

- 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;

- Αρχικά, 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.

- Σημείωση: σαν (σταθερές) παραμέτρους για τις μεθόδους χρησιμοποιήσαμε:

A = 3, B = 0.3

- 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:

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

- B. 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

- E. Μη κυρτό boundary, ακανόνιστο, που δεν εντάσσεται στις κατηγορίες A-Δ - 22 instances

- Διαμόρφωση δεδομένων για ανάλυση:

- Για την κατάλληλη διαμόρφωση των δεδομένων που θέλαμε να αναλύσουμε, χρησιμοποιήσαμε το python script `test_instances.py`. Αρχικά, αξιοποιούμε την κατηγοριοποίηση των instances με βάση το `tests_categorized` που αναφέραμε παραπάνω. Μέσω του script, για κάθε κατηγορία, τρέχουμε όλα τα instances της με όλους τους συνδυασμούς μεθόδων που αναφέραμε παραπάνω, έτσι ώστε να βρούμε τον πιο αποδοτικό για κάθε instance. Τα αποτελέσματα υπάρχουν στον κατάλογο `test_categories`. Πέρα από την καλύτερη μέθοδο, στον κατάλογο `instances_combinations_values` υπάρχουν για κάθε συνδυασμό μεθόδων κάθε instance κάθε κατηγορίας, ο ρυθμός σύγκλισης ή η ενέργεια και έπειτα η ενέργεια πριν την τυχαιοποίηση, αν αυτή χρησιμοποιήθηκε. Αν δε χρησιμοποιήθηκε τυχαιοποίηση, η δεύτερη τιμή είναι -1.

- 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 τις κατηγορίες A-D, η πιο αποδοτική μέθοδος είναι η `local search` με `projection`, ενώ για την κατηγορία E η πιο αποδοτική μέθοδος είναι η `sa` με `projection`. Πιο συγκεκριμένα, παρατίθενται τα αποτελέσματα του script:

```
Best method for category 'A' is '-ls -proj' with 23 occurrences.  
Best method for category 'B' is '-ls -proj' with 5 occurrences.  
Best method for category 'C' is '-ls -proj' with 31 occurrences.  
Best method for category 'D' is '-ls -proj' with 4 occurrences.  
Best method for category 'E' is '-sa -proj' with 8 occurrences.
```

- **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. Τον πίνακα A, 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 A και στη στήλη `ls -proj`, περιέχει τον μέσο όρο της ενέργειας (ή του ρυθμού σύγκλισης) όλων των instances της κατηγορίας A, 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 B, 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 A και `ls proj` περιέχει τον μέσο όρο της διαφοράς της ενέργειας όλων των instances της κατηγορίας A, όταν εκτελέστηκαν με τη μέθοδο `local search projection`, πριν και μετά τη τυχαιοποίηση. Αν σε ένα κελί υπάρχει η τιμή N/A, 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 A

Ant				
Category	ant-proj	ant-proj-centr-mid	ant-proj-centr-mid-circum-merge	ant-proj-circum-merge
A	172.785	172.98	172.665	172.56
B	139.2857143	139.1571429	138.6	138.6428571
C	227.0338235	226.8794118	227.7132353	226.9102941
D	108.2538462	107.4230769	108.8769231	108.3923077
E	127.3363636	127.4181818	127.6090909	127.3363636
LS				
Category	ls-proj	ls-proj-centr-mid	ls-proj-centr-mid-circum-merge	ls-proj-circum-merge
A	158.0825978	161.1050978	163.2500978	160.9025978
B	125.6804536	133.4804536	134.2518821	130.009025
C	214.4042798	219.1469269	219.6675151	217.5057504
D	91.82005292	96.94312985	96.92005292	94.31236062
E	114.005679	122.7873522	122.9100795	116.4329518
SA				
Category	sa-proj	sa-proj-centr-mid	sa-proj-centr-mid-circum-merge	sa-proj-circum-merge
A	166.2390707	170.7143853	170.2615798	165.9625554
B	134.9690593	135.1346759	136.69062	132.9382689
C	220.882537	223.7540713	224.7454293	220.883976
D	84.40651	90.34338508	92.55883154	84.08648869
E	104.480054	115.4226262	116.1630964	107.1518296

Παρατηρούμε ότι οι σκέτες `projection` μέθοδοι είναι κατά κύριο λόγο οι πιο αποτελεσματικές μέθοδοι. Επιπλέον, στις κατηγορίες A έως C η πιο αποτελεσματική μέθοδος είναι η `local search`, ενώ στις κατηγορίες D, E, τα παραδείγματα τρέχουν πιο αποδοτικά με τη `sim annealing`.

Όσον αφορά τις μεθόδους, παρατηρούμε ότι στη local search η projection είναι με διαφορά η πιο αποδοτική. Στη sim annealing, παρατηρούμε ότι οι projection με την projection, circumcenter, merge είναι πολύ κοντά, ενώ στην ant δεν παρατηρούνται μεγάλες αυξομειώσεις μεταξύ των συνδυασμών των μεθόδων Steiner.

- Πάμε τώρα και στον πίνακα B

Ant				
Category	ant-proj	ant-proj-centr-mid	ant-proj-centr-mid-circum-merge	ant-proj-circum-merge
A	N/A	N/A	N/A	N/A
B	N/A	N/A	N/A	N/A
C	N/A	N/A	N/A	N/A
D	1.5	1.15	1.35	1.41
E	1.377272727	1.47	1.418181818	1.404545455
LS				
Category	ls-proj	ls-proj-centr-mid	ls-proj-centr-mid-circum-merge	ls-proj-circum-merge
A	N/A	N/A	N/A	N/A
B	N/A	N/A	N/A	N/A
C	N/A	N/A	N/A	N/A
D	1.5	1.5	1.44	1.5
E	1.5	1.390909091	1.5	1.38
SA				
Category	sa-proj	sa-proj-centr-mid	sa-proj-centr-mid-circum-merge	sa-proj-circum-merge
A	N/A	N/A	N/A	N/A
B	N/A	N/A	N/A	N/A
C	N/A	N/A	N/A	N/A
D	1.5	1.5	1.5	1.5
E	1.5	1.5	1.5	1.5

Όπως είπαμε και πριν, το N/A σημαίνει πως κανένα instance των κατηγοριών A, B, 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 με τις βέλτιστες μεθόδους βρίσκονται στον κατάλογο outputs/test_instances.

- Παραθέτουμε ενδεικτικά τις πειραματικές λύσεις από 5 παραδείγματα, 1 από κάθε κατηγορία:

Όνομα instance: point-set_10_ae0fff93

Κατηγορία: A

Βέλτιστη μέθοδος: sa -projection

Τελικός Αριθμός Αμβλυγωνίων: 0

Αριθμός Steiner: 11

Ρυθμός Σύγκλισης: 1.30488

Ενέργεια: -

Τυχαιοποίηση: Όχι

Όνομα instance: simple-polygon-exterior-20_10_ce9152de

Κατηγορία: B

Βέλτιστη μέθοδος: sa -projection, -centroid, -midpoint

Τελικός Αριθμός Αμβλυγωνίων: 2

Αριθμός Steiner: 47

Ρυθμός Σύγκλισης: -

Ενέργεια: 20.1

Τυχαιοποίηση: Όχι

Όνομα instance: simple-polygon-exterior_10_c5616894

Κατηγορία: C

Βέλτιστη μέθοδος: ls -proj

Τελικός Αριθμός Αμβλυγωνίων: 0

Αριθμός Steiner: 45

Ρυθμός Σύγκλισης: 1.32022

Ενέργεια: -

Τυχαιοποίηση: Όχι

Όνομα instance: ortho_10_d2723dcc

Κατηγορία: D

Βέλτιστη μέθοδος: sa -proj, -centr, -mid, -circum, -merge

Τελικός Αριθμός Αμβλυγωνίων: 0

Αριθμός Steiner: 22

Ρυθμός Σύγκλισης: 1.27683

Ενέργεια: -

Τυχαιοποίηση: Όχι

Όνομα instance: simple-polygon_10_272aa6ea

Κατηγορία: E

Βέλτιστη μέθοδος: sa -projection, -centroid, -midpoint

Τελικός Αριθμός Αμβλυγωνίων: 0

Αριθμός Steiner: 6

Ρυθμός Σύγκλισης: 0.573618

Ενέργεια: -

Τυχαιοποίηση: Όχι