



Τεχνητή Νοημοσύνη

## 2<sup>ο</sup> Θέμα

26/2/2018

Λεούσης Σάββας

ΑΜ:03114945

ΣΗΜΜΥ

Εθνικό Μετσόβιο Πολυτεχνείο

## Εισαγωγή

Στο πλαίσιο αυτής της εργασίας υλοποιήθηκε ένα πρόγραμμα σε γλώσσα Java που αποτελεί τον βασικό κορμό μιας ευφυούς υπηρεσίας εξυπηρέτησης πελατών ταξί. Ως είσοδο, το πρόγραμμα δέχεται 5 αρχεία τύπου .csv και μία αριθμητική παράμετρο η οποία καθορίζει το μέγιστο μέγεθος μετώπου αναζήτησης. Η υπηρεσία εντοπίζει και ειδοποιεί το ταξί που μπορεί να μεταβεί πιο γρήγορα στη θέση του πελάτη. Απεικονίζει στον πελάτη τα κοντινότερα ταξί καθώς και την κατάταξή τους με βάση την αξιολόγησή τους. Για ό,τι αφορά τη γνώση σχετικά με τον κόσμο που διαθέτει το πρόγραμμα, χρησιμοποιείται η Prolog.

## Περιγραφή του Προγράμματος

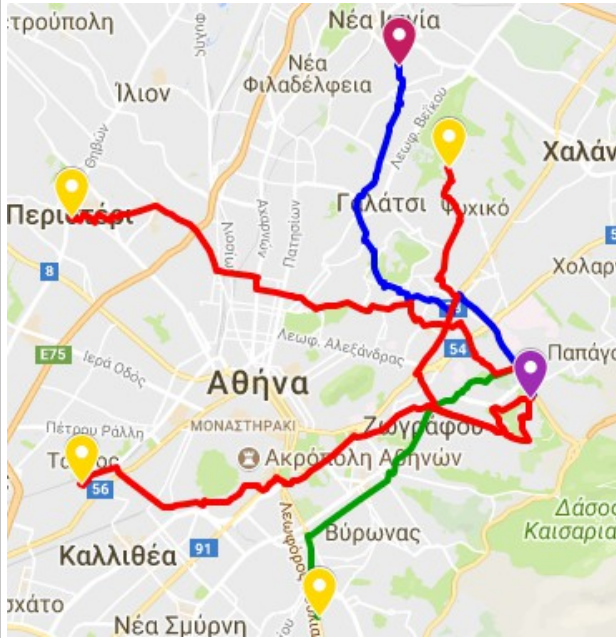
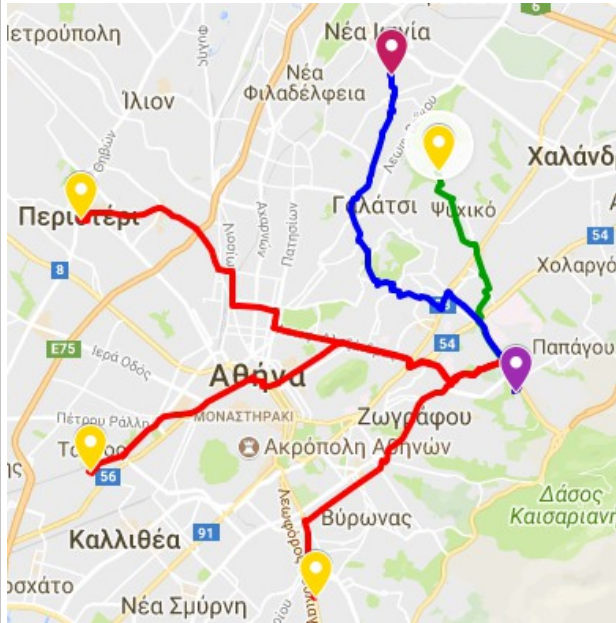
Η κλήση του προγράμματος απαιτεί από τον χρήστη και τον καθορισμό 6 παραμέτρων:

1. Ένα αρχείο client.csv που περιέχει τις συντεταγμένες X,Y του πελάτη.
2. Ένα αρχείο nodes.csv που περιέχει τις συντεταγμένες X,Y όλων των σημείων του χάρτη, το ID των δρόμων, καθώς και τις ονομασίες τους.
3. Ένα αρχείο taxis.csv που περιέχει τις συντεταγμένες X,Y όλων των ταξί στην περιοχή ενδιαφέροντος, μαζί με τον αντίστοιχο ID τους.
4. Ένα αρχείο lines.csv που περιέχει πληροφορίες για τους δρόμους του χάρτη,
5. Ένα αρχείο traffic.csv που περιέχει πληροφορίες για την κίνηση στους δρόμους του χάρτη,
6. Μία παράμετρο Q\_Cap, η οποία είναι ένας ακέραιος θετικός αριθμός που καθορίζει το μέγιστο μέτωπο αναζήτησης του αλγορίθμου A\*.

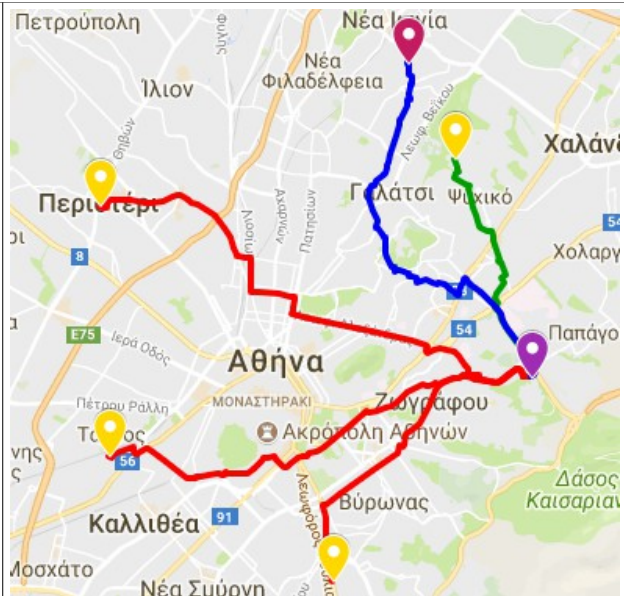
Σε αυτήν την εργασία προστέθηκε στο πρόγραμμα, και συγκεκριμένα στον αλγόριθμο A\*, ο έλεγχος της βάσης γνώσης Prolog, προτού κινηθεί ο αλγόριθμος σε κάποιο γειτονικό κόμβο καθώς και ο υπολογισμός της ευριστικής προτεραιότητάς του με βάση την κίνηση του δρόμου. Για την κατασκευή της βάσης γνώσης χρησιμοποιήθηκε η κλάση **KnowledgeBase**, η οποία δημιουργεί τα γεγονότα (facts) της Prolog με βάση τα csv. Συγκεκριμένα δημιουργεί 2 αρχεία γνώσης prolog\_base\_1.pl και prolog\_base\_2.pl, τα οποία περιέχουν τα γεγονότα pointRoadPosition(3) και highway(2),oneway(2), forbidden(1) και traffic(3). Επίσης κατασκευάζει μία βάση κανόνων prolog\_rules.pl με τα κατηγορήματα canMoveFromTo(X,Y,R) και priority(R,T,Z), έτσι ώστε να ελέγχεται η δυνατότητα πρόσβασης από έναν κόμβο στον άλλο, καθώς και ο βαθμός προτεραιότητας αντίστοιχα. Το pointRoadPosition δηλώνει το ID του κόμβου, το ID του δρόμου που ανήκει και τον αύξοντα αριθμό του στον δρόμο αυτό. Τα highway και oneway δηλώνουν αν ένας δρόμος με ένα ID είναι διπλής ή μονής κατεύθυνσης αντίστοιχα, μαζί με έναν αριθμό που δηλώνει τον τύπο δρόμου. Τέλος το traffic δηλώνει την κίνηση ενός δρόμου με το ID του, το εύρος ώρας και το μέγεθος της κίνησης εκείνη την ώρα. Η επέκταση στον A\* σε σχέση με την 1η εργασία βρίσκεται στα JIP queries που εκτελούνται πριν την πρόσθεση ενός γείτονα στην queue του αλγορίθμου. Πρώτα ελέγχεται αν είναι εφικτή η μετάβαση σε αυτό το σημείο, και μόνο αν είναι δυνατή η μετάβαση τελικά, τότε ελέγχει και το πόσο καλή είναι η μετάβαση σε αυτό το σημείο. Με αυτόν τον τρόπο βρίσκουμε και πάλι τα καλύτερα ταξί για τον πελάτη, και τυπώνουμε σε ένα αρχείο .txt μία κατάταξη με βάση την απόσταση και άλλη μία με βάση την αξιολόγηση (4 καλύτερα ταξί). Όπως και για όλα τα ταξί, έτσι και για τη διαδρομή Πελάτης-Προορισμός εκτελείται ο ίδιος αλγόριθμος A\* (με Prolog) και τυπώνεται το αντίστοιχο αρχείο .kml.

## Πειραματικά Αποτελέσματα

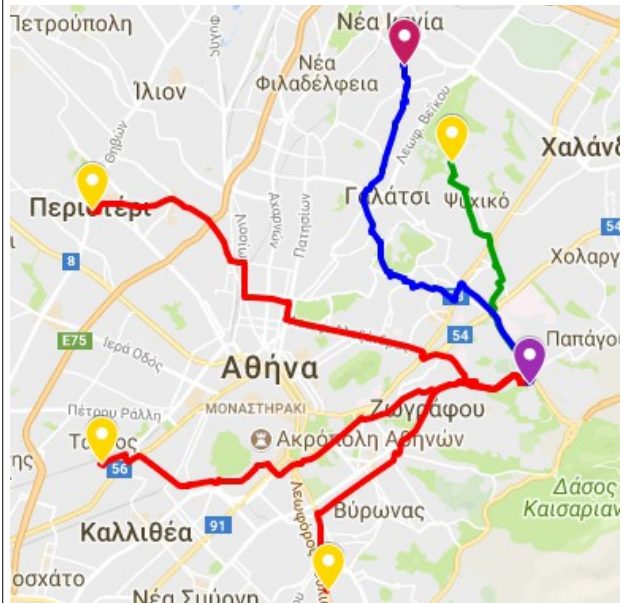
Παρακάτω παρατίθεται τα αποτελέσματα του προγράμματος για διαφορετικές εισόδους τοποθεσιών ταξί και πελάτη από τα δεδομένα αρχεία client.csv, nodes.csv, taxis.csv, lines.csv, traffic.csv (ή client\_ex.csv, taxis\_ex.csv για το δικό μας παράδειγμα) υπό τη μορφή πινάκων. Η πράσινη διαδρομή είναι η συντομότερη διαδρομή από όλα τα ταξί προς τον πελάτη, οι κόκκινες διαδρομές είναι των υπολοίπων ταξί και η μπλε διαδρομή είναι η διαδρομή από τον πελάτη στον προορισμό του:

Μέγεθος Μετώπου Αναζήτησης	Οπτική Απεικόνιση στο Google My Maps
50	
100	

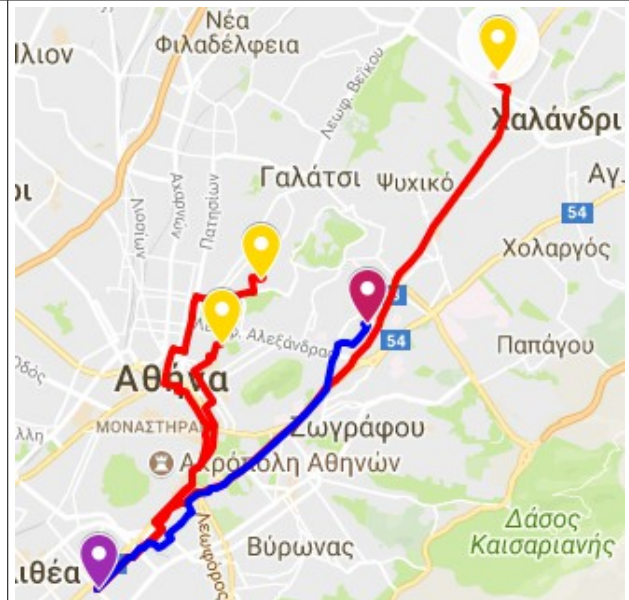
500



1000



Παράδειγμα - 500



Όπως παρατηρήσαμε και στην 1η εργασία, το μέγιστο μέγεθος μετώπου αναζήτησης παίζει σημαντικό ρόλο στην ποιότητα της διαδρομής που βρίσκει ο αλγόριθμος A\*. Όσο μεγαλύτερο είναι το μέγεθος, τόσο καλύτερη διαδρομή βρίσκει ο αλγόριθμος για όλα τα ταξί. Περισσότερα στατιστικά δεδομένα μπορούν να βρεθούν για κάθε μέγεθος μετώπου αναζήτησης στα αντίστοιχα Stats\_TAXI\_AI.out files για τις διαδρομές ταξί-πελάτης και πελάτης-προορισμός.

### **Αρχεία Εργασίας**

Στον φάκελο της εργασίας παρατίθενται:

1. Ο πηγαίος κώδικας "TAXI\_AI.java" του προγράμματος.
2. Η παρούσα αναφορά σε μορφή .pdf
3. Στο φάκελο "CSV, KML & OUT files" τα αρχεία .csv που χρησιμοποιήθηκαν (μαζί με τα csv files του παραδείγματός μας) και τα αρχεία .kml, .out, .txt, για κάθε μέγεθος μετώπου αναζήτησης και για κάθε διαδρομή (ταξί-πελάτης και πελάτης-προορισμός). Στα αρχεία Client\_TAXI\_AI\_\*.txt περιέχονται οι έξοδοι της υπηρεσίας προς τον πελάτη, όπως ακριβώς θα του τυπώνονταν ώστε να επιλέξει το ταξί του (όπου \* το μέγεθος μετώπου αναζήτησης). Στα αρχεία Stats\_TAXI\_AI\_\*.out περιέχονται στατιστικά δεδομένα για την εκτέλεση του A\* (αριθμός βημάτων, πραγματικό μέγιστο μέγεθος μετώπου αναζήτησης, απόσταση), όπου \* το μέγεθος μετώπου αναζήτησης.
4. Στον φάκελο "Prolog Base,Rules" τα αρχεία .pl που κατασκευάζουν την βάση γνώσης του προγράμματος, όπου prolog\_base\_1.pl και prolog\_base\_2.pl τα γεγονότα και prolog\_rules.pl οι κανόνες.