

ΚΑΜΙΝΑΚΗΣ ΝΙΚΟΛΑΟΣ

ΣΥΝΘΕΣΗ ΠΟΛΥΜΟΡΦΙΚΩΝ ΕΥΚΑΜΠΤΩΝ
ΜΗΧΑΝΙΣΜΩΝ ΜΕ ΧΡΗΣΗ ΤΟΠΟΛΟΓΙΚΗΣ
ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΙ ΕΞΕΛΙΚΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΦΕΒΡΟΥΑΡΙΟΣ 2008



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΠΑΡΑΓΩΓΗΣ

ΣΥΝΘΕΣΗ ΕΥΚΑΜΠΤΩΝ ΜΗΧΑΝΙΣΜΩΝ ΠΟΛΛΑΠΛΩΝ ΕΦΑΡΜΟΓΩΝ ΜΕ ΧΡΗΣΗ ΤΟΠΟΛΟΓΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΚΑΙ ΕΞΕΛΙΚΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

υπό
Καμινάκη Νικόλαου

*Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων για την
απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης*

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΣΤΡΟΦΗ

ΚΑΘ. ΣΤΑΥΡΟΥΛΑΚΗΣ Ε. ΓΕΩΡΓΙΟΣ (Επιβλέπων)	
ΕΠΙΚ. ΚΑΘ. ΓΕΩΡΓΙΑΚΗΣ ΓΕΩΡΓΙΟΣ	
ΚΑΘ. ΜΠΙΛΑΛΗΣ ΝΙΚΟΛΑΟΣ	

Στους γονείς μου

ΠΕΡΙΛΗΨΗ

Η μέθοδος της τοπολογικής βελτιστοποίησης επιλύει το βασικό πρόβλημα του βέλτιστου σχεδιασμού στην μηχανική, της διανομής μια περιορισμένης ποσότητας υλικού σε συγκεκριμένο, δεδομένο χώρο με στόχο τον σχεδιασμό ενός δομικού φορέα ή μηχανισμού ο οποίος εκπληρώνει τους στόχους σχεδιασμού κατά τον καλύτερο τρόπο. Η μέθοδος αυτή βρίσκει εφαρμογή στον βέλτιστο σχεδιασμό κατασκευών, μικρο-δομών, μηχανισμών, δομών βέλτιστης ροής, κ.α. Μια από τις πιο ενδιαφέρουσες εφαρμογές είναι ο σχεδιασμός των εύκαμπτων μηχανισμών. Ένας εύκαμπτος μηχανισμός είναι μια κατασκευή με τα εξής χαρακτηριστικά: *i)* η εξωτερική φόρτιση που επιβάλλεται στην κατασκευή μετατρέπεται σε κίνηση προς μια επιθυμητή κατεύθυνση σε ορισμένα σημεία της (διαφορετικά από εκείνα στα οποία επιβάλλεται η φόρτιση), *ii)* είναι μια μονολιθική κατασκευή, αποτελείται δηλαδή από μία και μόνο δομή χωρίς αρθρώσεις ή άλλο μηχανισμό, *iii)* είναι μια αρκετά εύκαμπτη κατασκευή ώστε να μπορεί να μεταδώσει συγκεκριμένη κίνηση, *iv)* είναι επιπλέον μια αρκετά δύσκαμπτη κατασκευή για να αντέχει την επιβαλλόμενη εξωτερική φόρτιση.

Οι εύκαμπτοι μηχανισμοί πολλαπλών εφαρμογών, είναι κατασκευές που παρουσιάζουν όλα τα παραπάνω χαρακτηριστικά και μπορούν να μετατρέψουν δύο ή και περισσότερες περιπτώσεις επιβαλλόμενων φορτίσεων σε δύο ή και περισσότερες κινήσεις, μια κίνηση ανά περίπτωση φόρτισης. Οι μηχανισμοί αυτοί που έχουν πολλές διαφορετικές λειτουργίες ονομάζονται εδώ "πολυμορφικοί μηχανισμοί".

Κατά την επίλυση προβλημάτων τοπολογικής βελτιστοποίησης εμφανίζονται προβλήματα βελτιστοποίησης με μεγάλο αριθμό αγνώστων. Ειδικότερα για την εύρεση της κατάλληλης κατασκευής που μπορεί να εξυπηρετεί πολλαπλές λειτουργίες, προκύπτει η ανάγκη της επίλυσης πολυκριτήριων προβλημάτων βελτιστοποίησης με μεγάλο αριθμό αγνώστων. Η χρήση τεχνικών επαναληπτικής τοπικής αναζήτησης πιθανόν να μην είναι αποτελεσματικές εξαιτίας της παρουσίας πολλών τοπικών βελτίστων, ακόμα και για την περίπτωση μιας περίπτωση φόρτισης. Στην παρούσα διατριβή παρουσιάζεται ένας υβριδικός αλγόριθμος που συνδυάζει τα πλεονεκτήματα τεχνικών γενικής βελτιστοποίησης, όπως εξελικτικοί αλγόριθμοι και συγκεκριμένα ο ο Διαφορικός Εξελικτικός Αλγόριθμος—*Differential Evolution* και αλγόριθμοι νοημοσύνης σμήνους σωματιδίων—*Particles Swarm Optimization*, και χρησιμοποιεί ως εργαλείο αξιολόγησης επαναληπτικούς αλγόριθμους τοπικής αναζήτησης. Τα αποτελέσματα της παρούσας διατριβής μπορούν να χρησιμοποιηθούν για την σύνθεση και σχεδιασμό μικρο-μηχανισμών και να επεκταθούν για την επίλυση προβλημάτων πολλών συνδυασμένων πεδίων (multi-physics).

ABSTRACT

Topology optimization solves the basic engineering design problem of distributing a limited amount of material in a design space in order to form a structure or a mechanism that works in an optimal way. The method is applicable to optimal design of structures, micro-structures, compliant mechanisms, optimal flow structures. One of the most interesting applications is the synthesis of compliant mechanisms. A compliant mechanism is a structure with the following features: *i)* the external load is transformed to motion along specific direction at a given point of the structure (different to the ones where the loading is applied), *ii)* it is a monolithic, single-piece structure without any joints and hinges, *iii)* the structure is flexible enough to deliver motion, *iv)* the structure is stiff enough to bear the applied load.

Multi-purpose compliant mechanisms are structures with all the above features that are able to deliver two or more different motions depending on the applied load cases, one delivered motion per applied load.

Several multi-objective topology optimization problems are first formulated in this paper. Due to local minima that arise *iterative local search* methods may not be effective. Therefore, a hybrid algorithm utilizing the advantages of global optimization techniques such as *Differential Evolution* and *Particle Swarms Optimization*, is proposed. The hybrid scheme uses iterative local search methods as the evaluation tool of the previously mentioned global optimization algorithms.

ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος σχημάτων	vi
Κατάλογος πινάκων	x
I	1
1 Εισαγωγή στην Τοπολογική Βελτιστοποίηση Κατασκευών	2
1.1 Εισαγωγή	2
1.2 Μόρφωση προβλήματος και παραμετροποίηση σχεδιασμού	5
1.2.1 Σχεδιασμός κατασκευής με ελαχιστοποίηση της ευκαμψίας	5
1.2.2 Μόρφωση διακριτού προβλήματος	6
1.2.3 Παραμετροποίηση σχεδιασμού	6
1.2.4 Συνθήκες βελτιστοποίησης και ανάλυση ευαισθησίας	8
1.3 Περιγραφή του αλγορίθμου	10
1.3.1 Υπολογιστική διαδικασία	10
1.3.2 Ορισμός του αρχικού χωρίου σχεδιασμού	10
1.3.3 Βελτιστοποίηση	11
1.3.4 Παρουσίαση των αποτελεσμάτων-Λήξη αλγορίθμου	12
1.3.5 Ανάλυση αλγορίθμου	12
1.4 Αριθμητικά προβλήματα	12
1.4.1 Εμφάνιση αριθμητικών προβλημάτων	12
1.4.2 Πρόβλημα τύπου σκακιέρας	13
1.4.3 Πρόβλημα εξάρτησης πλέγματος	13
1.5 Τοπολογική Βελτιστοποίηση με περισσότερες περιπτώσεις φορτίσεων	14
1.5.1 Εισαγωγή στο πρόβλημα πολλαπλών φορτίσεων	14
1.5.2 Μόρφωση προβλήματος	15
1.6 Υλοποίηση Τοπολογικής Βελτιστοποίησης σε Matlab	17
1.6.1 Γενική περιγραφή	17
1.6.2 Εκτέλεση του λογισμικού	17
1.6.3 Κύριο πρόγραμμα (γραμμές 1 έως 36)	17
1.6.4 Κριτήρια Βέλτιστου (γραμμές 37-48)	18

1.6.5	Εφαρμογή φίλτρου για το πρόβλημα εξάρτησης πλέγματος (γραμμές 49-64)	18
1.6.6	Ανάλυση πεπερασμένων στοιχείων (γραμμές 65-99)	19
1.6.7	Πολλές περιπτώσεις φορτίσεων	19
2	Τοπολογική Βελτιστοποίηση Εύκαμπτων Μηχανισμών	21
2.1	Σύνθεση εύκαμπτων μηχανισμών	21
2.1.1	Εισαγωγή στους εύκαμπτους μηχανισμούς	21
2.1.2	Μόρφωση προβλήματος	22
2.1.3	Υπολογιστική διαδικασία και ανάλυση ευαισθησίας	23
2.1.4	Σύνθεση πολυμορφικών εύκαμπτων μηχανισμών	24
2.2	Τοπολογική βελτιστοποίηση για σύνθεση μηχανισμών στο MATLAB	25
2.3	Υλοποίηση στο MATLAB: 2 περιπτώσεις φορτίσεων	27
2.3.1	Αντικειμενική συνάρτηση	27
2.3.2	Αριθμητικά προβλήματα	28
II	Στοχαστικοί Αλγόριθμοι Ολικής Βελτιστοποίησης	30
3	Εξελικτικοί αλγόριθμοι	31
3.1	Εισαγωγή	31
3.2	Εξελικτικοί Αλγόριθμοι	31
3.2.1	Ιστορική αναδρομή	31
3.2.2	Χαρακτηριστικά Εξελικτικών αλγορίθμων	33
3.2.3	Περιγραφή ενός εξελικτικού αλγορίθμου	33
3.3	Διαφορικοί Εξελικτικοί Αλγόριθμοι - DEA	34
3.3.1	Εισαγωγή στους DEA	34
3.3.2	Περιγραφή DEA	34
3.3.3	Η δομή των πληθυσμών στον DEA	35
3.3.4	Αρχικοποίηση πληθυσμού	38
3.3.5	Η μετάλλαξη στον DEA	38
3.3.6	Ο τελεστής του επιχιασμού στον DEA	39
3.3.7	Ο τελεστής της επιλογής στον DEA	40
4	Βελτιστοποίηση με Χρήση Νοημοσύνης Σμήνους Σωματιδίων (PSO)	41
4.1	Εισαγωγή	41
4.2	Περιγραφή PSO	41
4.2.1	Τι είναι ο αλγόριθμος PSO	41
4.2.2	Σύγκριση με εξελικτικούς αλγορίθμους	42
4.3	Ο αλγόριθμος PSO	42
4.3.1	Μοντελοποίηση	42
4.3.2	Γραφική αναπαράσταση PSO	43
4.3.3	Παρατηρήσεις	47
III	Επίλυση και Αποτελέσματα	48
5	Μοντελοποίηση Προβλήματος Ολικής Βελτιστοποίησης Για Σύνθεση Εύκαμπτων Μηχανισμών	49
5.1	Παρουσία τοπικών Βέλτιστων	49

5.2	Υβριδικό σχήμα Ολικής Βελτιστοποίησης	50
5.2.1	Η ιδέα	50
5.2.2	Περιγραφή αλγορίθμου	50
5.2.3	Μία περίπτωση φόρτισης: αντικειμενική συνάρτηση	52
5.2.4	Έλεγχος μετακίνησης	52
5.3	Σύνθεση πολυμορφικού εύκαμπτου μηχανισμού	53
5.3.1	Μόρφωση Προβλήματος	53
5.3.2	Περιορισμοί ελέγχου μετακίνησης	54
6	Αποτελέσματα	56
6.1	Εισαγωγή	56
6.2	Σύνθεση μηχανισμού για μια περίπτωση φόρτισης	56
6.2.1	Περιγραφή προβλήματος	56
6.2.2	Παράμετροι προβλήματος για DEA και αριθμητικά στοιχεία	56
6.2.3	Αποτελέσματα για μια περίπτωση φόρτισης και DEA	57
6.2.4	Παράμετροι προβλήματος για αλγόριθμο βελτιστοποίησης με χρήση νοημοσύνης σμήνους σωματιδίων και αριθμητικά στοιχεία - PSO	65
6.2.5	Αποτελέσματα για μια περίπτωση φόρτισης και PSO	65
6.3	Σύγκριση αποτελεσμάτων DEA και PSO για μια περίπτωση φόρτισης	70
7	Αποτελέσματα σύνθεσης πολυμορφικών μηχανισμών	71
7.1	Εισαγωγή	71
7.2	Σύνθεση μηχανισμού για μια περίπτωση φόρτισης	71
7.2.1	Περιγραφή προβλήματος	71
7.2.2	Παράμετροι προβλήματος για DEA και αριθμητικά στοιχεία	71
7.2.3	Αποτελέσματα για μια περίπτωση φόρτισης και DEA	72
7.2.4	Παράμετροι προβλήματος για αλγόριθμο βελτιστοποίησης με χρήση νοημοσύνης σμήνους σωματιδίων και αριθμητικά στοιχεία - PSO	79
7.2.5	Αποτελέσματα για δύο περιπτώσεις φόρτισης και PSO	80
7.3	Σύγκριση αποτελεσμάτων DEA και PSO για δύο περιπτώσεις φόρτισης	89
8	Συμπεράσματα και περαιτέρω έρευνα	90
8.1	Σύνοψη της εργασίας	90
8.2	Συμπεράσματα	90
8.3	Περαιτέρω έρευνα	90
IV	Παραρτήματα	92
A'	Παράρτημα A	93
A'.1	Κώδικας για Τοπολογική Βελτιστοποίηση Κατασκευών, top.m	93
A'.2	Τοπολογική Βελτιστοποίηση σύνθεσης εύκαμπτων μηχανισμών, μια περίπτωση φόρτισης	96
A'.3	Τοπολογική Βελτιστοποίηση σύνθεσης εύκαμπτων μηχανισμών, Δύο περιπτώσεις φορτίσεων	99
	Βιβλιογραφία	103

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

1.1	Κατασκευή Michell. α) Το χωρίο σχεδιασμού, β) Κατανομή υλικού με τοπολογική βελτιστοποίηση, γ) Ιδανική κατασκευή βασισμένη σε αναλυτικές μεθόδους.	3
1.2	Τρισδιάστατο χωρίο σχεδιασμού, δίκτυο κόμβων και ράβδων.	3
1.3	Τρεις κατηγορίες βελτιστοποίησης δομής.	4
1.4	Ορισμός αρχικού χωρίου σχεδιασμού τοπολογικής βελτιστοποίησης.	5
1.5	Διάγραμμα ροής Αλγόριθμου Τοπολογικής Βελτιστοποίησης [2]	11
1.6	Φαινόμενα τύπου σκακιέρας.	13
1.7	Φαινόμενα εξάρτησης πλέγματος.	14
1.8	Γραφική αναπαράσταση φίλτρου για το πρόβλημα εξάρτησης πλέγματος (γραμμική).	15
1.9	Τοπολογική βελτιστοποίηση κατασκευής: Επάνω παρουσιάζεται το αρχικό χωρίο σχεδιασμού, στην μέση παρουσιάζεται η απλοποίηση του προβλήματος με την χρήση συμμετρίας και οι συνοριακές συνθήκες, και κάτω παρουσιάζεται η τελική κατανομή [11].	18
1.10	Τοπολογική βελτιστοποίηση δοκού για δύο περιπτώσεις φορτίσεων: Αριστερά το χωρίο ορισμού με αντίθετες φορτίσεις στις πάνω και κάτω δεξιά κορυφές, στη μέση η τελική κατανομή όπως προκύπτει αν οι φορτίσεις εφαρμόζονται ταυτόχρονα και δεξιά η τελική κατανομή όπως προκύπτει στην περίπτωση των δύο διαφορετικών φορτίσεων [11].	20
2.1	Σύνθεση εύκαμπτων μηχανισμών: α) Ορισμός αρχικού πεδίου σχεδιασμού, β) Τελική κατανομή υλικού.	23
2.2	Τοπολογική βελτιστοποίηση για αντιστροφέα δύναμης. Αριστερά παρουσιάζεται το μισό χωρίο ορισμού, με τους περιορισμούς συμμετρίας, τις στηρίξεις και φορτίσεις ενώ δεξιά παρουσιάζεται η τελική κατανομή όπως προκύπτει από την εκτέλεση του <code>topm.m</code> : <code>topm(40,20,0.3,3.0,1.2)</code> .	27
2.3	Τοπολογική βελτιστοποίηση για πολυμορφικό αντιστροφέα δύναμης. Αριστερά η γραφική μόρφωση, και δεξιά μια τυχαία τελική κατανομή. Εκτέλεση σε περιβάλλον MATLAB: <code>topm2loads(30,30,0.3,3.0,1.2)</code> .	28
2.4	Πάνω αριστερά, τελική κατανομή για την μια περίπτωση φόρτισης, πάνω δεξιά τελική κατανομή για την δεύτερη περίπτωση φόρτισης, κάτω αριστερά και δεξιά τελικές κατανομές που ικανοποιούν και τις περιπτώσεις φορτίσεων.	29

3.1	Αρχικοποίηση πληθυσμού στον DEA, [8]	35
3.2	Υπολογισμός διαταράξεων στον πληθυσμό, [8]	35
3.3	Μετάλλαξη, [8]	36
3.4	Επιλογή. Το διάνυσμα u_0 αντικαθιστά το διάνυσμα με δείκτη 0 γιατί έχει καλύτερη τιμή αντικειμενικής συνάρτησης, [8].	36
3.5	Μετάλλαξη ενός άλλου διανύσματος του τρέχοντα πληθυσμού, [8].	37
3.6	Συγκρίση μεταξύ u_1 και x_1 . Στην περίπτωση το νέο διάνυσμα είναι χειρότερο από το υπάρχον, [8].	37
3.7	Ο τελεστής της διαφορικής μετάλλαξης: η κλιμακοποιημένη διαφορά $F \cdot (x_{r1,g} - x_{r2,g})$ προστίθεται στο βασικό διάνυσμα $x_{r0,g}$ για την παραγωγή του μεταλλαγμένου διανύσματος $v_{i,g}$, [8].	39
3.8	Τα πιθανά επιπλέον υπό δοκιμή διανύσματα $x'_{i,g}, x''_{i,g}$, όταν τα διανύσματα $v_{i,g}$ και $x_{i,g}$ διασταυρώνονται ομοιόμορφα, [8].	40
4.1	Αρχικό διάνυσμα $x_i^{(k)}$ μαζί με ταχύτητα $v_i^{(k)}$, το προσωπικό και ολικό βέλτιστο, $\hat{x}_i$ και $\hat{g}$	44
4.2	Δημιουργία των διανυσματικών διαφορών, $(\hat{g} - x_i)$ και $(\hat{x}_i - x_i)$ καθώς και η κλιμακοποίηση της ταχύτητας $\omega^k v_i$	44
4.3	Δημιουργία τυχαίων διανυσμάτων $c_1 r_1 \circ (\hat{x}_i - x_i)$ και $c_2 r_2 \circ (\hat{g} - x_i)$, συναρτήσει των παραπάνω διανυσματικών διαφορών.	45
4.4	Δημιουργία του $x_i^{(k+1)}$ (βήμα 1).	45
4.5	Δημιουργία του $x_i^{(k+1)}$ (βήμα 2).	46
4.6	Το διάνυσμα $x_i^{(k+1)}$ μαζί με την ταχύτητα του $v_i^{(k+1)}$	46
5.1	Ασύμμετρος μηχανισμός αντιστροφής δύναμης.	49
5.2	Διαφορετικές τελικές κατανομές υλικού με διαφορετικά σημεία εκκίνησης.	51
5.3	Λογικό διάγραμμα υβριδικού αλγορίθμου	52
5.4	Μεγιστοποίηση $u_{out}^{@X}$ και ελαχιστοποίηση $u_{out}^{@Y}$	53
5.5	Μόρφωση προβλήματος σύνθεσης μηχανισμών για δύο περιπτώσεις φορτίσεων.	54
6.1	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης	57
6.2	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή.	58
6.3	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή (με φίλτρο).	59
6.4	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Μετακινήσεις (με φίλτρο).	59
6.5	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή και Μετακινήσεις (με φίλτρο).	60
6.6	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο).	60
6.7	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Εξέλιξη αλγορίθμου.	61
6.8	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 1 έως 2	61
6.9	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 3 έως 10.	62
6.10	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 11 έως 19.	62

6.1.1	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 20 έως 23.	63
6.1.2	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 24 έως 26.	63
6.1.3	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 27 έως 45.	64
6.1.4	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 46 έως 50.	64
6.1.5	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή.	66
6.1.6	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή (με φίλτρο).	66
6.1.7	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Μετακινήσεις (με φίλτρο).	67
6.1.8	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή και Μετακινήσεις (με φίλτρο).	67
6.1.9	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο).	68
6.2.0	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Εξέλιξη αλγορίθμου.	68
6.2.1	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 1 έως 7.	69
6.2.2	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 8 έως 12.	69
6.2.3	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 13 έως 50.	70
7.1	Μόρφωση προβλήματος σύνθεσης μηχανισμών για δύο περιπτώσεις φορτίσεων	72
7.2	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, Τελική κατανομή	73
7.3	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, Τελική κατανομή (με φίλτρο).	74
7.4	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1 ^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)	74
7.5	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)	75
7.6	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)	75
7.7	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 2 ^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)	76
7.8	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 2 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)	76
7.9	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)	77
7.10	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, Εξέλιξη αλγορίθμου	77
7.11	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 1 έως 1	78

7.12	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 2 έως 20	78
7.13	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 21 έως 50	79
7.14	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, Τελική κατανομή	80
7.15	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, Τελική κατανομή (με φίλτρο)	81
7.16	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 1 ^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)	81
7.17	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 1 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)	82
7.18	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 1 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)	82
7.19	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 2 ^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)	83
7.20	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 2 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)	83
7.21	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1 ^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)	84
7.22	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, Εξέλιξη αλγορίθμου	84
7.23	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για την πρώτη επανάληψη	85
7.24	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για την δεύτερη επανάληψη	85
7.25	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 3 έως 6	86
7.26	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 7 έως 12	86
7.27	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 13 έως 17	87
7.28	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 18 έως 26	87
7.29	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 27 έως 38	88
7.30	Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 39 έως 50	88

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

6.1	Παράμετροι προβλήματος για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DEA	57
6.2	Αποτελέσματα για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DEA.	58
6.3	Παράμετροι προβλήματος για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO.	65
6.4	Αποτελέσματα για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO.	65
6.5	Σύγκριση αποτελεσμάτων DEA και PSO για μία περίπτωση φόρτισης	70
7.1	Παράμετροι προβλήματος για 2 περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DE	72
7.2	Αποτελέσματα για δύο περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DEA	73
7.3	Παράμετροι προβλήματος για 2 περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO	79
7.4	Αποτελέσματα για δύο περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO	80
7.5	Σύγκριση αποτελεσμάτων DEA και PSO για δύο περιπτώσεις φόρτισης . . .	89
7.6	Σύγκριση χρόνων αποπεράτωσης για 50x50 διακριτοποίηση για 2 περιπτώσεις φόρτισης	89

ΜΕΡΟΣ Ι

ΠΕΡΙ ΤΟΠΟΛΟΓΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

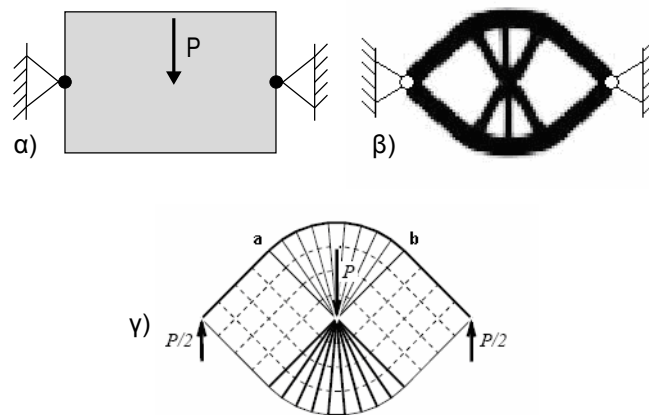
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΤΟΠΟΛΟΓΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΚΑΤΑΣΚΕΥΩΝ

1.1 Εισαγωγή

Η τοπολογική βελτιστοποίηση είναι η μέθοδος που χαρακτηρίζει την μόρφωση προβλημάτων βελτιστοποίησης και σχεδιασμού που επιτρέπουν τον καθορισμό της τοπολογίας μια κατασκευής ή ενός μηχανικού συστήματος. Το αποτέλεσμα της παραπάνω διαδικασίας είναι η εύρεση της τοπολογίας της κατασκευής, δηλαδή το σχήμα, το μέγεθος/διαστάσεις, και τη θέση των δομικών της, στοιχείων. Η τοπολογική βελτιστοποίηση είναι στην ουσία μια μέθοδος βέλτιστης διανομής υλικού σε συγκεκριμένο χωρίο και έχει την δυνατότητα να μπορεί να διαχειριστεί τα τρία προβλήματα, σχήματος, μεγέθους, και θέσης, ταυτόχρονα. Γενικότερα η μέθοδος γίνεται αντιληπτή εάν συγκριθεί με την πλειονότητα των κλασσικών προβλημάτων βέλτιστου σχεδιασμού, τα οποία για δεδομένη μορφή-τοπολογία περιορίζονται στον βέλτιστο υπολογισμό μερικών μόνο παραμέτρων σχεδιασμού.

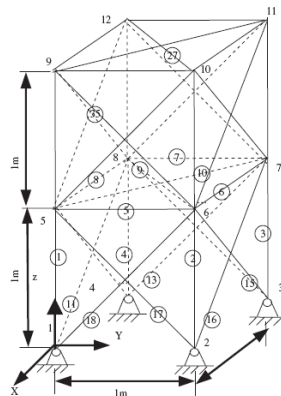
Η βασική ιδέα για λύση προβλημάτων κατανομής υλικού προήλθε από τον Michell [9] στις αρχές το 1904. Ο Michell επικεντρώθηκε στον σχεδιασμό κατασκευών από λεπτές ράβδους θέτοντας ως όριο σχεδιασμού την πλαστική μηχανική απόκριση του υλικού. Η μελέτη αυτή είχε βασιστεί σε αναλυτικές μεθόδους (βλ. σχήμα 1.1). Αρκετά αργότερα, οι μέθοδοι μαθηματικού προγραμματισμού, και πιο συγκεκριμένα, τεχνικές γραμμικού προγραμματισμού και η μέθοδος *simplex* εφαρμόστηκαν σε προβλήματα ελαχιστοποίησης του βάρους κατασκευών με δοκούς, με περιορισμούς τάσεων. Χρησιμοποιώντας ως μεταβλητή σχεδιασμού το εμβαδόν της διατομή της δοκού, το πρόβλημα εύρεσης της βέλτιστης θέσης της όταν η κατασκευή υπόκειται σε ένα και μόνο φορτίο, μπορεί να επιλυθεί χρησιμοποιώντας γραμμικό προγραμματισμό. Η τοπολογία καθορίζει ποιοι από τους κόμβοι είναι συνδεδεμένοι μεταξύ τους με δοκούς, από ένα ευρύτερο χωρίο σχεδιασμού που αποτελείται από ένα δίκτυο κόμβων και πιθανών ράβδων (βλ. σχήμα 1.2). Με την χρήση ενός αρκετά πυκνού αρχικού δικτύου δοκών (*structural aniverse*) τα προκύπτοντα βέλτιστα δικτυώματα κατά Michell προσδιορίζονται τελικώς με την χρήση κλασσικών τεχνικών βέλτιστου σχεδιασμού. Οι ράβδοι δοκοί που δεν χρειάζονται στην τελική θέση αφαιρούνται και τελικώς προκύπτει η ζητούμενη τοπολογία [3].

Η ιδέα διακριτοποίησης του αρχικού χωρίου σχεδιασμού σε κόμβους και ράβδους χρησιμοποιήθηκε στην περίπτωση όπου το χωρίο σχεδιασμού καλύπτεται από ένα συνεχές φυσικό μέσο. Μερικές από τις βασικές ιδέες αποσαφηνίστηκαν από θεωρητικές εργασίες που σχετίζονται με την ύπαρξη λύσης και την εφαρμογή τεχνικών ομογενοποίησης για τον καλύτερο ορισμό των προβλημάτων. Οι συγκεκριμένες εργασίες με την σειρά τους έγιναν η βάση για



Σχήμα 1.1: Κατασκευή Michell. α) Το χωρίο σχεδιασμού, β) Κατανομή υλικού με τοπολογική βελτιστοποίηση, γ) Ιδανική κατασκευή βασισμένη σε αναλυτικές μεθόδους.

την χρήση υπολογιστικών μεθόδων, τεχνικών διανομής υλικού, οι οποίες δουλεύουν με παραμετροποίηση σχεδιασμού και επιτρέπουν τον υπολογισμό της βέλτιστης κατανομής υλικού σε ένα συγκεκριμένο και εξαρχής ορισμένο χωρίο σχεδιασμού. Ενώ στις πρώτες εργασίες χρησιμοποιήθηκαν επαναληπτικές μεθόδους που βασίζονται στην χρήση μεθόδων κριτηρίων βελτίστου (optimality criteria methods) (M.P Bendsøe & Kikuchi, [1]) σήμερα οι περισσότερες μέθοδοι διανομής υλικού βασίζονται σε τεχνικές μαθηματικού προγραμματισμού μαζί με χρήση τεχνικών ανάλυσης με πεπερασμένα στοιχεία. Αυτό σημαίνει ότι οι θεμελιώδεις τεχνικές επίλυσης των μεθόδων διανομής υλικού μοιάζουν αρκετά με τις μεθόδους που αναπτύχθηκαν για διαστασιολογική βελτιστοποίηση, αλλά με ένα ευρύ φάσμα πολυπλοκότητας που σχετίζεται με την ειδική παραμετροποίηση που θα πρέπει να γίνει για τον σχεδιασμό τοπολογίας. Ιδιαίτερη προσοχή θα πρέπει να δίνεται στον ορισμό προβλημάτων τοπολογικής βελτιστοποίησης ειδικά όταν πρόκειται για προβλήματα μεγάλης κλίμακας [3].



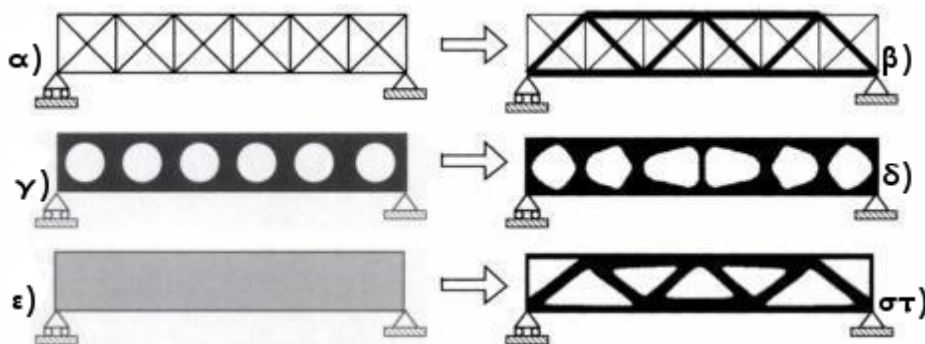
Σχήμα 1.2: Τρισδιάστατο χωρίο σχεδιασμού, δίκτυο κόμβων και ράβδων.

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα βασικά συστατικά της μεθόδου που θα αποκαλούμε μέθοδος διανομής υλικού για την εύρεση της βέλτιστης απόκρισης μιας γραμμικά ελαστικής δομής. Υπό το παρόν πρίσμα η χωροταξία της κατασκευής συμπεριλαμβάνει την πληροφορία για την τοπολογία, το σχήμα και την διαστασιολόγηση της κατασκευής, ενώ η μέθοδος διανομής υλικού μπορεί να διαχειριστεί και τα τρία προβλήματα ταυτόχρονα.

Τα προβλήματα βελτιστοποίησης της τοπολογίας, του σχήματος και του μεγέθους μιας κατασκευής ή μηχανικού συστήματος είναι διάφορες μορφές του γενικότερου προβλήματος εύρεσης της βέλτιστης δομής αυτών. Σε ένα τυπικό πρόβλημα βελτιστοποίησης *μεγέθους* ο στόχος θα μπορούσε να είναι η εύρεση της βέλτιστης κατανομής του πάχους μιας γραμμικά ελαστικής δοκού ή το πάχος των δοκών σε ένα δικτύωμα. Η βέλτιστη κατανομή πάχους ελαχιστοποιεί (ή μεγιστοποιεί) κάποια φυσική ιδιότητα όπως η μέση ευκαμψία (εξωτερικό έργο), μέγιστες τάσεις, μέγιστες μετακινήσεις κ.τ.λ, ενώ ταυτόχρονα ικανοποιούνται οι συνθήκες ισορροπίας, τα κριτήρια καταστάσεως καθώς και οι περιορισμοί για τις μεταβλητές σχεδιασμού. Ως μεταβλητή σχεδιασμού μπορεί να θεωρηθεί το πάχος της ελαστικής πλάκας ή το πάχος των δοκών, ενώ ως κριτήριο καταστάσεως μπορεί να θεωρηθούν οι τάσεις και οι μετακινήσεις που θα αναπτυχθούν όταν η συγκεκριμένη κατασκευή υπόκειται σε φορτίσεις. Το κύριο χαρακτηριστικό ενός προβλήματος *μεγέθους/διαστάσεων* είναι ότι η δομή της κατασκευής και οι παράμετροι καταστάσεως είναι γνωστές *a priori* και σταθερές κατά την διάρκεια της βελτιστοποίησης. Από την άλλη πλευρά, σε ένα πρόβλημα *βελτιστοποίησης σχήματος* σκοπός είναι η εύρεση του βέλτιστου σχήματος της δομής, και έτσι για το πρόβλημα βελτιστοποίησης σχήματος οι παράμετροι σχεδιασμού είναι η ίδια η δομή. Η *τοπολογική* βελτιστοποίηση στερεών κατασκευών προϋποθέτει τον καθορισμό συγκεκριμένων χαρακτηριστικών όπως ο αριθμός, η θέση, και το σχήμα των οπών, καθώς και η συνέχεια της δομής.

Στο σχήμα 1.3 παρουσιάζονται γραφικά οι τρεις κατηγορίες βελτιστοποίησης δομής [2]¹:

- **Διαστασιολογική Βελτιστοποίηση:** Αριστερά στο σχήμα α) παρουσιάζεται η δομή της κατασκευής. Η δομή αποτελείται από δοκούς σε συγκεκριμένες θέσεις και ζητείται να βρεθεί το βέλτιστο πάχος των δοκών ώστε η κατασκευή να αντέχει τις φορτίσεις. Δεξιά στο σχήμα β) παρουσιάζεται η κατασκευή με ορισμένες δοκούς να είναι αρκετά πιο παχιές σε σχέση με τις άλλες. Στην ουσία η δεξιά εικόνα παρουσιάζει με ποιες δοκούς μπορεί να κτιστεί η κατασκευή, και πόσο είναι το πάχος τους.
- **Βελτιστοποίηση σχήματος:** Στο σχήμα γ) παρουσιάζεται το αρχικό χωρίο σχεδιασμού με έξι οπές. Σκοπός είναι να βρεθεί το βέλτιστο σχήμα των οπών ώστε η κατασκευή να αντέξει τις φορτίσεις. Δεξιά στο σχήμα δ) παρουσιάζεται η τελική μορφή των οπών.
- **Τοπολογική βελτιστοποίηση:** Στο σχήμα ε) παρουσιάζεται το χωρίο σχεδιασμού χωρίς οπές, χωρίς προδιαγεγραμμένες θέσεις δοκών, και ζητείται ποια θα είναι η μορφή του χωρίου ώστε η κατασκευή να αντέξει τις φορτίσεις. Δεξιά στο σχήμα στ) παρουσιάζετε η βέλτιστη κατανομή του υλικού που ικανοποιεί τα κριτήρια σχεδιασμού.



Σχήμα 1.3: Τρεις κατηγορίες βελτιστοποίησης δομής.

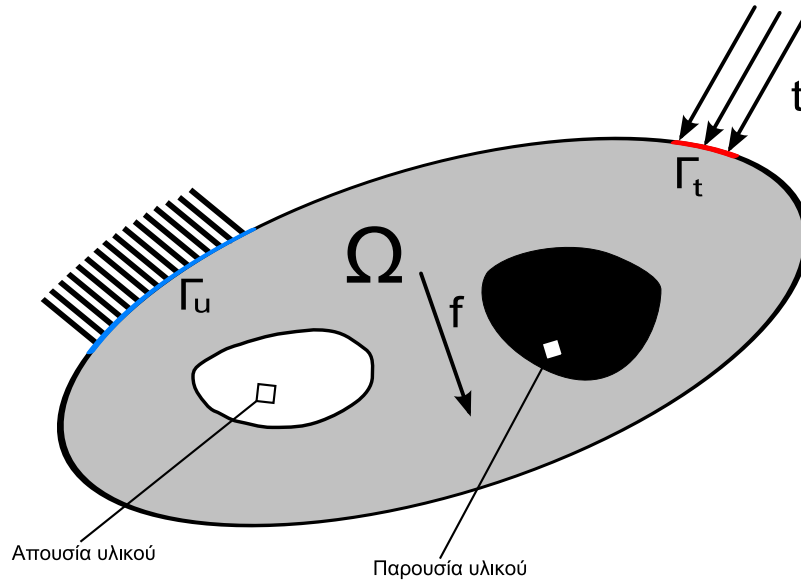
¹σελ: 1-2

1.2 Μόρφωση προβλήματος και παραμετροποίηση σχεδιασμού

Το πρόβλημα διανομής υλικού που θα οριστεί παρακάτω συνδυάζει μερικά χαρακτηριστικά από τα παραδοσιακά προβλήματα στο βέλτιστο σχεδιασμό κατασκευών. Σκοπός της τοπολογικής βελτιστοποίησης είναι η εύρεση της βέλτιστης κατανομής υλικού σε ένα δεδομένο χωρίο σχεδιασμού. Οι μόνες γνωστές παράμετροι του προβλήματος είναι οι δυνάμεις, οι πιθανές στηρίξεις, ο όγκος της κατασκευής που θα υλοποιηθεί, καθώς και επιπλέον περιορισμοί όπως η θέση και το σχήμα περιοχών με απουσία (οπές) ή παρουσία υλικού. Σε αυτό το πρόβλημα οι διαστάσεις, το σχήμα και η συνδεσμολογία (connectivity) είναι τα ζητούμενα. Η τοπολογία, το σχήμα και οι διαστάσεις της κατασκευής δεν αναπαρίστανται από συγκεκριμένες παραμετρικές συναρτήσεις, αλλά από ένα σύνολο κουκίδων (pixels), ενώ οι στόχοι σχεδιασμού προσομοιώνονται με διάφορες μαθηματικές συναρτήσεις.

1.2.1 Σχεδιασμός κατασκευής με ελαχιστοποίηση της ευκαμψίας

Στην συνέχεια θα περιγραφεί η μόρφωση του προβλήματος στο συνεχές μέσο. Ας υποθέσουμε ότι ένα ελαστικό σώμα καταλαμβάνει ένα χώρο Ω^{mat} και είναι μέρος ενός ευρύτερου χωρίου αναφοράς Ω που ανήκει στον δισδιάστατο $\mathbb{R}^2$ ή στον τρισδιάστατο χώρο $\mathbb{R}^3$. Το χωρίο αναφοράς Ω έχει επιλεγθεί έτσι ώστε να είναι δυνατός ο ορισμός των φορτίσεων και των συνοριακών συνθηκών. Η γραφική αναπαράσταση του χωρίου αναφοράς παρουσιάζεται στο σχήμα 1.4. Το πρόβλημα ορίζεται ως η εύρεση της βέλτιστης κατανομής υλικού έτσι ώστε το ολικό μέτρο δυσκαμψίας στο χωρίο ορισμού Ω , ή πιο απλά στον ορισμό του χωρίου Ω^{mat} μέσα στα όρια του Ω , ούτως ώστε να ελαχιστοποιείται η συνολική ευκαμψία και να ικανοποιούνται οι συνθήκες ισορροπίας. Το συγκεκριμένο πρόβλημα είναι ισοδύναμο με την μεγιστοποίηση της δυσκαμψίας της κατασκευής.



Σχήμα 1.4: Ορισμός αρχικού χωρίου σχεδιασμού τοπολογικής βελτιστοποίησης.

$$\begin{aligned}
& \min_{u \in U, E} l(u) \\
& \text{υπό τους περιορισμούς:} \\
& \alpha_E(u, v) = l(u), \\
& \text{για όλα τα } v \in U, \\
& E \in E_{ad}.
\end{aligned} \tag{1.1}$$

1.2.2 Μόρφωση διακριτού προβλήματος

Για να λυθεί το πρόβλημα του συνεχούς με υπολογιστικά μέσα, το αρχικό χωρίο αναφοράς Ω θα πρέπει να διακριτοποιηθεί χρησιμοποιώντας για παράδειγμα πεπερασμένα στοιχεία. Η θεμελιώδης ιδέα της *τεχνικής διανομής υλικού* για σχεδιασμό τοπολογίας είναι η συσχέτιση κάθε στοιχείου με μια ανεξάρτητη μεταβλητή σχεδιασμού που ορίζει την δυσκαμψία του υλικού E_e . Είναι σημαντικό να επισημανθούν στο πρόβλημα (1.1) δύο παράμετροι ιδιαίτερης σημασίας: οι μετακινήσεις $\mathbf{u}$ και ο γενικός συντελεστής δυσκαμψίας $\mathbf{E}$. Υπό το πρίσμα της ανάλυσης με πεπερασμένα στοιχεία το πρόβλημα (1.1) μπορεί να γραφτεί στην διακριτή του μορφή ως:

$$\begin{aligned}
& \min_{u \in U, E} \mathbf{f}^T \mathbf{u} \\
& \text{υπό τους περιορισμούς:} \\
& \mathbf{K}(E_e) \mathbf{u} = \mathbf{f}, \\
& E \in E_{ad},
\end{aligned} \tag{1.2}$$

όπου οι παράγοντες $\mathbf{u}$ και $\mathbf{f}$ είναι τα διανύσματα μετατοπίσεων και δυνάμεων, αντίστοιχα. Ο γενικός πίνακας δυσκαμψίας $\mathbf{K}$, εξαρτάται από την δυσκαμψία E_e κάθε πεπερασμένου στοιχείου e , όπου $e = 1, \dots, N$. Έτσι ο γενικός πίνακας δυσκαμψίας μπορεί να γραφτεί στην συμβολική μορφή (assembly):

$$\mathbf{K} = \sum_{e=1}^N \mathbf{K}_e(E_e), \tag{1.3}$$

Ο $\mathbf{K}_e$ είναι ο τοπικός πίνακας δυσκαμψίας του στοιχείου e .

1.2.3 Παραμετροποίηση σχεδιασμού

Για τον σχεδιασμό της τοπολογίας μιας κατασκευής, ενδιαφέρον έχει ο καθορισμός της βέλτιστης διανομής συγκεκριμένου ισότροπου υλικού, σε συγκεκριμένο χωρίο ορισμού Ω . Θα πρέπει δηλαδή, να καθοριστεί πια σημεία του χωρίου θα έχουν υλικό Ω^{mat} , και πια θα παραμείνουν κενά, χωρίς υλικό, $\Omega^{void} = \Omega - \Omega^{mat}$. Η γεωμετρική αναπαράσταση του χωρίου θα μπορούσε να αναπαρασταθεί από μια ασπρό-μαύρη εικόνα, όπου με *μαύρο* θα αναπαρίστανται οι περιοχές που περιέχουν υλικό και με *λευκό* οι περιοχές που δεν περιέχουν. Στην διακριτή μορφή, η γραφική αναπαράσταση της εικόνας γίνεται από *στοιχειώδη υποχωρία* (pixels), που καθορίζονται από την διακριτοποίηση των πεπερασμένων στοιχείων. Για κάθε ένα στοιχείο μπορεί να θεωρηθεί μια πυκνότητα ρ_e :

$$\rho_e = \begin{cases} \rho_0, & \text{αν } e \in \Omega^{mat} \\ 0, & \text{αν } e \in \Omega^{void} \end{cases} \tag{1.4}$$

όπου ρ_0 είναι η πυκνότητα του στοιχείου του υλικού όταν υπάρχει υλικό. Ορίζεται η σχετική πυκνότητα x_e ως το πηλίκο της πυκνότητας του στοιχείου e , προς ρ_0 , δηλαδή:

$$x_e = \frac{\rho_e}{\rho_0}, \quad e = 1, \dots, N \quad (1.5)$$

Με τον ορισμό της σχετικής πυκνότητας από την σχέση (1.5) το πρόβλημα βελτιστοποίησης μετατρέπεται σε πρόβλημα ακέραιου προγραμματισμού $0-1$. Η πιο συχνή στρατηγική που χρησιμοποιείται για την επίλυση προβλημάτων ακέραιου προγραμματισμού είναι η αντικατάσταση των ακέραιων μεταβλητών από συνεχείς μεταβλητές μέσα σε ένα διάστημα, εισάγοντας παράλληλα και ένα είδος ποινής ώστε οι συνεχείς μεταβλητές να οδηγούνται στα όρια του διαστήματος ορισμού τους. Έτσι το πρόβλημα σχεδιασμού για το δεδομένο χωρίο μετατρέπεται σε πρόβλημα διαστασιολόγησης μεταβάλλοντας τον πίνακα δυσκαμψίας ώστε να εξαρτάται από μια συνεχή συνάρτηση που εκφράζει την σχετική πυκνότητα του υλικού. Η συνάρτηση αυτή είναι η νέα μεταβλητή σχεδιασμού. Η μόνη απαίτηση για την συνάρτηση αυτή είναι, οι τιμές τις να δηλώνουν παρουσία ή απουσία υλικού. Μια τέτοια στρατηγική η οποία έχει αποδειχθεί εξαιρετικά επιτυχής είναι το μοντέλο *ανάλογης δυσκαμψίας ίσοτροπου υλικού με ποινή* ή αλλιώς *SIMP*². Έτσι μπορεί να αντικατασταθεί η δυσκαμψία $\mathbf{K}_e$ του στοιχείου e με:

$$\begin{aligned} \mathbf{K}_e &= x_e^p \mathbf{K}_0, \quad e = 1, \dots, N, \quad p > 1, \\ \mathbf{K}_e(x_e = 0) &= 0, \quad \mathbf{K}_e(x_e = 1) = \mathbf{K}_0. \end{aligned} \quad (1.6)$$

όπου $\mathbf{K}_0$ είναι η δυσκαμψία του στοιχείου όταν $x_e = 1$ και p ο συντελεστής ποινής που συνήθως είναι $p \geq 3$. Το ολικό μητρώο δυσκαμψίας $\mathbf{K}$ εκφράζεται ως:

$$\mathbf{K} = \sum_{e=1}^N x_e^p \mathbf{K}_0 \quad (1.7)$$

Ο όγκος του χωρίου που περιέχει υλικό $Vol(\Omega^{mat})$ θα πρέπει να είναι ένα ποσοστό του συνολικού όγκου του αρχικού χωρίου Ω . Αυτό μεταφράζεται ως περιορισμός όγκου:

$$\begin{aligned} \frac{V}{V_0} &\leq \varphi \Leftrightarrow V \leq \varphi V_0 \\ V &= \sum_{e=1}^N v_e x_e \leq \varphi V_0, \end{aligned} \quad (1.8)$$

όπου v_e είναι ο όγκος του στοιχείου e . Λαμβάνοντας υπόψιν τον περιορισμό όγκου (1.8) και την αλλαγή μεταβλητών σχεδιασμού, το πρόβλημα ελαχιστοποίησης της ευκαμψίας ορίζεται ως εξής:

$$\begin{aligned} \min_{\mathbf{u}, x_e} c(x_e) &= \mathbf{f}^T \mathbf{u}, \\ \text{υπό τους περιορισμούς:} \\ \left(\sum_{e=1}^N x_e^p \mathbf{K}_0 \right) \mathbf{u} &= \mathbf{f}, \\ \sum_{e=1}^N v_e x_e &\leq \varphi V_0, \\ 0 < x_{min} &\leq x_e \leq 1, \quad e = 1, \dots, N, \end{aligned} \quad (1.9)$$

²SIMP: Solid Isotropic Material with Penalization.

Θα πρέπει να σημειωθεί ότι η σχετική πυκνότητα έχει ένα κάτω όριο x_{min} ώστε να είναι πάντα δυνατή η επίλυση του προβλήματος ισορροπίας. Μια τυπική τιμή για το κάτω όριο μπορεί να θεωρηθεί $x_{min} = 10^{-3}$.

1.2.4 Συνθήκες βελτιστοποίησης και ανάλυση ευαισθησίας

Ο ορισμός της ευκαμψίας όπως προκύπτει από το πρόβλημα (1.9) η οποία πρέπει να ελαχιστοποιηθεί είναι:

$$c = \mathbf{f}^T \mathbf{u} = \mathbf{u}^T \mathbf{K} \mathbf{u} = \sum_{e=1}^N u_e \mathbf{K}_e u_e$$

$$\mathbf{K}_e = x_e^p \mathbf{K}_0$$

Έτσι η ευκαμψία διαμορφώνεται ως:

$$c = \sum_{e=1}^N x_e^p u_e \mathbf{K}_0 u_e, \quad e = 1, \dots, N,$$

Για την επίλυση του προβλήματος (1.9) θα χρησιμοποιηθεί η μέθοδος των πολλαπλασιαστών *Langrange*. Στην συνέχεια θα εφαρμοστεί μια ευρετική μέθοδος για την ανανέωση των μεταβλητών σχεδιασμού, ενώ η εύρεση του πολλαπλασιαστή *Lagrange* θα γίνεται μια επαναληπτική διαδικασία. Η συνάρτηση *Lagrange* ορίζεται ως εξής:

$$\mathcal{L} = c + \lambda(V - \varphi V_0) + \lambda_1^T (\mathbf{K} \mathbf{u} - \mathbf{f}) + \sum_{e=1}^N -\lambda_e (x_{min} - x_e) + \sum_{e=1}^N +\lambda_e (x_e - 1) \quad (1.11)$$

όπου λ και λ_1 οι συντελεστές *Lagrange* για τους περιορισμούς όγκου και ισορροπίας, και οι $-\lambda_e$ και $+\lambda_e$ για τους κάτω και άνω περιορισμούς διαστήματος των μεταβλητών σχεδιασμού. Ο συντελεστής *Lagrange* λ για τον περιορισμό όγκου είναι βαθμωτό, ο συντελεστής λ_1 είναι διάνυσμα, ενώ οι συντελεστές $-\lambda_e$ και $+\lambda_e$ είναι επίσης βαθμωτά μεγέθη. Το βέλτιστο βρίσκεται όταν οι μερικές παράγωγοι της συνάρτησης *Lagrange* ως προς τις μεταβλητές σχεδιασμού είναι ίσες με μηδέν:

$$\frac{\partial \mathcal{L}}{\partial x_e} = 0, \quad \text{για } e = 1, \dots, N$$

$$\frac{\partial \mathcal{L}}{\partial x_e} = \frac{\partial c}{\partial x_e} + \lambda \frac{\partial V}{\partial x_e} + \lambda_1^T \frac{\partial (\mathbf{K} \mathbf{u} - \mathbf{f})}{\partial x_e} - \lambda_e + \lambda_e \quad (1.12)$$

Υπολογίζοντας τις μερικές παραγώγους του όρου της αντικειμενικής συνάρτησης στην εξίσωση *Lagrange* έχουμε (χρησιμοποιώντας τον κανόνα της αλυσίδας):

$$\begin{aligned} \frac{\partial c}{\partial x_e} &= \frac{\partial (\mathbf{f}^T \mathbf{u})}{\partial x_e} \\ &= \frac{\partial (\mathbf{u}^T \mathbf{K} \mathbf{u})}{\partial x_e} \\ &= \frac{\partial \mathbf{u}^T}{\partial x_e} \mathbf{K} \mathbf{u} + \mathbf{u}^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} + \mathbf{u}^T \mathbf{K} \frac{\partial \mathbf{u}}{\partial x_e} \end{aligned} \quad (1.13)$$

Υπολογίζοντας τις μερικές παραγώγους του όρου περιορισμού ισορροπίας στην εξίσωση *Lagrange* έχουμε (χρησιμοποιώντας τον κανόνα της αλυσίδας):

$$\lambda_1^T \frac{\partial (\mathbf{K} \mathbf{u} - \mathbf{f})}{\partial x_e} = \lambda_1^T \left(\frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} + \mathbf{K} \frac{\partial \mathbf{u}}{\partial x_e} \right) \quad (1.14)$$

Για τον περιορισμό όγκου έχουμε:

$$\begin{aligned} \lambda \frac{\partial V}{\partial x_e} &= \lambda \frac{\partial \left(\sum_{e=1}^N v_e x_e \right)}{\partial x_e} \\ &= \lambda v_e \end{aligned} \quad (1.15)$$

Θεωρώντας ανενεργούς τους περιορισμούς διαστήματος $-\lambda_e = +\lambda_e = 0$ και ότι οι φορτίσεις είναι ανεξάρτητες των μεταβλητών σχεδιασμού, $\frac{\partial f}{\partial x_e} = 0$ τελικά οι μερικές παράγωγοι της συνάρτησης Lagrange διαμορφώνονται ως:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_e} &= \frac{\partial \mathbf{u}^T}{\partial x_e} \mathbf{K} \mathbf{u} + \mathbf{u}^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} + \mathbf{u}^T \mathbf{K} \frac{\partial \mathbf{u}}{\partial x_e} + \lambda v_e + \lambda_1^T \left(\frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} + \mathbf{K} \frac{\partial \mathbf{u}}{\partial x_e} \right) \\ \frac{\partial \mathcal{L}}{\partial x_e} &= \mathbf{u}^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} + \lambda_1^T \frac{\partial \mathbf{K}}{\partial x_e} + \frac{\partial \mathbf{u}}{\partial x_e} (2\mathbf{u}^T \mathbf{K} + \lambda_1^T \mathbf{K}) + \lambda v_e \end{aligned} \quad (1.16)$$

Αφού ο συντελεστής λ_1^T έχει οριστεί αυθαίρετα, μπορεί να επιλεχθεί έτσι ώστε οι μερικές παράγωγοι των μετακινήσεων ως προς της μεταβλητές σχεδιασμού να είναι μηδέν, $\frac{\partial \mathbf{u}}{\partial x_e} = 0$. Αφού $\mathbf{K} \neq 0$ θα πρέπει $\lambda_1^T = -2\mathbf{u}^T$, έτσι ώστε ο παράγοντας $2\mathbf{u}^T \mathbf{K} + \lambda_1^T \mathbf{K}$ να είναι ίσος με μηδέν. Έτσι οι μερικές παράγωγοι της συνάρτησης Lagrange διαμορφώνονται:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_e} &= -\mathbf{u}^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} + \lambda v_e \\ &= -p x_e^{p-1} u_e \mathbf{K}_0 u_e + \lambda v_e \\ &= -p x_e^{p-1} q_c + \lambda v_e = 0 \end{aligned} \quad (1.17)$$

όπου,

$$q_c = u_e \mathbf{K}_0 u_e, \quad (1.18)$$

που αντιπροσωπεύει την ενέργεια για το στοιχείο e , όταν αυτό περιέχει 100% υλικό, δηλαδή όταν η σχετική πυκνότητα $x_e = 1$. Η ανανέωση των μεταβλητών σχεδιασμού βασίζεται στην σχέση (1.18)

$$\frac{p x_e^{p-1} q_c}{\lambda v_e} = 1 \quad (1.19)$$

Η φυσική σημασία της σχέσης (1.18) είναι ότι η πυκνότητα ενέργειας στο σύστημα θα πρέπει να είναι σταθερή σε όλο το χωρίο σχεδιασμού. Ο πολλαπλασιαστής Lagrange λ λειτουργεί ως ένας συντελεστής κλιμακοποίησης ώστε η πυκνότητα ενέργειας να παραμένει σταθερή. Για την ανανέωση των μεταβλητών σχεδιασμού θα χρησιμοποιηθεί ένα ευρετικό σχήμα:

$$\begin{aligned} x_{e,k+1} &= x_{e,k} \left(\frac{p x_e^{p-1} q_c}{\lambda v_e} \right)^\zeta = x_{e,k} B_{e,k}^\zeta, \\ B_{e,k} &= \frac{p x_e^{p-1} q_c}{\lambda v_e}, \\ \zeta &= 0.5, \quad 0 < \zeta < 1 \end{aligned} \quad (1.20)$$

όπου το $x_{e,k}$ συμβολίζει την σχετική πυκνότητα του στοιχείου e στην k επανάληψη. Το ζ είναι ένας συντελεστής απόσβεσης συνήθως ίσος με 0.5, και χρησιμοποιείται για να σταθεροποιηθεί η επαναληπτική διαδικασία. Για να μην υπάρχουν μεγάλες αλλαγές στις μεταβλητές

σχεδιασμού μεταξύ δύο επαναλήψεων, όπως για παράδειγμα το στοιχείο e γίνει από πλήρες στερεό, κενό, εισάγεται ένα όριο μεταβολής των μεταβλητών σχεδιασμού. Αυτό γίνεται επίσης στα πλαίσια της σταθεροποίησης της επαναληπτικής διαδικασίας. Έτσι για το ευρετικό σχήμα, τα όρια μεταβολής των μεταβλητών σχεδιασμού, θα πρέπει να ληφθούν υπόψη:

$$x_{e,k+1} = \begin{cases} A_k & \text{Αν } x_{e,k} B_{e,k}^{\zeta} \leq A_k, \\ x_{e,k} B_{e,k}^{\zeta} & \text{Αν } A_k \leq x_{e,k} B_{e,k}^{\zeta} \leq C_k, \\ C_k & \text{Αν } x_{e,k} B_{e,k}^{\zeta} \geq C_k. \end{cases} \quad (1.21)$$

$$A_k = \max \{ (1 - m)x_{e,k}, x_{min} \} \quad (1.21a)$$

$$C_k = \min \{ (1 + m)x_{e,k}, 1 \} \quad (1.21b)$$

όπου m είναι το όριο κίνησης και στις περισσότερες περιπτώσεις παίρνει την τιμή 0.5, ενώ παίρνει τιμές μεταξύ 0 και 1, $0 < m < 1$. Το σχήμα (1.21) προσθέτει υλικό στα στοιχεία όπου η ενέργεια είναι μεγαλύτερη από τον συντελεστή Lagrange λ , και αυτό γίνεται όταν $B_k > 1$ και αφαιρεί υλικό όταν η ενέργεια είναι μικρότερη από λ . Αυτό συμβαίνει μόνο όταν οι νέες τιμές των μεταβλητών σχεδιασμού δεν παραβιάζουν τα όρια τους. Από την σχέση (1.19) φαίνεται ότι ο συντελεστής Lagrange είναι ανάλογος κατά p της μέσης πυκνότητας ενέργειας στις περιοχές τις κατασκευής όπου η σχετική πυκνότητα παίρνει ενδιάμεσες τιμές.

1.3 Περιγραφή του αλγορίθμου

1.3.1 Υπολογιστική διαδικασία

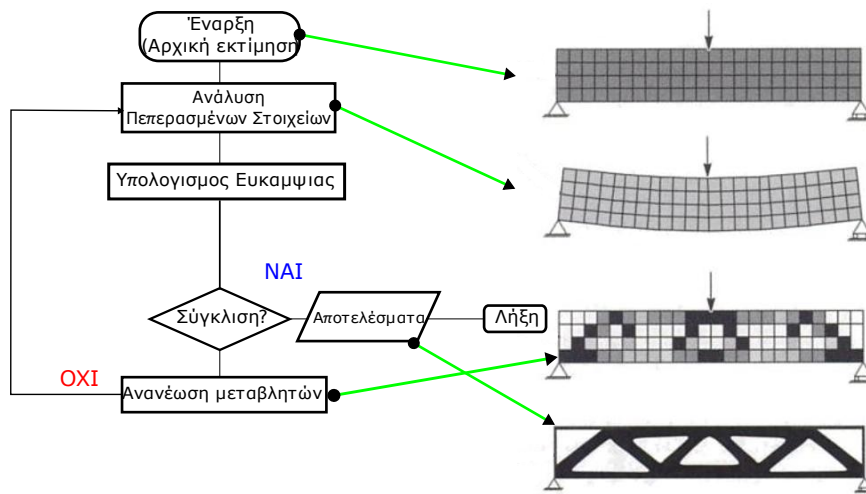
Ο σχεδιασμός τοπολογίας με χρήση μεθόδων διανομής υλικού βασίζεται στον αριθμητικό υπολογισμό της βέλτιστης διανομής της πυκνότητας του υλικού x . Το σχήμα που χρησιμοποιείται για την ποινικοποίηση των ενδιάμεσων τιμών των μεταβλητών σχεδιασμού ώστε η διανομή του υλικού να αποτελείται από στοιχεία με τιμές σχετικής πυκνότητας κοντά στο 0 ή 1, είναι η μέθοδος *SIMP*. Η μέθοδος των βέλτιστων κριτηρίων για την εύρεση της βέλτιστης τοπολογίας μιας κατασκευής που αποτελείται από ένα (και μόνο) ισότροπο υλικό μπορεί να αναλυθεί στα παρακάτω βήματα:

1. Ορισμός του αρχικού χωρίου σχεδιασμού.
2. Βελτιστοποίηση
3. Παρουσίαση αποτελεσμάτων

1.3.2 Ορισμός του αρχικού χωρίου σχεδιασμού

Στο στάδιο αυτό ορίζεται το αρχικό χωρίο σχεδιασμού Ω :

- ορίζονται τα όρια του χωρίου σχεδιασμού, καθώς και οι διάφορες στηρίξεις, κυλήσεις, δυνάμεις,
- ορίζονται εκείνες οι περιοχές του αρχικού χωρίου, όπου αναγκαστικά θα πρέπει να υπάρχει ή όχι υλικό,
- ορίζεται το πλέγμα των πεπερασμένων στοιχείων για το χωρίο σχεδιασμού. Το πλέγμα θα πρέπει να είναι αρκετά πυκνό ώστε η κατασκευή να μπορεί να αναπαρασταθεί ικανοποιητικά με στοιχεία 0-1, ικανοποιητικής ανάλυσης. Θα πρέπει επίσης να είναι δυνατός εκ των προτέρων ο ορισμός εκείνων των περιοχών της κατασκευής όπου οι



Σχήμα 1.5: Διάγραμμα ροής Αλγόριθμου Τοπολογικής Βελτιστοποίησης [2]

μεταβλητές σχεδιασμού έχουν σταθερή τιμή (παρουσία ή απουσία υλικού). Το πλέγμα θα πρέπει να παραμένει σταθερό καθ' όλη την διάρκεια της βελτιστοποίησης.

- ορίζονται τα πεδία των μετακινήσεων και των μεταβλητών σχεδιασμού ώστε να είναι δυνατή η μετεπεξεργασία των αποτελεσμάτων.

1.3.3 Βελτιστοποίηση

Στο στάδιο αυτό λαμβάνει χώρα ο υπολογισμός της βέλτιστης διανομής υλικού πάνω στο χωρίο αναφοράς για τις μεταβλητές σχεδιασμού x . Η διαδικασία της βελτιστοποίησης βασίζεται στην ανάλυση πεπερασμένων στοιχείων και στο σχήμα ανανέωσης των μεταβλητών σχεδιασμού με την χρήση κριτηρίων βελτιστοποίησης. Η δομή του αλγόριθμου είναι η παρακάτω:

1. **Βήμα 1.** Γίνεται αρχικοποίηση των μεταβλητών σχεδιασμού με ομοιόμορφη κατανομή υλικού σε όλο το χωρίο σχεδιασμού, εκτός των περιοχών που έχουν οριστεί εκ των προτέρων ως περιοχές με υποχρεωτική παρουσία ή απουσία υλικού.
2. **Βήμα 2.** Έναρξη των επαναλήψεων. Για την συγκεκριμένη κατανομή υλικού, γίνεται επίλυση του περιορισμού ισορροπίας με πεπερασμένα στοιχεία, και υπολογίζονται οι μετακινήσεις και οι παραμορφώσεις.
3. **Βήμα 3.** Γίνεται υπολογισμός της ευκαμψίας της κατασκευής. Εάν δεν υπάρχει βελτίωση (μέσα σε κάποιο διάστημα ε) σε σχέση με την προηγούμενη επανάληψη τότε γίνεται διακοπή της επαναληπτικής διαδικασίας. Αλλιώς συνεχίζεται.
4. **Βήμα 4.** Υπολογισμός των τιμών των μεταβλητών σχεδιασμού με βάση τη σχέση (1.21). Στο βήμα αυτό υπολογίζεται η τιμή του συντελεστή Lagrange λ για τον περιορισμό του όγκου.
5. **Βήμα 5.** Επανάληψη του βήματος {2}

1.3.4 Παρουσίαση των αποτελεσμάτων-Λήξη αλγορίθμου

Στο στάδιο αυτό γίνεται μια μετεπεξεργασία των τελικών αποτελεσμάτων, όπου παρουσιάζεται η τελική κατανομή του υλικού η οποία καθορίζει την βέλτιστη τοπολογία της κατασκευής.

1.3.5 Ανάλυση αλγορίθμου

Για τον παραπάνω αλγόριθμο, υπάρχει η δυνατότητα της επιλογής του βασικού σχήματος παρεμβολής, όπως για παράδειγμα το σχήμα SIMP. Είναι επίσης ενδιαφέρον να τονιστεί ότι η τοπολογική βελτιστοποίηση με χρήση του σχήματος SIMP με μια σχετικά μεγάλη τιμή για τον παράγοντα ποιότητος p δίνει πολύ καλά ορισμένες τοπολογίες που αποτελούνται από περιοχές με παρουσία ή απουσία υλικού, ενώ είναι λίγες οι περιοχές όπου οι πυκνότητες έχουν μια ενδιάμεση τιμή, δηλαδή απαλείφονται οι περιοχές χρώματος γκρι. Είναι επίσης σημαντικό να επισημανθεί ότι στον παραπάνω αλγόριθμο μπορεί να ενσωματωθεί οποιοσδήποτε τύπος πλέγματος πεπερασμένων στοιχείων για ένα οποιοδήποτε χωρίο σχεδιασμού Ω . Αυτό δίνει στον αλγόριθμο μια σημαντική ευελιξία στην συγκεκριμένη μέθοδο όσο αφορά τα όρια του χωρίου σχεδιασμού, αλλά και τον ορισμό συγκεκριμένων περιοχών του χωρίου σχεδιασμού με παρουσία ή απουσία υλικού. Ο ίδιος αλγόριθμος όπως λειτουργεί σε πεδία στις δύο διαστάσεις, μπορεί να επεκταθεί και στις τρεις διαστάσεις, όπου τα τετράπλευρα επίπεδα στοιχεία θα μπορούσαν να αντικατασταθούν με τρισδιάστατα στοιχεία. Όσο αφορά την υπολογιστική πολυπλοκότητα, ο συγκεκριμένος αλγόριθμος δεν απαιτεί ιδιαίτερη προγραμματιστική προσπάθεια. Για ένα δεδομένο κώδικα ανάλυσης πεπερασμένων στοιχείων, χρειάζεται μόνο η προσθήκη μερικών γραμμών κώδικα σχετικά με το σχήμα ανανέωσης των μεταβλητών σχεδιασμού, και τον υπολογισμό της ευκαμψίας [11].

1.4 Αριθμητικά προβλήματα

1.4.1 Εμφάνιση αριθμητικών προβλημάτων

Κατά την επίλυση προβλημάτων τοπολογικής βελτιστοποίησης είναι συχνή η εμφάνιση αριθμητικών προβλημάτων. Αυτά εξαρτώνται από:

- το είδος των πεπερασμένων στοιχείων,
- το πλήθος των στοιχείων,
- τον αλγόριθμο βελτιστοποίησης,
- τους συντελεστές του αλγορίθμου βελτιστοποίησης,
- την αρχικοποίηση του προβλήματος.

Ένα από τα πιο συχνά εμφανιζόμενα προβλήματα είναι το πρόβλημα της *εμφάνισης τυπολογίας τύπου σκακιέρας*. Οι σχετικές πυκνότητες βρίσκονται μεταξύ 0 και 1, και ως εκ τούτου συμβαίνει συχνά η εμφάνιση περιοχών της τοπολογίας όπου τα στοιχεία εναλλάσσονται μεταξύ 0 και 1, εμφανίζοντας εικόνες που μοιάζουν με σκακιέρα. Εξού και το όνομα. Ένα άλλο αριθμητικό πρόβλημα είναι η εμφάνιση διαφορετικών τοπολογιών ανάλογα με τον αριθμό των στοιχείων, δηλαδή με την πυκνότητα και το μέγεθος της διακριτοποίησης. Υπάρχει δηλαδή μια εξάρτηση της τοπολογίας από το μέγεθος συνεπώς και την πυκνότητα της διακριτοποίησης. Θα πρέπει να αναφερθεί επίσης ότι κατά την επίλυση προβλημάτων τοπολογικής βελτιστοποίησης υπάρχει η πιθανότητα εμφάνισης πολλών διαφορετικών λύσεων που αντιστοιχούν στην εμφάνιση πολλών τοπικών βέλτιστων για το κριτήριο που έχει επιλεχθεί. Για το ίδιο

πρόβλημα και την ίδια διακριτοποίηση μπορούν να προκύψουν πολλές διαφορετικές λύσεις και αυτό είναι κυρίως συνάρτηση της αρχικής μορφής που θα έχει το χωρίο σχεδιασμού Ω . Ενώ τα δύο πρώτα προβλήματα μπορούν να αντιμετωπιστούν με διάφορες τεχνικές που περιγράφονται παρακάτω, η εμφάνιση των πολλαπλών λύσεων για το ίδιο πρόβλημα και ίδια διακριτοποίηση δικαιολογεί την χρήση τεχνικών γενικής βελτιστοποίησης για την εύρεση ενός και μόνου ολικού βέλτιστου.

1.4.2 Πρόβλημα τύπου σκακιέρας

Όπως αναφέρθηκε και παραπάνω, στην τελική κατανομή υλικού που προκύπτει ως λύση του προβλήματος τοπολογικής βελτιστοποίησης υπάρχουν περιοχές που εμφανίζονται φαινόμενα τύπου σκακιέρας. Στην εικόνα 1.6 φαίνονται οι περιοχές αυτές. Το πρόβλημα προκύπτει γιατί οι δομές με διανομές τύπου σκακιέρας, εμφανίζουν μεγάλη δυσκαμψία σε σχέση με δομές που εμφανίζουν ομοιόμορφη διανομή [4].



Σχήμα 1.6: Φαινόμενα τύπου σκακιέρας

Ένας τρόπος για την αντιμετώπιση των φαινομένων αυτών είναι η εφαρμογή φίλτρων, που χρησιμοποιούνται στην επεξεργασία εικόνας. Στα περισσότερα εμπορικά λογισμικά επεξεργασίας εικόνας υπάρχουν λειτουργίες εξομάλυνσης και αφαίρεσης θορύβου από εικόνες. Τα εργαλεία αυτά αφαιρούν τις διάφορες ανωμαλίες στην εικόνα εφαρμόζοντας ένα τέτοιο φίλτρο σε κάθε pixel ή σε μια περιοχή από pixels. Στην παρακάτω εξίσωση παρουσιάζετε η μαθηματική μορφή του φίλτρου 3×3 , [12] η εφαρμογή του οποίου δεν αυξάνει απαγορευτικά το υπολογιστικό κόστος, [10]:

$$H = \frac{1}{(b+2)^2} \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}, \quad b \in [1, \infty) \quad (1.22)$$

Για μεγάλες τιμές του b τα βάρη για τα γειτονικά στοιχεία γίνονται μικρότερα. Έτσι το φίλτρο διαμορφώνεται ως εξής:

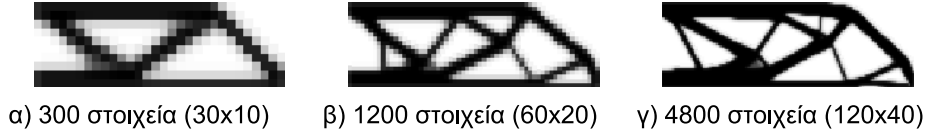
$$G(i, j) = \sum_{m=1}^3 \sum_{n=1}^3 F(m, n) H(m + i - C, n + j - C) \quad (1.23)$$

όπου $C = \frac{3+1}{2} = 2$

και $G(i, j)$ είναι η εξομαλυνμένη τιμή της σχετικής πυκνότητας $F(i, j)$. Όταν το b τείνει στο άπειρο το $G(i, j)$ τείνει στην αρχική τιμή $F(i, j)$, [10].

1.4.3 Πρόβλημα εξάρτησης πλέγματος

Όπως έχει αναφερθεί και παραπάνω διαφορετικές λύσεις προκύπτουν όταν αλλάξει το μέγεθος της διακριτοποίησης του Ω . Τα φαινόμενα αυτά παρουσιάζονται στην εικόνα 1.7 και δείχνουν ότι η τελική μορφή εξαρτάται από την πύκνωση του πλέγματος, [10].



Σχήμα 1.7: Φαινόμενα εξάρτησης πλέγματος

Φαίνεται ότι ακόμα και αν οι αρχικές συνθήκες είναι ίδιες όταν αλλάζει το μέγεθος της διακριτοποίησης η τελική λύση αλλάζει. Θα πρέπει να σημειωθεί ότι η αναλογία, των στοιχείων στην οριζόντια και κάθετη διεύθυνση παραμένει σταθερή. Η ιδέα αύξησης της διακριτοποίησης προκύπτει από την ανάγκη εύρεσης μιας καλύτερης λύσης και ένα καλύτερο ορισμό των συνόρων του Ω^{mat} . Η πιο λεπτομερής λύση που προκύπτει από την γ) περίπτωση με 4800 στοιχεία δημιουργεί περισσότερο προβλήματα στην πρακτική εφαρμογή της δομής αυτής. Ένας τρόπος για την εξάλειψη του προβλήματος αυτού είναι η εφαρμογή ενός φίλτρου σταθερού μεγέθους που να είναι ανεξάρτητο της διακριτοποίησης. Η μέθοδος είναι καθαρά ευρετική αλλά δίνει πολύ καλά αποτελέσματα με ένα μικρό υπολογιστικό κόστος. Καταρχήν ορίζεται η απόσταση r_{min} η οποία μπορεί να εξαρτάται από το μέγεθος οριζόντιας διακριτοποίησης, για παράδειγμα μπορεί να είναι το 5% της, και την απόσταση $dist(e, i)$ μεταξύ του στοιχείου e και i . Το φίλτρο μπορεί να εφαρμοστεί τόσο στις μεταβλητές σχεδιασμού αλλά και στις τιμές των παραγώγων και μπορεί να διαμορφωθεί ως εξής:

$$\widehat{H}_i = r_{min} - dist(e, i), \{i \in N_r \mid dist(e, i) \leq r_{min}, e = 1, \dots, N\} \quad (1.24)$$

όπου N_r είναι το σύνολο των στοιχείων που η απόσταση τους από το στοιχείο k είναι μικρότερη ή ίση από r_{min} . Προφανώς το $\widehat{H}$ είναι μηδέν για τα στοιχεία που απέχουν περισσότερο από r_{min} από το στοιχείο e . Η επίδραση του φίλτρου στις μεταβλητές σχεδιασμού είναι:

$$\widehat{x}_e = \frac{1}{\left(\sum_{i=1}^{N_r} \widehat{H}_f\right)} \sum_{i=1}^{N_r} \widehat{H}_i x_i \quad (1.25)$$

Η επίδραση του φίλτρου στις παραγώγους είναι:

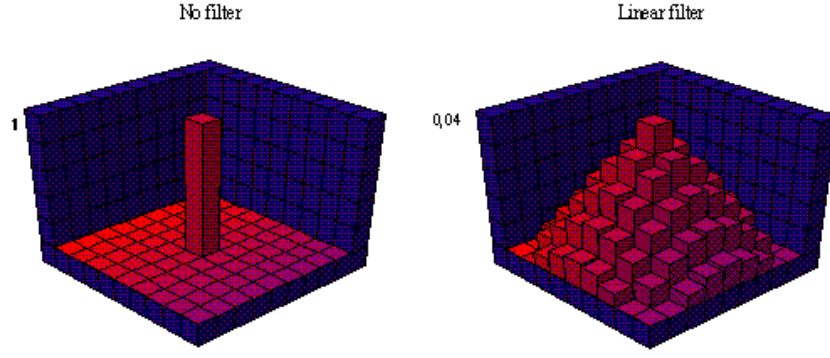
$$\frac{\partial \hat{c}}{\partial x_e} = \frac{1}{x_e \left(\sum_{f=1}^{N_r} \widehat{H}_f\right)} \sum_{i=1}^{N_r} \widehat{H}_i x_i \frac{\partial c}{\partial x_e} \quad (1.26)$$

και το οποίο μπορεί να αναπαρασταθεί γραφικά στο σχήμα 1.8:

1.5 Τοπολογική Βελτιστοποίηση με περισσότερες περιπτώσεις φορτίσεων

1.5.1 Εισαγωγή στο πρόβλημα πολλαπλών φορτίσεων

Το πρόβλημα ελαχιστοποίησης της ευκαμψίας μιας κατασκευής όταν αυτή υπόκειται σε μια και μόνη φόρτιση μπορεί να γενικευτεί και για περισσότερες φορτίσεις. Θα πρέπει να διευκρινιστεί ότι, όταν γίνεται αναφορά σε πολλαπλές φορτίσεις, οι φορτίσεις αυτές δεν επιβάλλονται ταυτόχρονα, αλλά κάθε φορά. Αν στην κατασκευή εφαρμόζονται περισσότερα από ένα φορτία ταυτόχρονα τότε οι φορτίσεις αυτές καθιστούν μια περίπτωση φόρτισης.



Σχήμα 1.8: Γραφική αναπαράσταση φίλτρου για το πρόβλημα εξάρτησης πλέγματος (γραμμική)

Όταν όμως η κατασκευή υπόκειται σε διαφορετικές φορτίσεις κάθε φορά τότε έχουμε μόρφωση προβλήματος τοπολογικής βελτιστοποίησης πολλαπλών φορτίσεων.

1.5.2 Μόρφωση προβλήματος

Στο συγκεκριμένο πρόβλημα ζητείται η ελαχιστοποίηση της ευκαμψίας της κατασκευής για όλες τις περιπτώσεις φορτίσεων:

$$\begin{cases} \min_{\mathbf{u}_1, x_e} c_1(x_e) &= \mathbf{f}_1^T \mathbf{u}_1, \\ \min_{\mathbf{u}_2, x_e} c_2(x_e) &= \mathbf{f}_2^T \mathbf{u}_2, \\ &\vdots \\ \min_{\mathbf{u}_m, x_e} c_m(x_e) &= \mathbf{f}_m^T \mathbf{u}_m, \end{cases} \quad (1.27)$$

όπου m είναι το σύνολο των διαφορετικών περιπτώσεων φόρτισης. Το πρόβλημα είναι πολυκριτήριο αφού κάθε κριτήριο αποτελείται από την ελαχιστοποίηση της ευκαμψίας για μία περίπτωση φόρτισης. Το πολυκριτήριο πρόβλημα μπορεί να αντικατασταθεί από ένα και μόνο κριτήριο που μπορεί να αποτελείται από το σταθμισμένο άθροισμα των ευκαμψιών [2]:

$$\begin{aligned} \min \sum_{i=1}^m w_i c_i &= \sum_{i=1}^m w_i \mathbf{u}_i^T \mathbf{K} \mathbf{u}_i \\ \text{υπό τους περιορισμούς:} \\ \left(\sum_{e=1}^N x_e^p \mathbf{K}_0 \right) \mathbf{u}_i &= \mathbf{f}_i, \quad i = 1, \dots, m \\ \sum_{e=1}^N v_e x_e &\leq \varphi V_0, \\ 0 < x_{\min} &\leq x_e \leq 1, \quad e = 1, \dots, N, \\ \sum_{i=1}^m w_i &= 1 \end{aligned} \quad (1.28)$$

όπου w_i το βάρος για κάθε περίπτωση φόρτισης. Θα πρέπει το σύνολο των βαρών να είναι ίσο με την μονάδα. Στην περίπτωση αυτή η συνάρτηση Lagrange $\mathcal{L}$ παίρνει την παρακάτω

μορφή σύμφωνα και με την σχέση (1.11):

$$\mathcal{L} = \sum_{i=1}^m w_i \mathbf{u}_i^T \mathbf{K} \mathbf{u}_i + \lambda(V - \varphi V_0) + \sum_{i=1}^m \lambda_i^T (\mathbf{K} \mathbf{u}_i - \mathbf{f}_i) + \sum_{e=1}^N -\lambda_e (x_{\min} - x_e) + \sum_{e=1}^N +\lambda_e (x_e - 1) \quad (1.29)$$

Οι μερικές παράγωγοι της συνάρτησης Lagrange υπολογίζονται όπως ακριβώς και στην περίπτωση της μιας φόρτισης (βλ. σχέση (1.16)):

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_e} &= 0, \quad \text{για } e = 1, \dots, N \\ \frac{\partial \mathcal{L}}{\partial x_e} &= \frac{\partial \left(\sum_{i=1}^m w_i \mathbf{u}_i^T \mathbf{K} \mathbf{u}_i \right)}{\partial x_e} + \lambda \frac{\partial V}{\partial x_e} + \sum_{i=1}^m \lambda_i^T \left(\frac{\partial (\mathbf{K} \mathbf{u}_i)}{\partial x_e} - \frac{\partial \mathbf{f}_i}{\partial x_e} \right) - \lambda_e + \lambda_e \Rightarrow \\ \frac{\partial \mathcal{L}}{\partial x_e} &= w_1 \frac{\partial \mathbf{u}_1^T}{\partial x_e} \mathbf{K} \mathbf{u}_1 + w_1 \mathbf{u}_1^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_1 + w_1 \mathbf{u}_1^T \mathbf{K} \frac{\partial \mathbf{u}_1}{\partial x_e} + \dots \\ &\quad + w_m \frac{\partial \mathbf{u}_m^T}{\partial x_e} \mathbf{K} \mathbf{u}_m + w_m \mathbf{u}_m^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_m + w_m \mathbf{u}_m^T \mathbf{K} \frac{\partial \mathbf{u}_m}{\partial x_e} + \lambda \frac{\partial V}{\partial x_e} \\ &\quad + \lambda_1^T \left(\frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_1 + \mathbf{K} \frac{\partial \mathbf{u}_1}{\partial x_e} \right) + \dots + \lambda_m^T \left(\frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_m + \mathbf{K} \frac{\partial \mathbf{u}_m}{\partial x_e} \right) \Rightarrow \\ \frac{\partial \mathcal{L}}{\partial x_e} &= \sum_{i=1}^m w_i \mathbf{u}_i^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_i + \sum_{i=1}^m w_i \lambda_i^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_i + \sum_{i=1}^m \frac{\partial \mathbf{u}_i}{\partial x_e} (2w_i \mathbf{u}_i^T \mathbf{K} + \lambda_i^T \mathbf{K}) + \lambda v_e \end{aligned} \quad (1.30)$$

Αφού ο συντελεστής λ_i^T έχουν οριστεί αυθαίρετα, μπορούν να επιλεγθούν έτσι ώστε οι μερικές παράγωγοι των μετακινήσεων ως προς της μεταβλητές σχεδιασμού για όλες τις περιπτώσεις φορτίσεων να είναι μηδέν, $\frac{\partial \mathbf{u}_i}{\partial x_e} = 0$, $i = 1, \dots, m$. Αφού $\mathbf{K} \neq 0$ θα πρέπει $\lambda_i^T = -2\mathbf{u}_i^T$, έτσι ώστε οι παράγοντες $2\mathbf{u}_i^T \mathbf{K} + \lambda_i^T \mathbf{K}$, $i = 1, \dots, m$ να είναι ίσος με μηδέν. Έτσι οι μερικές παράγωγοι της συνάρτησης Lagrange διαμορφώνονται:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_e} &= - \sum_{i=1}^m w_i \mathbf{u}_i^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_i + \lambda v_e \\ &= - \sum_{i=1}^m w_i (p x_e^{p-1}) u_{e,i} \mathbf{K}_0 u_{e,i} + \lambda v_e \\ &= - p x_e^{p-1} \sum_{i=1}^m w_i u_{e,i} \mathbf{K}_0 u_{e,i} + \lambda v_e \\ &= - p x_e^{p-1} \widehat{q}_c + \lambda v_e = 0 \end{aligned} \quad (1.31)$$

όπου,

$$\widehat{q}_c = \sum_{i=1}^m w_i u_{e,i} \mathbf{K}_0 u_{e,i}, \quad (1.32)$$

και αντιπροσωπεύει την ενέργεια για το στοιχείο e , όταν αυτό περιέχει 100% υλικό, δηλαδή όταν η σχετική πυκνότητα $x_e = 1$ για όλες τις φορτίσεις. Το ευρετικό σχήμα για την ανανέωση των μεταβλητών σχεδιασμού είναι αντίστοιχο αυτού για την μια περίπτωση φόρτισης (βλ.

σχέση 1.20):

$$\begin{aligned} x_{e,k+1} &= x_{e,k} \left(\frac{p x_e^{p-1} \hat{q}_c}{\lambda v_e} \right)^\zeta = x_{e,k} B_{e,k}^\zeta, \\ B_{e,k} &= \frac{p x_e^{p-1} \hat{q}_c}{\lambda v_e}, \\ \zeta &= 0.5, \quad 0 < \zeta < 1 \end{aligned} \quad (1.33)$$

ενώ το σχήμα για την ανανέωση των μεταβλητών σχεδιασμού είναι αυτό της σχέσης 1.21 μετά την αντικατάσταση του q με το $\hat{q}$.

1.6 Υλοποίηση Τοπολογικής Βελτιστοποίησης σε Matlab

1.6.1 Γενική περιγραφή

Ο αλγόριθμος για την τοπολογική βελτιστοποίηση κατασκευών μπορεί να υλοποιηθεί σε περιβάλλον προγραμματισμού MATLAB σε μόλις 99 γραμμές κώδικα [11]. Οι 99 γραμμές χωρίζονται σε διάφορα τμήματα: 36 γραμμές για την υλοποίηση του κύριου πρόγραμμα, 12 γραμμές για αλγόριθμο βελτιστοποίησης που βασίζεται σε κριτήρια βελτίστου (Optimality Criteria), 16 γραμμές για την εφαρμογή του φίλτρου που αντιμετωπίζει το αριθμητικό πρόβλημα εξάρτησης πλέγματος, και 35 γραμμές για τον ορισμό και την επίλυση του προβλήματος των πεπερασμένων στοιχείων. Με μερικές απλές προσθήσεις και μετατροπές ο ίδιος κώδικας μπορεί να μετασχηματιστεί για την επίλυση προβλημάτων για περισσότερες περιπτώσεις φορτίσεων και την σύνθεση εύκαμπτων μηχανισμών.

1.6.2 Εκτέλεση του λογισμικού

Το λογισμικό εκτελείται με την πληκτρολόγηση στην γραμμή εντολών του MATLAB:

```
top(nelx,nely,volfrac,penal,rmin)
```

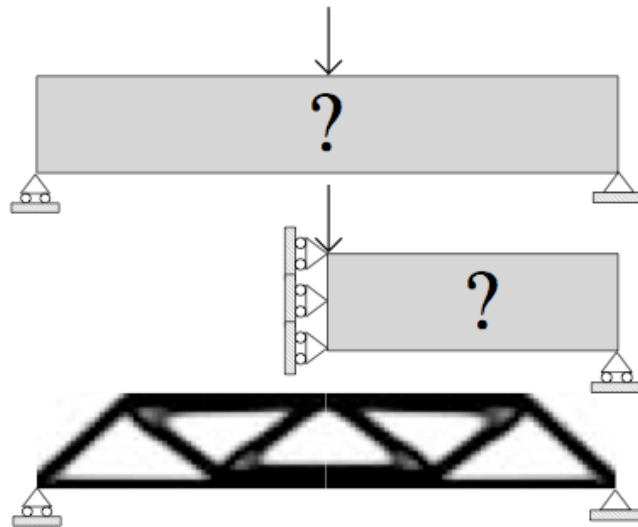
όπου, *nelx*, και *nely* καθορίζουν την διάσταση του πλέγματος, *volfrac* είναι το ποσοστό του τελικού όγκου που θα καταλαμβάνει η τελική κατανομή, *penal* η παράμετρος ποινής p για την μέθοδο SIMP, και *rmin* είναι η παράμετρος που καθορίζει την ακτίνα για την επιβολή του φίλτρου ομαλοποίησης για το αριθμητικό πρόβλημα εξάρτησης πλέγματος. Άλλες παράμετροι που καθορίζουν τους περιορισμούς κίνησης, τις συνοριακές συνθήκες, τις φορτίσεις, ορίζονται εντός του προγράμματος. Στο σχήμα 1.9 παρουσιάζεται ένα παράδειγμα εφαρμογής του συγκεκριμένου κώδικα που παρουσιάζεται στο παράρτημα Α':

```
\top(60,20,0.5,3.0,1.5)
```

Το φορτίο εφαρμόζεται στην πάνω αριστερά κορυφή, έχοντας ορίσει σαφείς περιορισμούς συμμετρίας στην αριστερή πλευρά, ενώ ταυτόχρονα η κατασκευή υποστηρίζεται στην κάτω δεξιά κορυφή. Λεπτομέρειες για το κώδικα σε MATLAB παρουσιάζονται παρακάτω.

1.6.3 Κύριο πρόγραμμα (γραμμές 1 έως 36)

Το κύριο πρόγραμμα (γραμμές 1-36) ξεκινά με τον ορισμό της αρχικής κατανομής στο χωρίο ορισμού. Συγκεκριμένα γίνεται ομοίομορφη κατανομή υλικού σε όλο το χωρίο σχεδιασμού (γραμμή 4). Έπειτα από το ορισμό κάποιων παραμέτρων ξεκινάει η επαναληπτική διαδικασία στην γραμμή 12, εκτελώντας την υπορουτίνα επίλυσης πεπερασμένων στοιχείων (FEM). Η υπορουτίνα επιστρέφει τις μετακινήσεις με το διάνυσμα $\mathbf{U}$. Δεδομένου ότι το τοπικό



Σχήμα 1.9: Τοπολογική βελτιστοποίηση κατασκευής: Επάνω παρουσιάζεται το αρχικό χωρίο σχεδιασμού, στην μέση παρουσιάζεται η απλοποίηση του προβλήματος με την χρήση συμμετρίας και οι συνοριακές συνθήκες, και κάτω παρουσιάζεται η τελική κατανομή [11].

μητρώο δυσκαμψίας για το υλικό είναι σταθερό για κάθε ένα στοιχείο όταν η πυκνότητα είναι ίση με την μονάδα, η υπορουτίνα για τον υπολογισμό του καλείται μόνο μια φορά (γραμμή 14). Στην συνέχεια υπολογίζονται για όλα τα στοιχεία η αντικειμενική συνάρτηση και οι παράγωγοι ευαισθησίας (16-24). Οι μεταβλητές $n1$, και $n2$, καθορίζουν την πάνω αριστερά και δεξιά κορυφή κάθε στοιχείου μέσα στο πλέγμα, και χρησιμοποιούνται για τον υπολογισμό των μετακινήσεων κάθε στοιχείου U_e από το γενικό διάνυσμα μετακινήσεων U . Μετά τον υπολογισμό της ευαισθησίας εκτελείται η υπορουτίνα για την εφαρμογή του φίλτρου για το αριθμητικό πρόβλημα εξάρτησης πλέγματος (mesh-independency filter) (γραμμή 26), και στην συνέχεια καλείται η υπορουτίνα βελτιστοποίησης. Η τρέχουσα τιμή της αντικειμενικής συνάρτησης (δυσκαμψία) καθώς και άλλες πληροφορίες εκτυπώνονται (γραμμές 30-33), και γίνεται γραφική αναπαράσταση της τρέχουσας κατανομής (γραμμή 35). Η επαναληπτική διαδικασία τερματίζεται αν δεν υπάρξει αλλαγή στις μεταβλητές σχεδιασμού μεγαλύτερη του 1%.

1.6.4 Κριτήρια Βέλτιστου (γραμμές 37-48)

Οι νέες τιμές των μεταβλητών σχεδιασμού υπολογίζονται από την συγκεκριμένη υπορουτίνα (OC) (γραμμές 37-48). Λαμβάνοντας υπόψιν ότι ο όγκος του υλικού ($\text{sum}(\text{sum}(\mathbf{x}_{\text{new}}))$) είναι φθίνουσα μονότονη συνάρτηση του πολλαπλασιαστή Lagrange, η τιμή του πολλαπλασιαστή μπορεί να βρεθεί με την μέθοδο διχοτόμησης (γραμμές 40-48). Η μέθοδος της διχοτόμησης ξεκινά με μια εκτίμηση των κάτω και πάνω ορίων, l_1 και l_2 του διαστήματος που περικλείει τον πολλαπλασιαστή Lagrange (γραμμή 39). Το διάστημα αυτό διχοτομείται συνεχώς στο μισό μέχρι να ικανοποιηθούν τα κριτήρια σύγκλισης (γραμμή 40), και έτσι υπολογίζεται η τιμή του πολλαπλασιαστή.

1.6.5 Εφαρμογή φίλτρου για το πρόβλημα εξάρτησης πλέγματος (γραμμές 49-64)

Στις γραμμές 49 έως 64 γίνεται υλοποίηση του φίλτρου για την αποφυγή του αριθμητικού προβλήματος εξάρτησης πλέγματος. Θα πρέπει να σημειωθεί ότι δεν ερευνώνται όλα τα στοιχεία του πλέγματος για να βρεθούν τα στοιχεία που βρίσκονται εντός της ακτίνας $r_{\min}$, αλλά μόνο αυτά που βρίσκονται εντός τετραγώνου, με πλευρά δύο φορές την ακτίνα $r_{\min}$.

γύρω από το συγκεκριμένο στοιχείο. Επιλέγοντας ακτίνα μικρότερη από την μονάδα, η εφαρμογή του φίλτρου δεν θα έχει καμία σημασία στην κατανομή.

1.6.6 Ανάλυση πεπερασμένων στοιχείων (γραμμές 65-99)

Ο κώδικας για την επίλυση του πλέγματος πεπερασμένων στοιχείων βρίσκεται στις γραμμές 65 έως 69. Θα πρέπει να σημειωθεί ότι ο επιλύτης λαμβάνει υπόψη την επιλογή για χρήση αραιών πινάκων (*sparse matrix*) στο MATLAB για γρήγορη επίλυση. Στην αρχή γίνεται σύνθεση του γενικού πίνακα δυσκαμψίας (γραμμές 70-77). Όπως έχει ειπωθεί και παραπάνω, οι μεταβλητές *n1* και *n2* δηλώνουν τις πάνω αριστερά και δεξιά κορυφές του στοιχείου μέσα στο πλέγμα και χρησιμοποιούνται για τοποθέτηση του πίνακα δυσκαμψίας κάθε στοιχείου στη σωστή θέση μέσα στον γενικό πίνακα δυσκαμψίας.

Η αρίθμηση των κόμβων και των στοιχείων γίνεται σε στήλες, αριστερά προς τα δεξιά. Κάθε κόμβος έχει δύο βαθμούς ελευθερίας (κάθετο και οριζόντιο), έτσι με την εντολή $F(2,1) = -1$ στην γραμμή 79 γίνεται δήλωση της εισερχόμενης δύναμης στον πρώτο κόμβο στην κάθετη κατεύθυνση. Ο πρώτος κόμβος είναι η πάνω αριστερά κορυφή του χωρίου ορισμού.

Οι στηρίξεις υλοποιούνται με τον ορισμό ενός διανύσματος *fixeddofs*. Η επίλυση στο MATLAB γίνεται με την εντολή στην γραμμή 84:

```
84 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
```

όπου *freedofs* οι βαθμοί ελευθερίας του πλέγματος. Συνήθως είναι πιο εύκολο να ορισθεί πρώτα το διάνυσμα με τους περιορισμένους βαθμούς ελευθερίας (*fixeddofs*) και έτσι το διάνυσμα των μη περιορισμένων βαθμών ελευθερίας προκύπτει αν από διάνυσμα *alldofs* αφαιρεθεί το διάνυσμα *fixeddofs*. Αυτό επιτυγχάνεται πολύ εύκολα με την εντολή στην γραμμή 82:

```
82 freedofs = setdiff(alldofs,fixeddofs);
```

Ο υπολογισμός του πίνακα δυσκαμψίας για το στοιχείο, βρίσκεται στις γραμμές 86-99. Ο 8x8 πίνακας για το τετραγωνικό γραμμικό επίπεδο στοιχείο με τέσσερις κόμβους και οκτώ βαθμούς ελευθερίας υπολογίζεται αναλυτικά ενώ ο λόγος Poisson *nu* και το μέτρο ελαστικότητας *E* ορίζονται στις γραμμές 88 και 89 αντίστοιχα.

1.6.7 Πολλές περιπτώσεις φορτίσεων

Ο συγκεκριμένος αλγόριθμος μπορεί πολύ εύκολα να επεκταθεί για την μοντελοποίηση περισσότερων φορτίσεων. Συγκεκριμένα στην γραμμή 69 θα πρέπει να οριστούν τα διανύσματα των δυνάμεων και μετακινήσεων (ανά δύναμη) ως:

```
69 F = sparse(2*(nely+1)*(nelx+1),2); U = sparse(2*(nely+1)*(nelx+1),2);
```

Η αντικειμενική συνάρτηση είναι το άθροισμα των δυσκαμψιών ανά φόρτιση:

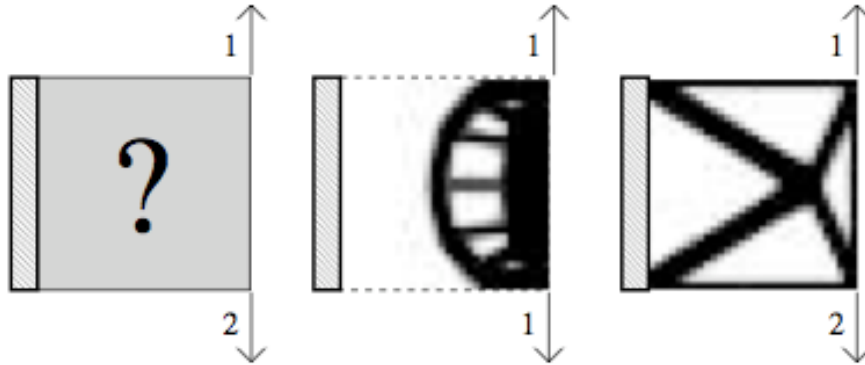
$$c = \sum_{i=1}^2 \mathbf{u}_i^T \mathbf{K} \mathbf{u}_i \quad (1.34)$$

και έτσι οι γραμμές 20-22 αντικαθίστανται από τις παρακάτω:

```
19b dc(ely,elx) = 0.;
19c for i = 1:2
20   Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2;2*n1+1;2*n1+2],i);
21   c = c + x(ely,elx)^penal*Ue'*KE*Ue;
22   dc(ely,elx) = dc(ely,elx) - penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
22b end
```

Θα πρέπει επίσης να προστεθεί η δεύτερη φόρτιση στον πίνακα φορτίσεων F (βλέπε σχήμα 1.10):

```
79 F(2*(nelx+1)*(nely+1),1) = -1.; F(2*(nelx)*(nely+1)+2,2) = 1.;
```



Σχήμα 1.10: Τοπολογική βελτιστοποίηση δοκού για δύο περιπτώσεις φορτίσεων: Αριστερά το χωρίο ορισμού με αντίθετες φορτίσεις στις πάνω και κάτω δεξιά κορυφές, στη μέση η τελική κατανομή όπως προκύπτει αν οι φορτίσεις εφαρμόζονται ταυτόχρονα και δεξιά η τελική κατανομή όπως προκύπτει στην περίπτωση των δύο διαφορετικών φορτίσεων [11].

Για την εκτέλεση του αλγορίθμου για το παραπάνω παράδειγμα:

```
top(30,30,0.4,3.0,1.2)
```

ΤΟΠΟΛΟΓΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΥΚΑΜΠΤΩΝ ΜΗΧΑΝΙΣΜΩΝ

2.1 Σύνθεση εύκαμπτων μηχανισμών

2.1.1 Εισαγωγή στους εύκαμπτους μηχανισμούς

Οι εύκαμπτοι μηχανισμοί είναι παραμορφώσιμα σώματα κατάλληλης μορφής τα οποία όταν φορτιστούν κατάλληλα λειτουργούν ως μηχανισμοί, σε αντίθεση με τους παραδοσιακούς μηχανισμούς που η κινητικότητα τους προέρχεται από αρθρώσεις, έδρανα κύλισης, οδοντωτούς τροχούς, κ.α. Τα κύρια πλεονεκτήματα των μηχανισμών είναι ότι μπορούν να κατασκευαστούν χρησιμοποιώντας λιγότερα εξαρτήματα, απαιτούν σαφώς λιγότερες διαδικασίες συναρμολόγησης ενώ δεν χρειάζονται λίπανση. Μια ευρεία εφαρμογή των εύκαμπτων μηχανισμών λαμβάνει χώρα στα MEMS (Micro Electro-Mechanical Systems) τα οποία δεν μπορούν να κατασκευαστούν με συμβατικές διαδικασίες παραγωγής και συναρμολόγησης.

Η λειτουργία ενός εύκαμπτου μηχανισμού βασίζεται στην μεταφορά μιας εισερχόμενης δύναμης που ασκείται σε ένα μέρος του μηχανισμού, σε ένα άλλο σημείο με την μορφή μετακινήσεων ή μορφή εξερχόμενης δύναμης. Ο εύκαμπτος μηχανισμός αποτελείται από ένα και μόνο σώμα (μονολιθική δομή). Το σώμα αυτό θα πρέπει να είναι αρκετά εύκαμπτο ώστε να μπορεί να μεταφέρει τις εισερχόμενες φορτίσεις σε μετακινήσεις, άλλα ταυτόχρονα θα πρέπει να είναι αρκετά στιβαρό ώστε να μπορεί να αντέξει τις φορτίσεις αυτές. Επίσης κατά τον σχεδιασμό ενός εύκαμπτου μηχανισμού και την μελέτη της κινητικότητας του θα πρέπει να ληφθεί υπόψιν και ο παράγοντας της κόπωσης. Αν οι απώλειες ενέργειας και ο παράγοντας αδράνειας θεωρηθούν αμελητέα σε ένα εύκαμπτο μηχανισμό η εισερχόμενη ενέργεια είναι ίση με το άθροισμα της εξερχόμενης ενέργειας και αυτής που είναι αποθηκευμένη στον μηχανισμό (strain energy). Ο λόγος της (εξερχόμενης) δύναμης στο σημείο εξόδου (output point) του μηχανισμού προς την εισερχόμενη δύναμη στο σημείο εισόδου (input point) ορίζεται ως *μηχανικό πλεονέκτημα*, (Mechanical Advantage, MA) ενώ ο λόγος της εξερχόμενης μετακίνησης στο σημείο εξόδου προς την μετακίνηση στο σημείο εφαρμογής της εισερχόμενης δύναμης ορίζεται ως το *Γεωμετρικό Πλεονέκτημα*, (Geometric Advantage, GA) [6].

$$\begin{aligned} MA &= \frac{F_{out}}{F_{in}} \\ GA &= \frac{U_{out}}{U_{in}} \end{aligned} \quad (2.1)$$

Ένας από τους πιο σημαντικούς στόχους στην σύνθεση εύκαμπτων μηχανισμών είναι ο έλεγχος των παραπάνω μαθηματικών λόγων αφού μπορούν να χρησιμοποιηθούν ως κριτήρια-στόχοι στο πρόβλημα βέλτιστου σχεδιασμού εύκαμπτου μηχανισμού

Ένα παράδειγμα εύκαμπτου μηχανισμού είναι αυτό που επιτυγχάνει την αντιστροφή της δύναμης. Σκοπός του μηχανισμού είναι να μετατρέψει μια εφαρμοσμένη δύναμη στο σημείο εισόδου σε μετακίνηση στην αντίθετη κατεύθυνση στο σημείο εξόδου (βλ. εικόνα). Για να είναι δυνατή η μεταφορά ενέργειας από το σημείο εισόδου ως το σημείο εξόδου η κατανομή του υλικού θα πρέπει να είναι τέτοια ώστε ο στόχος αυτός να είναι εφικτός. Ταυτόχρονα θα πρέπει να υπάρχει έλεγχος του μεγέθους των μετακινήσεων στο σημείο εξόδου. Η τοπολογική βελτιστοποίηση είναι το εργαλείο που αναλαμβάνει να βρει την βέλτιστη κατανομή υλικού ώστε να είναι πιο αποτελεσματική η μεταφορά της δύναμης σε κίνηση.

Για να γίνει πιο ρεαλιστικός σχεδιασμός ενός εύκαμπτου μηχανισμού θα πρέπει να χρησιμοποιηθεί θεωρία μη γραμμικών (μεγάλων) μετακινήσεων. Η χρήση μη γραμμικών πεπερασμένων στοιχείων επιτρέπει την μελέτη μεγάλων μετακινήσεων και μεγάλων παραμορφώσεων παράγοντες που είναι αρκετά σημαντικοί κατά την σύνθεση ενός εύκαμπτου μηχανισμού. Στην παρούσα εργασία για λόγους απλούστευσης έχει γίνει χρήση γραμμικών πεπερασμένων στοιχείων, ενώ μπορεί κάλλιστα να επεκταθεί χωρίς ιδιαίτερο κόπο και στην μην γραμμική περιοχή.

2.1.2 Μόρφωση προβλήματος

Το ζητούμενο σε ένα εύκαμπτο μηχανισμό είναι να βρεθεί η κατάλληλη τοπολογία για την μεταφορά δύναμης ή ενέργειας από ένα σημείο σε κάποιο άλλο. Για την μοντελοποίηση της εισερχόμενης στο σύστημα δύναμης, μπορεί να χρησιμοποιηθεί το μοντέλο δύναμης-ελατηρίου δεδομένης δυσκαμψίας k_{in} . Φυσικά αυτό, θα ήταν ισοδύναμο με ένα γραμμικό επενεργητή (actuator) που θα ασκούσε δύναμη σε συγκεκριμένη κατεύθυνση. Στην περίπτωση των MEMS, επενεργητής θα μπορούσε να ήταν κάποιο πιεζο-ηλεκτρικό ή ηλεκτροθερμικό στοιχείο το οποίο χαρακτηρίζεται από την δύναμη f_{in} που ασκεί και από την μετατόπιση $u_{in} = \frac{f_{in}}{k_{in}}$ που επιδέχεται. Σκοπός του προβλήματος βελτιστοποίησης είναι η μεγιστοποίηση της μετακίνησης u_{out} (της ενέργειας, ή της δύναμης f_{out}) στο σημείο εξόδου. Στο σημείο εξόδου θα μπορούσε να υπάρχει σύστημα ελατηρίου με συντελεστή δυσκαμψίας k_{out} . Για μικρές τιμές του k_{out} οι μετακινήσεις θα είναι μεγάλες και το αντίθετο. Με αυτό τον τρόπο επιτυγχάνεται έλεγχος στο μέγεθος των μετακινήσεων στο σημείο εξόδου. Για να υπάρξει μεγιστοποίηση της μεταφοράς ενέργειας με τις μικρότερες απώλειες, θα πρέπει το υλικό να διανεμηθεί κατάλληλα στο πεδίο σχεδιασμού. Ένα παράδειγμα μηχανισμού αντιστροφής δύναμης παρουσιάζεται στο παρακάτω σχήμα 2.1:

Έτσι το πρόβλημα σύνθεσης μηχανισμού με χρήση τοπολογικής βελτιστοποίησης μορφώνεται ως εξής:

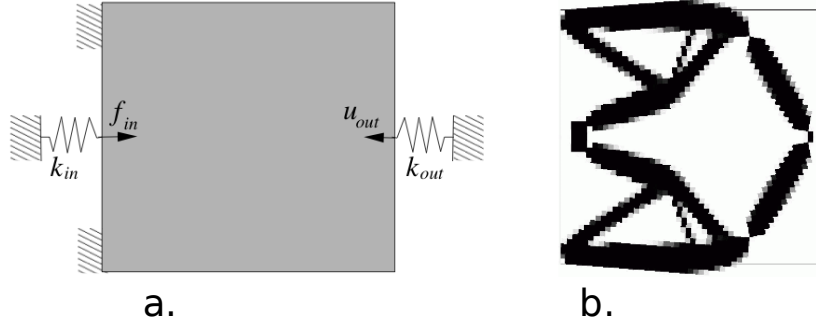
$$\max_{\mathbf{u}, \mathbf{x}_e} u_{out} = \mathbf{1}^T \mathbf{u},$$

υπό τους περιορισμούς:

$$\left(\sum_{e=1}^N x_e^p K_0 \right) \mathbf{u} = \mathbf{f}, \quad (2.2)$$

$$\sum_{e=1}^N v_e x_e \leq \phi V_0,$$

$$0 < x_{min} \leq x_e \leq 1, \quad e = 1, \dots, N,$$



Σχήμα 2.1: Σύνθεση εύκαμπτων μηχανισμών: α) Ορισμός αρχικού πεδίου σχεδιασμού, β) Τελική κατανομή υλικού

όπου $\mathbf{1}^T$ είναι ένα διάνυσμα όπου το εσωτερικό γινόμενο με το διάνυσμα των μετακινήσεων $\mathbf{u}$ δίνει την μετακίνηση στο σημείο εξόδου. Στην ουσία το διάνυσμα $\mathbf{1}$ έχει μηδενικά στοιχεία για όλους τους κόμβους και μονάδα στον κόμβο εξόδου.

2.1.3 Υπολογιστική διαδικασία και ανάλυση ευαισθησίας

Ο αλγόριθμος για την επίλυση του προβλήματος σύνθεσης εύκαμπτου μηχανισμού είναι ακριβώς ο ίδιος και στην περίπτωση εύρεσης της βέλτιστης κατασκευής. Τα βήματα που ακολουθούνται είναι τα ίδια όπως αυτά περιγράφονται στο σχήμα 1.5. Το μόνο που αλλάζει είναι ο υπολογισμός των παραγώγων και το ευρετικό σχήμα ανανέωσης των μεταβλητών σχεδιασμού. Για τον υπολογισμό των παραγώγων θα χρησιμοποιηθεί η μέθοδος του συζυγούς προβλήματος (adjoint problem).

Για την αντικειμενική συνάρτηση $u_{out} = \mathbf{1}^T \mathbf{u}$ χρησιμοποιείται το $\mathbf{u}$ τέτοιο ώστε να ικανοποιεί τον περιορισμό ισορροπίας έτσι ώστε $\mathbf{f} - \mathbf{K}\mathbf{u} = \mathbf{0}$. Για ένα αυθαίρετο διάνυσμα $\boldsymbol{\omega}$ έχουμε:

$$u_{out} = \mathbf{1}^T \mathbf{u} - \boldsymbol{\omega}^T (\mathbf{f} - \mathbf{K}\mathbf{u}) \quad (2.3)$$

Παραγωγίζοντας την παραπάνω σχέση ως προς τις μεταβλητές σχεδιασμού έχουμε:

$$\frac{\partial u_{out}}{\partial x_e} = \mathbf{1}^T \frac{\partial \mathbf{u}}{\partial x_e} - \frac{\partial \boldsymbol{\omega}^T}{\partial x_e} (\mathbf{f} - \mathbf{K}\mathbf{u}) - \boldsymbol{\omega}^T \left(\frac{\partial \mathbf{f}}{\partial x_e} - \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} - \mathbf{K} \frac{\partial \mathbf{u}}{\partial x_e} \right) \quad (2.4)$$

και δεδομένου ότι οι παράγωγοι της δύναμης και του διανυσμάτος $\boldsymbol{\omega}$ οφείλουν να γίνονται μηδέν έχουμε:

$$\frac{\partial u_{out}}{\partial x_e} = (\mathbf{1}^T + \boldsymbol{\omega}^T \mathbf{K}) \frac{\partial \mathbf{u}}{\partial x_e} + \boldsymbol{\omega}^T \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u} \quad (2.5)$$

όπου η $\boldsymbol{\omega}$ είναι η λύση του συζυγούς προβλήματος $\mathbf{K}\boldsymbol{\omega} = -\mathbf{1}$. Λαμβάνοντας υπόψιν την σχέση 1.7 και την παράγωγο αυτής έχουμε:

$$\frac{\partial u_{out}}{\partial x_e} = p x_e^{p-1} \boldsymbol{\omega}^T \mathbf{K}_0 \mathbf{u} \quad (2.6)$$

Το διάνυσμα $\mathbf{1}$ μπορεί να χαρακτηριστεί και ως το μοναδιαίο διάνυσμα φόρτισης το οποίο ασκείται στον κόμβο εξόδου του μηχανισμού. Η λύση του συζυγούς προβλήματος $\mathbf{K}\boldsymbol{\omega} = -\mathbf{1}$

στην ουσία είναι οι μετακινήσεις του σώματος όταν σε αυτό ασκείται μια δύναμη f_{out} στο κόμβο εξόδου, και έχει αντίθετη φορά με την εφαρμόζουσα δύναμη f_{in} στο κόμβο εισόδου και χαρακτηρίζεται από το αρνητικό πρόσημο στο συζυγές πρόβλημα. Γι' αυτό και συχνά η μοντελοποίηση της κίνησης στο σημείο εξόδου γίνεται με το μοντέλο ελατηρίου δυσκαμψίας k_{out} και δύναμης f_{out} .

Το πρόβλημα βελτιστοποίησης όπως διατυπώνεται στην σχέση 2.2 είναι της ίδιας μορφής με αυτό της ελαχιστοποίησης της ευκαμψίας με την έννοια ότι μια απλή αντικειμενική συνάρτηση θα πρέπει να ελαχιστοποιηθεί κάτω από γραμμικό περιορισμό του όγκου. Επομένως η μέθοδος κριτηρίων βελτίστου μπορεί να χρησιμοποιηθεί για την επίλυση ενώ το σχήμα ανανέωσης των μεταβλητών σχεδιασμού πρέπει να αλλάξει αφού οι παράγωγοι μπορούν να παίρνουν θετικές και αρνητικές τιμές. Το ευρετικό σχήμα μπορεί να διατυπωθεί ως:

$$x_e^{k+1} = x_e^k \left(\frac{\max(0, -\frac{\partial u_{out}}{\partial x_e})}{\lambda v_e} \right)^\eta \quad (2.7)$$

όπου η είναι μια σταθερά απόσβεσης στο διάστημα $(0,1)$. Στην περίπτωση ελαχιστοποίησης της ευκαμψίας η σταθερά η επιλέγεται ίση με 0.5, ως η βέλτιστη τιμή για την σύγκλιση του αλγορίθμου, ενώ στην περίπτωση σύνθεσης μηχανισμών επιλέγεται να είναι μικρότερη από 0.5. Σύμφωνα με την βιβλιογραφία για βέλτιστη σύγκλιση του αλγορίθμου η καλύτερη μέθοδος βελτιστοποίησης είναι η μέθοδος των κινούμενων ασύμπτωτων (MMA: Method of Moving Asymptotes) [15].

2.1.4 Σύνθεση πολυμορφικών εύκαμπτων μηχανισμών

Οι μηχανισμοί πολλαπλών εφαρμογών είναι εύκαμπτοι μηχανισμοί που εκτελούν διαφορετικές κινήσεις ανάλογα με την ασκούμενη φόρτιση. Αποτελούνται από ένα και μόνο εύκαμπτο σώμα και στην ουσία μεταδίδουν δύο ή και περισσότερες προδιαγεγραμμένες κινήσεις. Το ελαστικό σώμα θα πρέπει να είναι κατάλληλο ώστε να μπορεί να αντέχει όλες τις επιβαλλόμενες διαφορετικές φορτίσεις. Για την σύνθεση ενός μηχανισμού πολλαπλών εφαρμογών μορφώνονται πολυκριτήρια προβλήματα τοπολογικής βελτιστοποίησης. Η μόρφωση ενός προβλήματος σύνθεσης μηχανισμού πολλαπλών εφαρμογών είναι παρόμοια με την αντίστοιχη διατύπωση προβλήματος σχεδιασμού κατασκευής που υπόκειται σε πολλές φορτίσεις (βλ. σχέσεις 1.27 και 1.28 και δίνεται παρακάτω:

$$\begin{cases} \max_{u_1, x_e} c_1(x_e) &= \mathbf{1}_1^T \mathbf{u}_1, \\ \max_{u_2, x_e} c_2(x_e) &= \mathbf{1}_2^T \mathbf{u}_2, \\ &\vdots \\ \max_{u_m, x_e} c_m(x_e) &= \mathbf{1}_m^T \mathbf{u}_m, \end{cases} \quad (2.8)$$

Παίρνοντας το σταθμισμένο άθροισμα για την μετατροπή ενός πολυκριτήριου προβλήματος

σε μονοκριτήριο έχουμε:

$$\max \sum_{i=1}^m w_i c_i = u_{out}^{(\Sigma)} = \sum_{i=1}^m w_i \mathbf{1}_i^T \mathbf{u}_i$$

υπό τους περιορισμούς:

$$\begin{aligned} \left(\sum_{e=1}^N x_e^p \mathbf{K}_0 \right) \mathbf{u}_i &= \mathbf{f}_i, \quad i = 1, \dots, m \\ \sum_{e=1}^N v_e x_e &\leq \varphi V_0, \\ 0 < x_{min} &\leq x_e \leq 1, \quad e = 1, \dots, N, \\ \sum_{i=1}^m w_i &= 1 \end{aligned} \tag{2.9}$$

όπου $u_{out}^{(\Sigma)}$ το άθροισμα όλων των μετακινήσεων στους κόμβους εξόδου, w_i τα βάρη και $\mathbf{1}_i$ τα μοναδιαία συζυγή διανύσματα δυνάμεων για κάθε περίπτωση φόρτισης. Το συζυγές πρόβλημα στην περίπτωση πολλαπλών φορτίσεων γίνεται:

$$u_{out}^{(\Sigma)} = \sum_{i=1}^m w_i \left(\mathbf{1}_i^T \mathbf{u}_i - \omega_i (\mathbf{f} - \mathbf{K} \mathbf{u}_i) \right) \tag{2.10}$$

οπότε παραγωγίζοντας έχουμε:

$$\frac{\partial u_{out}^{(\Sigma)}}{\partial x_e} = \sum_{i=1}^m w_i \left(\left(\mathbf{1}_i^T + \omega_i \mathbf{K} \right) \frac{\partial \mathbf{u}_i}{\partial x_e} + \omega_i \frac{\partial \mathbf{K}}{\partial x_e} \mathbf{u}_i \right) \tag{2.11}$$

όπου ω_i είναι οι λύσεις για των συζυγή προβλήματα $\mathbf{K} \omega_i = -\mathbf{1}$, $i = 1, \dots, m$. Σε αυτή την περίπτωση οι παράγωγοι μορφώνονται ως:

$$\begin{aligned} \frac{\partial u_{out}^{(\Sigma)}}{\partial x_e} &= \sum_{i=1}^m w_i \left(p x_e^{p-1} \omega_i \mathbf{K}_0 \mathbf{u}_i \right) \\ &= p x_e^{p-1} \sum_{i=1}^m w_i \omega_i \mathbf{K}_0 \mathbf{u}_i \end{aligned} \tag{2.12}$$

Το σχήμα ανανέωσης των μεταβλητών σχεδιασμού είναι παρόμοιο της σχέσης ??,

$$x_e^{k+1} = x_e^k \left(\frac{\max(0, -\frac{\partial u_{out}^{(\Sigma)}}{\partial x_e})}{\lambda v_e} \right)^\eta \tag{2.13}$$

2.2 Τοπολογική βελτιστοποίηση για σύνθεση μηχανισμών στο MATLAB

Οι απαραίτητες αλλαγές στον βασικό κώδικα τοπολογικής βελτιστοποίησης για την μετατροπή του σε αλγόριθμο σύνθεσης μηχανισμών είναι μικρές και εύκολα υλοποιήσιμες [2]². Συγκεκριμένα αρκεί να:

1. Αντικατάσταση της γραμμής 12 με:

```
[U,c]=FE(nelx,nely,x,penal);
```

²σελ: 269-270

και διαγραφή της γραμμής 15

```
c=0;
```

2. Οι εκφράσεις για τις παραγώγους ευαισθησίας εξαρτώνται από την λύση του συζυγούς προβλήματος (adjoint problem). Οι λύσεις (οι μετακινήσεις) αποθηκεύονται στην δεύτερη στήλη του πίνακα μετακινήσεων **U**. Αντικατάσταση των γραμμών 20-22 με τις παρακάτω:

```
Ue1 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
Ue2 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],2);
dc(ely,elx) = penal*x(ely,elx)^(penal-1)*Ue1'*KE*Ue2;
```

3. Γίνεται βελτίωση των κριτηρίων σύγκλισης για τον αλγόριθμο διχοτόμησης που χρησιμοποιείται για την εύρεση του πολλαπλασιαστή Lagrange. Αντικατάσταση των γραμμών 39-40 με:

```
l1 = 0; l2 = 100000; move = 0.1;
while ((l2-l1)/(l2+l1) > 1e-4) && l2>1e-40
```

4. Για την σταθεροποίηση της διαδικασίας σύγκλισης του αλγορίθμου επιλέγεται η τιμή 0.3 για την παράμετρο η , ενώ ταυτόχρονα λαμβάνονται υπόψιν οι θετικές τιμές των παραγώγων ευαισθησίας. Αντικατάσταση της γραμμής 42 με:

```
xnew = max(0.001,max(x-move,min(1.,min(x+move,
x.*(max(1e-10,(-dc./lmid)).^0.3)))));
```

5. Γίνεται επιστροφή του πίνακα μετακινήσεων στο κύριο πρόγραμμα. Αντικατάσταση της γραμμής 66 με:

```
function [U,Uout]=FE(nelx,nely,x,penal)
```

6. Γίνεται προκαθορισμός των πινάκων δύναμης και μετακινήσεων για να ληφθούν υπόψιν η πραγματική και η συζυγής περίπτωση φόρτισης. Αντικατάσταση της γραμμής 69 με:

```
F = sparse(2*(nely+1)*(nelx+1),2);
U = sparse(2*(nely+1)*(nelx+1),2);
```

7. Γίνεται ορισμός των συνοριακών συνθηκών, στηρίξεων και συμμετρίας και των κόμβων εισόδου και εξόδου. Επιπλέον θα πρέπει να οριστεί η δυσκαμψία των ελατηρίων στο σημεία αυτά, ενώ κρίνεται απαραίτητη η επιστροφή τις μετακίνησης στο σημείο εξόδου πίσω στο πρόγραμμα (τιμή της αντικειμενικής συνάρτησης). Αντικατάσταση των γραμμών 78-80 με:

```
% DEFINE LOADS AND SUPPORTS (HALF FORCE INVERTER)
din=1;
dout=2*nelx*(nely+1) + 1;
F(din,1)=1;
F(dout,2)=-1;
K(din,din)=K(din,din) + 0.1;
K(dout,dout)=K(dout,dout) + 0.1;
fixeddofs = union([2:2*(nely+1):2*(nely+1)*(nelx+1)],
[2*(nely+1):-1:2*(nely+1)-3] );
```

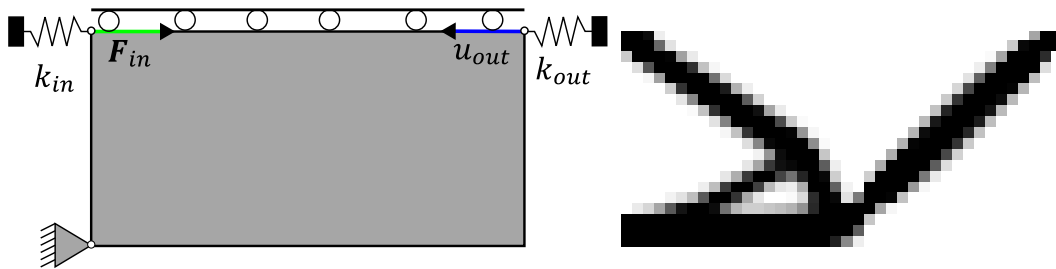
Πρόσθεση της γραμμής:

```
Uout=U(dout,1);
```

αμέσως μετά την γραμμή 85:

```
U(fixeddofs,:)= 0;
```

Η παραπάνω διάταξη περιγράφεται στην παρακάτω εικόνα:



Σχήμα 2.2: Τοπολογική βελτιστοποίηση για αντιστροφέα δύναμης. Αριστερά παρουσιάζεται το μισό χωρίο ορισμού, με τους περιορισμούς συμμετρίας, τις στηρίξεις και φορτίσεις ενώ δεξιά παρουσιάζεται η τελική κατανομή όπως προκύπτει από την εκτέλεση του `topm.m: topm(40,20,0.3,3.0,1.2)`

2.3 Υλοποίηση στο MATLAB: 2 περιπτώσεις φορτίσεων

2.3.1 Αντικειμενική συνάρτηση

Για την μετατροπή του παραπάνω κώδικα για την περίπτωση των δύο φορτίσεων αρκεί να γίνουν μερικές αλλαγές στις γραμμές που αφορούν τους υπολογισμούς των παραγώγων ευαισθησίας και της αντικειμενικής συνάρτησης. Στο σχήμα 2.3 παρουσιάζεται η γραφική μόρφωση του προβλήματος καθώς και μια τελική κατανομή που έχει προκύψει. Ως αντικειμενική συνάρτηση ορίζεται το σταθμισμένο άθροισμα των μετακινήσεων στο σημείο εξόδου για κάθε περίπτωση φόρτισης:

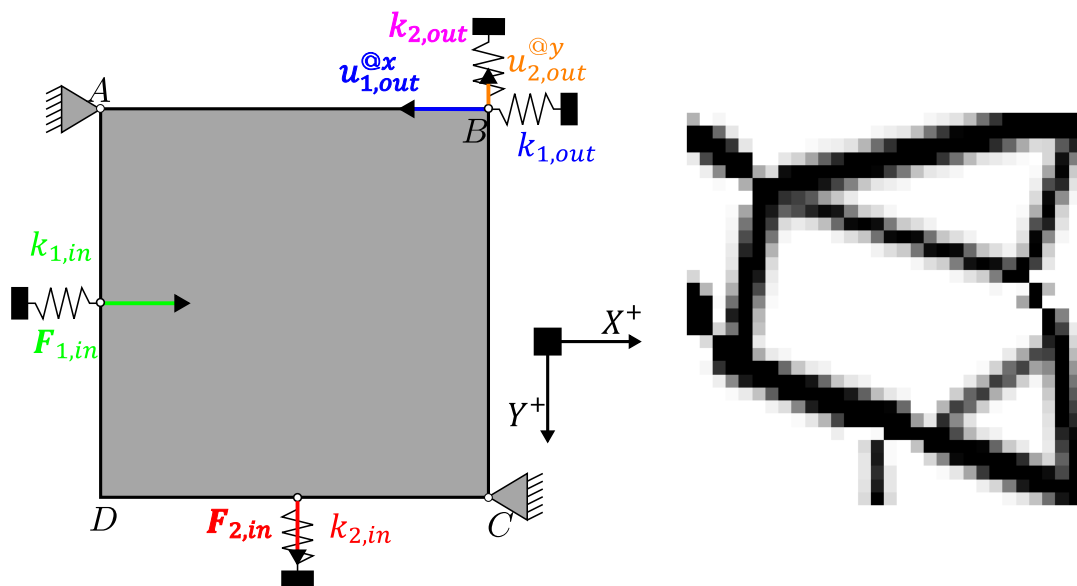
$$c = \sum_{i=1}^2 w_i c_i = \sum_{i=1}^2 w_i \mathbf{1}_i^T \mathbf{u}_i \quad (2.14)$$

Η γραμμή 151 περιγράφει την παραπάνω έκφραση:

```
151 Uout=0.5*abs(Uout1_x) + 0.5*abs(Uout2_y);
```

Οι μεταβλητές `Uout1_x` και `Uout2_y` είναι οι μετακινήσεις στον κόμβο εξόδου για την κάθε περίπτωση φόρτισης ξεχωριστά, ενώ θα πρέπει να ταυτόχρονα να προσαρμοστεί και ο κώδικας που αφορά τις παραγώγους ευαισθησίας (γραμμές 20-31)

```
for ely = 1:nely
    for elx = 1:nelx
        n1 = (nely+1)*(elx-1)+ely;
        n2 = (nely+1)* elx +ely;
        Ue1 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
        Ue2 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],2);
```



Σχήμα 2.3: Τοπολογική βελτιστοποίηση για πολυμορφικό αντιστροφέα δύναμης. Αριστερά η γραφική μάρφωση, και δεξιά μια τυχαία τελική κατανομή. Εκτέλεση σε περιβάλλον MATLAB: `topm2loads(30,30,0.3,3.0,1.2)`

```

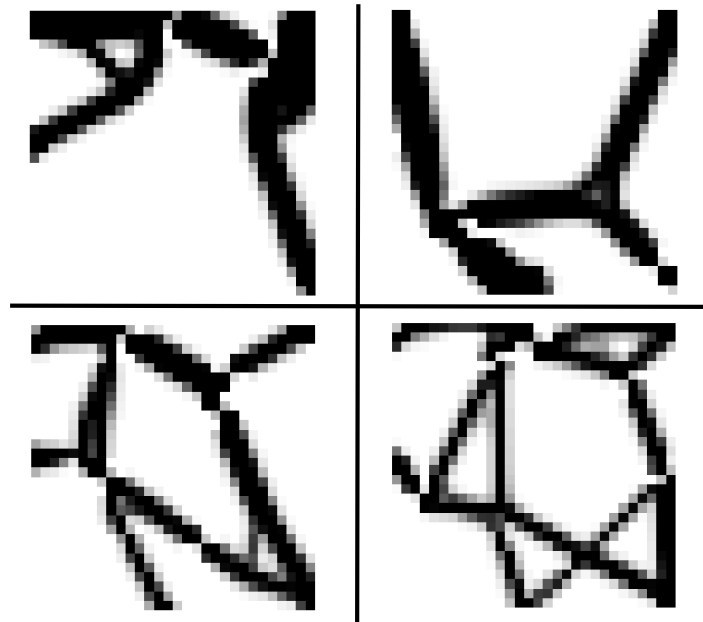
Ue3 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],3);
Ue4 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],4);
dc(ely,elx) = 0.5*penal*x(ely,elx)^(penal-1)*Ue1'*KE*Ue2+...
0.5*penal*x(ely,elx)^(penal-1)*Ue3'*KE*Ue4;
end
end

```

Αλλαγές στον κώδικα όσο αφορά τις συντομογραφίες, φορτίσεις, στηρίξεις, περιορισμούς κίνησης, παρουσιάζονται στις γραμμές 89-121.

2.3.2 Αριθμητικά προβλήματα

Με την χρήση του παραπάνω αλγόριθμου παρατηρήθηκε ότι για διαφορετικές εκτελέσεις του κώδικα, με ακριβώς τις ίδιες παραμέτρους, που ξεκινούσαν από τυχαίες αρχικές κατανομές, οι τελικές κατανομές ήταν πολύ διαφορετικές μεταξύ τους. Επίσης υπάρχουν περιπτώσεις που ο αλγόριθμος κατέληγε σε κατανομές που ήταν κατάλληλες μόνο για την μία ή την άλλη περίπτωση φόρτισης, ενώ υπήρχαν περιπτώσεις που η τελική κατανομή ήταν κατάλληλη και για τις δύο φορτίσεις. Αυτό φαίνεται καθαρά στο παρακάτω σχήμα 2.4. Σε όλες τις περιπτώσεις έγινε εκτέλεση του ίδιου κώδικα με ίδιες παραμέτρους, αλλά με τυχαίες αρχικές κατανομές: `topm_2loads(30,30,0.3,3.0,1.2)`, που μερικές όπως οι δύο πρώτες δεν εξυπηρετούν τον βασικό περιορισμό στο πρόβλημα, δηλαδή την σύνθεση ενός και μόνου εύκαμπτου μηχανισμού που να εξυπηρετεί και τις δύο περιπτώσεις φορτίσεων. Ακόμα και αν βρεθεί μια τέτοια κατανομή δεν γνωρίζουμε αν είναι τοπικό ή ολικό βέλτιστο. Γι' αυτόν τον λόγο κρίνεται απαραίτητη η χρήση στοχαστικών αλγορίθμων ολικής βελτιστοποίησης όπως Εξελικτικοί Αλγόριθμοι ή Αλγόριθμοι Βελτιστοποίησης Σμήνους Σημείων χρησιμοποιώντας τον επαναληπτικό αλγόριθμο τοπικής έρευνας ως εργαλείο αξιολόγησης για την εύρεση ενός ολικού βέλτιστου.



Σχήμα 2.4: Πάνω αριστερά, τελική κατανομή για την μια περίπτωση φόρτισης, πάνω δεξιά τελική κατανομή για την δεύτερη περίπτωση φόρτισης, κάτω αριστερά και δεξιά τελικές κατανομές που ικανοποιούν και τις περιπτώσεις φορτίσεων.

ΜΕΡΟΣ II

ΣΤΟΧΑΣΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΟΛΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

3.1 Εισαγωγή

Στο προηγούμενο κεφάλαιο είχε αναφερθεί ότι ένα από τα προβλήματα των επαναληπτικών διαδικασιών επίλυσης προβλημάτων τοπολογικής βελτιστοποίησης είναι η ύπαρξη πολλών λύσεων, πολλών τοπικών ακρότατων, στις περιπτώσεις που ο αλγόριθμος ξεκινά από διαφορετική αρχική κατάσταση. Αυτό το πρόβλημα έχει παρατηρηθεί τόσο για προβλήματα εύρεσης βέλτιστης κατασκευής, αλλά και πολύ περισσότερο σε προβλήματα σύνθεσης μηχανισμών. Για την εύρεση ενός γενικού ακρότατου προτείνεται η χρήση τεχνικών γενικής βελτιστοποίησης. Σε κάθε βήμα των παραπάνω αλγόριθμων το εργαλείο αξιολόγησης θα βασίζεται στη χρήση ενός επαναληπτικού αλγορίθμου τοπικής έρευνας που θα λύνει ένα πρόβλημα τοπολογικής βελτιστοποίησης. Στο παρόν κεφάλαιο θα περιγράφουν δύο βασικοί αλγόριθμοι γενικής βελτιστοποίησης: ο Διαφορικός Εξελικτικός Αλγόριθμος (Differential Evolution, ή DEA) από την οικογένεια των εξελικτικών αλγορίθμων, και ο κλασσικός αλγόριθμος βελτιστοποίησης με προσομοίωση κίνησης σμήνους σωματιδίων (Particles Swarm Optimization ή PSO).

3.2 Εξελικτικοί Αλγόριθμοι

3.2.1 Ιστορική αναδρομή

Ως εξέλιξη, νοούνται όλες οι αλλαγές στα χαρακτηριστικά των έμβιων οργανισμών στο πέρασμα των χρόνων, συμπεριλαμβανομένης και της εμφάνισης και νέων μορφών ζωής. Την δεκαετία του 40, η εξέλιξη ορίστηκε λεπτομερέστερα ως οι αλλαγές στην ακολουθία των αλληλόμορφων (διαφορετικές μορφές ενός γονιδίου) σε ένα πληθυσμό, από την τρέχουσα γενιά, μέχρι την επόμενη [16].

Η βιολογική εξέλιξη αναφέρεται σε όλες τις αλλαγές που παρουσιάζονται σε έναν πληθυσμό οργανισμών στη διάρκεια του χρόνου. Οι αλλαγές αυτές λαμβάνουν χώρα στους οργανισμούς σε γενετικό επίπεδο, καθώς τα γονίδια αυτών υπόκεινται σε μετάλλαξη ή/ και επιχiasμό κατά τη διάρκεια της αναπαραγωγικής διαδικασίας, οπότε μεταβιβάζονται στις επόμενες γενιές [16].

Κατά την διάρκεια της εξέλιξης, τα άτομα αυτά κληρονομούν νέα χαρακτηριστικά από τους γονείς τους, τα οποία αυξάνουν τις πιθανότητες επιβίωσης και αναπαραγωγής από τα υπόλοιπα ομοειδή άτομα του περιβάλλοντος τους. Τα χαρακτηριστικά που ενεργούν θετικά, και συνεπώς αυξητικά στην πιθανότητα επιβίωσης των ατόμων, τείνουν να κατέχουν μεγα-

λύτερη συχνότητα εμφάνισης στον πληθυσμό, ενώ αυτά που επιδρούν αρνητικά ή κατέχουν συγκριτικά χειρότερη θέση, τείνουν να ελαττωθούν και να εξαφανιστούν. Αυτή η διαδικασία της διαφοροποίησης της επιβίωσης και της αναπαραγωγής των ατόμων είναι γνωστή ως *φυσική επιλογή* [17].

Ο Κάρολος Δαρβίνος (1809-1882) ήταν ο πρώτος που διατύπωσε την θεωρία της εξέλιξης μέσω της φυσικής επιλογής και που στήριξε την θεωρία της οργανικής εξέλιξης με επαρκή αποδεικτικά στοιχεία και καθόρισε με ποιον τρόπο η διεργασία της φυσικής επιλογής επιφέρει την προσαρμογής των ατόμων στο περιβάλλον τους, αλλά απέφυγε για χρόνια να δημοσιεύσει τις ανακαλύψεις του. Η θεωρία της φυσικής επιλογής του Δαρβίνου υποστηρίζει ότι η ποικιλομορφία στους οργανισμούς εμφανίζεται τυχαία ενώ η επιβίωση ή η εξαφάνιση κάθε οργανισμού καθορίζεται από την ικανότητα του να προσαρμοστεί στο περιβάλλον του.

Εμπνευσμένοι από το έργο και τα αποτελέσματα των θεωριών του Δαρβίνου, πολλοί επιστήμονες και μηχανικοί εφαρμόζουν τις αρχές της φυσικής εξέλιξης στην έρευνα, με αποτέλεσμα την εμφάνιση ενός νέου είδους αλγορίθμων πολύ χρήσιμων στην επίλυση απαιτητικών και δύσκολων προβλημάτων. Ένα παράδειγμα χρήσης της θεωρίας της εξέλιξης, για την επίλυση τέτοιων προβλημάτων είναι οι Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms, EA). Αυτοί οι αλγόριθμοι είναι χρήσιμοι σε περιπτώσεις που ο χρήστης είναι σε θέση να καθορίσει μία καλύτερη λύση μεταξύ άλλων, αλλά η εύρεση της βέλτιστης είναι δύσκολη, αδύνατη ή απαιτεί πολύ χρόνο. Σε αυτές τις περιπτώσεις ένας EA παράγει συνεχώς καλύτερες λύσεις μέχρι να επιτευχθεί μία, η οποία να είναι αρκετά ικανοποιητική σε ότι αφορά τις απαιτήσεις που έχουν τεθεί, έστω κι αν αυτή δεν είναι η βέλτιστη όλων.

Με τον όρο Εξελικτικοί Αλγόριθμοι περιγράφονται οποιοιδήποτε ευρετικοί αλγόριθμοι βελτιστοποίησης, που χρησιμοποιούν μηχανισμούς και τεχνικές δανεισμένες από την βιολογική εξέλιξη, όπως η *επιλογή* (crossover), η *μετάλλαξη* (mutation), και ο *επιχiasμός* (crossover). Οι υποψήφιες λύσεις του προβλήματος προς επίλυση είναι τα άτομα σε ένα πληθυσμό και η αξία τους περιγράφεται από την *συνάρτηση καταλληλότητας* (fitness function). Η Εξέλιξη του πληθυσμού πραγματοποιείται μετά από μια επαναλαμβανόμενη εφαρμογή των προαναφερθέντων τελεστών [18]. Η διαδικασία εφαρμογής των τελεστών στα άτομα του πληθυσμού είναι ο παράγοντας που διαφοροποιεί τους EA μεταξύ τους, δημιουργώντας αρκετές παραλλαγές.

Οι EA είναι ευρέως διαδεδομένοι, γιατί είναι πολύ εύκολοι στην υλοποίηση και στην χρήση, ενώ βρίσκουν εφαρμογή σε ένα ευρύ πεδίο διαφορετικών προβλημάτων, στην επιστήμη και την μηχανική. Η συνεχής έρευνα στο συγκεκριμένο αντικείμενο οδήγησε τόσο στην ανακάλυψη κάποιων αδυναμιών τους, όπως η γρήγορη σύγκλιση, αλλά και σε βελτιώσεις αυτών, με στόχο την υπέρβαση εμποδίων που μπορεί να αντιμετωπίσουν. Οι χρήστες των EA είναι πολύ δύσκολο να έχουν πλήρη εποπτεία των βέλτιστων επιλογών των παραμέτρων του EA, αφού η επίδοσή τους εξαρτάται σε μεγάλο βαθμό από το συγκεκριμένο πρόβλημα προς επίλυση, και τις εκάστοτε παραμέτρους [18].

Οι EA εμφανίζονται σε τρεις διαφορετικές μορφές, οι οποίες ακολουθούν διακριτή πορεία, αλλά με ισχυρές αλληλεπιδράσεις μεταξύ τους:

- του Γενετικού Αλγόριθμου (Genetic Algorithms - GAs),
- τον Εξελικτικό Προγραμματισμό (Evolutionary Programming - EP) και
- τις Εξελικτικές Στρατηγικές (Evolution Strategies - ESs).

Ως παρακλάδι των Γενετικών Αλγορίθμων εξελίχθηκε πρόσφατα ο Γενετικός Προγραμματισμός (Genetic Programming - GP).

Μέσα από την προσπάθεια για βελτίωση των Εξελικτικών Αλγορίθμων δημιουργήθηκαν οι Διαφορικοί Εξελικτικοί Αλγόριθμοι (Differential Evolution, DE - Δ.Ε.) [14],[8], οι οποίοι αποτελούν μέρος της ευρύτερης ομάδας των Εξελικτικών Αλγορίθμων, αλλά μπορούν να

λειτουργούν πιο αξιόπιστα, χωρίς να παρουσιάζουν ιδιαίτερες αδυναμίες σε εμπόδια, στα οποία οι άλλοι αλγόριθμοι αντιμετωπίζουν δυσκολίες.

3.2.2 Χαρακτηριστικά Εξελικτικών αλγορίθμων

Οι ΕΑ παρουσιάζουν σαφή πλεονεκτήματα σε σχέση με τις κλασσικές μεθόδους βελτιστοποίησης, διότι δεν απαιτούν κυρτότητα, κοιλότητα συνέχεια ή την ύπαρξης παραγώγων της εκάστοτε αντικειμενικής συνάρτησης η οποία να μην είναι δυνατόν να υπολογιστεί αναλυτικά. Ένας ΕΑ διαφέρει από τις κλασσικές μεθόδους βελτιστοποίησης σε αρκετά σημεία [13]:

- **Στοχαστικότητα:** Αρχικά βασίζεται στην τυχαία δειγματοληψία. Αυτό καθιστά τον ΕΑ μια μη ντετερμινιστική μέθοδο, η οποία μπορεί να καταλήξει σε συνήθως αρκετά κοντινές αλλά διαφορετικές λύσεις για επαναλαμβανόμενες εκτελέσεις του αλγορίθμου για το ίδιο πρόβλημα. Αντιθέτως, οι ντετερμινιστικές μέθοδοι καταλήγουν πάντα στην ίδια λύση.
- **Πληθυσμός:** Αντίθετα από τις κλασσικές μεθόδους βελτιστοποίησης που διατηρούν μία μόνο βέλτιστη λύση που έχει βρεθεί μέχρι την τρέχουσα χρονική στιγμή, σε ένα ΕΑ διατηρείται ένα σύνολο από υποψήφιες λύσεις που σχηματίζουν τον πληθυσμό. Οι πολλές διαφορετικές λύσεις μπορούν να συνδυαστούν μεταξύ τους με την βοήθεια γενετικών τελεστών ώστε να σχηματιστούν επιπλέον λύσεις, και επομένως να επιτευχθεί η εξέλιξη.
- **Μετάλλαξη:** Ο τελεστής αυτός προέρχεται από την γενετική. Η μετάλλαξη πραγματοποιεί τυχαίες αλλαγές ή μεταλλάξεις στο γενετικό υλικό (DNA) σε ένα ή περισσότερα μέλη του πληθυσμού, διαφοροποιώντας τον πληθυσμό. Στους ΕΑ η μετάλλαξη επιτυγχάνει ακριβώς την ίδια δουλειά, μεταλλάσσοντας τυχαία μία η περισσότερες υποψήφιες λύσεις, οι οποίες μπορεί να είναι είτε καλύτερες είτε χειρότερες από τις υπάρχουσες. Υπάρχουν πολλοί διαφορετικοί τρόποι για την πραγματοποίηση μιας μετάλλαξης.
- **Επιχiasμός:** Στην γενετική ο τελεστής αυτός διασταυρώνει το γενετικό υλικό δύο οργανισμών και την δημιουργία νέων, στους ΕΑ ο επιχiasμός επιχειρεί να συνδυάσει στοιχεία από τις υπάρχουσες λύσεις, με στόχο την δημιουργία νέες με κάποια χαρακτηριστικά από τις υπάρχουσες λύσεις που επιχιάστικαν. Στην βιβλιογραφία υπάρχουν αρκετές μέθοδοι για την επίτευξη του επιχιασμού.
- **Επιλογή:** Ένας ΕΑ πραγματοποιεί μια διαδικασία επιλογής, κατά την οποία τα πιο αποδοτικά μέλη του πληθυσμού επιβιώνουν, ενώ τα λιγότερο αποδοτικά μέλη δεν επιβιώνουν. Η διαδικασία της επιλογής οδηγεί τον αλγόριθμο σε καλύτερες λύσεις.

3.2.3 Περιγραφή ενός εξελικτικού αλγορίθμου

Κατά την εκκίνηση του αλγορίθμου, πραγματοποιείται μια αρχικοποίηση του πληθυσμού. Στην ουσία παράγονται υποψήφιες λύσεις μέσα στον χώρο λύσεων. Κάθε λύση αξιολογείται ώστε να υπάρχει ένα μέτρο σύγκρισης μεταξύ των λύσεων. Στην συνέχεια αρχίζει μια επαναληπτική διαδικασία, κατά την οποία εφαρμόζονται οι τελεστές επιχιασμού και μετάλλαξης στον πληθυσμό, παράγοντας ένα νέο ενδιαμέσο πληθυσμό. Έπειτα, γίνεται μια αξιολόγηση κάθε νέας υποψήφιας λύσης. Στην συνέχεια επιδρά ο τελεστής της επιλογής, όπου με κάποιο τρόπο επιλέγονται οι καλύτερες λύσεις από τον παλιό πληθυσμό (γονείς) και τον ενδιαμέσο (παιδιά) για την συγκρότηση του νέου πληθυσμού. Στον νέο πληθυσμό υπάρχει η πιθανότητα να υπάρχουν όχι μόνο οι καλύτερες λύσεις, αλλά και μερικές κακές. Αυτό συνήθως γίνεται για να υπάρχει όσο το δυνατόν μεγαλύτερη ποικιλομορφία στον πληθυσμό, ούτως ώστε να μην υπάρχει περίπτωση ο ΕΑ να συγκλίνει γρήγορα και να οδηγηθεί σε κάποιο τοπικό βέλτιστο. Η ποικιλομορφία του πληθυσμού δεν θα πρέπει να είναι υπερβολική γιατί θα οδηγήσει τον

αλγόριθμο σε αργή σύγκλιση. Η επαναληπτική διαδικασία τερματίζει όταν ικανοποιηθούν τα κριτήρια σύγκλισης. Παρακάτω παρουσιάζεται σε μορφή ψευδοκώδικα η δομή ενός EA.

START

```
t=0, Random Initialization of 1st Population_1,
Evaluation of population_1
Keep the best member of population_1
Do while criteria are not satisfied
    t=t+1
    Crossover
    Mutation
    Evaluation
    Selection and assembly of population_t
    Keep the best so far
```

EndDo

END

3.3 Διαφορικοί Εξελικτικοί Αλγόριθμοι - DEA

3.3.1 Εισαγωγή στους DEA

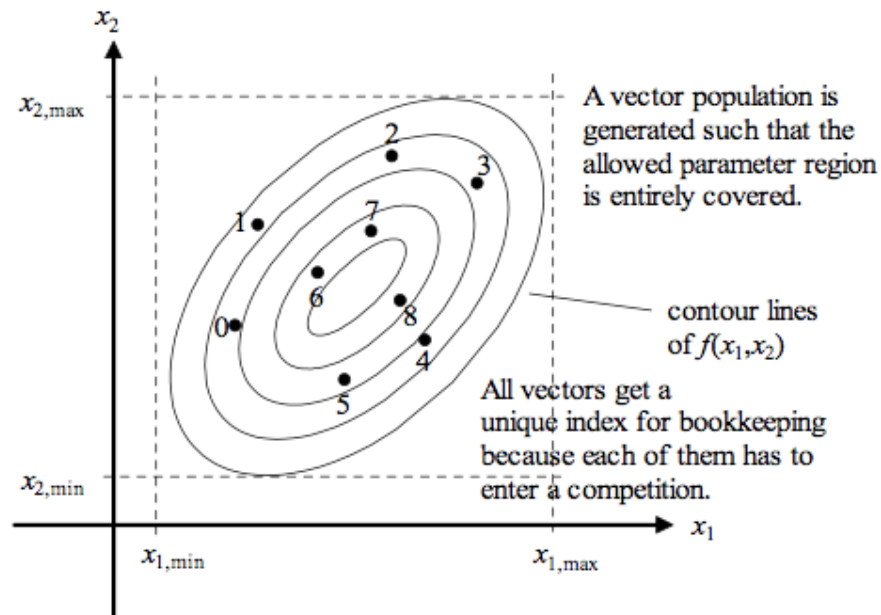
Οι Διαφορικοί Εξελικτικοί Αλγόριθμοι (DEA) αναπτύχθηκαν από τους Price και Storn το 1995 ως αλγόριθμοι για την λύση του πολυωνυμικού προβλήματος Chebychev [14]. Στην συνέχεια εξελίχθηκαν και έφτασαν στην σημερινή μορφή τους ως αλγόριθμοι βελτιστοποίησης. Ο DEA έχει τα όλα τα βασικά χαρακτηριστικά ενός Εξελικτικού Αλγορίθμου, αφού στην ουσία είναι ένας από αυτούς. Οι κυριότερες αλλαγές σε σχέση με τους EA είναι:

1. Στους DEA πρώτα εφαρμόζεται ο τελεστής της μετάλλαξης και στην συνέχεια ο τελεστής του επιχιασμού, σε αντίθεση με τους EA όπου η σειρά εκτέλεσης των τελεστών είναι αντίστροφη,
2. Η μετάλλαξη δεν γίνεται από μια ήδη γνωστή κατανομή πιθανότητας αλλά από τις διαφορές μεταξύ των λύσεων στον τρέχοντα πληθυσμό. Εξού και το όνομα *διαφορικός*.

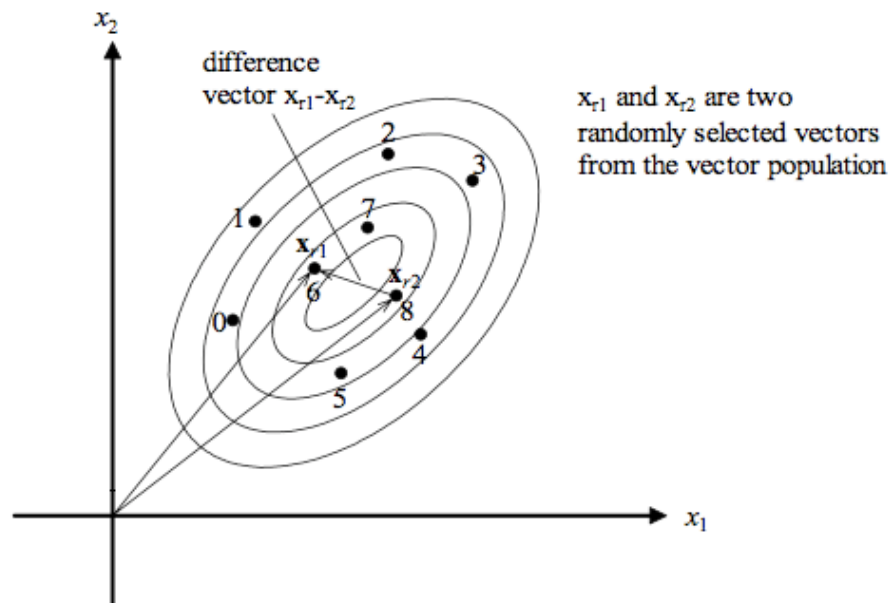
3.3.2 Περιγραφή DEA

Όπως και στους υπόλοιπους EA, οι DEA είναι αλγόριθμοι βασιζόμενοι στην εξέλιξη ενός πληθυσμού. Προκαθορισμένα όρια στις μεταβλητές σχεδιασμού ορίζουν τον χώρο των υποψήφιων λύσεων που αναπαριστώνται από N_p διάνυσμα στον χώρο $\mathbb{C}$ υποσύνολο του $\mathbb{R}^d$, όπου d είναι οι μεταβλητές σχεδιασμού. Ο αλγόριθμος περιγράφεται από τα παρακάτω βήματα:

1. Στην αρχή παράγονται τυχαίες λύσεις μέσα στο χώρο $\mathbb{C}$ οι οποίες αξιολογούνται,
2. Για να διαφοροποιηθεί ο υπάρχων πληθυσμός παράγονται διαταράξεις ως διαφορές δύο τυχαίων διανυσμάτων μέσα από τον πληθυσμό,
3. Τα νέα άτομα $\mathbf{u}_i$, $i = 0, \dots, N_p - 1$ προκύπτουν από την πρόσθεση μιας τυχαία κλιμακοποιημένης διαφοράς σε ένα τυχαίο τρίτο διάνυσμα του πληθυσμού,
4. Για την επιλογή τα νέο άτομο $\mathbf{u}_i$ συγκρίνονται με αυτό με ίδιο δείκτη του υπάρχοντα πληθυσμού $\mathbf{x}_i$, και το αντικαθιστά αν είναι καλύτερο, αλλιώς το παλιό παραμένει στον πληθυσμό.



Σχήμα 3.1: Αρχικοποίηση πληθυσμού στον DEA, [8]

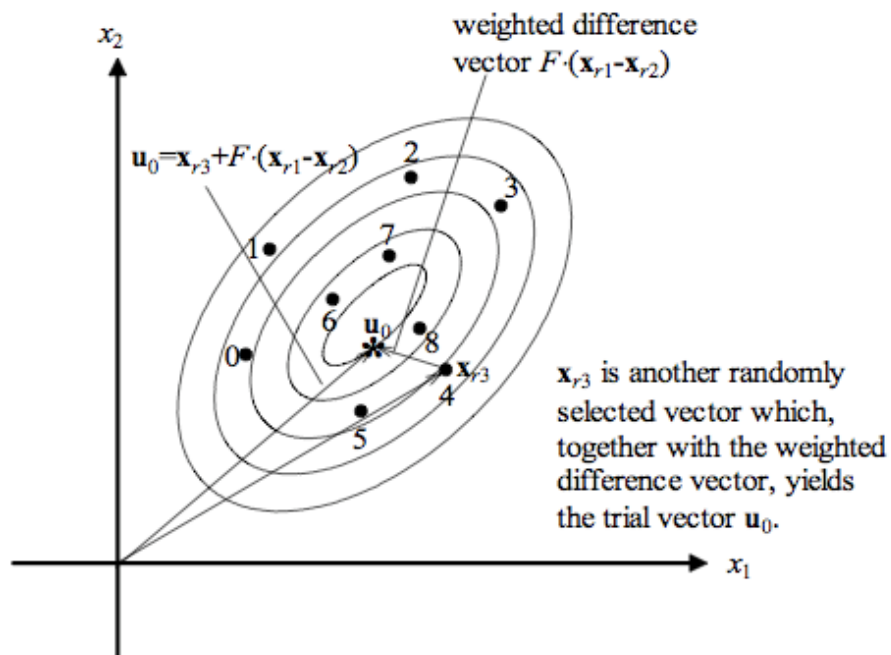


Σχήμα 3.2: Υπολογισμός διαταράξεων στον πληθυσμό, [8]

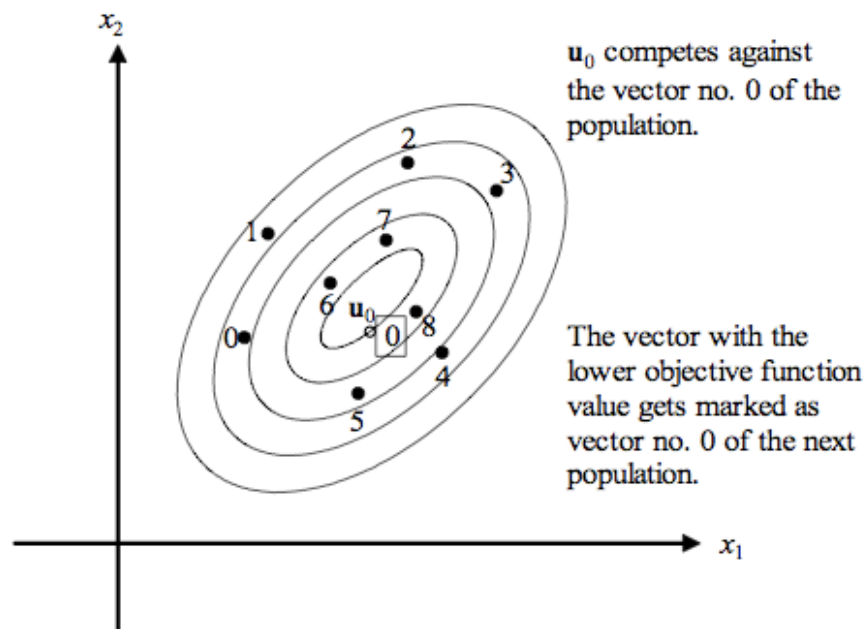
Τα παραπάνω βήματα περιγράφονται στα παρακάτω σχήματα 3.1 έως 3.4. Τα σχήματα 3.5 και 3.6 παρουσιάζουν την παραπάνω διαδικασία για όλα τα άτομα του πληθυσμού N_p . Μόλις ολοκληρωθούν όλες οι συγκρίσεις τότε τα άτομα του νέου πληθυσμού που προκύπτει θα αποτελέσουν τους γονείς για το επόμενη γενιά.

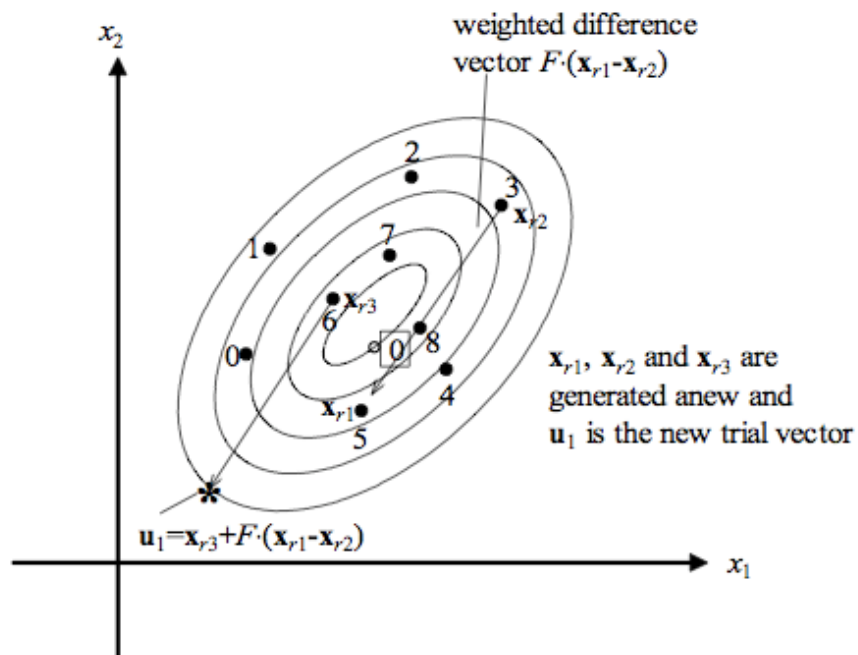
3.3.3 Η δομή των πληθυσμών στον DEA

Ο διαφορικός εξελικτικός αλγόριθμος διατηρεί δύο πληθυσμούς ατόμων που περιέχουν N_p d-διάστατα διανύσματα από πραγματικές τιμές. Ο τρέχον πληθυσμός, συμβολίζεται P_x , και αποτελείται από τα διανύσματα x_{ig} , τα οποία έχουν βρεθεί αποδεκτά είτε ως μέλη του

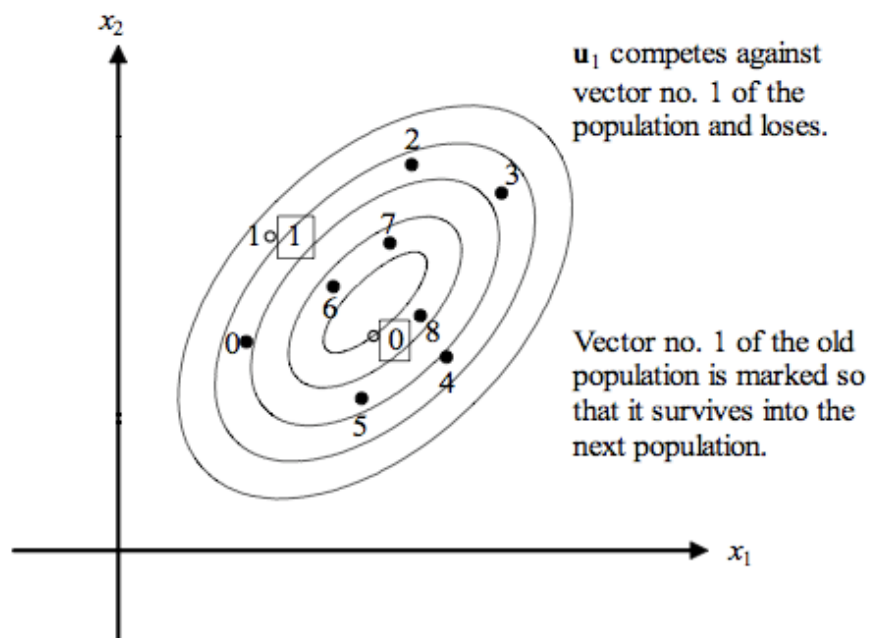


Σχήμα 3.3: Μετάλλαξη, [8]

Σχήμα 3.4: Επιλογή. Το διάνυσμα u_0 αντικαθιστά το διάνυσμα με δείκτη 0 γιατί έχει καλύτερη τιμής αντικειμενικής συνάρτησης, [8].



Σχήμα 3.5: Μετάλλαξη ενός άλλου διανύσματος του τρέχοντα πληθυσμού, [8].



Σχήμα 3.6: Συγκρίση μεταξύ u_1 και x_1 . Στην περίπτωση το νέο διάνυσμα είναι χειρότερο από το υπάρχον, [8].

αρχικού πληθυσμού είτε σε σύγκριση με τα άτομα του προηγούμενου:

$$\begin{aligned} P_{x,g} &= \{x_{i,g}\}, \quad i = 0, 1, \dots, N_p - 1, \quad g = 0, 1, \dots, g_{max}, \\ x_{i,g} &= \{x_{j,i,g}\}, \quad j = 0, 1, \dots, d - 1. \end{aligned} \quad (3.1)$$

Ο δείκτης g δείχνει την τρέχουσα γενιά ή πληθυσμό στον οποίο ανήκει ένα διάνυσμα. Επιπλέον κάθε διάνυσμα χαρακτηρίζεται μέσα στον πληθυσμό με τον δείκτη i , $i = 0, 1, \dots, N_p - 1$. Οι παράμετροι σχεδιασμού μέσα στο διάνυσμα χαρακτηρίζονται από τον δείκτη j , $j = 0, 1, \dots, d - 1$.

Μετά την αρχικοποίηση, ενεργοποιείται ο τελεστής της μετάλλαξης όπου τυχαία διανύσματα από τον πληθυσμό μεταλλάσσονται για την σύνθεση ενός ενδιάμεσου πληθυσμού $P_{v,g}$ από N_p μεταλλαγμένα διανύσματα v_g :

$$\begin{aligned} P_{v,g} &= \{v_{i,g}\}, \quad i = 0, 1, \dots, N_p - 1, \quad g = 0, 1, \dots, g_{max}, \\ v_{i,g} &= \{v_{j,i,g}\}, \quad j = 0, 1, \dots, d - 1. \end{aligned} \quad (3.2)$$

Κάθε διάνυσμα του ενδιάμεσου πληθυσμού διασταυρώνεται με ένα τυχαίο από τον τρέχοντα πληθυσμό, για την σύνθεση του υπό δοκιμή πληθυσμού P_u , που αποτελείται από N_p υπό δοκιμή διανύσματα $u_{i,g}$:

$$\begin{aligned} P_{v,g} &= \{v_{i,g}\}, \quad i = 0, 1, \dots, N_p - 1, \quad g = 0, 1, \dots, g_{max}, \\ v_{i,g} &= \{v_{j,i,g}\}, \quad j = 0, 1, \dots, d - 1. \end{aligned} \quad (3.3)$$

3.3.4 Αρχικοποίηση πληθυσμού

Πριν την αρχικοποίηση του πληθυσμού θα πρέπει να οριστούν τα ανώτερα και κατώτερα όρια για τις μεταβλητές σχεδιασμού. Οι ανώτατες και κατώτατες τιμές για τις μεταβλητές σχεδιασμού περιγράφονται από τα διανύσματα διάστασης d , b_U και b_L . Στην συνέχεια εφαρμόζεται η αρχικοποίηση με την χρήση μιας οποιαδήποτε γεννήτρια τυχαίων αριθμών για την παραγωγή αριθμών μεταξύ των συγκεκριμένων διαστημάτων ανά μεταβλητή σχεδιασμού. Για παράδειγμα η $j^{στη}$ παράμετρος του $i^{στη}$ διανύσματος του αρχικού πληθυσμού P_0 είναι:

$$x_{i,j,0} = rand_j(0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L} \quad (3.4)$$

Ο τελεστής $rand_j(0,1)$ επιστρέφει ένα τυχαίο αριθμό από μια ομοιόμορφη κατανομή στο διάστημα $[0,1)$ δηλαδή $0 \leq rand_j(0,1) < 1$.

3.3.5 Η μετάλλαξη στον DEA

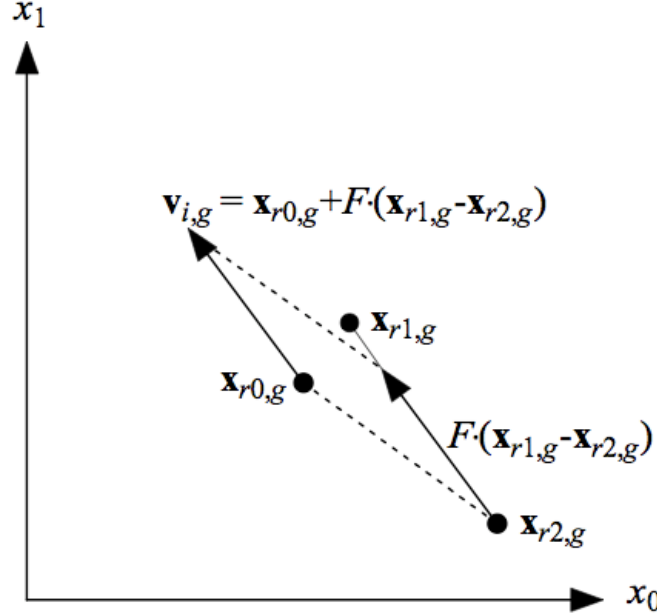
Μετά την αρχικοποίηση, επιδρά ο τελεστής της μετάλλαξης για την σύνθεση του ενδιάμεσου πληθυσμού P_v . Η σύνθεση γίνεται με την πρόσθεση μια κλιμακοποιημένης διαφοράς δύο τυχαίων διανυσμάτων που έχουν επιλεγεί από τον τρέχοντα πληθυσμό σε ένα τρίτο τυχαίο διάνυσμα από τον ίδιο πληθυσμό. Το μεταλλαγμένο διάνυσμα $v_{i,g}$ προκύπτει από την παρακάτω σχέση:

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}) \quad (3.5)$$

Ο συντελεστής $F \in (0,1+)$ είναι ένας θετικός πραγματικός αριθμός που στην ουσία ελέγχει την διατάραξη του πληθυσμού. Αν και δεν υπάρχει ανώτατο όριο στην τιμή του F , σπάνια χρησιμοποιούνται τιμές μεγαλύτερες από 1.0.

Το βασικό διάνυσμα με δείκτη $r0$ μπορεί να οριστεί με αρκετούς τρόπους, αλλά προς το παρόν θεωρείται ότι επιλέγεται τυχαία από τον τρέχοντα πληθυσμό αρκεί ο δείκτης του να είναι διαφορετικός από τον δείκτη i , δηλαδή, $r0 \neq i$. Για να υπάρχει πλήρης διαφοροποίηση

του ενδιάμεσου πληθυσμού P_v , τα διανύσματα με δείκτες $r1$, και $r2$, επιλέγονται τυχαία κάθε φορά που εκτελείται ο τελεστής της μετάλλαξης για κάθε ένα διάνυσμα του τρέχοντα πληθυσμού. Πιο λεπτομερής απεικόνιση του τελεστή της μετάλλαξης παρουσιάζεται στο Σχήμα 3.7 για ένα δυσδιάστατο πρόβλημα:



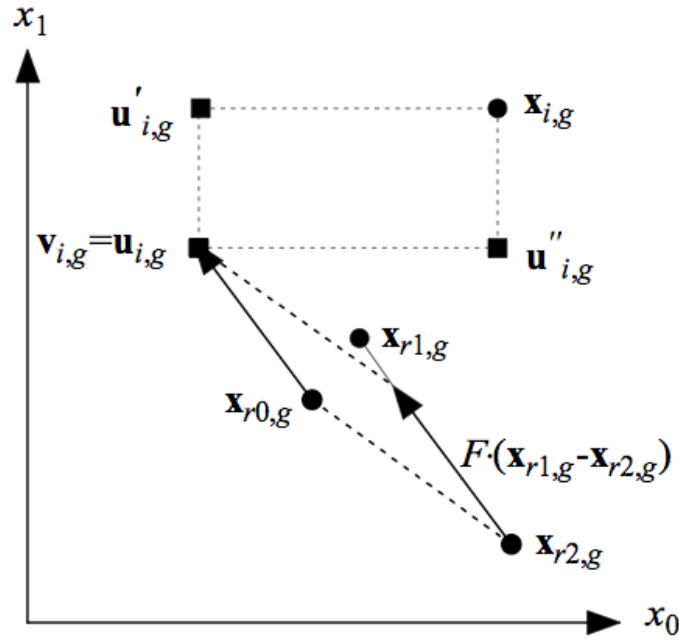
Σχήμα 3.7: Ο τελεστής της διαφορικής μετάλλαξης: η κλιμακοποιημένη διαφορά $F \cdot (x_{r1,g} - x_{r2,g})$ προστίθεται στο βασικό διάνυσμα $x_{r0,g}$ για την παραγωγή του μεταλλαγμένου διανύσματος $v_{i,g}$, [8].

3.3.6 Ο τελεστής του επιχιασμού στον DEA

Μετά την εφαρμογή της μετάλλαξης, γίνεται εφαρμογή του τελεστή επιχιασμού για την σύνθεση του υπό δοκιμή πληθυσμού P_u . Συγκεκριμένα επιχιάζονται τα διανύσματα από τον τρέχοντα πληθυσμό P_x και από τον ενδιάμεσο πληθυσμό P_v που έχει συντεθεί με την μετάλλαξη:

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{Αν } (rand_j(0,1) \leq C_r \text{ ή } j = j_{rand}), \\ x_{j,i,g} & \text{Αλλιώς.} \end{cases} \quad (3.6)$$

Η πιθανότητα επιχιασμού $C_r \in [0, 1]$, είναι μια μεταβλητή ορισμένη από τον χρήστη που ελέγχει τον αριθμό των παραμέτρων (σχεδιασμού) που θα αντιγραφούν από το μεταλλαγμένο διάνυσμα σε αυτό του τρέχοντα πληθυσμού. Η παράμετρος αυτή ορίζεται εξ' αρχής από τον χρήστη. Για να καθοριστεί ποιες παράμετροι θα συνδυαστούν, παράγεται με ομοιόμορφη κατανομή μια πιθανότητα $rand_j$, $0 \leq rand_j \leq 1$. Αν η πιθανότητα είναι μικρότερη ή ίση από C_r , τότε η παράμετρος j για το υπό δοκιμή διάνυσμα κληρονομείται από την j μεταβλητή του μεταλλαγμένου διανύσματος $v_{i,g}$ αλλιώς από το $x_{i,g}$. Επιπλέον η υπό δοκιμή παράμετρος με τυχαίο δείκτη j_{rand} αντιγράφεται από το μεταλλαγμένο διάνυσμα για να διασφαλιστεί ότι το υπό δοκιμή διάνυσμα δεν είναι αντίγραφο του $x_{i,g}$. Στο σχήμα 3.8 παρουσιάζονται όλα τα πιθανά υπό δοκιμή διανύσματα που μπορούν να προκύψουν από τον επιχιασμό ενός μεταλλαγμένου διανύσματος $v_{i,g}$ και του διανύσματος $x_{i,g}$.



Σχήμα 3.8: Τα πιθανά επιπλέον υπό δοκιμή διανύσματα $x'_{i,g}, x''_{i,g}$, όταν τα διανύσματα $v_{i,g}$ και $x_{i,g}$ διασταυρώνονται ομοιόμορφα.[8].

3.3.7 Ο τελεστής της επιλογής στον DEA

Στην περίπτωση που το υπό δοκιμή διάνυσμα $u_{i,g}$ έχει καλύτερη αντικειμενική συνάρτηση από το αντίστοιχο του στον τρέχοντα πληθυσμό $x_{i,g}$, τότε το αντικαθιστά στον επόμενο $(g + 1)$ πληθυσμό, αλλιώς το $x_{i,g}$ παραμένει για τουλάχιστον μια ακόμα γενιά.

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{Αν } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{Αλλιώς.} \end{cases} \quad (3.7)$$

Αφού ο νέος πληθυσμός έχει σχηματιστεί, γίνεται επανάληψη των τελεστών μετάλλαξης, επιχιασμού και επιλογής, μέχρι να βρεθεί το βέλτιστο ή μέχρι να ικανοποιηθούν τα κριτήρια τερματισμού, όπως για παράδειγμα, συμπληρωθεί ο μέγιστος αριθμός γενεών g_{max} ή η βελτίωση της αντικειμενικής συνάρτησης είναι μικρότερη από ένα δεδομένο μικρό αριθμό ε ώστε $f(x_{g+1}^*) - f(x_g^*) \leq \varepsilon$.

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕ ΧΡΗΣΗ ΝΟΗΜΟΣΥΝΗΣ ΣΜΗΝΟΥΣ ΣΩΜΑΤΙΔΙΩΝ (PSO)

4.1 Εισαγωγή

Η βελτιστοποίηση με την χρήση νοημοσύνης σμήνους σωματιδίων (Particles Swarm Optimization: PSO) είναι μια στοχαστική τεχνική βελτιστοποίησης που βασίζεται στην ανάπτυξη και εξέλιξη ενός πληθυσμού. Η εξέλιξη αυτή βασίζεται στην κοινωνική συμπεριφορά σμηνών και κοπαδιών, από έντομα, πουλιά και ψάρια. Αναπτύχθηκε από τους Eberhart και Kennedy το 1995, [5] [7], με κύριο σκοπό την επίλυση πολύ δύσκολων προβλημάτων (κυρίως αλλά όχι μόνο βελτιστοποίησης) αποτυπώνοντας με κάποιο τρόπο τις τεχνικές εκείνες που τα έμβια όντα σχηματίζουν κοπάδια, σμήνη και ομάδες και προσπαθούν να επιβιώσουν μέσα στο ευρύτερο περιβάλλον.

Η κοινωνική συμπεριφορά και η κοινωνική μάθηση που λαμβάνει ένα άτομο σε μια κοινωνία επιτρέπει στο άτομο να αναπτύξει μια γνώση ώστε μαζί με τα υπόλοιπα άτομα να επιβιώσουν και να εξελιχθούν. Οι άνθρωποι λύνουν διάφορα προβλήματα, μιλώντας για αυτά, με άλλους ανθρώπους, ανταλλάσσοντας σχέδια, σκέψεις, ιδέες και κοινωνικές αλλαγές συμπεριφοράς. Οι αλλαγές αυτές μπορούν να απεικονισθούν καθώς τα άτομα κινούνται μεταξύ τους μέσα στο κοινωνικό περιβάλλον. Αυτό είναι περισσότερο εμφανές στα κοπάδια από ψάρια και σμήνη από πουλιά και έντομα, όταν αυτά όλα μαζί αλλάζουν τους τρισδιάστατους σχηματισμούς τους είτε για να αντιμετωπίσουν ένα πιθανό εχθρό-κυνηγό, είτε για να βρουν τροφή, είτε για να μεταφερθούν από ένα μέρος σε κάποιο άλλο με τις λιγότερες απώλειες σε ενέργεια, [19].

4.2 Περιγραφή PSO

4.2.1 Τι είναι ο αλγόριθμος PSO

Στην οικογένεια των εξελικτικών αλγορίθμων συχνά παρουσιάζονται τέσσερις κύριες κατηγορίες αλγορίθμων: οι γεννητικοί αλγόριθμοι, ο εξελικτικός προγραμματισμός, οι εξελικτικές στρατηγικές και ο γεννητικός προγραμματισμός. Όπως στις παραπάνω τεχνικές, ο αλγόριθμος PSO, χρησιμοποιεί ένα πληθυσμό από υποψήφιες λύσεις για την εύρεσης μιας βέλτιστης ή κοντά στην βέλτιστη, λύσης στο πρόβλημα. Το κριτήριο για την βέλτιστη λύση καθορίζεται από τον χρήστη, με την χρήση μιας αντικειμενικής συνάρτησης.

Ο αλγόριθμος PSO, έχει τις βάσεις του στην *τεχνητή ζωή* και στις κοινωνικές σχέσεις, όπως επίσης στην μηχανική και την επιστήμη υπολογιστών, και διαφέρει από τις μεθόδους

εξελικτικών υπολογιστικών μεθόδων (evolutionary computation methods) όσο αφορά το γεγονός ότι τα μέλη του πληθυσμού (particles) οι υποψήφιες λύσεις μπορούν να κινηθούν ελεύθερα μέσα στον υπερχώρο λύσεων του προβλήματος. Αυτό σημαίνει ότι, όταν συσταθεί για πρώτη φορά ο πληθυσμός ως ένα σύνολο από τυχαία διανύσματα μέσα στον υπερχώρο λύσεων, για κάθε υποψήφια λύση αντιστοιχεί μια τυχαία ταχύτητα. Σε κάθε επανάληψη του αλγορίθμου, γίνεται αξιολόγηση κάθε υποψήφιας λύσης και αποθηκεύεται στην μνήμη η καλύτερη θέση που είχε μέχρι στιγμής (personal best) αλλά και η καλύτερη θέση μεταξύ όλου του πληθυσμού (global best). Στην συνέχεια κάθε σωματίδιο (λύση) επιταχύνεται τυχαία προς την συνισταμένη των δύο παραπάνω κατευθύνσεων. Με αυτό τον τρόπο η καλύτερη θέση κάθε σωματιδίου είναι διαθέσιμη ως πληροφορία και στα υπόλοιπα άτομα του πληθυσμού. Οι επαναλήψεις συνεχίζονται μέχρι να ικανοποιηθούν τα κριτήρια τερματισμού.

4.2.2 Σύγκριση με εξελικτικούς αλγορίθμους

Οι σημαντικότερη ομοιότητα που έχει ένας αλγόριθμος PSO, με ένα EA είναι ότι και οι δύο βασίζονται στην ανάπτυξη και εξέλιξη ενός πληθυσμού αρχικών λύσεων. Και στις δύο περιπτώσεις η εξέλιξη του πληθυσμού βασίζεται στην τύχη ενώ η αξιολόγηση κάθε λύσης γίνεται με την εφαρμογή μιας συνάρτησης προσαρμογής (fitness function), ενώ δεν είναι ποτέ σίγουρο ότι θα βρεθεί το ολικό βέλτιστο. Μια άλλη σημαντική ομοιότητα είναι η απλότητα των αλγορίθμων τόσο στην υλοποίηση όσο και στην χρήση. Οι παράμετροι που χρειάζεται να οριστούν εξ αρχής είναι λίγες και στις δύο περιπτώσεις ενώ η υλοποίηση μπορεί να γίνει σε λίγες γραμμές κώδικα, ίσως με τον PSO να υπερέχει λίγο.

Το μεγάλο πλεονέκτημα του αλγορίθμου PSO, σε σχέση με τους εξελικτικούς αλγορίθμους είναι ότι έχει μνήμη. Για κάθε ένα σωματίδιο έχει αποθηκευτεί η καλύτερη θέση του, όπως επίσης και όλα τα ολικά βέλτιστα που έχουν βρεθεί μέχρι στιγμής. Στον PSO δεν υπάρχει εφαρμογή τελεστών επιχιασμού, μετάλλαξης και επιλογής από όλα τα σημεία μέλη του πληθυσμού. Στον PSO, όλες οι λύσεις αναπτύσσονται παράλληλα τόσο σε τοπικό όσο και σε παγκόσμιο επίπεδο, ακολουθώντας κάθε φορά τις καλύτερες λύσεις μέχρι στιγμής. Συνήθως ο PSO είναι πιο αργός σε εξέλιξη και προφανώς στην σύγκλιση, αλλά ταυτόχρονα έχει το σημαντικό πλεονέκτημα ότι είναι δύσκολο να εγκλωβιστεί σε τοπικά βέλτιστα.

4.3 Ο αλγόριθμος PSO

4.3.1 Μοντελοποίηση

Το μεγάλο πλεονέκτημα του PSO είναι η εύκολη υλοποίηση του ως εκτελέσιμο πρόγραμμα στον Η/Υ. Και αυτό γιατί οι βασικές αρχές περιγράφονται από μερικές απλές εξισώσεις:

$$\begin{aligned} v_{i,j} &\leftarrow c_0 v_i + c_1 r_1 (\widehat{g}_j - x_{i,j}) + c_2 r_2 (\widehat{x}_{i,j} - x_{i,j}) \\ x_{i,j} &\leftarrow x_{i,j} + v_{i,j}, \quad i = 1, \dots, n, j = 1, \dots, m \end{aligned} \quad (4.1)$$

Έστω η $f: \mathbb{R}^m \rightarrow \mathbb{R}$, να είναι η συνάρτηση κατάστασης ή αντικειμενική συνάρτηση στόχος, όπου m οι μεταβλητές σχεδιασμού. Το παρόν σμήνος αποτελείται από n σωματίδια, όπου κάθε ένα έχει μια τυχαία θέση $x_i \in \mathbb{R}^m$ και μια τυχαία ταχύτητα $v_i \in \mathbb{R}^m$, $i = 1, \dots, n$. Έστω ως $\widehat{x}_i$ η καλύτερη θέση για κάθε σωματίδιο, και έστω ως $\widehat{g}$ το ολικό βέλτιστο. Ο αλγόριθμος περιγράφεται από τα παρακάτω βήματα:

1. Πραγματοποιείται σύνθεση του αρχικού πληθυσμού, αρχικοποιώντας τις υποψήφιες λύσεις x_i και αρχικές ταχύτητες v_i . Οι παράμετροι σχεδιασμού μπορούν να πάρουν αρχικές τιμές όπως στην περίπτωση των ΔΕΑ:

$$x_{i,j} = rand_j(0,1) \cdot (b_{j,U} - b_{j,L}) + b_{j,L}. \quad (4.2)$$

όπου $b_{j,U}$, και $b_{j,L}$ τα πάνω και κάτω όρια αντίστοιχα για την j^{th} παράμετρο, και $rand_j(0,1)$ ένας τυχαίος αριθμός με ομοιόμορφη κατανομή μεταξύ 0 και 1. Οι αρχικές ταχύτητες μπορούν να θεωρηθούν είτε τυχαίες, είτε μηδενικές, $v_i = 0$.

2. Γίνεται αξιολόγηση του αρχικού πληθυσμού, και αποθήκευση του ολικού βέλτιστου στην παράμετρο $\hat{g}$, ενώ ταυτόχρονα οι αρχικές λύσεις θεωρούνται ως οι καλύτερες για κάθε ένα σωματίδιο:

$$\begin{aligned} \hat{x}_i &\leftarrow x_i, \\ \hat{g} &\leftarrow best(f(x_i)), i = 1, \dots, n \end{aligned} \quad (4.3)$$

3. Εφόσον τα κριτήρια τερματισμού δεν έχουν ικανοποιηθεί

- i. $iter \leftarrow iter + 1$

- ii. Για κάθε σωματίδιο την τρέχουσα χρονική στιγμή πραγματοποιούνται τα παρακάτω, $1 \leq i \leq n$:

- α'. Γίνεται σύνθεση δύο τυχαίων διανυσμάτων, r_1 και r_2 , με ομοιόμορφη κατανομή στο $[0,1]$, δηλαδή $r_{1,j}, r_{2,j} = rand_j(0,1)$, $j = 1, \dots, m$

- β'. Γίνεται ενημέρωση της θέσεως του σωματιδίου: $x_i \leftarrow x_i + v_i$,

- γ'. Γίνεται ενημέρωση της ταχύτητας του σωματιδίου:

$$v_i \leftarrow \omega_{iter} v_i + c_1 r_1 \circ (\hat{x}_i - x_i) + c_2 r_2 \circ (\hat{g} - x_i),$$

- δ'. Γίνεται ανανέωση της καλύτερης θέσης που έχει το σωματίδιο μέχρι στιγμής: Αν $f(x_i)$ καλύτερο από $f(\hat{x}_i)$, τότε $\hat{x}_i \leftarrow x_i$,

- ε'. Γίνεται ανανέωση της καλύτερης θέσης όλων των σωματιδίων μέχρι στιγμής: Αν $f(x_i)$ καλύτερο από $f(\hat{g})$, τότε $\hat{g} \leftarrow x_i$,

4. Η βέλτιστη λύση είναι η $\hat{g}$, με τιμή αντικειμενικής συνάρτησης $f(\hat{g})$.

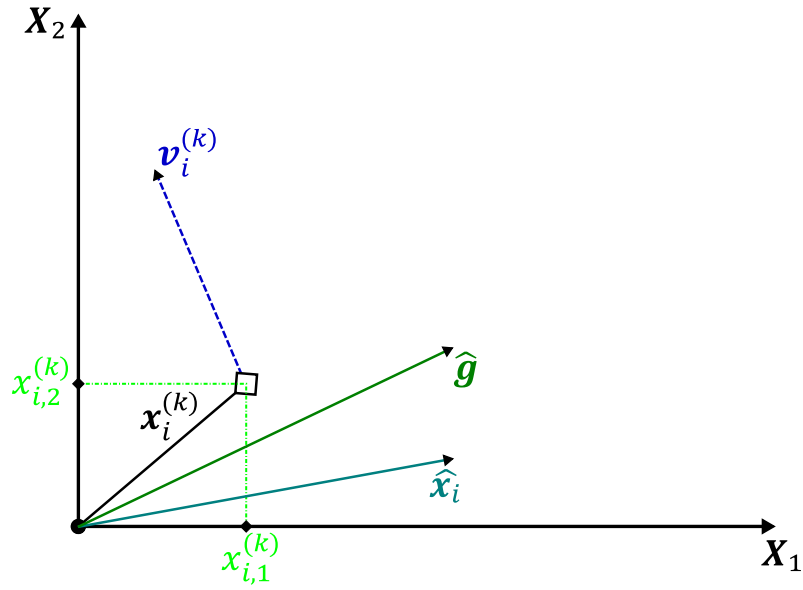
όπου, ω είναι μια σταθερά αδρανείας που επηρεάζει την εξέλιξη του αλγορίθμου. Για μεγάλες τιμές του ω ο PSO πραγματοποιεί έρευνα περισσότερο σε παγκόσμιο επίπεδο, ενώ μικρές τιμές θέτουν τον αλγόριθμο να ερευνά περισσότερο γύρω από τις περιοχές που βρίσκονται τα ίδια τα σωματίδια. Στην μεν πρώτη περίπτωση υπάρχει πιθανότητα ο αλγόριθμος ερευνώντας τυχαία παντού στον υπερχώρο λύσεων να μην μπορεί να συγκλίνει, ενώ στην δεύτερη περίπτωση να εγκλωβιστεί σε ένα τοπικό βέλτιστο. Γι' αυτό προτείνεται η σταδιακή (γραμμική) μείωση του ω κατά την διάρκεια των επαναλήψεων, ώστε στην αρχή, να υπάρχει η δυνατότητα έρευνας σε ευρύτερη περιοχή, αλλά καθώς οι επαναλήψεις αυξάνουν ο αλγόριθμος να επικεντρωθεί σε μια συγκεκριμένη περιοχή:

$$\omega_k = \omega_{max} - \frac{(\omega_{max} - \omega_{min})k}{ik_{max}} \quad (4.4)$$

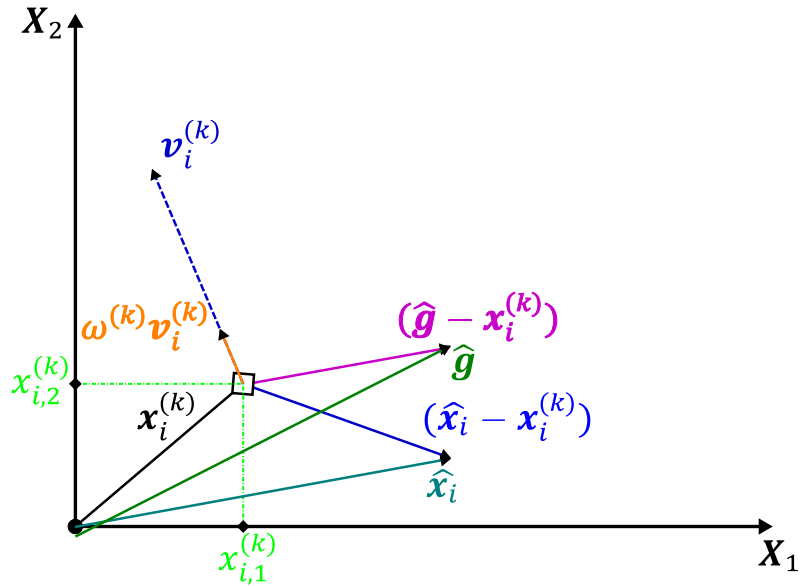
με ω_{max} και ω_{min} η μέγιστη και ελάχιστη τιμή για τον συντελεστή αδρανείας, και k_{max} ο μέγιστος αριθμός επαναλήψεων. Οι παράμετροι c_1 και c_2 είναι διάφοροι συντελεστές επιτάχυνσης που καθορίζουν το βήμα με το οποίο ο αλγόριθμος κινείται τόσο τοπικά, όσο και ευρύτερα. Συνήθως οι τιμές που παίρνουν είναι κοντά στο 2.

4.3.2 Γραφική αναπαράσταση PSO

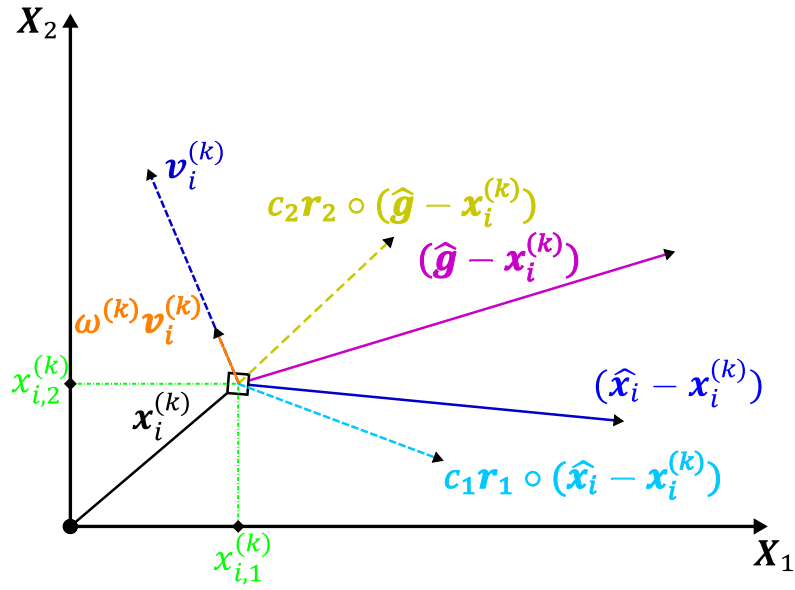
Στις παρακάτω εικόνες παρουσιάζεται γραφικά το βασικό σχήμα της εξίσωσης 4.1 στον $\mathbb{R}^2$ για ένα τυχαίο διάνυσμα $x_i^{(k)}$ στη $k^{στη}$ επανάληψη:



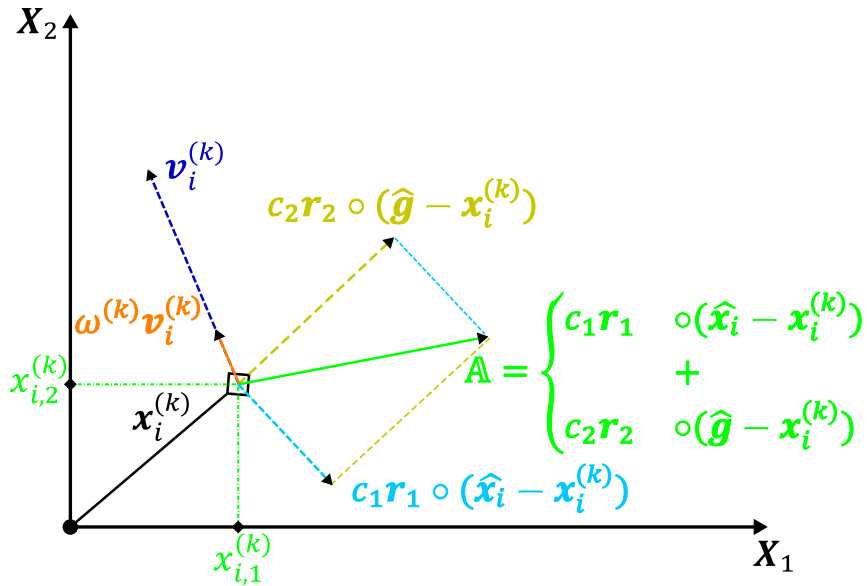
Σχήμα 4.1: Αρχικό διάνυσμα $x_i^{(k)}$ μαζί με ταχύτητα $v_i^{(k)}$, το προσωπικό και ολικό βέλτιστο, $\hat{x}_i$ και $\hat{g}$.



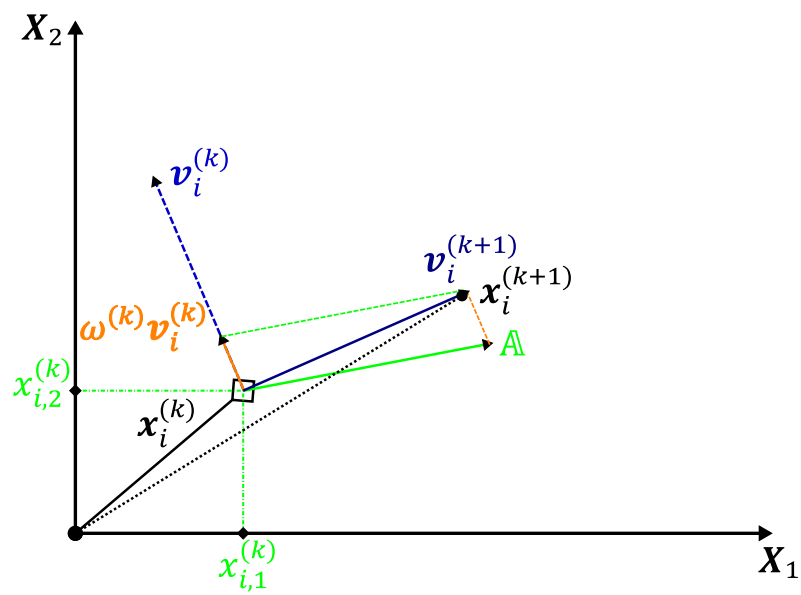
Σχήμα 4.2: Δημιουργία των διανυσματικών διαφορών, $(\hat{g} - x_i)$ και $(\hat{x}_i - x_i)$ καθώς και η κλιμακοποίηση της ταχύτητας $\omega^k v_i$.



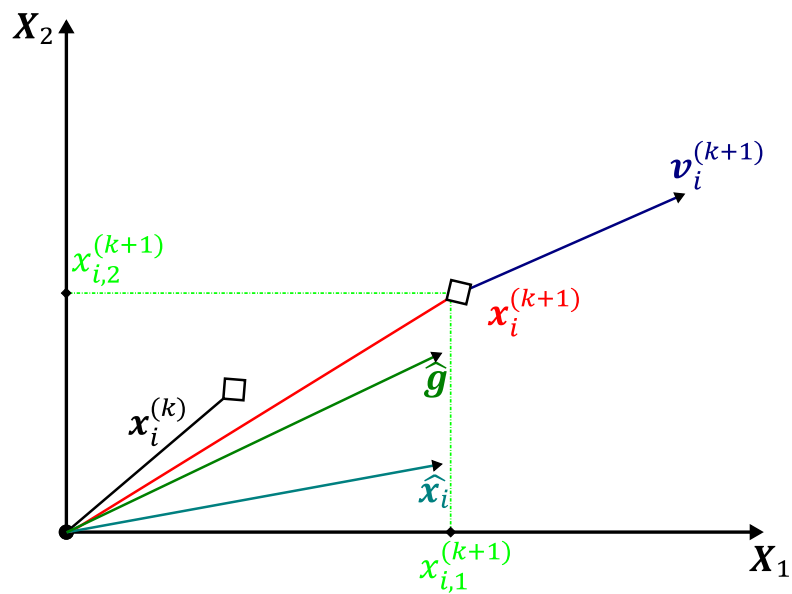
Σχήμα 4.3: Δημιουργία τυχαίων διανυσμάτων $c_1 r_1 \circ (\hat{x}_i - x_i)$ και $c_2 r_2 \circ (\hat{g} - x_i)$, συναρτήσει των παραπάνω διανυσματικών διαφορών.



Σχήμα 4.4: Δημιουργία του $x_i^{(k+1)}$ (βήμα 1).



Σχήμα 4.5: Δημιουργία του $x_i^{(k+1)}$ (βήμα 2).



Σχήμα 4.6: Το διάνυσμα $x_i^{(k+1)}$ μαζί με την ταχύτητα του $v_i^{(k+1)}$

4.3.3 Παρατηρήσεις

Μελετώντας τον αλγόριθμο κάποιος θα μπορούσε να ισχυριστεί ότι πρόκειται και μια προσομοίωση διακριτού χρόνου, αφού κάθε μια επανάληψη αναπαριστά μια στιγμή χρόνου. Τα σωματίδια επικοινωνούν μεταξύ τους ανταλλάσσοντας πληροφορίες σχετικά με την καλύτερη θέση που είχαν τοπικά, αλλά και ολικά. Όταν βρεθεί μια καινούργια βέλτιστη θέση τα σωματίδια μεταβάλλουν την θέση τους ανάλογα μεταδίδοντας ταυτόχρονα την πληροφορία αυτή και σε όλα τα υπόλοιπα μέλη του σμήνους. Τα σωματίδια πάντα κινούνται πάντα σε συνάρτηση της καλύτερης τους θέσης και της καλύτερης θέσης μέσα στο σμήνος. Ο συνδυασμός γίνεται τυχαία, χρησιμοποιώντας τυχαίους πολλαπλασιαστές r_1 και r_2 , κάτι που δίνει στο σμήνος την ικανότητα της στοχαστικής εξερεύνησης. Τα κριτήρια τερματισμού θα μπορούσε να είναι είτε η ολοκλήρωση ενός αριθμού επαναλήψεων, είτε όταν δεν υπάρχει βελτίωση στην τιμή της αντικειμενικής συνάρτησης δηλαδή: $f(g^{(k+1)}) - f(g^{(k)}) \leq \varepsilon$.

ΜΕΡΟΣ ΙΙΙ

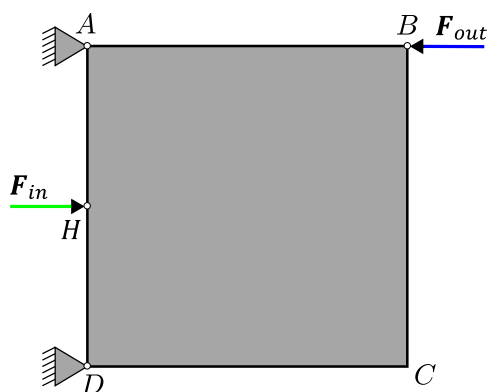
ΕΠΙΛΥΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ ΟΛΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΓΙΑ ΣΥΝΘΕΣΗ ΕΥΚΑΜΠΤΩΝ ΜΗΧΑΝΙΣΜΩΝ

5.1 Παρουσία τοπικών Βέλτιστων

Κατά την επίλυση του προβλήματος σύνθεσης μηχανισμών με χρήση τοπολογικής βελτιστοποίησης διαπιστώθηκε ότι ο επαναληπτικός αλγόριθμος τοπικής αναζήτησης όπως αυτός έχει υλοποιηθεί (βιβλιογραφία για μηχανισμούς) στην βιβλιογραφία καταλήγει σε τοπικά βέλτιστα. Αυτό το φαινόμενο είναι χαρακτηριστικό των αλγορίθμων αυτών, και αποτελεί ένα από τα μεγαλύτερα μειονεκτήματά τους, αφού η διαδικασία της βελτιστοποίησης βασίζεται σε ευρετικά σχήματα χωρίς να εγγυώνται την εύρεση ενός ολικού βέλτιστου. Η τελική κατανομή υλικού εξαρτάται σε πολύ μεγάλο βαθμό από το σημείο εκκίνησης του αλγορίθμου, δηλαδή πως είναι αρχικά κατανομημένο το υλικό μέσα στο χωρίο ορισμού Ω .

Ως παράδειγμα χρησιμοποιήθηκε ο ασύμμετρος μηχανισμός αντιστροφής της δύναμης, όπως φαίνεται στο παρακάτω σχήμα :



Σχήμα 5.1: Ασύμμετρος μηχανισμός αντιστροφής δύναμης.

Το πεδίο σχεδιασμού είναι το τετράγωνο ABCD, και το σημείο εφαρμογής της εισερχόμενης

δύναμης είναι στο σημείο H ενώ το σημείο μεταφοράς κίνησης είναι το σημείο B στην αντίθετη κατεύθυνση. Στα παρακάτω σχήματα φαίνονται καθαρά οι διαφορετικές τελικές κατανομές που προκύπτουν από τον ίδιο αλγόριθμο, ξεκινώντας πάντα από μια τυχαία αρχική κατανομή:

Και στις τέσσερις παραπάνω περιπτώσεις ο αλγόριθμος έτρεξε με τις ίδιες ακριβώς παραμέτρους αλλά με διαφορετικό αρχικό σημείο έναρξης, καταλήγοντας σε αρκετά διαφορετικές λύσεις που προφανώς αποτελούν τοπικά βέλτιστα.

5.2 Υβριδικό σχήμα Ολικής Βελτιστοποίησης

5.2.1 Η ιδέα

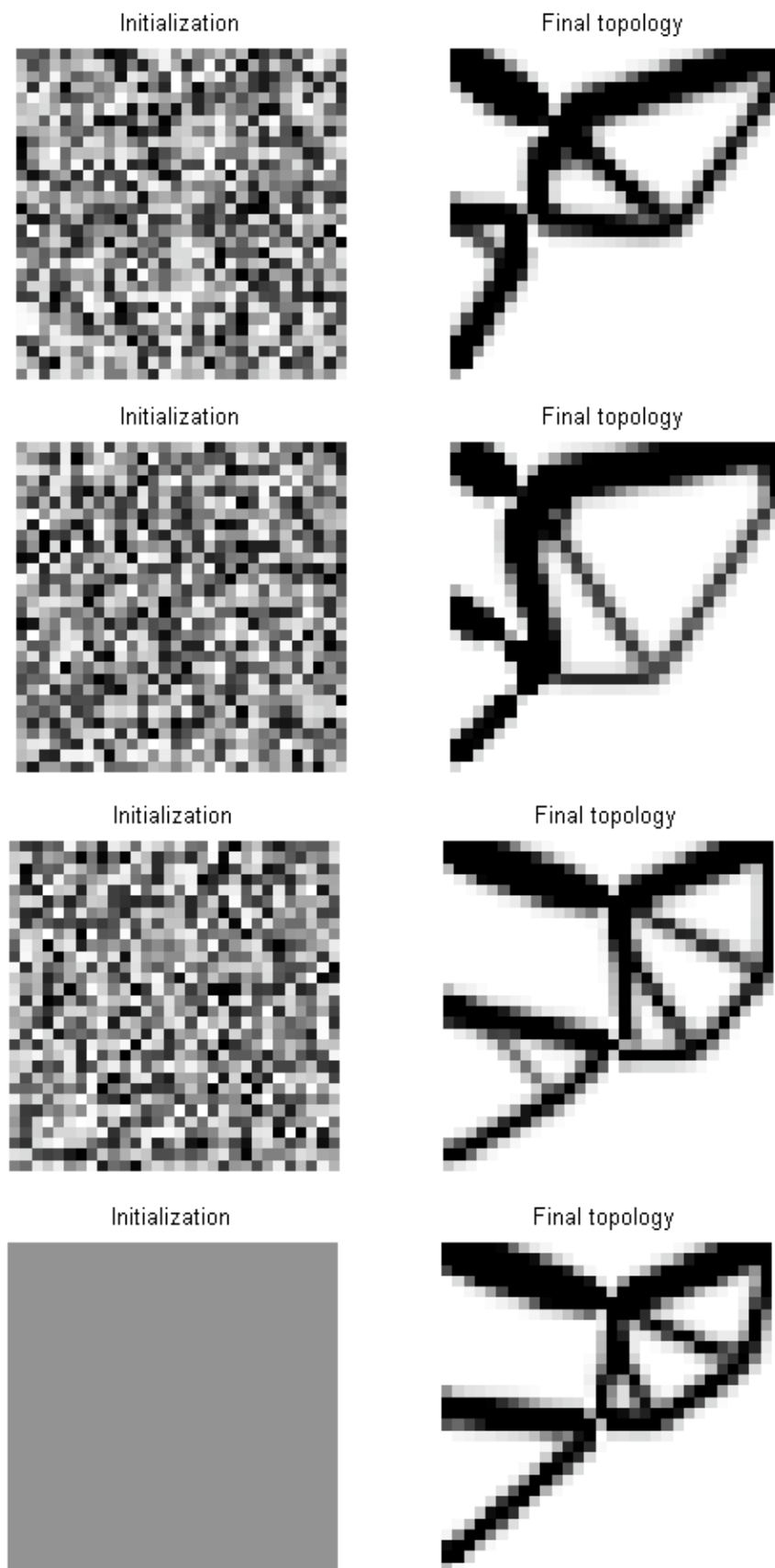
Για την εύρεση ενός ολικού βέλτιστου προτείνεται ένα υβριδικό σχήμα γενικής βελτιστοποίησης με χρήση επαναληπτικών αλγορίθμων τοπικής έρευνας. Οι αλγόριθμοι γενικής βελτιστοποίησης που χρησιμοποιούνται είναι Εξελικτικοί Αλγόριθμοι και Αλγόριθμοι Βελτιστοποίησης Σμήνους Σημείων (PSO). Από το είδος των ΕΑ χρησιμοποιήθηκε ο Διαφορικός Εξελικτικός Αλγόριθμος. Ως εργαλείο αξιολόγησης των υποψήφιων λύσεων μπορεί να χρησιμοποιηθεί ο επαναληπτικός αλγόριθμος τοπικής έρευνας. Με αυτόν τον τρόπο γίνεται εκμετάλλευση όλων των πλεονεκτημάτων των αλγορίθμων γενικής βελτιστοποίησης. Το μόνο μειονέκτημα είναι η αργή σύγκλιση.

Στο άρθρο *A 99 line topology optimization code written in MATLAB* [11], περιγράφεται η δομή του αλγορίθμου τοπολογικής βελτιστοποίησης και οι απαραίτητες μετατροπές που πρέπει να γίνουν για την επίλυση του προβλήματος σύνθεσης εύκαμπτων μηχανισμών. Κάθε μια υποψήφια λύση αποτελείται από μια τελική κατανομή που έχει προκύψει μετά από την εκτέλεση του παραπάνω αλγορίθμου για μια τυχαία αρχική κατανομή. Οι τελικές κατανομές είναι αυτές που τελικά συνδυάζονται μεταξύ τους μέσω των αλγορίθμων DE ή PSO για την εύρεση του ολικού βέλτιστου.

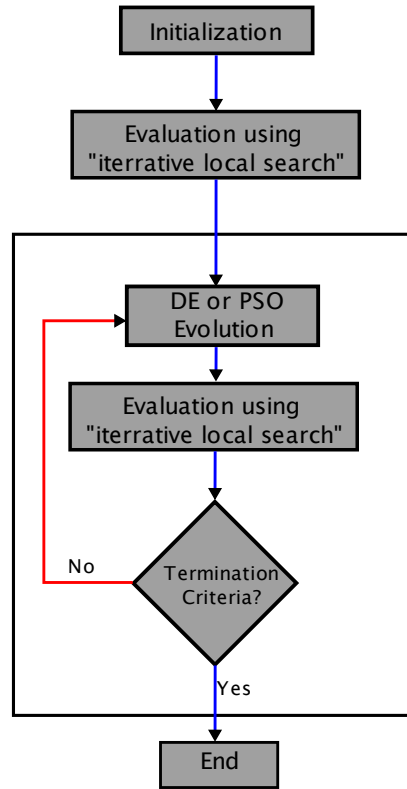
5.2.2 Περιγραφή αλγορίθμου

Ο αλγόριθμος περιγράφεται από τα παρακάτω βήματα:

1. Αρχικοποίηση: Κατά την έναρξη γίνεται αρχικοποίηση του πληθυσμού. Κάθε μέλος αποτελείται από μια τυχαία κατανομή υλικού μέσα στο χώρο Ω .
2. Αξιολόγηση: Για κάθε μία αρχική τυχαία κατανομή εφαρμόζεται ο αλγόριθμος τοπικής βελτιστοποίησης. Έτσι προκύπτουν αντίστοιχες τελικές κατανομές οι οποίες αξιολογούνται.
3. Έναρξη Αλγορίθμου γενικής βελτιστοποίησης :
 - i. Εφαρμογή εξελικτικής διαδικασίας του αλγορίθμου γενικής βελτιστοποίησης (DE ή PSO). Οι τελικές κατανομές που προκύπτουν από το βήμα 2, χρησιμοποιούνται ως τα αρχικά μέλη του πληθυσμού.
 - ii. Η αξιολόγηση κάθε μέλους γίνεται με την εφαρμογή του αλγορίθμου τοπικής έρευνας.
 - iii. Αν υπάρχει ικανοποίηση των κριτηρίων τερματισμού τότε γίνεται λήξη του αλγορίθμου.
4. Μετα-επεξεργασία: Γίνεται μετά-επεξεργασία των αποτελεσμάτων για εξαγωγή συμπερασμάτων.



Σχήμα 5.2: Διαφορετικές τελικές κατανομές υλικού με διαφορετικά σημεία εκκίνησης.



Σχήμα 5.3: Λογικό διάγραμμα υβριδικού αλγορίθμου

5.2.3 Μία περίπτωση φόρτισης: αντικειμενική συνάρτηση

Ως αντικειμενική συνάρτηση για την αξιολόγηση κάθε τελικής κατανομής για τον αλγόριθμο γενικής βελτιστοποίησης χρησιμοποιήθηκε η μεγιστοποίηση της μετακίνησης $u_{out}^{@X}$ στον κόμβο στο σημείο εξόδου (B) στην αντίθετη κατεύθυνση της εισερχόμενης δύναμης:

$$\max_x |u_{out}^{@X}| \quad (5.1)$$

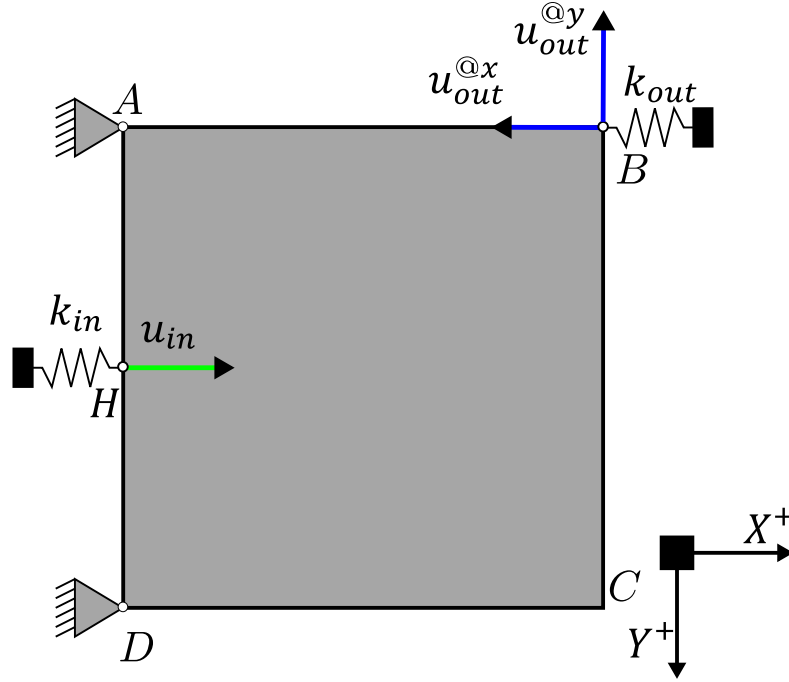
όπου x οι διάφορες κατανομές υλικού.

5.2.4 Έλεγχος μετακίνησης

Εκτός από την μεγιστοποίηση της μετακίνησης στον κόμβο εξόδου κατά την αντίθετη κατεύθυνση της εισερχόμενης δύναμης $u_{out}^{@X}$, ταυτόχρονα θα πρέπει να ελαχιστοποιηθεί η απόλυτη τιμή της μετακίνησης στην κάθετη κατεύθυνση $u_{out}^{@Y}$. Η ελαχιστοποίηση αυτή εκφράζεται ως ένα επιπλέον κριτήριο για το γενικό πρόβλημα βελτιστοποίησης:

$$\begin{cases} \max_x |u_{out}^{@X}| \\ \text{και} \\ \min_x |u_{out}^{@Y}| \end{cases} \quad (5.2)$$

Το δεύτερο κριτήριο μπορεί να γραφτεί ως μεγιστοποίηση του αντίστροφου του: $\max \left| \frac{1}{u_{out}^{@Y}} \right|$. Οπότε τα δύο κριτήρια θα μπορούν να συνδυαστούν σε ένα, με την μορφή κλάσματος, ώστε να ικανοποιούνται και τα δύο ταυτόχρονα:



Σχήμα 5.4: Μεγιστοποίηση $u_{out}^{@X}$ και ελαχιστοποίηση $u_{out}^{@Y}$.

$$\max_x \left| \frac{u_{out}^{@X}}{u_{out}^{@Y}} \right| \quad (5.3)$$

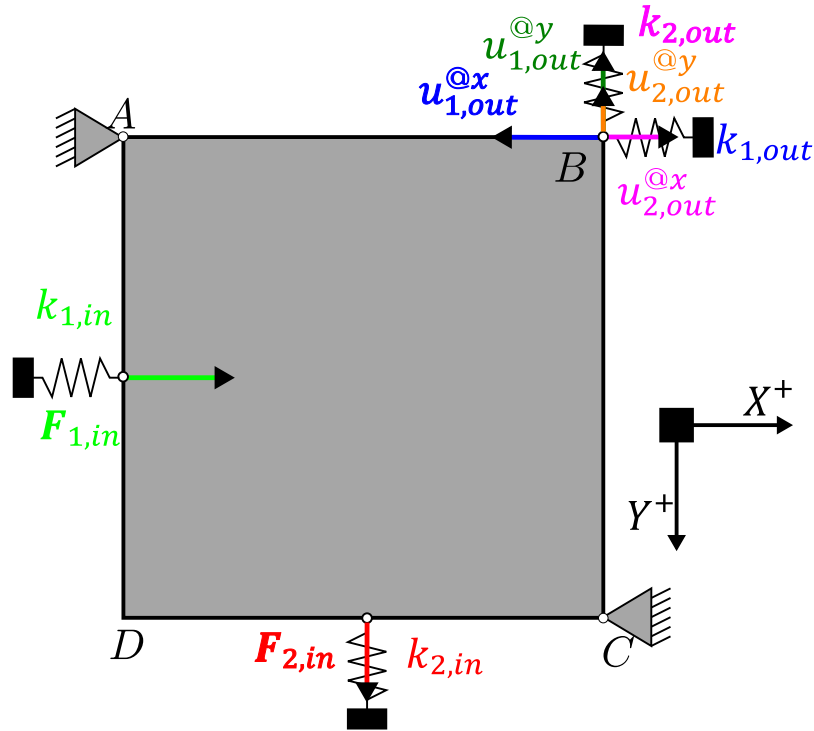
5.3 Σύνθεση πολυμορφικού εύκαμπτου μηχανισμού

5.3.1 Μόρφωση Προβλήματος

Στην περίπτωση αυτή σκοπός είναι να βρεθεί μια κατανομή υλικού ώστε για δύο ή και περισσότερες διαφορετικές φορτίσεις ο μηχανισμός να λειτουργεί διαφορετικά, μεταδίδοντας διαφορετική κίνηση ανά φόρτιση. Έτσι επιτυγχάνεται η πολυμορφικότητα του μηχανισμού. Στο παρακάτω σχήμα παρουσιάζεται η δομή του προβλήματος. Ως Ω επιλέγεται το τετράγωνο χωρίο "ABCD". Οι εισερχόμενες δυνάμεις $F_{1,in}$ και $F_{2,in}$ έχουν ως σημεία εφαρμογής τους κόμβους H και I. Το ζητούμενο είναι το σημείο B να κινηθεί αντίθετα στις κατευθύνσεις κάθε εισερχόμενης δύναμης $F_{1,in}$ ή $F_{2,in}$, να λειτουργήσει δηλαδή ο μηχανισμός διαφορετικά σε κάθε περίπτωση φόρτισης. Στο σχήμα 5.5 παρουσιάζονται γραφικά οι μεταβλητές του προβλήματος.

Στην περίπτωση αυτή το πρόβλημα μορφώνεται ως πολυκριτήριο πρόβλημα βελτιστοποίησης δύο κριτηρίων χωρίς έλεγχο μετακινήσεων:

$$\begin{cases} \max_x |u_{1,out}^{@X}| \\ \text{και} \\ \min_x |u_{2,out}^{@Y}| \end{cases} \quad (5.4)$$



Σχήμα 5.5: Μόρφωση προβλήματος σύνθεσης μηχανισμών για δύο περιπτώσεις φορτίσεων.

5.3.2 Περιορισμοί ελέγχου μετακίνησης

Για κάθε περίπτωση φόρτισης το σημείο B μπορεί να κινηθεί και στις δύο διευθύνσεις X, Y . Στην μεν πρώτη περίπτωση το σημείο B θα πρέπει να κινηθεί αποκλειστικά στην αντίθετη κατεύθυνση από την δύναμη $F_{1,in}$. Θα πρέπει δηλαδή η μετακίνηση στον άξονα $-X$, $u_{1,out}^{@X}$ να μεγιστοποιηθεί και ταυτόχρονα η μετακίνηση $u_{1,out}^{@Y}$ (είτε στον $+Y$, είτε στον $-Y$ άξονα) να τείνει προς το μηδέν. Αντιστοίχως, στην δεύτερη περίπτωση φόρτισης η μετακίνηση στον $-Y$ άξονα $u_{2,out}^{@Y}$ θα πρέπει να μεγιστοποιηθεί και ταυτόχρονα η μετακίνηση στον X , $u_{2,out}^{@X}$ να τείνει στο μηδέν. Λαμβάνοντας υπόψιν τους περιορισμούς αυτούς για κάθε ένα κριτήριο ξεχωριστά το πρόβλημα μορφώνεται ως εξής:

$$\begin{cases} \max_x |u_{1,out}^{@X}| \\ \min_x |u_{1,out}^{@Y}| \\ \text{και} \\ \max_x |u_{2,out}^{@Y}| \\ \min_x |u_{2,out}^{@X}| \end{cases} \quad (5.5)$$

Λαμβάνοντας την μόρφωση του προβλήματος για μια περίπτωση φόρτισης, το πρόβλημα μετατρέπεται ως εξής:

$$\left\{ \begin{array}{l} \max_x \left| \frac{u_{1,out}^{@X}}{u_{1,out}^{@Y}} \right| \\ \text{και} \\ \max_x \left| \frac{u_{2,out}^{@Y}}{u_{2,out}^{@X}} \right| \end{array} \right. \quad (5.6)$$

Για την μετατροπή του παραπάνω προβλήματος σε μονοκριτήριο αρκεί να γίνει μεγιστοποίηση του ελάχιστου των δύο παραπάνω κριτηρίων. Έτσι ταυτόχρονα γίνεται μεγιστοποίηση και των δύο κριτηρίων. Το πρόβλημα μορφώνεται ως:

$$\max_x \left\{ \min \left\{ \left| \frac{u_{1,out}^{@X}}{u_{1,out}^{@Y}} \right|, \left| \frac{u_{2,out}^{@Y}}{u_{2,out}^{@X}} \right| \right\} \right\} \quad (5.7)$$

6.1 Εισαγωγή

Στο κεφάλαιο παρουσιάζονται τα αποτελέσματα σύνθεσης εύκαμπτων μηχανισμών για μια περίπτωση φόρτισης. Σε κάθε μια περίπτωση η επίλυση έγινε χρησιμοποιώντας ως ολικό αλγόριθμο βελτιστοποίησης, τον διαφορικό εξελικτικό αλγόριθμο (DE), και αλγόριθμο βελτιστοποίησης με χρήση νοημοσύνης σμήνους σωματιδίων (PSO).

Σε κάθε παράδειγμα, και για κάθε περίπτωση γενικής βελτιστοποίησης χρησιμοποιήθηκαν οι ίδιες γενικές παράμετροι σχετικά με το μέγεθος και το πλήθος των γενιών για να υπάρχει μια κοινή βάση για σύγκριση των δύο γενικών αλγορίθμων μεταξύ τους.

6.2 Σύνθεση μηχανισμού για μια περίπτωση φόρτισης

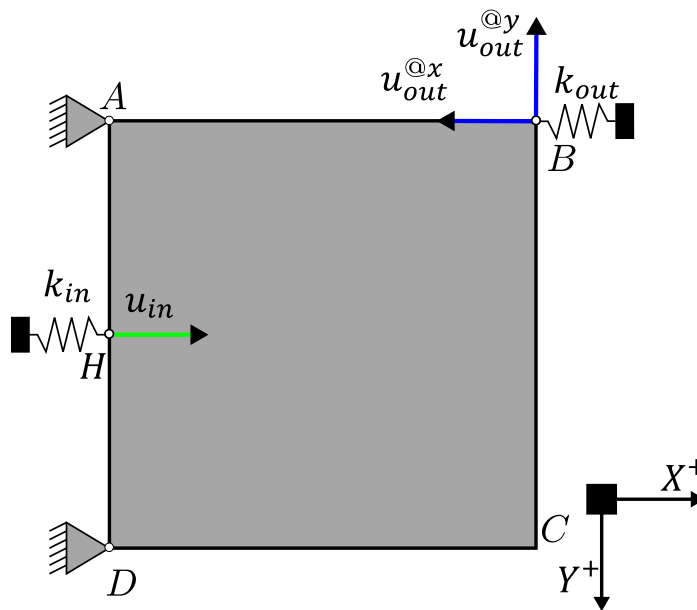
6.2.1 Περιγραφή προβλήματος

Στο συγκεκριμένο παράδειγμα μελετήθηκε το πρόβλημα του ασύμμετρου εύκαμπτου μηχανισμού. Ο ορισμός *ασύμμετρος* δίνεται για δικαιολογήσει το γεγονός ότι η αντιστροφή της δύναμης γίνεται εκτός της διεύθυνσης εφαρμογής της εισερχόμενης δύναμης. Για το συγκεκριμένο παράδειγμα ζητάτε να βρεθεί η κατάλληλη κατανομή ώστε όταν γίνεται εφαρμογή της δύναμης F_{in} στο σημείο H, το σημείο B να κινηθεί όσο το δυνατόν περισσότερο στην αντίθετη κατεύθυνση ελαχιστοποιώντας την μετατόπιση στην κάθετη διεύθυνση. Για το συγκεκριμένο παράδειγμα χρησιμοποιήθηκε ως γενικός αλγόριθμος ο Διαφορικός Εξελικτικός Αλγόριθμος. Το χωρίο ορισμού ABCD, οι φορτίσεις και οι περιορισμοί στήριξης παρουσιάζονται στο σχήμα 6.1.

6.2.2 Παράμετροι προβλήματος για DEA και αριθμητικά στοιχεία

Στο συγκεκριμένο πρόβλημα έγινε διακριτοποίηση του χωρίου σχεδιασμού σε 30×30 στοιχεία. Για κάθε ένα στοιχείο θα πρέπει να βρεθεί αν θα περιέχει υλικό ή όχι, και γιαυτό οι μεταβλητές σχεδιασμού είναι εννιακόσιες (900). Λόγω της διακριτοποίησης του χωρίου με τετράπλευρα στοιχεία οι συνολικοί κόμβοι του πλέγματος είναι $(30 + 1) \times (30 + 1) = 961$ κόμβοι. Ο κάθε κόμβος έχει δύο βαθμούς ελευθερίας, αρά οι συνολικοί βαθμοί ελευθερίας είναι 1922. Ο περιορισμός του όγκου είναι στο 30% για τον επαναληπτικό αλγόριθμο τοπικής αναζήτησης.

Για τον Διαφορικό Εξελικτικό το πλήθος των ατόμων σε κάθε γενιά είναι 20, και το συνολικό πλήθος γενεών είναι 50. Αυτό σημαίνει ότι οι συνολικός αριθμός εφαρμογής του



Σχήμα 6.1: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης

Παράμετροι προβλήματος για μια περίπτωση Φόρτισης με DEA	
Μέγεθος πληθυσμού	20
Σύνολο γενιών	50
Παράμετρος επιχιασμού C_r	0.9
Παράμετρος Μετάλλαξης F	1.5
Διακριτοποίηση	30x30
Αριθμός μεταβλητών	900
Βαθμοί ελευθερίας	$2 \times (30+1) \times (30+1) = 1922$
Επαναλήψεις τοπικού αλγορίθμου	60
Περιορισμός όγκου	30%

Πίνακας 6.1: Παράμετροι προβλήματος για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DEA

επαναληπτικού αλγορίθμου τοπικής αναζήτησης είναι 1000. Οι συνολικές επαναλήψεις για τον επαναληπτικό αλγόριθμο τοπικής αναζήτησης είναι 60. Αυτό σημαίνει ότι γίνονται 60000 επιλύσεις πεπερασμένων στοιχείων μέχρι το πέρας του αλγορίθμου. Θα πρέπει να σημειωθεί ότι η εφαρμογή του επαναληπτικού αλγορίθμου τοπικής έρευνας είναι ιδιαίτερα χρονοβόρα και καθυστερεί αρκετά την διαδικασία επίλυσης. Οι παράμετροι του προβλήματος παρουσιάζονται συνοπτικά στον πίνακα 6.1.

6.2.3 Αποτελέσματα για μια περίπτωση φόρτισης και DEA

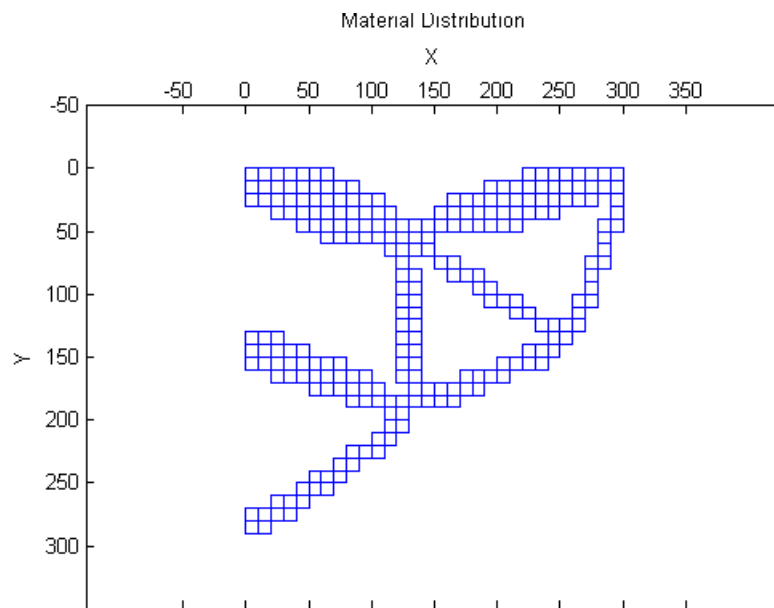
Περισσότερο για λόγους παρουσίασης και μόνο έγινε μια μετά-επεξεργασία στις εικόνες με την τελική κατανομή. Συγκεκριμένα εφαρμόστηκε ένα φίλτρο στην τελική εικόνα με σκοπό την απομόνωση των στοιχείων που η πυκνότητα είναι μεγαλύτερη από το 0.6. Αυτό έγινε περισσότερο για την οπτική παρουσίαση των μετακινήσεων στο σημείο εξόδου. Στον πίνακα 6.2 παρουσιάζονται συνοπτικά τα αποτελέσματα για μια περίπτωση φόρτισης με DEA:

Αποτελέσματα για μια περίπτωση Φόρτισης με DEA		
Τιμή αντικ. συνάρτησης	$\frac{u_{out}^{@X}}{u_{out}^{@Y}}$	51.4
Μετακίνηση στον X, $u_{out}^{@X}$		-1.199
Μετακίνηση στον Y, $u_{out}^{@Y}$		-0.023
Ευκλείδεια απόσταση		1.200
Γεωμ. Πλεονέκτημα (GA)		13.8%
Όγκος υλικού		30%

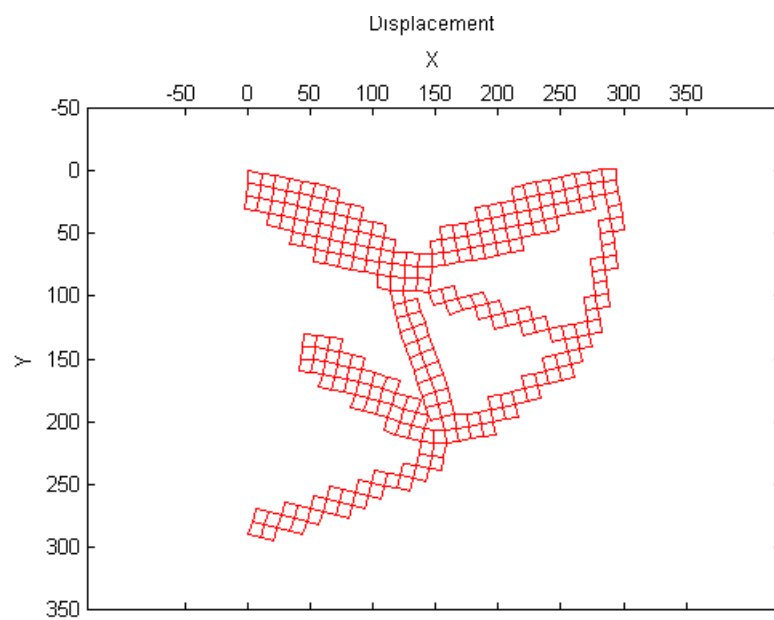
Πίνακας 6.2: Αποτελέσματα για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DEA.



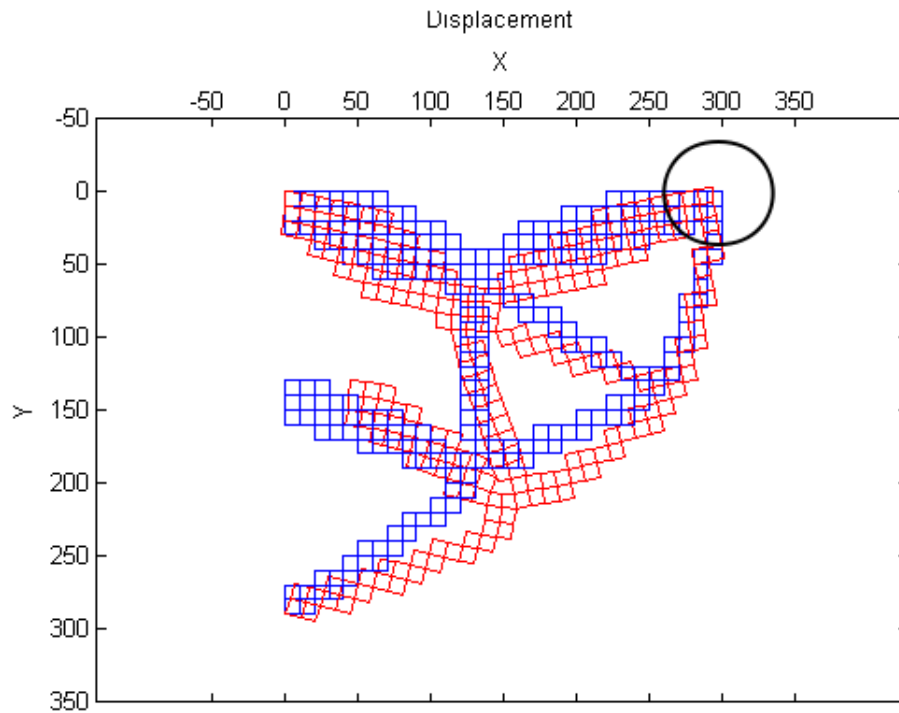
Σχήμα 6.2: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή.



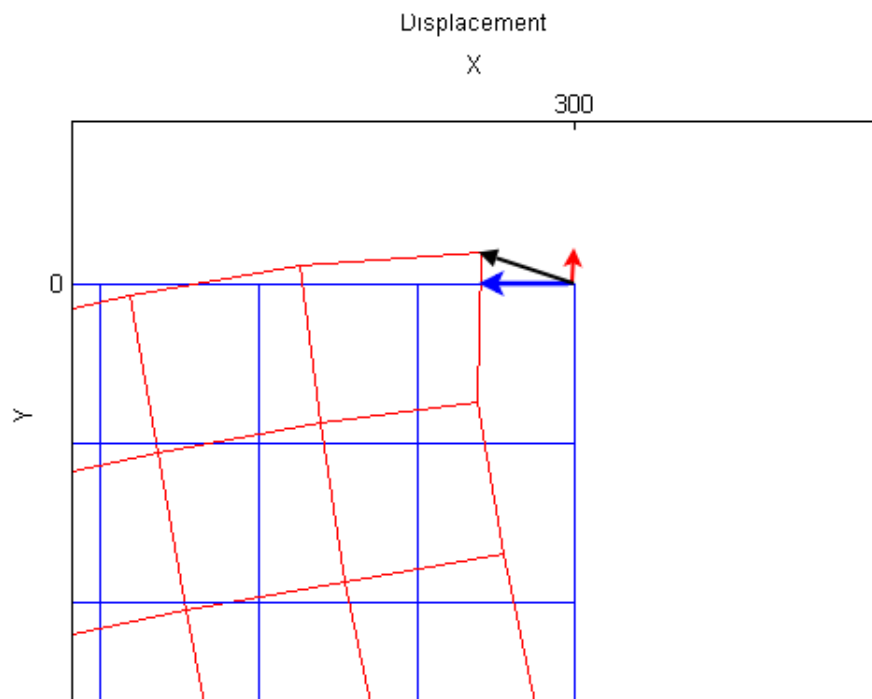
Σχήμα 6.3: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή (με φίλτρο).



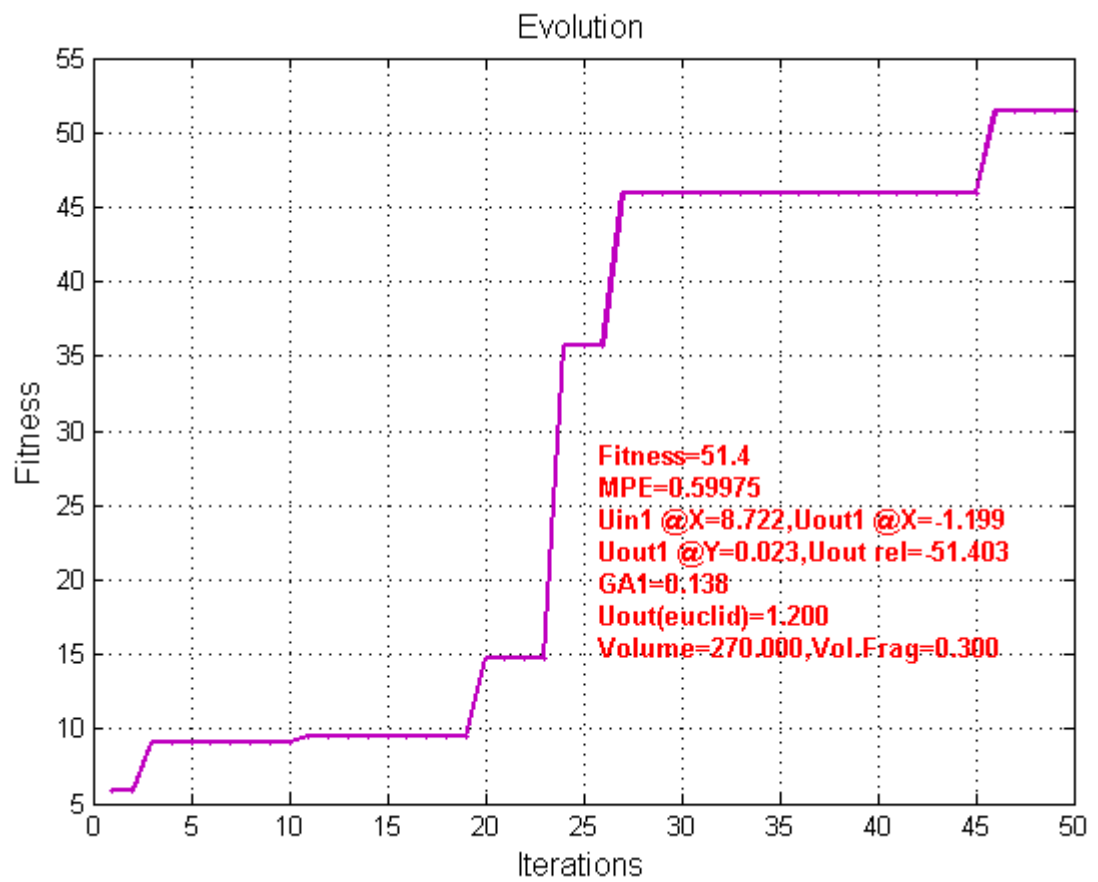
Σχήμα 6.4: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Μετακινήσεις (με φίλτρο).



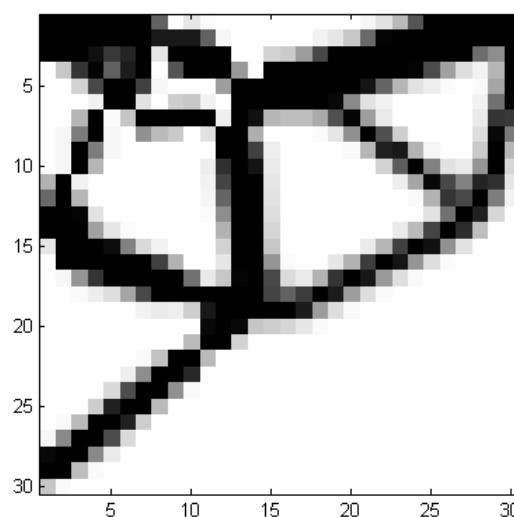
Σχήμα 6.5: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή και Μετακινήσεις (με φίλτρο).



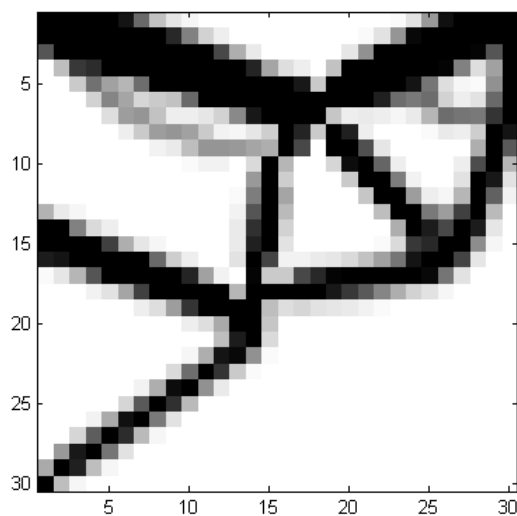
Σχήμα 6.6: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Τελική κατανομή και Μετακινήσεις , λεπτομέρεια (με φίλτρο).



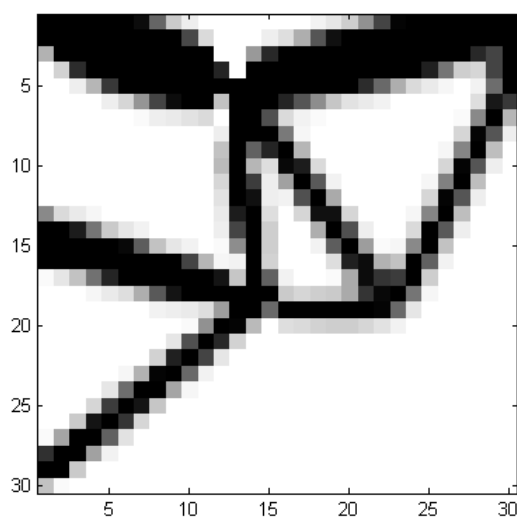
Σχήμα 6.7: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Εξέλιξη αλγορίθμου.



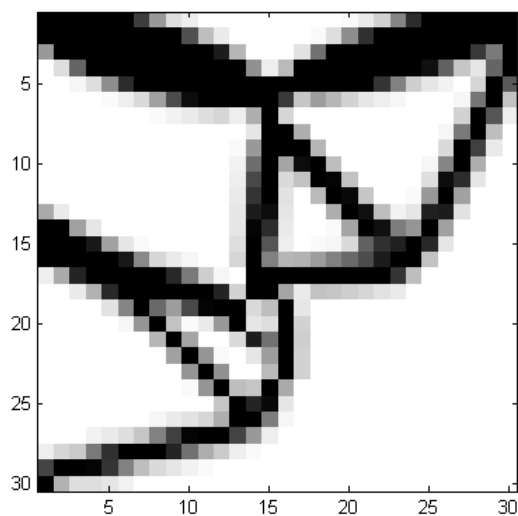
Σχήμα 6.8: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 1 έως 2



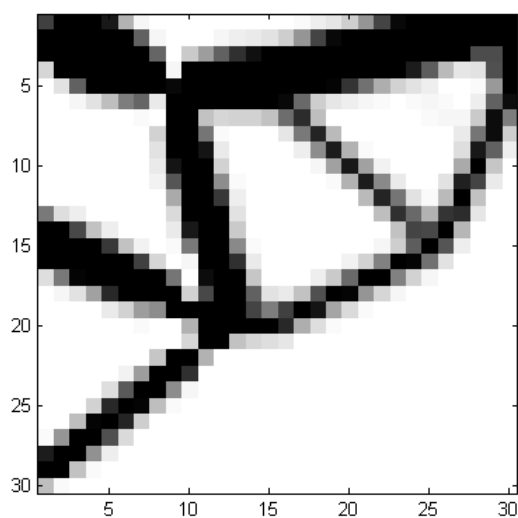
Σχήμα 6.9: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 3 έως 10.



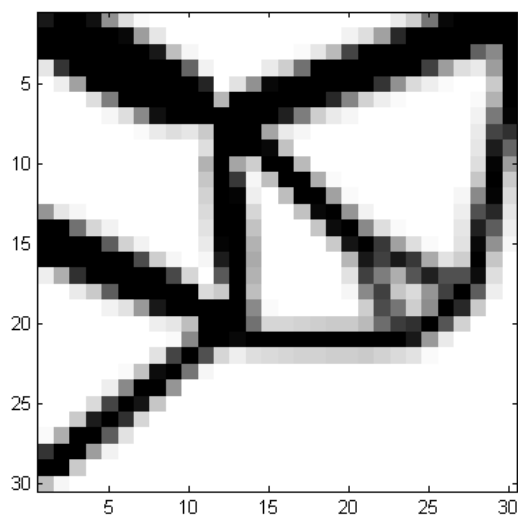
Σχήμα 6.10: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 11 έως 19.



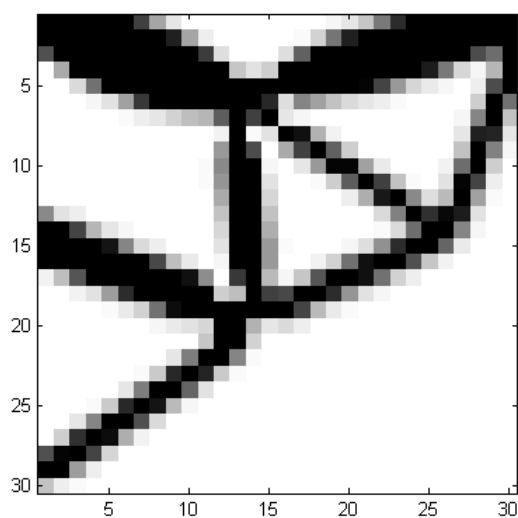
Σχήμα 6.11: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 20 έως 23.



Σχήμα 6.12: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 24 έως 26.



Σχήμα 6.13: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 27 έως 45.



Σχήμα 6.14: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 46 έως 50.

Παράμετροι προβλήματος για μια περίπτωση Φόρτισης με PSO	
Μέγεθος σμήνους	20
Σύνολο τοπικών αναζητήσεων	50
Συντελεστές επιτάχυνσης $c_1 = c_2$	2
Συντελεστές αδράνειας w_{max} και w_{min}	0.9 και 0.1
Διακριτοποίηση	30x30
Αριθμός μεταβλητών	900
Βαθμοί ελευθερίας	$2 \times (30+1) \times (30+1) = 1922$
Επαναλήψεις τοπικού αλγορίθμου	60
Περιορισμός όγκου	30%

Πίνακας 6.3: Παράμετροι προβλήματος για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO.

Αποτελέσματα για μια περίπτωση Φόρτισης με PSO		
Τιμή αντικ. συνάρτησης	$\frac{u_{out}^{@X}}{u_{out}^{@Y}}$	385.7
Μετακίνηση στον X, $u_{out}^{@X}$		-1.158
Μετακίνηση στον Y, $u_{out}^{@Y}$		-0.003
Ευκλείδεια απόσταση		1.158
Γεωμ. Πλεονέκτημα (GA)		13.5%
Όγκος υλικού		30%

Πίνακας 6.4: Αποτελέσματα για μια περίπτωση φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO.

6.2.4 Παράμετροι προβλήματος για αλγόριθμο βελτιστοποίησης με χρήση νοημοσύνης σμήνους σωματιδίων και αριθμητικά στοιχεία - PSO

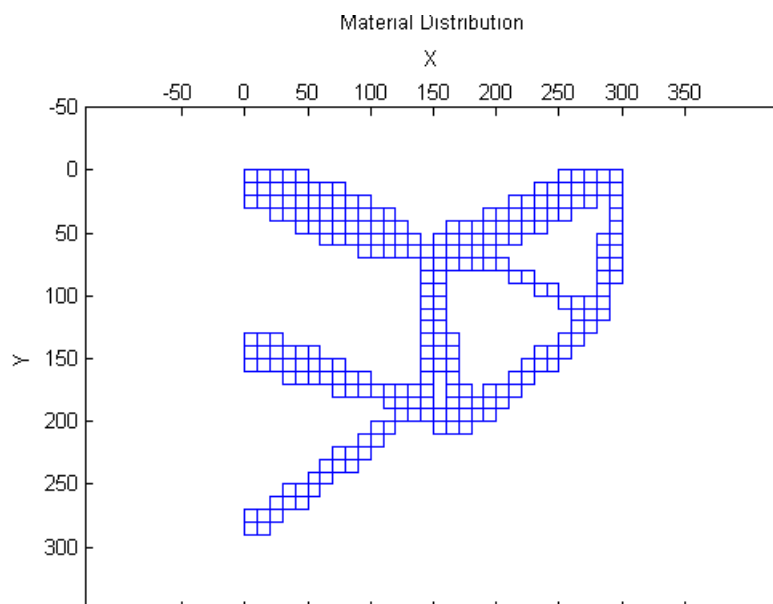
Για το συγκεκριμένο πρόβλημα οι παράμετροι του αλγορίθμου τοπικής έρευνας είναι ακριβώς οι ίδιοι όπως στην περίπτωση του διαφορικού εξελικτικού. Για τον PSO, το σμήνος αποτελείται από 20 σωματίδια, όπου κάθε σωματίδιο μπορεί να εκτελέσει 50 τοπικές αναζητήσεις για κάθε βήμα επανάληψης. Ο συνολικός αριθμός των αναζητήσεων για όλα τα σωματίδια είναι 1000. Οι υπόλοιπες παράμετροι του προβλήματος παρουσιάζονται συνοπτικά στον πίνακα 6.3.

6.2.5 Αποτελέσματα για μια περίπτωση φόρτισης και PSO

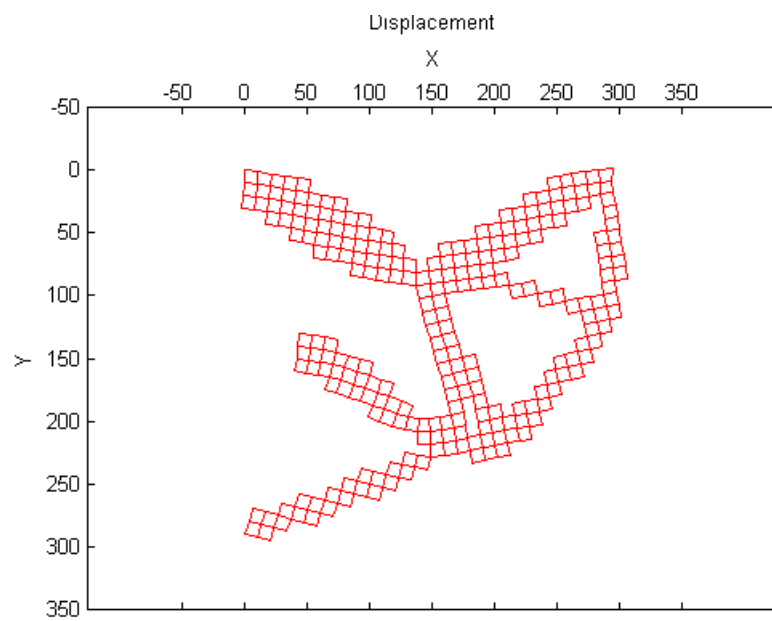
Παρακάτω παρουσιάζονται τα αποτελέσματα για τον ασύμμετρο αντιστροφέα δύναμης με χρήση PSO:



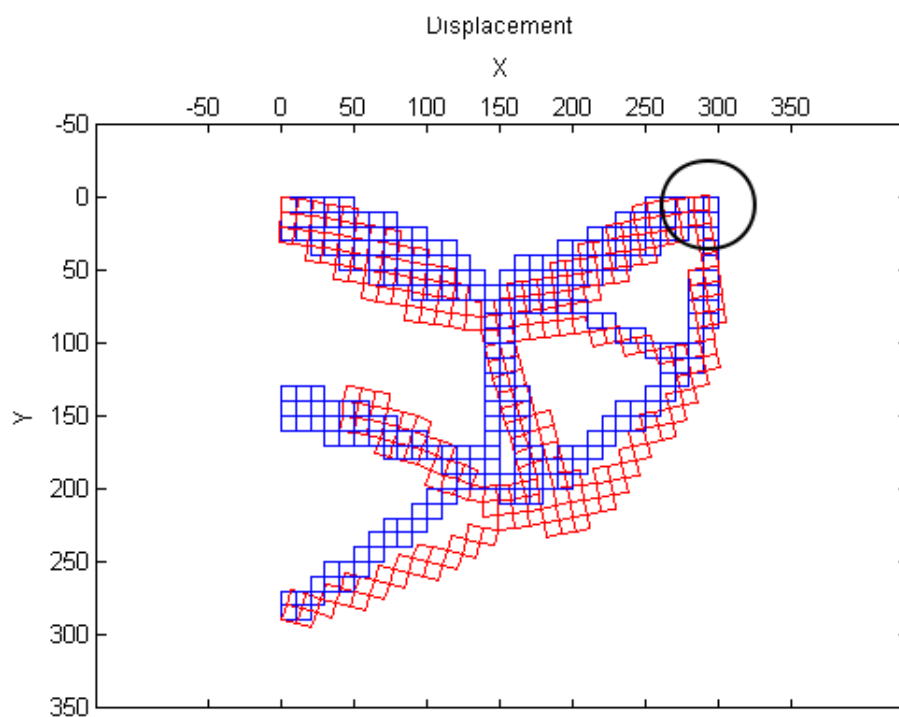
Σχήμα 6.15: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή.



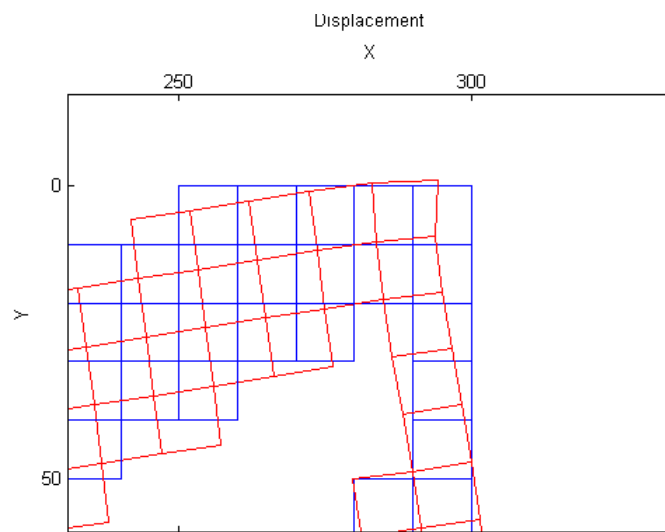
Σχήμα 6.16: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή (με φίλτρο).



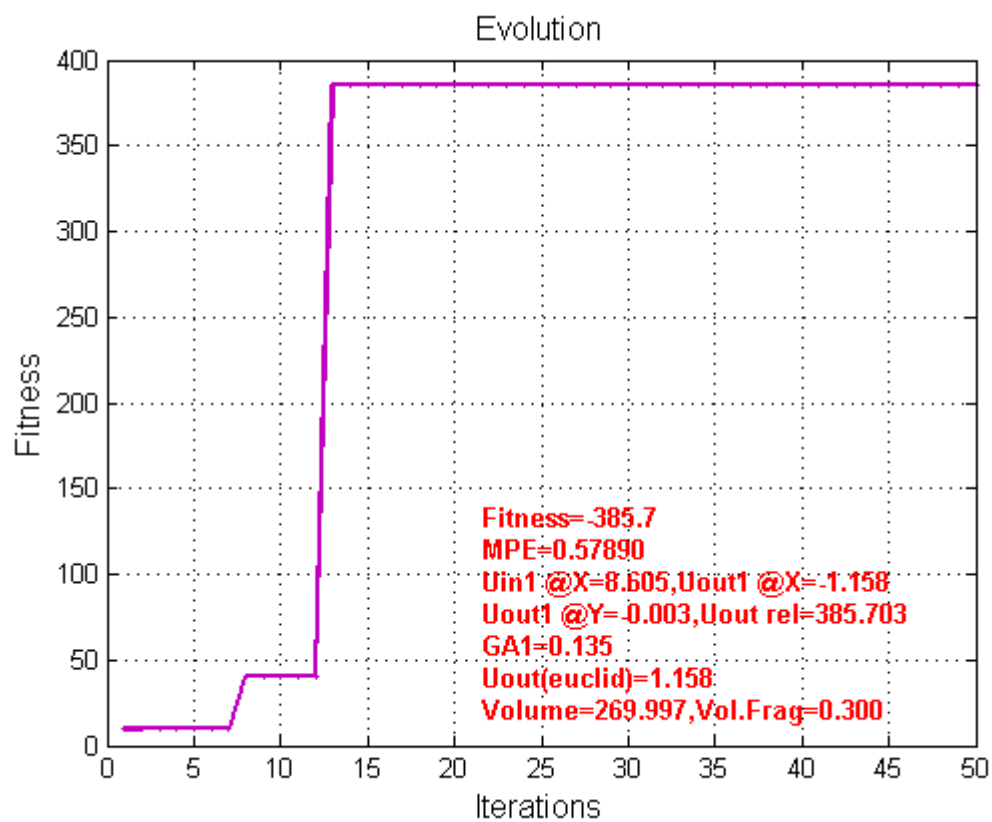
Σχήμα 6.17: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Μετακινήσεις (με φίλτρο).



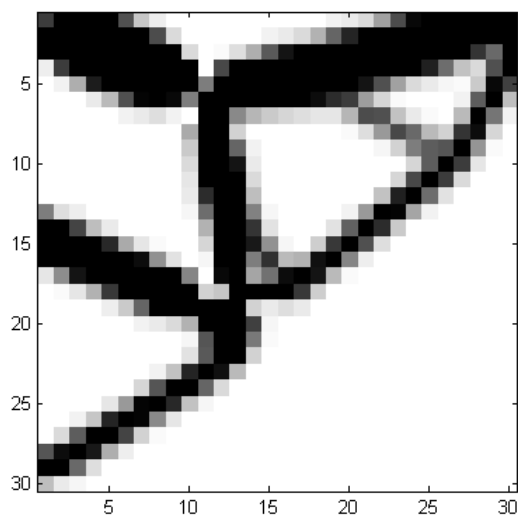
Σχήμα 6.18: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή και Μετακινήσεις (με φίλτρο).



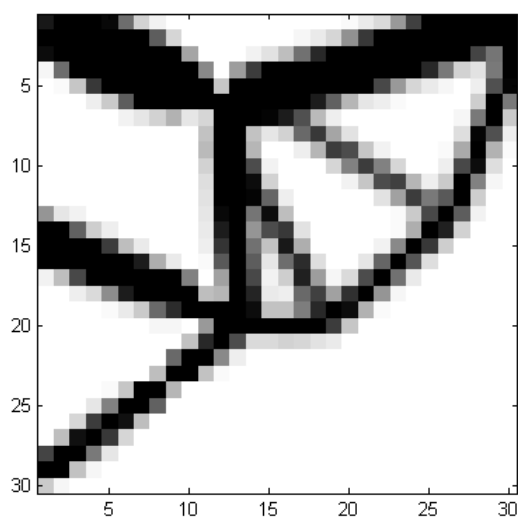
Σχήμα 6.19: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο).



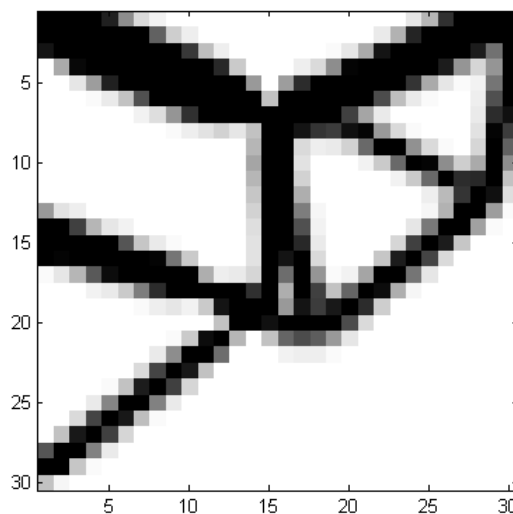
Σχήμα 6.20: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Εξέλιξη αλγορίθμου.



Σχήμα 6.21: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 1 έως 7.



Σχήμα 6.22: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 8 έως 12.



Σχήμα 6.23: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 13 έως 50.

6.3 Σύγκριση αποτελεσμάτων DEA και PSO για μια περίπτωση φόρτισης

Από τα αποτελέσματα φαίνεται ότι ο αλγόριθμος PSO έδωσε πολύ καλύτερη λύση σε σχέση με τον DEA. Οι τιμές των αντικειμενικών συναρτήσεων είναι 357.8 και 51.4 αντίστοιχα. Οι μετακινήσεις στον άξονα X και στις δύο περιπτώσεις είναι της ίδια τάξης μεγέθους, αλλά η μεγάλη διαφορά εμφανίζεται στις μετακινήσεις στον άξονα Y. Συγκεκριμένα ο PSO έδωσε $u_{out}^{@Y}=0.003$, και ο DEA, $u_{out}^{@Y}=0.023$, περίπου 7 φορές μεγαλύτερη τιμή. Το κριτήριο αυτό είναι στον παρανομαστή της αντικειμενικής συνάρτησης και θα πρέπει να ελαχιστοποιείται. Και οι δύο αλγόριθμοι κατέληξαν στην ίδια περίπου κατανομή υλικού, και στο ίδιο ποσοστό όγκου. Στον πίνακα 6.5 παρουσιάζονται τα αποτελέσματα σύγκρισης των δύο γενικών αλγορίθμων μεταξύ τους:

Σύγκριση αποτελεσμάτων DEA και PSO		
	DEA	PSO
Τελική κατανομή		
Αντικ. Συνάρτηση $\left \frac{u_{out}^{@X}}{u_{out}^{@Y}} \right $	51.4	385.7
Μετακίνηση στον X: $u_{out}^{@X}$	-1.199	-1.158
Μετακίνηση στον Y: $u_{out}^{@Y}$	-0.023	-0.003
Γεωμ. Πλεονέκτημα, GA	13.8%	13.5%

Πίνακας 6.5: Σύγκριση αποτελεσμάτων DEA και PSO για μία περίπτωση φόρτισης

ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΥΝΘΕΣΗΣ ΠΟΛΥΜΟΡΦΙΚΩΝ ΜΗΧΑΝΙΣΜΩΝ

7.1 Εισαγωγή

Στο κεφάλαιο παρουσιάζονται τα αποτελέσματα σύνθεσης πολυμορφικών εύκαμπτων μηχανισμών για δύο περιπτώσεις φορτίσεων. Όπως και στο προηγούμενο παράδειγμα για μια περίπτωση φόρτισης έτσι και σε αυτό με δύο περιπτώσεις φορτίσεων η επίλυση έγινε χρησιμοποιώντας ως ολικό αλγόριθμο βελτιστοποίησης, τον διαφορικό εξελικτικό αλγόριθμο (DE), και αλγόριθμο βελτιστοποίησης με χρήση νοημοσύνης σμήνους σωματιδίων (PSO).

Σε κάθε παράδειγμα, και για κάθε περίπτωση γενικής βελτιστοποίησης χρησιμοποιήθηκαν οι ίδιες γενικές παράμετροι σχετικά με το μέγεθος και το πλήθος των γενιών για να υπάρχει μια κοινή βάση για σύγκριση των δύο γενικών αλγορίθμων μεταξύ τους.

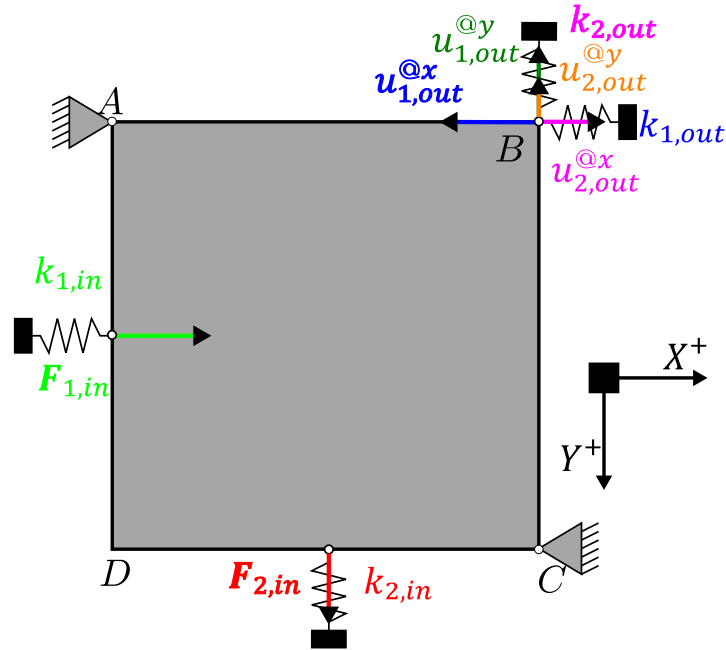
7.2 Σύνθεση μηχανισμού για μια περίπτωση φόρτισης

7.2.1 Περιγραφή προβλήματος

Στο συγκεκριμένο παράδειγμα μελετήθηκε το πρόβλημα του ασύμμετρου εύκαμπτου μηχανισμού για δύο περιπτώσεις φορτίσεων. Και για τις δύο περιπτώσεις φορτίσεων η μετάδοση της κίνησης είναι ασύμμετρη, δηλαδή ο άξονας μετάδοσης τις κίνησης δεν είναι συνευθειακός με τον άξονα εφαρμογής της φόρτισης ενώ το σημείο μετάδοσης της κίνησης είναι το ίδιο και στις δύο περιπτώσεις. Στο σχήμα 7.1 παρουσιάζεται γραφικά το χωρίο σχεδιασμού, οι φορτίσεις, και οι περιορισμοί στήριξης.

7.2.2 Παράμετροι προβλήματος για DEA και αριθμητικά στοιχεία

Στο συγκεκριμένο πρόβλημα έγινε διακριτοποίηση του χωρίου σχεδιασμού σε 50×50 στοιχεία. Για κάθε ένα στοιχείο θα πρέπει να βρεθεί αν θα περιέχει υλικό ή όχι, και γιαυτό οι μεταβλητές σχεδιασμού είναι 2500. Λόγω της διακριτοποίησης του χωρίου με τετράπλευρα στοιχεία οι συνολικοί κόμβοι του πλέγματος είναι $(50 + 1) \times (50 + 1) = 2601$ κόμβοι. Ο κάθε κόμβος έχει δύο βαθμούς ελευθερίας, αρά οι συνολικοί βαθμοί ελευθερίας είναι 5202. Ο περιορισμός του όγκου είναι στο 30% για τον επαναληπτικό αλγόριθμο τοπικής αναζήτησης. Λόγω της μεγαλύτερης διακριτοποίησης αυξήθηκε ο αριθμός των επαναλήψεων για τον αλγόριθμο τοπικής βελτιστοποίησης από 60 σε 80 για καλύτερη σύγκλιση.



Σχήμα 7.1: Μόρφωση προβλήματος σύνθεσης μηχανισμών για δύο περιπτώσεις φορτίσεων

Στο παράδειγμα αυτό χρησιμοποιήθηκαν για τον DEA οι ίδιες παράμετροι για στο παράδειγμα για μια περίπτωση φόρτισης. Οι παράμετροι του προβλήματος παρουσιάζονται συνοπτικά στον πίνακα 7.1.

7.2.3 Αποτελέσματα για μια περίπτωση φόρτισης και DEA

Περισσότερο για λόγους παρουσίασης και μόνο έγινε μια μετά-επεξεργασία στις εικόνες με την τελική κατανομή. Συγκεκριμένα εφαρμόστηκε ένα φίλτρο στην τελική εικόνα με σκοπό την απομόνωση των στοιχείων που η πυκνότητα είναι μεγαλύτερη από το 0.6. Αυτό έγινε περισσότερο για την οπτική παρουσίαση των μετακινήσεων στο σημείο εξόδου. Στον πίνακα 7.2 παρουσιάζονται συνοπτικά τα αποτελέσματα για μια περίπτωση φόρτισης με DEA:

Παράμετροι προβλήματος για δύο περιπτώσεις Φόρτισης με DEA	
Μέγεθος πληθυσμού	20
Σύνολο γενιών	50
Παράμετρος επιχιασμού C_r	0.9
Παράμετρος Μετάλλαξης F	1.5
Διακριτοποίηση	50x50
Αριθμός μεταβλητών	2500
Βαθμοί ελευθερίας	$2 \times (50+1) \times (50+1) = 5202$
Επαναλήψεις τοπικού αλγορίθμου	80
Περιορισμός όγκου	30%

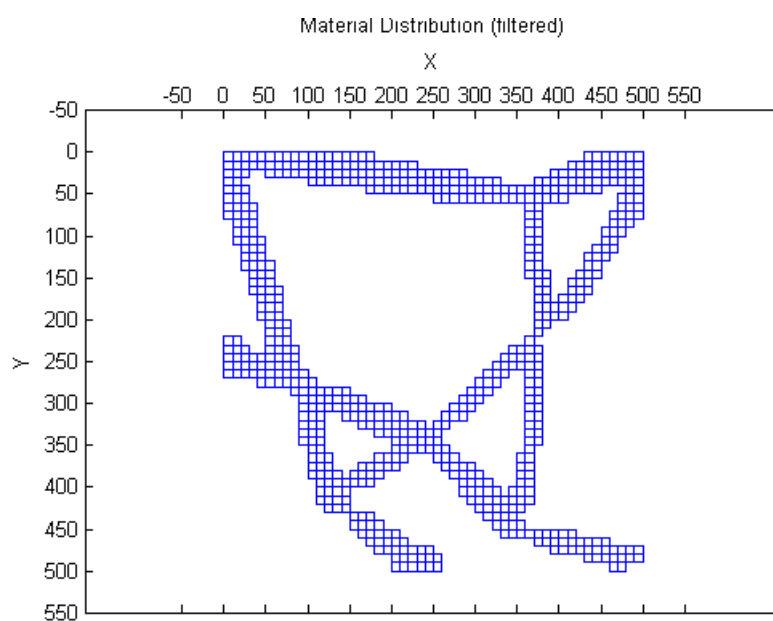
Πίνακας 7.1: Παράμετροι προβλήματος για 2 περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DE

Αποτελέσματα για δύο περιπτώσεις φόρτισης με DEA		
1 ^η Περίπτωση Φόρτισης	Τιμή ολικής αντικ. συνάρτησης	13.303
	Μετακίνηση στον Χ, $u_{1,out}^{@X}$	-0.696
	Μετακίνηση στον Υ, $u_{1,out}^{@Y}$	-0.020
	Τιμή αντικ. συνάρτησης $\frac{u_{1,out}^{@X}}{u_{1,out}^{@Y}}$	33.952
	Γεωμ. Πλεονέκτημα (GA)	8.2%
2 ^η Περίπτωση Φόρτισης	Μετακίνηση στον Χ, $u_{2,out}^{@X}$	0.030
	Μετακίνηση στον Υ, $u_{2,out}^{@Y}$	-0.398
	Τιμή αντικ. συνάρτησης $\frac{u_{2,out}^{@Y}}{u_{2,out}^{@X}}$	13.303
	Γεωμ. Πλεονέκτημα (GA)	4.3%

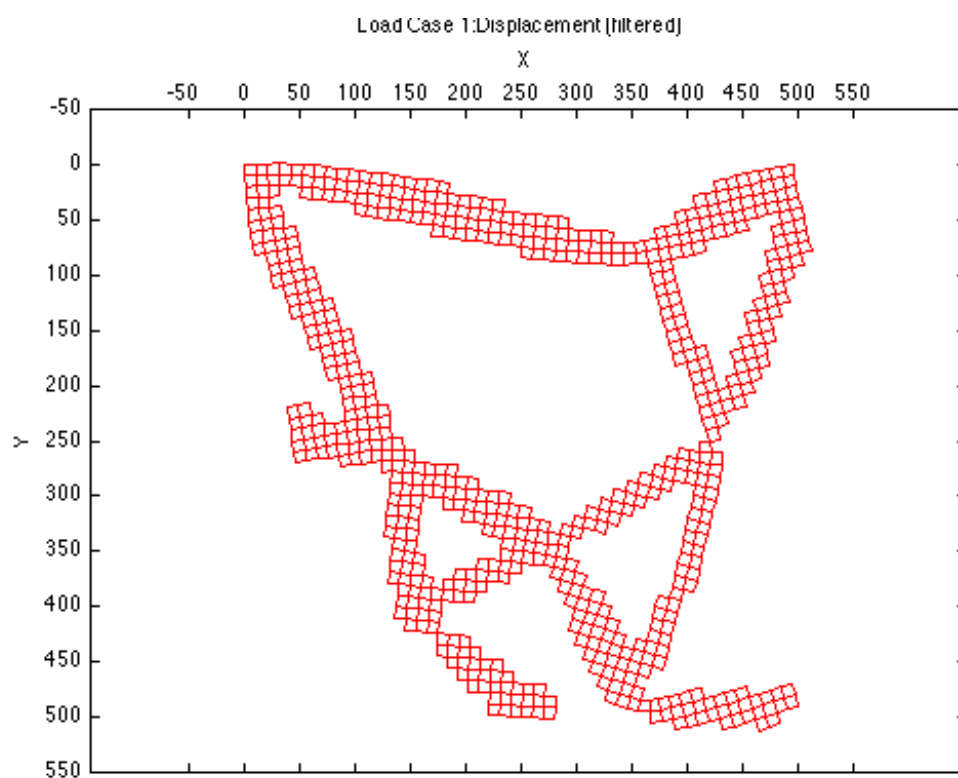
Πίνακας 7.2: Αποτελέσματα για δύο περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: Διαφορικός Εξελικτικός Αλγόριθμος - DEA



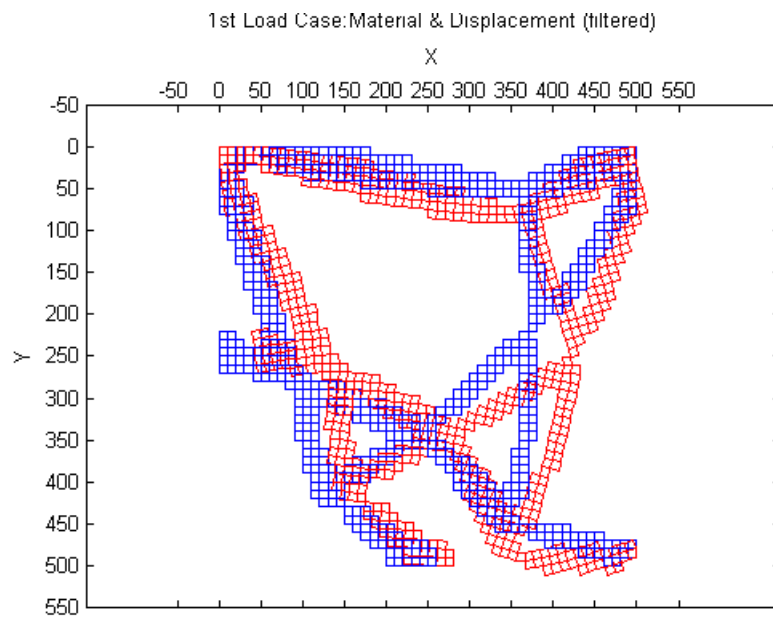
Σχήμα 7.2: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, Τελική κατανομή



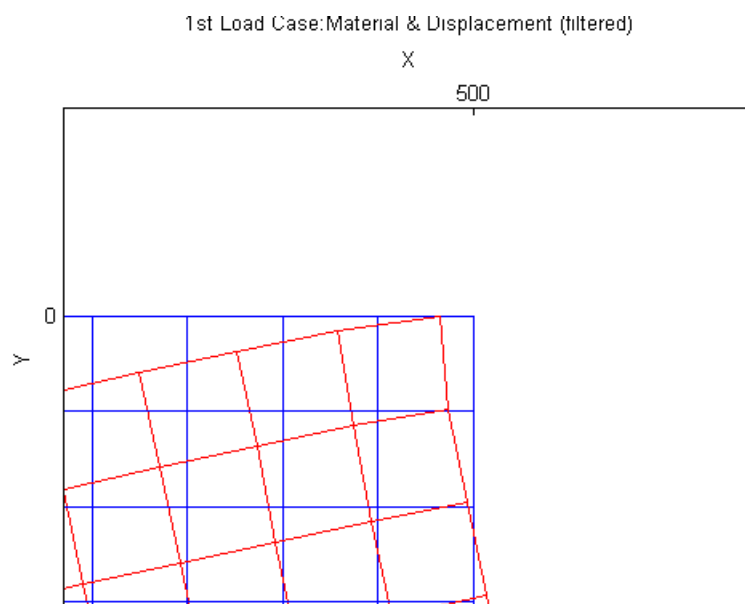
Σχήμα 7.3: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, Τελική κατανομή (με φίλτρο)



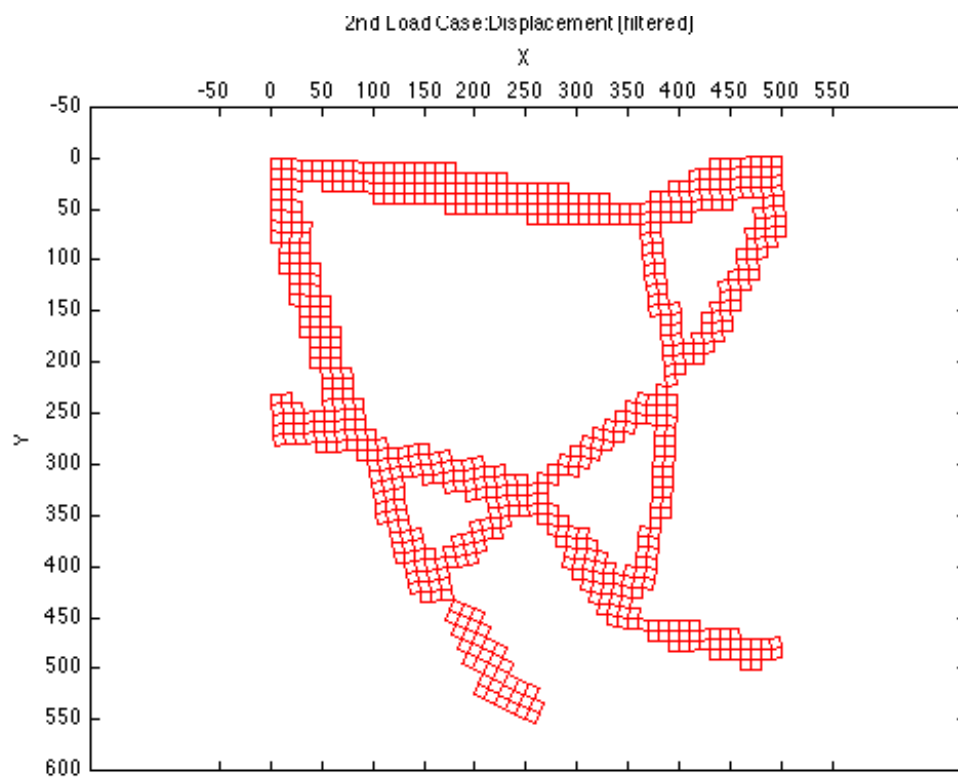
Σχήμα 7.4: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)



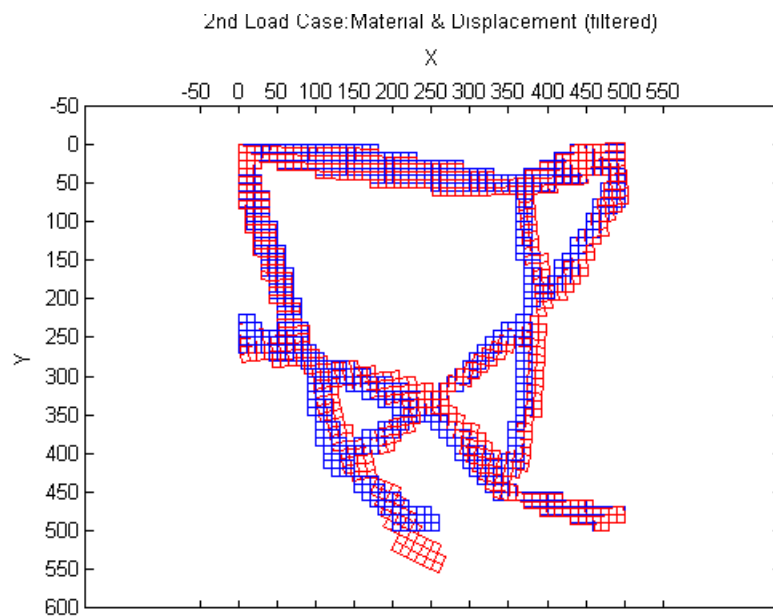
Σχήμα 7.5: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)



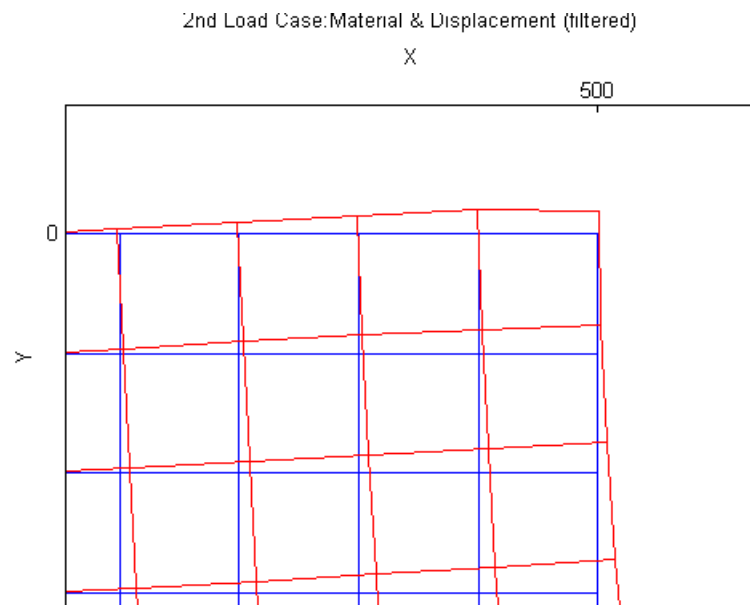
Σχήμα 7.6: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)



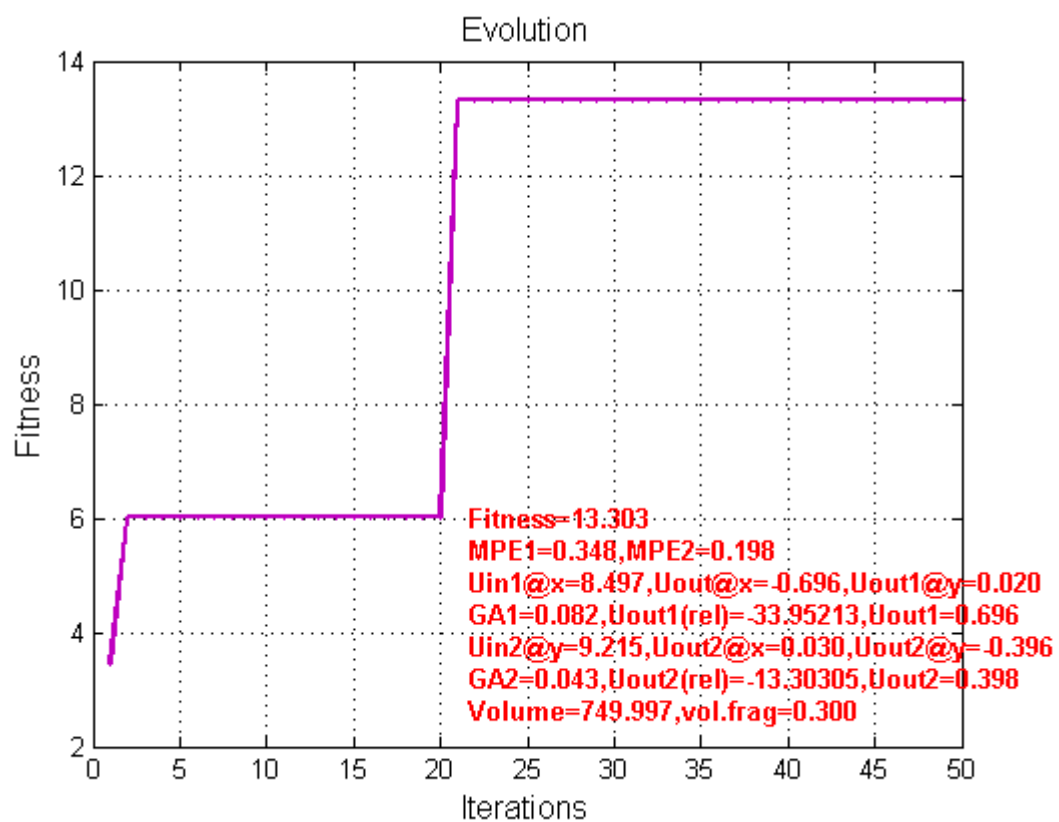
Σχήμα 7.7: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 2^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)



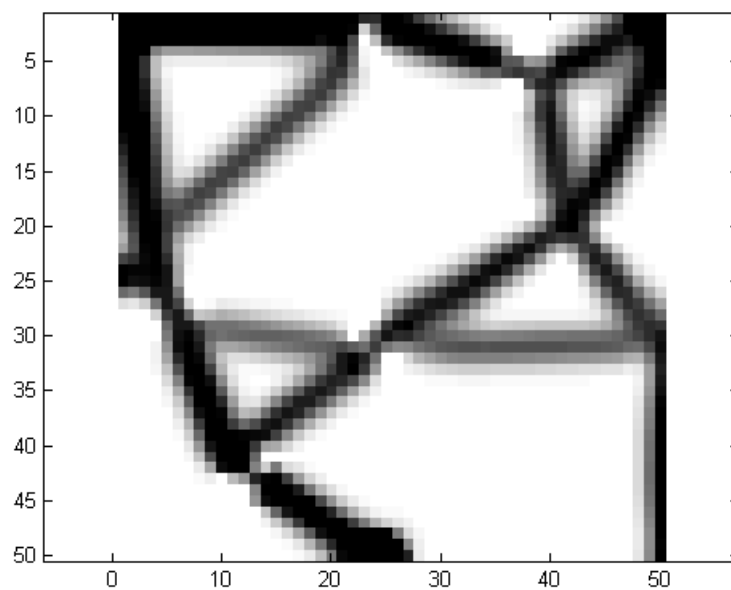
Σχήμα 7.8: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 2^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)



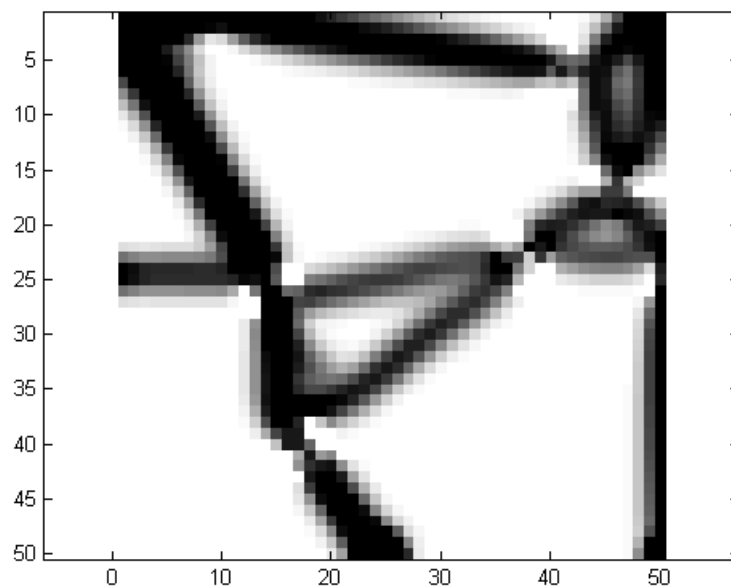
Σχήμα 7.9: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)



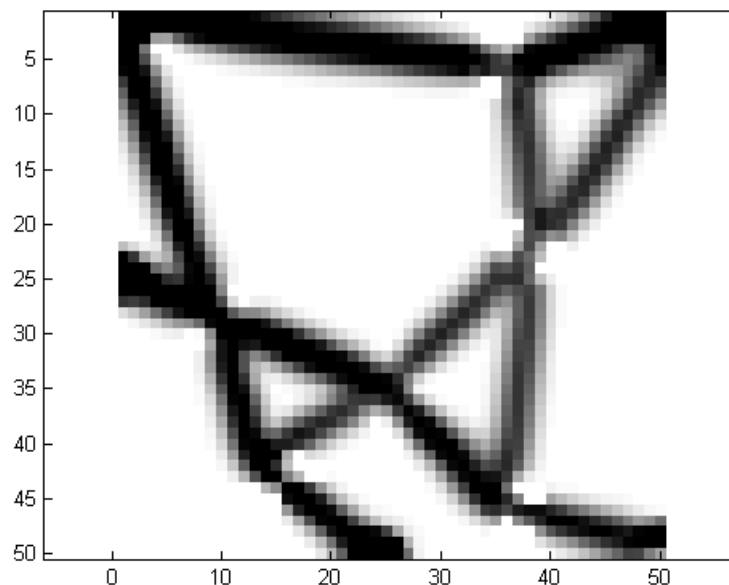
Σχήμα 7.10: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, Εξέλιξη αλγορίθμου



Σχήμα 7.11: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 1 έως 1



Σχήμα 7.12: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 2 έως 20



Σχήμα 7.13: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με DEA, Ενδιάμεση κατανομή για επαναλήψεις από 21 έως 50

7.2.4 Παράμετροι προβλήματος για αλγόριθμο βελτιστοποίησης με χρήση νοημοσύνης σμήνους σωματιδίων και αριθμητικά στοιχεία - PSO

Για το συγκεκριμένο πρόβλημα οι παράμετροι του αλγόριθμου τοπικής έρευνας είναι ακριβώς οι ίδιοι όπως στην περίπτωση του διαφορικού εξελικτικού. Για τον PSO, το σμήνος αποτελείται από 20 σωματίδια, όπου κάθε σωματίδιο μπορεί να εκτελέσει 50 τοπικές αναζητήσεις για κάθε βήμα επανάληψης. Ο συνολικός αριθμός των αναζητήσεων για όλα τα σωματίδια είναι 1000. Οι υπόλοιπες παράμετροι του προβλήματος παρουσιάζονται συνοπτικά στον πίνακα 7.3.

Παράμετροι προβλήματος για δύο περιπτώσεις φόρτισης με PSO	
Μέγεθος σμήνους	20
Σύνολο τοπικών αναζητήσεων	50
Συντελεστές επιτάχυνσης $c_1 = c_2$	2
Συντελεστές αδράνειας w_{max} και w_{min}	0.9 και 0.1
Διακριτοποίηση	50x50
Αριθμός μεταβλητών	2500
Βαθμοί ελευθερίας	$2 \times (50+1) \times (50+1) = 5202$
Επαναλήψεις τοπικού αλγορίθμου	80
Περιορισμός όγκου	30%

Πίνακας 7.3: Παράμετροι προβλήματος για 2 περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO

Αποτελέσματα για δύο περιπτώσεις φόρτισης με PSO		
1 ^η Περίπτωση Φόρτισης	Τιμή ολικής αντικ. συνάρτησης	13.348
	Μετακίνηση στον Χ, $u_{1,out}^{@X}$	-0.326
	Μετακίνηση στον Υ, $u_{1,out}^{@Y}$	-0.004
	Τιμή αντικ. συνάρτησης $\frac{u_{1,out}^{@X}}{u_{1,out}^{@Y}}$	76.655
	Γεωμ. Πλεονέκτημα (GA)	3.4%
2 ^η Περίπτωση Φόρτισης	Μετακίνηση στον Χ, $u_{2,out}^{@X}$	0.043
	Μετακίνηση στον Υ, $u_{2,out}^{@Y}$	-0.580
	Τιμή αντικ. συνάρτησης $\frac{u_{2,out}^{@Y}}{u_{2,out}^{@X}}$	13.348
	Γεωμ. Πλεονέκτημα (GA)	6.1%

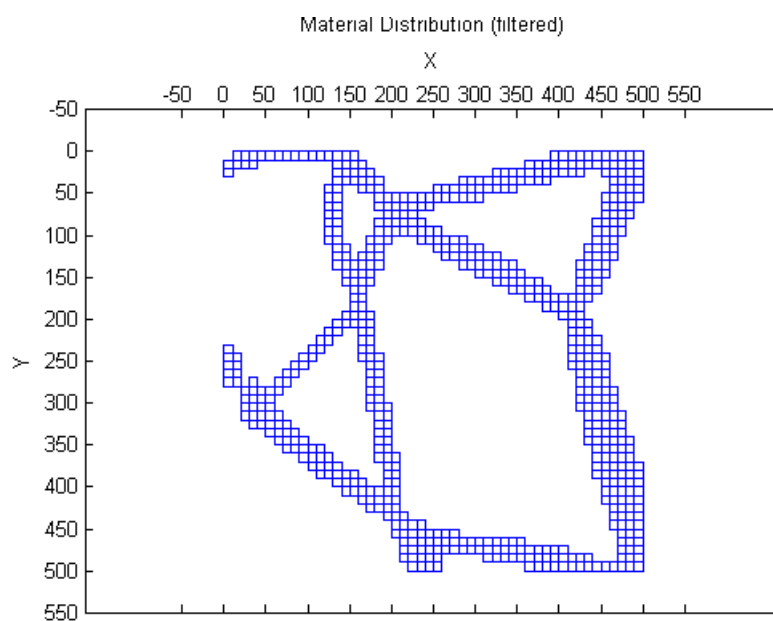
Πίνακας 7.4: Αποτελέσματα για δύο περιπτώσεις φόρτισης και γενικό αλγόριθμο βελτιστοποίησης: PSO

7.2.5 Αποτελέσματα για δύο περιπτώσεις φόρτισης και PSO

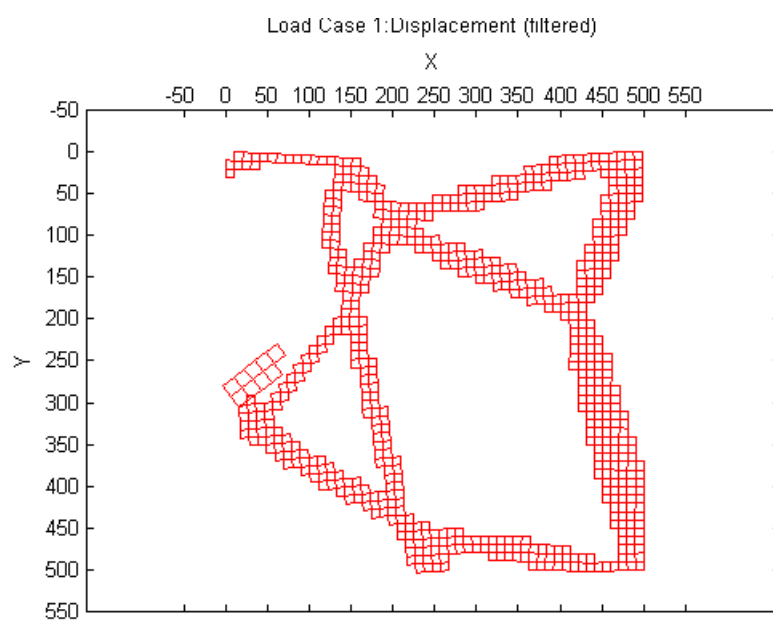
Στον πίνακα 7.4 παρουσιάζονται τα αποτελέσματα για τον ασύμμετρο αντιστροφέα δύναμης για περιπτώσεις φόρτισης με χρήση αλγορίθμου βελτιστοποίησης με χρήση νοημοσύνης σμήνους σωματιδίων - PSO:



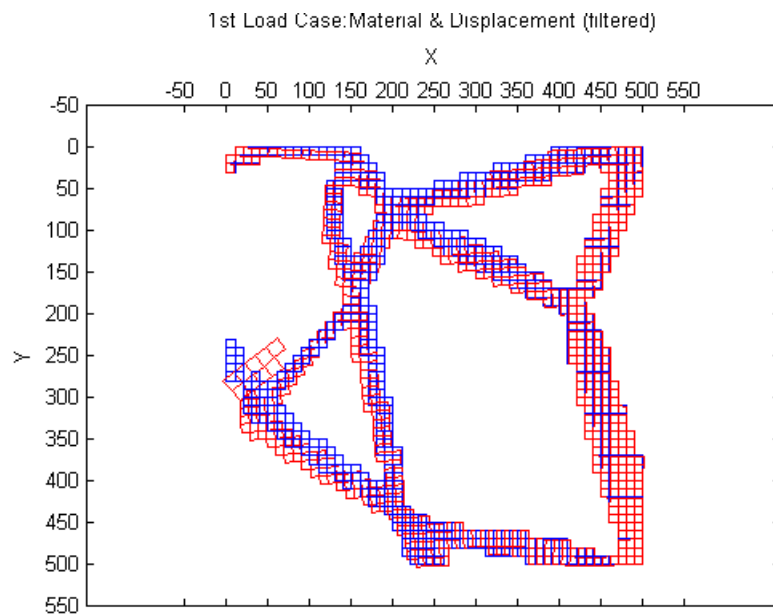
Σχήμα 7.14: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, Τελική κατανομή



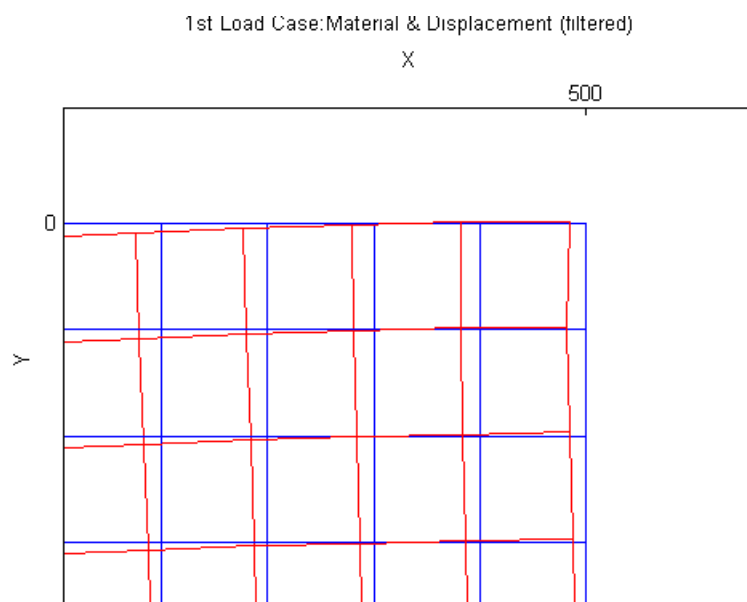
Σχήμα 7.15: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, Τελική κατανομή (με φίλτρο)



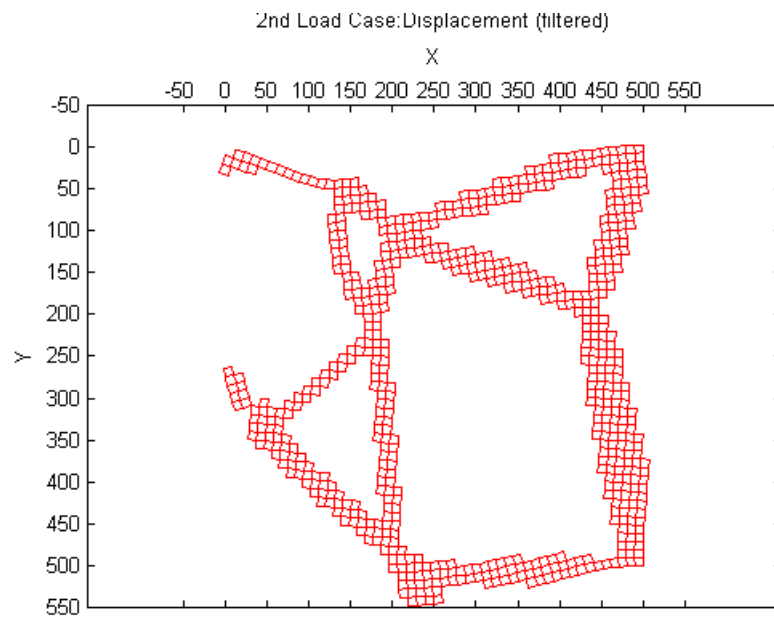
Σχήμα 7.16: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 1^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)



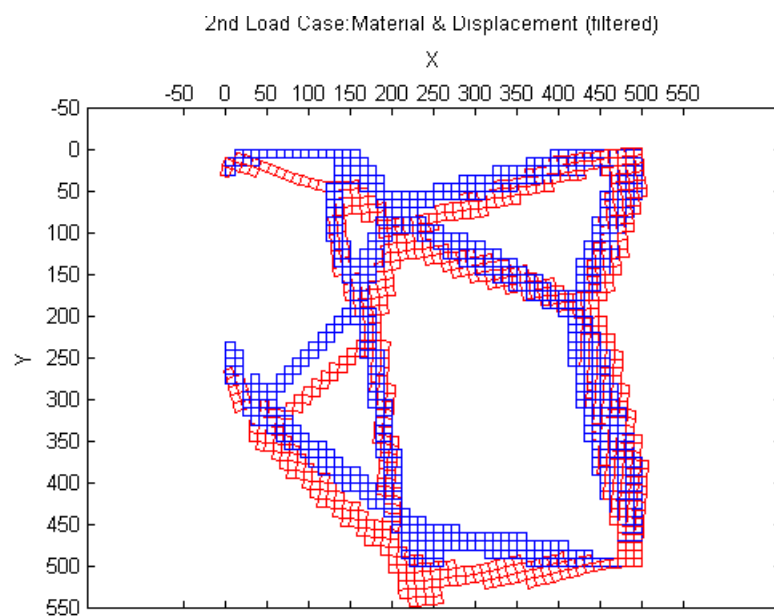
Σχήμα 7.17: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 1^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)



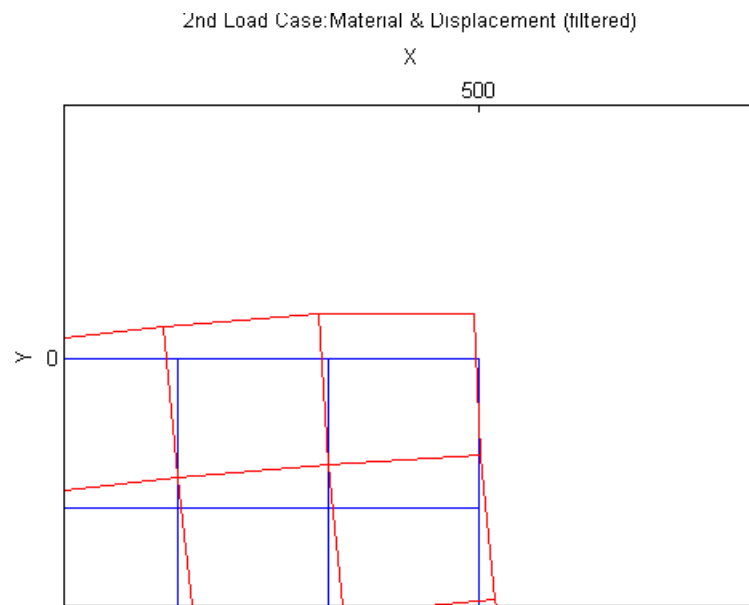
Σχήμα 7.18: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 1^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)



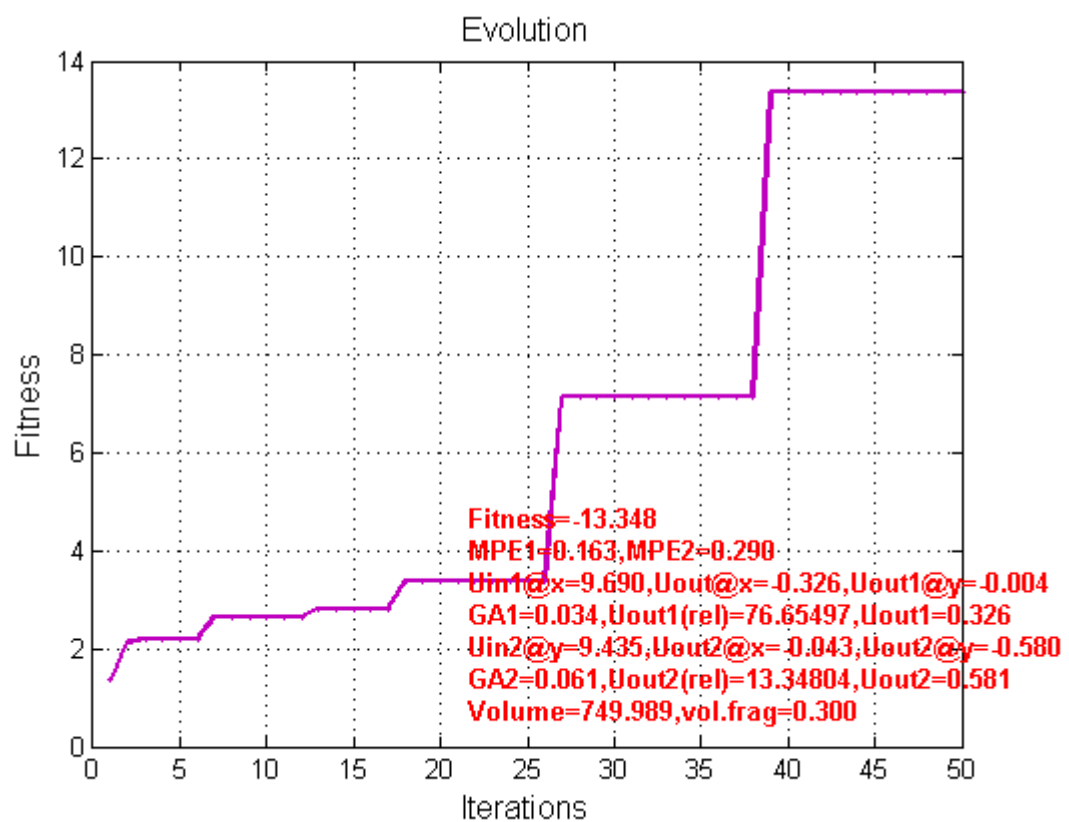
Σχήμα 7.19: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 2^η Περίπτωση φόρτισης Μετακινήσεις (με φίλτρο)



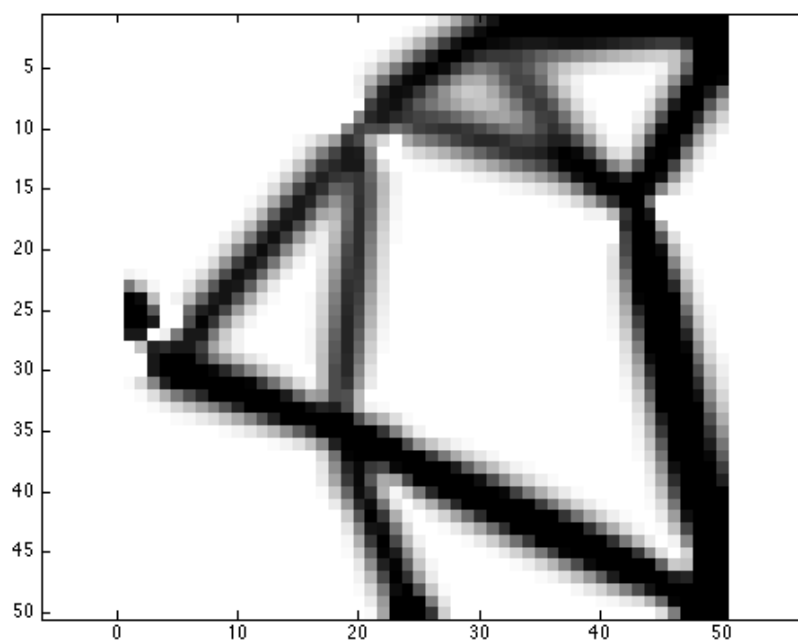
Σχήμα 7.20: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, 2^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις (με φίλτρο)



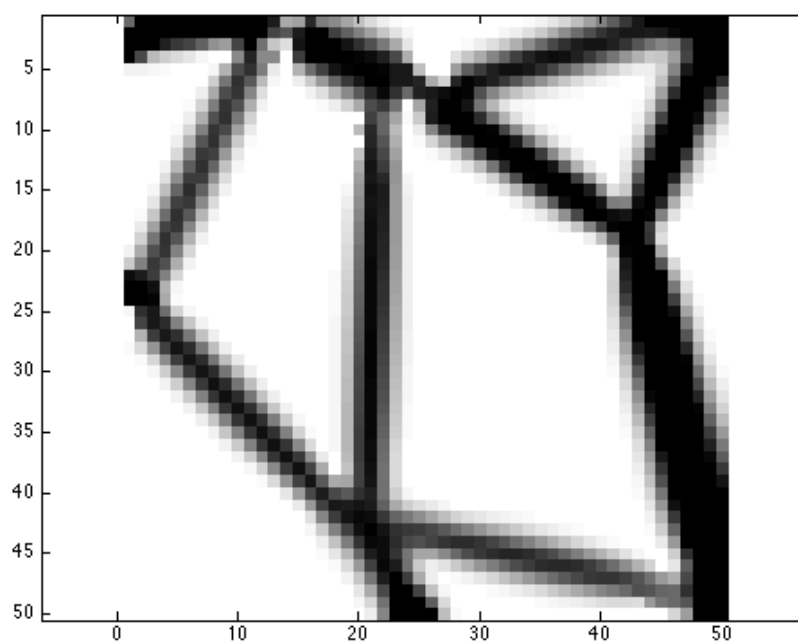
Σχήμα 7.21: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με DEA, 1^η Περίπτωση φόρτισης Τελική κατανομή και Μετακινήσεις, λεπτομέρεια (με φίλτρο)



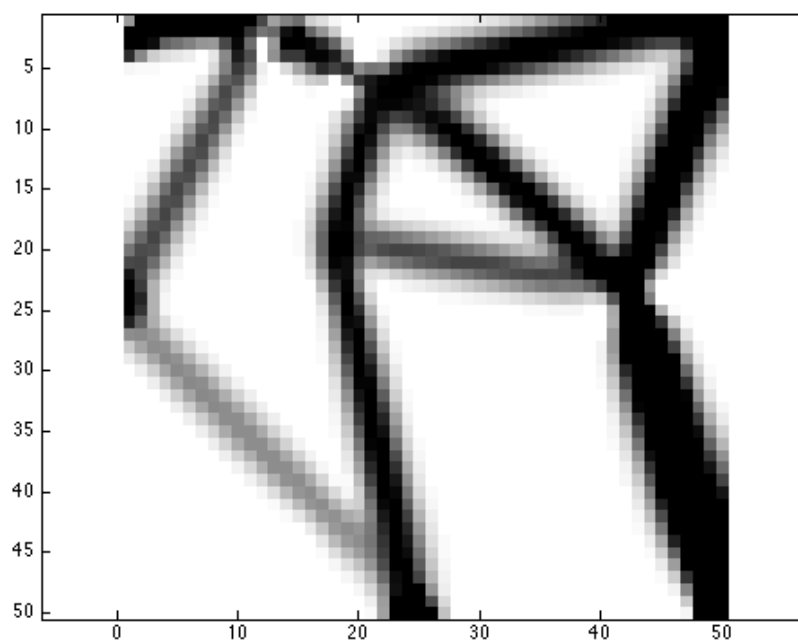
Σχήμα 7.22: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για δύο περιπτώσεις φόρτισης με PSO, Εξέλιξη αλγορίθμου



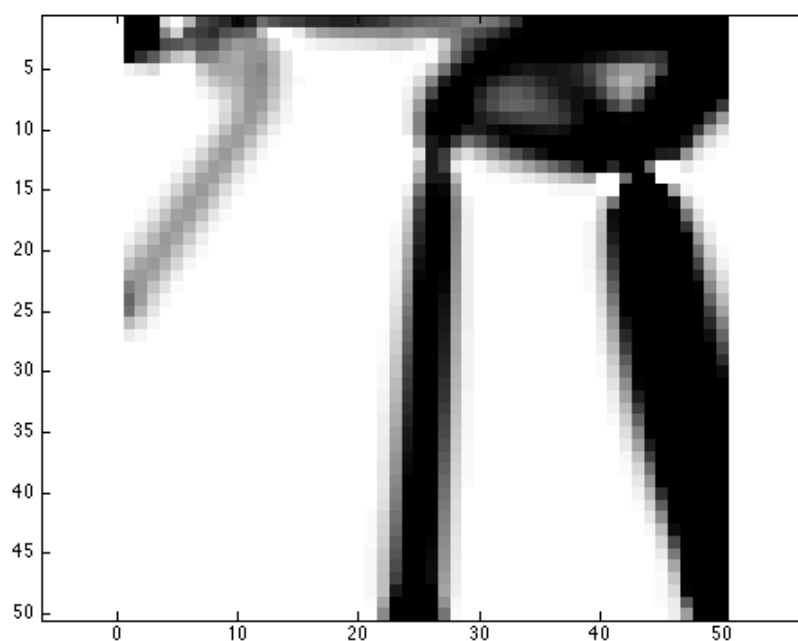
Σχήμα 7.23: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για την πρώτη επανάληψη



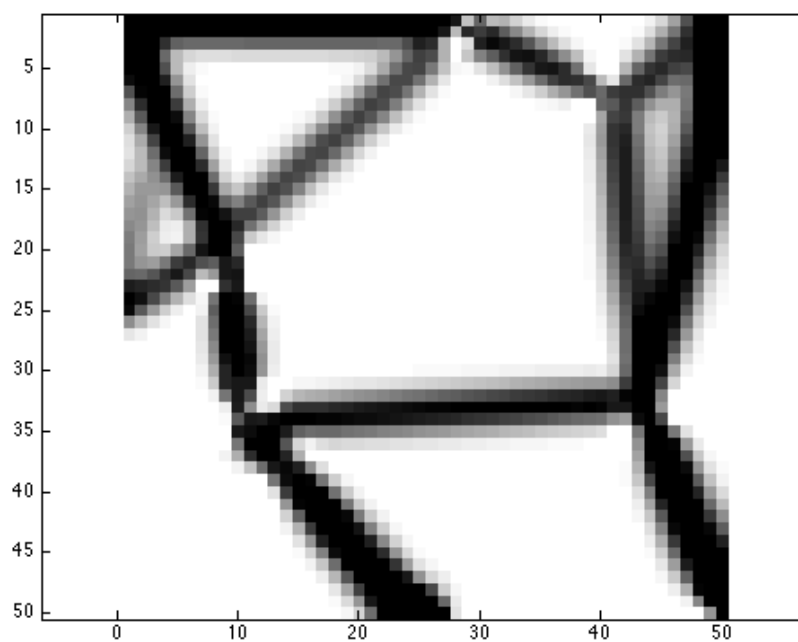
Σχήμα 7.24: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για την δεύτερη επανάληψη



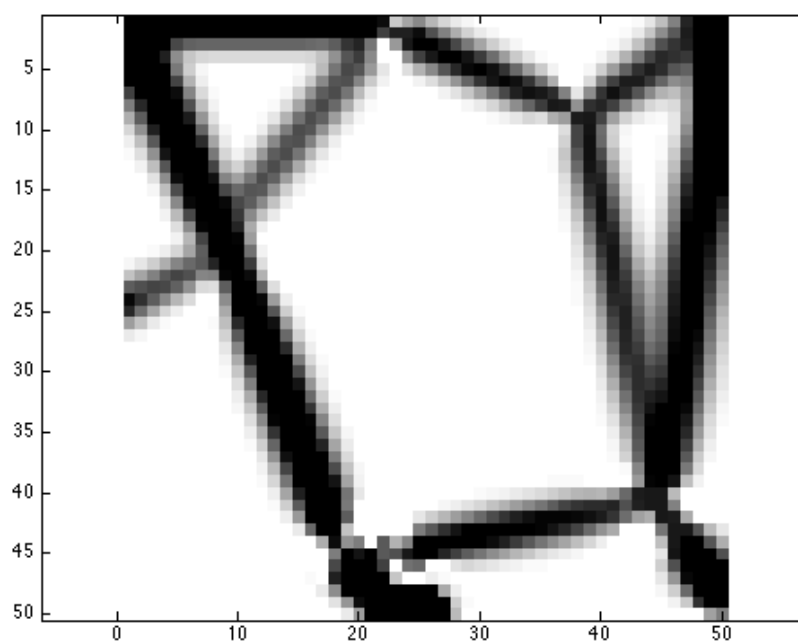
Σχήμα 7.25: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 3 έως 6



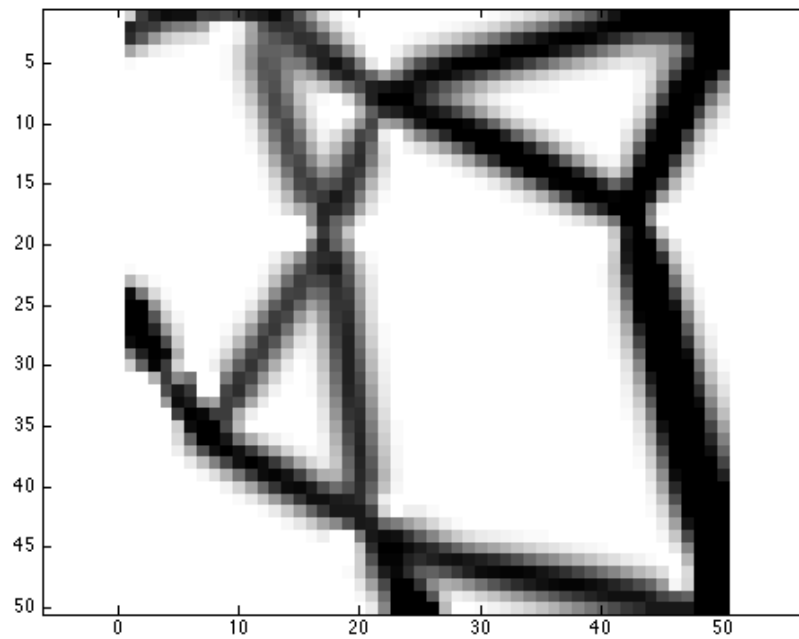
Σχήμα 7.26: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 7 έως 12



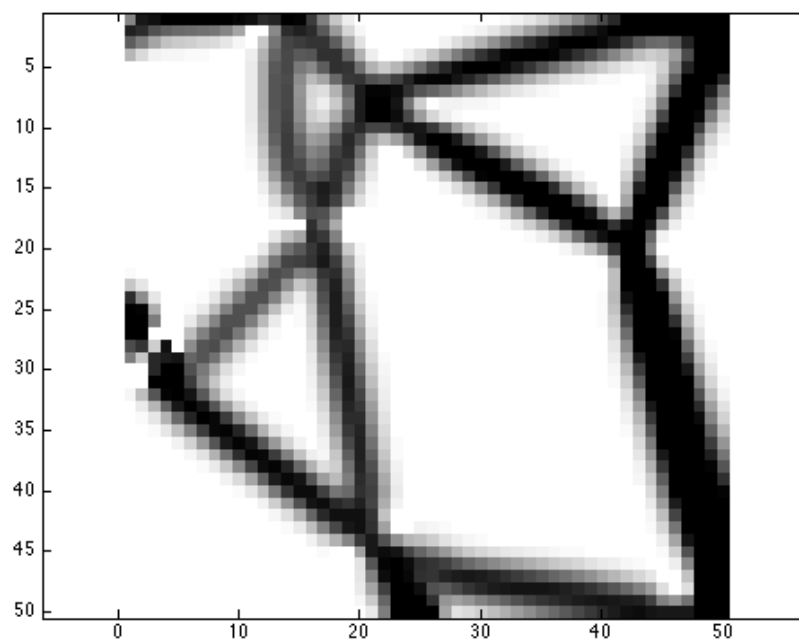
Σχήμα 7.27: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 13 έως 17



Σχήμα 7.28: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 18 έως 26



Σχήμα 7.29: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 27 έως 38



Σχήμα 7.30: Ασύμμετρος μηχανισμός αντιστροφής δύναμης για μια περίπτωση φόρτισης με PSO, Ενδιάμεση κατανομή για επαναλήψεις από 39 έως 50

7.3 Σύγκριση αποτελεσμάτων DEA και PSO για δύο περιπτώσεις φόρτισης

Από τα αποτελέσματα φαίνεται ότι ο αλγόριθμος PSO έδωσε λίγο καλύτερη λύση σε σχέση με τον DEA. Οι τιμές των αντικειμενικών συναρτήσεων είναι 13.348 και 13.303 αντίστοιχα με πολύ διαφορετική τελική κατανομή. Και στις δύο περιπτώσεις φόρτισης υπερτέρησε ο PSO έναντι όλων των κριτηρίων, εκτός από το γεωμετρικό πλεονέκτημα για την πρώτη περίπτωση φόρτισης με τον DEA δίνει τιμή 8.2% σε σχέση με 3.4% που έδωσε ο PSO, αν και το συγκεκριμένο κριτήριο δεν συμμετέχει στην αντικειμενική συνάρτηση. Οι σύγκριση των αποτελεσμάτων μεταξύ DEA και PSO παρουσιάζονται στον πίνακα 7.5. Στον πίνακα 7.6 παρουσιάζονται οι χρόνοι αποπεράτωσης για δύο διαφορετικά συστήματα.

			Σύγκριση αποτελεσμάτων DEA και PSO	
			DEA	PSO
Τελική κατανομή				
Γενική αντικ. συνάρτηση			13.303	13.348
1 ^η Περ. φόρτισης	$\frac{u_{1,out}^{@X}}{u_{1,out}^{@Y}}$		33.952	76.655
	$u_{1,out}^{@X}$		-0.696	-0.326
	$u_{1,out}^{@Y}$		-0.020	-0.004
	GA		8.2%	3.4%
1 ^η Περ. φόρτισης	$\frac{u_{2,out}^{@Y}}{u_{2,out}^{@X}}$		13.303	13.348
	$u_{2,out}^{@X}$		0.030	0.043
	$u_{2,out}^{@Y}$		-0.348	-0.580
	GA		4.3%	6.1%

Πίνακας 7.5: Σύγκριση αποτελεσμάτων DEA και PSO για δύο περιπτώσεις φόρτισης

Σύγκριση Χρόνων Αποπεράτωσης		
CPU	1x Intel Core 2 Duo E4300@1.8Ghz	2x Intel Quad Core Xeon 5420@2.5Ghz
Χρήση CPU	2/2	4/8
OS	MS Windows Vista Business 32bit	MS Windows XP 64bit
Μνήμη	2Gb	4Gb
50x50	16-18 hours	6.6 - 7 hours

Πίνακας 7.6: Σύγκριση χρόνων αποπεράτωσης για 50x50 διακριτοποίηση για 2 περιπτώσεις φόρτισης

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ

8.1 Σύνοψη της εργασίας

Στην παρούσα εργασία διερευνήθηκε η εφαρμογή τεχνικών ολικής βελτιστοποίησης για επίλυση προβλημάτων με μεγάλο αριθμό μεταβλητών, όπως προβλήματα τοπολογικής βελτιστοποίησης για σύνθεση κατασκευών και μηχανισμών. Συγκεκριμένα μελετήθηκε η σύνθεση εύκαμπτων μηχανισμών για μία και δύο περιπτώσεις φορτίσεων, με κύριο στόχο την εύρεση ενός ολικού βέλτιστου για κάθε περίπτωση. Διαπιστώθηκε ότι η χρήση αλγόριθμου τοπικής έρευνας για τέτοιου είδους προβλήματα βρίσκουν αρκετά διαφορετικές λύσεις ανάλογα με την αρχική κατάσταση έναρξης. Το υβριδικό σχήμα βελτιστοποίησης που παρουσιάζετε στην εργασία αυτή, φιλοδοξεί βρει το ολικό βέλτιστο για αυτά τα προβλήματα.

8.2 Συμπεράσματα

Τα συμπεράσματα που προκύπτουν από την μελέτη των δύο παραδειγμάτων σύνθεσης μηχανισμών είναι τα παρακάτω:

- Το υβριδικό σχήμα βελτιστοποίησης δίνει λύσεις,
- Και για τα δύο παραδείγματα, ο PSO έδωσε καλύτερα αποτελέσματα από τον DEA, χαρακτηριστικό που τον προκρίνει έναντι του DEA για περαιτέρω έρευνα, και χρήση.
- Η υβριδική μέθοδος είναι αρκετά αργή, λόγω του αλγορίθμου τοπικής έρευνας που χρησιμοποιείται για την κάθε επίλυση. Αυτό συμβαίνει λόγω του αυξημένου υπολογιστικού κόστους που έχει η μέθοδος αυτή για αρκετά μεγάλη διακρίτοποίηση. Εννοείτε ότι η χρήση μόνο αλγορίθμων γενικής βελτιστοποίησης θα ήταν πολύ πιο αργή διαδικασία, που προϋποθέτει και την χρήση ειδικών τεχνικών για την αντιμετώπιση προβλημάτων.

8.3 Περαιτέρω έρευνα

Τα συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας εργασίας οδηγούν σε σκέψεις για περαιτέρω επεκτάσεις που μπορούν να πραγματοποιηθούν στο μέλλον.

Όσο αφορά την ταχύτητα σύγκλισης του αλγορίθμου η χρήση Η/Υ με πολλούς επεξεργαστές μπορεί δώσει πιο γρήγορες λύσεις. Με την χρήση συστημάτων με πολλούς επεξεργαστές. Μια σκέψη θα μπορούσε να είναι η εκπαίδευση ενός νευρωνικού δικτύου για την πρόβλεψη της αντικειμενικής συνάρτησης ώστε να μειωθεί ο χρόνος επίλυσης. Η εκπαίδευση του νευρωνικού

δικτύου μπορεί να γίνει παράλληλα με την εξέλιξη του αλγορίθμου. Ένα σημαντικό *bottleneck* στην επίλυση είναι η διαδικασία σύνθεσης του πίνακα δυσκαμψίας για κάθε ένα βήμα. Η χρήση νέων τεχνικών για την σύνθεση του πίνακα δυσκαμψίας πλέγματος θα μπορούσε να μειώσει και άλλο τον χρόνο επίλυσης.

Στην παρούσα εργασία έγινε χρήση γραμμικών πεπερασμένων στοιχείων. Κάτι τέτοιο αποτελεί σχετικό μειονέκτημα για την μελέτη σύνθεσης μηχανισμών γιατί οι μετακινήσεις είναι αρκετά μικρές. Με την χρήση μη γραμμικών στοιχείων δίνεται η δυνατότητα σύνθεσης μηχανισμών για μεγάλες μετακινήσεις και παραμορφώσεις. Η χρήση μη γραμμικών στοιχείων αυξάνει αρκετά το υπολογιστικό κόστος γιατί έτσι γίνεται επίλυση μη γραμμικών προβλημάτων βελτιστοποίησης. Η χρήση μην γραμμικών στοιχείων θα μπορούσε να χρησιμοποιηθεί για σύνθεση εύκαμπτων μηχανισμών που θα μπορούσαν να μεταδώσουν την κίνηση σε μια συγκεκριμένη τροχιά (*path generation compliant mechanisms*). Επίσης η παρούσα εργασία θα μπορούσε να επεκταθεί για σύνθεση πολυμορφικών μηχανισμών και κατασκευών σε συνδυασμό με προβλήματα επαφής, ακόμα και για επίλυση προβλημάτων συνδυασμένων πεδίων (*multi-physics*) όπως σύνθεση μικρο-ηλεκτρο-μηχανισμών (MEMS).

Ακόμα και στο πρόβλημα για μια περίπτωση φόρτισης, τα προβλήματα βελτιστοποίησης που προκύπτουν είναι πολλά. Εκτός από την μεγιστοποίηση της μετατόπισης στο σημείο εξόδου και τον έλεγχο μετακίνησης, η χρήση κριτηρίων όπως η ελαχιστοποίηση των τάσεων που προκύπτουν από τις φορτίσεις και την κόπωση του μηχανισμού, ή η μεγιστοποίηση του γεωμετρικού πλεονεκτήματος είναι δυνατή. Στην περίπτωση των πολυμορφικών μηχανισμών ο αριθμός των κριτηρίων αυξάνει αρκετά. Άρα μπορεί να γίνει χρήση αλγορίθμων γενικής βελτιστοποίησης πολλαπλών κριτηρίων. Αυτό μπορεί να υλοποιηθεί με χρήση πολυκριτηρίου Διαφορικού Εξελικτικού Αλγορίθμου (*Multi-Objective Differential Evolution Algorithm*), και αλγορίθμου βελτιστοποίησης με χρήση νοημοσύνης πολλαπλών σημνών σωματιδίων (*Multiple-PSO*). Επίσης είναι δυνατή παραμετρική διερεύνηση των παραπάνω αλγορίθμων.

ΜΕΡΟΣ IV

ΠΑΡΑΡΤΗΜΑΤΑ



ΠΑΡΑΡΤΗΜΑ Α

Α.1 Κώδικας για Τοπολογική Βελτιστοποίηση Κατασκευών, top.m

```
1  %%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000
2  %%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
3  function top(nelx,nely,volfrac,penal,rmin)
4  % INITIALIZE
5  % x(1:nely,1:nelx) = rand;
6  x=0.001+(1-0.001)*rand(nely,nelx);
7  loop = 0;
8  change = 1.;
9  % START ITERATION
10 while change > 0.01
11     loop = loop + 1;
12     xold = x;
13 % FE-ANALYSIS
14 [U]=FE(nelx,nely,x,penal);
15 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
16 [KE] = lk;
17 c = 0.;
18 for ely = 1:nely
19     for elx = 1:nelx
20         n1 = (nely+1)*(elx-1)+ely;
21         n2 = (nely+1)* elx +ely;
22         Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
23         c = c + x(ely,elx)^penal*Ue'*KE*Ue;
24         dc(ely,elx) = -penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
25     end
26 end
27 % FILTERING OF SENSITIVITIES
28 [dc] = check(nelx,nely,rmin,x,dc);
29 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA MEIHDOD
30 [x] = OC(nelx,nely,x,volfrac,dc);
31 % PRINT RESULTS
32 change = max(max(abs(x-xold)));
33 disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
34       ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
35       ' ch.: ' sprintf('%6.3f',change) ])
36 % PLOT DENSITIES
37 colormap(gray);imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
```

```

38 end
39 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40 function [xnew]=OC(nelx , nely , x , volfrac , dc)
41 l1 = 0; l2 = 100000; move = 0.2;
42 while (l2-l1 > 1e-4)
43     lmid = 0.5*(l2+l1);
44     xnew = max(0.001 , ...
45         max(x-move, min(1 , min(x+move, x.*sqrt(-dc./lmid)))));
46     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
47         l1 = lmid;
48     else
49         l2 = lmid;
50     end
51 end
52 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
53 function [dcn]=check(nelx , nely , rmin , x , dc)
54 dcn=zeros(nely , nelx);
55 for i = 1:nelx
56     for j = 1:nely
57         sum=0.0;
58         for k = max(i-floor(rmin) , 1):min(i+floor(rmin) , nelx)
59             for l = max(j-floor(rmin) , 1):min(j+floor(rmin) , nely)
60                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
61                 sum = sum+max(0 , fac);
62                 dcn(j , i) = dcn(j , i) + max(0 , fac)*x(l , k)*dc(l , k);
63             end
64         end
65         dcn(j , i) = dcn(j , i)/(x(j , i)*sum);
66     end
67 end
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FE-ANALYSIS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
69 function [U]=FE(nelx , nely , x , penal)
70 [KE] = lk;
71 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
72 F = sparse(2*(nely+1)*(nelx+1), 1); U = zeros(2*(nely+1)*(nelx+1), 1);
73 for elx = 1:nelx
74     for ely = 1:nely
75         n1 = (nely+1)*(elx-1)+ely;
76         n2 = (nely+1)* elx +ely;
77         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
78         K(edof , edof) = K(edof , edof) + x(ely , elx)^penal*KE;
79     end
80 end
81 % DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
82 F(2 , 1) = -1;
83 fixeddofs = union([1:2:2*(nely+1)], [2*(nelx+1)*(nely+1)]);
84 alldofs = [1:2*(nely+1)*(nelx+1)];
85 freedofs = setdiff(alldofs , fixeddofs);
86 % SOLVING
87 U(freedofs , :) = K(freedofs , freedofs) \ F(freedofs , :);
88 U(fixeddofs , :) = 0;
89 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
90 function [KE]=lk
91 E = 1.;
92 nu = 0.3;
93 k=[ 1/2-nu/6    1/8+nu/8  -1/4-nu/12  -1/8+3*nu/8  ...
94     -1/4+nu/12  -1/8-nu/8    nu/6      1/8-3*nu/8];
95 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
96                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
97                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
98                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)]

```

[illegible]

A'.2 Τοπολογική Βελτιστοποίηση σύνθεσης εύκαμπτων μηχανισμών, μια περίπτωση φόρτισης

```

1  %%%% A 104 LINE TOPOLOGY OPTIMIZATION CODE compliant mechanisms
2  %%%% BY OLE SIGMUND, JANUARY 2000
3  %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
4  function topm(nelx , nely , volfrac , penal , rmin)
5  % INITIALIZE
6  x(1:nely , 1:nelx) = volfrac;
7  loop = 0;
8  change = 1.;
9  % START ITERATION
10 dc=zeros(nely , nelx);
11 while change > 0.01
12     loop = loop + 1;
13     xold = x;
14 % FE-ANALYSIS
15     [U,c]=FE(nelx , nely , x , penal);
16 % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
17     [KE] = lk;
18     for ely = 1:nely
19         for elx = 1:nelx
20             n1 = (nely+1)*(elx-1)+ely;
21             n2 = (nely+1)* elx +ely;
22             Ue1 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
23             Ue2 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],2);
24             dc(ely , elx) = penal*x(ely , elx)^(penal-1)*Ue1'*KE*Ue2;
25         end
26     end
27 % FILTERING OF SENSITIVITIES
28     [dc] = check(nelx , nely , rmin , x , dc);
29 % DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
30     [x] = OC(nelx , nely , x , volfrac , dc);
31 % PRINT RESULTS
32     change = max(abs(x-xold));
33     disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
34          ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
35          ' ch.: ' sprintf('%6.3f',change) ])
36 % PLOT DENSITIES
37     colormap(gray); imagesc(-x); axis equal; axis tight; axis off; pause(1e-6);
38 end
39 %%%%%%%%%% OPTIMALITY CRITERIA UPDATE %%%%%%%%%%
40 function [xnew]=OC(nelx , nely , x , volfrac , dc)
41 l1 = 0; l2 = 100000; move = 0.1;
42 while ((l2-l1)/(l2+l1) > 1e-4) && l2>1e-40
43     lmid = 0.5*(l2+l1);
44     xnew = max(0.001 , ...
45         max(x-move , min(1. , min(x+move , x.*(max(1e-10,(-dc./lmid))).^0.3))));
46     if sum(sum(xnew)) - volfrac*nelx*nely > 0;
47         l1 = lmid;
48     else
49         l2 = lmid;
50     end
51 end
52 %%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
53 function [dcn]=check( nelx , nely , rmin , x , dc)
54 dcn=zeros( nely , nelx );
55 for i = 1:nelx

```

```

56     for j = 1:nely
57         sum=0.0;
58         for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
59             for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
60                 fac = rmin-sqrt((i-k)^2+(j-l)^2);
61                 sum = sum+max(0,fac);
62                 dcn(j,i) = dcn(j,i) + max(0,fac)*x(1,k)*dc(1,k);
63             end
64         end
65         dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
66     end
67 end
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
69 function [U,Uout]=FE(nelx,nely,x,penal)
70 [KE] = lk;
71 K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
72 F = sparse(2*(nely+1)*(nelx+1),2); U = sparse(2*(nely+1)*(nelx+1),2);
73 for elx = 1:nelx
74     for ely = 1:nely
75         n1 = (nely+1)*(elx-1)+ely;
76         n2 = (nely+1)* elx +ely;
77         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
78         K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
79     end
80 end
81 % DEFINE LOADS AND SUPPORTS (HALF FORCE INVERTER)
82 % din=2*(floor((nely+1)/2)+1)-1;
83 din=1;
84 dout=2*nelx*(nely+1) + 1;
85 F(din,1)=1;
86 F(dout,2)=-1;
87 K(din,din)=K(din,din) + 0.1;
88 K(dout,dout)=K(dout,dout) + 0.1;
89 fixeddofs = union([2:2*(nely+1):2*(nely+1)*(nelx+1)], ...
90                  [2*(nely+1):-1:2*(nely+1)-3] );
91 alldofs = [1:2*(nely+1)*(nelx+1)];
92 freedofs = setdiff(alldofs,fixeddofs);
93 % SOLVING
94 U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
95 U(fixeddofs,:)= 0;
96 Uout=U(dout,1);
97 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98 function [KE]=lk
99 E = 1.;
100 nu = 0.3;
101 k=[ 1/2-nu/6    1/8+nu/8  -1/4-nu/12  -1/8+3*nu/8 ...
102    -1/4+nu/12  -1/8-nu/8   nu/6      1/8-3*nu/8];
103 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
104                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
105                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
106                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
107                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
108                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
109                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
110                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
111 %
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 % This Matlab code was written by Ole Sigmund, Department of Solid %
114 % Mechanics, Technical University of Denmark, DK-2800 Lyngby, Denmark %
115 % Please sent your comments to the author: sigmund@fam.dtu.dk %
116 %

```


A'3 Τοπολογική Βελτιστοποίηση σύνθεσης εύκαμπτων μηχανισμών, Δύο περιπτώσεις φορτίσεων

```

1  %%%% A 104 LINE TOPOLOGY OPTIMIZATION CODE compliant mechanisms
2  %%%% BY OLE SIGMUND, JANUARY 2000
3  %%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
4  %%%% CODE MODIFIED FOR TWO LOAD CASES, Febrouary 2009, By Kaminakis Nikos
5  %%%% nkaminakis@gmail.com
6  function topm_2loads(nelx,nely,volfrac,penal,rmin)
7  rand('twister',sum(100*clock));
8  % INITIALIZE
9  %x(1:nely,1:nelx) = volfrac; %Uniform distribution for the initialization
10                                     %of the domain
11  x=0.001+0.999*rand(nely,nelx);
12  loop = 0;
13  change = 1.;
14  % START ITERATION
15  dc=zeros(nely,nelx);
16  while change > 0.01
17      loop = loop + 1;
18      xold = x;
19      % FE-ANALYSIS
20      [U,c,c1,c2]=FE(nelx,nely,x,penal);
21      % OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
22      [KE] = lk;
23      for ely = 1:nely
24          for elx = 1:nelx
25              n1 = (nely+1)*(elx-1)+ely;
26              n2 = (nely+1)* elx +ely;
27              Ue1 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
28              Ue2 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],2);
29              Ue3 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],3);
30              Ue4 = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],4);
31              dc(ely,elx) = 0.5*penal*x(ely,elx)^(penal-1)*Ue1'*KE*Ue2+...
32                          0.5*penal*x(ely,elx)^(penal-1)*Ue3'*KE*Ue4;
33          end
34      end
35      % FILTERING OF SENSITIVITIES
36      [dc] = check(nelx,nely,rmin,x,dc);
37      % DESIGN UPDATE BY THE OPTIMALITY CRITERIA MEIHDOD
38      [x] = OC(nelx,nely,x,volfrac,dc);
39      % PRINT RESULTS
40      change = max(max(abs(x-xold)));
41      disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
42          ' Uout1_@x: ' sprintf('%10.4f',c1) ...
43          ' Uout2_@y: ' sprintf('%10.4f',c2) ...
44          ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
45          ' ch.: ' sprintf('%6.3f',change)])
46      % PLOT DENSITIES
47      colormap(gray);imagesc(-x);axis equal;axis tight; axis off;pause(1e-6);
48  end
49  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
50  function [xnew]=OC(nelx,nely,x,volfrac,dc)
51  l1 = 0; l2 = 100000; move = 0.1;
52  while ((l2-l1)/(l2+l1) > 1e-4) && l2>1e-40
53      lmid = 0.5*(l2+l1);
54      xnew = max(0.001,...
55          max(x-move,min(1.,min(x+move,x.*(max(1e-10,(-dc./lmid))).^0.3))));

```

```

56     if sum(sum(xnew)) - volfrac*nex*nely > 0;
57         l1 = lmid;
58     else
59         l2 = lmid;
60     end
61 end
62 %%%%%%%%%% MESH-INDEPENDENCY FILTER %%%%%%%%%%
63 function [dcn]=check( nex , nely , rmin , x , dc )
64 dcn=zeros( nely , nex );
65 for i = 1:nex
66     for j = 1:nely
67         sum=0.0;
68         for k = max(i-floor( rmin ) , 1):min( i+floor( rmin ) , nex )
69             for l = max( j-floor( rmin ) , 1):min( j+floor( rmin ) , nely )
70                 fac = rmin-sqrt( (i-k)^2+(j-l)^2 );
71                 sum = sum+max( 0 , fac );
72                 dcn( j , i ) = dcn( j , i ) + max( 0 , fac ) * x( l , k ) * dc( l , k );
73             end
74         end
75         dcn( j , i ) = dcn( j , i ) / ( x( j , i ) * sum );
76     end
77 end
78 %%%%%%%%%% FE-ANALYSIS %%%%%%%%%%
79 function [U,Uout,Uout1_x,Uout2_y]=FE( nex , nely , x , penal )
80 [KE] = lk;
81 K = sparse( 2*(nex+1)*(nely+1) , 2*(nex+1)*(nely+1) );
82 num_loads=2; % number of loads
83 F = sparse( 2*(nely+1)*(nex+1) , 2*num_loads ); %Allocate force matrix
84 U = sparse( 2*(nely+1)*(nex+1) , 2*num_loads ); %Allocate displacement matrix
85 for elx = 1:nex
86     for ely = 1:nely
87         n1 = (nely+1)*(elx-1)+ely;
88         n2 = (nely+1)* elx +ely;
89         edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
90         K(edof,edof) = K(edof,edof) + x(ely , elx)^penal*KE;
91     end
92 end
93 %% DEFINE LOADS AND SUPPORTS (HALF FORCE INVERTER)
94 % Definition of input and output points
95 din1=2*ceil( nely/2 )-1; %Point of input force 1
96 dout1=2*nex*(nely+1) + 1; %Point of output force1
97 din2=2*(ceil( nex/2 )*(nely+1)); %Point of input force 2
98 dout2=2*nex*(nely+1) + 2; %Point of output force2
99 % Definition of input and output forces
100 F(din1,1)=1; %direction of the 1st input load @ X+ direction
101 F(dout1,2)=-1; %direction of the 1st dummy load @ X- direction
102 F(din2,3)=1; %direction of the 2nd input load @ Y+ direction
103 F(dout2,4)=-1; %direction of the 2nd dummy load @ Y- direction
104 % define the stiffnesses of the springs in each input & output point
105 K(din1 , din1)=K(din1 , din1) + 0.1;
106 K(dout1 , dout1)=K(dout1 , dout1) + 0.1;
107 K(din2 , din2)=K(din2 , din2) + 0.1;
108 K(dout2 , dout2)=K(dout2 , dout2) + 0.1;
109 alldofs = (1:2*(nely+1)*(nex+1)); % all degrees of freedom
110 %% Definition of fixed degrees of freedom
111 % fixeddofs for the 1st load case
112 fixeddofs_1st_input_load=2*((nely/2)+1) ;
113 %dont allow motion for input load @
114 %this point @ Y(+,-) Direction
115 fixeddofs_1st_dummy_load=[];
116 fixeddofs_1st_load=[fixeddofs_1st_input_load fixeddofs_1st_dummy_load];

```

```

117 fixeddofs_temp1= union([1,2,3,4],...
118                        (2*(nely+1)*(nelx+1):-1:2*(nely+1)*(nelx+1)-3));
119                        %panw aristera kai katw de3ia gwnia
120 fixeddofs1=union(fixeddofs_temp1, fixeddofs_1st_load);
121 %% fixeddofs for the 2nd load case
122 fixeddofs_2nd_input_load=2*((nelx/2)+1)*(nely+1)-1;
123 fixeddofs_2nd_dummy_load=[];
124 fixeddofs_2nd_load=[fixeddofs_2nd_input_load fixeddofs_2nd_dummy_load];
125 fixeddofs_temp2=union([1,2,3,4],...
126                      (2*(nely+1)*(nelx+1):-1:2*(nely+1)*(nelx+1)-3));
127                      %panw aristera kai katw de3ia gwnia
128 fixeddofs2=union(fixeddofs_temp2, fixeddofs_2nd_load);
129 %% Solving 1st load case, and calculate some extra stuff
130 freedofs1 = setdiff(alldofs, fixeddofs1);
131 U(freedofs1,1:2) = K(freedofs1, freedofs1) \ F(freedofs1,1:2);
132 U(fixeddofs1,1:2)= 0;
133 Uin1_x=U(din1,1); %Input displacement of force1
134 Uout1_x=U(dout1,1); %%Output displacement of force1 @X direction
135 Uout1_y=U(dout1+1,1); %Output displacement of force1 @Y direction
136 GA1=abs(Uout1_x/Uin1_x); %Geometric Advantage of force1
137 Uout_rel1=Uout1_x/Uout1_y; %Output displacement @X/ Out displacement @Y
138 %maximize the displace @ X,
139 %minimize displacement @Y, so we maximize the
140 %fraction Uout1_x/Uout1_y
141 Uout1=sqrt(Uout1_x^2+Uout1_y^2);
142 MPE1=0.5*U(:,1)'*K*U(:,2); %Mutual Potential Energy for force1
143 %% Solving 2nd load case, and calculate some extra stuff
144 freedofs2 = setdiff(alldofs, fixeddofs2);
145 U(freedofs2,3:4) = K(freedofs2, freedofs2) \ F(freedofs2,3:4);
146 U(fixeddofs2,3:4)= 0;
147 Uin2_y=U(din2,3); %Input displacement of force2 @ Y direction
148 Uout2_y=U(dout2,3); %%Output displacement of force2 @Y direction
149 Uout2_x=U(dout2-1,3); %Output displacement of force2 @X direction
150 GA2=abs(Uout2_y/Uin2_y); %Geometric Advantage of force2
151 Uout_rel2=Uout2_y/Uout2_x; %Output displacement @X/ Out displacement @Y
152 %maximize the displace @ X,
153 %minimize displacement @Y, so we maximize the
154 %fraction Uout1_x/Uout1_y
155 Uout2=sqrt(Uout2_x^2+Uout2_y^2);
156 MPE2=0.5*U(:,3)'*K*U(:,4); %Mutual Potential Energy for force2
157 %% Calculate the fitness function = w1*Uout1_x + w2*Uout2_y
158 Uout=0.5*abs(Uout1_x) + 0.5*abs(Uout2_y);
159 %%%%%%%%%%% ELEMENT STIFFNESS MATRIX %%%%%%%%%%%
160 function [KE]=lk
161 E = 1.;
162 nu = 0.3;
163 k=[ 1/2-nu/6 1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
164    -1/4+nu/12 -1/8-nu/8 nu/6 1/8-3*nu/8];
165 KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
166                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
167                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
168                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
169                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
170                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
171                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
172                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
173 %
174 %%%%%%%%%%%
175 % This Matlab code was written by Ole Sigmund, Department of Solid %
176 % Mechanics, Technical University of Denmark, DK-2800 Lyngby, Denmark %
177 % Please sent your comments to the author: sigmund@fam.dtu.dk %

```


ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1998.
- [2] M. P. Bendsøe and O. Sigmund. *Topology Optimization, Theory, Methods and Applications*. Springer, 2004.
- [3] Martin P. Bendsøe and Ole Sigmund. *Optimization of Structural and Mechanical Systems*, chapter 6, pages 161–194. World Scientific, 2006.
- [4] A. Díaz and O. Sigmund. Checkerboard patterns in layout optimization. *Structural Optimization*, vii(10):40–45, 1995.
- [5] R. C. Eberhart and J Kennedy. A new optimizer using particle swarm theory. In *Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan*, pages 39–43, 1995.
- [6] Noboru Kikuchi Jinyong Joo, Sridhar Kota. Topological synthesis of compliant mechanisms using linear beam element. *Mechanics Based Design of Structures and Machines*, 28(4):245–280, 2000.
- [7] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ*, pages 1942–1948, 1995.
- [8] Jouni A. Lampinen Kenneth V. Price, Rainer M. Storn. *Differential Evolution - A Practical Approach to Global Optimization*. Springer, 2005.
- [9] A. G. M. Michell. The limit of economy of material in frame structures. *Philosophical Magazine*, 8(6):589–597, 1904.
- [10] Claus B.W Pedersen and Thomas Buhl. <http://www.topopt.dtu.dk/theory1/>, topology optimization (a one-semester project from the fall-semester of 1997).
- [11] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21:120–127, 2001.
- [12] Ole Sigmund. *Design of material structures using topology optimization*. PhD thesis, Department of Solid Mechanics, DTU, 1994.

- [13] Solver.com. Genetic and evolutionary algorithms, <http://www.solver.com/gabasics.htm>.
- [14] Price K.V. Storn R. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI, 1995.
- [15] K. Svanberg. The method of moving asymptotes - a new method of structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.
- [16] thefreedictionary.com. <http://encyclopedia.thefreedictionary.com/evolution>, 2001.
- [17] WGBH/NOVA. Evolution: Wgbh/nova science unit and clear blue sky productions, <http://pbs.org/wgbh/evolution>.
- [18] Wikipedia. <http://en.wikipedia.org/wiki/evolution>.
- [19] Wikipedia. http://en.wikipedia.org/wiki/particle_swarm_optimization.