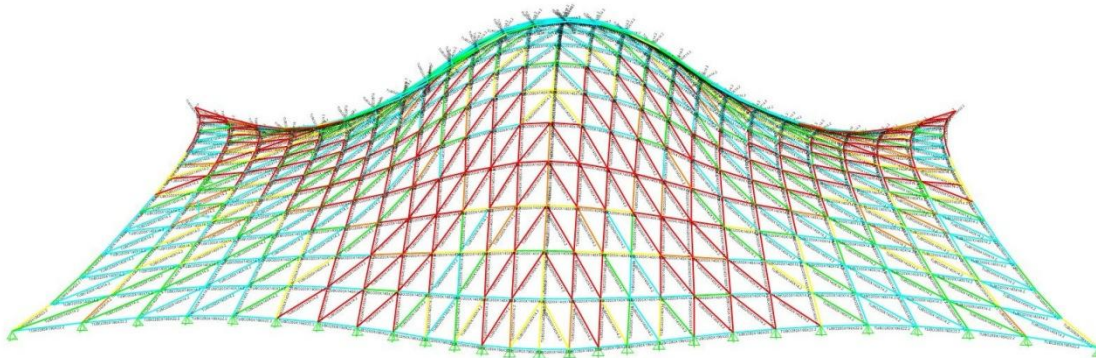




ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Σχολή Πολιτικών Μηχανικών
Εργαστήριο Στατικής & Αντισεισμικών Ερευνών

ΣΥΓΧΡΟΝΕΣ ΜΕΘΟΔΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΤΑΣΚΕΥΩΝ



Διπλωματική Εργασία
του
Σεραφείμ Κουρνούτου

Επιβλέπων: Αν. Καθηγητής Νικόλαος Λαγαρός

Αθήνα, Μάρτιος 2021

ΠΕΡΙΛΗΨΗ

Στις μέρες μας, ο σχεδιασμός ενός συστήματος που ικανοποιεί απλά και μόνο τις απαιτήσεις που σχετίζονται με την αντοχή και την ασφάλεια αυτού, δεν αρκεί πλέον, αφού κριτήρια όπως η ελαχιστοποίηση του κόστους ή γενικότερα των πόρων που απαιτούνται για την κατασκευή ή τη λειτουργία οποιουδήποτε συστήματος είναι εξίσου απαραίτητα. Έτσι, επιδιώκεται ο «βέλτιστος σχεδιασμός» ενός συστήματος δηλαδή ο σχεδιασμός εκείνος ο οποίος όχι μόνο ικανοποιεί τις απαιτήσεις λειτουργικότητας και αντοχής αλλά ικανοποιεί επίσης και κριτήρια όπως η ελαχιστοποίηση του κόστους, της ενέργειας ή του βάρους του συστήματος.

Σκοπός αυτής της εργασίας είναι η παρουσίαση των μεθόδων βελτιστοποίησης που έχουν αναπτυχθεί τα τελευταία χρόνια και αφορούν κυρίως σε μετα-ευρεστικούς (metaheuristic) αλγορίθμους, δηλαδή αλγοριθμικές διαδικασίες στοχαστικού χαρακτήρα εμπνευσμένες από τις βιολογικές και φυσικές διεργασίες, καθόσον η αυξημένη πολυπλοκότητα και το μέγεθος των σημερινών προβλημάτων έχει καταστήσει τις προηγούμενες μεθόδους αναποτελεσματικές. Λαμβάνοντας υπόψη ότι, τα περισσότερα προβλήματα περιέχουν μεταβλητές οι οποίες υπόκεινται σε πληθώρα περιορισμών τόσο απλών όσο και σύνθετων, γίνεται μια εκτεταμένη ανάλυση στις διάφορες τεχνικές διαχείρισης περιορισμών (constraint handling techniques) που ενσωματώνουν οι αλγόριθμοι αυτοί καθόσον ο κατάλληλος συνδυασμός αλγορίθμου και τεχνικής διαχείρισης περιορισμών εγγυάται ότι θα οδηγήσει την αναζήτηση στην καθολική βέλτιστη λύση.

Αρκετά παραδείγματα που αφορούν τόσο σε μεταλλικές κατασκευές όσο και σε κατασκευές από οπλισμένο σκυρόδεμα επιλύονται με διαφορετικούς αλγορίθμους και τα αποτελέσματά τους παρουσιάζονται και αναλύονται προκειμένου να αναδειχθεί αφενός η σημαντικότητα και αφετέρου η περιπλοκότητα και η ποικιλομορφία ενός «βέλτιστου σχεδιασμού». Συμπεραίνεται ότι δεν υπάρχει μία μοναδική μέθοδος που να βελτιστοποιεί όλα τα είδη των προβλημάτων. Άλλωστε, εάν ένας αλγόριθμος έχει καλή απόδοση σε μια συγκεκριμένη κατηγορία προβλημάτων, τότε εμφανίζει απαραίτητα χαμηλότερη απόδοση στο σύνολο όλων των υπόλοιπων προβλημάτων (no free lunch theorem).

ABSTRACT

Nowadays, designing a system that simply meets the requirements related to its durability and safety is no longer enough, since criteria such as minimizing the cost or in general the resources required for the construction or operation of any system are just as necessary. Thus, the "optimal design" of a system is sought, i.e. the design that not only meets the requirements of functionality and durability but also meets criteria such as minimizing the cost, energy or weight of the system.

The purpose of this diploma thesis is to present the optimization methods that have been developed in recent years and concern mainly on metaheuristic algorithms, i.e. stochastic algorithmic procedures inspired by biological and natural processes, as the increasing complexity and size of current problems has made the previous methods ineffective. Given that most problems contain variables that are subject to a variety of constraints both simple and complex, an extensive analysis of various constraint handling techniques incorporated by these algorithms is made, as the appropriate combination of an algorithm and a constraint handling technique will lead the search to the global optimal solution.

Several examples concerning both metal and reinforced concrete structures are solved with different algorithms and their results are presented and analyzed in order to highlight on the one hand the importance and on the other hand the complexity and diversity of an "optimal design". It is concluded that there is no single method that optimizes all kinds of problems. After all, if an algorithm performs well in a particular category of problems, then it necessarily performs lower in all other problems according to the "no free lunch theorem".

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1	ΕΙΣΑΓΩΓΗ	1
2	ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΤΑΣΚΕΥΩΝ	3
3	ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ	4
4	ΤΕΧΝΙΚΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΟΡΙΣΜΩΝ	6
4.1	Γενικά.....	6
4.2	Βιβλιογραφική επισκόπηση τεχνικών διαχείρισης περιορισμών.....	8
4.2.1	Μέθοδοι βασισμένες σε συναρτήσεις ποινής.....	9
4.2.2	Διαχωρισμός αντικειμενικής συνάρτησης και περιορισμών.....	11
4.2.3	Συνδυασμός μεθόδων διαχείρισης περιορισμών.....	15
4.2.4	Μέθοδοι βασισμένες σε αλγόριθμους επισκευής.....	17
4.2.5	Μέθοδοι βασισμένες στα όρια των μεταβλητών.....	18
4.2.6	Άλλες μέθοδοι.....	20
4.3	Ανάλυση δημοφιλέστερων τεχνικών διαχείρισης περιορισμών.....	21
4.3.1	Μέθοδοι συναρτήσεων ποινής.....	21
4.3.1.1	Προσαρμοζόμενη μέθοδος συναρτήσεων ποινής.....	21
4.3.2	Μέθοδος κανόνων εφικτότητας.....	22
4.3.2.1	Τροποποιημένη μέθοδος κανόνων εφικτότητας.....	23
4.3.3	Μέθοδος ε-περιορισμού.....	25
4.3.3.1	Βελτιωμένη μέθοδος ε-περιορισμού.....	25
4.3.3.2	Τροποποιημένη μέθοδος ε-περιορισμού.....	26
4.3.4	Συνονθύλευμα τεχνικών διαχείρισης περιορισμών.....	27
4.3.5	Μέθοδος στοχαστικής κατάταξης.....	28
5	ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ	29
5.1	Γενικά.....	29
5.2	Τύποι μοντέλων μελέτης.....	29
5.3	Αλγόριθμοι βελτιστοποίησης μελέτης.....	32
5.3.1	Αλγόριθμος PQN.....	32
5.3.2	Αλγόριθμος MATSuMoTo.....	36
5.3.3	Αλγόριθμος Latin Hypercube.....	40
5.3.4	Αλγόριθμος Differential Evolution.....	42
5.3.5	Αλγόριθμος PBA.....	43

6	ΑΠΟΤΕΛΕΣΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ	46
6.1	Ανάλυση 1 ^ο Μοντέλου.....	47
6.1.1	Εφαρμογή του αλγορίθμου PQN.....	48
6.1.2	Εφαρμογή του αλγορίθμου MATSuMoTo	49
6.1.3	Εφαρμογή του αλγορίθμου Latin Hypercube.....	51
6.1.4	Συγκριτική ανάλυση αποτελεσμάτων	52
6.2	Ανάλυση 2 ^ο Μοντέλου.....	54
6.2.1	Εφαρμογή του αλγορίθμου PQN.....	54
6.2.2	Εφαρμογή του αλγορίθμου MATSuMoTo	56
6.2.3	Εφαρμογή του αλγορίθμου Latin Hypercube.....	58
6.2.4	Συγκριτική ανάλυση αποτελεσμάτων	59
6.3	Ανάλυση 3 ^ο Μοντέλου.....	61
6.3.1	Εφαρμογή του αλγορίθμου PQN.....	61
6.3.2	Εφαρμογή του αλγορίθμου MATSuMoTo	63
6.3.3	Εφαρμογή του αλγορίθμου Latin Hypercube.....	64
6.3.4	Συγκριτική ανάλυση αποτελεσμάτων	65
6.4	Ανάλυση 4 ^ο Μοντέλου.....	67
6.4.1	Εφαρμογή του αλγορίθμου PQN.....	67
6.4.2	Εφαρμογή του αλγορίθμου MATSuMoTo	69
6.4.3	Εφαρμογή του αλγορίθμου Latin Hypercube.....	70
6.4.4	Συγκριτική ανάλυση αποτελεσμάτων	71
6.5	Ανάλυση 5 ^ο Μοντέλου.....	73
6.5.1	Εφαρμογή του αλγορίθμου PQN.....	73
6.5.2	Εφαρμογή του αλγορίθμου MATSuMoTo	75
6.5.3	Εφαρμογή του αλγορίθμου Latin Hypercube.....	76
6.5.4	Συγκριτική ανάλυση αποτελεσμάτων	78
6.6	Ανάλυση 6 ^ο Μοντέλου.....	79
6.6.1	Εφαρμογή του αλγορίθμου PQN.....	80
6.6.2	Εφαρμογή του αλγορίθμου MATSuMoTo	81
6.6.3	Εφαρμογή του αλγορίθμου Latin Hypercube.....	83
6.6.4	Συγκριτική ανάλυση αποτελεσμάτων	84
7	ΣΥΜΠΕΡΑΣΜΑΤΑ	86
	ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ	88

1 ΕΙΣΑΓΩΓΗ

Ο όρος «βελτιστοποίηση» στα εφαρμοσμένα μαθηματικά αναφέρεται στην αναζήτηση βέλτιστων παραμέτρων ενός συστήματος και μπορεί να οριστεί ως η διαδικασία εξεύρεσης των συνθηκών εκείνων που δίνουν το μέγιστο ή το ελάχιστο μίας συνάρτησης. Η ύπαρξη μεθόδων βελτιστοποίησης εντοπίζεται στις ημέρες των Newton, Lagrange και Cauchy.

Αντίστοιχα με τα μαθηματικά, έτσι και στη μηχανική, ο όρος «βέλτιστος σχεδιασμός κατασκευών» μπορεί να ερμηνευτεί με πολλούς τρόπους. Ο όρος «κατασκευή» χρησιμοποιείται για να περιγράψει τη διάταξη των στοιχείων ή / και των υλικών προκειμένου να δημιουργηθεί ένα σύστημα ικανό να αναλάβει τα φορτία που επιβάλλονται από τις απαιτήσεις σχεδιασμού. Ωστόσο, η πρόοδος της τεχνολογίας των υπολογιστών δημιούργησε περισσότερες απαιτήσεις και πλέον, ο σχεδιασμός ενός δομικού συστήματος που ικανοποιεί απλά και μόνο τις απαιτήσεις σχεδιασμού που σχετίζονται με την αντοχή και την ασφάλεια της κατασκευής, δεν αρκεί πια. Ο σωστός σχεδιασμός ενός δομικού συστήματος είναι ζωτικής σημασίας και μονόδρομος για την επίτευξη του επιθυμητού αποτελέσματος και περιλαμβάνει και άλλες παραμέτρους πέραν της αντοχής μιας κατασκευής. Έτσι, για το σωστό σχεδιασμό των κατασκευών εφαρμόζεται μια επαναληπτική διαδικασία με τελικό στόχο την επίτευξη του «βέλτιστου σχεδιασμού». Ο όρος βέλτιστος σχεδιασμός χρησιμοποιείται για τον σχεδιασμό εκείνον ο οποίος δεν ικανοποιεί μόνο τις απαιτήσεις λειτουργικότητας ή αντοχής αλλά ικανοποιεί επίσης και κριτήρια όπως η ελαχιστοποίηση του κόστους ή του βάρους του συστήματος.

Στόχος του μηχανικού είναι να βρει, μέσω μιας αλγοριθμικής διαδικασίας, έναν συνδυασμό ανεξάρτητων μεταβλητών με πραγματικές ή ακέραιες τιμές που ονομάζονται παράμετροι ή μεταβλητές σχεδιασμού, έτσι ώστε να βελτιστοποιηθεί η αντικειμενική συνάρτηση του προβλήματος σχεδιασμού. Έτσι, για την επίτευξη του βέλτιστου σχεδιασμού είναι απαραίτητο να εκτελεστούν δύο βασικά βήματα: η μαθηματική διατύπωση του προβλήματος βελτιστοποίησης και η εφαρμογή ενός αλγορίθμου. Η ύπαρξη αποτελεσματικών αλγορίθμων βελτιστοποίησης εγγυάται ότι το πρόβλημα του βέλτιστου σχεδιασμού θα αντιμετωπιστεί επιτυχώς. Η εμπειρία του μηχανικού είναι σημαντική παράμετρος για τη σωστή χρήση αυτών των αλγορίθμων. Η διαδικασία σχεδιασμού είναι μια επαναληπτική διαδικασία όπου η επανάληψη θεωρείται ως η διαδοχική δοκιμή υποψήφιων λύσεων και αξιολογεί αν κάθε υποψήφια λύση είναι καλύτερη ή όχι σε σύγκριση με τα προηγούμενα, ενώ ικανοποιεί τους περιορισμούς του προβλήματος. Η συμβατική διαδικασία που χρησιμοποιείται από τους μηχανικούς είναι αυτή της «δοκιμής και διόρθωσης». Φυσικά, με την αύξηση της πολυπλοκότητας και του μεγέθους των προβλημάτων, η χρήση τέτοιων εμπειρικών τεχνικών δεν οδηγεί στη βέλτιστη λύση. Έτσι κατέστη αναγκαία η αυτοματοποίηση της διαδικασίας του σχεδιασμού αξιοποιώντας τις

εξελίξεις στην τεχνολογία των υπολογιστών και την πρόοδο των αλγορίθμων βελτιστοποίησης.

Η υφιστάμενη βιβλιογραφία βρίθεται από έρευνες που σχετίζονται με την ανάπτυξη και εξέλιξη αλγοριθμικών διαδικασιών ικανών να επιλύσουν μια μεγάλη γκάμα προβλημάτων. Από τις πρώτες και απλούστερες μεθόδους που στηρίζονταν στη χρήση απλών παραγωγίσιμων συναρτήσεων για την επίλυση προβλημάτων βελτιστοποίησης, πλέον αναπτύσσονται πιο σύγχρονες μέθοδοι εμπνευσμένες από τις φυσικές και βιολογικές διεργασίες όπως για παράδειγμα η συμπεριφορά των εντόμων ή άλλων συναφών ειδών, η αναπαραγωγική διαδικασία, η δομή και λειτουργία των χρωμοσωμάτων και πολλές άλλες. Η πλειονότητα αυτών των αλγορίθμων, όμως, είναι σχεδιασμένη για προβλήματα με συνεχείς χώρους αναζήτησης στα οποία το πεδίο εξερεύνησης δεν περιορίζεται από κανένα όριο. Στην πραγματικότητα, ωστόσο, κάθε πρόβλημα υπόκειται σε πληθώρα περιορισμών, είτε απλών είτε σύνθετων, οι οποίοι μειώνουν το διαθέσιμο πεδίο εξερεύνησης και εκμετάλλευσης των αλγορίθμων και κατ' επέκταση και το πεδίο που μπορούν να κινηθούν οι υποψήφιες λύσεις κάθε προβλήματος. Ως εκ τούτου, πέραν από έναν αποτελεσματικό αλγόριθμο εξίσου σημαντική είναι και η ενσωμάτωση σε αυτόν μιας κατάλληλης τεχνικής διαχείρισης περιορισμών, ο συνδυασμός των οποίων θα οδηγήσει στην καθολική βέλτιστη λύση ενός προβλήματος.

Το υπόλοιπο της παρούσας εργασίας είναι δομημένο ως εξής: Στο Κεφάλαιο 2 περιγράφονται οι βασικοί τύποι προβλημάτων βελτιστοποίησης κατασκευών και στο Κεφάλαιο 3 οι μέθοδοι / αλγόριθμοι βελτιστοποίησης που έχουν αναπτυχθεί τα τελευταία χρόνια. Στο Κεφάλαιο 4 γίνεται μια βιβλιογραφική επισκόπηση των υφιστάμενων μεθόδων διαχείρισης περιορισμών καθώς και μια εκτεταμένη ανάλυση των δημοφιλέστερων εξ αυτών. Αρκετά παραδείγματα που αφορούν τόσο σε μεταλλικές κατασκευές όσο και σε κατασκευές από οπλισμένο σκυρόδεμα καθώς και οι αλγόριθμοι βελτιστοποίησης που χρησιμοποιήθηκαν για την επίλυσή τους παρουσιάζονται στο Κεφάλαιο 5 ενώ τα αποτελέσματά τους αναλύονται στο Κεφάλαιο 6. Τέλος, στο Κεφάλαιο 7 παρουσιάζονται τα συμπεράσματα από την ανάλυση των αποτελεσμάτων.

2 ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΤΑΣΚΕΥΩΝ

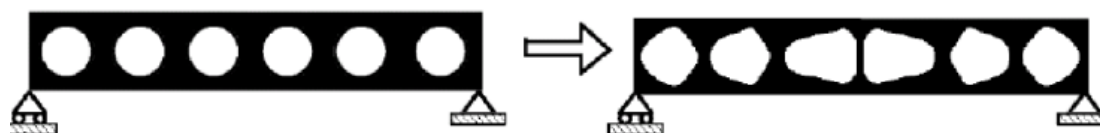
Υπάρχουν τρεις βασικές κατηγορίες προβλημάτων βελτιστοποίησης κατασκευών: μεγέθους (sizing), σχήματος (shape) και τοπολογίας (topology). Στα προβλήματα βελτιστοποίησης μεγέθους ο στόχος είναι συνήθως η ελαχιστοποίηση του βάρους της κατασκευής υπό ορισμένους περιορισμούς στη συμπεριφορά (τάσεις και μετατοπίσεις). Κατά αντιστοιχία, στα προβλήματα βελτιστοποίησης σχήματος ο στόχος είναι να βελτιωθεί η αποτελεσματικότητα της κατασκευής τροποποιώντας τα όριά της.

Αρχικά η βελτιστοποίηση κατασκευών επικεντρώθηκε στη βελτιστοποίηση του μεγέθους, όπως η βελτιστοποίηση της διατομής των δοκών και πλαισίων ή το πάχος των πλακών και των κελύφων (Σχ.1). Το επόμενο βήμα ήταν να μελετηθεί η δυνατότητα να βρεθούν τα βέλτιστα όρια μιας κατασκευής και επομένως να βελτιστοποιηθεί το σχήμα της (Σχ.2). Στην πρώτη περίπτωση η κατασκευή έχει σταθερή μορφή, ενώ στην τελευταία περίπτωση δεν είναι σταθερή αλλά έχει μια προκαθορισμένη τοπολογία.

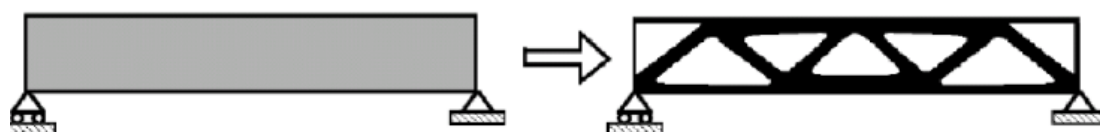
Τελευταία εξέλιξη είναι η βελτιστοποίηση της τοπολογίας, η οποία επιτρέπει στον σχεδιαστή να βελτιστοποιήσει τη διάταξη ή την τοπολογία μιας κατασκευής ανιχνεύοντας και αφαιρώντας το υλικό που παραλαμβάνει μικρή φόρτιση και που στην ουσία δεν χρησιμοποιείται αποτελεσματικά (Σχ.3). Μπορεί να θεωρηθεί ως μια διαδικασία βελτιστοποίησης της ορθολογικής κατανομής του διαθέσιμου υλικού και την εξάλειψη αυτού που δεν χρειάζεται.



Σχ.1 Βελτιστοποίηση μεγέθους (sizing) (Πηγή: <https://www.researchgate.net>)



Σχ.2 Βελτιστοποίηση σχήματος (shape) (Πηγή: <https://www.researchgate.net>)



Σχ.3 Βελτιστοποίηση τοπολογίας (topology) (Πηγή: <https://www.researchgate.net>)

3 ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Τα τελευταία χρόνια έχουν αναπτυχθεί αρκετές μέθοδοι βελτιστοποίησης των κατασκευών. Η κατηγοριοποίηση αυτών ποικίλλει μέσα στην υφιστάμενη βιβλιογραφία, ανάλογα με τον τύπο και τα χαρακτηριστικά των αλγορίθμων που χρησιμοποιούν. Οι μέθοδοι βελτιστοποίησης διακρίνονται γενικά σε:

- **Αιτιοκρατικού ή Προσδιοριστικού ή Ντετερμινιστικού** χαρακτήρα, δηλαδή στηρίζονται σε αλγόριθμους που κινούνται προς συνεχώς καλύτερες λύσεις με αυστηρά καθορισμένο τρόπο αξιοποιώντας πληροφορία από την «παράγωγο» της αντικειμενικής συνάρτησης (gradient-based).

- **Πιθανοκρατικού ή Τυχηματικού ή Στοχαστικού** χαρακτήρα, οι οποίες στηρίζονται στην τεχνική της δημιουργίας και ελέγχου μέσω της τυχαίας αναζήτησης (random search) και δεν εξαρτώνται από την παράγωγο της αντικειμενικής συνάρτησης (derivative-free). Αξιολογούν ένα σύνολο τυχαίων λύσεων του προβλήματος και επιλέγουν την βέλτιστη από αυτές. Δηλαδή δημιουργείται μια τυχαία υποψήφια λύση η οποία ελέγχεται και αν δεν ικανοποιεί κάποια κριτήρια, τότε δημιουργείται νέα υποψήφια λύση. Σε άλλες περιπτώσεις, αντί της μίας τυχαίας υποψήφιας λύσης χρησιμοποιούν έναν «πληθυσμό» (population) υποψήφιας λύσεων του προβλήματος. Οι λύσεις αυτές είναι ανεξάρτητες μεταξύ τους ενώ ο αρχικός πληθυσμός επιλέγεται με τυχαίο τρόπο και στη συνέχεια, ο πληθυσμός αλλάζει κινούμενος σε όλο και καλύτερες περιοχές του χώρου αναζήτησης.

Έτσι, για την επίλυση των προβλημάτων βελτιστοποίησης οι διάφορες μέθοδοι / αλγόριθμοι που έχουν αναπτυχθεί και εξελιχτεί, μπορούν να κατηγοριοποιηθούν σε:

- **Μαθηματικές μεθόδους**, οι οποίες στηρίζονται στις αρχές του μαθηματικού προγραμματισμού, είναι αιτιοκρατικού χαρακτήρα και μπορούν να ταξινομηθούν σε πέντε μεγάλες κατηγορίες:

Γραμμικός Προγραμματισμός (Linear Programming – LP)

Μη Γραμμικός Προγραμματισμός (Non Linear Programming – NLP)

Ακέραιος Προγραμματισμός (Integer Programming – IP)

Δυναμικός προγραμματισμός (Dynamic Programming – DP)

Γεωμετρικός προγραμματισμός (Geometric Programming – GP).

- Τις **ευρεστικές (heuristic) μεθόδους**, οι οποίες είναι εμπειρικές – προσεγγιστικές μέθοδοι τυχηματικού χαρακτήρα και είναι ικανές να εντοπίζουν

ικανοποιητικές λύσεις, όχι όμως απαραίτητα βέλτιστες, σε εύλογο χρόνο κυρίως σε προβλήματα μεγάλου μεγέθους ή περίπλοκης δομής. Οι αλγόριθμοι αυτοί πρακτικά σχεδιάζονται ανάλογα με το εκάστοτε πρόβλημα και ως εκ τούτου δεν υπάρχει γενική μορφή.

Στον τομέα της Επιστήμης Υπολογιστών, της τεχνητής νοημοσύνης, και στην Μαθηματική βελτιστοποίηση, ευρεστική καλείται η τεχνική που μπορεί να δώσει την λύση ενός προβλήματος – χωρίς να είναι απαραίτητα η βέλτιστη – αρκετά γρήγορα όταν οι κλασικές μέθοδοι είναι πολύ αργές, ή για την εύρεση λύσης, κατά προσέγγιση, όταν οι κλασικές μέθοδοι αποτύχουν να βρουν οποιαδήποτε ακριβή λύση. Αυτό επιτυγχάνεται βάζοντας ως προτεραιότητα την ταχύτητα σε βάρος της βελτιστοποίησης, της πληρότητας και της ακρίβειας της μεθόδου. Παρόλο που η λύση δεν είναι πάντοτε η βέλτιστη εξακολουθεί να είναι σημαντική, επειδή η εύρεσή της δεν απαιτεί απαγορευτικά μεγάλο χρόνο εκτέλεσης. Χαρακτηριστικό παράδειγμα ευρεστικού αλγορίθμου είναι η «δοκιμή και διόρθωση» (trial and error).

- Τους **μετα-ευρεστικούς (metaheuristic) αλγορίθμους**, οι οποίοι είναι επίσης τυχαματικού χαρακτήρα και εμπνευσμένοι από τις βιολογικές και φυσικές διεργασίες. Η αυξημένη πολυπλοκότητα και το μέγεθος κάποιων προβλημάτων έχει καταστήσει τις προηγούμενες μεθόδους αναποτελεσματικές και επιτακτική την ανάγκη ανάπτυξης εξελιγμένων αλγορίθμων βελτιστοποίησης.

Σύμφωνα με τους Glover και Sörensen, μετα-ευρεστική μέθοδος είναι ένα υψηλού επιπέδου αλγοριθμικό πλαίσιο που παρέχει ένα πακέτο κατευθυντήριων γραμμών και στρατηγικών με στόχο να αναπτύξει έναν ευρεστικό αλγόριθμο βελτιστοποίησης. Ο όρος χρησιμοποιείται επίσης για να αναφερθεί σε μια εφαρμογή συγκεκριμένου προβλήματος ενός ευρεστικού αλγορίθμου βελτιστοποίησης σύμφωνα με τις κατευθυντήριες γραμμές που εκφράζονται σε ένα τέτοιο πλαίσιο. Οι αλγόριθμοι αυτοί σε αντίθεση με τις μεθόδους αιτιοκρατικού χαρακτήρα, χρησιμοποιούν έναν «πληθυσμό» (population) υποψήφιων λύσεων και δεν εγκλωβίζονται, έτσι, σε τοπικά βέλτιστα με αποτέλεσμα να αυξάνονται οι πιθανότητες εύρεσης καθολικών βέλτιστων λύσεων. Μερικοί από τους σημαντικότερους **μετα-ευρεστικούς αλγορίθμους** είναι οι κάτωθι:

- Μέθοδος Στρατηγικών Εξέλιξης (Evolutionary Strategies – ES)
- Γενετικοί Αλγόριθμοι (Genetic Algorithms – GAs)
- Σμήνος Σωματιδίων (Particle Swarm Optimization – PSO)
- Αποικία Μυρμηγκιών (Ant Colony Optimization – ACO)
- Διαφορική Εξέλιξη (Differential Evolution – DE)

4 ΤΕΧΝΙΚΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΟΡΙΣΜΩΝ

4.1 Γενικά

Τα περισσότερα προβλήματα στις μέρες μας περιέχουν μεταβλητές οι οποίες υπόκεινται σε πληθώρα περιορισμών τόσο απλών (όρια μεταβλητών) όσο και σύνθετων (συναρτήσεις περιορισμών). Τέτοια προβλήματα ονομάζονται προβλήματα βελτιστοποίησης με περιορισμούς (constrained optimization problems – COPs) ή μη-γραμμικού προγραμματισμού (Non Linear Programming – NLP). Σύμφωνα με τον Deb (2000), η γενική μαθηματική διατύπωση τέτοιων προβλημάτων μπορεί να εκφραστεί ως ακολούθως:

Minimize $f(\mathbf{x})$,

Subject to

$$g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, J$$

$$h_k(\mathbf{x}) = 0, k = 1, 2, \dots, K$$

$$x_i^l \leq x_i \leq x_i^u, i = 1, 2, \dots, n$$

όπου $f(\mathbf{x})$ είναι η αντικειμενική συνάρτηση, $g_j(\mathbf{x})$ είναι ο j^{th} περιορισμός ανισότητας, $h_k(\mathbf{x})$ είναι ο k^{th} περιορισμός ισότητας και x_i^l, x_i^u το κάτω και άνω όριο του i^{th} στοιχείου του διανύσματος μεταβλητών \mathbf{x} μεγέθους n . Οι περιορισμοί ισότητας μπορούν να μετατραπούν εύκολα σε περιορισμούς ανισότητας σύμφωνα με την εξίσωση:

$$g_k(\mathbf{x}) = |h_k(\mathbf{x})| - \sigma \leq 0, k = 1, 2, \dots, K,$$

όπου το σ λαμβάνει πολύ μικρές θετικές τιμές. Έτσι, ο συνολικός αριθμός των περιορισμών ανισότητας γίνεται το άθροισμα $m = J + K$ και στην μαθηματική διατύπωση του προβλήματος, ο όρος J αντικαθίσταται από το m οπότε τελικά ισχύει $g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, m$.

Η πλειοψηφία των αλγορίθμων που έχουν αναπτυχθεί μέχρι σήμερα, όμως, έχει σχεδιαστεί μόνο για προβλήματα βελτιστοποίησης χωρίς περιορισμούς (unconstrained optimization problems). Έτσι, η ανάγκη να ξεπεραστεί το εμπόδιο της αποτελεσματικής διαχείρισης των περιορισμών (constraints handling) κατά τις αλγοριθμικές διαδικασίες καθώς και το γεγονός ότι δεν υπάρχει μια μοναδική μέθοδος διαχείρισης περιορισμών για να οδηγήσει την αναζήτηση κάθε προβλήματος στο καθολικό βέλτιστο, έχει μετατρέψει τη διαχείριση περιορισμών σε αντικείμενο μελέτης για πολλούς ερευνητές.

Ο Coello (2002) διεξήγαγε μια ολοκληρωμένη έρευνα για τις πιο δημοφιλείς τεχνικές διαχείρισης περιορισμών όπως συναρτήσεις ποινής (penalty functions),

ειδικές αναπαραστάσεις και τελεστές (operators), αλγόριθμοι επισκευής (repair algorithms), διαχωρισμός αντικειμενικής συνάρτησης και περιορισμών και υβριδικές μέθοδοι που χρησιμοποιούνται παράλληλα με εξελικτικούς αλγόριθμους. Εκτός από την περιγραφή αυτών των τεχνικών, ο συγγραφέας παρουσιάζει επίσης τα πλεονεκτήματα και τα μειονεκτήματα της καθεμίας και ορισμένα συγκριτικά πειραματικά αποτελέσματα μεταξύ πολλών τεχνικών βασισμένων σε ποινές που εφαρμόστηκαν σε ένα γενετικό αλγόριθμο για την επίλυση τριών μαθηματικών συναρτήσεων.

Στην εργασία των Mezura-Montes και Coello (2011), διεξάγεται μια παρουσίαση διαφόρων μεθόδων διαχείρισης περιορισμών που έχουν ενσωματωθεί σε αλγόριθμους εμπνευσμένους από τη φύση, όπως εξελικτικοί αλγόριθμοι και αλγόριθμοι που βασίζονται στην ευφυΐα των σμηνών (swarm intelligence). Οι τεχνικές που αναλύονται λεπτομερώς είναι οι κανόνες εφικτότητας (feasibility rules), η στοχαστική κατάταξη (stochastic ranking), η μέθοδος ε-περιορισμού (ε-constrained method), οι συναρτήσεις ποινών (penalty functions), οι ειδικοί τελεστές (special operators), οι πολλαπλές αντικειμενικές συναρτήσεις (multi-objective concept) και το συνονθύλευμα τεχνικών διαχείρισης περιορισμών (ensemble of constraint handling techniques).

Ο Mallipeddi κ.ά. (2012) συνέκριναν την απόδοση του αλγορίθμου Διαφορικής Εξέλιξης σε συνδυασμό από τη μία με τη συνάρτηση ποινής και από την άλλη με διαφορετικές τεχνικές διαχείρισης περιορισμών όπως η ανωτερότητα των εφικτών λύσεων (superiority of feasible solutions), η αυτοπροσαρμοζόμενη συνάρτηση ποινής (self-adaptive penalty), τη μέθοδο ε-περιορισμού (ε-constrained method), τη στοχαστική κατάταξη (stochastic ranking) και το συνονθύλευμα τεχνικών διαχείρισης περιορισμών (ensemble of constraint handling techniques) για την επίλυση προβλημάτων βελτιστοποίησης της μεταφοράς άεργης ισχύος (optimal reactive power dispatch – ORPD).

Ο Jordehi (2015) ανέλυσε την εφαρμοσιμότητα, τα πλεονεκτήματα και τα μειονεκτήματα των διαφορετικών στρατηγικών διαχείρισης περιορισμών που χρησιμοποιούνται στους ευρεστικούς αλγόριθμους βελτιστοποίησης και ιδιαίτερα στο σμήνος σωματιδίων (PSO). Σε αυτήν την εργασία, οι μέθοδοι διαχείρισης περιορισμών που ενσωματώνονται στον PSO κατηγοριοποιούνται ως συναρτήσεις ποινών, διαχωρισμού αντικειμενικής συνάρτησης και περιορισμών, συνδυασμού του PSO με άλλες τεχνικές βελτιστοποίησης καθώς και μεθόδους που δεν μπορούν να συμπεριληφθούν σε καμία από τις προηγούμενες κατηγορίες.

Στην εργασία των Miranda-Varela και Mezura-Montes (2018), τέσσερις παραλλαγές του αλγορίθμου της Διαφορικής Εξέλιξης με χρήση υποκατάστατων μοντέλων SA-DECV (Surrogate-Assisted Differential Evolution with Combined Variants) η καθεμία σε συνδυασμό με μια διαφορετική τεχνική διαχείρισης περιορισμών (κανόνες εφικτότητας, μέθοδος ε-περιορισμού, στοχαστική κατάταξη και μέθοδος διατήρησης ποικιλομορφίας (diversity maintenance)) αξιολογούνται για

να εξετάσουν 24 γνωστές μαθηματικές συναρτήσεις. Τα αποτελέσματα της καλύτερης παραλλαγής του αλγορίθμου της Διαφορικής Εξέλιξης συγκρίνονται με τα αποτελέσματα άλλων αλγορίθμων που χρησιμοποιούν υποκατάστατα μοντέλα όπως ο Γενετικός Αλγόριθμος (GA) και η Μέθοδος Στρατηγικών Εξέλιξης (ES).

Ο Ameca-Alducin κ.ά. (2018) ανέλυσαν την απόδοση ενός απλού αλγορίθμου Διαφορικής Εξέλιξης σε συνδυασμό με τέσσερις διαφορετικές τεχνικές διαχείρισης περιορισμών (συνάρτηση ποινής, κανόνες εφικτότητας, μέθοδος ε-περιορισμού και στοχαστική κατάταξη) χρησιμοποιώντας ένα κοινό σημείο αναφοράς για να προσδιοριστεί ποιες τεχνικές είναι οι πιο κατάλληλες για τους περισσότερο διαδεδομένους τύπους προβλημάτων δυναμικής βελτιστοποίησης με περιορισμούς (Dynamic Constrained Optimization Problems – DCOP).

Ο Lin κ.ά.(2019) εξέτασαν την απόδοση τεσσάρων διαφορετικών αλγορίθμων δυναμικής πολυτροπικής βελτιστοποίησης (dynamic multimodal optimization) βασισμένων σε «πληθυσμό» λύσεων (Clonal Selection, DE, PSO και EP) και πέντε διαφορετικές τεχνικές διαχείρισης περιορισμών (ανωτερότητα της εφικτής λύσης, μέθοδος ε-περιορισμού, συνάρτηση ποινής, δυναμική συνάρτηση ποινής και στοχαστική επιλογή και κατάταξη) σε προβλήματα δυναμικής πολυτροπικής βελτιστοποίησης με περιορισμούς.

Ο Caraffini κ.ά. (2019) ανέλυσαν τη συμπεριφορά πολλών δημοφιλών παραλλαγών του DE σε συνδυασμό με διαφορετικές τεχνικές διαχείρισης περιορισμών, όπως τεχνικές συνάρτησης ποινής και διόρθωσης (επισκευής) (κορεσμός, τοροειδής), προκειμένου καταρχήν να εντοπιστεί ο συνδυασμός εκείνος των τελεστών μετάλλαξης και διασταύρωσης που είναι υπεύθυνοι για την ύπαρξη «δομικής καθοδήγησης» (structural bias) σε αλγόριθμους DE αλλά και να γίνει γνωστό σε ποιο βαθμό μια συνετή επιλογή τόσο των παραμέτρων ελέγχου του αλγόριθμου DE, όσο και των τεχνικών διαχείρισης περιορισμών αλλά και το μέγεθος του «πληθυσμού», μπορεί τελικά να μετριάσει αυτή την «καθοδήγηση».

4.2 Βιβλιογραφική επισκόπηση τεχνικών διαχείρισης περιορισμών

Κατά τη διάρκεια των τελευταίων δύο δεκαετιών, αναπτύχθηκε μια μεγάλη γκάμα τεχνικών διαχείρισης περιορισμών που ενσωματώθηκαν σε διάφορους αλγόριθμους, προκειμένου να επιλυθούν προβλήματα βελτιστοποίησης με περιορισμούς σε ένα ευρύ φάσμα εφαρμογών. Πολλές από αυτές τις τεχνικές είναι παραλλαγές συγκεκριμένων μεθόδων διαχείρισης περιορισμών που έχουν αρχικά προταθεί και στη συνέχεια εξελιχθεί από άλλους ερευνητές, ενώ αρκετοί έχουν καταβάλει σημαντική προσπάθεια να αναπτύξουν καινοτόμες τεχνικές προκειμένου να επιτύχουν τη βέλτιστη απόδοση. Σε αυτή την εργασία, οι πιο αποτελεσματικές και ευρέως χρησιμοποιούμενες τεχνικές διαχείρισης, καθώς και πολλές νέες τεχνικές που προτάθηκαν πρόσφατα, συλλέχθηκαν από την υπάρχουσα βιβλιογραφία,

ομαδοποιήθηκαν σύμφωνα με τη βασική κατηγορία της μεθόδου διαχείρισης περιορισμών στις οποίες βασίζονται και παρουσιάζονται στη συνέχεια.

Παρακάτω, παρατίθεται μια σύντομη επισκόπηση της αρθρογραφίας που συλλέχθηκε με μια συνοπτική περιγραφή του κάθε άρθρου όσον αφορά στην εφαρμοζόμενη τεχνική διαχείρισης περιορισμών, στον αλγόριθμο που χρησιμοποιείται και στο είδος των προβλημάτων που εξετάζονται. Αυτές οι τεχνικές ομαδοποιούνται στις επόμενες γενικές κατηγορίες μεθόδων διαχείρισης περιορισμών:

4.2.1 Μέθοδοι βασισμένες σε συναρτήσεις ποινής

Η επιβολή «ποινής» (penalty) σε μια αντικειμενική συνάρτηση είναι η παλαιότερη αλλά ακόμα και σήμερα η περισσότερο χρησιμοποιούμενη τεχνική για την αντιμετώπιση των περιορισμών στα προβλήματα. Όλα αυτά τα χρόνια έχουν αναπτυχθεί αρκετές τεχνικές που βασίζονται στις συναρτήσεις ποινής, τεχνικές όπως η στατική ποινή (static penalty), η δυναμική ποινή (dynamic penalty), η προσαρμοζόμενη ποινή (adaptive penalty) και η «θανατική» ποινή (death penalty) μεταξύ άλλων.

Ο Miettinen κ.ά. (2003), συνδύασαν γενετικούς αλγόριθμους με πέντε τεχνικές διαχείρισης περιορισμών βασισμένες σε συναρτήσεις ποινής για καθολική βελτιστοποίηση 33 μαθηματικών προβλημάτων διαφορετικών τύπων. Η μέθοδος της προσαρμοζόμενης ποινής αποδείχθηκε η πιο αποτελεσματική, ενώ η μέθοδος της ποινής χωρίς παραμέτρους ήταν η πιο αξιόπιστη.

Ο Da Silva κ.ά. (2011), πρότειναν μια τεχνική προσαρμοζόμενης ποινής (APM) για τη διαχείριση περιορισμών κατά την εφαρμογή του αλγορίθμου της Διαφορικής Εξέλιξης για την εξέταση προβλημάτων βελτιστοποίησης δομικής και μηχανικής φύσης. Αυτή η τεχνική, η οποία δεν απαιτεί επιπλέον παραμέτρους, βασίζεται στη λήψη πληροφορίας, μέσω ανάδρασης, από την τρέχουσα κατάσταση του «πληθυσμού» των υποψηφίων λύσεων και επανακαθορίζει αυτόματα, για κάθε περιορισμό, τον αντίστοιχο συντελεστή ποινής.

Ο Montemurro κ.ά. (2013), χρησιμοποίησαν μια προσέγγιση βασισμένη σε συνάρτηση ποινής, που ονομάστηκε τεχνική Αυτόματης Δυναμικής Ποινολόγησης (Automatic Dynamic Penalization – ADP), η οποία είναι μια απλή και αποτελεσματική τεχνική διαχείρισης περιορισμών χωρίς παραμέτρους για τη βελτιστοποίηση πραγματικών εφαρμογών της επιστήμης του μηχανικού, όπως η μεγιστοποίηση του πρώτου φορτίου λυγισμού σύνθετων ελασμάτων με δεδομένες ελαστικές συμμετρίες. Η στρατηγική ADP υλοποιείται μέσω του αλγορίθμου GA BIANCA, ο οποίος είναι ένας γενετικός αλγόριθμος πολλαπλών «πληθυσμών» για προβλήματα βελτιστοποίησης μηχανικής που επιτρέπει την παράλληλη εξέλιξη ατόμων (individuals) του «πληθυσμού» που ανήκουν σε διαφορετικά είδη, χάρη στην εισαγωγή ορισμένων ειδικών γενετικών τελεστών για διασταύρωση και μετάλλαξη μεταξύ των ειδών.

Στην εργασία του Li κ.ά. (2019), εφαρμόζεται μια δυναμική βελτιστοποίηση βασισμένη σε υποκατάστατα μοντέλα (Dynamic Surrogate Based Optimization – DSBO) με χρήση του μοντέλου Kriging για την εξέταση έξι τυπικών μαθηματικών προβλημάτων. Στη μέθοδο αυτή χρησιμοποιούνται δύο κριτήρια για την επιλογή των απαραίτητων δειγμάτων στη διαδικασία βελτίωσης του μοντέλου: μεγιστοποίηση της συνάρτησης αναμενόμενης βελτίωσης (Expected Improvement – EI) και ελαχιστοποίηση της πρόβλεψης υποκατάστατων μοντέλων. Οι περιορισμοί ανισότητας διαχειρίζονται με τρία διαφορετικά μέσα: περιορισμό της συνάρτησης αναμενόμενης βελτίωσης, ποινολόγηση της πρόβλεψης υποκατάστατων μοντέλων και ποινολόγηση της αντικειμενικής συνάρτησης.

Στην εργασία του Kawachi κ.ά. (2019), μια παραλλαγή του αλγορίθμου της Διαφορικής Εξέλιξης που ονομάζεται LSHADE σε συνδυασμό με μια τεχνική προσαρμοζόμενης ποινής, εφαρμόζεται για τη βελτιστοποίηση των μαθηματικών προβλημάτων αναφοράς που παρέχονται από το αντίστοιχο συνέδριο για την εξελικτική υπολογιστική (Congress on Evolutionary Computation) CEC 2017. Αυτή η τεχνική προσαρμοζόμενης ποινής λαμβάνει υπόψη την ισορροπία μεταξύ των ορίων της τιμής της αντικειμενικής συνάρτησης και της τιμής της παραβίασης των περιορισμών στην περιοχή αναζήτησης και δεν καθοδηγεί την αναζήτηση αποκλειστικά μόνο προς την τιμή της αντικειμενικής συνάρτησης ή μόνο προς την τιμή της παραβίαση των περιορισμών.

Ο Datta κ.ά. (2019), πρότειναν μια τεχνική διαχείρισης περιορισμών με χρήση ατομικής ποινής (Constraint Handling with Individual Penalty – CHIP), η οποία είναι συνδυασμός ενός εξελικτικού αλγορίθμου του NSGA-II που χρησιμοποιείται για προβλήματα δύο αντικειμενικών συναρτήσεων, με μια κλασική συνάρτησης ποινής όπου όμως στη θέση της μίας μόνο παραμέτρου ποινής που σχετίζεται με την τιμή της συνολικής παραβίασης των περιορισμών, χρησιμοποιείται μία μεμονωμένη παράμετρος ποινής για κάθε περιορισμό και επομένως οι παράμετροι ποινής είναι ισάριθμες με τον αριθμό των περιορισμών. Οι παράμετροι ποινής για όλους τους περιορισμούς υπολογίζονται προσαρμοστικά μέσω της θεώρησης της συνολικής παραβίασης των περιορισμών ως βοηθητική αντικειμενική συνάρτηση προς ελαχιστοποίηση. Ο ρόλος της χρήσης της μεθόδου επίλυσης προβλημάτων δύο αντικειμενικών συναρτήσεων είναι να διαχειριστεί ένα πρόβλημα μίας αντικειμενικής συνάρτησης μετατρέποντάς το σε ένα ισοδύναμο πρόβλημα δύο αντικειμενικών συναρτήσεων.

Ο Li κ.ά. (2020), χρησιμοποίησαν έναν βελτιωμένο προσαρμοζόμενο αλγόριθμο Διαφορικής Εξέλιξης (JADE) σε συνδυασμό με μια τεχνική διαχείρισης περιορισμών αυτοπροσαρμοζόμενης ποινής (EJADE-SP) για την εύρεση της βέλτιστης λύσης στο πρόβλημα βέλτιστης ροής ισχύος (Optimal Power Flow – OPF) Ο EJADE-SP είναι μια βελτιωμένη έκδοση του JADE, όπου εφαρμόζονται τέσσερις βελτιώσεις για την ενίσχυση της απόδοσης του τελευταίου κατά την επίλυση του προβλήματος OPF: α) μηχανισμός ταξινόμησης της παραμέτρου crossover rate (CR), β) επανατυχαιοποίηση των παραμέτρων (CR και παράγοντα κλίμακας F), γ) χρήση

της στρατηγικής δυναμικής μείωσης του πληθυσμού και iv) ενσωμάτωση της τεχνικής διαχείρισης περιορισμών αυτοπροσαρμοζόμενης ποινής.

4.2.2 Διαχωρισμός αντικειμενικής συνάρτησης και περιορισμών

Στην εργασία του Deb (2000), προτείνεται μια αναπτυγμένη μέθοδος διαχείρισης περιορισμών, οι κανόνες εφικτότητας (feasibility rules), η οποία βασίζεται στις μεθόδους των συναρτήσεων ποινής αλλά όμως δεν απαιτεί καμία παράμετρο ποινής. Η προσέγγισή του έχει ως υπόβαθρο τη σύγκριση κάθε φορά δύο ατόμων του «πληθυσμού» στη διάρκεια της επιλογικής διαδικασίας σύμφωνα όμως με συγκεκριμένους κανόνες (τους κανόνες εφικτότητας) και ισχύει μόνο για μεθόδους αναζήτησης βασισμένες σε «πληθυσμούς», όπως οι γενετικοί αλγόριθμοι ή άλλες εξελικτικές μέθοδοι υπολογισμού.

Οι Runarsson και Yao (2000) εισάγουν μια νέα μέθοδο διαχείρισης περιορισμών, τη στοχαστική κατάταξη (Stochastic Ranking), για την εξισορρόπηση των τιμών της αντικειμενικής συνάρτησης και της συνάρτησης ποινής μέσω στοχαστικής προσέγγισης, παρουσιάζοντας μια νέα άποψη σχετικά με τις μεθόδους συναρτήσεων ποινής όσον αφορά στην «κυριαρχία» είτε της ποινής ή της αντικειμενικής συνάρτησης. Η ισορροπία επιτυγχάνεται μέσω μιας διαδικασίας κατάταξης των υποψηφίων λύσεων που βασίζεται στον στοχαστικό αλγόριθμο ταξινόμησης φυσαλίδας (stochastic bubble-sort algorithm). Η νέα μέθοδος διαχείρισης περιορισμών ενσωματώνεται στον αλγόριθμο των Στρατηγικών Εξέλιξης για την επίλυση 13 πρότυπων μαθηματικών προβλημάτων.

Στην εργασία των Coello και Montes (2002) για τη βελτιστοποίηση προβλημάτων δομικού σχεδιασμού εφαρμόζεται μια τεχνική γνωστή ως Niched – Pareto Genetic Algorithm (NPGA) η οποία χρησιμοποιεί μία επιλογική διαδικασία μέσω της αρχής της μη – κυριαρχίας. Η αρχή της μη – κυριαρχίας χρησιμοποιείται στη βελτιστοποίηση προβλημάτων πολλαπλών αντικειμενικών συναρτήσεων για να επισημάνει λύσεις που αντιπροσωπεύουν τον καλύτερο δυνατό συμβιβασμό σε ένα σύνολο αντικειμενικών συναρτήσεων. Σύμφωνα με την αρχή αυτή, καμία από τις τιμές των αντικειμενικών συναρτήσεων των μη – κυριαρχούμενων λύσεων δεν μπορεί να βελτιωθεί περαιτέρω χωρίς να επιδεινωθεί κάποια άλλη. Στην περίπτωση προβλημάτων βελτιστοποίησης με περιορισμούς μίας μόνο αντικειμενικής συνάρτησης, η μέθοδος NPGA αντιμετωπίζει τους περιορισμούς ως πρόσθετες αντικειμενικές συναρτήσεις.

Ο Wang κ.ά. (2009) χρησιμοποίησαν έναν υβριδικό εξελικτικό αλγόριθμο που χρησιμοποιεί ταυτόχρονα έναν απλό τελεστή διασταύρωσης και δύο τελεστές μετάλλαξης για τη δημιουργία του «πληθυσμού» των απογόνων προκειμένου να βελτιστοποιήσει 13 γνωστές πρότυπες μαθηματικές συναρτήσεις. Μια προσαρμοζόμενη τεχνική διαχείρισης περιορισμών έχει ενσωματωθεί στον αλγόριθμο αυτό η οποία αποτελείται από τρεις κύριες καταστάσεις (μη εφικτή, ημι-εφικτή και

εφικτή κατάσταση). Για κάθε περίπτωση, έχει σχεδιαστεί ένας μηχανισμός διαχείρισης περιορισμών ανάλογα με την τρέχουσα κατάσταση του «πληθυσμού».

Στην εργασία των Takahama και Sakai (2010) προτείνεται μια παραλλαγή του αλγόριθμου της Διαφορικής Εξέλιξης όσον αφορά στον τελεστή μετάλλαξης σε συνδυασμό με τη μέθοδο διαχείρισης ε - περιορισμού (ε DEag) προκειμένου να επιλυθούν 18 μαθηματικά προβλήματα που δόθηκαν σε ειδικό συνέδριο για «Βελτιστοποίηση προβλημάτων πραγματικών παραμέτρων με περιορισμούς μίας αντικειμενικής συνάρτησης» στο CEC2010.

Ένας μηχανισμός διαχείρισης περιορισμών που ονομάζεται «παραβίαση περιορισμού με αριθμητικό διάστημα (constraint violation with interval arithmetic – CVI) ενσωματώνεται από τον Mazhoud κ.ά. (2013) σε έναν προσαρμοσμένο αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων (PSO) για την επίλυση 24 πρότυπων μαθηματικών προβλημάτων και 3 προβλημάτων μηχανικής. Αυτή η μέθοδος έγκειται στην αξιολόγηση της εγγύτητας των λύσεων στην εφικτή περιοχή και η συνολική παραβίαση των περιορισμών εισάγεται ως αντικειμενική συνάρτηση για ελαχιστοποίηση ενώ το αριθμητικό διάστημα χρησιμοποιείται για την κανονικοποίηση των συνολικών παραβιάσεων. Το τελικό πρόβλημα επιλύεται χρησιμοποιώντας μια απλή λεξικογραφική μέθοδο.

Ο Zhang κ.ά. (2015) πρότειναν έναν νέο εξελικτικό αλγόριθμο που ονομάζεται Backtracking Search Algorithm (BSA) τον οποίον εφάρμοσαν για την επίλυση 13 πρότυπων συναρτήσεων και 4 προβλημάτων βελτιστοποίησης μηχανικής. Στην εργασία τους αυτή συνδυάζονται με τον BSA συνολικά τρεις μέθοδοι διαχείρισης περιορισμών για επίλυση προβλημάτων βελτιστοποίησης με περιορισμούς, οι κανόνες εφικτότητας και κυριαρχίας (Feasibility And Dominance – FAD), μια μέθοδος ε - περιορισμού με σταθερό τρόπο ελέγχου της τιμής του ε και μια ακόμη νέα μέθοδος ε - περιορισμού με αυτοπροσαρμοζόμενο τρόπο ελέγχου της τιμής του ε η οποία υπολογίζεται κάθε φορά σύμφωνα με τις ιδιότητες του τρέχοντος «πληθυσμού».

Ο Chehourι κ.ά. (2016) χρησιμοποίησαν ένα γενετικό αλγόριθμο ως αριθμητικό εργαλείο για να εφαρμόσουν μία τεχνική διαχείρισης περιορισμών χωρίς παραμέτρους με χρήση του «παράγοντα παραβίασης» (violation factor) ώστε να εξετάσουν γενικά προβλήματα μη γραμμικού προγραμματισμού (NLP). Ο «πληθυσμός» χωρίζεται σε δύο οικογένειες, την οικογένεια των εφικτών λύσεων (V_0) και αυτή των μη εφικτών (V_1) η οποία αποτελείται από άτομα που παραβιάζουν τουλάχιστον έναν περιορισμό. Η πρώτη ομάδα υποψήφιων λύσεων (V_0) ταξινομείται σύμφωνα με την τιμή της αντικειμενικής συνάρτησης ενώ η δεύτερη ομάδα (V_1) ταξινομείται σύμφωνα με κάποιους κανόνες σύγκρισης (κανόνες βασισμένους στην εφικτότητα) για την εξέλιξη του πληθυσμού σε κάθε γενιά.

Ο Peng κ.ά. (2018) χρησιμοποίησαν μια τεχνική βασισμένη σε «καθοδηγούμενα δυναμικά βάρη» (biased dynamic weights) για εξελικτικούς

αλγόριθμους με περιορισμούς, προκειμένου να βελτιστοποιήσουν 24 πρότυπα προβλήματα του CEC2006 και 18 προβλήματα του CEC2010. Τα «καθοδηγούμενα βάρη» χρησιμοποιούνται για την επιλογή ατόμων με χαμηλές τιμές αντικειμενικής συνάρτησης και χαμηλό βαθμό παραβίασης περιορισμών. Έτσι, κατά την εξελικτική διαδικασία, δίνεται περισσότερη έμφαση σε άτομα με χαμηλότερες τιμές αντικειμενικής και χαμηλότερο βαθμό παραβίασης περιορισμών ρυθμίζοντας δυναμικά τα βάρη αυτά. Αυτή η μέθοδος επιλέγει πρώτα τα μη εφικτά άτομα του πληθυσμού που βρίσκονται όμως κοντά στην εφικτή περιοχή από τον συνολικό πληθυσμό γονέων και απογόνων.

Ο Fan κ.ά. (2018) υιοθέτησαν μια βελτιωμένη μέθοδο ε-περιορισμού (IEpsilon) ενσωματωμένη σε μια βελτιωμένη έκδοση του αλγορίθμου LSHADE (που ονομάζεται LSHADE44) για την επίλυση 28 πρότυπων προβλημάτων του CEC2017. Η μέθοδος IEpsilon διαφέρει από την κλασική μέθοδο διαχείρισης ε-περιορισμού, καθώς σε αυτήν προσαρμόζεται διαρκώς η τιμή της παραμέτρου ϵ σύμφωνα με το ποσοστό των εφικτών λύσεων στον τρέχοντα πληθυσμό, δίνοντας έτσι τη δυνατότητα εξισορρόπησης της αναζήτησης μεταξύ εφικτών και μη εφικτών περιοχών, κατά την εξελικτική διαδικασία.

Ο Rodrigues κ.ά. (2018) πρότειναν τη μέθοδο Extended Balanced Ranking (E-BRM) ως τεχνική διαχείρισης περιορισμών για τη βελτίωση της αποτελεσματικότητας της αρχικής μεθόδου BRM, προκειμένου να επιλυθούν προβλήματα βελτιστοποίησης με εξελικτικούς αλγόριθμους, όπως οι πρότυπες συναρτήσεις του CEC2006 και του CEC2010 και τα πρότυπα προβλήματα μηχανικής. Η E-BRM περιλαμβάνει μια αυτοπροσαρμοζόμενη διαδικασία που κατατάσσει τις υποψήφιες λύσεις σε δύο ομάδες, εφικτές και μη εφικτές, οι οποίες όμως συγχωνεύονται κατά τη διαδικασία αναζήτησης ανάλογα με την ποσότητά τους εντός του πληθυσμού.

Ο Li κ.ά. (2019) υιοθέτησαν μια τεχνική ασαφούς καθοδήγησης της αναζήτησης που ονομάζεται Fuzzy Search Bias (FSB) η οποία κατευθύνεται από μια συνάρτηση μέλους, σε συνδυασμό με εξελικτικούς αλγόριθμους για τη βελτιστοποίηση 24 πρότυπων συναρτήσεων. Σε αυτήν τη μέθοδο χρησιμοποιείται η ασαφής γνώση (fuzzy knowledge) ότι μια λύση με μεγάλη τιμή παραβίασης περιορισμών καθοδηγεί την αναζήτηση προς λύσεις με μικρότερη τιμή παραβίασης περιορισμών και μια λύση με μικρή τιμή παραβίασης περιορισμών καθοδηγεί την αναζήτηση προς λύσεις με καλύτερη τιμή αντικειμενικής συνάρτησης.

Στην εργασία των Zhao κ.ά. (2020) μια νέα τεχνική διαχείρισης περιορισμών που ονομάζεται κανόνες εφικτότητας με ενσωμάτωση πληροφορίας από την αντικειμενική συνάρτηση (feasibility rule with the incorporation of objective function information – FROFI), εφαρμόζεται σε έναν αλγόριθμο Διαφορικής Εξέλιξης υποβοηθούμενο από ένα υποκατάστατο μοντέλο RBF για να εξεταστούν δύο πρότυπες συναρτήσεις και δύο μοντέλα δεξαμενών. Έτσι, για την εξισορρόπηση των περιορισμών και της αντικειμενικής συνάρτησης εφαρμόζεται αυτή η τεχνική η οποία

πρακτικά ενσωματώνει στη γνωστή μέθοδο διαχείρισης περιορισμών των κανόνων εφικτότητας πληροφορίες της αντικειμενικής συνάρτησης μέσω των τελεστών του αλγόριθμου της Διαφορικής Εξέλιξης και έναν μηχανισμό αντικατάστασης.

Με βάση τον κανόνα ότι μεταξύ δύο λύσεων, αυτή με την καλύτερη συνάρτηση καταλληλότητας (fitness function) ξεπερνά την άλλη, ο Liu κ.ά. (2020), ενσωμάτωσαν κάποιους βελτιωμένους κανόνες εφικτότητας του Deb (Improved Deb – IDeb) σε έναν αλγόριθμο βελτιστοποίησης τον Fruit fly algorithm (FOA) (μια στρατηγική αναζήτησης βασισμένη στη μνήμη και εμπνευσμένη από τη συμπεριφορά που εμφανίζουν οι μύγες κατά τη συλλογή τροφής από τα φρούτα) προκειμένου να βελτιστοποιήσουν δικτυωτές κατασκευές. Αυτή η μέθοδος αποθηκεύει τις υποψήφιες λύσεις σε μία μνήμη και στη συνέχεια χωρίζει τη διαδικασία αναζήτησης σε δύο φάσεις: Η πρώτη φάση στοχεύει στην αναζήτηση στην αποθηκευμένη μνήμη για εφικτές λύσεις χρησιμοποιώντας τον παραδοσιακό κανόνα του Deb, ενώ στη δεύτερη φάση, μόνο η λύση με καλύτερη τιμή αντικειμενικής συνάρτησης από αυτήν της χειρότερης λύσης που είναι αποθηκευμένη στην μνήμη, θα οδηγήσει σε υπολογισμό της τιμής παραβίασης των περιορισμών.

Ο Chu κ.ά. (2020) χρησιμοποίησαν τη μέθοδο ε-περιορισμού ως τεχνική για τη διαχείριση περιορισμών στον αλγόριθμο της Διαφορικής Εξέλιξης, προκειμένου να βελτιστοποιήσουν την τοπολογία των Polyline-based Core Sandwich Structures (PCSSs). Συνολικά αντιμετωπίζονται τρεις περιορισμοί που αφορούν στον όγκο, στη γωνία και στο πάχος. Οι περιορισμοί πάχους αντιμετωπίζονται από τα άνω και κάτω όρια, οι περιορισμοί γωνίας υπολογίζονται με απλές μαθηματικές παραγώγους και οι περιορισμοί όγκου με χρήση του υποκατάστατου μοντέλου RBF. Δεδομένου ότι ο περιορισμός όγκου είναι ο πιο σημαντικός περιορισμός, εισάγεται η μέθοδος διαχείρισης ε-περιορισμού για τη σύγκριση διαφορετικών μη εφικτών λύσεων.

Ο αλγόριθμος βελτιστοποίησης «έρευνας και διάσωσης» (Search and Rescue – SAR) σε συνδυασμό με την τεχνική διαχείρισης ε-περιορισμού χρησιμοποιείται από τον Shabani κ.ά. (2020) για την επίλυση πρότυπων συναρτήσεων και προβλημάτων μηχανικής με περιορισμούς. Αυτός ο μετα-ευριστικός αλγόριθμος μιμείται τη εξερευνητική συμπεριφορά των ανθρώπων κατά τη διάρκεια έρευνας και διάσωσης.

Στην εργασία του Stanonon κ.ά. (2020) αρκετές νέες μέθοδοι διαχείρισης περιορισμών που ακολουθούν τη βασική ιδέα της τεχνικής του ε-περιορισμού συνδυάζονται με τον αλγόριθμο Διαφορικής Εξέλιξης για τη βελτιστοποίηση των πρότυπων συναρτήσεων του CEC2017 καθώς και ένα σύνολο προβλημάτων κατανομής φορτίου για εξοικονόμηση ενέργειας (Economic Load Dispatch). Σε αυτές τις μεθόδους, το επίπεδο ϵ καθορίζεται με βάση την τρέχουσα παραβίαση στον «πληθυσμό» ενώ μεμονωμένα επίπεδα ϵ διατηρούνται για κάθε περιορισμό. Παράλληλα χρησιμοποιείται και ένας συνδυασμός των τιμών της συνάρτησης καταλληλότητας και της παραβίασης των περιορισμών για τον προσδιορισμό των μη εφικτών λύσεων.

Ο Rodrigues κ.ά. (2020) πρότειναν μια μέθοδο βασισμένη στον αλγόριθμο σμήνους σωματιδίων, την Weighted-Dynamic-Objective Constraint-Handling PSO Method (WDOCHM-PSO) για να σχεδιάσει τους πίνακες στάθμισης ενός μοντέλου πρόβλεψης ελέγχου (Model-Based Predictive Controller – MBPC) για μια επαγωγική γεννήτρια διπλής τροφοδοσίας (Doubly Fed Induction Generator – DFIG). Η ιδέα της DOCHM είναι να χωρίσει το πρόβλημα βελτιστοποίησης με περιορισμούς σε δύο αντικειμενικές συναρτήσεις χωρίς περιορισμούς. Η πρωτεύουσα αντικειμενική συνάρτηση αντιπροσωπεύει την απόσταση του ατόμου από την εφικτή περιοχή, ωστόσο εάν το άτομο βρίσκεται εντός της εφικτής περιοχής, τότε η διαδικασία ελαχιστοποίησης αλλάζει στη δευτερεύουσα αντικειμενική συνάρτηση που σχετίζεται με το πρόβλημα βελτιστοποίησης χωρίς περιορισμούς.

Ο Mao κ.ά. (2020) υιοθέτησαν την τεχνική διαχείρισης περιορισμών της στοχαστικής κατάταξης την οποία ενσωμάτωσαν σε έναν εξελικτικό αλγόριθμο τον shuffled complex evolution (SCE), προκειμένου να χρησιμοποιηθεί σαν μέθοδος καθολικής βελτιστοποίησης για την επίλυση προβλημάτων προγραμματισμού ταμειωτήρων με περιορισμούς. Η εν λόγω μέθοδος συνδυάζει τα πλεονεκτήματα της μεθόδου simplex και του εξελικτικού αλγόριθμου competitive complex evolution (CCE).

Στην εργασία του Kaucic κ.ά. (2020) χρησιμοποιείται μια έκδοση του αλγορίθμου PSO (IC-PSO) σε συνδυασμό με μια υβριδική διαδικασία διαχείρισης περιορισμών η οποία ενσωματώνει ταυτόχρονα έναν μηχανισμό επιδιόρθωσης και την αρχή της «κυριαρχίας» για τη βελτιστοποίηση αριθμητικών προβλημάτων. Ένας επισκευαστικός τελεστής προβάλλει στην εφικτή περιοχή τις υποψήφιες λύσεις που δεν πληρούν τους περιορισμούς και στη συνέχεια ενσωματώνει στον προηγούμενο μηχανισμό επιδιόρθωσης ένα σύνολο κανόνων εφικτότητας βασισμένους στην αρχή της «κυριαρχίας». Ο αλγόριθμος εφαρμόζεται για να εξεταστεί ένα αριθμητικό παράδειγμα, όπου ο δείκτης Euro Stoxx 50 χρησιμοποιείται ως πρότυπο.

4.2.3 Συνδυασμός μεθόδων διαχείρισης περιορισμών

Ο συνδυασμός διαφορετικών τεχνικών διαχείρισης περιορισμών, αντί της χρήσης μόνο μίας τεχνικής, κατά τη διαδικασία βελτιστοποίησης έχει γίνει πολύ δημοφιλής μέθοδος τα τελευταία χρόνια.

Στην εργασία των Mallipeddi και Suganthan (2010) χρησιμοποιείται ένα συνονθύλευμα τεχνικών διαχείρισης περιορισμών (Ensemble of Constraint Handling Techniques – ECHT), η ανωτερότητα των εφικτών λύσεων, η αυτοπροσαρμοζόμενη ποινή, η μέθοδος ε-περιορισμού και η στοχαστική κατάταξη, σε συνδυασμό με τους αλγόριθμους του Εξελικτικού Προγραμματισμού (Evolutionary Programming – EP) και της Διαφορικής Εξέλιξης για την επίλυση προβλημάτων βελτιστοποίησης πραγματικών παραμέτρων με περιορισμούς. Κάθε μέθοδος διαχείρισης περιορισμών έχει τον δικό της πληθυσμό και ένα σημαντικό χαρακτηριστικό της μεθόδου ECHT

είναι η χρήση κάθε συνάρτησης ξεχωριστά από κάθε πληθυσμό που σχετίζεται με την αντίστοιχη τεχνική διαχείρισης περιορισμών.

Ένα συνονθύλευμα των μεθόδων συνάρτησης ποινής και ανωτερότητας των εφικτών λύσεων (SOF) που ενσωματώνεται στον αλγόριθμο UDE χρησιμοποιείται από τον Trivedi κ.ά. (2018) για τη βελτιστοποίηση 28 πρότυπων προβλημάτων του CEC 2017. Σε αυτήν την τεχνική, η μέθοδος συνάρτησης ποινής υιοθετείται για το πρώτο 50% του μέγιστου επιτρεπόμενου αριθμού υπολογισμών της αντικειμενικής συνάρτησης, ενώ η μέθοδος SOF υιοθετείται για τους υπόλοιπους υπολογισμούς. Η τεχνική του συνονθυλεύματος προωθεί την «εξερεύνηση» του χώρου αναζήτησης στο πρώτο μισό της διαδικασίας υιοθετώντας τη μέθοδο της ποινής ενώ εστιάζει στην «εκμετάλλευση» αυτού στο τελευταίο μισό του σταδίου αναζήτησης χρησιμοποιώντας τη μέθοδο SOF.

Από τον Biswas κ.ά. (2018) υιοθετούνται η ανωτερότητα των εφικτών λύσεων (SF), η αυτοπροσαρμοζόμενη ποινή (SP) και ένα συνονθύλευμα αυτών των δύο, με την Διαφορική Εξέλιξη να είναι ο βασικός αλγόριθμος αναζήτησης, προκειμένου να βελτιστοποιηθεί το πρόβλημα της βέλτιστης ροής ισχύος (OPF). Τα πρότυπα IEEE 30, IEEE 57 και IEEE 118-bus systems εξετάζονται με αυτόν τον αλγόριθμο για διάφορες αντικειμενικές συναρτήσεις του OPF όπως το κόστος, η εκπομπή, η απώλεια ισχύος, η σταθερότητα τάσης κ.λπ.

Ο Malan (2018) χρησιμοποίησε μια τεχνική διαχείρισης περιορισμών με γνώμονα το διαμορφούμενο «τοπίο» σε συνδυασμό με τον αλγόριθμο της Διαφορικής Εξέλιξης για τη βελτιστοποίηση 18 προβλημάτων αναφοράς του CEC 2010. Αυτή η προσέγγιση διαφέρει από το συνονθύλευμα των μεθόδων (ECHT) στο ότι ο αλγόριθμος εναλλάσσεται διαρκώς μεταξύ τεσσάρων στρατηγικών διαχείρισης περιορισμών με βάση τα εκάστοτε χαρακτηριστικά του «τοπίου» που συλλέγονται συνεχώς ως μέρος της διαδικασίας αναζήτησης, επομένως μόνο μία τεχνική είναι ενεργή κάθε φορά.

Στην εργασία του Wang κ.ά. (2019) ένας αλγόριθμος Διαφορικής Εξέλιξης που ονομάζεται Variable Neighbourhood Search Differential Evolution (DEVNS), συνδυάζεται με τις μεθόδους των εφικτών λύσεων (SF) και του ε-περιορισμού για τη βελτιστοποίηση των προβλημάτων αναφοράς που παρέχονται για το CEC 2017. Ο DEVNS δημιουργεί κάθε δοκιμαστικό άτομο (trial individual) με χρήση της τοπικής αναζήτησης VNS που έχει δύο διαφορετικές στρατηγικές μετάλλαξης, καθώς και τυχαία ποσοστά διασταύρωσης και μετάλλαξης, προκειμένου να δημιουργηθούν πολλαπλές υποψήφιες λύσεις μεταξύ των οποίων η καλύτερη επιλέγεται τελικά ως δοκιμαστικό άτομο. Στη συνέχεια, το άτομο-στόχος (target individual) συγκρίνεται με το άτομο δοκιμής με τη μέθοδο SF προκειμένου να επιλεγεί ο απόγονος που θα περάσει στην επόμενη γενιά. Ωστόσο, στη μέθοδο αυτή, πρώτα απ' όλα θεωρείται ότι κάθε άτομο που έχει χαμηλότερη τιμή παραβίασης περιορισμών από το επίπεδο ε αντιμετώπιζεται ως εφικτή λύση.

Ο Javed κ.ά. (2019) χρησιμοποίησαν ένα συνονθύλευμα τεχνικών διαχείρισης περιορισμών (ECHT) που αποτελείται από τέσσερις διαφορετικές τεχνικές την ανωτερότητα των εφικτών λύσεων, την αυτοπροσαρμοζόμενη ποινή, τη μέθοδο ε-περιορισμού και τη στοχαστική κατάταξη, ενσωματωμένες στους αλγόριθμους Διαφορικής Εξέλιξης JADE και SADE για τη βελτιστοποίηση 24 προβλημάτων αναφοράς του CEC2006.

4.2.4 Μέθοδοι βασισμένες σε αλγόριθμους επισκευής

Στις συγκεκριμένες μεθόδους, για την επίλυση προβλημάτων βελτιστοποίησης με περιορισμούς, εφαρμόζεται μια συγκεκριμένη διαδικασία (συνήθως ευρεστική) που ονομάζεται αλγόριθμος επισκευής ή επιδιόρθωσης με στόχο τη μετατροπή ενός μη εφικτού ατόμου σε εφικτό (σ.σ. επισκευή). Το επιδιορθωμένο άτομο μπορεί στη συνέχεια να χρησιμοποιηθεί είτε απλά για εκτίμηση της τιμής της συνάρτησης καταλληλότητας ή ακόμα και για αντικατάσταση του αρχικού ατόμου στην επόμενη γενιά κατά τη διάρκεια της εξελικτικής διαδικασίας. Αυτό εξαρτάται κυρίως από τη φύση του προβλήματος αλλά και από το σχεδιασμό που κάνει κάθε ερευνητής. Κάποιοι από αυτούς δεν αντικαθιστούν ποτέ το αρχικό άτομο με το επισκευασμένο. Άλλοι αντικαθιστούν μόνο ένα μικρό ποσοστό από αυτά, ενώ υπάρχουν ερευνητές που αντικαθιστούν όλα τα μη εφικτά άτομα στον πληθυσμό με τα αντίστοιχα επισκευασμένα κατά τη διάρκεια της διαδικασίας.

Οι Chootinan και Chen (2006) πρότειναν μια διαδικασία επισκευής ενσωματωμένη σε έναν απλό γενετικό αλγόριθμο ως ειδικό τελεστή για τη βελτιστοποίηση 11 πρότυπων μαθηματικών προβλημάτων. Η βασική ιδέα αυτής της μεθόδου είναι να αξιοποιήσει την πληροφορία της παραγώγου που προκύπτει από το σύνολο των περιορισμών για τη συστηματική επιδιόρθωση των μη εφικτών λύσεων. Η πληροφορία της παραγώγου χρησιμοποιείται για να κατευθύνει τις μη εφικτές λύσεις προς την εφικτή περιοχή και προέρχεται είτε απευθείας από τους περιορισμούς ή μπορεί ακόμα και να προσεγγιστεί με μία όχι και τόσο ακριβή μέθοδο όπως οι πεπερασμένες διαφορές.

Στην εργασία του Salcedo-Sanz (2009) διεξάγεται μια ολοκληρωμένη έρευνα των βασικών ευρεστικών μεθόδων επισκευής που χρησιμοποιούνται με εξελικτικούς αλγόριθμους, προκειμένου να εξεταστεί η απόδοση των προβλημάτων βελτιστοποίησης με περιορισμούς.

Στην εργασία του Zein κ.ά. (2017) χρησιμοποιείται ένας τελεστής επισκευής ενσωματωμένος σε έναν βρόχο βελτιστοποίησης που χρησιμοποιεί έναν γενετικό αλγόριθμο για την εκτέλεση του προκαταρκτικού σχεδιασμού μιας σύνθετης κατασκευής. Ο αλγόριθμος βελτιστοποίησης αποτελείται από έναν διπλό βρόχο, όπου ο εσωτερικός λύνει το πρόβλημα βελτιστοποίησης χρησιμοποιώντας τον γενετικό αλγόριθμο και τα υποκατάστατα μοντέλα, ενώ ο εξωτερικός βρόχος προσθέτει τη βέλτιστη υποκατάστατη λύση στο σχεδιασμό και ανακατασκευάζει τα υποκατάστατα μοντέλα.

Ο Li κ.ά. (2018) ενσωμάτωσαν στον αλγόριθμο PSO και DE μια τροποποιημένη διαδικασία επιδιόρθωσης για να επιτύχουν την ικανοποίηση των περιορισμών ισότητας, η οποία παίρνει την παράγωγο της αντικειμενικής συνάρτησης ως ρυθμό αύξησης και ρυθμίζει την έξοδο της μονάδας διαμοιράζοντας τη μη ισορροπημένη ποσότητα της παραβίασης των περιορισμών του συστήματος. Αυτή η μέθοδος αξιολογείται για προβλήματα κατανομής φορτίου για εξοικονόμηση ενέργειας (Economic Dispatch) χρησιμοποιώντας έξι διαφορετικά συστήματα γεννητριών αναφοράς.

Ο Gandomi και ο Deb (2020) εφάρμοσαν μια τεχνική διαχείρισης περιορισμών που ονομάζεται μέθοδος ενημέρωσης των ορίων (Boundary Update – BU) σε εξελικτικούς και μαθηματικούς αλγόριθμους βελτιστοποίησης για τη δοκιμή επίλυσης αρκετών μαθηματικών προβλημάτων και προβλημάτων μηχανικής. Έχει σχεδιαστεί για την επιδιόρθωση των ορίων μίας ή περισσότερων επιλεγμένων μεταβλητών προκειμένου να ικανοποιηθούν οι περιορισμοί. Τα όρια συγκεκριμένων μεταβλητών (μεταβλητών προς επισκευή) περιορίζονται και αλλάζουν κατά τη διαδικασία βελτιστοποίησης και έτσι δημιουργούνται νέες λύσεις εντός των ενημερωμένων ορίων του χώρου αναζήτησης.

4.2.5 Μέθοδοι βασισμένες στα όρια των μεταβλητών

Οι περιορισμοί των προβλημάτων βελτιστοποίησης μπορούν να χωριστούν σε δύο κατηγορίες: σε περιορισμούς μορφής συνάρτησης και σε απλά όρια μεταβλητών. Η πρώτη κατηγορία περιλαμβάνει τους περιορισμούς ανισότητας και ισότητας και αντιπροσωπεύει έναν πιο περίπλοκο τύπο περιορισμών που ορίζονται ως συναρτήσεις, ενώ η δεύτερη κατηγορία αφορά τα άνω και κάτω όρια κάθε μεταβλητής τα οποία περιορίζουν τις τιμές που λαμβάνουν αυτές σε ένα πρόβλημα. Οι περισσότερες έρευνες προσπαθούν να βελτιστοποιήσουν τα προβλήματα με περιορισμούς χρησιμοποιώντας τεχνικές που διαχειρίζονται τις συναρτήσεις, ενώ σε λίγες από αυτές έχει καταβληθεί σημαντική προσπάθεια να διαχειριστούν σωστά τα όρια των μεταβλητών εφαρμόζοντας μεθόδους διαχείρισης ορίων (Boundary Constraint Handling Methods – BCHMs). Αυτές οι μέθοδοι είναι σχέσεις ελέγχου που προσπαθούν να τροποποιήσουν και να διορθώσουν τη θέση ενός μη εφικτού διανύσματος υποψήφιων λύσεων ενός προβλήματος και να το επανατοποθετήσουν μέσα στον χώρο αναζήτησης ώστε να γίνει εφικτό.

Στην εργασία των Gandomi και Yang (2012) ένα απλό και αποτελεσματικό εξελικτικό σχήμα για την διαχείριση των ορίων που ενσωματώνεται στον αλγόριθμο της Διαφορικής Εξέλιξης, συγκρίνεται με άλλες αντίστοιχες προσεγγίσεις για ένα ευρύ σύνολο προβλημάτων συναρτήσεων αναφοράς. Σε αυτό το σχήμα, όταν ένα στοιχείο υπερβαίνει το όριό του, τότε αντικαθίσταται από ένα τυχαίο στοιχείο μεταξύ του σχετικού ορίου και του αντίστοιχου στοιχείου της καλύτερης μέχρι στιγμής λύσης. Η χρήση αυτής της βέλτιστης λύσης έχει το πλεονέκτημα της δημιουργίας

νέων λύσεων χρησιμοποιώντας την ενημερωμένη πληροφορία για τις λύσεις και τον χώρο αναζήτησης και αυτή η ροή πληροφοριών εξελίσσεται διαρκώς.

Ο Gandomi και ο Kashani (2018) πρότειναν μια πιθανοτική εξελικτική μέθοδο διαχείρισης ορίων (Probabilistic Evolutionary Boundary Constraint Handling – PEBCH) σε συνδυασμό με τον αλγόριθμο PSO για τη βελτιστοποίηση 7 πρότυπων συναρτήσεων. Το EBCH είναι ένα σχήμα που επαναφέρει τα παραβιασμένα στοιχεία μέσα στον επιτρεπόμενο χώρο. Επειδή όμως, η παραβίαση των ορίων μπορεί να περιέχει πολλές χρήσιμες πληροφορίες, γι' αυτό ο τρόπος επαναφοράς των στοιχείων στο χώρο των αποδεκτών λύσεων επιδρά σημαντικά στην απόδοση του αλγορίθμου. Για το σκοπό αυτό, χρησιμοποιείται μια συνάρτηση κατανομής πιθανότητας η οποία έχει μεγαλύτερη πυκνότητα κοντά στο παραβιασμένο όριο και αλλάζει ανάλογα με την απόσταση μεταξύ παραβιασμένης λύσης και ορίων.

Οι επιπτώσεις διαφορετικών χαοτικών χαρτών στις δυνατότητες εξερεύνησης / εκμετάλλευσης των εξελικτικών αλγορίθμων που είναι εξοπλισμένοι με μια τεχνική γνωστή ως μέθοδος διαχείρισης ορίων, διερευνώνται από τον Mozaffari κ.ά. (2019). Οι στρατηγικές διαχείρισης ορίων (BCHSs) ελέγχουν τις θέσεις που λαμβάνουν τα παραγόμενα χρωμοσώματα και τις διορθώνουν όταν υπερβαίνουν το αποδεκτό πεδίο λύσεων. Οι BCHSs αναζητούν ένα σχήμα ελέγχου που να μπορεί να ανιχνεύσει και να τροποποιήσει τη θέση των παραβιασμένων λύσεων. Η τροποποίηση γίνεται χρησιμοποιώντας έναν χάρτη που μεταφέρει τις παραβιασμένες λύσεις εντός του αποδεκτού εύρους λύσεων.

Στην εργασία του Biedrzycki κ.ά. (2019), συνδυασμοί επτά διαφορετικών αλγορίθμων της Διαφορικής Εξέλιξης και δεκαεπτά διαφορετικών μεθόδων διαχείρισης περιορισμών (BCHM), που μπορούν να ταξινομηθούν ως μέθοδοι συνάρτησης ποιότητος, μέθοδοι επιδιόρθωσης και εξειδικευμένες μέθοδοι μετάλλαξης, αναλύονται για να αποτιμήσουν την αποτελεσματικότητά τους στην εκμετάλλευση και την εξερεύνηση προκειμένου να βελτιστοποιηθούν οι συναρτήσεις αναφοράς που παρέχονται από τους CEC 2017. Η αποτελεσματικότητα των αλγορίθμων διαφέρει σημαντικά ανάλογα με την επιλογή συγκεκριμένων BCHM.

Ο Juárez-Castillo κ.ά. (2019) εφάρμοσαν ένα προσαρμοζόμενο σχήμα διαχείρισης ορίων (ABCCHS) σε συνδυασμό με τον αλγόριθμο PSO και έναν αλγόριθμο DE για τη βελτιστοποίηση 60 προβλημάτων πραγματικών παραμέτρων μίας αντικειμενικής συνάρτησης με περιορισμούς. Το ABCCHS δρα σε δύο στάδια: στο πρώτο στάδιο, όταν δεν έχουν βρεθεί ακόμη εφικτές λύσεις, χρησιμοποιείται μια μέθοδος διαχείρισης ορίων (BCHM), ενώ στο δεύτερο στάδιο, ένα από τα k BCHM επιλέγεται τυχαία, αλλά σύμφωνα με τις πιθανότητες που σχετίζονται με κάθε μέθοδο. Αυτό το προσαρμοζόμενο σχήμα λαμβάνει ανάδραση από το σύστημα για την διαρκή ενημέρωση των πιθανοτήτων που σχετίζονται με καθένα από τα k BCHMs για τη δημιουργία καλύτερων επιδιορθωμένων διανυσμάτων λύσεων.

Στην εργασία των Arouri και Sayyafzadeh (2020) ένας αλγόριθμος με βάση την παράγωγο εφαρμόζεται για την επίλυση προβλημάτων βελτιστοποίησης παραγωγής. Αυτός ο αλγόριθμος είναι ένας συνδυασμός της στοχαστικής προσέγγισης simultaneous perturbation stochastic approximation (SPSA) για την εκτίμηση των παραγώγων, με ένα προσαρμοζόμενο υπολογιστικό πλαίσιο. Η κατεύθυνση αναζήτησης είναι ένα διάνυσμα που αποτελείται από τον μέσο όρο των προηγούμενων παραγώγων το οποίο κανονικοποιείται μέσω μιας εκτιμώμενης διαφοράς. Συνολικά χρησιμοποιούνται δύο μέθοδοι για την αντιμετώπιση των ορίων των μεταβλητών: η μέθοδος προβολής (projection method) και η μέθοδος λογαριθμικού μετασχηματισμού (logarithmic transformation method).

Στην εργασία του Biedrzycki κ.ά. (2020) αξιολογούνται οι συνδυασμοί του αλγορίθμου των Στρατηγικών Εξέλιξης Covariance Matrix Adaptation (CMA-ES) με 22 BCHMs μέσω της ανάλυσης των αποτελεσμάτων που προκύπτουν για μονοτροπικές και πολυτροπικές συναρτήσεις των CEC 2017 και BBOB. Όλα τα εξεταζόμενα BCHM μπορούν να αντιστοιχιστούν σε τέσσερις ομάδες: μέθοδοι επισκευής, μέθοδοι διατήρησης της εφικτότητας, εξειδικευμένες μέθοδοι που ισχύουν μόνο για τον αλγόριθμο CMA-ES και συναρτήσεις ποινής.

4.2.6 Άλλες μέθοδοι

Ο Atamna κ.ά. (2020) εφάρμοσαν μια επαυξημένη προσέγγιση Lagrangian (augmented Lagrangian method) ως τεχνική διαχείρισης περιορισμών στον αλγόριθμο των Στρατηγικών Εξέλιξης για να εξετάσουν δύο γραμμικές κυρτές τετραγωνικές συναρτήσεις με περιορισμούς, τη σφαίρα και το ελλειψοειδές, με έναν μέτριο αριθμό περιορισμών. Σε αυτήν την προσέγγιση, το αρχικό πρόβλημα μετατρέπεται σε πρόβλημα χωρίς περιορισμούς και η προς βελτιστοποίηση αντικειμενική συνάρτηση μετατρέπεται σε μια επαυξημένη Lagrangian συνάρτηση της οποίας οι παράμετροι προσαρμόζονται κατά τη διαδικασία βελτιστοποίησης.

Ο Qian κ.ά. (2020) χρησιμοποίησαν έναν γενετικό αλγόριθμο για την επίλυση εννέα αριθμητικών παραδειγμάτων και δύο πρακτικών προβλημάτων μηχανικής. Για να βελτιωθεί η αποτελεσματικότητα της επίλυσης αυτών των προβλημάτων, αντί για τους πραγματικούς περιορισμούς χρησιμοποιείται το υποκατάστατο μοντέλο του Kriging. Επειδή υπάρχουν σφάλματα πρόβλεψης μεταξύ του υποκατάστατου μοντέλου Kriging και των πραγματικών τιμών των περιορισμών, που μπορεί να οδηγήσουν σε μη εφικτές βέλτιστες λύσεις, αυτή η προσέγγιση ενημερώνει διαδοχικά τα υποκατάστατα μοντέλα Kriging για να διασφαλίσει την εφικτότητα των βελτιστοποιημένων λύσεων.

4.3 Ανάλυση δημοφιλέστερων τεχνικών διαχείρισης περιορισμών

Σε αυτήν την ενότητα περιγράφονται εκτενώς οι πιο συχνά χρησιμοποιούμενες τεχνικές διαχείρισης περιορισμών και επιπλέον κάποιες τεχνικές που έχουν προταθεί προσφάτως και δίνουν αρκετά αισιόδοξα αποτελέσματα.

4.3.1 Μέθοδοι συναρτήσεων ποινής

Σε αυτές τις μεθόδους, οι συντελεστές «ποινής» χρησιμοποιούνται για να μετατρέψουν το πρόβλημα με περιορισμούς σε ένα πρόβλημα χωρίς περιορισμούς, αθροίζοντας την αντικειμενική συνάρτηση με έναν όρο «ποινής». Αυτός ο όρος εξαρτάται από τις συναρτήσεις των περιορισμών και μια παράμετρο που ονομάζεται συντελεστής «ποινής» η τιμή της οποίας μπορεί να αλλάξει ή να παραμείνει ίδια καθ' όλη τη διαδικασία. Σύμφωνα με τον Coello (2002) η γενική διατύπωση μιας συνάρτησης ποινής αποδίδεται ως:

$$F(\mathbf{x}) = f(\mathbf{x}) \pm \left[\sum_{j=1}^J r_j \cdot G_j + \sum_{k=1}^K c_k \cdot L_k \right]$$

όπου $F(\mathbf{x})$ είναι η νέα αντικειμενική συνάρτηση ή συνάρτηση καταλληλότητας (fitness function), G_j και L_k είναι οι συναρτήσεις των περιορισμών ανισότητας $g_j(\mathbf{x})$ και των περιορισμών ισότητας $h_k(\mathbf{x})$, και r_j και c_k είναι θετικές σταθερές που ονομάζονται συντελεστές «ποινής». Μετατρέποντας τους περιορισμούς ισότητας σε αντίστοιχους ανισότητας όπως αναφέρθηκε σε προηγούμενη ενότητα, η παραπάνω εξίσωση ξαναγράφεται ως:

$$F(\mathbf{x}) = f(\mathbf{x}) \pm \sum_{j=1}^m r_j \cdot G_j$$

4.3.1.1 Προσαρμοζόμενη μέθοδος συναρτήσεων ποινής

Ο Kawachi κ.ά. (2019) πρότειναν έναν προσαρμοζόμενο τρόπο υπολογισμού του συντελεστή ποινής κατά τη διάρκεια της εξελικτικής διαδικασίας για να αποφευχθεί μια υπέρ ή υπό ποινολόγηση της αντικειμενικής συνάρτησης καθώς η τιμή του συντελεστή ποινής έχει σημαντική επίδραση στην αναζήτηση και έτσι, μια ακατάλληλη τιμή μπορεί να μην οδηγήσει την αναζήτηση στο καθολικό βέλτιστο. Αρχικά, η συνάρτηση καταλληλότητας ορίζεται ως:

$$F(\mathbf{x}) = f(\mathbf{x}) + PF \cdot \bar{v}(\mathbf{x})$$

όπου $\bar{v}(\mathbf{x})$ είναι η μέση τιμή παραβίασης των περιορισμών και PF είναι ο συντελεστής ποινής. Ακολούθως, απαιτούνται συνολικά τρία βήματα για την εφαρμογή της μεθόδου:

α. Υπολογισμός υποψήφιων συντελεστών ποινής PFCs

Οι υποψήφιοι συντελεστές ποινής (Penalty Factor Candidates – PFCs) χρησιμοποιούνται για τη σύγκριση δύο ατόμων του πληθυσμού και δίνονται από την ακόλουθη εξίσωση:

$$PFC_{k,l} = -\frac{f(\mathbf{x}_k) - f(\mathbf{x}_l)}{\bar{v}(\mathbf{x}_k) - \bar{v}(\mathbf{x}_l)}$$

όπου k και l είναι διαφορετικοί δείκτες. Αυτός ο τρόπος υπολογισμού λαμβάνει υπόψη τις διαφορές μεταξύ των τιμών της αντικειμενικής συνάρτησης και μεταξύ των τιμών παραβίασης των περιορισμών των διανυσμάτων \mathbf{x}_k και \mathbf{x}_l . Οι υποψήφιοι συντελεστές ποινής υπολογίζονται για όλους τους δυνατούς συνδυασμούς «γονέων» P (parents) και «παιδιών» C (children) του πληθυσμού δηλ. του συνόλου PUC συμπεριλαμβανομένων του υφιστάμενου πληθυσμού των γονέων P και των παραγόμενων παιδιών C.

β. Υπολογισμός συντελεστή ποινής PF από τους PFCs

Αν περισσότεροι από τους μισούς υποψήφιους συντελεστές ποινής PFCs δεν έχουν θετική τιμή, τότε ο συντελεστής ποινής PF της προηγούμενης γενιάς είναι και ο πραγματικός συντελεστής της επόμενης. Αν όμως περισσότεροι από τους μισούς υποψήφιους συντελεστές ποινής PFCs έχουν θετική τιμή, τότε ο συντελεστής ποινής PF της επόμενης γενιάς υπολογίζεται από το μέσο όρο αυτών των θετικών τιμών των PFCs.

γ. Εξάρτηση συντελεστή ποινής από την αναλογία εφικτών λύσεων στον πληθυσμό

Με βάση τα προηγούμενα βήματα, όταν αρκετά άτομα του συνόλου PUC του πληθυσμού είναι εφικτά, τότε ο συντελεστής ποινής της επόμενης γενιάς θα είναι πάντα ίδιος με αυτόν της προηγούμενης. Γι' αυτό το λόγο, αν η αναλογία r_f των εφικτών λύσεων στον πληθυσμό ξεπερνάει ένα όριο p_{feas} , τότε η συγκεκριμένη μέθοδος ενημερώνει τον συντελεστή ποινής και τον προσαρμόζει σύμφωνα με την ακόλουθη εξίσωση:

$$PF_{G+1} = p_{rate} \cdot PF_G$$

όπου $p_{rate} \in [0,1]$ και p_{feas} είναι παράμετροι που καθορίζονται από το χρήστη.

4.3.2 Μέθοδος κανόνων εφικτότητας

Στις μεθόδους συνάρτησης ποινής, οι χρήστες συνήθως πρέπει να δοκιμάσουν διαφορετικές τιμές του συντελεστή ποινής για να βρουν τη βέλτιστη λύση, ωστόσο ακατάλληλες τιμές μπορεί να οδηγήσουν την αναζήτηση μακριά από την καθολική βέλτιστη λύση. Για να ξεπεραστεί αυτό το πρόβλημα, ο Deb (2000) πρότεινε αρχικά μια διαφορετική προσέγγιση όπου χρησιμοποιείται ένας τελεστής επιλογής,

προκειμένου να συγκριθούν δύο λύσεις κάθε φορά, σύμφωνα με τα ακόλουθα κριτήρια:

- α. Οποιαδήποτε εφικτή λύση προτιμάται από οποιαδήποτε μη εφικτή λύση.
- β. Μεταξύ δύο εφικτών λύσεων προτιμάται αυτή που έχει καλύτερη τιμή αντικειμενικής συνάρτησης
- γ. Μεταξύ δύο μη εφικτών λύσεων, προτιμάται εκείνη που έχει μικρότερη τιμή παραβίασης περιορισμών.

Σε μαθηματικούς όρους εφαρμόζεται η ακόλουθη συνάρτηση καταλληλότητας:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } g_j(\mathbf{x}) \leq 0 \forall j = 1, 2, \dots, m \\ f_{max} + \sum_{j=1}^m \max\{0, g_j(\mathbf{x})\} & \text{διαφορετικά} \end{cases}$$

Η παράμετρος f_{max} είναι η τιμή της αντικειμενικής συνάρτησης της χειρότερης (μεγαλύτερης εφόσον πρόκειται για πρόβλημα ελαχιστοποίησης) εφικτής λύσης στον πληθυσμό. Αν δεν υπάρχει καμία εφικτή λύση στον πληθυσμό, τότε η f_{max} είναι ίση με το μηδέν. Σε αυτή τη μέθοδο, ο Deb αθροίζει όλες τις τιμές παραβίασης των περιορισμών και συγκρίνει μόνο μία τιμή και οι μη εφικτές λύσεις συγκρίνονται μόνο βάσει της τιμής της παραβίασης των περιορισμών.

4.3.2.1 Τροποποιημένη μέθοδος κανόνων εφικτότητας

Προσφάτως προτάθηκε μία διαφορετική μέθοδος βασισμένη στους κανόνες εφικτότητας του Deb στην οποία έχουν τροποποιηθεί οι κανόνες σύγκρισης που περιγράφονται παραπάνω για να καθοδηγηθεί η κατεύθυνση αναζήτησης προς καλύτερες εφικτές λύσεις και όχι απλά εφικτές, ώστε να επιτευχθεί τελικά μια καθολική εξερεύνηση του χώρου. Έτσι, για να υπολογιστεί η συνάρτηση καταλληλότητας ενός μη εφικτού ατόμου, εισάγεται ο συντελεστής $p_{violation}$ που είναι η κανονικοποιημένη μέγιστη τιμή παραβίασης των περιορισμών του ατόμου πολλαπλασιασμένο επί έναν όρο που λαμβάνει υπόψη τον αριθμό των παραβιασμένων συναρτήσεων περιορισμού μιας λύσης. Αυτός ο συντελεστής ορίζεται ως:

$$p_{violation} = \left\| \max\{\max\{0, g_j(\mathbf{x})\}\} \right\| \times \left(1 + \frac{n_{constviol}}{n_{const}} \right) > 1$$

όπου $n_{constviol}$ είναι ο αριθμός των παραβιασμένων συναρτήσεων περιορισμού και n_{const} είναι ο συνολικός αριθμός των συναρτήσεων περιορισμού. Στη συνέχεια, για τον υπολογισμό της συνάρτησης καταλληλότητας του ατόμου, ο συντελεστής $p_{violation}$ πολλαπλασιάζεται με τη μέγιστη τιμή της αντικειμενικής συνάρτησης

μεταξύ του καλύτερου εφικτού ατόμου που βρέθηκε μέχρι τώρα και του ίδιου του ατόμου. Η συνάρτηση καταλληλότητας διατυπώνεται ως:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } g_j(\mathbf{x}) \leq 0 \forall j = 1, 2, \dots, m \\ \max(f_{best\ feasible}, f(\mathbf{x})) \times p_{violation} & \text{διαφορετικά} \end{cases}$$

Σύμφωνα με τα παραπάνω, για να συγκριθούν δύο λύσεις κάθε φορά, εφαρμόζονται τα ακόλουθα κριτήρια:

α. Μεταξύ δύο εφικτών λύσεων, προτιμάται αυτή που έχει καλύτερη τιμή αντικειμενικής συνάρτησης.

β. Μεταξύ δύο μη εφικτών λύσεων, προτιμάται αυτή που έχει μικρότερη τιμή αντικειμενικής συνάρτησης και παραβίασης περιορισμών, αλλά ταυτόχρονα και μικρότερο αριθμό παραβιασμένων συναρτήσεων περιορισμού.

γ. Μεταξύ μίας εφικτής και μίας μη εφικτής λύσης, η νίκη της εφικτής λύσης εξαρτάται από την τιμή της αντικειμενικής συνάρτησης, την τιμή παραβίασης περιορισμών και τον αριθμό παραβιασμένων συναρτήσεων περιορισμού της μη εφικτής λύσης.

Σύμφωνα με αυτή τη μέθοδο, μια μικρή τιμή παραβίασης περιορισμών δεν είναι το μόνο κριτήριο για την επιλογή ενός ατόμου. Πέρα από τη μέγιστη τιμή παραβίασης περιορισμών, όταν συγκρίνονται δύο μη εφικτά άτομα, λαμβάνεται επίσης υπόψη και η τιμή της αντικειμενικής συνάρτησης αλλά και ο αριθμός των παραβιασμένων συναρτήσεων περιορισμού. Το γινόμενο αυτών δίνει τον τελικό νικητή. Αυτό σημαίνει ότι, εάν ένα άτομο έχει λίγο μικρότερη τιμή παραβίασης περιορισμών από ένα άλλο, αλλά η τιμή της αντικειμενικής συνάρτησής του είναι πολύ μεγαλύτερη, τότε η διατήρηση του πρώτου ατόμου στην επόμενη γενιά μπορεί να οδηγήσει σε ένα τοπικό βέλτιστο. Φυσικά, εάν το άλλο άτομο έχει πολύ μεγάλη τιμή παραβίασης περιορισμών, θα ποινολογηθεί με πολύ μεγαλύτερη τιμή. Προκειμένου να αποφευχθούν πολύ μικρές τιμές αντικειμενικής συνάρτησης που θα μπορούσαν να οδηγήσουν σε λάθος κατεύθυνση αναζήτησης, τίθεται η τιμή της αντικειμενικής συνάρτησης του καλύτερου εφικτού ατόμου που βρέθηκε μέχρι τώρα, ως το κατώτερο όριο. Αντίστοιχα, κατά τη σύγκριση ενός εφικτού και ενός μη εφικτού ατόμου, το δεύτερο θα μπορούσε να επιλεγεί τελικά ανάλογα φυσικά με το γινόμενο των τιμών της αντικειμενικής συνάρτησης και της παραβίασης περιορισμών. Αυτό συμβαίνει επειδή εάν η τιμή του εφικτού ατόμου είναι πολύ μεγάλη, θα μπορούσε να οδηγήσει ξανά σε ένα τοπικό βέλτιστο, οπότε το μη εφικτό άτομο πρέπει να προτιμηθεί. Τέλος, επειδή υπάρχει αβεβαιότητα στο ποια είναι καλύτερη μεταξύ, μιας λύσης με πολλές παραβιασμένες συναρτήσεις περιορισμού που έχουν μικρές τιμές και μιας λύσης με μία μόνο παραβιασμένη συνάρτηση πολύ μεγάλης τιμής όμως, λαμβάνεται υπόψη και ο συνολικός αριθμό παραβιασμένων συναρτήσεων περιορισμού πέρα από τη μέγιστη τιμή παραβίασης.

4.3.3 Μέθοδος ε -περιορισμού

Μια διαφορετική έκδοση των κανόνων εφικτότητας του Deb, προτάθηκε αρχικά από τους Takahama και Sakai (2005) για την παροχή κανόνων σύγκρισης για προβλήματα βελτιστοποίησης με περιορισμούς κατά τη διάρκεια της εξελικτικής διαδικασίας. Σε αυτή τη μέθοδο, χρησιμοποιείται η συνολική παραβίαση περιορισμών $\varphi(\mathbf{x})$ που ορίζεται είτε ως το μέγιστο είτε ως το άθροισμα όλων των περιορισμών και εφαρμόζεται η σύγκριση επιπέδου $\{<_{\varepsilon}\}$, στην οποία για κάθε $\varepsilon \geq 0$, ένα άτομο (f_1, φ_1) θεωρείται καλύτερο από ένα άλλο άτομο (f_2, φ_2) σύμφωνα με την ακόλουθη εξίσωση:

$$(f_1, \varphi_1) <_{\varepsilon} (f_2, \varphi_2) \Leftrightarrow \begin{cases} f_1 < f_2 & \text{if } \varphi_1, \varphi_2 \leq \varepsilon \\ f_1 < f_2 & \text{if } \varphi_1 = \varphi_2 \\ \varphi_1 < \varphi_2 & \text{διαφορετικά} \end{cases}$$

Αυτό σημαίνει ότι:

α. Εάν οι συνολικές τιμές παραβίασης περιορισμών φ δύο ατόμων είναι και οι δύο είτε μικρότερες είτε ίσες με ε , προτιμάται αυτό με τη μικρότερη τιμή αντικειμενικής συνάρτησης f .

β. Εάν οι συνολικές τιμές παραβίασης περιορισμών φ δύο ατόμων είναι ίσες μεταξύ τους, προτιμάται εκείνο με τη μικρότερη τιμή αντικειμενικής συνάρτησης f .

γ. Εάν μόνο η μία ή και οι δύο τιμές παραβίασης περιορισμών φ των δύο ατόμων είναι μεγαλύτερες από το ε , προτιμάται αυτό με τη μικρότερη τιμή παραβίασης περιορισμών φ .

Σύμφωνα με τους Takahama και Sakai (2010) το επίπεδο ε ελέγχεται συνήθως σύμφωνα με την ακόλουθη εξίσωση:

$$\varepsilon(t) = \begin{cases} \varphi_{\theta} & \text{if } t = 0 \\ \varepsilon(0) \left(1 - \frac{t}{T_C}\right)^{c_p} & \text{if } 0 < t < T_C \\ 0 & \text{if } t \geq T_C \end{cases}$$

όπου η αρχική τιμή του ε επιπέδου φ_{θ} είναι η τιμή παραβίασης του θ ατόμου στον αρχικό πληθυσμό, t είναι ο αριθμός των επαναλήψεων, T_C είναι ο μέγιστος αριθμός επαναλήψεων και c_p είναι μια παράμετρος που καθορίζεται από το χρήστη με τιμή όμως μεγαλύτερη ή ίση με 3.

4.3.3.1 Βελτιωμένη μέθοδος ε -περιορισμού

Ο Fan κ.ά. (2018) εφάρμοσαν τη σύγκριση επιπέδου $\{<_{\varepsilon}\}$ για να παρέχουν κανόνες σύγκρισης κατά τη διάρκεια της εξελικτικής διαδικασίας, με τη διαφορά ότι για να εξισορροπήσουν την εξελικτική αναζήτηση του πληθυσμού μεταξύ εφικτών

και μη εφικτών περιοχών, πρότειναν μια βελτιωμένη προσέγγιση υπολογισμού του επιπέδου ε ως εξής:

$$\varepsilon(k) = \begin{cases} \varphi_{\theta} & \text{if } k = 0 \\ \varepsilon(k-1) \left(1 - \frac{FES}{T_C}\right)^{c_p} & \text{if } r_k < \alpha \text{ and } FES < T_C \\ (1 + \tau)\varphi_{max} & \text{if } r_k \geq \alpha \text{ and } FES < T_C \\ 0 & \text{if } FES \geq T_C \end{cases}$$

όπου $\varepsilon(k)$ είναι η τιμή του ε , r_k είναι το ποσοστό των εφικτών λύσεων στη συγκεκριμένη γενιά, φ_{θ} είναι η θ μεγαλύτερη συνολική τιμή παραβίασης όλων των ατόμων στον αρχικό πληθυσμό (ταξινομημένα σε μια σειρά κατά τιμή παραβίασης) όπου $\theta = \gamma \cdot N$ (το N αναπαριστά το μέγεθος του τρέχοντος πληθυσμού) και $\gamma \in [0.2, 0.8]$, φ_{max} είναι η μέγιστη συνολική τιμή παραβίασης όλων των ατόμων, T_C είναι ο μέγιστος επιτρεπόμενος αριθμός υπολογισμών της αντικειμενικής συνάρτησης (Function Evaluations – FEs) όπου $T_C \in [0.1 \max FEs, 0.8 \max FEs]$ και τέλος $c_p \in [2, 10]$, $\tau \in [0, 1]$ και $\alpha \in [0, 1]$ είναι παράμετροι που καθορίζονται από το χρήστη.

4.3.3.2 Τροποποιημένη μέθοδος ε -περιορισμού

Προσφάτως προτάθηκε μία τροποποιημένη μέθοδος ε -περιορισμού, κατά αντιστοιχία με την τροποποιημένη μέθοδο των κανόνων εφικτότητας που αναφέρθηκε νωρίτερα, στην οποία όμως το επίπεδο ε υπολογίζεται ως εξής:

$$(f_1, p_1) <_{\varepsilon} (f_2, p_2) \Leftrightarrow \begin{cases} f_1 < f_2 & \text{if } p_1, p_2 \leq \varepsilon \\ f_1 < f_2 & \text{if } p_1 = p_2 \\ \max(f_{bf}, f_1) \times p_1 < \max(f_{bf}, f_2) \times p_2 & \text{if } p_1, p_2 \geq \varepsilon \\ f_1 < \max(f_{bf}, f_2) \times p_2 & \text{διαφορετικά} \end{cases}$$

όπου f_{bf} είναι η $f_{best\ feasible}$ και p_1, p_2 είναι οι συντελεστές $p_{violation_1}, p_{violation_2}$ αντίστοιχα.

Αυτό σημαίνει ότι:

α. Εάν οι μέγιστες τιμές παραβίασης περιορισμών δύο ατόμων είναι και οι δύο μικρότερες ή ίσες με ε , προτιμάται αυτή με τη μικρότερη τιμή αντικειμενικής συνάρτησης f .

β. Εάν οι μέγιστες τιμές παραβίασης περιορισμών δύο ατόμων είναι ίσες, προτιμάται αυτή με τη μικρότερη τιμή αντικειμενικής συνάρτησης f .

γ. Εάν και οι δύο μέγιστες τιμές παραβίασης περιορισμών δύο ατόμων είναι μεγαλύτερες από ε , προτιμάται αυτή με το μικρότερο γινόμενο τιμής παραβίασης περιορισμών και τιμής αντικειμενικής συνάρτησης.

δ. Εάν μόνο η μία μέγιστη τιμή παραβίασης περιορισμών δύο ατόμων είναι μεγαλύτερη από το ε , τότε προτιμάται το άλλο άτομο μόνο εάν η τιμή της αντικειμενικής του συνάρτησης είναι μικρότερη από το γινόμενο της τιμής παραβίασης περιορισμών και της τιμής αντικειμενικής συνάρτησης του πρώτου. Όπως και στην προηγούμενη προσέγγιση, πέρα από τη μέγιστη τιμή παραβίασης περιορισμών, λαμβάνεται επίσης υπόψη και ο συνολικός αριθμός των παραβιασμένων συναρτήσεων περιορισμού.

Τέλος, σε αυτή τη μέθοδο η τιμή του επιπέδου ε υπολογίζεται από τις αντίστοιχες σχέσεις που προτάθηκαν από τον Fan κ.ά. (2018) που αναφέρθηκαν νωρίτερα.

4.3.4 Συνονθύλευμα τεχνικών διαχείρισης περιορισμών

Δεν είναι πάντα εύκολο να προσδιοριστεί ποια μέθοδος διαχείρισης περιορισμών είναι καλύτερη στη χρήση, προκειμένου να εντοπιστεί η καθολική βέλτιστη λύση σε ένα δεδομένο πρόβλημα. Η χρήση μιας μοναδικής τεχνικής διαχείρισης περιορισμών μπορεί να είναι κατάλληλη για ένα συγκεκριμένο βήμα της διαδικασίας εξέλιξης αλλά ακατάλληλη για άλλο βήμα. Με άλλα λόγια, διαφορετικές μέθοδοι διαχείρισης περιορισμών μπορεί να είναι πιο αποτελεσματικές σε διαφορετικά στάδια της διαδικασίας αναζήτησης. Αυτό εξαρτάται από διάφορους παράγοντες όπως η πολυτροπικότητα του προβλήματος, η αναλογία μεταξύ του εφικτού χώρου αναζήτησης και ολόκληρου του χώρου αναζήτησης, ο επιλεγμένος αλγόριθμος καθώς και τα στάδια της καθολικής εξερεύνησης / τοπικής εκμετάλλευσης της διαδικασίας αναζήτησης.

Με κίνητρο αυτές τις παρατηρήσεις και όπως αναφέρθηκε στην προηγούμενη ενότητα, οι Mallipeddi και Suganthan (2010) πρότειναν ένα συνονθύλευμα τεχνικών διαχείρισης περιορισμών (ensemble of constraint handling techniques – ECHT) για την επίλυση προβλημάτων βελτιστοποίησης πραγματικών παραμέτρων με περιορισμούς, όπου κάθε μέθοδος διαχείρισης περιορισμών έχει τον δικό της πληθυσμό. Παρουσίασαν ένα ECHT με τέσσερις τεχνικές διαχείρισης περιορισμών (ανωτερότητα των εφικτών λύσεων, αυτοπροσαρμοζόμενη ποινή, μέθοδος ε-περιορισμού και στοχαστική κατάταξη) όπου η καθεμία έχει τον δικό της πληθυσμό και τις δικές τις παραμέτρους. Κάθε πληθυσμός που αντιστοιχεί σε μια μέθοδο διαχείρισης περιορισμών παράγει τους απογόνους του και τους αξιολογεί. Ο πληθυσμός των «γονέων» που αντιστοιχεί σε μια συγκεκριμένη μέθοδο διαχείρισης περιορισμών δεν ανταγωνίζεται μόνο τον δικό του πληθυσμό των «απογόνων» αλλά και τον πληθυσμό των απογόνων των άλλων τριών μεθόδων διαχείρισης περιορισμών. Λόγω αυτού, ένας απόγονος που παράγεται από μια συγκεκριμένη μέθοδο διαχείρισης περιορισμών μπορεί να απορριφθεί από τον δικό του πληθυσμό, αλλά θα μπορούσε να γίνει αποδεκτός από τους πληθυσμούς άλλων μεθόδων διαχείρισης περιορισμών.

4.3.5 Μέθοδος στοχαστικής κατάταξης

Μια άλλη πολύ συχνά χρησιμοποιούμενη μέθοδος για τη διαχείριση των περιορισμών είναι η μέθοδος της στοχαστικής κατάταξης (Stochastic Ranking) που προτάθηκε αρχικά από τους Runarsson και Yao (2000). Παρατηρώντας ότι οι τεχνικές διαχείρισης περιορισμών, όπως για παράδειγμα οι τεχνικές των συναρτήσεων ποινής, μπορεί να προκαλέσουν υπό ή υπέρ ποινολόγηση εάν ο σχεδιαστής επιλέξει μια ακατάλληλη τιμή για τον συντελεστή ποινής, διαπίστωσαν ότι πρέπει να εφαρμοστεί μια μέθοδος που θα μπορούσε να εξασφαλίσει τη σωστή ισορροπία μεταξύ της διατήρησης εφικτών ατόμων και της απόρριψης μη εφικτών. Έτσι, στη στοχαστική κατάταξη ο σκοπός είναι να συγκριθούν όλα τα γειτονικά άτομα, από την άποψη της κυριαρχίας είτε των συναρτήσεων ποινής ή της αντικειμενικής συνάρτησης, προκειμένου να ταξινομηθούν όλα τα άτομα ενός πληθυσμού από το καλύτερο στο χειρότερο.

Στη συγκεκριμένη μέθοδο, αρχικά, πρέπει να οριστεί από το χρήστη η πιθανότητα P_f να χρησιμοποιηθεί μόνο η αντικειμενική συνάρτηση για συγκρίσεις στη διάρκεια της κατάταξης στις μη εφικτές περιοχές του χώρου αναζήτησης. Αυτή η πιθανότητα παίρνει τιμές μεταξύ 0 και 1 ώστε να επιτευχθεί η καλύτερη απόδοση. Επιλέγοντας $P_f = 1/2$, σημαίνει ότι η πιθανότητα μιας σύγκρισης να γίνει με βάση την αντικειμενική συνάρτηση ή τη συνάρτηση ποινής θα είναι ίδια. Στη συνέχεια, η κατάταξη επιτυγχάνεται με μια διαδικασία τύπου ταξινόμησης φυσαλίδας. Τα άτομα λ ταξινομούνται συγκρίνοντας γειτονικά άτομα N σε αρκετές επαναλήψεις (τουλάχιστον λ) σε κάθε γενιά. Εάν και τα δύο γειτονικά άτομα είναι εφικτά ή ένας τυχαίος αριθμός u είναι μικρότερος από την επιλεγμένη πιθανότητα P_f , τότε τα άτομα συγκρίνονται σύμφωνα με τις τιμές της αντικειμενικής συνάρτησης. Διαφορετικά, εάν ένα ή και τα δύο άτομα είναι μη εφικτά, τότε η σύγκριση βασίζεται στην τιμή παραβίασης των περιορισμών. Η όλη διαδικασία τερματίζεται εάν δεν υπάρξει καμία αλλαγή στη σειρά κατάταξης σε μια πλήρη επανάληψη.

5 ΕΦΑΡΜΟΓΗ ΑΛΓΟΡΙΘΜΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

5.1 Γενικά

Στην παρούσα εργασία μελετήθηκαν διεξοδικά τα αποτελέσματα βελτιστοποίησης έξι συγκεκριμένων μοντέλων, μετά την εφαρμογή διάφορων αλγορίθμων βελτιστοποίησης. Τα κατασκευαστικά μοντέλα έχουν σχεδιαστεί και αναλυθεί στο πρόγραμμα κατασκευών SAP2000 v.20 της CSI. Η πλατφόρμα βελτιστοποίησης High Performance Optimization Computing Platform HP-OCP έχει σχεδιαστεί έτσι ώστε να είναι συμβατή και να μπορεί να λειτουργεί ταυτόχρονα με τα προγράμματα της CSI (SAP2000 και ETABS). Μέσω του HP-OCP, ο χρήστης επιλέγει το κατασκευαστικό μοντέλο που θέλει να βελτιστοποιήσει επιλέγοντας τις ακόλουθες επιθυμητές παραμέτρους, όπως:

- Αντικειμενική συνάρτηση (Objective function)
- Περιορισμούς (Constraints)
- Κριτήρια Σύγκλισης (Convergence Criteria)
- Όρια Διατομών (Bounds of Section Property Dimensions)

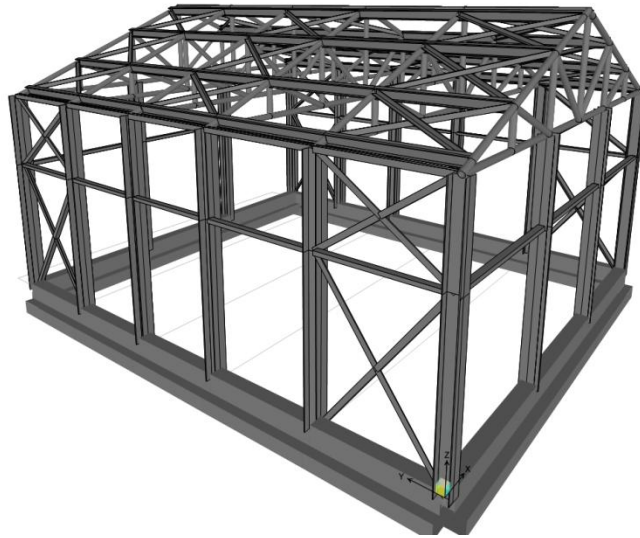
Έτσι, ανάλογα με τον αλγόριθμο που επιλέγεται κάθε φορά, το HP-OCP προσπαθεί να υπολογίσει την βέλτιστη εφικτή λύση μέσω επαναληπτικής διαδικασίας, σύμφωνα με τις παραμέτρους που έχει καθορίσει ο χρήστης, ώστε τελικά να μεγιστοποιήσει ή να ελαχιστοποιήσει την αντικειμενική συνάρτηση. Πρακτικά, το HP-OCP μεταβάλλει σε κάθε επανάληψη (δηλ. σε κάθε ανάλυση του μοντέλου που γίνεται μέσω του SAP2000) τις τιμές των μεταβλητών σχεδιασμού, όπως αυτές προκύπτουν από τον αλγόριθμο που χρησιμοποιείται, προκειμένου να βρει εκείνες τις τιμές που όταν εφαρμοστούν στο μοντέλο θα βελτιστοποιήσουν την αντικειμενική συνάρτηση.

Σε όλες τις περιπτώσεις που μελετήθηκαν, η αντικειμενική συνάρτηση είναι η ελαχιστοποίηση του βάρους της κατασκευής και οι περιορισμοί που έχουν τεθεί είναι οι σχεδιαστικοί (Design Violations) δηλ. οι κατασκευαστικοί και σεισμικοί κώδικες (EC-2, EC-3, EC-8, κτλ.).

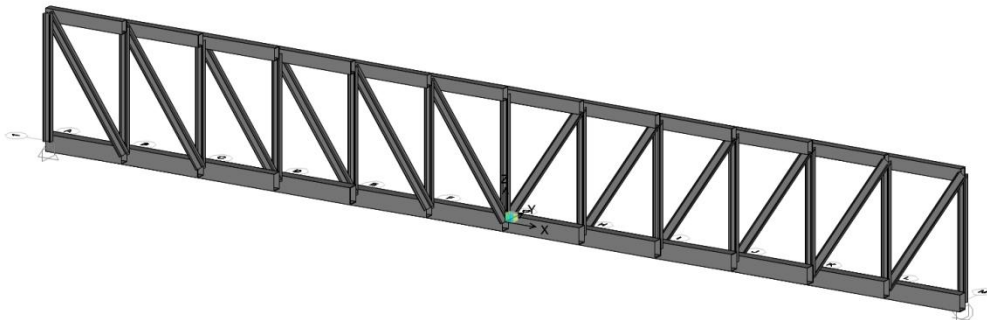
5.2 Τύποι μοντέλων μελέτης

Τα κατασκευαστικά μοντέλα που χρησιμοποιήθηκαν στο πλαίσιο της παρούσης διπλωματικής εργασίας, τα χαρακτηριστικά των οποίων αναλύονται διεξοδικότερα σε επόμενα κεφάλαια, είναι τα ακόλουθα:

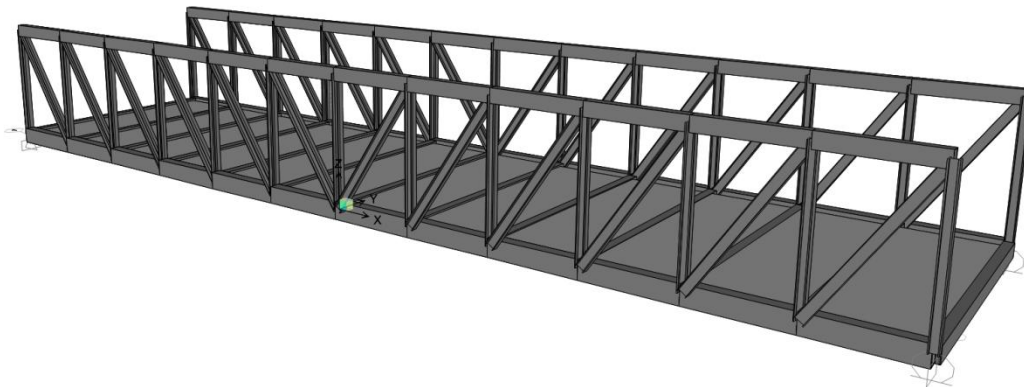
α. Ένας αχυρώνας από χαλύβδινα στοιχεία διαφόρων διατομών και θεμελίωση από οπλισμένο σκυρόδεμα.



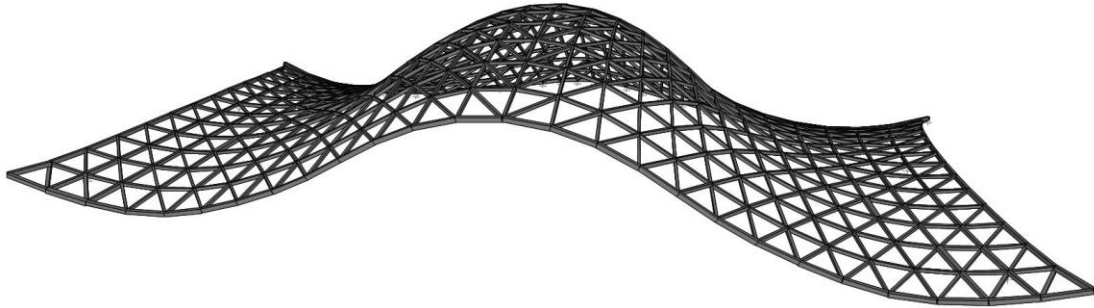
β. Μία 2D δικτυωτή γέφυρα με μεταλλικά στοιχεία διαφόρων διατομών.



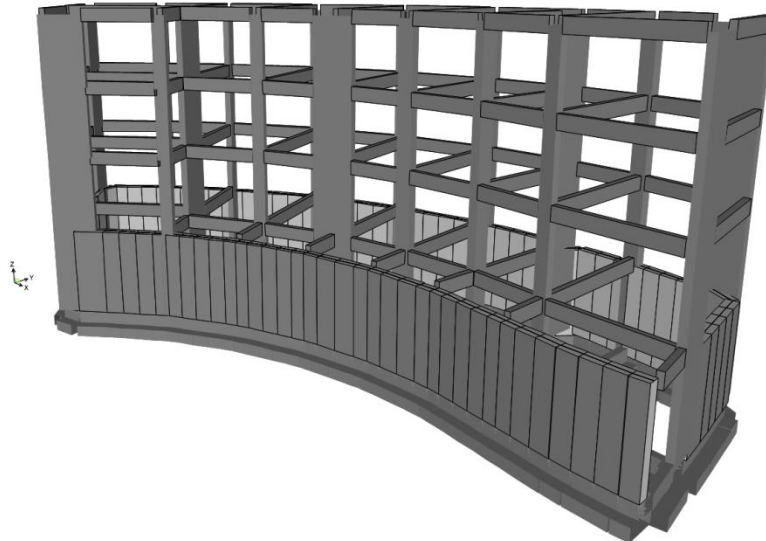
γ. Μία 3D γέφυρα κάτω διάβασης, με χαλύβδινο κατάστρωμα και χαλύβδινες δικτυωτές κύριες δοκούς.



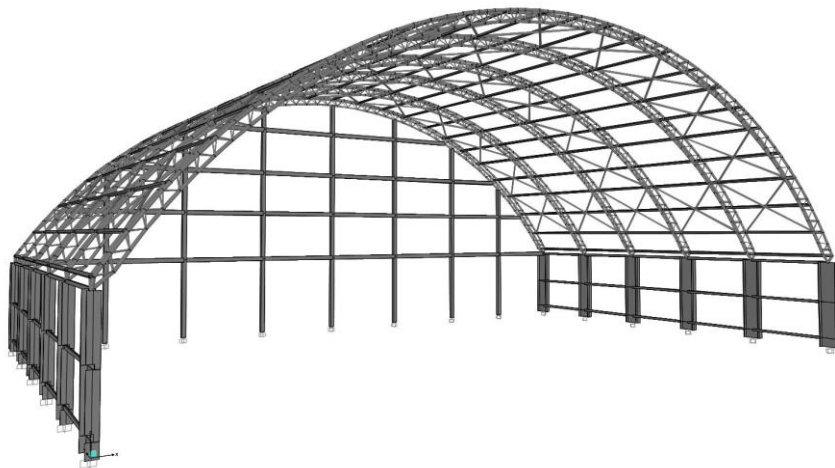
δ. Ένα στέγαστρο σχεδιασμένο ως χωροδικτύωμα με χαλύβδινα στοιχεία διαφόρων διατομών.



ε. Ένα σχολείο κατασκευασμένο από οπλισμένο σκυρόδεμα.



στ. Ένα υπόστεγο από χαλύβδινα στοιχεία διαφόρων διατομών.



5.3 Αλγόριθμοι βελτιστοποίησης μελέτης

Για τη μελέτη των παραπάνω μοντέλων χρησιμοποιήθηκαν οι ακόλουθοι αλγόριθμοι βελτιστοποίησης:

- PQN
- MATSuMoTo
- Latin Hypercube

Πέραν των ανωτέρω αλγορίθμων και για λόγους πληρότητας, περιγράφεται στη συνέχεια της εργασίας και η λειτουργία δύο ακόμα μετα-ευρεστικών αλγορίθμων που ανήκουν στους Εξελικτικούς Αλγορίθμους (evolution intelligence) και στους Αλγορίθμους Σμηνών (swarm intelligence) αντίστοιχα:

- Differential Evolution
- Pity Beetle Algorithm

Τα αποτελέσματα που προέκυψαν και τα οποία θα αναλυθούν διεξοδικά στη συνέχεια αποδεικνύουν την ντετερμινιστική ή στοχαστική φύση του κάθε αλγορίθμου βελτιστοποίησης. Διαπιστώθηκε ότι, κάθε αλγόριθμος συμπεριφέρεται τελείως διαφορετικά ανάλογα με τα χαρακτηριστικά της κάθε κατασκευής. Έτσι, κάποιοι αλγόριθμοι έδωσαν εξαιρετικά αποτελέσματα όταν εφαρμόστηκαν σε ένα συγκεκριμένο δομικό σύστημα αλλά οι ίδιοι αλγόριθμοι, σε άλλη κατασκευή, είτε εμφάνισαν «κακή» επίδοση είτε δεν κατάφεραν ποτέ να βρουν εφικτές λύσεις κάτω από τους τιθέμενους περιορισμούς της αντικειμενικής συνάρτησης.

5.3.1 Αλγόριθμος PQN

Ο αλγόριθμος αυτός είναι μια εφαρμογή της μεθόδου Implicit filtering, η οποία αποτελεί μία derivative-free μέθοδο βελτιστοποίησης που δεν κάνει άμεση χρήση «παραγώνων» και έχει αναπτυχθεί τα τελευταία χρόνια. Βασίζεται σε ένα συνδυασμό των γνωστών αλγορίθμων Quasi-Newton που χρησιμοποιούνται για βελτιστοποίηση προβλημάτων με περιορισμούς (bound constraints), των μη-γραμμικών προβλημάτων ελάχιστων τετραγώνων και ενός ντετερμινιστικού αλγορίθμου αναζήτησης. Τα διαφορικά όμως που απαιτούνται στον αλγόριθμο Quasi-Newton, εδώ προσεγγίζονται με πεπερασμένες διαφορές.

Η Implicit filtering είναι κατά βάση μια ντετερμινιστική μέθοδος δειγματοληψίας για βελτιστοποίηση προβλημάτων με οριακές συνθήκες (bound constrained optimization). Οι μέθοδοι δειγματοληψίας, γενικά, ελέγχουν την πρόοδο της βελτιστοποίησης αξιολογώντας τις εφικτές λύσεις της αντικειμενικής συνάρτησης, χωρίς τη χρήση «διαφορικών» (gradients). Σε κάποιες μεθόδους όμως, όπως στη συγκεκριμένη μέθοδο, γίνεται «εκτίμηση» και υπολογισμός της παραγώγου

ή/ και άλλων απαιτούμενων πληροφοριών από τη δειγματοληψία και γι' αυτό συγκαταλέγεται στις derivative-free μεθόδους.

Μέθοδοι όπως η Implicit filtering έχουν σχεδιαστεί για να ελαχιστοποιούν αντικειμενικές συναρτήσεις που είναι μη ομαλές, πιθανώς ασυνεχείς ή στοχαστικές και οι οποίες μπορεί να μην ορίζονται καν σε όλα τα σημεία του χώρου. Δεν είναι μέθοδοι για «ομαλά» προβλήματα όπου η συνάρτηση που θα ελαχιστοποιηθεί και οι περιορισμοί εκφράζονται σε όρους παραγωγίσιμων συναρτήσεων. Γιατί σε αυτή την περίπτωση υπάρχουν άλλες αποτελεσματικές, αυστηρά ντετερμινιστικές μέθοδοι, που βασίζονται στη χρήση παραγώγων, όπως η μέθοδος μεγίστης καθόδου (Steepest Descent) κτλ. Επίσης, μέθοδοι όπως η Implicit filtering, μπορεί να επιφέρουν σημαντικά οφέλη και σε περιπτώσεις όπου η χρήση άλλων μαθηματικών μεθόδων απαιτούν μεγάλο υπολογιστικό «κόστος» προκειμένου να δημιουργήσουν τις παραγώγους της αντικειμενικής συνάρτησης.

Η Implicit filtering θεωρείται επέκταση της μεθόδου αναζήτησης συντεταγμένων (coordinate search algorithm), την απλούστερη δυνατή προσέγγιση στη βελτιστοποίηση. Διαφέρει όμως από αυτήν, καθώς η Implicit filtering δημιουργεί ένα τοπικό μοντέλο της αντικειμενικής συνάρτησης χρησιμοποιώντας τη μέθοδο Quasi-Newton. Εν ολίγοις, η Implicit filtering επεκτείνει την μέθοδο αναζήτησης συντεταγμένων προσεγγίζοντας ένα διαφορικό με χρήση της μεθόδου παρεμβολής των ελαχίστων τετραγώνων, στη συνέχεια χρησιμοποιεί αυτήν την κατά προσέγγιση «παράγωγο» για να δημιουργήσει ένα νέο απλούστερο μοντέλο και έπειτα αναζητά μία καλύτερη λύση αντλώντας πλέον πληροφορίες από αυτό το μοντέλο. Πρακτικά, εκμεταλλεύεται τα πλεονεκτήματα αφενός των απλών μεθόδων δειγματοληψίας για μη ομαλά προβλήματα ή προβλήματα με ασυνέχειες (derivative-free μέθοδοι) και αφετέρου άλλων αποτελεσματικών μαθηματικών μεθόδων (gradient-based) οι οποίες χρησιμοποιούνται μόνο σε ομαλά και συνεχή προβλήματα και αποφέρουν γρήγορη σύγκλιση.

Η Implicit filtering είναι σχεδιασμένη για να επιλύει και να βελτιστοποιεί προβλήματα με περιορισμούς (bound constraint problems). Η μαθηματική διατύπωση των προβλημάτων αυτών εκφράζεται από την ακόλουθη εξίσωση:

$$\min_{x \in \Omega} f(x) \text{ όπου } \Omega = \{x \in R^N, L_i \leq (x)_i \leq U_i\}$$

Στην παραπάνω εξίσωση τα L_i , U_i είναι το πεπερασμένο άνω και κάτω όριο του i -στού στοιχείου $(x)_i$, του διανύσματος της μεταβλητής x και N ο αριθμός των μεταβλητών σχεδιασμού. Τα βασικά στάδια του αλγορίθμου αναλύονται παρακάτω:

α. Διαδικασία δειγματοληψίας

Ο αλγόριθμος ξεκινάει τη λειτουργία του με τη λήψη δειγμάτων, από τα οποία υπολογίζει την τιμή της αντικειμενικής συνάρτησης και στη συνέχεια κατασκευάζει ένα πίνακα (stencil) ώστε αργότερα να τον χρησιμοποιήσει για διάφορες διεργασίες,

όπως για παράδειγμα στον υπολογισμό των απαιτούμενων διαφορικών που θα χρησιμοποιηθούν από τη μέθοδο Quasi-Newton.

Αν θεωρήσουμε ως x_c την μεταβλητή μιας τυχαίας επανάληψης και $f(x_c)$ την αντίστοιχη τιμή της αντικειμενικής συνάρτησης, τότε ο αλγόριθμος προεπιλέγει $2N$ δείγματα $x_c \pm h\nu_i$, όπου $\nu_i = (L_i - U_i)e_i$ με $1 \leq \nu_i \leq N$. Ο όρος e_i αναφέρεται στο i -στο στοιχείο του μοναδιαίου διανύσματος και ο όρος h σε μια παράμετρο κλίμακας με $h = (2^{-n})_{n=start}^{depth}$, όπου η τιμή του h μειώνεται συνεχώς κατά τη διάρκεια της βελτιστοποίησης μέχρι να βρεθεί η βέλτιστη τιμή της $f(x)$. Οι τιμές που λαμβάνει το n (και άρα και το h) επιλέγονται από τον χρήστη κάθε φορά. Εάν ολοκληρωθεί η αλληλουχία της κλιμάκωσης δηλ. εξαντληθούν οι τιμές του h , τότε η διαδικασία βελτιστοποίησης τερματίζεται.

β. Προσέγγιση παραγώγων μέσω πεπερασμένων διαφορών

Με βάση τον προηγούμενο πίνακα που περιέχει τις τιμές της $f(x)$, σε κάθε κλιμάκωση h , υπολογίζεται ένας προσεγγιστικός πίνακας παραγώγων (stencil gradient) μέσω πεπερασμένων διαφορών και ο οποίος χρησιμοποιείται αργότερα για να υπολογιστεί η νέα κατεύθυνση στην οποία θα ψάξει ο αλγόριθμος για καλύτερες λύσεις.

Ο πίνακας παραγώγων είναι η λύση του γραμμικού προβλήματος ελαχίστων τετραγώνων $\min_{y \in \mathbb{R}^N} \|hV^T y - \delta(f, x, V, h)^T\|$ και ορίζεται ως $\nabla f(x, V, h) = Df(x, V, h)^T$.

V είναι ο πίνακας κατεύθυνσης με $V = (\nu_1, \nu_2, \dots, \nu_{2N})$

$Df(x, V, h)$ είναι ο Ιακωβιανός πίνακας, ο οποίος είναι ένα διάνυσμα – γραμμή και υπολογίζεται από τη σχέση $Df(x, V, h) = \frac{1}{h} \delta(f, x, V, h)V^+$, όπου

$$\delta(f, x, V, h) = \begin{pmatrix} f(x + h\nu_1) - f(x) \\ f(x + h\nu_2) - f(x) \\ \dots \\ f(x + h\nu_k) - f(x) \end{pmatrix}^T \text{ και } V^+ \text{ είναι ο ψευδο-αντίθετος πίνακας του } V.$$

γ. Υπολογισμός νέας κατεύθυνσης αναζήτησης καλύτερης λύσης

Η νέα κατεύθυνση προς την οποία πρέπει να κινηθεί ο αλγόριθμος υπολογίζεται από τη μέθοδο projected Quasi-Newton μέσω της κατασκευής ενός προσεγγιστικού πίνακα Hess, ο οποίος περιλαμβάνει πληροφορίες δεύτερης παραγώγου και σε κάθε επανάληψη ενημερώνεται (είτε μέσω της μεθόδου BFGS είτε της SR1) ώστε να υπολογίζεται κάθε φορά η νέα κατεύθυνση.

Η κατεύθυνση στη μέθοδο projected Quasi-Newton ορίζεται ως:

$$d = -R^{-1} \nabla f(x, V, h)$$

όπου R είναι ένας πίνακας που προσεγγίζει τον αντίστροφο του πίνακα Hess και ισχύει $R = P_{B^\epsilon}(x) + (I - P_{B^\epsilon}(x))H(I - P_{B^\epsilon}(x))$. Με τον τελεστή P ορίζεται η προβολή (projection) του συνόλου B^ϵ στο Ω με $B^\epsilon = B^\epsilon(x) = \{i \mid U_i - (x)_i \leq \epsilon \text{ και } \partial f(x) / \partial (x)_i \leq -\sqrt{\epsilon}\} \cup \{i \mid (x)_i - L_i \leq \epsilon \text{ και } \partial f(x) / \partial (x)_i \geq \sqrt{\epsilon}\}$ όπου $\epsilon = 10^{-6}$. Ο αρχικός πίνακας R τίθεται ίσος με τον μοναδιαίο πίνακα I και στη συνέχεια ενημερώνεται είτε μέσω της μεθόδου BFGS (Broyden – Fletcher – Goldfarb – Shanno) είτε της Symmetric Rank – One SR1

δ. Περιορισμοί

Οι περιορισμοί στις τιμές που μπορεί να λάβει η μεταβλητή x δημιουργούν πρόβλημα στη διαδικασία βελτιστοποίησης της αντικειμενικής συνάρτησης. Αυτό συμβαίνει γιατί, αν κατά τη διαδικασία αναζήτησης καλύτερης λύσης βρεθεί κάποια τιμή η οποία είναι εκτός του πεδίου ορισμού του Ω , τότε θα δώσει μια τιμή αντικειμενικής συνάρτησης που στην ουσία θα είναι μη εφικτή (infeasible). Όμως τόσο οι πίνακες των αντικειμενικών τιμών όσο και οι πίνακες των παραγώγων, σε όλη τη διάρκεια βελτιστοποίησης λαμβάνουν απλά τιμές, οπότε πρέπει να ενημερώνονται συνεχώς για τις μη εφικτές, ώστε να μην τις λαμβάνουν υπόψη. Έτσι, γίνεται ειδική διαχείριση των περιορισμών και ο αλγόριθμος ελέγχει όλες τις λύσεις, ώστε οι τιμές που είναι μη εφικτές να μη γίνονται αποδεκτές και έτσι η κατεύθυνση αναζήτησης λύσεων να είναι πάντα εντός του πεδίου εφικτών τιμών.

Μια απλή μορφή εφαρμογής της μεθόδου, όπου περιλαμβάνονται όλα τα παραπάνω στάδια, φαίνεται παρακάτω:

Pick initial x and h ; find $f(x)$ and the Difference Gradient $\nabla_h f(x)$.

Initialize the model Hessian B to the identity

while h and $\nabla_h f(x)$ satisfy conditions **do**

 Use $\nabla_h f(x)$ and B to calculate a descent direction d . This step is a quasi-Newton step.

 Perform a line search in the direction d , and signal success if some criteria are met.

if line search was successful **then**

 Accept new point and project into the box Q .

else

$h \leftarrow h/2$

end if

 Calculate the Difference Gradient $\nabla_h f(x)$.

 Update B with either a rank-one SR1 update, or a rank-two BFGS update.

end while

ε. Κριτήρια τερματισμού διαδικασίας βελτιστοποίησης (termination criteria)

Στην Implicit filtering γίνονται συνεχώς δύο ειδών επαναληπτικές διαδικασίες μία εξωτερική και μία εσωτερική. Η εξωτερική επανάληψη απλά ελέγχει την τιμή του h και τον συνολικό αριθμό των υπολογισμών της αντικειμενικής συνάρτησης f που έχουν γίνει. Η εσωτερική επανάληψη (ή Quasi-Newton επανάληψη), η οποία εκτελείται για κάθε τιμή της παραμέτρου κλίμακας h , είναι αυτή στην οποία ενημερώνονται οι πίνακες και υπολογίζεται η νέα κατεύθυνση σύμφωνα με τη μέθοδο Quasi-Newton.

Ως εκ τούτου, η εξωτερική επανάληψη θα σταματήσει όταν:

- ο συνολικός αριθμός (budget) των υπολογισμών της αντικειμενικής συνάρτησης f ξεπεράσει κάποιο όριο που έχει προκαθοριστεί ή
- ολοκληρωθεί η αλληλουχία της κλιμάκωσης δηλ. εξαντληθούν οι τιμές του h .

Αντίστοιχα, η εσωτερική επανάληψη θα σταματήσει όταν:

- η τιμή της f σε κάποιο σημείο είναι μικρότερη από οποιοδήποτε στοιχείο μέσα στον πίνακα πεπερασμένων διαφορών (stencil failure) ή
- ικανοποιηθούν τα εσωτερικά κριτήρια της κάθε επανάληψης της μεθόδου Quasi-Newton.

Τέλος, υπάρχει δυνατότητα να τεθούν και άλλα κριτήρια τερματισμού από τον χρήστη, κριτήρια όπως συγκεκριμένη τιμή – στόχος της αντικειμενικής συνάρτησης f , κ.α.

5.3.2 Αλγόριθμος MATSuMoTo

Ο αλγόριθμος MATSuMoTo (MATLAB Surrogate Model Toolbox) είναι μία μέθοδος βελτιστοποίησης για προβλήματα τα οποία απαιτούν σημαντικό υπολογιστικό κόστος επίλυσης, των οποίων η αντικειμενική συνάρτηση δεν είναι παραγωγίσιμη (derivative-free) και το πεδίο ορισμού των μεταβλητών σχεδιασμού λαμβάνει είτε συνεχείς ή ακέραιες τιμές ή συνδυασμό αυτών. Η μαθηματική διατύπωση των προβλημάτων αυτών εκφράζεται, όπως και στη μέθοδο Implicit filtering από την ακόλουθη εξίσωση:

$$\min_{x \in \Omega} f(x) \text{ όπου } \Omega = \{x \in R^N, L_i \leq (x)_i \leq U_i\}$$

Για την επίλυση των παραπάνω προβλημάτων, ο αλγόριθμος MATSuMoTo χρησιμοποιεί υποκατάστατα μοντέλα (surrogate models) της αρχικής αντικειμενικής συνάρτησης για την εύρεση βέλτιστων λύσεων, δηλαδή απλούστερες συναρτήσεις που απαιτούν έναν πολύ περιορισμένο αριθμό υπολογισμών. Έτσι, η αντικειμενική συνάρτηση $f(x)$ εκφράζεται ως $f(x)=s(x)+e(x)$, όπου $s(x)$ είναι η πρόβλεψη του

υποκατάστατου (surrogate) μοντέλου και $e(x)$ η διαφορά των δύο συναρτήσεων. Πρακτικά, η αρχική περίπλοκη (άρα και υπολογιστικά «ακριβή») αντικειμενική συνάρτηση διασπάται σε δύο απλούστερες συναρτήσεις. Η μία προσομοιώνει το υποκατάστατο μοντέλο που απαιτεί μικρότερο υπολογιστικό κόστος επίλυσης και η δεύτερη αφορά τη συνάρτηση διαφοράς μεταξύ της $f(x)$ και του υποκατάστατου μοντέλου. Με αυτό τον τρόπο, η αναζήτηση καλύτερων λύσεων γίνεται με βάση το υποκατάστατο μοντέλο, ενώ η $f(x)$ χρησιμοποιείται πλέον μόνο για τον υπολογισμό των τιμών των συγκεκριμένων σημείων που έχουν προκύψει από το υποκατάστατο μοντέλο.

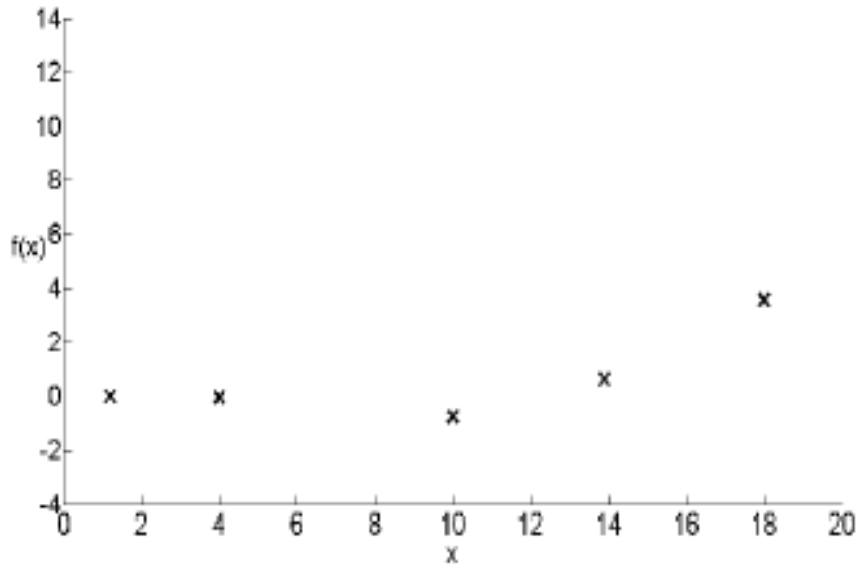
Η προσέγγιση αυτή, δηλαδή η χρήση υποκατάστατου μοντέλου, απαιτεί σε σύγκριση με αλγόριθμους που λειτουργούν με βάση τον «πληθυσμό» υποψήφιων λύσεων (όπως για παράδειγμα οι γενετικοί αλγόριθμοι) σημαντικά λιγότερους υπολογισμούς και επομένως είναι καταλληλότερη για προβλήματα που έχουν υψηλό υπολογιστικό κόστος.

Ο αλγόριθμος MATSuMoTo ακολουθεί τα παρακάτω βήματα προκειμένου να επιλύσει ένα πρόβλημα:

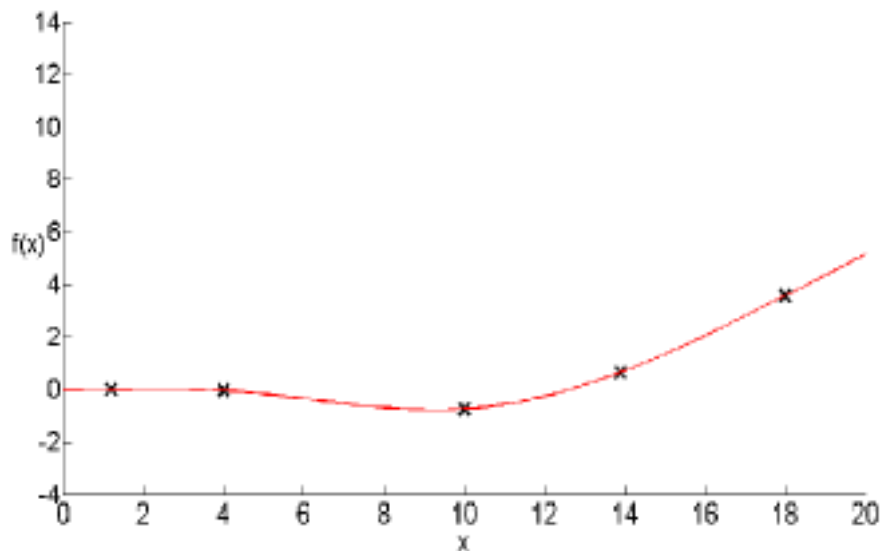
1. Δημιουργία ενός αρχικού μοντέλου με χρήση διάφορων μεθόδων όπως για παράδειγμα η μέθοδος δειγματοληψίας Latin hypercube.
2. Υπολογισμός τιμών αρχικής αντικειμενικής συνάρτησης για τα επιλεγμένα σημεία δειγματοληψίας.
3. Κατασκευή υποκατάστατου μοντέλου (surrogate model) δηλ. μιας απλούστερης συνάρτησης που προσαρμόζεται και ικανοποιεί τα παραπάνω σημεία, με χρήση μεθόδων όπως η Radial Basis Function (RBF), Polynomial Regression Models και η Multivariate Adaptive Regression Splines (MARS).
4. Επιλογή νέων σημείων από το πεδίο ορισμού με διάφορες στρατηγικές δειγματοληψίας (π.χ. ελάχιστο σημείο του υποκατάστατου μοντέλου, τοπική αναζήτηση υποψήφιου σημείου κοντά στο μέχρι στιγμής βέλτιστο, κ.α.) και με χρήση πληροφορίας πλέον από το νέο (απλούστερο) υποκατάστατο μοντέλο.
5. Υπολογισμός τιμών αρχικής αντικειμενικής συνάρτησης για τα νέα επιλεγμένα σημεία.
6. Έλεγχος κριτηρίων τερματισμού της διαδικασίας (stopping criterion), κριτήρια όπως μέγιστος αριθμός υπολογισμού τιμών αντικειμενικής συνάρτησης, ή μέγιστος αριθμός επαναλήψεων, κ.α.

Εφόσον τα κριτήρια τερματισμού δεν έχουν επιτευχθεί, ο αλγόριθμος κάνει μια επαναληπτική διαδικασία στα βήματα 3 έως 6, οπότε γίνεται εκ νέου προσαρμογή του υποκατάστατου μοντέλου (surrogate model) στα νέα επιλεγμένα σημεία κ.ο.κ.

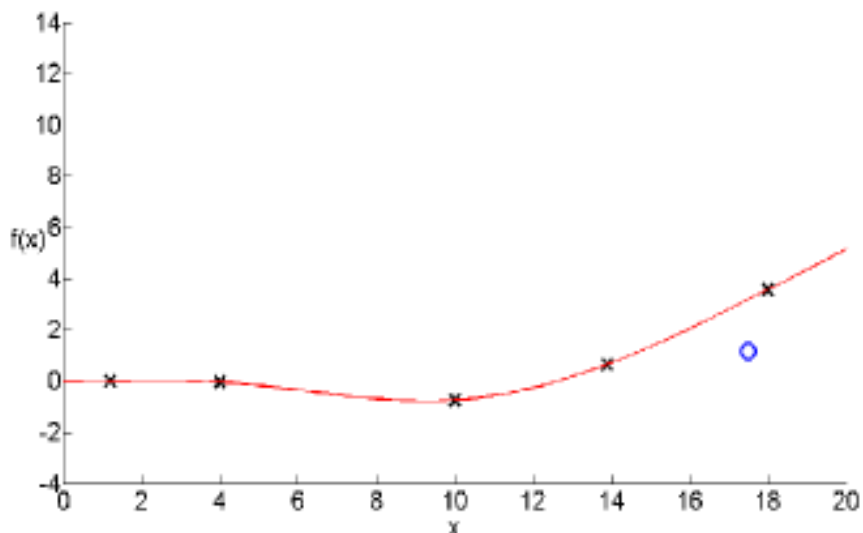
Στα επόμενα διαγράμματα φαίνεται σχηματικά η διαδικασία που ακολουθεί ο αλγόριθμος για μια τυχαία αντικειμενική συνάρτηση $f(x)$ μίας μεταβλητής σχεδιασμού x (Σχ.1 ως 4).



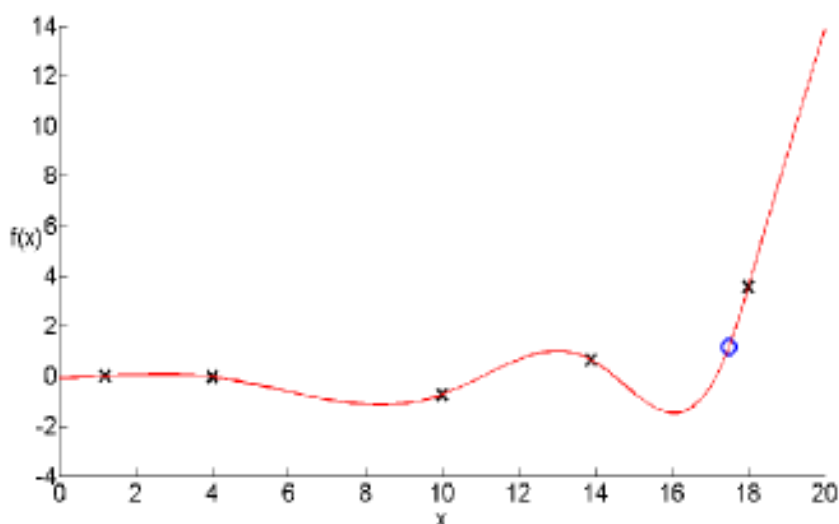
Σχ.1 Δημιουργία ενός αρχικού μοντέλου και υπολογισμός τιμών αρχικής αντικειμενικής συνάρτησης $f(x)$



Σχ.2 Προσαρμογή υποκατάστατου μοντέλου (surrogate model)(κόκκινη γραμμή)



Σχ.3 Επιλογή νέου σημείου από το πεδίο ορισμού (μπλε κύκλος) και υπολογισμός τιμής αρχικής αντικειμενικής συνάρτησης για τα νέο επιλεγμένο σημείο



Σχ.4 Προσαρμογή υποκατάστατου μοντέλου στο νέο σημείο (νέα κόκκινη γραμμή)

Η επιλογή των νέων σημείων από το πεδίο ορισμού, σημεία που θα πρέπει σε κάθε περίπτωση να ικανοποιούν τους σχεδιαστικούς περιορισμούς των μεταβλητών, είναι σημαντική διαδικασία καθώς θα καθορίσει τελικά το τοπικό ή καθολικό βέλτιστο της αρχικής αντικειμενικής συνάρτησης και ως εκ τούτου τη βέλτιστη λύση του προβλήματος. Για το λόγο αυτό, αν ο αλγόριθμος MATSuMoTo ύστερα από κάποιες προκαθορισμένες επαναλήψεις, δεν καταφέρει να βρει νέο σημείο το οποίο θα δώσει τιμή αρχικής αντικειμενικής καλύτερη από πριν, τότε ξεκινάει πάλι από την αρχή (1^ο βήμα). Δηλαδή, δημιουργεί ένα νέο αρχικό μοντέλο μέσω νέας δειγματοληψίας και χωρίς μάλιστα να λαμβάνει υπόψη τα σημεία που ελήφθησαν στο προηγούμενο μοντέλο. Με αυτό τον τρόπο αποφεύγει να εγκλωβιστεί σε τοπικά

βέλτιστα και να αναζητήσει καλύτερες λύσεις προκειμένου να βρει τελικά το καθολικό βέλτιστο.

5.3.3 Αλγόριθμος Latin Hypercube

Ο συγκεκριμένος αλγόριθμος είναι στην ουσία μία μέθοδος δειγματοληψίας, μια στατιστική μέθοδος για τη δημιουργία ενός (σχεδόν) τυχαίου δείγματος τιμών παραμέτρων από μια πολυδιάστατη κατανομή. Το Latin Hypercube Sampling (LHS) περιγράφηκε από τον Michael McKay του Εθνικού Εργαστηρίου του Los Alamos το 1979.

Στο πλαίσιο της στατιστικής δειγματοληψίας, ένα τετράγωνο πλέγμα που περιέχει θέσεις δείγματος είναι ένα Latin square, εάν (και μόνο εάν) υπάρχει μόνο ένα δείγμα σε κάθε σειρά και κάθε στήλη. Ένας Latin Hypercube είναι η γενίκευση αυτής της έννοιας σε έναν αυθαίρετο αριθμό διαστάσεων, όπου κάθε δείγμα είναι το μοναδικό σε κάθε άξονα – υπερεπίπεδο (hyperplane) που το εμπεριέχει.

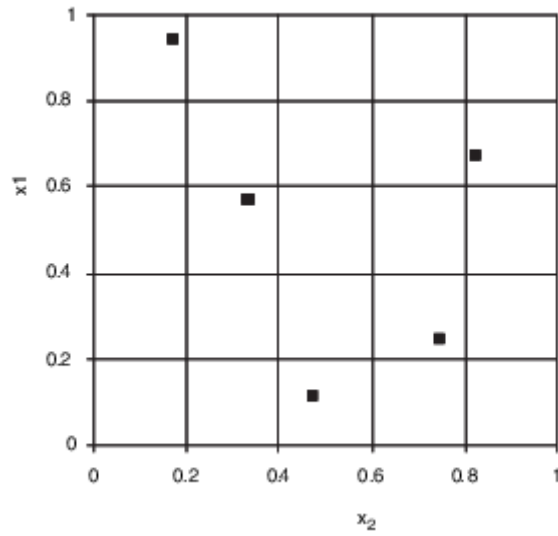
Κατά τη δειγματοληψία μιας συνάρτησης των K μεταβλητών x , το εύρος κάθε μεταβλητής διαιρείται σε N ίδια διαστήματα. Στη συνέχεια τοποθετούνται N δείγματα έτσι ώστε να ικανοποιούνται οι απαιτήσεις του Latin Hypercube. Αυτό υποχρεώνει τον αριθμό των διαιρέσεων N να είναι ίδιος για κάθε μεταβλητή (διάσταση). Αυτή η μορφή δειγματοληψίας δεν απαιτεί περισσότερα δείγματα για περισσότερες διαστάσεις (μεταβλητές). Ένα πλεονέκτημα είναι ότι τα τυχαία δείγματα μπορούν να ληφθούν ένα κάθε φορά και παράλληλα να θυμόμαστε ποια δείγματα ελήφθησαν μέχρι τώρα. Το μόνο που απαιτείται είναι να αποφασίσουμε πόσα σημεία δείγματος να χρησιμοποιήσουμε και για κάθε σημείο δειγματοληψίας να θυμόμαστε σε ποια σειρά και στήλη το σημείο δειγματοληψίας λήφθηκε.

Σε μαθηματική μορφή και σύμφωνα με τον McKay et al (1979), η ακριβής θέση του κάθε δείγματος X_{jk} στο πλέγμα, βρίσκεται με την ακόλουθη εξίσωση:

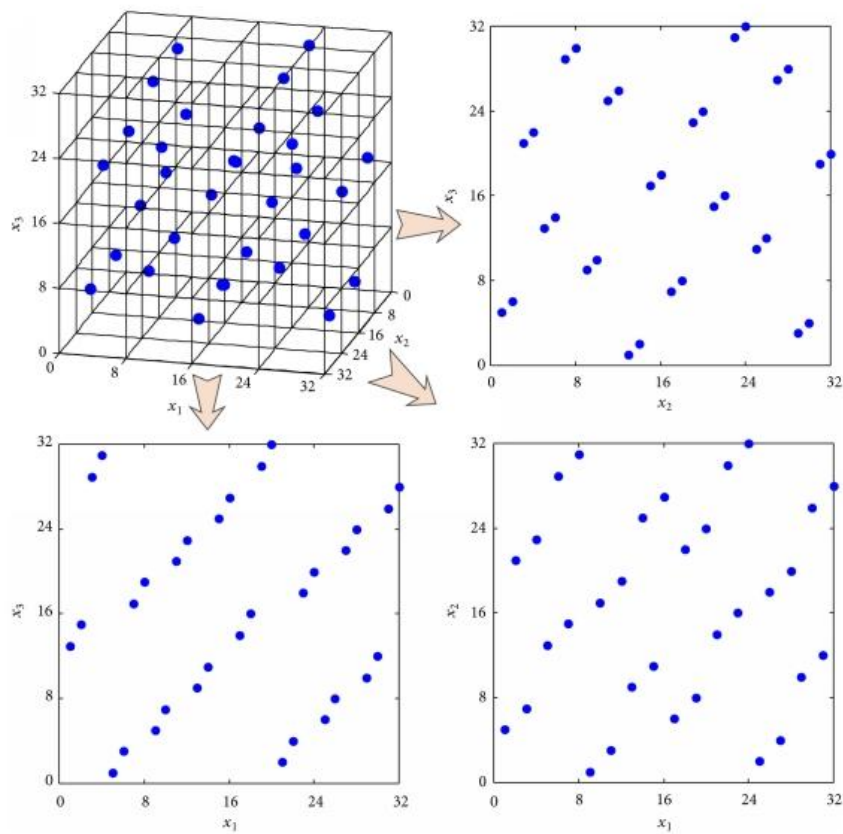
$$X_{jk} = F_k^{-1}\left(N^{-1}\left(p_{jk} - 1 + \xi_{jk}\right)\right), \text{ όπου } j=1, \dots, N \text{ και } k=1, \dots, K$$

Στη συνάρτηση αυτή, ο $P=(p_{jk})$ είναι ένας πίνακας $N \times K$ όπου κάθε στήλη είναι μια ανεξάρτητη τυχαία μετάθεση των αριθμών $1, \dots, N$. Έτσι, η τιμή του p_{jk} καθορίζει το «κελί» στο οποίο βρίσκεται το κάθε δείγμα και το ξ_{jk} (το οποίο είναι ομοιόμορφα κατανομημένο μεταξύ 0 και 1) την ακριβή του θέση μέσα στο «κελί». F_k είναι η αθροιστική συνάρτηση κατανομής των μεταβλητών x .

Στα επόμενα σχήματα φαίνεται απεικονιστικά η δειγματοληψία της Latin Hypercube Sampling για 2 και 3 μεταβλητές αντίστοιχα.



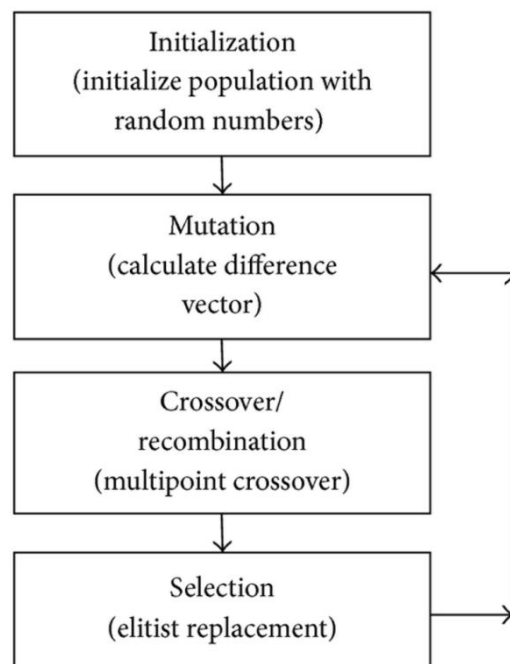
Παράδειγμα δειματοληψίας 5 δειγμάτων σε συνάρτηση 2 μεταβλητών x_1 και x_2



Παράδειγμα δειματοληψίας 32 δειγμάτων σε συνάρτηση 3 μεταβλητών x_1 , x_2 και x_3

5.3.4 Αλγόριθμος Differential Evolution

Ο αλγόριθμος της Διαφορικής Εξέλιξης (Differential Evolution Algorithm) είναι ένας Εξελικτικός Αλγόριθμος που προτάθηκε από τους Storn και Price (1997). Οι Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms – EA) είναι μετα-ευρεστικές μέθοδοι βελτιστοποίησης με βάση τον πληθυσμό (population), που έχουν προέλευση και έμπνευση από τον κόσμο της βιολογίας. Οι Εξελικτικοί Αλγόριθμοι ενσωματώνουν τις φυσικές διαδικασίες της επιλογής (selection), της μετάλλαξης (mutation) και της διασταύρωσης (crossover) ή αλλιώς του ανασυνδυασμού (recombination) και τις χρησιμοποιούν ως μηχανισμούς ή τελεστές αναζήτησης, ώστε να βρουν καλύτερες λύσεις γρηγορότερα σε προβλήματα βελτιστοποίησης.



Τα βασικά στάδια του αλγορίθμου Διαφορικής Εξέλιξης

Η Διαφορική Εξέλιξη είναι μια αποτελεσματική στοχαστική μέθοδος που βασίζεται στον «πληθυσμό» για καθολική βελτιστοποίηση σε συνεχείς χώρους αναζήτησης. Η Διαφορική Εξέλιξη δεν απαιτεί το πρόβλημα βελτιστοποίησης να είναι διαφορίσιμο, όπως απαιτείται από τις κλασικές μεθόδους βελτιστοποίησης. Ως Εξελικτικός Αλγόριθμος που βασίζεται στον πληθυσμό, η Διαφορική Εξέλιξη επιλύει προβλήματα βελτιστοποίησης εξελίσσοντας τον πληθυσμό των υποψήφιων λύσεων χρησιμοποιώντας τους τελεστές της μετάλλαξης, της διασταύρωσης και της επιλογής.

Η πρωταρχική ιδέα της Διαφορικής Εξέλιξης είναι να προσαρμόσει την αναζήτηση κατά τη διάρκεια της εξελικτικής διαδικασίας. Κατά την αρχική φάση της εξέλιξης, οι διακυμάνσεις είναι μεγάλες, καθόσον οι υποψήφιες λύσεις είναι πολύ μακριά η μία από την άλλη. Όσο η εξελικτική διαδικασία προχωράει, ο πληθυσμός συγκλίνει σε μια μικρή περιοχή και οι διακυμάνσεις γίνονται μικρότερες. Πρακτικά

δημιουργεί έναν αρχικό πληθυσμό υποψήφιων λύσεων και στη συνέχεια με χρήση απλών μαθηματικών τύπων συνδυάζονται οι θέσεις των υπαρχόντων λύσεων του πληθυσμού. Δημιουργούνται, έτσι, συνεχώς νέες υποψήφιες λύσεις, κρατώντας τελικά εκείνες που έχουν την καλύτερη «επίδοση». Η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί τελικά, αλλά όχι εγγυημένα, η βέλτιστη λύση.

5.3.5 Αλγόριθμος PBA

Ο Pity Beetle Algorithm (PBA) προτάθηκε από τους Καλλιώρα κ.ά. (2018) και είναι βασισμένος στον γνωστό αλγόριθμο Σμήνους Σωματιδίων (PSO). Εμπνεύστηκε από την κοινωνική συμπεριφορά κατά την αναζήτηση στέγης και τροφής του σκαθαριού *Pityogenes chalcographus* επίσης γνωστού και ως σκαθάρι της ερυθρελάτης (six-toothed spruce bark beetle). Αυτό το σκαθάρι έχει τη δυνατότητα να εντοπίζει και να αποικεί αρχικά στο φλοιό των εξασθενημένων δέντρων σε ένα δάσος, ενώ όταν ο πληθυσμός του υπερβαίνει ένα συγκεκριμένο όριο μπορεί να προσβάλει ακόμα και υγιή δέντρα.

Η διαδικασία που ακολουθείται από αυτόν τον αλγόριθμο βελτιστοποίησης χωρίζεται σε τρία στάδια. Αρχικά, στο πρώτο στάδιο (στάδιο αναζήτησης), τα «πρωτοπόρα» αρσενικά σκαθάρια εντοπίζουν ένα κατάλληλο «ξενιστή» (δέντρο) από τις εκπομπές χημικών ουσιών των εξασθενημένων δέντρων του δάσους. Στο δεύτερο στάδιο (στάδιο συγκέντρωσης), αυτά τα πρωτοπόρα σκαθάρια αρχίζουν να τρέφονται με τον ξενιστή οπότε και παράγουν μια φερομόνη που προσελκύει τα γύρω αρσενικά και θηλυκά σκαθάρια, αυξάνοντας έτσι τοπικά τον πληθυσμό. Στο τελευταίο στάδιο (στάδιο αντι-συγκέντρωσης), όταν ο πληθυσμός των σκαθαριών ξεπεράσει ένα συγκεκριμένο όριο, τότε οι μηχανισμοί άμυνας του ξενιστή δεν μπορούν πλέον να ανταπεξέλθουν σε αυτήν τη μαζική εισβολή, οπότε σε αυτό το επίπεδο πληθυσμού, τόσο τα εξασθενημένα όσο και τα υγιή πλέον δέντρα, είναι δυνατόν να δεχτούν εισβολή. Ωστόσο, επειδή ένα υπερπληθυσμένο δέντρο – ξενιστής προκαλεί προβλήματα στην επιβίωση των σκαθαριών (λιγότερος χώρος για τροφή, υψηλότερη πιθανότητα μολυσματικών ασθενειών), όταν η πυκνότητα των σκαθαριών γίνεται πολύ υψηλή, τότε τα σκαθάρια που τρέφονται από αυτό το δέντρο, εκπέμπουν μια διαφορετική φερομόνη η οποία αποθαρρύνει περισσότερα σκαθάρια να επιτεθούν σε αυτό, οδηγώντας τα έτσι σε άλλα δέντρα. Με αυτόν τον τρόπο, η εισβολή επεκτείνεται σταδιακά σε όλο το δάσος δημιουργώντας ομάδες νεκρών δέντρων γύρω από το αρχικά προσβεβλημένο δέντρο.

Σε μαθηματικούς όρους, η παραπάνω διαδικασία αναλύεται σε τρία βήματα:

a. *Initialization*

Στο πρώτο βήμα του αλγορίθμου (δηλ. στη γενιά $g = 0$), ένας αρχικός αριθμός σωματιδίων (σ.σ. σκαθαριών) N_{pop} , που ονομάζονται «πρωτοπόρα» σωματίδια, τοποθετείται τυχαία μέσα σε έναν υπερόγκο (hypervolume) ίσο με τον

καθολικό χώρο αναζήτησης, ψάχνοντας για διανύσματα βέλτιστης θέσης $\mathbf{x}_j^{(0)} \in R^D$. Η θέση κάθε πρωτοπόρου σωματιδίου στον αρχικό πληθυσμό επιλέγεται τυχαία μέσω μιας τεχνικής τυχαίας δειγματοληψίας (random sampling technique – RST) και δίδεται από την σχέση:

$$\mathbf{x}_j^{(0)} = RST(\mathbf{L}, \mathbf{U}, D, N_{pop}),$$

όπου $\mathbf{L} = [L_1, L_2, \dots, L_D]$ είναι το κάτω όριο, $\mathbf{U} = [U_1, U_2, \dots, U_D]$ το άνω όριο και $j = 1, 2, \dots, N_{pop}$.

Όταν πλέον όλα τα σωματίδια έχουν καταλάβει μια καθορισμένη θέση μέσα στον υπερόγκο (διάνυσμα υποψηφίων λύσεων), τότε αυτές οι λύσεις N_{pop} συγκρίνονται μεταξύ τους και το καλύτερο σωματίδιο προσελκύει τα υπόλοιπα και έτσι δημιουργείται ο πρώτος πληθυσμός. Συνολικά δημιουργούνται έξι τέτοιοι πληθυσμοί που ο κάθε ένας αποτελείται από N_{pop} σωματίδια ώστε στη συνέχεια ο κάθε ένας να δημιουργήσει νέες υποψήφιες λύσεις με βάση ένα μοτίβο επιλογής νέου υπερόγκου (new hypervolume selection pattern) σύμφωνα με διάφορα κριτήρια.

β. *New hypervolume selection pattern*

Στο δεύτερο βήμα του αλγορίθμου, όλα τα νεοσυσταθέντα σωματίδια θα ψάξουν μέσα στο χώρο αναζήτησης για μια καλύτερη λύση για να δημιουργήσουν τον δικό τους πληθυσμό. Με βάση τη συμπεριφορά του σκαθαριού που περιγράφηκε προηγουμένως, υπάρχουν πέντε μοτίβα επιλογής νέου υπερόγκου στον συγκεκριμένο αλγόριθμο:

1. *Γειτονικός όγκος αναζήτησης*

Μια νέα περιοχή αναζήτησης δημιουργείται γύρω από την αρχική θέση των σωματιδίων σύμφωνα με ένα συντελεστή f_{nb} (neighboring factor) ο οποίος καθορίζει το μέγεθος αυτής της περιοχής και κυμαίνεται μεταξύ 0.10 και 0.20. Το μοτίβο αυτό επιλέγεται είτε επειδή μερικά, όχι αρκετά ισχυρά σκαθάκια, δεν μπορούν να πετούν σε μεγάλες αποστάσεις ή επειδή υπάρχουν κατάλληλες θέσεις (σ.σ. εξασθενημένα δέντρα) σε κοντινή απόσταση από τη θέση γέννησής τους ή επειδή ο υπερπληθυσμός των σκαθαριών σε κοντινή απόσταση μπορεί να οδηγήσει σε εισβολή ακόμη και σε υγιή δέντρα που βρίσκονται μέσα σε αυτό το εύρος.

2. *Όγκος αναζήτησης μεσαίας κλίμακας*

Το μοτίβο αυτό επιλέγεται όταν το προηγούμενο δεν καταφέρνει να δώσει καλύτερες λύσεις συγκριτικά με την αρχική θέση. Όπως και πριν μια νέα περιοχή αναζήτησης δημιουργείται γύρω από την τρέχουσα καλύτερη θέση των σωματιδίων σύμφωνα με ένα συντελεστή f_{ms} (mid-scale factor) που κυμαίνεται μεταξύ 0.10 και 1.00.

3. Όγκος αναζήτησης μεγάλης κλίμακας

Το μοτίβο αυτό επιλέγεται είτε επειδή μερικά από τα νέα σκαθάρια, συνήθως τα πιο ισχυρά, μπορούν να πετάξουν σε μεγάλες αποστάσεις από τη θέση γέννησής τους προς αναζήτηση ενός άλλου κατάλληλου δέντρου – ξενιστή ή επειδή το μοτίβο γειτονικού όγκου αναζήτησης δεν καταφέρνει να δώσει καλύτερες λύσεις συγκριτικά με την αρχική θέση. Η νέα περιοχή αναζήτησης δημιουργείται σύμφωνα με ένα συντελεστή f_{1s} (large-scale factor) που κυμαίνεται μεταξύ 1 και 100.

4. Όγκος αναζήτησης καθολικής κλίμακας

Χρησιμοποιείται όταν ο αλγόριθμος υπερβεί έναν προκαθορισμένο μέγιστο αριθμό υπολογισμών της αντικειμενικής συνάρτησης και δεν καταφέρει να δώσει καλύτερες λύσεις.

5. Όγκος αναζήτησης με βάση τη μνήμη

Στο μοτίβο αυτό οι καλύτερες λύσεις που έχουν βρεθεί μέχρι τώρα από τον αλγόριθμο αποθηκεύονται σε μια μνήμη (MEM) μεγέθους ίση με N_{pop} και επιλέγεται όταν μια συγκεκριμένη περιοχή του χώρου αναζήτησης έχει πολλούς κατάλληλους (εξασθενημένους) ξενιστές, οπότε αυτό οδηγεί σε αυξημένη συγκέντρωση σκαθαριών σε αυτήν την περιοχή με αποτέλεσμα την πιθανότητα εισβολής σε οποιοδήποτε δέντρο ανεξάρτητα από την ανθεκτικότητά του.

γ. *Update location of populations*

Σε αυτό το βήμα οι θέσεις των απογόνων (θέσεις ζευγαρώματος αρσενικών και θηλυκών) ενημερώνονται και οι προηγούμενες απορρίπτονται, εκτός από αυτές που είναι αποθηκευμένες στη μνήμη (MEM). Συγκεκριμένα, όλοι οι προηγούμενοι απόγονοι εξαφανίζονται και οι νέες τοποθεσίες γίνονται θέσεις αναπαραγωγής για τις νέες γενιές. Στη συνέχεια, με βάση τους κανόνες που περιγράφηκαν προηγουμένως, νέα μοτίβα υπερόγκων επιλέγονται ξανά για τους νέους πληθυσμούς και αυτή η διαδικασία συνεχίζεται μέχρι να ικανοποιηθούν τα κριτήρια τερματισμού του αλγορίθμου PBA.

6 ΑΠΟΤΕΛΕΣΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Στη συνέχεια αναλύονται διεξοδικά τα αποτελέσματα της βελτιστοποίησης των έξι μοντέλων της παραγράφου 5.2 που χρησιμοποιήθηκαν στην παρούσα εργασία, μετά την εφαρμογή των τριών αλγορίθμων βελτιστοποίησης που αναφέρθηκαν νωρίτερα. Πιο συγκεκριμένα, για κάθε μοντέλο και αλγόριθμο βελτιστοποίησης που εφαρμόζεται, παρουσιάζεται η πορεία του αλγορίθμου όσον αφορά:

- α. Στο ποσοστό % βελτίωσης της τιμής της αντικειμενικής συνάρτησης σε σχέση με την τιμή σχεδιασμού.
- β. Στην τιμή της παραβίασης (violation) κάθε επανάληψης (ανάλυσης του μοντέλου).
- γ. Στην τιμή της αντικειμενικής συνάρτησης.
- δ. Στον αριθμό των «εφικτών» λύσεων, δηλ. των επαναλήψεων εκείνων όπου δεν υπάρχει παραβίαση και δίνουν αποδεκτή λύση.
- ε. Στην ταχύτητα εμφάνισης εφικτών λύσεων και ιδιαίτερα της πρώτης εφικτής λύσης και της βέλτιστης λύσης.

Για την καλύτερη κατανόηση των παραπάνω, παρουσιάζονται διαγράμματα στα οποία απεικονίζεται η πορεία του κάθε αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του κάθε μοντέλου. Στον οριζόντιο άξονα εμφανίζεται πάντα ο αριθμός της επανάληψης ενώ στον κατακόρυφο άξονα εμφανίζονται:

- α. **Το ποσοστό βελτίωσης (%)** της τιμής της αντικειμενικής συνάρτησης της κάθε επανάληψης σε σχέση με την τιμή σχεδιασμού, δηλαδή την τιμή της αντικειμενικής που λαμβάνει το μοντέλο με βάση τις αρχικές διατομές σχεδιασμού.
- β. **Η τιμή της παραβίασης** της κάθε επανάληψης. Με τον όρο «παραβίαση» εννοείται η υπέρβαση των ορίων σχεδιασμού – περιορισμών που έχουν τεθεί (design violation) και η οποία πρακτικά μας δείχνει την «απόσταση» που έχουν οι επιλεγμένες τιμές των μεταβλητών σχεδιασμού (δηλαδή οι διαστάσεις των διατομών) από τους σχεδιαστικούς περιορισμούς.

Μία μεγάλη τιμή παραβίασης σημαίνει ότι, οι διατομές που έχουν επιλεγεί είναι ενδεχομένως πολύ μικρές για να ικανοποιήσουν τα σχεδιαστικά κριτήρια, αν και δίνουν μικρή τιμή αντικειμενικής δηλ. μικρό βάρος κατασκευής. Σε αντίθεση, πολύ μικρή παραβίαση σημαίνει ότι έχουμε ενδεχομένως μία άκαμπτη κατασκευή με μεγάλες διατομές, η οποία όμως επιφέρει μεγάλο βάρος και άρα μεγάλη τιμή αντικειμενικής, κάτι που είναι αντίθετο με το ζητούμενο. Για το λόγο

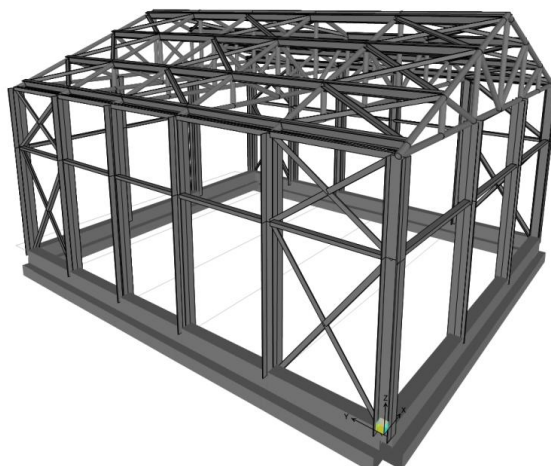
αυτό, η βέλτιστη λύση θα κινείται πάντα στο όριο της τιμής της παραβίασης δηλ. κοντά στο 1,0.

γ. **Η τιμή της αντικειμενικής συνάρτησης** αφενός κάθε επανάληψης και αφετέρου των επαναλήψεων εκείνων που δίνουν εφικτές λύσεις δηλ. που έχουν παραβίαση μικρότερη ή ίση με το 1,0.

Επισημαίνεται ότι, τα μοντέλα έχουν σχεδιαστεί και αναλύονται σε κάθε επανάληψη στο πρόγραμμα SAP2000 v.20 της CSI, ενώ για τη βελτιστοποίηση χρησιμοποιείται η πλατφόρμα High Performance Optimization Computing Platform HP-OCP. Επιπλέον, σε όλες τις περιπτώσεις που μελετήθηκαν, η αντικειμενική συνάρτηση (objective function) είναι η ελαχιστοποίηση του βάρους της κατασκευής και οι περιορισμοί που έχουν τεθεί είναι οι σχεδιαστικοί (Design Violations) δηλ. οι κατασκευαστικοί και σεισμικοί κώδικες (EC-2, EC-3, EC-8, κτλ.) οι οποίοι εμπεριέχονται στο SAP2000.

6.1 Ανάλυση 1^ο Μοντέλου

Το 1^ο μοντέλο, όπως προαναφέρθηκε, αφορά σε έναν αχυρώνα κατασκευασμένο από χαλύβδινα στοιχεία διαφόρων διατομών και θεμελίωση από οπλισμένο σκυρόδεμα. Αποτελείται συνολικά από 337 στοιχεία και 154 κόμβους, το ύψος μέχρι το στέγαστρο είναι 4,85m και το συνολικό ύψος 6,85m. Τέλος έχει διαστάσεις σε κάτοψη 11,25 x 9,80 m. και εμβαδό 110,25 m². Οι μεταλλικές διατομές είναι από χάλυβα S275 και η θεμελίωση από σκυρόδεμα C30/37.



Οι διατομές που έχουν επιλεγεί κατά το σχεδιασμό του αρχικού μοντέλου είναι οι ακόλουθες:

- Υποστυλώματα HEA280
- Θεμελίωση διατομής T h=700mm, w=700mm, t_f=300mm, t_w=300mm
- Κύριες Δοκοί HEA100
- Μηκίδες 125x75x6RHS

- Χιαστί σύνδεσμοι οροφής L90x12
- Χιαστί πλευρικοί σύνδεσμοι L100x10
- Τεγίδες IPE200
- Στοιχεία δικτυωμάτων οροφής 114.3x6CHS

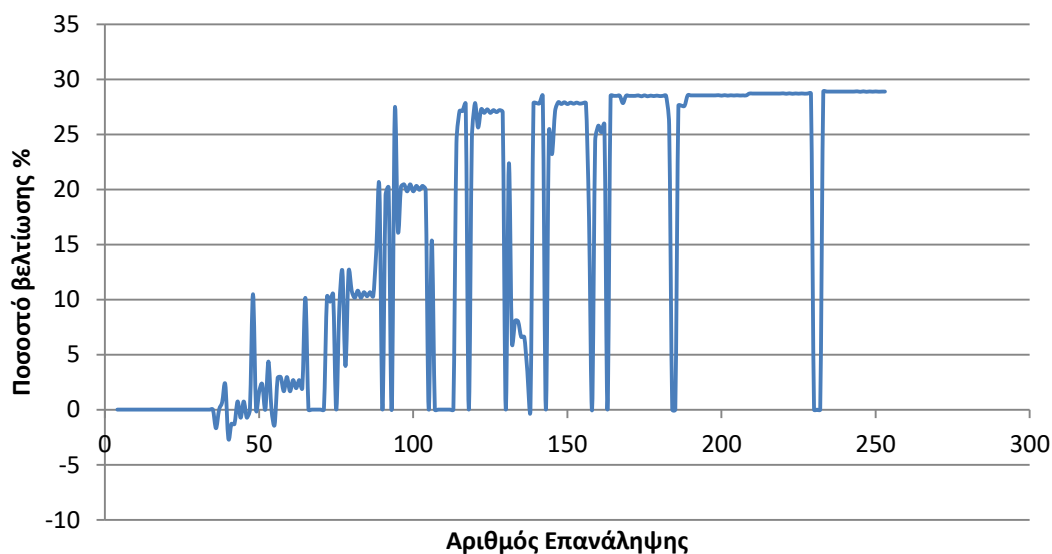
6.1.1 Εφαρμογή του αλγορίθμου PQN

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης PQN, η τιμή σχεδιασμού, δηλ. η τιμή της αντικειμενικής συνάρτησης του αρχικού μοντέλου πριν γίνει βελτιστοποίηση, είναι 14.347. Συνολικά γίνονται 253 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 244^η επανάληψη με τιμή 10.201 (βελτίωση σε ποσοστό 28,90% επί της τιμής σχεδιασμού).

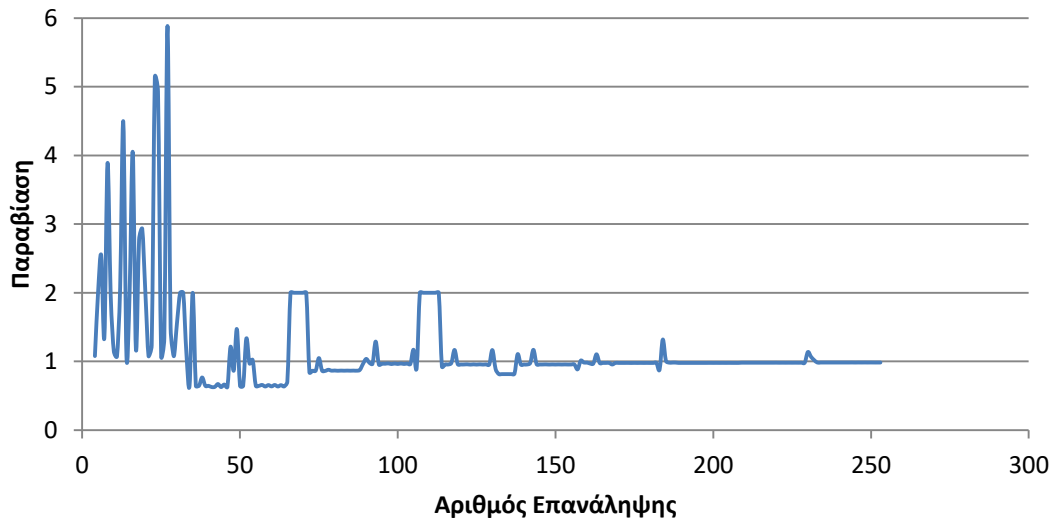
Οι αντικειμενικές τιμές στις πρώτες επαναλήψεις εμφανίζουν μεγάλη διακύμανση γύρω από την τιμή σχεδιασμού, μέχρι την επανάληψη εκείνη που δίνει την πρώτη εφικτή λύση. Από εκεί και μετά, ο αλγόριθμος αναζητάει «καλύτερες» εφικτές λύσεις σε σημεία κοντά στο υπάρχον τοπικό βέλτιστο, μειώνοντας έτσι συνεχώς την τιμή της αντικειμενικής συνάρτησης μέχρι να βρει την βέλτιστη τιμή και φυσικά ωστόσο σταματήσει να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Για το λόγο αυτό, αντίστοιχα και οι παραβιάσεις αρχικά είναι μεγάλες αλλά μόλις εντοπίσει την πρώτη εφικτή λύση μειώνονται σημαντικά και κινούνται γύρω από την τιμή 1,0.

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

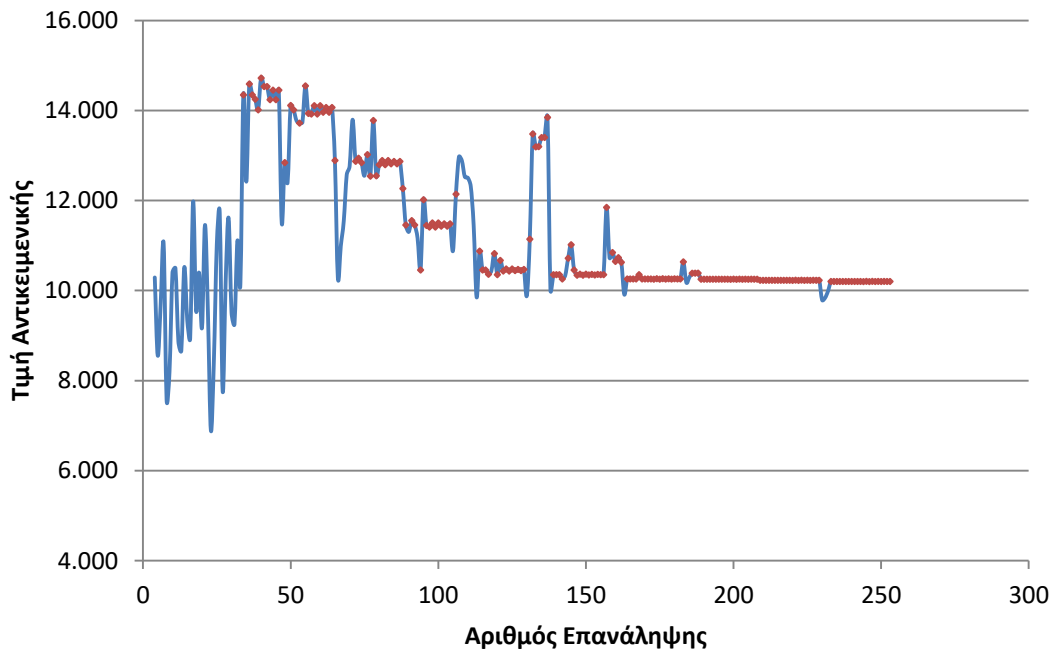
1ο Μοντέλο με PQN - Ποσοστό βελτίωσης



1ο Μοντέλο με PQN - Παραβίαση



1ο Μοντέλο με PQN - Τιμή αντικειμενικής



* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.1.2 Εφαρμογή του αλγορίθμου MATSuMoTo

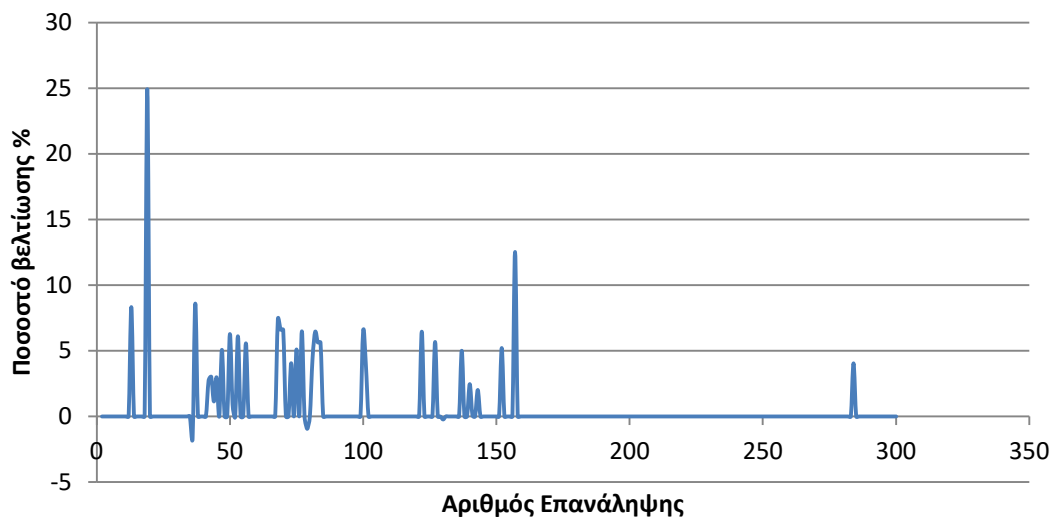
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης MATSuMoTo η τιμή σχεδιασμού είναι 14.347. Συνολικά γίνονται 300 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης

εμφανίζεται στην 19^η επανάληψη με τιμή 10.771 (βελτίωση σε ποσοστό 24,93% επί της τιμής σχεδιασμού).

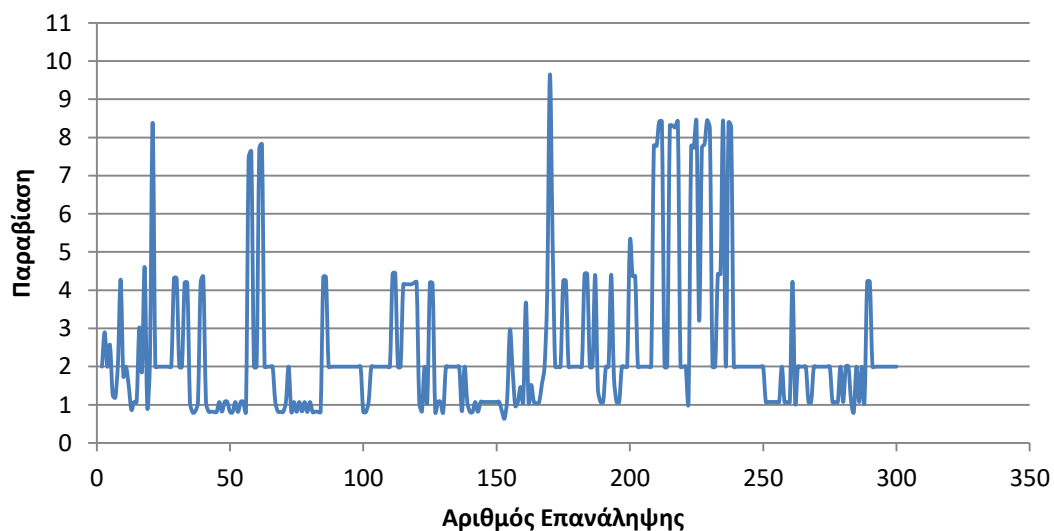
Σε όλες τις επαναλήψεις υπάρχει μεγάλη διακύμανση τιμών, τόσο της αντικειμενικής συνάρτησης όσο και των παραβιάσεων. Ο αλγόριθμος αναζητάει συνεχώς καλύτερες λύσεις σε όλο το πεδίο σχεδιασμού και έτσι δεν εγκλωβίζεται σε κάποιο ενδεχομένως τοπικό βέλτιστο, ενώ εντοπίζει πολύ γρήγορα τη βέλτιστη τιμή.

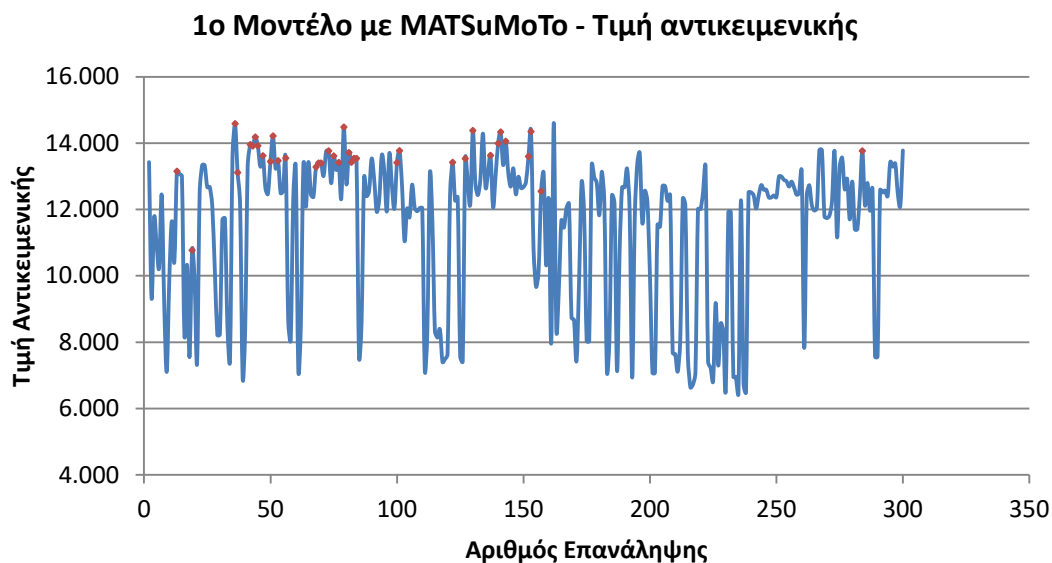
Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

1ο Μοντέλο με MATSuMoTo - Ποσοστό βελτίωσης



1ο Μοντέλο με MATSuMoTo - Παραβίαση





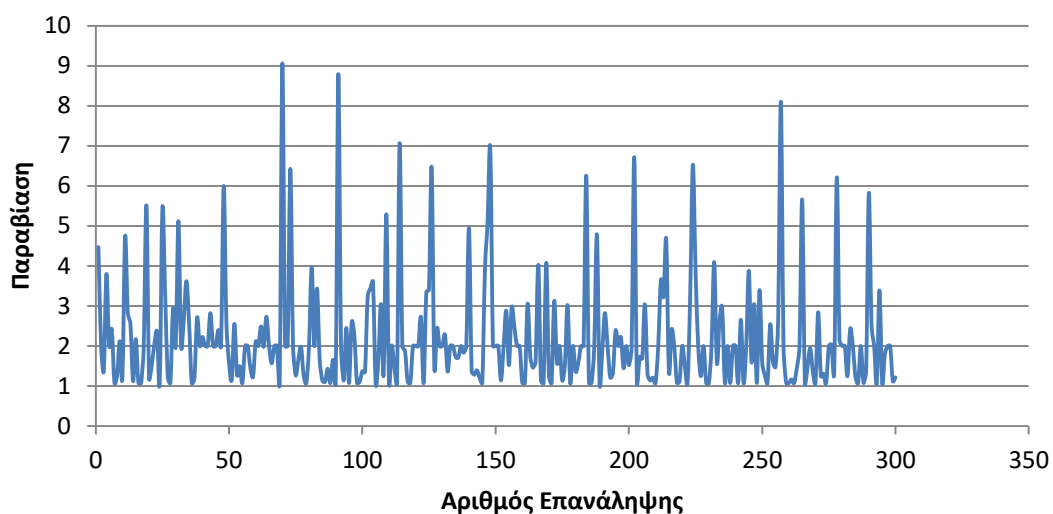
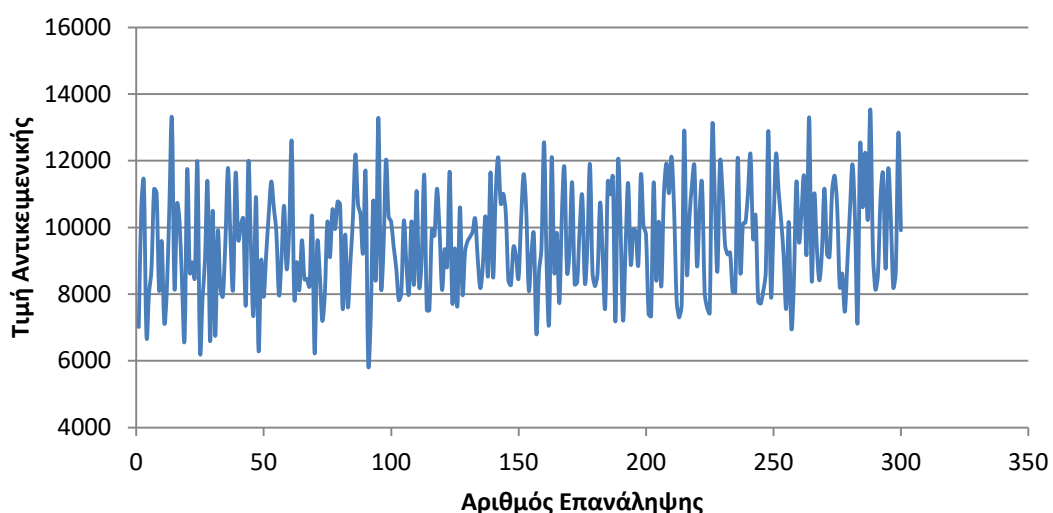
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.1.3 Εφαρμογή του αλγορίθμου Latin Hypercube

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης Latin Hypercube η πρώτη τιμή της αντικειμενικής συνάρτησης είναι 7.015 εμφανίζοντας αρκετά μεγάλη παραβίαση 4,471. Συνολικά γίνονται 300 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί.

Γενικά ο αλγόριθμος καθ' όλη τη διάρκεια της εφαρμογής του δεν καταφέρνει να βρει καμία εφικτή λύση. Οι τιμές της αντικειμενικής έχουν μεγάλη διακύμανση σε όλο το εύρος των επαναλήψεων με μέγιστη τιμή στην 288^η επανάληψη με τιμή 13.536 (παραβίαση 1,075) και ελάχιστη στην 91^η επανάληψη με τιμή 5.960 (παραβίαση 8,789).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

1ο Μοντέλο με Latin Hypercube - Παραβίαση**1ο Μοντέλο με Latin Hypercube - Τιμή αντικειμενικής**

* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.1.4 Συγκριτική ανάλυση αποτελεσμάτων

Κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου PQN, οι αντικειμενικές τιμές στις πρώτες επαναλήψεις εμφανίζουν μεγάλη διακύμανση γύρω από την τιμή σχεδιασμού, μέχρι την επανάληψη εκείνη που δίνει την πρώτη εφικτή λύση. Από εκεί και μετά, ο αλγόριθμος αναζητάει «καλύτερες» εφικτές λύσεις σε σημεία κοντά στο υπάρχον τοπικό βέλτιστο.

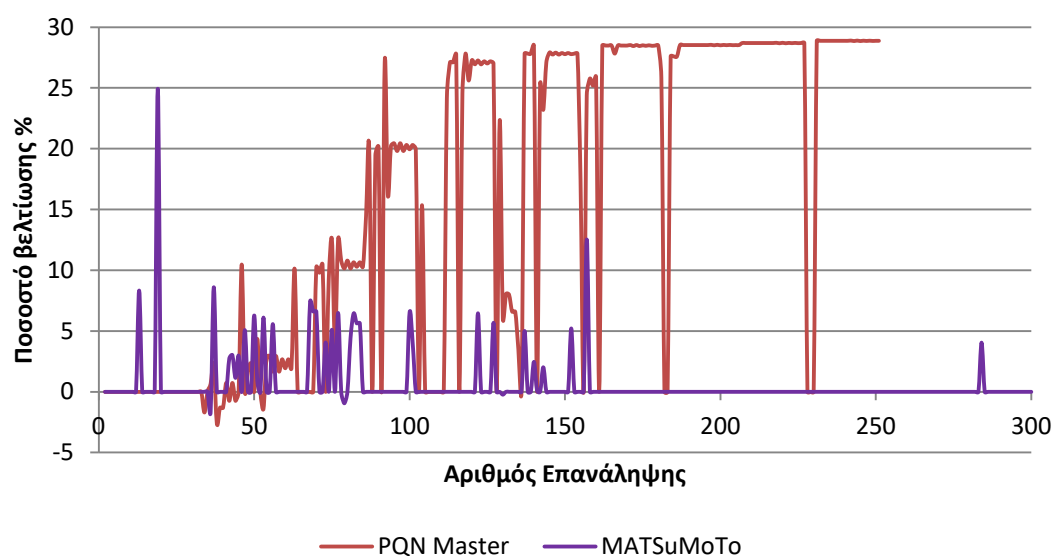
Κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου MATSuMoTo σε όλες τις επαναλήψεις υπάρχει μεγάλη διακύμανση τιμών, τόσο της αντικειμενικής συνάρτησης όσο και των παραβιάσεων. Εντοπίζει πολύ πιο γρήγορα τη βέλτιστη τιμή

η οποία όμως αποδεικνύεται τελικά «χειρότερη» από την βέλτιστη τιμή που δίνει ο PQN.

Κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου Latin Hypercube, δεν καταφέρνει να βρει κάποια αποδεκτή λύση.

Προς επίρρωση των παραπάνω, στο επόμενο διάγραμμα και πίνακα απεικονίζεται ταυτόχρονα η πορεία όλων των αλγορίθμων κατά τη διαδικασία βελτιστοποίησης του μοντέλου. Στο διάγραμμα που απεικονίζεται το ποσοστό βελτίωσης της αντικειμενικής συνάρτησης, δεν έχει σκόπιμα συμπεριληφθεί ο αλγόριθμος Latin Hypercube, καθώς ο συγκεκριμένος αλγόριθμος δεν καταφέρνει να βρει κάποια αποδεκτή λύση.

1ο Μοντέλο Σύγκριση Ποσοστών Βελτίωσης



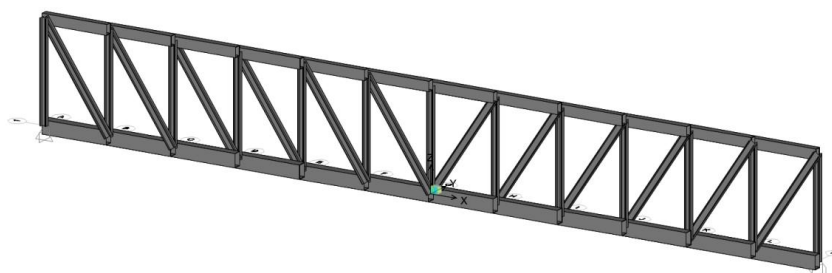
Συνοπτικός πίνακας σύγκρισης πορείας αλγορίθμων

	PQN	MATSuMoTo	Latin Hypercube
Μέγιστη τιμή	14.769	14.608	13.536
Ελάχιστη τιμή	6.442	6.460	5.960
Τιμή σχεδιασμού	14.347 (εφικτή)	14.347 (εφικτή)	-
Αριθμός Επαναλήψεων	253	300	300
Πρώτη εφικτή λύση	(34 ^η) 14.347 0%	(13 ^η) 13.152 8,33%	-
Πρώτη εφικτή με τιμή	(38 ^η) 14.252 0,66%	(13 ^η) 13.152 8,33%	-

μικρότερη του σχεδιασμού			
Βέλτιστη τιμή	(244 ^η) 10.201	(19 ^η) 10.771	-
Ποσοστό βελτίωσης %	28,90%	24,93%	-
Πορεία αλγορίθμου	(34 ^η) 14.347 0% (38 ^η) 14.252 0,66% (48 ^η) 12.844 10,48% (77 ^η) 12.542 12,58% (89 ^η) 11.458 20,14% (94 ^η) 10.458 27,11% (244 ^η) 10.201 28,90%	(13 ^η) 13.152 8,33% (19 ^η) 10.771 24,93%	Δεν βρίσκει εφικτές λύσεις

6.2 Ανάλυση 2^ο Μοντέλου

Το 2^ο μοντέλο, όπως προαναφέρθηκε, αφορά σε μία 2D δικτυωτή γέφυρα με μεταλλικά στοιχεία διαφόρων διατομών. Αποτελείται συνολικά από 49 στοιχεία και 26 κόμβους, το ύψος είναι 6m και το συνολικό μήκος 48m. Όλες οι μεταλλικές διατομές είναι από χάλυβα S355.



Οι διατομές που έχουν επιλεγεί κατά το σχεδιασμό του αρχικού μοντέλου είναι οι ακόλουθες:

- Άνω πέλμα κοίλης ορθογωνικής διατομής με $h=600\text{mm}$, $w=400\text{mm}$, $t_f=18\text{mm}$, $t_w=18\text{mm}$
- Κάτω πέλμα κοίλης ορθογωνικής διατομής με $h=800\text{mm}$, $w=500\text{mm}$, $t_f=12\text{mm}$, $t_w=12\text{mm}$
- Ορθοστάτες HEM300
- Διαγώνιοι HEA400

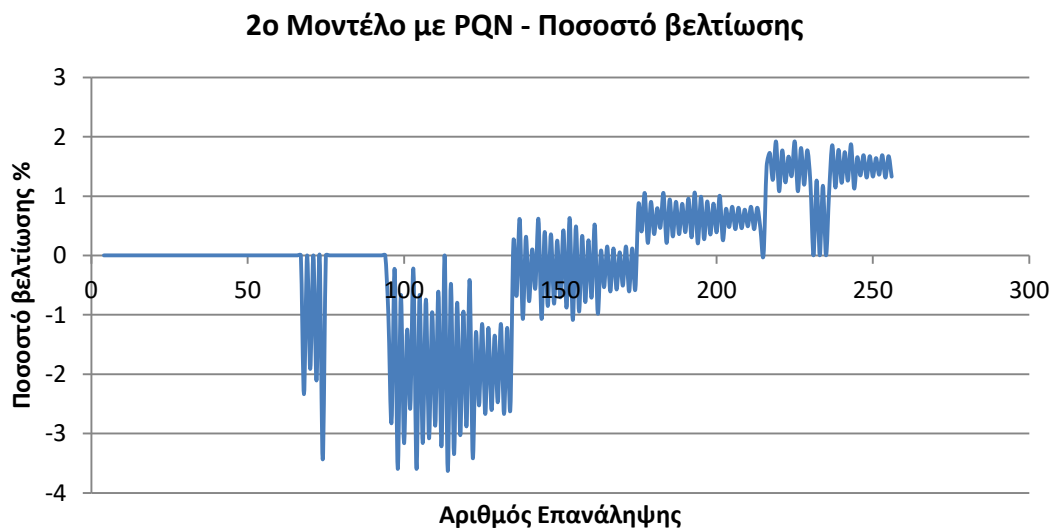
6.2.1 Εφαρμογή του αλγορίθμου PQN

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης PQN, η τιμή σχεδιασμού, δηλ. η τιμή της αντικειμενικής συνάρτησης του αρχικού μοντέλου πριν γίνει βελτιστοποίηση, είναι 47.847. Η επιλογή όμως των συγκεκριμένων (αρχικών) διατομών σχεδιασμού εμφανίζουν μικρή παραβίαση 1,02 γεγονός που

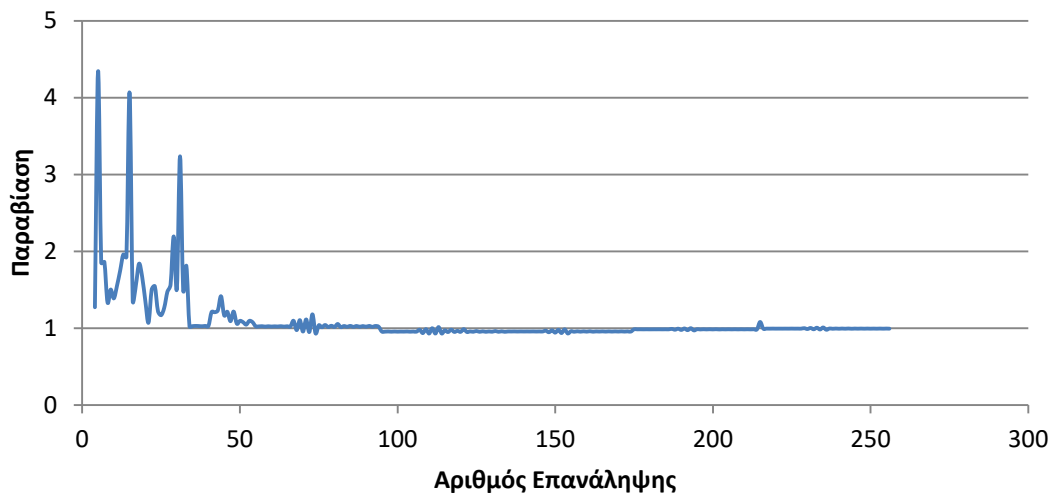
καθιστά τη συγκεκριμένη λύση μη εφικτή. Συνολικά γίνονται 256 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 225^η επανάληψη με τιμή 46.929 (μικρή βελτίωση σε ποσοστό 1,92% επί της τιμής σχεδιασμού).

Όπως και στο προηγούμενο μοντέλο, κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου PQN, οι αντικειμενικές τιμές στις πρώτες επαναλήψεις εμφανίζουν μεγάλη διακύμανση γύρω από την τιμή σχεδιασμού, μέχρι την επανάληψη εκείνη που δίνει την πρώτη εφικτή λύση. Από εκεί και μετά, ο αλγόριθμος αναζητάει «καλύτερες» εφικτές λύσεις σε σημεία κοντά στο υπάρχον τοπικό βέλτιστο, μειώνοντας έτσι συνεχώς την τιμή της αντικειμενικής συνάρτησης μέχρι να βρει την βέλτιστη τιμή.

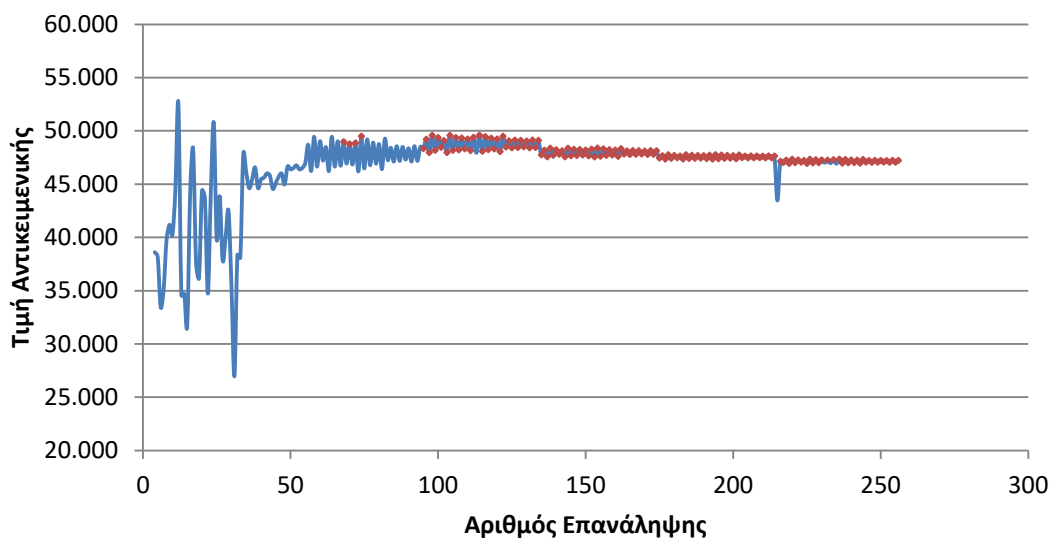
Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.



2ο Μοντέλο με PQN - Παραβίαση



2ο Μοντέλο με PQN - Τιμή αντικειμενικής



* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

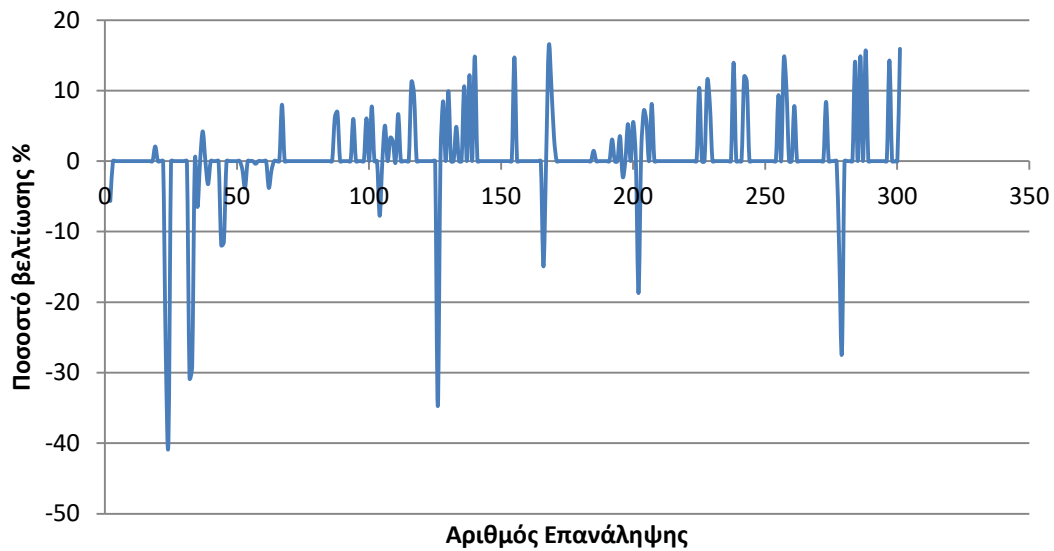
6.2.2 Εφαρμογή του αλγορίθμου MATSuMoTo

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης MATSuMoTo η τιμή σχεδιασμού είναι 48.001. Συνολικά γίνονται 301 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 168^η επανάληψη με τιμή 40.202 (βελτίωση σε ποσοστό 16,25% επί της τιμής σχεδιασμού).

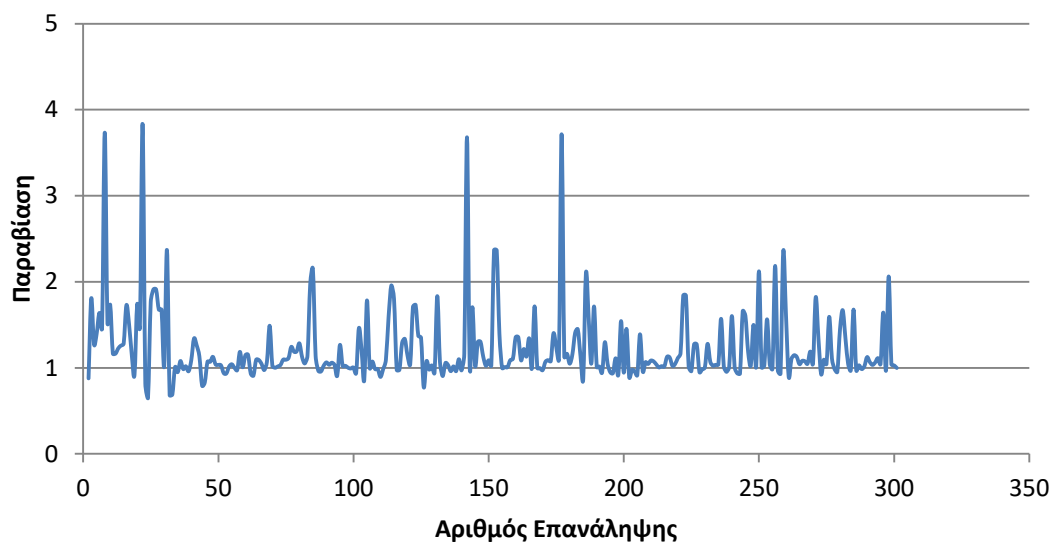
Αντίστοιχα με το προηγούμενο μοντέλο και σε αυτό, κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου MATSuMoTo, διαπιστώνεται ότι σε όλες τις επαναλήψεις υπάρχει μεγάλη διακύμανση τιμών, τόσο της αντικειμενικής συνάρτησης όσο και των παραβιάσεων.

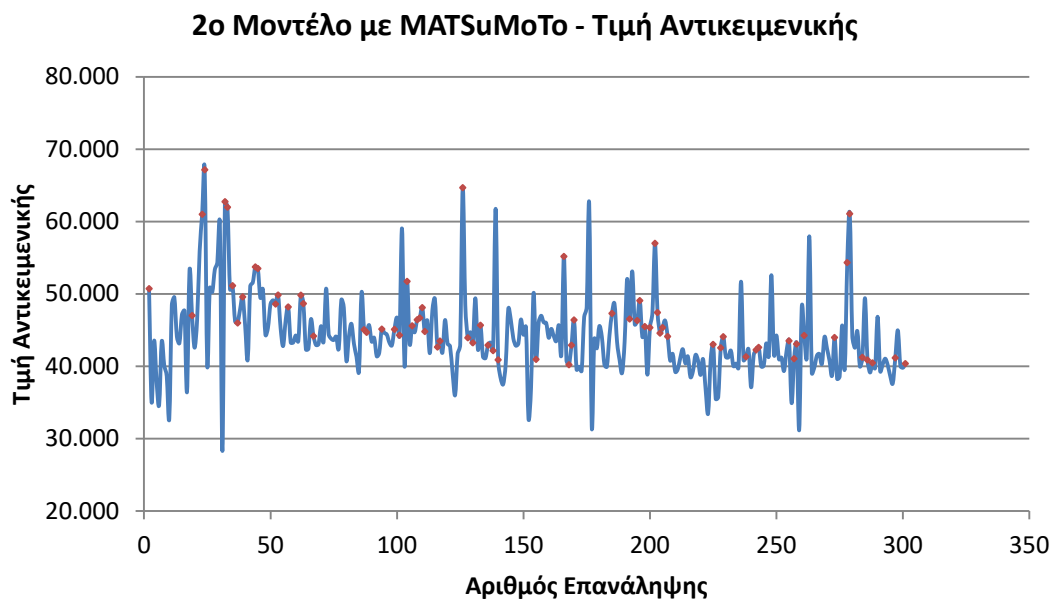
Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

2ο Μοντέλο με MATSuMoTo - Ποσοστό βελτίωσης



2ο Μοντέλο με MATSuMoTo - Παραβίαση





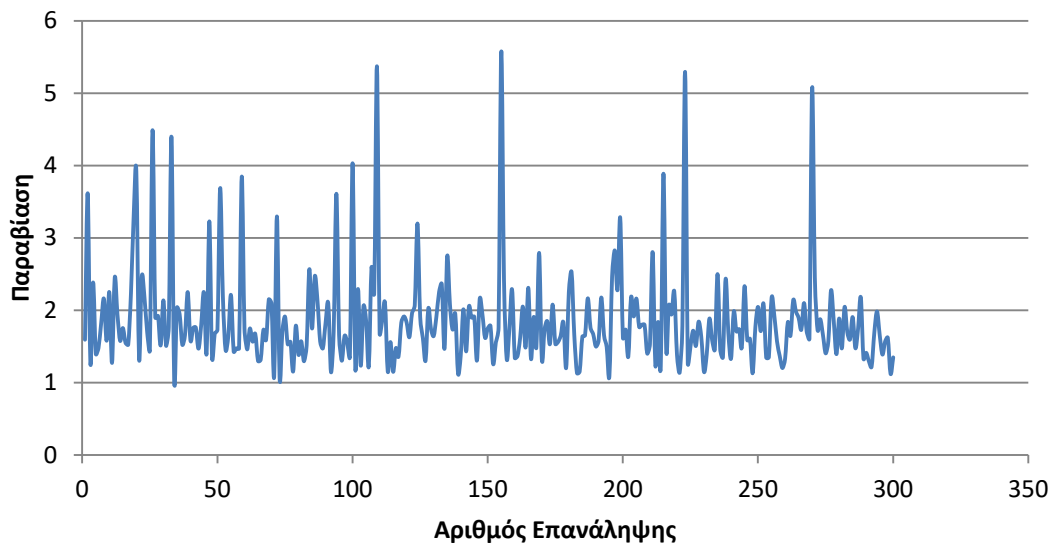
6.2.3 Εφαρμογή του αλγορίθμου Latin Hypercube

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης Latin Hypercube η πρώτη τιμή της αντικειμενικής συνάρτησης είναι 34.826 εμφανίζοντας μικρή παραβίαση 1,593. Συνολικά γίνονται 300 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί.

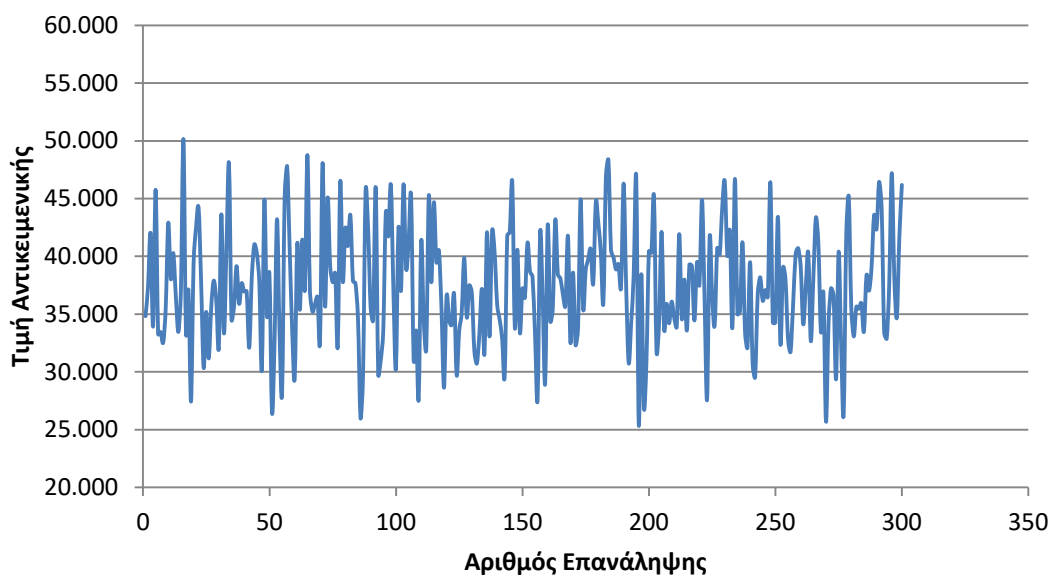
Γενικά ο αλγόριθμος καθ' όλη τη διάρκεια της εφαρμογής του δεν καταφέρνει να βρει καμία εφικτή λύση. Οι τιμές της αντικειμενικής έχουν μεγάλη διακύμανση σε όλο το εύρος των επαναλήψεων με μέγιστη τιμή στην 16^η επανάληψη με τιμή 50.142 (παραβίαση 1,549) και ελάχιστη στην 196^η επανάληψη με τιμή 25.428 (παραβίαση 2,468).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

2ο Μοντέλο με Latin Hypercube - Παραβίαση



2ο Μοντέλο με Latin Hypercube - Τιμή Αντικειμενικής



* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

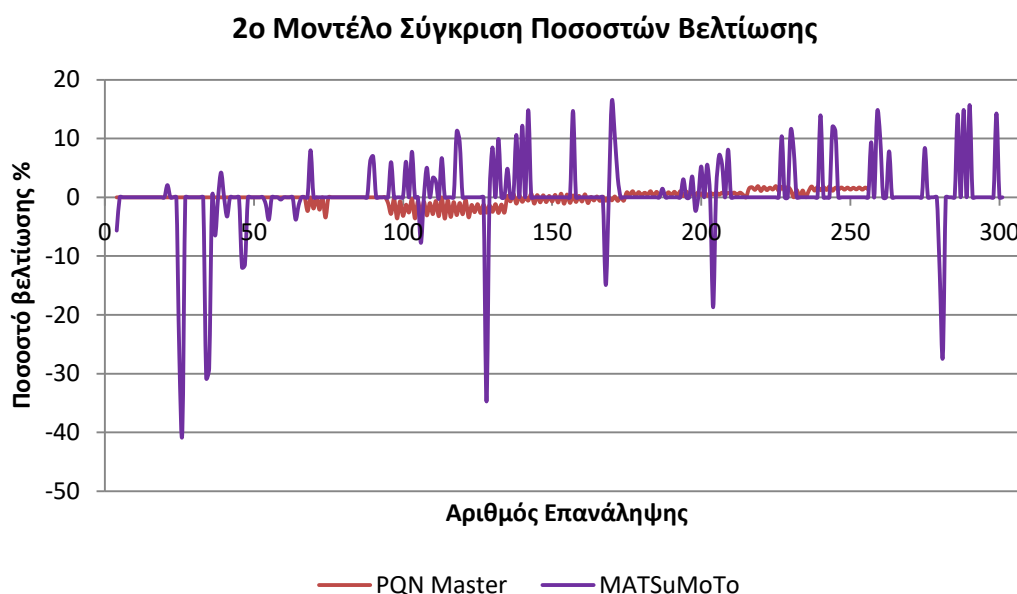
6.2.4 Συγκριτική ανάλυση αποτελεσμάτων

Όπως και στο προηγούμενο μοντέλο, κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου PQN, οι αντικειμενικές τιμές στις πρώτες επαναλήψεις εμφανίζουν μεγάλη διακύμανση γύρω από την τιμή σχεδιασμού, μέχρι την επανάληψη εκείνη που δίνει την πρώτη εφικτή λύση. Αντίστοιχα, κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου MATSuMoTo, διαπιστώνεται ότι σε όλες τις επαναλήψεις υπάρχει μεγάλη διακύμανση τιμών, τόσο της αντικειμενικής συνάρτησης όσο και των παραβιάσεων. Σε αυτήν την περίπτωση όμως η βέλτιστη τιμή αποδεικνύεται πολύ

«καλύτερη» από τη βέλτιστη τιμή που δίνει ο PQN, ο οποίος εδώ εγκλωβίστηκε σε κάποιο τοπικό βέλτιστο, σε αντίθεση με τον αλγόριθμο MATSuMoTo ο οποίος αναζητώντας συνεχώς καλύτερες λύσεις σε όλο το πεδίο σχεδιασμού τελικά εντόπισε πολύ καλύτερη λύση.

Κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου Latin Hypercube, διαπιστώνεται και πάλι ότι δεν καταφέρνει να βρει κάποια αποδεκτή λύση.

Προς επίρρωση των παραπάνω, στο επόμενο διάγραμμα και πίνακα απεικονίζεται ταυτόχρονα η πορεία όλων των αλγορίθμων κατά τη διαδικασία βελτιστοποίησης του μοντέλου.



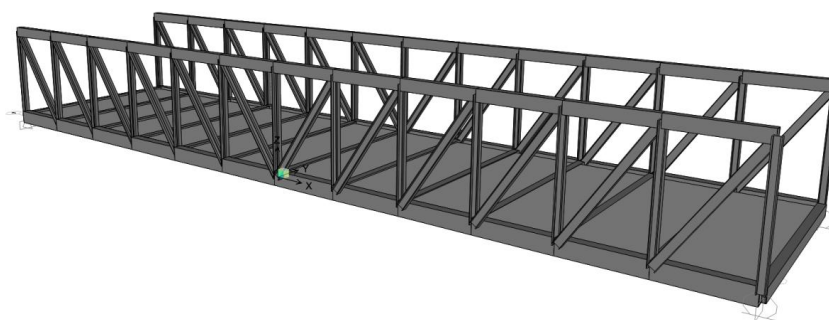
Συνοπτικός πίνακας σύγκρισης πορείας αλγορίθμων

	PQN	MATSuMoTo	Latin Hypercube
Μέγιστη τιμή	90.477	67.149	50.142
Ελάχιστη τιμή	12.786	28.273	25.428
Τιμή σχεδιασμού	47.847 (μη εφικτή)	48.001 (εφικτή)	-
Αριθμός Επαναλήψεων	256	301	300
Πρώτη εφικτή λύση	(68 ^η) 48.966 -2,34%	(19 ^η) 47.001 2,08%	-
Πρώτη εφικτή με τιμή μικρότερη του σχεδιασμού	(135 ^η) 47.740 0,22%	(19 ^η) 47.001 2,08%	-
Βέλτιστη τιμή	(225 ^η) 46.929	(168 ^η) 40.202	-
Ποσοστό βελτίωσης %	1,92%	16,25%	-

Πορεία αλγορίθμου	(135 ^η) 47.740 0,22%	(19 ^η) 47.001 2,08%	Δεν βρίσκει εφικτές λύσεις
	(153 ^η) 47.546 0,63%	(37 ^η) 45.976 4,22%	
	(175 ^η) 47.437 0,86%	(67 ^η) 44.160 8,00%	
	(193 ^η) 47.340 1,06%	(116 ^η) 42.651 11,15%	
	(216 ^η) 47.130 1,50%	(168 ^η) 40.202 16,25%	
	(225 ^η) 46.929 1,92%		

6.3 Ανάλυση 3^{ου} Μοντέλου

Το 3^ο μοντέλο, όπως προαναφέρθηκε, αφορά σε μία 3D γέφυρα κάτω διάβασης, με χαλύβδινο κατάστρωμα και χαλύβδινες δικτυωτές κύριες δοκούς. Αποτελείται συνολικά από 111 στοιχεία, 12 κελύφη και 52 κόμβους, το ύψος είναι 6m, το συνολικό μήκος 48m και το άνοιγμα του καταστρώματος 12m. Οι μεταλλικές διατομές των στοιχείων είναι από χάλυβα S355.



Οι διατομές που έχουν επιλεγεί κατά το σχεδιασμό του αρχικού μοντέλου είναι οι ακόλουθες:

- Άνω πέλμα κύριων δοκών κοίλης ορθογωνικής διατομής με $h=600\text{mm}$, $w=400\text{mm}$, $t_f=18\text{mm}$, $t_w=18\text{mm}$
- Κάτω πέλμα κύριων δοκών κοίλης ορθογωνικής διατομής με $h=800\text{mm}$, $w=500\text{mm}$, $t_f=12\text{mm}$, $t_w=12\text{mm}$
- Ορθοστάτες κύριων δοκών HEM300
- Διαγώνιοι κύριων δοκών HEA400
- Διαδοκίδες κοίλης ορθογωνικής διατομής με $h=600\text{mm}$, $w=400\text{mm}$, $t_f=18\text{mm}$, $t_w=18\text{mm}$
- Κατάστρωμα συμπαγούς διατομής πάχους 250mm.

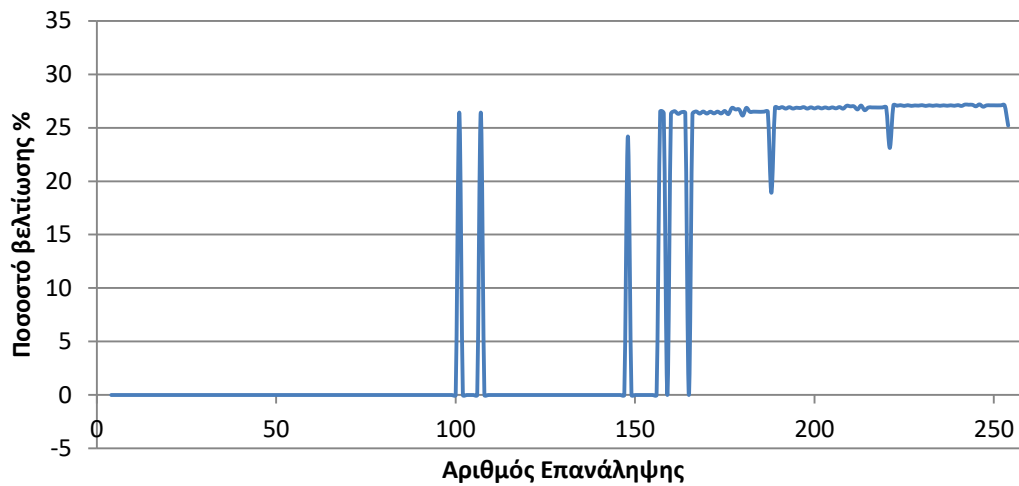
6.3.1 Εφαρμογή του αλγορίθμου PQN

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης PQN, η τιμή σχεδιασμού, δηλ. η τιμή της αντικειμενικής συνάρτησης του αρχικού μοντέλου πριν γίνει βελτιστοποίηση, είναι 1.151.303. Η επιλογή όμως των συγκεκριμένων (αρχικών) διατομών σχεδιασμού εμφανίζουν παραβίαση 1,914 γεγονός που καθιστά τη συγκεκριμένη λύση μη εφικτή. Συνολικά γίνονται 254 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping

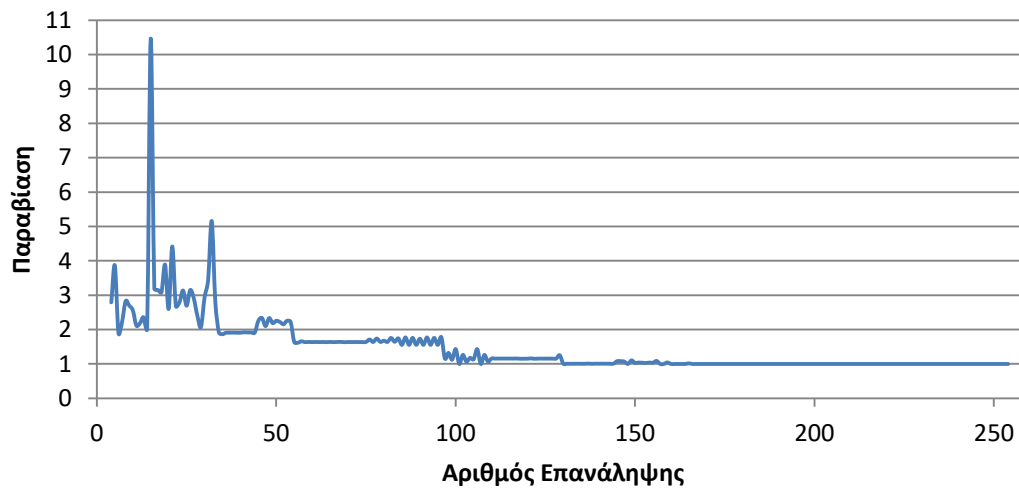
criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 246^η επανάληψη με τιμή 838.242 (βελτίωση σε ποσοστό 27,19% επί της τιμής σχεδιασμού).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

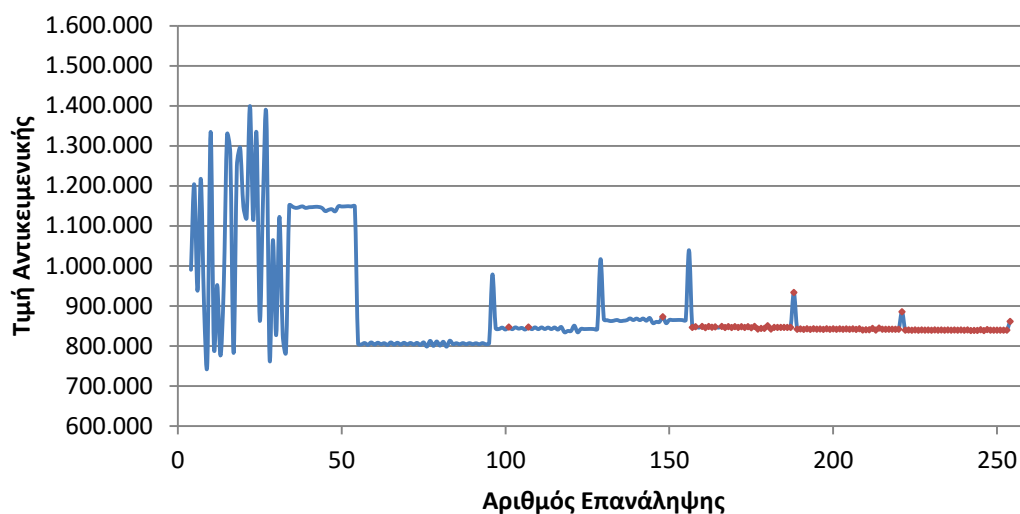
3ο Μοντέλο με PQN - Ποσοστό βελτίωσης



3ο Μοντέλο με PQN - Παραβίαση



3ο Μοντέλο με PQN - Τιμή Αντικειμενικής



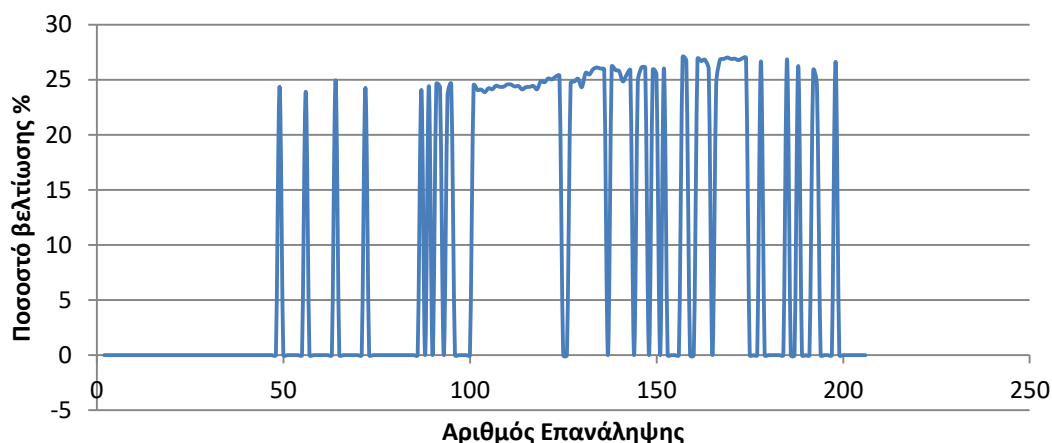
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.3.2 Εφαρμογή του αλγορίθμου MATSuMoTo

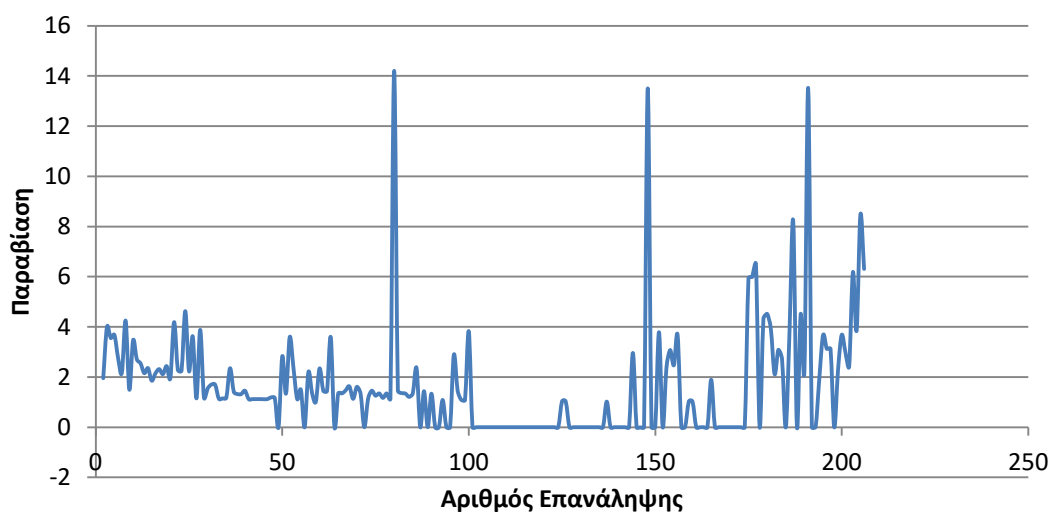
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης MATSuMoTo η τιμή σχεδιασμού είναι 1.150.588. Η επιλογή όμως των συγκεκριμένων (αρχικών) διατομών σχεδιασμού εμφανίζουν παραβίαση 1,85 γεγονός που καθιστά τη συγκεκριμένη λύση μη εφικτή. Συνολικά γίνονται 206 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 157^η επανάληψη με τιμή 839.367 (βελτίωση σε ποσοστό 27,05% επί της τιμής σχεδιασμού).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

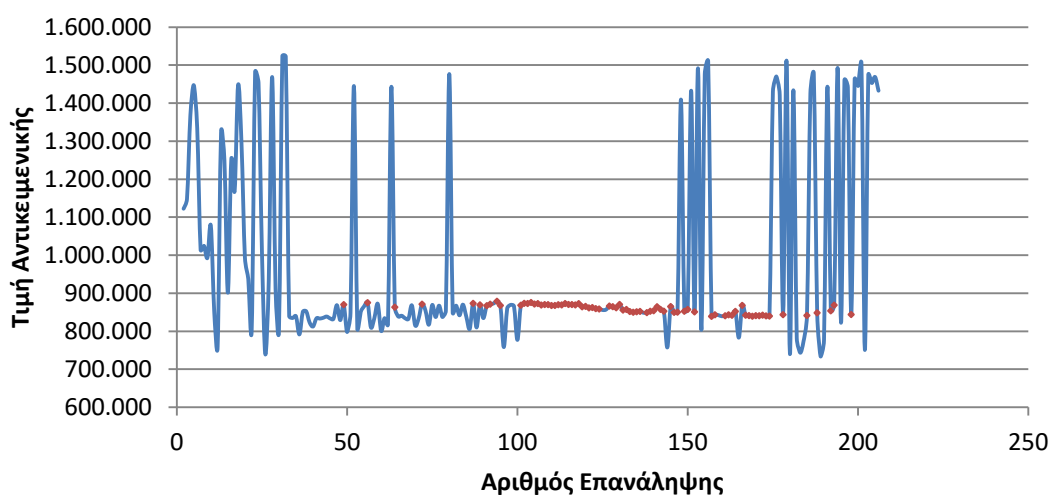
3ο Μοντέλο με MATSuMoTo - Ποσοστό βελτίωσης



3ο Μοντέλο με MATSuMoTo - Παραβίαση



3ο Μοντέλο με MATSuMoTo - Τιμή Αντικειμενικής



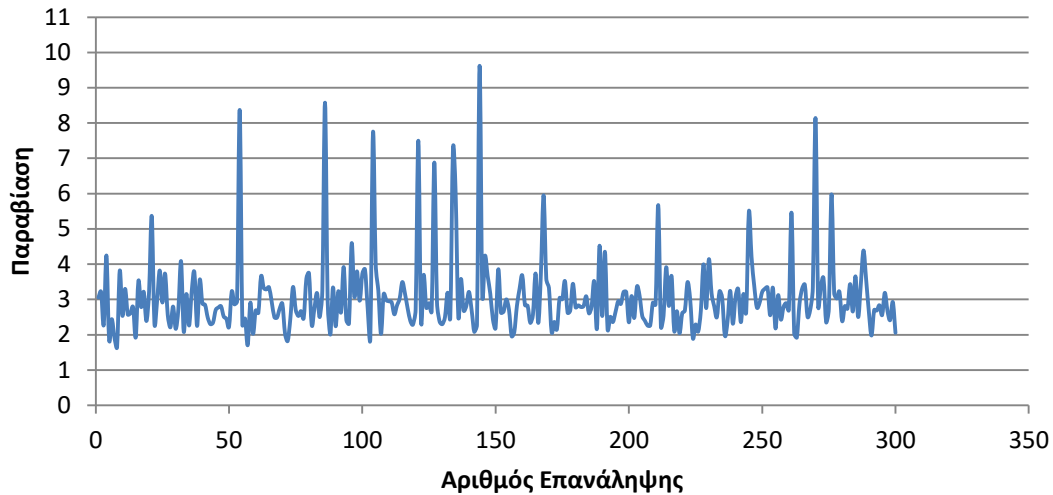
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.3.3 Εφαρμογή του αλγορίθμου Latin Hypercube

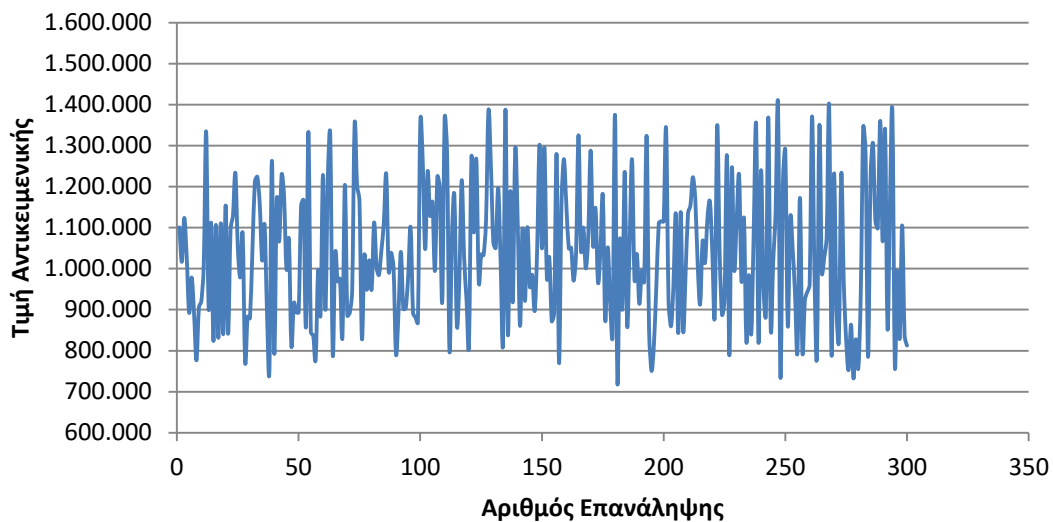
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης Latin Hypercube η πρώτη τιμή της αντικειμενικής συνάρτησης είναι 1.100.831 εμφανίζοντας παραβίαση 3,055. Συνολικά γίνονται 300 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Γενικά ο αλγόριθμος καθ' όλη τη διάρκεια της εφαρμογής του δεν καταφέρνει να βρει καμία εφικτή λύση. Οι τιμές της αντικειμενικής έχουν μεγάλη διακύμανση σε όλο το εύρος των επαναλήψεων με μέγιστη τιμή στην 247^η επανάληψη με τιμή 1.401.455 (παραβίαση 3,395) και ελάχιστη στην 181^η επανάληψη με τιμή 721.368 (παραβίαση 2,835).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

3ο Μοντέλο με Latin Hypercube - Παραβίαση



3ο Μοντέλο με Latin Hypercube - Τιμή Αντικειμενικής



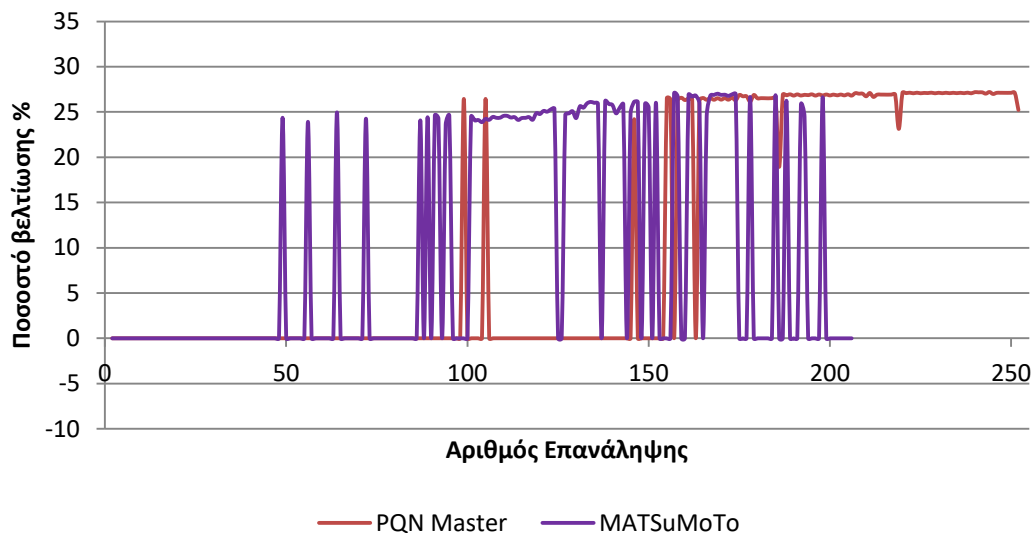
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.3.4 Συγκριτική ανάλυση αποτελεσμάτων

Στο συγκεκριμένο μοντέλο, τόσο κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου PQN, όσο και με εφαρμογή του αλγορίθμου MATSuMoTo, οι βέλτιστες λύσεις έχουν παραπλήσιες τιμές. Η κύρια διαφορά έγκειται στο ότι ο MATSuMoTo έφτασε πολύ γρηγορότερα σε αυτήν. Και πάλι, ο αλγόριθμος Latin Hypercube, δεν καταφέρνει να βρει κάποια αποδεκτή λύση.

Στο επόμενο διάγραμμα και πίνακα απεικονίζεται ταυτόχρονα η πορεία όλων των αλγορίθμων κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

3ο Μοντέλο Σύγκριση Ποσοστών Βελτίωσης

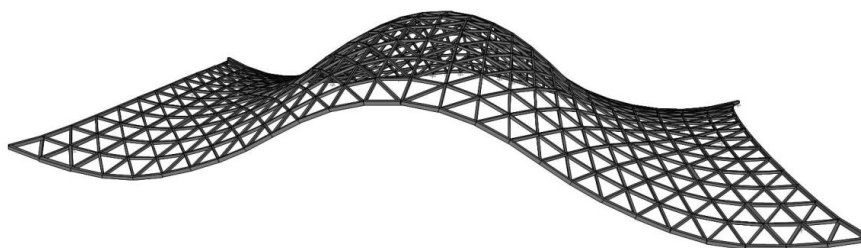


Συνοπτικός πίνακας σύγκρισης πορείας αλγορίθμων

	PQN	MATSuMoTo	Latin Hypercube
Μέγιστη τιμή	1.595.508	1.523.646	1.401.455
Ελάχιστη τιμή	688.366	733.501	721.368
Τιμή σχεδιασμού	1.151.303 (μη εφικτή)	1.150.588 (μη εφικτή)	-
Αριθμός Επαναλήψεων	254	206	300
Πρώτη εφικτή λύση	(101 ^η) 847.031 26,43%	(49 ^η) 870.251 24,36%	-
Πρώτη εφικτή με τιμή μικρότερη του σχεδιασμού	(101 ^η) 847.031 26,43%	(49 ^η) 870.251 24,36%	-
Βέλτιστη τιμή	(246 ^η) 838.242	(157 ^η) 839.367	-
Ποσοστό βελτίωσης %	27,19%	27,05%	-
Πορεία αλγορίθμου	(101 ^η) 847.031 26,43% (157 ^η) 846.174 26,50% (246 ^η) 838.242 27,19%	(49 ^η) 870.251 24,36% (157 ^η) 839.367 27,05%	Δεν βρίσκει εφικτές λύσεις

6.4 Ανάλυση 4^ο Μοντέλου

Το 4^ο μοντέλο, όπως προαναφέρθηκε, αφορά σε ένα στέγαστρο σχεδιασμένο ως χωροδικτύωμα με χαλύβδινα στοιχεία διαφόρων διατομών. Αποτελείται συνολικά από 1148 στοιχεία και 419 κόμβους και το μέγιστο ύψος είναι 16,6m. Τέλος έχει διαστάσεις σε κάτοψη 62 x 38,80 m.



Οι διατομές που έχουν επιλεγεί κατά το σχεδιασμό του αρχικού μοντέλου είναι όλες τυποποιημένες κοίλες ορθογωνικές από χάλυβα S355 ή S450 με τις κάτωθι διαστάσεις:

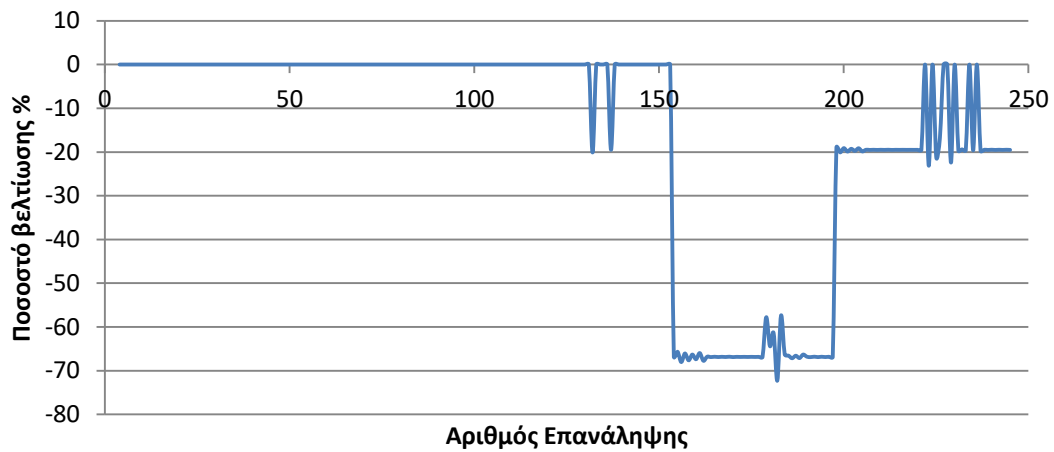
- TUBO200x140x14.2
- TUBO240x168x14.2
- TUBO280x196x22.2
- TUBO280x280x16
- TUBO300x210x25
- TUBO300x300x20

6.4.1 Εφαρμογή του αλγορίθμου PQN

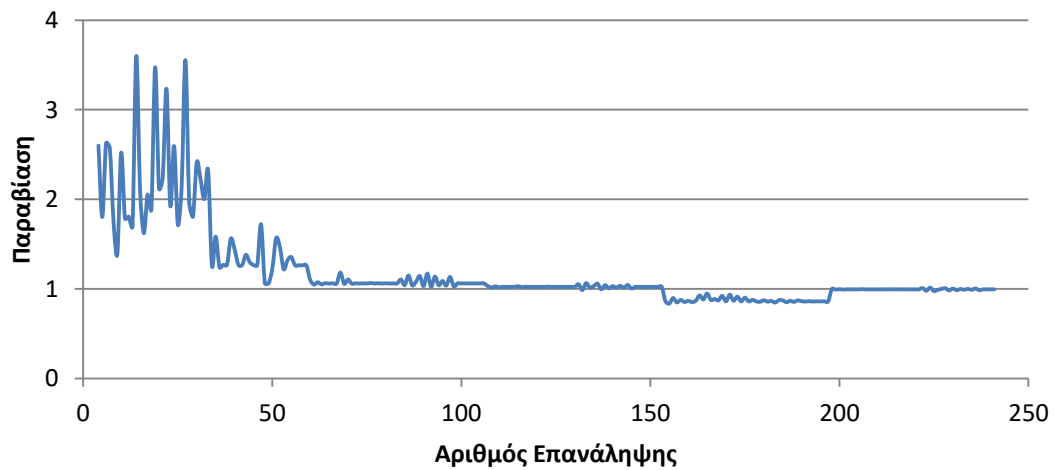
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης PQN, η τιμή σχεδιασμού είναι 209.290. Η επιλογή όμως των συγκεκριμένων (αρχικών) διατομών σχεδιασμού εμφανίζουν μικρή παραβίαση 1,263 γεγονός που καθιστά τη συγκεκριμένη λύση μη εφικτή. Συνολικά γίνονται 245 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης είναι μεγαλύτερη από την τιμή σχεδιασμού και εμφανίζεται στην 226^η επανάληψη με τιμή 244.490 (ποσοστό -16,82% επί της τιμής σχεδιασμού).

Ο αλγόριθμος βρίσκει αρκετές εφικτές λύσεις, όλες όμως με τιμή μεγαλύτερη της τιμής σχεδιασμού. Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

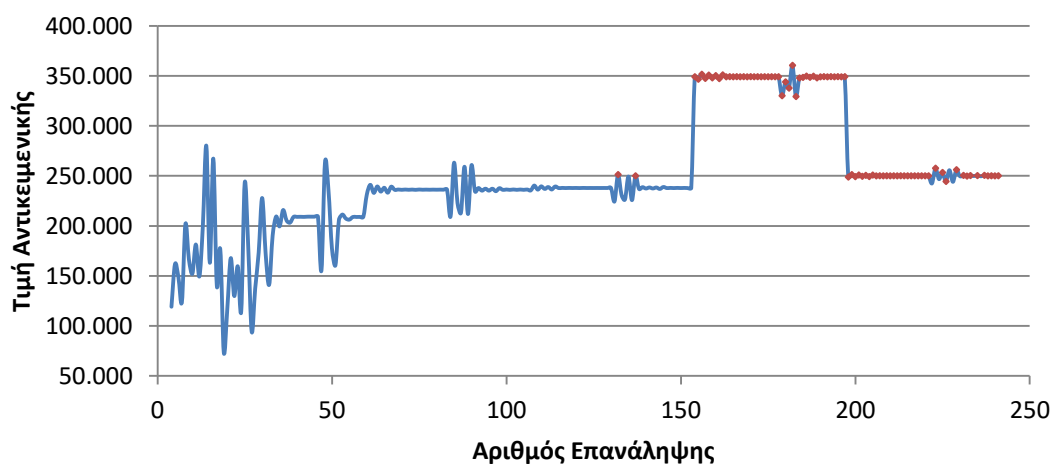
4ο Μοντέλο με RQN - Ποσοστό βελτίωσης



4ο Μοντέλο με RQN - Παραβίαση



4ο Μοντέλο με RQN - Τιμή Αντικειμενικής



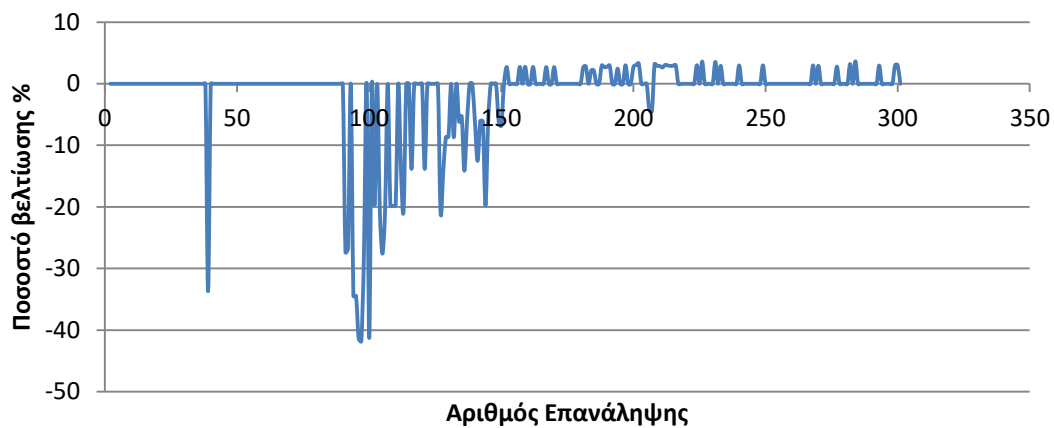
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.4.2 Εφαρμογή του αλγορίθμου MATSuMoTo

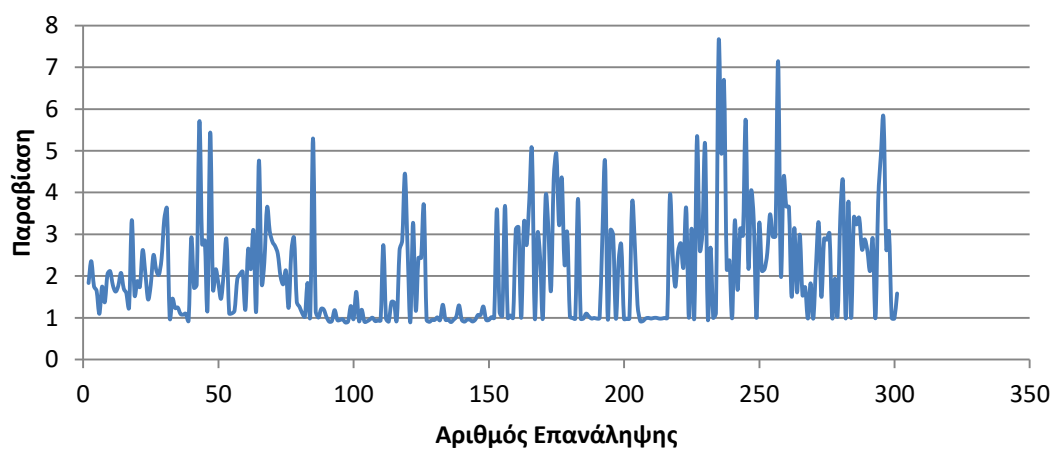
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης MATSuMoTo η τιμή σχεδιασμού είναι 225.743. Η επιλογή όμως των συγκεκριμένων (αρχικών) διατομών σχεδιασμού εμφανίζουν μικρή παραβίαση 1,186 γεγονός που καθιστά τη συγκεκριμένη λύση μη εφικτή. Συνολικά γίνονται 301 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 284^η επανάληψη με τιμή 217.476 (βελτίωση σε ποσοστό 3,66% επί της τιμής σχεδιασμού).

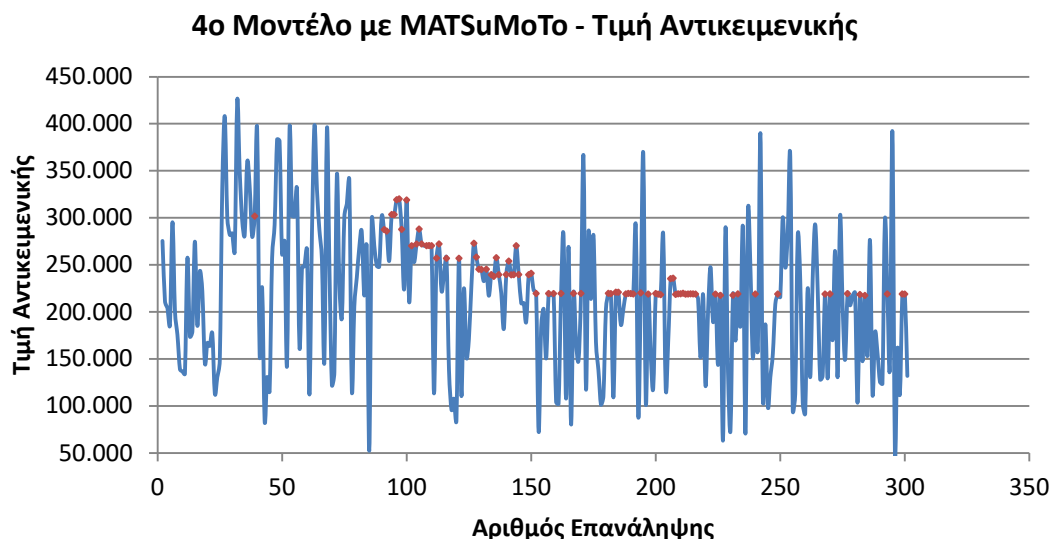
Γενικά ο αλγόριθμος παρουσιάζει αρκετές διακυμάνσεις σε όλο το εύρος των επαναλήψεων, δίνοντας αρκετές εφικτές λύσεις με τιμές όμως ελαφρά μικρότερες της τιμής σχεδιασμού. Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

4ο Μοντέλο με MATSuMoTo - Ποσοστό βελτίωσης



4ο Μοντέλο με MATSuMoTo - Παραβίαση





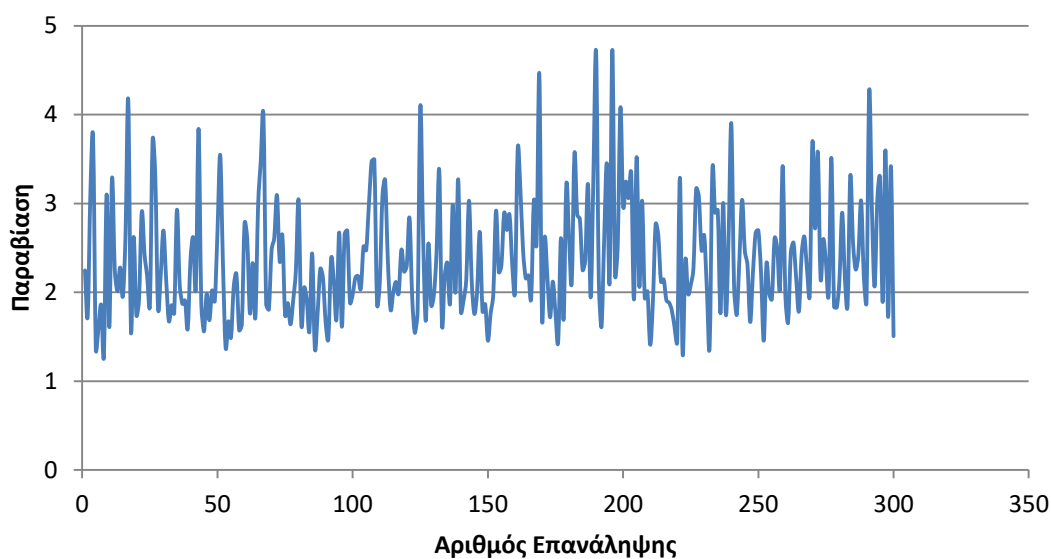
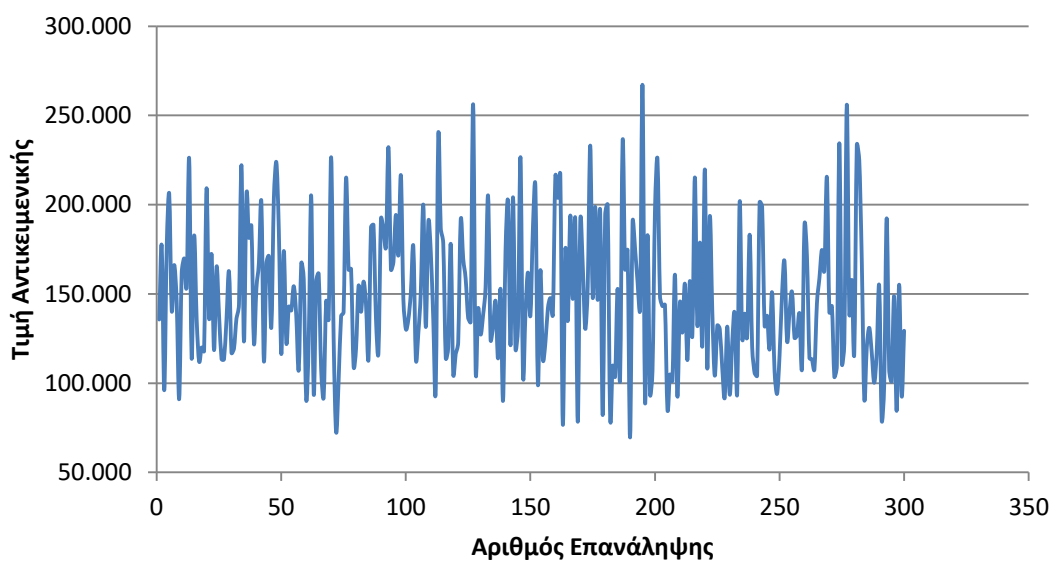
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.4.3 Εφαρμογή του αλγορίθμου Latin Hypercube

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης Latin Hypercube η πρώτη τιμή της αντικειμενικής συνάρτησης είναι 135.623 εμφανίζοντας παραβίαση 2,247. Συνολικά γίνονται 300 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί.

Γενικά ο αλγόριθμος καθ' όλη τη διάρκεια της εφαρμογής του δεν καταφέρνει να βρει καμία εφικτή λύση. Οι τιμές της αντικειμενικής έχουν μεγάλη διακύμανση σε όλο το εύρος των επαναλήψεων με μέγιστη τιμή στην 195^η επανάληψη με τιμή 266.464 (παραβίαση 2,107) και ελάχιστη στην 190^η επανάληψη με τιμή 69.556 (παραβίαση 4,71).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

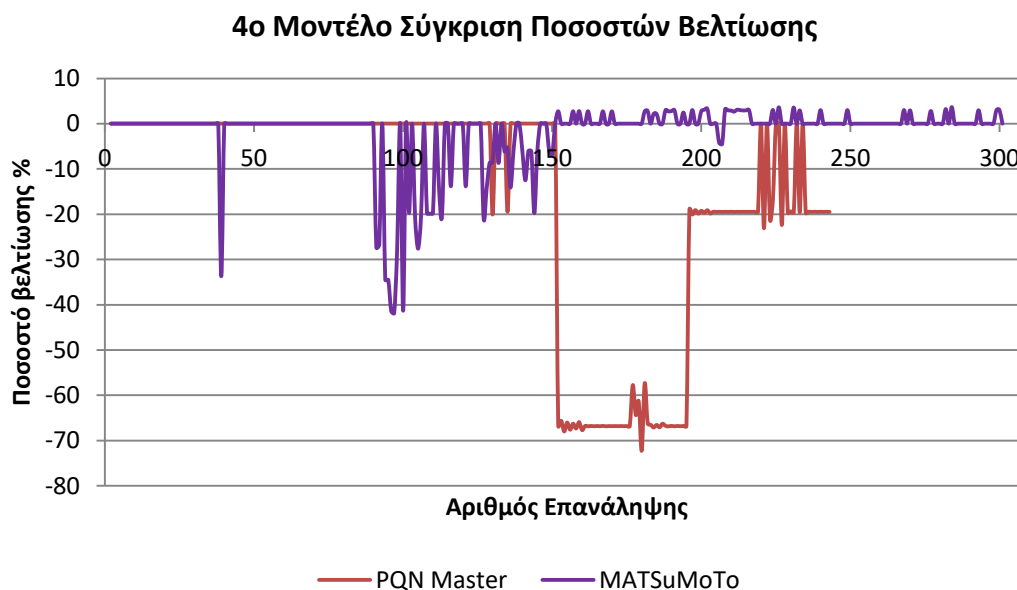
4ο Μοντέλο με Latin Hypercube - Παραβίαση**4ο Μοντέλο με Latin Hypercube - Τιμή Αντικειμενικής**

* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.4.4 Συγκριτική ανάλυση αποτελεσμάτων

Στο συγκεκριμένο μοντέλο ο αλγόριθμος PQN, αν και βρίσκει αρκετές εφικτές λύσεις, ωστόσο δεν μπορεί να εντοπίσει λύση με τιμή μικρότερη της τιμής σχεδιασμού, σε αντίθεση με τον MATSuMoTo που βρίσκει λίγο καλύτερη λύση από αυτή του σχεδιασμού. Και πάλι, ο αλγόριθμος Latin Hypercube, δεν καταφέρνει να βρει κάποια αποδεκτή λύση.

Στο επόμενο διάγραμμα και πίνακα απεικονίζεται ταυτόχρονα η πορεία όλων των αλγορίθμων κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

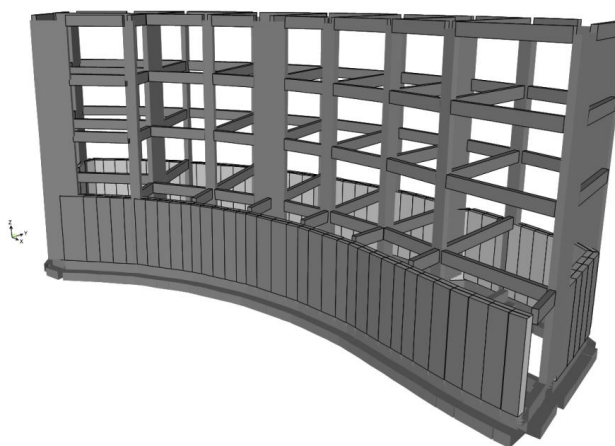


Συνοπτικός πίνακας σύγκρισης πορείας αλγορίθμων

	PQN	MATSuMoTo	Latin Hypercube
Μέγιστη τιμή	432.623	425.003	266.464
Ελάχιστη τιμή	44.905	51.391	69.556
Τιμή σχεδιασμού	209.290 (μη εφικτή)	225.743 (μη εφικτή)	-
Αριθμός Επαναλήψεων	245	301	300
Πρώτη εφικτή λύση	(132 ^η) 251.361 -20,10%	(39 ^η) 301.807 -33,69%	-
Πρώτη εφικτή με τιμή μικρότερη του σχεδιασμού	-	(152 ^η) 219.562 2,74%	-
Βέλτιστη τιμή	(226 ^η) 244.490	(284 ^η) 217.476	-
Ποσοστό βελτίωσης %	-16,82%	3,66%	-
Πορεία αλγορίθμου	Όλες οι λύσεις έχουν τιμές αντικειμενικής μεγαλύτερες από την τιμή σχεδιασμού	(152 ^η) 219.562 2,74% (284 ^η) 217.476 3,66%	Δεν βρίσκει εφικτές λύσεις

6.5 Ανάλυση 5^ο Μοντέλου

Το 5^ο μοντέλο, όπως προαναφέρθηκε, αφορά σε ένα σχολείο κατασκευασμένο από οπλισμένο σκυρόδεμα, με υπόγειο και τρεις ορόφους. Αποτελείται συνολικά από 378 ραβδωτά στοιχεία, 92 κελύφη και 255 κόμβους. Το ύψος του κάθε ορόφου είναι αντίστοιχα 3,50m (1^{ος}), 3,25m (2^{ος}) και 3,25m (3^{ος}). Το συνολικό ύψος από τη στάθμη του υπογείου μέχρι το δώμα είναι 14,00m. Τέλος έχει μέγιστες διαστάσεις σε κάτοψη 32,60 x 8,75 m. και συνολικό εμβαδό 275 m². Οι διατομές είναι από οπλισμένο σκυρόδεμα κατηγοριών C20/25, C25/30 και C30/37.



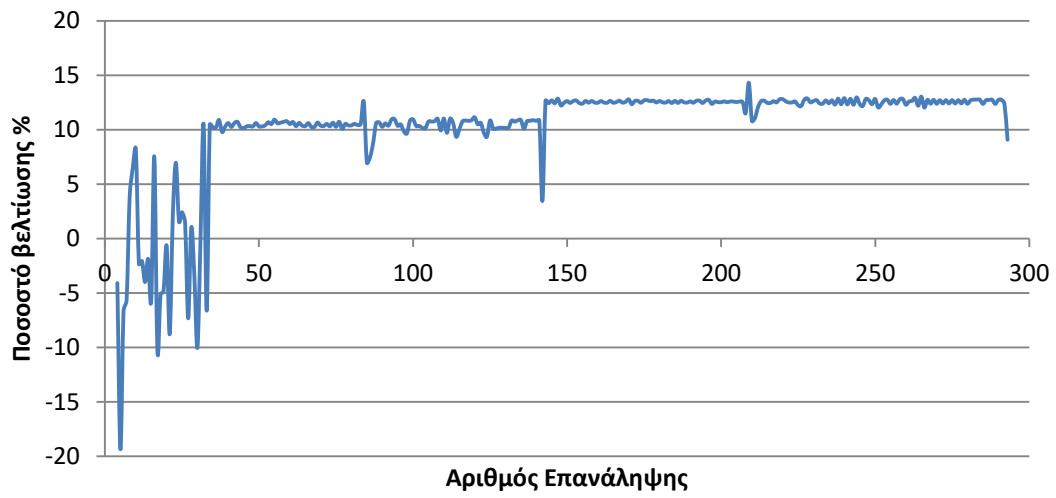
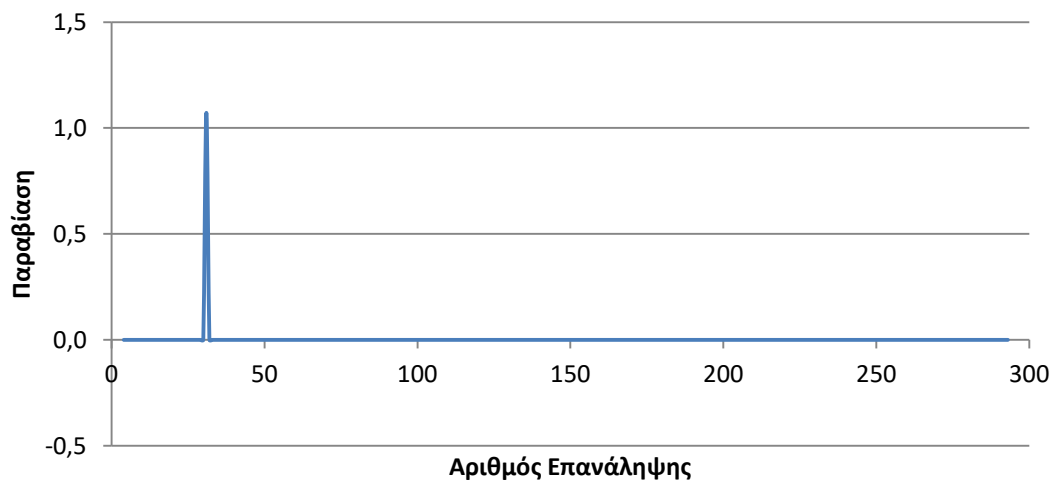
Οι διατομές που έχουν επιλεγεί κατά το σχεδιασμό του αρχικού μοντέλου είναι οι ακόλουθες:

- Δοκοί ορθογωνικής διατομής 600x250
- Υποστυλώματα και τοιχία ορθογωνικής διατομής διαφόρων διαστάσεων 2000x300, 600x300, 700x300, 700x400
- Περιμετρικό τοιχίο υπογείου πάχους 500mm
- Θεμελίωση διατομής T με $h=1100\text{mm}$, $w=1350\text{mm}$, $t_f=500\text{mm}$, $t_w=350\text{mm}$

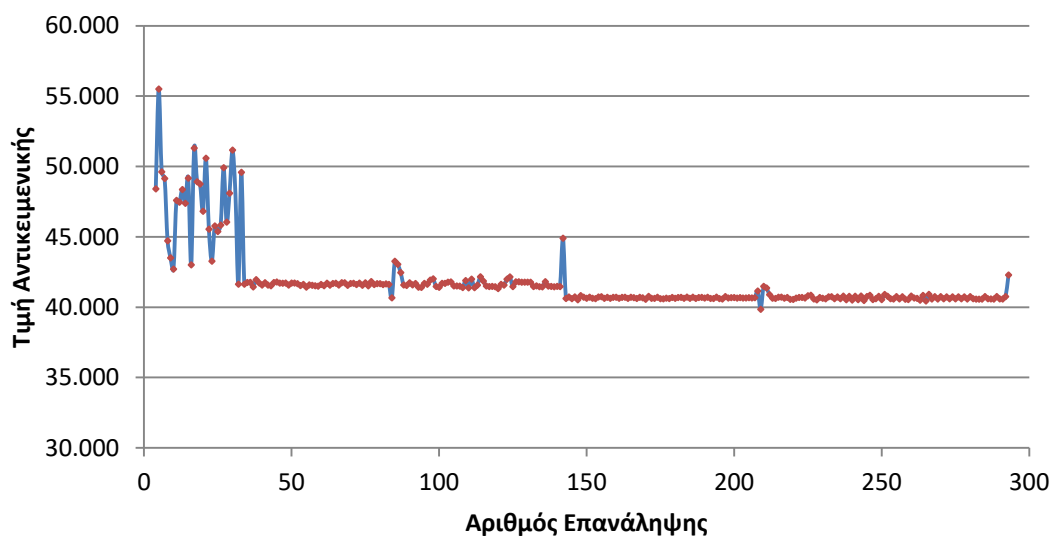
6.5.1 Εφαρμογή του αλγορίθμου PQN

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης PQN, η τιμή σχεδιασμού είναι 46.503. Συνολικά γίνονται 293 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 209^η επανάληψη με τιμή 39.849 (βελτίωση σε ποσοστό 14,31% επί της τιμής σχεδιασμού). Όλες οι λύσεις πλην της 31^{ης} επανάληψης είναι εφικτές.

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

5ο Μοντέλο με RQN - Ποσοστό βελτίωσης**5ο Μοντέλο με RQN - Παραβίαση**

5ο Μοντέλο με PQN - Τιμή Αντικειμενικής



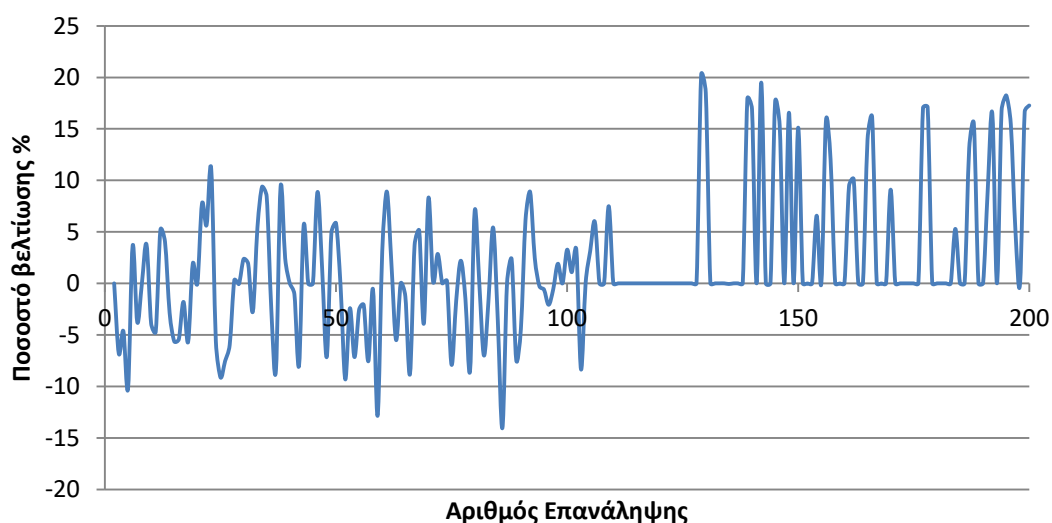
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.5.2 Εφαρμογή του αλγορίθμου MATSuMoTo

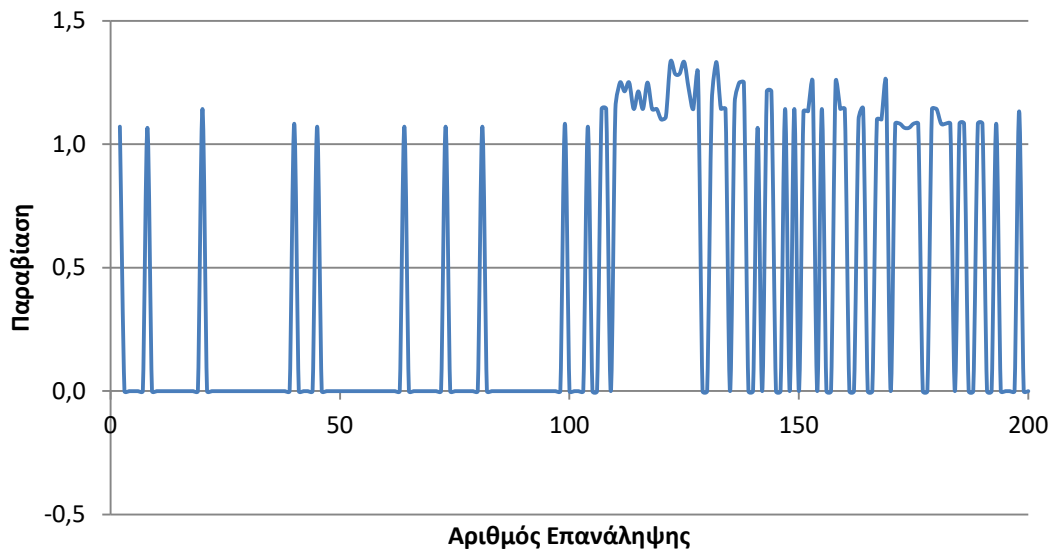
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης MATSuMoTo η τιμή σχεδιασμού είναι 46.503. Συνολικά γίνονται 200 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 129^η επανάληψη με τιμή 37.109 (βελτίωση σε ποσοστό 20,20% επί της τιμής σχεδιασμού).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

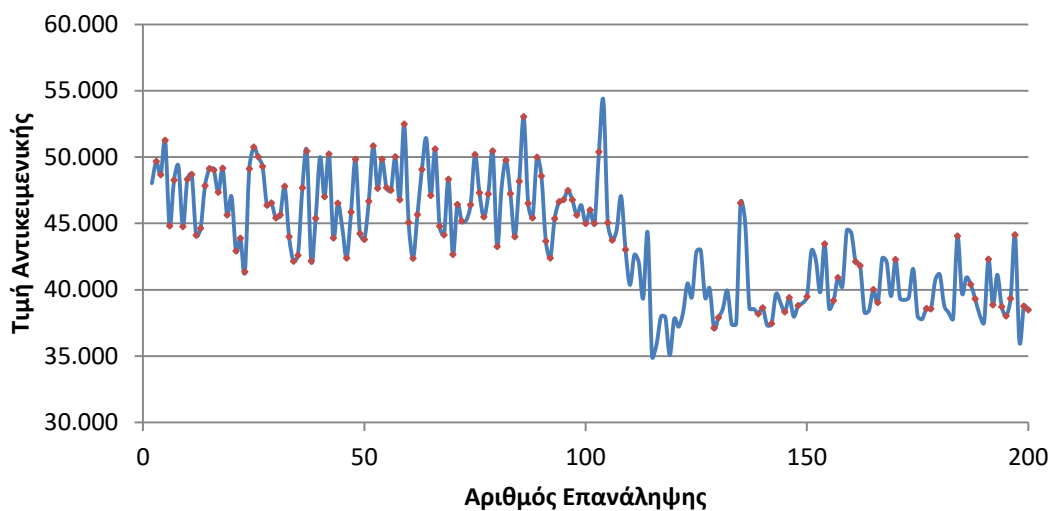
5ο Μοντέλο με MATSuMoTo - Ποσοστό βελτίωσης



5ο Μοντέλο με MATSuMoTo - Παραβίαση



5ο Μοντέλο με MATSuMoTo - Τιμή Αντικειμενικής



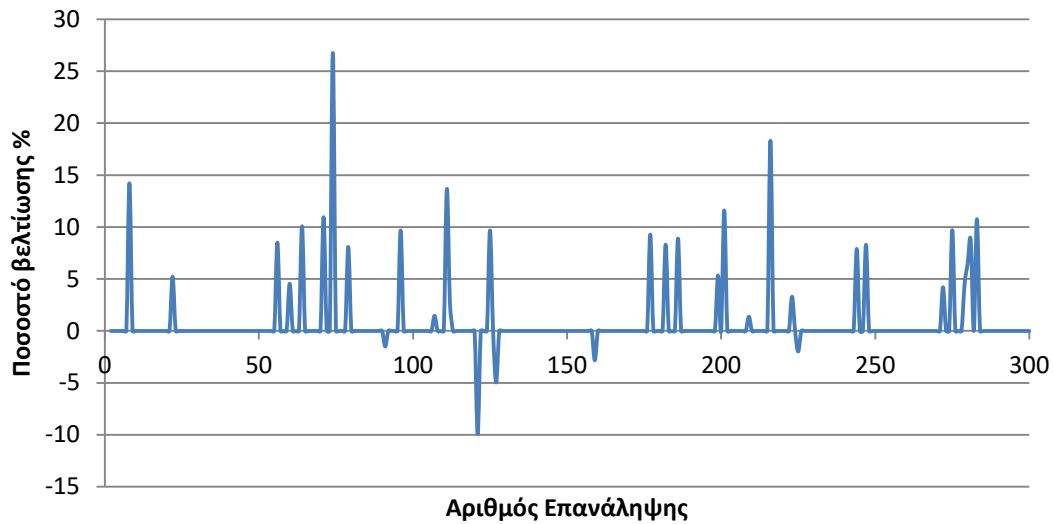
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.5.3 Εφαρμογή του αλγορίθμου Latin Hypercube

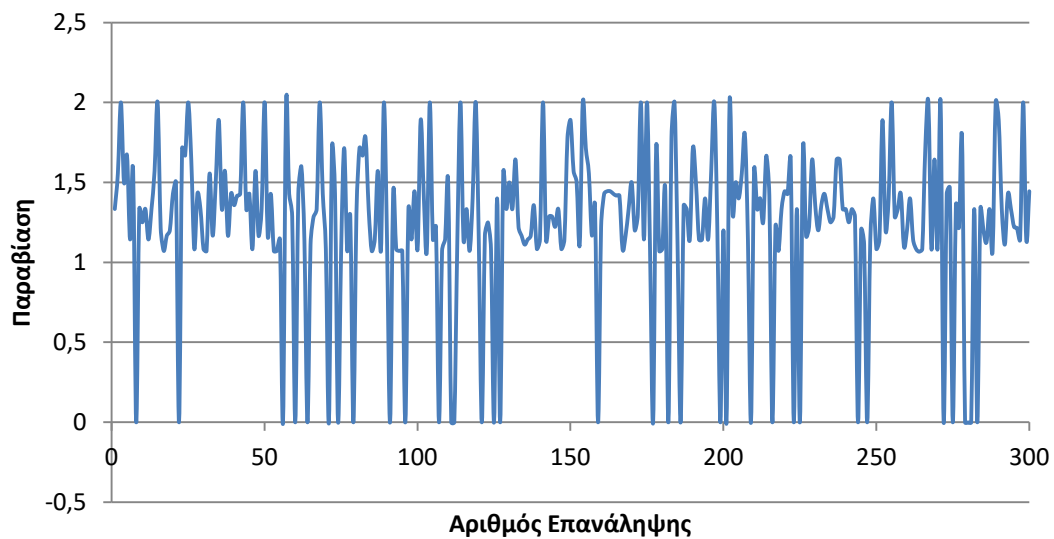
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης Latin Hypercube η πρώτη τιμή της αντικειμενικής συνάρτησης είναι 40.765 εμφανίζοντας παραβίαση 1,333. Συνολικά γίνονται 300 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 74^η επανάληψη με τιμή 34.062 (βελτίωση σε ποσοστό 16,44% με βάση την τιμή σχεδιασμού του αλγορίθμου PQN Master).

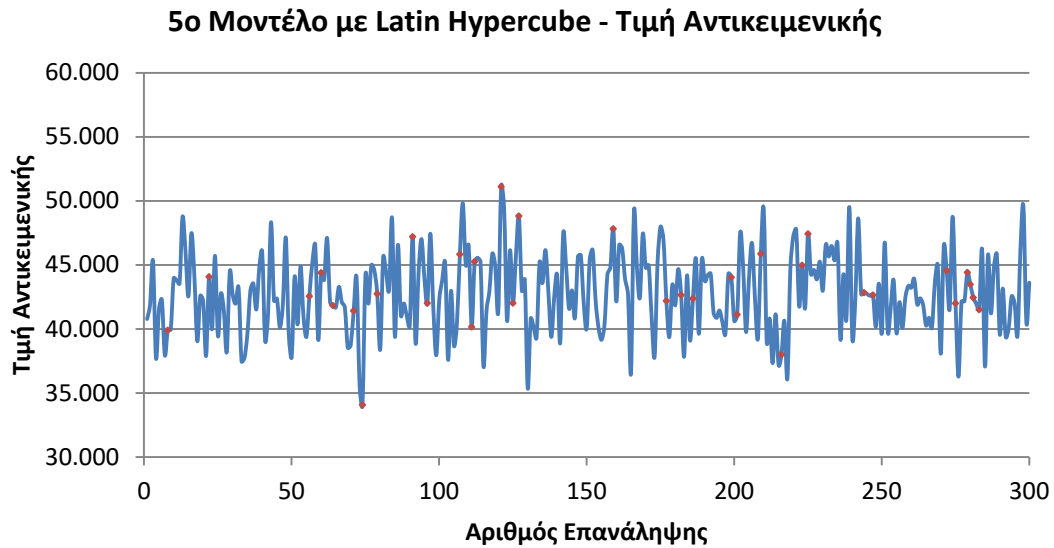
Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

5ο Μοντέλο με Latin Hypercube - Ποσοστό βελτίωσης



5ο Μοντέλο με Latin Hypercube - Παραβίαση



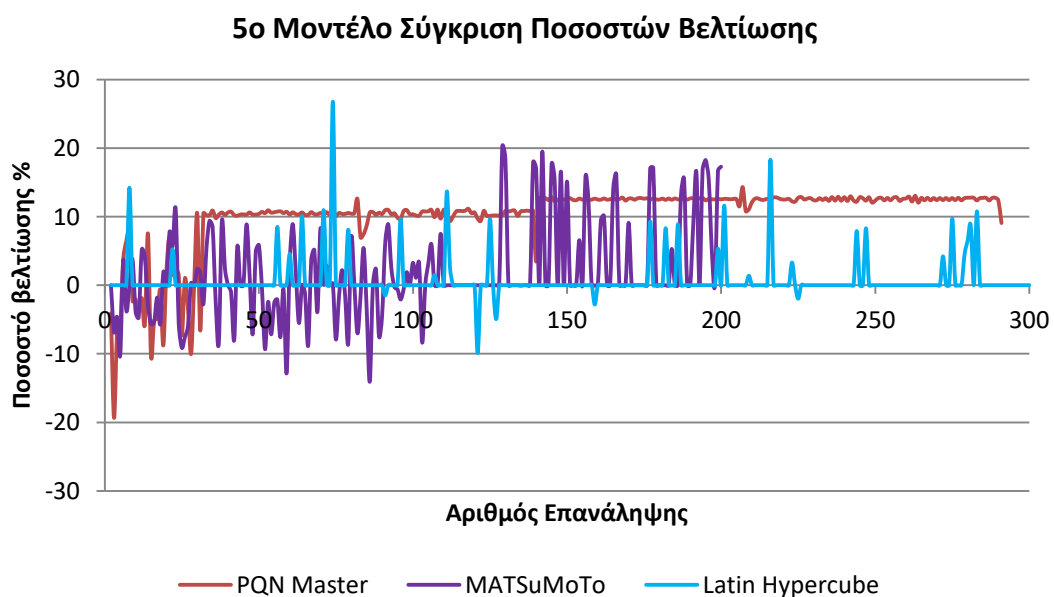


* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.5.4 Συγκριτική ανάλυση αποτελεσμάτων

Στο συγκεκριμένο μοντέλο ο αλγόριθμος MATSuMoTo βρίσκει αρκετά καλύτερη λύση από τον PQN και πολύ πιο γρήγορα. Όμως στην περίπτωση αυτή ο αλγόριθμος Latin Hypercube εντοπίζει τελικά την πιο καλή εφικτή λύση με την μικρότερη τιμή αντικειμενικής και από τις άλλες δύο μεθόδους.

Στο επόμενο διάγραμμα και πίνακα απεικονίζεται ταυτόχρονα η πορεία όλων των αλγορίθμων κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

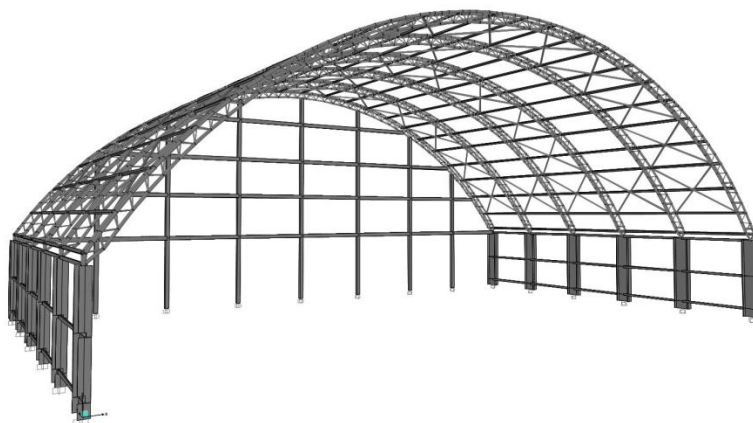


Συνοπτικός πίνακας σύγκρισης πορείας αλγορίθμων

	PQN	MATSuMoTo	Latin Hypercube
Μέγιστη τιμή	73.155	54.278	51.107
Ελάχιστη τιμή	27.995	34.967	34.062
Τιμή σχεδιασμού	46.503 (εφικτή)	46.503 (εφικτή)	46.503 (PQN Master)
Αριθμός Επαναλήψεων	293	200	300
Πρώτη εφικτή λύση	(4 ^η) 48.407 -4,10%	(4 ^η) 48.663 -4,64%	(8 ^η) 39.895 14,21%
Πρώτη εφικτή με τιμή μικρότερη του σχεδιασμού	(8 ^η) 44.717 3,84%	(6 ^η) 44.804 3,65%	(8 ^η) 39.895 14,21%
Βέλτιστη τιμή	(209 ^η) 39.849	(129 ^η) 37.109	(74 ^η) 34.062
Ποσοστό βελτίωσης %	14,31%	20,20%	26,75%
Πορεία αλγορίθμου	(8 ^η) 44.717 3,84% (9 ^η) 43.505 6,45% (10 ^η) 42.706 8,16% (32 ^η) 41.634 10,47% (84 ^η) 40.665 12,55% (209 ^η) 39.849 14,31%	(6 ^η) 44.804 3,65% (12 ^η) 44.102 5,16% (23 ^η) 41.343 11,09% (129 ^η) 37.109 20,20%	(8 ^η) 39.895 14,21% (74 ^η) 34.062 26,75%

6.6 Ανάλυση 6^{ου} Μοντέλου

Το 6^ο μοντέλο, όπως προαναφέρθηκε, αφορά σε ένα υπόστεγο από χαλύβδινα στοιχεία διαφόρων διατομών. Αποτελείται συνολικά από 3294 ραβδωτά στοιχεία και 1379 κόμβους, το ύψος μέχρι το καμπύλο στέγαστρο είναι 2,45m και το συνολικό ύψος 7,87m. Τέλος έχει διαστάσεις σε κάτοψη 12,35 x 19,94 m. και εμβαδό 246,26 m². Όλες οι μεταλλικές διατομές είναι από χάλυβα S355.



Οι διατομές που έχουν επιλεγεί κατά το σχεδιασμό του αρχικού μοντέλου είναι οι ακόλουθες:

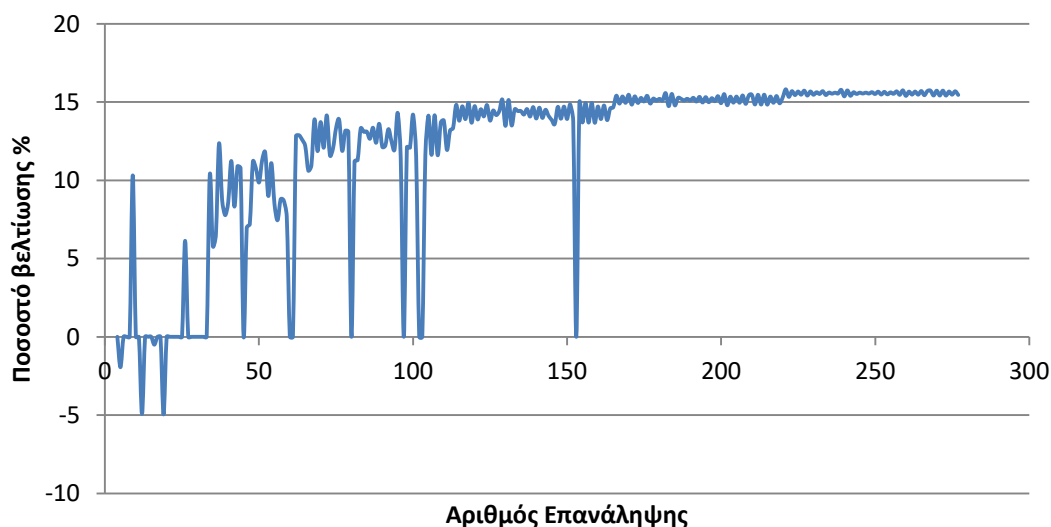
- Δοκοί οπίσθιας πλευράς κοίλης ορθογωνικής διατομής $h=125\text{mm}$, $w=75\text{mm}$, $t_f=6\text{mm}$, $t_w=6\text{mm}$
- Πλευρικοί δοκοί κοίλης ορθογωνικής διατομής $h=65\text{mm}$, $w=35\text{mm}$, $t_f=4\text{mm}$, $t_w=4\text{mm}$
- Υποστυλώματα οπίσθιας πλευράς κοίλης ορθογωνικής διατομής $h=125\text{mm}$, $w=125\text{mm}$, $t_f=9\text{mm}$, $t_w=9\text{mm}$
- Πλευρικά υποστυλώματα διατομής διπλού T $h=500\text{mm}$, $w_f=200\text{mm}$, $t_f=16\text{mm}$, $t_w=10.2\text{mm}$
- Δικτυωτές καμπύλες δοκοί κοίλης κυκλικής διατομής 48.3x5.4, 21.3x3.2 και 60.3x3.6
- Χιαστί σύνδεσμοι οροφής 48.3x5.4
- Τεγίδες κοίλης ορθογωνικής διατομής $h=65\text{mm}$, $w=35\text{mm}$, $t_f=3\text{mm}$, $t_w=3\text{mm}$

6.6.1 Εφαρμογή του αλγορίθμου PQN

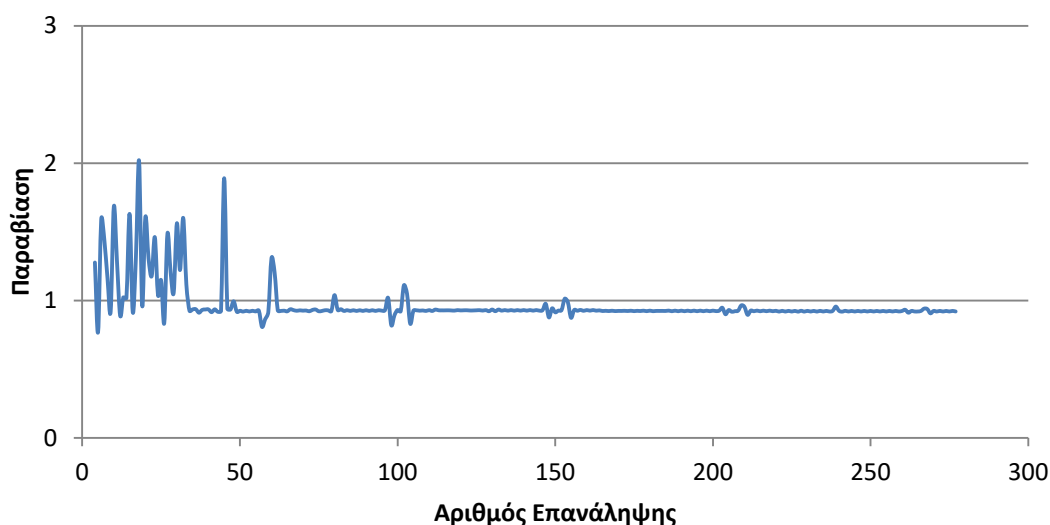
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης PQN, η τιμή σχεδιασμού είναι 10.180. Η επιλογή όμως των συγκεκριμένων (αρχικών) διατομών σχεδιασμού εμφανίζουν μικρή παραβίαση 1,072 γεγονός που καθιστά τη συγκεκριμένη λύση μη εφικτή. Συνολικά γίνονται 277 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 221^η επανάληψη με τιμή 8.570 (βελτίωση σε ποσοστό 15,82% επί της τιμής σχεδιασμού).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

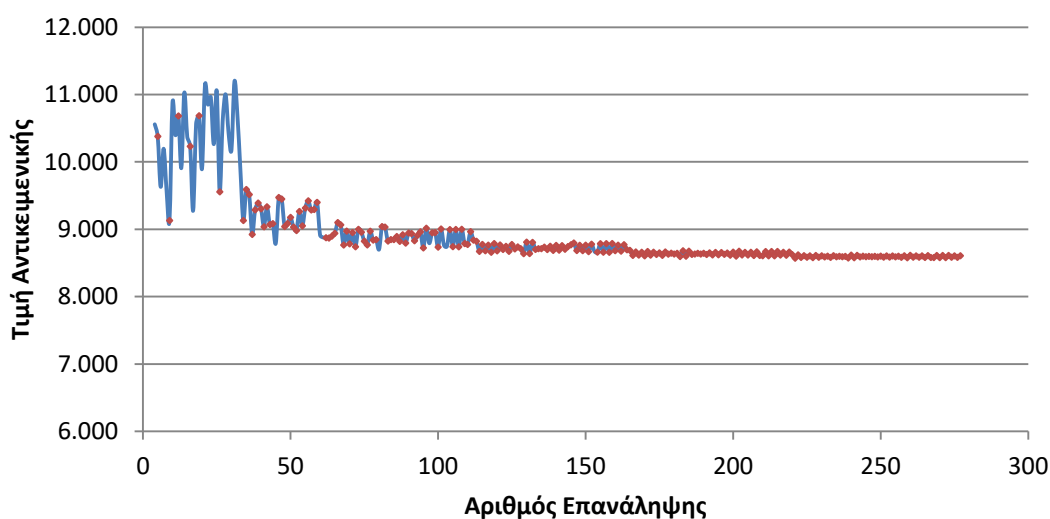
6ο Μοντέλο με PQN - Ποσοστό βελτίωσης



6ο Μοντέλο με PQN - Παραβίαση



6ο Μοντέλο με PQN - Τιμή Αντικειμενικής



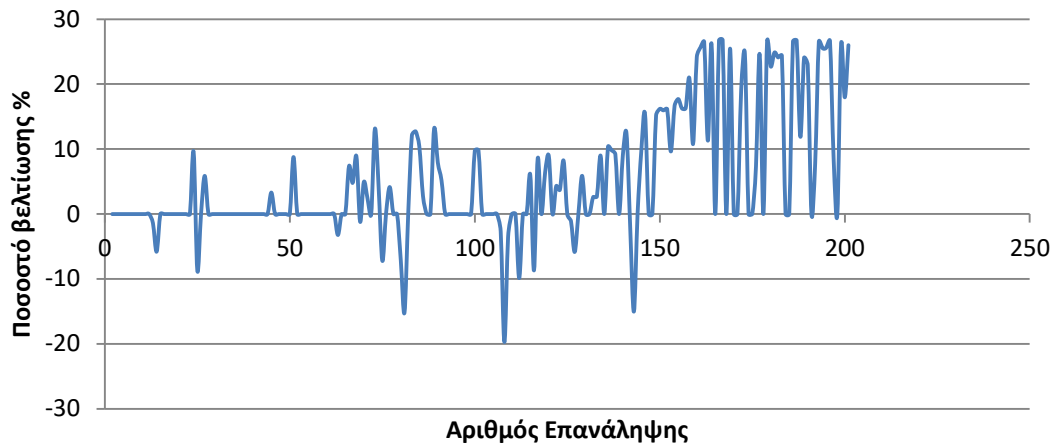
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.6.2 Εφαρμογή του αλγορίθμου MATSuMoTo

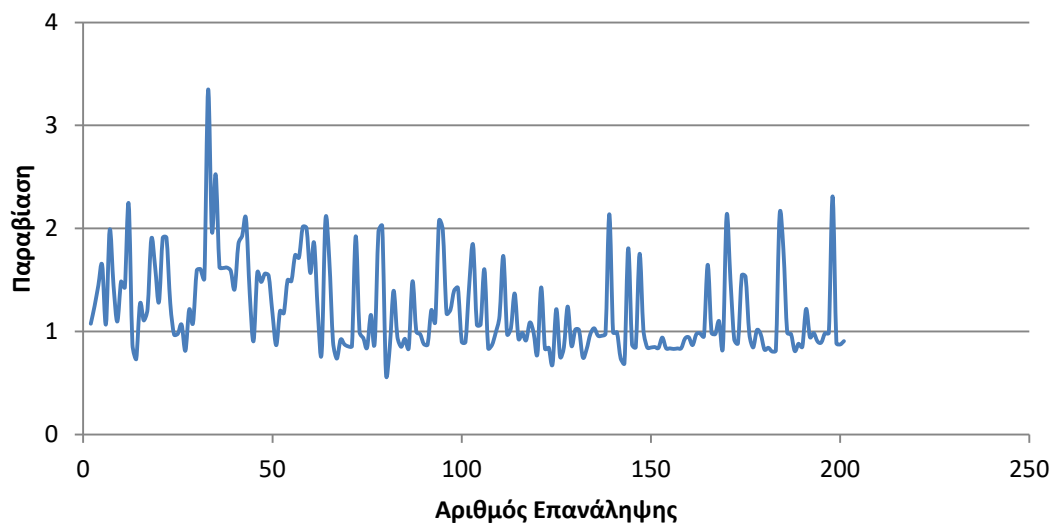
Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης MATSuMoTo η τιμή σχεδιασμού είναι 10.200. Η επιλογή όμως των συγκεκριμένων (αρχικών) διατομών σχεδιασμού εμφανίζουν μικρή παραβίαση 1,072 γεγονός που καθιστά τη συγκεκριμένη λύση μη εφικτή. Συνολικά γίνονται 201 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 167^η επανάληψη με τιμή 7.467 (βελτίωση σε ποσοστό 26,79% επί της τιμής σχεδιασμού).

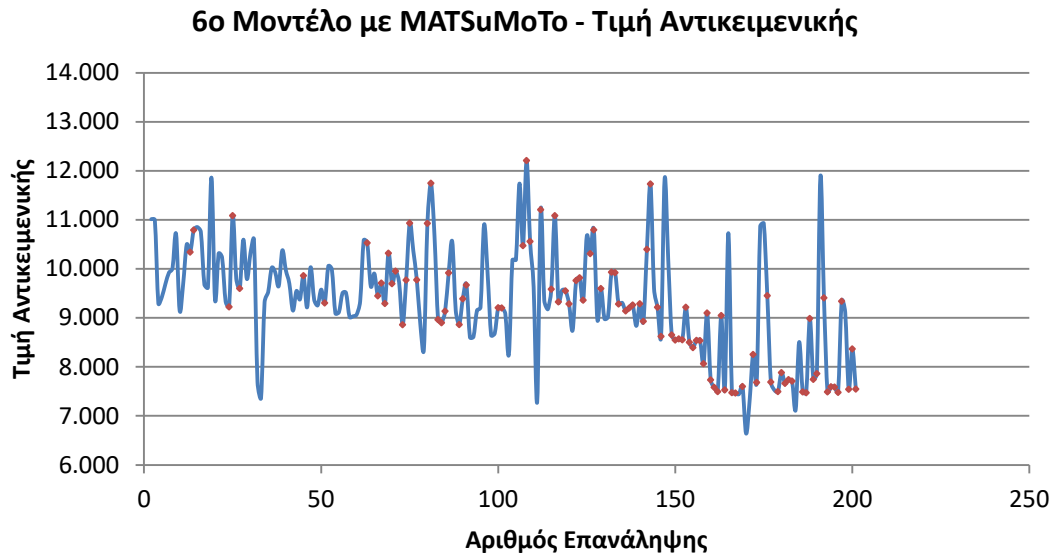
Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

6ο Μοντέλο με MATSuMoTo - Ποσοστό βελτίωσης



6ο Μοντέλο με MATSuMoTo - Παραβίαση



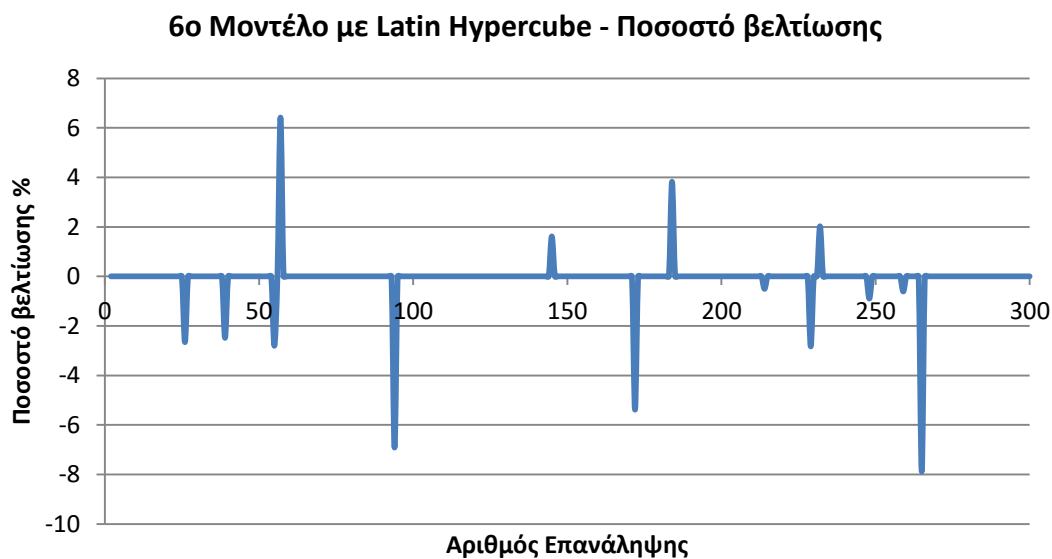


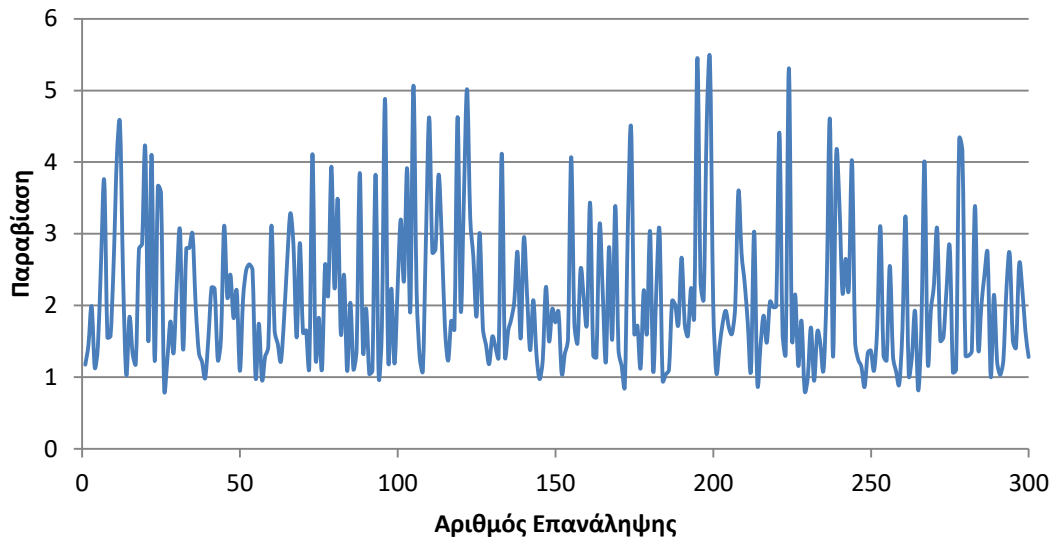
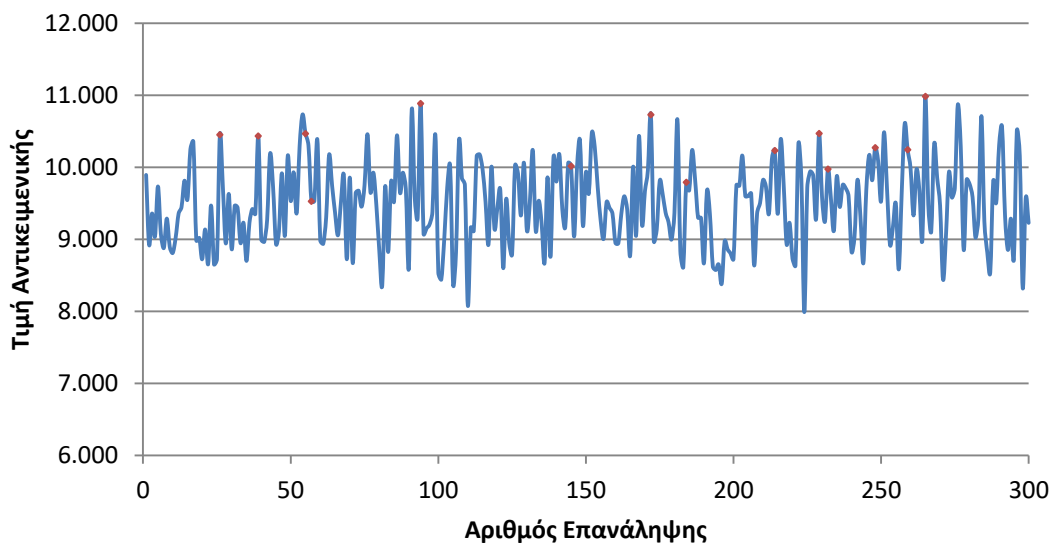
* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.6.3 Εφαρμογή του αλγορίθμου Latin Hypercube

Κατά την ανάλυση με εφαρμογή του αλγορίθμου βελτιστοποίησης Latin Hypercube η πρώτη τιμή της αντικειμενικής συνάρτησης είναι 9.892 εμφανίζοντας παραβίαση 1,174. Συνολικά γίνονται 300 επαναλήψεις πριν σταματήσει ο αλγόριθμος να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Η βέλτιστη τιμή της αντικειμενικής συνάρτησης εμφανίζεται στην 57^η επανάληψη με τιμή 9.527 (βελτίωση σε ποσοστό 3,69% με βάση την τιμή σχεδιασμού του αλγορίθμου PQN Master).

Στα επόμενα διαγράμματα απεικονίζεται η πορεία του συγκεκριμένου αλγορίθμου κατά τη διαδικασία βελτιστοποίησης του μοντέλου.



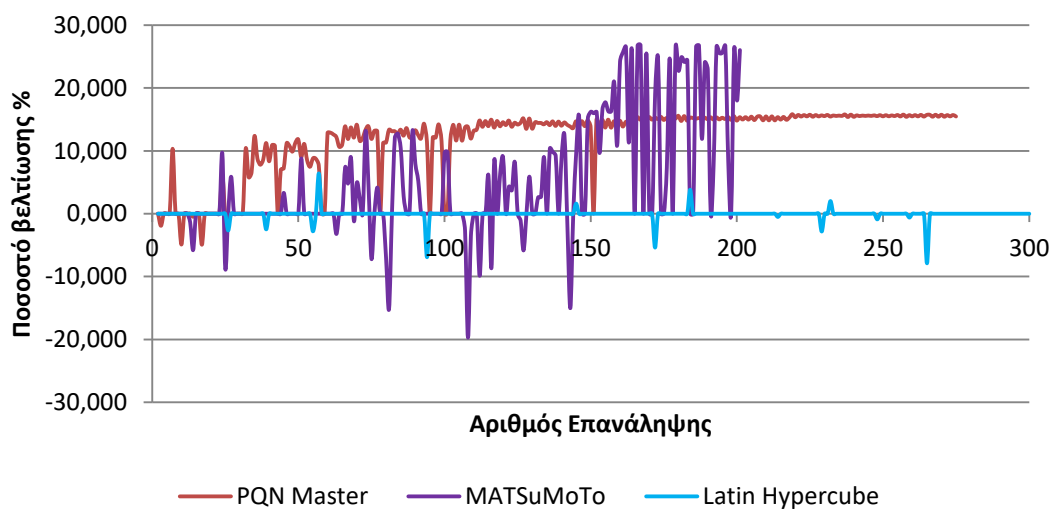
6ο Μοντέλο με Latin Hypercube - Παραβίαση**6ο Μοντέλο με Latin Hypercube - Τιμή Αντικειμενικής**

* Με κόκκινο χρώμα υποδεικνύονται οι εφικτές λύσεις (Παραβίαση μικρότερη ή ίση με 1,0)

6.6.4 Συγκριτική ανάλυση αποτελεσμάτων

Στο συγκεκριμένο μοντέλο ο αλγόριθμος MATSuMoTo βρίσκει αρκετά καλύτερη λύση τόσο από τον PQN όσο και από τον Latin Hypercube. Στο επόμενο διάγραμμα και πίνακα απεικονίζεται ταυτόχρονα η πορεία όλων των αλγορίθμων κατά τη διαδικασία βελτιστοποίησης του μοντέλου.

6ο Μοντέλο Σύγκριση Ποσοτών Βελτίωσης

Συνοπτικός πίνακας σύγκρισης πορείας αλγορίθμων

	PQN	MATSuMoTo	Latin Hypercube
Μέγιστη τιμή	17.310	12.210	10.984
Ελάχιστη τιμή	4.854	6.640	7.988
Τιμή σχεδιασμού	10.180 (μη εφικτή)	10.200 (μη εφικτή)	10.180 (PQN Master)
Αριθμός Επαναλήψεων	277	201	300
Πρώτη εφικτή λύση	(5 ^η) 10.378 -1,95%	(13 ^η) 10.341 -1,39%	(26 ^η) 10.451 -2,66%
Πρώτη εφικτή με τιμή μικρότερη του σχεδιασμού	(9 ^η) 9.130 10,32%	(24 ^η) 9.223 9,57%	(57 ^η) 9.527 6,42%
Βέλτιστη τιμή	(221 ^η) 8.570	(167 ^η) 7.467	(57 ^η) 9.527
Ποσοστό βελτίωσης %	15,82%	26,79%	6,42%
Πορεία αλγορίθμου	(9 ^η) 9.130 10,32% (37 ^η) 8.923 12,35% (62 ^η) 8.874 12,83% (68 ^η) 8.766 13,89% (72 ^η) 8.739 14,15% (221 ^η) 8.570 15,82%	(24 ^η) 9.223 9,57% (73 ^η) 8.860 13,14% (146 ^η) 8.621 15,48% (158 ^η) 8.065 20,93% (167 ^η) 7.467 26,79%	(57 ^η) 9.527 6,42%

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως προαναφέρθηκε διαπιστώθηκε ότι, κάθε αλγόριθμος συμπεριφέρεται τελείως διαφορετικά ανάλογα με τα χαρακτηριστικά της κάθε κατασκευής. Έτσι, κάποιοι αλγόριθμοι έδωσαν εξαιρετικά αποτελέσματα όταν εφαρμόστηκαν σε ένα συγκεκριμένο δομικό σύστημα αλλά οι ίδιοι αλγόριθμοι, σε άλλη κατασκευή, είτε εμφάνισαν «κακή» επίδοση είτε δεν κατάφεραν ποτέ να βρουν εφικτές λύσεις κάτω από τους τιθέμενους περιορισμούς της αντικειμενικής συνάρτησης. Στον επόμενο πίνακα απεικονίζονται συνοπτικά τα ποσοστά βελτίωσης που πέτυχε κάθε αλγόριθμος ανά μοντέλο μελέτης.

Πίνακας σύγκρισης ποσοστών βελτίωσης

	PQN	MATSuMoTo	Latin Hypercube
1^ο Μοντέλο	28,90%	24,93%	-
2^ο Μοντέλο	1,92%	16,25%	-
3^ο Μοντέλο	27,19%	27,05%	-
4^ο Μοντέλο	-16,82%	3,66%	-
5^ο Μοντέλο	14,31%	20,20%	26,75%
6^ο Μοντέλο	15,82%	26,79%	6,42%

Κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου PQN διαπιστώνεται ο πρακτικά ντετερμινιστικός χαρακτήρας αυτού βασισμένος σε gradient-based μεθόδους αν και ο ίδιος ο αλγόριθμος ανήκει στις derivative-free μεθόδους. Οι τιμές της αντικειμενικής συνάρτησης στις πρώτες επαναλήψεις εμφανίζουν μεγάλη διακύμανση γύρω από την τιμή σχεδιασμού, μέχρι την επανάληψη εκείνη που εντοπίζει την πρώτη εφικτή λύση. Από εκεί και μετά, ο αλγόριθμος αναζητάει «καλύτερες» εφικτές λύσεις σε σημεία κοντά στο υπάρχον τοπικό βέλτιστο, μειώνοντας έτσι συνεχώς την τιμή της αντικειμενικής συνάρτησης μέχρι να βρει την βέλτιστη τιμή και φυσικά ωστόσο σταματήσει να «τρέχει» με βάσει τα κριτήρια τερματισμού (stopping criterion) που έχουν τεθεί. Συχνά όμως εγκλωβίζεται σε τοπικά βέλτιστα και λόγω της δομής του δεν καταφέρνει να εντοπίσει το καθολικό βέλτιστο που αποτελεί και το ζητούμενο.

Σε αντίθεση με τα παραπάνω, κατά την βελτιστοποίηση με εφαρμογή του αλγορίθμου MATSuMoTo διαπιστώνεται ο καθαρά στοχαστικός χαρακτήρας του ως derivative-free μέθοδος. Σε όλες τις επαναλήψεις υπάρχει μεγάλη διακύμανση τιμών, τόσο της αντικειμενικής συνάρτησης όσο και των παραβιάσεων. Εντοπίζει πολύ πιο γρήγορα τη βέλτιστη τιμή η οποία όπως αποδείχθηκε τελικά είναι στις περισσότερες περιπτώσεις «καλύτερη» από την βέλτιστη τιμή που δίνουν οι άλλοι αλγόριθμοι.

Τέλος, κατά την βελτιστοποίηση με απλή εφαρμογή του αλγορίθμου Latin Hypercube, διαπιστώνεται ότι η, πρακτικά, μέθοδος τυχαίας δειγματοληψίας είναι ανεπαρκής από μόνη της στις περισσότερες περιπτώσεις, αφού δεν καταφέρνει να

βρει κάποια εφικτή και άρα αποδεκτή λύση. Ο συνδυασμός, όμως, με άλλες αποτελεσματικές μεθόδους, όπως στην περίπτωση του αλγορίθμου MATSuMoTo, δίνει τελείως διαφορετικά αποτελέσματα.

Συμπεραίνεται τελικά ότι δεν υπάρχει μία και μοναδική μέθοδος που να βελτιστοποιεί όλα τα είδη των προβλημάτων. Άλλωστε, σύμφωνα με το no free lunch theorem, εάν ένας αλγόριθμος έχει καλή απόδοση σε μια συγκεκριμένη κατηγορία προβλημάτων, τότε εμφανίζει απαραίτητα χαμηλότερη απόδοση στο σύνολο όλων των υπόλοιπων προβλημάτων.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

1. ACE – Hellas SA, ACE OCP version 2017 User's manual, General Purpose structural design optimization computing platform, 2017.
2. Ameca-Alducin, M.-Y., M. Hasani-Shoreh, W. Blaikie, F. Neumann and E. Mezura-Montes, A Comparison of Constraint Handling Techniques for Dynamic Constrained Optimization Problem, *2018 IEEE Congress on Evolutionary Computation, CEC 2018 – Proceedings*, 8477750, 2018.
3. Arouri, Y. and M. Sayyafzadeh, An accelerated gradient algorithm for well control optimization, *Journal of Petroleum Science and Engineering*, 190,106872, 2020.
4. Atamna, A., A. Auger and N. Hansen, On invariance and linear convergence of evolution strategies with augmented Lagrangian constraint handling, *Theoretical Computer Science*, 832, pp. 68-97, 2020.
5. Biedrzycki, R., Handling bound constraints in CMA-ES: An experimental study, *Swarm and Evolutionary Computation*, 52,100627, 2020.
6. Biedrzycki, R., J. Arabas and D. Jagodziński, Bound constraints handling in Differential Evolution: An experimental study, *Swarm and Evolutionary Computation*, 50,100453, 2019.
7. Biswas, P.P., P.N. Suganthan, R. Mallipeddi and G.A.J. Amaratunga, Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques, *Engineering Applications of Artificial Intelligence*, 68, pp. 81-100, 2018.
8. Caraffini, F., A.V. Kononova and D. Corne, Infeasibility and structural bias in differential evolution, *Information Sciences*, 496, pp. 161-179, 2019.
9. Chehouri, A., R. Younes, J. Perron and A. Ilinca, A Constraint-Handling Technique for Genetic Algorithms using a Violation Factor, *Journal of Computer Science*, 12(7), pp. 350-362, 2016.
10. Choi T.D., O.J. Eslinger, P.A. Gilmore, C.T. Kelley, H.A. Patrick, *User's Guide to IFFCO*, Version 2001, 51 pages, North Carolina State University, USA, 2001.
11. Chootinan, P. and A. Chen, Constraint handling in genetic algorithms using a gradient-based repair method, *Computers and Operations Research*, 33(8), pp. 2263-2281, 2006.
12. Chu, S., Z. Yang, M. Xiao, H. Qiu, K. Gao and L. Gao, Explicit topology optimization of novel polyline-based core sandwich structures using surrogate assisted evolutionary algorithm, *Computer Methods in Applied Mechanics and Engineering*, 369,113215, 2020.
13. Coello Coello, C.A., Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Computer Methods in Applied Mechanics and Engineering*, 191(11-12), pp. 1245-1287, 2002.

14. Coello Coello, C.A. and E.M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, *Advanced Engineering Informatics*, 16(3), pp. 193-203, 2002.
15. Da Silva, E.K., H.J.C. Barbosa and A.C.C. Lemonge, An adaptive constraint handling technique for differential evolution with dynamic use of variants in engineering optimization, *Optimization and Engineering*, 12(1-2), pp. 31-54, 2011.
16. Datta, R. K. Deb and J.-H. Kim, CHIP: Constraint Handling with Individual Penalty approach using a hybrid evolutionary algorithm, *Neural Computing and Applications*, 31(9), pp. 5255-5271, 2019.
17. Deb, K., An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, 186(2-4), pp. 311-338, 2000.
18. Fan, Z., Y. Fang, W. Li, Y. Yuan, Z. Wang and X. Bian, LSHADE44 with an Improved ϵ Constraint-Handling Method for Solving Constrained Single-Objective Optimization Problems, *2018 IEEE Congress on Evolutionary Computation, CEC 2018 – Proceedings*, 8477943, 2018.
19. Fang, Y., H. Xu, Q. Liu and D.T. Pham, Evolutionary optimization using epsilon method for resource-constrained multi-robotic disassembly line balancing, *Journal of Manufacturing Systems*, 56, pp. 392-413, 2020.
20. Gandomi, A.H. and K. Deb, Implicit constraints handling for efficient search of feasible solutions, *Computer Methods in Applied Mechanics and Engineering*, 363,112917, 2020.
21. Gandomi, A.H. and A.R. Kashani, Probabilistic evolutionary bound constraint handling for particle swarm optimization, *Operational Research*, 18(3), pp. 801-823, 2018.
22. Gandomi, A.H. and X.-S. Yang, Evolutionary boundary constraint handling scheme, *Neural Computing and Applications*, 21(6), pp. 1449-1462, 2012.
23. Javed, H., M.A. Jan, N. Tairan, W.K. Mashwani, R.A. Khanum, M. Sulaiman, H.U. Khan and H. Shah, On the Efficacy of Ensemble of Constraint Handling Techniques in Self-Adaptive Differential Evolution, *Mathematics*, 7(7),635, 2019.
24. Jordehi, A.R., A review on constraint handling strategies in particle swarm optimisation, *Neural Computing and Applications*, 26(6), pp. 1265-1275, 2015.
25. Juárez-Castillo, E., H.-G. Acosta-Mesa and E. Mezura-Montes, Adaptive boundary constraint-handling scheme for constrained optimization, *Soft Computing*, 23(17), pp. 8247-8280, 2019.
26. Kallioras, N.A., N.D. Lagaros and D.N. Avtzis, Pity beetle algorithm – A new metaheuristic inspired by the behavior of bark beetles, *Advances in Engineering Software*, 121, pp. 147-166, 2018
27. Kaucic, M., F. Barbini and F.J. Camerota Verdù, Polynomial goal programming and particle swarm optimization for enhanced indexation, *Soft Computing*, 24(12), pp. 8535-8551, 2020.

28. Kawachi, T., J.-I. Kushida, A. Hara and T. Takahama, Efficient Constraint Handling based on the Adaptive Penalty Method with Balancing the Objective Function Value and the Constraint Violation, *2019 IEEE 11th International Workshop on Computational Intelligence and Applications, IWCIA 2019 – Proceedings*, 8955094, pp. 121-128, 2019.
29. Kelley C.T., Software manual, *Implicit Filtering*, Version 2011, 184 pages, 2011.
30. Li, C., H. Fang and C. Gong, Expensive Inequality Constraints Handling Methods Suitable for Dynamic Surrogate-based Optimization, *2019 IEEE Congress on Evolutionary Computation, CEC 2019 – Proceedings*, 8790253, pp. 2010-2017, 2019.
31. Li, S., W. Gong, L. Wang, X. Yan and C. Hu, Optimal power flow by means of improved adaptive differential evolution, *Energy*, 198,117314, 2020.
32. Li, X., C. Xiao and Z. Lu, A New Constraint Handling Method for Economic Dispatch, *Journal of Electrical Engineering and Technology*, 13(3), pp. 1099-1109, 2018.
33. Li, Z., S. Chen, S. Zhang, S. Jiang, Y. Gu and M. Nouioua, FSB-EA: Fuzzy search bias guided constraint handling technique for evolutionary algorithm, *Expert Systems with Applications*, 119, pp. 20-35, 2019.
34. Lin, X., W. Luo, Y. Qiao, P. Xu and T. Zhu, Empirical Study of Population-Based Dynamic Constrained Multimodal Optimization Algorithms, *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, 9002835, pp. 722-730, 2019.
35. Liu, S., H. Zhu, Z. Chen and H. Cao, Frequency-constrained truss optimization using the fruit fly optimization algorithm with an adaptive vision search strategy, *Engineering Optimization*, 52(5), pp. 777-797, 2020.
36. Malan, K.M., Landscape-Aware Constraint Handling Applied to Differential Evolution, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11324 LNCS, pp. 176-187, 2018.
37. Mallipeddi, R., S. Jeyadevi, P.N. Suganthan and S. Baskar, Efficient constraint handling for optimal reactive power dispatch problems, *Swarm and Evolutionary Computation*, 5, pp. 28-36, 2012.
38. Mallipeddi, R. and P.N. Suganthan, Ensemble of Constraint Handling Technique, *IEEE Transactions on Evolutionary Computation*, 14(4),5395672, pp. 561-579, 2010.
39. Mao, J.-Q., M.-M. Tian, T.-F. Hu, K. Ji, L.-Q. Dai and H.-C. Dai, Shuffled complex evolution coupled with stochastic ranking for reservoir scheduling problems, *Water Science and Engineering*, 12(4), pp. 307-318, 2019.
40. Mazhoud, I., K. Hadj-Hamou, J. Bignon and P. Joyeux, Particle swarm optimization for solving engineering problems: A new constraint-handling mechanism, *Engineering Applications of Artificial Intelligence*, 26(4), pp. 1263-1273, 2013.

41. Mezura-Montes, E. and C.A. Coello Coello, Constraint-handling in nature-inspired numerical optimization: Past, present and future, *Swarm and Evolutionary Computation*, 1(4), pp. 173-194, 2011.
42. Miettinen, K., M.M. Mäkelä and J. Toivanen, Numerical Comparison of Some Penalty-Based Constraint Handling Techniques in Genetic Algorithms, *Journal of Global Optimization*, 27(4), pp. 427-446, 2003.
43. Miranda-Varela, M.-E. and E. Mezura-Montes, Constraint-handling techniques in surrogate-assisted evolutionary optimization. An empirical study, *Applied Soft Computing Journal*, 73, pp. 215-229, 2018.
44. Montemurro, M., A. Vincenti and P. Vannucci, The Automatic Dynamic Penalization method (ADP) for handling constraints with genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, 256, pp. 70-87, 2013.
45. Mozaffari, A., M. Emami and A. Fathi, A comprehensive investigation into the performance, robustness, scalability and convergence of chaos-enhanced evolutionary algorithms with boundary constraints, *Artificial Intelligence Review*, 52(4), pp. 2319-2380, 2019.
46. Muller J., MATSuMoTo: The MATLAB Surrogate Model Toolbox For Computationally Expensive Black-Box Global Optimization Problems, *ArXiv e-prints*, 2014a.
47. Muller J., *MATSuMoTo Code Documentation*, 15 pages, Cornell University, Ithaca, NY, USA, 2014b.
48. Peng, C., H.-L. Liu and F. Gu, A novel constraint-handling technique based on dynamic weights for constrained optimization problems, *Soft Computing*, 22(12), pp. 3919-3935, 2018.
49. Qian, J., J. Yi, Y. Cheng, J. Liu and Q. Zhou, A sequential constraints updating approach for Kriging surrogate model-assisted engineering optimization design problem, *Engineering with Computers*, 36(3), pp. 993-1009, 2020.
50. Qiao, B. and J. Liu, Multi-objective dynamic economic emission dispatch based on electric vehicles and wind power integrated system using differential evolution algorithm, *Renewable Energy*, 154, pp. 316-336, 2020.
51. Rodrigues, L.L., J. Sebastian Solis-Chaves, O.A.C. Vilcanqui and A.J.S. Filho, Predictive Incremental Vector Control for DFIG With Weighted-Dynamic Objective Constraint-Handling Method-PSO Weighting Matrices Design, *IEEE Access*, 8, 9121987, pp. 114112-114122, 2020.
52. Rodrigues, M.D.C., S. Guimarães and B.S.L.P. de Lima, E-BRM: A constraint handling technique to solve optimization problems with evolutionary algorithms, *Applied Soft Computing Journal*, 72, pp. 14-29, 2018.
53. Runarsson, T.P. and X. Yao, Stochastic ranking for constrained evolutionary optimization, *IEEE Transactions on Evolutionary Computation*, 4(3), pp. 284-294, 2000.

54. Salcedo-Sanz, S., A survey of repair methods used as constraint handling techniques in evolutionary algorithms, *Computer Science Review*, 3(3), pp. 175-192, 2009.
55. Shabani, A., B. Asgarian, M. Salido and S. Asil Gharebaghi, Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems, *Expert Systems with Applications*, 161,113698, 2020.
56. Stanovov, V., S. Akhmedova and E. Semenkin, Combined fitness–violation epsilon constraint handling for differential evolution, *Soft Computing*, 24(10), pp. 7063-7079, 2020.
57. Stein, M., Large Sample Properties of Simulations Using Latin Hypercube Sampling, *Technometrics*, 29(2), pp. 143-151, 1987.
58. Storn, R. and K. Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, 11(4), pp. 341-359, 1997.
59. Takahama, T. and S. Sakai, Constrained optimization by ε constrained particle swarm optimizer with ε -level control, *Advances in Soft Computing*, (AISC), pp. 1019-1029, 2005.
60. Takahama, T. and S. Sakai, Constrained Optimization by the ε Constrained Differential Evolution with an Archive and Gradient-Based Mutation, *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, 5586484, 2010.
61. Trivedi, A., N. Biswas, S. Chakroborty and D. Srinivasan, Extending Unified Differential Evolution with a New Ensemble of Constraint Handling Techniques, *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017- Proceedings*, 2018-January, pp. 1-8, 2018.
62. Wang, S., J. Kang, M.F. Tasgetiren, L. Gao and D. Kizilay, Differential Evolution Through Variable Neighborhood Search for Constrained Real Parameter Optimization Problems, *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, 9003026, pp. 2252-2260, 2019.
63. Wang, Y., Z. Cai, Y. Zhou and Z. Fan, Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique, *Structural and Multidisciplinary Optimization*, 37(4), pp. 395-413, 2009.
64. Yang S.H., U. Natarajan, M. Sekar and S. Palani, Prediction of surface roughness in turning operations by computer vision using neural network trained by differential evolution algorithm, *International Journal of Advanced Manufacturing Technology*, 51(9-12), pp. 965-971, 2010.
65. Zein, S., V. Madhavan and D. Dumas, A repair operator for the preliminary design of a composite structure using genetic algorithms, *Structural and Multidisciplinary Optimization*, 55(6), pp. 2099-2110, 2017.
66. Zhang, C., Q. Lin, L. Gao and X. Li, Backtracking Search Algorithm with three constraint handling methods for constrained optimization problems, *Expert Systems with Applications*, 42(21), pp. 7831-7845, 2015.

67. Zhao, X., K. Zhang, G. Chen, X. Xue, C. Yao, J. Wang, Y. Yang, H. Zhao and J. Yao, Surrogate-assisted differential evolution for production optimization with nonlinear state constraints, *Journal of Petroleum Science and Engineering*, 194,107441, 2020.
68. Βλάχος Α., Μετα-ευρεστικοί αλγόριθμοι βελτιστοποίησης και εφαρμογής σε προβλήματα συνδυαστικής βελτιστοποίησης, Διδακτορική Διατριβή, Πανεπιστήμιο Πειραιώς, Πειραιάς, 2005.
69. Γκουντή Α., Εξελικτικοί Αλγόριθμοι: Ο αλγόριθμος της διαφορικής εξέλιξης, Μεταπτυχιακή Διπλωματική Εργασία, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη, 2013.
70. Μιχάλογλου Α., Αλγόριθμοι Εξελικτικής Βελτιστοποίησης και Εφαρμογές, Πτυχιακή Εργασία, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θεσσαλονίκη, 2018.
71. Πουλή Ι., Αλγόριθμοι για τον βέλτιστο σχεδιασμό δικτυωμάτων μεγάλου μεγέθους σε συνδυασμό με τη μέθοδο των πεπερασμένων στοιχείων, Μεταπτυχιακή Εργασία, Πολυτεχνείο Κρήτης, Χανιά, 2014.