ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΔΟΜΟΣΤΑΤΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΜΕΘΟΔΟΛΟΓΙΑΣ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΟΝ ΥΠΟΛΟΓΙΣΜΟ ΕΡΥΘΡΑΣ ΓΡΑΜΜΗΣ ΤΗΣ ΜΗΚΟΤΟΜΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Εμμανουέλας Ι. Μιχαήλ

Επιβλέπων Καθηγητής: Φραγκιαδάκης Μιχαήλ
Συνεπιβλέπων Καθηγητής : Στάμος Αθανάσιος

Αθήνα, Νοέμβριος 2018

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα Καθηγητή του τμήματος Πολιτικών Μηχανικών Ε.Μ.Π. κ. Μ. Φραγκιαδάκη, για την ανάθεση και την επίβλεψη της Διπλωματικής μου Εργασίας. Στον συνεπιβλέποντα κ. Α. Στάμο, για τις ουσιαστικές παρατηρήσεις και συμβουλές του σε όλα τα στάδια εκπόνησης της Διπλωματικής Εργασίας, αφιερώνοντας αρκετό χρόνο εργασίας και συζήτησης καθώς και για την υπομονή του. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, καθώς και όσους βοήθησαν με τον οποιονδήποτε τρόπο στη διαμόρφωση της παρούσας Διπλωματικής Εργασίας, ιδιαιτέρως τον Σταμάτη.

Περίληψη

Η χάραξη μίας οδού είναι μια εξαιρετικά περίπλοκη διαδικασία καθώς από ένα σύνολο άπειρων λύσεων επιλέγεται μοναδική λύση η οποία αποτελεί συγκερασμό αλληλοσυγκρουόμενων μεγεθών. Στη παρούσα διπλωματική εργασία ερευνήθηκε η δυνατότητα αυτοματοποίησης της κατακόρυφης χάραξης της οδού και συντάχθηκε πρόγραμμα για την σχεδίασή της. Χρησιμοποιούνται για την βελτιστοποίηση της, δύο στοχαστικές μέθοδοι μια ευρετική, η μέθοδος Monte Carlo και μία μεθευρετική, η μέθοδος των Γενετικών Αλγορίθμων, η οποία θεωρείται αξιόπιστη για την βελτιστοποίηση της κατακόρυφης χάραξης σε σχέση με άλλες μεθόδους. Από το πρόγραμμα παράγεται βελτιστοποιημένη ερυθρά με κριτήρια την ελαχιστοποίηση των απόλυτων τιμών των εμβαδών των χωματισμών και την τήρηση των κανονισμών οδοποιίας. Ο χρήστης εισάγει την ταχύτητα μελέτης, την κατηγορία της οδού, τον τύπο του εδάφους και τα στοιχεία των χιλιομετρικών θέσεων και υψομέτρων του εδάφους για την οδό. Παράλληλα δίνεται η δυνατότητα εισαγωγής του πλήθους των θλάσεων της ερυθράς. Εναλλακτικά παράγονται αυτόματα από το πρόγραμμα τρεις λύσεις για κάθε πλήθος θλάσεων από μηδέν έως ένα μέγιστο πλήθος θλάσεων που εξαρτάται από το μήκος της οδού. Το πρόγραμμα ελέγχθηκε με στοιχεία πραγματικών οδών που έχουν ήδη μελετηθεί/κατασκευαστεί σε όλη την Ελλάδα. Οι υψομετρικές χαράξεις που προέκυψαν προσεγγίζουν κατά το δυνατόν το εδαφικό ανάγλυφο τηρώντας έτσι τις απαιτήσεις του οδικού σχεδιασμού. Σε όσες οδούς ήταν διαθέσιμη η υψομετρική χάραξη, έγινε σύγκριση με αυτή του προγράμματος, και η χάραξη του προγράμματος ήταν παρόμοια ή καλύτερη.

Λέξεις κλειδιά: Βελτιστοποίηση κατακόρυφης χάραξης οδού, Αλγόριθμος Monte Carlo, Γενετικοί Αλγόριθμοι, Εφαρμογή Βελτιστοποίησης

Abstract

Road alignment is a process of high complexity, as from a set of inexhaustible solutions a unique solution is chosen which is a mixture of conflicting variables. In this thesis an attempt was made to automate vertical alignment of the road. For the research implementation, two stochastic methods were used: a heuristic, the Monte Carlo method and a metaheuristic, genetic algorithms, which are considered reliable for the optimization of vertical alignment in relation to other methods. The optimized grade line is produced minimizing the volume of earthworks while respecting the applicable national road rules. The user inputs road speed, road category, soil type and a list of geometrical data for the road. At the same time it is possible to enter the number of inflection points of the grade line. Otherwise, the program produces three solutions with different number of inflection points of the grade line between zero and a maximum number of inflection points which depends on the length of the road. The program was tested with real world data of roads which have already been studied/constructed all over Greece. The grade lines produced by the program approach optimally the terrain, thus respecting the demands of road design. The produced grade lines were also compared to the original grade lines of the available road studies, and the produced grade lines were found to be similar or better.

Keywords: Optimization of Vertical Alignment, Genetic Algorithms, Monte Carlo Algorithm, Optimization Methods

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

1 Εισαγωγή

Στη παρούσα διπλωματική εργασία εξετάζεται το πολυσύνθετο ζήτημα της κατακόρυφης χάραξης μιας οδού. Το ζήτημα αυτό πρέπει να ικανοποιεί αλληλοσυγκρουόμενους παράγοντες όπως

- Χωροταξικούς
- πολεοδομικούς
- διαμόρφωσης του οδικού χώρου
- οικονομίας κατά τη κατασκευή και τη λειτουργία της οδού
- εξοικονόμησης ενέργειας
- μείωσης στην εκπομπή καυσαερίων
- προστασίας του φυσικού περιβάλλοντος και μέριμνας για το τοπίο
- ασφάλειας και άνεσης της μηχανοκίνητης και μη κυκλοφορίας,
- υλοποίησης των απαιτήσεων της μηχανοκίνητης κυκλοφορίας, έτσι ώστε με τον ορισμό των βασικών στοιχείων μελέτης να επιτυγχάνεται μία συνολικά ισορροπημένη λύση.

Επιπλέον πρέπει να ληφθεί υπόψη η σημασία των οδικών έργων καθώς τα έργα οδοποιίας συντελούν στην αύξηση της ποιότητας ζωής των ανθρώπων μιας και διευκολύνουν την σύνδεση της ανθρώπινης δραστηριότητας. Μάλιστα όσοπιοσύγχρονη είναι μια κοινωνία τόσο εξυγχρονισμένες είναι και οι οδικές της υποδομές.

Εκτός όμως από τον κρίσιμο ρόλο που διαδραματίζουν στην καθημερινότητά μας έχουν βαρύνουσα σημασία και στον οικονομικό προϋπολογισμό ενός κράτους καθώς όπως είναι γνωστό από έρευνες καταλαμβάνουν σημαντικό ποσοστό του ΑΕΠ.

Πραγματοποιήθηκε προσπάθεια προσέγγισης αυτού του προβλήματος με την σύνταξη ενός αλγορίθμου που παράγει βελτιστοποιημένη την ερυθρά της οδού με τις μεθόδους Monte Carlo και των γενετικών αλγορίθμων.

1.1 Δομή Διπλωματικής Εργασίας

Κεφάλαιο 1

Εισαγωγή

Κεφάλαιο 2

Σε αυτό το κεφάλαιο γίνεται η βιβλιογραφική ανασκόπηση. Μελετήθηκαν έρευνες βελτιστοποίησης κυρίως γενετικών αλγορίθμων καθώς και προγράμματα οδοποιίας ευρείας χρήσης. Με βάση τη βιβλιογραφική ανασκόπηση επιλέχθηκε η κατάλληλη μέθοδος βελτιστοποίησης για την επιλύση του ζητήματος της χάραξης της ερυθράς.

Κεφάλαιο 3

Σε αυτό το κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο της εργασίας. Δηλαδή οι ισχύοντες κανόνες της οδοποιίας και η μεθοδολογία της κατακόρυφης χάραξης. Στη συνέχεια παρουσιάζονται οι μέθοδοι βελτιστοποίησης Monte Carlo και Γενετικών Αλγορίθμων οι οποίες και θα χρησιμοποιηθούν.

Κεφάλαιο 4

Στο κεφάλαιο αυτό παρουσιάζεται ο αλγόριθμος που συντάχθηκε για την έρευνα του υπό εξέταση ζητήματος. Ο αλγόριθμος συντάχθηκε χρησιμοποιώντας την γλώσσα προγραμματισμού python. Ως συνάρτηση στόχου στην εργασία ορίστηκε η ελαχιστοποίηση των απόλυτων τιμών των εμβαδών που σχηματίζονται από τις γραμμές του εδάφους και την ερυθρά της οδού. Στη συνέχεια πραγματοποιείται η βελτιστοποίηση με τις μεθόδους που επιλέχθησαν. Για την επιτάχυσνη της εκτέλεσης του αλγορίθμου κατά τη χρήση της μεθόδου Monte Carlo χρησιμοποιήθηκε η βιβλιοθήκη της python multiprocessoring η οποία κατανέμει μέρος των εργασιών σε κάθε υπόεπεξεργαστή. Ενώ για τη μέθοδο των γενετικών αλγορίθμων χρησιμοποιήθηκε αναλυτική επίλυση με βάση τον κανόνα του Crammer.

Κεφάλαιο 5

Σε αυτό το κεφάλαιο παρουσιάζονται τμήματα οδών τα για τα οποία χράχθηκε η ερυθρά της οδού με τη χρήση του αλγόριθμου που συντάχθηκε. Επίσης συγκρίνονται κάποιες από τις μηκοτομές που παράχθησαν με αντίστοιχες που έχουν σχεδιαστεί από μηχανικό.

Κεφάλαιο 6

Στο κεφάλαιο αυτό αναλύονται τα αποτελέσματα της μεθόδου και παραθέτονται τα συμπεράσματα του ερευνητή.

2 Βιβλιογραφική Ανασκόπηση

2.1 Συναφείς έρευνες

Με βάση τις υπάρχουσες έρευνες η χρήση μαθηματικών μοντέλων επιταχύνει και βελτιστοποιεί τη διαδικασία της χάραξης ενός οδικού έργου. Η βελτιστοποίηση του σχεδιασμού της οριζόντιας και κατακόρυφης χάραξης σταδιακά ή παράλληλα έχει προσεγγιστεί από διάφορα πρότυπα τα οποία αναφέρονται στον παρακάτω πίνακα. Επίσης στον πίνακα γίνεται έκθεση των πλεονεκτημάτων και μειονεκτημάτων τους. Επιπλέον σχετική βιβλιογραφία για τη χρήση των γενετικών αλγόριθμων στη κατακόρυφη χάραξη της οδού αναπτύσσεται στη συνέχεια.

Μέθοδοι	Πλεονεκτήματα	Μειονεκτήματα
Λογισμός Μεταβολών	Αποδίδει ομαλή χάραξη	Πολύπλοκη προτυποποίηση
	Πιθανόν εντοπίζει το ολικό βέλτιστο	Υψηλές υπολογιστικές απαιτήσεις
	Έχει συνεχή χώρο αναζήτησης	Πιθανή παγίδευση σε τοπικό βέλτιστό
		Δεν αποδίδει ασυνεχή στοιχεία κόστους (απαιτεί πολύ καλά ανεπτυγμένες συναρτήσεις)
Δικτυακή βελτιστοποίηση	Εύχρηστη	Απαιτεί μεγάλη μνήμη υπολογιστικού συστήματος
	Ανεπτυγμένοι αλγόριθμοι επίλυσης	Δεν παράγει λεία χάραξη
	Πιθανότητα εύρεσης του ολικού βέλτιστου	Χρησιμοποιεί διακριτά διαστήματα επίλυσης αντί για συνεχείς χώρους
Δυναμικός Προγραμματισμός	Εύχρηστη	Απαιτεί μεγάλη μνήμη υπολογιστικού συστήματος
	Ανεπτυγμένοι αλγόριθμοι επίλυσης	Δεν παράγει λεία χάραξη

	Πιθανότητα εύρεσης του ολικού βέλτιστου	Χρησιμοποιεί διακριτά διαστήματα επίλυσης αντί για συνεχείς χώρους
		Δυσκολία εφαρμογής σε οπισθόδρομη χάραξη
Πλήρης Απαρίθμηση	Πιθανότητα παραγωγής ρεαλιστικής χάραξης	Μη αποδοτική
	Εξετάζει την πλειοψηφία των περιορισμών και κόστους	Χρησιμοποιεί διακριτά διαστήματα επίλυσης αντί για συνεχείς χώρους
	Πιθανότητα εύρεσης του ολικού βέλτιστου	
Γραμμικός προγραμματισμός	Εύχρηστη	Μόνο γραμμικά στοιχεία κόστους και περιορισμοί
	Αναπτυγμένοι αλγόριθμοι επίλυσης	Περιορισμοί κλίσης και καμπυλότητας διατυπώνονται για περιορισμένο αριθμό στοιχείων
	Πιθανότητα εύρεσης του ολικού βέλτιστου	
	Παράγει λεία χάραξη	
	Συνεχής χώρος επίλυσης	
Αριθμητική αναζήτηση	Παράγει λεία χάραξη	Πιθανή παγίδευση σε τοπικό βέλτιστο
	Συνεχής χώρος επίλυσης	Πολύπλοκη προτυποποίηση
	Εξετάζει την πλειοψηφία των περιορισμών και κόστους	Δυσκολία προτυποποίησης διακριτών στοιχείων κόστους
		Υψηλές υπολογιστικές απαιτήσεις
Γενετικοί Αλγόριθμοι και Σύστημα Γεωγραφικών Πληροφοριών	Παράγει ρεαλιστική χάραξη (οριζοντιογραφία και μηκοτομή)	Εξαρτάται από την ακρίβεια των χωρικών δεδομένων
	Πιθανότητα εύρεσης του ολικού βέλτιστου	Απαίτηση υπολογιστικού περιβάλλοντος υψηλών προδιαγραφών
	Συνεχής χώρος επίλυσης	
	Εύχρηστη	
	Ανεπτυγμένοι αλγόριθμοι επίλυσης	

Εξετάζει τους σημαντικότερους περιορισμούς	
Παράγει χαράξεις με οπισθοτομία	

Πηγή: Jong(1998)

Οι Fwa, Chan, & Sim (2002) ανέπτυξαν ένα πρότυπο στο NUS, βασισμένο σε γενετικό αλγόριθμο για την ανάλυση της βέλτιστης κατακόρυφης χάραξης. Στο πρότυπο η κατακόρυφη χάραξη πραγματοποιείται για δεδομένη οριζοντιογραφία και η κατακόρυφη χάραξη είναι εναρμονισμένη με την οριζόντια. Το πρότυπο περιλαμβάνει επίσης περιορισμό μέγιστης κατά μήκος κλίσης, και συσχέτισης οριζοντιογραφικών και κατακόρυφων καμπύλων. Η συνάρτηση στόχου επιδιώκει την ελαχιστοποίηση του κόστους των χωματουργικών και της κατασκευής των οδοστρωμάτων. Το πεδίο επίλυσης αναπαριστάται από έναν πλέγμα οριζόντιων και κατακόρυφων γραμμών. Οι κατακόρυφες γραμμές κατανέμονταν ανά ίσα διαστήματα κατά μήκος της οριζοντιογραφικής χάραξης ενώ τα υψόμετρα επί των γραμμών αυτών αντιμετωπίζονταν ως μεταβλητές απόφασης. Το πεδίο είναι παραλληλόγραμμο επίπεδο που ορίζεται από τέσσερις ευθείες με συντελεστή διεύθυνσης την μέγιστη κλίση, αρνητική και θετική, και διέρχονται από τα σημεία αρχής και τέλους της οδού. Η προσέγγιση αυτή μειώνει τον αριθμό των παραγόμενων μη εφικτών λύσεων αποφεύγοντας την αναζήτηση εκτός της ορισμένη περιοχής.

Αρχικά επιλέχθηκε πληθυσμός εξακοσίων ατόμων αλλά δε βελτίωσε την αποδοτικότητα του αλγορίθμου επομένως υιοθετήθηκε πληθυσμός εφτακοσίων ατόμων. Οι πιθανότητες διασταύρωσης και μετάλλαξης ήταν πενήντα πέντε εκατοστά (0,55) και τρία εκατοστά (0,03) αντίστοιχα. Σε κάθε επανάληψη, το 90% των απογόνων συνδυαζόταν με το καλύτερο 10% των προγόνων για την επόμενη αναπαραγωγή. Ο πληθυσμός συνέκλινε κοντά στην πεντακοσιοστή (500) επανάληψη με σημαντικές βελτιώσεις στις πρώτες 150 επαναλήψεις. Ενώ για κάθε παραβίαση περιορισμού δόθηκε σταθερή ποινή.

Η έρευνα έδειξε πως οι τεχνικές γενετικών αλγορίθμων παρέχουν μια αξιόπιστη μέθοδο σε σχέση με άλλες μεθόδους βελτιστοποίησης για την επίλυση του προβλήματος της κατακόρυφης χάραξης. Το πρότυπο ήταν αρκετά ευέλικτο και ήταν ικανό να συμπεριλάβει μια μεγάλη ποικιλία περιορισμών. Όπως οι περιορισμοί κατά μήκος κλίσης και καμπυλότητας, τα κρίσιμα μήκη για τον έλεγχο των κλίσεων, τα σταθερά σημεία υψομέτρου και μέριμνα για την υπέρθεση οριζοντιογραφικών και κατακόρυφων καμπυλών. Αυτοί οι περιορισμοί δε λαμβάνονται συνήθως υπόψη στα περισσότερα πρότυπα καθώς είναι δύσκολη η προτυποποίησή τους. **Βέβαια, και σε αυτήν την περίπτωση η χάραξη μπορεί να διέλθει πεπερασμένο σύνολο υψομέτρων σε κάθε σημείο. Κατά συνέπεια, ελέγχεται μόνο ένα υποσύνολο του πεδίου αναζήτησης και δεν εξασφαλίζεται η απαίτηση**

για ολικό βέλτιστο (Fan, 2004). Τέλος, η προκύπτουσα χάραξη αναπαρίσταται με ευθύγραμμα τμήματα, υστερώντας με τον τρόπο αυτόν σε απαιτούμενη ακρίβεια για πραγματική εφαρμογή.

Στην έρευνα του Σ. Πολίτη (2016) εξετάζεται η χάραξη της οριζοντιογραφίας και της μηκοτομής της οδού και η βελτιστοποίηση τους με τη χρήση γενετικών αλγορίθμων. Στην έρευνα γίνεται εκτενής αναφορά στα στάδια μιας μελέτης οδοποιίας. Περιγράφονται τα στοιχεία Οδικών Χαράξεων, οι Προδιαγραφές του σχεδιασμού της οδού, οι αρχές της χάραξης στο χώρο αλλά και οι απαιτήσεις για ασφάλεια. Στη συνέχεια αναφέρονται οι μέθοδοι βελτιστοποίησης, η βελτιστοποίηση στα οδικά έργα και η Εφαρμογή προτύπων βελτιστοποίησης χαράξεων με γενετικούς αλγόριθμους.

Το πρότυπο που έχει αναπτυχθεί από τον ερευνητή περιλαμβάνει ένα αρχείο excel και το Evolver® 5.5 ένα λογισμικό πακέτο γενετικών αλγορίθμων που αναπτύχθηκε από την εταιρεία Axcelis® Inc. Αρχικά εισάγονται τα σημεία αρχής και τέλους της οδού. Δημιουργείται αυτόματα ο ευθύγραμμος άξονας της οδού. Έπειτα δημιουργείται η ερυθρά της οδού που ενώνει απλά τα σημεία αρχής και τέλους της οδού. Έπειτα εισάγεται το πλήθος των κορυφών της πολυγωνικής της οριζοντιογραφίας. Οι κορυφές αυτές τοποθετούνται ανά ίσα διαστήματα επί του άξονα της οδού και ξεκινά η διαδικασία αυτόματης επίλυσης των καθορισμένων περιορισμών για την εύρεση μιας αρχικής λύσης την οποία μπορεί να ορίσει ο χρήστης ή το πρότυπο και στη συνέχεια βελτιστοποιείται με στόχο το ελάχιστο κόστος εκσκαφών.

Όσον αφορά την κατακόρυφη χάραξη οι κορυφές της τοποθετούνται σε τυχαίες θέσεις σε διαστήματα τα οποία έχουν ορισθεί από τον χρήστη αλλά επειδή η κατακόρυφη χάραξη είναι σκόπιμο να βρίσκεται σε αρμονία με την οριζόντια το πρότυπο εισάγει μερικούς επιπλέον περιορισμούς. Δηλαδή οι κορυφές της μηκοτομής πρέπει να περιέχονται στις καμπύλες της οριζοντιογραφίας, ει δυνατόν η σύμπτωση των σημείων καμπής τους. Ενώ το υψόμετρό τους κυμαίνεται μεταξύ της διαφοράς του μέγιστου βάθους εκσκαφής και του μέγιστου ύψους επιχώματος αντίστοιχα από το υψόμετρο του εδάφους. Επιπλέον εφαρμόζεται έλεγχος μέγιστης και ελάχιστης κατά μήκος κλίσης, έλεγχος επικαλύψεων των κατακόρυφων καμπύλων από το πρότυπο και εξετάζεται η θλάση της πολυγωνικής χωρίς εφαρμογή κατακόρυφης καμπύλης όπου αυτή είναι εφικτή.

Όσον αφορά την βελτιστοποίηση με το λογισμικό Evolver ορίζονται οι γενετικές παράμετροι το μέγεθος του πληθυσμού, ο βαθμός διασταύρωσης και ο βαθμός μετάλλαξης. Το μέγεθος του πληθυσμού που προτείνεται από το λογισμικό είναι τριάντα έως εκατό επιλέχθηκε πενήντα, ο βαθμός διασταύρωσης από ένα χιλιοστό (0.01) έως ένα, επιλέχθηκε πέντε δέκατα (0.5) και ο μέγιστος βαθμός μετάλλαξης δύο δέκατα (0.2). Ο αρχικός πληθυσμός προκύπτει τυχαία με εφαρμογή των περιορισμών και η διαδικασία τερματίζεται όταν επιτυγχάνονται δέκα χιλιάδες (10000) δοκιμές. Το πλήθος αυτό επιλέχθηκε ώστε η μέθοδος να συγκλίνει. Ο χρόνος υπολογισμού που απαιτείται είναι τριάντα πέντε (35) με σαράντα (40) λεπτά για την οριζοντιογραφία και περίπου

δεκαπέντε (15) λεπτά για την μηκοτομή, εξαρτάται βέβαια από τα τεχνικά χαρακτηριστικά του υπολογιστικού συστήματος. Αναφέρεται ότι εκτός από τον καθορισμό των γενετικών παραμέτρων η αρχική χάραξη επηρεάζει εξαιρετικά το αποτέλεσμα καθώς τα χρωμοσώματα του πληθυσμού που παράχθηκε εξαρτώνται από αυτή. Μάλιστα με δοκιμές που έγιναν διαπιστώθηκε πως όταν ο αρχικός πληθυσμός είχε χαμηλή συνάρτηση στόχου η δοκιμή σύγκλινε ταχύτερα σε βέλτιστα αποτελέσματα. Επίσης αναφέρεται πως ιδιαίτερα σημαντικός είναι ο παράγοντας της ταχύτητας μελέτης καθώς η αύξησή της αυξάνει και τον όγκο των χωματισμών.

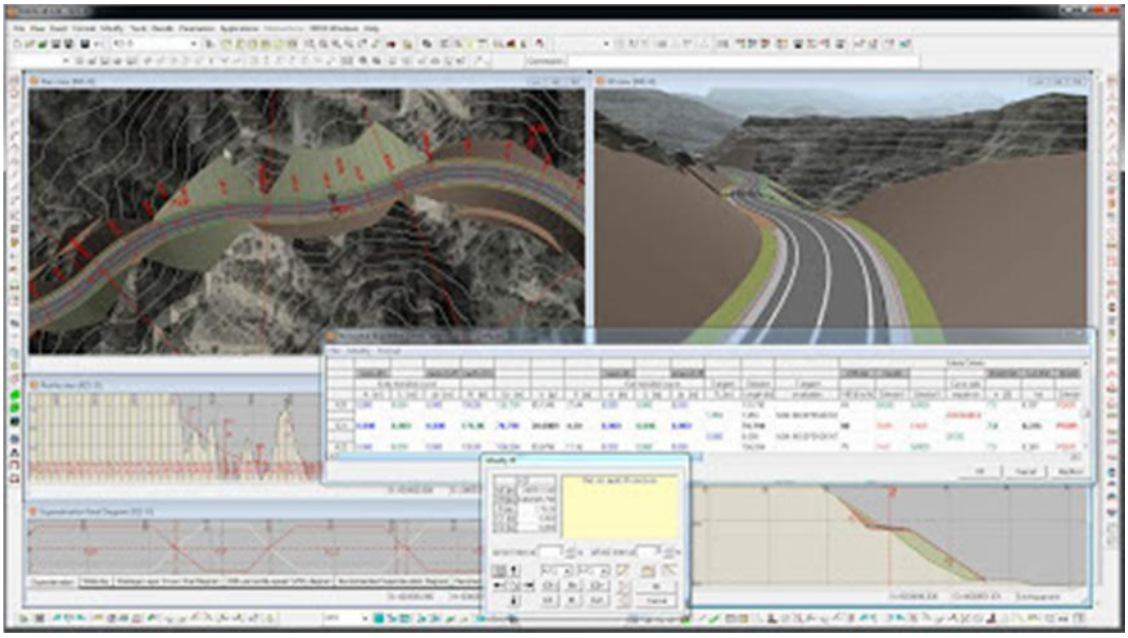
Ο ερευνητής συμπεραίνει πως αν και καμία μέθοδος δεν είναι καθολικά καλύτερη από τις υπόλοιπες οι γενετικοί αλγόριθμοι προσαρμόζονται αρκετά καλά στο πρόβλημα. Το πρότυπο που αναπτύχθηκε παράγει υψηλής ποιότητας αποτελέσματα σύμφωνα με τις παραδοχές και υποθέσεις που έγιναν.

2.2 Συνοπτική Παρουσίαση άλλων προγραμμάτων Οδοποιίας

Το ΟΔΟΣ 8, είναι η τελευταίας γενιάς έκδοση λογισμικού οδοποιίας της σειράς ΟΔΟΣ. Είναι κατ' εξοχήν πρόγραμμα σχεδιασμού οδικών έργων, αλλά μπορεί να χρησιμοποιηθεί και για την υποβοήθηση της μελέτης έργων «διαδρόμου» με την ευρύτερη έννοια (σιδηροδρομικών έργων, τροχοδρόμων αεροδρομίων, τάφρων, κλπ.).

Το πρόγραμμα, ΟΔΟΣ 8, διαβάζει απ' ευθείας οποιοδήποτε αρχείο DWG ως τοπογραφικό υπόβαθρο για τη μελέτη οδοποιίας και αξιοποιεί άμεσα την περιεχόμενη γεωμετρική πληροφορία του DWG (άμεση λήψη διατομών από τριδιάστατα στοιχεία γραμμών, πολυ-γραμμών, κλπ.). Ενώ βάσει των συντεταγμένων της περιοχής του υποβάθρου με τη χρήση του προγράμματος, Google Earth, καθίσταται δυνατή η αυτόματη συλλογή κατάλληλων αποσπασμάτων δορυφορικών εικόνων, γεωαναφορά και επίθεσή τους στο ψηφιακό υπόβαθρο της μελέτης. Παρέχεται επίσης η δυνατότητα ρύθμισης της εποπτείας των ατμοσφαιρικών συνθηκών στην περιοχή του έργου (καθαρότητα ατμόσφαιρας, θέση φωτισμού, κλπ.).

Ο αρχικός καθορισμός της χάραξης μιας οδού τόσο οριζοντιογραφικά όσο και μηκοτομικά, γίνεται μέσω της ελεύθερης γραφικής εισαγωγής πολυγωνικής. Ο λεπτομερής γεωμετρικός σχεδιασμός του άξονα, υλοποιείται μέσω των ειδικά σχεδιασμένων εντολών του ΟΔΟΣ 8, με απόλυτο έλεγχο της επηρεαζόμενης γεωμετρίας από το χρήστη και με δυνατότητες γεωμετρικών δεσμεύσεων (π.χ. Μεταβολή κυκλικού τόξου, έτσι ώστε να διέρχεται από δύο σημεία, με δεδομένη ακτίνα, διατήρηση της θέσης και του τόξου συναρμογής εξόδου της προηγούμενης καμπύλης και εφαπτομενική σύνδεση με την επόμενη κορυφή πολυγωνικής, «ολίσθηση» κλάδου επί καμπύλης, αυτόματη κατασκευή "S" , κλπ.).



Εικόνα 1: Λογισμικό ΟΔΟΣ 8 Πηγή: http://sxediase.blogspot.com

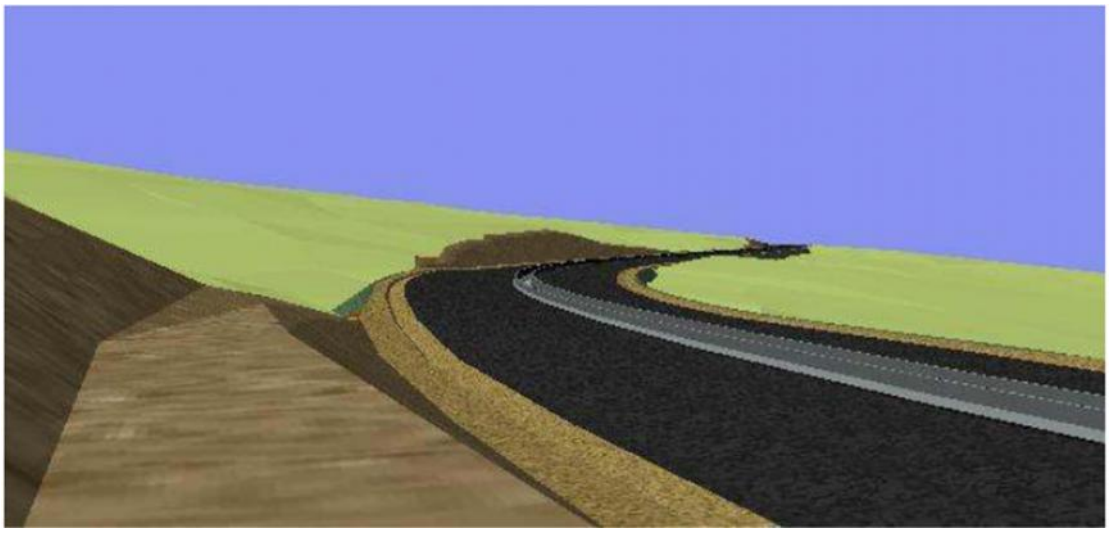
Στη συνέχεια ο σχεδιασμός των διατομών γίνεται εντελώς αυτόματα από το ΟΔΟΣ 8. Οι θέσεις των διατομών (πάσσαλοι) μπορούν να εισαχθούν εναλλακτικά με πολλούς τρόπους. Αυτόματα, στις χαρακτηριστικές θέσεις (Α, Ω, Δ, Ω', Α', κλπ.) των καμπυλών, σε επιλεγμένες θέσεις γραφικά από το χρήστη, ομαδικά σε επιλεγμένες περιοχές με βήμα, με ανάγνωση σειριακών αρχείων (π.χ. αρχείων τύπου GRD των ΠΨΥΜΟ), κλπ.

Οι υπολογισμοί των στοιχείων μελέτης κάθε οδού βασίζονται στη λειτουργική κατάταξή της, το μοντέλο V85 της Ελλάδας, την οριζοντιογραφική και υψομετρική χάραξή της και τις διαστάσεις του τυπικού οδοστρώματος.Το ΟΔΟΣ 8 έχει αφομοιώσει όλους τους προβλεπόμενους ελέγχους των ανωτέρω κανονισμών, στο μέγιστο βαθμό λεπτομέρειας. Όπως στοιχεία των ΟΜΟΕ-ΛΚΟΔ και ΟΜΟΕ-Δ σχετικά με τη λειτουργική κατάταξη και την εφαρμοστέα τυπική διατομή, των ΟΜΟΕ-ΚΑΟ για το σχεδιασμό αστικών οδών, καθώς και τις γερμανικές οδηγίες RAA 2008 για τον σχεδιασμό αυτοκινητοδρόμων και συνδετηρίων κλάδων ανισόπεδων κόμβων. Οι ΟΜΟΕ-Χ δίνουν έμφαση στην «ποσοτικοποίηση» της ποιότητας του σχεδιασμού από πλευράς ασφάλειας.

Επιπρόσθετα στα πλαίσια των ελέγχων του γεωμετρικού σχεδιασμού, το ΟΔΟΣ 8 παρέχει τη δυνατότητα εποπτικού ελέγχου κατά μήκος της οδού. Η προσομοίωση οδήγησης χρησιμοποιείται ως το κύριο εργαλείο του μελετητή στον ακριβή έλεγχο διάθεσης των απαιτουμένων μηκών ορατότητας για στάση και για προσπέραση αλλά και απορροής ομβρίων.

Τέλος πρέπει να σημειωθεί ότι μέσω της τεχνολογίας OpenDesign που ενσωματώνεται στο ΟΔΟΣ 8, η παραγωγή όλων των σχεδίων της μελέτης, γίνεται αυτόνομα σε αρχεία DWG. Τα παραγόμενα από το ΟΔΟΣ σχέδια είναι πλήρη, όσον αφορά τη γραμμογραφία και τους συμβολισμούς, τα υπομνήματα κειμένου και τη δομή τους.

Το **Diolkos** είναι ολοκληρωμένο και αυτόνομο λογισμικό σχεδιασμού οδών. Καλύπτει όλες τις περιπτώσεις οδοποιίας όπως αυτοκινητόδρομοι, επαρχιακές και αστικές οδοί, δασικές οδοί κλπ. Επίσης αντιμετωπίζει σύνθετα οδικά έργα όπως κόμβοι, παράλληλο δίκτυο service road κλπ. Επιπρόσθετα διαθέτει όλα τα απαραίτητα εργαλεία για την εκτέλεση επιμετρήσεων κατά την φάση κατασκευής των έργων. Η εταιρία Diolkos3d παρέχει μια σειρά βοηθητικών και συμπληρωματικών προγραμμάτων του Diolkos όπως το WaterNet CAD (για τη χρήση σε μελέτες υδραυλικών δικτύων), το FastTerrain, (για τη δημιουργία και επεξεργασία ψηφιακών μοντέλων εδάφους και υπολογισμού εκσκαφών/επιχώσεων) και το Gaies (για την βελτιστοποίηση της κίνησης γαιών σε οδικά έργα) .

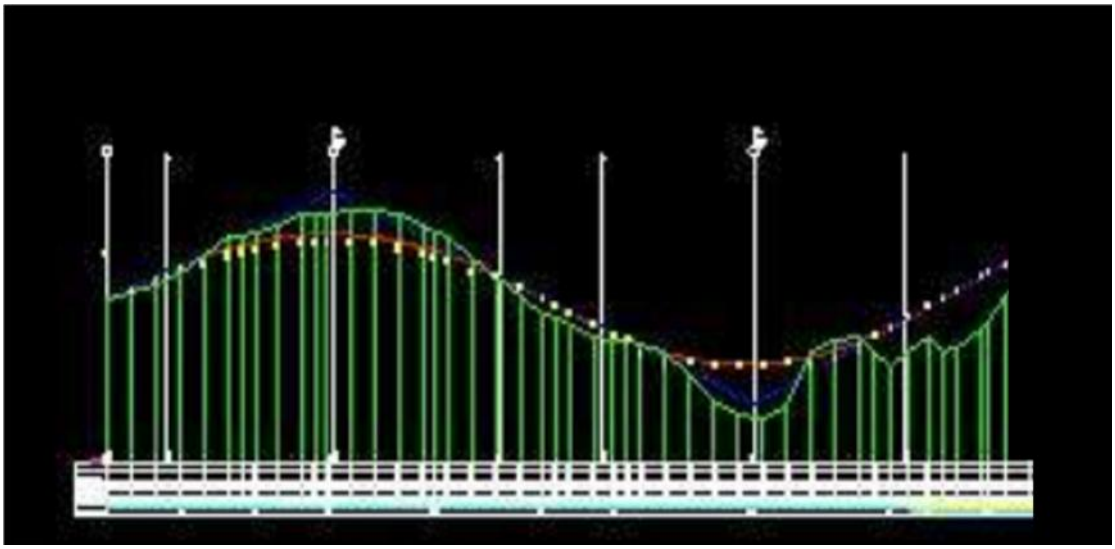


Εικόνα 2: Άποψη τριών διαστάσεων (3d) Πηγή: Diolkos Manual

Στο πρόγραμμα **DIOLKOS** γίνεται εισαγωγή της πολυγωνικής της χάραξης από το AutoCAD. Ο χρήστης μπορεί να χειρίζεται γραφικά στην οθόνη τις κορυφές της χάραξης (εισαγωγή, μετακίνηση, διαγραφή). Μπορεί επίσης να οριστεί οποιαδήποτε χιλιομετρική θέση ως αρχή της χάραξης. Παράλληλα ο χρήστης μπορεί να εισάγει το υπόβαθρο της μελέτης και να το οργανώσει σε σχεδιαστικά επίπεδα (layers).Οι πάσσαλοι εισάγονται είτε αυτόματα από το πρόγραμμα κατά μήκος της οδού ή χειροκίνητα από τον χρήστη. Ο χρήστης μπορεί να εισάγει ή να διαγράψει πασσάλους με το πληκτρολόγιο ή γραφικά στην οθόνη με το ποντίκι. Προβλέπεται αυτόματη εισαγωγή πασσάλων κατά μήκος του άξονα ανά συγκεκριμένη απόσταση και στις χαρακτηριστικές θέσεις (Α,

Ω, Δ, Ω’ Α’). Μπορούν να εισαχθούν μαζικά πάσσαλοι στις τομές του άξονα της χάραξης με το TIN. Επίσης μπορεί να γίνει μαζική εισαγωγή πασσάλων στις τομές μεταξύ του άξονα και των γραμμών υποβάθρου κάποιου δεδομένου layer ή να γίνει προβολή των πασσάλων άλλου οδικού άξονα. Με τους τρόπους αυτούς μπορούν να εισαχθούν αυτόματα όλοι οι απαιτούμενοι πάσσαλοι της μελέτης, αλλά και να επανεισάγονται αυτόματα σε κάθε αλλαγή της χάραξης.

Η ανάγνωση του εδάφους γίνεται αυτόματα από το TIN που προηγουμένως έχουμε εισάγει στο πρόγραμμα. **Ο χρήστης μπορεί να εισάγει γραφικά επί της οθόνης την πολυγωνική της μηκοτομής και να τροποποιεί τα δεδομένα των κορυφών (Χ.Θ. R, Ηερ) εύκολα και εποπτικά.** Η βάση του DIOLKOS είναι ο τρόπος οργάνωσης και ανάγνωσης του ανάγλυφου του εδάφους. Το DIOLKOS «διαβάζει» το έδαφος από το TIN που έχει αποθηκευμένο στην βάση δεδομένων του, το οποίο αποτελείται από τρίγωνα που σχηματίζουν ένα δίκτυο ακανόνιστων τριγώνων. Εναλλακτικά

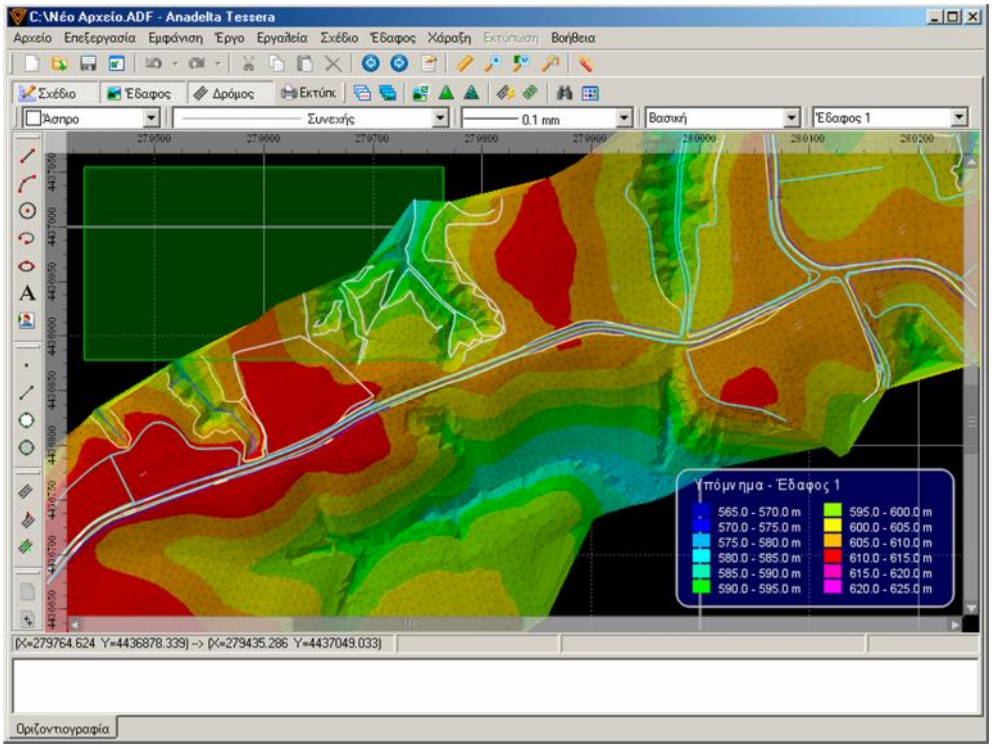


Εικόνα 3: Μηκοτομή Πηγή: Diolkos Manual

το DIOLKOS μπορεί να διαβάσει το ανάγλυφο του εδάφους από ένα σύνολο τρισδιάστατων γραμμών που περιγράφουν ικανοποιητικά το ανάγλυφο (π.χ από ισοϋψείς). Σε κάποιες περιπτώσεις μελέτης έργων, αυτός ο τρόπος είναι πιο πρόσφορος σε σχέση με το κλασικό TIN που αποτελείται από ακανόνιστα τρίγωνα. Όπως και με τις κορυφές της οριζοντιογραφίας, ο χρήστης μπορεί να χειρίζεται γραφικά επί της οθόνης τις σημαίες της μηκοτομής (εισαγωγή, τροποποίηση, διαγραφή). Επίσης μπορεί να εισάγει και να διαχειριστεί βοηθητικά σημεία ελέγχου της μηκοτομής Ενώ σε κάθε τροποποίηση των σημαιών, ο χρήστης βλέπει ταυτόχρονα στην οθόνη τις ενημερωμένες τιμές της κλίσης και της υψομετρικής διαφοράς με το έδαφος σε κάθε πάσσαλο.

Το **Anadelta Tessera** είναι ένα νέο Λογισμικό Οδοποιίας που αναπτύχθηκε από την εταιρία Anadelta Software. Οι αρχικοί στόχοι που η εταιρία έθεσε ήταν η ανάπτυξη σύγχρονων αλγόριθμων σχεδιασμού, η ευελιξία και η προσαρμοστικότητα στις ανάγκες του μελετητή, μέσα από ένα νέο ειδικό και ολοκληρωμένο γραφικό περιβάλλον. Το αποτέλεσμα είναι ένα αυτόνομο και ενιαίο λογισμικό κατασκευασμένο ειδικά για οδοποιία. Στο Tessera μαζί με τις λειτουργίες σχεδιασμού οδοποιίας συνυπάρχουν όλες οι απαραίτητες σχεδιαστικές δυνατότητες καθώς και ένα πλήρες περιβάλλον διαχείρισης του ψηφιακού μοντέλου εδάφους. Το συνεχώς εξελισσόμενο αυτό λογισμικό συνοδεύεται από υψηλής ποιότητας υποστήριξη και απευθύνεται σε μελετητές και κατασκευαστές οδικών έργων.

Η ύπαρξη του σχεδιαστικού περιβάλλοντος είναι κυρίως βοηθός στο περιβάλλον της Χάραξης, καθώς το εμπλουτίζει με δυνατότητες που ξεπερνούν τις αναμενόμενες από ένα πρόγραμμα Οδοποιίας. Επιτρέπεται στο μελετητή να έχει πλήρη έλεγχο του υποβάθρου πάνω στο οποίο θα κάνει τη νέα χάραξη ή τη βελτίωση. Ενώ παράλληλα παρέχεται πληθώρα σχεδιαστικών αντικειμένων: polylines, κύκλοι, ελλείψεις, κλωθοειδείς, παραβολές, truetype κείμενα, εικόνες και υποστηρίζονται κλασσικές λειτουργίες Cad όπως linetypes (.lin), hatches (.pat), Copy/Paste, Undo/Redo, περιστροφή, μεγέθυνση, snap σε αντικείμενα ή κάνναβο κ.α.. Τα αντικείμενα ταξινομούνται σε ενότητες (layers) και παίρνουν τις ιδιότητές τους από αυτές.



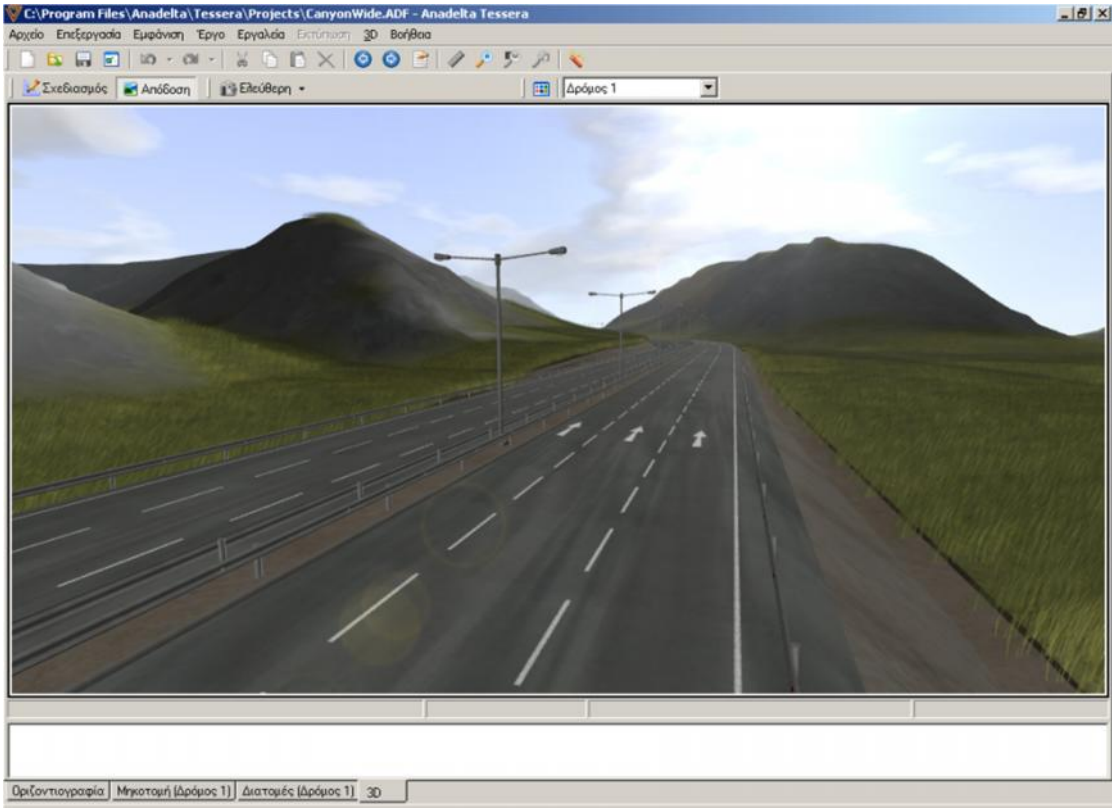
Εικόνα 4: Υπόμνημα εδάφους Πηγή: anadelta.com

Η εισαγωγή των στοιχείων του εδάφους εισάγονται από αρχεία ASCII ή από αρχεία Dxf , ενώ της πολυγωνικής της οριζόντιας χάραξης γραφικά με το ποντίκι, με ανάγνωση αρχείου ASCII ή με

πληκτρολόγηση σε φύλλο εργασίας του προγράμματος. Το πρόγραμμα προτείνει αρχικές τιμές για τα στοιχεία συναρμογής των κορυφών και προβάλλονται συνεχώς τα γεωμετρικά χαρακτηριστικά της κάθε καμπύλης. Έπειτα πραγματοποιείται έλεγχος της χάραξης και διόρθωση όπου απαιτείται. Οι διατομές τοποθετούνται αυτόματα από το πρόγραμμα με βάση τις ιδιότητες του δρόμου.

Μετά τον ορισμό των θέσεων των διατομών στην οριζόντια χάραξη, προκύπτει η μηκοτομή του εδάφους. Οποιαδήποτε μεταβολή της οριζόντιας χάραξης αντιστοιχεί σε μια νέα μηκοτομή εδάφους. **Η εισαγωγή της πολυγωνικής της κατακόρυφης χάραξης γίνεται με το ποντίκι, με πίνακες ή με εισαγωγή από αρχεία κειμένου**. Το πρόγραμμα υπολογίζει την ερυθρά από την πολυγωνική και τις ακτίνες καμπυλότητας των κορυφών της (προτείνονται κατάλληλες τιμές που προκύπτουν από τον έλεγχο του απαιτούμενου μήκους ορατότητας) ενώ διορθώνονται τυχόν λάθη.

Ανάλογα με τον τύπο του δρόμου γίνεται η επιλογή των τυπικών διατομών που θα χρησιμοποιηθούν. Υπάρχει εκτενής βιβλιοθήκη τυπικών διατομών που μπορεί να εμπλουτιστεί μέσω internet ή e-mail. Ο ενιαίος χαρακτήρας του προγράμματος επιτρέπει επίσης την άμεση διόρθωση του μοντέλου όταν στο επίπεδο της διατομής κριθεί ότι κάτι τέτοιο είναι απαραίτητο να γίνει.



Εικόνα 5: Άποψη τριών διαστάσεων (3d) Πηγή: Anadelta Tessera Manual

Ο πίνακας χωματισμών παραμετροποιείται πλήρως. Τη στιγμή του υπολογισμού - εκτύπωσης ο χρήστης μπορεί να επιλέξει ποιες από τις υπολογισμένες ποσότητες θα εμφανιστούν, με ποια σειρά και με ποια μορφή. Το Tessera καλύπτει πλήρως και τις ανάγκες της επιμέτρησης.

2.3 Σύνοψη - Κριτική αξιολόγηση

Στο κεφάλαιο αυτό παρουσιάστηκαν τα αποτελέσματα της βιβλιογραφικής ανασκόπησης που πραγματοποιήθηκε για τις ανάγκες της Διπλωματικής Εργασίας, αναφορικά με την χάραξη και τη βελτιστοποίηση της ερυθράς της οδού.

Αρχικά στο παρόν κεφάλαιο παρουσιάσθηκαν έρευνες που αντιμετώπισαν το ζήτημα. Σχετικά με όλες τις μορφές βελτιστοποίησης που αναφέρθηκαν οι γενετικοί αλγόριθμοι προσομοιάζουν και εκτελούν με μεγαλύτερη επιτυχία τη διαδικασία της κατακόρυφης χάραξης του οδικού άξονα. Καθώς επιτυγχάνουν ομαλή και ρεαλιστική χάραξη ενώ ελαχιστοποιούν την συνάρτηση στόχου καθώς αναζητούν λύση προς πολλές κατευθύνσεις διατηρώντας παράλληλα πιθανές λύσεις. Προωθούν δε την διαμόρφωση και ανταλλαγή πληροφοριών μεταξύ αυτών των κατευθύνσεων.

Στη συνέχεια έγινε μια συνοπτική παρουσίαση των προγραμμάτων οδοποιίας ευρείας χρήσης στη χώρα μας, των ΟΔΟΣ 8, Anadelta Tessera και Diolkos. Τα λογισμικά που αναφέρθηκαν είναι ιδιαίτερα χρήσιμα αλλά και εύχρηστα διότι προσομοιώνουν το έδαφος, γίνεται εύκολη εισαγωγή στοιχείων και επεξεργασία αποτελεσμάτων, οπτικός έλεγχος από σχέδια και τρισδιάστατα περιβάλλοντα αλλά και εκτύπωση σχεδίων από αρχεία τύπου CAD. Ακολουθούν δε την εξέλιξη της τεχνολογίας χρησιμοποιώντας την για να μειώσουν λάθη στη μελέτη που παλιότερα υποβάθμιζαν την ποιότητα και ασφάλεια της οδού. Η χάραξη της ερυθράς γίνεται γραφικά ή από αρχεία και δεν περιλαμβάνουν κάποια αυτοματοποιημένη διαδικασία χάραξης της. Σε κανένα από αυτά δεν γίνεται βελτιστοποίηση της.

3 Θεωρητικό Υπόβαθρο

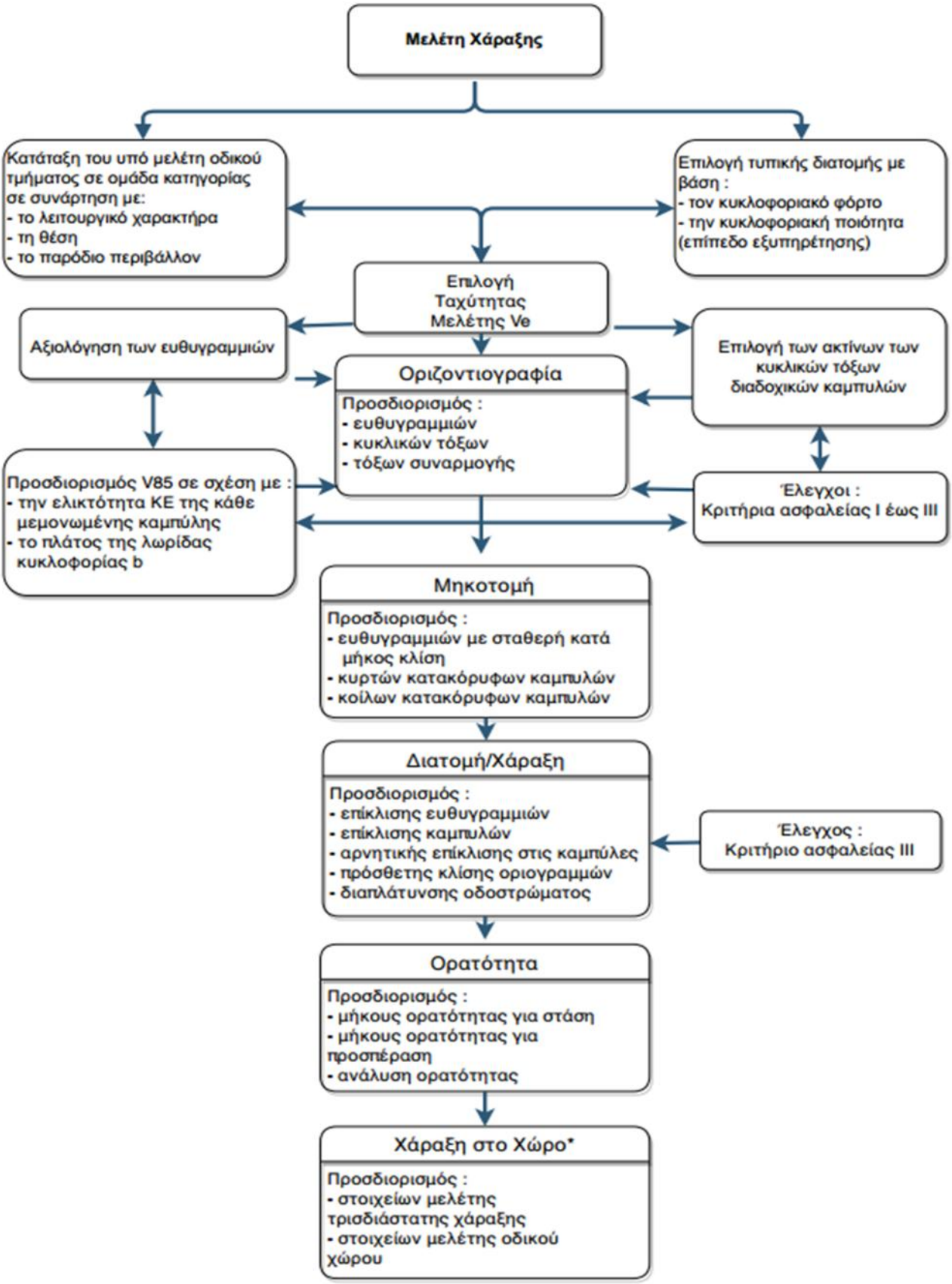
3.1 Οδοποιία

Το οδικό δίκτυο αποτελεί μέρος της συνολικής συγκοινωνιακής υποδομής και ταυτόχρονα το βασικότερο μέσο πρόσβασης. Ένα σύγχρονο οδικό δίκτυο σχεδιάζεται ακολουθώντας μια ολοκληρωμένη διαδικασία που αναπτύσσεται σταδιακά στο πλαίσιο της ιεράρχησης των αναγκών της χώρας, με βάση ένα διευρυμένο κατάλογο στόχων, που εκτός από τους παραδοσιακούς στόχους της ασφαλούς κυκλοφορίας και τις ανάγκες κυκλοφοριακής ροής, εξασφαλίζει τη διαφύλαξη της ποιότητας του φυσικού αλλά και ανθρωπογενούς περιβάλλοντος και μεριμνά για την εξοικονόμηση των διαθέσιμων φυσικών πόρων.

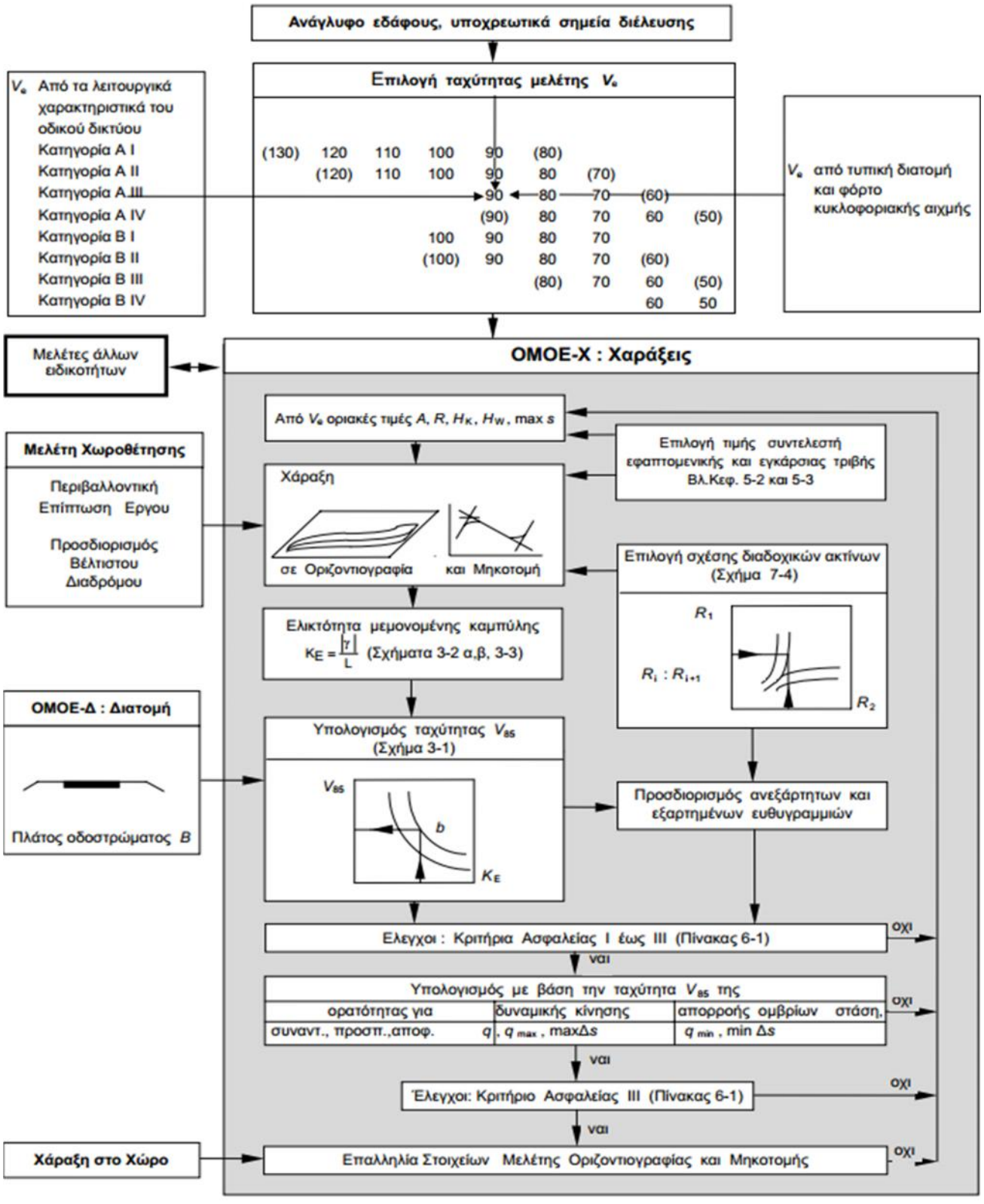
Ο σχεδιασμός ενός οδικού έργου προσεγγίζεται κατά κανόνα με γεωμετρική μέθοδο, σύμφωνα με την οποία η χάραξη μιας οδού πραγματοποιείται σε δύο διακριτά στάδια. Αρχικά γίνεται η σχεδίαση του άξονα της οδού στο οριζόντιο επίπεδο, η οποία ονομάζεται οριζοντιογραφία της οδού. Στη συνέχεια πραγματοποιείται η υψομετρική διαμόρφωση του άξονα της οδού. Η υψομετρική διαμόρφωση γίνεται επί του αναπτύγματος της κατακόρυφης επιφάνειας με οδηγό καμπύλη τον άξονα της οδού κατά την οριζοντιογραφία και καλείται μηκοτομή της οδού. Ο άξονας της οδού κατά τη μηκοτομή ονομάζεται ερυθρά της οδού ενώ η προβολή της επιφάνειας του εδάφους ονομάζεται γραμμή εδάφους. Το τελικό αποτέλεσμα της χάραξης στον χώρο προκύπτει από τη σύνθεση των επιμέρους δύο σταδίων.

3.1.1 Μελέτη Χάραξης

Κατά την μελέτη χάραξης, όπως αυτή περιγράφεται στους αντίστοιχους κανονισμούς, αρχικά η οδός κατατάσσεται σύμφωνα με το λειτουργικό χαρακτήρα, τη θέση και το παρόδιο περιβάλλον της (ΟΜΟΕ – ΛΚΟΔ) και πραγματοποιείται η κυκλοφοριακή μελέτη της οδού ορίζεται δηλαδή το επίπεδο εξυπηρέτησης της και μετριέται ο κυκλοφοριακός φόρτος. Στη συνέχεια ορίζεται η ταχύτητα μελέτης, σύμφωνα με αυτή υπολογίζονται τα περισσότερα γεωμετρικά στοιχεία της οδού και η λειτουργική ταχύτητα. Έπειτα εκτελούνται οι απαραίτητοι έλεγχοι, όταν αυτοί ικανοποιούνται χαράσσονται η οριζοντιογραφία, η μηκοτομή και οι διατομές της οδού και ελέγχεται η ορατότητά της. Τα διακριτά στάδια της μελέτης χάραξης καθώς και οι σχέσεις αλληλεπίδρασής τους περιγράφονται στη συνέχεια .



Σχήμα 1: Διάγραμμα ροής για τη μελέτη χάραξης Πηγή : ΟΜΟΕ - Χ



Σχήμα 2: Μεθοδολογία της πορείας της μελέτης χάραξης μίας οδού με στοιχεία μελέτης εναρμονισμένα μεταξύ τους. Πηγή: ΟΜΟΕ – Χ

Χαρακτηριστικές Ταχύτητες

Οι τιμές των παραμέτρων που χρησιμοποιούνται κατά την χάραξη της μελέτης ενός οδικού έργου, προκύπτουν από την αλληλεπίδραση της χάραξης της οδού, της οδικής συμπεριφοράς των οδηγών, της δυναμικής κίνησης των οχημάτων και τα γεωμετρικά χαρακτηριστικά της τροχιάς τους. Οι χαρακτηριστικές ταχύτητες της οδού αποτελούν καθοριστικό στοιχείο για τη χάραξη ενός οδικού έργου. Επηρεάζουν ποικίλους παράγοντες όπως οικονομικούς, τεχνικούς, κοινωνικούς και περιβαλλοντικούς. Επιπλέον καθορίζουν τα γεωμετρικά χαρακτηριστικά της οδού.

Η **επιτρεπόμενη ταχύτητα $V_{επιτρ}$** είναι το τοπικό ή σχετικά ισχύον μέγιστο όριο ταχύτητας. Η επιτρεπόμενη ταχύτητα διαφέρει ανάλογα με την κατηγορία της οδού. Η ταχύτητα αυτή θεωρείται καθοριστική, διότι η ταχύτητα μελέτης σχετίζεται με τα στοιχεία μελέτης της οδού και πρέπει να είναι μεγαλύτερη ή ίση με την επιτρεπόμενη ταχύτητα.

Η **ταχύτητα μελέτης V_e** αποτελεί καθοριστικό στοιχείο για τη χάραξη ενός οδικού έργου. Ορίζεται λαμβάνοντας υπόψη περιβαλλοντικά και οικονομικά κριτήρια, που ανταποκρίνονται στον προβλεπόμενο λειτουργικό χαρακτήρα της οδού στο οδικό δίκτυο και την επιδιωκόμενη ποιότητα κυκλοφοριακής ροής, όπως την αύξηση του όγκου των εργασιών και του χρόνου κατασκευής αλλά και του επίπεδου εξυπηρέτησης και της οδικής ασφάλειας. Προσδιορίζει δε, τα περισσότερα γεωμετρικά μεγέθη της οδού, τις ελάχιστες ακτίνες των κατακόρυφων και οριζόντιων καμπυλών, τις μέγιστες κατά μήκος κλίσεις και τις παραμέτρους των κλωθοειδών. Η τιμή της είναι καθοριστική και γι αυτό είναι σκόπιμο να παραμένει σταθερή κατά το δυνατόν, ειδικά σε τμήματα της οδού που αλληλοεξαρτούνται.

Η **λειτουργική ταχύτητα V_{85}** χρησιμοποιείται στο γεωμετρικό υπολογισμό μεμονωμένων στοιχείων μελέτης της οριζοντιογραφίας, της μηκοτομής και της διατομής. Επίσης έχει άμεση σχέση με τη δυναμική της κίνησης των οχημάτων. Προσδιορίζεται είτε για κάθε μεμονωμένο γεωμετρικό στοιχείο της οδού, καμπύλη ή ευθυγραμμία, είτε για οδικό τμήμα με ενιαία χαρακτηριστικά. Στις οδούς κατηγορίας Α αντιστοιχεί στην ταχύτητα με την οποία θα κινηθεί ανεμπόδιστα το 85% των επιβατηγών οχημάτων σε καθαρό και υγρό οδόστρωμα, ενώ στις Β στην μέγιστη επιτρεπόμενη ταχύτητα. Οι καθοριστικοί παράγοντες που επηρεάζουν την τιμή της ταχύτητας V_{85}, είναι η ελικτότητα της μεμονωμένης καμπύλης και το πλάτος της λωρίδας κυκλοφορίας (b). Η λειτουργική ταχύτητα V_{85} καθορίζει τα απαιτούμενα μήκη ορατότητας για στάση και προσπέραση, τις επικλίσεις στις οριζόντιες καμπύλες, τις ελάχιστες οριζόντιες ακτίνες σε περίπτωση αρνητικής επίκλισης, και τα στοιχεία απορροής ομβρίων. Παράλληλα η μεταβολή της τιμής της έχει αντίκτυπο στο κόστος και την ασφάλεια της οδού καθώς και το παρόδιο περιβάλλον.

Κριτήρια Ασφαλείας

Όπως αναφέρθηκε παραπάνω η ταχύτητα μελέτης της οδού είναι σκόπιμο να διατηρείται σταθερή κατά το δυνατόν είτε να μεταβάλλεται σταδιακά και να είναι εναρμονισμένη με την ταχύτητα λειτουργίας. Γι' αυτό ορίστηκαν τα κριτήρια ασφαλείας που αναφέρονται στο παραπάνω διάγραμμα ως ακολούθως.

Το **Κριτήριο Ασφαλείας Ι** διασφαλίζει την αρμονία και τη συνέχεια στην οδό, καθώς συνδέει τη ταχύτητα μελέτης με τη λειτουργική ταχύτητα ώστε εξασφαλίζεται συμβατότητα της οδικής συμπεριφοράς και των γεωμετρικών χαρακτηριστικών. Το κριτήριο ασφαλείας Ι αναφέρεται σε ένα στοιχείο της χάραξης δηλαδή σε μια καμπύλη ή σε μια ανεξάρτητη ευθυγραμμία.

Το **Κριτήριο Ασφαλείας ΙΙ** εξασφαλίζει την αρμονία και συνέχεια της λειτουργικής ταχύτητας κατά μήκος της οδού. Σε οδούς μεγάλου μήκους τα γεωμετρικά χαρακτηριστικά της χάραξης και άρα και της ταχύτητας συχνά διαφοροποιούνται. Επομένως πρέπει να δοθεί ιδιαίτερη προσοχή στην επιλογή των στοιχείων μελέτης, ώστε να μεταβάλλονται σταδιακά. Το κριτήριο ορίζει ικανή σχέση συναρμογής στην $V_{8\ i}$ και την $V_{8\ i+1}$

Το **Κριτήριο Ασφαλείας ΙΙΙ** ελέγχει την σχέση μεταξύ του τροχού και του οδοστρώματος σε συνάρτηση με την ταχύτητα του οχήματος. Υπάρχουν πολλές και ουσιαστικές διαφορές ως προς την επιλογή των επιτρεπόμενων τιμών του συντελεστή της εφαπτομενικής και της εγκάρσιας συνιστώσας της τριβής. Οι διαφορές αυτές έχουν με τη σειρά τους καθοριστική επίδραση στον υπολογισμό των ελαχίστων ακτινών των οριζόντιων καμπυλών, των απαιτουμένων μηκών ορατότητας, των ελάχιστων ακτινών των κατακόρυφων καμπυλών κλπ. σε αλληλεξάρτηση με τις εκάστοτε ισχύουσες ταχύτητες μελέτης V_e, V_{85}. Αξίζει να αναφερθεί πως μη ικανοποίηση του κριτηρίου οδηγεί σε ιδιαίτερα αντιοικονομικές κατασκευές με ανεπιθύμητες επιπτώσεις στο περιβάλλον.

Οριζοντιογραφία

Οριζοντιογραφία καλείται το σχέδιο που προκύπτει από την αναπαράσταση του άξονα της οδού στο προβολικό επίπεδο. Αποτελείται από αλληλουχία ευθύγραμμων και καμπύλων τμημάτων. Στοιχεία της οριζοντιογραφίας αποτελούν οι ευθυγραμμίες, τα κυκλικά τόξα και τα τόξα συναρμογής τα οποία παρεμβάλλονται μεταξύ ευθυγραμμιών και κυκλικών τόξων για την ομαλή μετάβαση σε αυτά. Για την χάραξη της οριζοντιογραφίας σχεδιάζεται πρώτα η ισοκλινής της οδού, μια τεθλασμένη γραμμή που διατηρεί σταθερή κλίση. Από αυτή καθορίζεται η πολυγωνική της χάραξης, οι πλευρές της οποίας αποτελούν τις ευθυγραμμίες της οδού και στις κορυφές της εγγράφονται οι καμπύλες της. Στόχος της παραπάνω διαδικασίας είναι όταν εγγραφούν οι καμπύλες ο άξονας της οδού να πλησιάζει ει δυνατό την ισοκλινή. Επιπλέον μετά τη χάραξη της οριζοντιογραφίας υπολογίζεται η

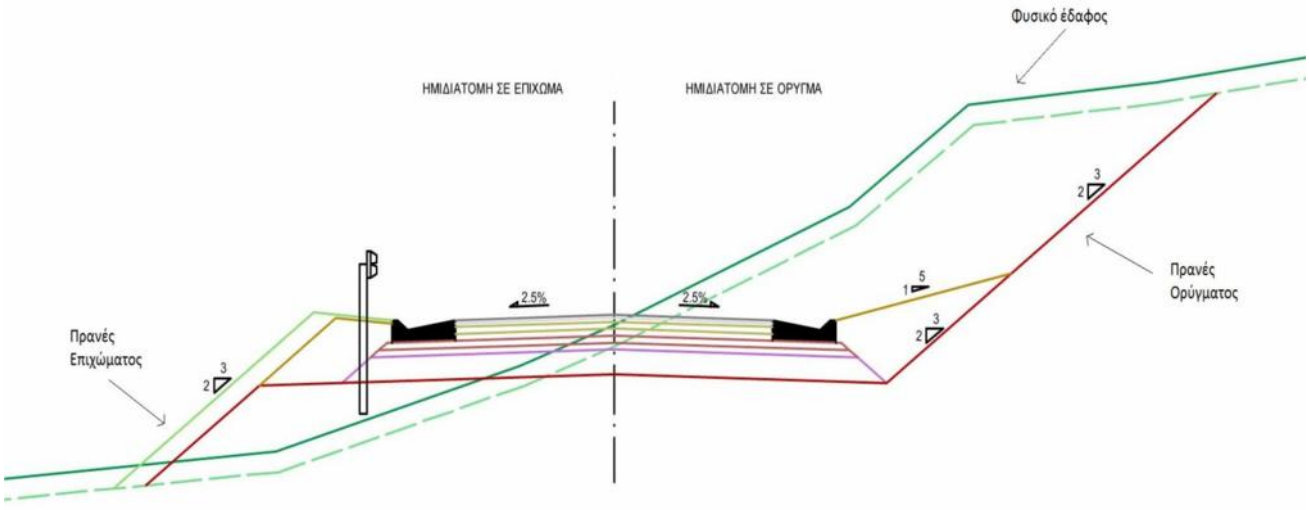
ελικτότητα της κάθε καμπύλης ως ο λόγος των απόλυτων τιμών των αλλαγών διεύθυνσης του άξονα προς το μήκος του.

Μηκοτομή

Η μηκοτομή της οδού είναι το ανάπτυγμα της τομής της οδού και του εδάφους με την κυλινδρική επιφάνεια που έχει οδηγό καμπύλη τον άξονα της οδού και γενέτειρες κατακόρυφες ευθείες. Η απεικόνιση του εδάφους στη μηκοτομή ονομάζεται γραμμή εδάφους ενώ η απεικόνιση του άξονα της οδού, στη φάση που έχει κατασκευασθεί η τελική επιφάνεια κύλισης, ονομάζεται ερυθρά γραμμή. Στα αντίστοιχα σχέδια, η μηκοτομή της οδού παρίσταται ως επίπεδη γραμμή σε ορθογώνιο σύστημα αξόνων όπου οι τετμημένες είναι οι χιλιομετρικές θέσεις των σημείων του άξονα, όπως προκύπτουν από την οριζοντιογραφία, και οι τεταγμένες τα υψόμετρά τους. Για την χάραξη της μηκοτομής σχεδιάζονται ευθυγραμμίες με σταθερή κατά μήκος κλίση. Τα σημεία θλάσης της ερυθράς ομαλοποιούνται με τη χρήση τετραγωνικής παραβολής ως προσέγγιση του εγγύτατου κύκλου της καμπύλης.

Διατομές

Η διατομή της οδού προκύπτει εάν κάνουμε τομή κάθετα προς τον άξονα της οδού και προβάλλουμε το αποτέλεσμα στο κατακόρυφο επίπεδο. Η μορφή της διατομής διακρίνεται σε διατομή σε όρυγμα, επίχωμα και μικτή. Κατά τη σχεδίαση της διατομής ιδιαίτερη μέριμνα πρέπει να δίνεται στην επικλίση , στην εγκάρσια κλίση, του οδοστρώματος για την καλή απορροή των ομβρίων και καλή δυναμική της κίνησης.



Εικόνα 7: Τυπική διατομή οδού. Πηγή: TD2 – www.topografos .me

Ορατότητα

Κατά τη μελέτη της χάραξης εξετάζεται και η ορατότητα της οδού που είναι απαραίτητη για την ασφάλεια της κυκλοφορίας και της ποιότητας της κυκλοφοριακής ροής. Μήκος ορατότητας ορίζεται το τμήμα της οδού που εκτίθεται στο οπτικό πεδίο του οδηγού σε κάθε χρονική στιγμή και απαιτείται ώστε οδηγός οχήματος κινούμενου με ταχύτητα V_{85} να λάβει μια πληροφορία και να αντιδράσει σε αυτήν. Διακρίνεται σε μήκος ορατότητας για στάση, η απαιτούμενη απόσταση για ακινητοποίηση του οχήματος στη θέαση ενός εμποδίου, μήκος ορατότητας για προσπέραση, που είναι η απαιτούμενη απόσταση για την ασφαλή προσπέραση προπορευόμενου οχήματος , μήκος ορατότητας για συνάντηση, ώστε δύο αντίθετα κινούμενα οχήματα να ακινητοποιηθούν έγκαιρα.

3.1.2 Αρχές Χάραξης Μηκοτομής

Κατά μήκος κλίσεις

Οι μικρές κατά μήκος κλίσεις:

- βελτιώνουν την κυκλοφοριακή ασφάλεια (εξαίρεση: περιοχές προσαρμογής των επικλίσεων με στροφή του οδοστρώματος γύρω από τον άξονα περιστροφής, όπου η επίκλιση διέρχεται από την οριζόντια θέση)
- βελτιώνουν την ποιότητα της κυκλοφοριακής ροής
- μειώνουν τις δαπάνες λειτουργίας, καθώς και τις δαπάνες για τους χρήστες της οδού
- μειώνουν τις εκπομπές.

Αντίθετα, με τις μεγαλύτερες κατά μήκος κλίσεις:

- βελτιώνεται η προσαρμογή στο φυσικό τοπίο
- μειώνονται οι παρεμβάσεις στο φυσικό και αστικό περιβάλλον
- μειώνεται το κόστος κατασκευής.

Αν δεν είναι δυνατόν να αναπτυχθούν οι αναμενόμενες λειτουργικές ταχύτητες σε ορισμένα ανωφερικά τμήματα στις οδούς της ομάδας Α, πρέπει να διερευνάται η πιθανότητα διάταξης μίας επιπλέον λωρίδας για τα βαρέα οχήματα (πρόσθετη λωρίδα) ή αλλαγής της χάραξης της οδό.

Στρογγύλευση των κορυφών της μηκοτομής

Στις κορυφές της μηκοτομής χαράσσονται τόξα συναρμογής για να εξομαλυνθεί η διαφορά κλίσης μεταξύ δυο κλάδων της μηκοτομής. Συνήθως τα τόξα συναρμογής είναι τετραγωνικές παραβολές, δεδομένου ότι για τις κατά κανόνα εφαρμοζόμενες μεγάλες ακτίνες και την επιδιωκόμενη ακρίβεια το

κυκλικό τόξο και η τετραγωνική παραβολή συμπίπτουν. Καθοριστικό γεωμετρικό μέγεθος της κατακόρυφης καμπύλης είναι η ακτίνα καμπυλότητας Η στη θέση αλλαγής της κατά μήκος κλίσης

(*s* = 0%).

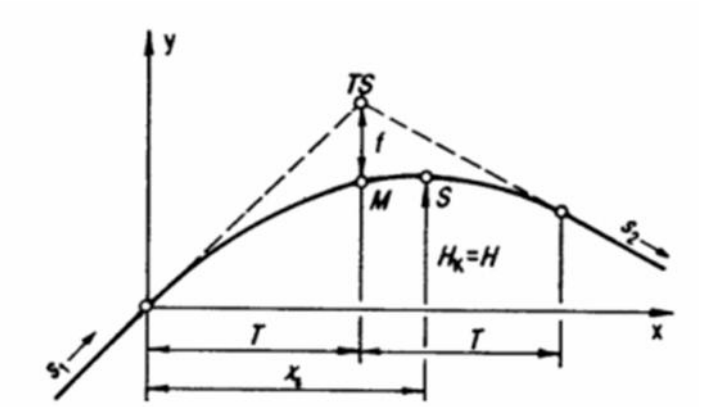
Οι ακτίνες των τόξων συναρμογής θα πρέπει να επιλέγονται κατά τέτοιον τρόπο, ώστε

- σε συνδυασμό με τα στοιχεία της οριζοντιογραφίας να παράγεται μια ισορροπημένη χάραξη στον χώρο
- να παρέχουν κυκλοφοριακή ασφάλεια μέσω τήρησης των μηκών ορατότητας για στάση (ΜΟΣ)
- να προσαρμόζονται όσο το δυνατόν καλύτερα στην τοπογραφία και να διατηρούν κατά το δυνατόν ανέπαφη την εικόνα του φυσικού τοπίου
- να λαμβάνονται υπόψη τα πολεοδομικά δεδομένα (ΤΜΑ 3).

Οι ελάχιστες ακτίνες κυρτής καμπύλης λαμβάνουν υπόψη την απαίτηση τήρησης του μήκους ορατότητας στάσης (ΜΟΣ) για την αναγνωρισιμότητα του τερματικού σημείου τμήματος κυκλοφοριακής συμφόρησης και επίσης εξασφαλίζουν το μήκος ορατότητας στάσης (ΜΟΣ) σε ευθυγραμμίες της οριζοντιογραφίας. Σε καμπύλες της οριζοντιογραφίας, καθώς και κατά την υπέρβαση προς τα κάτω των ελάχιστων τιμών πρέπει να καταδεικνύεται μελετητικά στο χώρο η τήρηση του μήκους ορατότητας για στάση.

Οι ελάχιστες ακτίνες κοίλης καμπύλης εξασφαλίζουν την τήρηση του μήκους ορατότητας για στάση (ΜΟΣ) ακόμη και κάτω από τεχνικά έργα. Για λόγους αρμονικής χάραξης πρέπει κατά το δυνατό να μην επιλεγούν μικρότερες τιμές από το ήμισυ των ακτινών κυρτώματος $H_W \geq 0{,}5 \cdot H_K$

Οι βασικές σχέσεις υπολογισμού μίας κατακόρυφης καμπύλης δίδονται παρακάτω:



$$x_S = -\frac{s_1}{100} \cdot H$$

$$s(x) = s_1 + \frac{x}{H} \cdot 100$$

$$y(x) = \frac{s_1}{100} \cdot x + \frac{x^2}{2 \cdot H}$$

$$T = \frac{H}{2} \cdot \frac{s_2 - s_1}{100}$$

$$f = \frac{T^2}{2 \cdot H} = \frac{T}{4} \cdot \frac{s_2 - s_1}{100} = \frac{H}{8} \cdot \left(\frac{s_2 - s_1}{100} \right)^2$$

Εικόνα 6: Σχηματική απεικόνιση και τύποι χάραξης τετραγωνικής παραβολής Πηγή : ΟΜΟΕ - Χ

Προκειμένου να αποφεύγεται η οπτική εικόνα θλάσης της χάραξης της μηκοτομής στις κορυφές της, το μήκος της εφαπτόμενης Τ πρέπει να είναι :

- για οδούς της ομάδας Α: $T_{min} = V_e$
- για οδούς της ομάδας Β: $T_{min} = 0,75 \cdot V_e$

όπου :

T_{min} [m] = ελάχιστο μήκος εφαπτομένης
V_e [km/h] = ταχύτητα μελέτης

Σε περιοχές με μικρές μεταβολές της κατά μήκος κλίσης επιτρέπεται η παράλειψη καμπύλης συναρμογής όταν η μέγιστη διαφορά των διαδοχικών κατά μήκος κλίσεων είναι: $Ds_{max} = 0,3 / V_e^2$ όπου Ds_{max} [m/ m] = η διαφορά κλίσεων.

Η ελάχιστη επιτρεπόμενη απόσταση μεταξύ διαδοχικών θλάσεων της χάραξης της μηκοτομής χωρίς καμπύλη συναρμογής είναι :

- για V_e > 70 km/ h, 30 m

- για V_e ≤ 70 km/ h, 15 m

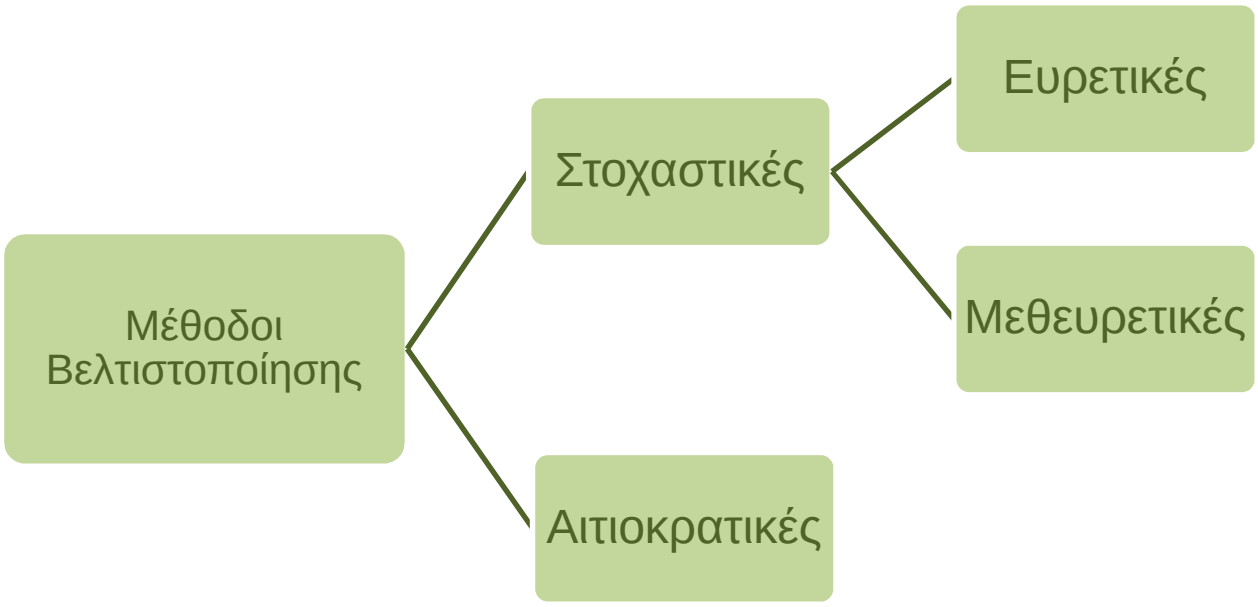
Όταν η αλλαγή της κλίσης γίνεται χωρίς καμπύλη συναρμογής στην πραγματικότητα στο σημείο θλάσης γίνεται στρογγύλευση από τον κατασκευαστή. Η παράλειψη της καμπύλης συναρμογής συνιστάται να αποφεύγεται στις εξής περιπτώσεις :

- σε θέσεις γεφυρών (περιλαμβάνονται και τα άκρα των γεφυρών)
- σε θέσεις ιρλανδικών ρείθρων (ροή επάνω στην οδό)
- σε άλλες θέσεις όπου απαιτείται προσεκτική διαμόρφωση των κλίσεων.

Τέλος, πρέπει να δίδεται προσοχή στον κατάλληλο συνδυασμό των στοιχείων μελέτης της μηκοτομής με εκείνα της οριζοντιογραφίας, έτσι ώστε να πληρούνται οι κανόνες χάραξης της οδού στο χώρο.

3.2 Βελτιστοποίηση

Βελτιστοποίηση είναι η διαδικασία εντοπισμού της καλύτερης δυνατής λύσης. Στα προβλήματα βελτιστοποίησης κατασκευάζεται μία συνάρτηση-εκπρόσωπος του προβλήματος, η αντικειμενική συνάρτηση και χρησιμοποιείται ως μέτρο σύγκρισης των πιθανών λύσεων. Σκοπός μας είναι να εντοπίσουμε τις τιμές των μεταβλητών στις οποίες η αντικειμενική συνάρτησή λαμβάνει την μέγιστη ή ελάχιστη τιμή.



Σχήμα 3: Μέθοδοι Βελτιστοποίησης

Οι μέθοδοι βελτιστοποίησης διακρίνονται σε αιτιοκρατικές και στοχαστικές. Οι αιτιοκρατικές μέθοδοι είναι ιδιαίτερα αποδοτικές εφόσον οδηγούν στη βέλτιστη επίλυση προβλημάτων μεγάλου εύρους πρακτικών εφαρμογών. Οι αιτιοκρατικοί αλγόριθμοι υποθέτουν ότι η “τέλεια” πληροφορία είναι γνωστή, δηλαδή ότι γνωρίζουμε πλήρως το μοντέλο, τις ιδιότητες και την συμπεριφορά της αντικειμενικής συνάρτησης και, συνεπώς, όλες αυτές οι πληροφορίες χρησιμοποιούνται για την εύρεση της ιδανικής κατεύθυνσης προς την οποία θα κινηθούμε σε κάθε βήμα. Το αποτέλεσμα μιας αιτιοκρατικής μεθόδου καθορίζεται επακριβώς από τις παραμέτρους και τις αρχικές συνθήκες, που καθορίζουν τις παραμέτρους. Συνεπώς, ένα πείραμα που θα επαναληφθεί με τις ίδιες ακριβώς συνθήκες, θα αποδώσει ακριβώς το ίδιο αποτέλεσμα. Στις στοχαστικές μεθόδους εισάγεται ένας βαθμός αβεβαιότητας. Η ίδια διαδικασία δεν αποδίδει πάντα το ίδιο αποτέλεσμα, αλλά ένα σύνολο δυνατών αποτελεσμάτων. Οι αλγόριθμοι αυτής της κατηγορίας, ψάχνουν σε όλο το χώρο αναζήτησης, αξιολογούν τα αποτελέσματα και ξαναπροσπαθούν κινούμενοι προς τυχαίες κατευθύνσεις. Συγκριτικά με τους αιτιοκρατικούς αλγόριθμους, οι στοχαστικοί αλγόριθμοι βελτιστοποίησης, έχουν καλύτερη συμπεριφορά σε μοντέλα που δεν είναι πλήρως γνωστά, είναι ανεκτικοί σε ανακρίβειες και δεν παγιδεύονται τόσο εύκολα σε τοπικά ελάχιστα, επομένως έχουν μεγαλύτερες πιθανότητες να εντοπίσουν την περιοχή της καθολικά βέλτιστης λύσης. Το βασικό τους μειονέκτημα είναι πως απαιτούν μεγάλο πλήθος υπολογισμών. Πολλές σύγχρονες μέθοδοι βελτιστοποίησης χρησιμοποιούν στοιχεία και από τις δύο μεθόδους. Οι στοχαστικές μέθοδοι διακρίνονται σε ευρετικές και μεθευρετικές. Οι ευρετικές αναζητούν λύση η οποία είναι σχεδόν βέλτιστη. Γίνεται αποδεκτή διότι ικανοποιεί κάποια κριτήρια όπως η ποιότητα της λύσης, δηλαδή η απόκλισή της από τη βέλτιστη, η ευκολία απόκτησης μιας λύσης. Οι μεθευρετικές μέθοδοι εφαρμόζουν γενικές στρατηγικές αναζήτησης, οι οποίες συνήθως είναι εμπνευσμένες από τις φυσικές διαδικασίες. Οι μεθευρετικές μέθοδοι διαχειρίζονται ένα πληθυσμό λύσεων ταυτόχρονα σε πολλές κατευθύνσεις.

3.2.1 Μέθοδος Monte Carlo

Η μέθοδος Monte Carlo είναι μια αριθμητική μέθοδος για την επίλυση μαθηματικών προβλημάτων με τη χρήση γεννήτριας τυχαίων αριθμών. Χρησιμοποιείται για την επίλυση αλλά και προσομοίωση προβλημάτων μαθηματικής φύσης. Σύμφωνα με πιο γενικούς ορισμούς της μεθόδου, κάθε πείραμα στο οποίο χρησιμοποιούνται τυχαίοι αριθμοί για την εξέταση του προβλήματος, καλείται Monte Carlo πείραμα. Η βασική τους ιδέα είναι να χρησιμοποιηθούν τυχαίες λύσεις για την επίλυση προβλημάτων που μπορεί να έχουν ντετερμινιστική λύση αλλά είναι αδύνατο να βρεθεί με ντετερμινιστική μέθοδο. Η μέθοδος δανείστηκε το όνομα της πόλης του Πριγκιπάτου του Monaco,

που είναι γνωστό για τα καζίνο του το οποίο φιλοξενεί μια από τις απλούστερες συσκευές παραγωγής τυχαίων αριθμών, τη ρουλέτα.

Η μέθοδος αποδίδεται στους N. Metropolis & S. Ulam οι οποίοι το 1949 στο άρθρο τους με τίτλο "Η μέθοδος Monte Carlo" στο Journal of the American Statistics Association πρωτοπαρουσίασαν την μέθοδο. Χρησιμοποιήθηκε δε από τους Ulam και von Neumann για την προτυποποίηση ενός συστήματος διάχυσης νετρονίων μέσω σχάσιμων υλικών. Αν και ήδη αρκετά προβλήματα στατιστικής λύνονταν μέσω τυχαίας δειγματοληψίας, που είναι στην ουσία η μέθοδος Monte Carlo, πολύ πριν από το 1949. Συμπεραίνουμε λοιπόν ότι η θεωρητική βάση της μεθόδου ήταν γνωστή ωστόσο η προσομοίωση τυχαίων μεταβλητών είναι μια δύσκολη διαδικασία για να γίνει χειρωνακτικά, επομένως η μέθοδος διαδόθηκε με την εμφάνιση και εξέλιξη των υπολογιστών.

Η μέθοδος Monte Carlo επιτρέπει την προσομοίωση συστημάτων τα οποία ακολουθούν τους φυσικούς νόμους με απλό τρόπο σε ποσοστό ακριβείας πολλές φορές υψηλότερο από τη λύση περίπλοκων αναλυτικών μεθόδων. Με την μέθοδο Monte Carlo, μεγάλα συστήματα μπορούν να εκπροσωπηθούν από έναν αριθμό τυχαίων δειγματοληψιών, και τα εξαγόμενα δεδομένα μπορούν να χρησιμοποιηθούν για την περιγραφή του συστήματος. Χρησιμοποιούνται κυρίως σε τρεις κατηγορίες προβλημάτων βελτιστοποίησης, αριθμητικής ολοκλήρωσης και δημιουργίας γενεών από τυχαία δειγματοληψία. Εξαιτίας των πλεονεκτημάτων της όμως, παραλλαγές της εφαρμόζονται σε πλήθος επιστημονικών κλάδων, από την οικονομία και την ψυχολογία ως την χημεία και τη ρύθμιση της κυκλοφορίας.

Στη φυσική, οι μέθοδοι Monte Carlo είναι χρήσιμες για την προσομοίωση συστημάτων με πολλούς βαθμούς ελευθερίας, όπως υγρά, ανομοιογενή υλικά, ισχυρά συζευγμένα στερεά και κυτταρικές δομές (βλέπε μοντέλο κυτταρικού Potts, αλληλεπιδρώντα συστήματα σωματιδίων, διαδικασίες McKean-Vlasov, κινητικά μοντέλα αερίων). Ακόμη χρησιμοποιούνται για τη μοντελοποίηση φαινόμενων με σημαντική αβεβαιότητα παραμέτρων όπως ο υπολογισμός του κινδύνου στις επιχειρήσεις και, στα μαθηματικά, για τον υπολογισμό πολυδιάστατων ολοκληρωμάτων με περίπλοκες οριακές συνθήκες. Χρησιμοποιούνται ακόμη σε προβλήματα αεροδιαστημικής και εξόρυξης πετρελαίου, όπου υπολογίζουν τις προβλέψεις της αποτυχίας και τις υπερβάσεις κόστους και προγράμματος καλύτερα από την ανθρώπινη διαίσθηση και εναλλακτικές μεθόδους.

Οι μέθοδοι Monte Carlo μπορούν να χρησιμοποιηθούν για την επίλυση προβλημάτων που έχουν πιθανολογική ερμηνεία. Σύμφωνα το νόμο των μεγάλων αριθμών, τα ολοκληρώματα που χρησιμοποιούν την αναμενόμενη τιμή κάποιας τυχαίας μεταβλητής μπορούν κατά προσέγγιση να επιλυθούν με τη χρήση του εμπειρικού μέσου (του μέσου του δείγματος) ανεξάρτητων δειγμάτων της μεταβλητής.

Αλγόριθμος Monte Carlo

Μια μέθοδος Monte Carlo είναι ένας υπολογιστικός αλγόριθμος που βασίζεται στην επαναλαμβανόμενη τυχαία δειγματοληψία για να υπολογίσει τα αποτελέσματά της. Με λίγα λόγια, αντί να εκτελούνται πολύπλοκες υπολογισμούς, εκτελούμε ένα μεγάλο αριθμό "πειραμάτων" χρησιμοποιώντας μια σχεδόν τυχαία γενιά αριθμών για να επιλύσουμε το πρόβλημα. Οι αλγόριθμοι Monte Carlo τείνουν να χρησιμοποιούνται όταν είναι αδύνατο να υπολογιστεί ένα ακριβές αποτέλεσμα με έναν ντετερμινιστικό αλγόριθμο.

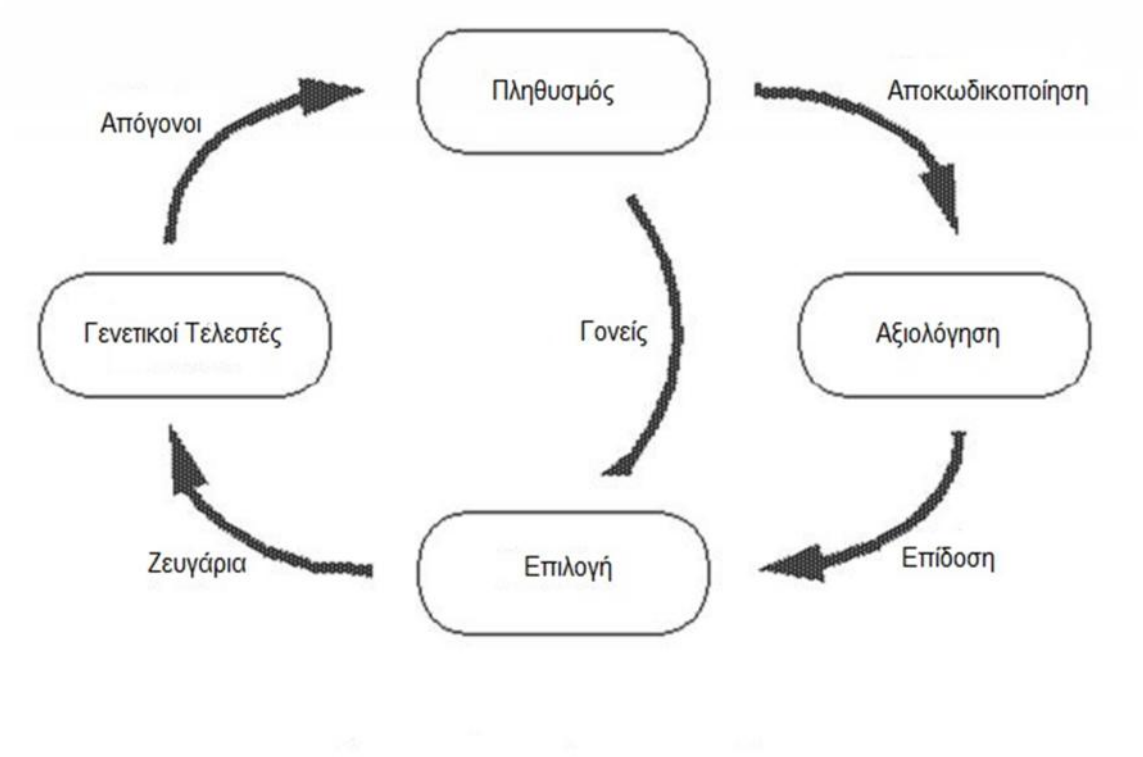
Η μεθοδολογία της Monte Carlo είναι σχετικά απλή. Προκειμένου όμως να εξασφαλιστεί η ακρίβεια των αποτελεσμάτων της πρέπει να οριστούν επαρκώς τα απαραίτητα για την εκτέλεσή της βήματα:

- Βήμα 1: Δημιουργείται κατάλληλο παραμετρικό πρότυπο, $f(x_1, x_2, ..., x_n)$. Προκειμένου να επιλυθεί το εκάστοτε πρόβλημα μέσω της επαναληπτικής μεθόδου θα πρέπει να παρασταθεί μέσω ενός μαθηματικού προτύπου
- Βήμα 2: Δημιουργείται ένα σύνολο τυχαίων εισόδων, $x_{i1}, x_{i2}, ..., x_{in}$ και $y_{i1}, y_{i2}, ..., y_{in}$. Αφού κατασκευαστεί το μαθηματικό μοντέλο θα πρέπει να εξασφαλιστεί ότι οι τιμές εισόδου της συνάρτησης είναι μεν τυχαίες αλλά παράγονται μέσω τυχαίας επιλογής αριθμών μιας κατανομής, "ψευδοτυχαίοι" αριθμοί. Σκοπός είναι η παράγωγή αλληλουχίας ανεξάρτητων, ομοιόμορφα κατανεμημένων (U (0,1)) τυχαίων μεταβλητών. Σε ορισμένες περιπτώσεις χρησιμοποιούνται οι U(0,1) τυχαίοι αριθμοί που παράγονται, ωστόσο στις περισσότερες περιπτώσεις χρησιμοποιούνται για να παραχθούν δευτερευόντως τυχαίοι αριθμοί οι οποίοι έχουν διαφορετικές κατανομές. Σε περίπτωση που δεν εξασφαλιστεί η παραπάνω προϋπόθεση η μέθοδος κινδυνεύει να εγκλωβιστεί σε κάποιο τοπικό μέγιστο. Όπως αναφέρθηκε και παραπάνω οι έρευνες που έχουν γίνει είναι για αριθμούς που παράγονται από διάφορες κατανομές πιθανότητας.
- Βήμα 3: Αξιολογείται το μοντέλο και αποθηκεύονται τα αποτελέσματα ως fi.
- Βήμα 4: Επαναλαμβάνονται τα βήματα 2 και 3 για i = 1 έως n. Όπως αναφέρθηκε η Monte Carlo είναι μια επαναληπτική μέθοδος. Είναι εξαιρετικής σημασίας να οριστεί δόκιμο πλήθος επαναλήψεων. Προφανώς όσο μεγαλύτερο είναι το πλήθος των επαναλήψεων τόσο ακριβέστερα θα είναι και τα αποτελέσματα της μεθόδου. Όσο όμως αυξάνονται οι επαναλήψεις η μέθοδος γίνεται χρονοβόρα. Πολλές έρευνες που πραγματοποιήθηκαν συμπεραίνουν ότι η επαναληπτική διαδικασία τερματίζεται όταν το σφάλμα των αποτελεσμάτων ισούται με $1/\sqrt{N}$. Ωστόσο στη παρούσα διπλωματική δεν είναι δυνατό να υπολογιστεί το σφάλμα της διαδικασίας για αυτό πραγματοποιήθηκε διερεύνηση για να βρεθεί το κατάλληλο πλήθος επαναλήψεων. Τονίζεται δε ότι το πλήθος επαναλήψεων που επιλέχθηκε μεταβάλλεται ανάλογα με το πλήθος των προς εύρεση παραμέτρων.

- Βήμα 5: Αναλύονται τα αποτελέσματα χρησιμοποιώντας ιστογράμματα, συνοπτικά στατιστικά στοιχεία, διαστήματα εμπιστοσύνης κ.λπ.

3.2.2 Γενετικοί Αλγόριθμοι

Στην κατηγορία των στοχαστικών μεθευρετικών μεθόδων βελτιστοποίησης ανήκουν οι Γενετικοί Αλγόριθμοι (ΓΑ) και αποτελούν παρακλάδι των Εξελικτικών Αλγορίθμων. Οι Γενετικοί Αλγόριθμοι στηρίχτηκαν στη μίμηση των διαδικασιών της εξέλιξης που συναντάμε στην φύση. Σύμφωνα με την Θεωρία Εξέλιξης των Ειδών του Δαρβίνου, στη φύση επιβιώνει ο πιο δυνατός οργανισμός, καθώς το φυσικό περιβάλλον μεταβάλλεται οι οργανισμοί θα πρέπει να εξελίσσονται συνεχώς προκειμένου να επιβιώσουν. Βασισμένοι σε αυτή την ιδέα οι Γενετικοί Αλγόριθμοι διατηρούν ένα πληθυσμό πιθανών λύσεων ο οποίος σε κάθε επανάληψη του αλγορίθμου βελτιώνεται, με την έννοια ότι τα άτομα που τον αποτελούν πλησιάζουν όλο και περισσότερο στη λύση του προβλήματος.



Εικόνα 8 : Διαδικασία Γενετικών Αλγορίθμων

Η διαδικασία της βιολογικής εξέλιξης των οργανισμών αποτυπώνεται στα χρωμοσώματά τους. Το χρωμόσωμα είναι μία οργανωμένη δομή DNA η οποία βρίσκεται στα κύτταρα των οργανισμών και αποτελείται από γονίδια. Τα γονίδια περιέχουν αποθηκευμένη μία συγκεκριμένη γενετική

πληροφορία, δηλαδή μία τιμή για κάποιο χαρακτηριστικό. Τα γονίδια, τα οποία επηρεάζουν συγκεκριμένα χαρακτηριστικά, βρίσκονται σε συγκεκριμένες θέσεις του χρωμοσώματος που ονομάζονται τόποι. Γονίδια τα οποία αναφέρονται στο ίδιο χαρακτηριστικό λέγονται αλληλόμορφα, ενώ το σύνολο όλων των γονιδίων ενός οργανισμού καλείται γονότυπος. Ο γονότυπος ενός ατόμου καθορίζει τα φυσικά χαρακτηριστικά του, όπως το ύψος, το χρώμα ματιών κ.α., τα οποία καλούνται φαινότυπος. Βασικές λειτουργίες της εξέλιξης των ειδών είναι η αναπαραγωγή και η μετάλλαξη. Στην αναπαραγωγή κάθε απόγονος λαμβάνει ένα γονίδιο από τον κάθε γονέα για κάθε χαρακτηριστικό. Σε περίπτωση που τα γονίδια διαφωνούν στην τιμή τους, στον απόγονο θα επικρατήσει η τιμή του ισχυρότερου γονιδίου το οποίο λέγεται κυρίαρχο ή επικρατές, ενώ η τιμή του πιο αδύναμου γονιδίου (υπολειπόμενο γονίδιο) θα αγνοηθεί. Η μετάλλαξη συμβαίνει σπάνια και με τυχαίο τρόπο, είτε από κάποιο λάθος στην αντιγραφή του γονιδίου, είτε λόγω κάποιου εξωγενή παράγοντα και κάποιες φορές οδηγεί σε θάνατο, ενώ κάποιες άλλες σε σημαντική βελτίωση του χαρακτηριστικού. Παρόμοια οι έννοιες άτομα (individuals), γονότυπος (genotype), χρωμόσωμα (chromosome), γονίδιο (gene), τόπος ή θέση (loci), αλληλόμορφο (allele), φαινότυπος (phenotype), διασταύρωση (crossover), μετάλλαξη (mutate), αντικειμενική συνάρτηση(objective, fitness function) κ.α. εμφανίζονται και στους γενετικούς αλγόριθμους.

Η εισαγωγή στους γενετικούς αλγορίθμους έγινε από τον Friedberg το 1958 ο οποίος επιχείρησε την αυτόματη παραγωγή σύνθετων προγραμμάτων Fortran με τον συνδυασμό μικρότερων προγραμμάτων ωστόσο, τα προγράμματα που προέκυπταν τις περισσότερες φορές δεν ήταν εκτελέσιμα. Η πρώτη εργασία που αφορά τις αρχές που διέπουν τους γενετικούς αλγορίθμους δημοσιεύτηκε από τον Holland το 1975. Ο Holland εισάγοντας την κωδικοποίηση των λύσεων σε σειρές δυαδικών ψηφίων έδωσε μια νέα ώθηση στο χώρο. Αποτέλεσμα αυτής της εργασίας ήταν οι γενετικοί αλγόριθμοι, μια νέα, εξελισσόμενη και πολλά υποσχόμενη τεχνική αναζήτησης και βελτιστοποίησης. Που μπορεί να βρει τη συνολική βέλτιστη λύση σε πολυσύνθετους χώρους, με πολλές διαστάσεις. Οι αρχές των γενετικών αλγορίθμων έχουν αναπτυχθεί αρκετά και περιγραφεί σε αρκετά κείμενα όπως αυτά των Goldberg (1989) και Michalewicz (1996)

Ο γενετικός αλγόριθμος αφού δημιουργηθεί ένας αρχικός πληθυσμός, λειτουργεί επαναληπτικά. Αρχικά, διασταυρώνει τους γονείς και δημιουργεί το νέο πληθυσμό. Στη συνέχεια, μεταλλάσει τα γονίδια ορισμένων απογόνων με βάση το συντελεστή μετάλλαξης. Ο πληθυσμός αξιολογείται με βάση την αντικειμενική συνάρτηση και επιλέγεται ένα κομμάτι του συνολικού πληθυσμού συνήθως ίσο με τον αρχικό το οποίο χρησιμοποιείται ως η νέα γενιά γονέων. Η διαδικασία αυτή επαναλαμβάνεται μέχρι τον τερματισμό του αλγορίθμου όταν

- επιτευχθεί ο μέγιστος αριθμός γενιών (επαναλήψεων)
- για ένα συγκεκριμένο αριθμό γενιών δεν υπάρχει βελτίωση στην καλύτερη λύση του πληθυσμού.

- δεν υπάρχει αλλαγή σε κανένα μέρος του πληθυσμού.
- μία αποδεκτή λύση έχει βρεθεί.

Τα βασικά βήματα των γενετικών αλγορίθμων είναι:

1. Δημιουργία ενός αρχικού πληθυσμού από πιθανές λύσεις
2. Επιλογή ενός νέου πληθυσμού με βάση την απόδοση κάθε μέλους του πληθυσμού
3. Αξιολόγηση κάθε λύσης χρησιμοποιώντας την αντικειμενική συνάρτηση
4. Γενετικοί τελεστές
5. Νέα γενιά (επιστροφή στο βήμα 2)
6. Τερματισμός αλγορίθμου

Αρχικός πληθυσμός

Το πρώτο βήμα κάθε γενετικού αλγορίθμου είναι η δημιουργία ενός αρχικού πληθυσμού. Συνήθως, ο αρχικός πληθυσμός αποτελείται από τυχαία διαμορφωμένες λύσεις. Το μέγεθος του πληθυσμού είναι δύσκολο να οριστεί. Σημαντικό δε είναι από τον αρχικό πληθυσμό να μπορούν να προκύψουν νέα άτομα από όλο τον χώρο αναζήτησης ώστε να καλύπτονται όλες οι δυνατές λύσεις. Δηλαδή εκτός από το μέγεθος του πληθυσμού πρέπει να εξασφαλίζεται και η ποικιλομορφία του. Επίσης πρέπει να αναφερθεί πως όσο μεγαλώνει το μέγεθος του πληθυσμού μεγαλώνει ο χώρος των δυνατών λύσεων που θα καλυφθεί αλλά αυξάνεται και ο χρόνος εκτέλεσης του προγράμματος.

Κωδικοποίηση

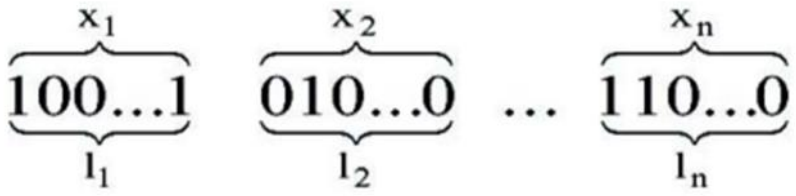
Για τη λειτουργία του γενετικού αλγορίθμου απαιτείται η αναπαράσταση των λύσεων σε μια ακολουθία που λέγεται χρωμόσωμα. Για να μετατραπεί μια λύση σε χρωμόσωμα πρέπει να προηγηθεί μια διαδικασία κωδικοποίησης και για το αντίστροφο, να μετατραπεί δηλαδή ένα χρωμόσωμα σε λύση, μια διαδικασία αποκωδικοποίησης. Στη συνέχεια περιγράφονται οι κυριότερες μέθοδοι κωδικοποίησης.

Δυαδική κωδικοποίηση (Binary Encoding)

Η δυαδική κωδικοποίηση είναι ο πιο διαδεδομένος τόπος αναπαράστασης ενός ατόμου, χρησιμοποιήθηκε δε και στην παρούσα έρευνα. Σε αυτή τη μέθοδο κάθε χρωμόσωμα του πληθυσμού αναπαρίσταται με μία δυαδική συμβολοσειρά, δηλαδή μία ακολουθία από 0 και 1. Η θέση και η τιμή κάθε ψηφίου (bit) της ακολουθίας αναπαριστά κάποια πληροφορία για ένα χαρακτηριστικό του ατόμου που αντιπροσωπεύει. Το μήκος των γονιδίων καθορίζεται από την επιθυμητή ακρίβεια. Ο συνολικός αριθμός των bits με τα οποία αναπαρίσταται κάθε λύση καθορίζεται από το πλήθος των αγνώστων μεταβλητών

Η δυαδική απεικόνιση εξασφαλίζει τεράστια ευελιξία, καθώς επιτρέπει το χειρισμό τόσο πραγματικών όσο και ακέραιων μεταβλητών, καθώς και μεταβλητών που δεν περιγράφονται

αριθμητικά, όπως λογικές εκφράσεις. Για να μπορέσει να κωδικοποιηθεί ο χώρος αναζήτησης μιας συνάρτησης και να επιτύχουμε την απαιτούμενη ακρίβεια μιας μεταβλητής x_i με n ακρίβεια απαιτούμενων ψηφίων έστω διάστημα $[x_1, x_2]$, θα χωριστεί σε ίσα υποδιαστήματα $(x_2 - x_1)*10^n$ όπου n είναι η απαιτούμενη ακρίβεια μιας μεταβλητής x_i. Αν l_i ο μικρότερος ακέραιος που ικανοποιεί τη σχέση $(x_2 - x_1)*10^n \leq 2^{l_i} - 1$ τότε η αναπαράσταση της μεταβλητής x_i σε δυαδική συμβολοσειρά μήκους l_i , ικανοποιεί την απαίτηση για ακρίβεια n δεκαδικών ψηφίων. Μία τυπική μορφή ενός χρωμοσώματος, είναι η ακόλουθη:



Εικόνα 9: Δυαδική Κωδικοποίηση

Κωδικοποίηση μετάθεσης (Permutation Encoding)

Στην κωδικοποίηση μετάθεσης ο συμβολισμός του χρωμοσώματος (δηλαδή των υποψήφιων λύσεων) αποτελείται από αριθμούς που παριστάνουν τη θέση σε μια ακολουθία. Η κωδικοποίηση αυτή μπορεί να χρησιμοποιηθεί σε συνδυαστικά προβλήματα όπως, για παράδειγμα στο πρόβλημα του πλανόδιου πωλητή.

Εικόνα 10: Κωδικοποίηση Μετάθεσης

Κωδικοποίηση δένδρου (Tree Encoding)

Το χρωμόσωμα (υποψήφια λύση) σε αυτήν την περίπτωση μπορεί να παραλληλιστεί με ένα δένδρο κατάλληλα κωδικοποιημένων αντικειμένων, όπου τα αντικείμενα περιλαμβάνουν συναρτήσεις ή εντολές σε γλώσσα προγραμματισμού. Αυτό το είδος κωδικοποίησης είναι ιδιαίτερα χρήσιμο για εξελισσόμενα προγράμματα (evolving programs).

Κωδικοποίηση τιμών (Value Encoding)

Η κωδικοποίηση τιμής χρησιμοποιείται σε κάποια ιδιαίτερα και περίπλοκα προβλήματα στα οποία η δυαδική κωδικοποίηση θα ήταν πολύ δύσκολο να εφαρμοστεί. Το κάθε χρωμόσωμα, εδώ, μπορεί να είναι μία ακολουθία πραγματικών αριθμών, χαρακτήρων ή ακόμα και αντικειμένων άμεσα συσχετιζόμενων με το προς βελτιστοποίηση πρόβλημα.

Η χρήση των γονιδίων με πραγματικές τιμές θεωρείται ότι πλεονεκτεί σε σχέση με τις δυαδικές κωδικοποιήσεις στην αριθμητική συνάρτηση βελτιστοποίησης. Η αποδοτικότητα των GA αυξάνεται δεδομένου ότι δεν μετατρέπονται τα χρωμοσώματα σε φαινοτύπους πριν από κάθε αξιολόγηση της συνάρτησης, επίσης απαιτείται λιγότερη μνήμη αφού αποδοτικές floating-point εσωτερικές αναπαραστάσεις στον υπολογιστή μπορούν να χρησιμοποιηθούν άμεσα, δεν μειώνεται η ακρίβεια εξαιτίας της κωδικοποίησης σε δυαδικές ή άλλες τιμές, και υπάρχει μεγαλύτερη ελευθερία να χρησιμοποιηθούν διαφορετικοί γενετικοί τελεστές.

Χρωμόσωμα(1)	1.231 2.436 7.497 0.841
Χρωμόσωμα(2)	HFKDJSOWEPATHJURP
Χρωμόσωμα(3)	(back), (left), (left), (right)

Εικόνα 11: Κωδικοποίηση Τιμών

Αντικειμενική Συνάρτηση

Η συνάρτηση καταλληλότητας ή η αντικειμενική συνάρτηση αποτελεί το κριτήριο για την αξιολόγηση των ατόμων, χρωμοσωμάτων. Η συνάρτηση δέχεται ως είσοδο ένα χρωμόσωμα και επιστρέφει έναν αριθμό που υποδηλώνει το βαθμό καταλληλότητας του. Η αντικειμενική συνάρτηση παίζει το ρόλο του περιβάλλοντος στο τεχνικό μοντέλο. Όταν η τιμή της συνάρτησης καταλληλότητας ενός χρωμοσώματος είναι σε μεγάλη απόσταση από τις αντίστοιχες τιμές άλλων χρωμοσωμάτων, τότε το προαναφερθέν χρωμόσωμα συχνά λειτουργεί ως υπεράτομο μέσα στον πληθυσμό και οδηγεί σε πρόωρη σύγκλιση.

Επιλογή

Κατά τη διαδικασία της επιλογής ουσιαστικά υλοποιείται η επιβίωση του πιο ισχυρού ατόμου (καλύτερης λύσης). Αφού επιλέγονται γονείς προς αναπαραγωγή καινούριων λύσεων με βάση την απόδοση τους ως προς την αντικειμενική συνάρτηση. Ωστόσο έρευνες έχουν δείξει ότι η επιλογή

μόνο ‘κατάλληλων’ γονέων κινδυνεύει να παγιδεύσει την συνάρτηση σε τοπικά ακρότατα και να μειώσει την ευλυγισία του αλγορίθμου. Προκειμένου να εξασφαλιστούν τα παραπάνω κατασκευάστηκαν διάφορες μέθοδοι επιλογής οι οποίες εξασφαλίζουν την επιβίωση και ενός, μικρότερου βέβαια, μέρους ‘ακατάλληλων’ γονέων. Αφού αντιγραφούν τα χρωμοσώματα σε μια δεξαμενή(mating pool) με μέγεθος ίσο με τον αρχικό πληθυσμό πραγματοποιείται μια από τις παρακάτω μεθόδους.

Επιλογή Ρουλέτας (Roulette Wheel Selection)

Η μέθοδος επιλογής ρουλέτας παριστάνεται από μία ρουλέτα η οποία έχει χωριστεί σε τμήματα, όσα και τα άτομα του πληθυσμού. Κάθε τμήμα αντιπροσωπεύει ένα άτομο του πληθυσμού και το μέγεθός του είναι ανάλογο με την πιθανότητα επιλογής που έχει το συγκεκριμένο άτομο σύμφωνα με την τιμή της αντικειμενικής του συνάρτησης. Κάθε φορά που η ρουλέτα περιστρέφεται επιλέγεται ένα άτομο μέχρι να συμπληρωθεί το επιθυμητό μέγεθος πληθυσμού.

Η μέθοδος της ρουλέτας είναι στοχαστική επομένως είναι δυνατό να επιλεγεί ένα τμήμα με χαμηλή αντικειμενική συνάρτηση οι πιθανότητες όμως ευνοούν τα άτομα με μεγάλη τιμή αντικειμενικής συνάρτησης. Σε αυτή τη περίπτωση η μέθοδος κινδυνεύει να εγκλωβιστεί σε τοπικά βέλτιστα και να δημιουργήσει μια στασιμότητα στο πληθυσμό. Ο αλγόριθμος τότε θα έχει πολύ αργή σύγκλιση αφού από γενιά σε γενιά δεν θα υπάρχει σημαντική βελτίωση στην ατομική και συνολική καταλληλότητα.

Γραμμική Ταξινόμηση (Linear Ranking Selection)

Στη μέθοδο γραμμικής ταξινόμησης τα άτομα ενός πληθυσμού ταξινομούνται με βάση την τιμή της αντικειμενικής τους συνάρτησης. Η επιλογή γίνεται με βάση την πιθανότητα που τους αντιστοιχεί με βάση τη θέση που έλαβαν στην λίστα ταξινόμησης. Η πιθανότητα υπολογίζεται από τον τύπο, $p_i = {2 * (n - j)}/{n * (n - 1)}$ όπου n είναι το μέγεθος του πληθυσμού και j η θέση του ατόμου i στην κατάταξη.

Στην μέθοδο αυτή πρέπει να συνταχθεί μια διαδικασία που θα εξασφαλίζει ότι δεν έχει επιλεχθεί ως γονέας το ίδιο άτομο. Διότι αυτό θα είχε ως αποτέλεσμα ο απόγονος να είναι ένα ακριβές αντίγραφο του γονέα, γεγονός που εμποδίζει την εξέλιξη του αλγορίθμου. Επίσης πρέπει να εξασφαλιστεί ότι ένα άτομο δεν επιλέγεται ως γονέας σε πολλές διαδικασίες διασταύρωσης καθώς η κυριάρχηση του γονέα θα οδηγήσει τη μέθοδο σε πρόωρη σύγκλιση.

Επιλογή Ταξινόμησης (Rank selection)

Στη μέθοδο επιλογής ταξινόμησης τα άτομα αξιολογούνται από τη συνάρτηση στόχου και ταξινομούνται με βάση την τιμή της αξιολόγησής τους σε αύξουσα σειρά. Η πιθανότητα επιλογής τους υπολογίζεται με βάση τη θέση που έχουν στη ταξινομημένη λίστα από τον τύπο

p = $\frac{2-s}{\mu} + \frac{2*i*(s-1)}{\mu*(\mu-1)}$, όπου s ∈ (1, 2] παράμετρος και μ η τάξη του ατόμου i.

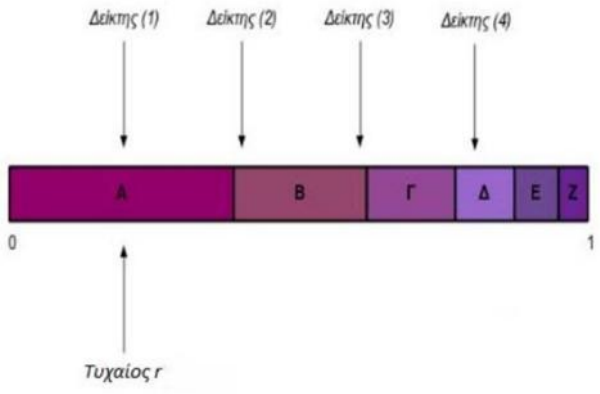
Με αυτόν τον τρόπο, αντιμετωπίζεται το πρόβλημα της πρόωρης σύγκλισης όταν κάποιο άτομο του πληθυσμού έχει υψηλή τιμή αντικειμενικής συνάρτησης .Όμως είναι πολύ πιθανό να έχουμε αργή σύγκλιση του αλγορίθμου, αφού τα ικανότερα άτομα δεν διαφέρουν και πολύ από τα υπόλοιπα.

Επιλογή του καλύτερου συζύγου (Top Mate Selection)

Η μέθοδος αυτή είναι μια αιτιοκρατική μέθοδος και είναι η απλούστερη μέθοδος επιλογής γονέων. Σε αυτήν τη μέθοδο ο πληθυσμός ταξινομείται με φθίνουσα σειρά σύμφωνα με την τιμή της αντικειμενικής συνάρτησης. Στη συνέχεια, ένα ποσοστό (συνήθως το 50%) του πληθυσμού με τα λιγότερο κατάλληλα χρωμοσώματα αγνοείται, ενώ τα υπόλοιπα χρωμοσώματα χρησιμοποιούνται για να παραχθεί η επόμενη γενιά. Πλεονέκτημα αυτής της μεθόδου είναι η απλότητα της αφού το μόνο που απαιτείται είναι η επιλογή των καλύτερων λύσεων στην υπάρχουσα γενιά και στη συνέχεια τυχαίο ζευγάρωμα. Μειονέκτημα της είναι η απώλεια γενετικής πληροφορίας από τα αδύναμα χρωματοσώματα (φαινομενικά ακατάλληλες λύσεις). Η μέθοδος αυτή συχνά οδηγεί σε πρόωρη σύγκλιση λόγο της απώλειας χαρακτηριστικών σε πρώιμο στάδιο και των εγκλωβισμό σε τοπικό ακρότατο.

Ομοιόμορφη στοχαστική δειγματοληψία (Stochastic Universal Sampling)

Στην Ομοιόμορφη στοχαστική δειγματοληψία τα χρωμοσώματα τοποθετούνται σε έναν άξονα, σε κάθε χρωμόσωμα αντιστοιχεί ένα τμήμα του άξονα ανάλογο με την καταλληλότητά του, όπως και στην μέθοδο της ρουλέτας. Στη συνέχεια τοποθετούνται στον άξονα τόσοι δείκτες όσο και το απαιτούμενο μέγεθος τους πληθυσμού. Οι δείκτες απέχουν ανά δυο 1/k και η θέση του πρώτου δείκτη πάνω στον άξονα ανήκει στο διάστημα [0, 1/k] και επιλέγεται τυχαία.



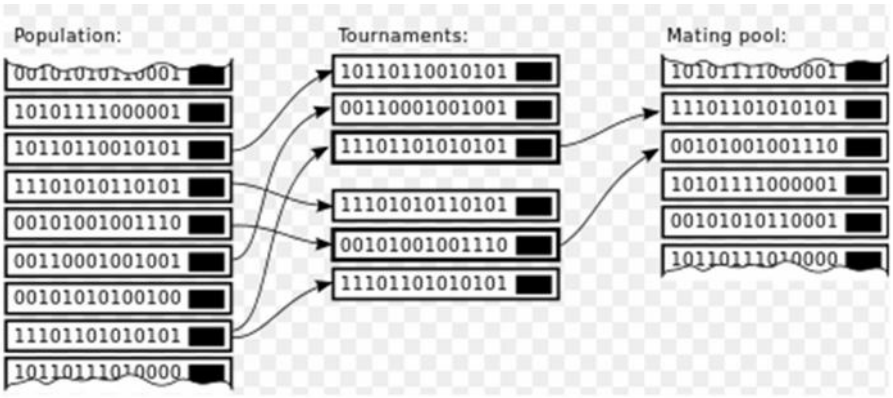
Εικόνα 12: Ομοιόμορφη Στοχαστική Δειγματοληψία

Η στοχαστική ομοιόμορφη δειγματοληψία, σε γενικά επιλέγει ικανότερα άτομα συγκριτικά με την μέθοδο της ρουλέτας.

Επιλογή με Διαγωνισμό (Tournament Selection)

Η επιλογή με διαγωνισμό είναι μια ευρέως διαδεδομένη στοχαστική μέθοδος επιλογής. Σε αυτή τη μεθοδολογία επιλέγονται τυχαία δύο ή περισσότερα άτομα του πληθυσμού, αξιολογούνται και επιλέγονται τα άτομα με τις μεγαλύτερες τιμές της αντικειμενικής συνάρτησης. Η διαδικασία επαναλαμβάνεται μέχρι να επιλεχθεί ο απαιτούμενος αριθμός γονέων.

Πλεονέκτημα αυτής της μεθόδου είναι ότι ελέγχεται παραμετρικά η πιθανότητα επιλογής που έχουν τα άτομα του πληθυσμού, μέσω του καθορισμού του μεγέθους του εκάστοτε διαγωνισμού. Όσο αυξάνεται ο αριθμός των υποψήφιων λύσεων που συμμετέχουν στον διαγωνισμό, τόσο ελαττώνεται η πιθανότητα επιλογής των ατόμων του πληθυσμού με χαμηλή τιμή καταλληλότητας και αντίστροφα. Η τεχνική αυτή οδηγεί γενικά σε πληθυσμούς με μεγαλύτερη ποικιλία απ' ότι οι άλλες μέθοδοι.



Εικόνα 13: Μέθοδος επιλογής με διαγωνισμό

Ελιτισμός (Elitism)

Στον ελιτισμό προκειμένου να αποτραπεί η πιθανότητα να χαθούν τα ικανότερα χρωμοσώματα, επιλέγεται το ικανότερο ή τα ικανότερα χρωμοσώματα και αντιγράφονται κατευθείαν στην νέα γενιά. Τα υπόλοιπα επιλέγονται με κάποια κλασσική μέθοδο, όπως αυτές που αναφέρθηκαν παραπάνω. Βασικό μειονέκτημα του ελιτισμού είναι η πιθανή παγίδευση σε τοπικά ακρότατα και αυτός είναι ο λόγος που προτιμάται η εφαρμογή του ελιτισμού σε συνδυασμό με κάποια άλλη μεθοδολογία.

Διασταύρωση

Αφού καθοριστεί ο αρχικός πληθυσμός κωδικοποιηθεί και αξιολογηθεί επιλέγονται οι γονείς και δημιουργείται η πισίνα ζευγαρώματος προκειμένου να δημιουργηθούν οι απόγονοι στη συνέχεια. Η

δημιουργία των απογόνων πραγματοποιείται είτε με τη διαδικασία της διασταύρωσης, είτε με τη διαδικασία της μετάλλαξης είτε και με τις δύο. Οι τελεστές διασταύρωσης είναι ο τελεστής όπου ο απόγονος δημιουργείται από ένα μόνο γονέα (asexual), ο τελεστής όπου οι δύο γονείς χρησιμοποιούνται για τη δημιουργία δύο ή περισσοτέρων απογόνων (sexual) και ο τελεστής όπου περισσότεροι από δύο γονείς χρησιμοποιούνται για τη δημιουργία δύο ή περισσοτέρων απογόνων (multi-recombination). Το ποσοστό των ατόμων του πληθυσμού που υπόκεινται σε μία διαδικασία διασταύρωσης είναι συνήθως μεταξύ του πενήντα (50%) και του ενενήντα (90%) τοις εκατό με συνηθέστερη τιμή το ογδόντα τοις εκατό (80%).

Από δύο ‘κατάλληλες’ λύσεις είναι πιθανόν να εμφανιστούν δύο ‘ακατάλληλες’ λύσεις που θα πρέπει είτε να απορριφθούν είτε να διορθωθούν ώστε να μπορούν να χρησιμοποιηθούν στη συνέχεια. Εάν είναι προτιμητέο να απορριφθεί ή όχι μια λύση εξαρτάται από το ποσοστό των απογόνων που θα είναι ‘ακατάλληλοι’. Δηλαδή εάν οι απόγονοι που παράγουν ‘ακατάλληλες’ λύσεις είναι η πλειοψηφία των απογόνων εάν αυτοί διαγραφούν ο πληθυσμός θα απαρτίζεται στην ουσία από μέρος του αρχικού πληθυσμού και οι γενιές δεν θα εξελίσσονται.

Μετάλλαξη

Η δεύτερη διαδικασία που χρησιμοποιείται για την επιλογή των απογόνων είναι η διαδικασία της μετάλλαξης. Η μετάλλαξη είναι η διαδικασία όπου ένα υποσύνολο από μεταβλητές (γονίδια) επιλέγονται τυχαία και οι τιμές τους αλλάζουν. Με τη μετάλλαξη πραγματοποιείται επιπλέον εξερεύνηση του χώρου αναζήτησης διότι δημιουργούνται εντελώς καινούρια άτομα. Έτσι αποτρέπεται ο γενετικός αλγόριθμος από το να παγιδευτεί σε κάποιο τοπικό μέγιστο ενώ παράλληλα διατηρεί την ποικιλομορφία του πληθυσμού του.

Γενικά, κατά τον σχεδιασμό ενός τελεστή μετάλλαξης πρέπει να εξασφαλιστεί ότι οι λύσεις κινούνται σε όλο το χώρο αναζήτησης, ότι παράγονται εφικτές λύσεις χωρίς οι αλλαγές που γίνονται να αλλάζουν εντελώς τη λύση παρά μόνο να επιφέρουν μια μικρή αλλαγή στην λύση. Άρα θα πρέπει να μπορεί να ελέγχεται το μέγεθος του τελεστή μετάλλαξης ώστε να μην καταστρέφει πολύ καλές λύσεις και γι’ αυτό τον λόγο συνήθως εφαρμόζεται σε ένα πολύ μικρό ποσοστό του πληθυσμού. Η μέγιστη τιμή του είναι 20% γιατί αλλιώς οδηγεί σε τυχαιοποιημένη αναζήτηση.

Πλεονεκτήματα γενετικών αλγορίθμων

Τα βασικότερα πλεονεκτήματα τα οποία προκύπτουν από τη χρήση των γενετικών αλγορίθμων, είναι τα παρακάτω (Λυκοθανάσης, 2001):

1. Έχουν τη δυνατότητα να επιλύουν δύσκολα προβλήματα γρήγορα και αξιόπιστα. Μάλιστα, η αποδοτικότητα των γενετικών αλγορίθμων αποτελεί ένα βασικό λόγο για τη χρησιμοποίησή τους. Μπορούν να αντιμετωπίσουν την επίλυση προβλημάτων με πολλές, δύσκολα

προσδιορισμένες λύσεις, ενώ ακόμη συναρτήσεις που παρουσιάζουν μεγάλες διακυμάνσεις, δε δημιουργούν πρόβλημα στη χρήση των γενετικών αλγορίθμων, σε σύγκριση με άλλες μεθόδους.

2. Είναι εύκολα επεκτάσιμοι και εξελίξιμοι, καθώς οι γενετικοί αλγόριθμοι δεν αντιστέκονται σε αλλαγές, επεκτάσεις και μετεξελίξεις, ανάλογα με την κρίση του σχεδιαστή.
3. Για τη βελτιστοποίηση με γενετικούς αλγορίθμους απαιτείται μόνο η γνώση της αντικειμενικής συνάρτησης και όχι πρόσθετων πληροφοριών. Αυτό σημαίνει πως δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται, σε αντίθεση με τις παραδοσιακές μεθόδους οι οποίες απαιτούν την ύπαρξη περιορισμών, με αποτέλεσμα να καθίστανται δύσκαμπτες και ακατάλληλες. Επίσης, οι γενετικοί αλγόριθμοι «επικοινωνούν» με το περιβάλλον τους μέσω της συνάρτησης καταλληλότητας, χωρίς να ενδιαφέρονται για τη σημασία της υπό εξέταση πληροφορίας, και με αυτό τον τρόπο εγγυάται η επιτυχία τους ανεξαρτήτως της σημασίας του προβλήματος, χωρίς όμως αυτό να σημαίνει πως δεν υπάρχουν άλυτα προβλήματα για τους γενετικούς αλγορίθμους.
4. Έχουν τη δυνατότητα να συνεργάζονται εύκολα με τα ήδη υπάρχοντα μοντέλα και συστήματα. Αυτό συμβαίνει καθώς οι γενετικοί αλγόριθμοι δίνουν ένα σημαντικό πλεονέκτημα, να μπορούν να χρησιμοποιηθούν με πρόσθετο τρόπο στα μοντέλα που χρησιμοποιούνται σήμερα, χωρίς να απαιτούν τον επανασχεδιασμό τους. Επίσης, μπορούν να συνεργαστούν με τους υπάρχοντες κώδικες, καθώς χρησιμοποιούν μόνο πληροφορίες της διαδικασίας ή της συνάρτησης που θα βελτιστοποιήσουν, χωρίς να ενδιαφέρει άμεσα ο ρόλος της στο σύστημα ή όλη δομή του συστήματος.
5. Έχουν τη δυνατότητα να συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους. Παρόλο που οι γενετικοί αλγόριθμοι έχουν μεγάλη ισχύ, σε ορισμένες περιπτώσεις προβλημάτων, δίνεται η δυνατότητα χρησιμοποίησης ενός υβριδικού σχήματος γενετικού αλγορίθμου με διαφορετική μέθοδο, κάτι το οποίο συμβαίνει χάρη στην ευελιξία που προσφέρουν οι γενετικοί αλγόριθμοι.
6. Μπορούν να εφαρμοστούν σε πολύ περισσότερα πεδία από οποιαδήποτε άλλη μέθοδο. Το χαρακτηριστικό μάλιστα που τους δίνει αυτή τη δυνατότητα είναι η ελευθερία επιλογής των κριτηρίων που καθορίζουν την επιλογή μέσα στο τεχνικό περιβάλλον. Κατ' αυτόν τον τρόπο, οι γενετικοί αλγόριθμοι μπορούν να χρησιμοποιηθούν σε πλήθος τομέων, όπως την οικονομία, το σχεδιασμό μηχανών ή ακόμα και την επίλυση μαθηματικών εξισώσεων.
7. Οι γενετικοί αλγόριθμοι αποτελούν τη μοναδική μέθοδο που εξερευνά το χώρο αναζήτησης και ταυτόχρονα εκμεταλλεύεται την ήδη επεξεργασμένη πληροφορία. Με το τυχαίο ψάξιμο γίνεται καλή εξερεύνηση του χώρου, αλλά δεν γίνεται εκμετάλλευση της πληροφορίας, ενώ με την αναζήτηση γίνεται καλή εκμετάλλευση της πληροφορίας, αλλά όχι καλή εξερεύνηση, δύο ανταγωνιστικά χαρακτηριστικά, τα οποία είναι επιθυμητό να συνυπάρχουν ταυτόχρονα. Οι γενετικοί αλγόριθμοι επιτυγχάνουν το βέλτιστο συνδυασμό εξερεύνησης και εκμετάλλευσης, κάτι το οποίο τους καθιστά αποδοτικούς και ελκυστικούς.

8. Έχουν από τη φύση τους το στοιχείο του παραλληλισμού. Ένας γενετικός αλγόριθμος, σε κάθε του βήμα επεξεργάζεται μεγάλες ποσότητες πληροφορίας, καθώς κάθε άτομο θεωρείται αντιπρόσωπος πολλών άλλων. Έχει υπολογιστεί ότι 10 άτομα αντιπροσωπεύουν περίπου 1000. Έτσι, μπορούν να καλύψουν μεγάλους χώρους σε μικρούς χρόνους.
9. Επιδέχονται παράλληλη υλοποίηση. Έχουν τη δυνατότητα να εκμεταλλευτούν τα πλεονεκτήματα των παράλληλων μηχανών, αφού εύκολα μπορούν να δεχτούν παράλληλη υλοποίηση. Το χαρακτηριστικό αυτό τους καθιστά ακόμη περισσότερο αποδοτικούς, ενώ σπανίως συναντάται στις παραδοσιακές μεθόδους.

4 Παρουσίαση του Προγράμματος

4.1 Python

Η Python είναι μια διερμηνευόμενη, (interpreted), αλληλεπιδραστική, (interactive), αντικειμενοστραφής, (object – oriented), ανοικτού πηγαίου κώδικα (open source), υψηλού επιπέδου γλώσσα προγραμματισμού. Δημιουργήθηκε από τον Guido van Rossum ως διάδοχος της γλώσσας ABC και κυκλοφόρησε για πρώτη φορά το 1991. Η φιλοσοφία σχεδίασής της δίνει έμφαση στην αναγνωσιμότητα, απλότητα και συνεκτικότητα του κώδικα αλλά και στην αύξηση της παραγωγικότητας του προγραμματιστή. Επιπλέον η χρήση κενών διαστημάτων, εσοχών, καθιστά τον κώδικα ξεκάθαρο και δομημένο με συνέπεια την αποφυγή συντακτικών λαθών και την ενίσχυση της προγραμματιστικής σκέψης καθώς επιτρέπει στον προγραμματιστή να συγκεντρώνεται στη λύση του προβλήματος παρά στην ίδια τη γλώσσα.

Η Python συνδυάζει σημαντική ισχύ με σαφή σύνταξη. Διαθέτει σύστημα δυναμικού τύπου και αυτόματη διαχείριση μνήμης. Έχει μια μεγάλη και περιεκτική, πρότυπη βιβλιοθήκη και μια ευρεία συλλογή τυπικών βιβλιοθηκών και πρόσθετων πακέτων. Χρησιμοποιεί ενότητες, (modules), τάξεις, (classes), εξαιρέσεις, (exceptions), καθώς και πολύ υψηλού επιπέδου δυναμικούς τύπους δεδομένων. Ενώ παράλληλα έχει τη δυνατότητα επέκτασης προσθέτοντας καινούρια modules γραμμένα σε μια γλώσσα που μεταγλωττίζεται όπως η C ή η C++. Τα modules αυτά μπορούν να ορίσουν νέες συναρτήσεις, (functions), μεταβλητές, (variables), τύπους δεδομένων, (object types). Υποστηρίζει πολλαπλά παραδείγματα προγραμματισμού, συμπεριλαμβανομένων των αντικειμενοστραφών, επιτακτικών, λειτουργικών και διαδικαστικών. Ενώ τέλος η ευελιξία της Python επιτρέπει στον προγραμματιστή να αναπτύξει της δεξιότητες του σε αρκετούς τομείς της πληροφορικής όπως η εξόρυξη δεδομένων, η δημιουργία εφαρμογών ιστού ή η εκμάθηση μηχανών. Καθώς χρησιμοποιείται από σχεδόν όλες τις διανομές της Linux, μεγάλα πανεπιστήμια αλλά και τη NASA.

4.2 Περιγραφή του Προγράμματος

4.2.1 Δεδομένα εισόδου

Τα δεδομένα εισόδου (input) για τον προτεινόμενο αλγόριθμο είναι οι συντεταγμένες, η χιλιομετρική θέση και το υψόμετρο εδάφους, των σημείων της οριζοντιογραφίας που θα χρησιμοποιηθούν για να χαραχθεί η ερυθρά της οδού και να σχεδιαστεί η μηκοτομή της. Τα δεδομένα αυτά πρέπει να είναι περασμένα σε αρχείο με δυο υποχρεωτικές και μια προαιρετική στήλες. Η πρώτη στήλη περιέχει τις χιλιομετρικές θέσεις των σημείων, η δεύτερη τα υψόμετρα του έδαφος των σημείων και η προαιρετική στήλη την ονομασία κάθε σημείου.

Επίσης εισάγονται η ταχύτητα μελέτης και η κατηγορία της οδού καθώς και ο τύπος του ανάγλυφου του εδάφους όταν η κατηγορία της οδού είναι ‘Α’. Οι παράμετροι αυτοί καθορίζουν την επιλογή των μέγιστων παραμέτρων της κατά μήκους κλίσης και των ακτινών της κατακόρυφης καμπύλης όπως αυτές ορίζονται στο τεύχος ‘Οδηγίες Μελετών Οδικών Έργων Χαράξεις (ΟΜΟΕ – Χ)’. Η μέγιστη κατά μήκος κλίση και οι μέγιστες ακτίνες χρησιμοποιούνται για να εισάγουν τους απαραίτητους περιορισμούς κατά τη χάραξη της οδού όπως αυτοί ορίζονται από τη νομοθεσία. Εκτενέστερη ανάπτυξη των περιορισμών θα γίνει σε επόμενο κεφάλαιο.

4.2.2 Δεδομένα εξόδου

Κατά τη διάρκεια των υπολογισμών στο αντίστοιχο παράθυρο του προγράμματος εμφανίζεται η χρονική εξέλιξη της διαδικασίας αλλά και τα βέλτιστα, σύμφωνα με το πρότυπο, αποτελέσματα κάθε διαδικασίας. Επιπλέον τα παραπάνω αποθηκεύονται και σε αρχείο με όνομα ίδιο με του αρχείου που εισήχθη και περιέχει τις συντεταμένες της οδού. Δίνεται επίσης η δυνατότητα παραγωγής αρχείου dxf με την ερυθρά της επιλογής του ερευνητή. Σκόπιμο είναι να αναφερθεί ότι το πρόγραμμα παράγει τρεις διαφορετικές εκδοχές της μηκοτομής της υπό μελέτη οδούς για κάθε διαφορετικό αριθμό σημείων θλάσεως της μηκοτομής. Σε επόμενο κεφάλαιο θα αναφερθούν τα κριτήρια επιλογής του αριθμού θλάσεων της μηκοτομής.

4.2.3 Αντικειμενική συνάρτηση

Η αντικειμενική συνάρτηση ή αλλιώς συνάρτηση στόχου είναι βασικό συστατικό της βελτιστοποίησης καθώς με βάση αυτή συγκρίνονται τα αποτελέσματα των επαναληπτικών δοκιμών και επικρατεί η βέλτιστη λύση.

Στη παρούσα διπλωματική εργασία η παράμετρος με την οποία συγκρίνονται οι λύσεις είναι το άθροισμα των εμβαδών των επιχωμάτων και ορυγμάτων. Πιο αναλυτικά στην πρώτη δοκιμή το άθροισμα των εμβαδών συγκρίνεται με τη δέκατη δύναμη του δέκα (10^1). Αν το εμβαδό που υπολογιστεί από τη συνάρτηση είναι μικρότερο από δυο επί τη δέκατη δύναμη του δέκα (10^1) εκχωρείται στις αντίστοιχες μεταβλητές όπως και οι συντεταγμένες του.

```
if fabs(trial_epos) + fabs(trial_eneg) < fabs(epos) + fabs(eneg):
    yers, xers = trial_yers, trial_xers
    epos, eneg =trial_epos, trial_eneg
```

Αλγόριθμος συνάρτησης στόχου

Κάθε φορά που επαναλαμβάνεται η δοκιμή ελέγχονται τα εμβαδά αν το άθροισμα των εμβαδών της δοκιμής είναι μικρότερο από την τιμή της αντίστοιχης μεταβλητής εκχωρείται σε αυτήν όπως και οι

συντεταγμένες της αντίστοιχης ερυθράς της οδού από την οποία υπολογίστηκε εκχωρούνται στις αντίστοιχες μεταβλητές. Στο τέλος της διαδικασίας η κύρια συνάρτηση, main(), εμφανίζει το βέλτιστο εμβαδό, τις κορυφές της ερυθράς και την κλίση, ως λίστες, καθώς και ένα σκαρίφημα.

4.2.4 Χάραξη

Το πρότυπο χαράσσει ευθείες γραμμές με τύπο $(y - y_0) = \frac{y_1 - y_0}{x_1 - x_0} * (x - x_0)$. Στόχος της έρευνας είναι να βρεθούν τα σημεία από τα οποία διέρχονται οι βέλτιστες ευθείες ώστε το άθροισμα των χωματισμών, επιχώματα και ορύγματα, να είναι το ελάχιστο δυνατό. Στη συνέχεια θα περιγραφούν τα επιμέρους στάδια του προτύπου. Αρχικά θα αναφερθούν οι περιορισμοί που εισάγονται από τους ισχύοντες κρατικούς κανονισμούς ,στη συνέχεια θα περιγραφούν η κατασκευή της αντικειμενικής συνάρτησης, η βελτιστοποίηση με τη μέθοδο Monte Carlo και τέλος η βελτιστοποίηση με τη μέθοδο των γενετικών αλγορίθμων.

Εισαγωγή Προδιαγραφών Οδοποιίας

Σύμφωνα με τις προδιαγραφές χάραξης που ορίζονται η μέγιστη κατά μήκος κλίση εξαρτάται από την κατηγορία της οδού και την ταχύτητα μελέτης. Οι επιτρεπόμενες τιμές αναφέρονται στους 3παρακάτω πίνακες.

V_e [km/h]	s_{max} [%] για τις οδούς της ομάδας			
	A			B (πλήν BI)
	πεδινά εδάφη	λοφώδη εδάφη	ορεινά εδάφη	όλες οι κατηγορίες εδαφών
50	7 (8)	8 (9)	10 (11)	8 (12)
60	6 (8)	7 (9)	9 (10)	7 (10)
70	5 (7)	6 (8)	8 (9)	6 (9)
80	4 (6)	5 (7)	7 (9)	5 (7)
90	4 (5)	5 (6)	7 (8)	-
100	3 (5)	4 (6)	6 (8)	-
110	3 (5)	4 (6)	5 (6)	-
120	3 (5)	4 (6)	-	-
130	3 (4)	-	-	-

Οι τιμές σε ( ) εμφανίζονται σε εξαιρετικές περιπτώσεις

Πινακας 1: Μέγιστες κατά μήκος κλίσεις Πηγή :ΟΜΟΕ - Χ

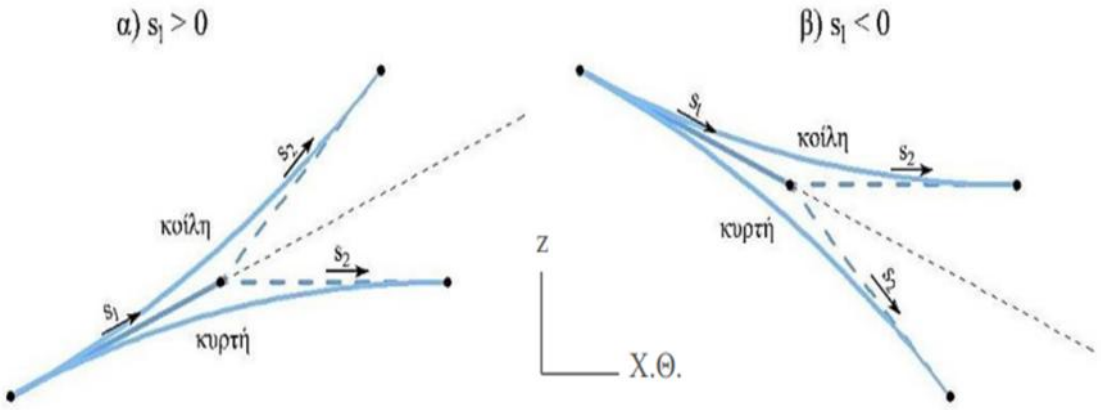
V_e [km/h]	S_{max} [%]	Μέγιστο μήκος L_{max} [m]
50	8 (12)	100
60	7	200
70	6	300
80	5	600

Πίνακας 2 : Μέγιστες κατά μήκος κλίσεις οδών κατηγορίας ΒΙ (αστικών αυτοκινητόδρομων)
Πηγή : ΟΜΟΕ - Χ

Για την εφαρμογή του περιορισμού της κατά μήκος κλίσης ορίστηκε η συνάρτηση ποινής κλίσης. Εάν το πρόγραμμα επιλέξει δύο σημεία τα οποία ορίζουν ευθεία με κλίση μεγαλύτερη από την επιτρεπόμενη στην αντικειμενική συνάρτηση προστίθεται επιπλέον $30.0 * \Delta x * |s_m - s_i| / s_m$

όπου Δx είναι η διαφορά μεταξύ δύο διαδοχικών θλασεων ερυθράς, s_m η μέγιστη επιτρεπόμενη κατά μήκος κλίση, s_i η κατά μήκος κλίση του κλάδου i της ερυθράς.

Από τους κανονισμούς επίσης ορίζεται η ακτίνα της κατακόρυφης καμπύλης Η. Η καμπύλη που εγγράφεται στη γωνία που σχηματίζουν δύο διαδοχικοί κλάδοι της ερυθράς. Η καμπύλη που θα χαραχθεί μπορεί να είναι κυρτή ή κοίλη. Η μορφή της καμπύλης σε σχέση με τη κατά μήκος κλίση είναι η ακόλουθη:



Εικόνα 14: Διάκριση μεταξύ κυρτών και κοίλων καμπυλών. Πηγή :Μπουτσάκης (2013)

a) Εάν το πρώτο ευθύγραμμο τμήμα έχει θετική κλίση **s1** (ανωφέρεια):
   - Αν **s2 > s1**, τότε η καμπύλη είναι κοίλη.

- Αν **s2 < s1**, τότε η καμπύλη είναι κυρτή.

b) Εάν το πρώτο ευθύγραμμο τμήμα έχει αρνητική κλίση **s1** (κατωφέρεια):

- Αν **s2 > s1**, τότε η καμπύλη είναι κοίλη.
- Αν **s2 < s1**, τότε η καμπύλη είναι κυρτή.

Επομένως, σε κάθε περίπτωση ισχύει ότι η καμπύλη είναι **κυρτή εφόσον *s2 > s1***, και **κοίλη εφόσον *s2 < s1***.(Μπουτσάκης 2013)

Η ακτίνα της καμπύλης που θα χαραχθεί ορίζεται από τους κανονισμούς της ΟΜΟΕ – Χ στους παρακάτω πίνακες.

V_e [km/h]	$H_w$min [m]
50	1.350
60	1.900
70	2.500
80	3.300
90	4.200
100	5.200
110	6.300
120	7.500
130	10.000

Πίνακας 3: Ελάχιστες τιμές ακτινών κοίλων κατακόρυφων καμπυλών συναρμογής για οδούς των ομάδων Α και Β

V_e [km/h]	Ορατότητα για στάση	Ορατότητα για προσπέραση	
	Επιτρεπόμενη περιοχή H_K Διατίθεται το απαραίτητο μήκος ορατότητας για στάση [m]	Αποφευκτέα περιοχή H_K Δεν διατίθεται το απαραίτητο μήκος ορατότητας για προσπέραση [m]	Επιτρεπόμενη περιοχή H_K Διατίθεται το απαραίτητο μήκος ορατότητας για προσπέραση [m]
1	2	3	4
Οδοί ενιαίας επιφάνειας κυκλοφορίας και διαχωρισμένης με			
50	800 έως 7.000	7.000 έως 27.000	≥ 27.000
60	2.000 έως 7.600	7.600 έως 30.000	≥ 30.000
70	3.000 έως 8.400	8.400 έως 34.000	≥ 34.000
80	4.500 έως 10.000	10.000 έως 40.000	≥ 40.000
90	6.200 έως 12.000	12.000 έως 48.000	≥ 48.000
100	8.500 έως 14.000	14.000 έως 55.000	≥ 55.000
Οδοί διαχωρισμένων επιφανειών κυκλοφορίας με Α.Κ.			
60	3.000 έως ∞	–	–
70	4.500 έως ∞	–	–
80	6.200 έως ∞	–	–
90	8.500 έως ∞	–	–
100	11.000 έως ∞	–	–
110 έως 130	15.000 έως ∞	–	–

Προσοχή : Μετά την επιλογή τιμών από τον πίνακα και πριναπό την ορτιστική υιοθέτησή τους, πρέπει οπωσδήποτε να επαληθεύεται ότι αυτές ικανοποιούν όλες τις απαιτήσεις ορατότηας όπως αυτές διατυπώνονται στους ισχύοντες κανονισμούς ΟΜΟΕ – Χ §10.3

Πίνακας 4: Οριακές τιμές ακτινών κυρτών κατακόρυφων καμπυλών συναρμογής για οδούς των ομάδων Α και Β. Πηγή : ΟΜΟΕ – Χ

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο στη γωνία της θλάσης της μηκοτομής χαράσσεται τετραγωνική παραβολή σύμφωνα με τους τύπους που αναφέρθηκαν. Συνήθως σε κάθε οδικό τμήμα που πραγματοποιείται μελέτη χάραξης της μηκοτομής χαράσσονται περισσότερες από μια

καμπύλες ανάλογα βέβαια και με τις ανάγκες της οδού. Σε περίπτωση που θα χαραχθούν περισσότερες από μια καμπύλες πρέπει να εξασφαλίζεται η συνέχεια και η ομαλή χάραξη της οδού. Για να εξασφαλιστεί αυτό δημιουργήθηκε η συνάρτηση ποινής επικάλυψης η οποία στη συνάρτηση στόχου προσθέτει σε περίπτωση υπερκάλυψης διαδοχικών καμπύλων επιπλέον $30 * t_{i+1} + t_i * \Delta x$ όπου t_i το μήκος της εφαπτομένης της κατακόρυφης καμπύλης και Δx είναι η διαφορά μεταξύ δύο διαδοχικών θλασεων ερυθράς.

Είναι επιπλέον σημαντικό να αναφερθεί πως το παρόν πρότυπο χαράσσει την ερυθρά της οδού για τον δυνατό αριθμό σημείων θλάσης. Το κριτήριο που χρησιμοποιεί για την εύρεση του αριθμού των σημείων θλάσης είναι $L //500$, όπου L είναι το συνολικό μήκος της οδού σε μέτρα , το σύμβολο $//$ συμβολίζει την ακέραια διαίρεση. Η Παραπάνω θεώρηση έγινε λαμβάνοντας υπόψη ότι ‘ Όταν στην Ερυθρά περιέχονται δύο διαδοχικές κατακόρυφες καμπύλες προς την ίδια κατεύθυνση που χωρίζονται από ένα σύντομο ευθύγραμμο τμήμα τότε οι καμπύλες ονομάζονται Broken-Back Curves (κατεστραμμένες κατακόρυφες καμπύλες). Οι καταστραμμένες κατακόρυφες καμπύλες πρέπει να αποφεύγονται όπου είναι δυνατό. Αποστάσεις μικρότερες από 1500 ft (500 m) μεταξύ των κατακόρυφων σημείων τομής θεωρείται ότι είναι κατεστραμμένες κατακόρυφες (broken back)’. (3Bureau Of Design And Environment Manual, Illinois, 2010). Η παραδοχή αυτή χρησιμοποιήθηκε μόνο για τον ορισμό του πλήθους θλάσεων της ερυθράς της οδού και δεν εξασφαλίζει την απόσταση μεταξύ δύο διαδοχικών καμπύλων η οποία καθορίζεται τυχαία και ελέγχεται από την συνάρτηση ποινής υπερκάλυψης που ορίζεται σε επόμενο κεφάλαιο. Η διατήρηση του παραπάνω περιορισμού θα επιβάρυνε σημαντικά την ταχύτητα εκτέλεσης του προτύπου. Επιπλέον ο ορισμός του αριθμού θλάσεων με βάση το κριτήριο που αναφέρθηκε θεωρείται ότι καλύπτει την απαίτηση η χάραξη της ερυθράς της οδού να μην προσομοιάζει το φαινόμενο του λουνα-παρκ. Δεν έγινε ωστόσο περαιτέρω διερεύνηση.

4.2.5 Βελτιστοποίηση

Η κατασκευή ενός δρόμου είναι μια εξαιρετικά περίπλοκη διαδικασία. Καθώς από ένα σύνολο άπειρων λύσεων επιλέγεται μοναδική λύση η οποία αποτελεί συγκερασμό αλληλοσυγκρουόμενων μεγεθών. Εκτός δηλαδή του εύρους της επιφάνειας που δίνεται για την κατασκευή του δρόμου η οποία περιορίζεται από τους ισχύοντες κανονισμούς, υπεισέρχονται και άλλα περιβαλλοντικά, κοινωνικοοικονομικά και κατασκευαστικά κριτήρια τα οποία επηρεάζουν και πολλές φορές δυσχεραίνουν το σχεδιασμό. Η επιλογή δε της βέλτιστης λύσης εναπόκειται στον μελετητή.

Η δυσκολία λοιπόν της πραγμάτωσης μιας οδικής μελέτης καθώς και η ανάπτυξη της τεχνολογίας είχε ως αποτέλεσμα την εύρεση διάφορων υπολογιστικών μεθόδων βελτιστοποίησης.

Monte Carlo

Για τη δημιουργία ένα σύνολο τυχαίων εισόδων, x_{i1}, x_{i2}, …, x_{in} και y_{i1}, y_{i2}, ..., y_{in} χρησιμοποιείται μια ενσωματωμένη βιβλιοθήκη της python για την παραγωγή ψευδοτυχαίων αριθμών γεγονός που αποτελεί καινοτομία του προγράμματος. Καθώς η εύρεση της χιλιομετρικής θέσης και του υψομέτρου αλλαγής κλίσεως δεν γίνεται από τον χρήστη αλλά από το ίδιο το πρόγραμμα χρησιμοποιώντας τη βιβλιοθήκη **random**. Η βιβλιοθήκη αυτή δημιουργεί ψευδοτυχαίες γεννήτριες αριθμών είτε από ομοιόμορφη επιλογή από μια περιοχή για ακέραιους αριθμούς είτε από μια κατανομή για ακολουθίες (όπως ομοιόμορφη, κανονική (Gaussian), lognormal, αρνητικού εκθέτη, γάμμα και βήτα) και παράγει ένα τυχαίο δεκαδικό αριθμό σε εύρος [0.0, 1.0) χρησιμοποιώντας τη Mersenne Twister ως γεννήτρια πυρήνα, όλες οι συναρτήσεις της βασίζονται σε αυτή.

Στο πρότυπο χρησιμοποιείται συνάρτηση uniform() της βιβλιοθήκης random(random.uniform()). Από τη random.uniform() επιστρέφεται ένας δεκαδικός από ένα κλειστό ή ανοιχτό διάστημα. Τα διαστήματα επιλογής που δίνονται είναι αυτό της χιλιομετρικής θέσης που δόθηκε μέσω του αρχείου, δηλαδή από την οριζόντια συντεταγμένη αρχής έως την οριζόντια συντεταγμένη του τέλους της οδού, για την οριζόντια συντεταγμένη. Ενώ για τη κατακόρυφη συντεταγμένη να απέχει τριάντα (30 m) μέτρα από το υψόμετρο του εδάφους που έχει η αντίστοιχη οριζόντια συντεταγμένη. Για να υπολογιστεί το υψόμετρο του εδάφους, εάν αυτό δεν ανήκει στο αρχείο που εισήχθη στο πρόγραμμα στην αρχή, χρησιμοποιήθηκε η συνάρτηση γραμμικής παρεμβολής που έχει συνταχθεί.

Δοκιμαστικά ζητήθηκε από τη συνάρτηση η αναζήτηση πέντε σημείων με μοναδική συνθήκη ελέγχου οι οριζόντιες συντεταγμένες να απέχουν μεταξύ τους είκοσι (20 m) μέτρα. Αφού βρεθούν οι οριζόντιες συντεταγμένες ταξινομούνται με αύξουσα σειρά έπειτα υπολογίζονται οι αντίστοιχες κατακόρυφες συντεταγμένες. Σκόπιμο είναι να αναφερθεί πως η ταξινόμηση των οριζόντιων συντεταγμένων είναι απαραίτητο να γίνει πριν την εύρεση των κατακόρυφων διότι δεν υπάρχει τρόπος να γίνει ταξινόμηση σε αυτή τη λίστα. Αλλά και ότι θα μπορούσε η λίστα των **yers** να προκύπτει από το διάστημα που ορίζεται από τις ευθείες

$$(y_i - y_{i-1}) = \pm s_m * (x_i - x_{i-1}) .$$

Όπου:

- y_i η προς αναζήτηση κατακόρυφη συντεταγμένη
- (x_0, y_0) είναι το αρχικό σημείο της οδού και δίνεται από το χρήστη
- (x_{-1}, y_{-1}) είναι το τελικό σημείο της οδού και δίνεται από τον χρήστη
- x_i είναι η οριζόντια ψευδοτυχαία συντεταγμένη η οποία επιλέγεται από τη συνάρτηση uniform() της βιβλιοθήκης random της Python

- $\pm s_m$ το διάστημα που ορίζεται από την επιτρεπόμενη από τους κανονισμούς κλίση

Η παραπάνω λύση απορρίφθηκε για να ενισχυθεί η τυχαιότητα. Ωστόσο για τη διασφάλιση της τήρησης των κανονισμών εισήχθη η συνάρτηση ποινής κλίσης που αναφέρθηκε παραπάνω.

Επιπλέον το μήκος της τετραγωνικής παραβολής που ισούται με το διπλάσιο του μήκους της εφαπτόμενης του κύκλου που εγγράφεται στη γωνία θλάσης της μηκοτομής, 2 * T, συχνά είναι μεγαλύτερο από είκοσι μέτρα που εισάγαμε δοκιμαστικά στον αλγόριθμο. Γι' αυτό συντάχθηκε καινούριος έλεγχος ο οποίος αφού δημιουργήσει μια λίστα με τις ψευδοτυχαίες τιμές της οριζόντιας συντεταγμένης των σημείων θλάσης της ερυθράς της οδού υπολογίζει το μήκος της εφαπτομένης, T, και ελέγχει αν το μήκος των εφαπτόμενων δυο διαδοχικών κύκλων που εγγράφονται στις γωνίες θλάσης της μηκοτομής, $T_i + T_{i+1}$, είναι μικρότερα από την απόσταση δύο σημείων θλάσεως της μηκοτομής ($T_i + T_{i+1} \leq x_{i+1} - x_i$). Αν αυτός ο έλεγχος δεν ικανοποιείται στο άθροισμα των εμβαδών προστίθεται το αποτέλεσμα της συνάρτησης ποινής υπερκάλυψης που αναφέρθηκε παραπάνω.

Παραπάνω ορίστηκε το μαθηματικό πρότυπο που είναι η συνάρτηση στόχου και οι στοχαστικές είσοδοι x, y που αποτελούν τα πρώτα βήματα του αλγορίθμου Monte Carlo όπως αναφέρθηκαν σε προηγούμενο κεφάλαιο. Στη συνέχεια θα αναφερθούμε στο τέταρτο βήμα του αλγορίθμου στον ορισμό δηλαδή του πλήθους επαναλήψεων. Διερευνήθηκαν οι τιμές της συνάρτησης στόχου για πλήθος επαναλήψεων ίσο με 2 * N * K όπου δύο είναι το πλήθος των συντεταγμένων x, y. N είναι το πλήθος των βαθμών ελευθερίας δηλαδή των θλάσεων της ερυθράς και K παράμετρος η οποία θα διερευνηθεί. Ως τιμές της παραμέτρου K ορίζονται οι τιμές διακόσια δυο χιλιάδες και είκοσι χιλιάδες. Η διερεύνηση του πλήθους επαναλήψεων παρουσιάζεται παρακάτω με τη μορφή πινάκων.

Δοκιμές	Ταχύτητα	Σημεία Θλάσης	Ελάχιστο	Μέσος Όρος	Πλήθος	Τυπική Απόκλιση
200	50	1	16790.38	16979.33	51	178.82
		2	16438	17153.57	51	391.13
		3	16184.91	17474.24	51	477.84
		4	12745.95	17883.89	51	1267.86
		5	16725.73	19059.66	51	1265.01
		6	17273.45	19958.22	51	1764.6
		7	17671.24	21384.69	51	1756.15
		8	17046.81	23289.13	51	2452.21
		9	20252.02	25886.48	51	2671.82
	100	1	16784.52	17001.21	74	179.34
		2	16504.48	17077.56	74	264.74
		3	15830.15	17444.49	71	700.52
		4	16725.14	18393.62	64	830.36
		5	16570.59	20177.77	55	1682.51
		6	19962.59	23593.77	55	2004.03
		7	21311.54	27961.55	55	3853.8
		8	13414.41	34775.04	66	10163.54
		9	29575.43	49168.36	55	9214.9

Δοκιμές	Ταχύτητα	Σημεία Θλάσης	Ελάχιστο	Μέσος Όρος	Πλήθος	Τυπική Απόκλιση
2000	50	1	16782.75	16816.79	68	24.92
		2	15957.05	16607.59	39	234.62
		3	15541.55	16544.77	12	499.67
		4	15111.38	16630.49	12	545.83
		5	15534.86	16552.63	12	690.08
		6	12419.77	15738.96	50	1405.23
		7	15286.34	17371	8	1283.26
		8	16631.67	19385.4	8	1726.07
		9	17708.81	20135.26	8	1572.22
	100	1	16784.25	16818.5	51	31.31
		2	16479.54	16767.93	51	125.24
		3	15869.39	16754.59	51	312.2
		4	16149.27	17039.37	51	414.27
		5	15568.3	17543.29	51	832.59
		6	16276.76	19209.98	51	1205.56
		7	17161.22	22517.73	51	2458.33
		8	17375.89	26644.53	51	3299.11
		9	25492.6	35612.05	18	5523.02

Δοκιμές	Ταχύτητα	Σημεία Θλάσης	Ελάχιστο	Μέσος Όρος	Πλήθος	Τυπική Απόκλιση
20000	50	1	16782.87	16785.83	50	3.84
		2	15827.44	16274.15	50	211.94
		3	14732.46	15752.93	50	337.86
		4	13056.31	14632.39	50	764.35
		5	12543.62	14595.89	40	810.09
		6	13296.9	15100.59	12	846.59
		7	15943.16	16375.08	2	431.92
	100	1	16783.41	16786.03	50	2.03
		2	16440.13	16572.35	50	63.72
		3	15754.23	16212.67	50	224.74
		4	15587.5	16043.41	16	218.64
		5	13037.04	15078.69	51	869.21
		6	13257.83	15521.91	45	1032.13
		7	17256.94	18559.86	4	1176.78
		8	21319.69	21745.94	4	296.34
		9	24560.74	28268.33	4	2176.78

Από τις δοκιμές που έγιναν παρατηρείται ότι η μέθοδος είναι αποδοτική για μικρές ταχύτητες και μικρό αριθμό βαθμών ελευθερίας. Παρατηρείται ότι με αυτές τις συνθήκες η μέθοδος παρέχει περίπου ίσα αποτελέσματα ανεξαρτήτως της τιμής που λαμβάνει η παράμετρος K. Μικρές τιμές της παραμέτρου K συντελούν σε σημαντική μείωση του χρόνου εκτέλεσης του αλγορίθμου. Επιπλέον παρατηρείται ότι με την αύξηση των δοκιμών μειώνεται σημαντικά η διασπορά του δείγματος.

Για περισσότερους βαθμούς ελευθερίας η μέθοδος είναι εξίσου αποδοτική καθώς η τιμή της συνάρτησης στόχου μειώνεται όσο αυξάνονται οι βαθμοί ελευθερίας εντούτοις συνήθως απαιτούνται περισσότερες επαναλήψεις, δηλαδή μεγαλύτερες τιμές της παραμέτρου K, παρατηρείται επίσης μείωση της διασποράς με την αύξηση του αριθμού των δοκιμών. Η μέθοδος μάλιστα για τιμές βαθμού ελευθερίας από τρία έως έξι εμφάνισε σημαντικά μικρότερη τιμή της συνάρτησης στόχου.

Τέλος για πολλούς βαθμούς ελευθερίας η μέθοδος δεν ενδείκνυται. Γι’ αυτές τις τιμές η συνάρτηση στόχου αυξήθηκε σημαντικά παρόλο που η αύξηση των βαθμών ελευθερίας επιτρέπει στην ερυθρά να προσεγγίσει καλύτερα την γραμμή εδάφους και η συνάρτηση στόχου θα έπρεπε να μειώνεται. Επίσης αυξάνεται η τιμή της τυπικής απόκλισης άρα και η διασπορά του δείγματος καθώς και ο απαιτούμενος χρόνος εκτέλεσης του αλγορίθμου.

Για να μειωθεί ο απαιτούμενος χρόνος χρησιμοποιήθηκε η βιβλιοθήκη multiprocessing της Python. Το σύνολο εντολών multiprocessing της python υποστηρίζει την αναδημιουργία διαδικασιών χρησιμοποιώντας Διεπαφή Προγραμματιστικών Εφαρμογών (Application Programming Interface). Η

Διεπαφή Προγραμματιστικών Εφαρμογών είναι η δυνατότητα που παρέχει ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή προκειμένου να επιτρέπει να γίνονται αιτήσεις από άλλα προγράμματα ή/και ανταλλαγή δεδομένων προς αυτό. Με την εντολή multiprocessing παρακάμπτεται το κλείδωμα του καθολικού διερμηνέα (Global Interpreter Lock) ο οποίος φροντίζει να εκτελείται μόνο μία διεργασία Python τη φορά. Το κλείδωμα του διερμηνέα διευκολύνει τον διαμοιρασμό των εργασιών σε υποεπεξεργαστές, δυνατότητα που παρέχεται από υπολογιστικά συστήματα πολλαπλών επεξεργαστών. Εξαιτίας αυτού, η μονάδα πολλαπλών επεξεργασιών επιτρέπει στον προγραμματιστή να αξιοποιήσει πλήρως πολλούς επεξεργαστές σε ένα συγκεκριμένο μηχάνημα.

Η διαδικασία Pool της ενότητας multiprocessing δίνει την δυνατότητα του παραλληλισμού καθώς επιτρέπει την ταυτόχρονη εκτέλεση μιας συνάρτησης με διαφορετικές τιμές εισόδου, διανέμοντας τα δεδομένα εισόδου στις διεργασίες (παραλληλισμός δεδομένων). Κατά την εκτέλεση του προγράμματος με την συνάρτηση cpu_count() μετριούνται οι υποεπεξεργαστές και το πηλίκο της ακέραιης διαίρεσης των καθορισμένων επαναλήψεων δια του αριθμού των υποεπεξεραστών μοιράζεται σε κάθε πυρήνα. Η Pool() διαμοιράζει τα δεδομένα ώστε να ξεκινήσουν οι παράλληλες διαδικασίες εύρεσης των σημείων.

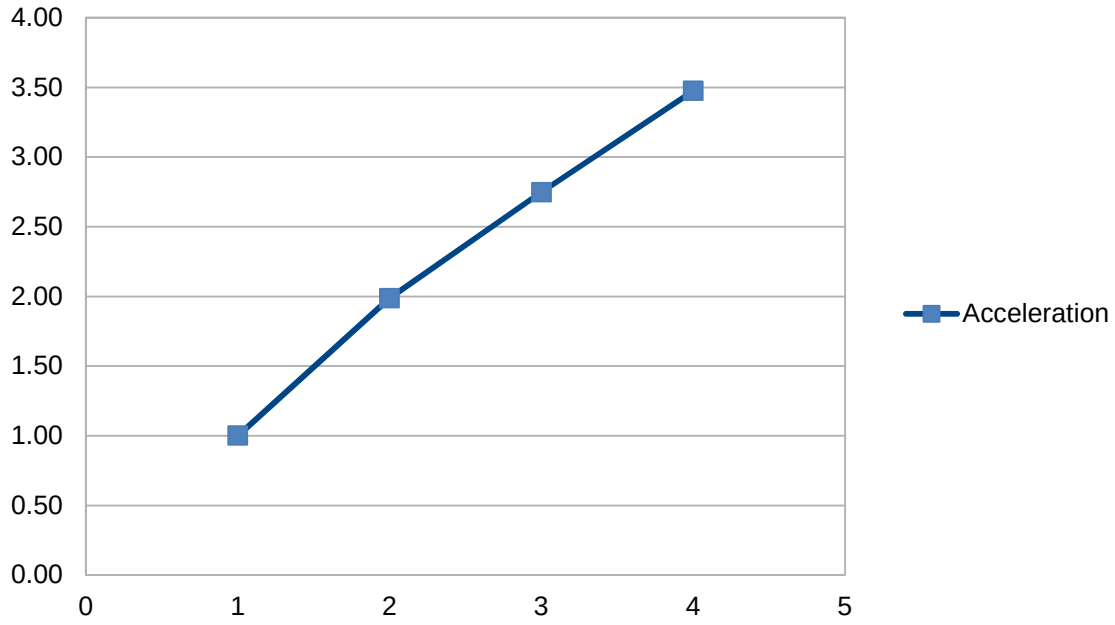
```
def monteCarlo(xth, y, a):
    """Using Monte Carlo find optimum grade points."""
    Point.setGround(xth, y)
    ntries = a*2 * ntriesperdim
    pomin = Point(a)

    nprocesses = multiprocessing.cpu_count()
    ntries1 = ntries//nprocesses
    pool = multiprocessing.Pool(nprocesses)

        argslist = [(a, ntries1)] * nprocesses
    for po1 in pool.imap_unordered(doSome, argslist):
        if po1.emb < pomin.emb:
            pomin = po1
    return pomin
```

Αλγόριθμος : multiprocessing

Η ταχύτητα εκτέλεσης του προγράμματος βελτιώνεται αισθητά μετά από τον διαμοιρασμό των επαναλήψεων. Η εκτέλεση του προγράμματος σε έναν υπολογιστή Intel I7, με 4 πυρήνες (υποεπεξεργαστές) ήταν τρεισήμισι (3,47)φορές ταχύτερη. Για την πραγματοποίηση διακόσιων επαναλήψεων η μέθοδος διήρκησε τρεισήμισι λεπτά έναντι έντεκα με τη χρήση ενός υποεπεξεργαστή. Η επιτάχυνση του αλγορίθμου φαίνεται στο διάγραμμα



Σχήμα 4: Επιτάχυνση αλγορίθμου με τη χρήση της εντολής multiprocessing

Επιπλέον για να επιταχυνθεί η διαδικασία διερευνήθηκε η εύρεση της κατακόρυφης συντεταγμένης y των σημείων θλάσεως της ερυθράς με τον 'Κανόνα του Crammer'. Σύμφωνα με τον κανόνα του Crammer αν θεωρήσουμε ένα γραμμικό σύστημα της μορφής [A] *[X] = [B], όπου A τετραγωνικός και αντιστρέψιμος πίνακας , τότε το σύστημα έχει πάντα λύση της μορφής $[A]^T * [B] = [A]^T * [A] * [X]$

Αποδεικνύεται δηλαδή πως αν οι επιμέρους κλάδοι της ερυθράς της οδού παρασταθούν από ευθεία με εξίσωση $y = ax + b$ ότι η βέλτιστη γραμμή, δηλαδή που ελαχιστοποιεί τις διαφορές των y της ερυθράς της οδού από το y του εδάφους, και ελαχιστοποιεί τα χωματουργικά υπολογίζεται πολλαπλασιάζοντας την μητρωική εξίσωση επί τον ανάστροφο του πίνακα [A].

Υποθέτουμε σημείο u_1, όπου u_1 σημείο εντός της οδού που δόθηκε προς διερεύνηση και u_1 σημείο τομής δύο ερυθρών γραμμών. Ειδικά για $x=u_1$ και λόγω της συνέχειας της ερυθράς γραμμής ισχύει:

$$y = a_1u_1 + b_1 = a_2u_1 + b_2 \Rightarrow b_2 = a_1u_1 + b_1 - a_2u_1$$

Θέτοντας k τον αριθμό για τον οποίο ισχύει:

$$x_i \leq u_1, 1 \leq i \leq k$$

$$x_i > u_1, k < i \leq N$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} x_1 & 1 & 0 \\ x_2 & 1 & 0 \\ \vdots & & \\ x_k & 1 & 0 \\ u_1 & 1 & x_{k+1} - u_1 \\ u_1 & 1 & x_{k+2} - u_1 \\ \vdots & \vdots & \vdots \\ u_1 & 1 & x_N - u_1 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ \vdots \\ a_n \\ a_{n+1} \end{bmatrix} \quad \text{ή} \quad [B] = [A] \cdot [X]$$

Η παραπάνω μητρωική εξίσωση λύνεται με βάση το κανόνα του Crammer. Όμως η μεθοδολογία αυτή προϋποθέτει την γνώση της παραμέτρου u, στο πρόγραμμα τέθηκε u η λίστα με τις χιλιομετρικές θέσεις του εδάφους που εισήχθη στην αρχή του προγράμματος. Ως x τέθηκαν τυχαίες συντεταγμένες που βρέθηκαν μέσω του τύπου

x = random.rand(n) * L + u_1 + 10

Η συνάρτησης random.rand(δημιουργεί μια λίστα με πλήθος n που ορίζεται από τον χρήστη και εισάγει τυχαίους αριθμούς από το διάστημα [0, 1) σύμφωνα με τη λογική της βιβλιοθήκης random που αναφέρθηκε παραπάνω. u1 είναι η αρχή της οδού και L το συνολικό μήκος της οδού. Ακολουθώντας τον κανόνα του Crammer υπολογίζεται ερυθρά για τα τυχαία x .Η τιμή που δίνει τον ελάχιστο όγκο V χωματουργικών είναι η βέλτιστη.

Γενετικοί Αλγόριθμοι

Δημιουργία αρχικού πληθυσμού

Το πρώτο βήμα της υπολογιστικής διαδικασίας είναι η κατασκευή του αρχικού πληθυσμού δηλαδή της πρώτης γενιάς των υποψήφιων λύσεων. Το μέγεθος του πληθυσμού, συμβολίζεται στο πρόγραμμα με τη μεταβλητή nAnim είτε εισάγεται από τον χρήστη είτε υπολογίζεται από το πρόγραμμα.

Η κατασκευή του μεγέθους του αρχικού πληθυσμού από το πρόγραμμα γίνεται από τη συνάρτηση nAnimCompute() χρησιμοποιώντας την παράμετρο nValmax, η οποία επίσης μπορεί να εισαχθεί από τον χρήστη, τον αριθμό των χρωμοσωμάτων που είναι είκοσι έξι (26), και το πλήθος των ατόμων ανά δυαδικό χρωμόσωμα που ισούται με πέντε (5). Σε περίπτωση που το μέγεθος του πληθυσμού εισαχθεί από τον χρήστη και είναι μικρότερο από είκοσι (20) άτομα ορίζεται μέγεθος πληθυσμού nAnim ίσο με είκοσι.

Κωδικοποίηση

Στην παρούσα έρευνα χρησιμοποιήθηκε δυαδική κωδικοποίηση σύμφωνα με την εξίσωση που διατυπώθηκε στο αντίστοιχο κεφάλαιο του Θεωρητικού Υποβάθρου για τους γενετικούς αλγόριθμους $(x_2 - x_1)*10^n \leq 2^{l_i} - 1$ επιλέγεται η παράμετρος nValmax ώστε να εξασφαλιστεί η απαιτούμενη ποικιλομορφία και η ακρίβεια του πληθυσμού. Έστω ότι θέλουμε να

κωδικοποιήσουμε τους αριθμούς που ανήκουν στο διάστημα [-5.000, 1.000] με ακρίβεια τριών δεκαδικών ψηφίων. Πρέπει δηλαδή να κωδικοποιήσουμε έξι χιλιάδες ένα (6001) αριθμούς με την ακόλουθη σειρά -5.000, -4.999, -4.998, ...,1.000. Άρα με βάση την εξίσωση θα πρέπει ο αριθμός έξι χιλιάδες ένα να εκφραστεί σαν δύναμη του δύο, όμως $2^{12} = 4096$ και $2^{13} = 8192$. Άρα χρειαζόμαστε δεκατρία ψηφία για να αναπαραστήσουμε τα νούμερα.

Με την ίδια λογική στον αλγόριθμο χρησιμοποιώντας την εντολή ορίζονται τα χρωμοσώματα chrom[i] = random.Random.randrange(nValmax) τα οποία ταξινομούνται με αύξουσα σειρά στη συνέχεια. Οι κωδικοποιημένες τιμές που χρησιμοποιήθηκαν για τη κατασκευή του πληθυσμού με τη χρήση της συνάρτησης crhom2Fenotype() και της παραμέτρου της $dx = \frac{L_{\text{ού}}}{n^{1}}$ αποκωδικοποιούνται και ορίζουν τις συντεταγμένες του πληθυσμού. Οι οριζόντιες συντεταγμένες ορίζονται από την σχέση $x_{i} = x_{\text{αύ}} + \text{chrom}_{i} * dx$ και οι κατακόρυφες συντεταγμένες με την αναλυτική μέθοδο, κανόνας του Crammer που αναφέρθηκε. Δημιουργείται έτσι μια δεξαμενή τυχαίων τιμών και στη συνέχεια υπολογίζεται η καταλληλότητα (τιμή της αντικειμενικής συνάρτησης) όλων αυτών των υποψήφιων λύσεων.

Επιλογή γονέων

Η μεθοδολογία που επιλέχθηκε για την επιλογή γονέων για αναπαραγωγή είναι αυτή του τουρνουά. Η μέθοδος του τουρνουά επιλέχθηκε διότι θεωρείται ότι ο τρόπος με τον οποίο γίνεται η επιλογή των ατόμων έως τη συμπλήρωση του κατάλληλου αριθμού πληθυσμού εξασφαλίζει την απαραίτητη ποικιλομορφία του. Επιλέγονται τόσοι γονείς όσο το μέγεθος του πληθυσμού, και αυτοί αποθηκεύονται σε μια δεξαμενή γονέων για να χρησιμοποιηθούν αργότερα στο στάδιο της διασταύρωσης αναπαραγωγής.

Διασταύρωση – Αναπαραγωγή

Από τη δεξαμενή γονέων που δημιουργήθηκε στο πιο πάνω βήμα δεν θα αναπαραχθούν όλοι οι γονείς αλλά αφού από τη δεξαμενή γονέων επιλεγούν είκοσι άτομα τυχαία με την συνάρτηση random.sample() θα επιλεχθούν δέκα άτομα με βάση την απόδοσή τους στη συνάρτηση στόχου, δηλαδή το ελάχιστο εμβαδό. Η συνάρτηση random.sample() ανήκει στη βιβλιοθήκη random και έχει συνταχθεί από τους εκδότες της python. Επιστρέφει μια λίστα δοσμένου μήκους μοναδικών στοιχείων από ένα σύνολο ή ακολουθία το οποίο δέχεται.

Τα άτομα που επιλέγονται ζευγαρώνουν ανά δύο. Κατά τη διαδικασία του ζευγαρώματος τα χρωμοσώματα των απογόνων που δημιουργούνται είναι προϊόν τυχαίας σύνθεσης των χρωμοσωμάτων των γονέων. Αφού αντιγραφούν τα χρωμοσώματα των γονέων αυτά ταξινομούνται με αύξουσα σειρά. Και δημιουργούνται οι απόγονοι ενώ οι γονείς διαγράφονται. Η διαδικασία επαναλαμβάνεται έως ότου να πραγματοποιηθεί κάποια από τις συνθήκες τερματισμού.

Κρίνεται ικανό να αναφερθεί ότι μόλις αξιολογηθούν οι γονείς αποθηκεύεται η ελάχιστη τιμή της γενιάς στο τέλος της επαναληπτικής διαδικασίας εμφανίζεται η ελάχιστη τιμή της συνάρτησης στόχου όλων των γενεών. Επίσης πρέπει να αναφερθεί πως το μέγεθος του πληθυσμού δεν πρέπει να μειώνεται με την επαναληπτική διαδικασία.

Μετάλλαξη

Μετά την δημιουργία του νέου πληθυσμού είναι το στάδιο της μετάλλαξης δηλαδή της αλλοίωσης των χαρακτηριστικών μιας πιθανής λύσης. Κάθε άτομο του πληθυσμού και αντίστοιχα κάθε χαρακτηριστικό του ατόμου αυτού μπορούν αντικατασταθούν με μία άλλη τιμή αφού πρώτα εξεταστεί η πιθανότητα μετάλλαξης (mutation_probability). Η μετάλλαξη στο πρόγραμμα πραγματοποιείται μια φορά στις πέντε γενιές. Ενώ αν δεν οριστεί από τον χρήστη πραγματοποιείται μια μετάλλαξη ανά γενιά. Αφού κάποιο χαρακτηριστικό μεταλλαχθεί τα χρωμοσώματα ταξινομούνται σε αύξουσα σειρά και το άτομο αξιολογείται για να συμμετέχει στην αναπαραγωγή.

Συνθήκες τερματισμού

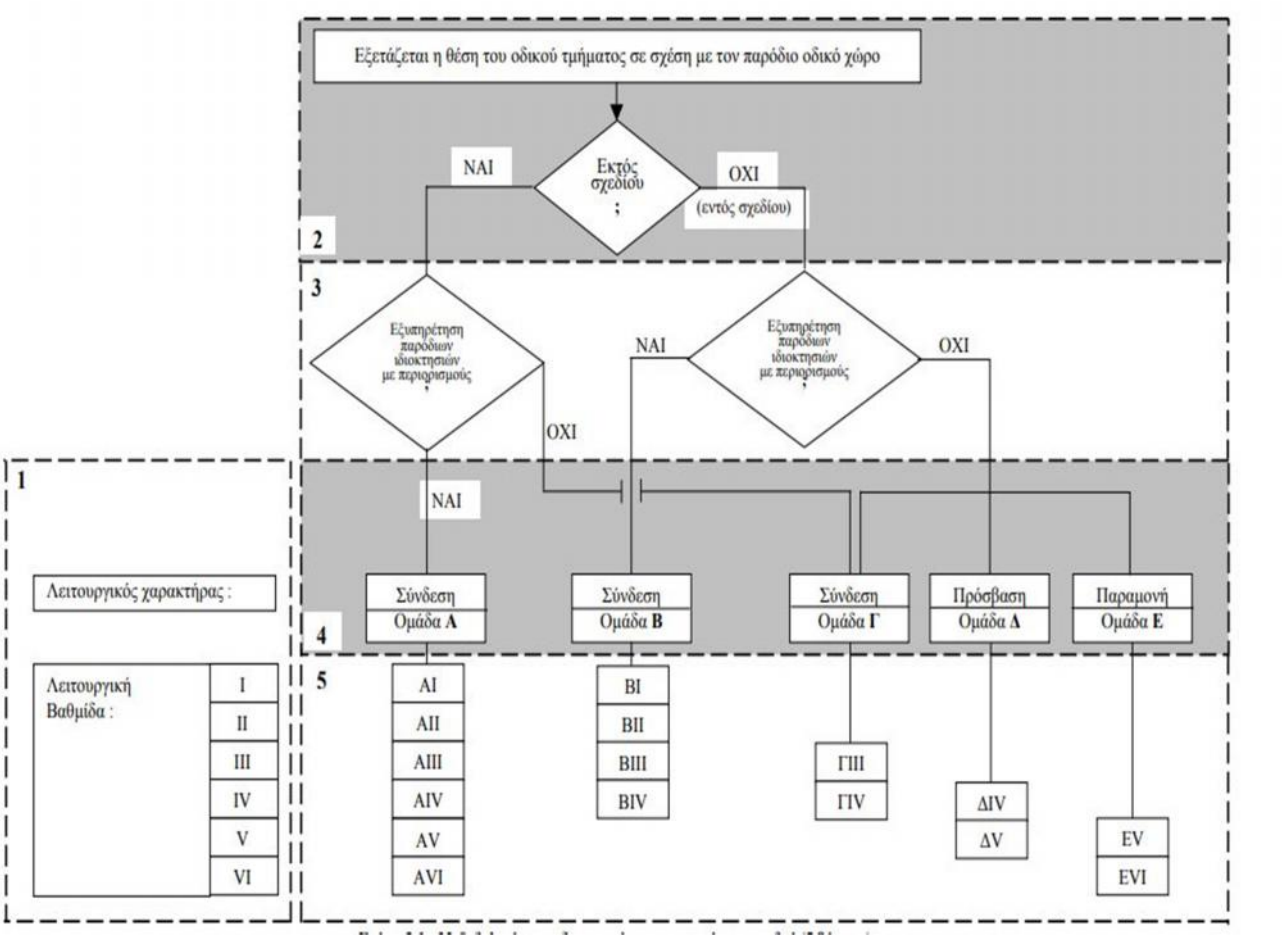
Το πρώτο κριτήριο τερματισμού της διαδικασία είναι η υπέρβαση του μέγιστου αριθμού γενεών. Ορίστηκε πλήθος τερματισμού του αλγόριθμου ίσο με διακόσιες γενιές. Το δεύτερο κριτήριο τερματισμού που ορίστηκε μέσω της συνάρτησης converged() είναι ότι αφού ο αλγόριθμος έχει κάνει πάνω από εκατό επαναλήψεις αν το αποτέλεσμα δεν έχει βελτιωθεί για είκοσι συναπτά βήματα η μέθοδος συγκλίνει και τερματίζει.

Δηλαδή με τη συμπλήρωση των εκατό πρώτων γενεών ο αλγόριθμος μπορεί να τερματίσει αν για είκοσι γενιές δεν έχει βελτιωθεί η ελάχιστη τιμή του αθροίσματος των απόλυτων τιμών των εμβαδόν που περικλείονται από την γραμμή του εδάφους και την ερυθρά της οδού.

5 Εφαρμογή Μεθοδολογίας – Αποτελέσματα

5.1 Εφαρμογές - Αποτελέσματα

Σε αυτό το κεφάλαιο παρουσιάζονται οι μηκοτομές οι οποίες χαράχθηκαν και σχεδιάστηκαν με το πρόγραμμα που συντάχθηκε κατά την διάρκεια της διπλωματικής εργασίας. Το πρόγραμμα παρέχει δυνατότητα επιλογής στον χρήστη του αριθμού των θλάσεων της ερυθράς, της ταχύτητας μελέτης, της κατηγορίας οδού και του τύπου του εδάφους όπου χρειάζεται. Αν ο χρήστης δεν ορίσει τον αριθμό των θλάσεων ερυθράς εμφανίζονται από το πρόγραμμα τρεις διαφορετικές εκδοχές για όσα σημεία θλάσης επιτρέπονται σύμφωνα με τον περιορισμό που αναφέρθηκε σε προηγούμενο κεφάλαιο. Οι μηκοτομές που παρουσιάζονται επιλέχθηκαν με αποκλειστικό κριτήριο το ελάχιστο άθροισμα εμβαδών των απόλυτων τιμών των χωματισμών. Η συνάρτηση στόχου για όλες τις εφαρμογές υπολογίστηκε με την μέθοδο των γενετικών αλγορίθμων εκτός από την Επαρχιακή Οδό του νομού Μεσσηνίας η συνάρτηση στόχου της οποίας υπολογίστηκε με την μέθοδο Monte Carlo. Η ταχύτητα κάθε οδού επιλέχθηκε με βάση το τεύχος ‘Λειτουργική Κατάταξη Οδών’ της ΟΜΟΕ από το οποίο προκύπτει το παρακάτω σχεδιάγραμμα. Επίσης πραγματοποιήθηκε επισκόπηση της οδού με τη χρήση του προγράμματος Google Earth.



Σχήμα 5: Λειτουργική κατάταξη οδού Πηγή : ΟΜΟΕ – ΛΚΟΔ

5.1.1 Εγνατία

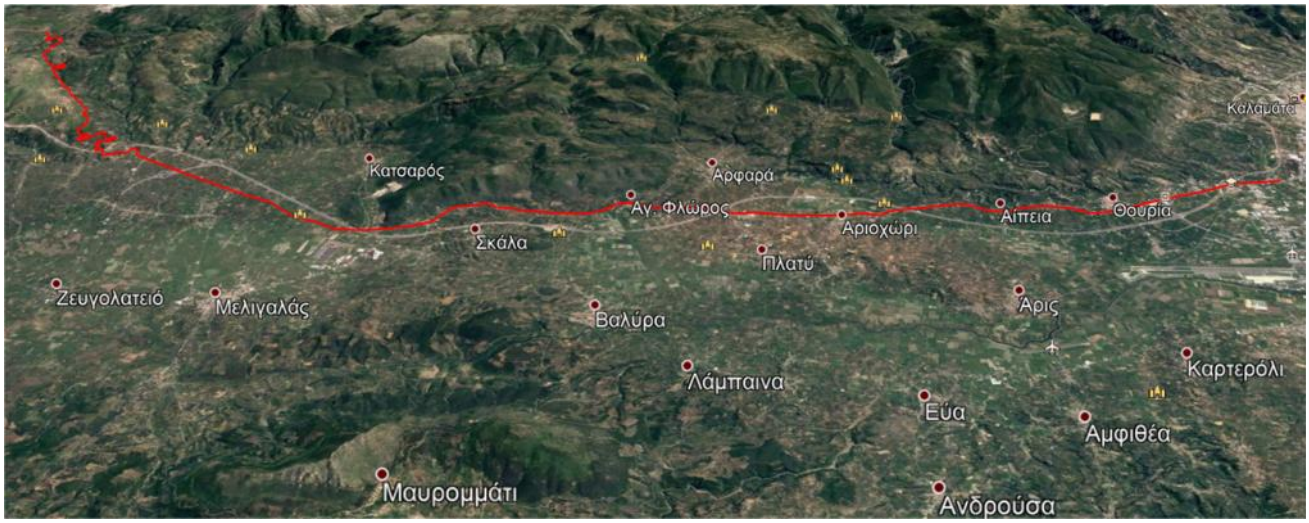
Για το τμήμα της Εγνατίας οδού μήκους 4466,42 m από την χιλιομετρική θέση -1+ 162.71 έως την χιλιομετρική θέση 3+ 403.71 επιλέχθηκε βάση των κανονισμών της ΟΜΟΕ – ΛΚΟΔ κατηγορία οδού ΑΙ ,ορεινού ανάγλυφου και ταχύτητα μελέτης V_e = 120 km/ h. Οι ακτίνες των κατακόρυφων καμπύλων ορίστηκαν από τους κανονισμούς ΟΜΟΕ – Χ, κοίλη κατακόρυφη ακτίνα h_w = 7500 και κυρτή κατακόρυφη ακτίνα h_k = 15000. Το ελάχιστο εμβαδό χωματισμών υπολογίστηκε 13284.98 m^2 για επτά σημεία θλάσης της ερυθράς τα οποία απεικονίζονται στον παρακάτω πίνακα.

Χ.Θ.	Υ
-1+162.71	88.14
-0+202.27	64.28
0+430.97	70.43
0+845.97	54.09
1+703.01	50.38
2+427.66	36.27
2+760.72	42.50
3+069.53	33.91
3+403.71	34.00

Εγνατία

ΚΛΙΜΑΚΑ ΥΨΩΝ: 1/ 500	
ΚΛΙΜΑΚΑ ΜΗΚΩΝ: 1/ 5000	
ΥΨΟΜΕΤΡΙΑ ΟΡΙΟΓΡΑΜΜΗΣ	
ΥΨΟΜΕΤΡΙΑ ΔΑΦΟΥΣ	
ΥΠΟΔΟΜΕΣ ΔΙΑΒΑΣΕΩΝ	
ΑΠΟΣΤΑΣΕΙΣ ΣΗΜΕΙΩΝ	
ΑΠΟΣΤΑΣΕΙΣ ΚΑΙ ΚΛΙΣΕΙΣ	
ΚΑΜΠΥΛΟΤΗΤΕΣ	
ΕΛΑΧΙΣΤΟΙ ΑΚΤΙΝΕΣ ΚΑΜΠΥΛΟΤΗΤΑΣ	
ΤΟΞΑ - ΚΛΙΣΕΙΣ	

5.1.2 Επαρχιακή Οδός Νομού Μεσσηνίας



Εικόνα 15: Επαρχιακή Οδός νομού Μεσσηνίας

Για το τμήμα, μήκους 692.91 m, της Επαρχιακής οδού του νομού Μεσσηνίας, που απεικονίζεται στην φωτογραφία, επιλέχθηκε βάση των κανονισμών της ΟΜΟΕ – ΛΚΟΔ κατηγορία οδού ΑΙΙ με ορεινό ανάγλυφο και ταχύτητα μελέτης V_e = 80 km/ h. Οι ακτίνες των κατακόρυφων καμπύλων ορίστηκαν από τους κανονισμούς ΟΜΟΕ – Χ, κοίλη κατακόρυφη ακτίνα h_w = 2500 και κυρτή κατακόρυφη ακτίνα h_k = 4500. Το ελάχιστο εμβαδό χωματισμών υπολογίστηκε 25.64m^2 και ορίστηκε ένα σημείο θλάσης που απεικονίζεται μαζί με τα σημεία αρχήςκαι τέλους της οδού στον παρακάτω πίνακα.

Χ.Θ.	Υ
0+000.00	74.43
0+424.97	71.43
0+692.91	70.57

Επ. Οδός Νομού Μεσσηνίας

ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/200
ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000

ΤΟΞΑ - ΚΛΙΣΕΙΣ	ΕΥΘΥΓΡΑΜΜΙΕΣ-ΚΑΜΠΥΛΕΣ	ΧΙΛΙΟΜΕΤΡΗΣΗ	ΑΠΟΣΤΑΣΕΙΣ ΑΠΟ ΑΡΧΗΣ	ΑΠΟΣΤΑΣΕΙΣ ΜΕΤΑΞΥ	ΑΡΙΘΜΗΣΗ ΔΙΑΤΟΜΩΝ	ΥΨΟΜΕΤΡΑ ΕΔΑΦΟΥΣ	ΥΨΟΜΕΤΡΑ ΕΡΥΘΡΩΝ
			0.0		1	74.43	74.43
				16.8			
			16.8		2	74.28	74.31
				16.8			
			33.6		3	74.17	74.19
				20.0			
			53.6		4	74.02	74.05
				14.5			
			68.1		5	73.93	73.95
				14.5			
			82.6		6	73.80	73.85
				20.0			
			102.6		7	73.50	73.71
				16.5			
			119.2		8	73.45	73.59
				16.5			
			135.7		9	73.15	73.47
				20.0			
			155.7		10	73.30	73.33
				16.5			
			172.2		11	73.17	73.21
				16.5			
			188.8		12	73.07	73.10
				20.0			
			208.8		13	72.95	72.96
				14.5			
			223.3		14	72.87	72.85
				14.5			
			237.8		15	72.77	72.75
				20.0			
			257.8		16	72.60	72.61
				20.0			
			277.8		17	72.48	72.47
				20.0			
			297.8		18	72.33	72.33
				20.0			
			317.8		19	72.17	72.19
				20.0			
			337.8		20	72.05	72.05
				20.0			
			357.8		21	71.90	71.90
				14.2			
			372.0		22	71.80	71.80
				14.2			
			386.2		23	71.72	71.70
				20.0			
			406.2		24	71.62	71.56
				20.0			
			426.2		25	71.50	71.43
				20.0			
			446.2		26	71.30	71.36
				14.4			
			460.6		27	71.33	71.32
				14.4			
			475.1		28	71.32	71.27
				20.0			
			495.1		29	71.10	71.20
				20.0			
			515.1		30	71.20	71.14
				10.0			
			525.1		31	71.14	71.11
				10.0			
			535.2		32	71.09	71.08
				20.0			
			555.2		33	71.00	71.01
				20.0			
			575.2		34	70.93	70.95
				10.1			
			585.2		35	70.92	70.92
				10.0			
			595.3		36	70.87	70.88
				20.0			
			615.3		37	70.85	70.82
				20.0			
			635.3		38	70.78	70.76
				20.0			
			655.3		39	70.70	70.69
				14.5			
			669.7		40	70.65	70.64
				14.4			
			684.2		41	70.59	70.60
			692.9	8.8	42	70.57	70.57

-0.7 %
4216

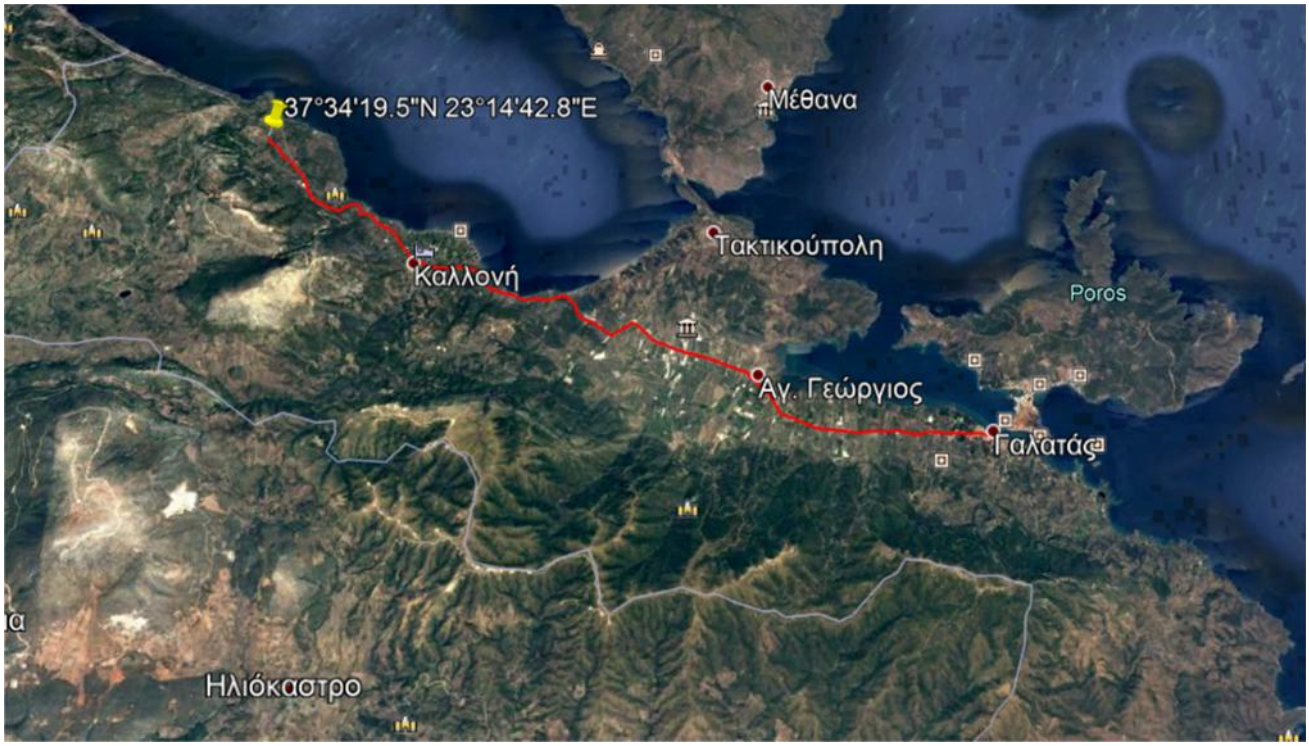
-0.3 %
2813

H=66.0

0.00
0.03
0.02
0.03
0.02
0.05
0.21
0.14
0.32
0.03
0.04
0.03
0.01
-0.02
-0.02
0.01
-0.01
0.00
0.02
0.00
0.00
0.00
-0.02
-0.06
420.2
-0.07
71.41
0.06
-0.01
-0.05
0.10
-0.06
-0.03
-0.01
0.01
0.02
0.00
0.01
-0.03
-0.02
-0.01
-0.01
0.01
0.00

H= 71.43
R= 2500
T= 4.81
L= 9.62
G= 0.00

5.1.3 Δρυόπη - Γαλατάς

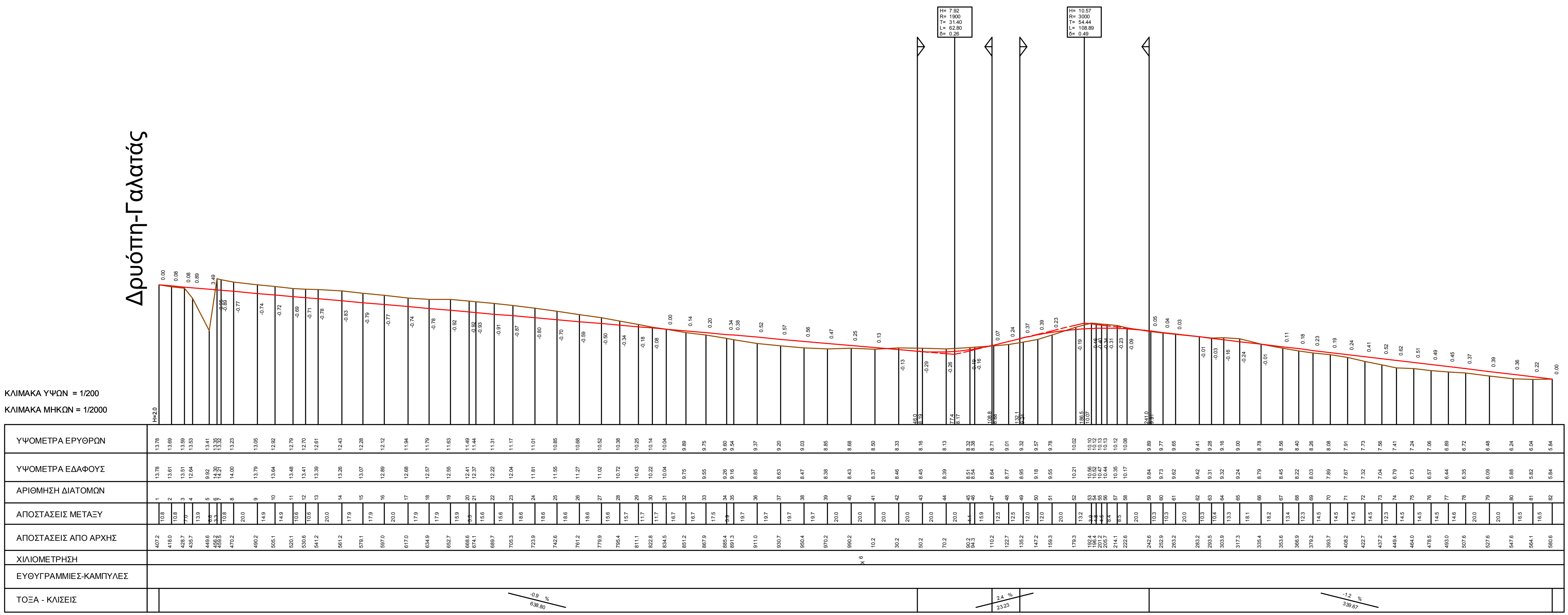


Εικόνα 16: Επαρχιακή οδός σύνδεσης της Δρυόπης – Γαλατά

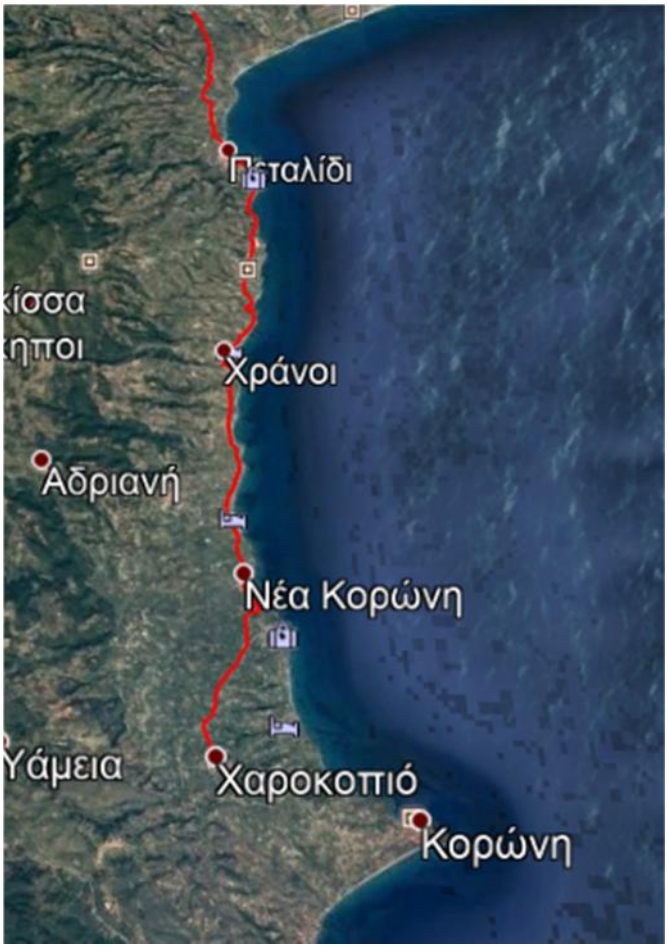
Για το τμήμα, μήκους 1173.4 m, της επαρχιακής οδού σύνδεσης της Δρυόπης με τον Γαλατά από την χιλιομετρική θέση 5+ 407.23 έως την χιλιομετρική θέση 6+ 580.63 επιλέχθηκε βάση των κανονισμών της ΟΜΟΕ – ΛΚΟΔ κατηγορία οδού AIV, λοφώδες ανάγλυφο και ταχύτητα μελέτης V_e = 60 km/ h. Οι ακτίνες των κατακόρυφων καμπύλων ορίστηκαν από τους κανονισμούς ΟΜΟΕ – Χ, κοίλη κατακόρυφη ακτίνα h_w = 1900 και κυρτή κατακόρυφη ακτίνα h_k = 3000. Το ελάχιστο εμβαδό χωματισμών υπολογίστηκε 506.57 m^2 και ορίστηκαν δύο σημεία θλάσης. Οι κορυφές της ερυθράς φαίνονται στον παρακάτω πίνακα.

Χ.Θ.	Υ
5+407.23	13.78
6+077.44	7.92
6+186.51	10.57
6+580.63	5.84

Δρυόπη-Γαλατάς



5.1.4 Ριζόμυλος - Κορώνη

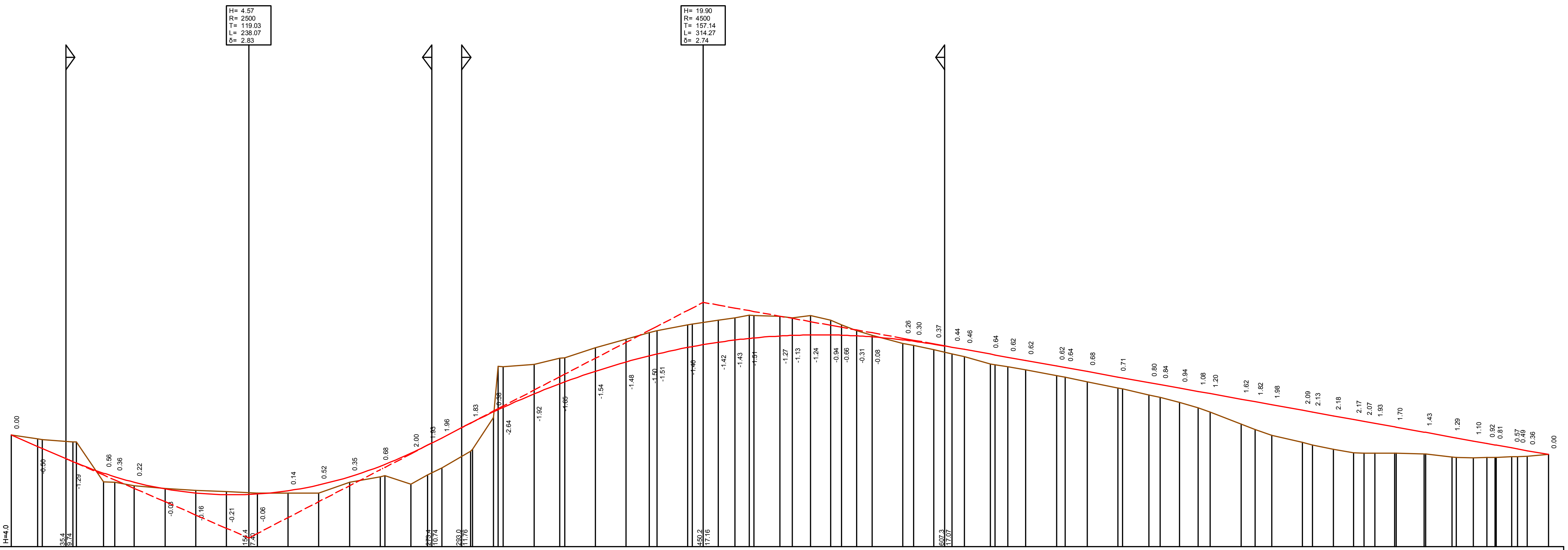


Εικόνα 17: Επαρχιακής οδού σύνδεσης του Ριζόμυλου – Κορώνης

Για το τμήμα, μήκους 1000 m, της επαρχιακής οδού σύνδεσης του Ριζόμυλου με την Κορώνη από την χιλιομετρική θέση 3+ 000.00 έως την χιλιομετρική θέση 4+ 000.00 επιλέχθηκε βάση των κανονισμών της ΟΜΟΕ – ΛΚΟΔ κατηγορία οδού ΑΙΙΙ, λοφώδες ανάγλυφο και ταχύτητα μελέτης V_e = 70 km/ h. Οι ακτίνες των κατακόρυφων καμπύλων ορίστηκαν από τους κανονισμούς ΟΜΟΕ – Χ, κοίλη κατακόρυφη ακτίνα h_w = 2500 και κυρτή κατακόρυφη ακτίνα h_k = 4500. Το ελάχιστο εμβαδό χωματισμών υπολογίστηκε 820,23 m^2 και ορίστηκαν δύο σημεία θλάσης. Οι κορυφές της ερυθράς φαίνονται στον παρακάτω πίνακα.

Χ.Θ.	Υ
3+000.00	11.27
3+154.40	4.57
3+450.16	19.90
4+000.00	10.00

Ρυζόμυλος - Κορώνη



ΚΛΙΜΑΚΑ ΥΨΩΝ = 1/200

ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 1/2000

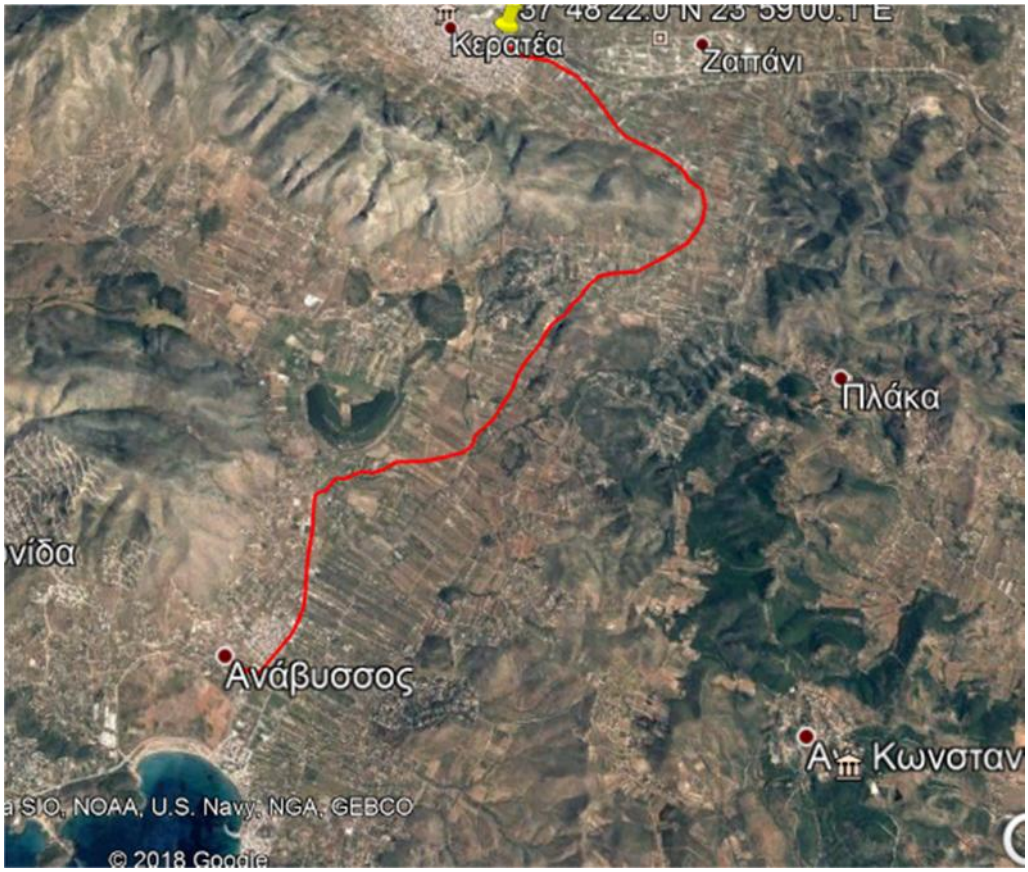
ΑΡΙΘΜΗΣΗ ΔΙΑΤΟΜΩΝ	ΥΨΟΜΕΤΡΑ ΕΡΥΘΡΩΝ	ΥΨΟΜΕΤΡΑ ΕΔΑΦΟΥΣ	ΑΠΟΣΤΑΣΕΙΣ ΑΠΟ ΑΡΧΗΣ	ΑΠΟΣΤΑΣΕΙΣ ΜΕΤΑΞΥ (to the next section)
1	11.27	11.27	0.0	17.2
2	10.52	11.02	17.2	20.0
4	9.54	10.83	40.0	17.8
6	8.79	8.23	60.0	7.2
7	8.56	8.20	67.2	12.8
8	8.20	7.98	80.0	20.0
9	7.77	7.80	100.0	20.0
10	7.50	7.66	120.0	20.0
11	7.38	7.59	140.0	20.0
12	7.43	7.49	160.0	20.0
13	7.64	7.50	180.0	20.0
14	8.01	7.49	200.0	20.0
15	8.54	8.19	220.0	20.0
16	9.23	8.55	240.0	16.9
18	10.08	8.08	260.0	10.8
19	10.61	8.68	270.8	9.2
20	11.08	9.12	280.0	18.6
21	12.04	10.21	298.6	13.8
23	12.78	12.40	313.8	3.2
25	13.07	15.71	320.0	20.0
26	13.94	15.86	340.0	16.9
27	14.61	16.26	356.9	20.0
29	15.42	16.96	380.0	20.0
30	16.03	17.51	400.0	15.1
31	16.43	17.93	415.1	4.9
32	16.55	18.06	420.0	20.0
33	16.97	18.43	440.0	17.1
35	17.31	18.73	460.0	10.7
36	17.46	18.89	470.7	9.3
37	17.56	19.07	480.0	16.9
39	17.72	18.99	500.0	8.1
40	17.76	18.89	508.1	11.9
41	17.80	19.04	520.0	13.1
42	17.79	18.73	533.1	6.9
43	17.78	18.44	540.0	10.0
44	17.74	18.05	550.0	10.1
45	17.67	17.75	560.1	19.9
46	17.48	17.22	580.0	7.0
47	17.39	17.09	587.0	13.0
48	17.20	16.83	600.0	12.1
49	16.99	16.55	612.1	7.9
50	16.84	16.38	620.0	17.0
51	16.54	15.90	637.0	8.4
53	16.33	15.71	648.4	11.6
54	16.12	15.50	660.0	20.0
55	15.76	15.14	680.0	5.6
56	15.66	15.02	685.6	14.4
57	15.40	14.72	700.0	20.0
58	15.04	14.33	720.0	17.2
60	14.68	13.88	740.0	7.4
61	14.55	13.71	747.4	12.6
62	14.32	13.38	760.0	12.0
63	14.11	13.03	772.0	8.0
64	13.96	12.76	780.0	20.0
65	13.60	11.98	800.0	9.2
66	13.44	11.62	809.2	10.8
67	13.24	11.26	820.0	20.0
68	12.88	10.79	840.0	6.4
69	12.76	10.63	846.4	13.6
70	12.52	10.34	860.0	13.2
71	12.28	10.11	873.2	6.8
72	12.16	10.09	880.0	7.1
73	12.03	10.10	887.1	12.9
74	11.80	10.10	900.0	18.2
76	11.46	10.03	919.2	17.3
78	11.13	9.84	937.3	11.3
80	10.88	9.78	951.3	8.7
81	10.72	9.80	960.0	5.2
82	10.63	9.82	965.2	10.0
84	10.43	9.86	976.1	3.9
85	10.36	9.87	980.0	6.1
86	10.25	9.89	986.1	13.9
87	10.00	10.00	0.0	

ΧΙΛΙΟΜΕΤΡΗΣΗ: x 3 (at section 1); x 4 (at section 87)

ΕΥΘΥΓΡΑΜΜΙΕΣ-ΚΑΜΠΥΛΕΣ

ΤΟΞΑ - ΚΛΙΣΕΙΣ: 5.2 % / 19.59; -1.8 % / 392.70

5.1.5 Κερατέα - Ανάβυσσος



Εικόνα 18: Οδός μεταξύ οικισμών Αναβύσσου – Κερατέας

Για το τμήμα, μήκους 8994.99 m, της επαρχιακής οδού σύνδεσης της Κερατέας με την Ανάβυσσο από την χιλιομετρική θέση 0+ 000.00 έως την χιλιομετρική θέση 8+ 994.99 επιλέχθηκε βάση των κανονισμών της ΟΜΟΕ – ΛΚΟΔ κατηγορία οδού ΑΙΙΙ ορεινό ανάγλυφο και ταχύτητα μελέτης V_e = 70 km/ h. Οι ακτίνες των κατακόρυφων καμπύλων ορίστηκαν από τους κανονισμούς ΟΜΟΕ – Χ, κοίλη κατακόρυφη ακτίνα h_w = 1900 και κυρτή κατακόρυφη ακτίνα h_k = 3000. Το ελάχιστο εμβαδό χωματισμών υπολογίστηκε 3958.18 m^2 και ορίστηκαν δεκαπέντε σημεία θλάσης της ερυθράς. Οι κορυφές της ερυθράς φαίνονται στον παρακάτω πίνακα.

Χ.Θ.	Υ
0+000.00	148.15
0+568.23	145.05
1+238.02	146.09
1+920.44	156.10
2+338.79	133.41
2+958.07	139.65
3+405.51	149.19
3+648.73	149.80
4+381.65	169.51

Χ.Θ.	Υ
5+843.12	81.88
6+144.53	72.27
6+256.53	69.29
6+716.60	59.39
7+870.07	44.65
8+180.81	49.57
8+736.96	33.31
8+994.99	37.13


OLCE KEPATEAL AWARE

5.1.6 Οδός Κιάφα – Σπάτα

Εικόνα19 : Οδός Κιάφα, Σπάτα

Για την αστική αρτηρία που μελετάται για την οδό Κιάφα, μήκους 4014.80 m, από την χιλιομετρική θέση 0+ 000.00 έως την χιλιομετρική θέση 4+ 014.80 επιλέχθηκε βάση των κανονισμών της ΟΜΟΕ – ΛΚΟΔ κατηγορία οδού ΒΙΙΙ και ταχύτητα μελέτης V_e = 70 km/ h. Το ελάχιστο εμβαδό χωματισμών υπολογίστηκε 1382.03 m^2 και ορίστηκαν οχτώ σημεία θλάσης. Οι κορυφές της ερυθράς απεικονίζονται στον παρακάτω πίνακα.

Χ.Θ.	Υ
0+000.00	109.46
0+406.53	96.08
0+628.54	90.64
0+841.97	88.31
1+930.21	72.70
2+238.48	76.04
2+472.49	65.29
2+887.35	64.00
3+121.13	70.51
4+014.80	110.11

ΚΑΛΑΦΑΣ

ΚΛΙΜΑΚΑ ΥΨΩΝ = 100
ΚΛΙΜΑΚΑ ΜΗΚΩΝ = 5000

ΥΨΟΜΕΤΡΑ ΕΡΥΘΡΟΥ
ΥΨΟΜΕΤΡΑ ΣΗΜΕΙΩΝ
ΑΡΙΘΜΗΣΗ ΔΙΑΤΟΜΩΝ
ΑΠΟΣΤΑΣΕΙΣ ΜΕΤΑΞΥ
ΑΠΟΣΤΑΣΕΙΣ ΚΟΙΝΗΣ
ΚΛΙΣΕΙΣ/ΠΤΩΣΕΙΣ
ΕΥΘΥΓΡΑΜΜΕΣ ΜΕΤΡΗΣΕ
ΤΟΠΟΣ - ΛΗΞΕΙΣ

5.2 Σύγκριση με χαράξεις Μηχανικών

Για την επαρχιακή οδό του νομού Μεσσηνίας και για την επαρχιακή οδό Ριζόμυλου – Κορώνης πραγματοποιήθηκε σύγκριση των απόλυτων τιμών των εμβαδών των χωματισμών των χαράξεων που παράχθηκαν από το πρόγραμμα με χαράξεις που παράχθηκαν από μηχανικούς. Οι χωματισμοί των οδών που χαράχθηκαν από μηχανικούς υπολογίστηκαν με πρόγραμμα το οποίο συντάχθηκε βάσει του αρχικού προγράμματος και παρατίθεται στο παράρτημα.

Στη περίπτωση της Επαρχιακής οδού του νομού Μεσσηνίας οι απόλυτες τιμές των εμβαδών των χωματισμών που παράχθηκαν από το πρόγραμμα είναι μικρότερες από τις αντίστοιχες της ερυθράς που σχεδιάστηκε από τον μηχανικό. Στην μηκοτομή της οδού Ριζόμυλος – Κορώνη οι απόλυτες τιμές των εμβαδών των χωματισμών που παράχθηκαν από το πρόγραμμα είναι μεγαλύτερες από τις αντίστοιχες της ερυθράς που σχεδιάστηκε από τον μηχανικό. Πιο συγκεκριμένα για την Επαρχιακή Οδό του νομού Μεσσηνίας οι χωματισμοί που παράχθηκαν από μηχανικούς υπολογίστηκαν 69.68 m^2 ενώ οι χωματισμοί της παραγόμενης από το πρόγραμμα ερυθράς είχε απόλυτη τιμή ίση με 25.64m^2. Για την επαρχιακή οδό Ριζόμυλου – Κορώνης οι χωματισμοί που υπολογίστηκαν για την ερυθρά που σχεδιάστηκε από μηχανικούς υπολογίστηκαν 801.46 m^2 ενώ οι χωματισμοί της παραγόμενης από το πρόγραμμα ερυθρά είχε απόλυτη τιμή ίση με 820,23 m^2.

Εν κατακλείδι οι απόλυτες τιμές των εμβαδών των χωματισμών που παράγονται από το πρόγραμμα είναι παραπλήσιες με τις αντίστοιχες τιμές που υπολογίστηκαν για την ερυθρά της οδού Ριζόμυλου – Κορώνης από την υπάρχουσα μηκοτομή. Ενώ οι χωματισμοί της μηκοτομής της επαρχιακής οδού του νομού Μεσσηνίας που υπολογίστηκαν με βάση την ερυθρά που παράγεται από το πρόγραμμα είναι Μεσσηνίας είναι σημαντικά λιγότεροι από αυτούς που υπολογίστηκαν από την ερυθρά που παράχθηκε από τον μηχανικό, δηλαδή το πρόγραμμα έδωσε πολύ καλύτερη λύση (σχήμα σελ. 63). Επομένως από τις εφαρμογές που έγιναν συμπεραίνουμε από το πρόγραμμα παράγεται η ερυθρά της οδού οι χωματισμοί της οποίας είναι είτε παραπλήσιοι είτε λιγότεροι από χωματισμούς που προκύπτουν από χαράξεις μηχανικών. Το αποτέλεσμα λοιπόν του προγράμματος είναι είτε ίδιο είτε καλύτερο από το αντίστοιχο αποτέλεσμα που παράγεται από έναν μηχανικό. Ένα επιπλέον στοιχείο που είναι σκόπιμο να επισημανθεί είναι πως μέσω της χρήσης του προγράμματος για την χάραξη της ερυθράς εξασφαλίζεται ο κανονισμός που τέθηκε από την ΟΜΟΕ – Χ για εναρμόνιση των κυρτών και κοίλων καμπύλων κατά τη κατακόρυφη χάραξη καθώς λαμβάνεται σταθερή ταχύτητα καθ’ όλο το μήκος της υπό μελέτη οδού.

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΝΟΜΑΡΧΙΑΚΗ ΑΥΤΟΔΙΟΙΚΗΣΗ ΜΕΣΣΗΝΙΑΣ

Ε.Ο. ΤΣΑΚΩΝΑ - ΚΑΛΑΜΑΤΑ
Χ.Θ. 63 ΧΜ - 64 ΧΜ

ΜΕΛΕΤΗ ΟΔΟΠΟΙΙΑΣ

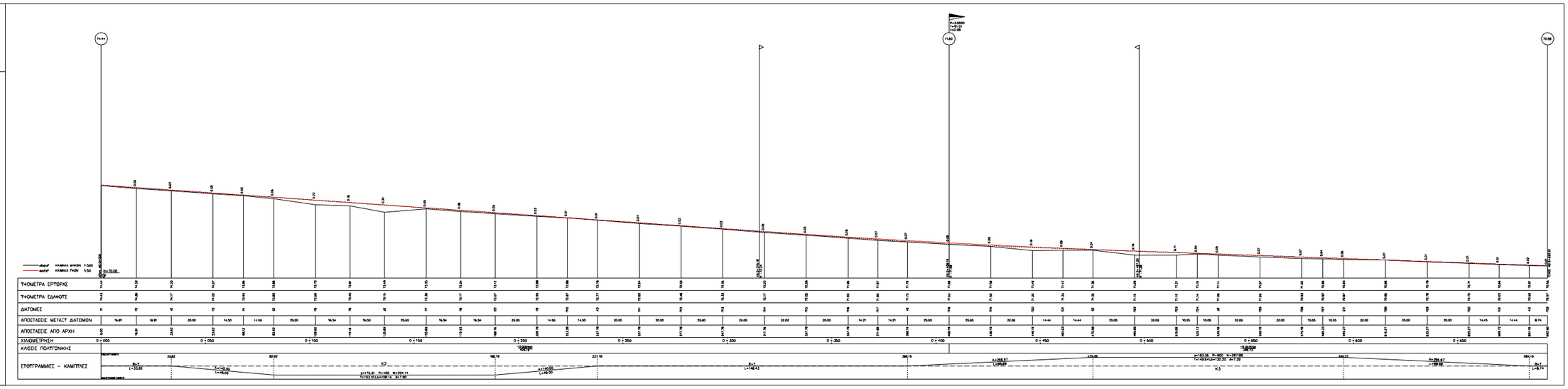
ΤΙΤΛΟΣ ΣΧΕΔΙΩΝ: ΜΗΚΟΤΟΜΗ	ΑΡ. ΣΧΕΔΙΟΥ:	2
	ΚΩΔ.ΜΕΛΕΤΗΣ:	2
	ΚΛΙΜΑΚΑ:	1:500/1:50
	REV:	00
	ΗΜΕΡΟΜΗΝΙΑ:	ΙΟΥΛΙΟΣ 2004

ΣΥΝΤΑΞΑΝΤΕΣ:

ΑΡΕΤΟΜΕΝΗΣ ΣΤΑΦΟΓΙΩΡΓΑΣ
Πρ/κός Μηχανικός

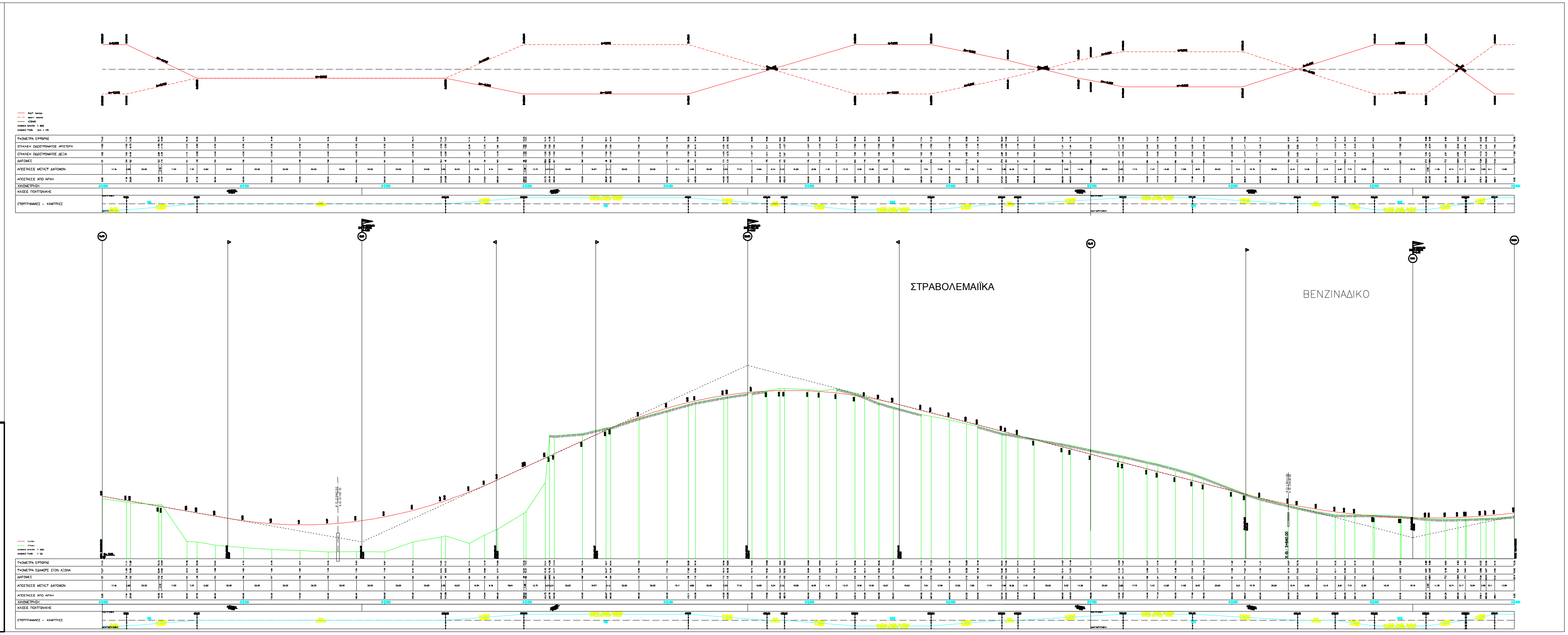
ΜΑΡΙΑΝΝΑ ΑΣΚΟΥΝΑ
Πρ/κός Μηχανικός

ΔΙΑΓ/ΝΩΣΕΙΣ:	ΠΡΟΣΠΟΜΕΙΕΣ:	ΣΥΚΡΙΣΕΙΣ:



ΠΑΡΑΤΗΡΗΣΗ :

Ο ΑΚΡΙΒΗΣ ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΩΝ ΤΕΧΝΙΚΩΝ ΓΙΝΕΤΑΙ ΣΤΙΣ ΜΕΛΕΤΕΣ ΑΠΟΧΕΤΕΥΣΗΣ ΚΑΙ ΤΕΧΝΙΚΩΝ ΕΡΓΩΝ ΑΝΤΙΣΤΟΙΧΑ.



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΕΡΙΦΕΡΕΙΑ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΓΕΝΙΚΗ ΔΙΕΥΘΥΝΣΗ ΠΕΡΙΦΕΡΕΙΑΣ
ΔΙΕΥΘΥΝΣΗ ΔΗΜΟΣΙΩΝ ΕΡΓΩΝ

ΜΕΛΕΤΗ: "ΔΡΟΜΟΣ ΡΙΖΟΜΥΛΟΣ - ΚΟΡΩΝΗ"

ΑΡΙΘΜ. ΕΡΓΟΥ : Κ.Α. 2002 ΜΠ 0263 0030

**ΟΡΙΣΤΙΚΗ ΜΕΛΕΤΗ ΑΡΤΗΡΙΑΣ
ΚΑΙ ΑΠΟΚΑΤΑΣΤΑΣΗΣ ΟΔΙΚΟΥ ΔΙΚΤΥΟΥ**

ΤΜΗΜΑ Χ.Θ.: 0+000 :-10+160.00

ΘΕΜΑ ΣΧΕΔΙΟΥ:	ΜΗΚΟΤΟΜΗ - ΔΙΑΓΡΑΜΜΑ ΕΠΙΚΛΙΣΕΩΝ ΑΡΤΗΡΙΑΣ ΑΠΟ Χ.Θ.:3+000.00 ΕΩΣ Χ.Θ.:4+000.00		
ΑΡ. ΣΧΕΔΙΟΥ: Μ-4Ν	ΑΝΑΔΟΧΟΣ ΜΕΛΕΤΗΣ		
ΚΛΙΜΑΚΑ: 1:500	ΑΞΩΝ ΜΕΛΕΤΑΙ Α.Τ.Ε.		
		ΥΠΟΓΡΑΦΗ	ΗΜΕΡΟΜΗΝΙΑ
ΣΥΝΤΑΧΘΗΚΕ:	ΑΞΩΝ ΜΕΛΕΤΑΙ Α.Τ.Ε.		ΝΟΕΜΒΡΙΟΣ 2006
ΑΝΑΘΕΩΡΗΣΕΙΣ:	A [illegible]		ΟΚΤΩΒΡΙΟΣ 2008

ΠΕΡΙΦΕΡΕΙΑ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΔΙΕΥΘΥΝΣΗ ΔΗΜΟΣΙΩΝ ΕΡΓΩΝ

	ΟΝΟΜΑΤΕΠΩΝΥΜΟ	ΥΠΟΓΡΑΦΗ	
ΔΙΕΥΘΥΝΟΥΣΑ ΥΠΗΡΕΣΙΑ — Ο Ελεγξας Τρίπολη, .../..../....			
Για την ομάδα Επίβλεψης Τρίπολη, .../..../....	ΑΝΤ. Α.ΠΑΠΑΔΗΜΗΤΡΙΟΥ ΓΕΩΛΟΓΟΣ ΜΕ ΣΥΜΒΑΣΗ		
Ο Προϊστάμενος Τμημ. ΣΥΓΚ.ΕΡΓΩΝ Τρίπολη, .../..../....	Γ.ΚΩΣΤΟΓΙΑΝΝΗΣ ΠΟΛ.ΜΗΧΑΝΙΚΟΣ ΜΕ Αβ		
ΘΕΩΡΗΘΗΚΕ — Ο Διευθυντής Τρίπολη, .../..../....	Γ.ΚΟΚΟΣΙΟΥΛΗΣ ΠΟΛ.ΜΗΧΑΝΙΚΟΣ ΜΕ Αβ		
Γνωμοδότηση Σ.Δ.Ε			
Απόφαση Έγκρισης			

6 Συμπεράσματα

Στη παρούσα διπλωματική εργασία πραγματοποιήθηκε προσπάθεια για την αυτοματοποίηση της και βελτιστοποίησης της κατακόρυφης χάραξης. Για την πραγματοποίηση της έρευνας χρησιμοποιήθηκαν δύο στοχαστικές μέθοδοι μια ευρετική, η μέθοδος Monte Carlo και μία μεθευρετική, η μέθοδος των γενετικών αλγορίθμων. Στη συνέχεια για τον έλεγχο της απόδοσης του αλγορίθμου σχεδιάστηκαν, με τη βοήθεια ενός προγράμματος του c_mhk.exe (Στάμος, 2018) οι μηκοτομές έξι οδών. Για τις δύο από αυτές ο ερευνητής είχε στη διάθεσή του προϋπάρχοντα σχέδια μηκοτομής με χαραγμένη την ερυθρά της οδού.

6.1 Σύνοψη αποτελεσμάτων

Με βάση τις δοκιμές που πραγματοποιήθηκαν στις οδούς συμπεραίνεται ότι ο αλγόριθμος που συντάχθηκε αποτελεί μια δόκιμη προσπάθεια για την αυτοματοποίηση της κατακόρυφης χάραξης της οδού. Από τις μηκοτομές των οδών για τις οποίες χαράχθηκε η ερυθρά παρατηρούμε ότι ο αλγόριθμος έχει ως αποτέλεσμα ομοιόμορφη χάραξη της ερυθράς της οδού χωρίς ασυνέχειες η οποία προσεγγίζει κατά το δυνατόν το εδαφικό ανάγλυφο τηρώντας τις απαιτήσεις του οδικού σχεδιασμού.

Από την έρευνα που έγινε συμπεραίνουμε ότι η μέθοδος Monte Carlo είναι αποδοτική για τμήματα οδών μικρού μήκους, με μικρό πλήθος θλάσεων ερυθράς. Η αποδοτικότητά της αυξάνεται για μικρή ταχύτητα μελέτης. Ενώ είναι ακατάλληλη για δρόμους μεγάλου μήκους με πολλούς βαθμούς ελευθερίας, δηλαδή θλάσεις της ερυθράς. Επιπλέον σκόπιμο είναι να αναφερθεί ότι όσο αυξάνεται το πλήθος των δοκιμών αυξάνεται και ο χρόνος εκτέλεσης. Η διαδικασία θεωρείται γενικά χρονοβόρα εκτός από την περίπτωση μικρών οδικών τμημάτων.

Η μέθοδος των Γενετικών αλγορίθμων ενδείκνυται για προβλήματα οδοποιίας τα οποία είναι πολυσύνθετα, με ευρύ χώρο αναζήτησης. Κατά τη διαδικασία των δοκιμών τα αποτελέσματα της κατακόρυφης χάραξης προσέγγιζαν όλα τη βέλτιστη τιμή που παρήγαγε ο αλγόριθμος και είχαν μικρή διασπορά μεταξύ τους. Επιπλέον τα αποτελέσματα προσέγγιζαν το εδαφικό ανάγλυφο ιδιαίτερα με την αύξηση του πλήθους των βαθμών ελευθερίας όπως ήταν αναμενόμενο.

6.2 Συνολικά συμπεράσματα

Από τις δύο μεθόδους που χρησιμοποιήθηκαν και οι δύο έλαβαν υπόψη τους περιορισμούς της μέγιστης κατά μήκος κλίσης και της καμπυλότητας της οδού και αναζήτησαν την ερυθρά της οδού για την οποία ελαχιστοποιείται το άθροισμα των απόλυτων τιμών των χωματισμών. Η βελτιστοποιημένη χάραξη της ερυθράς της οδού που προκύπτει μπορεί με ασφάλεια να αποτελέσει μία καλή λύση της κατακόρυφης χάραξης μίας οδού, ή τουλάχιστον μία πολύ καλή αρχική λύση για ένα έμπειρο μηχανικό.

6.3 Προτάσεις - Περαιτέρω έρευνα

Κατά την έρευνα που πραγματοποιήθηκε παράγεται βελτιστοποιημένη η ερυθρά της οδού. Η κατακόρυφη χάραξη της οδού είναι όμως σκόπιμο να είναι εναρμονισμένη με την οριζόντια χάραξη της οδού. Τα σημεία θλάσης της ερυθράς θα πρέπει να είναι σε συμφωνία με τα σημεία θλάσης της οριζοντιογραφίας. Σύμφωνα με τους κανόνες της οδοποιίας τα σημεία θλάσης της μηκοτομής πρέπει να συμπίπτουν με τα σημεία θλάσης της οριζοντιογραφίας ή αν αυτό δεν είναι εφικτό να ανήκουν στα εφαπτόμενα τμήματά των καμπύλων της οριζοντιογραφίας. Ωστόσο ο ορισμός ενός αυστηρού σημείου ή διαστήματος για τις οριζόντιες συντεταγμένες των σημείων θλάσης μειώνει τον χώρο αναζήτησης και την τυχαιότητα.

Επιπλέον ο αλγόριθμος που αναπτύχθηκε θα μπορούσε να βελτιστοποιηθεί χρησιμοποιώντας την δυνατότητα της Python να συνδυάζεται με άλλες γλώσσες προγραμματισμού, όπως η Java και η C++, οι οποίες θα βελτιώσουν την χρονική απόδοση του.

Βιβλιογραφία

Βασιλάς, Α., (2013), «Ανάπτυξη Λογισμικού για Ψηφιακή Απόδοση της Χάραξης Υφιστάμενης Οδού», Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Πολιτικών Μηχανικών, Αθήνα.

Γιαννάκογλου Κ.Χ., (2012), «Μέθοδοι αιτιοκρατικής και Στοχαστικής Βελτιστοποίησης και Εφαρμογές», Σημειώσεις για το Μεταπτυχιακό μάθημα ΔΠΜΣ, Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα

Γιαννής, Γ., (2010), «Οδηγίες για την Ετοιμασία της Διπλωματικής Εργασίας». Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Πολιτικών Μηχανικών, Αθήνα.

Ηροδότου Π., (2018), «Ανάπτυξη εξελικτικού αλγόριθμου για τη βελτιστοποίηση παραμέτρων θερμοδυναμικών μοντέλων», Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Χημικών Μηχανικών, Αθήνα.

Κανελλαΐδης Γ., Μαλέρδος Γ., Καλτσούνης Α., Γλαρός Γ., (2008)., «Σημειώσεις για το Γεωμετρικό Σχεδιασμό των Οδών», Εθνικό Μετσόβιο Πολυτεχνείο, Αθήνα

Καρλαύτης Μ. Γ., Λαγαρός Ν. Δ., (2010), «Επιχειρησιακή έρευνα και βελτιστοποίηση για μηχανικούς». Αθήνα: Συμμετρία

Κεχαγιά Φ., Κιμούνδρης Α., Τσώχος Γ., (2005), «Η γεωμετρία της οδού ως περιβαλλοντική συνιστώσα», 2ο Πανελλήνιο Συνέδριο Οδοποιίας, Βόλος

Λεβεντέας Δ., «Εκμάθηνση python βήμα βήμα. Οδηγός Python μέσω παραδειγμάτων», Πάτρα.

Λυκοθανάσης, Σ., (2001), «Γενετικοί Αλγόριθμοι και Εφαρμογές». Ελληνικό Ανοικτό Πανεπιστήμιο, Σχολή Θετικών Επιστημών και Τεχνολογίας, Πάτρα.

Μαγούτης Κ., Νικολάου Χ., (2015), «ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΜΕ PYTHON» , Σ.Ε.Α.Β , Αθήνα

Μόσχος Ε.Ι., (2015), «Σχεδιασμός και υλοποίηση υβριδικού μεταευρετικού αλγορίθμου βελτιστοποίησης για το πράσινο πρόβλημα δρομολόγησης στόλου ετερογενών οχημάτων σε αστικό περιβάλλον», Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Μηχανολόγων Μηχανικών, Αθήνα.

Μπουτσάκης Α., (2013), «Χάραξη οδού στον χώρο με χρήση καμπύλων ελεύθερης μορφής NURBS (Non-Uniform Rational Basis Spline)», Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Πολιτικών Μηχανικών, Αθήνα.

Παπανικολάου Α. Ν., (2015), «Μια αποτελεσματικότερη αρχικοποίηση του πληθυσμού των Γενετικών Αλγορίθμων για βελτιστοποίηση συναρτήσεων», Μεταπτυχιακή Διπλωματική Εργασία, Πανεπιστήμιο Πατρών, Τμήμα Μαθηματικών, Τμήμα Μηχαικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής, Πάτρα

Πισπιρίγκος Α.,(2014), «Ανάπτυξη λογισμικού για την βελτιστοποίηση της κίνησης γαιών», Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Πολιτικών Μηχανικών, Αθήνα.

Πολίτης Σ., (2016), «Βελτιστοποίηση Οδικών Χράξεων», Διπλωματική εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Αγρονόμων Τοπογράφων Μηχανικών, Αθήνα.

Στάμος Α., (2018), Ανέκδοτες σημειώσεις Οδοποιίας.

Τερζίδου Α., (2013), “Python programming”, Πανεπιστήμιο Δυτικής Μακεδονίας, Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών., Κοζάνη

ΥΠΕΧΩΔΕ, (2001), «Οδηγίες Μελετών Οδικών Έργων (ΟΜΟΕ) – Τεύχος 1: Λειτουργική Κατάταξη Οδικού Δικτύου», Υπουργείο Περιβάλλοντος, Χωροταξίας & Δημοσίων Έργων, Αθήνα 2001

ΥΠΕΧΩΔΕ, (2001), «Οδηγίες Μελετών Οδικών Έργων (ΟΜΟΕ) – Τεύχος 3: Χαράξεις», Υπουργείο Περιβάλλοντος, Χωροταξίας & Δημοσίων Έργων, Αθήνα

Φελούρης Α. Γ., (2009), «Γραμμική Άλγεβρα και Αναλυτική Γεωμετρία», Έκδοση δεύτερη , Αθήνα: Συμεών

Al-Sayed A. Al-Sobky ,(2014), “An Optimization Approach for Highway Vertical Alignment Using the Earthwork Balance Condition”, World Applied Sciences Journal 29 (7): 884-891, Cairo Egypt

Bäck T. and Schwefel H.-P.. (1993). “An Overview of Evolutionary Algorithms for Parameter Optimization”, Evol. Comput., vol. 1, no. 1, pp. 1–23.

BUREAU OF DESIGN AND ENVIRONMENT, (2010), BUREAU OF DESIGN AND ENVIRONMENT MANUAL, “Vertical Alignment”, Chapter 33, Illinois

Coley D. A., (1999), “An Introduction to Genetic Algorithms for Scientists and Engineers”, WORLD SCIENTIFIC.

D. E. (David E. Goldberg and D. E.), (1989), “Genetic algorithms in search, optimization, and machine learning”, Addison-Wesley Longman Publishing Co., Inc..

Darwin C., (1859), “On the Origin of the Species”, vol. 5.

Dessalegn Amenu Hirpa, (2014), “Simultaneous optimization of vertical and horizontal road alignments”, The University Of British Columbia.

Davey N., Dunstall S., Halgamuge S., (2017), “Optimal road design through ecologically sensitive areas considering animal migration dynamics”, Transportation Research, Part C 77, pp. 478–494

Davis L. D. and Mitchell M., (1991), “Handbook of Genetic Algorithms”, VAN NOSTRAND REINHOLD, vol. 15, no. 1, pp. 4–6.

Drezner Z. and Drezner T., (2005), “Genetic Algorithms”, Biomimetics.

Easa S. M., (1987),“SELECTION OF ROADWAY GRADES THAT MINIMIZE EARTHWORK COST USING LINEAR PROGRAMMING”,Rex.-A Vol. 2, No. 2, pp 121-136., Ontario

Fan Tao, (2004), “BI-LEVEL GENETIC ALGORITHM APPROACH FOR 3D ROAD ALIGNMENT OPTIMIZATION”, Thesis for the degree of Master of Engineering Department of Civil Engineering, National University of Singapore

Friedberg R. M., (1958), “A learning machine: Part I”, IBM J. Res. Dev., vol. 2, no. 1, pp. 2–13.

Fwa, T. F., Chan, W. T., & Sim, Y. P. (2002, September). Optimal Vertical Alignment Analysis for Highway Design. Journal of Transportation Engineering, 128(5), pp. 395–402.

Goktepe A. B., Lav A. H. and Altun S., (2009), “ Method for optimal vertical alignment of highways”, Proceedings of the Institution of Civil Engineers, Transport 162, Issue TR4, Pages 177–188

Holland J., (1975), Adaption in Natural and Artificial Systems, vol. 11.

Jong J.-C. (1998). Optimizing Highway Alignments with Genetic Algorithms. PhD dissertation, University of Maryland, Department of Civil Engineering, College Park, MD.

Kang M.-W., Schonfeld P., & Jong, J.-C. (2007, March). Highway Alignment Optimization through Feasible Gates. Journal of Advanced Transportation, 41(2), pp. 115-144.

Kang M. W., Shariat S., (2013), “New highway geometric design methods for minimizing vehicular fuel consumption and improving safety”, Transportation Research Part C Emerging Technologies

Landau D. P., Binder K.,(2009), “A Guide to Monte Carlo Simulationsin Statistical Physics”, Third Edition, Cambridge University Press

Lee W. , Kim H. –Y., (2005), “Genetic Algorithm Implementation in Python”, Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS’05)

Michalewicz Z., (1996), “Genetic Algorithms + Data Structures = Evolution Programs,” Computational Statistics & Data Analysis, vol. 24, no. 3, pp. 372–373

Mitchell M., (1998), “An Introduction to Genetic Algorithms (Complex Adaptive Systems),” MIT Press. no. 6, p. 221, 1996.

Press W., Teukolsky S., Vetterling W., Flannery B., (2007), “Numerical recipes. The art of scientific computing”, Third edition, Cambridge University Press

Transit New Zealand, (2002), “STATE HIGHWAY GEOMETRIC DESIGN MANUAL SECTION 5: VERTICAL ALIGNMENT”, Section 5, New Zealand

http://www.anadelta.com

http://www.diolkos3d.com

http://www.google.com

http://www.imola.odos.gr

http://www.mathesis.gr

Παραρτήματα

Αρχεία Εισόδου Συντεταγμένων

Εγνατία

Χ.Θ.	Υ	Χ.Θ.	Υ	Χ.Θ.	Υ
-1+162.71	88.14	-0+776.57	80.00	-0+389.86	78.00
-1+156.23	89.00	-0+766.64	81.00	-0+384.63	77.80
-1+150.48	90.00	-0+755.00	82.00	-0+374.23	77.40
-1+143.53	91.00	-0+746.94	82.53	-0+363.83	77.00
-1+133.89	92.00	-0+739.65	83.00	-0+344.94	76.00
-1+127.30	93.00	-0+732.64	84.00	-0+333.67	75.00
-1+120.71	94.00	-0+725.30	85.00	-0+329.11	74.00
-1+104.92	95.00	-0+712.91	86.00	-0+324.54	73.00
-1+086.81	95.00	-0+707.30	87.00	-0+316.25	71.99
-1+078.21	95.00	-0+695.96	88.00	-0+310.41	71.00
-1+066.75	94.00	-0+686.51	89.00	-0+303.19	70.00
-1+055.81	93.50	-0+678.88	90.00	-0+295.95	69.00
-1+044.87	93.00	-0+668.08	91.00	-0+288.71	68.00
-1+030.63	92.00	-0+658.71	92.00	-0+281.40	67.00
-1+020.14	91.50	-0+647.55	93.00	-0+271.24	66.00
-1+009.65	91.00	-0+631.83	94.00	-0+265.31	65.00
-0+999.58	90.50	-0+612.04	94.00	-0+259.37	64.00
-0+989.51	90.00	-0+601.31	94.00	-0+252.84	63.00
-0+973.73	89.00	-0+590.59	94.00	-0+244.98	62.00
-0+958.33	88.00	-0+580.42	93.00	-0+241.41	62.00
-0+948.05	87.00	-0+572.73	92.00	-0+229.26	62.00
-0+937.72	86.00	-0+564.09	91.00	-0+217.12	62.00
-0+927.80	85.00	-0+557.64	90.00	-0+201.29	61.00
-0+914.94	84.00	-0+551.80	89.00	-0+194.17	60.00
-0+908.47	83.00	-0+542.14	88.00	-0+190.17	59.00
-0+900.36	82.00	-0+532.28	87.00	-0+186.18	58.00
-0+888.70	81.00	-0+522.39	86.00	-0+182.18	57.00
-0+875.73	80.00	-0+515.50	85.00	-0+170.11	56.42
-0+870.85	79.00	-0+504.89	84.00	-0+161.42	56.00
-0+865.96	78.00	-0+498.60	83.53	-0+158.12	55.97
-0+861.56	78.00	-0+491.62	83.00	-0+139.24	55.78
-0+858.21	79.00	-0+480.01	82.39	-0+119.41	55.59
-0+855.15	80.00	-0+472.52	82.00	-0+099.78	55.40
-0+842.20	80.56	-0+461.32	81.64	-0+081.27	55.22
-0+832.16	81.00	-0+451.21	81.32	-0+069.97	55.11
-0+822.57	81.00	-0+441.10	81.00	-0+058.67	55.00
-0+811.21	81.00	-0+431.95	80.00	-0+055.23	54.00
-0+799.29	80.00	-0+423.26	79.49	-0+051.68	53.00
-0+796.46	79.00	-0+414.88	79.00	-0+047.53	52.00
-0+786.29	79.00	-0+404.11	78.57	-0+038.42	52.00

X.Θ.	Y	X.Θ.	Y	X.Θ.	Y
-0+033.77	53.00	0+396.43	74.00	0+848.56	54.00
-0+028.87	54.00	0+405.59	74.37	0+851.26	53.00
-0+023.96	55.00	0+421.54	75.00	0+865.53	53.00
-0+010.82	55.52	0+433.05	75.00	0+877.86	53.00
0+000.00	55.95	0+444.56	75.00	0+880.18	53.00
0+001.32	56.00	0+456.13	74.00	0+887.07	53.00
0+015.92	57.00	0+465.79	73.00	0+905.03	53.60
0+030.18	58.00	0+473.30	72.00	0+916.77	54.00
0+040.16	59.00	0+474.38	71.88	0+926.76	55.00
0+046.77	60.00	0+482.65	71.00	0+944.15	55.00
0+051.85	61.00	0+495.52	70.00	0+962.44	55.00
0+056.28	62.00	0+505.55	69.50	0+977.46	55.00
0+060.73	63.00	0+515.57	69.00	0+984.58	55.00
0+065.52	64.00	0+525.27	68.00	1+000.57	55.00
0+070.45	65.00	0+531.66	67.49	1+011.36	55.00
0+075.11	66.00	0+537.75	67.00	1+022.15	55.00
0+084.52	67.00	0+551.97	66.00	1+032.18	54.50
0+096.18	68.00	0+554.16	66.00	1+042.20	54.00
0+099.01	68.00	0+571.12	65.00	1+052.23	53.50
0+101.43	68.00	0+590.56	64.00	1+062.27	53.00
0+108.35	69.00	0+607.72	63.41	1+078.03	52.00
0+114.08	69.19	0+619.51	63.00	1+093.83	52.00
0+132.76	69.81	0+632.51	62.36	1+097.68	52.00
0+138.69	70.00	0+639.77	62.00	1+116.55	52.00
0+156.45	71.00	0+650.73	61.69	1+119.20	53.00
0+172.00	71.14	0+668.92	61.17	1+120.21	54.00
0+190.39	71.00	0+675.01	61.00	1+121.25	55.00
0+209.40	71.00	0+685.70	60.00	1+128.63	55.00
0+219.05	71.00	0+688.55	60.00	1+136.15	54.00
0+227.43	70.00	0+700.63	59.00	1+156.09	53.68
0+237.19	69.57	0+706.61	58.72	1+174.21	53.40
0+249.77	69.00	0+720.15	58.00	1+193.99	53.08
0+258.21	68.00	0+735.80	57.00	1+199.34	53.00
0+268.16	67.00	0+740.87	57.00	1+212.48	52.68
0+270.02	67.00	0+751.30	56.50	1+231.12	52.23
0+285.44	68.00	0+761.72	56.00	1+240.73	52.00
0+296.22	69.00	0+777.72	55.57	1+250.12	52.00
0+299.85	69.00	0+790.63	55.23	1+263.23	52.00
0+312.30	69.00	0+799.24	55.00	1+269.37	52.00
0+323.33	70.00	0+808.87	54.52	1+289.02	52.00
0+330.46	70.48	0+819.16	54.00	1+308.24	52.00
0+338.14	71.00	0+821.47	54.00	1+327.29	52.00
0+353.29	72.00	0+827.89	54.00	1+345.81	52.00
0+370.90	73.00	0+842.52	54.00	1+364.99	51.77
0+385.69	73.58	0+845.26	54.00	1+380.65	52.00

Χ.Θ.	Υ	Χ.Θ.	Υ	Χ.Θ.	Υ
1+391.73	52.00	1+953.09	44.00	2+506.16	36.00
1+402.81	52.00	1+955.75	44.00	2+511.81	36.00
1+422.31	52.00	1+966.26	44.00	2+514.35	37.00
1+433.62	52.00	1+976.77	44.00	2+524.45	37.50
1+444.93	52.00	1+990.77	43.00	2+534.55	38.00
1+460.77	51.30	1+993.25	43.00	2+553.32	38.70
1+467.61	51.00	2+009.29	43.00	2+561.09	39.00
1+479.63	51.00	2+028.64	42.00	2+562.23	39.00
1+482.67	51.00	2+032.78	42.00	2+571.39	39.46
1+490.30	51.00	2+050.22	42.00	2+582.24	40.00
1+499.36	51.00	2+056.59	42.00	2+589.77	40.41
1+517.84	51.00	2+070.04	42.00	2+600.63	41.00
1+537.02	51.00	2+088.72	42.00	2+606.89	41.00
1+556.72	51.00	2+093.82	42.00	2+622.37	41.64
1+576.31	51.00	2+097.09	42.00	2+631.36	42.00
1+595.56	50.91	2+099.85	42.00	2+640.27	42.00
1+615.11	51.00	2+108.38	42.00	2+653.03	42.00
1+633.46	51.00	2+121.03	42.00	2+661.19	42.12
1+652.68	51.00	2+131.13	42.00	2+680.40	42.40
1+671.29	51.00	2+142.38	42.00	2+700.34	42.69
1+691.11	50.73	2+150.53	41.91	2+710.87	42.85
1+701.76	51.00	2+168.60	41.70	2+721.39	43.00
1+704.72	52.00	2+188.29	41.47	2+733.69	43.00
1+710.46	52.00	2+207.85	41.25	2+746.03	42.64
1+726.16	52.00	2+218.81	41.13	2+757.92	42.28
1+728.21	51.00	2+229.76	41.00	2+766.90	42.00
1+730.23	50.00	2+244.65	40.65	2+774.21	42.00
1+731.31	50.00	2+263.33	40.24	2+776.18	42.00
1+738.08	50.00	2+274.59	40.00	2+787.08	41.50
1+749.74	49.74	2+282.37	39.80	2+797.97	41.00
1+768.32	49.32	2+301.65	39.33	2+802.82	41.00
1+782.63	49.00	2+315.94	39.00	2+813.49	40.54
1+791.27	49.00	2+327.63	38.56	2+826.00	40.00
1+798.06	49.00	2+344.24	38.00	2+832.92	39.76
1+806.81	49.00	2+355.25	37.90	2+843.71	39.38
1+815.36	49.00	2+366.26	37.80	2+854.49	39.00
1+825.65	48.68	2+384.74	37.63	2+864.76	38.50
1+845.43	48.00	2+403.91	37.47	2+875.03	38.00
1+861.62	47.00	2+421.91	37.33	2+883.57	38.00
1+881.25	46.00	2+439.93	37.19	2+893.62	37.00
1+901.24	45.22	2+458.14	37.02	2+903.74	36.77
1+906.97	45.00	2+463.21	37.00	2+917.12	36.45
1+919.50	44.71	2+476.98	37.00	2+935.04	36.00
1+934.64	44.32	2+496.09	37.00	2+939.43	36.00
1+950.53	44.00	2+503.30	37.00	2+958.91	36.00

Χ.Θ.	Υ
2+974.18	35.40
2+984.17	35.00
2+989.14	35.00
3+001.40	35.00
3+009.92	35.00
3+020.36	35.00
3+030.80	35.00
3+042.50	35.00
3+054.19	35.00
3+067.45	35.00
3+077.69	35.00
3+081.94	35.00
3+099.73	35.00
3+109.03	35.00
3+113.61	35.00
3+131.98	35.00
3+150.81	35.00
3+162.58	35.00
3+174.36	35.00
3+188.44	35.00
3+207.21	35.00
3+225.70	35.00
3+244.00	35.00
3+257.86	35.00
3+263.08	35.00
3+275.19	35.00
3+287.30	35.00
3+296.85	34.90
3+310.11	34.78
3+328.04	34.58
3+347.41	34.38
3+366.55	34.18
3+383.68	34.00
3+397.36	34.00
3+403.71	34.00

$

Επαρχιακή Οδός Νομού Μεσσηνίας

Χ. Θ.	Y
0+000.00	74.43
0+016.81	74.28
0+033.62	74.17
0+053.62	74.02
0+068.12	73.93
0+082.62	73.80
0+102.62	73.50
0+119.16	73.45
0+135.69	73.15
0+155.69	73.30
0+172.23	73.17
0+188.76	73.07
0+208.76	72.95
0+223.26	72.87
0+237.76	72.77
0+257.76	72.60
0+277.76	72.48
0+297.76	72.33
0+317.76	72.17
0+337.76	72.05
0+357.76	71.90
0+371.98	71.80
0+386.19	71.72
0+406.19	71.62
0+426.19	71.50
0+446.19	71.30
0+460.63	71.33
0+475.08	71.32
0+495.08	71.10
0+515.08	71.20
0+525.13	71.14
0+535.18	71.09
0+555.18	71.00
0+575.18	70.93
0+585.23	70.92
0+595.27	70.87
0+615.27	70.85
0+635.27	70.78
0+655.27	70.70
0+669.72	70.65
0+684.16	70.59
0+692.91	7.57

Δρυόπη – Γαλατάς

Χ.Θ.	Υ	Χ.Θ.	Υ	Χ.Θ.	Υ
3+000.00	11.27	3+540.00	18.44	3+980.00	9.87
3+017.18	11.02	3+550.00	18.05	3+986.11	9.90
3+020.00	10.96	3+560.07	17.75	4+000.00	10.00
3+040.00	10.83	3+580.00	17.22	$	
3+042.19	10.83	3+587.04	17.09		
3+060.00	8.23	3+600.00	16.83		
3+067.18	8.20	3+612.05	16.55		
3+080.00	7.98	3+620.00	16.38		
3+100.00	7.80	3+637.04	15.90		
3+120.00	7.66	3+640.00	15.85		
3+140.00	7.60	3+648.39	15.71		
3+160.00	7.49	3+660.00	15.50		
3+180.00	7.50	3+680.00	15.14		
3+200.00	7.49	3+685.62	15.02		
3+220.00	8.19	3+700.00	14.72		
3+240.00	8.55	3+720.00	14.33		
3+243.08	8.63	3+722.85	14.27		
3+260.00	8.08	3+740.00	13.88		
3+270.84	8.68	3+747.42	13.71		
3+280.00	9.12	3+760.00	13.38		
3+298.64	10.21	3+771.98	13.03		
3+300.00	10.34	3+780.00	12.76		
3+313.75	12.40	3+800.00	11.98		
3+316.76	15.75	3+809.21	11.62		
3+320.00	15.71	3+820.00	11.26		
3+340.00	15.86	3+840.00	10.79		
3+356.87	16.26	3+846.44	10.63		
3+360.00	16.29	3+860.00	10.34		
3+380.00	16.96	3+873.19	10.11		
3+400.00	17.51	3+880.00	10.09		
3+415.11	17.93	3+887.10	10.10		
3+420.00	18.06	3+900.00	10.10		
3+440.00	18.43	3+901.02	10.09		
3+442.90	18.48	3+919.18	10.03		
3+460.00	18.73	3+920.00	10.03		
3+470.66	18.89	3+937.34	9.84		
3+480.00	19.07	3+940.00	9.82		
3+483.10	19.05	3+951.26	9.78		
3+500.00	18.99	3+960.00	9.80		
3+508.09	18.89	3+965.17	9.82		
3+520.00	19.04	3+966.11	9.83		
3+533.10	18.73	3+976.11	9.86		

Ριζόμυλος – Κορώνη

Χ.Θ.	Υ	Χ.Θ.	Υ	Χ.Θ.	Υ
3+000.00	11.27	3+540.00	18.44	3+980.00	9.87
3+017.18	11.02	3+550.00	18.05	3+986.11	9.90
3+020.00	10.96	3+560.07	17.75	4+000.00	10.00
3+040.00	10.83	3+580.00	17.22	$	
3+042.19	10.83	3+587.04	17.09		
3+060.00	8.23	3+600.00	16.83		
3+067.18	8.20	3+612.05	16.55		
3+080.00	7.98	3+620.00	16.38		
3+100.00	7.80	3+637.04	15.90		
3+120.00	7.66	3+640.00	15.85		
3+140.00	7.60	3+648.39	15.71		
3+160.00	7.49	3+660.00	15.50		
3+180.00	7.50	3+680.00	15.14		
3+200.00	7.49	3+685.62	15.02		
3+220.00	8.19	3+700.00	14.72		
3+240.00	8.55	3+720.00	14.33		
3+243.08	8.63	3+722.85	14.27		
3+260.00	8.08	3+740.00	13.88		
3+270.84	8.68	3+747.42	13.71		
3+280.00	9.12	3+760.00	13.38		
3+298.64	10.21	3+771.98	13.03		
3+300.00	10.34	3+780.00	12.76		
3+313.75	12.40	3+800.00	11.98		
3+316.76	15.75	3+809.21	11.62		
3+320.00	15.71	3+820.00	11.26		
3+340.00	15.86	3+840.00	10.79		
3+356.87	16.26	3+846.44	10.63		
3+360.00	16.29	3+860.00	10.34		
3+380.00	16.96	3+873.19	10.11		
3+400.00	17.51	3+880.00	10.09		
3+415.11	17.93	3+887.10	10.10		
3+420.00	18.06	3+900.00	10.10		
3+440.00	18.43	3+901.02	10.09		
3+442.90	18.48	3+919.18	10.03		
3+460.00	18.73	3+920.00	10.03		
3+470.66	18.89	3+937.34	9.84		
3+480.00	19.07	3+940.00	9.82		
3+483.10	19.05	3+951.26	9.78		
3+500.00	18.99	3+960.00	9.80		
3+508.09	18.89	3+965.17	9.82		
3+520.00	19.04	3+966.11	9.83		
3+533.10	18.73	3+976.11	9.86		

Κερατέα – Ανάβυσσος

Χ.Θ.	Υ	Χ.Θ.	Υ	Χ.Θ.	Υ
0+000.00	148.15	0+599.77	145.13	1+220.30	145.97
0+016.97	148.16	0+619.47	145.14	1+238.36	146.19
0+033.95	148.17	0+639.05	145.17	1+257.45	146.41
0+047.80	148.18	0+657.43	145.24	1+275.48	146.62
0+061.65	148.19	0+676.91	145.27	1+295.31	146.81
0+080.28	148.20	0+685.21	145.30	1+314.02	147.00
0+099.09	148.21	0+688.58	145.32	1+333.63	147.05
0+101.90	148.24	0+691.95	145.34	1+352.73	147.16
0+108.79	148.26	0+709.78	145.43	1+356.78	147.20
0+115.67	148.24	0+727.39	145.45	1+363.78	147.27
0+134.93	148.16	0+730.33	145.44	1+370.77	147.35
0+153.44	148.10	0+733.27	145.45	1+390.02	147.65
0+173.26	148.02	0+752.32	145.43	1+408.13	148.09
0+188.65	147.97	0+771.31	145.30	1+427.36	148.77
0+198.29	147.94	0+789.85	145.04	1+446.43	149.57
0+207.93	147.91	0+792.34	145.02	1+456.37	149.96
0+227.34	147.94	0+811.03	144.93	1+467.06	150.28
0+245.47	148.02	0+827.31	144.93	1+477.55	150.59
0+263.73	148.03	0+845.42	144.91	1+489.16	150.80
0+282.31	148.06	0+862.27	144.94	1+498.72	150.93
0+301.23	147.67	0+882.23	145.07	1+518.58	150.84
0+314.92	147.24	0+901.02	145.12	1+537.54	150.84
0+317.70	147.16	0+919.91	145.20	1+557.25	150.77
0+320.48	147.08	0+939.49	145.31	1+571.28	150.87
0+338.67	146.55	0+958.37	145.41	1+585.31	150.92
0+357.12	146.34	0+967.76	145.46	1+604.19	151.37
0+375.53	146.21	0+973.52	145.50	1+623.66	152.10
0+395.16	146.07	0+979.28	145.53	1+641.82	152.69
0+413.67	145.99	0+998.63	145.71	1+646.36	152.79
0+432.06	145.81	1+018.31	146.10	1+661.85	153.14
0+437.40	145.75	1+036.70	146.45	1+677.33	153.27
0+443.72	145.69	1+039.72	146.51	1+695.92	153.11
0+450.03	145.65	1+052.57	146.64	1+715.13	152.96
0+469.86	145.46	1+063.46	146.63	1+733.71	153.04
0+488.66	145.41	1+076.03	146.37	1+752.77	153.14
0+506.79	145.33	1+087.21	146.18	1+771.90	153.28
0+525.72	145.22	1+106.41	145.90	1+790.80	153.54
0+544.30	145.18	1+125.91	145.67	1+809.22	154.02
0+563.71	145.13	1+144.97	145.51	1+828.00	154.74
0+583.33	145.04	1+163.20	145.48	1+846.25	155.34
0+592.46	145.09	1+182.50	145.62	1+865.36	155.93
0+596.12	145.12	1+201.13	145.75	1+883.39	156.17

X.Θ.	Y	X.Θ.	Y	X.Θ.	Y
1+895.59	156.08	2+627.22	136.24	3+231.29	145.37
1+907.23	155.88	2+639.35	136.46	3+249.61	145.79
1+919.21	155.54	2+648.03	136.64	3+266.18	146.32
1+931.07	155.01	2+666.14	136.75	3+281.06	146.71
1+950.34	154.03	2+685.23	136.95	3+293.60	147.00
1+969.73	153.07	2+705.06	137.16	3+307.97	147.26
1+989.19	152.06	2+724.94	137.36	3+321.03	147.56
2+005.74	151.15	2+733.87	137.45	3+335.19	147.88
2+024.86	150.06	2+747.00	137.57	3+351.87	148.23
2+043.84	149.09	2+748.89	137.58	3+359.58	148.36
2+063.29	148.24	2+763.92	137.67	3+371.56	148.55
2+081.95	147.28	2+783.79	137.82	3+391.31	148.91
2+100.81	146.26	2+803.60	138.01	3+407.93	149.02
2+120.73	145.23	2+816.44	138.11	3+421.41	149.15
2+139.43	144.17	2+826.54	138.19	3+433.33	149.21
2+155.92	143.28	2+842.66	138.34	3+448.02	149.27
2+173.58	142.27	2+862.04	138.57	3+462.71	149.32
2+178.89	141.99	2+873.13	138.72	3+478.67	149.36
2+182.03	141.80	2+884.21	138.87	3+493.57	149.44
2+195.37	141.15	2+901.16	139.07	3+500.71	149.45
2+208.13	140.50	2+909.01	139.22	3+507.85	149.46
2+224.85	139.61	2+915.03	139.23	3+524.06	149.49
2+239.11	138.83	2+921.04	139.24	3+538.19	149.51
2+253.36	138.10	2+921.08	139.30	3+542.48	149.50
2+272.86	137.14	2+939.36	139.63	3+546.77	149.50
2+291.29	136.12	2+959.35	140.01	3+566.68	149.61
2+309.68	135.19	2+978.03	140.37	3+584.79	149.59
2+328.10	134.39	2+979.77	140.42	3+603.92	149.62
2+347.77	134.13	2+986.33	140.54	3+623.66	149.63
2+366.00	134.11	2+992.89	140.65	3+642.22	149.58
2+384.52	134.22	3+012.06	140.94	3+649.14	149.68
2+403.28	134.27	3+024.77	141.06	3+662.49	150.08
2+421.06	134.30	3+028.35	141.16	3+675.83	150.37
2+440.38	134.32	3+038.44	141.45	3+695.32	150.83
2+457.61	134.37	3+052.11	141.67	3+698.46	150.92
2+475.74	134.48	3+071.30	142.02	3+701.61	151.05
2+494.18	134.67	3+090.76	142.38	3+721.01	151.70
2+513.37	134.88	3+110.61	142.74	3+739.83	152.43
2+526.99	135.06	3+130.02	143.07	3+757.85	153.14
2+538.50	135.22	3+149.27	143.45	3+777.39	154.03
2+548.80	135.33	3+168.00	143.87	3+783.98	154.30
2+560.23	135.46	3+187.38	144.33	3+795.40	154.73
2+570.62	135.59	3+205.45	144.74	3+806.61	155.09
2+588.74	135.77	3+208.42	144.81	3+818.77	155.61
2+608.52	135.92	3+211.39	144.88	3+829.25	155.86

Χ.Θ.	Υ	Χ.Θ.	Υ	Χ.Θ.	Υ
3+848.04	156.31	4+447.40	165.93	4+608.22	156.00
3+867.09	156.63	4+459.45	166.00	4+617.99	155.00
3+886.92	156.83	4+464.05	165.00	4+621.86	154.64
3+905.96	157.11	4+464.45	165.00	4+627.82	154.00
3+924.34	157.35	4+464.55	165.00	4+627.86	154.00
3+942.40	157.69	4+466.45	165.00	4+628.33	154.00
3+945.78	157.75	4+466.95	165.00	4+641.45	153.00
3+948.62	157.81	4+466.96	165.00	4+641.77	153.00
3+951.46	157.87	4+467.51	165.00	4+642.24	153.00
3+969.92	158.21	4+471.50	165.00	4+643.14	153.00
3+989.42	158.62	4+483.88	164.00	4+654.48	153.00
4+008.40	159.01	4+483.91	164.00	4+663.83	152.00
4+027.77	159.50	4+484.36	164.00	4+664.41	152.00
4+031.28	159.60	4+492.04	163.82	4+678.44	151.23
4+035.13	159.70	4+498.20	163.00	4+679.98	151.00
4+038.96	159.81	4+498.41	163.00	4+687.96	150.84
4+058.69	160.34	4+498.62	163.00	4+695.97	150.00
4+078.28	160.88	4+500.15	162.94	4+697.48	149.97
4+096.37	161.37	4+508.25	162.89	4+699.05	150.00
4+100.74	161.49	4+519.89	162.00	4+699.07	150.00
4+105.10	161.58	4+519.94	162.00	4+699.12	150.00
4+123.58	161.93	4+520.06	162.00	4+699.57	150.00
4+142.62	162.21	4+520.38	162.00	4+699.87	150.00
4+161.46	162.76	4+520.40	162.00	4+700.10	150.00
4+179.48	163.10	4+520.68	162.00	4+705.36	149.00
4+198.37	163.45	4+521.00	162.00	4+719.58	148.00
4+202.37	163.75	4+530.12	161.00	4+719.76	148.00
4+213.30	164.10	4+531.50	161.00	4+720.16	148.00
4+224.22	164.20	4+532.30	161.00	4+721.16	148.00
4+244.07	164.60	4+539.35	160.82	4+721.19	148.00
4+263.32	164.94	4+554.01	160.49	4+738.87	147.00
4+265.01	165.00	4+559.05	161.00	4+740.24	146.91
4+273.08	166.05	4+565.00	161.00	4+747.58	146.85
4+281.14	165.25	4+565.04	161.00	4+753.86	146.00
4+299.46	165.50	4+568.67	161.49	4+754.00	146.00
4+317.73	165.67	4+572.42	162.00	4+754.40	146.00
4+337.61	165.85	4+581.09	162.00	4+754.92	145.93
4+357.38	166.06	4+589.79	161.00	4+769.44	145.00
4+376.19	166.27	4+596.47	160.00	4+774.45	144.74
4+378.99	166.29	4+597.99	159.00	4+786.44	144.00
4+392.65	166.36	4+597.99	159.00	4+789.34	143.77
4+406.31	166.41	4+600.38	158.00	4+803.64	142.90
4+424.69	166.35	4+600.56	158.00	4+818.99	141.78
4+443.58	166.01	4+600.62	158.00	4+832.83	140.71
4+446.18	166.00	4+604.43	157.00	4+851.72	139.47

X.Θ.	Y	X.Θ.	Y	X.Θ.	Y
4+871.62	138.18	5+420.10	107.44	6+030.61	75.93
4+890.32	137.00	5+431.44	106.73	6+049.89	75.33
4+909.97	135.82	5+442.77	106.02	6+069.04	74.71
4+920.11	135.34	5+458.34	105.09	6+086.97	74.08
4+930.26	134.86	5+476.95	103.87	6+092.93	73.88
4+946.83	134.30	5+496.53	102.68	6+098.90	73.70
4+949.59	134.20	5+515.17	101.45	6+117.00	73.13
4+964.17	133.68	5+533.58	100.40	6+136.92	72.53
4+977.81	133.23	5+552.17	99.17	6+156.88	71.98
4+992.74	132.70	5+571.13	97.89	6+176.62	71.44
5+006.03	132.26	5+590.71	96.70	6+195.32	70.97
5+025.99	131.59	5+605.86	95.72	6+212.24	70.56
5+045.40	130.97	5+617.41	95.02	6+220.47	70.34
5+063.55	130.31	5+628.96	94.31	6+228.69	70.11
5+065.93	130.25	5+647.76	93.15	6+246.79	69.61
5+071.74	129.94	5+653.25	92.82	6+266.20	69.03
5+076.01	129.76	5+666.74	91.92	6+285.58	68.53
5+086.09	129.23	5+685.43	90.61	6+304.85	68.11
5+104.22	127.99	5+703.93	89.24	6+324.70	67.68
5+120.30	127.01	5+723.20	88.03	6+333.66	67.49
5+125.46	126.69	5+743.15	86.86	6+337.45	67.43
5+130.62	126.38	5+763.08	85.71	6+341.23	67.37
5+150.36	125.12	5+769.61	85.32	6+360.26	67.01
5+169.60	123.88	5+776.71	84.88	6+378.04	66.69
5+188.21	122.71	5+783.81	84.45	6+381.60	66.62
5+206.52	121.52	5+803.40	83.31	6+385.17	66.54
5+225.77	120.28	5+822.97	82.34	6+404.04	66.18
5+227.33	120.19	5+827.93	82.11	6+423.56	65.75
5+234.19	119.74	5+830.24	82.03	6+442.19	65.37
5+241.06	119.28	5+832.56	81.88	6+461.01	65.04
5+260.67	117.89	5+850.96	81.25	6+479.18	64.71
5+279.19	116.64	5+867.06	80.66	6+497.73	64.30
5+286.60	116.14	5+873.14	80.55	6+516.36	63.78
5+292.57	115.73	5+879.22	80.36	6+518.71	63.73
5+298.28	115.42	5+898.58	79.89	6+521.05	63.67
5+303.40	115.12	5+916.78	79.48	6+540.28	63.20
5+307.96	114.88	5+924.28	79.26	6+560.01	62.71
5+312.51	114.59	5+931.66	79.03	6+565.41	62.64
5+331.46	113.40	5+939.04	78.78	6+570.58	62.58
5+351.20	112.11	5+957.16	78.19	6+575.74	62.48
5+369.03	110.91	5+977.02	77.66	6+594.07	62.02
5+379.62	110.19	5+996.26	77.02	6+610.80	61.61
5+390.21	109.48	6+005.43	76.73	6+616.57	61.45
5+402.80	108.59	6+008.27	76.63	6+622.35	61.29
5+411.45	108.03	6+011.10	76.53	6+641.73	60.88

Χ.Θ.	Υ	Χ.Θ.	Υ	Χ.Θ.	Υ
6+661.48	60.51	7+384.11	51.34	8+109.64	48.56
6+679.97	60.15	7+402.96	51.14	8+123.85	48.70
6+686.37	59.99	7+422.54	50.94	8+142.01	48.91
6+692.16	59.84	7+441.04	50.75	8+161.75	49.10
6+697.96	59.70	7+459.32	50.44	8+181.69	49.11
6+717.49	59.18	7+477.75	49.83	8+199.72	49.12
6+736.00	58.85	7+496.83	49.52	8+203.74	49.12
6+754.65	58.56	7+515.71	49.15	8+218.29	49.22
6+773.59	58.29	7+534.46	48.79	8+232.85	48.97
6+791.94	58.00	7+553.46	48.41	8+251.05	48.39
6+810.07	57.80	7+571.68	48.13	8+270.34	47.58
6+829.20	57.60	7+582.78	47.97	8+274.70	47.41
6+848.48	57.42	7+593.35	47.81	8+282.05	47.11
6+850.87	57.41	7+603.93	47.63	8+289.40	46.77
6+869.69	57.34	7+622.96	47.28	8+308.36	46.09
6+888.52	57.29	7+641.20	47.01	8+327.76	45.42
6+908.17	57.11	7+660.59	46.80	8+347.27	44.75
6+927.15	56.94	7+679.42	46.58	8+367.26	44.07
6+945.84	56.77	7+697.83	46.34	8+367.36	44.06
6+962.73	56.58	7+716.69	46.08	8+386.41	43.53
6+973.90	56.42	7+735.14	45.92	8+405.45	42.80
6+985.08	56.20	7+753.30	45.75	8+425.01	41.84
6+998.64	55.81	7+772.30	45.61	8+444.75	40.97
7+015.40	55.35	7+781.11	45.53	8+464.45	40.23
7+030.02	55.04	7+790.16	45.43	8+482.45	39.65
7+047.06	54.87	7+799.20	45.39	8+502.06	39.06
7+061.41	54.73	7+817.21	45.41	8+505.70	38.96
7+080.41	54.56	7+837.01	45.42	8+520.26	38.59
7+098.69	54.32	7+843.91	45.33	8+534.35	38.19
7+118.41	54.18	7+863.25	45.34	8+548.99	37.93
7+135.68	54.07	7+883.22	45.37	8+562.99	37.59
7+155.30	53.98	7+899.46	45.48	8+581.62	37.35
7+172.66	53.84	7+919.23	45.57	8+600.76	37.30
7+191.58	53.60	7+938.37	45.74	8+613.21	37.31
7+211.02	53.31	7+954.99	45.94	8+627.21	37.26
7+230.34	53.09	7+973.19	46.12	8+640.04	36.98
7+249.02	52.88	7+993.07	46.29	8+653.50	36.35
7+263.49	52.73	8+011.89	46.64	8+666.87	35.88
7+272.28	52.68	8+025.47	46.90	8+685.46	35.36
7+281.07	52.62	8+030.42	47.01	8+704.54	35.05
7+301.02	52.41	8+035.37	47.12	8+724.19	34.80
7+319.61	52.14	8+053.96	47.46	8+742.94	34.61
7+338.72	51.87	8+073.02	47.86	8+761.25	34.49
7+354.39	51.68	8+092.49	48.24	8+779.98	34.30
7+369.25	51.50	8+095.43	48.32	8+799.05	34.16

Χ.Θ.	Υ
8+818.39	34.10
8+829.08	34.09
8+846.08	34.21
8+863.08	34.48
8+876.33	34.66
8+891.97	34.82
8+907.61	35.39
8+927.46	36.14
8+946.76	36.70
8+949.60	36.76
8+952.44	36.82
8+972.18	37.19
8+990.74	37.14
8+994.99	37.13

$

Οδός Κιάφα

Χ.Θ.	Υ	Χ.Θ.	Υ
0.00	109.46	2286.58	73.75
34.49	108.37	2334.13	71.46
73.14	107.32	2406.45	68.04
107.70	106.41	2462.76	65.69
139.28	105.75	2490.94	65.55
146.85	105.87	2146.21	75.04
155.54	105.83	2201.67	76.30
218.77	101.75	2286.58	73.75
261.14	100.69	2334.13	71.46
298.29	99.66	2406.45	68.04
353.89	97.79	2462.76	65.69
409.06	96.03	2490.94	65.55
472.00	94.47	2526.53	65.36
529.11	92.98	2615.90	64.67
571.32	91.97	2688.31	64.34
638.34	90.76	2740.38	64.58
705.88	89.67	2799.11	64.32
764.73	89.09	2855.20	64.30
803.22	88.73	2928.50	65.09
832.29	88.39	2992.88	66.64
865.95	88.19	3075.58	69.57
905.55	87.51	3136.97	71.20
945.59	87.03	3182.78	72.96
978.66	86.52	3235.37	74.53
1015.91	85.94	3302.68	77.80
1027.36	85.69	3371.40	82.14
1102.70	84.32	3432.53	84.58
1160.04	83.42	3481.38	86.53
1249.38	82.13	3556.78	88.49
1316.54	81.01	3613.87	92.84
1405.59	80.12	3660.37	95.86
1468.49	79.32	3683.61	96.95
1518.45	78.75	3713.16	97.86
1565.23	78.28	3760.22	98.84
1613.34	77.53	3808.53	99.41
1660.96	76.88	3882.95	101.86
1723.25	75.60	3939.51	105.43
1766.55	74.59	3984.95	109.20
1852.72	73.72	4014.80	110.11
1921.23	73.33	$	
1997.16	73.35		
2085.57	73.78		
2146.21	75.04		
2201.67	76.30		

Αλγόριθμος

Μέθοδος Monte Carlo

```
import fdata
# import dxf
import repeat_xy
import geom

def pyMain():
    """Start here."""
    fn = "kalamata"
    fr = open(fn+".mhk", "r", encoding="iso-8859-7")
    amhk, xth, y, name = fdata.readData(fr)
    #fdata.plotmhk(amhk, xth, y, name)
    pl = fdata.PlotManyMhk()
    na = int((xth[-1] - xth[0])//500)
    ncarlo = 3
    for a in range(1, na+1):
        for i in range(ncarlo+1):
            print("Monte Carlo: nodes={}/{}   try={}/{}".format(a, na, i,
ncarlo))
            po = repeat_xy.monteCarlo(xth, y, a)
            repeat_xy.saveCarlo(fn, po)
            # pl.plotmhk("MonteCarlo_nodes={}_area={:.1f}".format(po.a, po.emb),
xth, y, name, po.xers, po.yers)
            print("    Area: {:.2f}  (geom={:.2f}  overlapping={:.2f}
klish={:.2f})".format(po.emb, po.embGeom, po.embOverlapping, po.embKlish))
            s = geom.klisi(po.xers, po.yers)
            # fdata.showmhk(amhk, xth, y, name, po.xers, po.yers) #  Συναρτήσεις
για προβολή σκαριφήματος της ερυθράς
            # for z in range(0, len(po.xers)-1, 2):
            #     x_cur = [a[0] for a in po.edafery()]
            #     y_cur = [a[2] for a in po.edafery()]
            #     fdata.show_curve(x_cur, y_cur)
            for slope in s:
                print("    Slope : {:.2f}%".format(slope*100))
            print("    main", po.yers, po.xers)
            # fdata.showend()
    # pl.plotend()
    fr.close()

pyMain()
```

Αρχείο fdata

```
import matplotlib.pyplot as plt
import numpy
import dxf
from scipy.interpolate import spline
```

```
def readData(f):
    "Read ground data from a file."
    # αρχικοποίηση λίστας.
    fxth = []
    fy = []
    fname = []
    iline = 0
    for amhk in f:
        iline += 1
        break
    for line in f:
        iline += 1
        if line.strip() == "$": break
        # σπάμε το string(line) όπου έχει κενό
        # επειδή μας ενδιαφέρουν μόνο οι γραμμές  που έχουν τρία στοιχεία
        try:
            a, b, c = line.split()
            fxth.append(float(a))
            fy.append(float(b))
            fname.append(c)
        except (IndexError, ValueError):
            raise ValueError("Syntax error at line {} of file {}".format(iline,
f.name))
    return amhk, fxth, fy, fname


def writeRes(fw, xers, yers, emb):
    "Write the results to a file."
    fw.write("Area: {:.2f}\n".format(emb))
    fw.write("Grade points:\n")
    for x1, y1 in zip(xers, yers):
        fw.write("{:15.3f}{:15.3f}\n".format(x1, y1))
    fw.write("$\n")


def plotmhk(amhk, fxth, fy, fname, xers=(), yers=()):
    "Plots the profile in dxf format, both ground and grade line."
    dxf.plotstart()
    plotmhk1(amhk, fxth, fy, fname, xers, yers)
    dxf.plotend()

def plotmhk1(amhk, fxth, fy, fname, xers=(), yers=()):
    "Plots the profile in dxf format, both ground and grade line."
    dxf.plotsetlayer("EDAFOS")
    h = 10.0*10   #text size/height
    dxf.plotsymbol(fxth[0]-(len(amhk)+2)*h, fy[0]*10, h, amhk, 0.0)

    for m in range(1, len(fxth)):
        dxf.plotline(fxth[m - 1], fy[m - 1] * 10, fxth[m], fy[m] * 10)
        # plotline(fxth[m-1],fy[m-1]*10, fxth[m-1], min(fy)-)
        # plotsymbol(x1,y1,0.1,name[m-1],0)
    dxf.plotsetlayer("ERYTHRA")
```

```
    for m in range(1, len(xers)):
        dxf.plotline(xers[m - 1], yers[m - 1] * 10, xers[m], yers[m] * 10)


class PlotManyMhk:
    "Call to plot multiple profiles to a single dxf."
    def __init__(self):
        "Open new drawing and set horizon."
        dxf.plotstart()
        self.hor = 0.0   #horizon (min of the profile on the drawing)

    def plotmhk(self, amhk, fxth, fy, fname, xers=(), yers=()):
        "Plots the profile in dxf format, both ground and grade line."
        temp = tuple(fy)+tuple(yers)
        fymin = min(temp)
        dy = max(temp) - fymin
        self.hor -= dy*1.1
        fyn = [self.hor+(temp-fymin) for temp in fy]
        yersn = [self.hor+temp-fymin for temp in yers]
        plotmhk1(amhk, fxth, fyn, fname, xers, yersn)

    def plotend(self):
        "Close the drawing."
        dxf.plotend()


def showmhk(amhk, fxth, fy, fname, xers=(), yers=()):
    "Plots the profile to a pyplot diagram."
    plt.figure(figsize=(1000, 1))
    plt.plot(fxth, fy, 'g-', xers, yers, 'b')
    # plt.axis('scaled')


def show_curve(x_c, y_c):
    # y = s1/100*x+x^2/(2*h)
    '''xnew = numpy.linspace(numpy.array(x_c).min(),numpy.array(x_c).max(),300)
    curve_smooth = spline(numpy.array(x_c),numpy.array(y_c),xnew)
    plt.plot(xnew, curve_smooth,'r--')'''
    plt.plot(x_c, y_c, 'r--')



def showend():
    plt.show()
```

Αρχείο repeat_xy

```
"""This module implements the Monte Carlo algorithm."""
import datetime
from pointcl import Point
import geom
```

```
import prodiagrafes
from math import fabs

ntriesperdim = 2000


def monteCarlo(xth, y, a):
    """Using Monte Carlo find optimum grade points."""
    Point.setGround(xth, y)
    po = Point(a)
    pomin = Point(a)
    ntries = a*2 * ntriesperdim


    for itry in range(ntries):
        if itry>0 and itry %1000 == 0:
            print("      {}/{}".format(itry, ntries))
        for iover in range(3):  #Make at most 3 tries if ovelapping happens
            po.setRandomErs()
            po.computeGeom()
            if po.embOverlapping == 0.0: break
        if po.emb < pomin.emb:
            pomin, po = po, pomin
    return pomin


def saveCarlo(fn, po):
    "Append the results of a Monte-Carlo run to a file."
    d = datetime.datetime.today()
    with open(fn+".car", "a", encoding="iso-8859-7") as fw:  # open the file for
append
        fw.write("#Monte Carlo: nodes={:<3d}  carlotries={:<5d}
{}\n".format(po.a,
                ntriesperdim, d.isoformat(" ")))
        fw.write("#              Area={:<10.3f}  (geom={:<10.3f}
overlapping={:<10.3f}  klish={:<10.3f})\n".format(po.emb, po.embGeom,
po.embOverlapping, po.embKlish))
        fw.write("{:.2f}\n".format(po.emb))
        fw.write("{:d}\n".format(len(po.xers)))
        r = list(po.h)
        r.insert(0, 0.0)
        r.append(0.0)
        for i in range(len(po.xers)):
            fw.write("{:15.3f}{:15.3f}{:15.3f}\n".format(po.xers[i], po.yers[i],
r[i]))
        fw.write("\n")   #blank line
```

Αρχείο pointcl

```
import random
import prodiagrafes
```

```
import parametroi_kampilis
from geom import plinint, klisi, d_klisi, proarea, elegxos_s
from math import fabs


class Point:
    """Computes random grade points, the elevation of the grade points
     and the area between grade and ground line."""
    ve, s_max, hk, hw = prodiagrafes.vasikes_parametroi()
    embmax = 1.0e100

    @classmethod
    def setGround(cls, xth, y):
        "Set the ground and number of nodes for all objects in this class."
        cls.xth = tuple(xth)
        cls.y = tuple(y)

    def __init__(self, a, xers=None, yers=None):
        "Set the geometry of the profile; compute random nodes if needed."
        self.a = a
        if xers is None:
            self.setRandomErs()
        else:
            assert len(self.xers) == a+2
            self.xers = list(xers)
            self.yers = list(yers)
        self.computeGeom()

    def computeGeom(self):
        "(Re)compute geometry of the profile."
        self.s = klisi(self.xers, self.yers)
        self.ds = d_klisi(self.xers, self.yers)
        self.h = parametroi_kampilis.dtrmn_h(Point.hw, Point.hk, self.s)
        self.t = parametroi_kampilis.dis_t(self.h, self.ds)
        self.embOverlapping = self.penaltyOverlapping()
#Additional area (penalty) due to overlapping
        self.embKlish = fabs(elegxos_s(self.xers, self.yers, Point.s_max))
#Additional area (penalty) due to excess slope
        self.embGeom = proarea(self.edafery())                                  #Area
of profile
        self.emb = self.embGeom + self.embOverlapping + self.embKlish


    def setRandomErs(self):
        "(Re)set random nodes."
        xers = []
        xers.append(self.xth[0])
        xers.append(self.xth[-1])
        for i in range(self.a):    # a is the number of grade points between the
first and last station
            xers.append(random.uniform(xers[0], xers[1]))
        xers.sort()
        self.xers = xers
```

```
        self.yers = Point.random_yers(self.xers, self.xth, self.y)


    @staticmethod
    def random_yers(xers, xth, y):
        """Computes the elevation of the grade points: ground elevation +-30
m."""
        yers = []
        yers.append(y[0])
        for i in range(1, (len(xers)-1)):
            yers.append(plinint(xth, y, xers[i])+random.uniform(-30.0, 30.0))
        yers.append(y[-1])
        return yers

    def penaltyOverlapping(self):
        "Penalty due to overlapping of two consequtive vertical curves."
        penalty = 0.0
        for e in range(1, len(self.t)):
            tt = fabs(self.t[e]) + fabs(self.t[e-1])
            xx = self.xers[e] - self.xers[e-1]
            if tt > xx:
                penalty += (tt-xx)*30.0
        return penalty

    def edafery(self):
        """Computes the grade line for all ground points and the ground line for
all grade points."""
        xeds2 = self.xth
        yeds2 = self.y
        xarx = xeds2[0]
        xtel = xeds2[-1]
        basic_x = parametroi_kampilis.curve(self.xers, self.ds, self.h)   #Start
x and end x of all curves
        cc3 = [(b_x, plinint(xeds2, yeds2, b_x), plinint(self.xers, self.yers,
b_x)) for b_x in basic_x
               if xarx<b_x<xtel]    #If we have overalapping, basic_x may be out
of the profile
        cc1 = [(xer, plinint(xeds2, yeds2, xer),
                parametroi_kampilis.in_out_curve(self.xers, self.yers, xer,
self.ds, self.h))
                for (xer, yer) in zip(self.xers, self.yers)]
        cc2 = [(xed, yed, parametroi_kampilis.in_out_curve(self.xers, self.yers,
xed, self.ds, self.h)) for (xed, yed) in zip(xeds2, yeds2)]
        cc2.extend(cc1)
        cc2.extend(cc3)
        cc2.sort()
        return cc2
```

Αρχείο geom

```
"""This modules defines geometry functions."""
from math import fabs


def plinint(xers, yers, x):
    "Interpolates x in polyline defined by xers, yers."
    if x < xers[0] or x > xers[-1]:raise ValueError("Out of range")
    for i in range(1, len(xers)):
        if x <= xers[i]:
            return linint(xers[i-1], yers[i-1], xers[i], yers[i], x)


def linint(x1, y1, x2, y2, x):
    "Linear interpolation without checking for zero denominator."
    return y1 + (y2-y1)/(x2-x1) * (x-x1)


def d_klisi(xers, yers):
    return [j-i for i, j in zip(klisi(xers, yers)[:-1], klisi(xers, yers)[1:])]


def klisi(xers, yers):
    return [(yers[i]-yers[i-1])/(xers[i]-xers[i-1]) for i in range(1, len(xers))]


def proarea(cc):
    """Computes the positive and negative areas between grade and ground line."""
    def area(x1, dy1, x2, dy2):
        """Computes area between two points and adds it to positive or negative
sums."""
        nonlocal epos, eneg
        e = (dy1+dy2) * (x2-x1) / 2
        if e > 0: epos += e
        else:   eneg += e

    epos = eneg = 0.0
    x = [ca[0] for ca in cc]
    dy = [ca[1]-ca[2] for ca in cc]
    for i in range(len(x)-1):
        j = i + 1
        if dy[i]*dy[j] >= 0:
            area(x[i], dy[i], x[j], dy[j])
        else:
            xm = linint(dy[i], x[i], dy[j], x[j], 0.0)
            area(x[i], dy[i], xm, 0.0)
```

```
            area(xm, 0.0, x[j], dy[j])
    return epos-eneg


def elegxos_s(xers, yers, s_max):
    "Additional penalty (area units) in case some slope exceeds max slope."
    kl = klisi(xers, yers)
    eplus = 0.0
    for i in range(1, len(xers)):
        akl = fabs(kl[i-1])
        if akl > s_max:
            eplus += 30.0*(xers[i]-xers[i-1]) * fabs(akl-s_max)/s_max
    return eplus
```

Αρχείο prodiagrafes

```
def vasikes_parametroi():
    ve_d = [   50,    60,    70,    80,    90,   100,  110,  120,   130]
    hk_d = [-1400, -3000, -4500, -6200, -8000, -10000, -15000, -15000, -15000]
#Crest vertical curve radius OMOE-X teyxos 3 8/6/2001
    hw_d = [ 1350,  1900,  2500,  3300,  4200,  5200, 6300, 7500, 10000] #Sag
vertical curve radius
    global ve, s_max, hk, hw
    ve = 70   # ταχύτητα μελέτης
    kat_odou = "A"
    kat_ed = "Λοφώδες"
    if kat_odou == "A":
        s_max = katigoria_a(ve, ve_d, kat_odou, kat_ed)
    elif kat_odou == "B" :
        s_d = [0.08, 0.07, 0.06, 0.05]
        s_max = s_d[ve_d.index(ve)]
    else:
        raise ValueError("Wrong road type")
    hk = hk_d[ve_d.index(ve)]
    hw = hw_d[ve_d.index(ve)]

    return ve, s_max, hk, hw


def katigoria_a(ve, ve_d, kat_odou, kat_ed):
    """Computes maximum grade."""
    if kat_ed == "Πεδινό":
        s_d = [0.07, 0.06, 0.05, 0.04, 0.04, 0.03, 0.03, 0.03, 0.03]
    elif kat_ed == "Λοφώδες":
        s_d = [0.08, 0.07, 0.06, 0.05, 0.05, 0.04, 0.04, 0.04]
    elif kat_ed == "Ορεινό":
        s_d = [0.10, 0.09, 0.08, 0.07, 0.07, 0.06, 0.05]
    else:
        raise ValueError("Wrong terrain type")
    s = s_d[ve_d.index(ve)]    #This may raise IndexError
    return s
```

Αρχείο parametroi_kampilis

```
import geom


def dtrmn_h(hw, hk, s):  # xers list has two more elements than h
    """Finds a list about radius h between two lines"""
    h = []
    for i in range(1, len(s)):
        if s[i] < s[i-1]: h.append(hk)    #sag
        else: h.append(hw)                #crest
    return h


def dis_t(h, ds):  # t list has as much elements as xers list. The first and the
last is 0
    """Finds a list about distance t in vertical curve"""
    t = list(map(lambda x, y: (x*y)/2, h, ds))
    t.append(0.0)
    t.insert(0, 0.0)
    return t


def curve(xers, ds, h):
    """Estimates the first and the last x coordinate of a vertical curve"""
    basic_x = [xers[i] - dis_t(h, ds)[i] for i in range(1, len(xers)-1)]
    basicx_1 = [xers[i] + dis_t(h, ds)[i] for i in range(1, len(xers)-1)]
    basic_x.extend(basicx_1)
    basic_x.sort()
    return basic_x


def in_out_curve(xers, yers, x, ds, h):
    d_t = dis_t(h, ds)
    # print("parametroi kampilis inout", d_t)
    if x < xers[0] or x > xers[-1]: raise ValueError("Out of range")
    for e in range(1, len(xers)-1):
        if x>=xers[e]-d_t[e] and x<=xers[e]: return alt_curve_x(xers, yers, x,
d_t, h, e)
        elif x>xers[e] and x<=xers[e]+d_t[e]: return alt_curve_x_plus(xers, yers,
x, d_t, h, e)
    else: return geom.plinint(xers, yers, x)


def alt_curve_x(xers, yers, x, d_t, h, e):
    """Estimates the altitude of a point in a vertical curve"""
    alt = geom.linint(xers[e-1], yers[e-1], xers[e], yers[e], x)+((x-(xers[e]-
d_t[e]))**2/(2*h[e-1]))
    return alt


def alt_curve_x_plus(xers, yers, x, d_t, h, e):
```

```
    """Estimates the altitude of a point in a vertical curve"""
    alt = geom.linint(xers[e], yers[e], xers[e+1], yers[e+1], x)+((x-
(xers[e]+d_t[e]))**2/(2*h[e-1]))
    return alt
```

Αρχείο dxf

```
out = None
lay = "0"   #default layer


def plotstart():
    global out
    out = open("data5.dxf", "w")
    for line in "0 SECTION 2 HEADER 9 ACADVER 1 $AC1009 0 ENDSEC 0 SECTION 2
ENTITIES".split():
        out.write(line+"\n")


def plotline(x1, y1, x2, y2):
    plot = "0 LINE 8 {0} 10 {1} 20 {2} 11 {3} 21 {4}".format(lay, x1, y1, x2, y2)
    for line in plot.split():
        out.write(line+"\n")


def plotend():
    for line in "0 ENDSEC 0 EOF".split():
        out.write(line+"\n")
    out.close()


def plotsymbol(x1, y1, h, txt,      theta):
    ps = "0 TEXT 8 {0} 10 {1} 20 {2} 40 {3} 1 {4} 50 {5}".format(lay, x1, y1, h,
txt, theta)
    for line in ps.split():
        out.write(line+"\n")


def plotsetlayer(laynew):
    "Set layer for all following objects."
    global lay
    lay = laynew
```

Multiprocessoring

```
import multiprocessing
import fdata
import repeat_xy
import korerythras
import geom
from pointcl import Point


def monteCarlo(xth, y, a):                                  # comment for linux
    """Using Monte Carlo find optimum grade points."""
    Point.setGround(xth, y)
    ntries = a*2 * repeat_xy.ntriesperdim
    pomin = Point(a)

    nprocesses = multiprocessing.cpu_count()
    ntries1 = ntries//nprocesses
    pool = multiprocessing.Pool(nprocesses)

    argslist = [(a, ntries1)] * nprocesses
    for po1 in pool.imap_unordered(repeat_xy.doSome, argslist):
        if po1.emb < pomin.emb:
            pomin = po1
    return pomin


def pyMain():
    """Start here."""
    fn = "road_name"
    fr = open(fn+".mhk", "r", encoding="iso-8859-7")
    amhk, xth, y, name = fdata.readData(fr)
    #fdata.plotmhk(amhk, xth, y, name)
    pl = fdata.PlotManyMhk()
    na = int((xth[-1] - xth[0])//500)
    ncarlo = 38



    for a in range(1, na+1):

        Point.setGround(xth, y)
        ntries = a*2 * repeat_xy.ntriesperdim
        pomin = Point(a)

        if __name__ == '__main__':                          #  comment for linux
            nprocesses = multiprocessing.cpu_count()
            ntries1 = ntries//nprocesses
            pool = multiprocessing.Pool(nprocesses)

            argslist = [(a, ntries1)] * nprocesses
            for po1 in pool.imap_unordered(repeat_xy.doSome, argslist):
```

```
                if po1.emb < pomin.emb:
                    pomin = po1

            for i in range(1, ncarlo+1):
                print("Monte Carlo: nodes={}/{}   try={}/{}".format(a, na, i,
ncarlo))
                # po = repeat_xy.monteCarlo(xth, y, a)  # uncomment for linux
                po = monteCarlo(xth, y, a)
                repeat_xy.saveCarlo(fn, po)
                pl.plotmhk("MonteCarlo_nodes={}_area={:.1f}".format(po.a,
po.emb), xth, y, name, po.xers, po.yers)
                print("    Area: {:.2f}  (geom={:.2f}  overlapping={:.2f}
klish={:.2f})".format(po.emb, po.embGeom, po.embOverlapping, po.embKlish))
                s = geom.klisi(po.xers, po.yers)
                # fdata.showmhk(amhk, xth, y, name, xers, yers)
                # for z in range(0, len(xers)-1, 2):
                #     x_cur = [a[0] for a in korerythras.edafery(xers, yers, xth,
y)]
                #     y_cur = [a[2] for a in korerythras.edafery(xers, yers, xth,
y)]
                #     fdata.show_curve(x_cur, y_cur)
                for slope in s:
                    print("    Slope : {:.2f}%".format(slope*100))
                print("    main", po.yers, po.xers)
                    # fdata.showend()
    pl.plotend()
    fr.close()



pyMain()
```

Αρχείο repeat_xy

```
"""This module implements the Monte Carlo algorithm."""
import datetime
from pointcl import Point
import geom
import prodiagrafes
from math import fabs
import multiprocessing

ntriesperdim = 2000


# def monteCarlo(xth, y, a):                    # uncomment for linux
#     """Using Monte Carlo find optimum grade points."""
#     Point.setGround(xth, y)
#     ntries = a*2 * ntriesperdim
#     pomin = Point(a)
#
#     if __name__ == '__main__':
```

```
#         nprocesses = multiprocessing.cpu_count()
#         ntries1 = ntries//nprocesses
#         pool = multiprocessing.Pool(nprocesses)
#
#         argslist = [(a, ntries1)] * nprocesses
#     for po1 in pool.imap_unordered(doSome, argslist):
#         if po1.emb < pomin.emb:
#             pomin = po1
#     return pomin


def monteCarlosingle(xth, y, a):
    """Using Monte Carlo find optimum grade points; using a singe cpu/core."""
    Point.setGround(xth, y)
    ntries = a*2 * ntriesperdim


    pomin = doSome((a, ntries))
    return pomin


def doSome(args):
    "Do some of the iterations."
    a, ntries = args
    po = Point(a)
    pomin = Point(a)
    for itry in range(ntries):
        if itry>0 and itry %1000 == 0:
            print("      {}/{}".format(itry, ntries))
        for iover in range(3):  #Make at most 3 tries if ovelapping happens
            po.setRandomErs()
            po.computeGeom()
            if po.embOverlapping == 0.0: break
        if po.emb < pomin.emb:
            pomin, po = po, pomin
    return pomin


def saveCarlo(fn, po):
    "Append the results of a Monte-Carlo run to a file."
    d = datetime.datetime.today()
    with open(fn+".car", "a", encoding="iso-8859-7") as fw:  # open the file for
append
        fw.write("#Monte Carlo: nodes={:<3d}  carlotries={:<5d}
{}\n".format(po.a,
                 ntriesperdim, d.isoformat(" ")))
        fw.write("#              Area={:<10.3f}  (geom={:<10.3f}
overlapping={:<10.3f}  klish={:<10.3f})\n".format(po.emb, po.embGeom,
po.embOverlapping, po.embKlish))
        fw.write("{:.2f}\n".format(po.emb))
        fw.write("{:d}\n".format(len(po.xers)))
        r = list(po.h)
        r.insert(0, 0.0)
```

```
        r.append(0.0)
        for i in range(len(po.xers)):
            fw.write("{:15.3f}{:15.3f}{:15.3f}\n".format(po.xers[i], po.yers[i],
r[i]))
        fw.write("\n")   #blank line
```

Μέθοδος Γενετικών Αλγορίθμων

```
import datetime
import geom
import fdata
from genal import GeneticAnimal, GeneticAlgorithm
from pointcl import Point
import genal


class PointAnimal(GeneticAnimal):

    def crhom2Fenotype(self, ga):
        """Transfer the instructions in chromosomes to the physical nature of the
animal."""
        a = ga.comNchrom
        xtha = Point.xth[0]
        xtht = Point.xth[-1]
        dx = (xtht - xtha)/ga.comNvalmax
        xers = [xtha]
        for i in range(a):
            xers.append(xtha + dx*self.chrom[i])
        xers.append(xtht)
        xers.sort()
        self.po = Point(a, xers)


    def justFitness(self, ga):
        """The fitness of the animal; the bigger the better."""
        return self.po.emb


def prg(fr, tags=()):
    "Print converting to DOS greek only if we run windows (tags is for
compatibility)."
    print(fr)



def pyMain():
    """Start here."""
    fn = "art1"
    fr = open(fn+".mhk", "r", encoding="iso-8859-7")
    amhk, xth, y, name = fdata.readData(fr)
    pl = fdata.PlotManyMhk()
```

```
    Point.setGround(xth, y)

    na = int((xth[-1] - xth[0])//500)
    ncarlo = 3
    ga = GeneticAlgorithm(prt=prg, nValmax=16384, nChrom=1, fitmax=30.0*(xth[-1]-
xth[0]), biggerIsBetter=False)
    for j in range(2, na + 1, 2):
        for a in range(j, j + 1):
            for i in range(1, ncarlo+1):
                print("Genetic: nodes={}/{}   try={}/{}".format(a, na, i,
ncarlo))
                ga = GeneticAlgorithm(prt=prg, nValmax=16384, nChrom=a,
fitmax=30.0*(xth[-1]-xth[0]), biggerIsBetter=False)
                ga.config(pmut = ga.pmut/5.0)  #1 mutation every 5 generations

                ga.genetic(PointAnimal)
                po = ga.bestall.po
                saveGenetic(fn, po)
                pl.plotmhk("Genetic_nodes={}_area={:.1f}".format(po.a, po.emb),
xth, y, name, xth, [c[2] for c in po.edafery()])  #po.xers, po.yers)
                print("    Area: {:.2f}  (geom={:.2f}  overlapping={:.2f}
klish={:.2f})".format(po.emb, po.embGeom, po.embOverlapping, po.embKlish))
                s = geom.klisi(po.xers, po.yers)
                # fdata.showmhk(amhk, xth, y, name, xers, yers)
                # for z in range(0, len(xers)-1, 2):
                #     x_cur = [a[0] for a in korerythras.edafery(xers, yers, xth,
y)]
                #     y_cur = [a[2] for a in korerythras.edafery(xers, yers, xth,
y)]
                #     fdata.show_curve(x_cur, y_cur)
                for slope in s:
                    print("    Slope : {:.2f}%".format(slope*100))
                print("    main", po.yers, po.xers)
                # fdata.showend()
    pl.plotend()
    fr.close()


def saveGenetic(fn, po):
    "Append the results of a Monte-Carlo run to a file."
    d = datetime.datetime.today()
    with open(fn+".car", "a", encoding="iso-8859-7") as fw:  # open the file for
append
        fw.write("#Genetic alg: nodes={:<3d}   {}\n".format(po.a, d.isoformat("
")))
        fw.write("#              Area={:<10.3f}  (geom={:<10.3f}
overlapping={:<10.3f}  klish={:<10.3f})\n".format(po.emb, po.embGeom,
po.embOverlapping, po.embKlish))
        fw.write("{:.2f}\n".format(po.emb))
        fw.write("{:d}\n".format(len(po.xers)))
        r = list(po.h)
        r.insert(0, 0.0)
        r.append(0.0)
```

```
        for i in range(len(po.xers)):
            fw.write("{:15.3f}{:15.3f}{:15.3f}\n".format(po.xers[i], po.yers[i],
r[i]))
        fw.write("\n")   #blank line


pyMain()
```

Αρχείο genal

```
from math import log2, fabs
import random
import copy


def prg(fr, tags=()):
    "Print converting to DOS greek only if we run windows (tags is for
compatibility)."
    print(fr)


class GeneticAlgorithm(object):
    """The class implements the genetic algorithm for optimisation."""

    def __init__(self, *args, **kw):

        # stin arxi eixe **kw egw to ekana *args kai sti seira 45

        """Initialize genetic algorithm procedure.

        These values are typical. They will be overwritten by genetic()."""
        self.prt = prg
        self.nAnim = None       # (initial?) number of animals
        self.animals = set()    # set of all animals alive
        self.bests = []         # Best animal of each generation
        self.bestall = None     # Best animal of all
        self.nGen = 200         # Number of generations
        self.igen = 0           # Current generation
        self.pmut = 0.0         # probability of mutation (if not set by user,
once every generation)

        self.r = random.Random()
        self.nRuns = 1          # Number of runs that the algorithm will be
executed

        self.comNvalmax = 2     # Integer max value (0..value-1) that a
chromosome may have (usually 2)
        self.comNchrom = 26     # Number of (different) chromosomes
        self.comFitmax = 0.0    # A big fitness; in fact it should be the best
fitness (biggest) possible (if known)
                                # It is used to compute deficiency due to common
chromosomes between parents.
```

```
                                # If comBiggerIsBetter is False, comFitamx should
again be a big value (poor):
                                # it is used convert small values (which are
better) to big values (which the GA
                                # tries to maximise).
                                # If zer0, then the biggest fitness of all
animals (which are random) at the beginning is taken
        self.nSample = 10       # Number animals taken is sample in
tournamentThanasis(); the least fit of these animals dies.
        self.comBiggerIsBetter = True    # If true justFitness() returns a number
which must be maximised
                                         # If false justFitness() returns a
number which must be minimised
        self.config(*args, **kw)

    def config(self, prt=None, nAnim=None, nGen=None, pmut=None, nValmax=None,
nChrom=None, fitmax=None, biggerIsBetter=None):
        """Change default values."""
        if prt     is not None: self.prt     = prt
        if nAnim   is not None:
            if nAnim < self.nSample: raise ValueError("At least %d animals are
required (for tournamentThanasis())" % (self.nSample,))
            self.nAnim = nAnim
        if nGen    is not None: self.nGen = nGen
        if pmut    is not None: self.pmut = pmut
        if nValmax is not None: self.comNvalmax = nValmax
        if nChrom  is not None: self.comNchrom = nChrom
        if fitmax  is not None: self.comFitmax= fitmax
        if biggerIsBetter is not None: self.comBiggerIsBetter = biggerIsBetter
        if self.nAnim is None:    # If bo value, compute it
            self.nAnimCompute()
        elif nAnim is None:       # If nAnim is not None, then the user wants
nAnim (which is already set)
            if nValmax is not None or nChrom is not None:  # If nValmax and
nChrom are both None, there is nothing new to compute
                self.nAnimCompute()
        if self.pmut <= 0.0:
            self.pmutCompute()
        elif nAnim is not None:    # If the number of animals changes, the
probability must change
            self.pmutCompute()

    def nAnimCompute(self):
        """Compute the number of animals from other values (and change the value
of mutation probability).

        According to tournamentThanasis() we choose a random set of nSample (10)
animals
        delete the one with the lowest fitness, until 50% of animals are deleted.
        Thus we must have at least 2*nSample (20) animals, so that when we delete
half of them,
        we still have nSample (10) animals to choose a random set."""
        self.nAnim = int(5 * log2(self.comNvalmax) * self.comNchrom)    # 5
```

```
animals per binary chromosome
        self.nAnim = max(self.nAnim, 2*self.nSample)   # At least 2*nSample (20)
animals anyway
        self.pmutCompute()

    def pmutCompute(self):
        """Compute the probability of mutations: once per generation.

        According to tournamentThanasis() and breed(), the whole population
        of animals is replaced in every generation. 50% of the population dies.
        The remaining 50% are going to be parents. However only the parents get
        mutations. Thus nAnim/2 animals may mutate in each generation."""
        self.pmut = 1.0 / (self.nAnim/2.0)

    def fitmaxCompute(self, DisAnimal):
        """Compute a big fitness; in fact it should be the best fitness possible.

        At the beginninbgt the animals are random. Thus the biggest fitness of
all animals is
        a reasonable big fitness.
        """
        self.comFitmax = max(anim.justFitness(self) for anim in self.animals)

    def genetic(self, DisAnimal):
        "Execute the genetic algorithm."
        self.animals.clear()
        del self.bests[:]
        for i in range(self.nAnim):
            self.animals.add(DisAnimal(self))
        if self.comFitmax == 0: self.fitmaxCompute(DisAnimal)
        for self.bestall in self.animals: break     # Give a initial value to
bestall
        for self.igen in range(self.nGen):
            # print "Generation%4d" % igen
            self.bestIngen(self.igen)
            if self.converged(): break
            self.tournamentthanasis()
            self.breed()
        else:
            self.bestIngen(self.nGen)
            self.prt("Genetic algorithm terminated due to max generations",
"can1")
        # self.prt("\nBest fitness for every generation", "info1")
        # for i,b in enumerate(self.bests):
        #    self.prt("%4d: %.2f(%.2f)" % (i, b.fitness(self), b.fitnessr(self)))
        self.prt("")
        self.prt("best of all:", "info")
        if self.comBiggerIsBetter:
            self.prt("%.2f(%.2f)" % (self.bestall.fitness(self),
self.bestall.fitnessr(self)))
        else:
            self.prt("%.2f(%.2f)" % (self.comFitmax-self.bestall.fitness(self),
self.comFitmax-self.bestall.fitnessr(self)))
```

```
    def converged(self):
        "Check if the genetic algoithm converged; 20 steps with no optimisation."
        if len(self.bests) < 100: return False    # At least 100 generations
        nc = 30
        if len(self.bests) < nc: return False
        flast = self.bests[-1].fitnessr(self)
        for anim in self.bests[-nc:]:
            f = anim.fitnessr(self)
            if thanNearx(f, flast): continue # flast is equal to f
            if flast > f: return False       # flast is a better value than f;
not converged
        return True

    def bestIngen(self, igen):
        """Find and save the best animal in generation igen."""
#        print "----------------------------------------------------------------
---"
        best = max(self.animals, key=lambda a:a.fitnessr(self))
        if self.comBiggerIsBetter:
            self.prt("generation%4d  best: %.2f(%.2f)" % (igen,
best.fitness(self), best.fitnessr(self)))
        else:
            self.prt("generation%4d  best: %.2f(%.2f)" % (igen, self.comFitmax-
best.fitness(self),  self.comFitmax-best.fitnessr(self)))
        self.bests.append(best)

    def breed(self):
        """Animals mate and give birth to children."""
        anims = set()
        while len(self.animals) > 1:
            a, b = self.r.sample(self.animals, 2)
            self.animals.remove(a)
            self.animals.remove(b)
            a.mutate(self)
            b.mutate(self)
            for i in range(4):
                ch = a.mate(self, b)
                anims.add(ch)
        anims.update(self.animals)
        self.animals = anims

    def tournamentclassic(self):
        "Select the best part of the population to breed; allow for a bit random
selection."
        pdie = 0.30
        nkeep = int(self.nAnim*(1-pdie))
        keep = set()
        for i in range(nkeep):
            pop = self.r.sample(self.animals, 4)
            a = max(pop, key=lambda a:a.fitness(self))
            keep.add(a)
            self.animals.remove(a)
```

```
            f = a.fitness(self)
#            print "%f -> %f -> %.1f  ακόμα ζωντανός" % (f, com.fitmax,
f/fa*100.0)
        for a in self.animals:
            f = a.fitness(self)
#            print "%f -> %f -> %.1f  πεθαίνει" % (f, com.fitmax, f/fa*100.0)

    def tournamentthanasis(self):
        """Select the best part of the population to breed; allow for a bit
random selection."""
        pdie = 0.50
        ndie = int(self.nAnim*pdie)
        for i in range(ndie):
            pop = self.r.sample(self.animals, self.nSample)
            a = min(pop, key=lambda a:a.fitness(self))
            self.animals.remove(a)
#            print "%.2f(%.2f) <= %.2f  πεθαίνει" % (a.fitness(), a.fitnessr(),
com.fitmax)
#        for a in self.animals:
#            print("%.2f(%.2f) <= %.2f  ζει" % (a.fitness(), a.fitnessr(),
com.fitmax))


class GeneticAnimal(object):
    """An animal representing the problem to optimise."""
    comR = random.Random()

    def __init__(self, ga, chrom1=None, deficiency=0.0):
        """Create chromosomes for the distribution animal."""
        if chrom1 == None:
            self.chrom = [0]*ga.comNchrom
            for i in range(ga.comNchrom): self.chrom[i] =
self.comR.randrange(ga.comNvalmax)
        else:
            assert len(chrom1) == ga.comNchrom
            self.chrom = list(chrom1)
        self.chrom.sort()      #2018_09_30
        self.crhom2Fenotype(ga)
        self.defic = deficiency                  # Deficiency due to degeneration
        self.ftns = None

    def crhom2Fenotype(self, ga):
        """Transfer the instructions in chromosomes to the physical nature of the
animal."""
        pass

    def mutate(self, ga):
        """Perform a random mutation."""
        r = self.comR
        if r.random() > ga.pmut: return     #Mutate 1 in 1000 (look com.py)
        old = self.fitness(ga)
        oldr = self.fitnessr(ga)
        n = len(self.chrom)
```

```
        if r.random() < -1.0:   #2018_09_30
            i1 = r.randrange(n)
            monkey1 = self.chrom[i1]
            for i in range(10):
                i2 = r.randrange(n)
                monkey2 = self.chrom[i2]
                if monkey1 != monkey2: break
            else:
                ga.prt("Warning: can't find 2 crhomosomes with different values",
"can1")
                return
            self.chrom[i1] = monkey2
            self.chrom[i2] = monkey1
        else:
            i1 = r.randrange(n)
            monkey1 = self.chrom[i1]
            for i in range(10):
                monkey2 = r.randrange(ga.comNvalmax)
                if monkey1 != monkey2: break
            else:
                ga.prt("Warning: can't find different value to assign to
chromosome", "can1")
                return
            self.chrom[i1] = monkey2
        self.chrom.sort()      #2018_09_30
        self.crhom2Fenotype(ga)
        self.ftns = None
        self.defic = 0.0
        if ga.comBiggerIsBetter:
            ga.prt("           mutation: %.2f(%.2f) -> %.2f(%.2f)" % (old, oldr,
self.fitness(ga), self.fitnessr(ga)), "info1")
        else:
            ga.prt("           mutation: %.2f(%.2f) -> %.2f(%.2f)" %
(ga.comFitmax-old, ga.comFitmax-oldr, ga.comFitmax-self.fitness(ga),
ga.comFitmax-self.fitnessr(self)), "info1")

    def mate2(self, other):
        """Make 2 children which replace the parents, with change of a sequence
of chromosomes."""
        r = self.comR
        if r.random() < 0.7: return   # 30% of the parents do not mate
        n = len(self.ban)
        n2 = int(n//2)
        i1 = r.randrange(n2)
        i2 = r.randrange(n2, n+1)
        self.chrom[i1:i2], other.chrom[i1:i2] = other.chrom[i1:i2],
self.chrom[i1:i2]
        self.ftns = None
        other.ftns = None

    def mate(self, ga, other):
        """Make 1 children with mixed chromosomes of the parents."""
        r = self.comR
```

```
        n = len(self.chrom)
        n2 = int(n//2)
        i1 = r.randrange(n2)
        i2 = r.randrange(n2, n+1)
        m = self.chrom[:i1]+other.chrom[i1:i2]+self.chrom[i2:]
        d = 0
        for i in range(i1,i2):
            if self.chrom[i] == other.chrom[i]: d += 1
        d = float(d)/(i2-i1)
        if d > 0.5: d = ga.comFitmax*0.1*d
        else:       d = 0.0
        return self.__class__(ga, m, d)

    def fitness(self, ga):
        """The fitness corrected according to big or small is better, and
corrected with deficiency."""
        if self.ftns is None:
            self.ftns = self.justFitness(ga)   # Either bigger or smaller is
better
            if not ga.comBiggerIsBetter: self.ftns = ga.comFitmax - self.ftns
            rep = self.ftns > ga.bestall.fitnessr(ga)
            self.ftns -= self.defic
            if rep: ga.bestall = copy.copy(self)
        return self.ftns

    def justFitness(self, ga):
        """The fitness of the animal; the bigger the better."""
        raise AttributeError("Method justFitness() should be overriden.")

    def fitnessr(self, ga):
        """The real fitness (i.e. plus deficiency)."""
        return self.fitness(ga)+self.defic

    def detail(self, ga, fw):
        """Details about the distribution represented by the animal."""
        pass


thanThresholdx = 1e-10  # Threshold for coordinate difference; if less the
coordinates are the same
                        # This constant acommodates ThanCad too.


def thanNearx(xa, xb):
    "Checks if two coordinates (x or y) coincide taking numerical error into
account."
    d = fabs(xb-xa)
    v = (fabs(xa)+fabs(xb))*0.5
    if v < thanThresholdx: return d < thanThresholdx
    return d < v*thanThresholdx
```

Αρχείο pointcl

```
from math import fabs
import random
import prodiagrafes
import parametroi_kampilis
from geom import plinint, klisi, d_klisi, proarea, elegxos_s
import ery


class Point:
    """Computes random grade points, the elevation of the grade points
     and the area between grade and ground line."""
    ve, s_max, hk, hw = prodiagrafes.vasikes_parametroi()
    embmax = 1.0e100

    @classmethod
    def setGround(cls, xth, y):
        "Set the ground and number of nodes for all objects in this class."
        cls.xth = tuple(xth)
        cls.y = tuple(y)

    def __init__(self, a, xers=None, yers=None):
        "Set the geometry of the profile; compute random nodes if needed."
        self.a = a
        if xers is None:
            self.setRandomErs()
        else:
            assert len(xers) == a+2
            self.xers = list(xers)
            self.analyticYers()
        self.computeGeom()


    def computeGeom(self):
        "(Re)compute geometry of the profile."
        self.s = klisi(self.xers, self.yers)
        self.ds = d_klisi(self.xers, self.yers)
        self.h = parametroi_kampilis.dtrmn_h(Point.hw, Point.hk, self.s)
        self.t = parametroi_kampilis.dis_t(self.h, self.ds)
        self.embOverlapping = self.penaltyOverlapping()
#Additional area (penalty) due to overlapping
        self.embKlish = fabs(elegxos_s(self.xers, self.yers, Point.s_max))
#Additional area (penalty) due to excess slope
        self.embGeom = proarea(self.edafery())                                  #Area
of profile
        self.emb = self.embGeom + self.embOverlapping + self.embKlish
```

```
    def setRandomErs(self):
        "(Re)set random nodes."
        xers = []
        xers.append(self.xth[0])
        xers.append(self.xth[-1])
        for i in range(self.a):    # a is the number of grade points between the
first and last station
            xers.append(random.uniform(xers[0], xers[1]))
        xers.sort()
        self.xers = xers
        self.analyticYers()


    def analyticYers(self):
        "Compute yers analytically."
        import numpy as np
        xth = np.array(self.xth)
        a, b, k = ery.exis(xth, np.array(self.y), self.xers[1:-1])
        self.yers = ery.ypolNodes(self.xers[1:-1], a, b)
        self.yers.insert(0, self.y[0])
        self.yers.append(self.y[-1])


    def penaltyOverlapping(self):
        "Penalty due to overlapping of two consequtive vertical curves."
        penalty = 0.0
        for e in range(1, len(self.t)):
            tt = fabs(self.t[e]) + fabs(self.t[e-1])
            xx = self.xers[e] - self.xers[e-1]
            if tt > xx:
                penalty += (tt-xx)*30.0
        return penalty


    def edafery(self):
        """Computes the grade line for all ground points and the ground line for
all grade points."""
        xeds2 = self.xth
        yeds2 = self.y
        xarx = xeds2[0]
        xtel = xeds2[-1]
        basic_x = parametroi_kampilis.curve(self.xers, self.ds, self.h)   #Start
x and end x of all curves
        cc3 = [(b_x, plinint(xeds2, yeds2, b_x), plinint(self.xers, self.yers,
b_x)) for b_x in basic_x
               if xarx<b_x<xtel]    #If we have overalapping, basic_x may be out
of the profile
        cc1 = [(xer, plinint(xeds2, yeds2, xer),
                parametroi_kampilis.in_out_curve(self.xers, self.yers, xer,
self.ds, self.h))
                for (xer, yer) in zip(self.xers, self.yers)]
        cc2 = [(xed, yed, parametroi_kampilis.in_out_curve(self.xers, self.yers,
```

```
xed, self.ds, self.h)) for (xed, yed) in zip(xeds2, yeds2)]
        cc2.extend(cc1)
        cc2.extend(cc3)
        cc2.sort()
        return cc2
```

Αρχείο ery

```
import bisect
import numpy as np
from numpy.linalg import lstsq, LinAlgError
from matplotlib import pyplot as pl

def ery(filnam):
    v = np.loadtxt(filnam)
    xth = v[:, 0]
    yed = v[:, 1]
    del v

    nu = 10
    dx = xth[-1]-10.0 - (xth[0] + 10.0)
    u = np.random.rand(nu) * dx + (xth[0] + 10.0)
    u.sort()
    a, b, k = exis(xth, yed, u)
    yer = ypoler(xth, a, b, k)
    pl.figure()
    pl.plot(xth, yed)
    #pl.hold(True)
    pl.plot(xth, yer, 'r')
    pl.savefig('q1.jpg')


def ypolNodes(u, a, b):
    "Compute the y of the fit line at the ndoes."
    yu = []
    for j in range(len(u)):
        yu.append(a[j] * u[j] + b[j])
    return yu



def ypoler(xth, a, b, k):
    "Compute grade line as the best line to fit the points."
    NN = len(xth)
    n = len(a)
    yer = np.zeros((NN,))
    jp = 0
    for j in range(n):
        if j < n-1: je = k[j]
        else:       je = NN;
        yer[jp:je] = a[j]*xth[jp:je] + b[j];
        jp = je
```

```
    return yer


def exis(xth, yed, u):
    "find the equations of lines that best fit the points."
    n = len(u)
    NN = len(xth)
    assert n+2 <= NN, 'More unknowns than equations'
    k = np.zeros((n,), np.int)
    for j in range(n):
        k[j] = bisect.bisect(xth, u[j]) #this is 1 position to the right of the
value (xth)
    A = np.zeros((NN, n+2))
    A[:,      0] = 1.0;
    A[0:k[0], 1] = xth[0:k[0]]
    A[k[0]:,  1] = u[0]
    for j in range(3-1, n+1):
        A[k[j-2]:k[j-1], j] = xth[k[j-2]:k[j-1]] - u[j-2]
        A[k[j-1]:,       j] = u[j-1] - u[j-2]
    A[k[n-1]:, n+2-1] = xth[k[n-1]:] - u[n-1]
    x, terr = lsmsolve(A, yed)
    if x is None: raise ValueError(terr)
    a = x[1:]
    b = np.zeros((n+1,))
    b[0] = x[0]
    for j in range(n):
        b[j+1] = a[j]*u[j]+b[j]-a[j+1]*u[j]
    return a, b, k


def lsmsolve(A, B):
    "Solve the Least square method problem defined by matrixes A and B."
    try:
        x, residual, rank, s = lstsq(A, B, rcond=-1.0)
    except LinAlgError as why:
        print(why)
        return None, why
    try: return x[:, 0], ""  #2016_01_12thanasis:This converts shape from (333,
1) to (333,)
                             #Otherwise x[1] is an array, not a scalar
    except: return x, ""  #Thanasis2016_04_16: Changed again to normal???


if __name__ == "__main__":
    ery("art1.mhk")
```