ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΚΩΔΙΚΑ ΣΤΑΤΙΚΗΣ ΕΠΙΛΥΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΕΠΙΠΕΔΟΥ ΔΙΚΤΥΩΜΑΤΟΣ

Διπλωματική Εργασία

Μανουσέλης Εμμανουήλ

Επιβλέπων καθηγητής
Βουγιούκας Εμμανουήλ, Λέκτορας

Συνεπιβλέπων καθηγητής
Στάμος Αθανάσιος, Ε.ΔΙ.Π.

Αθήνα, Μάρτιος 2017

Copyright © Εμμανουήλ Μανουσέλης, 2017.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα της εργασίας μου κ. Εμμανουήλ Βουγιούκα, για τη σημαντική στήριξη και βοήθεια του. Θερμές ευχαριστίες θα ήθελα να αποδώσω στον κ. Αθανάσιο Στάμο, συνεπιβλέποντα της διπλωματικής εργασίας, για τη συνεχή παρακολούθηση της προόδου της εργασίας και την πολύτιμη καθοδήγηση του καθ’όλη τη διάρκεια της εκπόνησης της. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου, για την ηθική συμπαράσταση τους κατά την διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η δημιουργία κώδικα σε γλώσσα προγραμματισμού Fortran, για τη βελτιστοποίηση ενός τυχαίου διδιάστατου δικτυώματος ως προς το σχήμα (θέση των κόμβων), το μέγεθος (διαστασιολόγηση διατομών) και την τοπολογία του. Παράμετρος βελτιστοποίησης αποτελεί το βάρος της κατασκευής. Για τη στατική επίλυση του δικτυώματος χρησιμοποιείται η μέθοδος Άμεσης Δυσκαμψίας-Στιβαρότητας. Στη συγκεκριμένη εργασία, για τη διαστασιολόγηση των ράβδων χρησιμοποιούνται μόνο τετραγωνικές κοίλες διατομές. Για τη βελτιστοποίηση του δικτυώματος χρησιμοποιούνται δύο αλγόριθμοι, ο αλγόριθμος Μόντε Κάρλο (Monte Carlo) καθώς και ο στοχαστικός αλγόριθμος Προσομοιωμένης Ανόπτησης (Simulated Annealing). Στο τέλος της εργασίας παρουσιάζονται συγκεκριμένα παραδείγματα δικτυωμάτων επιλυμένα και με τους δύο αλγορίθμους, συνδυάζοντας τις μεθόδους βελτιστοποίησης και εξάγονται συμπεράσματα συγκρίνοντας τα αποτελέσματα τους.

Λέξεις-Κλειδιά: Επίπεδο Δικτύωμα, Φόρτραν, Άμεση Δυσκαμψία, Βελτιστοποίηση, Μόντε Κάρλο, Προσομοιωμένη Ανόπτηση, Βελτιστοποίηση Σχήματος, Βελτιστοποίηση Μεγέθους, Βελτιστοποίηση Τοπολογίας

ABSTRACT

The main objective of this diploma thesis is to create code in Fortran language to optimise the shape (position of the joints), the size (the sections of the bars), and the topology of the truss. For the static analysis of the truss, the Direct Stiffness Method is used. For this particular thesis, for the sizing of the members, only round hollow structural sections are used. In order to perform the optimization, two algorithms are utilised, Monte Carlo simulation and the stochastic algorithm of Simulated Annealing. Some examples of trusses are presented, solved by both Monte Carlo and Simulated Annealing, by combining the three methods of optimization and there are conclusions drawn by the comparisons made between the results.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

ΠΙΝΑΚΕΣ

ΕΙΣΑΓΩΓΗ

Σκοπός της διπλωματικής είναι να δημιουργηθεί κώδικας σε γλώσσα Fortran για τη στατική επίλυση ενός οποιουδήποτε επίπεδου δικτυώματος, καθώς και για την βελτιστοποίηση του, ως προς το σχήμα (Shape Optimization), το μέγεθος (Size Optimization) και την τοπολογία (Topology Optimization).

Στόχος της διπλωματικής είναι η εξοικίωση με προγραμματισμό στην Μέθοδο Πεπερασμένων Στοιχείων, με τους αλγόριθμους βελτιστοποίησης καθώς και με την βελτιστοποίηση των κατασκευών γενικότερα. Λόγω της εξέλιξης της τεχνολογίας, σε συνδυασμό με την κρίση των τελευταίων χρόνων, η εύρεση του βέλτιστου σχεδιασμού στις κατασκευές αποκτά όλο και μεγαλύτερη αξία.

Προτιμήθηκε η γλώσσα Fortran έναντι των άλλων γλωσσών για την ευκολία χρήσης σε συνδυασμό με την ταχύτητα εκτέλεσης του προγράμματος (κάτι που πρέπει να ληφθεί υπόψιν στις βελτιστοποιήσεις, καθώς χρειάζονται χιλιάδες επαναλήψεις κάθε φορά) . Ο κώδικας γράφτηκε σε Fortran 90 στον μεταγλωττιστή της, Force 2.0.

Στο 1^{ο} Κεφάλαιο γίνεται μια εισαγωγή στη Μέθοδο Πεπερασμένων Στοιχείων (Finite Element Method). Πρόκειται τη βάση με την οποία γίνεται δυνατό να υπολογισθεί μέσω Η/Υ το δικτύωμα της παρούσας εργασίας. Γίνεται ιστορική αναδρομή για την εξέλιξη της από την αρχαιότητα μέχρι το αποκορύφωμα της στα μέσα του 20^{ου} αιώνα και στην εδραίωση της. Ακόμη περιγράφεται συνοπτικά η διαδικασία με την οποία λειτουργούν τα εμπορικά προγράμματα που χρησιμοποιούν την Μέθοδο.

Στο 2^{ο} Κεφάλαιο περιγράφεται η Μέθοδος με την οποία γίνεται η στατική επίλυση του δικτυώματος. Πρόκειται για τη Μέθοδο Άμεσης Δυσκαμψίας-Στιβαρότητας (Direct Stiffness Method). Επεξηγείται μέσω ενός απλού παραδείγματος η πορεία επίλυση της.

Στο 3^{ο} Κεφάλαιο παρουσιάζονται οι αλγόριθμοι βελτιστοποίησης που χρησιμοποιούνται στην παρούσα εργασία. Πρόκειται για τη Μέθοδο Μόντε Κάρλο(Monte Carlo Simulation) καθώς και για τον αλγόριθμο Προσοιωμένης Ανόπτησης (Simulated Annealing). Η Μέθοδος Μόντε Κάρλο επιλέχθηκε λόγω της ευκολίας χρήσης και κατανόησης της. Ο αλγόριθμος Προσομοιωμένης Ανόπτησης επιλέχθηκε γιατί παρουσιάζει πιο ακριβή αποτελέσματα σε σχέση με την Μόντε Κάρλο, καθώς λόγω ότι δεν θεωρεί αποδεκτές μόνο τις βέλτιστες λύσεις,αλλά αποδέχεται και μερικές αρνητικές λύσεις, δεν παγιδεύεται στο τοπικό ακρότατο της υπό βελτιστοποίηση συνάρτησης αλλά βρίσκει με καλή ακρίβεια το ολικό μέγιστο ακρότατο της.

Στο 4^{ο} Κεφάλαιο αναλύεται η Βελτιστοποίηση των κατασκευών. Αρχικά αναπτύσσεται η Βελτιστοποίηση Σχήματος (Shape Optimization). Μεταβλητή σχεδιασμού στην παρούσα διπλωματική εργασία είναι οι συντεταγμένες των κόμβων τους δικτυώματος. Παράμετρος βελτιστοποίησης είναι η ελαχιστοποίηση του βάρους του δικτυώματος. Αρχικά επεξηγείται η διαδικασία με την οποία αναπτύσσονται οι Μόντε Κάρλο και Προσομοιωμένη Ανόπτηση ως προς τη βελτιστοποίηση σχήματος και στη συνέχεια δίδονται παραδείγματα επιλυμένα και με τους δύο μεθόδους και γίνεται σύγκριση μεταξύ τους.

Στη συνέχεια αναλύεται η Βελτιστοποίηση Μεγέθους (Size Optimization). Η διαστασιολόγηση των διατομών γίνεται αποκλειστικά μεταξύ Κοίλων Τετραγωνικών Διατομών. Πρέπει να ληφθεί υπόψιν ο λυγισμός σε περίπτωση θλιπτικής αξονικής δύναμης στις ράβδους. Αφού παρουσιαστεί η μεθοδολογία των δύο μεθόδων,δίδονται παραδείγματα. Τέλος, παρουσιάζεται και η Βελιστοποίηση Τοπολογίας. Αφού αναλυθούν τα κριτήρια αστοχίας και η διαδικασία επίλυσης, ακολουθεί παράδειγμα.

Στο 5^{ο} Κεφάλαιο δίδονται παραδείγματα στα οποία πέρα από τα κατακόρυφα φορτία ή οριζόντια φορτία, λαμβάνεται υπόψιν και το ίδιο βάρος των ράβδων ως φορτίο κατακόρυφο ασκούμενο στους κόμβους.Παράλληλα,γίνεται συνδυασμός των μεθόδων βελτιστοποίησης.

Στο 6^{ο} Κεφάλαιο παρουσιάζονται συμπεράσματα που προέκυψαν από τις επιλύσεις με τους δύο αλγορίθμους και τους τρεις τρόπους βελτιστοποίησης.

Στο Παράρτημα αναπτύσσεται ο κώδικας σε Fortran, σύμφωνα με τον οποίο εκτελέστηκαν τα διάφορα παραδείγματα, μαζί με τα αρχεία INPUT, που χρησιμοποιήθηκαν στην εκτέλεση των προγραμμάτων.

ΚΕΦΑΛΑΙΟ 1: ΜΕΘΟΔΟΣ ΠΕΠΕΡΑΣΜΕΝΩΝ ΣΤΟΙΧΕΙΩΝ

1.1 ΤΙ ΕΙΝΑΙ Η ΜΕΘΟΔΟΣ ΠΕΠΕΡΑΣΜΕΝΩΝ ΣΤΟΙΧΕΙΩΝ

Η Μέθοδος Πεπερασμένων Στοιχείων (Finite Element Method) είναι αριθμητική μέθοδος για τον υπολογισμό προσεγγιστικών λύσεων σε προβλήματα συνοριακών συνθηκών για μερικές διαφορικές εξισώσεις.

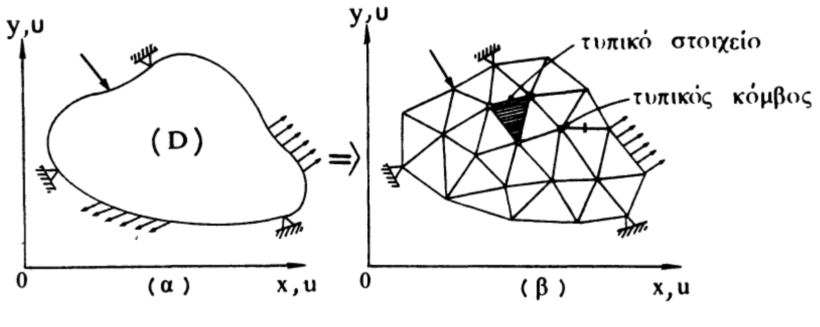
Η Μέθοδος αυτή μπορεί να θεωρηθεί ως μια ειδική διατύπωση των μεθόδων Rayleigh-Ritz, και σταθμικών υπολοίπων οι οποίες είναι από τις πιο διαδεδομένες προσεγγιστικές μεθόδους για την επίλυση προβλημάτων της Μηχανικής. Η Μ.Π.Σ. έχει το πλεονέκτημα της μεγαλύτερης ευκολίας αντιμετώπισης πολύπλοκων γεωμετριών λόγω του προγραμματισμού σε Η/Υ.

Η βασική έννοια της μεθόδου είναι η δυνατότητα αντικατάστασης του γεωμετρικά σύνθετου πεδίου του προβλήματος με ένα σύνολο απλών υποπεδίων τα οποία ονομάζονται πεπερασμένα στοιχεία [1]. Τα τεχνητά αυτά υποπεδία, ή πεπερασμένα στοιχεία είναι συνήθως τετράπλευρα ή τριγωνικά, και συνδέονται σε κόμβους πεπερασμένου πλήθους (εξ ου και η προέλευση της ονομασίας της μεθόδου). Η διαδικασία αυτή έχει ως αποτέλεσμα ένα σύνολο αλγεβρικών εξισώσεων.

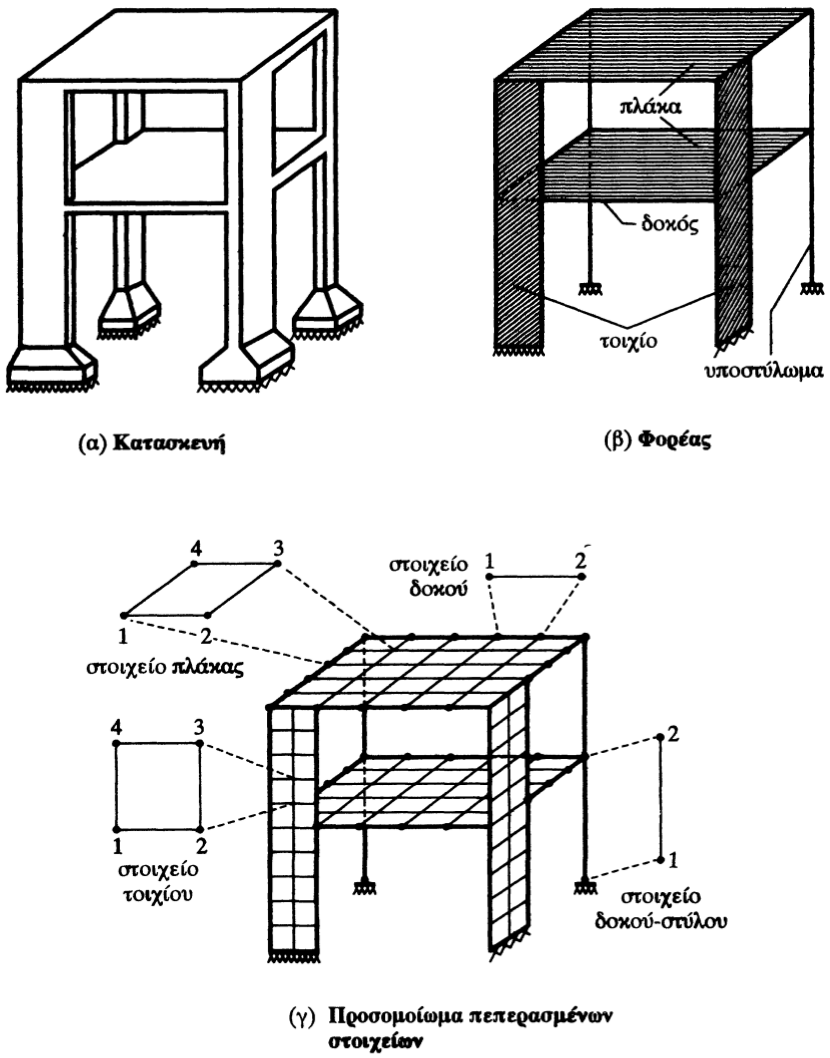
1.2 ΠΩΣ ΛΕΙΤΟΥΡΓΕΙ Η ΜΕΘΟΔΟΣ

Η κατασκευή υποδιαιρείται με τη βοήθεια ιδεατών γραμμών ή επιφανειών σ' έναν αριθμό στοιχείων. Πρέπει να τονιστεί ότι η υποδιαίρεση αυτή είναι δυνατό να μην αναπαριστά με απόλυτη ακρίβεια την κατασκευή, επειδή παραδείγματος χάρη, το καμπύλο σύνορο του σώματος δεν μπορεί να παρασταθεί με απόλυτη ακρίβεια. Άρα η διακεκριμενοποίηση της κατασκευής οδηγεί στον υπολογισμό μιας κατασκευής που είναι αρκετα κοντινή με την αρχική αλλα δεν ταυτίζεται απόλυτα με αυτήν [13] .

Τα στοιχεία συνδέονται μεταξύ τους σε διακριτά σημεία, τους κόμβους. Αυτό σημαίνει ότι κατά μία έννοια, η κατασκευή μετασχηματίζεται σε ένα είδος δικτυώματος. Το σχήμα στην πραγματικότητα είναι πολύ πιο σύνθετο, δεδομένου ότι μπορεί να υπάρχουν και εσωτερικοί κόμβοι.



***ΣΧΗΜΑ 1.1** : α) Τυχαία κατασκευή D . β) Διακεκριμενοποίηση της κατασκευής(χωρισμός σε πεπερασμένα στοιχεία (πηγή: Γεώργιος Τσαμασφύρος - Μέθοδος των Πεπερασμένων Στοιχείων)*



ΣΧΗΜΑ 1.2 : *α) Κατασκευή,β) Φορέας,*
γ) Προσομοίωμα πεπερασμένων στοιχείων
(πηγή: Γεώργιος Τσαμασφύρος - Μέθοδος των Πεπερασμένων Στοιχείων)

1.2.1 ΒΗΜΑΤΑ ΕΠΙΛΥΣΗΣ

Τα βήματα με τα οποία λειτουργούν τα εμπορικά προγράμματα πεπερασμένων στοιχείων είναι:

- Προεργασία: Μοντελοποίηση
- Επίλυση: Απόκτηση αποτελεσμάτων
- Επεξεργασία αποτελεσμάτων
- Βελτιστοποίηση

ΠΡΟΕΡΓΑΣΙΑ:

- Καθορισμός τύπου προβλήματος.
- Χρησιμοποίηση κόμβων, στοχείων, διατομών για την κατασκευή της γεωμετρίας του προβλήματος.
- Ανάθεση σωστών υλικών και ιδιοτήτων στα επιμέρους στοιχεία της κατασκευής.

ΕΠΙΛΥΣΗ:

- Εφαρμογή φορτίων και συνθηκών στήριξης.
- Υπολογισμός προβλήματος.
- Επανεκκίνηση υπολογισμών σε περίπτωση σφάλματος.

ΕΠΕΞΕΡΓΑΣΙΑ ΑΠΟΤΕΛΕΣΜΑΤΩΝ:

- Εμφάνιση αποτελεσμάτων σε γραφήματα ή διαγράμματα.
- Εξέταση έαν είναι αποδεκτά τα αποτελέσματα.
- Εάν είναι μη αποδεκτά, επανάληψη από το στάδιο Προεργασίας.

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ:

- Υπολογισμός αλλάζοντας στοιχεία ή ιδιότητες της κατασκευής με σκοπό να βρεθούν καλύτερα και οικονομικότερα αποτελέσματα εξασφαλίζοντας ακρίβεια και ασφάλεια.

1.3 ΕΦΑΡΜΟΓΕΣ

Η Μ.Π.Σ. έχει μεγάλο έυρος εφαρμογών σε προβλήματα της μηχανικής. Πρακτικά χρησιμοποιείται σε οποιοδήποτε πρόβλημα που λύνεται με μερικές διαφορικές εξισώσεις. Η πλειονότητα των εφαρμογών περιλαμβάνονται στις κατηγορίες της Μηχανικής των Ρευστών και της Μηχανικής των Στερεών. Τις τελευταίες δεκαετίες η Μ.Π.Σ. χρησιμοποείται και σε προβλήματα ηλεκτρικού και ηλεκτρομαγνητικού περιεχομένου, καθώς και στην βιομηχανική. Οι εφαρμογες της Μ.Π.Σ. μπορούν να χωριστούν σε 3 κατηγοριες [4] :

- Προβλήματα ισορροπίας
- Προβλήματα ιδιοτιμών
- Παροδικά προβλήματα

Τα προβλήματα ισορροπίας κυμαίνονται γενικά σε προβλήματα σταθερής κατάστασης όπως ο υπολογισμός της τάσης και της μετατόπισης σε προβλήματα Μηχανικής των Στερεών, ο υπολογισμός της θερμοκρασίας σε θερμικά προβλήματα, η εκτίμηση της ταχύτητας και της πίεσης σε προβλήματα της Μηχανικής των Ρευστών.

Τα προβλήματα ιδιοτιμών περιλαμβάνουν εκτίμηση της δόνησης και της συχνότητας στα στερεά και ρευστά σώματα.

Στα παροδικα προβλήματα η Μ.Π.Σ χρησιμοποιείται σε προβλήματα της μηχανικής με συνάρτηση τον χρόνο.

Παρακάτω περιγράφονται περιληπτικά εφαρμογές της Μ.Π.Σ σε διάφορους κλάδους της μηχανικής.

- Μηχανική των Κατασκευών: Τα προβλήματα ισορροπίας περιλαμβάνουν ανάλυση των ράβδων, πλαισίων, υπολογισμό της τάσης και της στρέψης διάφορων κατασκευών. Τα προβλήματα ιδιοτιμών περιλαμβάνουν ανάλυση της σταθερότητας της κατασκευής, των δονήσεων και της συχνότητάς της. Τα παροδικά προβλήματα περιλαμβάνουν δυναμική ανάλυση των κατασκευών με περιοδικά φορτία.
- Γεωτεχνική Μηχανική: Ανάλυση της τάσης, έλεγχος ασφαλείας σε ανατροπή ή ολίσθηση, έλεγχος διαρροής των υγρών στα εδαφικά στρώματα.
- Μηχανολογία: Σταθερή και παροδική θερμική ανάλυση των στερεών και ρευστών, σχεδιασμός αυτοκίνησης, καθώς και σχεδιασμός παραγωγικής διαδικασίας
- Πυρηνική Μηχανική: Σταθερή και δυναμική ανάλυση των κατασκευών αντιδραστήρων, ανάλυση της παροδικής μετάδοσης θερμοκρασίας στις κατασκευές.

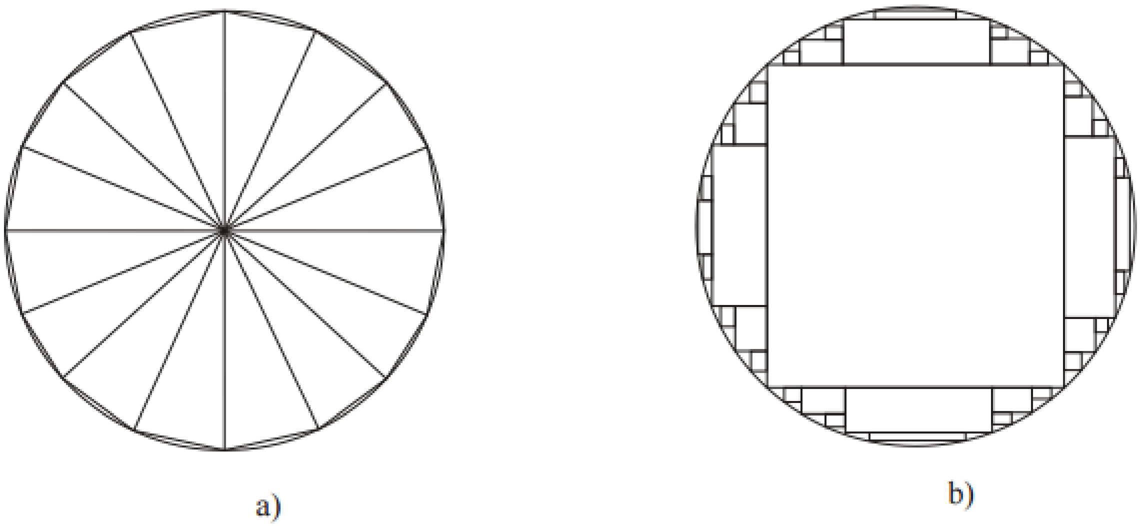
1.4 ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

1.4.1 ΣΤΟΝ ΑΡΧΑΙΟ ΚΟΣΜΟ

Εφαρμογές:

1. Περίμετρος και εμβαδόν κύκλου
2. Όγκος κυλίνδρου και σφαίρας
3. Άλλα πολύπλοκα γεωμετρικά σχήματα

Ο Αρχιμήδης χρησιμοποιούσε πεπερασμένα στοιχεία για να υπολογίσει τον όγκο των στερεών σωμάτων. Πιο πρόσφατα ο Euler, στις μελέτες του για τη δημιουργία του λογισμού των μεταβολών, χώρισε το διάστημα μιας μονοδιάστατης συνάρτησης σε πεπερασμένα διαστήματα και θεώρησε γραμμική μεταβολή μεταξύ τους, καταλήγοντας σε αυτό που λέμε σήμερα Διαφορική Εξίσωση Euler-Langrange.



ΣΧΗΜΑ 1.3: *Προσέγγιση της περιφέρειας του κύκλου*
(πηγή: http://www.mm.bme.hu/~szeki/files/2012_vem-en.pdf)

Για να υπολογίσουμε την περίμετρο του κύκλου, πρέπει να χωρίσουμε τον κύκλο σε n στοχεία όπως φαίνεται στο σχήμα 1.3a. Η προσεγγιστική τιμή του π και το σφάλμα του δίνονται στους παρακάτω τύπους:

$$\pi \approx n\cos\left(\frac{360^\circ}{2n}\right)\sin\left(\frac{360^\circ}{2n}\right), \quad \Sigma\varphi\acute{\alpha}\lambda\mu\alpha = \frac{\pi - n\cos\left(\frac{360^\circ}{2n}\right)\sin\left(\frac{360^\circ}{2n}\right)}{\pi}\cdot 100\%.$$

Ο Κινέζος μηχανικός Tsu Ch'ung-Chih το 480 μ.Χ. χρησιμοποιώντας τα ορθογώνια, προσεγγιστικά υπολόγισε ότι η τιμή του π είναι μεταξύ 3,1415926 - 3,1415927 [14].

1.4.2 ΣΤΗΝ ΣΥΧΡΟΝΗ ΙΣΤΟΡΙΑ

Στα τέλη της δεκαετίας του 1940 εμφανίστηκαν τα αεροπλάνα jet, τα οποία λόγω της προηγμένης τεχνολογίας και ταχύτητας, απαιτούσαν πιο περίπλοκες κατασκευές, όπως τα φτερά δέλτα (τριγωνικά). Οι προηγούμενες μέθοδοι για να σχεδιαστούν τόσο σύνθετες κατασκευές, κρίθηκαν ανεπαρκείς, καθώς η αναξιοπιστία των υπολογισμών δεν θα μπορούσε να αντισταθμιστεί. Χρειαζόταν μια πιο αξιόπιστη και ακριβής μέθοδος υπολογισμού των σύνθετων γεωμετριών.
Ο Levy εφάρμοσε πρώτος την Μέθοδο των Δυνάμεων, η οποία βασίζεται στην παραδοχή ότι οι μετακινήσεις υπολογίζονται από την ισορροπία των δυνάμεων.
Η Μέθοδος Πεπερασμένων Στοιχείων αναπτύχθηκε αισθητά και διεδόθηκε τις δεκαετίες ’60, 70 χάρη στον Ιωάννη Αργύρη μαζί με τους συνεργάτες του στο πανεπιστήμιο της Στουτγκάρδης. Από εκείνη την εποχή και μετά, η εξέλιξη της Μεθόδου συμβάδιζε με την εξέλιξη των Η/Υ καθώς με την μέθοδο αυτή καταλήγουμε σε αριθμό αλγεβρικών εξισώσεων ίσο με τον βαθμό ελευθερίας του προβλήματος, συνεπώς είναι απαραίτητη η χρήση του Η/Υ για την επίλυση των πολύπλοκων προβλημάτων.

Η **Μέθοδος Άμεσης Δυσκαμψίας-Στιβαρότητας** (με την οποία επιλύεται ο κώδικας στην συνέχεια), είναι από τις πιο διαδεδομένες εφαρμογές των Πεπερασμένων Στοιχείων. Μεταξύ 1934-1938 οι A.R. Collar και W. J. Duncan δημοσίευσαν τις πρώτες εργασίες σχετικά με την αναπαράσταση των μητρώων όπως αυτά χρησιμοποιούνται σήμερα [15]. Το 2^{ο} μεγαλύτερο αποκορύφωμα της μεθόδου Άμεσης Ακαμψίας συνέβη γύρω στο 1955 όταν ο καθηγητής Ιωάννης Αργύρης κατάφερε να μετατρέψει τα στοιχεία της κατασκευής σε σύστημα εξισώσεων.

Την περίοδο μετά τον πόλεμο στο Βιετνάμ, η εξέλιξη της μεθόδου είχε βυθιστεί σε τέλμα.Παρόλα αυτά, εμπορικά προγράμματα, τα οποία χρησιμοποιούσαν την Μέθοδο, άρχιζαν σιγά σιγά να εμφανίζονται. Από την περίοδο του 1980 και μετά, τα προγράμματα γίνονταν όλο και πιο σύνθετα, λύνοντας μη γραμμικά και δυναμικά συστήματα [16].

ΚΕΦΑΛΑΙΟ 2: ΜΕΘΟΔΟΣ ΑΜΕΣΗΣ ΔΥΣΚΑΜΨΙΑΣ-ΣΤΙΒΑΡΟΤΗΤΑΣ

2.1 ΕΙΣΑΓΩΓΗ

Οι ραβδωτοί φορείς που αποτελούνται από ευθύγραμμες ράβδους, των οποίων τα άκρα συνδέονται αρθρωτά σε κόμβους και μεταφέρουν μόνο αξονικές δυνάμεις, ονομάζονται δικτυώματα [30]. Στην περίπτωση κατά την οποία όλες οι ράβδοι του δικτυώματος βρίσκονται σε ένα επίπεδο και η φόρτιση του ανήκει στο επίπεδο αυτό, το δικτύωμα αναφέρεται ως επίπεδο, ενώ σε αντίθετη περίπτωση,αναφέρεται ως χωρικό δικτύωμα.

2.2 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΜΕΘΟΔΟΥ

Η Μέθοδος Άμεσης Δυσκαμψίας (Direct Stiffness Method) αποτελεί την πιο διαδεδομένη εφαρμογή της μεθόδου Πεπερασμένω Στοιχείων. Το πρόβλημα το οποίο θέλουμε να επιλυθεί πρέπει να αναλυθεί σε ένα σύστημα στοιχείων συνδεδεμένων σε κόμβους. Στη συνέχεια μορφώνονται τα μητρώα δυσκαμψίας των στοιχείων τα οποία ελέγχουν την συμπεριφορά της κατασκευής.

Μέσω των μητρώων δυσκαμψίας μπορούμε να υπολογίσουμε τις μετακινήσεις των ελεύθερων κόμβων της κατασκευής, τις αντιδράσεις στις στηρίξεις καθώς και τα εντατικά μεγέθη στα άκρα του κάθε μέλους

Η μέθοδος προήλθε από τον κλάδο της αεροδιαστημικής. Για να προσεγγιστεί η ανάλυση σύνθετων κατασκευών αεροπλάνων, χρησιμοποιήθηκαν μέθοδοι όπως η θεωρία της Ελαστικότητας, η μέθοδος της Ευκαμψίας, η μέθοδος των Μητρώων στιβαρότητας, και μέσω αυτών των θεωριών εμφανίστηκε η Μέθοδος Άμεσης Δυσκαμψίας ως μια αποτελεσματική μέθοδος για ενσωμάτωση σε υπολογιστικό περιβάλλον.

2.3 ΒΑΣΙΚΟΙ ΌΡΟΙ

- **Κόμβοι**: Γενική ονομασία για την σύνδεση μεταξύ γειτονικών στοιχείων. Χωρίζονται σε ελεύθερους (όπου επιτρέπονται όλες οι δυνατές μετατοπίσεις) και δεσμευμένους (απογορεύονται κάποιες ή όλες οι μετατοπίσεις).

- **Στοιχεία**: Τα ραβδωτά μέλη στα οποία χωρίζουμε την κατασκευή. Υποβάλλονται σε αξονική ένταση. Η ένταση αυτή παραλαμβάνεται από ομοιόμορφα κατανεμημένες ορθές τάσεις.

- **Βαθμοί Ελευθερίας**: Ο αριθμός των συνολικών δυνατών διευθύνσεων κατά τις οποίες μπορεί να μετακινηθεί και να στραφεί ο κόμβος. Για παράδειγμα:
    - Επίπεδο Δικτύωμα: 2 βαθμοί ελευθερίας σε κάθε κόμβο:μετακίνηση κατά x,και κατά y
    - Συνεχής Δοκός: 2 βαθμοί ελευθερίας σε κάθε κόμβο: κατακόρυφη μετακίνηση και στροφή
    - Επίπεδο Πλαίσιο: 3 βαθμοί ελευθερίας σε κάθε κόμβο: μετακίνηση κατά x,κατά y και στροφή κόμβου

  Συνεπώς ένα επίπεδο δικτύωμα με 10 κόμβους έχει 20 βαθμούς ελευθερίας.

- **Μητρώα Δυσκαμψίας**: Το μητρώο δυσκαμψίας ενός μέλους περιέχει τις εντάσεις που προκαλούνται στα άκρα του από μοναδιαίες μετακινήσεις στους κόμβους.

$$[\,\mathbf{K_i}\,]=\begin{bmatrix} \frac{E_iA_i}{L_i} & 0 & -\frac{E_iA_i}{L_i} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{E_iA_i}{L_i} & 0 & \frac{E_iA_i}{L_i} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{(2.1)}$$

όπου E_i το μέτρο Ελαστικότητας, A_i το εμβαδόν διατομής, και L_i το μήκος του μέλους i.

- Η σχέση για το καθολικό μητρώο στιβαρότητας και στα δυο άκρα του κάθε μέλους είναι:

$$[\,\overline{\mathbf{K}}_{\mathbf{i}}] = [\,\mathbf{\Lambda_i}\,]^{\mathbf{T}} \times [\,\mathbf{K_i}\,] \times [\,\mathbf{\Lambda_i}\,] \qquad \mathbf{(2.2)}$$

όπου

$$[\mathbf{\Lambda_i}\,] = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & \cos\varphi & \sin\varphi \\ 0 & 0 & -\sin\varphi & \cos\varphi \end{bmatrix} \qquad \mathbf{(2.3)}$$

και φ η αριστερόστροφη γωνία μεταξύ της ράβδου και του άξονα x.

Κάνοντας τον πολλαπλασιασμό προκύπτει:

$$[\,\overline{\mathbf{K}}_{\mathbf{i}}]=\frac{\mathrm{E_iA_i}}{\mathrm{L_i}}\times\begin{bmatrix} \cos^2\varphi & \cos\varphi\sin\varphi & -\cos^2\varphi & -\cos\varphi\sin\varphi \\ \cos\varphi\sin\varphi & \sin^2\varphi & -\cos\varphi\sin\varphi & -\sin^2\varphi \\ -\cos^2\varphi & -\cos\varphi\sin\varphi & \cos^2\varphi & \cos\varphi\sin\varphi \\ -\cos\varphi\sin\varphi & -\sin^2\varphi & \cos\varphi\sin\varphi & \sin^2\varphi \end{bmatrix} \qquad \mathbf{(2.4)}$$

- Η σχέση 2.4 εφαρμόζεται για κάθε ράβδο του δικτυώματος. Θεωρώντας τις μετακινήσεις κοινές σε κάθε ράβδο και τις εξωτερικές δυνάμεις ως άθροισμα των ακραίων δυνάμεων των ράβδων, οι σχέσεις 2.4 συνθέτονται στη μητρωική σχέση για όλο το δικτύωμα:

$$[P]=[K]\times[\Delta] \qquad (2.5)$$

όπου [Ρ] το μητρώο εξωτερικών δυνάμεων και [Δ] το μητρώο μετακινήσεων.

2.4 ΒΑΣΙΚΕΣ ΠΑΡΑΔΟΧΕΣ

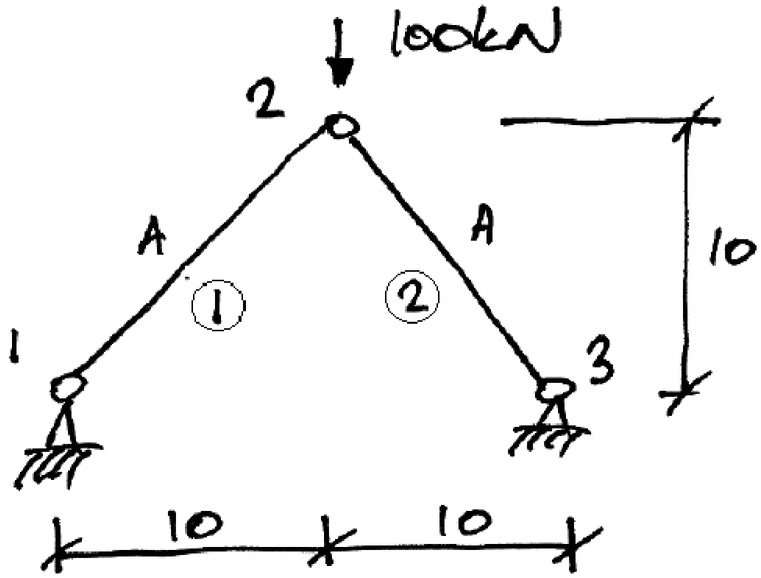
Για την ανάλυση των δικτυωμάτων γίνονται οι εξής απλοποιητικές παραδοχές [30]:

- Τα εξωτερικά φορτία αναλύονται αποκλειστικά σε δυνάμεις που ενεργούν επί των κόμβων ή είναι παράλληλες με τους άξονες των στοιχείων.
- Οι ράβδοι του δικτυώματος θεωρούνται εγκαρσίως αφόρτιστες.

Στην πραγματικότητα μπορεί να υπάρχουν αποκλίσεις από τις παραδοχές για διάφορους λόγους. Για παράδειγμα, στους πραγματικούς κόμβους, η τριβή στην άρθρωση δεν είναι μηδενική, και υπάρχει πρόσθετη μικρή καμπτική ένταση λόγω της έκκεντρης σύνδεσης των ράβδων στους κόμβους.

2.5 ΠΟΡΕΙΑ ΕΠΙΛΥΣΗΣ ΔΙΚΤΥΩΜΑΤΟΣ ΜΕΣΩ ΚΩΔΙΚΑ

Αρχικά εισάγεται η μορφή του δικτυώματος. Δίδονται δηλαδή οι συντεταγμένες των κόμβων, οι θέσεις των στηρίξεων, και οι φορτίσεις των κόμβων.
Για παράδειγμα έχουμε το παρακάτω δικτύωμα.



ΣΧΗΜΑ 2.1: *Δικτύωμα 3 κόμβων*
(πηγή: http://www.colincaprani.com/files/notes/SAIV/4%20-%20Matrix%20Stiffness%20Method.pdf)

Το εμβαδόν της κάθε διατομής είναι: A=0.005 m^2, και το μέτρο Ελαστικότητας: E=210 GPA.

ΕΠΙΛΥΣΗ:

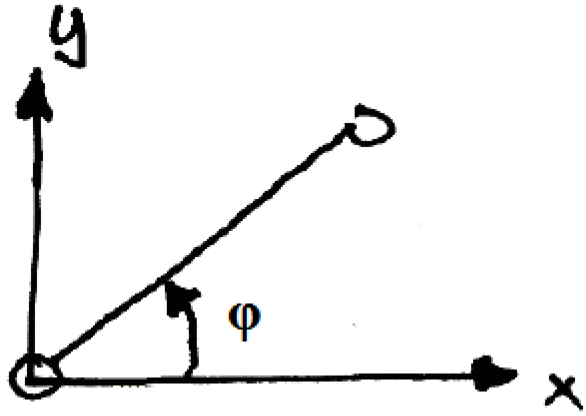
1) Τα μήκη των ράβδων προκύπτουν από την σχέση:

$$\sqrt{(\boldsymbol{x2}-\boldsymbol{x1})^2+(\boldsymbol{y2}-\boldsymbol{y1})^2} \qquad \textbf{(2.6)}$$

Στο παράδειγμα προκύπτει:

- L_1=14.14 m
- L_2=14.14 m

2) Υπολογίζουμε την γωνία φ για την κάθε ράβδο, η οποία προκύπτει αριστερόστροφα, ξεκινώντας από τον άξονα των x μέχρι να τμήσει την ράβδο.



ΣΧΗΜΑ 2.2: *Θετική φορά στροφής*
(πηγή: http://www.colincaprani.com/files/notes/SAIV/4%20-%20Matrix%20Stiffness%20Method.pdf)

Ο υπολογισμός στον κώδικα γίνεται παίρνοντας τις περιπτώσεις:

- ΑΝ $x2 - x1 = 0$ ΚΑΙ $y2 - y1>0$ ΤΟΤΕ → Φ=90^o
- ΑΝ $x2 - x1 = 0$ ΚΑΙ $y2 - y1<0$ ΤΟΤΕ → Φ=270^o
- ΑΝ $y2 - y1>0$ ΚΑΙ $x2 - x1>0$ ΤΟΤΕ → Φ=$\tan^{-1}(\frac{\Delta y}{\Delta x})$
- ΑΝ $y2 - y1>0$ ΚΑΙ $x2 - x1<0$ ΤΟΤΕ → Φ=90^o+ $\tan^{-1}(\frac{\Delta x}{\Delta y})$
- ΑΝ $y2 - y1<0$ ΚΑΙ $x2 - x1>0$ ΤΟΤΕ → Φ=360^o- $\tan^{-1}(\frac{\Delta y}{\Delta x})$
- ΑΝ $y2 - y1<0$ ΚΑΙ $x2 - x1<0$ ΤΟΤΕ → Φ=180^o+ $\tan^{-1}(\frac{\Delta y}{\Delta x})$
- ΑΝ $y2 - y1=0$ ΤΟΤΕ → Φ=0^o

Στο παράδειγμα προκύπτει:

- φ_1=45^o (με αρχή τον κόμβο 1)
- φ_2= 315^o (με αρχή τον κόμβο 2)

3) Το τοπικό μητρώο στιβαρότητας για το ένα άκρο του κάθε μέλους είναι:

$$\mathbf{K_i} = \begin{vmatrix} \frac{E_iA_i}{L_i} & -\frac{E_iA_i}{L_i} \\ -\frac{E_iA_i}{L_i} & \frac{E_iA_i}{L_i} \end{vmatrix} \quad \textbf{(2.7)}$$

- $K_1=\begin{bmatrix} 74257,4 & -74257,4 \\ -74257,4 & 74257,4 \end{bmatrix}$

- $K_2=\begin{bmatrix} 74257,4 & -74257,4 \\ -74257,4 & 74257,4 \end{bmatrix}$

4) Το μητρώο μετασχηματισμού για το ένα άκρο του κάθε μέλους είναι:

$$[\mathbf{\Lambda_i}]=\begin{bmatrix} \boldsymbol{cos\varphi} & \boldsymbol{sin\varphi} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{cos\varphi} & \boldsymbol{sin\varphi} \end{bmatrix} \quad \textbf{(2.8)}$$

- $[\Lambda_1]=\begin{bmatrix} 0,707 & 0,707 & 0 & 0 \\ 0 & 0 & 0,707 & -0,707 \end{bmatrix}$

- $[\Lambda_2]=\begin{bmatrix} 0,707 & -0,707 & 0 & 0 \\ 0 & 0 & 0,707 & -0,707 \end{bmatrix}$

5) Για το κάθε μέλος το καθολικό μητρώο στιβαρότητας είναι:

$$[\overline{\mathbf{K}}_\mathbf{i}] = [\mathbf{\Lambda_i}]^\mathbf{T} \times [\mathbf{K_i}] \times [\mathbf{\Lambda_i}]$$

- $[\overline{K}_1]=\begin{bmatrix} 37123{,}1 & 37123{,}1 & -37123{,}1 & -37123{,}1 \\ 37123{,}1 & 37123{,}1 & -37123{,}1 & -37123{,}1 \\ -37123{,}1 & -37123{,}1 & 37123{,}1 & 37123{,}1 \\ -37123{,}1 & -37123{,}1 & 37123{,}1 & 37123{,}1 \end{bmatrix}$

- $[\overline{K}_2]=\begin{bmatrix} 37123{,}1 & -37123{,}1 & -37123{,}1 & 37123{,}1 \\ -37123{,}1 & 37123{,}1 & 37123{,}1 & -37123{,}1 \\ -37123{,}1 & 37123{,}1 & 37123{,}1 & -37123{,}1 \\ 37123{,}1 & -37123{,}1 & -37123{,}1 & 37123{,}1 \end{bmatrix}$

6) Διατυπώνουμε το ολικό μητρώο στιβαρότητας από τα επιμέρους καθολικά. Το κάθε καθολικό είναι πίνακας 4X4 ενώ το ολικό θα είναι NXN όπου N ο συνολικός αριθμός βαθμών ελευθερίας. Για το κάθε στοιχείο i θα χρησιμοποιήσουμε τις παρακάτω σχέσεις:

- $K(2*(ΑΡΧΗ-1)+1\,,2*(ΑΡΧΗ-1)+1)=K(2*(ΑΡΧΗ-1)+1\,,2*(ΑΡΧΗ-1)+1)+K_i(1{,}1)$
- $K(2*(ΑΡΧΗ-1)+1\,,2*(ΑΡΧΗ-1)+2)=K(2*(ΑΡΧΗ-1)+1\,,2*(ΑΡΧΗ-1)+2)+K_i(1{,}2)$
- $K(2*(ΑΡΧΗ-1)+1{,}2*(ΤΕΛΟΣ-1)+1)=K(2*(ΑΡΧΗ-1)+1{,}2*(ΤΕΛΟΣ-1)+1)+K_i(1{,}3)$
- $K(2*(ΑΡΧΗ-1)+1{,}2*(ΤΕΛΟΣ-1)+2)=K(2*(ΑΡΧΗ-1)+1{,}2*(ΤΕΛΟΣ-1)+2)+K_i(1{,}4)$
- $K(2*(ΑΡΧΗ-1)+2{,}2*(ΑΡΧΗ-1)+1)=K(2*(ΑΡΧΗ-1)+2{,}2*(ΑΡΧΗ-1)+1)+K_i(2{,}1)$
- $K(2*(ΑΡΧΗ-1)+2{,}2*(ΑΡΧΗ-1)+2)=K(2*(ΑΡΧΗ-1)+2{,}2*(ΑΡΧΗ-1)+2)+K_i(2{,}2)$
- $K(2*(ΑΡΧΗ-1)+2{,}2*(ΤΕΛΟΣ-1)+1)=K(2*(ΑΡΧΗ-1)+2{,}2*(ΤΕΛΟΣ-1)+1)+K_i(2{,}3)$
- $K(2*(ΑΡΧΗ-1)+2{,}2*(ΤΕΛΟΣ-1)+2)=K(2*(ΑΡΧΗ-1)+2{,}2*(ΤΕΛΟΣ-1)+2)+K_i(2{,}4)$
- $K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΑΡΧΗ-1)+1)=K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΑΡΧΗ-1)+1)+K_i(3{,}1)$
- $K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΑΡΧΗ-1)+2)=K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΑΡΧΗ-1)+2)+K_i(3{,}2)$
- $K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΤΕΛΟΣ-1)+1)=K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΤΕΛΟΣ-1)+1)+K_i(3{,}3)$
- $K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΤΕΛΟΣ-1)+2)=K(2*(ΤΕΛΟΣ-1)+1{,}2*(ΤΕΛΟΣ-1)+2)+K_i(3{,}4)$
- $K(2*(ΤΕΛΟΣ-1)+2{,}2*(ΑΡΧΗ-1)+1)=K(2*(ΤΕΛΟΣ-1)+2{,}2*(ΑΡΧΗ-1)+1)+K_i(4{,}1)$

- $K(2*(ΤΕΛΟΣ-1)+2), 2*(ΑΡΧΗ-1)+2) = K(2*(ΤΕΛΟΣ-2)+1), 2*(ΑΡΧΗ-1)+2) + K(4,2)$
- $K(2*(ΤΕΛΟΣ-1)+2{,}2*(ΤΕΛΟΣ-1)+1) = K(2*(ΤΕΛΟΣ-1)+2{,}2*(ΤΕΛΟΣ-1)+1) + K_i(4,3)$
- $K(2*(ΤΕΛΟΣ-1)+2{,}2*(ΤΕΛΟΣ-1)+2) = K(2*(ΤΕΛΟΣ-1)+2{,}2*(ΤΕΛΟΣ-1)+2) + K_i(4,4)$

όπου ΑΡΧΗ και ΤΕΛΟΣ αναφέρονται στους αριθμούς των κόμβων της ράβδου που χουν δηλωθεί ως αφετηρία και τέλος αντίστοιχα.

Το δικτύωμα έχει 6 βαθμούς ελευθερίας, άρα ολικό μητρώο στιβαρότητας 6Χ6:

- $[K] = \begin{bmatrix} 37123{,}1 & 37123{,}1 & -37123{,}1 & -37123{,}1 & 0 & 0 \\ 37123{,}1 & 37123{,}1 & -37123{,}1 & -37123{,}1 & 0 & 0 \\ -37123{,}1 & -37123{,}1 & 74246{,}2 & 0 & -37123{,}1 & 37123{,}1 \\ -37123{,}1 & -37123{,}1 & 0 & 74246{,}2 & 37123{,}1 & -37123{,}1 \\ 0 & 0 & -37123{,}1 & 37123{,}1 & 37123{,}1 & -37123{,}1 \\ 0 & 0 & 37123{,}1 & -37123{,}1 & -37123{,}1 & 37123{,}1 \end{bmatrix}$

7) Γίνεται αναδιάταξη του μητρώου στιβαρότητας χρησιμοποιώντας την μέθοδο **Computer Oriented Master Stiffness**. Ουσιαστικά για τους δεσμευμένους βαθμούς ελευθερίας μηδενίζονται οι αντίστοιχες στήλες και σειρές του μητρώου κρατώντας στα διαγώνια στοιχεία την τιμή 1.
Στην πράξη, στο συγκεκριμένο παράδειγμα, οι βαθμοί ελευθερίας 1, 2, 5, 6 λόγω αρθρώσεων είναι δεσμευμένοι οπότε θα μηδενιστούν οι σειρές 1,2,5,6, και στήλες 1, 2, 5, 6 ενώ τα διαγώνια στοιχεία στις σειρές αυτές θα έχουν την τιμή 1.

- $[K] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 74246{,}2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 74246{,}2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

8) Υπολογίζεται ο αντίστροφος πίνακας του παραπάνω μητρώου. Ο αλγόριθμος με τον οποίο υπολογίζεται ο αντίστροφος λέγεται **Μέθοδος Gauss-Jordan** .

- $[K]^{-1}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.34*10^{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.34*10^{-5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

9) Από την σχέση:

[P]=[K]×[Δ]

Πολλαπλασιάζοντας και τα δύο μέλη με $[K]^{-1}$ προκύπτει:

$$[K]^{-1}\times[P]=[\Delta] \qquad \textbf{(2.10)}$$

όπου [P] το μητρώο εξωτερικών επικόμβιων δράσεων, το $[K]^{-1}$ το αντίστροφο μητρώο στιβαρότητας και [Δ] το μητρώο επικόμβιων μετακινήσεων.

$[P]=\begin{bmatrix} 0 \\ 0 \\ 0 \\ -100 \\ 0 \\ 0 \end{bmatrix}$ (KN) $\quad [\Delta]=\begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.34*10^{-3} \\ 0 \\ 0 \end{bmatrix}$ (m)

Άρα μετακίνηση του $2^{ου}$ κόμβου κατά y ↓ 134 mm.

10) Οι επικόμβιες δυνάμεις υπολογίζονται από την σχέση (2.5), αφού είναι πλέον γνωστό το μητρώο επικόμβιων μετακινήσεων.

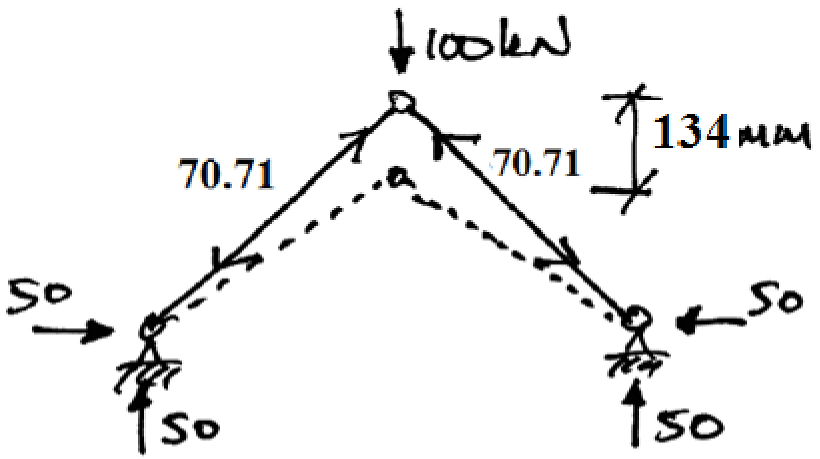
- $[P]=\begin{bmatrix} 50 \\ 50 \\ 0 \\ -100 \\ -50 \\ 50 \end{bmatrix}$

11) Η αξονική για το κάθε στοιχείο προκύπτει από την σχέση:

$$\mathbf{N}=\frac{\boldsymbol{E\times A}}{\boldsymbol{L}}\times[\boldsymbol{\cos\varphi} \quad \mathbf{sin\varphi}]\times\left\{\frac{\boldsymbol{\Delta x\ \tau\acute{\varepsilon}\lambda o\upsilon\varsigma-\Delta x\ \alpha\rho\chi\acute{\eta}\varsigma}}{\boldsymbol{\Delta y\ \tau\acute{\varepsilon}\lambda o\upsilon\varsigma-\Delta y\ \alpha\rho\chi\acute{\eta}\varsigma}}\right\} \qquad \mathbf{(2.11)}$$

όπου Δx είναι η μετακίνηση κατά x και Δy η μετακίνηση κατά y.

- N_1= -70.71 KN
- N_2= -70.71 KN



ΣΧΗΜΑ 2.3: *Δικτύωμα 3 κόμβων επιλυμένο*
(πηγή: http://www.colincaprani.com/files/notes/SAIV/4%20-%20Matrix%20Stiffness%20Method.pdf)

12) Το βάρος της κατασκευής υπολογίζεται από την σχέση:

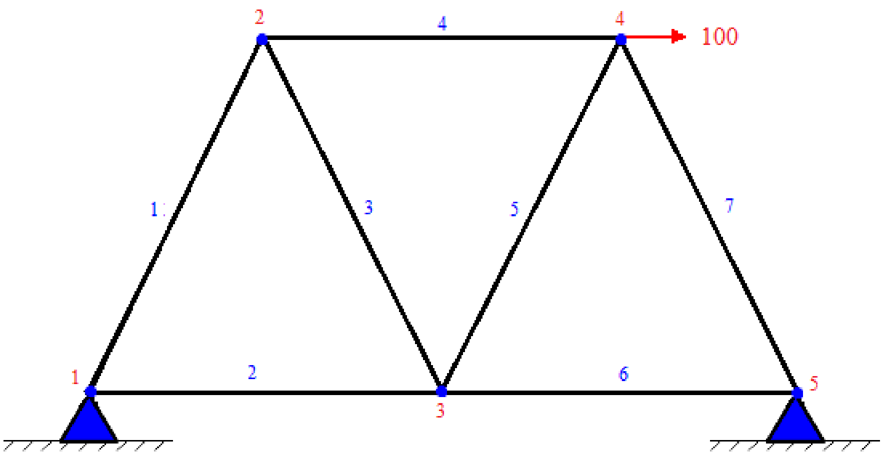
$$\mathbf{W}=\sum_{i=1}^{n}(\boldsymbol{\rho}\mathbf{i}\times\mathbf{Ai}\times\mathbf{Li}) \qquad \textbf{(2.12)}$$

όπου n=αριθμός στοιχείων, ρ η πυκνότητα των στοιχείων. Για χάλυβα ισχύει: ρ=7850 kg/m^3

Στο παράδειγμα προκύπτει:

- W= 1110,157 kg

ΠΑΡΑΔΕΙΓΜΑ 2:



ΣΧΗΜΑ 2.4: Δικτύωμα 5 κόμβων

Η παραπάνω μεθοδολογία εφαρμόστηκε και στο δικτύωμα του σχήματος 2.4 και έδωσε τα αποτελέσματα που συνοπτικά φαίνονται παρακάτω.

Πίνακας 2.1: Μετακινήσεις των κόμβων δικτυώματος 5 κόμβων

Κόμβος	X (m)	Y (m)
1	0	0
2	1.21194462E-03	5.95239653E-05
3	2.38095250E-04	2.38095410E-04
4	1.68813532E-03	1.78571732E-04
5	0	0

Πίνακας 2.2: Αντιδράσεις των κόμβων δικτυώματος 5 κόμβων

Κόμβος	X (KN)	Y (KN)
1	-50	-50
2	-1.26622617E-05	-1.06990046E-06
3	4.83018312E-07	-4.29055035E-07
4	100	2.90713797E-06
5	-50	50

Πίνακας 2.3: Αξονικές των στοιχείων δικτυώματος 5 κόμβων

Ράβδος	Αξονική (KN)
1	55.901680
2	25.00
3	-55.901691
4	50.00
5	55.901688
6	-25.00
7	-55.901691

- Το βάρος της κατασκευής προκύπτει: W = 2932.8135 kg.

ΚΕΦΑΛΑΙΟ 3: ΑΛΓΟΡΙΘΜΟΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

3.1 ΤΙ ΕΙΝΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Μια από τις πιο βασικές αρχές του κόσμου είναι η αναζήτηση για τη βέλτιστη κατάσταση. Ξεκινά από το μικρόκοσμο, στον οποίο τα άτομα προσπαθούν να δημιουργήσουν δεσμούς με σκοπό να ελαχιστοποιήσουν την ενέργεια από τα ηλεκτρόνια [11]. Όταν τα μόρια δημιουργήσουν στέρεα μορφή, προσπαθούν να αποκτήσουν ενεργειακά βέλτιστες κρυσταλλικές δομές. Αυτές οι διεργασίες, φυσικά, προέρχονται από τους νόμους της φυσικής και όχι από ανθρώπινη παρέμβαση.

Το ίδιο ισχύει και για τη βιολογική αρχή της ανθρωπότητας, την επιβίωση των καλύτερα προσαρμοσμένων, κατά την οποία, μέσω της βιολογικής εξέλιξης, η ανθρωπότητα οδηγείται σε καλύτερη προσαρμογή με το περιβάλλον. Στην καθημερινότητα, ο άνθρωπος στοχεύει στο καλύτερο και πιο κοντινό στο τέλειο, σε διάφορους τομείς. Στην οικονομία, το κέρδος και οι πωλήσεις πρέπει να μεγιστοποιηθούν, και τα κόστη να ελαχιστοποιηθούν. Ως εκ τούτου, η βελτιστοποίηση είναι από τις πιο αρχαίες επιστήμες οι οποίες εκτείνονται και στην καθημερινή ζωή.

Βελτιστοποίηση είναι η αναζήτηση του βέλτιστου μεταξύ διαφόρων παραμέτρων σ' ένα σύνθετο πρόβλημα. Συνήθως το πρόβλημα υπόκειται σε συγκεκριμένους περιορισμούς οι οποίοι πρέπει να ικανοποιούνται.

Ελαχιστοποίηση της f(x)
η οποία υπόκειται στους περιορισμούς:

$$g_i(x)>b_i \quad i=1,\ldots,m$$
$$h_i(x)<c_j \quad j= 1,\ldots,n$$

Σύμφωνα με τους C. Coello et al , ο Γαλιλαίος ήταν ο πρώτος επιστήμονας που μελέτησε την βελτιστοποίηση στις κατασκευές. Η τεχνολογία σήμερα επιτρέπει την χρήση πολλαπλών στατικών και δυναμικών επιλύσεων σε μια κατασκευή ώστε να βρεθεί το βέλτιστο αποτέλεσμα. Έτσι επιτυγχάνεται μείωση του κόστους , ενώ παράλληλα παραμένει ασφαλής η κατασκευή.

Η παρούσα εργασία επικεντρώνεται στη βελτιστοποίηση σχήματος (shape optimization), στη βελτιστοποίηση μεγέθους (size optimization), καθώς και στη βελτιστοποίηση τοπολογίας (topology optimization).

Για την επίλυση προβλημάτων, ερευνητές χρησιμοποιούν αλγόριθμους οι οποίοι τερματίζουν σε πεπερασμένο αριθμό βημάτων, ή επαναληπτικές μεθόδους οι οποίες συγκλίνουν σε λύση.

3.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

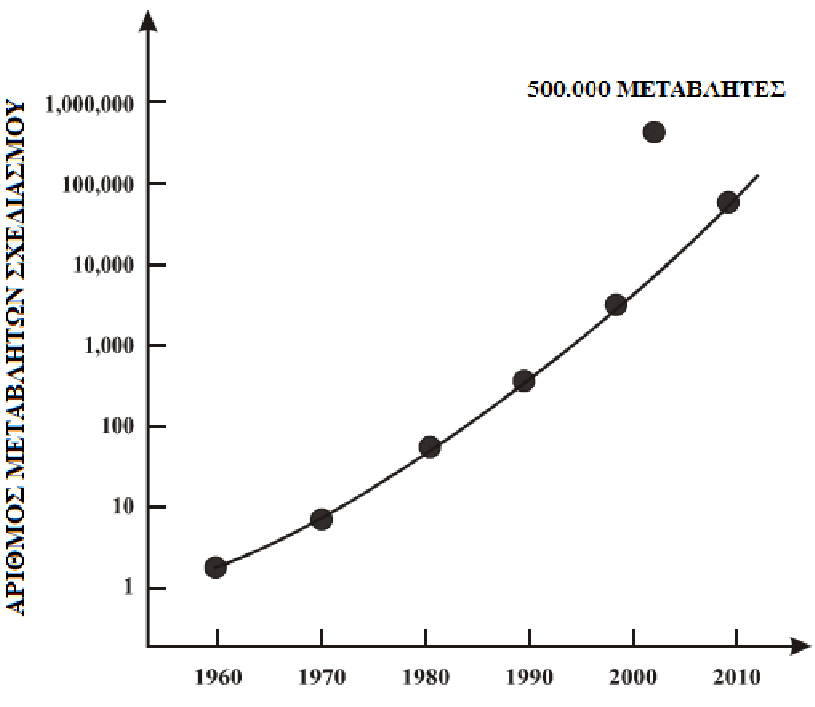
Κατά την περίοδο 1950-1960, οι τυχαίες μέθοδοι αναζήτησης ήταν δημοφιλείς, σύμφωνα με τις οποίες, τα στοιχεία του x διανύσματος επιλεγόταν τυχαία, και μέσω αναλύσεων, εάν βρισκόταν βελτιωμένος σχεδιασμός, διατηρούσαν την εκάστοτε αλλαγή. Αυτή η διαδικασία επαναλαμβανόταν, έως ότου να μην υπάρχει καμία βελτίωση, ή οι υπολογιστικοί πόροι να είχαν εξαντληθεί [12]. Κάποιοι επιστήμονες παρατήρησαν ότι, μετά από κάποιο διάστημα, μπορούσαν να δημιουργήσουν ένα διάνυσμα ορισμένο από τον χειρότερο προς τον καλύτερο σχεδιασμό και να επιταγχύνουν την διαδικασία. Τέτοιες μέθοδοι ήταν εύκολοι να προγραμματιστούν αλλά ήταν αναποτελεσματικές και περιορισμένες σε λίγες μεταβλητές.

Μελέτες στα μέσα του 1960 συμπεριέλαβαν τον Διαδοχικό Γραμμικό Προγραμματισμό (Kelly 1960) , τις Διαδοχικές Τεχνικές Ελαχιστοποίησης χωρίς Περιορισμούς (Fiacco, 1968) και τις Μεθόδους Εφικτών Κατευθύνσεων (Zoutendijk, 1960) .

Κατά την διάρκεια της δεκαετίας 1970, αναπτύχθηκαν ο Επαυξημένος Πολλαπλασιαστής Langrange (Rockefellar, 1973) και οι Μέθοδοι Γενικώς Μειωμένης Βαθμίδας (Gabriel-Ragsdell, 1977). Το πλεονέκτημα των συγκεκριμένων μεθόδων ήταν το ισχυρό θεωρητικό υπόβαθρο στις συνθήκες Kuhn-Tucker για βελτιστοποίηση.

Το 1980-1990 ήταν περίοδος όπου ανανεώθηκε το ενδιαφέρον για τυχαίες μεθόδους στην κοινότητα των μηχανικών. Οι μέθοδοι συμπεριέλαβαν την Γενετική Αναζήτηση (Hajela, 1990), την Προσομοιωμένη Ανόπτηση (Nemhauser-Wolsey, 1988) και συγγενείς μεθόδους που αποσκοπούσαν να μιμηθούν φυσικές εξελικτικές διαδικασίες.

Όσο οι αλγόριθμοι βελτιστοποίησης βελτιώνονταν, το μέγεθος και η πολυπλοκότητα των εφαρμογών στην μηχανική αυξανόταν. Στο παρακάτω σχήμα, ξεκινώντας από το 1960, παρουσιάζεται η τάση στην αύξηση της πολυπλοκότητας στα προβλήματα των μηχανικών.



ΣΧΗΜΑ 3.1*: Ανάπτυξη πολυπλοκότητας προβλημάτων*
(πηγή: Garret N. Vanderplaats - Structural Optimization for Statics, Dynamics and Beyond)

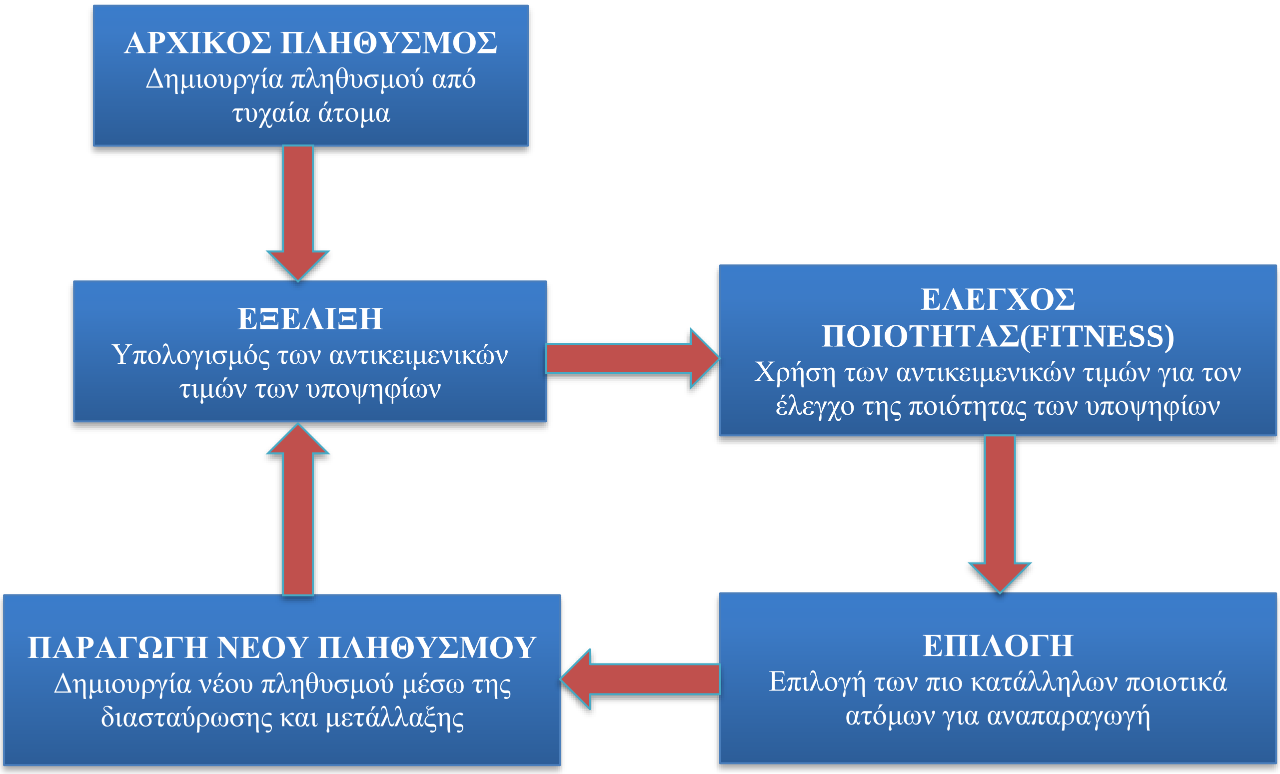
3.3 ΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Ευρετικοί είναι οι αλγόριθμοι που παράγουν λύσεις, όχι απαραίτητα βέλτιστες, σε προβλήματα συνδυαστικής βελτιστοποίησης. Βασίζονται στον μηχανισμό της ευρετικής. Ευρετική είναι μια στρατηγική, βασισμένη στην γνώση για το συγκεκριμένο πρόβλημα, που χρησιμοποιείται σαν βοήθημα για την γρηγορότερη επίλυση του. Μερικοί διαδεδομένοι ευρετικοί αλγόριθμοι είναι :

- Εξελικτικοί Αλγόριθμοι
- Βελτιστοποίηση σμήνους σωματιδίων
- Αλγόριθμος Τεχνηκής αποικίας μυρμηγκιών
- **Αλγορίθμος Προσομοιωμένης Ανόπτησης**

3.3.1 ΕΞΕΛΙΚΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

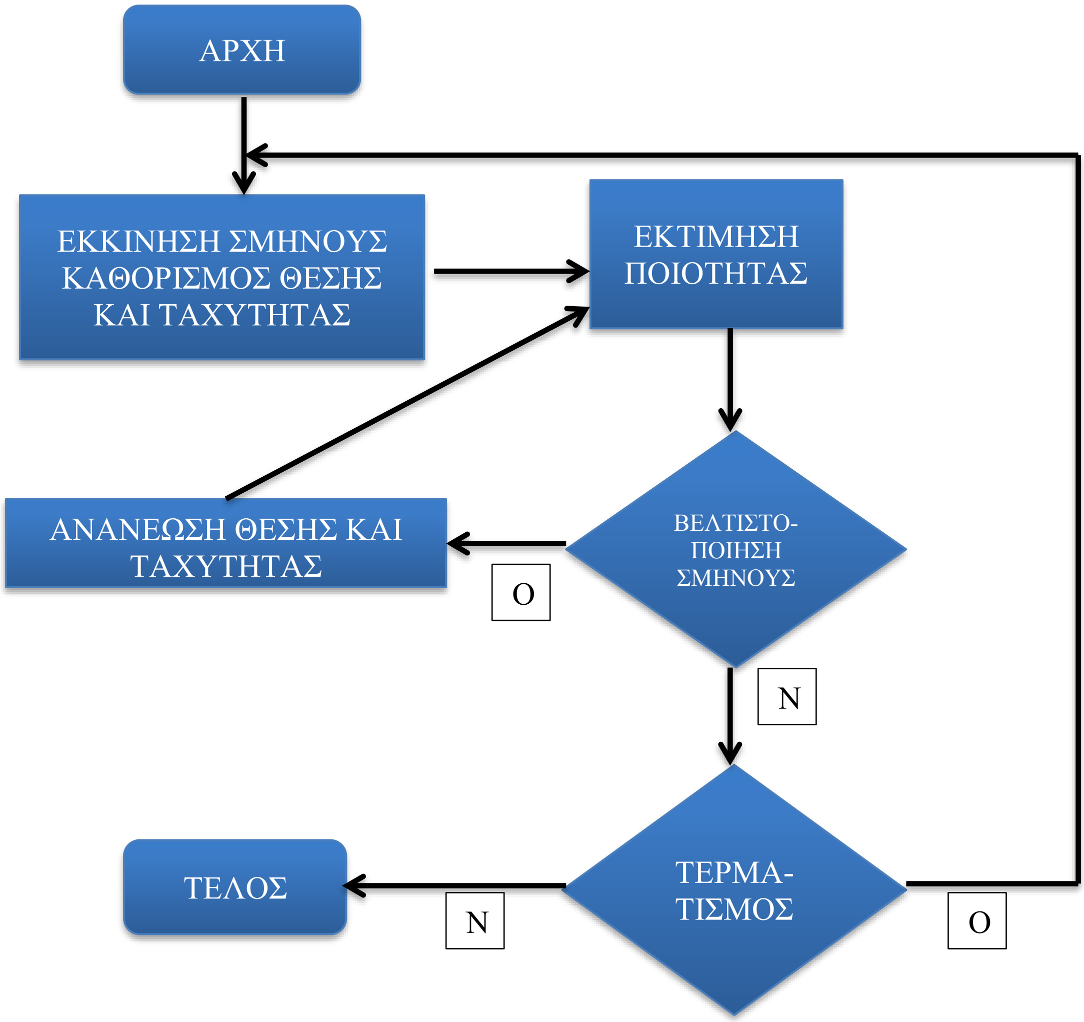
Είναι στοχαστικές μέθοδοι που εξομοιώνουν τη φυσική επιλογή και εξέλιξη στον κόσμο της βιολογίας. Διαφέρουν από άλλες μεθόδους βελτιστοποίησης, στο γεγονός ότι διατηρούν έναν πληθυσμό δυνητικών λύσεων ως προς το πρόβλημα και όχι μια λύση.



ΣΧΗΜΑ 3.3: *Βασικός Κύκλος εξελικτικού αλγόριθμου*
(πηγή: Thomas Weise - Global Optimization Algorithms–Theory and Application)

3.3.2 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΜΗΝΟΥΣ ΣΩΜΑΤΙΔΙΩΝ

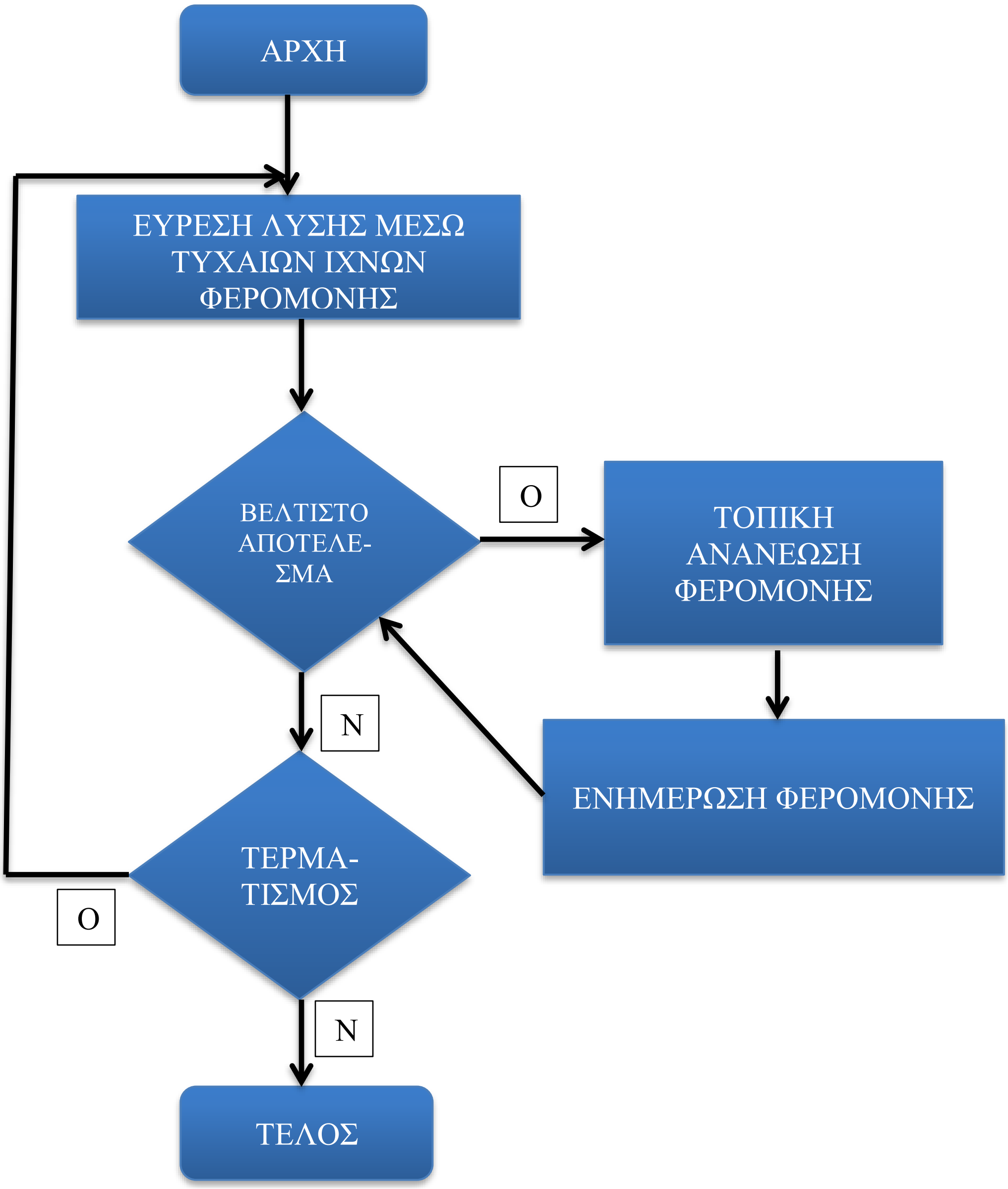
Η βελτιστοποίηση σμήνους σωματιδίων είναι μια στοχαστική τεχνική βελτιστοποίησης βασισμένη σε πληθυσμό που αναπτύχθηκε από τους Everhart-Kennedy το 1995, εμπευσμένοι από την κοινωνική συμπεριφορά που έχουν τα σμήνη πουλιών. Έχει πολλές ομοιότητες με τους εξελικτικούς αλγόριθμους.



ΣΧΗΜΑ 3.4: *Διάγραμμα Ροής αλγόριθμου βελτιστοποίησης σμήνους σωματιδίων (πηγή: https://pbcm.files.wordpress.com/2011/06/060911_1919_implementin33.png?w=630)*

3.3.3 ΑΛΓΟΡΙΘΜΟΣ ΤΕΧΝΙΚΗΣ ΑΠΟΙΚΙΑΣ ΜΥΡΜΗΓΚΙΩΝ

Οι αλγόριθμοι βελτιστοποίησης με βάση την λειτουργία των αποικιών των μυρμηγκιών μελετούνται κυρίως στον κλάδο της Επιχειρησιακής Έρευνας. Πρόκειται για πιθανολογικό υπολογιστικό εργαλείο επίλυσης προβλημάτων εύρεσης βέλτιστου. Προτάθηκε το 1992 από τον Marco Dorigo.



ΣΧΗΜΑ 3.5: *Διάγραμμα Ροής αλγορίθμου τεχνικής αποικίας μυρμηγκιών*
(πηγή: https://www.researchgate.net/figure/275075222_fig4_Flow-chart-of-the-Ant-Colony-Optimization)

3.4 ΜΟΝΤΕ ΚΑΡΛΟ ΠΡΟΣΟΜΟΙΩΣΗ

3.4.1 ΕΙΣΑΓΩΓΗ

Η προσομοίωση Μόντε Κάρλο είναι μια ευέλικτη μέθοδος για την ανάλυση της συμπεριφοράς ορισμένων δραστηριοτήτων ή διαδικασιών που αφορούν την αβεβαιότητα. Μιλώντας γενικά, κάθε πείραμα στο οποίο γίνεται χρήση τυχαίων αριθμών για την εξέταση του προβλήματος, ονομάζεται προσομοίωση Μόντε Κάρλο.
Πήρε το όνομα της από την ομώνυμη πόλη του Μόντε Κάρλο, από το φημισμένο της καζίνο. Η λειτουργία της προσομοίωσης βασίζεται στη γεννήτρια τυχαίων (για την ακρίβεια ψευδοτυχαίων) αριθμών [17].
Η μέθοδος ‘‘γεννήθηκε’’ το 1949 σε ένα άρθρο των N. Metropolis & S. Ulam με τίτλο "Η μέθοδος Monte Carlo" στο Journal of the American Statistics Association [8]. Παρ’όλα αυτά η θεωρητική βάση της μεθόδου ήταν γνωστή πριν από το 1949, αφού στατιστικά προβλήματα λύνονταν μέσω τυχαίας δηγματοληψίας.

3.4.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ-ΜΕΙΟΝΕΚΤΗΜΑΤΑ

Πλεονεκτήματα της Μόντε Κάρλο είναι :

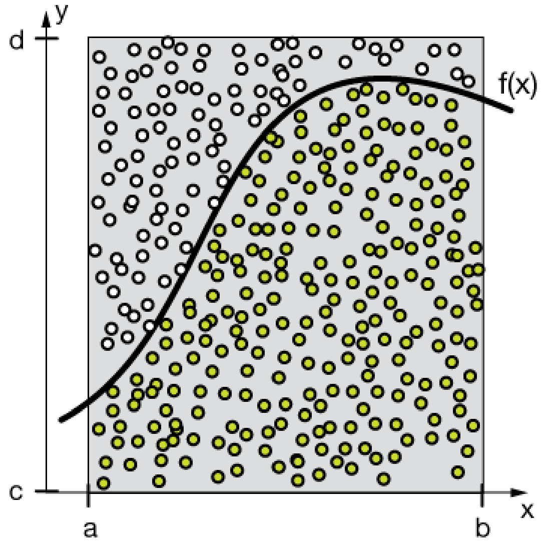
- Ελαστικότητα στην ανάλυση, αφού πρακτικά δεν υπάρχει εμπόδιο στην οποιαδήποτε ανάλυση.
- Ευκολία ανάπτυξης και ευκολία κατανόησης της από τρίτους χωρίς να χρειάζεται ιδιαίτερο μαθηματικό υπόβαθρο.
- Θεωρείται πολύ δυνατή στα πολυδιάστατα προβλήματα, γιατί γενικά η ακρίβεια της εξαρτάται από την πολυπλοκότητα του προβλήματος.

Μειονεκτήματα της είναι:

- Απαιτεί ηλεκτρονικό υπολογιστή.
- Οι υπολογισμοί μπορεί να διαρκέσουν περισσότερο από άλλες μεθόδους.
- Τα αποτελέσματα της δεν είναι τόσο ακριβή σε σχέση με άλλες μεθόδους (όπως ο αλγόριθμος Προσομοιωμένης Ανόπτησης).

3.4.3 ΕΦΑΡΜΟΓΕΣ

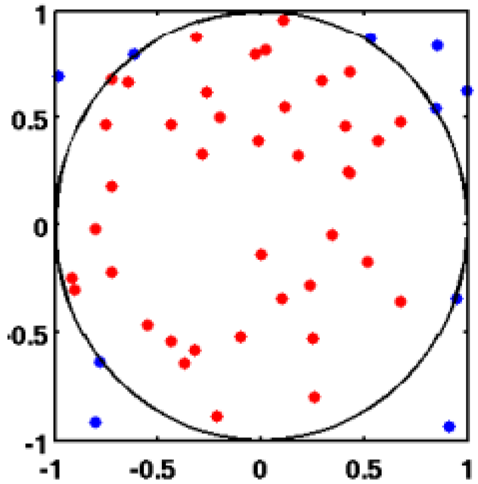
Μια απλή εφαρμογή της μεθόδου, έχει να κάνει με τον προσδιορισμό ολοκληρωμάτων. Η μέθοδος χρησιμοποιεί την τοποθέτηση τυχαίων σημείων εντός του χώρου, ο οποίος περιέχει την συνάρτηση.



ΣΧΗΜΑ 3.6

Το ολοκλήρωμα της συνάρτησης δίνεται από τον λόγο του πλήθους των σημείων που βρίσκονται κάτω από την γραφική παράσταση ως προς το σύνολο των σημείων.

Μία ακόμη ενδιαφέρουσα εφαρμογή της μεθόδου είναι ο υπολογισμός του π.



ΣΧΗΜΑ 3.7
(πηγή: https://www.unige.ch/sciences/astro/files/2713/8971/4086/3_Paltani_MonteCarlo.pdf)

1) Τοποθετώ τυχαία N σημεία (x,y) ομοιόμορφα σε ένα τετράγωνο.
2) Μετρώ τα K σημεία για τα οποία ισχύει $x^2 + y^2 < 1$
3) Το πηλίκο K/N συγκλίνει στο π/4. Προσεγγιστικά δηλαδή για τα τυχαία N σημεία N×π/4 θα πέσουν μέσα στον κύκλο.

3.4.4 ΠΑΡΑΓΩΓΗ ΤΥΧΑΙΩΝ ΑΡΙΘΜΩΝ

Στη Fortran 90 στην οποία θα χρησιμοποιήσουμε την προσομοίωση Μόντε Κάρλο, καλώντας την εντολή: **CALL RANDOM_NUMBER**, κάνουμε χρήση της γεννήτριας ψευδοτυχαίων αριθμών που υπάρχει ενσωματωμένη [3]. Η εντολή αυτή παράγει αριθμούς που βρίσκονται στο διάστημα (0,1) συνεπώς αν χρειάζονται αριθμοί στο διάστημα (A,B) γράφουμε:

```
call random_number (X)
X=A+X*(B-A)
```

Συνίσταται η χρήση της εντολής **CALL RANDOM_SEED** με όρο σχετιζόμενο με το χρόνο έναρξης της εκτέλεσης του προγράμματος, γιατί διαφορετικά οι ψευδοτυχαίοι αριθμοί παραμένουν ίδιοι σε κάθε κλήση της εντολής.
Στον κώδικα για το δικτύωμα έχουν χρησιμοποιηθεί οι παρακάτω εντολές.

```
call date_and_time(values=values)
call random_seed (size=k)
allocate(seed(1:k))
seed(:) = values(8)
call random_seed(put=seed)
```

3.5 ΑΛΓΟΡΙΘΜΟΣ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ

3.5.1 ΕΙΣΑΓΩΓΗ

Το 1953 οι Metropolis et al. ανέπτυξαν μια εκδοχή της μεθόδου Μόντε Κάρλο ώστε να υπολογίζει τις ιδιότητες οποιασδήποτε ουσίας που μπορεί να θεωρηθεί ότι αποτελείται από αλληλοεπιδρώμενα μόρια. Αυτή η εκδοχή ενέπνευσε τους Kirkpatrick et al. να δημιουργήσουν τον αλγόριθμο Προσομοιωμένης Ανόπτησης στις αρχές του 1980 και να τον εφαρμόσουν σε διάφορα συνδυαστικά προβλήματα βελτιστοποίησης [11].
Ο αλγόριθμος προσομοιωμένης ανόπτησης (Simulated Annealing) είναι ένας στοχαστικός αλγόριθμος βελτιστοποίησης. Σκοπός του αλγορίθμου είναι να βρει το ολικό μέγιστο της υπό βελτιστοποίηση συνάρτησης. Το όνομα και η προέλευση πρόερχονται από την ανόπτηση στην μεταλλουργία, τεχνική στην οποία ενα μέταλλο ή κράμα, υποβάλλεται σε θερμική κατεργασία, προκειμένου στην συνέχεια υποβαλλόμενο σε ψύξη να βελτιωθεί η ευκαμψία του.

Στην φυσική, η κάθε θέση όλων των ατόμων ενός συστήματος, ελέγχεται από τον πιθανοτικό παράγοντα Boltzmann $e^{-\frac{E(POS)}{K^T}}$ όπου E(pos) είναι η ενέργεια της διαμόρφωσης, T η θερμοκρασία μετρούμενη σε Κέλβιν και k η σταθερά Boltzmann k= 1.38065052×10^{-23}J/K

Η μέθοδος Metropolis ήταν ένα ακριβές αντίγραφο αυτής της φυσικής διαδικασίας, η οποία προσομοίωνε μια συλλογή από άτομα σε θερμοδυναμική ισορροπία σε δεδομένη θερμοκρασία [11].

3.5.2 ΠΛΕΟΝΕΚΤΗΜΑΤΑ-ΜΕΙΟΝΕΚΤΗΜΑΤΑ

Πλεονεκτήματα του αλγορίθμου είναι:

- Μπορεί να ασχοληθεί με αυθαίρετα συστήματα και συναρτήσεις κόστους, με μη γραμμικά μοντέλα που έχουν πολλούς περιορισμούς. Είναι μια εύρωστη και γενική τεχνική.
- Στατιστικά εγγυάται την εύρεση του ολικού βέλτιστου.
- Είναι σχετικά εύκολος να προγραμματιστεί, ακόμα και σε σύνθετα προβλήματα.

Στα μειονεκτήματα του αλγορίθμου περιλαμβάνονται:

- Η ταχύτητα εκτέλεσης του προγράμματος.
- Σε προβλήματα που έχουν λίγα τοπικά μέγιστα, η χρήση της μεθόδου είναι συνήθως περιττή.
- Η μέθοδος δεν μπορεί με κάποιο τρόπο να ενημερώσει άμα έχει βρει την ολική βέλτιστη λύση.

3.5.3 ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ

Για να μπορέσει να χρησιμοποιηθεί ο αλγόριθμος Προσομοιωμένης Ανόπτησης σε προβλήματα που δεν ανήκουν από θερμοδυναμικά συστήματα, πρέπει να αποσαφηνιστούν τα παρακάτω στοιχεία [7]:

1. Περιγραφή των πιθανών διαμορφώσεων του συστήματος.
2. Μια γεννήτρια τυχαίων αλλαγών στην διαμόρφωση. Αυτές οι αλλαγές αποτελούν τις επιλογές που παρουσιάζονται στο σύστημα.
3. Μια αντικειμενική συνάρτηση E (ανάλογη της ενέργειας) της οποίας η ελαχιστοποίηση είναι ο σκοπός της διαδικασίας.
4. Μια παράμετρος ελέγχου T (ανάλογη της θερμοκρασίας) και ένα πρόγραμμα ανόπτησης το οποίο περιγράφει τον τρόπο που μειώνεται, για παράδειγμα μετά από πόσες τυχαίες αλλαγές στην διαμόρφωση, το T μειώνεται κατά ένα ποσοστό (βήμα), και ποιο είναι αυτό το ποσοστό.

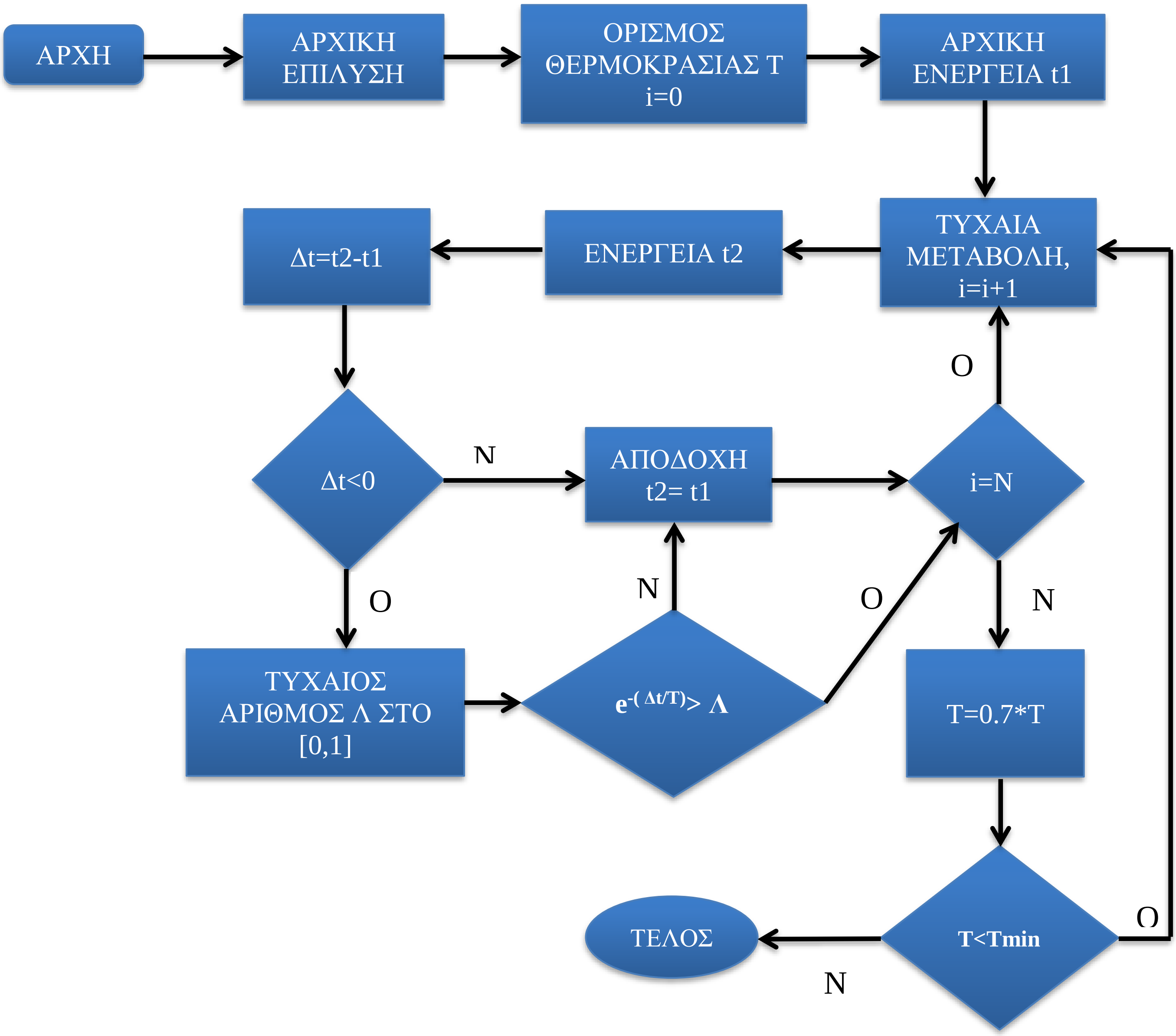
3.5.4 ΔΙΑΔΙΚΑΣΙΑ

Ο αλγόριθμος ξεκινάει από την αρχική επίλυση του προβλήματος, έστω στατική επίλυση ενός δικτυώματος όπως γίνεται στην παρούσα εργασία. Παράγει μια αρχική λύση (ή αρχική ενέργεια), το βάρος του δικτυώματος. Στη συνέχεια ''γεννάει'' μια νέα δοκιμαστική λύση. Αν η νέα λύση είναι καλύτερη της αρχικής (στην προκειμένη χαμηλότερο βάρος), τη δέχεται και την κρατά ως νέα τρέχουσα λύση. Διαφορετικά, δεν την απορρίπτει αυτόματα όπως θα συνέβαινε στην προσομοίωση Μόντε Κάρλο, αλλά ανάλογα με την πιθανότητα αποδοχής τη δέχεται ή την απορρίπτει. Η πιθανότητα αποδοχής εξαρτάται από την αρχική και την τρέχουσα λύση σε συνδυασμό με την τιμή της θερμοκρασίας [5], [6].

Υπολογίζεται από την παρακάτω σχέση:

$$\mathbf{Pr(\Delta t, T)} = e^{\left(-\frac{\Delta t}{T}\right)} \;,\; \mathbf{\Delta t = t_2 - t_1 > 0} \qquad \mathbf{(3.1)}$$

Όπου t_1 είναι η αρχική λύση, δηλαδή το αρχικό βάρος του δικτυώματος, t_2 η τρέχουσα λύση,και T η θερμοκρασία. Γενικά η αρχική τιμή της θερμοκρασίας είναι αρκετά μεγαλύτερη απο το Δt, και στην συνέχεια μειώνεται κατά ένα ποσοστό, συνήθως 10-30%.
Λόγω της αρχικής υψηλής τιμής της θερμοκρασίας, σχεδόν όλες οι λύσεις γίνονται δεκτές [7]. Στην συνέχεια μείωνεται η θερμοκρασία, οπότε περιορίζονται οι αποδεκτές τιμές, έως ότου δεν γίνεται καμία δεκτή.



ΣΧΗΜΑ 3.8: *Διάγραμμα Ρόης αλγορίθμου Προσομοιωμένης Ανόπτησης*
(πηγή: https://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec10.pdf)

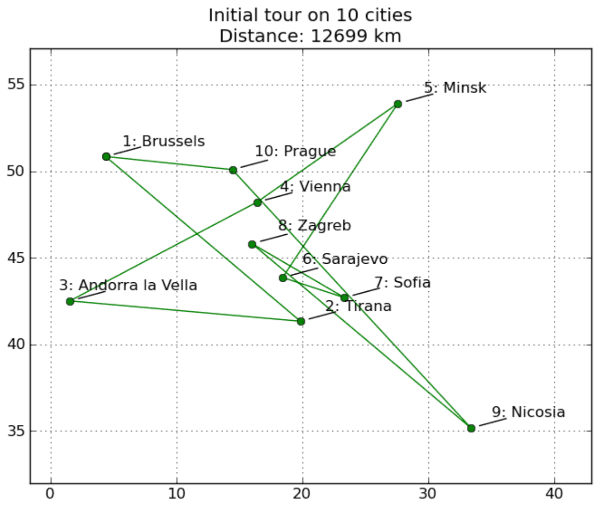
3.5.5 ΕΦΑΡΜΟΓΕΣ

Χαρακτηριστικό παράδειγμα Προσομοιωμένης Ανόπτησης είναι το πρόβλημα **'Traveling Salesman's Problem'**. Σκοπός είναι να βρεθεί η πιο γρήγορη διαδρομή ανάμεσα σε συγκεκριμένες τοποθεσίες που πρέπει να ακολουθήσει ο ταξιδιώτης. Η κάθε τοποθεσία πρέπει να περαστεί μόνο μια φορά και η διαδρομή πρέπει να τερματιστεί όταν ο ταξιδιώτης βρεθεί ξανά στην αρχική τοποθεσία. Το πρόβλημα αυτό μπορεί να επιλυθεί σχετικά εύκολα, υπολογίζοντας την κάθε πιθανή διαδρομή ξεχωριστά, όταν ο αριθμός των τοποθεσιών είναι μικρός. Έστω ότι πρέπει να ταξιδεύσει σε 5 τοποθεσίες. Ο αριθμός των διαδρομών που ξεκινούν απο την Α τοποθεσία και καταλήγουν στην Α είναι 4! =24. Όταν είναι 10 τοποθεσίες ο αριθμός είνιαι 9!= 362880. Γίνεται κατανοητό ότι όταν ο αριθμός αυξάνεται σημαντικά, δεν είναι δυνατό να επιλυθεί το πρόβλημα με τέτοιο τρόπο [20].
Ο τρόπος με τον οποίο επιλύνεται αυτό το πρόβλημα μέσω του αλγόριθμου Προσομοιωμένης Ανόπτησης είναι ο εξής:

1. **Διαμόρφωση**: Οι πόλεις (τοποθεσίες) είναι αριθμημένες i=1,..,N και έχουν συντεταγμένες (x_i,y_i). Διαμόρφωση είναι η μετάθεση των αριθμων 1,..,N ώστε να προκύπτει η σειρά με την οποία θα γίνει η διαδρομή.
2. **Αναδιατάξεις**: Μια αποτελεσματική σειρά κινήσεων είναι η αφαίρεση ενός κομματιού της διαδρομής και αντικατάσταση του σε ένα άλλο τυχαίο τμήμα μεταξύ 2 πόλεων της διαδρομής.
3. **Αντικειμενική Συνάρτηση**: Στην πιο απλή μορφή του προβλήματος, E είναι το συνολικό μήκος της διαδρομής.

$$E=\sum_{i}^{N}\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

4. **Πρόγραμμα Ανόπτησης**: Αυτό το βήμα απαιτεί πειραματισμούς. Παράγονται πρώτα κάποιες τυχαίες αναδιατάξεις ώστε να προκύψει μια διαφορά τιμών ΔΕ. Επιλέγεται μια αρχική τιμή Τ, η οποία είναι αρκετά μεγαλύτερα από την διαφορά ΔΕ. Η τιμή Τ, μειώνεται κατά 30% μετά από ένα συγκεκριμένο αριθμό βημάτων. Κρατιέται σταθερή την τιμή Τ για αριθμό αναδιατάξεων 100Ν (τυχαίο νούμερο). Όταν το Τ φτάσει κοντά στο 0, τερματίζεται ο αλγόριθμος.



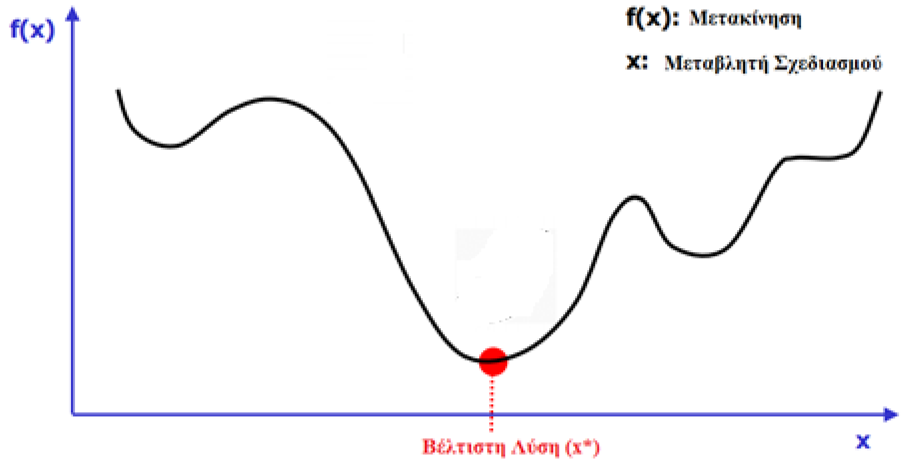
***ΣΧΗΜΑ 3.9:** Γράφημα διαδρομών μεταξύ πόλεων*

ΚΕΦΑΛΑΙΟ 4: ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΚΑΤΑΣΚΕΥΩΝ

4.1 ΕΙΣΑΓΩΓΗ

Η έρευνα και ο σχεδιασμός για τη βελτιστοποίηση στις κατασκευές έχουν τραβήξει την προσοχή τα τελευταία 20 χρόνια. Σαν αποτέλεσμα, πληθώρα από μεθόδους και αλγορίθμους έχουν αναπτυχθεί, που επικεντρώνονται σε διάφορα προβλήματα και σενάρια. Συνήθεις τομείς βελτιστοποίησης στις κατασκευές και κυρίως στα δικτυώματα είναι η βελτιστοποίηση σχήματος, η βελτιστοποίηση μεγέθους και η βελτιστοποίηση τοπολογίας.

Βέλτιστη λύση σε κατασκευή για συγκεκριμένη μεταβλητή σχεδιασμού υπάρχει όταν βρεθεί το ολικό ελάχιστο της συνάρτησης σχεδιασμού:



ΣΧΗΜΑ 4.1 : *Γραφική Αναπαράσταση βέλτιστου σχεδιασμού*
(πηγή: http://web.mit.edu/16.810/www/16.810_L8_Optimization.pdf)

4.1.1 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Ο Lucien Schmit το 1960 ήταν ο πρώτος που αναγνώρισε τη δυνατότητα συνδυασμού τεχνικών βελτιστοποίησης με τη δομική σχεδίαση [31]. Παρακάτω παρουσιάζονται οι βασικές κατηγορίες βελτιστοποίησης στις κατασκευές:

- Βελτιστοποίηση Σχήματος
- Βελτιστοποίηση Μεγέθους
- Βελτιστοποίηση Τοπολογίας

4.2 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΗΜΑΤΟΣ

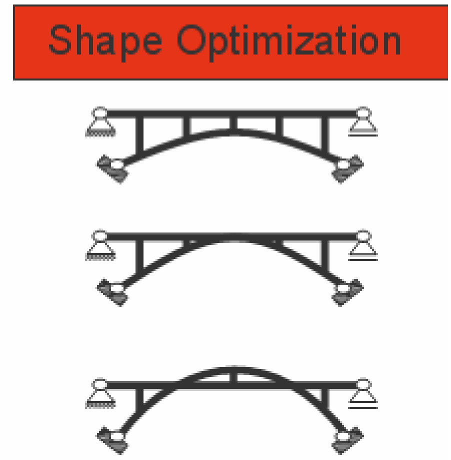
4.2.1 ΕΙΣΑΓΩΓΗ

Στόχος της βελτιστοποίησης σχήματος είναι η εύρεση του βέλτιστου σχήματος της κατασκευής, δεδομένης της τοπολογίας της. Η κατασκευή καθίσταται πιο οικονομική και λειτουργική. Ένα πρόβλημα βελτιστοποίησης σχήματος περιγράφεται με τον εξής τρόπο:

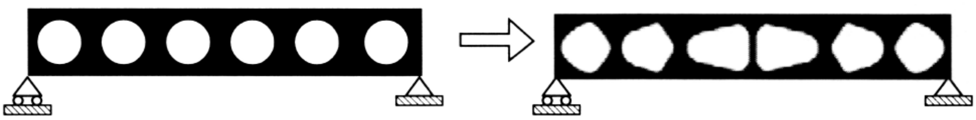
$$\min_{\Omega \in \mathcal{U}_{ad}} J(\Omega),$$

όπου U_{ad} είναι το σύνολο των αποδεκτών σχημάτων, J μεταβλητή σχεδιασμού και Ω το πεδίο ορισμού της μεταβλητής.

Μεταβλητή σχεδιασμού στην παρούσα εργασία είναι οι συντεταγμένες των κόμβων του δικτυώματος. Παράμετρος βελτιστοποίησης είναι η ελαχιστοποίηση του βάρους του δικτυώματος.



ΣΧΗΜΑ 4.2: *Βελτιστοποίηση σχήματος γέφυρας*
(πηγή: https://steelandglassart.com/topology-optimization-truss-structure-diy-plans/)



ΣΧΗΜΑ 4.3: *Βελτιστοποίηση σχήματος δοκού*
(πηγή: http://topshape.bm-rd.com/topology-optimization-overview/)

4.2.2 ΚΡΙΤΗΡΙΑ ΑΣΤΟΧΙΑΣ

1) Μέγιστη Μετακίνηση Κόμβων

Οι περιορισμοί μετακινήσεων είναι σημαντικό στοιχείο στις κατασκευές. Η κατασκευή δεν επιτρέπεται να υπερβεί το όριο. Γενικά το όριο κυμαίνεται ανάλογα με το πλάτος ανοίγματος της κατασκευής π.χ. $\delta_{max} = L/300$. Συνήθως το όριο κυμαίνεται μεταξύ $L/500 < \delta_{max} < L/150$. Στην παρούσα εργασία το όριο της μέγιστης μετακίνησης είναι η διπλάσια μετακίνηση της μέγιστης μετακίνησης των κόμβων της στατικής επίλυσης πριν την βελτιστοποίηση σχήματος [9]. Δηλαδη: $\delta_{max} < 2\,\delta_{max,αρχικό}$

2) Σταθερότητα Κατασκευής

Το δικτύωμα πρέπει να είναι κινηματικά σταθερό. Ένας τρόπος για να διαπιστωθεί άμα η κατασκευή είναι κινηματικά σταθερή, είναι να υπολογιστεί η ορίζουσα του μητρώου δυσκαμψίας. Άμα προκύψει μηδενική τότε η κατασκευή δεν είναι κινηματικά σταθερή.

3) Περιορισμός Μορφολογίας

Για να μην αλλάζει δραματικά η μορφολογία του δικτυώματος, οι κόμβοι στους οποίους υπάρχουν στηρίξεις δεν μετακινούνται. Επιπρόσθετα οι συντεταγμένες κατά y των κόμβων, που βρίσκονται στην ίδια ράβδο μεταξύ δύο στηρίξεων, παραμένουν σταθερές. Τέλος, υπάρχει περιορισμός στις θέσεις των κόμβων που εξετάζονται ώστε να μην συμπέσουν ενδιάμεσοι κόμβοι, και πιο συγκεκριμένα, οι συντεταγμένες του κάθε κόμβου μπορούν να μετατοπιστούν στον x άξονα και y άξονα κατά αποστάσεις που κυμαίνονται από -0.2L έως 0.2L, όπου L είναι το μήκος της μικρότερης ράβδου του δικτυώματος.

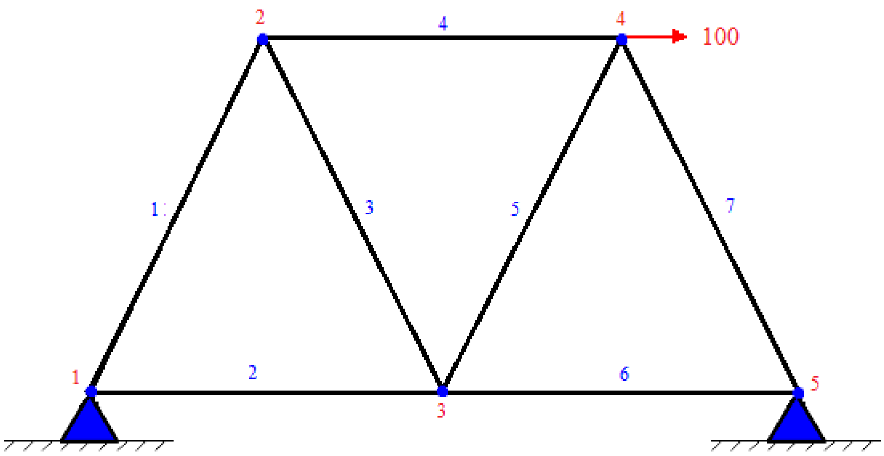
4.2.3 ΔΙΑΔΙΚΑΣΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΣΧΗΜΑΤΟΣ

Χρησιμοποιούνται δύο διαφορετικές μέθοδοι για τη βελτιστοποίηση που φαίνονται παρακάτω.

4.2.3.1 ΜΕΣΩ ΜΟΝΤΕ ΚΑΡΛΟ

1) Στατική επίλυση δικτυώματος, από την οποία προκύπτει το αρχικό βάρος του δικτυώματος.
2) Σε τυχαίο κόμβο, τυχαία μετατόπιση L_1 κατά x και L_2 κατά y.
3) Νέα στατική επίλυση του δικτυώματος.
4) Εάν το νέο βάρος είναι μικρότερο του αρχικού: εάν ικανοποιούνται τα κριτήρια αστοχίας τότε κρατάμε το νέο βάρος και τις συντεταγμένες του κόμβου. Αλλιώς επαναφορά στο βήμα 2.
5) Επαναφορά στο βήμα 2
6) Μόλις τελειώσουν οι επαναλήψεις έχουμε κρατήσει τις βέλτιστες συντεταγμένες και το τελικό βάρος.

Ο αριθμός επαναλήψεων είναι 10.000×N όπου N ο αριθμός των κόμβων.

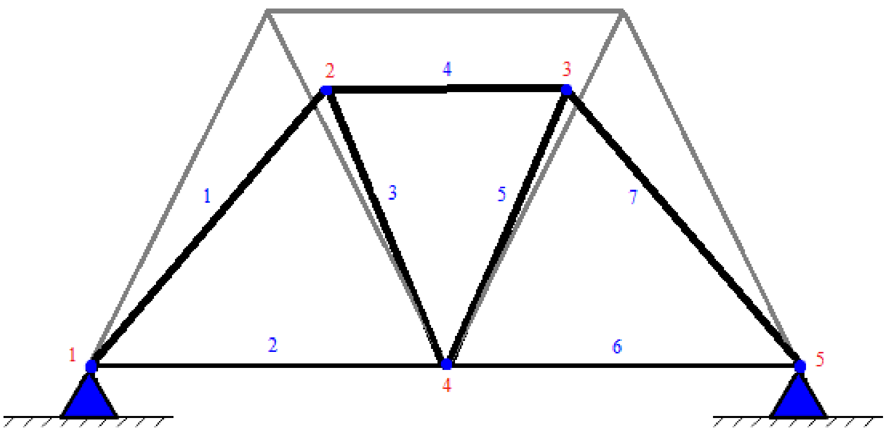


ΣΧΗΜΑ 4.4: *Δικτύωμα 5 κόμβων*

Για A=0.005 m^2, E=210 GPA, η στατική επίλυση χωρίς βελτιστοποίηση δίνει βάρος 2932.8135 kg.

Με βελτιστοποίηση σχήματος προκύπτει νέο βάρος 2529.079 kg.

Άρα επιτυγχάνεται μείωση βάρους κατά 14%.



ΣΧΗΜΑ 4.5: *Δικτύωμα 5 κόμβων βελτιστοποιημένο σε σχήμα*

Πίνακας 4.1: Συντεταγμένες Κόμβων μέσω Μόντε Κάρλο δικτυώματος 5 κόμβων

Κόμβος	X(m)	Y(m)
1	0	0
2	6,674	7,836
3	9,974	0
4	13,478	7,903
5	20	0

4.2.3.2 ΜΕΣΩ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ

1) Στατική επίλυση δικτυώματος, από την οποία προκύπτει το αρχικό βάρος του δικτυώματος.
2) Η αρχική Ενέργεια t1 ισούται με το αρχικό βαρος
3) Η θερμοκρασια Τ ισούται με Τ=10.000
4) Σε τυχαίο κόμβο, τυχαία μετατόπιση L_1 κατά x και L_2 κατά y.
5) Νέα στατική επίλυση του δικτυώματος
6) Η ενέργεια t2 ισούται με το νέο βάρος
7) Εάν το νέο βάρος είναι μικρότερο του αρχικού: εάν ικανοποιούνται τα κριτήρια αστοχίας τότε κρατάμε το νέο βάρος και τις συντεταγμένες του κόμβου και επιστρέφουμε στο βήμα 4. Αλλιώς κοιτάμε την πιθανότητα αποδοχής $P= e^{-(\Delta t/T)}$
8) Παραγωγή τυχαίου αριθμού Λ στο διάστημα [0,1]
9) Αν Ρ>Λ και εάν ικανοποιούνται τα κριτήρια αστοχίας τότε κρατάμε το νέο βάρος ως αρχικό και τις συντεταγμένες του κόμβου και επιστρέφουμε στο βήμα 4.
10) Τα βήματα 4-9 επαναλαμβάνονται για N=1000×(αριθμό κόμβων) φορές.
11) Η θερμοκρασία μειώνεται T=0,7×T και έαν $T>T_{min}=0,5$ επιστρέφουμε στο βήμα 4.
12) Μόλις τελειώσουν οι επαναλήψεις έχουμε κρατήσει τις βέλτιστες συντεταγμένες και το τελικό βάρος.

Στο ίδιο παράδειγμα δικτύωματος με 5 κόμβους που βρίσκεται στην σελίδα 40 λύνοντας με την μέθοδο Προσομοιωμένης Ανόπτησης προκύπτουν τα εξής αποτελέσματα:

Πίνακας 4.2: Συντεταγμένες Κόμβων μέσω Προσομοιωμένης Ανόπτησης δικτυώματος 5 κόμβων

Κόμβος	X(m)	Y(m)
1	0.0000000	0.0000000
2	6.1556106	7.8460779
3	8.0202971	0.0000000
4	12.995628	7.8967490
5	20.000000	0.0000000

- Τελικό βάρος δικτυώματος: 2525,662 kg.

Παρατηρούμε ότι είναι 2525,662 kg < 2529,079 kg που είναι το βάρος που προέκυψε με την μέθοδο Μόντε Κάρλο. Προκύπτει δηλαδή διαφορά 0,16% μεταξύ των δύο μεθόδων.

4.3 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕΓΕΘΟΥΣ

4.3.1 ΕΙΣΑΓΩΓΗ

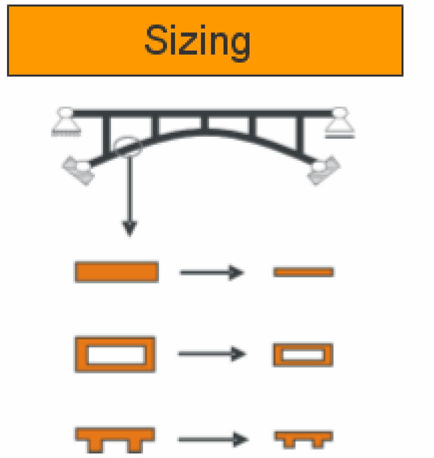
Στόχος της βελτιστοποίησης μεγέθους (Size Optimization) είναι να ελαχιστοποιηθούν τα υλικά της κατασκευής καθώς παράλληλα μειώνεται το βάρος της, χωρίς όμως να διακινδυνεύει την ασφάλεια και την απόδοση της.

Η μαθηματική σχέση που περιγράφει το πρόβλημα είναι:

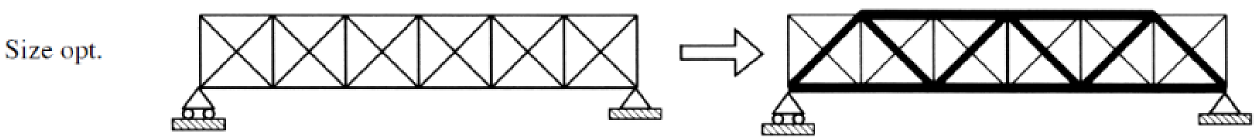
$$\mathbf{minW}=\sum_i^n p_i L_i A_i$$

$$\mathbf{A_{min}<A_i<A_{max}}$$

όπου W είναι το βάρος της κατασκευής, p_i το ειδικό βάρος, L_i το μήκος και A_i το εμβαδόν διατομής του i στοιχείου. Στη παρούσα εργασία, η βελτιστοποίηση μεγέθους επιτυγχάνεται, ορίζοντας το μικρότερο δυνατό εμβαδόν στην κάθε διατομής, δεδομένης της αξονικής που ασκείται στην κάθε ράβδο.



ΣΧΗΜΑ 4.6: *Βελτιστοποίηση διαστασιόλογησης*
Διατομής γέφυρας
(πηγή: https://steelandglassart.com/topology-optimization-truss-structure-diy-plans/)



ΣΧΗΜΑ 4.7: *Βελτιστοποίηση διαστασιόλογησης*
Διατομής γέφυρας
(πηγή: http://topshape.bm-rd.com/topology-optimization-overview/)

4.3.2 ΚΡΙΤΗΡΙΑ ΑΣΤΟΧΙΑΣ

Ισχύουν οι ίδιοι περιορισμοί για την μέγιστη μετακίνηση κόμβων και την σταθερότητα της κατασκευής, που ισχύουν στην Βελτιστοποίηση Σχήματος.

4.3.3 ΔΙΑΣΤΑΣΙΟΛΟΓΗΣΗ

Η διαστασιολόγηση του δικτυώματος θα γίνει σύμφωνα με τον Ευρωκώδικα 3. Το υλικό των ράβδων είναι S235 και η επιλογή των διατομών θα γίνει αποκλειστικά μεταξύ κοίλων τετραγωνικών διατομών. Πρέπει να ληφθεί υπόψιν ο λυγισμός σε περίπτωση θλιπτικών αξονικών δυνάμεων στις ράβδους.

Για την κάθε ράβδο:

- Για ΕΦΕΛΚΥΣΜΟ (EN1993-1-1§6.2.3)

$$\mathbf{N_{ed}/N_{pl,rd} \leq 1 \ \ \mu\varepsilon \ \ N_{pl,rd} = \frac{A*fy}{\gamma\mu 0}} \qquad \textbf{(5.1)}$$

όπου N_{ed} είναι η αξονική της ράβδου,Α το εμβαδόν της διατομής, fy η τάση διαρροής του χάλυβα και $\gamma_{\mu o}$ ο συντελεστής ασφαλείας.

- Για ΘΛΙΨΗ

Αντοχή Μελών έναντι καμπτικού Λυγισμού (EN1993-1-1§6.3.1)

$$\mathbf{N_{ed}/N_{b,rd} \leq 1 \ \mu\varepsilon \ \ N_{b,rd} = \frac{x*A*fy}{\gamma\mu 1}} \qquad \textbf{(5.2)}$$

Μειωτικός συντελεστής λυγισμού:

$$\boldsymbol{\chi = \frac{1}{\Phi+\sqrt{\Phi^2-\bar{\lambda}^2}} \leq 1{,}0 \ \mu\varepsilon \ \Phi = 0{,}5[1+\alpha(\bar{\lambda}^2\text{-}0{,}2)+\bar{\lambda}^2)} \qquad \textbf{(5.3)}$$

Ο συντελεστής ατελειών α που αντιστοιχεί στην ανάλογη καμπύλη λυγισμού θα λαμβάνεται από τον ακόλουθο πίνακα (EN1993-1-1 Πίν. 6.1)

Καμπύλη λυγισμού	a_0	a	b	c	d
Συντελεστής ατελειών **α**	0,13	0,21	0,34	0,49	0,76

Ανηγμένη λυγηρότητα:

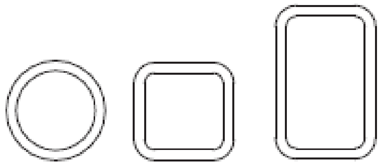
$$\bar{\boldsymbol{\lambda}}= \sqrt{\frac{A*fy}{Ncr}}= \frac{Lcr}{i}\frac{1}{\lambda 1} \text{ για διατομές κατηγορίας 1,2,3} \qquad (5.4)$$

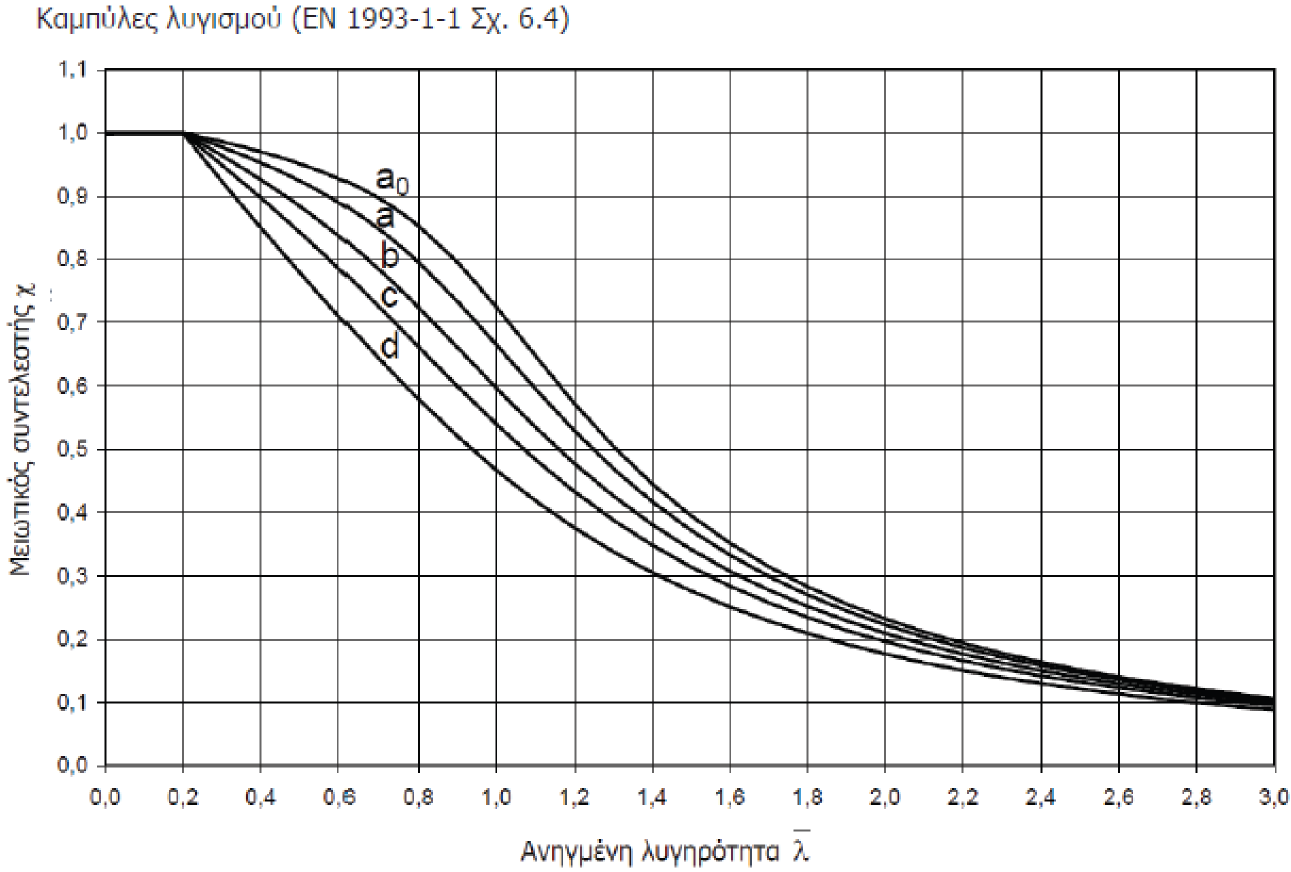
$$\mathbf{Ncr}=\frac{\pi^2 EI}{Lcr^2} \quad \lambda_1 = \pi\sqrt{\frac{E}{fy}} \qquad (5.5)$$

$$\varepsilon=\sqrt{\frac{235}{fy}} \qquad (5.6)$$

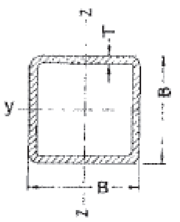
$$\mathbf{i}=\sqrt{\frac{I}{A}} \qquad (5.7)$$

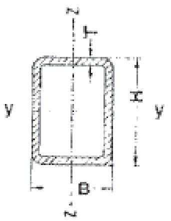
Επιλογή καμπύλης λυγισμού (ΕΝ1993-1-1 Πίν. 6.2):

Κοίλες Διατομές		Εν θερμώ έλαση	Κάθε	a	a_0
		Ψυχρή έλαση	Κάθε	c	c



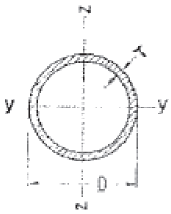
ΣΧΗΜΑ 4.8: *Καμπύλες λυγισμού*





Hohlprofile

Rechenwerte $M_{pl,y}$ $W_{pl,y}$ I_T C_t siehe Seiten 27, 29, 31
Mantelfläche U siehe Seite 39



B mm	T mm	A cm²	M kg/m	I cm⁴	W cm³	i cm	B mm	T mm	A cm²	M kg/m	I cm⁴	W cm³	i cm	B mm	T mm	A cm²	M kg/m	I cm⁴	W cm³	i cm
Quadrat-Hohlprofile (warmgefertigt, nahtlos oder geschweißt), Auswahl aus **DIN EN 10210-2, Ausgabe November 1997**																				
40	3	4,34	3,41	9,78	4,89	1,50	**120**	10	42,9	33,7	852	142	4,46	**220**	8	67,2	52,7	5 002	455	8,63
	4	5,59	4,39	11,8	5,91	1,45	**140**	5	26,7	21,0	807	115	5,50		10	82,9	65,1	6 050	550	8,54
50	3	5,54	4,35	20,2	8,08	1,91		6,3	33,3	26,1	984	141	5,44		12,5	102	80,1	7 254	659	8,43
	4	7,19	5,64	25,0	9,99	1,86		8	41,6	32,6	1 195	171	5,36	**250**	6,3	61,0	47,9	6 014	481	9,93
60	3	6,74	5,29	36,2	12,1	2,32		10	50,9	40,0	1 416	202	5,27		8	76,8	60,3	7 455	596	9,86
	4	8,79	6,90	45,4	15,1	2,27	**150**	5	28,7	22,6	1 002	134	5,90		10	94,9	74,5	9 055	724	9,77
	5	10,7	8,42	53,3	17,8	2,23		6,3	35,8	28,1	1 223	163	5,85		16	147	115	13 267	1 061	9,50
70	3	7,94	6,24	59,0	16,9	2,73		8	44,8	35,1	1 491	199	5,77	**260**	8	80,0	62,8	8 423	648	10,3
	4	10,4	8,15	74,7	21,3	2,68		10	54,9	43,1	1 773	236	5,68		10	89,9	77,7	10 242	788	10,2
	5	12,7	9,99	88,5	25,3	2,64	**160**	6,3	38,3	30,1	1 499	187	6,26		12,5	122	95,8	12 365	951	10,1
80	4	12,0	9,41	114	28,6	3,09		8	48,0	37,6	1 831	229	6,18		16	153	120	15 061	1 159	9,91
	5	14,7	11,6	137	34,2	3,05		10	58,9	46,3	2 186	273	6,09	**300**	8	92,8	72,8	13 128	875	11,9
	6,3	18,1	14,2	156	40,5	2,99		12,5	72,1	56,6	2 576	322	5,98		10	115	90,2	16 026	1 068	11,8
90	4	13,6	10,7	166	37,0	3,50	**180**	6,3	43,3	34,0	2 168	241	7,07		12,5	142	112	19 442	1 296	11,7
	5	16,7	13,1	200	44,4	3,45		8	54,4	42,7	2 661	296	7,00		16	179	141	23 855	1 590	11,5
	6,3	20,7	16,2	238	53,0	3,40		10	66,9	52,5	3 193	355	6,91	**350**	8	109	85,4	21 129	1 207	13,9
100	4	15,2	11,9	232	46,4	3,91		12,5	82,1	64,4	3 790	421	6,80		10	135	106	25 884	1 479	13,9
	5	18,7	14,7	279	55,9	3,86	**200**	6,3	48,4	38,0	3 011	301	7,89		12,5	167	131	31 541	1 802	13,7
	6,3	23,2	18,2	336	67,1	3,80		8	60,8	47,7	3 709	371	7,81		16	211	166	38 942	2 225	13,6
120	5	22,7	17,8	498	83,0	4,68		10	74,9	58,8	4 471	447	7,72	**400**	10	155	122	39 128	1 956	15,9
	6,3	28,2	22,2	603	100	4,62		12,5	92,1	72,3	5 336	534	7,61		12,5	192	151	47 839	2 392	15,8
	8	35,2	27,6	726	121	4,55	**220**	6,3	53,4	41,9	4 049	368	8,71		16	243	191	59 344	2 967	15,6

ΣΧΗΜΑ 4.9: *Πίνακας κοίλων τετραγωνικών διατομών*

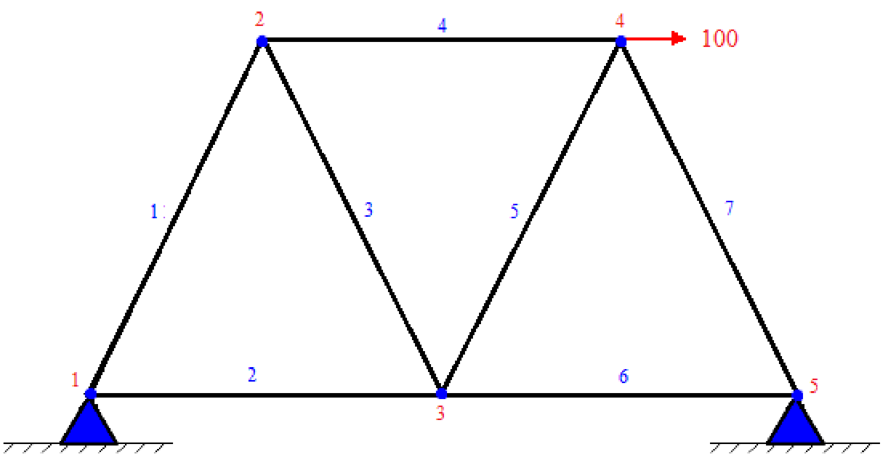
4.3.4 ΔΙΑΔΙΚΑΣΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΜΕΓΕΘΟΥΣ

Χρησιμοποιούνται δύο διαφορετικές μέθοδοι για τη βελτιστοποίηση που φαίνονται παρακάτω.

4.3.4.1 ΜΕΣΩ ΜΟΝΤΕ ΚΑΡΛΟ

1) Στατική επίλυση δικτυώματος από την οποία προκύπτει το αρχικό βάρος του δικτυώματος.
2) Για κάθε ράβδο, παραγωγή τυχαίου αριθμού Ν στο διάστημα (0.2,1), και υπολογισμός νέου εμβαδού διατομής Α=Α×Ν
3) Αντιστοίχιση των διατομών των ράβδων με τις αμέσως μεγαλύτερες από τον πίνακα κοίλων τετραγωνικών διατομών.
4) Νέα στατική επίλυση δικτυώματος.
5) Έλεγχος για την κάθε ράβδο άμα αντέχει σε λυγισμό ή εφελκυσμό ανάλογα με την αξονική δύναμη που της ασκείται. Αν κάποια ράβδος δεν αντέχει επιστροφή στο βήμα 2
6) Υπολογισμός βάρους. Αν είναι μικρότερο από το ελάχιστο, θεωρούμε το νέο βάρος ως ελάχιστο και κρατάμε τις διατομές των ράβδων. Επιστροφή στο βήμα 2.
7) Τα βήματα 2-6 επαναλαμβάνονται για Ν=100.000 φορές
8) Μόλις τελειώσουν οι επαναλήψεις έχουμε κρατήσει τις βέλτιστες διατομές και το τελικό βάρος.

Στο παρακάτω δικτύωμα 5 κόμβων:



Για Α=0.005 m^2,Ε=210 GPA, η επίλυση μέσω μεθόδου Μόντε Κάρλο δίνει τα παρακάτω αποτελέσματα.

Πίνακας 4.3: Εμβαδόν Διατομής Ράβδων μέσω Μόντε Κάρλο δικτυώματος 5 κόμβων

Ράβδος	Εμβαδόν (cm^2)
1	12.7
2	10.7
3	28.2
4	12.7
5	14.7
6	21.3
7	35.2
Συνολικό Βάρος (kg)	**1162,72**

- Τελικό Βάρος δικτυώματος: 1162,72 kg (έναντι αρχικού βάρους 2932,81 kg)

4.3.4.2 ΜΕΣΩ ΠΡΟΣΟΜΟΙΩΜΕΝΗΣ ΑΝΟΠΤΗΣΗΣ

1) Στατική επίλυση δικτυώματος,από την οποία προκύπτει αρχικό βάρος του δικτυώματος.
2) Η αρχική Ενέργεια t1 ισούται με το αρχικό βάρος
3) Η θερμοκρασία T ισούται με T=10.000
4) Σε τυχαία ράβδο τοποθετούμε τυχαία διατομή
5) Νέα στατική επίλυση δικτυώματος.
6) Η ενέργεια t2 ισούται με το νέο βάρος
7) Έλεγχος σε θλίψη και εφελκυσμό των ράβδων. Εάν κάποια ράβδος δεν αντέχει επιστροφή στο βήμα 4.
8) Εαν Δt=t2-t1 <0 τότε κρατάμε το νέο βάρος ως αρχικό, t1=t2 και αποδεχόμαστε την καινούρια διατομή. Επιστροφή στο βήμα 4
9) Αλλιώς κοιτάμε την πιθανότητα αποδοχής $P= e^{-(\Delta t/T)}$
10) Παραγωγή τυχαιόυ αριθμού Λ στο διάστημα [0,1]
11) Αν P>Λ τότε κρατάμε το νέο βάρος ως αρχικό, t1=t2 και αποδεχόμαστε την καινούρια διατομή και επιστρέφουμε στο βήμα 4.
12) Τα βήματα 4-11 επαναλαμβάνονται για N=100×(αριθμό ράβδων) φορές
13) Η θερμοκρασία μειώνεται T=0,7×T και εάν T >Tmin=0,5 επιστρέφουμε στο βήμα 4
14) Μόλις τελειώσουν οι επαναλήψεις έχουμε κρατήσει τις βέλτιστες διατομές και το τελικό βάρος.

Για το δικτύωμα με τους 5 κόμβους προκύπτουν τα παρακάτω εμβαδά των διατομών:

Πίνακας 4.4: Εμβαδόν Διατομής Ράβδων μέσω Προσομοιωμένης Ανόπτησης δικτυώματος 5 κόμβων

Ράβδος	Εμβαδόν (cm^2)
1	10.7
2	18.7
3	23.1
4	14.7
5	12.7
6	14.7
7	42.9
Συνολικό Βάρος (kg)	**1158.37**

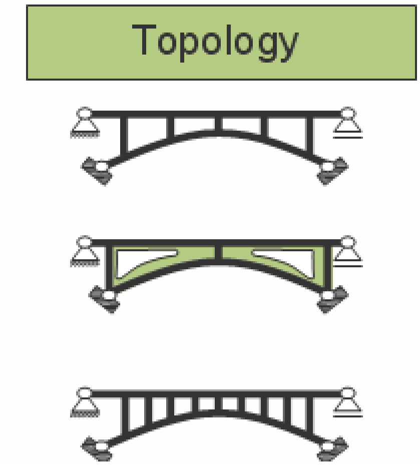
➢ Τελικό βάρος δικτυώματος: 1158,37 kg (έναντι αρχικού βάρους 1162,72 kg)
Παρατηρούμε πάλι ότι η Προσομοιωμένη Ανόπτηση παράγει καλύτερα αποτελέσματα.

4.4 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΟΠΟΛΟΓΙΑΣ

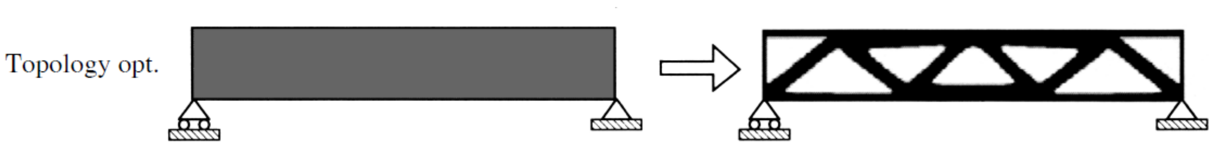
4.4.1 ΕΙΣΑΓΩΓΗ

Τις τελευταίες δεκαετίες υπήρξε τεράστια εξέλιξη στον τομέα της βελτιστοποίησης τοπολογίας. Πατέρες της συγκεκριμένης μεθόδου βελτιστοποίησης είναι οι Bendsoe και Kikuchi. Η πιο γενική μορφή της βελτιστοποίησης των κατασκευών είναι η βελτιστοποίηση τοπολογίας [10]. Όπως και με τις βελτιστοποίησεις σχήματος και μεγέθους, σκοπός της είναι η εύρεση της βέλτιστης διανομής των υλικών. Με την βελτιστοποίηση τοπολογίας, το τελικό σχήμα της κατασκευής δεν είναι γνωστό.

Στόχος της παρούσας εργασίας είναι η αφαίρεση τυχαίων στοιχείων της κατασκευής, ώστε να προκύπτει μείωση του βάρους της κατασκευής, λαμβάνοντας υπόψιν κάποιους συγκεκριμένους περιορισμούς που εξασφαλίζουν την ασφάλεια και ομαλή λειτουργία της κατασκευής.



ΣΧΗΜΑ 4.10: *Βελτιστοποίηση τοπολογίας γέφυρας*
(πηγή: https://steelandglassart.com/topology-optimization-truss-structure-diy-plans/)



ΣΧΗΜΑ 4.11: *Βελτιστοποίηση τοπολογίας γέφυρας*
(πηγή: http://topshape.bm-rd.com/topology-optimization-overview/)

4.4.2 ΚΡΙΤΗΡΙΑ ΑΣΤΟΧΙΑΣ

1) Μέγιστη Μετακίνηση Κόμβων

Ομοίως την παράγραφο 4.2.2

2) Σταθερότητα Κατασκευής

Με τη συνεχή αφαίρεση στοιχείων, ελλοχεύει ο κίνδυνος το δικτύωμα να γίνει μηχανισμός. Γι' αυτό θα πρέπει να ληφθεί υπόψιν η σχέση για την εύρεση των βαθμών ελευθερίας του δικτυώματος.
Αν b ο αριθμός των ράβδων του δικτυώματος, r ο αριθμός των αντιδράσεων και n ο αριθμός των κόμβων τότε:

- $b+r = 2n$ → Ισοστατικό(ή ολικά ισοστατικό) Δικτύωμα
- $b+r > 2n$ →Υπερστατικό Δικτύωμα
- $b+r < 2n$ →Υποστατικό Δικτύωμα ή Μηχανισμός

Για να εξασφαλιστεί η κινηματική σταθερότητα της κατασκευής, γίνεται ένας ακόμη έλεγχος, υπολογίζοντας την ορίζουσα του μητρώου δυσκαμψίας. Αν προκύπτει μηδενική (ή δεν υπάρχει) για αφαίρεση συγκεκριμένης ράβδου, τότε κινδυνεύει η κατασκευή, οπότε πρέπει να επιλεγεί γι' αφαίρεση κάποια άλλη ράβδος. Ο υπολογισμός της ορίζουσας [29] γίνεται μετατρέποντας τον τετραγωνικό πίνακα του μητρώου δυσκαμψίας, σε άνω τριγωνικό και στην συνέχεια πολλαπλασιάζοντας τα στοιχεία της διαγωνίου του.

3) Αξονικές Τάσεις των Ράβδων

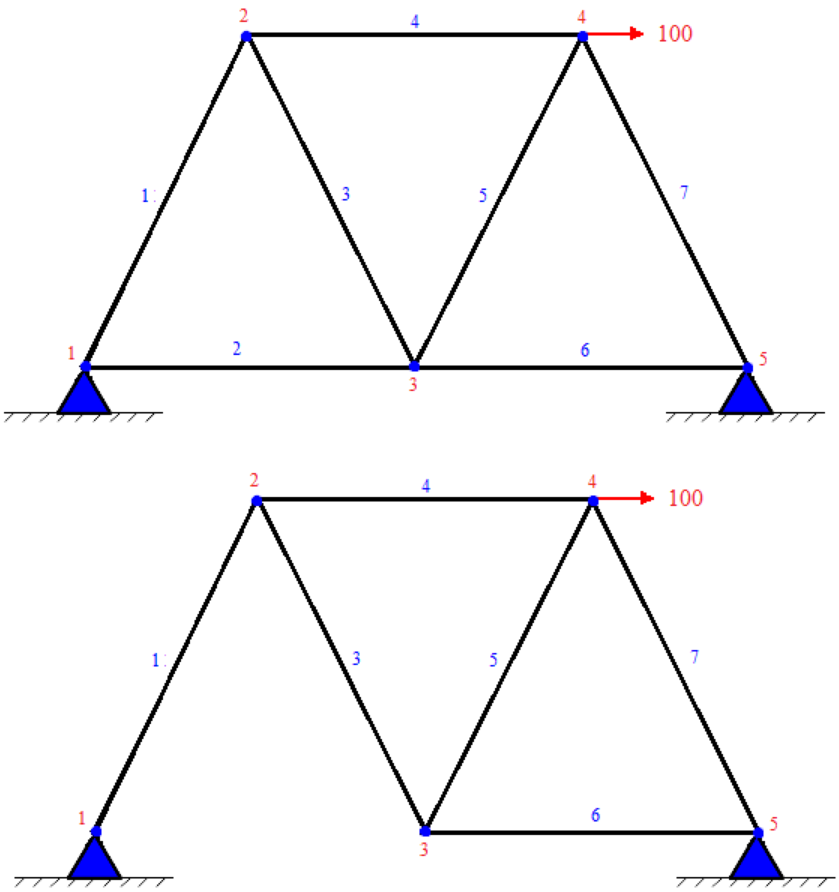
Η κάθε απομείνουσα ράβδος του δικτυώματος πρέπει να αντέχει σε εφελκυσμό ή θλίψη ανάλογα, όπως έχει περιγραφεί στην παράγραφο 4.3.3.

4.4.2 ΔΙΑΔΙΚΑΣΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΤΟΠΟΛΟΓΙΑΣ

4.4.2.1 ΜΕΣΩ ΜΟΝΤΕ ΚΑΡΛΟ

1) Στατική επίλυση δικτύωματος, από την οποία προκύπτει το αρχικό βάρος του δικτυώματος
2) Αφαίρεση τυχαίας ράβδου
3) Νέα στατική Επίλυση δικτυώματος.
4) Έλεγχός για ορίζουσα. Αν προκύπτει μηδενική (ή δεν υπάρχει), τότε επιστροφή στο βήμα 2.
5) Έλεγχος για μέγιστη μετακίνηση κόμβων. Αν υπάρχει σφάλμα τότε επιστροφή στο 2.
6) Έλεγχος για αξονικά φορτία. Αν υπάρχει σφάλμα τότε επιστροφή στο βήμα 2.
7) Έλεγχος αν επιτρέπεται νέα αφαίρεση ράβδων. Αν επιτρέπεται, επιστροφή στο βήμα 2.
8) Αν το βάρος της κατασκευής είναι μικρότερο του αρχικού, τότε κρατάμε το νέο ως αρχικό και επιστρέφουμε στο βήμα 2
9) Τα βήματα 2-8 επαναλαμβάνονται για N=100×(αριθμό κόμβων) φορές.
10) Μόλις τελειώσουν οι επαναλήψεις έχουμε αφαιρέσει τις ράβδους και έχουμε κρατήσει το τελικό βάρος.

Για το δικτύωμα :



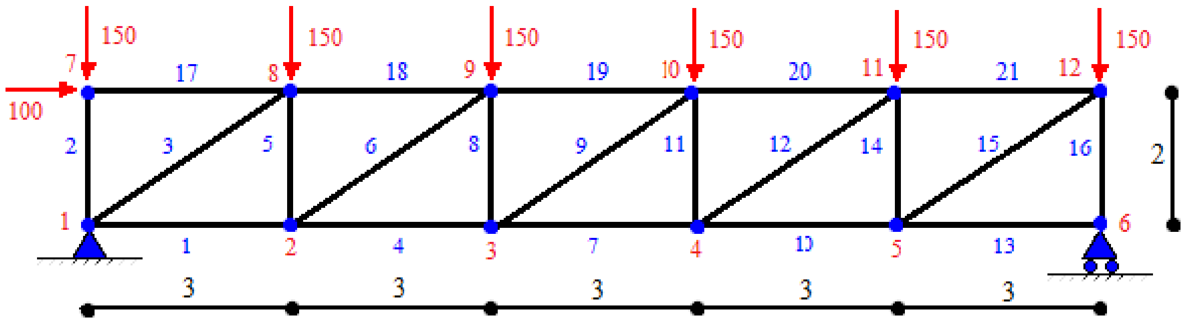
ΣΧΗΜΑ 4.15: *Δικτύωμα 5 κόμβων βελτιστοποιημένο σε τοπολογία*

- Τελικό βάρος δικτυώματος: 2586 kg (έναντι αρχικού βάρους 2932 kg)

ΚΕΦΑΛΑΙΟ 5: ΣΥΝΔΥΑΣΜΟΣ ΜΕΘΟΔΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

5.1 ΔΙΚΤΥΩΜΑ 12 ΚΟΜΒΩΝ

Το δικτύωμα φορτίζεται με το ίδιο βάρος των ράβδων, που θεωρείται οι ασκείται στους κόμβους των ράβδων, και με το ωφέλιμο φορτίο δύο φορτηγών αυτοκινήτων βάρους 60t και 30t αντίστοιχα. Το δικτύωμα επίσης φορτίζεται με ισοδύναμο οριζόντιο σεισμικό φορτίο που λαμβάνεται 100 KN ως ποσοστό της συνισταμένης των κατακόρυφων φορτίων. Το σεισμικό φορτίο και το ωφέλιμο φορτίο μοιράζονται ομοιόμορφα στους κόμβους της άνω σειράς του δικτυώματος.



ΣΧΗΜΑ 5.1: *Δικτύωμα 12 κόμβων*

Μήκος ανοίγματος 3m, ύψος 2m,εμβαδόν διατομής ράβδων 0,008 m^{2}.
Λόγω μορφής δικτυώματος, δεν υπάρχει δυνατότητα αφαίρεσης ράβδου, άρα η βελτιστοποίηση τοπολογίας είναι περιττή**.**

5.1.1 ΣΤΑΤΙΚΗ ΑΝΑΛΥΣΗ

Πίνακας 5.1: Εξωτερικές δυνάμεις των κόμβων δικτύωματος 12 κόμβων

Κόμβος	Κατά X (KN)	Κατά Y (KN)
1	0.0000000	-2.7021432
2	0.0000000	-3.6441431
3	0.0000000	-3.6441431
4	0.0000000	-3.6441431
5	0.0000000	-3.6441431
6	0.0000000	-1.5700001
7	100.00000	-151.57001
8	0.0000000	-153.64415

9	0.0000000	-153.64415
10	0.0000000	-153.64415
11	0.0000000	-153.64415
12	0.0000000	-152.70215

Πίνακας 5.2: Μετακινήσεις των κόμβων δικτύωματος 12 κόμβων

Κόμβος	X(m)	Y(m)
1	0.0000000	0.0000000
2	9.85467224E-04	-9.53088421E-03
3	2.35652667E-03	-1.48210321E-02
4	3.69187188E-03	-1.47136627E-02
5	4.57019778E-03	-9.31592844E-03
6	4.57019778E-03	0.0000000
7	5.01457183E-03	-1.80440256E-04
8	4.83600004E-03	-9.35517251E-03
9	3.85053083E-03	-1.48325674E-02
10	2.47946824E-03	-1.49124451E-02
11	1.14412140E-03	-9.70195606E-03
12	2.65795039E-04	-5.72155463E-04

Πίνακας 5.3: Αντιδράσεις των κόμβων δικτύωματος 12 κόμβων

Κόμβος	Κατά X (KN)	Κατά Y(KN)
1	-99.998283	452.81232
2	2.80346721E-04	-3.6434534
3	2.35382468E-04	-3.6438897
4	4.31153923E-04	-3.6431761
5	-6.69190194E-05	-3.6453536
6	-5.54000362E-05	480.61047
7	100.00014	-151.57001
8	-1.12390518E-03	-153.64485
9	2.27302313E-04	-153.64403
10	-1.19745731E-03	-153.64441
11	-1.10563636E-03	-153.64240

12	-2.05118122E-04	-152.70209

Πίνακας 5.4: Αξονικές των ράβδων δικτύωματος 12 κόμβων

Ράβδος	Από –Προς κόμβο	Αξονική(KN)
1	1-2	551.86163
2	1-7	-151.57001
3	1-8	-543.07245
4	2-3	767.79327
5	2-8	147.59770
6	2-9	-259.51810
7	3-4	747.79333
8	3-9	-9.6897593
9	3-10	24.036743
10	4-5	491.86249
11	4-10	166.97720
12	4-11	307.59024
13	5-6	0.0000000
14	5-11	-324.26309
15	5-12	591.14532
16	6-12	-480.61047
17	7-8	-100.00001
18	8-9	-551.86273
19	9-10	-767.79504
20	10-11	-747.79425
21	11-12	-491.86276

- Το βάρος του δικτυώματος είναι: 3769,7429 kg.

5.1.2 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΗΜΑΤΟΣ

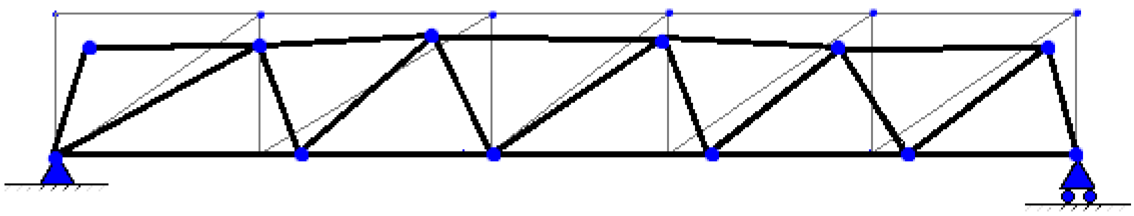
Πίνακας 5.5: Τελικές Συντεταγμένες δικτύωματος 12 κόμβων

ΜΟΝΤΕ ΚΑΡΛΟ

Κόμβος	X (m)	Y(m)
1	0.0000000	0.0000000
2	3.5922933	0.0000000
3	6.3963556	0.0000000
4	9.5796404	0.0000000
5	12.554180	0.0000000
6	15.000000	0.0000000
7	0.47427642	1.5396473
8	2.9307699	1.5746598
9	5.5584575	1.6980579
10	8.9219055	1.6623536
11	11.492367	1.5343535
12	14.579304	1.5280004
Τελικό Βάρος(kg)	3410.5093	

ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ

Κόμβος	X (m)	Y(m)
1	0.0000000	0.0000000
2	3.5260050	0.0000000
3	6.0813289	0.0000000
4	9.5466127	0.0000000
5	12.557337	0.0000000
6	15.000000	0.0000000
7	0.45972705	1.6631296
8	3.0276148	1.5747353
9	5.6684275	1.6889786
10	8.6740618	1.5804803
11	11.415113	1.4974372
12	14.423829	1.6170398
Τελικό Βάρος(kg)	3383.6924	



ΣΧΗΜΑ 5.2: *Δικτύωμα 12 κόμβων βελτιστοποιημένο σε σχήμα*

- ✓ Παρατηρούμε ότι προκύπτει μείωση βάρους 10,2% έναντι του αρχικού λόγω βελτιστοποίησης σχήματος.

5.1.3 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕΓΕΘΟΥΣ

Πίνακας 5.6: Εμβαδόν διατομής Ράβδων δικτύωματος 12 κόμβων

ΜΟΝΤΕ ΚΑΡΛΟ

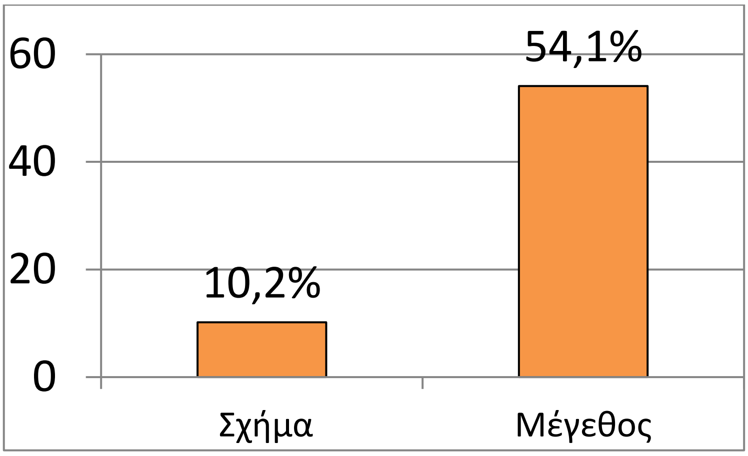
Ράβδος	Από –Προς κόμβο	Εμβαδόν Διατομής (cm^2)
1	1-2	28.20
2	1-7	23.20
3	1-8	42.90
4	2-3	42.90
5	2-8	50.90
6	2-9	35.20
7	3-4	82.10
8	3-9	42.90
9	3-10	20.70
10	4-5	42.90
11	4-10	18.10
12	4-11	18.10
13	5-6	23.20
14	5-11	54.90
15	5-12	35.20
16	6-12	28.20
17	7-8	28.20
18	8-9	42.90
19	9-10	54.90
20	10-11	58.90
21	11-12	72.10
Τελικό Βάρος(kg)	1780.7086	

ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ

Ράβδος	Από –Προς κόμβο	Εμβαδόν Διατομής (cm^2)
1	1-2	28.20
2	1-7	18.10
3	1-8	50.90
4	2-3	72.10
5	2-8	50.90
6	2-9	23.20
7	3-4	35.20
8	3-9	35.20
9	3-10	18.10
10	4-5	72.10
11	4-10	20.70
12	4-11	28.20
13	5-6	23.20
14	5-11	18.10
15	5-12	35.20
16	6-12	72.10
17	7-8	18.10
18	8-9	42.90
19	9-10	50.90
20	10-11	50.90
21	11-12	72.10
Τελικό Βάρος(kg)	1730.8633	

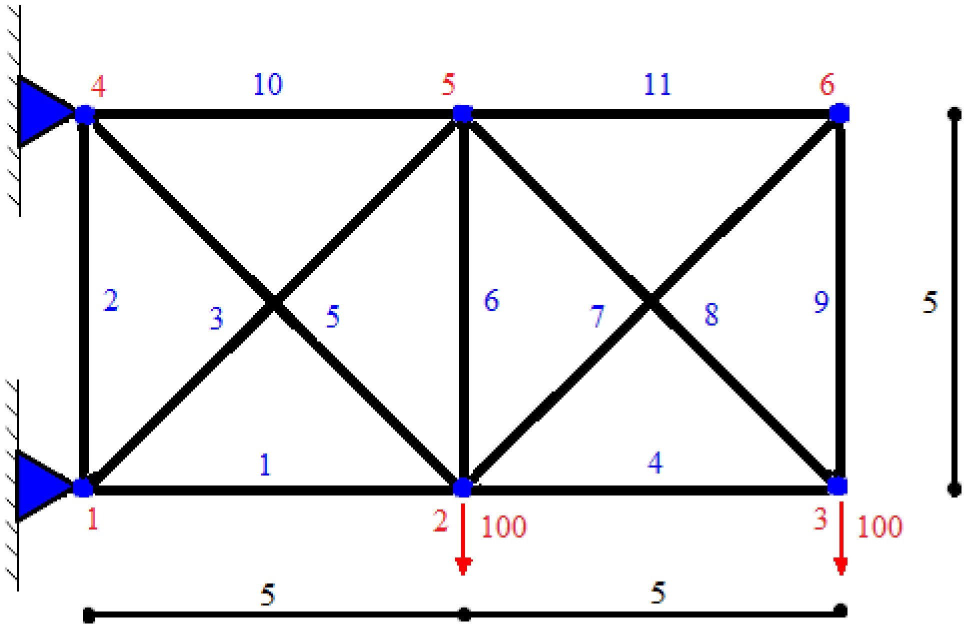
- ✓ Παρατηρούμε ότι προκύπτει μείωση βάρους 54,1% έναντι του αρχικού λόγω βελτιστοποίησης μεγέθους.

Τα ποσοστά βελτιστοποίησης των δύο μεθόδων φαίνονται στο παρακάτω διάγραμμα.



ΣΧΗΜΑ 5.3: Ποσοστά βελτιστοποίησης μεθόδων

5.2 ΔΙΚΤΥΩΜΑ 6 ΚΟΜΒΩΝ



ΣΧΗΜΑ 5.4: *Δικτύωμα 6 κόμβων*

Εμβαδόν διατομής ράβδων: A=0.008 m^2, μέτρο Ελαστικότητας: E=210 GPA

5.2.1 ΣΤΑΤΙΚΗ ΑΝΑΛΥΣΗ

Πίνακας 5.7: Εξωτερικές δυνάμεις των κόμβων (ΙΒ και εξωτερικά φορτία) δικτυώματος 6 κόμβων

Κόμβος	Κατά X (KN)	Κατά Y (KN)
1	0.0000000	-5.3603153
2	0.0000000	-109.15063
3	0.0000000	-105.36031
4	0.0000000	-5.3603158
5	0.0000000	-9.1506310
6	0.0000000	-5.3603158

Πίνακας 5.8: Μετακινήσεις των κόμβων δικτυώματος 6 κόμβων

Κόμβος	X(m)	Y(m)
1	0.0000000	0.0000000
2	-6.84126862E-04	-1.68709049E-03
3	-8.78280785E-04	-3.66543443E-03
4	0.0000000	0.0000000
5	6.56537479E-04	-1.58146652E-03
6	7.91909348E-04	-3.54601606E-03

Πίνακας 5.9: Αντιδράσεις των κόμβων δικτυώματος 6 κόμβων

Κόμβος	Κατά X (KN)	Κατά Y(KN)
1	339.74261	109.87596
2	-2.92953191E-05	-109.15067
3	-6.56564007E-05	-105.36015
4	-339.74261	119.14600
5	7.89382648E-06	-9.1506720
6	2.36666638E-05	-5.3604026

Πίνακας 5.10: Αξονικές των ράβδων δικτυώματος 6 κόμβων

Ράβδος	Από –Προς κόμβο	Αξονική(KN)
1	1-2	-229.86665
2	1-4	0.0000000
3	1-5	- 155.38809
4	2-3	-65.235725
5	2-4	168.49789
6	2-5	35.489639
7	3-5	-64.325432
8	3-6	92.257133
9	3-10	40.124554
10	4-5	220.59662
11	5-6	45.484951

- Το βάρος του δικτυώματος είναι: 3974.2522 kg.

5.2.2 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΗΜΑΤΟΣ

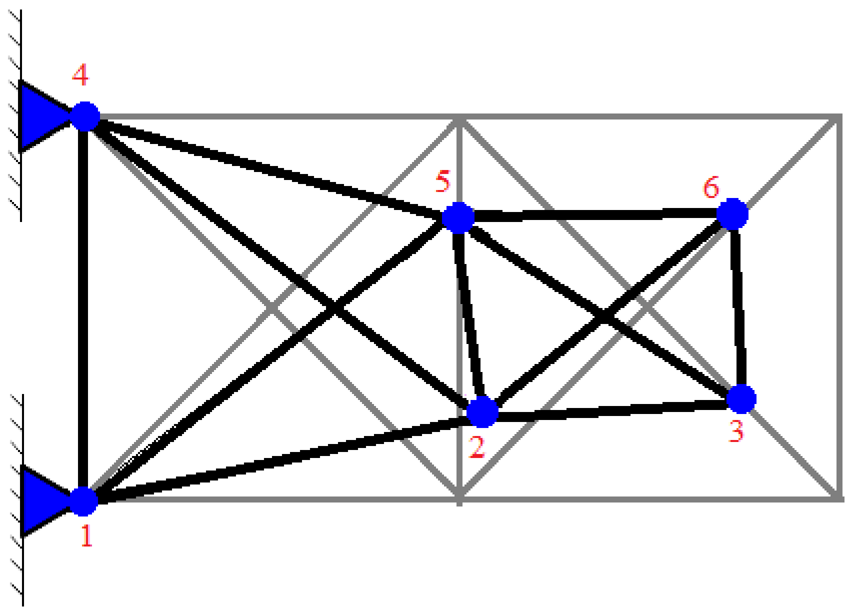
Πίνακας 5.11: Τελικές Συντεταγμένες δικτυώματος 6 κόμβων

ΜΟΝΤΕ ΚΑΡΛΟ

Κόμβος	X (m)	Y(m)
1	0.0000000	0.0000000
2	5.3065686	1.0700945
3	8.7873898	1.2746906
4	0.0000000	5.0000000
5	4.9180217	3.7199383
6	8.6238661	3.7610846
Τελικό Βάρος(kg)	3078.8962	

ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ

Κόμβος	X (m)	Y(m)
1	0.0000000	0.0000000
2	4.4646149	0.67204475
3	8.7418804	1.1849377
4	0.0000000	5.0000000
5	5.0717640	3.7694495
6	8.2124214	4.4800081
Τελικό Βάρος(kg)	3018.9863	



ΣΧΗΜΑ 5.5: *Δικτύωμα 6 κόμβων βελτιστοποιημένο σε σχήμα*

- ✓ Παρατηρούμε ότι προκύπτει μείωση βάρους 24,1% έναντι του αρχικού λόγω βελτιστοποίησης σχήματος.

5.2.3 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕΓΕΘΟΥΣ

Πίνακας 5.12:Εμβαδόν Διατομής Ράβδων δικτυώματος 6 κόμβων

ΜΟΝΤΕ ΚΑΡΛΟ

Ράβδος	Από –Προς κόμβο	Εμβαδόν Διατομής (cm^2)
1	1-2	28.20
2	1-7	23.20
3	1-8	42.90
4	2-3	42.90
5	2-8	50.90
6	2-9	35.20
7	3-4	82.10
8	3-9	42.90
9	3-10	20.70
10	4-5	42.90
11	4-10	18.10
Τελικό Βάρος(kg)	1894.7581	

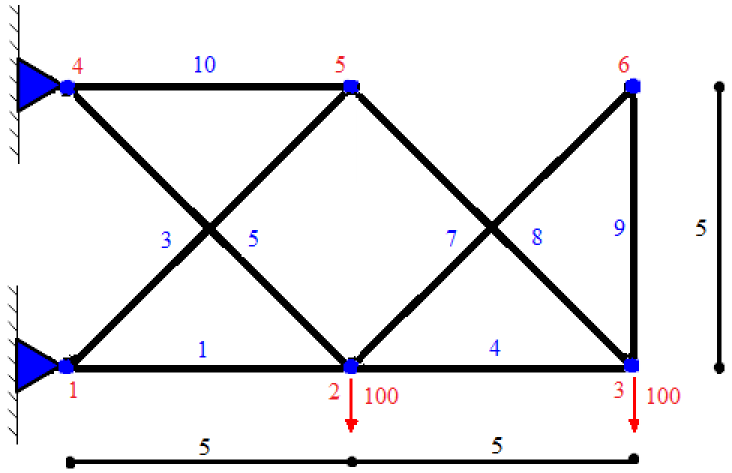
ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ

Ράβδος	Από –Προς κόμβο	Εμβαδόν Διατομής (cm^2)
1	1-2	28.20
2	1-7	18.10
3	1-8	50.90
4	2-3	72.10
5	2-8	50.90
6	2-9	23.20
7	3-4	35.20
8	3-9	35.20
9	3-10	18.10
10	4-5	72.10
11	4-10	20.70
Τελικό Βάρος(kg)	1874.8341	

- ✓ Παρατηρούμε ότι προκύπτει μείωση βάρους 52,8% έναντι του αρχικού λόγω βελτιστοποίησης μεγέθους.

5.2.4 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΟΠΟΛΟΓΙΑΣ

Το σύνολο αντιδράσεων και ράβδων είναι 15, ενώ οι κόμβοι είναι 6. Οπότε υπάρχει δυνατότητα αφαίρεσης 3 ράβδων (=15-6×2). Αφαιρούνται οι ράβδοι 2,6,11.

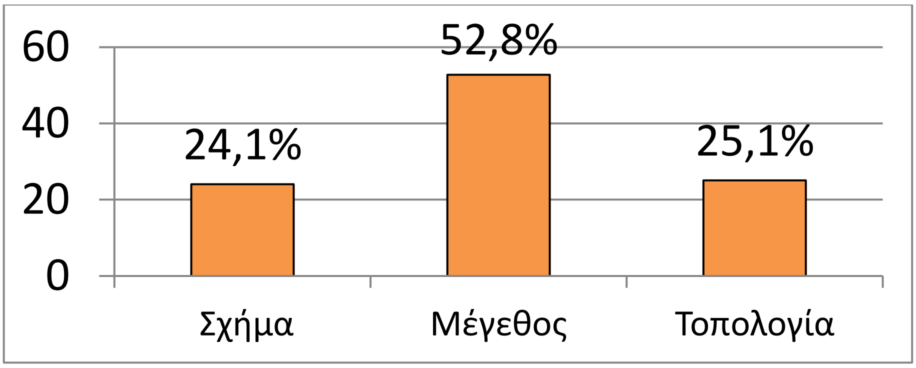


ΣΧΗΜΑ 5.6: *Δικτύωμα 6 κόμβων βελτιστοποιημένο σε τοπολογία*

- ➢ Τελικό βάρος δικτυώματος: 2978 kg (έναντι αρχικού βάρους 3974 kg)

- ✓ Παρατηρούμε ότι προκύπτει μείωση βάρους 25,1% έναντι του αρχικού λόγω βελτιστοποίησης τοπολογίας.

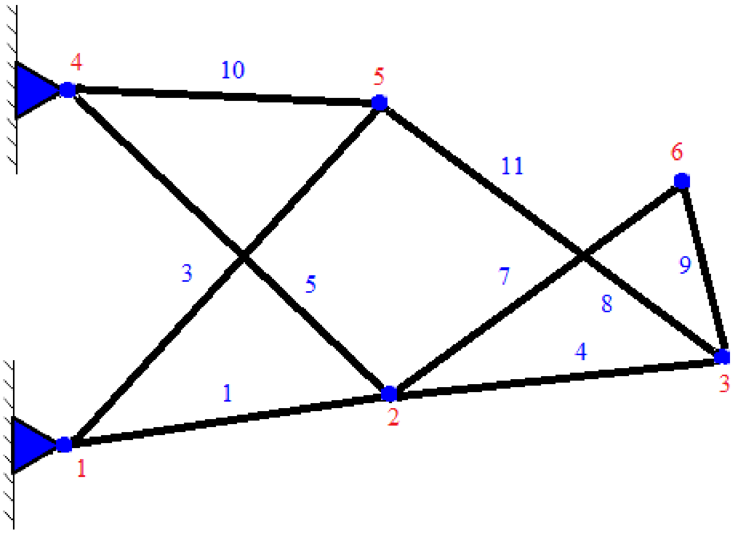
Τα ποσοστά βελτιστοποίησης των τριών μεθόδων φαίνονται στο παρακάτω διάγραμμα.



ΣΧΗΜΑ 5.7: *Ποσοστά βελτιστοποίησης μεθόδων*

5.2.5 ΤΑΥΤΟΧΡΟΝΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΗΜΑΤΟΣ, ΜΕΓΕΘΟΥΣ, ΤΟΠΟΛΟΓΙΑΣ

Αφού πρώτα πραγματοποιήθηκε βελτιστοποίηση τοπολογίας, κατά την οποία αφαιρέθηκαν οι ράβδοι 2,6,11, στην συνέχεια πραγματοποίηθηκε βελτιστοποίηση σχήματος και τέλος βελτιστοποίηση μεγέθους.



ΣΧΗΜΑ 5.8: *Δικτύωμα 6 κόμβων βελτιστοποιημένο ταυτόχρονα σε σχήμα, μέγεθος, τοπολογία.*

Πίνακας 5.13: Τελικές Συντεταγμένες

Κόμβος	X (m)	Y(m)
1	0.0000000	0.0000000
2	4.4131498	0.68919986
3	9.2808104	1.2041485
4	0.0000000	5.0000000
5	4.3989897	4.7681351
6	8.6632757	3.7023556

Πίνακας 5.14: Εμβαδόν Διατομών

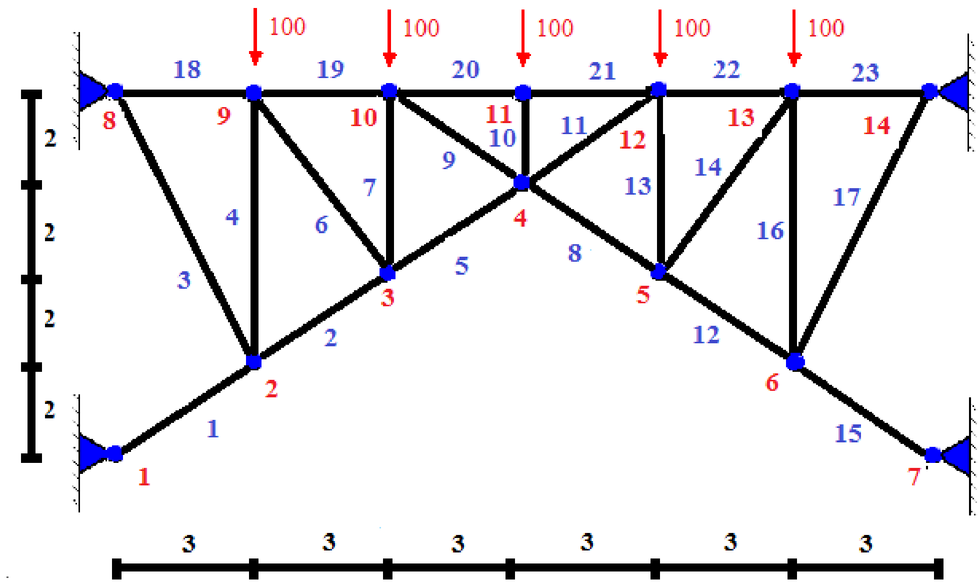
Ράβδος	Από –Προς κόμβο	Εμβαδόν(cm^2)
1	1-2	28.20
2	1-5	23.10
3	2-3	18.10
4	2-4	35.20
5	2-6	23.20
6	3-5	23.20
7	3-6	28.20
8	4-5	35.20

- Τελικό Βάρος δικτυώματος: 840.69 kg.

- ✓ Παρατηρούμε ότι προκύπτει τελική μείωση βάρους 78,8 % έναντι του αρχικού.

5.3 ΔΙΚΤΥΩΜΑ 14 ΚΟΜΒΩΝ

Στο παρακάτω δικτύωμα γίνεται ταυτόχρονη βελτιστοποίηση Τοπολογίας, Σχήματος και Μεγέθους:



ΣΧΗΜΑ 5.9: *Δικτύωμα 14 κόμβων*

Εμβαδόν διατομής ράβδων A=0,005 m^2, μέτρο Ελαστικότητας E=210GPA

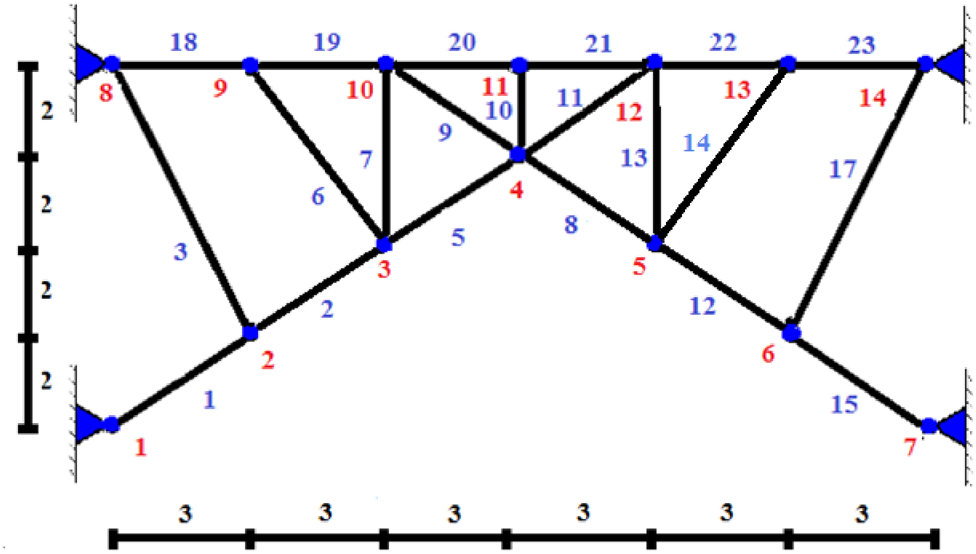
Το αρχικό βάρος κατασκευής χωρίς βελτιστοποίηση είναι: 3621.2368 kg

- Η σειρά με την οποία θα βελτιστοποιηθεί το δικτύωμα είναι:

    1. Τοπολογία
    2. Σχήμα
    3. Μέγεθος

5.3.1 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΟΠΟΛΟΓΙΑΣ

Από την ανάλυση για βελτιστοποίηση τοπολογίας,μετά από αφαίρεση των ράβδων 4, 16, προκύπτει το εξής δικτύωμα:

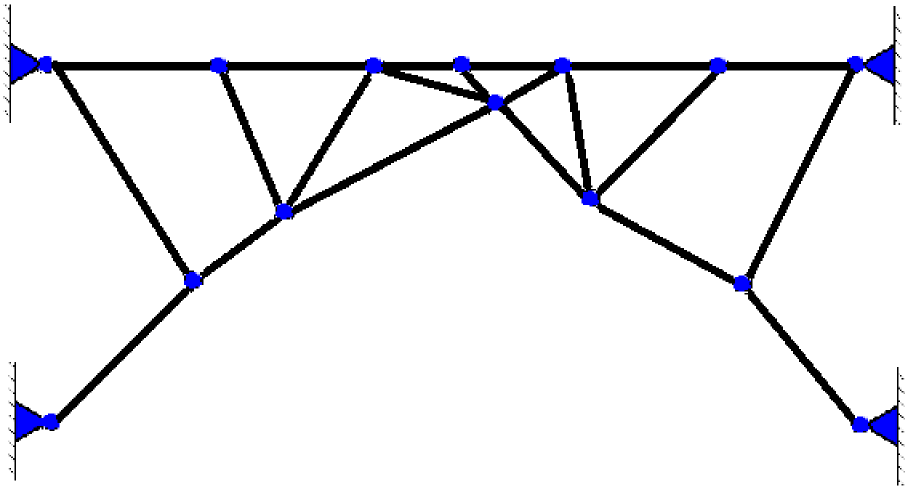


ΣΧΗΜΑ 5.10: *Δικτύωμα 14 κόμβων βελτιστοποιημένο σε τοπολογία*

5.3.2 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΗΜΑΤΟΣ

Πίνακας 5.15: Τελικές Συντεταγμένες δικτυώματος 14 κόμβων

Κόμβος	X(m)	Y(m)
1	0.0000000	0.0000000
2	3.1612418	3.1083453
3	5.0840073	4.5766053
4	9.9782829	7.1572008
5	11.986761	4.9659095
6	15.485084	2.9880230
7	18.000000	0.0000000
8	0.0000000	8.0000000
9	4.1976857	8.0000000
10	7.1696439	8.0000000
11	9.0000000	8.0000000
12	11.472699	8.0000000
13	15.012588	8.0000000
14	18.000000	8.0000000



ΣΧΗΜΑ 5.11: *Δικτύωμα 14 κόμβων βελτιστοποιημένο σε τοπολογία,σχήμα*

Πίνακας 5.16: Αξονικές των ράβδων δικτυώματος 14 κόμβων

Ράβδος	Από –Προς κόμβο	Αξονική(KN)
1	1-2	-450.69424
2	2-3	-450.69510
3	2-8	6.33805030E-05
4	3-4	-540.83246
5	3-9	-125.00013
6	3-10	150.00038
7	4-5	-540.83459
8	4-10	-450.69342
9	1-11	-100.00008
10	4-12	-450.69281
11	5-6	-450.69458
12	5-12	149.99907
13	5-13	-124.99998
14	6-7	-450.69525
15	6-14	-2.70770579E-05
16	8-9	-175.00002
17	9-10	-99.999924

18	10-11	274.99963
19	11-12	274.99966
20	12-13	-99.999680
21	13-14	-174.99971

Πίνακας 5.17: Τελικές μετακινήσεις των κόμβων δικτυώματος 14 κόμβων

Κόμβος	X(m)	Y(m)
1	0.0000000	0.0000000
2	-8.71878467E-04	-4.35939699E-04
3	3.28668160E-03	-8.41754023E-03
4	3.62342689E-09	-5.58002992E-03
5	-3.28666740E-03	-8.41752253E-03
6	8.71880737E-04	-4.35940194E-04
7	0.0000000	0.0000000
8	0.0000000	8.0000000
9	-3.12500022E-04	-1.15819573E-02
10	-4.91071318E-04	-8.06039665E-03
11	-5.34782885E-10	-5.69907762E-03
12	4.91070328E-04	-8.06038175E-03
13	3.12499469E-04	-1.15819275E-02
14	0.0000000	0.0000000

5.3.3 ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕΓΕΘΟΥΣ

Πίνακας 5.18: Εμβαδόν διατομής ράβδων δικτυώματος 14 κόμβων

Ράβδος	Από –Προς κόμβο	Εμβαδόν Διατομής (cm^2)
1	1-2	35.20
2	2-3	35.20
3	2-8	20.70
4	3-4	50.90
5	3-9	35.20
6	3-10	20.70
7	4-5	50.90
8	4-10	35.20
9	1-11	58.90
10	4-12	35.20
11	5-6	50.90
12	5-12	72.10
13	5-13	35.20
14	6-7	35.20
15	6-14	23.20
16	8-9	42.90
17	9-10	18.10
18	10-11	72.10
19	11-12	35.20
20	12-13	20.70
21	13-14	18.10

- Τελικό βάρος δικτυώματος: 2309.42 kg (έναντι αρχικού 3621.24 kg)

- ✓ Προκύπτει δηλαδή τελική μείωση βάρους **36%** .

Σύμφωνα με την εργασία του Max Hultman (2010) [9], στο ίδιο δικτύωμα προέκυψε τελικό βάρος 1245 kg. Παρ’όλα αυτά, δεν είναι σωστό να γίνει άμεση σύγκριση, μια και στην εργασία του, η βελτιστοποίηση τοπολογίας περιελάμβανε και αφαίρεση κόμβων, πέρα από αφαίρεση ράβδων. Συνεπώς το τελικό δικτύωμα αποτελείτο από 11 κόμβους και 14 ράβδους.

ΚΕΦΑΛΑΙΟ 6: ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη παρούσα διπλωματική αναπτύχθηκαν αλγόριθμοι βελτιστοποίησης δικτυώματος μέσω της Μεθόδου Μόντε Κάρλο και Μεθόδου Προσομοιωμένης Ανόπτησης, ως προς το σχήμα, το μέγεθος και την τοπολογία.

Με βάση τα αποτελέσματα των παραδειγμάτων συμπεραίνουμε ότι η Μέθοδος Προσομοιωμένης Ανόπτησης, αν και πιο χρονοβόρα ως προς την εκτέλεση της (για την ακρίβεια, ο χρόνος εκτέλεσης των επαναλήψεων ήταν υπερδιπλάσιος της Μόντε Κάρλο), παρουσιάζει καλύτερα αποτελέσματα σε σχέση με την Μέθοδο Μόντε Κάρλο.

Όσον αφορά τις βελτιστοποίησεις σχήματος, μεγέθους και τοπολογίας, συμπεραίνουμε ότι η βελτιστοποίηση μεγέθους προσφέρει αισθητά μεγαλύτερη οικονομία και καλύτερα αποτελέσματα ειδικά σε περίπτωση που έχει γίνει συντηρητική προδιαστασιολόγηση της κατασκευής. Ο συνδυασμός όμως και των τριών αποφέρει το μεγαλύτερο κέρδος και οικονομία στην κατασκευή.

Ο κώδικας μπορεί εύκολα να ενσωματωθεί σε υφιστάμενα εμπορικά προγράμματα, παίρνοντας ως αρχικό βήμα την παρούσα διπλωματική εργασία.

ΠΑΡΑΡΤΗΜΑ

I. ΚΩΔΙΚΑΣ ΓΙΑ ΣΤΑΤΙΚΗ ΕΠΙΛΥΣΗ

```
PROGRAM DIKTYWMA

REAL:: X(50),Y(50),E(50),A(50,50),AZ(50),WEIGHT(50),&
L(50),P(50,1),SLOPE(50),TMAT(50,2,4),TTMAT(50,4,2),F(50),&
AKTOT1(50,4,2),AK(50,2,2),AKTOT(50,4,4),KTOT(50,50),B(50,50),U(50),col(50,1
),KT2(50,50)
INTEGER:: START(50),FINISH(50)

PRINT*, 'DWSTE ARITHMO KOMBWN'
READ(*,*) NODES
PRINT*,'DWSTE ARITHMO RABDWN'
READ(*,*) ELEMENTS
DO I=1,NODES
PRINT*,'DWSTE TIS SYNTETAGMENES TOY KOMBOY',I
READ(*,*) X(I),Y(I)
17 PRINT*,'O KOMBOS',I,'EINAI KOMBOS STHRIKSHS? 1 GIA NAI ,2 GIA OXI'
READ(*,*) K
IF (K.EQ.1) THEN
20 PRINT*, 'AN EINAI DESMEYMENOS KATA X PATA 1,KATA Y 2,KAI STIS 2 DIEYTH.
PATA 0.'
READ(*,*) J
IF (J.NE.1.AND. J.NE.2.AND. J.NE.0) THEN
PRINT*,'DWSATE MH APODEKTH TIMH'
GO TO 20
ELSE IF (J.EQ.1) THEN
U(2*(I-1)+1)=0.5
ELSE IF (J.EQ.2) THEN
U(2*(I-1)+2)=0.5
ELSE IF (J.EQ.0) THEN
U(2*(I-1)+1)=0.5
U(2*(I-1)+2)=0.5
END IF
ELSE IF (K.NE.1.AND.K.NE.2) THEN
PRINT*, 'DWSATE MH APODEKTH TIMH'
GO TO 17
ELSE
END IF
PRINT*,'ASKEITAI FORTIO STON KOMBO',I,'?;AN NAI PATHSTE 1'
READ(*,*) K
IF (K.EQ.1) THEN
PRINT*, 'FORTIO KATA X'
READ(*,*) P(2*(I-1)+1,1)
PRINT*,'FORTIO KATA Y'
READ(*,*) P(2*(I-1)+2,1)
ELSE
P(2*(I-1)+1,1)=0
P(2*(I-1)+2,1)=0
END IF
END DO
DO I=1,ELEMENTS
PRINT*,'DWSE ARXH KAI TELOS TOY STOIXEIOY',I
READ(*,*)START(I),FINISH(I)
```

```
PRINT*,'DWSE EMBADON DIATOMHS A TOU STOIXEIOY ' ,I
READ(*,*) AZ(I)
E(i)=210000000
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))
IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF

TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
```

```
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1) +
AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2) +
AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1) +
AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2)) +
AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1) +
AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2) +
AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1) +
AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2) +
AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2) +
AKTOT(I,4,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) +
AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1) +
AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2) +
AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)+
AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)+
AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)+
AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)+
AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
```

```
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO



IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO

PRINT*,'OI EKSWTERIKES DYNAMES EINAI'
DO i=1,2*nodes
PRINT*, P(i,1)
END DO

DO I=1,n
DO J1=1,1
COL(I,J1)=0
```

```
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO
print*, 'OI METAKINHSEIS TWN KOMBWN EINAI'
do I=1,2*NODES
print*, COL(I,1)
END DO

DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do
PRINT*,'OI ANTIDRASEIS TWN KOMBWVN EINAI'
DO I=1,2*NODES
print*,P(I,1)
END DO
DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))
PRINT*, 'H AKSONIKH TOY STOXEIOY',I,  START(I),'-',FINISH(I), 'EINAI', F(I)
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO
PRINT*, 'TO BAROS THS KATASKEYHS EINAI',TOTWEIGHT,'KG'

END PROGRAM
```

II. ΚΩΔΙΚΑΣ ΓΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΗΜΑΤΟΣ ΜΕ ΜΟΝΤΕ ΚΑΡΛΟ

```
PROGRAM DIKTYWMA

REAL::X(50),Y(50),E(50),A(50,50),AZ(50),WEIGHT(50),C(50),K,J,N,XORIG(50),YO
RIG(50),&
L(50),P(50,1),SLOPE(50),TMAT(50,2,4),TTMAT(50,4,2),F(50),XFIN(50),YFIN(50),
MINWEIGHT,&
AKTOT1(50,4,2),AK(50,2,2),AKTOT(50,4,4),KTOT(50,50),B(50,50),U(50),col(50,1
),KT2(50,50),p2(50,1)
INTEGER:: START(50),FINISH(50),values(1:8),s
integer, dimension(:), allocatable :: seed


OPEN (unit = 1, file = "Input.txt", status='old', action='read' )
read(1,*) nodes,elements
do i=1,nodes
read(1,*) x(i),y(i)
end do

do i=1,2*nodes
read(1,*) U(i)
end do

do i=1,elements
read(1,*) start(i),finish(i),az(i)
end do

do i=1,2*nodes
read(1,*) P(i,1)
end do


close (UNIT=1)

do i=1,nodes
if (U(2*(i-1) +1).eq.0.5) then
C(I)=5
else if (U(2*(i-1)+2).eq.0.5) then
C(I)=5
end if
end do


DO I=1,ELEMENTS
E(i)=210000000
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))
IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
```

```
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF

TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)

END DO
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO

DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1) +
AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2) +
AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1) +
AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-
1)+2)) + AKTOT(i,1,4)
```

```
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1) +
AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2) +
AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1)  = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+1)  + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+2)  + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+2)  + AKTOT(I,4,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1)  = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+1)  + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1) +
AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2) +
AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)+
AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)+
AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)+
AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)+
AKTOT(I,4,2)
END DO

do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0          !
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J)  = 0.0
END DO
B(I,I)  = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
```

```
END IF
END DO

IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO

DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO


open (unit=2,file="Output.txt",action="write",status="replace")

WRITE(2,*) 'OI EKSWTERIKES DYNAMES EINAI'
DO i=1,2*nodes
p2(i,1)=p(i,1) !gia na epanaferw tis ekswterikes dunameis
WRITE(2,*) P(i,1)
END DO

DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO
WRITE(2,*) 'OI METAKINHSEIS TWN KOMBWN EINAI'
do I=1,2*NODES
WRITE(2,*) COL(I,1)
END DO
```

```
DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do
WRITE(2,*)'OI ANTIDRASEIS TWN KOMBWVN EINAI'
DO I=1,2*NODES
WRITE(2,*)P(I,1)
END DO
DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))
WRITE(2,*) 'H AKSONIKH TOY STOXEIOY',I,  START(I),'-',FINISH(I), 'EINAI',
F(I)
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO
WRITE(2,*) 'TO BAROS THS KATASKEYHS EINAI',TOTWEIGHT,'KG'


DO I=1,NODES
XORIG(I)=X(I)
YORIG(I)=Y(I)
XFIN(I)=X(I)
YFIN(I)=Y(I)
END DO





do i=1,nodes
if (c(i).eq.5) then
x2=x(i)
y2=y(i)
k2=i
end if
end do
do i=1,k2-1
if (c(i).eq.5) then
x1=x(i)
y1=y(i)
k1=i
end if
end do
if (x2-x1.eq.0) then
slopesupport=1.57
else
slopesupport= atan((y2-y1)/(x2-x1))
end if
```

```
COLMAX=COL(1,1)
DO I=2,2*NODES
IF (COLMAX.LT.COL(I,1)) THEN
COLMAX= COL(I,1)
END IF
END DO

MINWEIGHT=TOTWEIGHT
LMIN=L(1)
DO W=2,ELEMENTS
IF (LMIN.LT.L(W)) THEN
LMIN=L(W)
ELSE
END IF
END DO

call date_and_time(values=values)
CALL RANDOM_SEED (size=s)
allocate(seed(1:s))
seed(:) = values(8)
call random_seed(put=seed)
DO O=1,10000*NODES
CALL RANDOM_NUMBER(Q)
Q=Q*(NODES+1)
Q=INT(Q)
IF (X(Q)-X(K1).EQ.0) THEN
SL1=1.57
ELSE
SL1=ATAN((Y(Q)- Y(K1))/(X(Q)-X(K1)))
END IF
IF (C(Q).EQ.5) THEN
XFIN(Q)=XORIG(Q)
YFIN(Q)=YORIG(Q)
ELSE IF (C(Q).NE.5) THEN
CALL RANDOM_NUMBER (Q1)
Q1=-0.2+Q1*0.4
CALL RANDOM_NUMBER (Q2)
Q2=-0.2+Q2*0.4
X(Q)=XORIG(Q)+Q1*LMIN
Y(Q)=YORIG(Q)+Q2*LMIN
IF  (SLOPESUPPORT.EQ.SL1) THEN
Y(Q)=YORIG(Q)
END IF
TOTWEIGHT=0
DO I=1,ELEMENTS
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO


do i=1,2*nodes
p(i,1)=p2(i,1)
end do
```

```
IF (TOTWEIGHT.LT.MINWEIGHT) THEN


DO I=1,ELEMENTS
IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF

TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)           !
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
```

```
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+1) + AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+2) + AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-
1)+1),(2*(FINISH(I)-1)+2)) + AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+1) + AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+2) + AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+2) + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,4,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+1)+ AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+2)+ AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+1)+ AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+2)+ AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
```

```
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO
```

```
DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO


DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do


DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))

END DO



COLMAX2=COL(1,1)
DO I=2,2*NODES
IF (COLMAX2.LT.COL(I,1)) THEN
COLMAX2= COL(I,1)
END IF
END DO


IF (COLMAX.LT.2*COLMAX2) THEN

XFIN(Q)= X(Q)
YFIN(Q)=Y(Q)

MINWEIGHT=TOTWEIGHT
else
x(q)=XORIG(q)
y(q)=yORIG(q)

END IF

END IF


else
x(q)=XORIG(q)
y(q)=yORIG(q)
END IF
END DO
```

```
WRITE(2,*) 'X'
DO j=1,NODES
WRITE(2,*) Xfin(j)
END DO
WRITE(2,*) 'Y'
DO j=1,NODES
WRITE(2,*) Yfin(j)
END DO

WRITE(2,*) 'BAROS MIN', MINWEIGHT

END PROGRAM
```

III. ΚΩΔΙΚΑΣ ΓΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΧΗΜΑΤΟΣ ΜΕ ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ

```
PROGRAM DIKTYWMA

REAL::
X(50),Y(50),E(50),A(50,50),AZ(50),WEIGHT(50),C(50),K,J,N,XORIG(50),YORIG(50
),TEMPERAT,Z1,&
L(50),P(50,1),SLOPE(50),TMAT(50,2,4),TTMAT(50,4,2),F(50),XFIN(50),YFIN(50),
MINWEIGHT,PROBAB,T1,T2,&
AKTOT1(50,4,2),AK(50,2,2),AKTOT(50,4,4),KTOT(50,50),B(50,50),U(50),col(50,1
),KT2(50,50),p2(50,1)
INTEGER:: START(50),FINISH(50),values(1:8),s
integer, dimension(:), allocatable :: seed


OPEN (unit = 1, file = "Input.txt", status='old', action='read' )
read(1,*) nodes,elements
do i=1,nodes
read(1,*) x(i),y(i)
end do

do i=1,2*nodes
read(1,*) U(i)
end do

do i=1,elements
read(1,*) start(i),finish(i),az(i)
end do

do i=1,2*nodes
read(1,*) P(i,1)
end do


close (UNIT=1)

do i=1,nodes
if (U(2*(i-1) +1).eq.0.5) then
C(I)=5
else if (U(2*(i-1)+2).eq.0.5) then
C(I)=5
end if
end do


DO I=1,ELEMENTS
E(i)=210000000
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))
IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
```

```
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF

TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1) +
AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2) +
AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1) +
AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-
1)+2)) + AKTOT(i,1,4)
```

```
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1) +
AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2) +
AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+1) + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+2) + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+2) + AKTOT(I,4,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+1) + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1) +
AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2) +
AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)+
AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)+
AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)+
AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)+
AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO
```

```
DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO

open (unit=2,file="Output.txt",action="write",status="replace")

WRITE(2,*)'OI EKSWTERIKES DYNAMES EINAI'
DO i=1,2*nodes
p2(i,1)=p(i,1)
WRITE(2,*) P(i,1)
END DO

DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
```

```
END DO
END DO
END DO
WRITE(2,*) 'OI METAKINHSEIS TWN KOMBWN EINAI'
do I=1,2*NODES
WRITE(2,*) COL(I,1)
END DO

DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do
WRITE(2,*)'OI ANTIDRASEIS TWN KOMBWVN EINAI'
DO I=1,2*NODES
WRITE(2,*)P(I,1)
END DO
DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))
WRITE(2,*) 'H AKSONIKH TOY STOXEIOY',I,  START(I),'-',FINISH(I), 'EINAI',
F(I)
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO
WRITE(2,*) 'TO BAROS THS KATASKEYHS EINAI',TOTWEIGHT,'KG'


DO I=1,NODES
XORIG(I)=X(I)
YORIG(I)=Y(I)

END DO

do i=1,nodes
if (c(i).eq.5) then
x2=x(i)
y2=y(i)
k2=i
end if
end do
do i=1,k2-1
if (c(i).eq.5) then
x1=x(i)
y1=y(i)
k1=i
end if
end do
if (x2-x1.eq.0) then
slopesupport=1.57
else
slopesupport= atan((y2-y1)/(x2-x1))
end if
```

```
COLMAX=COL(1,1)
DO I=2,2*NODES
IF (COLMAX.LT.COL(I,1)) THEN
COLMAX= COL(I,1)
END IF
END DO

T1=TOTWEIGHT
LMIN=L(1)
DO W=2,ELEMENTS
IF (LMIN.LT.L(W)) THEN
LMIN=L(W)
ELSE
END IF
END DO


TEMPERAT=1000


call date_and_time(values=values)
CALL RANDOM_SEED (size=s)
allocate(seed(1:s))
seed(:) = values(8)
call random_seed(put=seed)
317  DO O=1,1000*NODES
!
CALL RANDOM_NUMBER(Q)
Q=Q*(NODES+1) s
Q=INT(Q)
IF (X(Q)-X(K1).EQ.0) THEN
SL1=1.57
ELSE
SL1=ATAN((Y(Q)- Y(K1))/(X(Q)-X(K1)))
END IF
IF (C(Q).EQ.5) THEN
XFIN(Q)=XORIG(Q)
YFIN(Q)=YORIG(Q)
ELSE IF (C(Q).NE.5) THEN
CALL RANDOM_NUMBER (Q1)
Q1=-0.2+Q1*0.4
CALL RANDOM_NUMBER (Q2)
Q2=-0.2+Q2*0.4
X(Q)=XORIG(Q)+Q1*LMIN
Y(Q)=YORIG(Q)+Q2*LMIN
IF  (SLOPESUPPORT.EQ.SL1) THEN
Y(Q)=YORIG(Q)
END IF
TOTWEIGHT=0
DO I=1,ELEMENTS
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO
T2=TOTWEIGHT
```

```
do i=1,2*nodes
p(i,1)=p2(i,1)
end do


DO I=1,ELEMENTS
IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF

TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
```

```
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+1) + AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+2) + AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-
1)+1),(2*(FINISH(I)-1)+2)) + AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+1) + AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+2) + AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+2) + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,4,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+1)+ AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+2)+ AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+1)+ AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+2)+ AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
```

```
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO
```

```
DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO


DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do


DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))

END DO



COLMAX2=COL(1,1)
DO I=2,2*NODES
IF (COLMAX2.LT.COL(I,1)) THEN
COLMAX2= COL(I,1)
END IF
END DO


IF (COLMAX.LT.2*COLMAX2) THEN

IF (T2.LT.T1) THEN

XFIN(Q)= X(Q)
YFIN(Q)=Y(Q)
T1=T2
ELSE
PROBAB=EXP(-(T2-T1)/TEMPERAT)
CALL RANDOM_NUMBER(Z1)
IF (PROBAB.GT.Z1) THEN
XFIN(Q)= X(Q)
YFIN(Q)=Y(Q)
T1=T2
else
x(q)=XORIG(q)
y(q)=yORIG(q)

END IF


END IF
```

```
else
x(q)=XORIG(q)
y(q)=yORIG(q)
END IF

else
x(q)=XORIG(q)
y(q)=yORIG(q)

END IF
END DO


IF (TEMPERAT.GT.0.5) THEN
TEMPERAT=0.7*TEMPERAT
GO TO 317
END IF

WRITE(2,*) 'X'
DO j=1,NODES
WRITE(2,*) XFIN(j)
END DO
WRITE(2,*) 'Y'
DO j=1,NODES
WRITE(2,*) YFIN(j)
END DO

WRITE(2,*) 'BAROS MIN', T1


END PROGRAM
```

IV. ΚΩΔΙΚΑΣ ΓΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕΓΕΘΟΥΣ ΜΕ ΜΟΝΤΕ ΚΑΡΛΟ

```
PROGRAM DIKTYWMA

REAL::X(50),Y(50),E(50),A(50,50),AZ(50),WEIGHT(50),XM,NBRD(50),AZF(50),ISMF
(50),LSM,FSM,XSM,&
L(50),P(50,1),SLOPE(50),TMAT(50,2,4),TTMAT(50,4,2),F(50),ISM(150,150),RHS(1
50,150),ESM,LS1,NBRDF(50),ISMFINAL(50),&
AKTOT1(50,4,2),AK(50,2,2),AKTOT(50,4,4),KTOT(50,50),B(50,50),U(50),col(50,1
),KT2(50,50),p2(50,1),QINTERMEDIATE(50)
INTEGER:: START(50),FINISH(50),values(1:8) ,k
integer, dimension(:), allocatable :: seed

OPEN (unit = 1, file = "Input.txt", status='old', action='read' )

read(1,*) nodes,elements
do i=1,nodes
read(1,*) x(i),y(i)
end do

do i=1,2*nodes
read(1,*) U(i)
end do

do i=1,elements
read(1,*) start(i),finish(i),az(i)
end do

do i=1,2*nodes
read(1,*) P(i,1)
end do


close (UNIT=1)

DO I=1,ELEMENTS
E(i)=210000000
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))
IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
```

```
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF

TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+1) + AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+2) + AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-
1)+1),(2*(FINISH(I)-1)+2)) + AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+1) + AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+2) + AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1)  = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+2) + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,4,4)
```

```
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+1)+ AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+2)+ AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+1)+ AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+2)+ AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
```

```
A(IROW,K) = DUM
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO


open (unit=2,file="Output.txt",action="write",status="replace")
WRITE(2,*)'OI EKSWTERIKES DYNAMES EINAI'
DO i=1,2*nodes
p2(i,1)=p(i,1)
WRITE(2,*) P(i,1)
END DO



DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO
WRITE(2,*) 'OI METAKINHSEIS TWN KOMBWN EINAI'
do I=1,2*NODES
WRITE(2,*) COL(I,1)
END DO

DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
```

```
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do
WRITE(2,*)'OI ANTIDRASEIS TWN KOMBWVN EINAI'
DO I=1,2*NODES
WRITE(2,*)P(I,1)
END DO
DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))
WRITE(2,*) 'H AKSONIKH TOY STOXEIOY',I,  START(I),'-',FINISH(I), 'EINAI',
F(I)
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO

WRITE(2,*) 'ARXIKO BAROS', TOTWEIGHT



MINWEIGHT=TOTWEIGHT
DO I =1,ELEMENTS
AZF(I)=AZ(I)
END DO

RHS(1,1)= 0.000434
RHS(1,2)= 0.000559
RHS(2,1)= 0.000554
RHS(2,2)= 0.000719
RHS(3,1)= 0.000674
RHS(3,2)= 0.000879
RHS(3,3)= 0.00107
RHS(4,1)= 0.000794
RHS(4,2)= 0.00104
RHS(4,3)= 0.00127
RHS(5,1)=0.00120
RHS(5,2)= 0.00147
RHS(5,3)= 0.00181
RHS(6,1)= 0.00136
RHS(6,2)= 0.00167
RHS(6,3)= 0.00207
RHS(7,1)= 0.00152
RHS(7,2)= 0.00187
RHS(7,3)= 0.00232
RHS(8,1)= 0.00227
RHS(8,2)= 0.00282
RHS(8,3)= 0.00352
RHS(9,1)= 0.00429
RHS(10,1)=  0.00267
RHS(10,2)= 0.00333
RHS(10,3)= 0.00416
RHS(10,4)= 0.00509
RHS(11,1)=  0.00287
RHS(11,2)=  0.00358
RHS(11,3)=  0.00448
```

```
RHS(11,4)=  0.00549
RHS(12,1)= 0.00383
RHS(12,2)= 0.0048
RHS(12,3)= 0.00589
RHS(12,4)= 0.00721
RHS(13,1)= 0.00433
RHS(13,2)=   0.00544
RHS(13,3)=0.00669
RHS(13,4)=  0.00821
RHS(14,1)=  0.00484
RHS(14,2)=   0.00608
RHS(14,3)=  0.00749
RHS(14,4)=  0.00921
RHS(15,1)=  0.00534
RHS(15,2)=   0.00672
RHS(15,3)=   0.00829
RHS(15,4)=    0.0102
RHS(16,1)=  0.0061
RHS(16,2)=  0.00768
RHS(16,3)=  0.00949
RHS(16,4)=  0.0147
RHS(17,1)=   0.008
RHS(17,2)=   0.00899
RHS(17,3)=   0.0122
RHS(17,4)=   0.0153
RHS(18,1)=    0.0928
RHS(18,2)=    0.0115
RHS(18,3)=    0.0142
RHS(18,4)=    0.0179
```

```
ISM(1,1)= 0.015
ISM(1,2)= 0.0145
ISM(2,1)= 0.0191
ISM(2,2)= 0.0186
ISM(3,1)= 0.0232
ISM(3,2)= 0.0227
ISM(3,3)= 0.0223
ISM(4,1)= 0.0273
ISM(4,2)= 0.0268
ISM(4,3)= 0.0264
ISM(5,1)= 0.0309
ISM(5,2)= 0.0305
ISM(5,3)= 0.0299
ISM(6,1)= 0.035
ISM(6,2)= 0.0345
ISM(6,3)= 0.034
ISM(7,1)= 0.0391
ISM(7,2)= 0.0386
ISM(7,3)= 0.038
ISM(8,1)= 0.0468
ISM(8,2)= 0.0462
ISM(8,3)= 0.0455
ISM(9,1)= 0.0446
ISM(10,1)=  0.055
ISM(10,2)= 0.0544
ISM(10,3)= 0.0536
ISM(10,4)= 0.0527
ISM(11,1)= 0.059
```

```
ISM(11,2)=  0.0585
ISM(11,3)=  0.0577
ISM(11,4)=  0.0568
ISM(12,1)= 0.0626
ISM(12,2)= 0.0618
ISM(12,3)= 0.0609
ISM(12,4)= 0.0598
ISM(13,1)=  0.077
ISM(13,2)=   0.07
ISM(13,3)=0.0691
ISM(13,4)=  0.068
ISM(14,1)=  0.0789
ISM(14,2)=   0.0781
ISM(14,3)=  0.0772
ISM(14,4)=  0.0761
ISM(15,1)=  0.0871
ISM(15,2)=   0.0863
ISM(15,3)=   0.0854
ISM(15,4)=    0.0843
ISM(16,1)=    0.0993
ISM(16,2)=    0.0986
ISM(16,3)=    0.0977
ISM(16,4)=    0.0950
ISM(17,1)=   0.103
ISM(17,2)=   0.102
ISM(17,3)=   0.101
ISM(17,4)=   0.0991
ISM(18,1)=   0.119
ISM(18,2)=   0.118
ISM(18,3)=   0.117
ISM(18,4)=   0.115

DO I=1,ELEMENTS
QINTERMEDIATE(I)=AZ(I)
END DO
call date_and_time(values=values)
CALL RANDOM_SEED (size=k)
allocate(seed(1:k))
seed(:) = values(8)
call random_seed(put=seed)

DO O=1,100000


do i=1,ELEMENTS

CALL RANDOM_NUMBER(V)
V=0.2 + 0.8*V
AZ(I)=V*QINTERMEDIATE(I)
END DO
DO I=1,ELEMENTS
JXM=1
IXM=1
DO WHILE (RHS(IXM,JXM).LT.AZ(I))
DO Jxm=2,4
IF (RHS(IXM,JXM).GT.AZ(I)) THEN
GO TO 100
END IF
END DO
JXM=1
```

```
IXM=IXM+1
END DO
100 AZ(I)=RHS(IXM,JXM)
ismf(i)=ism(ixm,jxm)

END DO


110   AZ(I)=RHS(IXM,JXM)
ismf(i)=ism(ixm,jxm)

do i=1,2*nodes
p(i,1)=p2(i,1)
end do


DO I=1,ELEMENTS
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1) +
AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2) +
AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1) +
AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-
1)+2)) + AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1) +
AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2) +
AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1)  = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+1)  + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+2)  + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+2)  + AKTOT(I,4,4)
```

```
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+1) + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1) +
AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2) +
AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)+
AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)+
AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)+
AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)+
AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
```

```
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO


DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO


DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do

DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))

END DO
```

```
FY=235000
ESM= 1


DO I=1,ELEMENTS

IF (F(I).LT.0) THEN
LS1=3.14*SQRT(E(I)/FY)
LSM= L(I)/(ISMf(I)*LS1)
FSM=0.5*(1+0.21*(LSM-0.2)+LSM**2)
XSM=1/( FSM + SQRT(FSM**2 - LSM**2) )
NBRD(I)= XSM*FY*AZ(I)

ELSE IF (F(I).GT.0) THEN
NBRD(I)=AZ(I)*FY

END IF

IF (NBRD(I).LT.abs(F(I))) THEN

GO TO 50


END IF
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO


IF (TOTWEIGHT.LT.MINWEIGHT) THEN
MINWEIGHT=TOTWEIGHT
DO I=1,ELEMENTS
AZF(I)=AZ(I)
ISMFINAL(I)=ISMF(I)
NBRDF(I)=NBRD(I)
END DO
END IF

50 END DO

TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZF(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO

WRITE(2,*) 'TELIKO BAROS', totweight

DO I=1,ELEMENTS
WRITE(2,*) 'EMBADON DIATOMHS',i,AZF(I)

WRITE(2,*) 'NBRD', NBRDF(I)
END DO

END PROGRAM
```

V. ΚΩΔΙΚΑΣ ΓΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΜΕΓΕΘΟΥΣ ΜΕ ΠΡΟΣΟΜΟΙΩΜΕΝΗ ΑΝΟΠΤΗΣΗ

```
PROGRAM DIKTYWMA

REAL::
X(50),Y(50),E(50),A(50,50),AZ(50),WEIGHT(50),U1,XM,Z1,TEMPERAT,intermediate
,nbrdf(100),lsm,ls1,PROBAB,&
L(50),P(50,1),SLOPE(50),TMAT(50,2,4),TTMAT(50,4,2),F(50),RHS(100,100),ISM(1
00,100),ISMF(100),NBRD(100),azf(100),&
AKTOT1(50,4,2),AK(50,2,2),AKTOT(50,4,4),KTOT(50,50),B(50,50),U(50),col(50,1
),KT2(50,50),P2(50,1),QINTERMEDIATE(100),&
T1,T2,DT
INTEGER:: START(50),FINISH(50),values(1:8) ,k,ixm,jxm
integer, dimension(:), allocatable :: seed
OPEN (unit = 1, file = "Input.txt", status='old', action='read' )
read(1,*) nodes,elements
do i=1,nodes
read(1,*) x(i),y(i)
end do

do i=1,2*nodes
read(1,*) U(i)
end do

do i=1,elements
read(1,*) start(i),finish(i),az(i)
end do

do i=1,2*nodes
read(1,*) P(i,1)
end do


close (UNIT=1)

DO I=1,ELEMENTS
E(I)=210000000
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))
IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
```

```
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF

TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1) +
AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2) +
AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1) +
AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-
1)+2)) + AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1) +
AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2) +
AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+1) + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+2) + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+2) + AKTOT(I,4,4)
```

```
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+1) + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1) +
AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2) +
AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)+
AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)+
AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)+
AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)+
AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
```

```
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO

open (unit=2,file="Output.txt",action="write",status="replace")

WRITE(2,*)'OI EKSWTERIKES DYNAMES EINAI'
DO i=1,2*nodes
p2(i,1)=p(i,1)
WRITE(2,*) P(i,1)
END DO



DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO
WRITE(2,*) 'OI METAKINHSEIS TWN KOMBWN EINAI'
do I=1,2*NODES
WRITE(2,*) COL(I,1)
END DO

DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
```

```
END DO
end do
WRITE(2,*)'OI ANTIDRASEIS TWN KOMBWVN EINAI'
DO I=1,2*NODES
WRITE(2,*)P(I,1)
END DO
DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))
WRITE(2,*) 'H AKSONIKH TOY STOXEIOY',I,  START(I),'-',FINISH(I), 'EINAI',
F(I)
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO

WRITE(2,*) 'ARXIKO BAROS', TOTWEIGHT



RHS(1,1)= 0.000434
RHS(1,2)= 0.000559
RHS(2,1)= 0.000554
RHS(2,2)= 0.000719
RHS(3,1)= 0.000674
RHS(3,2)= 0.000879
RHS(3,3)= 0.00107
RHS(4,1)= 0.000794
RHS(4,2)= 0.00104
RHS(4,3)= 0.00127
RHS(5,1)=0.00120
RHS(5,2)= 0.00147
RHS(5,3)= 0.00181
RHS(6,1)= 0.00136
RHS(6,2)= 0.00167
RHS(6,3)= 0.00207
RHS(7,1)= 0.00152
RHS(7,2)= 0.00187
RHS(7,3)= 0.00232
RHS(8,1)= 0.00227
RHS(8,2)= 0.00282
RHS(8,3)= 0.00352
RHS(9,1)= 0.00429
RHS(10,1)=  0.00267
RHS(10,2)= 0.00333
RHS(10,3)= 0.00416
RHS(10,4)= 0.00509
RHS(11,1)=  0.00287
RHS(11,2)=  0.00358
RHS(11,3)=  0.00448
RHS(11,4)=  0.00549
RHS(12,1)= 0.00383
RHS(12,2)= 0.0048
RHS(12,3)= 0.00589
RHS(12,4)= 0.00721
RHS(13,1)= 0.00433
RHS(13,2)=   0.00544
```

```
RHS(13,3)=0.00669
RHS(13,4)=  0.00821
RHS(14,1)=  0.00484
RHS(14,2)=   0.00608
RHS(14,3)=  0.00749
RHS(14,4)=  0.00921
RHS(15,1)=  0.00534
RHS(15,2)=   0.00672
RHS(15,3)=   0.00829
RHS(15,4)=    0.0102
RHS(16,1)=  0.0061
RHS(16,2)=  0.00768
RHS(16,3)=  0.00949
RHS(16,4)=  0.0147
RHS(17,1)=   0.008
RHS(17,2)=   0.00899
RHS(17,3)=   0.0122
RHS(17,4)=   0.0153
RHS(18,1)=    0.0928
RHS(18,2)=    0.0115
RHS(18,3)=    0.0142
RHS(18,4)=    0.0179
```

```
ISM(1,1)= 0.015
ISM(1,2)= 0.0145
ISM(2,1)= 0.0191
ISM(2,2)= 0.0186
ISM(3,1)= 0.0232
ISM(3,2)= 0.0227
ISM(3,3)= 0.0223
ISM(4,1)= 0.0273
ISM(4,2)= 0.0268
ISM(4,3)= 0.0264
ISM(5,1)= 0.0309
ISM(5,2)= 0.0305
ISM(5,3)= 0.0299
ISM(6,1)= 0.035
ISM(6,2)= 0.0345
ISM(6,3)= 0.034
ISM(7,1)= 0.0391
ISM(7,2)= 0.0386
ISM(7,3)= 0.038
ISM(8,1)= 0.0468
ISM(8,2)= 0.0462
ISM(8,3)= 0.0455
ISM(9,1)= 0.0446
ISM(10,1)=  0.055
ISM(10,2)= 0.0544
ISM(10,3)= 0.0536
ISM(10,4)= 0.0527
ISM(11,1)= 0.059
ISM(11,2)=  0.0585
ISM(11,3)=  0.0577
ISM(11,4)=  0.0568
ISM(12,1)= 0.0626
ISM(12,2)= 0.0618
ISM(12,3)= 0.0609
ISM(12,4)= 0.0598
```

```
ISM(13,1)=  0.077
ISM(13,2)=   0.07
ISM(13,3)=0.0691
ISM(13,4)=  0.068
ISM(14,1)=  0.0789
ISM(14,2)=   0.0781
ISM(14,3)=  0.0772
ISM(14,4)=  0.0761
ISM(15,1)=  0.0871
ISM(15,2)=   0.0863
ISM(15,3)=   0.0854
ISM(15,4)=    0.0843
ISM(16,1)=    0.0993
ISM(16,2)=    0.0986
ISM(16,3)=    0.0977
ISM(16,4)=    0.0950
ISM(17,1)=   0.103
ISM(17,2)=   0.102
ISM(17,3)=   0.101
ISM(17,4)=   0.0991
ISM(18,1)=   0.119
ISM(18,2)=   0.118
ISM(18,3)=   0.117
ISM(18,4)=   0.115




DO I=1,ELEMENTS
QINTERMEDIATE(I)=AZ(I)
END DO

TEMPERAT= 1000
NTR=100*ELEMENTS
T1=TOTWEIGHT
call date_and_time(values=values)
CALL RANDOM_SEED (size=k)
allocate(seed(1:k))
seed(:) = values(8)
call random_seed(put=seed)


60 DO O=1,NTR
do i=1,ELEMENTS

CALL RANDOM_NUMBER(V)
V=0.2 + 0.8*V                        )
AZ(I)=V*QINTERMEDIATE(I)
END DO
DO I=1,ELEMENTS
JXM=1
IXM=1
DO WHILE (RHS(IXM,JXM).LT.AZ(I))
DO Jxm=2,4
IF (RHS(IXM,JXM).GT.AZ(I)) THEN
GO TO 100
END IF
END DO
JXM=1
IXM=IXM+1
```

```
END DO
100 AZ(I)=RHS(IXM,JXM)
ismf(i)=ism(ixm,jxm)

END DO

110   AZ(I)=RHS(IXM,JXM)
ismf(i)=ism(ixm,jxm)


do i=1,2*nodes
p(i,1)=p2(i,1)
end do

DO I=1,ELEMENTS
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO
DO J1=1,4
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1) +
AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2) +
AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1) +
AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-
1)+2)) + AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1) +
AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2) +
AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1)  = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+1)  + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-
1)+2)  + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2)  = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+2)  + AKTOT(I,4,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1)  = KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-
1)+1)  + AKTOT(I,4,3)
```

```
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1) +
AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2) +
AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)+
AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)+
AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)+
AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)+
AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)
end do
end do
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
```

```
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO



DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO


DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
END DO
end do

DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))

END DO


TOTWEIGHT=0
```

```
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO


FY=235000
ESM= 1
DO I=1,ELEMENTS

IF (F(I).LT.0) THEN
LS1=3.14*SQRT(E(I)/FY)
LSM= L(I)/(ISMf(I)*LS1)
FSM=0.5*(1+0.21*(LSM-0.2)+LSM**2)
XSM=1/( FSM + SQRT(FSM**2 - LSM**2) )
NBRD(I)= XSM*FY*AZ(I)


ELSE IF (F(I).GT.0) THEN
NBRD(I)=AZ(I)*FY

END IF

IF (NBRD(I).LT.abs(F(I))) THEN

GO TO 50



END IF
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO

T2=TOTWEIGHT
DT=T2-T1

IF (DT.LT.0) THEN
T1=T2
do i=1,elements
nbrdf(i)=nbrd(i)
azf(i)=az(i)
end do
ELSE
PROBAB= EXP(-DT/TEMPERAT)
CALL RANDOM_NUMBER(Z1)
IF (PROBAB.LE. Z1) THEN

GO TO 50
ELSE
T1=T2

do i=1,elements
nbrdf(i)=nbrd(i)
azf(i)=az(i)
end do
END IF
```

```
END IF

50 END DO

IF (TEMPERAT.GT.0.5) THEN
TEMPERAT=0.7*TEMPERAT
GO TO  60
END IF


TOTWEIGHT=0

DO I=1,ELEMENTS
WRITE(2,*)'rhs', I, AZf(I)
WRITE(2,*)'nbrd',nbrdf(i)


WEIGHT(I)= 7850*AZf(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)



END DO
WRITE(2,*) 'teliko baros',totweight


END PROGRAM
```

VI. ΚΩΔΙΚΑΣ ΓΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΟΠΟΛΟΓΙΑΣ ΜΕ ΜΟΝΤΕ ΚΑΡΛΟ

```
PROGRAM DIKTYWMA

REAL::
X(50),Y(50),E(50),A(50,50),AZ(50),WEIGHT(50),C(50),K,J,N,XORIG(50),YORIG(50
),&
L(50),P(50,1),SLOPE(50),TMAT(50,2,4),TTMAT(50,4,2),F(50),XFIN(50),YFIN(50),
MINWEIGHT,&
AKTOT1(50,4,2),AK(50,2,2),AKTOT(50,4,4),KTOT(50,50),B(50,50),U(50),col(50,1
),&
KT2(50,50),p2(50,1),XSM,ISM(150,150),RHS(150,150),ESM,LS1,finstart(50),finf
inish(50),&
m, temp
,matrix(50,50),XM,NBRD(50),AZF(50),ISMF(50),LSM,FSM,origstart(50),origfinis
h(50),maxD
INTEGER:: START(50),FINISH(50),values(1:8),s,i,il
,tempstart(50),tempfinish(50)
integer, dimension(:), allocatable :: seed
LOGICAL :: DetExists = .TRUE.

OPEN (unit = 1, file = "Input.txt", status='old', action='read' )
read(1,*) nodes,elements
do i=1,nodes
read(1,*) x(i),y(i)
end do

do i=1,2*nodes
read(1,*) U(i)
end do

do i=1,elements
read(1,*) start(i),finish(i),az(i)
end do

do i=1,2*nodes
read(1,*) P(i,1)
end do


close (UNIT=1)

do i=1,nodes
if (U(2*(i-1) +1).eq.0.5) then
C(I)=5
else if (U(2*(i-1)+2).eq.0.5) then
C(I)=5
end if
end do
DO i=1,2*nodes
p2(i,1)=p(i,1)

END DO



DO I=1,ELEMENTS
E(i)=210000000
az(i)=0.005
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
```

```
L(I)=ABS(L(I))
end do
minweight=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
minweight= minweight+WEIGHT(I)
END DO

maxD=0.05

REACT=0
DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
REACT=REACT+1
END IF
END DO
DOF=ELEMENTS+REACT-2*NODES

origdof=dof

do i=1,elements
tempstart(i)=start(i)
origstart(i)=start(i)
tempfinish(i)=finish(i)
origfinish(i)=finish(i)
end do
tempelements=elements
origelements=elements

open (unit=2,file="Output.txt",action="write",status="replace")
call date_and_time(values=values)
CALL RANDOM_SEED (size=s)
allocate(seed(1:s))
seed(:) = values(8)
call random_seed(put=seed)
DO O=1,100*NODES
elements=origelements
tempelements=origelements
do i=1,origelements
start(i)=origstart(i)
tempstart(i)=start(i)
finish(i)=origfinish(i)
tempfinish(i)=finish(i)
end do
dof=origdof
40  CALL RANDOM_NUMBER(Q)
q=q*elements+1
q=int(q)

do i=Q,elements
start(i)=start(i+1)
finish(i)=finish(i+1)
end do
elements=elements-1
```

```
do i=1,2*nodes
p(i,1)=p2(i,1)
end do
DO I=1,ELEMENTS
E(i)=210000000
az(i)=0.005
L(I)=((X(FINISH(I))- X(START(I)))**2+(Y(FINISH(I))-Y(START(I)))**2)**(0.5)
L(I)=ABS(L(I))

IF (X(FINISH(I))- X(START(I)).EQ.0) THEN
IF (Y(FINISH(I))- Y(START(I)).GT.0) THEN
SLOPE(I)=1.5707963268
ELSE
SLOPE(I)=4.7123889804
END IF
ELSE
IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))- X(START(I)).GT.0)
THEN
SLOPE(I)=ATAN((ABS(Y(FINISH(I))-Y(START(I))))/(ABS(X(FINISH(I))-
X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).GT.0 .AND. X(FINISH(I))-
X(START(I)).LT.0) THEN
SLOPE(I)=1.5707963268+ ATAN((ABS(X(FINISH(I))-
X(START(I))))/(ABS(Y(FINISH(I))- Y(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).LT.0 .AND. X(FINISH(I))-
X(START(I)).GT.0) THEN
SLOPE(I)=6.2831853072- ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
ELSE IF (Y(FINISH(I))- Y(START(I)).EQ.0) THEN
SLOPE(I)=0
ELSE
SLOPE(I)=3.1415926536+ATAN((ABS(Y(FINISH(I))-
Y(START(I))))/(ABS(X(FINISH(I))- X(START(I)))))
END IF
END IF



TMAT(I,1,1)=COS(SLOPE(I))
TMAT(I,1,2)=SIN(SLOPE(I))
TMAT(I,2,3)=COS(SLOPE(I))
TMAT(I,2,4)=SIN(SLOPE(I))
TTMAT(I,1,1)=COS(SLOPE(I))
TTMAT(I,2,1)=SIN(SLOPE(I))
TTMAT(I,3,2)= COS(SLOPE(I))
TTMAT(I,4,2)=SIN(SLOPE(I))
AK(I,1,1)=E(I)*AZ(I)/L(I)
AK(I,1,2)= -E(I)*AZ(I)/L(I)
AK(I,2,1)= -E(I)*AZ(I)/L(I)
AK(I,2,2)= E(I)*AZ(I)/L(I)
DO J1=1,4
DO J2= 1,2
AKTOT1(I,J1,J2)=0
DO J3=1,2
AKTOT1(I,J1,J2)=AKTOT1(I,J1,J2)+TTMAT(I,J1,J3)*AK(I,J3,J2)
END DO
END DO
END DO

DO J1=1,4
```

```
DO J2= 1,4
AKTOT(I,J1,J2)=0
DO J3=1,2
AKTOT(I,J1,J2)=AKTOT(I,J1,J2)+AKTOT1(I,J1,J3)*TMAT(I,J3,J2)
END DO
END DO
END DO
END DO
DO J1=1,2*NODES
DO J2=1,2*NODES
KTOT(J1,J2)=0
END DO
END DO
DO I=1,ELEMENTS
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+1) + AKTOT(I,1,1)
KTOT(2*(START(I)-1)+1,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+1,2*(START(I)-
1)+2) + AKTOT(I,1,2)
KTOT(2*(START(I)-1)+1,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,1,3)
KTOT((2*(START(I)-1)+1),(2*(FINISH(I)-1)+2))=KTOT((2*(START(I)-
1)+1),(2*(FINISH(I)-1)+2)) + AKTOT(i,1,4)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+1)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+1) + AKTOT(I,2,1)
KTOT(2*(START(I)-1)+2,2*(START(I)-1)+2)= KTOT(2*(START(I)-1)+2,2*(START(I)-
1)+2) + AKTOT(I,2,2)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+1) + AKTOT(I,3,3)
KTOT(2*(FINISH(I)-1)+1,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-
1)+1,2*(FINISH(I)-1)+2) + AKTOT(I,3,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+2) = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,4,4)
KTOT(2*(FINISH(I)-1)+2,2*(FINISH(I)-1)+1) = KTOT(2*(FINISH(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,4,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+1)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+1) + AKTOT(I,2,3)
KTOT(2*(START(I)-1)+2,2*(FINISH(I)-1)+2)=KTOT(2*(START(I)-
1)+2,2*(FINISH(I)-1)+2) + AKTOT(I,2,4)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+1)+ AKTOT(I,3,1)
KTOT(2*(FINISH(I)-1)+1,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+1,2*(START(I)-1)+2)+ AKTOT(I,3,2)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+1)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+1)+ AKTOT(I,4,1)
KTOT(2*(FINISH(I)-1)+2,2*(START(I)-1)+2)=KTOT(2*(FINISH(I)-
1)+2,2*(START(I)-1)+2)+ AKTOT(I,4,2)
END DO
do i=1,2*nodes
do j= 1,2*nodes
KT2(i,j)=KTOT(i,j)


end do
end do
```

```
il = 1
n=2*nodes
do i=1,2*nodes
do j=1,2*nodes
Matrix(i,j)=Ktot(i,j)

end do
end do

DO k = 1, n-1

IF (matrix(k,k) == 0) THEN
DetExists = .FALSE.
elements=origelements
do s=1,elements
start(s)=origstart(s)
finish(s)=origfinish(s)
end do
GO TO 40

DO i = k+1, n
IF (matrix(i,k) /= 0) THEN
DO j = 1, n
temp = matrix(i,j)
matrix(i,j)= matrix(k,j)
matrix(k,j) = temp
END DO
DetExists = .TRUE.
il=-il
EXIT
ENDIF
END DO
IF (DetExists .EQV. .FALSE.) THEN
FindDet = 0
return
END IF
ENDIF
DO j = k+1, n
m = matrix(j,k)/matrix(k,k)
DO i = k+1, n
matrix(j,i) = matrix(j,i) - m*matrix(k,i)
END DO
END DO
END DO


FindDet = il
DO i = 1, n
FindDet = FindDet * matrix(i,i)
END DO




if(ISNAN(finddet)) then
```

```
elements=tempelements
do i=1,elements
start(i)=tempstart(i)
finish(i)=tempfinish(i)
end do
go to 40
end if

if (finddet.eq.0) then
elements=tempelements
do i=1,elements
start(i)=tempstart(i)
finish(i)=tempfinish(i)
end do
go to 40
end if



DO I=1,2*NODES
IF (U(I).EQ.0.5) THEN
DO J=1,2*NODES
KTOT(I,J)=0
KTOT(J,I)=0
KTOT(I,I)=1
END DO
ELSE
END IF
END DO

DO I=1,2*NODES
DO J=1,2*NODES
B(I,J)=0
A(I,J)=KTOT(I,J)
END DO
END DO

N=2*NODES
DO I = 1,N
DO J = 1,N
B(I,J) = 0.0
END DO
B(I,I) = 1.0
END DO

DO I = 1,N
BIG = A(I,I)
DO J = I,N
IF (A(J,I).GT.BIG) THEN
BIG = A(J,I)
IROW = J
END IF
END DO


IF (BIG.GT.A(I,I)) THEN
DO K = 1,N
DUM = A(I,K)
A(I,K) = A(IROW,K)
A(IROW,K) = DUM
```

```
DUM = B(I,K)
B(I,K) = B(IROW,K)
B(IROW,K) = DUM
END DO
END IF

DUM = A(I,I)
DO J = 1,N
A(I,J) = A(I,J)/DUM
B(I,J) = B(I,J)/DUM
END DO

DO J = I+1,N
DUM = A(J,I)
DO K = 1,N
A(J,K) = A(J,K) - DUM*A(I,K)
B(J,K) = B(J,K) - DUM*B(I,K)
END DO
END DO
END DO


DO I = 1,N-1
DO J = I+1,N
DUM = A(I,J)
DO W = 1,N
A(I,W) = A(I,W)-DUM*A(J,W)
B(I,W) = B(I,W)-DUM*B(J,W)
END DO
END DO
END DO




DO i=1,2*nodes
p2(i,1)=p(i,1)

END DO


DO I=1,n
DO J1=1,1
COL(I,J1)=0
DO J2=1,n
COL(I,J1)=COL(I,J1)+B(I,J2)*P(J2,J1)
END DO
END DO
END DO

do I=1,2*NODES

END DO

DO I=1,n
DO J1=1,1
P(I,J1)=0
do J2=1,n
P(I,J1)=P(I,J1)+KT2(I,J2)*COL(J2,J1)
END DO
```

```
END DO
end do

DO I=1,2*NODES

END DO
DO i=1,ELEMENTS
F(I)= E(I)*AZ(I)/L(I)*(COS(SLOPE(I))*(col(2*(FINISH(I)-1)+1,1)-
col(2*(START(I)-1)+1,1))&
+SIN(SLOPE(I))*( col(2*(FINISH(I)-1)+2,1)-col(2*(START(I)-1)+2,1)))

END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO


RHS(1,1)= 0.000434
RHS(1,2)= 0.000559
RHS(2,1)= 0.000554
RHS(2,2)= 0.000719
RHS(3,1)= 0.000674
RHS(3,2)= 0.000879
RHS(3,3)= 0.00107
RHS(4,1)= 0.000794
RHS(4,2)= 0.00104
RHS(4,3)= 0.00127
RHS(5,1)=0.00120
RHS(5,2)= 0.00147
RHS(5,3)= 0.00181
RHS(6,1)= 0.00136
RHS(6,2)= 0.00167
RHS(6,3)= 0.00207
RHS(7,1)= 0.00152
RHS(7,2)= 0.00187
RHS(7,3)= 0.00232
RHS(8,1)= 0.00227
RHS(8,2)= 0.00282
RHS(8,3)= 0.00352
RHS(9,1)= 0.00429
RHS(10,1)=  0.00267
RHS(10,2)= 0.00333
RHS(10,3)= 0.00416
RHS(10,4)= 0.00509
RHS(11,1)=  0.00287
RHS(11,2)=  0.00358
RHS(11,3)=  0.00448
RHS(11,4)=  0.00549
RHS(12,1)= 0.00383
RHS(12,2)= 0.0048
RHS(12,3)= 0.00589
RHS(12,4)= 0.00721
RHS(13,1)= 0.00433
RHS(13,2)=   0.00544
RHS(13,3)=0.00669
RHS(13,4)=  0.00821
RHS(14,1)=  0.00484
RHS(14,2)=   0.00608
```

```
RHS(14,3)=  0.00749
RHS(14,4)=  0.00921
RHS(15,1)=  0.00534
RHS(15,2)=   0.00672
RHS(15,3)=   0.00829
RHS(15,4)=    0.0102
RHS(16,1)=  0.0061
RHS(16,2)=  0.00768
RHS(16,3)=  0.00949
RHS(16,4)=  0.0147
RHS(17,1)=   0.008
RHS(17,2)=   0.00899
RHS(17,3)=   0.0122
RHS(17,4)=   0.0153
RHS(18,1)=    0.0928
RHS(18,2)=    0.0115
RHS(18,3)=    0.0142
RHS(18,4)=    0.0179
```

```
ISM(1,1)= 0.015
ISM(1,2)= 0.0145
ISM(2,1)= 0.0191
ISM(2,2)= 0.0186
ISM(3,1)= 0.0232
ISM(3,2)= 0.0227
ISM(3,3)= 0.0223
ISM(4,1)= 0.0273
ISM(4,2)= 0.0268
ISM(4,3)= 0.0264
ISM(5,1)= 0.0309
ISM(5,2)= 0.0305
ISM(5,3)= 0.0299
ISM(6,1)= 0.035
ISM(6,2)= 0.0345
ISM(6,3)= 0.034
ISM(7,1)= 0.0391
ISM(7,2)= 0.0386
ISM(7,3)= 0.038
ISM(8,1)= 0.0468
ISM(8,2)= 0.0462
ISM(8,3)= 0.0455
ISM(9,1)= 0.0446
ISM(10,1)=  0.055
ISM(10,2)= 0.0544
ISM(10,3)= 0.0536
ISM(10,4)= 0.0527
ISM(11,1)= 0.059
ISM(11,2)=  0.0585
ISM(11,3)=  0.0577
ISM(11,4)=  0.0568
ISM(12,1)= 0.0626
ISM(12,2)= 0.0618
ISM(12,3)= 0.0609
ISM(12,4)= 0.0598
ISM(13,1)=  0.077
ISM(13,2)=   0.07
ISM(13,3)=0.0691
ISM(13,4)=  0.068
```

```
ISM(14,1)=  0.0789
ISM(14,2)=   0.0781
ISM(14,3)=  0.0772
ISM(14,4)=  0.0761
ISM(15,1)=  0.0871
ISM(15,2)=   0.0863
ISM(15,3)=   0.0854
ISM(15,4)=    0.0843
ISM(16,1)=    0.0993
ISM(16,2)=    0.0986
ISM(16,3)=    0.0977
ISM(16,4)=    0.0950
ISM(17,1)=   0.103
ISM(17,2)=   0.102
ISM(17,3)=   0.101
ISM(17,4)=   0.0991
ISM(18,1)=   0.119
ISM(18,2)=   0.118
ISM(18,3)=   0.117
ISM(18,4)=   0.115


do i=1,2*nodes
if  ( abs(COL(I,1)).gt.maxD)  then

elements=tempelements
do k=1,elements
start(k)=tempstart(k)
finish(k)=tempfinish(k)
end do
go to 40
end if
end do



DO I=1,ELEMENTS
JXM=1
IXM=1
DO WHILE (RHS(IXM,JXM).LT.AZ(I))
DO Jxm=2,4
IF (RHS(IXM,JXM).GT.AZ(I)) THEN
GO TO 100
END IF
END DO
JXM=1
IXM=IXM+1
END DO
100 AZ(I)=RHS(IXM,JXM)
ismf(i)=ism(ixm,jxm)

END DO

FY=235000
ESM= 1

DO I=1,ELEMENTS

IF (F(I).LT.0) THEN
LS1=3.14*SQRT(E(I)/FY)
```

```
LSM= L(I)/(ISMf(I)*LS1)
FSM=0.5*(1+0.21*(LSM-0.2)+LSM**2)
XSM=1/( FSM + SQRT(FSM**2 - LSM**2) )
NBRD(I)= XSM*FY*AZ(I)

ELSE IF (F(I).GT.0) THEN
NBRD(I)=AZ(I)*FY

END IF

IF (NBRD(I).LT.abs(F(I))) THEN
elements=tempelements
do k=1,elements
start(k)=tempstart(k)
finish(k)=tempfinish(k)
end do
GO TO 40


END IF
END DO
TOTWEIGHT=0
DO I=1,ELEMENTS
WEIGHT(I)= 7850*AZ(I)*L(I)
TOTWEIGHT= TOTWEIGHT+WEIGHT(I)
END DO

DOF=DOF-1

tempelements=elements
do i=1,elements
tempstart(i)=start(i)
tempfinish(i)=finish(i)
end do
IF (DOF.EQ.0) THEN
if (totweight.lt.minweight) then
finelements=elements
minweight=totweight
do i=1,finelements
finstart(i)= start(i)
finfinish(i)=finish(i)
end do
end if
else
go to 40
END IF

end do

60  do i =1,finelements
write(*,*) finstart(i),finfinish(i)
end do
print*, 'baros',minweight
pause

end program
```

VII. ΑΡΧΕΙΑ INPUT ΤΩΝ ΕΠΙΛΥΜΕΝΩΝ ΔΙΚΤΥΩΜΑΤΩΝ

- **Δικτύωμα 5 κόμβων:**

```
5, 7
0,0
5,10
10,0
15,10
20,0
0.5
0.5
0
0
0
0
0
0
0.5
0.5
1,2,0.005
1,3,0.005
2,3,0.005
2,4,0.005
3,4,0.005
3,5,0.005
4,5,0.005
0
0
0
0
0
0
100
0
0
0
```

- **Δικτύωμα 6 κόμβων:**

6,11
0,0
5,0
10,0
0,5
5,5
10,5
0.5
0.5
0
0
0
0
0.5
0.5
0
0
0
0
1,2,0.008
1,4,0.008
1,5,0.008
2,3,0.008
2,4,0.008
2,5,0.008
2,6,0.008
3,5,0.008
3,6,0.008
4,5,0.008
5,6,0.008
0
0
0
-100
0
-100
0
0
0
0
0
0

- **Δικτύωμα 12 κόμβων:**

12, 21
0,0
3,0
6,0
9,0
12,0
15,0
0,2
3,2
6,2
9,2
12,2
15,2
0.5
0.5
0
0
0
0
0
0
0
0
0.5
0.5
0
0
0
0
0
0
0
0
0
0
0
0
1,2,0.008
1,7,0.008
1,8,0.008
2,3,0.008
2,8,0.008
2,9,0.008
3,4,0.008
3,9,0.008
3,10,0.008
4,5,0.008

4,10,0.008
4,11,0.008
5,6,0.008
5,11,0.008
5,12,0.008
6,12,0.008
7,8,0.008
8,9,0.008
9,10,0.008
10,11,0.008
11,12,0.008
0
0
0
0
0
0
0
0
0
0
0
0
100
-150
0
-150
0
-150
0
-150
0
-150
0
-150

- **Δικτύωμα 14 κόμβων:**

14, 23
0,0
3,2
6,4
9,6
12,4
15,2
18,0
0,8
3,8
6,8
9,8
12,8
15,8
18,8
0.5
0.5
0
0
0
0
0
0
0
0
0
0
0.5
0.5
0.5
0.5
0
0
0
0
0
0
0
0
0
0
0.5
0.5
1,2,0.005
2,3,0.005
2,8,0.005
2,9,0.005

3,4,0.005
3,9,0.005
3,10,0.005
4,5,0.005
4,10,0.005
4,11,0.005
4,12,0.005
5,6,0.005
5,12,0.005
5,13,0.005
6,7,0.005
6,13,0.005
6,14,0.005
8,9,0.005
9,10,0.005
10,11,0.005
11,12,0.005
12,13,0.005
13,14,0.005
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
-100
0
-100
0
-100
0
-100
0
-100
0
0

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΑΝΑΦΟΡΕΣ ΑΠΟ ΑΡΘΡΑ-ΒΙΒΛΙΑ

[1] Παπαδρακάκης Μανόλης - Ανάλυση Φορέων με την Μέθοδο των Πεπερασμένων Στοιχείων, 2001

[2] Γέωργιος Τσαμασφύρος - Μέθοδος των Πεπερασμένων Στοιχείων, 1990

[3] Ματαράς Δημήτρης Σ. - Προγραμματισμός Fortran 90/95 για επιστήμονες και μηχανικούς , 2001

[4] S. S. Rao - The Finite Element Method in Engineering , 2005

[5] Α.Α. Στάμος και Ι. Τζουβαδάκης 2009 - Χάραξη πολεοδομικού Ιστού για βελτιστοποίηση βιοκλιματικου σχεδιασμού κτιρίων

[6] A. A. Stamos 2008 - Optimization of Highway Gradeline using the Simulated Annealing Method

[7] William H. Press 1986 - Numerical Recipes in Fortran 77.The Art of Scientific Computing

[8] Καλαμαράς Δημήτρης 2001 - Αριθμητική Ανάλυση

[9] Max Hultman 2010 - Weight optimization of steel trusses by a genetic algorithm

[10] Anton Olason-Daniel Tilman 2010 - Methodology for Topology and Shape Optimization in the Design Process

[11] Thomas Weise 2009-Global Optimization Algorithms–Theory and Application

[12] Garret N. Vanderplaats 2006 - Structural Optimization for Statics, Dynamics and Beyond

ΑΝΑΦΟΡΕΣ ΑΠΟ ΔΙΑΔΙΚΤΥΟ

[13] http://users.ntua.gr/caridis/methodoi/keimena/chap%2003/Chapter%2003.pdf

[14] http://www.mm.bme.hu/~szeki/files/2012_vem-en.pdf

[15] https://en.wikipedia.org/wiki/Direct_stiffness_method#Member_stiffness_relations

[16]
http://www.colorado.edu/engineering/cas/courses.d/IFEM.d/IFEM.AppO.d/IFEM.AppO.pdf

[17] https://el.wikipedia.org/wiki/Monte_Carlo_(%CE%BC%CE%AD%CE%B8%CE%BF%CE%B4%CE%BF%CF%82)

[18] https://www.unige.ch/sciences/astro/files/2713/8971/4086/3_Paltani_MonteCarlo.pdf

[19] https://en.wikipedia.org/wiki/Simulated_annealing

[20] http://codecapsule.com/2010/04/06/simulated-annealing-traveling-salesman/

[21] http://topshape.bm-rd.com/topology-optimization-overview/

[22] https://steelandglassart.com/topology-optimization-truss-structure-diy-plans/

[23] https://neos-guide.org/content/optimization-taxonomy

[24] http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1934938

[25] https://pbcm.files.wordpress.com/2011/06/060911_1919_implementin33.png?w=630

[26] https://www.researchgate.net/figure/275075222_fig4_Flow-chart-of-the-Ant-Colony-Optimization

[27] https://ocw.mit.edu/courses/engineering-systems-division/esd-77-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec10.pdf

[28] http://www.colincaprani.com/files/notes/SAIV/4%20-%20Matrix%20Stiffness%20Method.pdf

[29] http://web.hku.hk/~gdli/UsefulFiles/Example-Fortran-program.html

[30] http://users.ntua.gr/cvsapoun/2-Plane%20Trusses.pdf

[31] http://slideplayer.gr/slide/2480681/

[32] http://web.mit.edu/16.810/www/16.810_L8_Optimization.pdf