# Εργασία 2:

Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Μέλη Ομάδας 38:

* **ΦΟΥΤΡΗΣ ΒΑΣΙΛΕΙΟΣ 1115202000231**
* **ΠΑΠΑΔΗΜΗΤΡΙΟΥ ΜΑΡΙΝΑ 1115202100136**

**Περιγραφή Εργασίας**

Αυτό η εργασία υλοποιεί αλγορίθμους καθολικής βελτιστοποίησης για την εισαγωγή σημείων Steiner σε τριγωνοποιήσεις. Οι αλγόριθμοι που χρησιμοποιούνται περιλαμβάνουν:

1. Τοπική Αναζήτηση
2. Προσομοιωμένη Ανόπτηση
3. Βελτιστοποίηση μέσω Αποικίας Μυρμηγκιών`

**Λειτουργίες**

**1. Τοπική Αναζήτηση**

Η μέθοδος τοπικής αναζήτησης επιδιώκει την επαναληπτική βελτίωση της τριγωνοποίησης μέσω της εισαγωγής σημείων Steiner. Η διαδικασία περιλαμβάνει:

Δοκιμή και επιλογή της καταλληλότερης θέσης μεταξύ 5 εναλλακτικών. Αξιολόγηση της αποτελεσματικότητας της θέσης ως προς τη μείωση των αμβλυγωνίων τριγώνων. Ο χρήστης μπορεί να καθορίσει τον μέγιστο αριθμό επαναλήψεων (L) για τη συνθήκη τερματισμού.

**2. Προσομοιωμένη Ανόπτηση**

Στην προσομοιωμένη ανόπτηση:

Υπολογίζεται η «ενέργεια» της αρχικής τριγωνοποίησης. Επιλέγεται η κατάλληλη θέση εισαγωγής σημείου Steiner με βάση τη μείωση της ενέργειας. Γίνονται αποδεκτές τοπικά χειρότερες θέσεις βάσει του κριτηρίου Metropolis για την αποφυγή εγκλωβισμού σε τοπικό ελάχιστο. Οι παράμετροι περιλαμβάνουν τα βάρη α και β για τον προσδιορισμό της «ενέργειας» και τον αριθμό επαναλήψεων (L).

**3. Βελτιστοποίηση μέσω Αποικίας Μυρμηγιών**

Στη βελτιστοποίηση μέσω αποικίας μυρμηγκιών:

Ένα σύνολο τεχνητών μυρμηγκιών (πρακτόρων) επιχειρεί την επαναληπτική βελτίωση της τριγωνοποίησης. Κάθε μυρμήγκι επιλέγει την επόμενη θέση βάσει πιθανότητας που προσδιορίζεται από: Τον ευρετικό συντελεστή της κάθε εναλλακτικής. Το ίχνος της φερομόνης που αντιπροσωπεύει τη μαθημένη προτίμηση. Οι παράμετροι περιλαμβάνουν τα βάρη για τον προσδιορισμό της ενέργειας, τους συντελεστές χ και ψ για τον υπολογισμό της πιθανότητας επιλογής, τον συντελεστή «εξάτμισης» λ, τον αριθμό των μυρμηγκιών (Κ) και τον αριθμό των κύκλων (L).

**Σημείωση**

Αυτή η εφαρμογή απαιτεί τη βιβλιοθήκη CGAL και nlohmann/json για την ανάλυση και την έξοδο των δεδομένων. Βεβαιωθείτε ότι έχετε εγκαταστήσει αυτές τις βιβλιοθήκες πριν από την εκτέλεση του προγράμματος.

*To run manually: -del dir built/mkdir built -cd built -cmake .. -make -./triangulation\_program*

*For the run.sh : chmod +x run.sh*

*For the cleanup.sh : chmod +x cleanup.sh*

*Before commitng changes run the cleanup.sh*

**Περιγραφή Συναρτήσεων:**

**1. isObtuse**

Αυτή η συνάρτηση ελέγχει αν ένα τρίγωνο είναι οξεία γωνία. Υπολογίζει τις τετραγωνικές αποστάσεις μεταξύ των κορυφών του τριγώνου και ελέγχει αν το άθροισμα των τετραγωνικών αποστάσεων δύο πλευρών είναι μικρότερο από την τετραγωνική απόσταση της τρίτης πλευράς.

**2. hasObtuseAngle**

Αυτή η συνάρτηση ελέγχει αν το τρίγωνο που σχηματίζεται από τρεις δοσμένες κορυφές έχει οξεία γωνία. Υπολογίζει τις τετραγωνικές αποστάσεις μεταξύ των κορυφών και ελέγχει αν κάποια από τις γωνίες είναι οξεία.

**3. countObtuseTriangles**

Αυτή η συνάρτηση μετράει τον αριθμό των οξεών τριγώνων σε μια δομή τριγωνοποίησης. Διατρέχει όλες τις πεδία (τρίγωνα) της τριγωνοποίησης και χρησιμοποιεί τη συνάρτηση **hasObtuseAngle** για να ελέγξει αν το τρίγωνο έχει οξεία γωνία.

**4. getMidpoint**

Αυτή η συνάρτηση υπολογίζει το μέσο σημείο μεταξύ δύο δοσμένων κορυφών.

**5. getCentroid**

Αυτή η συνάρτηση υπολογίζει το κέντρο βάρους (κεντροειδές) ενός τριγώνου που σχηματίζεται από τρεις δοσμένες κορυφές.

**6. getCircumcenter**

Αυτή η συνάρτηση υπολογίζει το κέντρο του περιγεγραμμένου κύκλου ενός τριγώνου, χρησιμοποιώντας τις συντεταγμένες των κορυφών του τριγώνου.

**7. projectVertexOntoEdge**

Αυτή η συνάρτηση υπολογίζει την προβολή μιας κορυφής πάνω σε μια πλευρά του τριγώνου.

**8. sign**

Αυτή η συνάρτηση υπολογίζει το πρόσημο της περιοχής του παραλληλόγραμμου που σχηματίζεται από τρεις δοσμένες κορυφές, χρησιμοποιώντας τις συντεταγμένες τους.

**9. isPointInsideTriangle**

Αυτή η συνάρτηση ελέγχει αν ένα σημείο βρίσκεται μέσα σε ένα τρίγωνο, χρησιμοποιώντας την μέθοδο του προσήμου.

**10. generateCandidatePoints**

Αυτή η συνάρτηση δημιουργεί υποψήφια σημεία (Steiner points) για ένα τρίγωνο, συμπεριλαμβανομένων του περιγεγραμμένου κέντρου, των μέσων σημείων των πλευρών και του κεντροειδούς.

**11. evaluateTriangulation**

Αυτή η συνάρτηση αξιολογεί την τριγωνοποίηση και ποινικοποιεί τα οξεία τρίγωνα, υπολογίζοντας ένα σκορ που βασίζεται στον αριθμό των οξεών τριγώνων.

**12. localSearchForSteinerPoints**

Αυτή η συνάρτηση εκτελεί τοπική αναζήτηση για σημεία Steiner, προσπαθώντας να βελτιώσει την τριγωνοποίηση μειώνοντας τον αριθμό των οξεών τριγώνων.

**13. extractTriangulationResults**

Αυτή η συνάρτηση εξάγει τα αποτελέσματα της τριγωνοποίησης σε μορφή JSON.

**14. initializePheromones**

Αυτή η συνάρτηση αρχικοποιεί τις φερομόνες για κάθε πεδίο της τριγωνοποίησης.

**15. calculateHeuristic**

Αυτή η συνάρτηση υπολογίζει μια ηθική τιμή για ένα τρίγωνο, βασισμένη στην αναλογία του περιγεγραμμένου κύκλου προς την μεγαλύτερη πλευρά.

**16. updatePheromones**

Αυτή η συνάρτηση ενημερώνει τις φερομόνες με βάση τις βελτιώσεις που έγιναν κατά τη διάρκεια ενός κύκλου.

**17. antColonyOptimization**

Αυτή η συνάρτηση εκτελεί βελτιστοποίηση με αποικία μυρμηγκιών για την αναζήτηση βελτιωμένων σημείων Steiner, χρησιμοποιώντας φερομόνες για να καθοδηγήσει την αναζήτηση και να μειώσει τον αριθμό των οξεών τριγώνων.

**18. generateRandomSteinerPoint**

Αυτή η συνάρτηση δημιουργεί τυχαία σημεία Steiner, επιλέγοντας τυχαία τρεις κορυφές από το κυρτό περίγραμμα και υπολογίζοντας ένα σημείο μέσα στο τρίγωνο που σχηματίζεται από αυτές.

**19. simulatedAnnealing**

Αυτή η συνάρτηση εκτελεί προσομοιωμένη αναγόμωση για να βελτιώσει την τριγωνοποίηση, προσπαθώντας να μειώσει τον αριθμό των οξεών τριγώνων μέσω τυχαίων προσθηκών σημείων Steiner και αξιολόγησης της ενέργειας του συστήματος.

**20. performTriangulation**

Αυτή η συνάρτηση εκτελεί την τριγωνοποίηση με βάση τα δεδομένα εισόδου, εισάγοντας σημεία και περιορισμούς, και επιλέγοντας τη μέθοδο τριγωνοποίησης που θα χρησιμοποιηθεί.

**21. writeOutput**

Αυτή η συνάρτηση γράφει τα αποτελέσματα της τριγωνοποίησης σε αρχείο JSON, περιλαμβάνοντας τις κορυφές, τις ακμές και τα σημεία Steiner.

**22. parseInput**

Αυτή η συνάρτηση αναλύει τα δεδομένα εισόδου από ένα αρχείο JSON και τα αποθηκεύει σε μια μεταβλητή τύπου json για περαιτέρω επεξεργασία.

**Link to GitHub repository:**[**https://github.com/HackeRinaa/Non-Obtuse-Triangulation-PSLG.git**](https://github.com/HackeRinaa/Non-Obtuse-Triangulation-PSLG.git)