- ΟΝΟΜΑΤΕΠΩΝΥΜΑ ΜΕΛΩΝ ΚΑΙ ΑΜ:

- Δημήτριος Χρυσός - 1115202100275

- Αναστάσιος Μουμουλίδης - 1115202100108

- Οι μέθοδοι που επιλέξαμε:

- Και για τις 3 μεθόδους (ls, sa, ant) έχουμε επιλέξει να testάρουμε τους ίδιους συνδυασμούς μεθόδων Steiner:

- projection

- projection, circumcenter, merge

- projection, midpoint, centroid

- όλες (projection, centroid, midpoint, circumcenter, merge)

- Γιατί επιλέξαμε αυτές τις μεθόδους;

- Αρικά, όπως παρατηρήσαμε και από τις προηγούμενες εργασίες, η πιο αποτελεσματική μέθοδος με τεράστια διαφορά από τη 2η είναι η projection. Για τον λόγο αυτόν, επιλέξαμε να αφιερώσουμε ένα τεστ μόνο σε αυτή τη μέθοδο. Επιπλέον, ο παραπάνω είναι και ο λόγος που απορρίψαμε όλους τους συνδυασμούς μεθόδων που δεν περιέχουν την projection, καθώς αφενός θα είναι μη αποδοτικοί και αφετέρου στα μεγάλα instances είναι μη βιώσιμοι για το σύστημα.

- Ο δεύτερος συνδυασμός που επιλέξαμε είναι αυτός της projection με circumcenter, merge. Αυτό γιατί οι μέθοδοι circumcenter, merge είναι οι μόνες δύο μέθοδοι που ελέγχουν την εισαγωγή σημείου έξω από το αυστηρό πλαίσιο του ενός συγκεκριμένου αμβλυγωνίου τριγώνου. Επομένως, σε παραδείγματα με κλειστούς περιορισμούς, κάτι τέτοιο ίσως είναι αρκετά πιο αποδοτικό από τη σκέτη projection.

- Ο τρίτος συνδυασμός που επιλέξαμε είναι αυτός της projection με midpoint, centroid. Αυτό γιατί παρότι οι midpoint, centroid ελέγχουν την εισαγωγή σημείου μόνο μέσα στο πλαίσιο ενός συγκεκριμένου τριγώνου, όπως και η projection, θέλουμε να καλύψουμε και την περίπτωση που σε κάποιο παράδειγμα κάποια από τις πρώτες δύο μεθόδους είναι πιο αποτελεσματική από την projection.

- Ο τελευταίος συνδυασμός που επιλέξαμε είναι αυτός που περιέχει όλες τις μεθόδους. Κι αυτό για να δούμε τη συμπεριφορά των instances όταν έχουν στη διάθεσή τους όλη τη γκάμα των μεθόδων που υλοποιήσαμε, και να τη συγκρίνουμε με αυτές των προηγούμενων συνδυασμών.

- Κατηγοριοποίηση των instances:

- Για τον διαχωρισμό των instances σε 5 κατηγορίες, με βάση την εκφώνηση, τρέξαμε το script count\_categories.py. Το πόσα και ποια instances υπάγονται σε κάθε κατηγορία, μπορείτε να το δείτε στον φάκελο tests\_categorized.

- Πάμε να δούμε πόσα instances έχει η κάθε κατηγορία:

- Α. Κυρτό boundary (που ταυτίζεται με το Κυρτό Περίβλημα ) χωρίς περιορισμούς - 40 instances

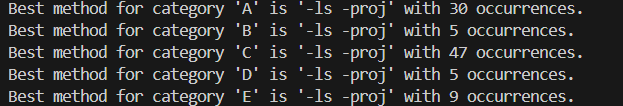
- Β. Κυρτό boundary με «ανοιχτούς» περιορισμούς - 7 instances

- Γ. Κυρτό boundary με περιορισμούς που αποτελούν ακμές πολυγώνων που σχηματίζονται εντός του boundary, συμπεριλαμβανομένων των ακμών που βρίσκονται επί του boundary («κλειστοί περιορισμοί») - 68 instances

- Δ. Μη κυρτό boundary με ευθύγραμμα τμήματα παράλληλα στους άξονες χωρίς περιορισμούς - 13 instances

- Ε. Μη κυρτό boundary, ακανόνιστο, που δεν εντάσσεται στις κατηγορίες Α-Δ - 22 instances

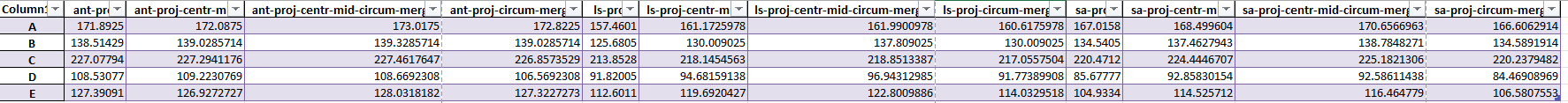
- Πιο αποδοτικοί συνδυασμοί μεθόδων ανά κατηγορία:

- Για να βρούμε αυτή τη πληροφορία, τρέξαμε το python script extract\_best\_method\_per\_category.py και ανακαλύψαμε ότι για όλες τις κατηγορίες Α-Ε, η πιο αποδοτική μέθοδος είναι η local search με projection. Πιο συγκεκριμένα, παρατίθενται τα αποτελέσματα του script:  


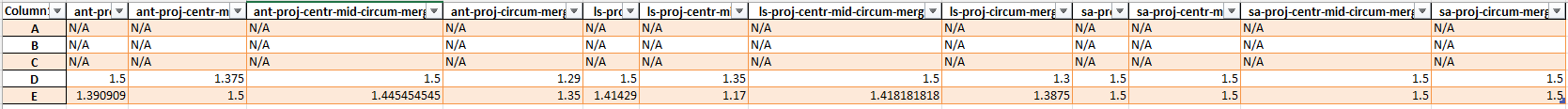
- Πάμε τώρα και στους πίνακες. Για την ανάλυση των αποτελεσμάτων μας, έχουμε φτιάξει δύο βοηθητικούς πίνακες, οι οποίοι κατασκευάστηκαν με τα python scripts create\_table\_A και create\_table\_B:

A. Tον πίνακα Α, που περιέχει ανά κατηγορία, τον μέσο όρο της ενέργειας (ή του ρυθμού σύγκλισης) για κάθε συνδυασμό μεθόδων. Π.χ. το κελί που αντιστοιχεί στη γραμμή Α και στη στήλη ls -proj, περιέχει τον μέσο όρο της ενέργειας (ή του ρυθμού σύγκλισης) όλων των instances της κατηγορίας Α, όταν εκτελέστηκαν με τη μέθοδο local search με projection. Τον πίνακα μπορείτε να δείτε αναλυτικά στον κατάλογο instances\_combinations\_values, στο αρχείο method\_averages.xlsx

B. Τον πίνακα Β, που περιέχει ανά κατηγορία, τον μέσο όρο της διαφοράς ενέργειας πριν και μετά την τυχαιοποίηση (δηλαδή το τελική ενέργεια – ενέργεια πριν τη τυχαιοποίηση) για κάθε συνδυασμό μεθόδων. Π.χ. το κελί που αντιστοιχεί στη γραμμή Α και ls proj περιέχει τον μέσο όρο της διαφοράς της ενέργειας όλων των instances της κατηγορίας Α, όταν εκτελέστηκαν με τη μέθοδο local search projection, πριν και μετά τη τυχαιοποίηση. Αν σε ένα κελί υπάρχει η τιμή Ν/Α, αυτό σημαίνει ότι σε κανένα από τα instances της κατηγορίας που εκτελέστηκαν με τη συγκεκριμένη μέθοδο δε χρησιμοποιήθηκε η τυχαιοποίηση. Τον πίνακα μπορείτε να δείτε αναλυτικά στον κατάλογο instances\_combinations\_values, στο αρχείο compare\_randomization.xlsx

- Ας ξεκινήσουμε ρίχνοντας μια ματιά στον πίνακα Α

Πράγμστι, παρατηρούμε ότι για όλες τις κατηγορίες η local search με projection έχει τον χαμηλότερο μέσο όρο ενέργειας.

- Πάμε τώρα και στον πίνακα Β

Όπως είπαμε και πριν, το Ν/Α σημαίνει πως κανένα instance των κατηγοριών Α, Β, C δε χρησιμοποιεί την τυχαιοποίηση. Αυτό συμβαίνει είτε επειδή τα αμβλυγώνια μηδενίζονται (το παράδειγμα συγκλίνει), είτε επειδή τελείωσε ο μέγιστος αριθμός επαναλήψεων L στη local search, είτε επειδή κάθε τυχαίο σημείο που εισάγεται αυξάνει τα αμβλυγώνια. Πάμε τώρα στις υπόλοιπες τιμές. Παρατηρούμε ότι όλοι οι αριθμοί είναι θετικοί, το οποίο σημαίνει πως η τελική ενέργεια είναι μεγαλύτερη από την ενέργεια πριν την τυχαιοποίηση. Άρα, η τυχαιοποίηση, όπου χρησιμοποιείται, όχι μόνο δε βελτιώνει την τριγωνοποίηση, αλλά τη κάνει και ελάχιστα χειρότερη (επειδή εισάγει 5 επιπλέον σημεία Steiner, χωρίς να μειώνει τα αμβλυγώνια).

- Τέλος, οι βέλτιστες μέθοδοι όλων των instances βρίσκονται στον κατάλογο test\_categories, ενώ τα outputs όλων των instances βρίσκονται στον κατάλογο outputs/test\_instances.