

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Διατύπωση και εφαρμογές στην αεροδυναμική της
Choreography Composition-Like (CCLO) μεθόδου
βελτιστοποίησης σε μονο/πολυκριτηριακά προβλήματα,
με ή χωρίς περιορισμούς

Διπλωματική Εργασία

Ναταλία Βάβουλα

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2020

Ευχαριστίες

Θα ήθελα να ευχαριστήσω από καρδιάς τους ανθρώπους που με βοήθησαν στην ολοκλήρωση αυτής της εργασίας.

Αρχικά θα ήθελα να πω ένα μεγάλο ευχαριστώ στον καθηγητή μου, Κ.Χ.Γιαννάκογλου που ήταν δίπλα μου σε όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας, παρά τον περιορισμένο χρόνο του. Η τεράστια μεταδοτικότητα του, οι βαθιές του γνώσεις πάνω στο αντικείμενο και η διάθεσή του για αναζήτηση νέων ιδεών και μεθόδων ήταν κρίσιμες για την ολοκλήρωση αυτής της εργασίας, αλλά και των σπουδών μου στο σύνολο.

Επιπλέον, θα ήθελα να αναγνωρίσω την κρίσιμη συμβολή της Δρ. Ασούτη, που με βοήθουσε καθ'όλη τη διάρκεια, εξηγώντας μου τη λειτουργία και χρήση ανεπτυγμένων από το εργαστήριο κωδίκων, παρατηρώντας και συμβουλευόντας με για τα αποτελέσματα που ο κώδικάς που έγραψα έδινε και εφοδιάζοντάς με με απαραίτητο λογισμικό.

Θα ήθελα ακόμα να ευχαριστήσω τον Δρ. Καψούλη, για την όλο το χρόνο και ενέργεια που διέθεσε για να μου μεταδώσει τις απαραίτητες γνώσεις, την υπομονή του στα λάθη μου, την εμπύχωση και την προθυμία του να με βοηθήσει, κάθε φορά που το χρειαζόμουν.

Αλλά και όλο το προσωπικό της Μονάδας Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης του ΕΜΠ, για την καλή ατμόσφαιρα και βοήθεια όποτε τους δινόταν η ευκαιρία.

Ένα μεγάλο ευχαριστώ θέλω επίσης να πω και στους καθηγητές μου, Σ.Πόνη και στην ερευνητική του ομάδα, με έμφαση στον Υ.Δ. Αγαλιανό, όπως επίσης και στον Σ.Πρωτοσύγγελο, που με συμβούλεψαν για τη δημιουργία της ενότητας της Οικονομικής Ποσότητας Παραγγελίας και μου έκαναν χρήσιμες παρατηρήσεις για διορθώσεις.

Τέλος, η διπλωματική εργασία δεν θα είχε ολοκληρωθεί χωρίς τη διαρκή υποστήριξη και αγάπη από την οικογένεια και τους φίλους μου, που στάθηκαν μαζί μου σε όλη τη διάρκεια, κατανοώντας την προσπάθεια που κατέβαλα και δίνοντας μου δύναμη να πετύχω περισσότερα.



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Διατύπωση και εφαρμογές στην αεροδυναμική της
Choreography Composition-Like (CCLO) μεθόδου
βελτιστοποίησης σε μονο/πολυκριτηριακά προβλήματα,
με ή χωρίς περιορισμούς

Διπλωματική Εργασία

Ναταλία Βάβουλα

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ
Αθήνα, 2020

Περίληψη

Υπάρχουν πολλές πληθυσμιακές στοχαστικές μέθοδοι βελτιστοποίησης. Οι εξελικτικοί αλγόριθμοι είναι ίσως η πιο διαδεδομένη. Μια από τις σχετικά πρόσφατες μεθόδους είναι η Harmony Search, η οποία μιμείται τη διαδικασία σύνθεσης της τζαζ μουσικής για την εύρεση της βέλτιστης λύσης. Η βέλτιστη λύση παραλληλίζεται με τη βέλτιστη αρμονία, η σύνθεση της οποίας είναι το ζητούμενο, και οι μεταβλητές σχεδιασμού αντιστοιχούν στις νότες κάθε μουσικού οργάνου οι οποίες τη συνθέτουν.

Στο ίδιο πλαίσιο, η ΜΠΥΡΒ/ΕΜΠ έχει, τα τελευταία 20 χρόνια αναπτύξει ιδιαίτερα το λογισμικό EASY, το οποίο βασίζεται σε εξελικτικούς αλγόριθμους που υποβοηθούνται από μεταπρότυπα (συνήθως, αλλά όχι αποκλειστικά, τεχνητά νευρωνικά δίκτυα) ώστε να επιλύονται προβλήματα μονοκριτηριακής και πολυκριτηριακής βελτιστοποίησης σε χρόνο πολύ μικρότερο από ότι χωρίς αυτά. Είναι εύλογο, η τεχνογνωσία που αυτή τη στιγμή υπάρχει στον EASY να μεταφέρεται σε οποιοδήποτε άλλο λογισμικό βελτιστοποίησης αντίστοιχης λογικής.

Έτσι, σκοπός αυτός της εργασίας είναι να γίνει μια πρώτη προσπάθεια να μεταφερθούν δοκιμασμένα στοιχεία που περιέχει ο EASY σε μια νέα μέθοδο στοχαστικής πληθυσμιακής βελτιστοποίησης η οποία δομείται βασισμένη στη γενική ιδέα του Harmony Search. Παράλληλα, στη μέθοδο αυτή εφαρμόζονται νέοι τελεστές, για τη βελτίωση της κλασσικής μεθόδου HS. Με τον τρόπο αυτό, η παραγόμενη μέθοδος μετουσιώνεται αποκτώντας νέα χαρακτηριστικά και η αναζήτηση της βέλτιστης λύσης μιμείται διαδικασίες σύνθεσης χορογραφίας. Όλοι οι τελεστές της μεθόδου αποκτούν σχετική φυσική σημασία από τη σύνθεση χορογραφίας και, για αυτό, στη μέθοδο προσδίνεται ένα νέο όνομα, «Choreography Composition-Like Optimization» (CCLO). Η προτεινόμενη μέθοδος εμπλέκει χρήση τεχνητής νοημοσύνης και δίνει δυνατότητα επίλυσης προβλη-

μάτων πολλών στόχων, με ή χωρίς περιορισμούς, υπολογίζοντας το μέτωπο Pareto.

Για την πιστοποίησή της, η μέθοδος εφαρμόζεται σε αριθμητικά προβλήματα ελαχιστοποίησης, σε προβλήματα αεροδυναμικής και οικονομικών. Η περίπτωση αεροδυναμικής βελτιστοποίησης αφορά βελτιστοποίηση της μορφής μεμονωμένης αεροτομής με στόχο είτε μόνο τη μεγιστοποίηση του συντελεστή άνωσης ή και την ελαχιστοποίηση του συντελεστή οπισθέλκουσας, με ή χωρίς περιορισμούς για το συντελεστή ροπής. Στην περίπτωση των οικονομικών, υπολογίζεται η οικονομική ποσότητα παραγγελίας προϊόντος, από επιχείρηση που χρησιμοποιεί το μοντέλο των προγραμματισμένων καθυστερήσεων για το προϊόν αυτό, όταν το κόστος προμήθειας του προϊόντος ακολουθεί κλιμακωτή χρέωση.



National Technical University of Athens
School of Mechanical Engineering
Fluids Section
Parallel CFD & Optimization Unit

**Formulation and applications of the Choreography
Composition-Like Optimization (CCLO) method in
unconstrained and constrained, single- and multi-objective
problems**

Diploma Thesis

Natalia Vavoula

Supervisor: Kyriakos C. Giannakoglou, Professor NTUA
Athens, 2020

Abstract

There are many population – based stochastic optimization methods. The evolutionary algorithms may be the most ubiquitous. One of the most recent methods is Harmony Search, which mimics the Jazz music’s composition process for finding the optimal solution. The optimal solution is an analogy of the optimal harmony, the composition of which is the objective, and the design variables correspond to the notes of each musical instrument that composes it.

Similarly, over the last 20 years, the Parallel CFD & Optimization Unit of NTUA has developed EASY, a software that is based on evolutionary algorithms supported by metamodels (radial basis function networks mostly, but not always) to solve single and multi – objective optimization problems more rapidly than without them. It stands to reason that, the already developed practices of EASY, will be transmitted to any similar optimization software.

The objective of the current diploma thesis is a first attempt to transmit already tested practices of EASY, in a new population – based stochastic optimization method that is based on the concept of Harmony Search. At the same time, new operators are applied for the improvement of the classical Harmony Search method. As a result, the optimization method is transmuted, obtaining new characteristics and the search for optimal solution mimics processes of choreography composition. All the method’s operators are obtaining meaning from the theory of dance composition and as a result, the method is renamed as Choreography Composition-Like Optimization (CCLO). The proposed method uses artificial intelligence and is able to solve multi-objective problems with or without constraints by computing Pareto fronts.

For its assessment, the method is applied to a number of computational minimization

problems as well as to problems in aerodynamics and in finance. In aerodynamics, the optimization concerns the shape of an isolated airfoil, either for maximum lift coefficient or for both maximum lift and minimum drag coefficients. The shape of the airfoil is also examined under constraints on the moment coefficient. In the second case, the economic order quantity of a product in a company is computed. The company uses the shortage model and the supplier applies tiered pricing to the product.

Ακρωνύμια

EMΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΕΘΣ	Εργαστήριο Θερμικών Στροβιλομηχανών
ΜΠΤΡ&Β	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΥΡΔ	Υπολογιστική Ρευστοδυναμική
ΜΕΔ	Μέσες Ετήσιες Δαπάνες
CFD	Computational Fluid Dynamics
NTUA	National Technical University of Athens
PCopt	Parallel CFD & Optimization unit
EA	Evolutionary Algorithms
GA	Genetic Algorithms
ES	Evolution Strategies
SPEA	Strength Pareto Evolutionary Algorithm
HS	Harmony Search
CCLO	Choreography Composition-Like Optimization
HVI	Hypervolume Indicator
RBF	Radial Basis Function Networks
HM	Harmony Memory
CS	Choreography Set
HMS	Harmony Memory Size
CSS	Choreography Set Size

HMCR	Harmony Memory Consideration Rate
MMR	Mimic Movement Rate
PAR	Pitch Adjustment Rate
LSR	Legato-Staccato Rate
EOQ	Economic Order Quantity

Περιεχόμενα

Περιεχόμενα	i
1 Εισαγωγή	1
1.1 Εισαγωγή στη βελτιστοποίηση	1
1.2 Στοχαστικές Μέθοδοι Βελτιστοποίησης	3
1.2.1 Μέτωπο Pareto	3
1.2.2 Μεταπρότυπα	4
1.2.3 Αλγόριθμοι Στοχαστικής Βελτιστοποίησης	5
1.3 Στόχος διπλωματικής εργασίας	10
2 Βάσεις Νέου Αλγορίθμου	11
2.1 Harmony Search	11
2.1.1 Αλγόριθμος βελτιστοποίησης HS	11
2.1.2 Αλγόριθμος δημιουργίας νέου ατόμου με την τεχνική HS	12
2.2 (μ,λ) Εξελικτικός Αλγόριθμος	13
2.2.1 Λογισμικό EASY	13
2.2.2 (μ,λ) EA, Μονοκριτηριακή Βελτιστοποίηση	13
2.2.3 (μ,λ) Πολυκριτηριακή Βελτιστοποίηση	15
2.2.4 EA Υποβοηθούμενοι από Μεταπρότυπα	17
2.2.5 EAs και MAEAs σε Προβλήματα με Περιορισμούς	19
2.2.6 Διάγραμμα Ροής EASY	20
3 Choreography Composition-Like Optimization	21
3.1 Διαφορές HS - CCLO	21

3.2	Ανάλυση δημιουργίας χορογραφίας	23
3.3	Ανάλυση αλγορίθμου CCLO	24
3.3.1	Αλγόριθμος δημιουργίας νέου ατόμου με την τεχνική CCLO	24
3.3.2	CCLO χωρίς χρήση μεταπροτύπων	25
3.3.3	CCLO με χρήση μεταπροτύπων	26
3.4	Ομοιότητες και διαφορές CCLO - (μ,λ) EA	28
4	Μετάβαση από HS σε CCLO	31
4.1	Εξεταζόμενες συναρτήσεις	31
4.2	Αποτελέσματα αλλαγών	33
4.2.1	1 ^η Αλλαγή: Πλήθος νέων υποψήφιας λύσεων	33
4.2.2	2 ^η Αλλαγή: Διαχείριση υπέρβασης ορίων μεταβλητών	34
4.2.3	3 ^η Αλλαγή: Εύρος BW	34
4.2.4	4 ^η Αλλαγή: Διασταύρωση	34
4.2.5	5 ^η Αλλαγή: Μεταπρότυπα	35
4.2.6	Συγκεντρωτικά αποτελέσματα	36
5	Πιστοποίηση κώδικα σε μαθηματικές συναρτήσεις	39
5.1	Συναρτήσεις Μονοκριτηριακής Βελτιστοποίησης	39
5.1.1	Συνάρτηση Rosenbrock	39
5.1.2	Συνάρτηση Sphere	42
5.2	Συναρτήσεις Πολυκριτηριακής Βελτιστοποίησης	45
5.2.1	Συνάρτηση Kursawe	45
5.2.2	Συνάρτηση Poloni	48
5.2.3	Συνάρτηση Schaffer N.1	51
5.2.4	Συνάρτηση Schaffer N.1 με περιορισμούς	54
6	Εφαρμογές στην Αεροδυναμική και τα Οικονομικά	55
6.1	Μεμονωμένη Αεροτομή	55
6.1.1	Πρόβλημα ενός στόχου	57
6.1.2	Πρόβλημα δύο στόχων	59

6.1.3	Πρόβλημα δύο στόχων υπό περιορισμούς	62
6.2	Οικονομική Ποσότητα Παραγγελίας	65
6.2.1	Θεωρητικό Υπόβαθρο	65
6.2.2	Πρόβλημα βελτιστοποίησης	67
7	Συμπεράσματα	73
A'	Σταθερές CCLO	75
B'	Κώδικας εύρεσης δείκτη υπερόγκου	77
B'.1	Ανάγκη δημιουργίας κώδικα	77
B'.2	Βήματα υπολογισμού δείκτη υπερόγκου	78
	Βιβλιογραφία	79

Κεφάλαιο 1

Εισαγωγή

Στο κεφάλαιο αυτό γίνεται μια εισαγωγή στις βασικές έννοιες της βελτιστοποίησης. Στη συνέχεια αναλύονται οι μέθοδοι στοχαστικής βελτιστοποίησης, τις οποίες αφορά αυτή η διπλωματική εργασία και παρουσιάζονται ορισμένες από αυτές. Τέλος, αναφέρεται ο στόχος της εργασίας αυτής.

1.1 Εισαγωγή στη βελτιστοποίηση

Η βελτιστοποίηση χρησιμοποιείται ευρέως στο σχεδιασμό προϊόντων, υπηρεσιών ή διεργασιών, στα οικονομικά, στη διάθεση πόρων, στον προγραμματισμό και στον έλεγχο.

Οι μέθοδοι βελτιστοποίησης χρησιμοποιούνται για την εύρεση της καλύτερης λύσης ενός προβλήματος, μέσα από ένα σύνολο πιθανών λύσεων, τηρώντας τους εκάστοτε περιορισμούς. Στόχος της βελτιστοποίησης είναι είτε η ελαχιστοποίηση συναρτήσεων κόστους (cost functions) είτε η μεγιστοποίηση συναρτήσεων καταλληλότητας (fitness functions).

Το πρόβλημα μπορεί να αποτελείται από μία συνάρτηση - στόχο (Single Objective Optimization - SOO) ή από ένα σύνολο συναρτήσεων - στόχων (Multi Objective Optimization - MOO). Στην περίπτωση των πολλών στόχων, οι στόχοι είναι συνήθως ανταγωνιστικοί, δηλαδή η βελτίωση ενός χειροτερεύει την απόδοση ως προς τους άλλους. Στην περίπτωση αυτή υπολογίζεται ένα μέτωπο Pareto, εξίσου καλών λύσεων που κυριαρχούν στις υπόλοιπες λύσεις του προβλήματος, δηλαδή είναι καλύτερες από τις υπόλοιπες λύσεις ως προς κάθε συνάρτηση - στόχο.

Η συνάρτηση ενός προβλήματος ελαχιστοποίησης εκφράζεται, στη γενική μορφή της,

από τη σχέση 1.1

$$\min \vec{f}(\vec{x}) = \min [f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})] \quad (1.1)$$

Στη σχέση 1.1, με \vec{f} συμβολίζεται το διάνυσμα των συναρτήσεων - στόχων, το διάνυσμα \vec{x} περιλαμβάνει τις τιμές των N σε πλήθος μεταβλητών σχεδιασμού και οι $f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})$ αποτελούν τις επιμέρους συναρτήσεις - στόχους.

Το πρόβλημα μπορεί να υπόκειται σε περιορισμούς ισότητας ή/και ανισότητας, όπως φαίνεται στην 1.2, όπου M_1 και M_2 τα πλήθη των περιορισμών ισότητας και ανισότητας αντίστοιχα

$$\begin{aligned} h_i(\vec{x}) &= 0, & i &= 1, \dots, M_1 \\ g_i(\vec{x}) &\leq 0, & i &= 1, \dots, M_2 \end{aligned} \quad (1.2)$$

Η λύση που προκύπτει είναι το άτομο του οποίου το σύνολο των μεταβλητών σχεδιασμού, δίνει τη βέλτιστη τιμή στη συνάρτηση - στόχο. Οι μεταβλητές σχεδιασμού μπορεί να είναι συνεχείς ή διακριτές. Κάθε μεταβλητή του προβλήματος x_i , με το i να ανήκει στο σύνολο των N μεταβλητών, μπορεί να λάβει τιμές εντός ενός προκαθορισμένου διαστήματος αναζήτησης (search space), από x_i^{low} έως x_i^{up} .

Βελτιστοποίηση είναι η διαδικασία ανίχνευσης του χώρου των υποψήφιων λύσεων και εντοπισμού της καλύτερης (optimal solution). Η ποιότητα της μεθόδου βελτιστοποίησης κρίνεται από το χρόνο που απαιτείται για να εντοπισθεί η λύση αυτή και από την πορεία σύγκλισης, δηλαδή τις καλύτερες λύσεις που έχουν εντοπισθεί κάθε στιγμή της αναζήτησης.

Για τη βελτιστοποίηση είναι απαραίτητοι δύο αντικρουόμενοι παράγοντες. Η ανίχνευση (exploration) και η εκμετάλλευση (exploitation). Ανίχνευση είναι η δοκιμή λύσεων σε όλο το εύρος της περιοχής αναζήτησης, με σκοπό την εύρεση της επιμέρους περιοχής στην οποία βρίσκεται η βέλτιστη λύση. Εκμετάλλευση είναι η εστίαση σε μια συγκεκριμένη περιοχή και αναζήτηση της καλύτερης λύσης στην περιοχή αυτή.

Κατά τη βελτιστοποίηση σε συναρτήσεις με πολλά ακρότατα (multimodal functions), σε περίπτωση ανεπαρκούς ανίχνευσης, υπάρχει ο κίνδυνος εντοπισμού κάποιας περιοχής τοπικού ακρότατου και «εγκλωβισμού» σε αυτή, δηλαδή μεγάλης εκμετάλλευσης της περιοχής αυτής, γεγονός που θα έχει ως αποτέλεσμα υποβέλτιστες λύσεις (sub-optimal solutions). Λόγω αυτής της δυσκολίας, εξετάζονται συναρτήσεις με πολλά ακρότατα για την αξιολόγηση των διαφόρων μεθόδων αξιολόγησης.

1.2 Στοχαστικές Μέθοδοι Βελτιστοποίησης

Οι μέθοδοι βελτιστοποίησης διακρίνονται σε στοχαστικές (stochastic) και αιτιοκρατικές (deterministic). Οι αιτιοκρατικές μέθοδοι επιλύουν με χρήση της παραγώγου της συνάρτησης - στόχου, επομένως εξαρτώνται από το προς επίλυση πρόβλημα και την αρχικοποίηση που θα γίνει στις μεταβλητές σχεδιασμού του. Αντίθετα, οι στοχαστικές χαρακτηρίζονται από τυχαιότητα στην αναζήτηση λύσης, κατά συνέπεια είναι πιο χρονοβόρες, άρα και κοστοβόρες. Το πλεονέκτημά τους είναι ότι ο αλγόριθμος βελτιστοποίησης είναι γενικής χρήσης, αντιμετωπίζουν δηλαδή το πρόβλημα ως «μαύρο κουτί» (“black box”), επομένως δεν χρειάζεται ανάπτυξη από το μηχανικό - προγραμματιστή για κάθε αλλαγή στο πρόβλημα.

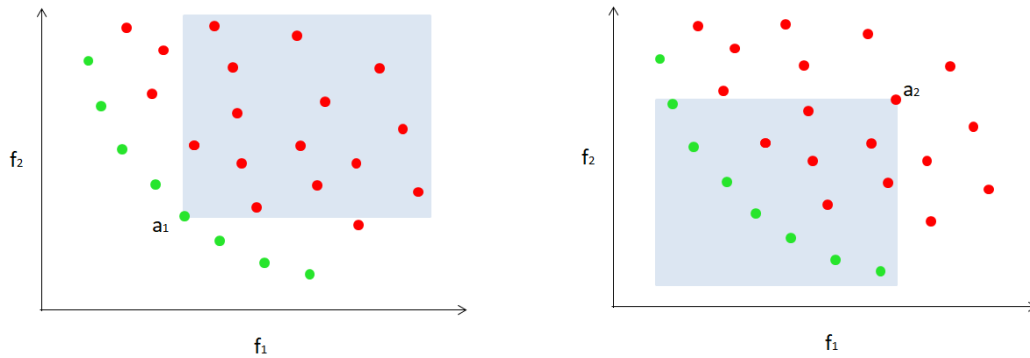
1.2.1 Μέτωπο Pareto

Κατά τη στοχαστική βελτιστοποίηση, προκύπτει ένα σύνολο λύσεων, από το οποίο πρέπει να βρεθούν οι βέλτιστες. Στην περίπτωση του ενός στόχου, η σύγκριση των λύσεων είναι απλή. Καλύτερη είναι η λύση που βελτιστοποιεί, δηλαδή ελαχιστοποιεί στην περίπτωση που εξετάζεται, την τιμή της συνάρτησης - στόχο [1]. Στην περίπτωση των πολλών, αντικρουόμενων στόχων, η σύγκριση είναι πιο πολύπλοκη. Συνήθως, δεν υπάρχει λύση που ταυτόχρονα να ελαχιστοποιεί όλες τις συναρτήσεις, όπως φαίνεται στο σχήμα 1.1. Για τα πράσινα σημεία για παράδειγμα, μείωση της f_1 προκαλεί αύξηση στην f_2 .

Δημιουργείται λοιπόν η ανάγκη εύρεσης ενός μετώπου, που να περιέχει τις καλύτερες λύσεις. Το μέτωπο αυτό ονομάζεται Pareto. Οι λύσεις που περιλαμβάνονται στο μέτωπο είναι οι μη-κυριαρχούμενες. Ως μη-κυριαρχούμενη ορίζεται μια λύση όταν δεν υπάρχει άλλη λύση με μικρότερες τιμές όλων των συναρτήσεων - στόχων ταυτόχρονα.

Για παράδειγμα, στο σχήμα 1.1 αριστερά, το σημείο x_1 κυριαρχεί σε όλες τις λύσεις που περιέχονται στο γκρι πλαίσιο, αφού έχει μικρότερες τιμές για τις συναρτήσεις f_1 και f_2 ταυτόχρονα. Δεν υπάρχει όμως καμία λύση που να κυριαρχεί σε αυτό, αφού κανένα σημείο δεν έχει μικρότερες τιμές ταυτόχρονα και για τις δύο συναρτήσεις. Επομένως, το άτομο a_1 ανήκει στο μέτωπο των μη-κυριαρχούμενων λύσεων. Αντίθετα, στα δεξιά του σχήματος 1.1 φαίνεται ένα άλλο άτομο, το a_2 . Με την ίδια λογική, το άτομο αυτό κυριαρχείται από όλα τα σημεία της γκρι περιοχής, αφού έχει μικρότερη τιμή των συναρτήσεων f_1 και f_2 .

Χάρη στο μέτωπο Pareto, δίνεται η δυνατότητα εύρεσης ενός συνόλου εξίσου καλών λύσεων και επιλογής της επιθυμητής με βάση τα προσωπικά κριτήρια υπεύθυνου λήψης αποφάσεων.



Σχήμα 1.1: Αριστερά: Κυριαρχία το ατόμου x_1 σε αυτά της γκρι περιοχής. Δεξιά: Σημεία από τα οποία κυριαρχείται το άτομο x_2 .

1.2.2 Μεταπρότυπα

Βασικό μειονέκτημα των στοχαστικών μεθόδων βελτιστοποίησης είναι το μεγάλο υπολογιστικό κόστος αξιολογήσεων των ατόμων που δημιουργούνται κατά την αναζήτηση της βέλτιστης λύσης. Οι κλήσεις του λογισμικού αξιολόγησης απαιτούν χρόνο, ειδικότερα όταν πρόκειται για βελτιστοποίηση στην αεροδυναμική όπου το λογισμικό αξιολόγησης είναι κώδικας υπολογιστικής ρευστοδυναμικής (Computational Fluid Dynamics - CFD). Για το λόγο αυτό, οι στοχαστικές μέθοδοι βελτιστοποίησης συχνά υποβοηθούνται από μεταπρότυπα [2].

Σκοπός των μεταπρωτύπων είναι ο γρήγορος προσεγγιστικός υπολογισμός της συνάρτησης - στόχου. Χάρη στα μεταπρότυπα, δεν αξιολογούνται όλα τα άτομα με το κοστοβόρο λογισμικό αξιολόγησης. Από τα άτομα που δημιουργούνται σε κάθε στάδιο της βελτιστοποίησης, επιλέγονται μόνο αυτά με τις προσεγγιστικά καλύτερες τιμές συναρτήσεων - στόχων και επαναυπολογίζονται με το λογισμικό ακριβούς αξιολόγησης [3].

Τα μεταπρότυπα είναι είτε συνδεδεμένα είτε αποσυνδεδεμένα από τη διαδικασία της βελτιστοποίησης. Στα αποσυνδεδεμένα, η εκπαίδευση του μεταπρωτύπου γίνεται πριν ξεκινήσει η διαδικασία της βελτιστοποίησης, επομένως οι λύσεις που εντοπίζονται κατά τη διάρκεια της βελτιστοποίησης δεν μεταβάλλουν τη συμπεριφορά του μεταπρωτύπου.

Αντίθετα, στα συνδεδεμένα, όπως αυτά έχουν αναπτυχθεί και χρησιμοποιούνται από τη ΜΠΥΡ&Β του ΕΜΠ [4], κατά τη διάρκεια της βελτιστοποίησης δημιουργείται προσωποποιημένο τοπικό μεταπρότυπο (personalized local metamodel) για κάθε νέα υποψήφια λύση, από το οποίο προσεγγίζεται η τιμή της συνάρτησης - στόχου της. Η εκπαίδευση του μεταπρωτύπου γίνεται από τα σημεία που έχουν κατά τη διάρκεια της βελτιστοποίησης υπολογιστεί με το λογισμικό αξιολόγησης και απέχουν την ελάχιστη απόσταση από το νέο άτομο. Με τον τρόπο αυτό γίνεται επιπλέον εκμετάλλευση των ατόμων που βρίσκονται κατά τη διάρκεια της αξιολόγησης, με αποτέλεσμα, ειδικά στις περιοχές που εντοπίζονται καλές λύσεις, να υπάρχει μεγάλο πλήθος ατόμων, επομένως

να γίνονται καλύτερες προσεγγίσεις.

1.2.3 Αλγόριθμοι Στοχαστικής Βελτιστοποίησης

Τα τελευταία χρόνια έχουν δημιουργηθεί πολλοί αλγόριθμοι στοχαστικής βελτιστοποίησης, οι οποίοι βασίζονται σε μεταφορές που περιγράφουν φυσικές διεργασίες, κοινωνικές συμπεριφορές ή τεχνητά φαινόμενα.

Μερικοί από αυτούς είναι οι Particle Swarm Optimization [5], Ant Colony Optimization [6], Harmony Search [7], Fly optimization algorithm [8], Fruit fly optimization algorithm [9], Termite colony optimization [10], Intelligent water drops algorithm [11], Imperialist competitive algorithm [12], Shuffled frog-leaping algorithm [13], Bee colony algorithm [14], Bees optimization algorithm [15], Honey-bees mating optimization [16], The great salmon run [17], Cat swarm optimization [18] και Artificial fish swarm optimization [19].

Στην ενότητα αυτή παρουσιάζονται ορισμένοι βασικοί στοχαστικοί αλγόριθμοι βελτιστοποίησης, και η δομή τους σε αναλογία με τη μεταφορά από την οποία προκύπτουν.

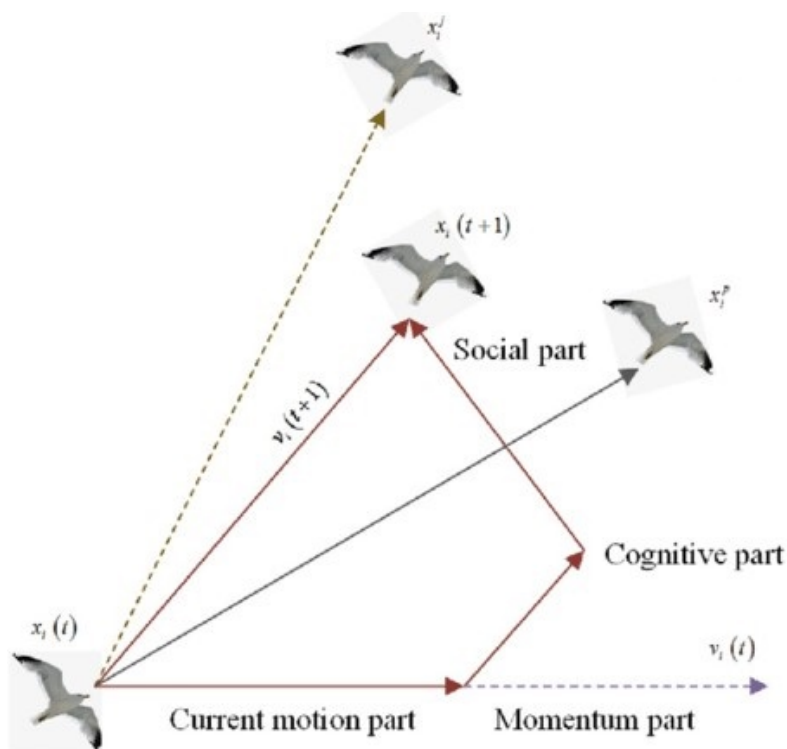
Particle Swarm Optimization

Η μέθοδος Particle Swarm Optimization (PSO) αποτελεί μια στοχαστική μέθοδο βελτιστοποίησης εμπνευσμένη από τη βιολογία. Αναπτύχθηκε το 1995 από τους J. Eberhart και R. Kennedy [5], και βασίζεται στους μηχανισμούς πλοήγησης ή αναζήτησης τροφής κοινωνιών ζώων που δεν έχουν αρχηγούς, όπως σμήνη πτηνών ή κοπάδια ψαριών [20].

Τα άτομα κοπαδιών χωρίς αρχηγούς βρίσκουν τροφή τυχαία, ακολουθώντας τα άτομα που βρίσκονται πιο κοντά σε πηγές τροφής [21]. Άτομα που βρίσκονται κοντά σε πηγές τροφής, επικοινωνούν με τα υπόλοιπα, ενημερώνοντάς τα για τη τοποθεσία τους. Τα υπόλοιπα άτομα αλλάζουν την πορεία τους, ώστε να πλησιάσουν την πηγή τροφής. Αυτό συμβαίνει επαναλαμβανόμενα, μέχρι να εντοπισθεί η καλύτερη δυνατή περιοχή.

Στη μέθοδο PSO, τα σωματίδια particles αντιπροσωπεύουν τις πιθανές λύσεις, και το σύνολο των πιθανών λύσεων αποτελεί το σμήνος swarm. Η εύρεση τροφής σε κάθε περιοχή αποτελεί την αντικειμενική συνάρτηση. Τα σωματίδια κινούνται στο χώρο. Η νέα θέση κάθε σωματιδίου εξαρτάται από την προηγούμενη θέση του και την ονομαζόμενη «ταχύτητα» κίνησης, δηλαδή η απόσταση που το σωματίδιο διένυσε κατά την κίνηση στη νέα θέση. Η ταχύτητα της κίνησης αποτελεί συνδυασμό της αδράνειας (momentum part), δηλαδή της ταχύτητας που το σωματίδιο έχει, της βέλτιστης λύσης που έχει μέχρι στιγμής εντοπίσει το συγκεκριμένο σωματίδιο (cognitive part) και της βέλτιστης τιμής που έχει εντοπίσει συνολικά το σμήνος (social part), όπως φαίνεται στο σχήμα 1.2.

Σε κάθε νέα θέση που βρίσκεται κάθε σωματίδιο του σμήνους, η βέλτιστη θέση που έχει



Σχήμα 1.2: Διαδικασία εύρεσης νέας ταχύτητας κίνησης [22].

βρεθεί μέχρι στιγμής από το σμήνος γνωστοποιείται σε όλα τα άτομα και η διαδικασία επαναλαμβάνεται.

Η αντίχνηση και η εκμετάλλευση εξαρτώνται κυρίως από τη βαρύτητα που η αδράνεια έχει στον εντοπισμό της νέας θέσης. Εάν η αδράνεια έχει μεγάλη βαρύτητα, η αντίχνηση είναι μεγάλη και η εκμετάλλευση μικρή. Το αντίθετο συμβαίνει για μικρή βαρύτητα αδράνειας. Στις πρώτες αναζητήσεις η βαρύτητα της αδράνειας είναι μεγάλη και, στη συνέχεια, γραμμικά με τις αναζητήσεις πέφτει. Με τον τρόπο αυτό αρχικά σαρώνεται ολόκληρο το πεδίο για την εύρεση της βέλτιστης περιοχής και, στη συνέχεια, η αναζήτηση επικεντρώνεται στην περιοχή αυτή για εύρεση του βέλτιστου σημείου.

Ant Colony Optimization

Η μέθοδος Ant Colony Optimization (ACO) αποτελεί μια στοχαστική μέθοδο βελτιστοποίησης που αναπτύχθηκε το 1992 από τους M. Dorigo, A. Colomni και V. Maniezzo [6]. Η πρώτη εφαρμογή της μεθόδου αυτής σε βελτιστοποίηση στην αεροδυναμική έγινε από τη ΜΠΥΡ&Β/ΕΜΠ [23]. Η τεχνική εμπνεύστηκε από τη διαδικασία αναζήτησης τροφής ορισμένων ειδών μυρμηγκιών, που αφήνουν φερομόνη καθώς κινούνται για να σηματοδοτήσουν το μονοπάτι που πρέπει να ακολουθηθεί από άλλα άτομα της αποικίας

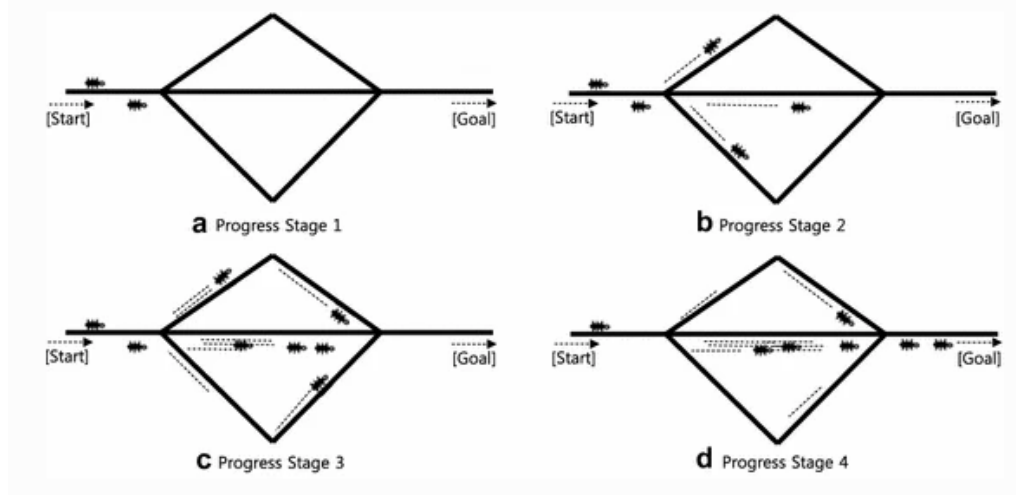
[24].

Κατά την αναζήτηση τροφής, τα μυρμήγκια αρχικά εξερευνούν την περιοχή τυχαία [25]. Μόλις βρουν την πηγή τροφής, αξιολογούν την ποσότητά και την ποιότητά της, και τη μεταφέρουν στη φωλιά τους αφήνοντας φερομόνη στο δρόμο της επιστροφής. Η ποσότητα φερομόνης που αφήνουν αντιστοιχεί στην ποσότητα και την ποιότητα της τροφής που εντόπισαν [26].

Τροποποιώντας το περιβάλλον τους, τα άτομα του πληθυσμού επικοινωνούν έμμεσα, ώστε να βρουν την πηγή τροφής μεταξύ της φωλιάς τους και της πηγής τροφής [27].

Στη μέθοδο ACO, οι διαδρομές που ακολουθούν τα μυρμήγκια αποτελούν το σύνολο των πιθανών λύσεων. Βέλτιστη λύση είναι η διαδρομή με τη μεγαλύτερη ποσότητα φερομόνης, δηλαδή η διαδρομή που ακολουθήθηκε της περισσότερες φορές [28].

Σε κάθε κόμβο της διαδρομής, το άτομο εξετάζει τις πιθανές κατευθύνσεις που μπορεί να ακολουθήσει. Η πιθανότητα του ατόμου να ακολουθήσει κάθε κατεύθυνση είναι μεγαλύτερη για τις κατευθύνσεις που υπάρχει μεγαλύτερη ποσότητα φερομόνης. Στο σχήμα 1.3 φαίνεται γραφικά η διαδικασία αυτή. Καθώς όλο και περισσότερα άτομα διαλέγουν τη μεσαία διαδρομή για να κινηθούν, η ποσότητα φερομόνης στη διαδρομή αυτή αυξάνεται, με αποτέλεσμα όλο και περισσότερα άτομα του πληθυσμού να την ακολουθούν.



Σχήμα 1.3: Διαδικασία επιλογής διαδρομής με κριτήριο την ποσότητα φερομόνης [29].

Μέρος της φερομόνης εξατμίζεται μεταξύ διαδοχικών αναζητήσεων. Μεγάλος βαθμός εξατμίσεως προκαλεί καλύτερη ανίχνευση, ενώ μικρότερος καλύτερη εκμετάλλευση.

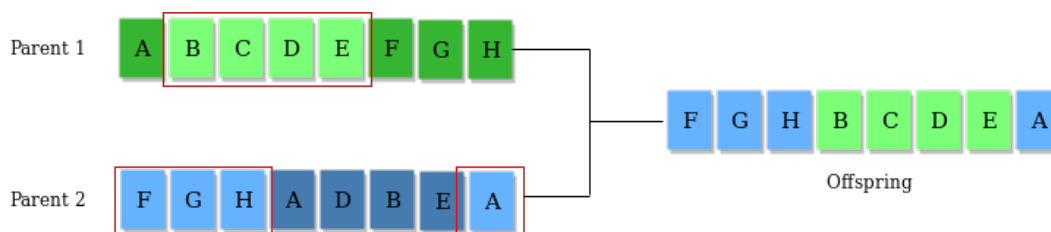
Evolutionary Algorithm

Η μέθοδος Evolutionary Algorithm (EA) αποτελεί μια στοχαστική μέθοδο βελτιστοποίησης που περιλαμβάνει τους γενετικούς αλγορίθμους (genetic algorithms) [30], τις στρατηγικές εξέλιξης (evolution strategies) [31], το γενετικό προγραμματισμό (genetic programming) [32] και τον εξελικτικό προγραμματισμό evolutionary programming. Η μέθοδος έχει βασιστεί στη θεωρία της Δαρβινικής εξέλιξης των ειδών [33][34].

Κατά τη Δαρβινική εξέλιξη των ειδών, νέα άτομα δημιουργούνται από συνδυασμό των γονιδίων των γονέων και μεταλλάξεις. Τα νέα άτομα είναι περισσότερα από τους γονείς, συνεπώς δεν μπορούν να επιβιώσουν όλα. Μέσω της φυσικής επιλογής, επιβιώνουν και αναπαράγονται τα καταλληλότερα άτομα. Η διαδικασία συνεχίζεται, με τα άτομα αυτά να αποτελούν γονείς για τη νέα γενιά [35].

Στον EA, τα άτομα αποτελούν το σύνολο των πιθανών λύσεων. Κάθε άτομο χαρακτηρίζεται από ένα χρωμόσωμα, με τα γονίδια του ατόμου. Βέλτιστο είναι το άτομο που παρουσιάζει τον καλύτερο συνδυασμό γονιδίων [36].

Κάθε γενιά πιθανών λύσεων αξιολογείται με κριτήρια καταλληλότητας. Τα πιο κατάλληλα άτομα διασταυρώνονται, δηλαδή συνδυάζουν τα γονιδιά τους, για να δημιουργήσουν απογόνους, όπως φαίνεται στο σχήμα 1.4. Οι απόγονοι κάθε γενιάς είναι πιθανό να υποστούν κάποια μετάλλαξη, δηλαδή κάποιο από τα γονίδια του χρωμοσώματός τους να τροποποιηθεί, όπως φαίνεται στο σχήμα 1.5. Περισσότερες πιθανότητες επιβίωσης και αναπαραγωγής έχουν οι καταλληλότεροι απόγονοι. Από τα άτομα αυτά νέα χρωμοσώματα προκύπτουν και η διαδικασία της εξέλιξης συνεχίζεται. Στόχος είναι η εύρεση του καλύτερου ατόμου από όλες τις γενιές που ολοκληρώθηκαν.



Σχήμα 1.4: Διαδικασία διασταύρωσης [37].



Σχήμα 1.5: Διαδικασία μετάλλαξης [37].

Μεγάλη πιθανότητα μετάλλαξης έχει ως αποτέλεσμα καλύτερη ανίχνευση και μικρή καλύτερη εκμετάλλευση.

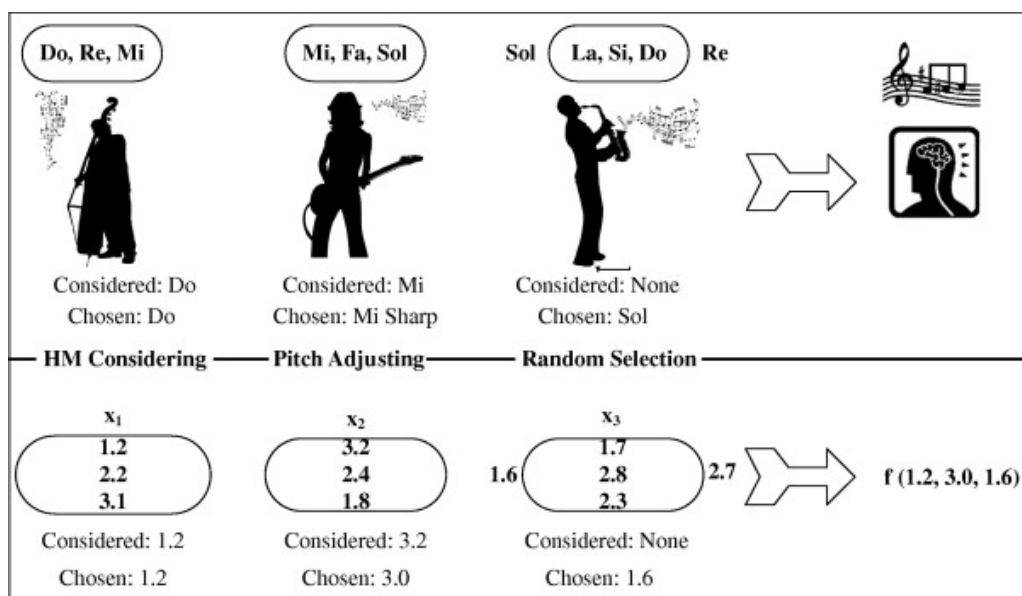
Harmony Search

Η Harmony Search (HS) αποτελεί μια μέθοδο πληθυσμιακής στοχαστικής βελτιστοποίησης, που αναπτύχθηκε από τους Z.W.Geem, J.H.Kim και G.V. Loganathan, το 2001 [7]. Κατά τη μέθοδο αυτή η διαδικασία σύνθεσης τζαζ μουσικής παραλληλίζεται με μια διαδικασία βελτιστοποίησης, όπου ο σκοπός είναι η βελτιστοποίηση της αρμονίας. Αρχικά ο αλγόριθμος αφορούσε βελτιστοποίηση για διακριτές μεταβλητές, στη συνέχεια ωστόσο, δημιουργήθηκε το ανάλογό του και για συνεχείς μεταβλητές [38]. Μέχρι τώρα, ο αλγόριθμος έχει εφαρμοστεί σε προβλήματα, όπως στο σχεδιασμό δικτύων διανομής νερού [39], στην εύρεση παραμέτρων φρεζαρίσματος [40], στην εξοικονόμηση ενέργειας [41], στην επίλυση sudoku [42] και σε πολλά άλλα. Έχουν γίνει πολλές δημοσιεύσεις σχετικά με βελτιώσεις της μεθόδου HS [43][44][45][46][47], και υβριδικές μεθόδους της HS με άλλες μεθόδους βελτιστοποίησης. [48][49][50][51][52].

Η αναλογία εύρεσης μιας αρμονίας στη τζαζ μουσική με τους αλγορίθμους βελτιστοποίησης φαίνεται στο σχήμα 1.6. Στη μουσική, αναζητείται ο βέλτιστος συνδυασμός νότων από κάθε μουσικό όργανο για τη δημιουργία μιας αρμονίας. Αντίστοιχα, οι αλγόριθμοι βελτιστοποίησης αναζητούν το σύνολο μεταβλητών σχεδιασμού που θα δώσει το βέλτιστο αποτέλεσμα. Στη μουσική, ο συνδυασμός των νότων αξιολογείται από κριτήρια αισθητικής, ενώ στους αλγορίθμους βελτιστοποίησης από τη συνάρτηση-στόχο. Τέλος, στη μουσική μεταβάλλονται οι νότες ή το pitch, η συχνότητα ταλάντωσης του ηχητικού κύματος της κάθε νότας, για την εύρεση της αρμονίας, ενώ στους αλγορίθμους βελτιστοποίησης μεταβάλλονται οι τιμές των μεταβλητών.

Στον αλγόριθμο HS χρησιμοποιούνται τρεις βασικές σταθερές. Η Harmony Memory Size (HMS), που είναι το μέγεθος της βάσης δεδομένων που περιέχει τις καλύτερες λύσεις που έχουν βρεθεί στις, έως εκείνη τη στιγμή, αξιολογήσεις, η Harmony Memory Consideration Rate (HMCR), που είναι η πιθανότητα να δημιουργηθεί νέο, τυχαίο άτομο στον πληθυσμό και η Pitch Adjustment Rate (PAR), που είναι η πιθανότητα να μεταβληθούν εντός κάποιου μικρού εύρους οι τιμές του νέου ατόμου [38]. Το εύρος μεταβολής αποτελεί μια επιπλέον σταθερά, που ονομάζεται εύρος ζώνης (bandwidth - bw). Στην περίπτωση των διακριτών μεταβλητών το εύρος είναι αύξηση ή μείωση κατά μια μονάδα, σταθερό για κάθε πρόβλημα. Αντίθετα, στην περίπτωση των συνεχών μεταβλητών, αποτελεί μια ακόμα μεταβλητή που πρέπει να ορισθεί από τον χρήστη. Τέλος, ορίζεται ο μέγιστος αριθμός επαναλήψεων, ίδιος με τον αριθμό αξιολογήσεων, για τον τερματισμό της επαναληπτικής διαδικασίας.

Η ικανότητα της μεθόδου HS να πετύχει μια ισορροπία μεταξύ εκμετάλλευσης και ανίχνευσης είναι ο κύριος λόγος επιτυχίας της. Η εκμετάλλευση επιτυγχάνεται μέσω της PAR και του bw, ενώ η ανίχνευση μέσω της HMCR.



Σχήμα 1.6: Αναλογία εύρεσης αρμονίας με αλγόριθμο βελτιστοποίησης [53].

1.3 Στόχος διπλωματικής εργασίας

Στόχος της διπλωματικής εργασίας είναι η εξέταση του αλγορίθμου HS με χρήση στοιχείων που έχουν αναπτυχθεί και χρησιμοποιούνται σε αλγορίθμους βελτιστοποίησης της ΜΠΥΡ&Β, όπως μεταπρότυπα, μέθοδοι διαχείρισης περιορισμών και μετώπων Pareto καθώς και η τροποποίηση της δομής του αλγορίθμου για επίτευξη καλύτερων αποτελεσμάτων.

Αρχικά προγραμματίζεται ο αλγόριθμος HS, σε γλώσσα C++, χρησιμοποιώντας κλάσεις και δείκτες ανάλογους αυτών του λογισμικού EASY για συμβατότητα. Στη συνέχεια στον κλασικό αλγόριθμο HS γίνονται κατάλληλες τροποποιήσεις για να χρησιμοποιηθούν τα στοιχεία της ΜΠΥΡ&Β. Τέλος, δοκιμάζονται σε αυτόν τελεστές, οι οποίοι είναι είτε εμπνευσμένοι από άλλες μεθόδους βελτιστοποίησης είτε νέοι, για την εύρεση καλύτερων λύσεων. Όσοι δίνουν καλύτερα αποτελέσματα, εφαρμόζονται.

Ο νέος αλγόριθμος που δημιουργείται μετά από αλλαγές, προσθήκες, και χρήση στοιχείων άλλων αλγορίθμων, αποφασίστηκε να λάβει ένα νέο όνομα, Choreography Composition-Like Optimization, αφού η νέα διαδικασία βελτιστοποίησης μιμείται τη διαδικασία σύνθεσης μίας χορογραφίας.

Εάν τα αποτελέσματα που θα βρεθούν είναι ικανοποιητικά σε μια ευρεία περιοχή εφαρμογών, πιθανόν να εξεταστεί το ενδεχόμενο να προσαρτηθεί ο αλγόριθμος αυτός στο λογισμικό που η ΜΠΥΡ&Β χρησιμοποιεί για βελτιστοποίηση.

Κεφάλαιο 2

Βάσεις Νέου Αλγορίθμου

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά οι μέθοδοι βελτιστοποίησης από τις οποίες υιοθέτησε στοιχεία η μέθοδος βελτιστοποίησης που δημιουργήθηκε. Παρουσιάζεται η μέθοδος HS, στην οποία βασίστηκε η νέα μέθοδος βελτιστοποίησης, και ο (μ, λ) εξελικτικός αλγόριθμος, με τις προσθήκες που έχουν γίνει σε αυτόν από τη ΜΠΥΡ&Β, για τη χρήση του στο λογισμικό EASY.

2.1 Harmony Search

Η νέα μέθοδος που δημιουργήθηκε βασίστηκε στην πληθυσμιακή στοχαστική μέθοδο HS, το θεωρητικό υπόβαθρο της οποίας παρουσιάστηκε στην ενότητα 1.2.3. Στην ενότητα αυτή θα παρουσιαστεί αναλυτικά ο αλγόριθμος της μεθόδου αυτής.

2.1.1 Αλγόριθμος βελτιστοποίησης HS

Τα βήματα βελτιστοποίησης για τον αλγόριθμο HS παρουσιάζονται παρακάτω:

Αρχικοποίηση της βάσης δεδομένων Harmony Memory (HM) με HMS τυχαία άτομα.

Αξιολόγηση των HMS τυχαίων ατόμων από λογισμικό αξιολόγησης.

Εύρεση χειρότερου ατόμου στην HM.

Δημιουργία και αξιολόγηση ενός νέου ατόμου με την τεχνική HS (βλ 2.1.2).

Αν το νέο άτομο είναι καλύτερο από το χειρότερο στη HM, τότε:

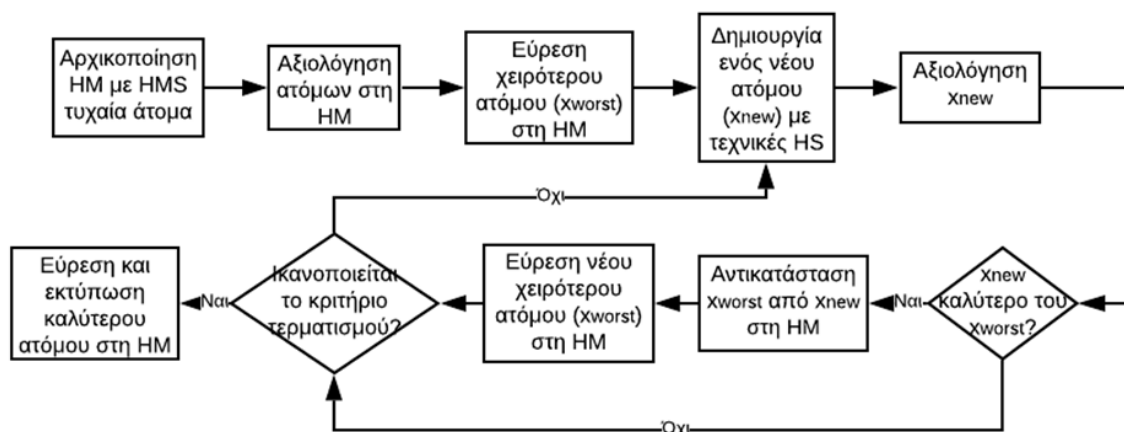
Αντικατάσταση του χειρότερου ατόμου από το νέο.

Εύρεση χειρότερου ατόμου στην HM.

Επαναλήψεις μέχρι την ικανοποίηση του κριτηρίου τερματισμού.

Εκτύπωση της καλύτερης λύσης.

Στη συνέχεια παρουσιάζεται το διάγραμμα ροής της μεθόδου που αναλύθηκε παραπάνω.



Σχήμα 2.1: Βήματα υλοποίησης HS.

2.1.2 Αλγόριθμος δημιουργίας νέου ατόμου με την τεχνική HS

Η δημιουργία ενός νέου ατόμου, γίνεται ως εξής. Για καθεμία από τις μεταβλητές σχεδιασμού του νέου ατόμου, επιλέγεται η αντίστοιχη μεταβλητή τυχαίου ατόμου από την HM, με πιθανότητα HMCR, εναλλακτικά λαμβάνεται μια τυχαία τιμή εντός του πεδίου ορισμού. Κάθε μεταβλητή που δημιουργήθηκε στο προηγούμενο βήμα, μεταβάλλεται κατά μια τυχαία τιμή εντός του bw, με πιθανότητα PAR, εναλλακτικά η τιμή της μεταβλητής αυτής παραμένει σταθερή [7].

Παρακάτω παρουσιάζεται σε μορφή αλγορίθμου η διαδικασία δημιουργίας του νέου διανύσματος μεταβλητών σχεδιασμού [54].

Για κάθε μεταβλητή x_i , όπου $i = 1, 2, \dots, N$

Εάν $rand(0, 1) \leq HMCR$ τότε $x_{new,i} = x_i^j$, όπου j τυχαία τιμή στο $1, 2, \dots, HMS$.

Εάν $rand(0, 1) \leq PAR$ τότε $x_{new,i} = x_{new,i} + rand(-bw/2, bw/2)$

Εάν το $x_{new,i}$ είναι εκτός ορίων, παίρνει την τιμή του εκάστοτε ορίου.

Αλλιώς, εάν $\text{rand}(0, 1) \geq \text{HMCR}$ τότε $x_{\text{new},i} = L$ όπου L τυχαία τιμή στο πεδίο ορισμού της μεταβλητής.

Επιστροφή νέου ατόμου $X_{\text{new}} = x_{\text{new},1}, x_{\text{new},2}, \dots, x_{\text{new},N}$.

2.2 (μ,λ) Εξελικτικός Αλγόριθμος

Η μέθοδος που δημιουργήθηκε στο πλαίσιο αυτής της διπλωματικής εργασίας υιοθέτησε στοιχεία από το λογισμικό EASY (Evolutionary Algorithms SYstem). Το λογισμικό αυτό βελτιστοποιεί με, μεταξύ άλλων μεθόδων, (μ,λ) EA (βλ. 1.2.3), οι οποίοι στη συνέχεια αποτέλεσαν βάση σύγκρισης για τη νέα μέθοδο που δημιουργήθηκε. Στην ενότητα αυτή παρουσιάζονται οι (μ,λ) EA, συνεχών μεταβλητών, για τους τελεστές του λογισμικού EASY που επιλέχθηκαν να χρησιμοποιηθούν για τη σύγκριση με το νέο λογισμικό.

2.2.1 Λογισμικό EASY

Το λογισμικό EASY αποτελεί ένα πολυδοκιμασμένο εργαλείο βελτιστοποίησης που έχει δημιουργηθεί και αναπτύσσεται από τη ΜΠΥΡ&Β [2][55][56][57]. Πλήθος διπλωματικών και διδακτορικών εργασιών έχουν γίνει πάνω σε αυτό τα τελευταία είκοσι χρόνια.

Το λογισμικό EASY αποτελεί ένα γενικής χρήσης εργαλείο για βελτιστοποίηση. Χρησιμοποιείται για βελτιστοποίηση προβλημάτων ενός ή περισσότερων στόχων και έχει τη δυνατότητα χειρισμού περιορισμών με μεθόδους ποινών [58].

Ιδιαίτερες τεχνικές έχουν αναπτυχθεί για τη μείωση του χρόνου υπολογισμού. Πιο συγκεκριμένα, το λογισμικό EASY δίνει τη δυνατότητα χρήσης μεταπροτύπων, όπως τεχνητών νευρωνικών δικτύων και μεθόδων μαθηματικής παρεμβολής για προσεγγιστικό, μικρού υπολογιστικού κόστους υπολογισμού των συναρτήσεων στόχων. Επιπλέον, δίνεται η δυνατότητα παράλληλης αξιολόγησης [59].

Το λογισμικό EASY υποστηρίζει στοχαστικές μεθόδους βελτιστοποίησης. Η κωδικοποίηση των μεταβλητών μπορεί να είναι πραγματική ή δυαδική.

2.2.2 (μ,λ) EA, Μονοκριτηριακή Βελτιστοποίηση

Στο πλαίσιο της εργασίας αυτής, η κωδικοποίηση των μεταβλητών επιλέγεται να είναι πραγματική. Οι αλγόριθμοι που παρουσιάζονται αφορούν συναρτήσεις ελαχιστοποίησης. Στις περιπτώσεις που το πρόβλημα περιλαμβάνει συναρτήσεις καταλληλότητας, οι

οποίες πρέπει να μεγιστοποιηθούν, γίνεται αντιστροφή προσήμων, ώστε να αναζητούνται οι αντίθετες τιμές τους και να αναχθούν σε συναρτήσεις ελαχιστοποίησης.

Στους (μ,λ) EA, με μ συμβολίζεται ο πληθυσμός των γονέων και με λ ο πληθυσμός των απογόνων. Η διαδικασία βελτιστοποίησης είναι η ακόλουθη [60].

Αρχικοποίηση (Initialization):

Αρχικά δημιουργούνται λ πλήθους τυχαία άτομα. Κάθε άτομο είναι ένα σύνολο από τις μεταβλητές σχεδιασμού του εκάστοτε προβλήματος.

Αξιολόγηση πληθυσμού (Evaluation):

Τα λ άτομα αξιολογούνται από το λογισμικό αξιολόγησης του εκάστοτε προβλήματος. Το λογισμικό αυτό επιστρέφει τις πραγματικές τιμές των συναρτήσεων - στόχων του προβλήματος για κάθε σετ μεταβλητών σχεδιασμού.

Ελιτισμός (Elitism):

Τα άτομα με τις καλύτερες τιμές στις συναρτήσεις - στόχους συλλέγονται σε μια βάση δεδομένων. Λίγα από τα επίλεκτα άτομα, με τις καλύτερες τιμές συνάρτησης - στόχου, αντικαθιστούν άτομα των λ πλήθους απογόνων.

Επιλογή πληθυσμού γονέων (Parent selection):

Από τα λ σε πλήθος άτομα που έχουν δημιουργηθεί, επιλέγονται μ για να χρησιμοποιηθούν ως γονείς. Μεγαλύτερη πιθανότητα να επιλεγθούν έχουν τα άτομα με καλύτερες τιμές συνάρτησης κόστους.

Διασταύρωση (Crossover):

Η νέα γενιά λ πλήθους απογόνων προκύπτει από λ διασταυρώσεις γονέων. Για κάθε νέο άτομο χρησιμοποιούνται ρ άτομα γονέων, όπου $\rho \geq 2$, και συνήθως $\rho=3$. Ο τελεστής διασταύρωσης με τον οποίο γίνεται η σύγκριση είναι ο Simulated Binary Crossover (SBX) [61][62].

Ο τελεστής SBX αποτελείται από δύο όρους, τη μέση τιμή (average factor) και τη διάδοση (spread factor). Στη σχέση 2.1 φαίνεται η σχέση υπολογισμού κάθε μεταβλητής του νέου ατόμου

$$x_i^{new} = \begin{cases} \frac{x_i^1 + x_i^{random}}{2} - \frac{\beta}{2} \frac{x_i^1 + x_i^{random}}{2}, & rand(0,1) < 0.5 \\ \frac{x_i^1 + x_i^{random}}{2} + \frac{\beta}{2} \frac{x_i^1 + x_i^{random}}{2}, & rand(0,1) \geq 0.5 \end{cases} \quad (2.1)$$

όπου x_i^1 η αντίστοιχη μεταβλητή σχεδιασμού από τα ρ άτομα που επιλέχθηκαν, x_i^{random} η αντίστοιχη μεταβλητή τυχαία επιλεγμένου ατόμου από το σύνολο ρ, διαφορετικό για κάθε μεταβλητή σχεδιασμού του νέου ατόμου, β ο παράγοντας διάδοσης, και $rand(0,1)$ τυχαία τιμή στο διάστημα [0,1], διαφορετική για κάθε μεταβλητή σχεδιασμού.

Ο παράγοντας διάδοσης υπολογίζεται από τη σχέση

$$\beta = \begin{cases} (2rand(0, 1))^n, & rand(0, 1) < 0.5 \\ (\frac{1}{2rand(0,1)})^{n+2}, & rand(0, 1) \geq 0.5 \end{cases} \quad (2.2)$$

όπου η σταθερά n ορίζεται από τον χρήστη και συνήθως παίρνει τιμές στο διάστημα $[2,5]$.

Μετάλλαξη (Mutation):

Κατά τη μετάλλαξη, οι μεταβλητές κάθε στοιχείου της νέας γενιάς απογόνων που προέκυψαν κατά τη διασταύρωση, έχουν μια μικρή πιθανότητα να μεταβληθούν κατά μία μικρή τιμή, όπως φαίνεται στη σχέση 2.3 [62]

$$x_i^{new} = \begin{cases} x_i^{new}, & P > rand(0, 1) \\ M(x, D), & P \leq rand(0, 1) \end{cases} \quad (2.3)$$

όπου P πιθανότητα που ορίζει ο χρήστης, $rand(0,1)$ τυχαία τιμή στο διάστημα $[0,1]$ και το $M(x, D)$ υπολογίζεται από τη σχέση 2.4

$$M(x, D) = \begin{cases} x_i + D(g, U_i - b), & rand_1(0, 1) > 0.5 \\ x_i + D(g, b - L_i), & rand_1(0, 1) \leq 0.5 \end{cases} \quad (2.4)$$

όπου $rand_1(0, 1)$ νέα τυχαία τιμή στο διάστημα $[0,1]$, g ο αριθμός των ολοκληρωμένων γενιών, U_i και L_i τα ανώτερα και κατώτερα όρια της μεταβλητής και $D(g, y)$ να υπολογίζεται από τη σχέση 2.5

$$D(g, y) = y \times rand_2(0, 1) \times (1 - g/g_{max})^{0.2} \quad (2.5)$$

όπου $rand_1(0, 1)$ διαφορετική τυχαία τιμή στο διάστημα $[0,1]$ και g_{max} ο μέγιστος αριθμός γενιών του ΕΑ.

Έλεγχος κριτηρίου τερματισμού (Stopping criteria check):

Στη συνέχεια, γίνεται νέα αξιολόγηση του πληθυσμού και η διαδικασία επαναλαμβάνεται μέχρι να ικανοποιηθεί το κριτήριο τερματισμού, το οποίο στην περίπτωση που εξετάζεται αφορά πλήθος αξιολογήσεων.

2.2.3 (μ,λ) Πολυκριτηριακή Βελτιστοποίηση

Στην περίπτωση των περισσότερων της μίας συνάρτησης στόχου, στην παραπάνω διαδικασία γίνονται ορισμένες προσθήκες για κατάλληλη διαχείριση των συναρτήσεων.

Όπως αναφέρθηκε στην ενότητα 1.2.1, για σύγκριση σε συναρτήσεις πολλών μεταβλητών, απαραίτητη είναι η δημιουργία μετώπων Pareto. Υπάρχουν πολλές μέθοδοι διαχείρισης μη-κυριαρχούμενων λύσεων, αλλά αυτή που χρησιμοποιείται από το λογισμικό EASY είναι η SPEA II (Strength Pareto Evolutionary Algorithm II). Η μέθοδος υπολογισμού του μετώπου είναι η εξής [63]:

Αρχικά υπολογίζεται η τιμή της δύναμης (strength) για κάθε άτομο, ως το πλήθος των ατόμων που κυριαρχούνται από το άτομο αυτό, όπως φαίνεται στη σχέση 2.6

$$S_i = \text{Num of dominated points from member } i \quad (2.6)$$

Στη συνέχεια, υπολογίζεται η απόσταση κάθε ατόμου από τα γειτονικά του. Οι αποστάσεις αυτές τοποθετούνται σε αύξουσα σειρά. Από τις αποστάσεις αυτές επιλέγεται αυτή της θέσης k , όπου k η τετραγωνική ρίζα του πλήθους των ατόμων, και η τιμή αυτή συμβολίζεται με d_i^k . Η τιμή της πυκνότητας (density) είναι ο λόγος της μονάδας προς την απόσταση αυτή αυξημένη κατά 2, όπως φαίνεται στη σχέση 2.7

$$D_i = \frac{1}{d_i^k + 2} \quad (2.7)$$

Υπολογίζεται μια βοηθητική ποσότητα (raw fitness), ως το άθροισμα όλων των τιμών δύναμης των ατόμων τα οποία κυριαρχούν στο υπο εξέταση άτομο, όπως φαίνεται στη σχέση 2.8

$$R_i = \sum S_{\text{dominators}} \quad (2.8)$$

Τέλος, υπολογίζεται η τιμή καταλληλότητας (utility), με τα περισσότερα «κατάλληλα» άτομα να είναι αυτά με τις μικρότερες τιμές καταλληλότητας. Αποτελείται από το άθροισμα της τιμής πυκνότητας με τη βοηθητική ποσότητα, όπως φαίνεται στη σχέση 2.9

$$\phi_i = D_i + R_i \quad (2.9)$$

Από τα παραπάνω παρατηρείται ότι τα άτομα στο μέτωπο Pareto έχουν μηδενική τιμή βοηθητικής ποσότητας. Άτομα του μετώπου που απέχουν μεγαλύτερες αποστάσεις από τα υπόλοιπα άτομα του μετώπου, επομένως οι τιμές τους είναι απαραίτητες για την οριοθέτηση του μετώπου, έχουν μικρές τιμές πυκνότητας.

Στη διαδικασία που παρουσιάστηκε στην ενότητα 2.2.2, γίνονται ορισμένες προσθήκες για να είναι σε θέση να διαχειριστεί περισσότερες της μίας συναρτήσεις - στόχους. Συγκεκριμένα, προστίθεται ένα νέο βήμα ανάμεσα στην Αξιολόγηση πληθυσμού και στον Ελιτισμό, η εύρεση καταλληλότητας.

Εύρεση Καταλληλότητας (Fitness Assignment):

Υπολογίζεται η τιμή της συνάρτησης ϕ , ίση με τη συνάρτηση που έχει προκύψει από την αξιολόγηση στην περίπτωση της μιας συνάρτησης - στόχου, και υπολογισμένη με χρήση της μεθόδου SPEAII, στην περίπτωση των περισσοτέρων.

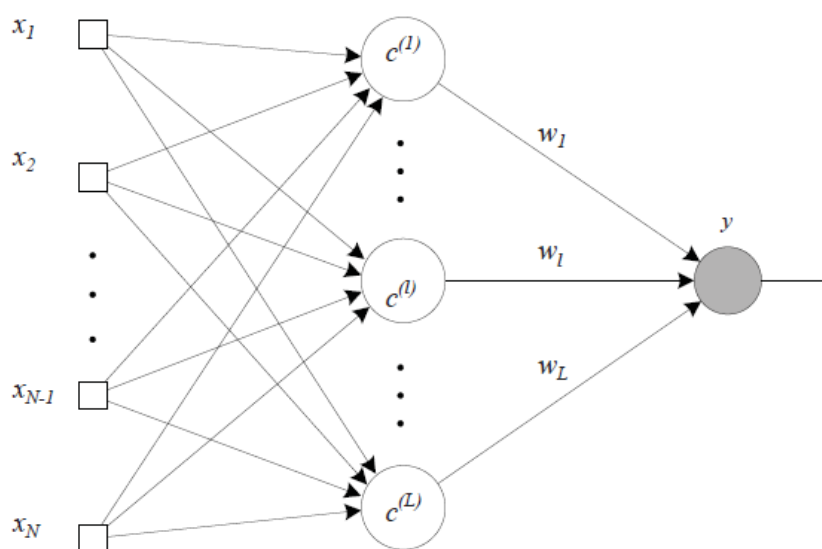
Στη συνέχεια η διαδικασία συνεχίζεται όπως παρουσιάζεται στην 2.2.2, χρησιμοποιώντας την τιμή ϕ αντί της τιμής της συνάρτησης στόχου που περιγράφηκε.

2.2.4 ΕΑ Υποβοηθούμενοι από Μεταπρότυπα

Στην ενότητα 1.2.2 έχουν παρουσιαστεί τα μεταπρότυπα. Στην περίπτωση χρήσης μεταπροτύπων από τους ΕΑ (Metamodel-Assisted EAs - MAEAs) δεν αξιολογούνται από το κοστοβόρο λογισμικό αξιολόγησης όλα τα άτομα που δημιουργούνται από τις διαδικασίες διασταύρωσης και μετάλλαξης, αλλά μόνο αυτά για τα οποία προβλέπεται καλύτερη τιμή συναρτήσεων - στόχων [4]. Το λογισμικό EASY δίνει τη δυνατότητα χρήσης on-line μεταπροτύπων.

Χρησιμοποιούνται Τεχνητά Νευρωνικά Δίκτυα (Radial Basis Function Networks - RBFN) [64][65].

Τα RBFN αποτελούνται από τρία επεξεργαστικά επίπεδα, το επίπεδο εισόδου, το κρυφό επίπεδο και το επίπεδο εξόδου, όπως φαίνεται στο σχήμα 2.2.



Σχήμα 2.2: RBFN τριών επιπέδων με έναν νευρώνα εξόδου [66].

Η διαδικασία προσεγγιστικού υπολογισμού των συναρτήσεων - στόχων είναι η ακόλουθη.

Αρχικά, για τον υπολογισμό κάθε νέου ατόμου επιλέγονται, από τα άτομα που κατά τη

διαδικασία της βελτιστοποίησης έχουν υπολογιστεί με το λογισμικό αξιολόγησης, αυτά των οποίων οι μεταβλητές σχεδιασμού απέχουν τις ελάχιστες αποστάσεις από το νέο άτομο. Οι τιμές αυτές είναι οι x_1 έως x_N του επιπέδου εισόδου του σχήματος 2.2.

Στη συνέχεια, κάθε μεταβλητή σχεδιασμού των ατόμων που επιλέχθηκαν χρησιμοποιείται από τον αντίστοιχο νευρώνα της μεταβλητής στο κρυφό επίπεδο. Οι νευρώνες του επιπέδου αυτού είναι ίσοι σε πλήθος με τις μεταβλητές σχεδιασμού και στον καθένα έχουν αντιστοιχηθεί N πλήθους μεταβλητές σχεδιασμού. Οι νευρώνες αυτοί ονομάζονται Κέντρα Ακτινικής Βάσης.

Σε κάθε νευρώνα γίνεται ένας μη γραμμικός μετασχηματισμός, διαφορετικός ανάλογα τον τύπο των μεταπροτύπων που επιλέγεται, που εξαρτάται από την απόσταση των μεταβλητών των ατόμων που επιλέχθηκαν από αυτές του νέου ατόμου.

Στο επίπεδο εξόδου, αθροίζονται οι τιμές των εξισώσεων που βρέθηκαν στο κρυφό επίπεδο, σταθμισμένες με τις τιμές των συνάψεων w_i .

Το αποτέλεσμα y είναι η προσεγγιστικά υπολογισμένη τιμή της συνάρτησης - στόχου του νέου ατόμου.

Προστίθενται τρία βήματα στη διαδικασία της ενότητας 2.2.2 για να χρησιμοποιηθούν on-line μεταπρότυπα όπως αυτά αναλύθηκαν. Αρχικά, πριν ξεκινήσει η διαδικασία της βελτιστοποίησης γίνονται αξιολογήσεις, ώστε να δημιουργηθεί μια βάση δεδομένων με επαρκή άτομα για τον προσεγγιστικό υπολογισμό νέων. Επιπλέον, προστίθενται τα βήματα της προσεγγιστικής αξιολόγησης πληθυσμού και της κατάταξής τους, πριν την πραγματική αξιολόγηση του πληθυσμού. Τα βήματα αυτά παρουσιάζονται αναλυτικά παρακάτω.

Δημιουργία Βάσης Δεδομένων Αξιολογημένων Ατόμων:

Δημιουργούνται T τυχαία άτομα, όπου ο αριθμός T ορίζεται από τον χρήστη, η τιμή των οποίων υπολογίζεται από το λογισμικό αξιολόγησης.

Προσεγγιστική Αξιολόγηση Πληθυσμού:

Από τα άτομα που έχουν υπολογιστεί από το λογισμικό αξιολόγησης, προσεγγίζεται η τιμή των λ ατόμων που δημιουργήθηκαν.

Εύρεση Καλύτερων Ατόμων:

Τα άτομα που αξιολογήθηκαν προσεγγιστικά κατατάσσονται σε αύξουσα σειρά με βάση την τιμή της συνάρτησης - στόχου τους. Τα λ πρώτα σε κατάταξη, δηλαδή καλύτερα, χρησιμοποιούνται κατά την αξιολόγηση με το λογισμικό αξιολόγησης.

Τέλος, κατά την αξιολόγηση, η διαδικασία της ενότητας 2.2.2 μεταβάλλεται ελαφρώς, με λ άτομα να αξιολογούνται στην περίπτωση αυτή και όλα τα άτομα που αξιολογούνται να προστίθενται στη Βάση Δεδομένων Αξιολογημένων Ατόμων.

2.2.5 EAs και MAEAs σε Προβλήματα με Περιορισμούς

Το λογισμικό EASY δίνει επιπλέον δυνατότητα διαχείρισης περιορισμών των συναρτήσεων στόχων με τη μέθοδο των ποινών στους εξελικτικούς αλγορίθμους με ή χωρίς χρήση μεταπροτύπων. Η μέθοδος αυτή παρουσιάζεται παρακάτω.

Αρχικά, οι περιορισμοί ανισο-ισότητας γράφονται στην παρακάτω μορφή

$$C_i(\vec{x}) \leq C_{THR,i}, \quad i = 1, \dots, M \quad (2.10)$$

όπου C_i είναι η συνάρτηση περιορισμών, \vec{x} το διάνυσμα των μεταβλητών σχεδιασμού και $C_{THR,i}$ η μέγιστη επιτρεπτή τιμή της συνάρτησης.

Στην περίπτωση περιορισμών ισότητας, η εξίσωση γράφεται ως

$$|C_i(\vec{x}) - C_{THR,i}| \leq \epsilon, \quad i = 1, \dots, M \quad (2.11)$$

όπου ϵ μικρή ποσότητα που ορίζει ο χρήστης, με μια συνήθη τιμή το 10^{-4}

Οι περιορισμοί ισότητας και πάλι μοντελοποιούνται ως δυο περιορισμοί ανισο-ισότητας

$$\begin{aligned} C_i(\vec{x}) - \epsilon &\leq C_{THR,i}, & i = 1, \dots, M \\ -C_i(\vec{x}) - \epsilon &\leq -C_{THR,i}, & i = 1, \dots, M \end{aligned} \quad (2.12)$$

Συχνά, στην αρχή της διαδικασίας βελτιστοποίησης, ορισμένες λύσεις που θα συνέβαλλαν στη δημιουργία ατόμων εντός περιορισμών και με βέλτιστες τιμές, τίθενται εκτός περιορισμών και παύουν να χρησιμοποιούνται. Για το λόγο αυτό, εισάγεται μια νέα σταθερά, η σταθερά ποινής θανάτου ($C_{DTH,i}$).

Όταν η τιμή της συνάρτησης περιορισμών λάβει τιμές μεγαλύτερες της τιμής ποινής θανάτου, η τιμή της συνάρτησης στόχου αυξάνεται κατά μια άπειρη ποσότητα, επομένως απορρίπτεται κατά την εξέλιξη που ακολουθεί.

Όταν η τιμή της συνάρτησης των περιορισμών είναι μεταξύ της μέγιστης επιτρεπτής και της τιμής θανάτου, προστίθεται μια ποινή (penalty) στη συνάρτηση - στόχο, επιτρέποντας σε λύσεις με μικρές τιμές - στόχους να παραμείνουν στη βάση δεδομένων, παρά την παραβίαση των περιορισμών. Η ποινή υπολογίζεται από τη σχέση 2.13, όπου PGF είναι ο τελεστής ενίσχυσης της ποινής, που επιλέγεται από τον χρήστη. Μία συνήθης τιμή του είναι το 3.

$$Penalty_i = \exp(PGF_i \times \frac{C_i - C_{THR,i}}{C_{DTH,i} - C_i}) \quad (2.13)$$

Στην περίπτωση αυτή προστίθεται ένα νέο βήμα στη διαδικασία της ενότητας 2.2.2, μετά την Αξιολόγηση Πληθυσμού και πριν την Εύρεση Καταλληλότητας.

Προσθήκη Ποινών:

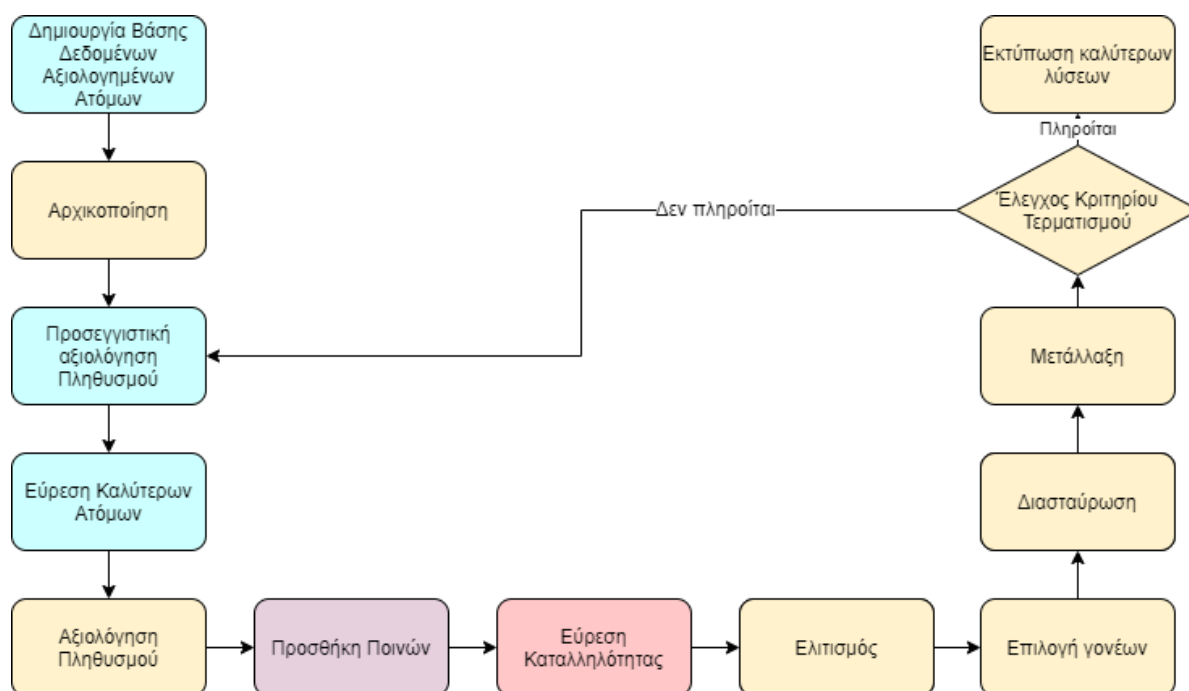
Στην περίπτωση που το πρόβλημα υπόκειται σε περιορισμούς, οι τιμές των συναρτήσεων - στόχων του προβλήματος αυξάνονται κατά μια ποινή, όπως αυτή υπολογίστηκε από τη μέθοδο των ποινών.

Στη συνέχεια, κατά την Εύρεση Καταλληλότητας, χρησιμοποιείται η τιμή στην οποία έχει προστεθεί ποινή.

2.2.6 Διάγραμμα Ροής EASY

Όσα αναφέρθηκαν στα κεφάλαια 2.2.2 έως 2.2.5 για τη διαδικασία βελτιστοποίησης του EASY συνοψίζονται στο διάγραμμα ροής 2.3.

Σε μπεζ πλαίσιο παρουσιάζονται τα βήματα της ενότητας 2.2.2, σε ροζ της 2.2.3, σε μπλε της 2.2.4 και σε μοβ της 2.2.5.



Σχήμα 2.3: Διάγραμμα ροής λογισμικού EASY.

Κεφάλαιο 3

Choreography Composition-Like Optimization

Στο κεφάλαιο αυτό αναλύεται η μετάβαση από HS σε Choreography Composition-Like Optimization (CCLO) και παρουσιάζεται η νέα μέθοδος. Στη συνέχεια, η νέα μέθοδος συγκρίνεται με τον (μ,λ) εξελικτικό αλγόριθμο του EASY, από τον οποίο χρησιμοποιήθηκαν στοιχεία και του οποίου τα αποτελέσματα συγκρίνονται στα επόμενα κεφάλαια με τα αντίστοιχα του CCLO.

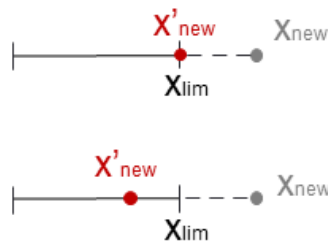
3.1 Διαφορές HS - CCLO

Σκοπός της διπλωματικής εργασίας αυτής ήταν η δημιουργία μιας πληθυσμιακής στοχαστικής μεθόδου βελτιστοποίησης, βασισμένη στη μέθοδο HS. Η νέα μέθοδος ονομάστηκε Choreography Composition-Like Optimization, με την ονομασία να προέρχεται από το χορό, αφού η διαδικασία βελτιστοποίησης ακολουθεί τη λογική της διαδικασίας σύνθεσης μιας νέας χορογραφίας.

Οι αλλαγές που έγιναν στον αλγόριθμο Harmony Search είναι οι ακόλουθες:

- Στον HS, σε κάθε νέα αναζήτηση λύσεων δημιουργείται ένα νέο άτομο από τα στοιχεία του πληθυσμού HM. Ωστόσο, με δοκιμές βρέθηκε ότι επιτυγχάνονται καλύτερα αποτελέσματα εάν δημιουργούνται ταυτόχρονα περισσότερα του ενός στοιχεία από τον πληθυσμό, και με την ίδια διαδικασία, όσα είναι καλύτερα από στοιχεία του πληθυσμού HM, να πάρουν τη θέση των χειρότερων στον πληθυσμό.
- Στον αλγόριθμο HS, όταν οι μεταβλητές που δημιουργούνται ξεπερνούν τα όρια σχεδιασμού τους, η τιμή τους εξισώνεται με το όριο που υπερέβησαν. Αυτό

έχει ως αποτέλεσμα όλες οι μεταβλητές των νέων πιθανών λύσεων που ξεπεράσαν το ίδιο όριο να έχουν ακριβώς την ίδια τιμή. Για να αποφευχθεί αυτό, στο νέο αλγόριθμο γίνεται αναδίπλωση με την ακόλουθη τεχνική. Η απόλυτη τιμή της υπέρβασης αφαιρείται από τη νέα μεταβλητή στην περίπτωση του άνω ορίου και προστίθεται στην περίπτωση του κάτω, πολλαπλασιασμένη με την ποσότητα $1 + rand(0, 1)$, όπου $rand(0, 1)$ τυχαίος αριθμός στο διάστημα $[0,1]$. Σε περίπτωση που η αναδίπλωση έχει ως αποτέλεσμα να ξεπεραστεί το αντίθετο όριο, η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί λύση εντός της εφικτής περιοχής. Η διαδικασία αυτή φαίνεται στο σχήμα 3.1.



Σχήμα 3.1: Διαχείριση μεταβλητών που ξεπερνούν τα όρια σχεδιασμού τους, με τον αλγόριθμο HS (πάνω) και για τον νέο αλγόριθμο (κάτω). Με x_{lim} συμβολίζεται το όριο της μεταβλητής, με x_{new} η τιμή που η μεταβλητή είχε, και με x'_{new} η τιμή που θα χρησιμοποιηθεί.

- Το εύρος (bw) μέσα στο οποίο μεταβάλλεται η τιμή της νέας μεταβλητής που έχει δημιουργηθεί, αν η τυχαία πιθανότητα που προκύπτει είναι μικρότερη της PAR, στον αλγόριθμο HS ορίζεται από τον χρήστη και είναι σταθερό. Αυτό συμβαίνει επειδή ο αλγόριθμος HS όταν δημιουργήθηκε αφορούσε αποκλειστικά διακριτές μεταβλητές, στις οποίες το εύρος αυτό ήταν σταθερό και ίσο με τη μονάδα [7]. Στο νέο αλγόριθμο, η τιμή του ορίου μειώνεται καθώς αυξάνονται οι αξιολογήσεις του αλγόριθμου, μέσω της συνάρτησης $bw'(Ev) = bw \times e^{-Ev/Ev.Max}$, όπου bw σταθερά που ορίζεται από τον χρήστη, Ev οι αξιολογήσεις που έχουν πραγματοποιηθεί και $Ev.Max$ ο μέγιστος αριθμός αξιολογήσεων που έχει ορίσει ο χρήστης.
- Στον αλγόριθμο HS, για πιθανότητα μικρότερη από HMCR, η τιμή της μεταβλητής που εξετάζεται λαμβάνει την τιμή της αντίστοιχης μεταβλητής ενός τυχαία επιλεγμένου ατόμου στην HM. Στο νέο αλγόριθμο, η τιμή της μεταβλητής αυτής προκύπτει ως γραμμικός συνδυασμός δύο τυχαίων μεταβλητών x_1, x_2 της HM, δηλαδή $x' = rand(0, 1) \times x_1 + (1 - rand(0, 1)) \times x_2$, όπου $rand(0, 1)$ τυχαίος αριθμός στο διάστημα $[0,1]$.
- Για να αποφευχθεί περιττή χρήση του κοστοβόρου λογισμικού αξιολόγησης, στον CCLO χρησιμοποιούνται τεχνητά νευρωνικά δίκτυα. Με χρήση αυτών γίνονται προσεγγιστικοί υπολογισμοί των συναρτήσεων - στόχων, με αποτέλεσμα να μη χρησιμοποιείται το λογισμικό αξιολόγησης στις περιπτώσεις που προσεγγιστικά φαίνεται ότι οι λύση που θα προκύψει θα είναι χειρότερη των υπολοίπων.

- Με τη χρήση της μεθόδου SPEA II, ο αλγόριθμος είναι σε θέση να επιλύσει προβλήματα περισσότερων της μίας συνάρτησης - στόχου, δυνατότητα που δεν υπήρχε στην αρχική μορφή του [7][38].
- Στο νέο αλγόριθμο δίνεται η δυνατότητα τήρησης περιορισμών στις συναρτήσεις - στόχους, χωρίς να απορρίπτονται όλες τις λύσεις που υπερβαίνουν τους περιορισμούς, αλλά δίνοντας κατάλληλες ποινές, αντίστοιχες της υπέρβασης.

Στο νέο αλγόριθμο, στις σταθερές του HS δόθηκαν νέα ονόματα, που να ανταποκρίνονται στη διαδικασία που αυτός μιμείται. Παρακάτω φαίνονται οι ονομασίες των αντίστοιχων σταθερών για τους αλγορίθμους HS και CCLO.

Harmony Memory Size	Choreography Set Size
Harmony Memory Consideration Rate	Mimic Movement Rate
Pitch Adjustment Rate	Legato-Staccato Rate

3.2 Ανάλυση δημιουργίας χορογραφίας

Χορογραφία είναι μία αλληλουχία κινήσεων, πάνω σε ένα, συγκεκριμένης χρονικής διάρκειας, μουσικό κομμάτι.

Σύνηθες είναι, ο χορογράφος αρχικά να αναζητήσει έμπνευση για τη χορογραφία που δημιουργεί είτε σε προηγούμενη δουλειά του, είτε σε χορογραφίες που έχουν γίνει από άλλους χορογράφους πάνω στο συγκεκριμένο κομμάτι. Δημιουργείται έτσι μια αρχική λίστα χορογραφιών Choreography Set (CS), που περιλαμβάνει Choreography Set Size (CSS) στοιχεία.

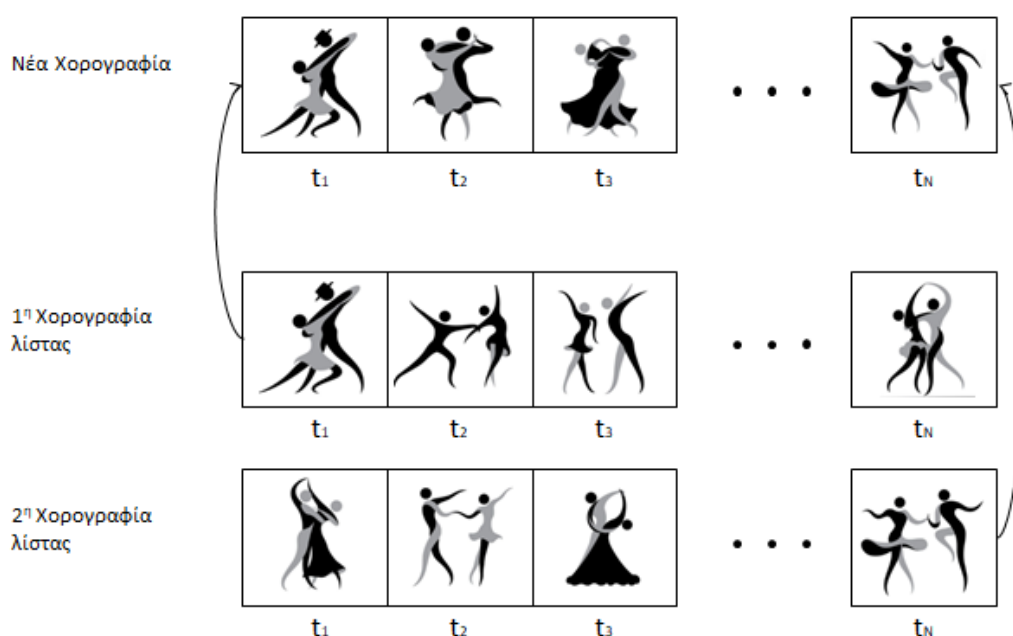
Στη συνέχεια, για κάθε «χρόνο» του κομματιού, δηλαδή κάθε επιμέρους χρονική διάρκεια ικανή να περιλάβει μία κίνηση, ο χορογράφος είτε επιλέγει να χρησιμοποιήσει κινήσεις κάποιας άλλης χορογραφίας, συχνά συνδιάζοντας υπάρχουσες κινήσεις μεταξύ τους, με πιθανότητα χρήσης της λίστας Mimic Movement Rate (MMR), είτε δημιουργεί μια νέα.

Και στην περίπτωση επιλογής μιας υπάρχουσας κίνησης υπάρχει πλήθος τροποποιήσεων που μπορούν να γίνουν. Ο χορογράφος, με πιθανότητα Legato-Staccato Rate (LSR) μπορεί να αλλάξει μεμονωμένα την κίνηση κάποιου σημείου του σώματος ή να προσθέσει legato και staccato, δηλαδή εναλλαγές μεταξύ αργών και γρήγορων βημάτων, ώστε να δημιουργήσει εντάσεις και εκτονώσεις.

Στη συνέχεια, η χορογραφία θα κριθεί με βάση την αισθητική του χορογράφου και, αν είναι καλύτερη από τη χειρότερη στη λίστα χορογραφιών που έχει δημιουργήσει, θα την αντικαταστήσει. Με τον τρόπο αυτό, ο χορογράφος θα συνεχίσει να δημιουργεί χορογραφίες, ανάλογα με το χρόνο που είναι πρόθυμος να διαθέσει για τη διαδικασία αυτή. Στο τέλος, θα εφαρμόσει την καλύτερη χορογραφία που δημιούργησε.

Ένα παράδειγμα όσων αναλύθηκαν παραπάνω φαίνεται στο σχήμα 3.2. Η λίστα περιέχει

$CSS = 2$ χορογραφίες. Κάθε χορογραφία έχει αναλυθεί σε N χρονικά διαστήματα, δηλαδή μεταβλητές σχεδιασμού. Στη νέα χορογραφία, στο πρώτο διάστημα, λόγω τυχαίας πιθανότητας μικρότερης της MMR , επιλέγεται να χρησιμοποιηθεί η κίνηση της πρώτης χορογραφίας. Η κίνηση αυτή, λόγω τυχαίας πιθανότητας μεγαλύτερης της LSR , αντιγράφεται αυτούσια. Στο δεύτερο και στο τρίτο διάστημα, λόγω τυχαίας πιθανότητας μεγαλύτερης της MMR , επιλέγονται νέες κινήσεις, που δεν υπήρχαν στη λίστα. Τέλος, στο νιοστό διάστημα, λόγω τυχαίας πιθανότητας μεγαλύτερης της LSR , επιλέγεται η αντίστοιχη κίνηση της δεύτερης χορογραφίας. Αυτήν τη φορά ωστόσο, λόγω τυχαίας πιθανότητας μικρότερης της LSR , η κίνηση μεταβάλλεται ελαφρώς στη θέση των ελεύθερων άνω άκρων.



Σχήμα 3.2: Μέθοδος δημιουργίας νέας χορογραφίας από υπάρχουσες στη λίστα και νέες κινήσεις.

3.3 Ανάλυση αλγορίθμου CCLO

3.3.1 Αλγόριθμος δημιουργίας νέου ατόμου με την τεχνική CCLO

Ο αλγόριθμος για τη δημιουργία νέου ατόμου, μετά της αλλαγές της ενότητας 3.1, έχει την παρακάτω μορφή.

Για κάθε μεταβλητή x_i , όπου $i = 1, 2, \dots, N$

Εάν $rand(0, 1) \leq MMR$ τότε

$$x_{new,i} = rand_1(0, 1) \times x_i^j + (1 - rand_1(0, 1)) \times x_i^k, \text{ με } j, k \text{ τυχαίες τιμές στο } (1, 2, \dots, CSS).$$

Εάν $rand(0, 1) \leq LSR$ τότε

$$x_{new,i} = x_{new,i} + rand(-bw/2, bw/2) \times e^{-Ev/Ev.Max}$$

Επανάλαβε, όσο το x είναι εκτός ορίων

$$\text{Εάν } x_{new,i} > x_{up,i} \text{ τότε } x_{new,i} = x_{new,i} - (x_{new,i} - x_{up,i}) \times (1 + rand_3(0, 1))$$

$$\text{Εάν } x_{new,i} < x_{low,i} \text{ τότε } x_{new,i} = x_{new,i} + (x_{low,i} - x_{new,i}) \times (1 + rand_4(0, 1))$$

Αλλιώς $x_{new,i} = L$

Επιστροφή νέου ατόμου $X_{new} = x_{new,1}, x_{new,2}, \dots, x_{new,N}$.

Στον παραπάνω αλγόριθμο, με $rand(X, Y)$ συμβολίζεται τυχαία τιμή στο διάστημα $[X, Y]$, με x_{new} το νέο άτομο που δημιουργείται, με Ev οι αξιολογήσεις που έχουν πραγματοποιηθεί, με $Ev.Max$ οι συνολικές αξιολογήσεις που έχει ορίσει ο χρήστης να γίνουν, με x_{up} και x_{low} το ανώτερο και κατώτερο όριο της μεταβλητής αντίστοιχα και με L τυχαία τιμή στο πεδίο ορισμού της μεταβλητής.

3.3.2 CCLO χωρίς χρήση μεταπροτύπων

Τα βήματα εύρεσης της βέλτιστης λύσης, χωρίς χρήση μεταπροτύπων, φαίνονται παρακάτω:

1. Αρχικά δημιουργείται ο πληθυσμός CS, αποτελούμενος από CSS πλήθους άτομα τυχαίων μεταβλητών.
2. Υπολογίζεται η πραγματική τιμή της ή των συναρτήσεων - στόχων τους, με χρήση του λογισμικού αξιολόγησης.
3. Υπολογίζεται η τιμή των συναρτήσεων στόχων μετά την εφαρμογή κατάλληλης ποινής λόγω υπέρβασης των περιορισμών, η οποία είναι ίση με την πραγματική στην περίπτωση που δεν υπάρχουν περιορισμοί, ή που αυτοί δεν έχουν καταπατηθεί.
4. Υπολογίζεται η τιμή της συνάρτησης ϕ , ίση με τη συνάρτηση στην οποία έχει εφαρμοστεί ποινή στην περίπτωση της μιας συνάρτησης - στόχου, και υπολογισμένη με χρήση της μεθόδου SPEAII, στην περίπτωση των περισσοτέρων.

5. Για την περίπτωση των πολλών συναρτήσεων - στόχων εντοπίζεται το μέτωπο των μη-κυριαρχούμενων λύσεων και δημιουργείται ο πληθυσμός των επίλεκτων ατόμων (elite). Για την περίπτωση της μίας μεταβλητής εντοπίζεται η βέλτιστη τιμή και αποθηκεύεται. Τα αποτελέσματα αυτά γράφονται και σε αρχεία εξόδου για διευκόλυνση του χρήστη.
6. Δημιουργούνται νέα άτομα, και δημιουργείται ο πληθυσμός απογόνων (offspring), οι μεταβλητές του οποίου παίρνουν τιμές με χρήση του αλγορίθμου της ενότητας 3.3.1.
7. Για τα άτομα του πληθυσμού offspring υπολογίζεται η πραγματική τιμή της ή των συναρτήσεων - στόχων, η τιμή των συναρτήσεων στόχων μετά την εφαρμογή κατάλληλης ποινής λόγω υπέρβασης περιορισμών και η τιμή της συνάρτησης ϕ , όπως παραπάνω.
8. Από τη σύνθεση των πληθυσμών offspring, CS και elite, υπολογίζονται ο νέος πληθυσμός CS που περιέχει τα καλύτερα CSS πλήθους άτομα που έχουν βρεθεί μέχρι στιγμής, η οποία αντικαθιστά τον προηγούμενο, καθώς και ο νέος πληθυσμός μη-κυριαρχούμενων επίλεκτων ατόμων, για την περίπτωση των πολλών συναρτήσεων - στόχων. Αν ο πληθυσμός επίλεκτων ατόμων έχει περισσότερα στοιχεία από αυτά που έχει ορίσει ο χρήστης, διαγράφονται τα άτομα με τις μεγαλύτερες τιμές χρησιμότητας, για να ικανοποιηθεί ο περιορισμός. Και τα νέα αποτελέσματα γράφονται στα αρχεία εξόδου.
9. Ελέγχεται το κριτήριο τερματισμού και, αν δεν έχει ικανοποιηθεί, επαναλαμβάνεται η διαδικασία από το βήμα 6 έως το βήμα 9.
10. Τέλος τυπώνεται η καλύτερη λύση, στην περίπτωση της μίας συνάρτησης - στόχου ή το μέτωπο Pareto στην περίπτωση των περισσότερων.

3.3.3 CCLO με χρήση μεταπροτύπων

Τα βήματα εύρεσης της βέλτιστης λύσης, με χρήση μεταπροτύπων, φαίνονται παρακάτω:

1. Αρχικά δημιουργούνται ένας πληθυσμός, ατόμων τυχαίων μεταβλητών. Ο χρήστης ορίζει το πλήθος (Ev.Beg.) των ατόμων αυτών. Από τον πληθυσμό αυτό θα χρησιμοποιηθούν άτομα για την εκπαίδευση των μεταπροτύπων.
2. Στη συνέχεια, υπολογίζονται οι πραγματικές τιμές της ή των συναρτήσεων - στόχων του πληθυσμού, και αποθηκεύονται στη βάση δεδομένων DB.
3. Υπολογίζεται η τιμή των συναρτήσεων - στόχων μετά την εφαρμογή ποινής λόγω υπέρβασης των περιορισμών, η οποία είναι ίση με αυτή του βήματος 2 στην περίπτωση που το πρόβλημα δεν υπόκειται σε περιορισμούς, ή οι περιορισμοί δεν έχουν καταπατηθεί. Αποθηκεύεται στην DB.
4. Υπολογίζεται η τιμή της συνάρτησης ϕ για κάθε άτομο, ίση με τη συνάρτηση στην οποία έχει εφαρμοστεί ποινή στην περίπτωση της μίας συνάρτησης -

στόχου, και υπολογισμένη με χρήση της μεθόδου SPEAII, στην περίπτωση των περισσοτέρων. Και αυτές αποθηκεύονται στην DB.

5. Από τα στοιχεία της βάσης δεδομένων DB δημιουργείται ένας νέος πληθυσμός, ο CS, αποτελούμενος από τα CSS καλύτερα άτομα που έχουν βρεθεί. Στο σημείο αυτό αρχίζει η διαδικασία της εξέλιξης.
6. Στη συνέχεια, για την περίπτωση των πολλών συναρτήσεων - στόχων, εντοπίζεται το μέτωπο των μη-κυριαρχούμενων λύσεων, που αποτελεί τον πληθυσμό επίλεκτων ατόμων (elite). Για την περίπτωση της μίας μεταβλητής εντοπίζεται και αποθηκεύεται η βέλτιστη τιμή. Τα αποτελέσματα αυτά γράφονται και σε κατάλληλα αρχεία για διευκόλυνση του χρήστη.
7. Δημιουργούνται νέα άτομα, οι μεταβλητές των οποίων παίρνουν τιμές με χρήση του αλγορίθμου της ενότητας 3.3.1 και δημιουργείται ο πληθυσμός offspring.
8. Για κάθε στοιχείο του πληθυσμού offspring, βρίσκονται τα στοιχεία εκείνα της βάσης DB οι τιμές των μεταβλητών σχεδιασμού των οποίων απέχουν τις ελάχιστες αποστάσεις από αυτό, για εκπαίδευση των τοπικών μεταπροτύπων.
9. Με χρήση των μεταπροτύπων υπολογίζεται μια προσεγγιστική τιμή ϕ' για τα άτομα αυτά.
10. Μόνο για τα άτομα του offspring με τις καλύτερες προσεγγιστικές τιμές ϕ' βρίσκονται η πραγματική τιμή της ή των συναρτήσεων - στόχων μέσω του λογισμικού αξιολόγησης, η τιμή των συναρτήσεων στόχων μετά την εφαρμογή κατάλληλης ποινής λόγω υπέρβασης περιορισμών και η τιμή της συνάρτησης ϕ , όπως παραπάνω.
11. Τα άτομα που υπολογίστηκαν με το λογισμικό αξιολόγησης προστίθενται στη βάση δεδομένων DB για μελλοντική εξάσκηση μεταπροτύπων.
12. Από τη σύνθεση των πληθυσμών offspring για τους υπολογισμούς με το λογισμικό αξιολόγησης, CS και elite, υπολογίζονται ο νέος πληθυσμός CS που περιέχει τα καλύτερα CSS πλήθους άτομα που έχουν βρεθεί μέχρι στιγμής, ο οποίος αντικαθιστά τον προηγούμενο, καθώς και ο νέος πληθυσμός elite, για την περίπτωση των πολλών συναρτήσεων - στόχων. Αν ο πληθυσμός elite έχει περισσότερα στοιχεία από αυτά που έχει ορίσει ο χρήστης, τα λιγότερο χρήσιμα διαγράφονται, για να ικανοποιηθεί ο περιορισμός. Και τα νέα αποτελέσματα γράφονται στα αρχεία εξόδου.
13. Ελέγχεται το κριτήριο τερματισμού και, αν δεν έχει ικανοποιηθεί, επαναλαμβάνεται η διαδικασία από το βήμα 7 έως το βήμα 13.
14. Τέλος τυπώνεται η καλύτερη λύση, στην περίπτωση της μίας συνάρτησης - στόχου και το μέτωπο Pareto στην περίπτωση των περισσότερων.

3.4 Ομοιότητες και διαφορές CCLO - (μ,λ) ΕΑ

Από την παραπάνω ανάλυση, είναι εμφανές ότι ο CCLO και ο (μ,λ) εξελικτικός αλγόριθμος που χρησιμοποιείται στο λογισμικό EASY, ως πληθυσμιακές στοχαστικές μέθοδοι, ακολουθούν όμοια διαδικασία για τη βελτιστοποίηση. Συγκεκριμένα:

- Και στις δύο περιπτώσεις γίνεται τυχαία αρχικοποίηση ενός πληθυσμού, το πλήθος των ατόμων του οποίου ορίζεται από το χρήστη.
- Ο αρχικός πληθυσμός αξιολογείται, ελέγχεται ως προς τους περιορισμούς, θέτοντας κατάλληλες ποινές σε περιπτώσεις υπέρβασης, και υπολογίζεται η τιμή της συνάρτησης καταλληλότητάς των στοιχείων του, με τη μέθοδο SPEAII και στις δύο περιπτώσεις. Για προβλήματα πολλών συναρτήσεων - στόχων, από τις μη κυριαρχούμενες λύσεις δημιουργείται ένας πληθυσμός elite στοιχείων.
- Από τα υπολογισμένα άτομα επιλέγονται ορισμένα και, από συνδυασμό των τιμών τους με παράγοντες τυχαιότητας, προκύπτει η νέα γενιά.
- Για τα άτομα της νέας γενιάς υπολογίζονται προσεγγιστικά οι τιμές των συναρτήσεων - στόχων με χρήση RBFN. Τα άτομα της γενιάς με τις χαμηλότερες τιμές συναρτήσεων - στόχων, αξιολογούνται με το πραγματικό λογισμικό αξιολόγησης, και οι τιμές τους αποθηκεύονται για επόμενη χρήση των RBFN.
- Τα άτομα που έχουν αξιολογηθεί με το λογισμικό αξιολόγησης ελέγχονται ως προς τους περιορισμούς και βρίσκεται η τιμή της συνάρτησης καταλληλότητάς τους.
- Τέλος, ελέγχεται η σύγκλιση και η διαδικασία συνεχίζεται μέχρι να ικανοποιηθεί το κριτήριο τερματισμού, που αφορά αριθμό κλήσεων του λογισμικού αξιολόγησης.

Οι διαφορές των δύο αλγορίθμων είναι οι ακόλουθες:

- Κατά την επιλογή ατόμων από τα οποία θα δημιουργηθεί η νέα γενιά. Στην περίπτωση του CCLO, τα νέα άτομα δημιουργούνται από τα στοιχεία του πληθυσμού CS, που είναι τα CSS πλήθους στοιχεία με τις βέλτιστες τιμές καταλληλότητας που έχουν βρεθεί σε όλη τη διάρκεια της βελτιστοποίησης. Αντίθετα, στον εξελικτικό αλγόριθμο τα νέα άτομα δημιουργούνται από τα άτομα που δημιουργήθηκαν στην προηγούμενη γενιά. Μεγαλύτερη πιθανότητα επιλογής έχουν άτομα με καλύτερες τιμές συνάρτησης καταλληλότητας, αλλά μπορούν να επιλεγούν και άτομα με λιγότερο καλές τιμές.
- Κατά τη διαδικασία της διασταύρωσης. Στην περίπτωση του CCLO, επιλέγεται τιμή για καθεμία από τις μεταβλητές σχεδιασμού του νέου ατόμου τυχαία ως συνδυασμός δύο από τις υπάρχουσες τιμές των αντίστοιχων μεταβλητών της CS, ή επιλέγεται μια εντελώς νέα, τυχαία μεταβλητή. Στον εξελικτικό αλγόριθμο το νέο άτομο προκύπτει αποκλειστικά από διασταύρωση, με τις μεταβλητές των

νέων στοιχείων να λαμβάνονται ως συνδυασμός των αντίστοιχων μεταβλητών ενός συγκεκριμένου, μικρού υποσυνόλου των γονέων, συχνά μεγέθους τριών ατόμων.

- Κατά τη διαδικασία της μετάλλαξης. Και στις δύο περιπτώσεις κατά τη μετάλλαξη, με μικρή πιθανότητα η τιμή ορισμένων μεταβλητών σχεδιασμού έχει μια μικρή αύξηση ή μείωση, που εξαρτάται από τον αριθμό των αξιολογήσεων που έχουν ολοκληρωθεί. Στην περίπτωση του CCLO η μικρή αυτή μεταβολή εξαρτάται επιπλέον από μια σταθερά (hw) που θέτει ο χρήστης. Στην περίπτωση των εξελικτικών αλγορίθμων, η τιμή αυτή είναι μικρότερη, και εξαρτάται επιπλέον από την απόσταση της μεταβλητής σχεδιασμού από τα επιτρεπτά όριά της.
- Στην περίπτωση υπέρβασης των περιορισμών. Στον CCLO, στην περίπτωση που η μεταβλητή σχεδιασμού ξεπεράσει τα όρια που έχουν τεθεί, γίνεται αναδίπλωση. Στον EA, η μεταβλητή αυτή λαμβάνει την τιμή του ορίου που ξεπεράστηκε.
- Ένα επιπλέον χαρακτηριστικό του EA είναι ότι, για να μπορούν να χρησιμοποιηθούν ως γονείς τα καλύτερα άτομα που έχουν βρεθεί σε όλη τη διάρκεια της βελτιστοποίησης, ώστε να γίνεται αναζήτηση στην περιοχή τους, λίγα άτομα των λ απογόνων σε κάθε γενιά αντικαθίστανται από τα άτομα αυτά.

Κεφάλαιο 4

Μετάβαση από HS σε CCLO

Στην ενότητα αυτή παρουσιάζονται μια προς μια οι μεταβολές που έγιναν στον αλγόριθμο HS, και τα αποτελέσματα που αυτές έφεραν στην πορεία σύγκλισης. Δεν παρουσιάζονται οι μεταβολές που έγιναν για να προστεθεί η δυνατότητα διαχείρισης μετώπων Pareto και περιορισμών με την μέθοδο των ποινών. Χρησιμοποιούνται δύο συναρτήσεις για την πιστοποίηση της βελτίωσης της πορείας σύγκλισης, η συνάρτηση Rastrigin και η συνάρτηση Levy No13.

4.1 Εξεταζόμενες συναρτήσεις

Συνάρτηση Rastrigin

Η μαθηματική έκφραση της συνάρτησης Rastrigin είναι [67]

$$f(x) = 3 \times 10 - \sum_{n=1}^{n=3} (x_i^2 - 10 \cos(2\pi x_i)) \quad (4.1)$$

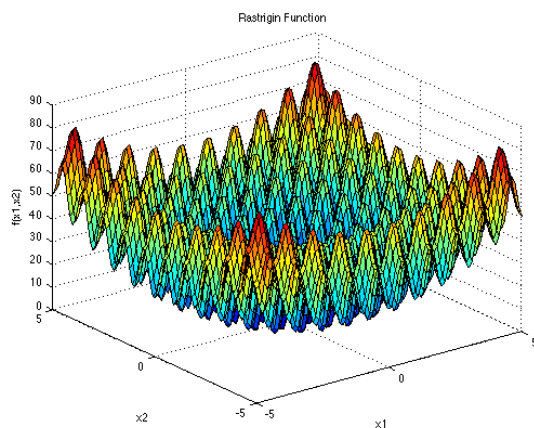
Οι μεταβλητές σχεδιασμού που επιλέχθηκαν να χρησιμοποιηθούν είναι 3, με όρια

$$-5.12 \leq x_1, x_2, x_3 \leq 5.12 \quad (4.2)$$

Η συνάρτηση παρουσιάζει ελάχιστο στο 0, για $x_1 = x_2 = x_3 = 0$.

Στο σχήμα 4.1 φαίνεται η συνάρτηση Rastrigin στον z άξονα για το αντίστοιχο πρόβλημα δύο μεταβλητών σχεδιασμού, στους άξονες x και y. Ήδη με τις 2 μεταβλητές είναι φανερό ότι η συνάρτηση έχει μεγάλο πλήθος τοπικών ακρότατων, επομένως μεγαλύτε-

ρο βάρος για την εύρεση βέλτιστων αποτελεσμάτων έχει η ικανότητα ανίχνευσης του αλγορίθμου.



Σχήμα 4.1: Συνάρτηση Rastrigin [68].

Συνάρτηση Levy No.13

Η μαθηματική έκφραση της συνάρτησης Levy No.13 είναι [69]

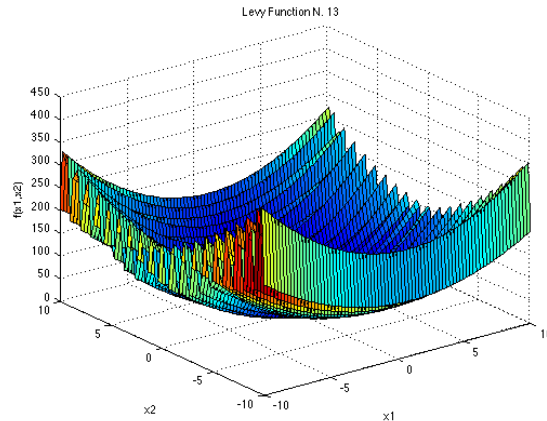
$$f(x) = \sin^2 3\pi x_1 + (x_1 - 1)^2(1 + \sin^2 3\pi x_2) + (x_2 - 1)^2(1 + \sin^2 2\pi x_2) + (x_2 - 1)^2(1 + \sin^2 2\pi x_2) \quad (4.3)$$

Οι μεταβλητές σχεδιασμού της συνάρτησης είναι 2, με όρια

$$-10 \leq x_1, x_2 \leq 10 \quad (4.4)$$

Η συνάρτηση παρουσιάζει ελάχιστο στο 0, για $x_1 = x_2 = 1$.

Στο σχήμα 4.2 φαίνεται η συνάρτηση Levy No.13. Στην περίπτωση αυτή, στο κέντρο υπάρχει μια περιοχή καλών λύσεων, στην οποία χρειάζεται επαρκή ανίχνευση από το πρόγραμμα βελτιστοποίησης για την εύρεση της βέλτιστης λύσης.



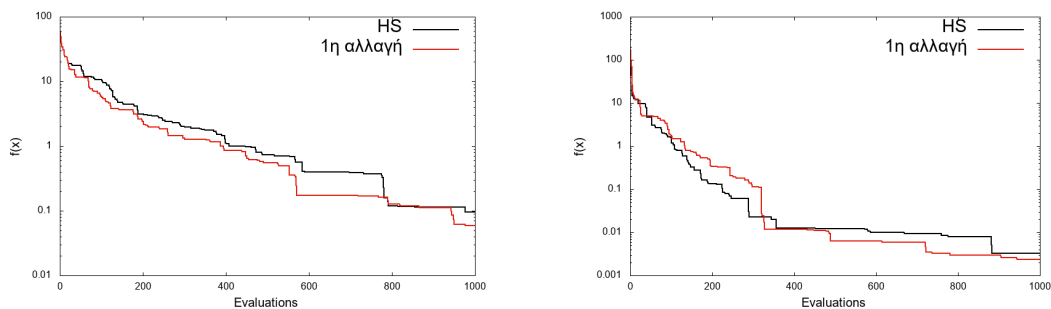
Σχήμα 4.2: Συνάρτηση Levy No13 [68].

4.2 Αποτελέσματα αλλαγών

Στην ενότητα αυτή, παρουσιάζεται σταδιακά η μεταβολή στην πορεία σύγκλισης του αλγορίθμου HS καθώς σε αυτόν προστίθενται ή μεταβάλλονται στοιχεία.

4.2.1 1^η Αλλαγή: Πλήθος νέων υποψήφιων λύσεων

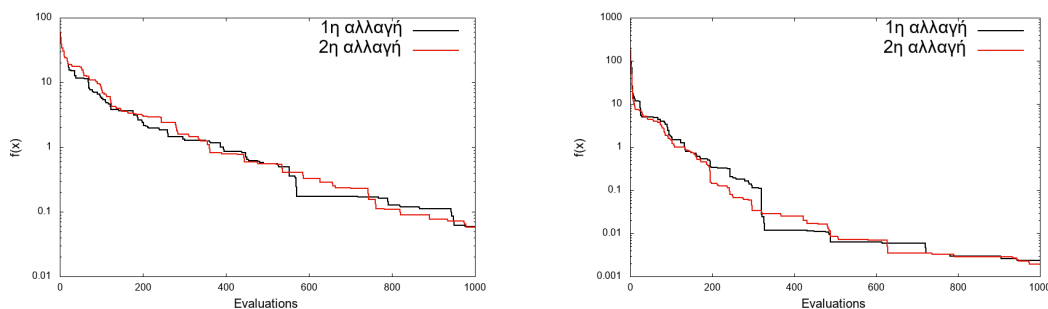
Στον αλγόριθμο HS δημιουργείται μία νέα υποψήφια λύση σε κάθε επανάληψη. Στον CCLO, χρησιμοποιούνται περισσότερες της μίας. Η τιμή που επιλέχθηκε για το παράδειγμα αυτό είναι 25. Τα αποτελέσματα παρουσιάζονται στο σχήμα 4.3 και στους πίνακες 4.1 και 4.2.



Σχήμα 4.3: Πορεία σύγκλισης συναρτήσεων Rastrigin (αριστερά) και Levy No13 (δεξιά) με χρήση ενός ατόμου σε κάθε επανάληψη (μαύρη γραμμή), και με χρήση 25 ατόμων (κόκκινη γραμμή). Μέση τιμή πέντε κλήσεων του κώδικα.

4.2.2 2^η Αλλαγή: Διαχείριση υπέρβασης ορίων μεταβλητών

Στον HS, όταν οι μεταβλητές σχεδιασμού υπερβαίνουν τα όριά τους, παίρνουν την τιμή του ορίου που ξεπεράστηκε. Στο νέο αλγόριθμο, σε περίπτωση υπέρβασης του ορίου, η μεταβλητή αναδιπλώνεται, με τη μέθοδο που παρουσιάστηκε στην ενότητα 3.3.1. Στο σχήμα 4.4 συγκρίνεται ο υπό δημιουργία αλγόριθμος, με την παλιά και τη νέα μέθοδο διαχείρισης των μεταβλητών που ξεπερνούν τα όρια.



Σχήμα 4.4: Πορεία σύγκλισης συναρτήσεων *Rastrigin* (αριστερά) και *Levy No13* (δεξιά) με τις τιμές που ξεπερνάνε τα όρια να παίρνουν την τιμή τους (μαύρη γραμμή), και αναδίπλωση (κόκκινη γραμμή). Μέση τιμή πέντε κλήσεων του κώδικα.

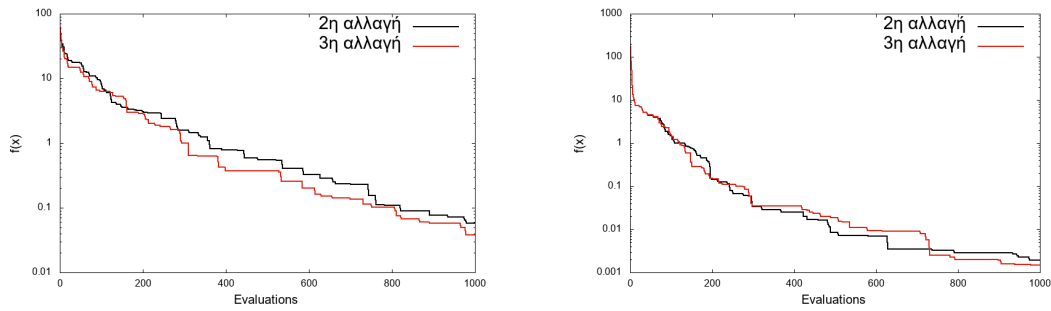
Να σημειωθεί ότι στις συναρτήσεις αυτές ήταν ελάχιστες οι μεταβλητές που ξεπέρασαν τα όρια, γιατί και η διαφορά δεν είναι σημαντική.

4.2.3 3^η Αλλαγή: Εύρος BW

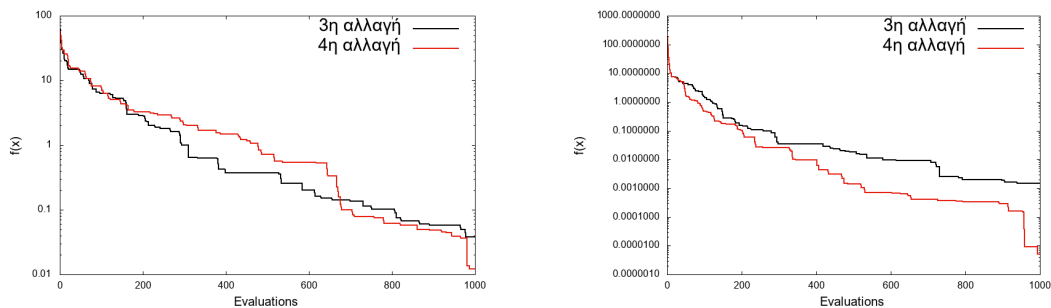
Στον HS, το εύρος μέσα στο οποίο μπορεί να μεταβληθεί η νέα τιμή που δημιουργήθηκε είναι σταθερό. Στο νέο αλγόριθμο το εύρος μειώνεται με χρήση της συνάρτησης $bw'(Ev) = bw \times e^{-Ev/Ev.Max}$. Στο σχήμα 4.5 συγκρίνεται ο αλγόριθμος που δημιουργείται με χρήση σταθερού και μειούμενου bw.

4.2.4 4^η Αλλαγή: Διασταύρωση

Στον αλγόριθμο HS, εάν μια τυχαία πιθανότητα είναι μικρότερη από HMCR, η νέα μεταβλητή λαμβάνει την τιμή μιας υπάρχουσας μεταβλητής στην HM. Στο νέο αλγόριθμο, η νέα μεταβλητή προκύπτει ως γραμμικός συνδυασμός δύο μεταβλητών της HM, δηλαδή χρησιμοποιείται σχήμα αντίστοιχο της διασταύρωσης των EA. Ο αλγόριθμος πριν και μετά από την αλλαγή αυτή παρουσιάζεται στο σχήμα 4.6.



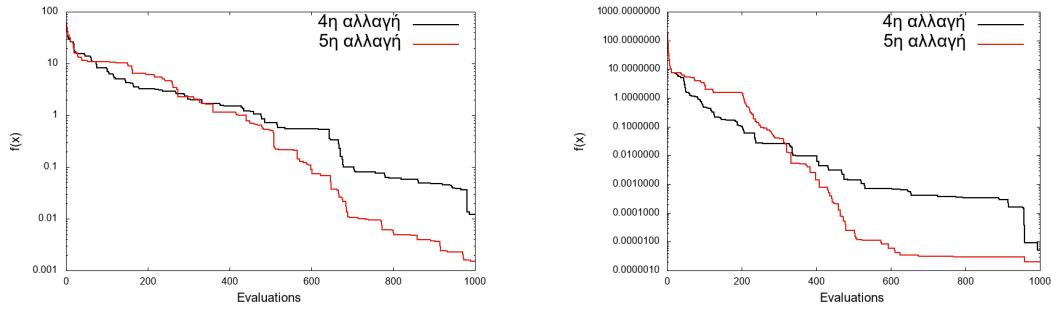
Σχήμα 4.5: Πορεία σύγκλισης συναρτήσεων *Rastrigin* (αριστερά) και *Levy No13* (δεξιά) με σταθερό bw (μαύρη γραμμή), και μειούμενο (κόκκινη γραμμή). Μέση τιμή πέντε κλήσεων του κώδικα.



Σχήμα 4.6: Πορεία σύγκλισης συναρτήσεων *Rastrigin* (αριστερά) και *Levy No13* (δεξιά) χωρίς διασταύρωση (μαύρη γραμμή), και με διασταύρωση (κόκκινη γραμμή). Μέση τιμή πέντε κλήσεων του κώδικα.

4.2.5 5^η Αλλαγή: Μεταπρότυπα

Τέλος, στον αλγόριθμο χρησιμοποιείται τεχνητή νοημοσύνη, για παράλειψη αξιολογήσεων στοιχείων με το ακριβές λογισμικό αξιολόγησης, τα οποία προσεγγιστικά φαίνεται ότι δεν θα έχουν ικανοποιητικές τιμές συναρτήσεων - στόχων. Η μεταβολή στην πορεία σύγκλισης από την προσθήκη αυτή φαίνεται στο σχήμα 4.7. Καλύτερα αποτελέσματα από το σχήμα φαίνεται ότι παρουσιάζονται όσο αυξάνεται ο αριθμός των πραγματικών αξιολογήσεων που έχουν ολοκληρωθεί. Αυτό συμβαίνει διότι αυξάνεται το μέγεθος της βάσης δεδομένων από την οποία επιλέγονται άτομα για τις προβλέψεις, με αποτέλεσμα οι τελευταίες να γίνονται πιο ακριβείς.



Σχήμα 4.7: Πορεία σύγκλισης συναρτήσεων *Rastrigin* (αριστερά) και *Levy No13* (δεξιά) χωρίς χρήση μεταπροτύπων (κόκκινη γραμμή), και με χρήση τους (μαύρη γραμμή). Μέση τιμή πέντε κλήσεων του κώδικα.

4.2.6 Συγκεντρωτικά αποτελέσματα

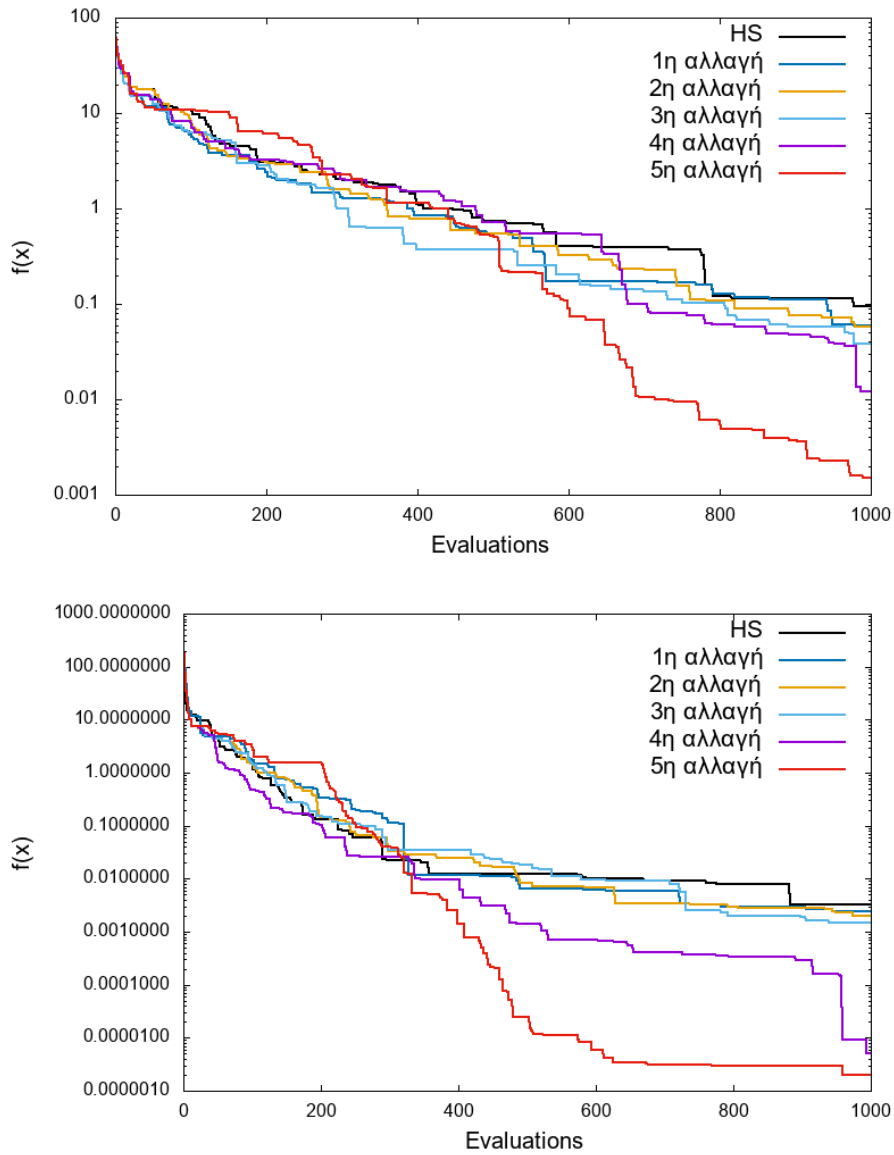
Τα αποτελέσματα που έδωσε καθεμία από της πέντε κλήσεις του κώδικα που έγιναν για την κάθε μεταβολή, μετά το τέλος των χιλίων αξιολογήσεων, παρουσιάζονται στους πίνακες 4.1 και 4.2. Συγκεντρωτικά οι μεταβολές στην πορεία λόγω των αλλαγών παρουσιάζονται στο διάγραμμα 4.8.

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS ($\times 10^{-2}$)	9.592	15.095	3.836	3.360	2.711	1.503	36.550
1 ^η Αλλαγή ($\times 10^{-2}$)	5.881	5.453	4.918	15.113	3.400	0.765	5.210
2 ^η Αλλαγή ($\times 10^{-2}$)	5.755	4.974	10.971	5.337	1.218	10.671	0.577
3 ^η Αλλαγή ($\times 10^{-2}$)	3.796	6.706	1.596	1.223	15.742	0.139	0.279
4 ^η Αλλαγή ($\times 10^{-2}$)	1.228	1.269	1.416	0.003	0.446	3.286	0.989
5 ^η Αλλαγή ($\times 10^{-2}$)	0.151	0.279	0.008	0.002	0.016	0.647	0.081

Πίνακας 4.1: Μέση τιμή, τυπική απόκλιση και βέλτιστο κάθε κλήσης του κώδικα, για τη συνάρτηση *Rastrigin*.

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS ($\times 10^{-4}$)	32.938	33.061	35.356	6.885	87.612	27.820	7.016
1 ^η Αλλαγή ($\times 10^{-4}$)	23.856	28.637	2.670	64.395	43.868	1.629	6.720
2 ^η Αλλαγή ($\times 10^{-4}$)	19.526	26.534	2.670	64.395	22.862	0.985	6.720
3 ^η Αλλαγή ($\times 10^{-4}$)	14.835	17.396	9.687	1.708	16.640	43.959	2.182
4 ^η Αλλαγή ($\times 10^{-4}$)	0.051	0.047	0.118	0.049	0.010	0.005	0.074
5 ^η Αλλαγή ($\times 10^{-4}$)	0.020	0.044	0.001	0.000	0.000	0.001	0.098

Πίνακας 4.2: Μέση τιμή, τυπική απόκλιση και βέλτιστο κάθε κλήσης του κώδικα, για τη συνάρτηση *Levy No13*.



Σχήμα 4.8: Πορεία σύγκλισης συναρτήσεων *Rastrigin* (πάνω) και *Levy No13* (κάτω) καθώς γίνονται μεταβολές στη δομή του αλγορίθμου. Μέση τιμή πέντε κλήσεων του κώδικα.

Οι σταθερές που χρησιμοποιήθηκαν για την αναζήτηση αυτή, φαίνονται στον πίνακα 4.3.

Constants	Rastrigin	Levy No13
CSS	15	20
MMR	0.8	0.85
LSR	0.5	0.6
bw	2	3
Off	25	25
Ev.Beg.	150	200
Pool	9	6

Πίνακας 4.3: Χρησιμοποιούμενες σταθερές στον *CCL0*.

Κεφάλαιο 5

Πιστοποίηση κώδικα σε μαθηματικές συναρτήσεις

Στο κεφάλαιο αυτό, δοκιμάζεται ο κώδικας που δημιουργήθηκε στο πλαίσιο της διπλωματικής αυτής εργασίας σε μαθηματικές συναρτήσεις μονοκριτηριακής και πολυκριτηριακής βελτιστοποίησης για πέντε κλήσεις κάθε λογισμικού σε κάθε πρόβλημα. Συγκρίνονται οι κώδικες CCLO και HS. Σε μερικές περιπτώσεις έχει γίνει σύγκριση και με τα αντίστοιχα αποτελέσματα του λογισμικού EASY με μεταπρότυπα. Η σύγκριση με το λογισμικό EASY δεν αποσκοπεί σε συμπεράσματα, μιας και ο EASY δεν έχει ενδεχομένως τρέξει με τις καλύτερες παραμέτρους που ένας έμπειρος χρήστης θα μπορούσε να θέσει. Σκοπός του κεφαλαίου αυτού είναι να συγκριθούν κυρίως οι κώδικες CCLO και HS, τόσο ως προς τις τελικές τιμές που επιστρέφουν στον χρήστη, όσο και προς τη πορεία σύγκλισης τους.

5.1 Συναρτήσεις Μονοκριτηριακής Βελτιστοποίησης

5.1.1 Συνάρτηση Rosenbrock

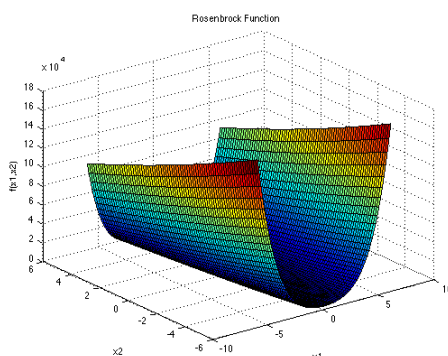
Η μαθηματική έκφραση της συνάρτησης Rosenbrock είναι [67]

$$f(x) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2 \quad (5.1)$$

Οι μεταβλητές σχεδιασμού που χρησιμοποιήθηκαν είναι 2, με όρια

$$-10 \leq x_1, x_2 \leq 10 \quad (5.2)$$

Στο σχήμα 5.1 φαίνεται η συνάρτηση Rosenbrock στον z άξονα για δύο μεταβλητές σχεδιασμού, στους άξονες x και y.



Σχήμα 5.1: Συνάρτηση Rosenbrock [68].

Η συνάρτηση παρουσιάζει ελάχιστο στο 0, για $x_1 = x_2 = 1$.

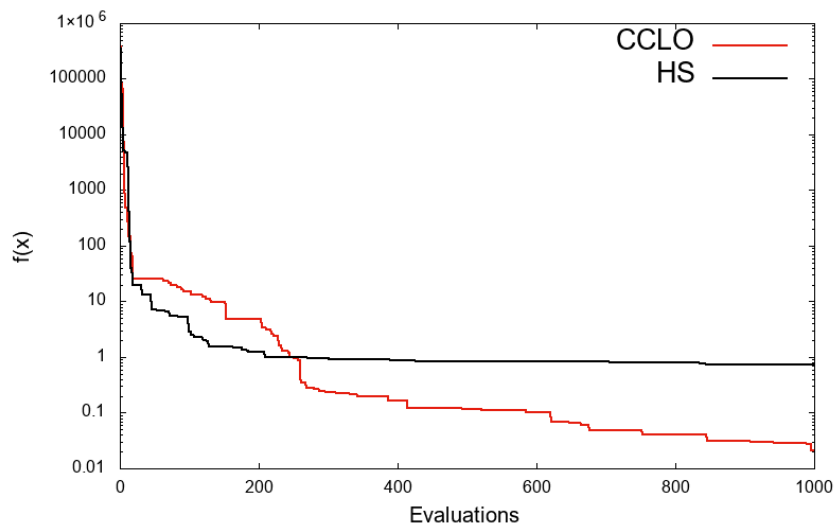
Στον πίνακα 5.1 φαίνονται η μέση τιμή και η τυπική απόκλιση των κωδίκων CCLO και HS, μετά από χίλιες αξιολογήσεις. Στο σχήμα 5.2 φαίνεται η πορεία της μέσης τιμής κατά τη διάρκεια εκτέλεσης του κάθε κώδικα. Τα σχήματα 5.3 παρουσιάζουν αναλυτικά τη μεταβολή της μέσης τιμής και της τυπικής απόκλισης καθώς αυξάνονται οι αξιολογήσεις σε κάθε κώδικα. Οι σταθερές που χρησιμοποιήθηκαν στους HS και CCLO φαίνονται στον πίνακα 5.2.

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS ($\times 10^{-1}$)	7.264	14.403	0.897	33.024	0.995	0.330	1.073
CCLO ($\times 10^{-1}$)	0.210	0.175	0.206	0.345	0.001	0.078	0.419

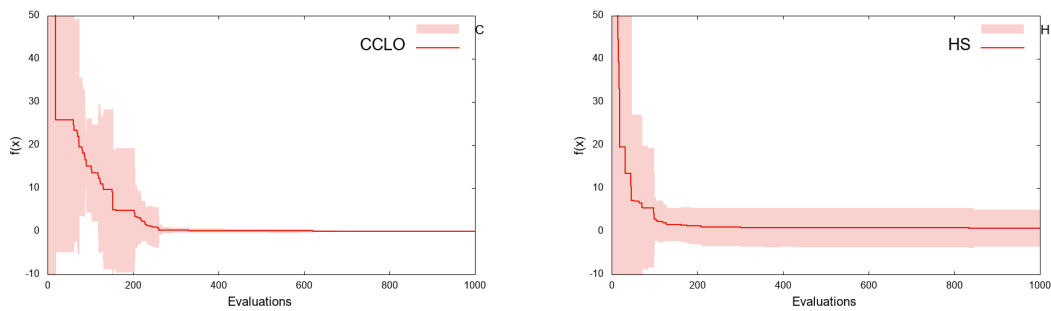
Πίνακας 5.1: Μέση τιμή, τυπική απόκλιση και βέλτιστο κάθε κλήσης CCLO και HS.

Σχολιασμός αποτελεσμάτων

Μετά από την ολοκλήρωση των 1000 αξιολογήσεων, οι τιμές που έχουν βρεθεί με τον κώδικα CCLO με χρήση μεταπροτύπων είναι σημαντικά μικρότερες από αυτές του HS. Στους πρώτους 250 υπολογισμούς της συνάρτησης - στόχου, για τους 200 από τους οποίους δημιουργείται η βάση δεδομένων για την εκπαίδευση των μεταπροτύπων στον CCLO, ο κώδικας HS παρουσιάζει καλύτερα αποτελέσματα. Στη συνέχεια ωστόσο, και μέχρι το τέλος των χιλίων αξιολογήσεων, ο CCLO έχει μικρότερες τιμές. Ο HS έχει καλές τιμές στις πρώτες αξιολογήσεις, σύντομα όμως εγκλωβίζεται σε υποβέλτιστες



Σχήμα 5.2: Πορεία σύγκλισης της συνάρτησης *Rosenbrock* για *CCLO* (κόκκινη γραμμή) και *HS* (μαύρη γραμμή) για πέντε κλήσεις κάθε κώδικα, σε λογαριθμική κλίμακα.



Σχήμα 5.3: Μέση τιμή και τυπική απόκλιση για κάθε κώδικα.

λύσεις, με αποτέλεσμα η τιμή του, μετά τις πρώτες 200 αξιολογήσεις, παραμένει πρακτικά σταθερή. Η τυπική απόκλιση του *HS* είναι σημαντικά μεγαλύτερη αυτής του *CCLO*, επομένως ήταν μεγάλη η διακύμανση της πορείας σύγκλισης για τις πέντε κλήσεις που έγιναν.

Constants	CCLO	HS
CSS	20	20
MMR	0.9	0.9
LSR	0.5	0.5
bw	2.5	2
Off	25	1
Ev.Beg.	200	-
Pool	6	-

Πίνακας 5.2: Χρησιμοποιούμενες σταθερές στον CCLO και τον HS (βλ. Α'). CSS, MMR, LSR και bw οι σταθερές του CCLO, Off ο πληθυσμός κάθε νέας αναζήτησης λύσεων, Ev.Beg. οι αξιολογήσεις που έγιναν πριν ξεκινήσει η διαδικασία της βελτιστοποίησης για εκπαίδευση των μεταπροτύπων, και Pool το πλήθος των γειτονικών ατόμων από τα οποία εκπαιδεύεται το μεταπρότυπο.

5.1.2 Συνάρτηση Sphere

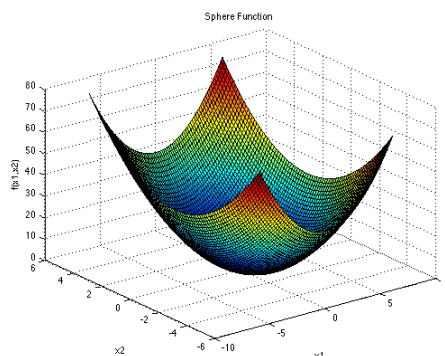
Η μαθηματική έκφραση της συνάρτησης Sphere είναι [69]

$$f(x) = \sum_{n=1}^{n=4} x^2 \quad (5.3)$$

Επιλέγεται να χρησιμοποιηθούν 4 μεταβλητές σχεδιασμού, με όρια

$$-10 \leq x_1, x_2, x_3, x_4 \leq 10 \quad (5.4)$$

Στο σχήμα 5.4 φαίνεται η συνάρτηση Sphere στον z άξονα για το αντίστοιχο πρόβλημα δύο μεταβλητών σχεδιασμού, στους άξονες x και y.



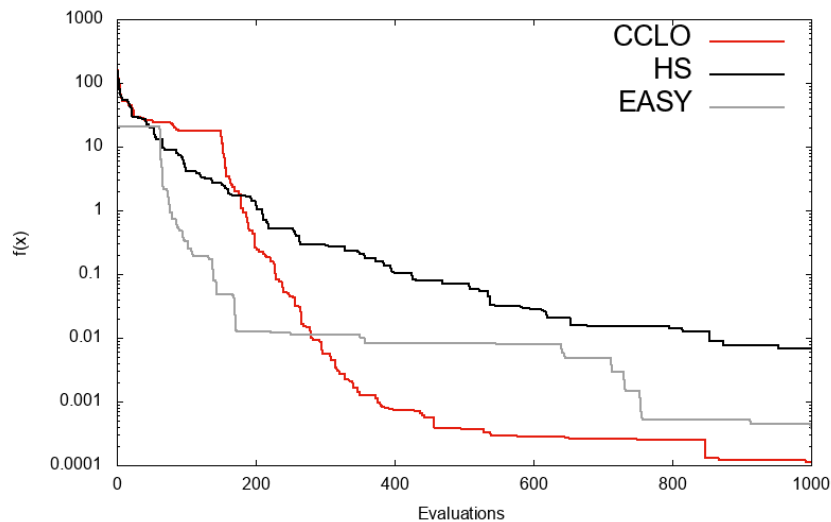
Σχήμα 5.4: Συνάρτηση Sphere [68].

Η συνάρτηση παρουσιάζει ελάχιστο στο 0, για $x_1 = x_2 = x_3 = x_4 = 0$.

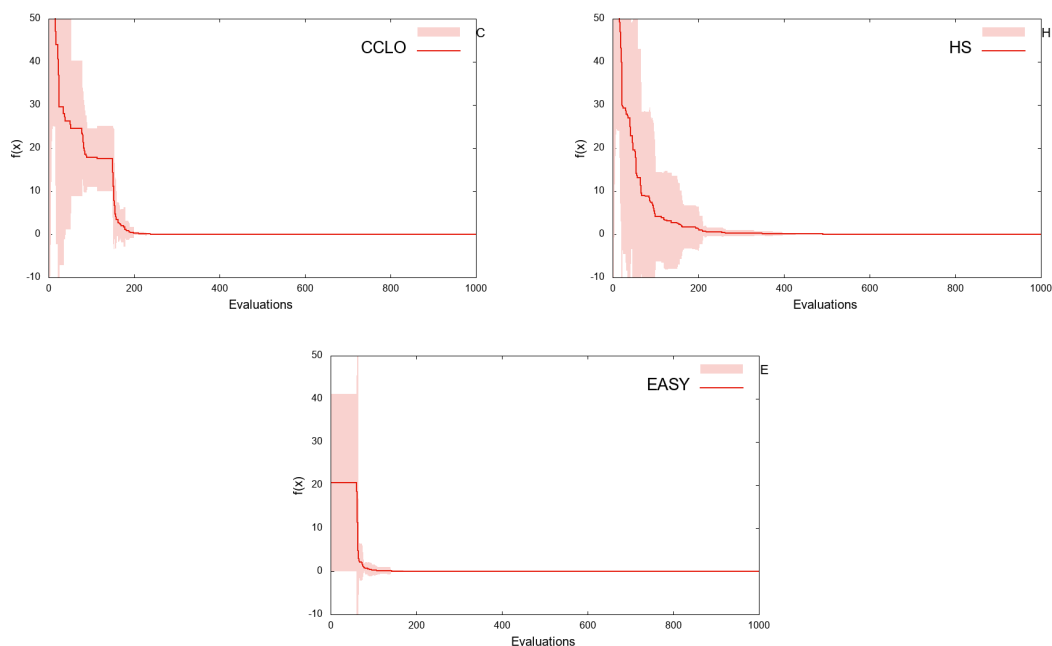
Στον πίνακα 5.3 φαίνονται η μέση τιμή και η τυπική απόκλιση των κωδίκων CCLO, HS αλλά και του λογισμικού EASY, αφού ολοκληρώθηκαν οι 1000 αξιολογήσεις, οι οποίες αποτελούν «αυθαίρετο» μεν, δίκαιο δε, κριτήριο τερματισμού. Στο σχήμα 5.5 συγκρίνεται η πορεία της μέσης τιμής για τα τρία λογισμικά. Στο σχήμα 5.6 παρουσιάζεται ξεχωριστά η μέση τιμή και η τυπική απόκλιση από κάθε λογισμικό κατά την εκτέλεση του κώδικα. Οι σταθερές που χρησιμοποιήθηκαν στους HS και CCLO φαίνονται στον πίνακα 5.4. Για το λογισμικό EASY επιλέχθηκε $(\mu, \lambda) = (20, 60)$, χωρίς κατ'ανάγκη να αποτελεί τη βέλτιστη επιλογή για το πρόβλημα.

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS ($\times 10^{-3}$)	6.764	5.277	5.256	0.969	7.726	4.714	15.154
CCLO ($\times 10^{-3}$)	0.112	0.100	0.002	0.174	0.232	0.016	0.134
EASY ($\times 10^{-3}$)	0.376	0.308	0.874	0.462	0.237	0.213	0.096

Πίνακας 5.3: Μέση τιμή, τυπική απόκλιση και βέλτιστο κάθε κλήσης CCLO, HS και EASY.



Σχήμα 5.5: Πορεία σύγκλισης μέσης τιμής της συνάρτησης *Sphere* για CCLO (κόκκινη γραμμή), HS (μαύρη γραμμή) και EASY (γκρι γραμμή) για πέντε κλήσεις κάθε λογισμικού, σε λογαριθμική κλίμακα. Το λογισμικό EASY ενδεχομένως δεν έτρεξε με τις καλύτερες παραμέτρους που θα μπορούσαν να επιλεγούν.



Σχήμα 5.6: Μέση τιμή και τυπική απόκλιση για κάθε κώδικα.

Constants	CCLO	HS
CSS	15	15
MMR	0.9	0.9
LSR	0.4	0.4
bw	3	3
Off	25	1
Ev.Beg.	150	-
Pool	10	-

Πίνακας 5.4: Χρησιμοποιούμενες σταθερές στον CCLO και τον HS.

Σχολιασμός αποτελεσμάτων

Μετά από τη λήξη των 1000 αξιολογήσεων, οι τιμές που έχουν βρεθεί με τον κώδικα CCLO είναι ελαφρώς καλύτερες από τις αντίστοιχες του EASY. Ο HS έδωσε αποτελέσματα μιας τάξης μεγέθους χειρότερα από τα άλλα δύο λογισμικά. Το λογισμικό EASY έχει καλύτερα αποτελέσματα για μικρό πλήθος αξιολογήσεων, ωστόσο, μετά τις 300 πρώτες ο CCLO δίνει καλύτερα αποτελέσματα. Ο HS έχει καλύτερα αποτελέσματα από τον CCLO για τις πρώτες 200 αξιολογήσεις, οι 150 από τις οποίες χρησιμοποιήθηκαν για δημιουργία της βάσης δεδομένων για εκπαίδευση των μεταπροτύπων στον CCLO. Και πάλι, η τυπική απόκλιση του λογισμικού HS είναι σημαντικά μεγαλύτερη, ωστόσο καμία από τις τιμές των πέντε κλήσεων του δεν είναι καλύτερη από τη μεγαλύτερη του CCLO.

5.2 Συναρτήσεις Πολυκριτηριακής Βελτιστοποίησης

Στην ενότητα αυτή, στόχος είναι η σύγκριση των λογισμικών σε συναρτήσεις πολυκριτηριακής βελτιστοποίησης. Ωστόσο, επειδή το λογισμικό HS δεν διαθέτει μέθοδο διαχείρισης πολλών στόχων στην αρχική του μορφή, για να είναι δυνατή η σύγκριση, και σε αυτό έχει προστεθεί η συνάρτηση SPEAII.

Στην ενότητα αυτή χρησιμοποιείται ο δείκτης υπερόγκου (HVI), που περιγράφεται στο παράρτημα Β', και παρουσιάζονται τα μέτωπα των μη-κυριαρχούμενων λύσεων για κάθε λογισμικό, τα οποία έχουν προκύψει από σύνθεση των ατόμων των μετώπων Pareto από τις πέντε κλήσεις που έγιναν σε κάθε λογισμικό.

5.2.1 Συνάρτηση Kursawe

Η μαθηματική έκφραση της συνάρτησης Kursawe είναι [70]

$$\begin{aligned} f_1(x) &= \sum_{n=1}^{n=2} [-10 \times e^{-0.2\sqrt{x_i^2 - x_{i+1}^2}}] \\ f_2(x) &= \sum_{n=1}^{n=3} [|x_i|^{0.8} + 5 \sin(x_i^3)] \end{aligned} \quad (5.5)$$

Οι μεταβλητές σχεδιασμού είναι 3, με όρια

$$-5 \leq x_1, x_2, x_3 \leq 5 \quad (5.6)$$

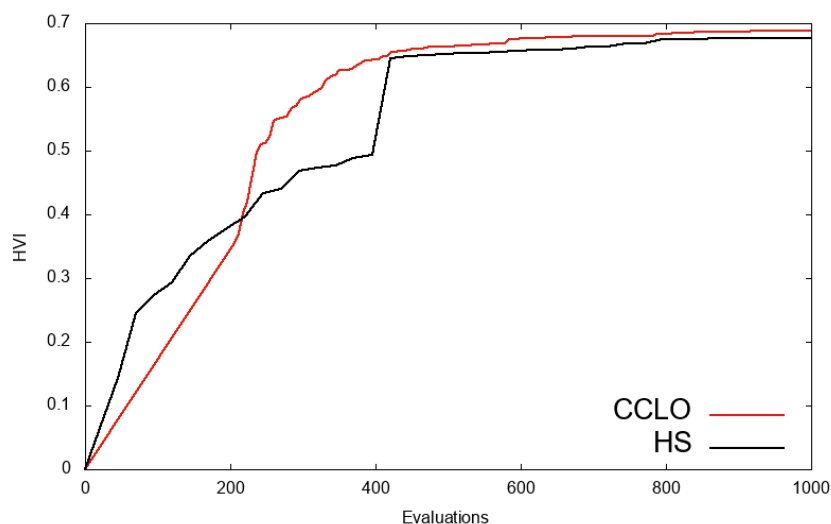
Στον πίνακα 5.5 φαίνεται ο δείκτης υπερόγκου για τα μέτωπα Pareto των κωδίκων CCLO και HS. Στη συνέχεια, στο σχήμα 5.7 φαίνεται η πορεία του δείκτη υπερόγκου συναρτήσει του αριθμού αξιολογήσεων, για τα διαδοχικά μέτωπα μη-κυριαρχούμενων λύσεων των κωδίκων αυτών. Τα σημεία ναδίρ και ζενίθ που χρησιμοποιήθηκαν ήταν τα (-9,2) και (-20,-12) αντίστοιχα.

Στο σχήμα 5.8 συγκρίνονται τα μέτωπα των μη-κυριαρχούμενων σημείων που προέκυψαν από σύνθεση των μετώπων Pareto των πέντε κλήσεων του κάθε κώδικα, για τους κώδικες HS και CCLO. Στο σχήμα 5.9 φαίνονται αναλυτικά τα μέτωπα Pareto των πέντε κλήσεων του κώδικα CCLO.

Στο σχήμα 5.10 φαίνεται το διάγραμμα παράλληλων συντεταγμένων του μετώπου των

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS	0.677	0.030	0.688	0.690	0.695	0.623	0.690
CCLO	0.688	0.007	0.692	0.697	0.691	0.679	0.682

Πίνακας 5.5: Μέση τιμή και τυπική απόκλιση δείκτη υπερόγκου, και αποτέλεσμα κάθε κλήσης κώδικα, για HS και CCLO.



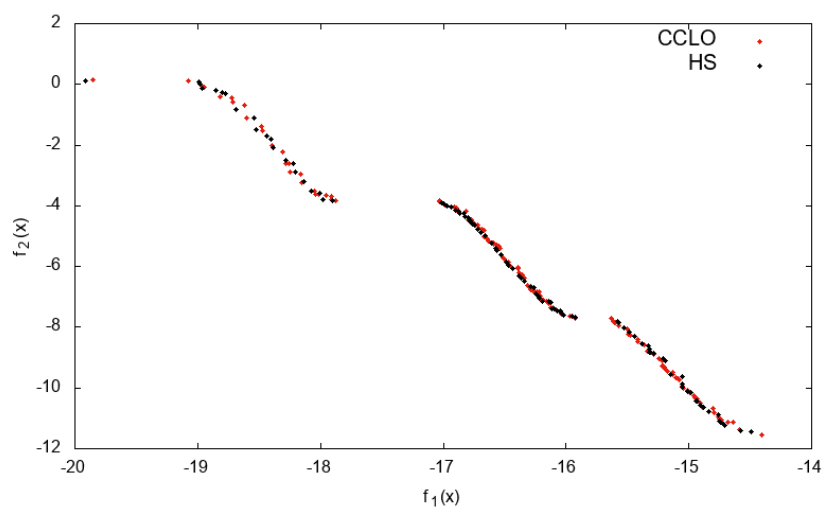
Σχήμα 5.7: Δείκτης υπερόγκου για διαδοχικά μέτωπα μη-κυριαρχούμενων λύσεων του κώδικα CCLO και HS, κόκκινη και μαύρη γραμμή αντίστοιχα.

μη-κυριαρχούμενων ατόμων για τον κώδικα CCLO.

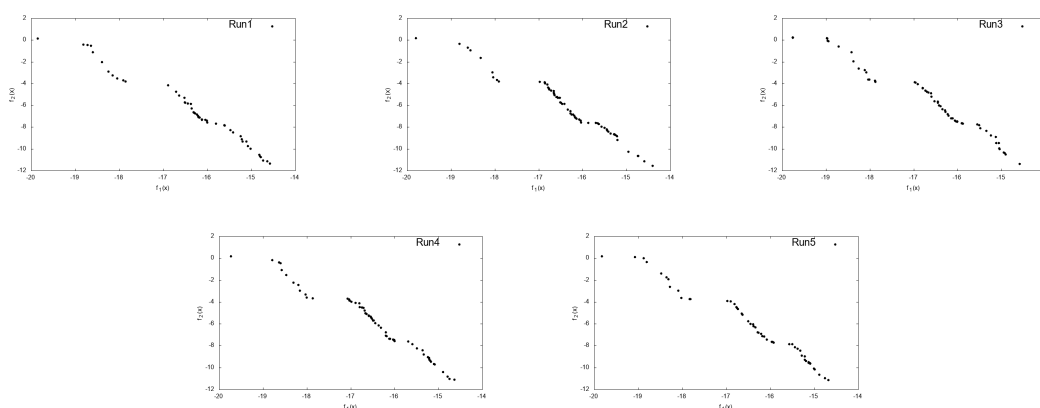
Οι σταθερές που χρησιμοποιήθηκαν για την εξαγωγή των αποτελεσμάτων αυτών για τους CCLO και HS, φαίνονται στον πίνακα 5.6.

Constants	CCLO	HS
CSS	20	20
MMR	0.8	0.9
LSR	0.6	0.5
bw	3	2
Off	25	1
Ev.Beg.	200	-
Pool	9	-

Πίνακας 5.6: Χρησιμοποιούμενες σταθερές στον CCLO και HS.



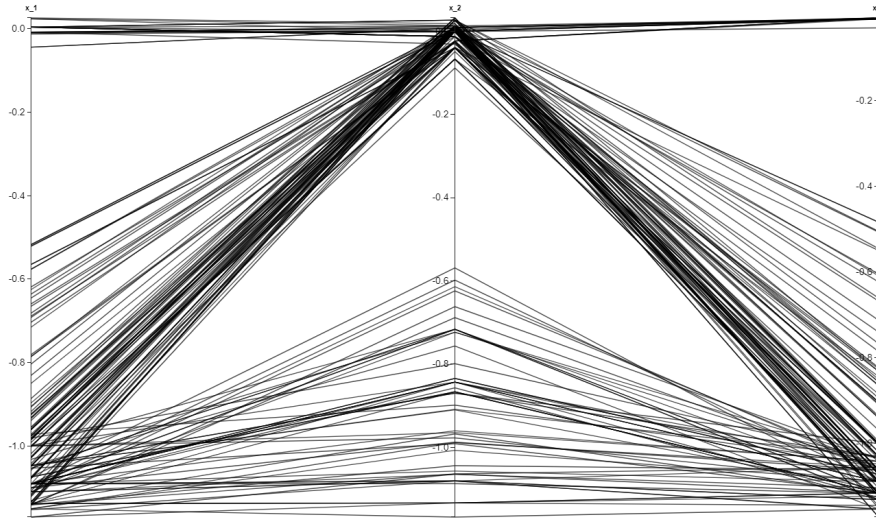
Σχήμα 5.8: Τελικό μέτωπο Pareto μετά τη σύνθεση των 5 επιμέρους μετώπων για HS (μαύρο) και CCLO (κόκκινο).



Σχήμα 5.9: Μέτωπα Pareto πέντε κλήσεων του κώδικα CCLO.

Σχολιασμός αποτελεσμάτων

Μετά την ολοκλήρωση των 1000 αξιολογήσεων, η μέση τιμή του δείκτη υπερόγκου που βρέθηκε με τον κώδικα CCLO ήταν, όπως αναμενόταν, καλύτερη από αυτή του HS. Από την πορεία της μέσης τιμής του δείκτη υπερόγκου του σχήματος 5.7, είναι εμφανές ότι σε όλη τη διάρκεια της βελτιστοποίησης, μετά τους πρώτους 200 υπολογισμούς που χρησιμοποιούνται για τη δημιουργία της βάσης για την εκπαίδευση των μεταπροτύπων, ο αλγόριθμος CCLO έχει καλύτερα αποτελέσματα από τον HS. Και οι δύο κώδικες βρίσκουν σε ικανοποιητικό βαθμό το μέτωπο Pareto, συμπεριλαμβανομένης της μικρής περιοχής κοντά στο $(-20,0)$ όπως φαίνεται στο σχήμα 5.8. Από το σχήμα 5.9, φαίνεται ότι ο κώδικας CCLO, εντόπισε όλες τις περιοχές σε κάθε κλήση του, αλλά σε ορισμένες από αυτές τα σημεία ήταν πιο πυκνά από άλλες. Τέλος, από το διάγραμμα παράλληλων συντεταγμένων φαίνεται ότι σε γενικές γραμμές, οι τιμές των μεταβλητών του μετώπου



Σχήμα 5.10: Διάγραμμα παράλληλων συντεταγμένων μη-κυριαρχούμενων ατόμων σύνθεσης μετώπων Pareto για CCLO.

Pareto του CCLO, είχαν τιμές εντός του διαστήματος $(-1.2, 0)$.

5.2.2 Συνάρτηση Poloni

Η μαθηματική έκφραση της συνάρτησης Poloni είναι [71]

$$\begin{aligned} f_1(X) &= 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2 \\ f_2(X) &= (x_1 + 3)^2 + (x_2 + 1)^2 \end{aligned} \quad (5.7)$$

Όπου

$$\begin{aligned} A_1 &= 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2) \\ A_2 &= 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2) \\ B_1 &= 0.5\sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2) \\ B_2 &= 1.5\sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2) \end{aligned} \quad (5.8)$$

Οι μεταβλητές σχεδιασμού είναι 2, με όρια

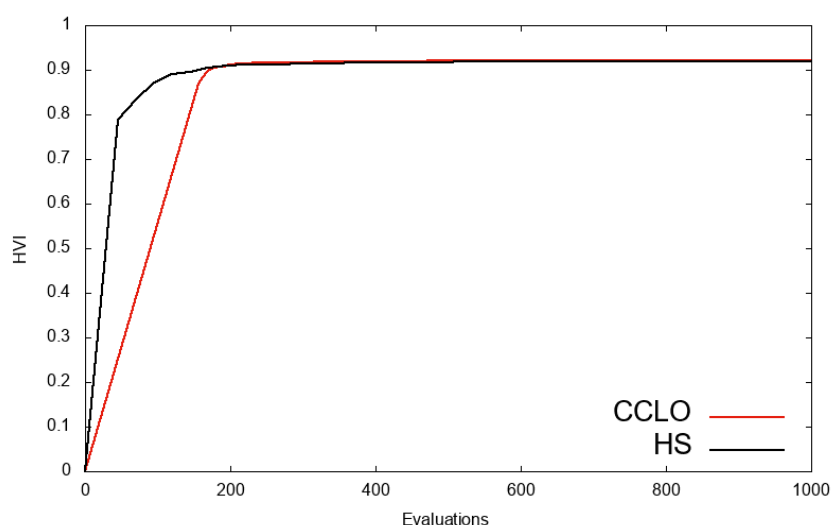
$$-\pi \leq x_1, x_2 \leq \pi \quad (5.9)$$

Στον πίνακα 5.7 φαίνεται ο δείκτης υπερόγκου για τα μέτωπα Pareto των κωδίκων HS CCLO και του λογισμικού EASY. Τα σημεία ναδίρ και ζενίθ που χρησιμοποιήθηκαν

ήταν τα (30,25) και (0,0) αντίστοιχα.

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS	0.919	0.002	0.921	0.920	0.918	0.917	0.921
CCLO	0.920	0.001	0.921	0.921	0.920	0.920	0.920
EASY	0.920	0.002	0.921	0.921	0.918	0.921	0.918

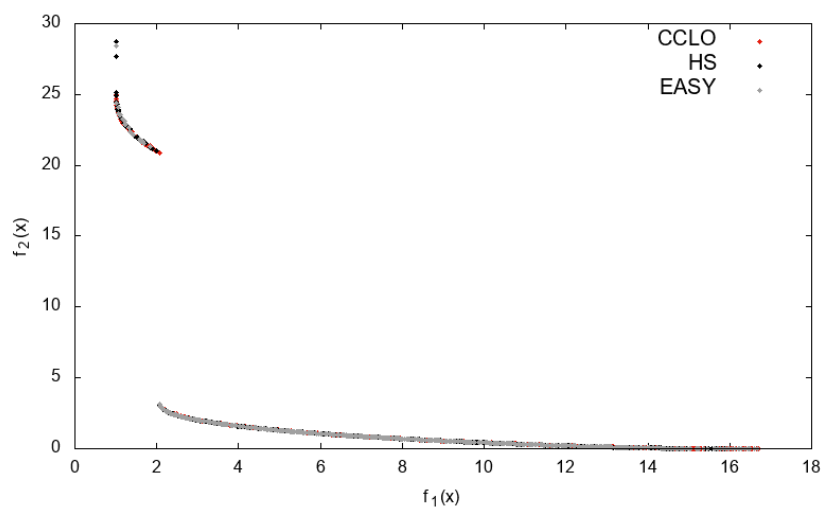
Πίνακας 5.7: Μέση τιμή και τυπική απόκλιση δείκτη υπερόγκου, και αποτέλεσμα δείκτη για κάθε κλήση κώδικα, για HS, CCLO και EASY. Το λογισμικό EASY ενδεχομένως δεν έτρεξε με τις καλύτερες παραμέτρους που θα μπορούσαν να επιλεγούν.



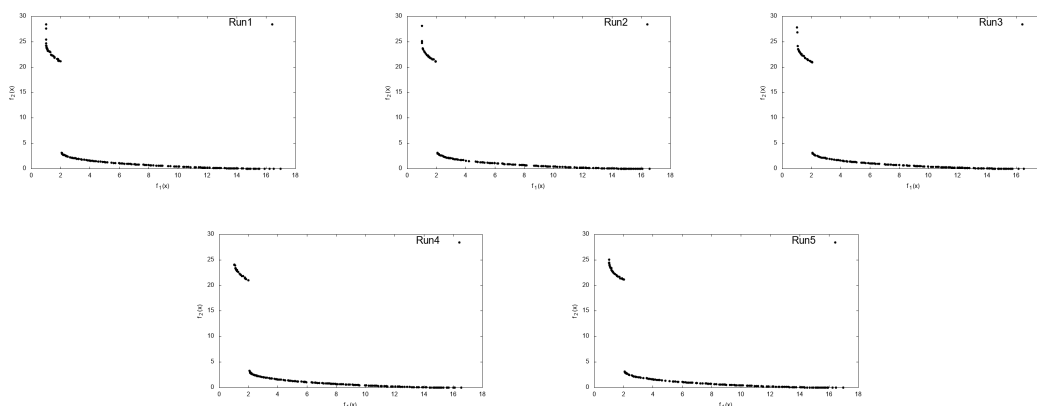
Σχήμα 5.11: Δείκτης υπερόγκου για διαδοχικά μέτωπα μη-κυριαρχούμενων λύσεων του κώδικα CCLO και HS, κόκκινη και μαύρη γραμμή αντίστοιχα.

Στο σχήμα 5.12 συγκρίνονται τα μέτωπα μη-κυριαρχούμενων σημείων από σύνθεση των μετώπων Pareto που προέκυψαν από πέντε κλήσεις του κάθε κώδικα, για τους κώδικες HS, CCLO και EASY. Στο σχήμα 5.13 φαίνονται αναλυτικά τα μέτωπα Pareto των πέντε κλήσεων του κώδικα CCLO.

Οι σταθερές που χρησιμοποιήθηκαν για την εξαγωγή των αποτελεσμάτων αυτών για τους CCLO και HS φαίνονται στον πίνακα 5.8. Για τον εξελικτικό αλγόριθμο του λογισμικού EASY, χρησιμοποιήθηκε $(\mu, \lambda) = (20, 40)$, με τις τιμές αυτές να έχουν επιλεγεί αυθαίρετα.



Σχήμα 5.12: Μέτωπο μη-κυριαρχούμενων σημείων μετά από τη σύνθεση πέντε επιμέρους μετώπων για HS (μαύρο), CCLO (κόκκινο) και EASY (γκρι).



Σχήμα 5.13: Μέτωπα Pareto πέντε κλήσεων του κώδικα CCLO.

Σχολιασμός αποτελεσμάτων

Μετά την ολοκλήρωση των 1000 αξιολογήσεων, η μέση τιμή και η τυπική απόκλιση του δείκτη υπερόγκου που βρέθηκαν με τα τρία λογισμικά, ήταν πρακτικά ίδιες, με τον HS να παρουσιάζει ελαφρώς χειρότερα αποτελέσματα. Ο HS είχε λίγο χαμηλότερες τιμές στην τρίτη και τέταρτη κλήση του. Μετά τις 150 αξιολογήσεις για εξάσκηση των μεταπροτύπων, η πορεία της μέσης τιμής του δείκτη υπερόγκου είναι πρακτικά ίδια για τους κώδικες HS και CCLO, όπως φαίνεται στο σχήμα 5.11. Από το σχήμα 5.12, φαίνεται ότι και τα τρία λογισμικά εντόπισαν όλες τις περιοχές του μετώπου των μη-κυριαρχούμενων λύσεων. Στο σχήμα 5.13 παρατηρείται ότι ο κώδικας CCLO δεν κατάφερε να εντοπίσει τα σημεία με μικρές τιμές f_1 και μεγάλες τιμές f_2 που απείχαν μεγαλύτερες αποστάσεις από τα υπόλοιπα σημεία του μετώπου, στην τέταρτη και πέμπτη κλήση του.

Constants	CCLO	HS
CSS	20	20
MMR	0.8	0.8
LSR	0.6	0.5
bw	1	1
Off	25	1
Ev.Beg.	150	-
Pool	6	-

Πίνακας 5.8: Χρησιμοποιούμενες σταθερές στους CCLO και HS.

5.2.3 Συνάρτηση Schaffer N.1

Η μαθηματική έκφραση της συνάρτησης Schaffer N.1 είναι [71]

$$\begin{aligned} f_1(x) &= x^2 \\ f_2(x) &= (x - 2)^2 \end{aligned} \quad (5.10)$$

Η μεταβλητή σχεδιασμού είναι 1, και για όρια επιλέγονται τα

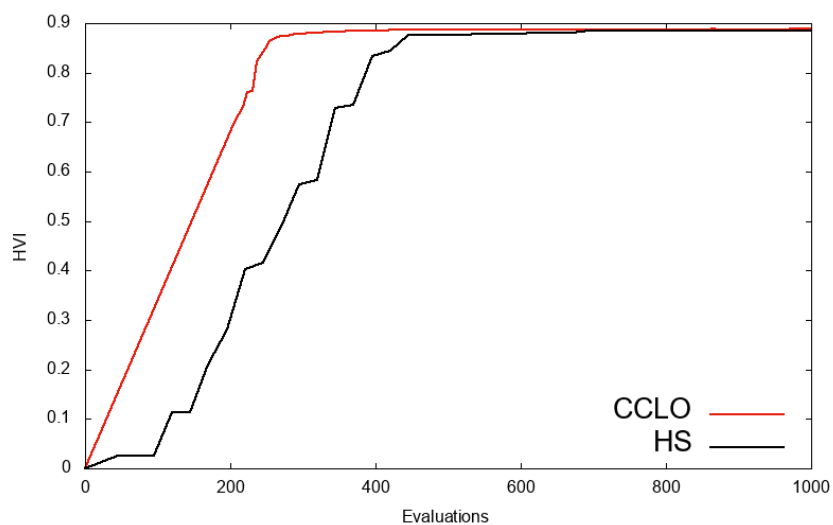
$$-100 \leq x_1 \leq 100 \quad (5.11)$$

Στον πίνακα 5.9 φαίνεται ο HVI για τα μέτωπα Pareto στο τέλος των χιλίων αξιολογήσεων, για τους HS και CCLO. Στο σχήμα 5.14 φαίνεται η μέση πορεία του δείκτη υπερόγκου για τα διαδοχικά μέτωπα μη-κυριαρχούμενων λύσεων για τους κώδικες αυτούς. Τα σημεία ναδίρ και ζενίθ που χρησιμοποιήθηκαν ήταν τα (5,5) και (0,0) αντίστοιχα.

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS	0.886	0.003	0.887	0.886	0.889	0.880	0.887
CCLO	0.888	0.000	0.889	0.888	0.888	0.888	0.889

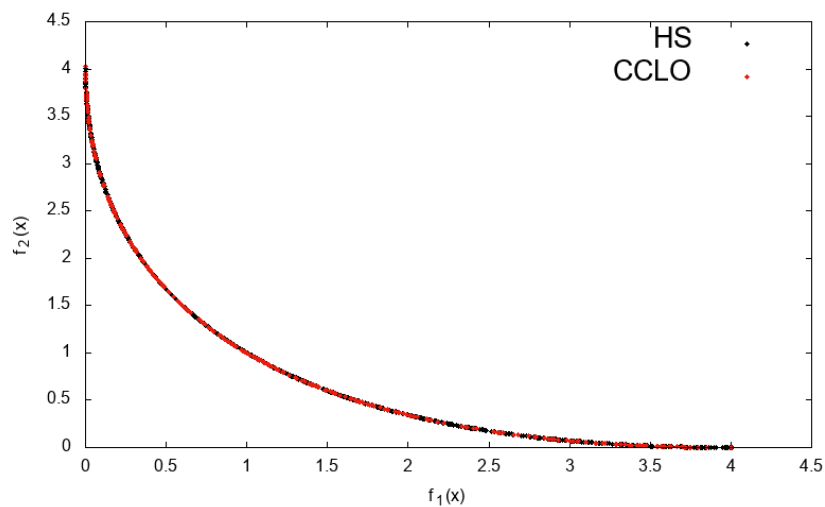
Πίνακας 5.9: Μέση τιμή και τυπική απόκλιση δείκτη υπερόγκου για πέντε κλήσεις κώδικα, και αποτέλεσμα κάθε κλήσης, για HS και CCLO.

Στο σχήμα 5.15 συγκρίνονται μη-κυριαρχούμενα άτομα που προέκυψαν από σύνθεση των μετώπων Pareto για πέντε κλήσεις του κάθε κώδικα, για τους κώδικες HS



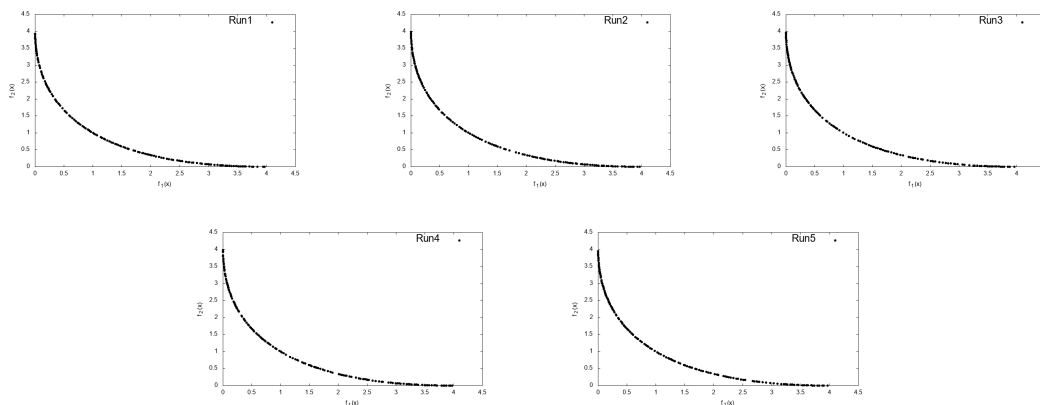
Σχήμα 5.14: Δείκτης υπερόγκου για διαδοχικά μέτωπα μη-κυριαρχούμενων λύσεων του κώδικα CCLO και HS, κόκκινη και μαύρη γραμμή αντίστοιχα.

και CCLO. Στο σχήμα 5.16 παρουσιάζονται αναλυτικά τα μέτωπα Pareto των πέντε κλήσεων του κώδικα CCLO.



Σχήμα 5.15: Τελικό μέτωπο Pareto μετά τη σύνθεση των 5 επιμέρους μετώπων, για HS (μαύρο), και CCLO (κόκκινο).

Οι σταθερές που χρησιμοποιήθηκαν για την εξαγωγή των αποτελεσμάτων αυτών για τους CCLO και HS, φαίνονται στον πίνακα 5.10.



Σχήμα 5.16: Μέτωπα Pareto πέντε κλήσεων του κώδικα CCLO.

Constants	CCLO	HS
CSS	20	20
MMR	0.9	0.9
LSR	0.4	0.5
bw	6	6
Off	25	1
Ev.Beg.	200	-
Pool	5	-

Πίνακας 5.10: Χρησιμοποιούμενες σταθερές στους CCLO και HS.

Σχολιασμός αποτελεσμάτων

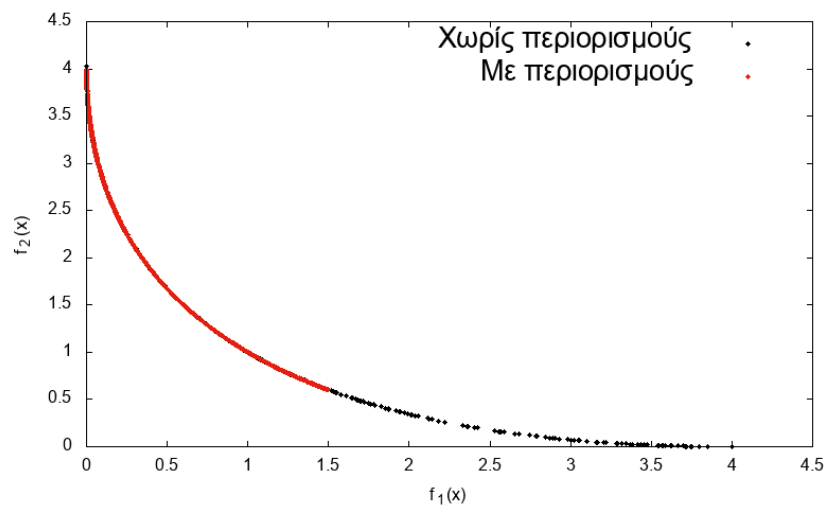
Μετά την ολοκλήρωση των χιλίων αξιολογήσεων, η μέση τιμή του δείκτη υπερόγκου που βρέθηκε με τον κώδικα CCLO ήταν, και σε αυτήν την περίπτωση, καλύτερη από αυτή του κώδικα HS. Ήδη από την ολοκλήρωση των 200 πρώτων αξιολογήσεων, που χρησιμοποιήθηκαν για τη δημιουργία της βάσης δεδομένων για εκπαίδευση των μεταπροτύπων για το λογισμικό CCLO, από το σχήμα 5.14 φαίνεται ότι ο CCLO έχει καλύτερα αποτελέσματα από τον HS. Από το σχήμα 5.15, φαίνεται ότι και τα δύο λογισμικά εντόπισαν όλες τις περιοχές του μετώπου των μη-κυριαρχούμενων λύσεων. Μάλιστα, στο σχήμα 5.16 παρατηρείται ότι ο κώδικας CCLO κατάφερε να εντοπίσει κάθε περιοχή του μετώπου Pareto, με ομοιόμορφη κατανομή σημείων σε όλο το μέτωπο.

5.2.4 Συνάρτηση Schaffer N.1 με περιορισμούς

Τέλος, για πιστοποίηση της λειτουργίας των περιορισμών, παρουσιάζεται η συνάρτηση Schaffer N.1 της ενότητας 5.2.3, υπό τον περιορισμό

$$f_1(x) \leq 1.5 \quad (5.12)$$

Τα αποτελέσματα του κώδικα CCLO με και χωρίς περιορισμούς φαίνονται στο σχήμα 5.17.



Σχήμα 5.17: Μέτωπο Pareto για κώδικα CCLO με και χωρίς περιορισμούς, μαύρα και κόκκινα σημεία αντίστοιχα.

Σχολιασμός αποτελεσμάτων

Παρατηρείται ότι στην περίπτωση των περιορισμών, το μέτωπο Pareto είναι πιο πυκνό και χωρίς σημαντικά κενά στην περιοχή αυτή. Αυτό είναι αναμενόμενο, αφού η αναζήτηση γίνεται σε μικρότερη περιοχή, το μέτωπο θα εντοπιστεί καλύτερα.

Κεφάλαιο 6

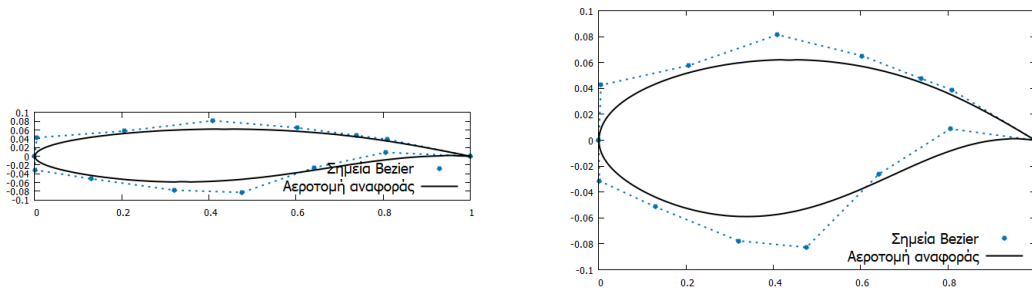
Εφαρμογές στην Αεροδυναμική και τα Οικονομικά

Στο κεφάλαιο αυτό εφαρμόστηκε ο κώδικας CCLO σε μοντελοποιημένα προβλήματα αεροδυναμικής και οικονομικών, και τα αποτελέσματα συγκρίθηκαν με τα αντίστοιχα του λογισμικού EASY με μεταπρότυπα, και του HS. Το πρόβλημα της μεμονωμένης αεροτομής ήταν μοντελοποιημένο από τη ΜΠΥΡ&Β, και χρησιμοποιήθηκε ως «μαύρο κουτί». Αντίθετα, το πρόβλημα της οικονομικής ποσότητας παραγγελίας μοντελοποιήθηκε κατά την εκπόνηση της διπλωματικής αυτής εργασίας. Για τον λόγο αυτό, το πρόβλημα περιγράφεται αναλυτικά στο κεφάλαιο αυτό.

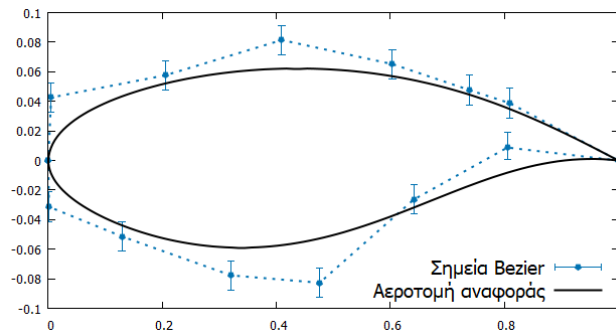
6.1 Μεμονωμένη Αεροτομή

Στην περίπτωση αυτή εξετάζεται μια μεμονωμένη αεροτομή, η RAE 2822, το σχήμα της οποίας μεταβάλλεται ώστε να μεγιστοποιηθεί ο συντελεστής άνωσης, ή και να ελαχιστοποιηθεί ο συντελεστής οπισθέκουσας. Στην περίπτωση των δύο συναρτήσεων - στόχων εξετάζεται και η περίπτωση στην οποία τίθεται περιορισμός ως προς το συντελεστή ροπής. Οι συνθήκες της επί άπειρον ροής είναι για τον αριθμό Mach, $M_\infty=0.725$ και για τη γωνία της ροής, $\alpha_\infty = 5^\circ$. Γίνεται η παραδοχή ότι η ροή είναι ατριβής, επομένως χρησιμοποιούνται οι εξισώσεις Euler.

Σε πρώτο στάδιο, από τη ΜΠΥΡ&Β μοντελοποιήθηκε η αεροτομή με δύο πολυώνυμα Bezier 8 σημείων ελέγχου για το καθένα, για τις πλευρές υποπίεσης και υπερπίεσης, με το 1° και το 8° να ταυτίζονται για τις δύο πλευρές, αποτελώντας τα σημεία πρόσπτωσης και εκφυγής όπως φαίνεται στο σχήμα 6.1. Με τα σημεία αυτά έγινε προσπάθεια η αεροτομή να προσεγγίζει με τον καλύτερο δυνατό τρόπο την αεροτομή αναφοράς [62].



Σχήμα 6.1: Δημιουργία αεροτομής από σημεία Bezier. Αριστερά σε κλίμακα, δεξιά χωρίς κλίμακα.



Σχήμα 6.2: Εύρη σημείων Bezier για τη δημιουργία αεροτομής.

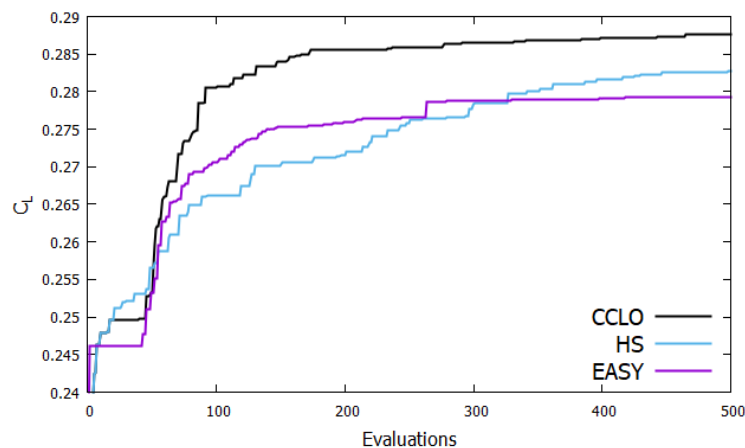
Στη συνέχεια, για καθένα από τα σημεία και για τις δύο πλευρές, πλην του 1^{ου} και το 8^{ου}, δόθηκε ένα επιτρεπτό εύρος κίνησης στον y άξονα, της τάξης του 20% της τιμής τους, μέσα στο οποίο μπορούν να μεταβληθούν οι τιμές τους για να επιτευχθεί η βέλτιστη αεροτομή. Τα επιτρεπτά εύρη κίνησης φαίνονται στο σχήμα 6.2.

Για τις αεροτομές που δημιουργήθηκαν υπολογίστηκαν οι συντελεστές άνωσης, οπισθέλκουσας και ροπής, με χρήση του επιλύτη ροής PUMA [72]. Για κάθε αεροτομή δημιουργήθηκε μη-δομημένο πλέγμα με τη μέθοδο (λογισμικό) του προελαύνοντος μετώπου (advancing-front method), το οποίο χρησιμοποιήθηκε για τον υπολογισμό του πεδίου ροής γύρω από την αεροτομή. Από το πεδίο αυτό υπολογίζονται οι ζητούμενοι συντελεστές.

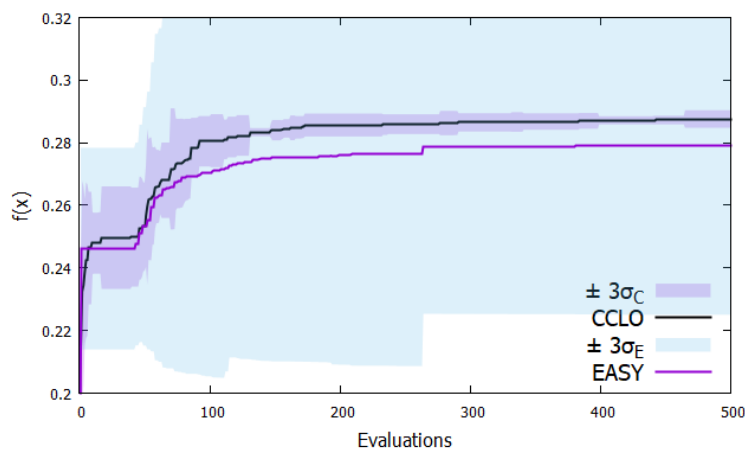
6.1.1 Πρόβλημα ενός στόχου

Στο πρόβλημα αυτό, μοναδικός στόχος ήταν η μεγιστοποίηση του συντελεστή άνωσης (C_L) της αεροτομής.

Στο σχήμα 6.5 φαίνεται η πορεία της μέσης τιμής του συντελεστή C_L των πέντε κλήσεων των HS, CCLO και EASY συναρτήσει των κλήσεων του λογισμικού Υπολογιστικής Ρευστοδυναμικής. Στο σχήμα 6.5 φαίνεται η τυπική απόκλιση στους CCLO και EASY, η οποία παρατηρείται ότι είναι σημαντικά μεγαλύτερη για το λογισμικό EASY.



Σχήμα 6.3: Μέση τιμή του συντελεστή άνωσης για 5 κλήσεις των κωδίκων HS, CCLO και EASY. Το λογισμικό EASY ενδεχομένως δεν έτρεξε με τις καλύτερες παραμέτρους που θα μπορούσαν να επιλεγούν.



Σχήμα 6.4: Μέση τιμή και τυπική απόκλιση συντελεστή άνωσης για 5 κλήσεις των κωδίκων CCLO και EASY.

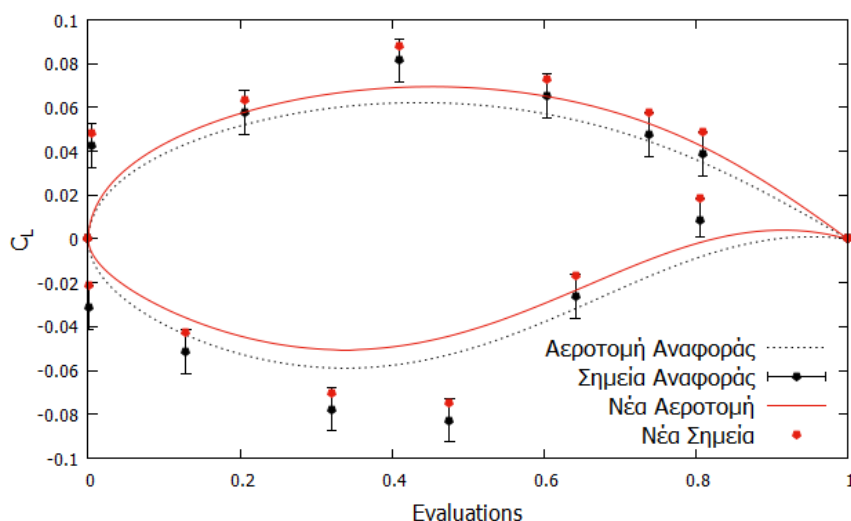
Στον πίνακα 6.1 φαίνονται η μέση τιμή και η τυπική απόκλιση των CCLO, HS και EASY στο τέλος των 500 αξιολογήσεων. Τα σημεία αυτά αντιστοιχούν στις αεροτομές του σχήματος 6.5. Παρά τα τελικά αποτελέσματα της μέσης τιμής, ο κώδικας EASY έδωσε

εξίσου καλά αποτελέσματα με τον CCLO σε όλες της κλήσεις του, πλην της δεύτερης που τα αποτελέσματα ήταν σημαντικά χειρότερα, γιαυτό και μειώθηκε η μέση τιμή.

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS	0.2827	0.0009	0.2819	0.2817	0.2837	0.2834	0.2827
CCLO	0.2876	0.0009	0.2865	0.2877	0.2871	0.2875	0.2890
EASY	0.2792	0.0181	0.2877	0.2469	0.2873	0.2870	0.2873

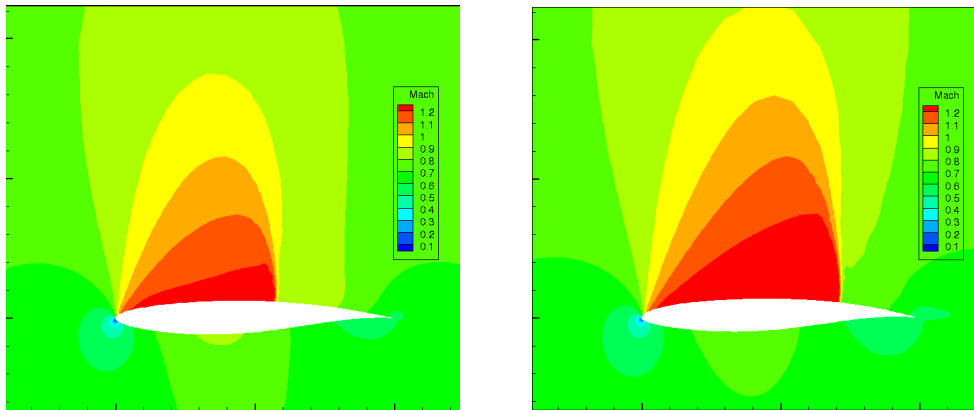
Πίνακας 6.1: Μέση τιμή, τυπική απόκλιση και βέλτιστο κάθε κλήσης του κώδικα για CCLO και EASY.

Η νέα αεροτομή που προέκυψε από τον κώδικα CCLO, καθώς και η αρχική αεροτομή RAE 2822 με τα όρια που τέθηκαν για τις μεταβλητές σχεδιασμού της, φαίνονται στο σχήμα 6.5. Στόχος ήταν η μεγιστοποίηση του συντελεστή άνωσης, επομένως η βελτιστοποιημένη αεροτομή έχει μεγαλύτερη καμπυλότητα, γεγονός που επιτρέπει μεγαλύτερη ταχύτητα στην περιοχή υποπίεσης, δηλαδή μικρότερη πίεση στην περιοχή αυτή, επομένως μεγαλύτερη δύναμη άνωσης.



Σχήμα 6.5: Νέα αεροτομή (κόκκινο) σε σύγκριση με την αεροτομή αναφοράς (μαύρο) υπό κλίμακα.

Ο αριθμός Mach γύρω από τη βέλτιστη αεροτομή που δημιουργήθηκε από τον CCLO και την αρχική, δηλαδή τη μοντελοποιημένη RAE 2822, φαίνονται στο σχήμα 6.6. Ο υπερηχητικός θύλακας φαίνεται σημαντικά αυξημένος στη βελτιστοποιημένη αεροτομή, ενώ ο αριθμός Mach στην πλευρά υπερπίεσης φαίνεται πρακτικά σταθερός. Αυτό έχει ως αποτέλεσμα μικρότερη πίεση στην πλευρά υποπίεσης και σταθερή στην πλευρά υπερπίεσης, κατά συνέπεια μεγαλύτερη διαφορά πίεσης των δύο περιοχών, δηλαδή μεγαλύτερη δύναμη άνωσης.



Σχήμα 6.6: Αριθμός Mach γύρω από το αρχικό περύγιο στα αριστερά, και γύρω από το βελτιωμένο στα δεξιά.

Για τους CCLO και HS, οι σταθερές που χρησιμοποιήθηκαν φαίνονται στον πίνακα 6.7. Στον αλγόριθμο EASY χρησιμοποιήθηκαν οι σταθερές $(\mu, \lambda) = (20, 40)$.

Constants	CCLO	HS
CSS	20	20
MMR	0.8	0.8
LSR	0.5	0.5
bw	0.004	0.004
Off	25	1
Ev.Beg.	50	-
Pool	15	-

Πίνακας 6.2: Χρησιμοποιούμενες σταθερές στον CCLO και τον HS.

6.1.2 Πρόβλημα δύο στόχων

Στο πρόβλημα αυτό, στόχος ήταν η μεγιστοποίηση του συντελεστή άνωσης (C_L) με παράλληλα την ελαχιστοποίηση του συντελεστή οπισθέλκουσας (C_D) της αεροτομής.

Στον πίνακα 6.3 φαίνεται η τελική τιμή του συντελεστή υπερόγκου, για τα ίδια σημεία ζενίθ και ναδίρ, για τα μέτωπα Pareto του κωδικών CCLO και EASY.

Στο σχήμα 6.7 φαίνονται το μέτωπο των μη-κυριαρχούμενων σημείων μετά από σύνθεση των μετώπων Pareto που έδωσαν οι πέντε κλήσεις των λογισμικών CCLO και EASY. Στο ίδιο σχήμα παρουσιάζεται επιπλέον η μορφή της αεροτομής τριών σημείων του μετώπου, και το πεδίο Mach γύρω της.

Παρατηρείται ότι τα δύο λογισμικά έχουν εντοπίσει ικανοποιητικά το ζητούμενο μέτωπο, με τον δείκτη υπερόγκου του CCLO να παρουσιάζει λίγο καλύτερα αποτελέσματα. Το

Software	μ	σ	Run1	Run2	Run3	Run4	Run5
CALO	0.758	0.005	0.760	0.759	0.758	0.749	0.762
EASY	0.756	0.005	0.759	0.755	0.749	0.755	0.762

Πίνακας 6.3: Μέση τιμή και τυπική απόκλιση δείκτη υπερόγκου και δείκτη υπερόγκου κάθε κλήσης του κώδικα για CALO EASY.

λογισμικό EASY ωστόσο, ενδεχομένως δεν έτρεξε με τις καλύτερες παραμέτρους που θα μπορούσαν να επιλεγούν.

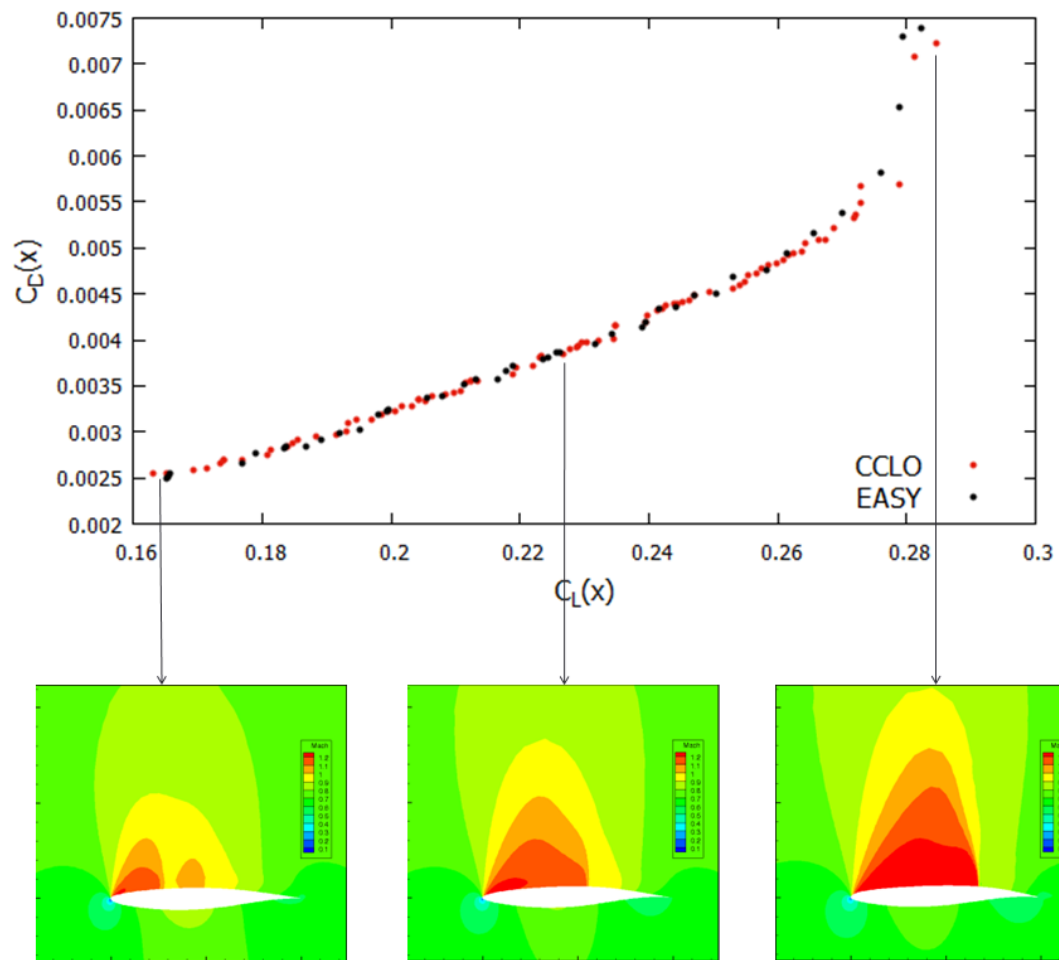
Για σημεία με μεγαλύτερες τιμές στους συντελεστές άνωσης και οπισθέλκουσας, η αεροτομή έχει μεγαλύτερη καμπυλότητα. Λόγω της μεγάλης καμπυλότητας, η απόσταση που πρέπει να διανυθεί στην πλευρά υποπίεσης είναι μεγαλύτερη, ενώ στην πλευρά υπερπίεσης πρακτικά σταθερή για κάθε περίπτωση. Αυτό έχει ως αποτέλεσμα ο αριθμός Mach του υπερηχητικού θύλακα να είναι μεγαλύτερος, ενώ η ταχύτητα στην πλευρά υπερπίεσης είναι πρακτικά ίδια σε όλες τις περιπτώσεις, όπως φαίνεται στο σχήμα. Η μεγάλη διαφορά πίεσης των δύο πλευρών προκαλεί μεγάλη δύναμη άνωσης. Επιπλέον, λόγω της μεγάλης ανάκτησης πίεσης μετά το κάθετο κύμα κρούσης του υπερηχητικού θύλακα, και η δύναμη οπισθέλκουσας θα είναι μεγαλύτερη.

Στο σχήμα 6.8 φαίνονται οι παράλληλες συντεταγμένες των μεταβλητών του μετώπου Pareto. Παρατηρείται ότι οι περισσότερες μεταβλητές έλαβαν οριακές τιμές, μέγιστες ή ελάχιστες, και συγκεκριμένα για τις μεταβλητές x_7 έως x_9 οι τιμές που προέκυψαν ήταν κοντά στο κάτω όριο ενώ για την x_{10} οι τιμές που προέκυψαν ήταν κοντά στο άνω όριο.

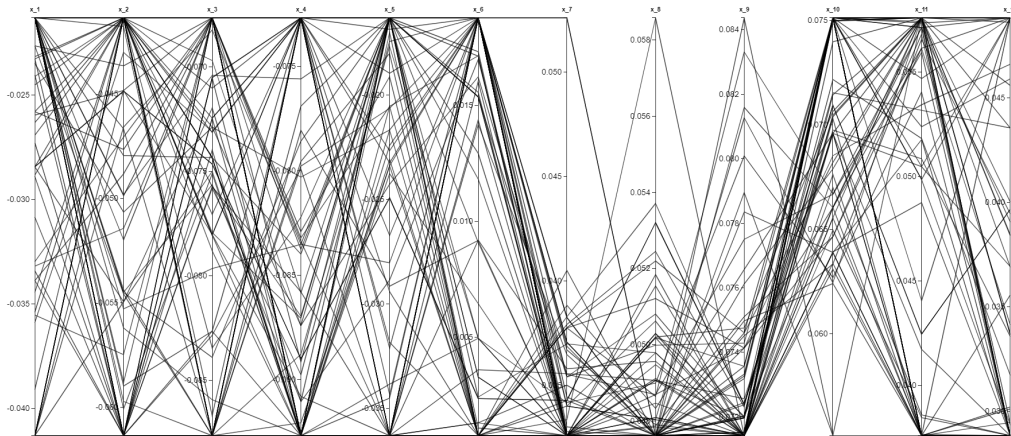
Οι σταθερές που χρησιμοποιήθηκαν για τον κώδικα CALO φαίνονται στον πίνακα 6.7. Στον αλγόριθμο EASY χρησιμοποιήθηκαν οι σταθερές $(\mu, \lambda) = (20, 40)$.

Constants	CALO
CSS	30
MMR	0.8
LSR	0.5
bw	0.008
Off	25
Ev.Beg.	100
Pool	30

Πίνακας 6.4: Χρησιμοποιούμενες σταθερές στον CALO και τον CALO.



Σχήμα 6.7: Πάνω: Μη-κυριαρχούμενα σημεία μετά τη σύνθεση των 5 επιμέρους μετώπων Pareto από τις κλήσεις του κώδικα CCLO και του λογισμικού EASY. Κάτω: Ροή Mach γύρω από το πτερύγιο για τρία σημεία του μετώπου του κώδικα CCLO.



Σχήμα 6.8: Παράλληλες συντεταγμένες για το μέτωπο μη-κυριαρχούμενων λύσεων των πέντε μετώπων Pareto του CCLO.

6.1.3 Πρόβλημα δύο στόχων υπό περιορισμούς

Στο πρόβλημα αυτό, στόχος ήταν η μεγιστοποίηση του συντελεστή άνωσης (C_L) και ελαχιστοποίηση του συντελεστή οπισθέλκουσας (C_D) της αεροτομής, υπό περιορισμό για το συντελεστή ροπής (C_M). Η ροπή έχει υπολογιστεί γύρω από το 1/4 της ακτίνας της αεροτομής.

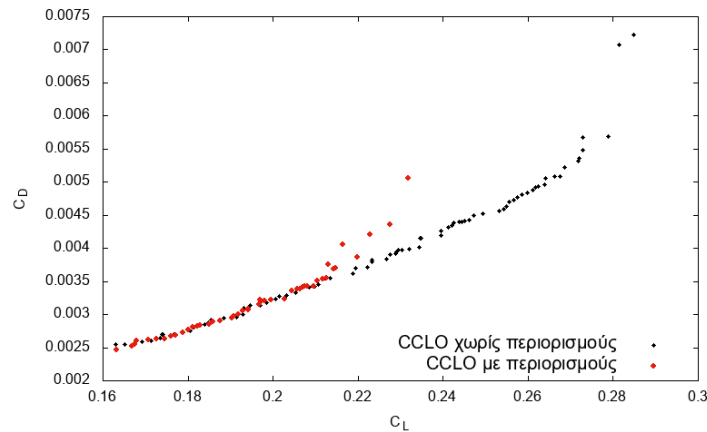
Ο περιορισμός που τέθηκε ήταν

$$C_M < 0.03 \quad (6.1)$$

Τα μη-κυριαρχούμενα σημεία της σύνθεσης των μετώπων Pareto με τον αλγόριθμο CCLO, για τις περιπτώσεις της εφαρμογής και μη περιορισμού για τον συντελεστή ροπής, φαίνεται στο σχήμα 6.9.

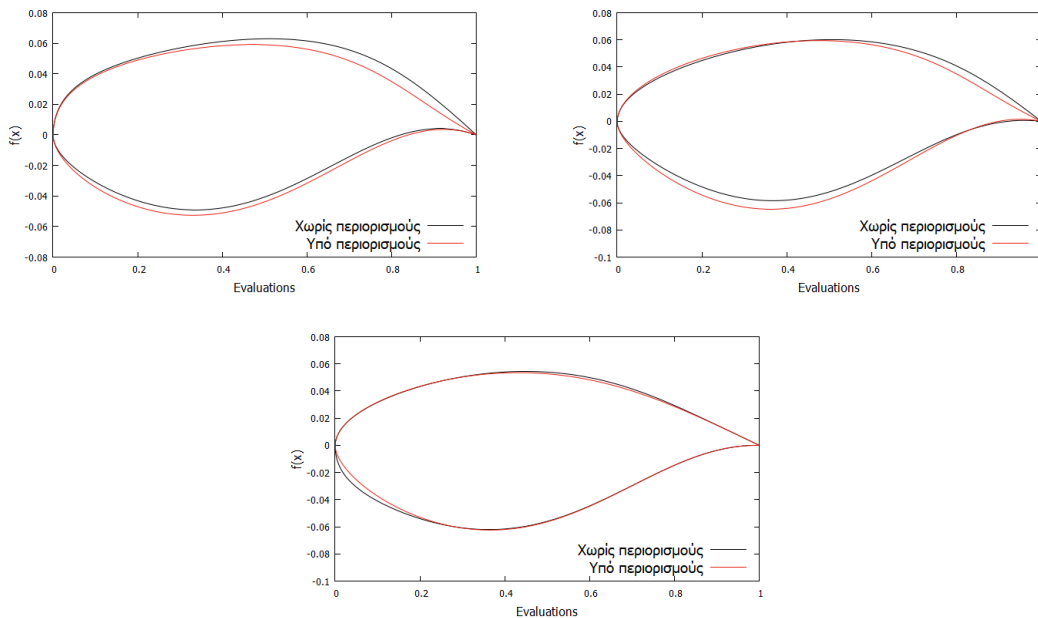
Παρατηρείται ότι στην περίπτωση των περιορισμών δεν εντοπίζονται σημεία με μεγάλους συντελεστές άνωσης και οπισθέλκουσας. Αυτό συμβαίνει επειδή, όπως έχει περιγραφεί στις προηγούμενες ενότητες, τα πτερύγια που παρουσιάζουν μεγάλους συντελεστές έχουν και μεγάλη καμπυλότητα. Λόγω της καμπυλότητας αυτής ο συντελεστής ροπής είναι αυξημένος, επομένως τα πτερύγια απορρίπτονται από την εφαρμογή του περιορισμού.

Στη συνέχεια, στο σχήμα 6.10 συγκρίνονται τρία πτερύγια των μετώπων Pareto του προβλήματος των δύο συναρτήσεων - στόχων, με και χωρίς περιορισμούς. Το πρώτο πτερύγιο έχει μικρές τιμές στους συντελεστές C_L και C_D του μετώπου Pareto, το δεύ-



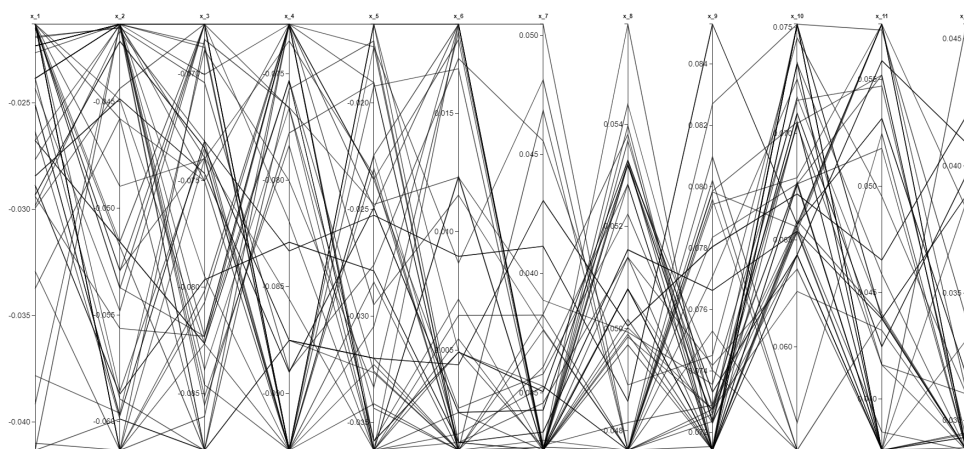
Σχήμα 6.9: Μη-κυριαρχούμενα σημεία μετώπων Pareto σε πρόβλημα υπό περιορισμούς (κόκκινο), συγκρινόμενο με το πρόβλημα χωρίς περιορισμούς (μαύρο).

τερο μεσαίες και το τρίτο μεγάλες. Στη σύγκριση, χρησιμοποιούνται τα πτερύγια του σχήματος 6.7 για την περίπτωση που δεν τίθενται περιορισμοί. Παρατηρείται ότι όσο μεγαλύτεροι είναι οι συντελεστές τόσο μεγαλύτερη απόκλιση έχουν τα δύο πτερύγια. Αυτό συμβαίνει διότι πτερύγια με μεγάλες τιμές στους συντελεστές άνωσης και οπισθέλκουσας έχουν μεγάλη καμπυλότητα, με αποτέλεσμα να απορρίπτονται. Αντίθετα, πτερύγια μικρών συντελεστών έχουν και μικρότερη καμπυλότητα, με αποτέλεσμα να μη ξεπερνάνε τον επιτρεπτό συντελεστή ροπής.



Σχήμα 6.10: Σύγκριση πτερυγίων με περιορισμό συντελεστή ροπής (κόκκινη γραμμή) και χωρίς (μαύρη γραμμή). Το πρώτο σχήμα προκύπτει σε μεγάλες τιμές συντελεστών άνωσης και οπισθέλκουσας, το δεύτερο μεσαίες και το τρίτο για μικρές

Το διάγραμμα παράλληλων συντεταγμένων φαίνεται στο σχήμα 6.11.



Σχήμα 6.11: Παράλληλες συντεταγμένες για τη σύνθεση των μετώπων Pareto των πέντε κλήσεων του CCLO.

Οι σταθερές που χρησιμοποιήθηκαν στον κώδικα CCLO φαίνονται στον πίνακα 6.7.

Constants	CCLO
CSS	20
MMR	0.8
LSR	0.5
bw	0.008
Off	25
Ev.Beg.	100
Pool	30

Πίνακας 6.5: Χρησιμοποιούμενες σταθερές στον CCLO.

6.2 Οικονομική Ποσότητα Παραγγελίας

6.2.1 Θεωρητικό Υπόβαθρο

Στην περίπτωση αυτή εξετάζεται ένα πρόβλημα διαχείρισης αποθέματος που βασίζεται στο μοντέλο Οικονομικής Ποσότητας Παραγγελίας (Economic Order Quantity - EOQ) [73], στην περίπτωση των προγραμματισμένων καθυστερήσεων [74]. Το EOQ αποτελεί σύστημα ελέγχου αποθέματος που υπολογίζει την ποσότητα που πρέπει να παραγγελθεί για να ελαχιστοποιηθεί το κόστος διαχείρισης αποθέματος (handling cost) και διαχείρισης της παραγγελίας (ordering cost) [75].

Η διοίκηση της εφοδιαστικής αλυσίδας είναι κομβικής σημασίας για τη λειτουργία κάθε επιχείρησης. Το κεφάλαιο που είναι κάθε στιγμή επενδυμένο σε αποθέματα μπορεί να φτάσει έως και 50% του ενεργητικού της επιχείρησης [76]. Ο ανεπαρκής έλεγχος των αποθεμάτων, μπορεί να οδηγήσει στη δημιουργία υπεραποθέματος ή υποαποθέματος. Η δημιουργία υπεραποθέματος έχει ως συνέπεια την κατασπατάληση πόρων για τη διαχείρισή του, που αποτελεί σημαντική επιβάρυνση για τη λειτουργία της επιχείρησης. Το υποαπόθεμα οδηγεί σε ελλείψεις, οι οποίες βλάπτουν το όνομα της επιχείρησης στην αγορά, προκαλώντας μείωση στους πελάτες της. Εναλλακτικά, για τη διαχείριση των καταστάσεων αυτών η επιχείρηση μπορεί να προβεί σε υπερωρίες, υπεργολαβίες προϊόντων, ή μη προγραμματισμένες μειώσεις τιμών, παράγοντες που προκαλούν επιπλέον οικονομικές επιβαρύνσεις στην επιχείρηση [77].

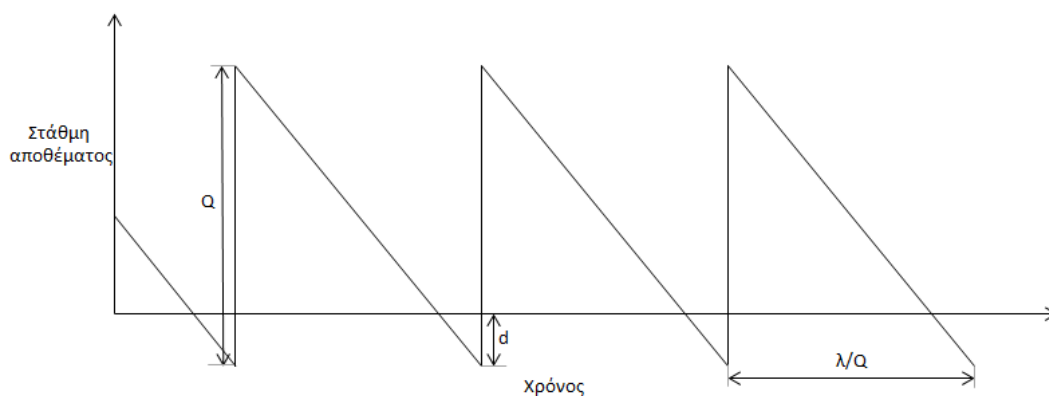
Η μεγάλη δυσκολία του ελέγχου αποθεμάτων έγκειται κυρίως στο γεγονός ότι για την κατάλληλη διαχείρισή των αποθεμάτων πρέπει να γίνουν ακριβείς προβλέψεις για την εξέλιξη της αγοράς. Η ζήτηση, μέσω παρακολούθησης ιστορικών στοιχείων, εποχικότητας και πορείας της αγοράς, μπορεί να προβλεφθεί σε ικανοποιητικό βαθμό, ωστόσο ποτέ δεν μπορεί να υπολογισθεί με βεβαιότητα. Ομοίως, οι μελλοντικές μεταβολές στην οικονομία μπορούν να υπολογισθούν, αλλά και πάλι είναι αδύνατο να σταθμιστούν όλοι οι παράγοντες που θα την επηρεάσουν.

Οι παραδοχές που γίνονται στο μοντέλο αυτό είναι οι ακόλουθες [78]:

- Το μοντέλο EOQ θα εφαρμοσθεί για κάθε προϊόν της επιχείρησης ξεχωριστά.
- Η ζήτηση του προϊόντος είναι γνωστή με ακρίβεια.
- Το κόστος παραγγελίας είναι γνωστό και σταθερό σε όλη τη διάρκεια του χρόνου.
- Το κόστος διαχείρισης του αποθέματος είναι γνωστό και σταθερό σε όλη τη διάρκεια του χρόνου.
- Η ποσότητα παραγγελίας είναι σταθερή σε κάθε παραγγελία.
- Η ποσότητα παραγγελίας φτάνει ολόκληρη κατά την ίδια μεταφορά στην επιχείρηση.

- Η τοποθέτηση των προϊόντων της παραγγελίας στην αποθήκη γίνεται τη στιγμή της παραλαβής τους.
- Η παραγγελία φτάνει την προβλεπόμενη ώρα στην επιχείρηση.
- Τα προϊόντα που ήταν σε καθυστέρηση χρησιμοποιούνται τη στιγμή της νέας παραλαβής.

Στην περίπτωση των προγραμματισμένων καθυστερήσεων που εξετάζεται, ο προγραμματισμός γίνεται με τρόπο ώστε για μια μικρή χρονική περίοδο να υπάρχει καθυστέρηση στις παραγγελίες. Αυτό είναι επιθυμητό στην περίπτωση που το κόστος διαχείρισης του επιπλέον αποθέματος είναι μεγαλύτερο από αυτό της έλλειψης. Η ποσότητα προϊόντων στο απόθεμα συναρτήσει του χρόνου φαίνεται στο σχήμα 6.12.



Σχήμα 6.12: Διακύμανση αποθέματος συναρτήσει χρόνου.

Οι μέσες συνολικές ετήσιες δαπάνες αποτελούνται από τις τρεις συνιστώσες που αναλύονται παρακάτω [76].

Μέση δαπάνη προμήθειας

Η δαπάνη αυτή περιλαμβάνει το κόστος αγοράς του προϊόντος καθώς και το κόστος επικοινωνίας, διαχείρισης, μεταφοράς, παραλαβής και ελέγχου του προϊόντος. Σε κάθε παραγγελία το κόστος αυτό αναλύεται στη σταθερή και τη μεταβλητή δαπάνη,

$$\Delta_{\pi} = \lambda c + A \frac{\lambda}{Q} \quad (6.2)$$

όπου με Δ_{π} συμβολίζεται η δαπάνη προμήθειας, με λ η συνολική ετήσια ζήτηση, με c η μοναδιαία αξία του προϊόντος, με A οι σταθερές δαπάνες ανά παραγγελία λόγω επικοινωνίας, διαχείρισης, μεταφοράς, παραλαβής και ελέγχου του προϊόντος και με Q η ποσότητα προϊόντων ανά παραγγελία.

Μέση δαπάνη αποθήκευσης ή αποθεματοποίησης

Η δαπάνη αυτή περιλαμβάνει το κόστος διαχείρισης και διατήρησης των προϊόντων στην αποθήκη, λόγω του κόστους της ενοικίασης της αποθήκης, της λειτουργίας της, της

ασφάλισης των προϊόντων, της φθοράς τους, της μείωσης της αξίας τους, του κόστους δέσμευσης του κεφαλαίου κ.ά. Η δαπάνη αυτή θεωρείται ανάλογη με την ποσότητα του αποθέματος. Η έκφρασή της είναι η ακόλουθη

$$\Delta_{\alpha} = \frac{cI(Q-d)^2}{2Q} \quad (6.3)$$

όπου Δ_{α} η δαπάνη αποθήκευσης, d η ποσότητα που βρίσκεται σε καθυστέρηση όταν καταφθάνει η νέα παραγγελία και I το ποσοστιαίο κόστος διατήρησης αποθέματος ανά μονάδα χρόνου, σε €/έτος.

Μέση δαπάνη υποαποθέματος

Η δαπάνη αυτή οφείλεται στο κέρδος που χάθηκε από πωλήσεις που δεν έγιναν λόγω ανεπαρκούς αποθέματος ή στην έμμεση ζημία που προκλήθηκε λόγω επιπτώσεων στη φήμη της επιχείρησης. Η έκφρασή της παρουσιάζεται παρακάτω

$$\Delta_{\nu} = \frac{C_d d^2}{2Q} \quad (6.4)$$

όπου με Δ_{ν} συμβολίζεται η δαπάνη υποαποθέματος, με C_d το κόστος καθυστέρησης μιας μονάδας αποθέματος ανά έτος.

Από τα παραπάνω, οι Μέσες Ετήσιες Δαπάνες (ΜΕΔ) υπολογίζονται ως άθροισμα των δαπανών που παρουσιάστηκαν

$$\Delta = \Delta_{\pi} + \Delta_{\alpha} + \Delta_{\nu} \quad (6.5)$$

6.2.2 Πρόβλημα βελτιστοποίησης

Παρακάτω παρουσιάζεται το πρόβλημα ΕΟQ, όπως αυτό τίθεται στο [79], το οποίο έχει ελαφρώς τροποποιηθεί και στο οποίο έχουν προστεθεί προγραμματισμένες καθυστερήσεις.

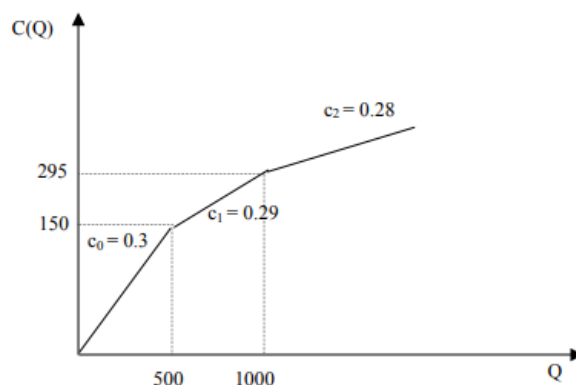
Η ετήσια ζήτηση ενός υγρού προϊόντος A σε μια επιχείρηση είναι $\lambda = 600$ lt/έτος, το σταθερό κόστος παραγγελίας A είναι $A = 8$ €/παρ., το μοναδιαίο κόστος διατήρησης του A , που αντιστοιχεί στο κόστος διατήρησης αποθέματος αξίας ενός ευρώ για ένα χρόνο, είναι $I = 20\%$ €/έτος και το κόστος καθυστέρησης του A 5€/ltέτος.

Στη μοναδιαία αξία του προϊόντος A γίνονται κλιμακωτά εκπτώσεις, με την τιμή του προϊόντος όπως φαίνεται στον πίνακα 6.6.

Από τα παραπάνω προκύπτει ότι το κόστος αγοράς της συνολικής ποσότητας κάθε παραγγελίας συναρτῆσει της ποσότητας ακολουθεί την τεθλασμένη γραμμή του σχήματος 6.13.

Ποσότητα	Αξία ανά τεμάχιο
$0 \leq Q < 500$	0.3
$500 \leq Q < 1000$	0.29
$1000 \leq Q$	0.28

Πίνακας 6.6: Κλιμακωτή χρέωση προϊόντων.



Σχήμα 6.13: Κόστος αγοράς παραγγελίας συναρτήσει ποσότητας παραγγελίας [79].

Συνεπώς, η συνολική αξία της παραγγελίας C υπολογίζεται με βάση τη σχέση 6.6

$$C(Q) = \begin{cases} 0, 30Q, & 0 \leq Q < 500 \\ 150 + 0.29(Q - 500) = 5 + 0.29Q, & 500 \leq Q < 1000 \\ 295 + 0.28(Q - 1000) = 15 + 0.28Q, & 1000 \leq Q \end{cases} \quad (6.6)$$

Η μοναδιαία ποσότητα παραγγελίας βρίσκεται από τη σχέση 6.7

$$c(Q) = C(Q)/Q \quad (6.7)$$

Η συνάρτηση των ΜΕΔ είναι

$$\Delta = \lambda c(Q) + A \frac{\lambda}{Q} + \frac{c(Q)I}{2} \frac{(Q-d)^2}{Q} + \frac{C_d d^2}{2Q} \quad (6.8)$$

Με αντικατάσταση των δεδομένων του προβλήματος προκύπτει η συνάρτηση - στόχος του προβλήματος

$$\Delta = 600 \frac{C(Q)}{Q} + 8 \frac{600}{Q} + \frac{0.2C(Q)}{2Q} \frac{(Q-d)^2}{Q} + 2.5 \frac{d^2}{Q} \quad (6.9)$$

Οι μεταβλητές σχεδιασμού είναι δύο, η ποσότητα παραγγελίας (Q) και η ποσότητα που βρίσκεται σε καθυστέρηση όταν καταφθάνει η νέα παραγγελία (d). Οι μεταβλητές αυτές είναι συνεχείς, αφού πρόκειται για μη συσκευασμένο υγρό.

Η συνάρτηση περιορισμών του προβλήματος είναι

$$d - Q < 0 \quad (6.10)$$

αφού αν η ποσότητα σε καθυστερήσεις υπερβεί την ποσότητα παραγγελίας, θεωρείται ότι είναι αδύνατο να καλυφθεί εγκαίρως.

Τα όρια των μεταβλητών που χρησιμοποιήθηκαν είναι

$$0 \leq Q, d \leq 5000 \quad (6.11)$$

Το πρόβλημα αυτό, αλγεβρικά λύνεται με διαδοχική αντικατάσταση κάθε κλάδου της συνάρτησης συνολικής αξίας της παραγγελίας στη συνάρτηση ΜΕΔ, εξίσωση των παραγώγων της ΜΕΔ ως προς Q και d με το μηδέν, και επίλυση των τριών συστημάτων που προκύπτουν. Στη συνέχεια συγκρίνονται οι λύσεις που τα συστήματα αυτά δίνουν και επιλέγεται αυτή που ελαχιστοποιεί το συνολικό κόστος.

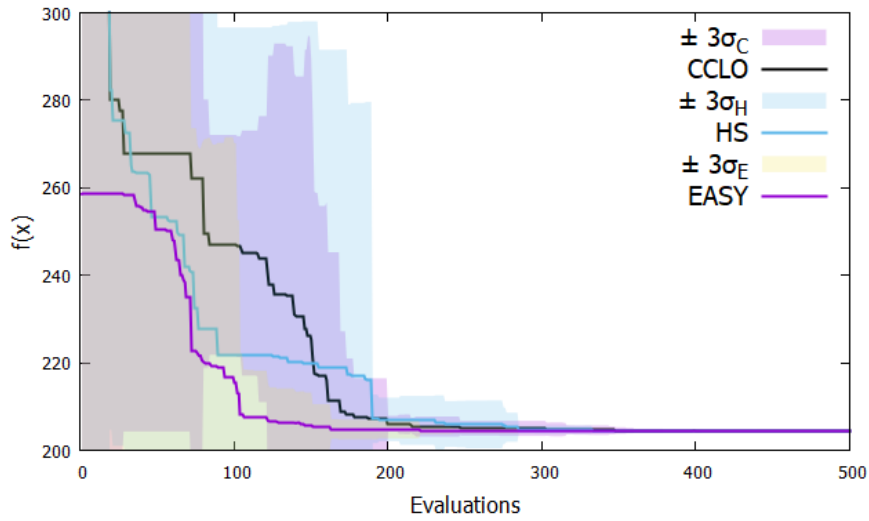
Η διαδικασία αυτή, δεδομένου ότι πρέπει να γίνει για κάθε προϊόν της επιχείρησης, αλλά ακόμα και για το ίδιο προϊόν τα κόστη συχνά αλλάζουν, είναι ιδιαίτερα χρονοβόρα. Για το λόγο αυτό, προτείνεται η λύση με αλγόριθμο βελτιστοποίησης.

Στον πίνακα 6.7 φαίνονται η μέση τιμή και η τυπική απόκλιση των κωδίκων CCLO και HS και του λογισμικού EASY, μετά από πενταχόσιες αξιολογήσεις. Στις 500 αξιολογήσεις που έγιναν, τα λογισμικά CCLO και EASY έχουν καταφέρει να εντοπίσουν την ελάχιστη τιμή για κάθε τρέξιμό τους.

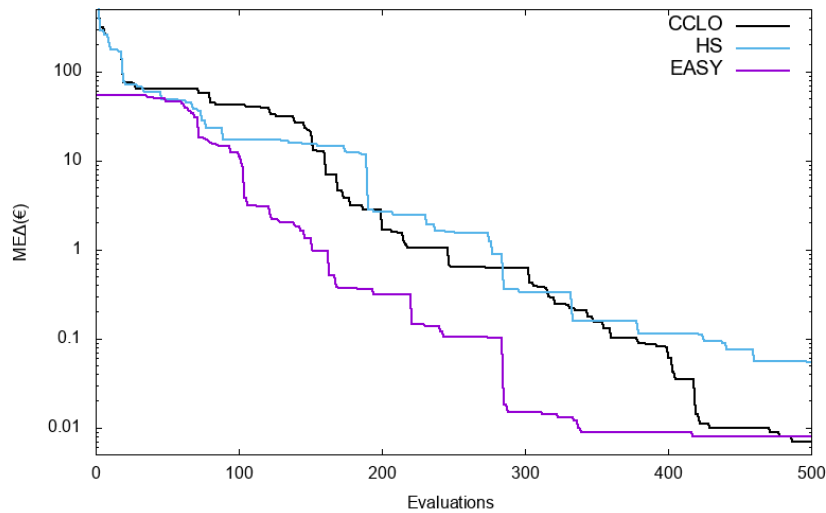
Software	μ	σ	Run1	Run2	Run3	Run4	Run5
HS	204.45	0.059	204.40	204.52	204.41	204.40	204.50
CCLO	204.40	0.002	204.40	204.40	204.40	204.40	204.40
EASY	204.40	0.004	204.40	204.40	204.40	204.40	204.40

Πίνακας 6.7: Μέση τιμή, τυπική απόκλιση και βέλτιστη λύση κάθε τρεξίματος των CCLO, HS και EASY.

Στο σχήμα 6.14 παρουσιάζεται η πορεία σύγκλισης της μέσης τιμής και της τυπικής απόκλισης των CCLO, HS και EASY. Στη συνέχεια, στο σχήμα 6.15, φαίνεται η πορεία σύγκλισης της μέσης τιμής των προς εξέταση λογισμικών σε λογαριθμική κλίμακα. Για καλύτερη αναπαράσταση στην περίπτωση αυτή, κάθε τιμή του y άξονα παρουσιάζεται μειωμένη κατά 204.39€.



Σχήμα 6.14: Πορεία σύγκλισης μέσης τιμής και τυπική απόκλιση μέσω των ετήσιων δαπανών για CCLO (μαύρη γραμμή), HS (μπλε γραμμή) και EASY (μοβ γραμμή).



Σχήμα 6.15: Πορεία σύγκλισης μέσης τιμής συνάρτησης μέσω των ετήσιων δαπανών σε λογαριθμική κλίμακα, μειωμένη κατά 204.39€, για CCLO (μαύρη γραμμή), HS (μπλε γραμμή) και EASY (μοβ γραμμή).

Για το λογισμικό EASY χρησιμοποιήθηκαν $(\mu, \lambda) = (20, 60)$. Για τους CCLO και HS, οι σταθερές φαίνονται στον πίνακα 6.7.

Τέλος, οι ποσότητες Q και d που προτείνεται να χρησιμοποιηθούν, όπως υπολογίστηκαν από κάθε κλήση του κώδικα CCLO φαίνονται στον πίνακα 6.9. Φαίνεται ότι για ένα εύρος της τάξης των 10 λίτρων παραγγελίας, οι μέσες ετήσιες δαπάνες παραμένουν σταθερά ελάχιστες.

Constants	CCLO	HS
CSS	20	20
MMR	0.8	0.8
LSR	0.5	0.5
bw	300	300
Off	25	1
Ev.Beg.	100	-
Pool	5	-

Πίνακας 6.8: Χρησιμοποιούμενες σταθερές στον *CCLO* και τον *HS*.

ΜΕΔ	Q	d
204.40	515.23	6.33
204.40	512.52	5.78
204.40	520.81	6.27
204.40	521.14	6.09
204.40	516.87	6.13

Πίνακας 6.9: ΜΕΔ και ποσότητες *Q* και *d* για κάθε κλήση του κώδικα *CCLO*.

Κεφάλαιο 7

Συμπεράσματα

Στη διπλωματική αυτή εργασία δημιουργήθηκε και προγραμματίστηκε μια νέα μέθοδος βελτιστοποίησης, η CCLO, η οποία βασίστηκε στη μέθοδο HS. Για τη δημιουργία της, αντλήθηκε επιπλέον λογισμικό ανεπτυγμένο από τη ΜΠΥΡ&Β/ΕΜΠ, το οποίο χρησιμοποιείται στον EASY. Στόχος ήταν η σύγκριση του νέου λογισμικού τόσο με την μέθοδο HS για διαπίστωση της βελτίωσης που επιτεύχθηκε, όσο με τους (μ,λ) ΕΑ, που χρησιμοποιεί ο EASY, για εξέταση πιθανής μελλοντικής ένταξής του σε αυτόν. Ο κώδικας γράφτηκε σε γλώσσα προγραμματισμού C++, σε αναλογία με τον κώδικα του λογισμικού EASY, για να διευκολυνθεί η πιθανή προσάρτηση σε αυτό.

Στον αλγόριθμο HS δοκιμάστηκαν τροποποιήσεις, και τελικά αποφασίστηκε να μεταβληθεί ο αριθμός των υποψήφιων λύσεων που δημιουργούνται σε κάθε αναζήτηση λύσεων που γίνεται, η μέθοδος διαχείρισης των μεταβλητών σχεδιασμού που υπερβαίνουν τα αποδεκτά τους όρια, η τιμή του εύρους (bw) μέσα στο οποίο γίνεται η μετάλλαξη, η μέθοδος δημιουργίας νέου ατόμου από αυτά που βρίσκονται στην ΗΜ, η μέθοδος διαχείρισης των συναρτήσεων - στόχων που ξεπερνούν τους περιορισμούς και επιπλέον να προστεθεί λογισμικό μεταπροτύπων για προσεγγιστική αξιολόγηση και να γίνουν οι απαραίτητοι χειρισμοί για να είναι δυνατή η επίλυση προβλημάτων πολυκριτηριακής βελτιστοποίησης. Ο τροποποιημένος αλγόριθμος ονομάστηκε CCLO.

Ο CCLO πράγματι επιτυγχάνει ικανοποιητικά αποτελέσματα, τόσο σε προβλήματα μονοκριτηριακής βελτιστοποίησης όσο και σε πολυκριτηριακής. Στις περιπτώσεις βελτιστοποίησης μίας συνάρτησης - στόχου, μετά την ολοκλήρωση των αξιολογήσεων που τέθηκαν ως κριτήριο τερματισμού, τα αποτελέσματα του CCLO είναι καλύτερα από αυτά του HS. Ωστόσο, κατά τις αρχικές αξιολογήσεις, ο αλγόριθμος CCLO, δεν έχει εξίσου καλά αποτελέσματα. Αυτό οφείλεται στο ότι οι αρχικές αξιολογήσεις που γίνονται στον CCLO, οι οποίες συνήθως επιλέγεται να είναι το 10-20% των συνολικών αξιολογήσεων, είναι καθαρά για εκπαίδευση των μεταπροτύπων, και η διαδικασία της βελτιστοποίησης ξεκινά μετά το τέλος τους. Στις περιπτώσεις βελτιστοποίησης πε-

ρισσότερων της μίας συναρτήσεων - στόχων, τα αποτελέσματα του νέου κώδικα είναι συγκρίσιμα με αυτά του λογισμικού EASY. Και στην πολυκριτηριακή βελτιστοποίηση, σε όλα τα προβλήματα που εξετάστηκαν τα αποτελέσματα του CCLO είναι καλύτερα από τα αντίστοιχα του HS.

Συνοψίζοντας, ο κώδικας που δημιουργήθηκε, δίνει πράγματι σε κάθε πρόβλημα που εξετάστηκε στη διπλωματική αυτή εργασία αποτελέσματα καλύτερα από αυτά του HS. Επιπλέον, δίνει αποτελέσματα συγκρίσιμα με αυτά του λογισμικού EASY, πρέπει να ληφθεί όμως υπόψη ότι ο τρόπος που εδώ χρησιμοποιείται ο EASY μπορεί να μην είναι ο βέλτιστος δυνατός, λόγω περιορισμένης σχετικής εμπειρίας.

Παράρτημα Α΄

Σταθερές CCLO

Για τον HS, οι τιμές των σταθερών που έχουν προταθεί στη βιβλιογραφία [80] για τις HMS, HMCR και PAR, οι οποίες, στον CCLO, αντιστοιχούν στις CSS, MMR και LSR είναι

$$\begin{aligned} 10 &\leq \text{HMS} \leq 50 \\ 0.7 &\leq \text{HMCR} \leq 0.95 \\ 0.2 &\leq \text{PAR} \leq 0.5 \end{aligned} \tag{A.1}$$

Για τη χρήση του λογισμικού CCLO, ο χρήστης πρέπει να ορίσει τις σταθερές που θα χρησιμοποιηθούν. Οι σταθερές αυτές περιλαμβάνουν τόσο τις σταθερές του CCLO, όσο και σταθερές σχετικές με το πλήθος των συναρτήσεων - στόχων, τον μέγιστο αριθμό αξιολογήσεων που θα εκτελεστούν, τα όρια και το πλήθος των μεταβλητών σχεδιασμού και των περιορισμών και τις σταθερές των μεταπροτύπων. Οι σταθερές αυτές ορίζονται σε ένα αρχείο, η μορφή του οποίου φαίνεται στο σχήμα Α΄.1. Στην αρχή του προγράμματος διαβάζεται το αρχείο αυτό.

Σε κάθε πρόβλημα που παρουσιάζεται έχει γίνει διερεύνηση, για την εύρεση των σταθερών που δίνουν τα καλύτερα αποτελέσματα. Διερεύνηση δεν έχει γίνει μόνο στον CCLO, αλλά και στους HS και EASY.

Σε κάθε πρόβλημα που παρουσιάζεται στη διπλωματική αυτή εργασία, συμπληρώνεται ένας πίνακας που περιλαμβάνει τις σταθερές που βρέθηκε ότι έδωσαν τα καλύτερα αποτελέσματα και τελικά χρησιμοποιήθηκαν. Στο σχήμα Α΄.1 φαίνεται ο πίνακας αυτός, και επεξηγούνται οι σταθερές κάθε γραμμής.

```

CCL0 Parameters
-----
Driver
-----
2          - Objectives to optimize
1000      - Maximum exact evaluations to perform
-----
CCL0 FEATURES
-----
20        - Choreography Set Size (CSS)
0.9      - Mimic Movement Rate (MMR)
0.6      - Legato - Staccato Rate (PAR)
3        - Bandwidth (bw)
25       - Population Of Each Search
150      - Elite individuals to keep (max archive size)
-----
PARAMETERIZATION
-----
#-----Design Variables (Format: lower_bound upper_bound comment)
2          - Total number of variables
-10 10  var1
-10 10  var2
#-----Constraints (format: Maximum_accepted_value Fail_value Intensity)
0          - Total number of constraints
-----
METAMODELS
-----
6          - Size of each pool
200       - Calculations for metamodels training
0         - CCL0 with metamodels(0) - CCL0 without metamodels (1)

```

Σχήμα Α'.1: Αρχείο ορισμού σταθερών προβλήματος.

Constants	Επεξήγηση
CSS	Choreography Set Size
MMR	Mimic Movement Rate
LSR	Legato-Staccato Rate
bw	Bandwidth
Off	Πληθυσμός κάθε νέας αναζήτησης λύσεων
Ev.Beg.	Αξιολογήσεις πριν ξεκινήσει η βελτιστοποίηση, για εκπαίδευση των μεταπροτύπων
Pool	Πλήθος ατόμων από τα οποία εκπαιδεύεται το μεταπρότυπο

Πίνακας Α'.1: Χρησιμοποιούμενες σταθερές στον CCL0.

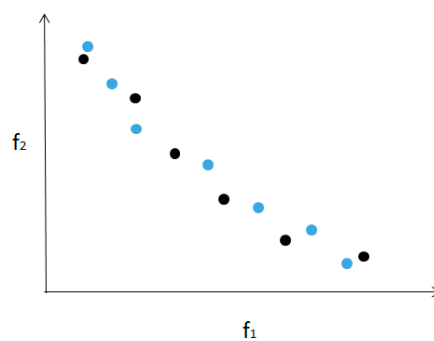
Παράρτημα Β΄

Κώδικας εύρεσης δείκτη υπερόγκου

Στα πλαίσια της εργασίας γράφτηκε επιπλέον ένας κώδικας για τον υπολογισμό του δείκτη υπερόγκου (hypervolume indicator), για σύγκριση αποτελεσμάτων σε προβλήματα περισσότερων της μίας συναρτήσεων - στόχων.

Β΄.1 Ανάγκη δημιουργίας κώδικα

Σε ένα πρόβλημα δύο συναρτήσεων - στόχων, κάθε κλήση του κώδικα δίνει τις διαδοχικές καμπύλες των μη-κυριαρχούμενων ατόμων σε κάθε γενιά του εξελικτικού αλγορίθμου. Στην περίπτωση αυτή υπάρχει η ανάγκη παρακολούθησης του αλγορίθμου και της ταχύτητας που αυτός συγκλίνει. Επιπλέον, υπάρχει η ανάγκη σύγκρισης διαφορετικών μετώπων Pareto, από κλήσεις του ίδιου κώδικα με διαφορετική γεννήτρια τυχαίων αριθμών. Στη σύγκριση αυτή σημεία διαφορετικής κλήσεις μπορεί να έχουν καλύτερα αποτελέσματα ανά περιοχή, κάνοντας τη σύγκριση αδύνατη, όπως φαίνεται στο σχήμα Β΄.1.



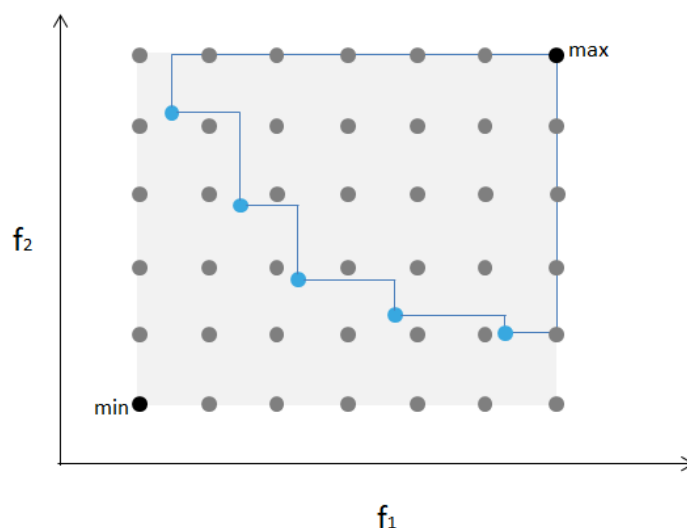
Σχήμα Β΄.1: Σύγκριση διαφορετικών μετώπων Pareto.

Προέκυψε λοιπόν η ανάγκη δημιουργίας ενός δείκτη ικανού να συγκρίνει άμεσα τις καμπύλες, στα προβλήματα δύο συναρτήσεων - στόχων, αλλά και των πολυδιάστατων σχημάτων, στην περίπτωση που υπάρχουν περισσότερες των δύο συναρτήσεις - στόχοι.

B'.2 Βήματα υπολογισμού δείκτη υπερόγκου

Για την περίπτωση των δύο διαστάσεων, η λογική που ακολουθήθηκε, όπως φαίνεται και στο σχήμα B'.2 ήταν η εξής:

1. Αρχικά ο χρήστης ορίζει έναν ορθογώνιο χώρο ελέγχου, όπως φαίνεται με γκρι χρώμα στο σχήμα B'.2, με δύο σημεία αναφοράς, το ελάχιστο (ζενιθ) και το μέγιστο (ναδιρ) στους δύο άξονες, όπως φαίνεται με μαύρο.
2. Μέσα στο χώρο ελέγχου δημιουργείται ένα πλέγμα, τα γκρι σημεία του σχήματος B'.2.
3. Για κάθε άτομο του πλέγματος που δημιουργήθηκε, εξετάζεται εάν υπάρχει άτομο του μετώπου που να έχει μικρότερη τιμή ταυτόχρονα τεταγμένης και τετμημένης. Τα άτομα που ικανοποιούν το κριτήριο αυτό είναι εντός ή στο όριο του μπλε πλαισίου στο σχήμα B'.2, και είναι 21 σε πλήθος.
4. Το σύνολο των τυχαίων ατόμων που πληρούν το βήμα 3, προς το σύνολο των ατόμων του πλέγματος που δημιουργήθηκε, αποτελούν το δείκτη υπερόγκου. Στο σχήμα B'.2 είναι $HVI = 21/42 = 0.5$



Σχήμα B'.2: Διαδικασία εύρεσης υπερόγκου.

Τα παραπάνω γενικεύονται για την περίπτωση που οι συναρτήσεις - στόχοι είναι περισσότεροι των δύο.

Βιβλιογραφία

- [1] Zitzler, Eckart and Thiele, Lothar: *Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach*. IEEE transactions on Evolutionary Computation, 3(4):257–271, 1999.
- [2] Giannakoglou, Kyriakos C, Giotis, Alexios P, and Karakasis, Marios K: *Low-cost genetic optimization based on inexact pre-evaluations and the sensitivity analysis of design parameters*. Inverse Problems in Engineering, 9(4):389–412, 2001.
- [3] Emmerich, Michael, Giotis, Alexios, Özdemir, Mutlu, Bäck, Thomas, and Giannakoglou, Kyriakos: *Metamodel—assisted evolution strategies*. In *International Conference on parallel problem solving from nature*, pages 361–370. Springer, 2002.
- [4] Karakasis, Marios K and Giannakoglou, Kyriakos C: *On the use of metamodel-assisted, multi-objective evolutionary algorithms*. Engineering Optimization, 38(8):941–957, 2006.
- [5] Kennedy, James and Eberhart, Russell: *Particle swarm optimization*. In *Proceedings of ICNN'95-International Conference on Neural Networks*, volume 4, pages 1942–1948. IEEE, 1995.
- [6] Dorigo, Marco, Colorni, Alberto, and Maniezzo, Vittorio: *Distributed optimization by ant colonies*, 1991.
- [7] Geem, Zong Woo, Kim, Joong Hoon, and Loganathan, Gobichettipalayam Vasudevan: *A new heuristic optimization algorithm: harmony search*. simulation, 76(2):60–68, 2001.
- [8] Abidin, Zulkiffi Zainal, Ngah, Umi Kalthum, Arshad, Mohd Rizal, and Ping, Ong Boon: *A novel fly optimization algorithm for swarming application*. In *2010 IEEE Conference on Robotics, Automation and Mechatronics*, pages 425–428. IEEE, 2010.
- [9] Pan, Wen Tsao: *A new fruit fly optimization algorithm: taking the financial distress model as an example*. Knowledge-Based Systems, 26:69–74, 2012.

- [10] Hedayatzadeh, Ramin, Salmassi, Foad Akhavan, Keshtgari, Manijeh, Akbari, Reza, and Ziarati, Koorush: *Termite colony optimization: A novel approach for optimizing continuous problems*. In *2010 18th Iranian conference on electrical engineering*, pages 553–558. IEEE, 2010.
- [11] Shah-Hosseini, Hamed: *The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm*. *International Journal of Bio-inspired computation*, 1(1-2):71–79, 2009.
- [12] Atashpaz-Gargari, Esmail and Lucas, Caro: *Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition*. In *2007 IEEE congress on evolutionary computation*, pages 4661–4667. Ieee, 2007.
- [13] Eusuff, Muzaffar, Lansey, Kevin, and Pasha, Fayzul: *Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization*. *Engineering optimization*, 38(2):129–154, 2006.
- [14] Karaboga, Dervis: *An idea based on honey bee swarm for numerical optimization*. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer . . . , 2005.
- [15] Nakrani, Sunil and Tovey, Craig: *On honey bees and dynamic server allocation in internet hosting centers*. *Adaptive Behavior*, 12(3-4):223–240, 2004.
- [16] Haddad, Omid Bozorg, Afshar, Abbas, and Marino, Miguel A: *Honey-bees mating optimization (hbmo) algorithm: a new heuristic approach for water resources optimization*. *water resources management*, 20(5):661–680, 2006.
- [17] Mozaffari, Ahmad, Fathi, Alireza, and Behzadipour, Saeed: *The great salmon run: a novel bio-inspired algorithm for artificial system design and optimisation*. *International Journal of Bio-Inspired Computation*, 4(5):286–301, 2012.
- [18] Chu, Shu Chuan, Tsai, Pei Wei, and Pan, Jeng Shyang: *Cat swarm optimization*. In *Pacific Rim international conference on artificial intelligence*, pages 854–858. Springer, 2006.
- [19] Li, XL and Qian, JX: *Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques*. *Journal of circuits and systems*, 1:1–6, 2003.
- [20] Rini, Dian Palupi, Shamsuddin, Siti Mariyam, and Yuhaniz, Siti Sophiyati: *Particle swarm optimization: technique, system and challenges*. *International journal of computer applications*, 14(1):19–26, 2011.
- [21] Hu, Xiaohui and Eberhart, Russell: *Solving constrained nonlinear optimization problems with particle swarm optimization*. In *Proceedings of the sixth world multiconference on systemics, cybernetics and informatics*, volume 5, pages 203–206. Citeseer, 2002.

- [22] Oliva, Diego, Elaziz, Mohamed Abd, Elsheikh, Ammar H, and Ewees, Ahmed A: *A review on meta-heuristics methods for estimating parameters of solar cells*. Journal of Power Sources, page 126683, 2019.
- [23] Zervogiannis, T, Assouti, V, Gagas, K, Kaounis, A, and Giannakoglou, KC: *Inverse design of aerodynamic shapes using ant colony optimization*. In *6th NATIONAL CONGRESS OF MECHANICS*, page 72, 2001.
- [24] Dorigo, Marco and Di Caro, Gianni: *Ant colony optimization: a new meta-heuristic*. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, volume 2, pages 1470–1477. IEEE, 1999.
- [25] Blum, Christian: *Ant colony optimization: Introduction and recent trends*. Physics of Life reviews, 2(4):353–373, 2005.
- [26] Dorigo, Marco and Blum, Christian: *Ant colony optimization theory: A survey*. Theoretical computer science, 344(2-3):243–278, 2005.
- [27] Dorigo, Marco, Birattari, Mauro, and Stutzle, Thomas: *Ant colony optimization*. IEEE computational intelligence magazine, 1(4):28–39, 2006.
- [28] Socha, Krzysztof and Dorigo, Marco: *Ant colony optimization for continuous domains*. European journal of operational research, 185(3):1155–1173, 2008.
- [29] Moon, YoungJu, Yu, HeonChang, Gil, Joon Min, and Lim, JongBeom: *A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments*. Human-centric Computing and Information Sciences, 7(1):28, 2017.
- [30] Grefenstette, John J: *Genetic algorithms for machine learning*, volume 165. Springer, 1994.
- [31] Back, Thomas, Hoffmeister, Frank, and Schwefel, Hans Paul: *A survey of evolution strategies*. In *Proceedings of the fourth international conference on genetic algorithms*, volume 2. Morgan Kaufmann Publishers San Mateo, CA, 1991.
- [32] Back, Thomas: *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.
- [33] Clark, David E and Westhead, David R: *Evolutionary algorithms in computer-aided molecular design*. Journal of Computer-Aided Molecular Design, 10(4):337–358, 1996.
- [34] Chipperfield, Andrew and Fleming, Peter: *Evolutionary algorithms for control engineering*. IFAC Proceedings Volumes, 29(1):1163–1168, 1996.
- [35] Deb, Kalyanmoy: *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.

- [36] Coello, Carlos A Coello, Lamont, Gary B, Van Veldhuizen, David A, *et al.*: *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.
- [37] *Genetic Algorithms*, webpage. <https://www.geeksforgeeks.org/genetic-algorithms/>.
- [38] Lee, Kang Seok and Geem, Zong Woo: *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice*. Computer methods in applied mechanics and engineering, 194(36-38):3902–3933, 2005.
- [39] Geem, Zong Woo: *Optimal cost design of water distribution networks using harmony search*. Engineering optimization, 38(03):259–277, 2006.
- [40] Zarei, O, Fesanghary, M, Farshi, B, Saffar, R Jalili, and Razfar, MR: *Optimization of multi-pass face-milling via harmony search algorithm*. Journal of materials processing technology, 209(5):2386–2392, 2009.
- [41] Ngonkham, Sukayapong and Buasri, Panhathai: *Harmony search algorithm to improve cost reduction in power generation system integrating large scale wind energy conversion system*. In *2009 World Non-Grid-Connected Wind Power and Energy Conference*, pages 1–5. IEEE, 2009.
- [42] Geem, Zong Woo: *Harmony search algorithm for solving sudoku*. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 371–378. Springer, 2007.
- [43] Wang, Chia Ming and Huang, Yin Fu: *Self-adaptive harmony search algorithm for optimization*. Expert Systems with Applications, 37(4):2826–2837, 2010.
- [44] Mahdavi, Mehrdad, Fesanghary, Mohammad, and Damangir, E: *An improved harmony search algorithm for solving optimization problems*. Applied mathematics and computation, 188(2):1567–1579, 2007.
- [45] Sivasubramani, Swarup and Swarup, KS: *Multi-objective harmony search algorithm for optimal power flow problem*. International Journal of Electrical Power & Energy Systems, 33(3):745–752, 2011.
- [46] Omran, Mahamed GH and Mahdavi, Mehrdad: *Global-best harmony search*. Applied mathematics and computation, 198(2):643–656, 2008.
- [47] Geem, Zong Woo: *Improved harmony search from ensemble of music players*. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 86–93. Springer, 2006.
- [48] Li, Hong qi and Li, Li: *A novel hybrid particle swarm optimization algorithm combined with harmony search for high dimensional optimization problems*. In *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*, pages 94–97. IEEE, 2007.

- [49] Fesanghary, M, Mahdavi, Mehrdad, Minary-Jolandan, M, and Alizadeh, Y: *Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems*. Computer methods in applied mechanics and engineering, 197(33-40):3080–3091, 2008.
- [50] Yıldız, Ali Rıza: *Hybrid taguchi-harmony search algorithm for solving engineering optimization problems*. International Journal of Industrial Engineering, 15(3):286–293, 2008.
- [51] Wang, Ling, Pan, Quan Ke, and Tasgetiren, M Fatih: *Minimizing the total flow time in a flow shop with blocking by using hybrid harmony search algorithms*. Expert Systems with Applications, 37(12):7929–7936, 2010.
- [52] Kaveh, A and Talatahari, S: *Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures*. Computers & Structures, 87(5-6):267–283, 2009.
- [53] Alatas, Bilal: *Chaotic harmony search algorithms*. Applied Mathematics and Computation, 216(9):2687–2699, 2010.
- [54] Wang, Xiaolei, Gao, Xiao Zhi, and Zenger, Kai: *An introduction to harmony search optimization method*. Springer, 2015.
- [55] Giannakoglou, KC: *Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence*. Progress in Aerospace Sciences, 38(1):43–76, 2002.
- [56] Karakasis, Marios K, Giotis, Alexios P, and Giannakoglou, Kyriakos C: *Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization*. International Journal for Numerical Methods in Fluids, 43(10-11):1149–1166, 2003.
- [57] Kampolis, Ioannis C, Papadimitriou, Dimitrios I, and Giannakoglou, Kyriakos C: *Evolutionary optimization using a new radial basis function network and the adjoint formulation*. Inverse Problems in Science and Engineering, 14(4):397–410, 2006.
- [58] Kapsoulis, Dimitrios, Tsiakas, Konstantinos, Trompoukis, Xenofon, Asouti, Varvara, and Giannakoglou, Kyriakos: *A pca-assisted hybrid algorithm combining eas and adjoint methods for CFD-based optimization*. Applied Soft Computing, 73:520–529, 2018.
- [59] *Evolutionary Algorithms SYstem, webpage*. <http://velos0.ltt.mech.ntua.gr/EASY/>.
- [60] Γιώτης, Α.: *Χρήση εξελικτικών τεχνικών, υπολογιστικής ευφυΐας και μεθόδων υπολογιστικής ρευστομηχανικής στη βελτιστοποίηση-αντίστροφη σχεδίαση περυγώσεων στροβιλομηχανών, μέσω παράλληλης επεξεργασίας*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2003.

- [61] Deb, Kalyanmoy, Agrawal, Ram Bhushan, *et al.*: *Simulated binary crossover for continuous search space*. *Complex systems*, 9(2):115–148, 1995.
- [62] Kapsoulis, D.: *LowCost MetamodelAssisted Evolutionary Algorithms with Application in Shape Optimization in Fluid Dynamics*. PhD thesis, Laboratory of Thermal Turbomachines, Parallel CFD Optimization Unit, NTUA., Athens, 2019.
- [63] Zitzler, Eckart, Laumanns, Marco, and Thiele, Lothar: *Spea2: Improving the strength pareto evolutionary algorithm*. TIK-report, 103, 2001.
- [64] Orr, Mark JL: *Recent advances in radial basis function networks*. Institute for Adaptive and Neural Computation, pages 25–29, 1999.
- [65] Powell, Michael JD: *Radial basis functions for multivariable interpolation: a review*. *Algorithms for approximation*, 1987.
- [66] Καμπόλης, Ι.: *Πολυεπίπεδοι, πολυεπεξεργαστικοί αλγόριθμοι αεροδυναμικής βελτιστοποίησης στις στροβιλομηχανές*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2009.
- [67] Li, Shaomin, Zhang, Deyuan, Shao, Zhenyu, and Tang, Hui: *Information feedback self-adaptive harmony search algorithm for the bovine cortical bone vibration-assisted drilling optimization*. *Measurement*, 149:107020, 2020.
- [68] *Virtual Library Of Simulation Experiments, webpage*. <https://www.sfu.ca/~ssurjano/optimization.html>.
- [69] Zhu, Guopu and Kwong, Sam: *Gbest-guided artificial bee colony algorithm for numerical function optimization*. *Applied mathematics and computation*, 217(7):3166–3173, 2010.
- [70] Panda, Arnapurna and Pani, Sabyasachi: *Multi-objective colliding bodies optimization*. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, pages 651–664. Springer, 2016.
- [71] *Virtual Library of Simulation Experiments, webpage*. <https://www.sfu.ca/~ssurjano/index.html>.
- [72] Τσόπελας, Η.: *Ένταξη του GPU Επιλύτη Ροής PUMA σε Ολοκληρωμένο Εργαλείο Ανάλυσης και Βελτιστοποίησης Στροβιλομηχανών της Βιομηχανίας. Δοκίμες και Πιστοποίηση*. Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2018.
- [73] Cárdenas-Barrón, Leopoldo Eduardo, Chung, Kun Jen, and Treviño-Garza, Gerardo: *Celebrating a century of the economic order quantity model in honor of ford whitman harris*, 2014.
- [74] Silver, Edward A, Pyke, David F, and Thomas, Douglas J: *Inventory and production management in supply chains*. CRC Press, 2016.

- [75] Agarwal, Sachin: *Economic order quantity model: a review*. VSRD International Journal of Mechanical, Civil, Automobile and Production Engineering, 4(12):233–236, 2014.
- [76] Πρωτοσύγγελος, Στέφανος: *Έλεγχος Αποθεμάτων*. Αθήνα, 2016.
- [77] Kumar, Rakesh: *Economic order quantity (EOQ) model*. Global Journal of finance and economic management, 5(1):1–5, 2016.
- [78] Senthilnathan, Samithamby: *Economic order quantity (EOQ)*. Available at SSRN 3475239, 2019.
- [79] Βλάχος, Δρ. Δημήτρης: *Σημειώσεις στη Διαχείριση Αποθεμάτων*. Θεσσαλονίκη, 2005.
- [80] Zhang, Tonghua and Geem, Zong Woo: *Review of harmony search with respect to algorithm structure*. Swarm and Evolutionary Computation, 48:31–43, 2019.