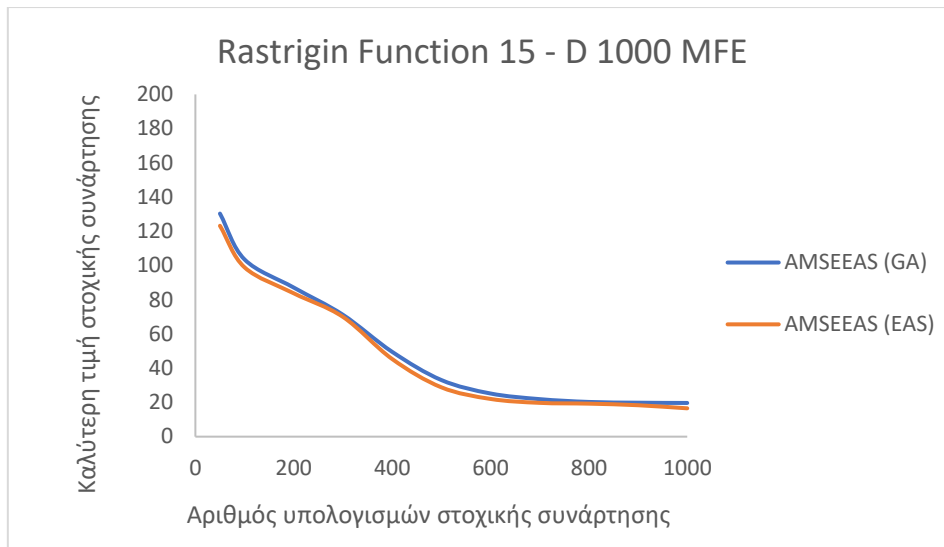
Σχήμα 3: Καμπύλες σύγκλισης των 4 περιπτώσεων της Griewank Function (OF3)



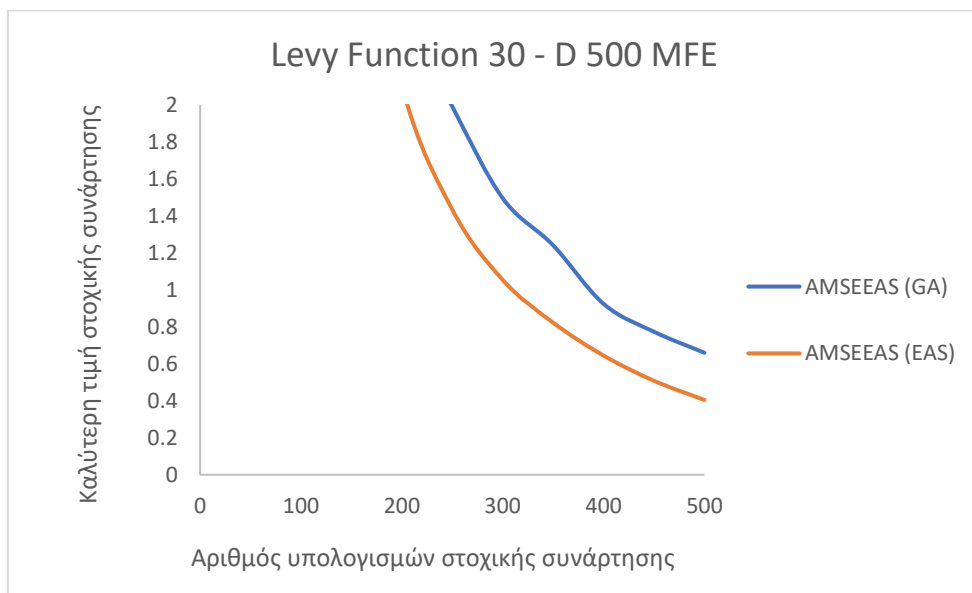
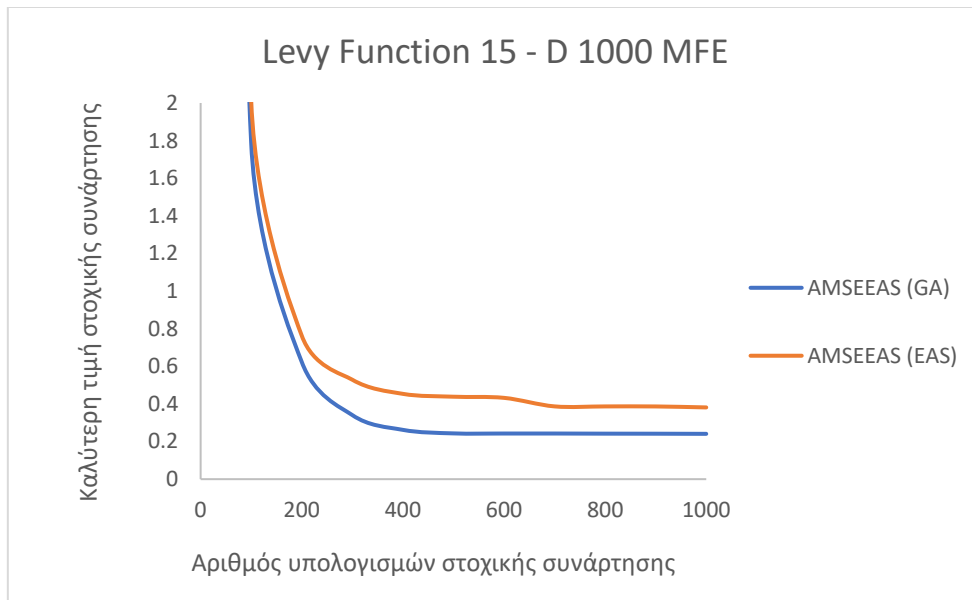
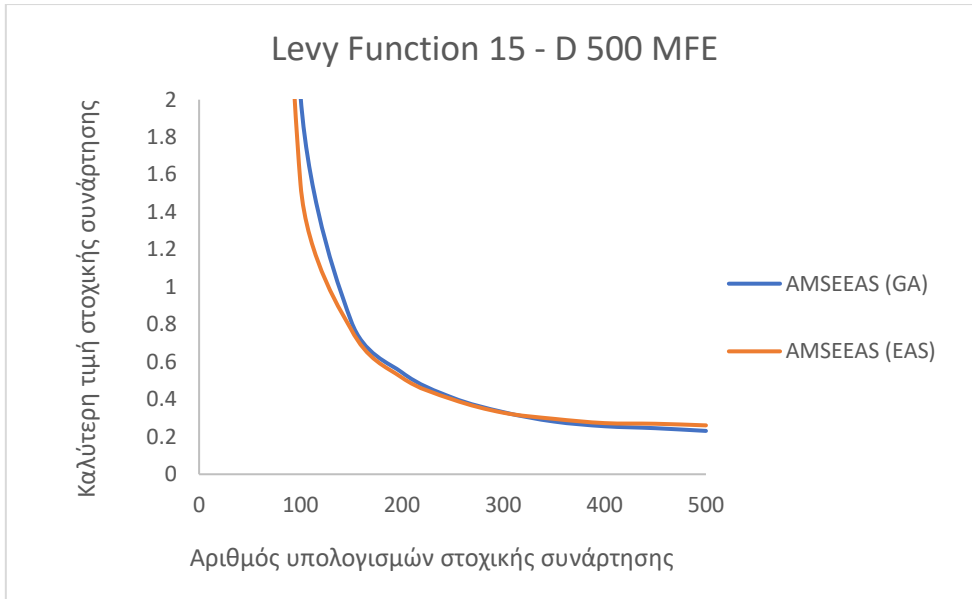


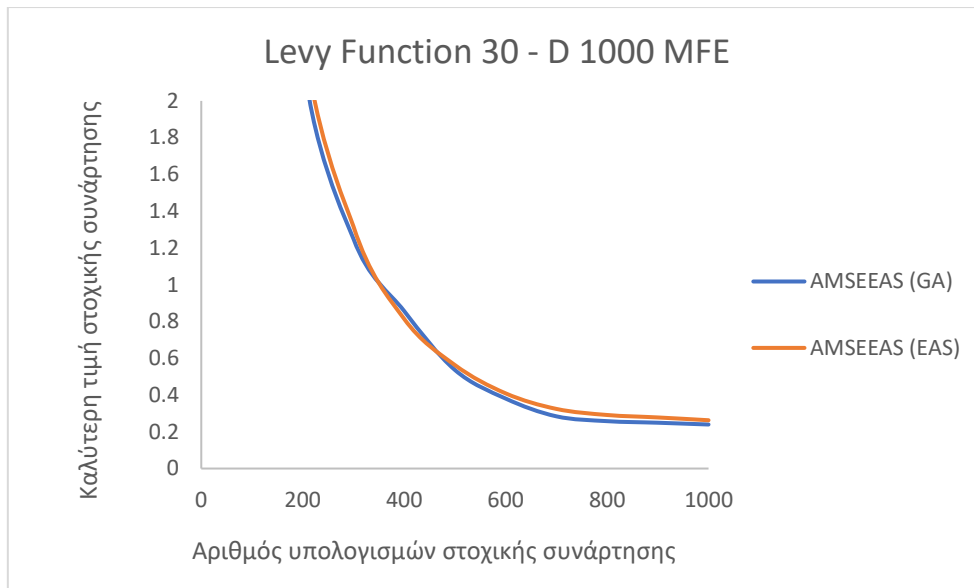
Σχήμα 4: Καμπύλες σύγκλισης των 4 περιπτώσεων της Zakharov Function (OF4)





Σχήμα 5: Καμπύλες σύγκλισης των 4 περιπτώσεων της Rastrigin Function (OF5)





Σχήμα 6: Καμπύλες σύγκλισης των 4 περιπτώσεων της Levy Function (OF6)