



Πολυτεχνείο Κρήτης
Τμήμα Μηχανικών Παραγωγής και Διοίκησης
Τομέας Επιχειρησιακής Έρευνας

Ανάπτυξη αλγορίθμου εκτίμησης χρόνου διαδρομής σε αστικά οδικά δίκτυα με τη χρήση κυκλοφοριακών μετρήσεων

Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων για την απόκτηση
Μεταπτυχιακού Διπλώματος Ειδίκευσης

Υπό

Κωνσταντίνου Κόντου

Χανιά, Μάιος 2008

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου, κ. Ηλία Κοσματόπουλο, επίκουρο καθηγητή στο τμήμα Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης, για την βοήθεια, την υποστήριξη και την εμπιστοσύνη που μου έδειξε σε όλα τα βήματα της παρούσης εργασίας. Ακόμα, ευχαριστώ τον κ. Μάρκο Παπαγεωργίου, καθηγητή, και τον κ. Ιωάννη Παπαμιχαήλ, λέκτορα, στο τμήμα Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης, για το χρόνο που αφιέρωσαν για την ανάγνωση της παρούσης εργασίας καθώς και για τις διορθώσεις που πρότειναν για την βελτίωση της.

Ακόμη θα ήθελα να ευχαριστήσω την Σάντρα για την υποστήριξη που μου έδωσε όλον αυτόν τον καιρό που γνωριζόμαστε. Ευχαριστώ τους φίλους μου Μιχάλη, Νικόλα, Γιώργο, Δημήτρη, Ηλία, Λουκά, Νίκο, Γρηγόρη και Ιπποκράτη για τα φοιτητικά χρόνια που περάσαμε μαζί.

Νιώθω την ανάγκη να ευχαριστήσω και γραπτώς την οικογένεια μου, που με στήριξε οικονομικά και που με εμπιστεύτηκε όλα αυτά τα χρόνια των σπουδών μου στα Χανιά αλλά και κατά τη διάρκεια της στρατιωτικής μου θητείας.

Σας ευχαριστώ,

Κωνσταντίνος Κόντος

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΥΧΑΡΙΣΤΙΕΣ	ii
ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ	v
ΛΙΣΤΑ ΠΙΝΑΚΩΝ	vii

ΚΕΦΑΛΑΙΟ

1 ΕΙΣΑΓΩΓΗ	8
2 ΘΕΩΡΙΑ	11
2.1 Εισαγωγή	11
2.2 Μοντελοποίηση συστημάτων κυκλοφοριακής ροής – Μικροσκοπικά μοντέλα προσομοίωσης	11
2.2.1 Car Following Models	13
2.2.2 Μοντέλα αλλαγής λωρίδας της κυκλοφορίας (lane-changing models)..	16
2.3 Φωρατές επαγωγικού βρόχου (Loop Detectors)	16
2.4 Θεωρία Προσέγγισης (Approximation Theory) και Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks)	17
3 ΜΕΘΟΔΟΛΟΓΙΑ	21
3.1 Εισαγωγή	21
3.2 Μικροσκοπικό Μοντέλο Προσομοίωσης.....	21
3.2.1 Περιγραφή του μικροσκοπικού μοντέλου	21
3.2.2 Περιπτώσεις πράσινου/κόκκινου.....	24
3.3 Προσεγγιστικός Αλγόριθμος	32
3.3.1 Κατασκευή προσεγγιστικής συνάρτησης.....	32
3.3.2 Κατασκευή του πίνακα δεδομένων	34
3.3.3 Εύρεση Σφάλματος.....	37

4	ΑΠΟΤΕΛΕΣΜΑΤΑ	42
4.1	Εισαγωγή	42
4.2	Συγκριτικά αποτελέσματα για τα σφάλματα	42
4.3	Εκτιμώμενος χρόνος.....	45
4.3.1	Περίπτωση Α	46
4.3.2	Περίπτωση Β	47
4.3.3	Περίπτωση Γ	49
4.3.4	Περίπτωση Δ	51
4.3.5	Σύγκριση των τεσσάρων περιπτώσεων	53
4.4	Βελτιωμένη έκδοση προσεγγιστικού αλγορίθμου.....	54
5	ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	59
5.1	Συμπεράσματα.....	59
5.2	Μελλοντικές επεκτάσεις.....	60
	ΠΑΡΑΡΤΗΜΑ Α	61
	ΠΑΡΑΡΤΗΜΑ Β	97
	ΒΙΒΛΙΟΓΡΑΦΙΑ	98

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

Σχήμα 1. Car following model (Rothery R. W., 2002)	14
Σχήμα 2. Οι θέσεις των φωτεινών σηματοδοτών και των φωρατών.....	22
Σχήμα 3. Η θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για N=5000 επαναλήψεις της περίπτωσης Α.....	27
Σχήμα 4. Η θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για N=5000 επαναλήψεις της περίπτωσης Β.....	28
Σχήμα 5. Η θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για N=5000 επαναλήψεις της περίπτωσης Γ	30
Σχήμα 6. θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για N=5000 επαναλήψεις της περίπτωσης Δ.....	31
Σχήμα 7. Σχηματική απεικόνιση του συστήματος μας, των εισερχόμενων και εξερχόμενων μεταβλητών.....	33
Σχήμα 8. Η μορφή του κάθε τμήματος του πίνακα Α για κάθε περίπτωση και κάθε τιμή της μεταβλητής alpha	37
Σχήμα 9. Σφάλματα για t=50 προηγούμενες χρονικές στιγμές.....	43
Σχήμα 10. Γραφικές παραστάσεις των τριών σφαλμάτων για t=100 προηγούμενες χρονικές στιγμές	43
Σχήμα 11. Γραφικές παραστάσεις των τριών σφαλμάτων για t=300 προηγούμενες χρονικές στιγμές	44
Σχήμα 12. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης.....	45
Σχήμα 13. Γραφικές παραστάσεις των σφαλμάτων για την πρώτη μόνο περίπτωση	46
Σχήμα 14. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης.....	47
Σχήμα 15. Γραφικές παραστάσεις των σφαλμάτων για τη δεύτερη μόνο περίπτωση ...	48
Σχήμα 16. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης.....	49
Σχήμα 17. Γραφικές παραστάσεις των σφαλμάτων για την τρίτη μόνο περίπτωση	50
Σχήμα 18. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης.....	51
Σχήμα 19. Γραφικές παραστάσεις των σφαλμάτων για την τέταρτη μόνο περίπτωση ..	52

Σχήμα 20. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης.....	53
Σχήμα 21. Γραφικές παραστάσεις των σφαλμάτων για occupancy sum ίσο και μικρότερο του 60	55
Σχήμα 22. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης.....	56
Σχήμα 23. Γραφικές παραστάσεις των σφαλμάτων για occupancy sum ίσο και μικρότερο του 60	57
Σχήμα 24. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης.....	58
Σχήμα 25. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης για τις t=50 προηγούμενες χρονικές στιγμές	97
Σχήμα 26. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης για τις t=100 προηγούμενες χρονικές στιγμές	97

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

Πίνακας 1. Συνδυασμοί χρόνων πράσινου/κόκκινου για τους τρεις φωτεινούς σηματοδότες	25
Πίνακας 2. Συνδυασμοί χρόνων πράσινου/κόκκινου για την περίπτωση Α	26
Πίνακας 3. Συνδυασμοί χρόνων πράσινου/κόκκινου για την περίπτωση Β.....	27
Πίνακας 4. Συνδυασμοί χρόνων πράσινου/κόκκινου για την περίπτωση Γ	29
Πίνακας 5. Συνδυασμοί χρόνου πράσινου/κόκκινου για την περίπτωση Δ	30
Πίνακας 6. Συνδυασμοί βαρών της συνάρτησης \hat{y} και αντίστοιχα σφάλματα	40
Πίνακας 7. Οι τιμές που λάμβαναν τα βάρη όταν κάθε φορά αφαιρείτο κάποιο από την εκτιμήτρια συνάρτηση	40
Πίνακας 8. Συγκριτικός πίνακας των σφαλμάτων για τα τρία διαφορετικά σενάρια	44
Πίνακας 9. Συγκριτικός πίνακας σφαλμάτων για τις τέσσερις περιπτώσεις	53
Πίνακας 10. Οι τιμές που λάμβαναν τα βάρη όταν κάθε φορά αφαιρείτο κάποιο από την εκτιμήτρια συνάρτηση	55
Πίνακας 11. Οι τιμές που λάμβαναν τα βάρη όταν κάθε φορά αφαιρείτο κάποιο από την εκτιμήτρια συνάρτηση	57
Πίνακας 12. Συγκριτικός πίνακας σφαλμάτων, όπως προέκυψαν με τον βελτιστοποιημένο προσεγγιστικό αλγόριθμο και χωρίς αυτόν	58

1 ΕΙΣΑΓΩΓΗ

Σκοπός της παρούσης εργασίας είναι η εκτίμηση του χρόνου διαδρομής (travel time) σε έναν υποθετικό αστικό δρόμο μιας λωρίδας κυκλοφορίας, όπου είναι τοποθετημένοι τρεις φωτεινοί σηματοδότες, με τη βοήθεια ενός προσεγγιστικού αλγορίθμου που βασίζεται στα Τεχνητά Νευρωνικά Δίκτυα.

Η εκτίμηση των χρόνων διαδρομής είναι πολύ σημαντικό θέμα κατά τον σχεδιασμό της κυκλοφοριακής ροής στα οδικά δίκτυα. Μιας και είναι πιο δύσκολο να γίνουν αλλαγές στις οδικές υποδομές, γίνεται προσπάθεια κατανόησης της υπάρχουσας κατάστασης και αποτελεσματικότερης χρήσης των υπάρχοντων δρόμων (Misca M. et al., 2005; Bellemans T. et al., 2002). Αυτό που επιθυμούμε είναι η γρήγορη μετακίνηση των οχημάτων και ταυτόχρονα η αποφυγή της κυκλοφοριακής συμφόρησης, η οποία συνεπάγεται ταλαιπωρία, χάσιμο χρόνου και μόλυνση του περιβάλλοντος από τους ρύπους των οχημάτων. Σε έναν ηλεκτρονικό υπολογιστή, ο κακός σχεδιασμός της κυκλοφοριακής ροής μπορεί να αποφευχθεί εύκολα, γρήγορα και με χαμηλό κόστος. Μερικά παραδείγματα συστημάτων προσομοίωσης σε ηλεκτρονικό υπολογιστή είναι το MiOS και το CORSIM.

Στο πολυτεχνείο TUD στο Delft της Ολλανδίας έχει αναπτυχθεί το MiOS, ένα υπολογιστικό σύστημα που προβλέπει με επιτυχία την κυκλοφοριακή ροή σε ένα δρόμο για χρονικό ορίζοντα 30 λεπτών (Misca M. et al., 2005). Ένα άλλο σύστημα προσομοίωσης για μικροσκοπικά μοντέλα είναι το CORSIM (CORridor SIMulation). Το CORSIM αναπτύχθηκε τη δεκαετία του '70 και έκτοτε έχουν γίνει διάφορες προσθήκες. Μέχρι και σήμερα αποτελεί ένα από τα πιο διαδεδομένα συστήματα προσομοίωσης της κυκλοφοριακής ροής στις Η.Π.Α. αλλά και στον υπόλοιπο κόσμο (Owen L. et al., 2000).

Στην παρούσα εργασία, αρχικά κατασκευάστηκε ένα μικροσκοπικό μοντέλο προσομοίωσης της κυκλοφοριακής ροής σε ένα δρόμο μονής κυκλοφορίας, στον οποίο βρίσκονται εγκαταστημένοι τρεις φωτεινοί σηματοδότες. Ο δρόμος αυτός είναι εικονικός και κατασκευάστηκε για την εκπόνηση της παρούσης εργασίας. Το μοντέλο

που χρησιμοποιήθηκε είναι ένα απλό μοντέλο, στο οποίο υπάρχουν περιορισμοί. Το κάθε όχημα δε θα πρέπει να επικαλύπτει το άλλο. Σε περίπτωση που το όχημα βρεθεί πριν από κόκκινο φωτεινό σηματοδότη, τότε επιβραδύνει μέχρι να σταματήσει εντελώς, σε απόσταση ασφαλείας από το προπορευόμενο όχημα. Από το μικροσκοπικό αυτό μοντέλο και πραγματοποιώντας δοκιμές στους χρόνους που οι φωτεινοί σηματοδότες επιτρέπουν την διέλευση των οχημάτων, εξάγονται τα δεδομένα της κυκλοφοριακής ροής.

Έπειτα, κατασκευάστηκε ο προσεγγιστικός αλγόριθμος, βασιζόμενος στη θεωρία των Τεχνητών Νευρωνικών Δικτύων, και με την βοήθεια των δεδομένων κυκλοφοριακής ροής που εξάχθηκαν από το μικροσκοπικό μοντέλο προσομοίωσης εξετάστηκε η πρόβλεψη των χρόνων ταξιδιού των οχημάτων. Εξετάστηκε η καλή πρόβλεψη του αλγορίθμου για διαφορετικά σενάρια των χρόνων πράσινου/κόκκινου των φωτεινών σηματοδοτών και επιλέχθηκε εκείνο για το οποίο ο προσεγγιστικός αλγόριθμος εμφάνισε την καλύτερη πρόβλεψη για να βελτιωθεί περαιτέρω.

Η εργασία αναπτύχθηκε σε τέσσερα βασικά μέρη. Στη θεωρία που χρησιμοποιήθηκε, στη μεθοδολογία που ακολουθήθηκε, στην παρουσίαση των αποτελεσμάτων που προέκυψαν και στα συμπεράσματα/μελλοντικές επεκτάσεις που προτάθηκαν.

Η θεωρία που χρησιμοποιήθηκε στην παρούσα εργασία παρουσιάζεται στο 2^ο Κεφάλαιο. Δίνονται βασικοί ορισμοί για το τι ονομάζουμε προσομοίωση, σύστημα και μοντέλο. Εξετάζονται τα είδη των μοντέλων προσομοίωσης και ειδικότερα τα μικροσκοπικά μοντέλα μιας και το μοντέλο προσομοίωσης που κατασκευάστηκε είναι μικροσκοπικό. Γίνεται μια αναφορά στη θεωρία προσέγγισης και στα Τεχνητά Νευρωνικά Δίκτυα, στα οποία βασίστηκε η κατασκευή του προσεγγιστικού αλγορίθμου.

Η μεθοδολογία που ακολουθήθηκε αναλύεται στο 3^ο Κεφάλαιο. Περιγράφεται το μικροσκοπικό μοντέλο προσομοίωσης και ο προσεγγιστικός αλγόριθμος που κατασκευάστηκαν. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε ήταν η MATLAB.

Τα αποτελέσματα που προέκυψαν, σχετικά με την εκτίμηση του χρόνου διαδρομής των οχημάτων σε τρία τμήματα του υπό μελέτη δρόμου, παρουσιάζονται στο 4^ο Κεφάλαιο. Πιο συγκεκριμένα, το πρώτο τμήμα ξεκινά από την αρχή του δρόμου έως και το πρώτο φωτεινό σηματοδότη, το δεύτερο ξεκινά από το δεύτερο έως και τον τρίτο φωτεινό σηματοδότη και τέλος το τρίτο τμήμα ξεκινά από τον τρίτο έως και τον τέταρτο φωτεινό σηματοδότη. Παρουσιάζονται τα τρία σφάλματα Μέσο Τετραγωνικό Σφάλμα (Mean Square Error – M.S.E.), Μέσο Τετραγωνικό Σφάλμα 2 (Mean Square Error 2 – M.S.E. 2) και Μέσο Τετραγωνικό Σφάλμα 3 (Mean Square Error 3 –M.S.E. 3) που προέκυψαν για τρία διαφορετικά σενάρια t προηγούμενων χρονικών στιγμών. Το πρώτο σενάριο είναι για $t=50$, το δεύτερο για $t=100$ και το τρίτο για $t=300$. Τα σφάλματα συγκρίνονται μεταξύ τους και επιλέγεται το σενάριο όπου εμφανίζεται το μικρότερο σφάλμα. Στη συνέχεια, παρουσιάζονται οι γραφικές παραστάσεις των εκτιμώμενων χρόνων \hat{y} και των πραγματικών χρόνων διαδρομής y για το σενάριο που επιλέχθηκε πρωτύτερα. Τα δεδομένα χωρίζονται σε τέσσερις περιπτώσεις χρόνου πράσινου/κόκκινου, που παρουσιάζονται στο δεύτερο κεφάλαιο, δίνονται οι γραφικές παραστάσεις των εκτιμώμενων χρόνων \hat{y} και των πραγματικών χρόνων διαδρομής y και επιλέγεται η περίπτωση για την οποία ο προσεγγιστικός αλγόριθμος που κατασκευάστηκε δίνει τα καλύτερα αποτελέσματα. Για την περίπτωση του προσεγγιστικού αλγορίθμου που δίνει τα καλύτερα αποτελέσματα, κατασκευάστηκε ένας βελτιωμένος προσεγγιστικός αλγόριθμος ο οποίος κάνει προβλέψεις με ακόμη μεγαλύτερη ακρίβεια.

Στο 5^ο Κεφάλαιο παρουσιάζονται τα **συμπεράσματα** που προέκυψαν από την παρούσα εργασία και προτείνονται **μελλοντικές επεκτάσεις** της.

2 ΘΕΩΡΙΑ

2.1 Εισαγωγή

Στο παρόν κεφάλαιο εξετάζεται η θεωρία που χρησιμοποιήθηκε στην παρούσα εργασία. Αρχικά, δίνονται βασικοί ορισμοί για το τι ονομάζουμε προσομοίωση, σύστημα και μοντέλο. Έπειτα, εξετάζονται τα είδη των μοντέλων προσομοίωσης και ειδικότερα τα μικροσκοπικά μοντέλα μιας και το μοντέλο προσομοίωσης που κατασκευάστηκε είναι μικροσκοπικό. Στο τελευταίο τμήμα του κεφαλαίου γίνεται μια αναφορά στη θεωρία προσέγγισης και στα Τεχνητά Νευρωνικά Δίκτυα, στα οποία βασίστηκε η κατασκευή του προσεγγιστικού αλγορίθμου.

2.2 Μοντελοποίηση συστημάτων κυκλοφοριακής ροής – Μικροσκοπικά μοντέλα προσομοίωσης

Για να είναι δυνατή η εξέταση σε βάθος της κυκλοφοριακής ροής ενός οδικού δικτύου, κατασκευάζεται πρώτα ένα μοντέλο του στον υπολογιστή. Με την προσομοίωση της λειτουργίας του μοντέλου, είμαστε σε θέση να το βελτιώνουμε αλλάζοντας το εύκολα, γρήγορα και χωρίς μεγάλο κόστος. Ακολουθώς, δίνονται οι ορισμοί της προσομοίωσης, του συστήματος και του μοντέλου.

«Προσομοίωση είναι η μίμηση της λειτουργίας συστημάτων ή της εξέλιξης διαδικασιών μέσα στο χρόνο με τη βοήθεια υπολογιστή. Σύστημα ονομάζεται ένα σύνολο στοιχείων τα οποία εξελίσσονται και αλληλεπιδρούν σύμφωνα με κάποιους κανόνες. Οι κανόνες αυτοί εκφράζονται με μαθηματικές ή λογικές σχέσεις, και αποτελούν το μοντέλο του συστήματος» (Κουικόγλου, 2002).

Υπάρχει μεγάλη ποικιλία μοντέλων κυκλοφοριακής ροής τα οποία μπορούν να κατηγοριοποιηθούν ανάλογα με τις ιδιότητες τους. Τέσσερις κατηγορίες τέτοιων μοντέλων είναι οι ακόλουθες (Bellemans T. et al, 2002):

- Φυσική ερμηνεία
- Επίπεδο λεπτομέρειας
- Διακριτά και Συνεχή
- Ντετερμινιστικά και Στοχαστικά

Όσον αφορά το επίπεδο λεπτομέρειας, οι δύο κύριοι τύποι μοντελοποίησης της κυκλοφοριακής ροής που εφαρμόζονται είναι οι εξής (Owen L. et al, 2000, Bellemans T. et al, 2002, Hoogendoorn and Bovy, 2000):

- 1) Μακροσκοπική μοντελοποίηση της κυκλοφοριακής ροής: Η κίνηση των οχημάτων αναπαρίσταται σαν τη ροή ενός υγρού. Δε παρακολουθούνται μεμονωμένα τα οχήματα.
- 2) Μικροσκοπική μοντελοποίηση της κυκλοφοριακής ροής: Η κίνηση κάθε οχήματος καθορίζεται από τα χαρακτηριστικά του οδηγού, της απόδοσης του οχήματος και των αλληλεπιδράσεων με τη γεωμετρία του οδικού δικτύου και των υπόλοιπων οχημάτων σε αυτό.

Τα μακροσκοπικά μοντέλα δεν είναι σε θέση να χρησιμοποιηθούν στις περιπτώσεις όπου απαιτείται λεπτομερής ανάλυση του ελέγχου της κυκλοφοριακής ροής κι αυτό γιατί μοντελοποιούν την κυκλοφοριακή ροή σαν ένα υγρό που κυλά σε κανάλια. Ένα μακροσκοπικό μοντέλο καθορίζει μια σχέση ανάμεσα στην πυκνότητα της κυκλοφοριακής ροής, της μέσης ταχύτητας και του εύρους της ταχύτητας που μπορεί να αναπτύξει η ροή των οχημάτων. Η πυκνότητα της κυκλοφοριακής ροής ορίζεται ως τον αριθμό των οχημάτων ανα χιλιόμετρο ανα λωρίδα κυκλοφορίας ενώ η ροή των οχημάτων ορίζεται ως ο αριθμός των οχημάτων που περνούν από ένα συγκεκριμένο σημείο ανα ώρα. Τα πιο παλιά μακροσκοπικά μοντέλα που προτάθηκαν, ήταν εκείνο των Lighthill και Whitham το 1955 και εκείνο του Richards το 1956. Η μόνη παράμετρος που είχαν εκείνα τα μοντέλα ήταν η πυκνότητα της κυκλοφοριακής ροής. Αργότερα, τα μοντέλα αυτά επεκτάθηκαν ούτως ώστε να ανταποκρίνονται σε καταστάσεις προχωρώ-σταματώ (stop and go) που συμβαίνουν σε καταστάσεις κυκλοφοριακής συμφόρησης. Το πλεονέκτημα των μοντέλων αυτών είναι ότι απαιτούν λιγότερο υπολογιστικό φόρτο από τα μικροσκοπικά μοντέλα επειδή δεν περιγράφουν με λεπτομέρεια τα οχήματα που βρίσκονται στο οδικό δίκτυο. Αυτό επιτρέπει τη γρήγορη

προσομοίωση του μοντέλου του οδικού δικτύου που μελετάται, κάτι το οποίο είναι πολύ σημαντικό στον on-line έλεγχο πρόβλεψης του δικτύου (Bellemans et al, 2002).

Τα μικροσκοπικά μοντέλα κυκλοφοριακής ροής ξεκινούν από την περιγραφή του οδικού δικτύου που επιθυμούμε να προσομοιώσουμε. Το οδικό δίκτυο περιγράφεται από τη μέγιστη ταχύτητα που μπορούν να αναπτύξουν τα οχήματα σε αυτό, τον αριθμό των λωρίδων κυκλοφορίας που διαθέτει, τη δυνατότητα που έχουν τα οχήματα να προσπεράσουν άλλα οχήματα ή μη, κ.λ.π. Επίσης, το κάθε όχημα που βρίσκεται στο δρόμο περιγράφεται από ορισμένες παραμέτρους οι οποίες είναι η αφετηρία του, ο προορισμός του, ο τύπος του οχήματος, η υπομονή του οδηγού, η επιθετική του συμπεριφορά κ.λ.π. Κατά τη διάρκεια της προσομοίωσης, τα οχήματα εισέρχονται στο οδικό δίκτυο και οι παράμετροι των οχημάτων (π.χ. τύπος οχήματος, επιτάχυνση που διαθέτει) και των οδηγών (π.χ. επιθετικότητα, χρόνος αντίδρασης, υπομονή), εισάγονται μέσω μιας στοχαστικής συνάρτησης κατανομής, η οποία έχει βασιστεί σε πραγματικές μετρήσεις. Τα οχήματα με τους οδηγούς τους, έχοντας καταχωρημένες τις παραμέτρους που τα χαρακτηρίζουν, προσπαθούν να φτάσουν στον προορισμό τους χρησιμοποιώντας το οδικό δίκτυο και αλληλεπιδρώντας με τα υπόλοιπα οχήματα-οδηγούς που συναντούν στο δρόμο καθώς και με τη γεωμετρία του οδικού δικτύου και των κανόνων που το διέπουν (π.χ. όρια ταχύτητας).

Τα βασικά χαρακτηριστικά της μικροσκοπικής προσομοίωσης της κυκλοφοριακής ροής είναι η λογική της ακολουθίας των αυτοκινήτων (car-following) και της αλλαγής λωρίδας κυκλοφορίας (lane-changing) (Owen L. et al, 2000).

2.2.1 Car Following Models

Η μοντελοποίηση υποθέτει ότι κάθε οδηγός προπορευόμενου οχήματος είναι ένα ενεργό και προβλέψιμο στοιχείο ελέγχου του συστήματος οδηγός-όχημα-δρόμος. Η απλή και συνήθης διαδικασία της οδήγησης ενός οχήματος, ακολουθώντας κάποιο άλλο, σε ένα δρόμο μονής λωρίδας κυκλοφορίας που δεν επιτρέπεται η προσπέραση, μπορεί να κατηγοριοποιηθεί σε τρεις συγκεκριμένες υπό-διαδικασίες.

Οι υπο-διαδικασίες αυτές είναι οι ακόλουθες τρεις (Rothery R. W., 2002).

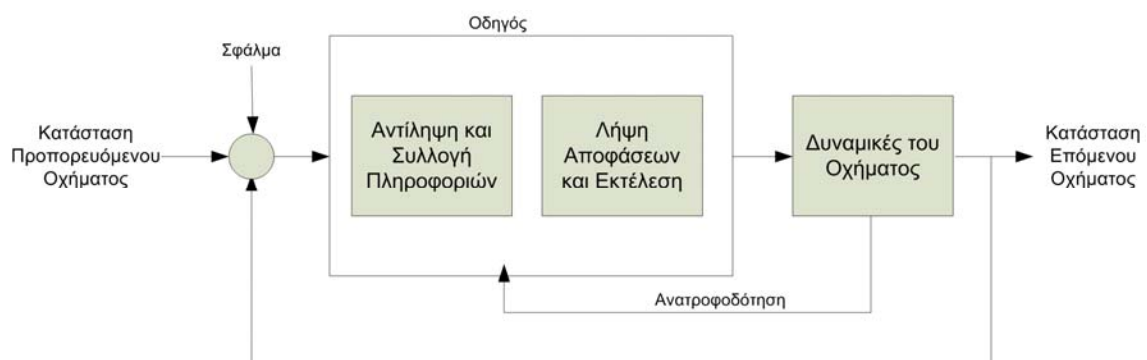
- 1) Αντίληψη (Perception)
- 2) Λήψη Αποφάσεων (Decision Making)
- 3) Έλεγχος (Control)

Κατά την υπό-διαδικασία της αντίληψης (perception) ο οδηγός συγκεντρώνει σχετικές πληροφορίες, κυρίως μέσω της όρασης. Οι πληροφορίες αυτές προκύπτουν κυρίως από την κίνηση του προπορευόμενου οχήματος και του δικού του. Μερικές πληροφορίες που συλλέγει είναι η απόσταση του με το προπορευόμενο όχημα, η ταχύτητα του οχήματος του, η επιτάχυνση/επιβράδυνση που ασκεί, κ.τ.λ.

Όσον αφορά την υπό-διαδικασία της λήψης αποφάσεων (decision making), ο οδηγός φιλτράρει τις πληροφορίες που συνέλεξε και τις ενοποιεί στην πορεία του χρόνου ούτως ώστε να είναι σε θέση να λάβει σωστές αποφάσεις. Η ερμηνεία των πληροφοριών γίνεται με βάση τη εμπειρία που διαθέτει ο οδηγός και του τύπου αυτοκινήτου που οδηγεί.

Ο έμπειρος οδηγός μπορεί να εκτελεί εντολές ελέγχου (control) με επιδεξιότητα και συντονισμό, βασισμένος στην ανατροφοδότηση που λαμβάνει από τις δικές του αντιδράσεις, οι οποίες συσχετίζονται άμεσα με το προπορευόμενο όχημα τη γεωμετρία του οδικού δικτύου και τις δυναμικές του οχήματος του.

Όπως παρατηρούμε και στο σχήμα 1 στο car following model ο οδηγός αντιλαμβάνεται και συλλέγει πληροφορίες σχετικά με την κατάσταση του προπορευόμενου οχήματος



Σχήμα 1. Car following model (Rothery R. W., 2002)

Μερικά είδη car following models είναι τα ακόλουθα (Γαλανόπουλος, 2002):

- Μοντέλο τριών οχημάτων
- Μοντέλο του Helly
- Μη συμμετρικά μοντέλα
- Μοντέλο του Gipps
- Μοντέλο των Nagel-Schreckenberg
- Μοντέλα βασιζόμενα στη θεωρία βέλτιστου ελέγχου

2.2.1.1 Γραμμικά Μοντέλα

Η εξίσωση της θέσης ενός οχήματος που βρίσκεται σε δρόμο μονής λωρίδας κυκλοφορίας που ακολουθεί το car following model δίνεται από την παρακάτω σχέση 2.1:

$$\ddot{x}_{n+1}(t+\tau) = \lambda [\dot{x}_n(t) - \dot{x}_{n+1}(t)] \quad (2.1)$$

Όπου $n=1,2,3\dots$ ο αριθμός του προπορευόμενου οχήματος και $n+1$ ο αριθμός του οχήματος που ακολουθεί, τ ο χρόνος αντίδρασης και λ ο συντελεστής ευαισθησίας.

2.2.1.2 Μη γραμμικά μοντέλα

Στα μη γραμμικά μοντέλα ο συντελεστής ευαισθησίας λ είναι μεταβλητός και προσδιορίζεται από τον ακόλουθο τύπο 2.2:

$$\lambda = \frac{\lambda_1}{x_n(t) - x_{n+1}(t)} \quad (2.2)$$

, όπου λ_1 είναι μια νέα παράμετρος η οποία θεωρείται σταθερή και συμβολίζει και πάλι έναν συντελεστή ευαισθησίας. Η εξίσωση της θέσης του οχήματος δίνεται από την παρακάτω σχέση 2.3.

$$\ddot{x}_{n+1}(t+\tau) = \frac{\lambda_1}{x_n(t) - x_{n+1}(t)} \cdot [\dot{x}_n(t) - \dot{x}_{n+1}(t)] \quad (2.3)$$

Όπου $n=1,2,3\dots$ ο αριθμός του προπορευόμενου οχήματος και $n+1$ ο αριθμός του οχήματος που ακολουθεί, τ ο χρόνος αντίδρασης και λ ο συντελεστής ευαισθησίας.

2.2.2 Μοντέλα αλλαγής λωρίδας της κυκλοφορίας (lane-changing models)

Τα μοντέλα αλλαγής λωρίδας της κυκλοφορίας περιγράφουν πως οι οδηγοί των οχημάτων μπορούν να προσπεράσουν προπορευόμενα οχήματα. Οι ιδιότητες του οχήματος και του οδηγού χρησιμοποιούνται σε αυτά τα μοντέλα και είναι για παράδειγμα η ταχύτητα που επιθυμεί να κινηθεί ο οδηγός και οι δυνατότητες του οχήματος του. Ο αριθμός των λωρίδων κυκλοφορίας εξαρτάται από το είδος του δρόμου που κινούνται τα οχήματα.

Η αλληλεπίδραση των οχημάτων είναι ένας πολύ σημαντικός παράγοντας. Κάποιος οδηγός θα τολμήσει προσπέραση, αλλάζοντας λωρίδα κυκλοφορίας, βάσει της ταχύτητας που κινείται και της ταχύτητας των οχημάτων που βρίσκονται στην αριστερή του λωρίδα καθώς επίσης βάσει του διαθέσιμου κενού στο δρόμο που θα του επιτρέψει να τολμήσει την προσπέραση με ασφάλεια.

Οποιοδήποτε κι αν είναι το είδος του μικροσκοπικού μοντέλου της κυκλοφορίας, κρίνεται απαραίτητη η χρήση κάποιου μέσου συλλογής κυκλοφοριακών δεδομένων. Ο πιο κοινός τρόπος είναι οι φωρατές επαγωγικού βρόχου (loop detectors), οι οποίοι παρουσιάζονται ακολούθως.

2.3 Φωρατές επαγωγικού βρόχου (Loop Detectors)

Οι φωρατές επαγωγικού βρόχου (loop detectors) είναι αισθητήρες που χρησιμοποιούνται για την εκτίμηση πολύτιμων κυκλοφοριακών δεδομένων σε συστήματα κυκλοφορίας οχημάτων. Από την πρώτη κιόλας στιγμή της κυκλοφορίας τους, τη δεκαετία του 1960, αποτέλεσαν την πιο διαδεδομένη μορφή ανίχνευσης οχημάτων (Wang Y. and Nihan N., 2003). Πολλά δίκτυα αυτοκινητόδρομων

χρησιμοποίησαν και χρησιμοποιούν ακόμα μονούς φωρατές επαγωγικού βρόχου (single-loop detectors) για την καταγραφή του πλήθους των οχημάτων που περνούν από το δρόμο στη μονάδα του χρόνου (volume, flow) και το χρόνο που τα οχήματα βρίσκονται πάνω από τον φωρατή (lane occupancy).

Εκτός από τους μονούς φωρατές επαγωγικού βρόχου, σε πιο σύνθετα δίκτυα αυτοκινητόδρομων, χρησιμοποιούνται οι διπλοί φωρατές επαγωγικού βρόχου (dual-loop detectors) οι οποίοι παρέχουν ακριβείς μετρήσεις της ταχύτητας και του τύπου των οχημάτων που περνούν πάνω από αυτούς. Άλλοι τύποι αισθητήρων είναι οι αισθητήρες ultrasonic doppler και οι αισθητήρες εικόνας (vision-based detectors).

Οι φωρατές επαγωγικού βρόχου, σε γενικές γραμμές, λειτουργούν μέσω του φαινομένου της επαγωγής, χρησιμοποιώντας συρμάτινους κύκλους θαμμένους στο δρόμο, οι οποίοι δημιουργούν ένα μαγνητικό πεδίο. Αντιλαμβάνονται τα οχήματα που παίρνουν από πάνω τους μέσω της διατάραξης που προκαλείται στο μαγνητικό τους πεδίο.

Στην παρούσα εργασία χρησιμοποιήθηκαν εικονικοί φωρατές επαγωγικού βρόχου στο μικροσκοπικό μοντέλο προσομοίωσης που κατασκευάστηκε ούτως ώστε να συλλεχθούν πολύτιμα κυκλοφοριακά δεδομένα.

2.4 Θεωρία Προσέγγισης (Approximation Theory) και Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks)

Η Θεωρία Προσέγγισης (Approximation Theory) προσπαθεί να βρει τρόπους με τους οποίους πολύπλοκες συναρτήσεις μπορούν να αντικατασταθούν από άλλες απλούστερες, παρουσιάζοντας το μικρότερο δυνατό σφάλμα.

Αναφορικά με το θέμα της εκτίμησης του χρόνου ταξιδιού, που απασχολεί και την παρούσα εργασία, υπάρχουν διάφορες μέθοδοι που μπορούν να χρησιμοποιηθούν για να εκτιμήσουν χρόνους διαδρομής. Οι μέθοδοι αυτοί χωρίζονται σε πέντε κατηγορίες (Sermpis et al.):

1. Μέθοδοι μέτρησης σημειακής ταχύτητας (Spot speed measurement methods)
2. Χωρικές μέθοδοι χρόνου διαδρομής (Spatial travel time methods)
3. Τεχνολογίες οχημάτων εφοδιασμένων με μηχανισμούς επικοινωνίας με το κέντρο ελέγχου κυκλοφορίας (Probe vehicle technologies)
4. Παλινδρομικά μοντέλα (Regression models)
5. Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural networks)

Η μέθοδος εκτίμησης του χρόνου διαδρομής που χρησιμοποιήθηκε στην παρούσα εργασία είναι τα Τεχνητά Νευρωνικά Δίκτυα.

Σύγχρονες ερευνητικές δραστηριότητες σχετικά με τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) κατέδειξαν ότι τα ΤΝΔ έχουν ισχυρές ιδιότητες ταξινόμησης και αναγνώρισης προτύπων. Εμπνευσμένα από τα βιολογικά συστήματα, ιδιαίτερα από την έρευνα του ανθρώπινου εγκεφάλου, τα ΤΝΔ είναι σε θέση να μαθαίνουν και να γενικεύουν από την εμπειρία.

Αποτελούν εκπαιδευσιμα αναλυτικά εργαλεία που επιχειρούν να μιμηθούν τα πρότυπα επεξεργασίας πληροφοριών του εγκεφάλου. Καθώς δεν απαιτούν απαραίτητα υποθέσεις για την κατανομή του δείγματος, οικονομολόγοι και μαθηματικοί χρησιμοποιούν τα Τεχνητά Νευρωνικά Δίκτυα για την ανάλυση δεδομένων ολοένα και περισσότερο. Όχι μόνο δεν απαιτούν υποθέσεις για την κατανομή του υποκείμενου δείγματος, αλλά επιπλέον συνιστούν ισχυρά εργαλεία πρόβλεψης συγκεντρώνοντας τις πλέον πρόσφατες εξελίξεις στην έρευνα τεχνητής νοημοσύνης .

Τα ΤΝΔ, τα οποία αρχικά αναπτύχθηκαν για τη μίμηση βασικών βιολογικών νευρικών συστημάτων, πιο συγκεκριμένα του ανθρώπινου εγκεφάλου, αποτελούνται από αλληλοσυνδεδεμένα απλά λειτουργικά στοιχεία που καλούνται νευρώνες ή κόμβοι. Κάθε κόμβος λαμβάνει ένα σήμα εισόδου, που αποτελεί τη συνολική «πληροφορία» από τους άλλους κόμβους ή το εξωτερικό ερέθισμα, το οποίο επεξεργάζεται τοπικά μέσω της συνάρτησης ενεργοποίησης ή μεταφοράς και παράγει ένα μετασηματισμένο σήμα εξόδου προς τους άλλους κόμβους ή τις εξωτερικές εξόδους. Μολονότι κάθε μεμονωμένος νευρώνας εκτελεί τη λειτουργία του αρκετά αργά και ατελώς, συλλογικά ένα δίκτυο μπορεί να πραγματοποιήσει ικανοποιητικά πλήθος εργασιών. Αυτό το χαρακτηριστικό επεξεργασίας πληροφοριών καθιστά τα ΤΝΔ ισχυρούς υπολογιστικούς

μηχανισμούς οι οποίοι είναι σε θέση να μάθουν από παραδείγματα και έπειτα να γενικεύουν σε παραδείγματα που δεν έχουν ξανασυναντήσει.

Οι νευρώνες ατομικά αποτελούν απλές δομές και η χρησιμότητα τους έγκειται στην ικανότητα τους να συνδέονται μεταξύ τους ως μέρος ενός μεγαλύτερου δικτύου. Τα τεχνητά νευρωνικά δίκτυα αποτελούνται από το σύνολο δεδομένων x_i , τα βάρη w_i , το κατώφλι u , τη συνάρτηση μεταφοράς ή ενεργοποίησης f και το δεδομένο εξόδου y , όπου i είναι ο βαθμός (ο αριθμός των εισόδων) του νευρώνα. Τα βάρη αντιπροσωπεύουν την ενίσχυση ή την από-ενίσχυση της διαδικασίας.

Τυπικώς, **οι τιμές των δεδομένων εισόδου, x_i** , είναι εξωτερικές και προκαλούνται από το περιβάλλον. Εναλλακτικά είναι δυνατό να δημιουργούνται από τα δεδομένα εξόδου των άλλων τεχνητών νευρώνων. Μπορεί να είναι διακριτές τιμές ενός συνόλου, όπως το $[0,1]$ ή πραγματικοί αριθμοί.

Τα βάρη, w_i , είναι πραγματικοί αριθμοί που καθορίζουν τη συνεισφορά κάθε δεδομένου εισόδου στο συνολικό άθροισμα βαρών του νευρώνα και τελικά στο δεδομένο εξόδου του. Ο στόχος των αλγορίθμων εκπαίδευσης των νευρωνικών δικτύων είναι ο προσδιορισμός του καλύτερου δυνατού συνόλου βαρών για το κάθε πρόβλημα. Η εύρεση του βέλτιστου συνόλου είναι τις περισσότερες φορές συνδυασμός υπολογιστικού χρόνου και ελαχιστοποίησης του σφάλματος του δικτύου.

Το κατώφλι, u , είναι ένα πραγματικός αριθμός που αφαιρείται από το συνολικό άθροισμα των βαρών των δεδομένων εισόδου. Ορισμένες φορές το κατώφλι αναφέρεται ως τιμή κλήσης. Σ' αυτή την περίπτωση, ο πραγματικός αριθμός προστίθεται στο άθροισμα των βαρών. Χάριν απλότητας, το κατώφλι μπορεί να θεωρηθεί ως ένα επιπλέον ζεύγος δεδομένων εισόδου/ βάρους, όπου $w_0 = u$ και $x_0 = -1$.

Η συνάρτηση ενεργοποίησης, f , μπορεί να είναι ιδιαίτερα απλή, παραδείγματος χάριν μπορεί να είναι η βηματική (unit step) συνάρτηση. Ωστόσο, το μοντέλο του τεχνητού νευρώνα έχει επεκταθεί ώστε να περιλαμβάνει και άλλες συναρτήσεις όπως τη σιγμοειδή, την τμηματικά γραμμική (piecewise linear), και τη συνάρτηση Gauss.

Τα δεδομένα εξόδου του τεχνητού νευρώνα, y , υπολογίζουν τα αποτελέσματα σύμφωνα με τις εξισώσεις των παραπάνω συναρτήσεων. Αυτή είναι η έξοδος της συνάρτησης ενεργοποίησης για το τρέχον άθροισμα βαρών, μείον το κατώφλι. Αυτή η τιμή μπορεί να είναι διακριτή ή πραγματική, ανάλογα με τη συνάρτηση ενεργοποίησης που χρησιμοποιήθηκε. Μόλις υπολογιστεί η έξοδος μεταβιβάζεται σε έναν άλλο νευρώνα (ή σύνολο νευρώνων) ή χρησιμοποιείται ως παράδειγμα στο εξωτερικό περιβάλλον. Η ερμηνεία της εξόδου του νευρώνα εξαρτάται από το εξεταζόμενο πρόβλημα.

Στο επόμενο κεφάλαιο περιγράφεται αναλυτικά η κατασκευή του μικροσκοπικού μοντέλου προσομοίωσης και του προσεγγιστικού αλγορίθμου, ο οποίος βασίστηκε στα ΤΝΔ.

3 ΜΕΘΟΔΟΛΟΓΙΑ

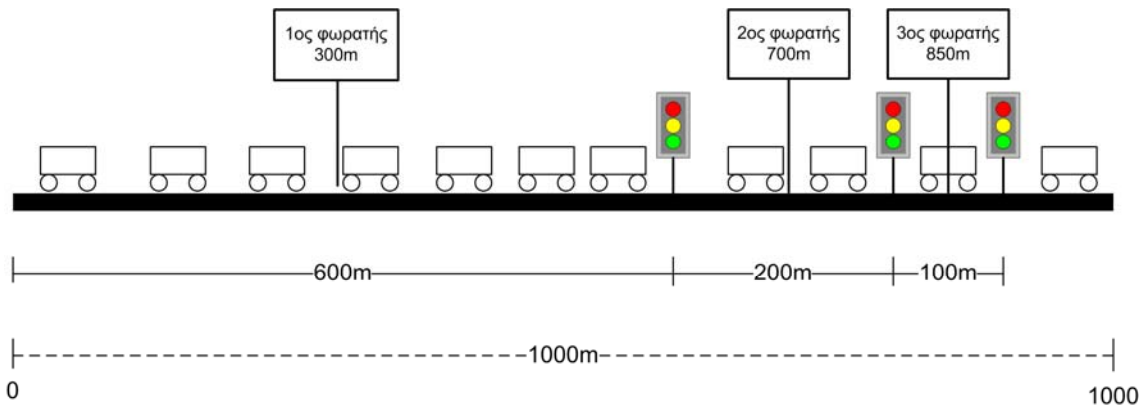
3.1 Εισαγωγή

Στο κεφάλαιο 3 παρουσιάζεται η μεθοδολογία που χρησιμοποιήθηκε στην παρούσα εργασία. Αρχικά, περιγράφεται το μικροσκοπικό μοντέλο προσομοίωσης που κατασκευάστηκε. Κατόπιν, περιγράφεται ο προσεγγιστικός αλγόριθμος που κατασκευάστηκε με σκοπό να προσεγγίσει τα αποτελέσματα της προσομοίωσης. Το μικροσκοπικό μοντέλο και ο προσεγγιστικός αλγόριθμος κατασκευάστηκαν σε γλώσσα προγραμματισμού MATLAB.

3.2 Μικροσκοπικό Μοντέλο Προσομοίωσης

3.2.1 Περιγραφή του μικροσκοπικού μοντέλου

Στο μικροσκοπικό μοντέλο προσομοίωσης, κατασκευάστηκε ένα τμήμα δρόμου μήκους ενός χιλιομέτρου με μονή λωρίδα κυκλοφορίας, στο οποίο βρίσκονται σταθερά τοποθετημένοι τρεις φωτεινοί σηματοδότες. Όπως παρατηρούμε και στο σχήμα 2, ο πρώτος φωτεινός σηματοδότης βρίσκεται στο σημείο των εξακοσίων μέτρων, ο δεύτερος στο σημείο των οκτακοσίων μέτρων και ο τρίτος στο σημείο των εννιακοσίων μέτρων.



Σχήμα 2. Οι θέσεις των φωτεινών σηματοδοτών και των φωρατών

Πρίν απο κάθε φωτεινό σηματοδότη τοποθετείται ένας μονός φωρατής επαγωγικού βρόχου (single inductive loop detector). Όπως παρατηρούμε και στο σχήμα 2 ο πρώτος φωρατής βρίσκεται πριν από τον πρώτο φωτεινό σηματοδότη, στο σημείο των τριακοσίων μέτρων. Ο δεύτερος φωρατής βρίσκεται, πριν από τον δεύτερο φωτεινό σηματοδότη, στο σημείο των επτακοσίων μέτρων και ο τρίτος φωρατής βρίσκεται, πριν από τον τρίτο φωτεινό σηματοδότη, στο σημείο των οκτακοσίων πενήντα μέτρων.

Κατά την προσομοίωση, τα οχήματα εισέρχονται στο δρόμο με τη βοήθεια μιας γεννήτριας τυχαίων χρόνων εισόδου. Η παράμετρος που χρησιμοποιήθηκε για να μπορεί να μεταβληθεί εύκολα η συχνότητα εισόδου οχημάτων στο δρόμο ήταν το alpha. Η παράμετρος alpha λαμβάνει τιμές στα πειράματα που ακολούθησαν απο 1 έως και 10. Όσο μεγαλύτερη τιμή παίρνει η παράμετρος αυτή, τόσο αραιά εισέρχονται αυτοκίνητα στο δρόμο. Αντιθέτως, όσο μικρότερη η τιμή της, τόσο πυκνότερα εισέρχονται οχήματα.

Τα οχήματα εισέρχονται στο δρόμο με τη μέγιστη επιθυμητή ταχύτητα η οποία είναι $y_{d,i}=16.5$ m/sec, δηλαδή περίπου 60km/h. Η ταχύτητα αυτή είναι η μέγιστη που μπορεί να λάβει ένα όχημα όσο βρίσκεται στο δρόμο και υπολογίζεται για κάθε όχημα i , για κάθε χρονικό βήμα $t=0,1,2,\dots,N$ (time step) του χρόνου προσομοίωσης βάσει του τύπου 3.1.

$$y_{d,i}(i,t-1) = \Lambda[\delta_i(i,t) - D], \quad (3.1)$$

όπου $\Lambda=0.7\text{sec}^{-1}$, $D=1\text{m}$ η απόσταση ασφαλείας και $\delta_i(i,t)$ είναι η απόσταση σε μέτρα που υπάρχει μεταξύ δύο διαδοχικών οχημάτων i και $i+1$. Εάν η επιθυμητή ταχύτητα υπολογιστεί μεγαλύτερη του επιτρεπόμενου ορίου, τότε τίθεται ίση με τη μέγιστη τιμή που επιτρέπεται να λάβει, δηλαδή $y_{d,i}=16.5 \text{ m/sec}$.

Τα οχήματα που βρίσκονται μέσα στο δρόμο οφείλουν να τηρούν μια απόσταση ασφαλείας μεταξύ τους. Η απόσταση αυτή έχει οριστεί ως D και είναι ίση με ένα μέτρο. Έτσι, για να αποφευχθεί η σύγκρουση δύο διαδοχικών αυτοκινήτων, το όχημα $i+1$ θα πρέπει να επιβραδύνει όταν η απόσταση $\delta_i(i,t)$ που έχει με το προπορευόμενο όχημα i τείνει να μειωθεί στην απόσταση ασφαλείας. Όταν δεν τίθεται τέτοιο θέμα κινδύνου σύγκρουσης, τότε το όχημα είναι ελεύθερο να επιταχύνει τόσο ώστε η ταχύτητα του να γίνει ίση με την μέγιστη επιθυμητή ταχύτητα $y_{d,i}$ που έχουμε θέσει.

Για κάθε όχημα i που βρίσκεται μέσα στο δρόμο υπολογίζεται η επιτάχυνση ή επιβράδυνση του με τη βοήθεια του τύπου 3.2.

$$a(i,t-1) = g(y_{d,i}(i,t-1) - y(i,t-1)), \quad (3.2)$$

όπου $g=2\text{sec}^{-1}$, $y_{d,i}$ η επιθυμητή ταχύτητα και y η ταχύτητα κάθε οχήματος i . Η επιτάχυνση του οχήματος δε θα πρέπει να υπερβεί μια μέγιστη τιμή $a_{\max} = 1.5\text{m/sec}^2$. Το ίδιο συμβαίνει και για την επιβράδυνση, καθώς δε θα πρέπει να είναι μικρότερη μιας κατώτατης τιμής $a_{\min} = -6\text{m/sec}^2$. Στην περίπτωση που η επιτάχυνση του οχήματος i υπολογιστεί μεγαλύτερη της τιμής a_{\max} , τότε $a(i,t) = a_{\max}$. Ομοίως, εάν η επιβράδυνση του οχήματος i υπολογιστεί μικρότερη της τιμής a_{\min} , τότε $a(i,t) = a_{\min}$.

Για κάθε όχημα i που βρίσκεται μέσα στο δρόμο υπολογίζεται η ταχύτητα του με τη βοήθεια του τύπου 3.3.

$$y_i = y(i,t-1) + a(i,t-1)T_{\text{sim}}, \quad (3.3)$$

όπου y η ταχύτητα, a η επιτάχυνση/επιβράδυνση και T_{sim} το βήμα της προσομοίωσης, το οποίο έχει οριστεί ίσο με 0,25.

Τέλος, για κάθε όχημα i που βρίσκεται μέσα στο δρόμο υπολογίζεται η θέση του για κάθε χρονική στιγμή με τη βοήθεια του τύπου 3.4.

$$x(i,t) = x(i,t-1) + T_{sim} \cdot y(i,t-1) + 0.5 \cdot T_{sim}^2 \cdot a(i,t-1), \quad (3.4)$$

όπου T_{sim} είναι το βήμα της προσομοίωσης, $x(i,t-1)$ είναι η θέση, $y(i,t-1)$ η ταχύτητα και $a(i,t-1)$ η επιτάχυνση/επιβράδυνση που είχε το όχημα i την αμέσως προηγούμενη χρονική στιγμή $t-1$.

Έτσι για κάθε χρονική στιγμή t της προσομοίωσης υπολογίζονται για κάθε όχημα i οι τρεις βασικές προαναφερθείσες μεταβλητές:

- η θέση του
- η ταχύτητα του
- η επιτάχυνση/επιβράδυνση του

3.2.2 Περιπτώσεις πράσινου/κόκκινου

Ο καθένας από τους τρεις φωτεινούς σηματοδότες που χρησιμοποιήθηκε μπορεί να λαμβάνει διάφορες τιμές για το χρόνο που θα ανάβει το πράσινο φανάρι και το χρόνο που θα ανάβει το κόκκινο. Το άθροισμα των δύο αυτών χρόνων είναι 90.

Στον πίνακα 1 παρατηρούμε τους τέσσερις συνδυασμούς των χρόνων των πράσινων και των κόκκινων φαναριών για τους τρεις φωτεινούς σηματοδότες. Οι συνδυασμοί αυτοί επιλέχθηκαν με τέτοιο τρόπο ώστε να είναι αρκετά διαφορετικοί μεταξύ τους.

Στην περίπτωση A, ο χρόνος που διαρκεί το πράσινο για κάθε φωτεινό σηματοδότη είναι ο ίδιος και πολύ μικρότερος από το χρόνο που διαρκεί το κόκκινο φανάρι. Μέσα από αυτό το συνδυασμό μπορούμε να παρατηρήσουμε τι συμβαίνει όταν δημιουργείται μποτιλιάρισμα στον υπό μελέτη δρόμο.

Στην περίπτωση Β οι χρόνοι που διαρκεί το πράσινο φανάρι αυξάνουν απο το πρώτο μέχρι και το τρίτο φανάρι. Δημιουργείται δηλαδή ένα μποτιλιάρισμα στο ύψος του πρώτου φωτεινού σηματοδότη το οποίο σταδιακά επιδιώκεται να εξαλειφθεί με τη διάρκεια όλο και μεγαλύτερων χρόνων για το πράσινο φανάρι στους επόμενους δύο φωτεινούς σηματοδότες.

Η περίπτωση Γ είναι το αντίστροφο της περίπτωσης Β. Οι χρόνοι που διαρκεί το πράσινο φανάρι στους τρεις φωτεινούς σηματοδότες μικραίνουν σταδιακά από τον πρώτο έως και τον τρίτο φωτεινό σηματοδότη.

Στην περίπτωση Δ, ο χρόνος που διαρκεί το πράσινο φανάρι είναι πολύ μεγαλύτερος απο το χρόνο που διαρκεί το κόκκινο και ίδιος και για τους τρεις φωτεινούς σηματοδότες. Έτσι, η κυκλοφοριακή ροή γίνεται όσο το δυνατόν γρηγορότερη αφού τα οχήματα δεν αναγκάζονται να σταματούν στο υπό μελέτη τμήμα του δρόμου.

Πίνακας 1. Συνδυασμοί χρόνων πράσινου/κόκκινου για τους τρεις φωτεινούς σηματοδότες

Περιπτώσεις	Alpha	1 ^{ος} φωτεινός σηματοδότης		2 ^{ος} φωτεινός σηματοδότης		3 ^{ος} φωτεινός σηματοδότης	
		πράσινο	κόκκινο	πράσινο	κόκκινο	πράσινο	κόκκινο
A	1,3,5,8,10	20	70	20	70	20	70
B	1,3,5,8,10	30	60	40	50	70	20
Γ	1,3,5,8,10	70	20	40	50	30	60
Δ	1,3,5,8,10	80	10	80	10	80	10

Για καθεμιά απο τις παραπάνω περιπτώσεις χρησιμοποιήθηκαν διαφορετικές τιμές της παραμέτρου alpha (1,3,5,8,10), η οποία καθορίζει τη συχνότητα των χρονικών στιγμών που εισέρχεται όχημα στο δρόμο. Όσο αυξάνεται η τιμή της παραμέτρου alpha, τόσο πιο αραιά εισέρχονται οχήματα στο δρόμο και όσο μειώνεται η τιμή της, τόσο πυκνότερα εισέρχονται οχήματα στο δρόμο.

Ο χρόνος της προσομοίωσης είναι $N=5000$ και το βήμα της (time step) είναι $T=0.25$.

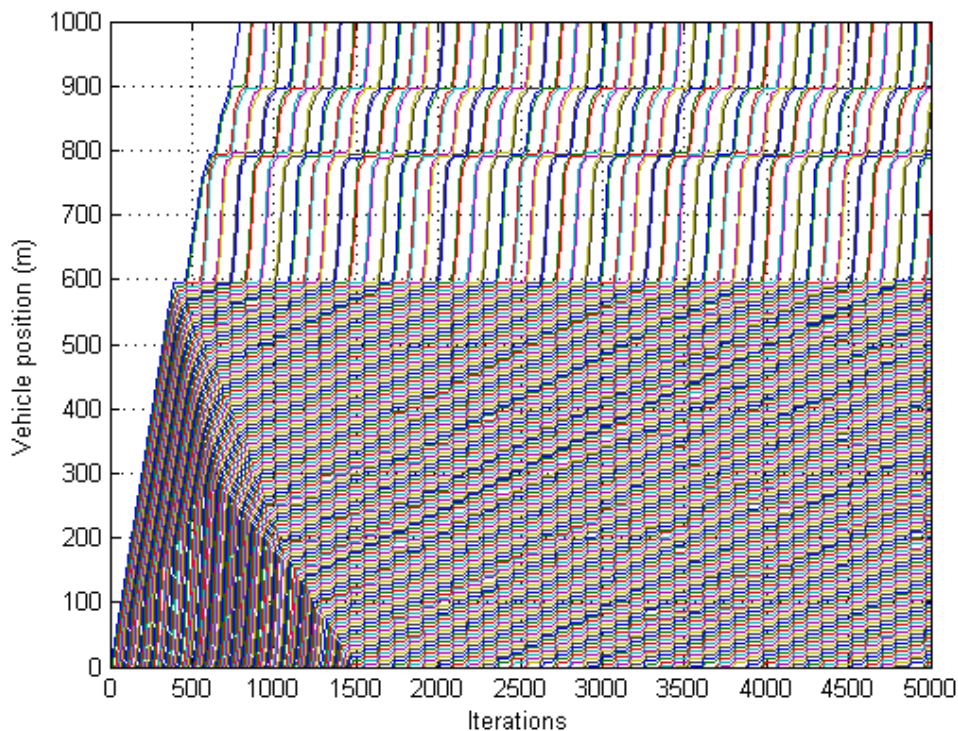
3.2.2.1 Περίπτωση Α

Η περίπτωση Α είναι εκείνη όπου δημιουργείται μποτιλιάρισμα στο δρόμο. Οι χρόνοι που διαρκεί το πράσινο φανάρι σε κάθε φωτεινό σηματοδότη είναι πολύ μικρότεροι από εκείνους που διαρκεί το κόκκινο. Πιο συγκεκριμένα ο χρόνος που διαρκεί το πράσινο φανάρι σε κάθε φωτεινό σηματοδότη είναι ίσος με 20. Όπως ήδη αναφέρθηκε οι χρόνοι που διαρκούν το πράσινο και το κόκκινο φανάρι σε κάθε φωτεινό σηματοδότη αθροίζουν στη τιμή 90. Συνεπώς ο χρόνος που θα διαρκεί το κόκκινο φανάρι σε κάθε φωτεινό σηματοδότη θα είναι ίσος με 90.

Πίνακας 2. Συνδυασμοί χρόνων πράσινου/κόκκινου για την περίπτωση Α

Περιπτώσεις	Alpha	1 ^{ος} φωτεινός σηματοδότης		2 ^{ος} φωτεινός σηματοδότης		3 ^{ος} φωτεινός σηματοδότης	
		πράσινο	κόκκινο	πράσινο	κόκκινο	πράσινο	κόκκινο
A	1,3,5,8,10	20	70	20	70	20	70

Στο σχήμα 3 απεικονίζονται οι θέσεις των οχημάτων x_i συναρτήσει του χρόνου t και για αριθμό προσομοιώσεων $N=5000$. Παρατηρούμε ότι έχει δημιουργηθεί κυκλοφοριακή συμφόρηση μέχρι το ύψος του πρώτου φωτεινού σηματοδότη, δηλαδή στα 600 μέτρα. Αυτό οφείλεται στο γεγονός της μεγάλης χρονικής διάρκειας του κόκκινου φαναριού του πρώτου φωτεινού σηματοδότη. Συνεπώς, δεν είναι μεγάλος ο αριθμός των οχημάτων που παίρνουν τον πρώτο φωτεινό σηματοδότη. Οι καμπύλες που παρατηρούνται στα 800 και 900 μέτρα οφείλονται στην ύπαρξη του φωτεινού σηματοδότη δύο και τρία. Όταν το φανάρι είναι κόκκινο, το όχημα επιβραδύνει και σταματά πέντε μέτρα πίσω από το φανάρι. Με τη σειρά τους τα οχήματα που ακολουθούν επιβραδύνουν και σταματούν διαδοχικά πίσω από το προπορευόμενο όχημα του καθενός. Όταν το φανάρι γίνει πράσινο, το πρώτο όχημα επιταχύνει και με τη σειρά τους επιταχύνουν και τα οχήματα που ακολουθούν διατηρώντας πάντα την απόσταση ασφαλείας που τους έχει δωθεί.



Σχήμα 3. Η θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για $N=5000$ επαναλήψεις της περίπτωσης A

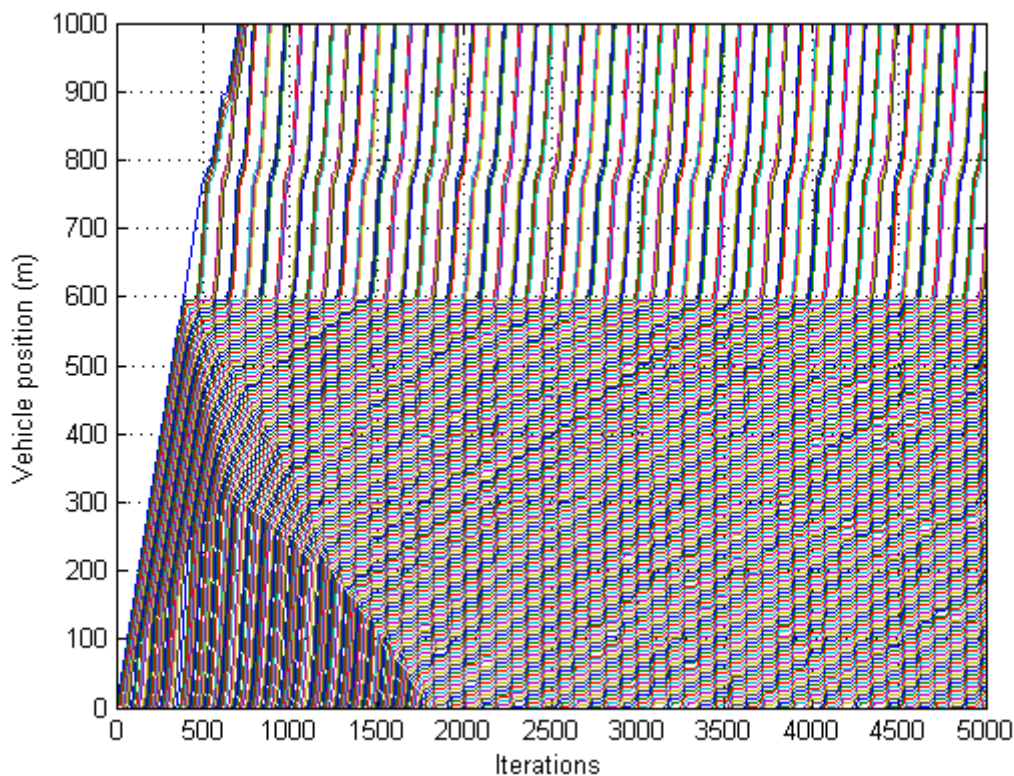
3.2.2.2 Περίπτωση B

Στην περίπτωση B οι χρόνοι που διαρκεί το πράσινο φανάρι αυξάνουν απο το πρώτο μέχρι και το τρίτο φανάρι. Πιο συγκεκριμένα, ο χρόνος που διαρκεί το πράσινο φανάρι στον πρώτο φωτεινό σηματοδότη είναι ίσος με 30 και ο χρόνος που διαρκεί το κόκκινο ίσος με 60. Ο χρόνος που διαρκεί το πράσινο φανάρι στο δεύτερο και τον τρίτο φωτεινό σηματοδότη είναι 40 και 70 αντίστοιχα. Οπότε, ο χρόνος που διαρκεί το κόκκινο φανάρι είναι 50 και 20 αντίστοιχα.

Πίνακας 3. Συνδυασμοί χρόνων πράσινου/κόκκινου για την περίπτωση B

Περιπτώσεις	Alpha	1 ^{ος} φωτεινός σηματοδότης		2 ^{ος} φωτεινός σηματοδότης		3 ^{ος} φωτεινός σηματοδότης	
		πράσινο	κόκκινο	πράσινο	κόκκινο	πράσινο	κόκκινο
B	1,3,5,8,10	30	60	40	50	70	20

Δημιουργείται, δηλαδή, ένα μποτιλιάρισμα στο ύψος του πρώτου φωτεινού σηματοδότη το οποίο σταδιακά επιδιώκεται να εξαλειφθεί με τη διάρκεια όλο και μεγαλύτερων χρόνων για το πράσινο φανάρι στους επόμενους δύο φωτεινούς σηματοδότες. Αυτό ακριβώς φαίνεται και στο σχήμα 4 όπου απεικονίζονται οι θέσεις των οχημάτων x_i συναρτήσει του χρόνου t και για αριθμό προσομοιώσεων $N=5000$. Παρατηρούμε ότι στο ύψος του πρώτου φωτεινού σηματοδότη υπάρχει μεγάλος συνωστισμός οχημάτων παρόμοιας κλίμακας με εκείνον που παρατηρήθηκε στην περίπτωση Α. Όμως, στο ύψος του δεύτερου και του τρίτου φωτεινού σηματοδότη δεν είναι τόσο έντονες οι καμπύλες της θέσης των οχημάτων όσο ήταν εκείνες της περίπτωσης Α. Ακόμα, οι καμπύλες που παρατηρούμε στο ύψος του τρίτου φωτεινού σηματοδότη, στα 900 μέτρα, είναι μικρότερες από εκείνες που παρατηρούμε στο ύψος του δεύτερου φωτεινού σηματοδότη, στα 800 μέτρα. Αυτό συμβαίνει γιατί το χρόνο που διαρκεί το πράσινο φανάρι στον τρίτο φωτεινό σηματοδότη είναι μεγαλύτερος από εκείνον του δεύτερου φωτεινού σηματοδότη, επιτρέποντας έτσι μεγαλύτερη διέλευση οχημάτων.



Σχήμα 4. Η θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για $N=5000$ επαναλήψεις της περίπτωσης Β

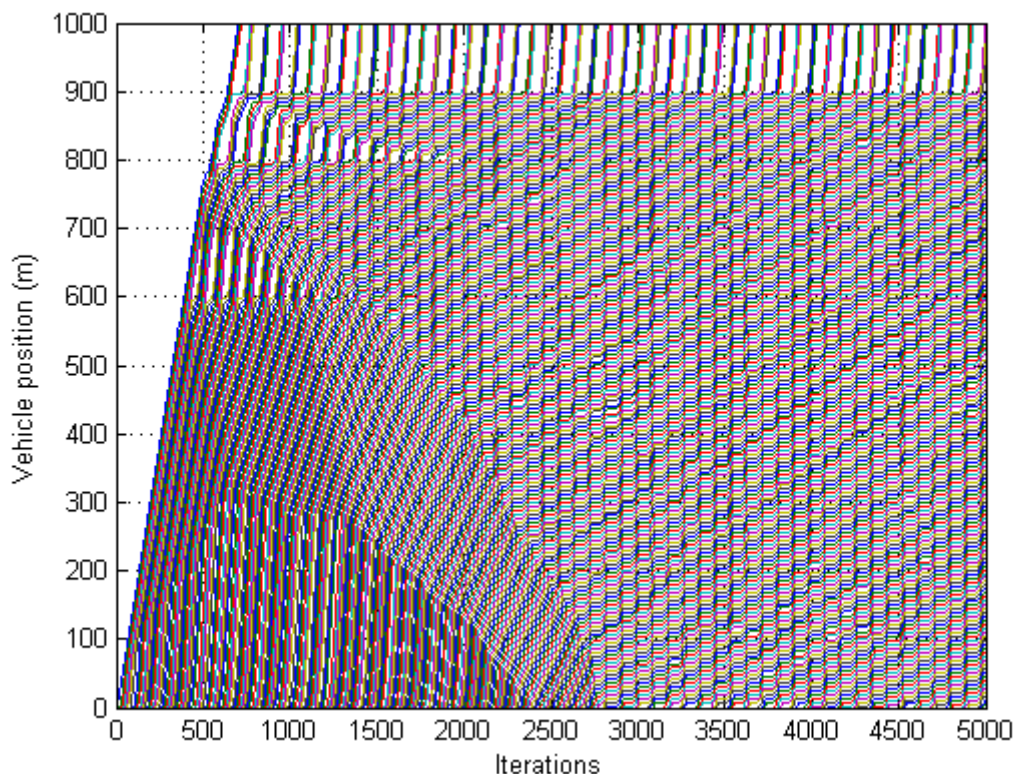
3.2.2.3 Περίπτωση Γ

Η περίπτωση Γ είναι το αντίστροφο της περίπτωσης Β. Οι χρόνοι που διαρκεί το πράσινο φανάρι στους τρεις φωτεινούς σηματοδότες μικραίνουν σταδιακά από τον πρώτο έως και τον τρίτο φωτεινό σηματοδότη. Πιο συγκεκριμένα, ο χρόνος που διαρκεί το πράσινο φανάρι στον πρώτο φωτεινό σηματοδότη είναι ίσος με 70 και ο χρόνος που διαρκεί το κόκκινο ίσος με 20. Ο χρόνος που διαρκεί το πράσινο φανάρι στο δεύτερο και τον τρίτο φωτεινό σηματοδότη είναι 40 και 30 αντίστοιχα. Οπότε, ο χρόνος που διαρκεί το κόκκινο φανάρι είναι 50 και 60 αντίστοιχα.

Πίνακας 4. Συνδυασμοί χρόνων πράσινου/κόκκινου για την περίπτωση Γ

Περίπτώσεις	Alpha	1 ^{ος} φωτεινός σηματοδότης		2 ^{ος} φωτεινός σηματοδότης		3 ^{ος} φωτεινός σηματοδότης	
		πράσινο	κόκκινο	πράσινο	κόκκινο	πράσινο	κόκκινο
Γ	1,3,5,8,10	70	20	40	50	30	60

Στο σχήμα 5 μπορούμε να παρατηρήσουμε ότι η διέλευση των οχημάτων στο ύψος του πρώτου φωτεινού σηματοδότη (600 μέτρα) είναι ομαλή, λόγω του μεγάλου χρόνου που διαρκεί το πράσινο φανάρι. Στο ύψος του δεύτερου φωτεινού σηματοδότη (800 μέτρα) παρατηρούμε ότι οι καμπύλες που αναπαριστούν τη θέση των οχημάτων στη διάρκεια του χρόνου είναι πιο έντονες από εκείνες του πρώτου φωτεινού σηματοδότη. Οι καμπύλες στο ύψος του τρίτου φωτεινού σηματοδότη είναι ακόμα πιο έντονες.



Σχήμα 5. Η θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για $N=5000$ επαναλήψεις της περίπτωσης Γ

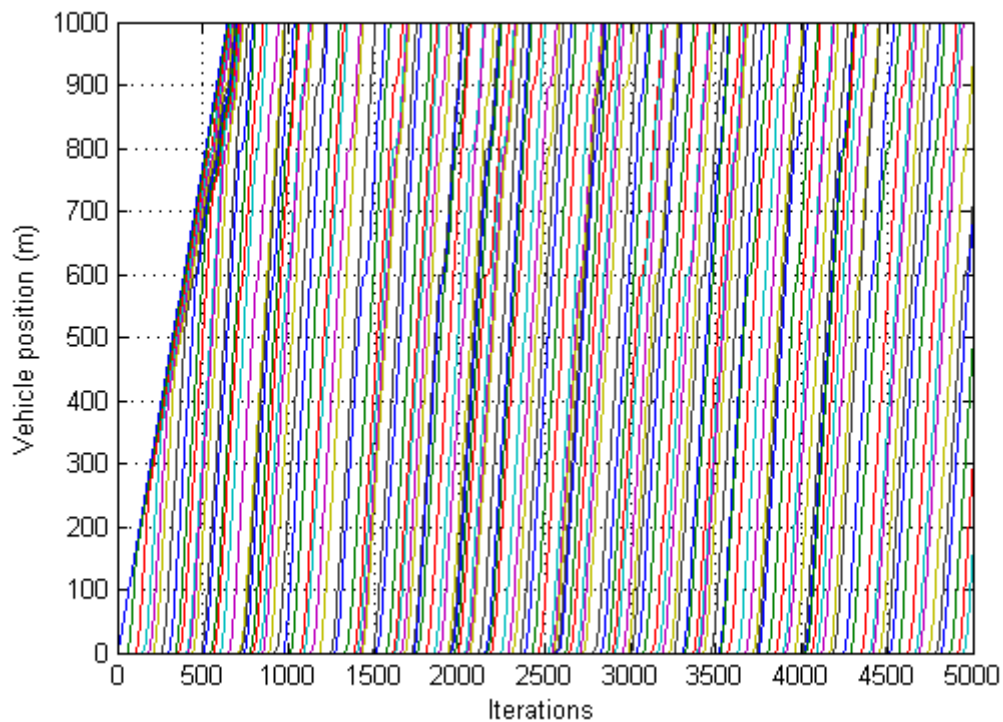
3.2.2.4 Περίπτωση Δ

Στην περίπτωση Δ, ο χρόνος που διαρκεί το πράσινο φανάρι είναι πολύ μεγαλύτερος από το χρόνο που διαρκεί το κόκκινο και ίδιος και για τους τρεις φωτεινούς σηματοδότες. Πιο συγκεκριμένα, ο χρόνος που διαρκεί το πράσινο φανάρι και στους τρεις φωτεινούς σηματοδότες είναι ίσος με 80 και άρα ο χρόνος του κόκκινου είναι ίσος με 10. Έτσι, η κυκλοφοριακή ροή γίνεται όσο το δυνατόν γρηγορότερη αφού τα οχήματα δεν αναγκάζονται να σταματούν στο υπό μελέτη τμήμα του δρόμου

Πίνακας 5. Συνδυασμοί χρόνου πράσινου/κόκκινου για την περίπτωση Δ

Περίπτώσεις	Alpha	1 ^{ος} φωτεινός σηματοδότης		2 ^{ος} φωτεινός σηματοδότης		3 ^{ος} φωτεινός σηματοδότης	
		πράσινο	κόκκινο	πράσινο	κόκκινο	πράσινο	κόκκινο
Δ	1,3,5,8,10	80	10	80	10	80	10

Ακριβώς γι' αυτό το λόγο, όπως μπορούμε να παρατηρήσουμε στο σχήμα 6 τα οχήματα κινούνται ομαλά και γρήγορα χωρίς να χρειάζεται να σταματούν στους φωτεινούς σηματοδότες και να δημιουργούνται οι καμπύλες που παρατηρήθηκαν στις προηγούμενες τρεις περιπτώσεις. Σε αυτό το σημείο σημειώνεται ότι η παράμετρος α για το σχήμα 6. είναι ίση με δέκα, το οποίο σημαίνει ότι λιγότερα οχήματα εισέρχονται στο δρόμο που κατασκευάστηκε σε σχέση με τις προηγούμενες περιπτώσεις. Έτσι είναι ευκολότερη η παρατήρηση της περίπτωσης αυτής.



Σχήμα 6. Θέση των οχημάτων στο δρόμο ανά βήμα προσομοίωσης για $N=5000$ επαναλήψεις της περίπτωσης Δ

3.3 Προσεγγιστικός Αλγόριθμος

3.3.1 Κατασκευή προσεγγιστικής συνάρτησης

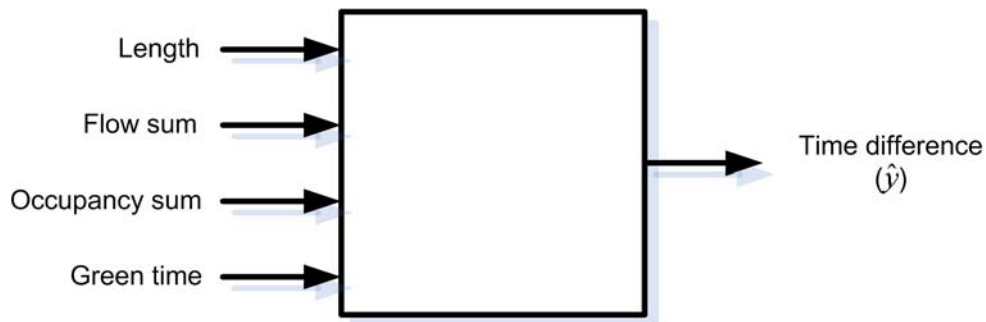
Έπειτα από την κατασκευή του μοντέλου προσομοίωσης και της επιλογής των τεσσάρων περιπτώσεων χρόνων πράσινου/κόκκινου των φωτεινών σηματοδοτών που ήδη αναφέρθηκαν, κατασκευάστηκε μια συνάρτηση με στόχο την όσο δυνατόν καλύτερη προσέγγιση του χρόνου που χρειάζεται κάθε όχημα για να διασχίσει καθένα από τα τρία τμήματα του υπό μελέτη δρόμου. Η συνάρτηση αυτή ονομάζεται «εκτιμήτρια» του προβλήματος μας.

Στο σύστημα μας εισέρχονται κάποιες είσοδοι (inputs) και εξέρχονται κάποιοι έξοδοι (outputs). Η εκτιμήτρια συνάρτηση προσπαθεί να προσεγγίσει την έξοδο που επιθυμούμε, εν προκειμένω το χρόνο που χρειάζεται κάθε όχημα για να διανύσει καθένα από τα τρία τμήματα του δρόμου, y .

Οι είσοδοι που επιλέχθηκαν για το σύστημα μας ήταν οι εξής:

- Μήκος του κάθε τμήματος (length)
- Άθροισμα των τιμών flow για τις t προηγούμενες χρονικές στιγμές του αντίστοιχου φωρατή (flow sum)
- Άθροισμα των τιμών occupancy για τις t προηγούμενες χρονικές στιγμές του αντίστοιχου φωρατή (occupancy sum)
- Χρόνος που είναι πράσινο το φανάρι του φωτεινού σηματοδότη του τμήματος στο οποίο βρίσκεται το όχημα κάθε φορά (green time)

Στο σχήμα 7 το τετράγωνο αναπαριστά το σύστημα μας. Στο σύστημα αυτό εισέρχονται οι μεταβλητές που προαναφέρθηκαν και εξέρχεται η μεταβλητή y , την οποία και προσπαθούμε να προσεγγίσουμε με όσο το δυνατόν μικρότερο σφάλμα. Η εκτίμηση της μεταβλητής y θα συμβολίζεται με \hat{y} .



Σχήμα 7. Σχηματική απεικόνιση του συστήματος μας, των εισερχόμενων και εξερχόμενων μεταβλητών

Η έξοδος του συστήματος θα είναι ο εκτιμώμενος χρόνος που χρειάζεται κάθε όχημα για να διασχίσει καθένα από τα τρία τμήματα του δρόμου (time difference), \hat{y} . Ο εκτιμώμενος χρόνος \hat{y} υπολογίστηκε μέσω της ακόλουθης συνάρτησης 3.5.

Εκτιμήτρια Συνάρτηση (3.5)

$$\hat{y} = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + w_4 \cdot x_4 \quad (\text{όροι } 1^{\text{ης}} \text{ τάξης})$$

$$+ w_5 \cdot x_1 \cdot x_2 + w_6 \cdot x_1 \cdot x_3 + w_7 \cdot x_1 \cdot x_4 + w_8 \cdot x_2 \cdot x_3 + w_9 \cdot x_2 \cdot x_4 + w_{10} \cdot x_3 \cdot x_4 \quad (\text{όροι } 2^{\text{ης}} \text{ τάξης})$$

$$+ w_{11} \cdot x_1 \cdot x_2 \cdot x_3 + w_{12} \cdot x_2 \cdot x_3 \cdot x_4 + w_{13} \cdot x_1 \cdot x_3 \cdot x_4 + w_{14} \cdot x_1 \cdot x_2 \cdot x_4 \quad (\text{όροι } 3^{\text{ης}} \text{ τάξης})$$

, όπου οι παράμετροι $w=1,2,\dots,14$ είναι τα βάρη κάθε όρου. Τα βάρη είναι άγνωστοι όροι οι οποίοι θα πρέπει να βρεθούν. Υπολογίζονται μέσω μιας επαναληπτικής διαδικασίας των ακόλουθων τύπων (3.6) – (3.19)

$$w_1 = w_1 + \gamma \cdot e \cdot x_1 \quad (3.6)$$

$$w_2 = w_2 + \gamma \cdot e \cdot x_2 \quad (3.7)$$

$$w_3 = w_3 + \gamma \cdot e \cdot x_3 \quad (3.8)$$

$$w_4 = w_4 + \gamma \cdot e \cdot x_4 \quad (3.9)$$

$$w_5 = w_5 + \gamma \cdot e \cdot x_1 \cdot x_2 \quad (3.10)$$

$$w_6 = w_6 + \gamma \cdot e \cdot x_1 \cdot x_3 \quad (3.11)$$

$$w_7 = w_7 + \gamma \cdot e \cdot x_1 \cdot x_4 \quad (3.12)$$

$$w_8 = w_8 + \gamma \cdot e \cdot x_2 \cdot x_3 \quad (3.13)$$

$$w_9 = w_9 + \gamma \cdot e \cdot x_2 \cdot x_4 \quad (3.14)$$

$$w_{10} = w_{10} + \gamma \cdot e \cdot x_3 \cdot x_4 \quad (3.15)$$

$$w_{11} = w_{11} + \gamma \cdot e \cdot x_1 \cdot x_2 \cdot x_3 \quad (3.16)$$

$$w_{12} = w_{12} + \gamma \cdot e \cdot x_2 \cdot x_3 \cdot x_4 \quad (3.17)$$

$$w_{13} = w_{13} + \gamma \cdot e \cdot x_1 \cdot x_3 \cdot x_4 \quad (3.18)$$

$$w_{14} = w_{14} + \gamma \cdot e \cdot x_1 \cdot x_2 \cdot x_4 \quad (3.19)$$

, όπου γ είναι το ποσοστό μάθησης (learning rate) το οποίο ορίζεται από τον χρήστη, e το σφάλμα μεταξύ των μεταβλητών y και \hat{y} , x_1 η τιμή της παραμέτρου length, x_2 η τιμή της παραμέτρου flow sum, x_3 η τιμή της παραμέτρου occupancy sum και x_4 η τιμή της παραμέτρου green time. Η τιμή της παραμέτρου γ επιλέχθηκε ίση με 0.000000000000001 μιας και το σφάλμα ήταν το μικρότερο που μπορούσε να βρεθεί. Για μικρότερες τιμές το σφάλμα γινόταν μεγαλύτερο και για μικρότερες το κόστος του υπολογιστικού χρόνου παρουσίαζε πάρα πολύ μεγάλη αύξηση.

3.3.2 Κατασκευή του πίνακα δεδομένων

Κατασκευάστηκε ένας πίνακας, ο οποίος ονομάστηκε A, στον οποίο συγκεντρώθηκαν τα δεδομένα που προέκυψαν από την προσομοίωση. Όπως ήδη αναφέρθηκε, τέσσερις είναι οι περιπτώσεις των συνδυασμών των χρόνων πράσινου/κόκκινου των φωτεινών σηματοδοτών που μελετήθηκαν. Σε καθεμιά από τις περιπτώσεις μεταβαλλόταν η παράμετρος alpha της γεννήτριας χρόνων εισόδου της προσομοίωσης, η οποία ελέγχει την συχνότητα εισόδου των οχημάτων στο δρόμο. Οι τιμές που λαμβάνει η μεταβλητή alpha κυμαίνονται από ένα έως και δέκα, όπου η μεγαλύτερη τιμή συνεπάγεται μεγαλύτερα διαστήματα ανάμεσα στους χρόνους εισόδων των οχημάτων, δηλαδή λιγότερα οχήματα στο δρόμο. Πιο συγκεκριμένα, οι τιμές που έλαβε η μεταβλητή alpha ήταν 1, 3, 5, 8 και 10. Συνεπώς οι περιπτώσεις που μελετήθηκαν έφτασαν στο πλήθος τον αριθμό είκοσι. Για καθεμιά από τις περιπτώσεις αυτές κατασκευάστηκε ένας πίνακας A_i , $i = 1, 2, \dots, 20$, τμήμα του πίνακα A ο οποίος παρουσιάζεται στη συνέχεια.

Στην πρώτη στήλη του πίνακα A, υπάρχει η μεταβλητή “length” (x_1), η οποία δείχνει σε ποιο κομμάτι του δρόμου βρίσκεται το όχημα που εξετάζεται. Μπορεί να λάβει μια από τις τιμές 600, 200 ή 100, οι οποίες αντιστοιχούν σε τρία τμήματα του δρόμου. Η τιμή 600 αντιστοιχεί στο τμήμα του δρόμου που ξεκινά από την αρχή του έως και το πρώτο φανάρι, η τιμή 200 αντιστοιχεί στο τμήμα του δρόμου μεταξύ του πρώτου φωτεινού σηματοδότη έως και του δεύτερου και η τιμή 100 στο τμήμα του δρόμου μεταξύ του δεύτερου φωτεινού σηματοδότη έως και του τρίτου.

Στη δεύτερη στήλη του πίνακα A υπάρχει η μεταβλητή “flow sum” (x_2), η οποία έχει προκύψει μετά από τις μετρήσεις των τριών φωρατών που τοποθετήθηκαν σε τρία τμήματα του δρόμου κατά την προσομοίωση. Η μεταβλητή αυτή προκύπτει από το άθροισμα των μετρήσεων των t προηγούμενων χρονικών στιγμών της μεταβλητής “flow” του φωρατή ο οποίος βρίσκεται στο αντίστοιχο τμήμα του δρόμου με το όχημα που εξετάζεται την κάθε φορά. Για παράδειγμα, εάν το όχημα που εξετάζουμε βρίσκεται στο τρίτο τμήμα του δρόμου που έχει μήκος 100 μέτρα, τότε θα χρησιμοποιήσουμε τις μετρήσεις του τρίτου κατά σειρά φωρατή στο δρόμο. Όσον αφορά την επιλογή του αριθμού των προηγούμενων χρονικών στιγμών (t), επιλέχθηκαν τρία σενάρια. Στο πρώτο το t έλαβε την τιμή 50, στο δεύτερο την τιμή 100 και στο τρίτο σενάριο την τιμή 300.

Στην τρίτη στήλη του πίνακα A υπάρχει η μεταβλητή “occupancy sum” (x_3), η οποία έχει προκύψει κι αυτή μετά από τις μετρήσεις των τριών φωρατών που τοποθετήθηκαν σε τρία τμήματα του δρόμου κατά την προσομοίωση. Η μεταβλητή αυτή προκύπτει από το άθροισμα των μετρήσεων των t προηγούμενων χρονικών στιγμών της μεταβλητής “occupancy” του φωρατή ο οποίος βρίσκεται στο αντίστοιχο τμήμα του δρόμου με το όχημα που εξετάζεται την κάθε φορά. Για παράδειγμα, εάν το όχημα που εξετάζουμε βρίσκεται στο δεύτερο τμήμα του δρόμου που έχει μήκος 200 μέτρα, τότε θα χρησιμοποιήσουμε τις μετρήσεις του δεύτερου κατά σειρά φωρατή στο δρόμο. Όσον αφορά την επιλογή του αριθμού των προηγούμενων χρονικών στιγμών, ισχύουν ακριβώς τα ίδια με τη μεταβλητή “flow sum”.

Στην τέταρτη στήλη του πίνακα A υπάρχει η μεταβλητή “green time” (x_4), η οποία δείχνει το χρόνο που είναι πράσινο το φανάρι στον αντίστοιχο φωτεινό σηματοδότη που

έχει μπροστά του το όχημα, ανάλογα με το τμήμα του δρόμου στο οποίο βρίσκεται κάθε φορά. Έτσι, εάν το όχημα βρίσκεται για παράδειγμα στο πρώτο τμήμα του δρόμου και μελετούμε την περίπτωση Β, τότε η μεταβλητή “length” θα είναι ίση με 600 και η μεταβλητή “green time” θα είναι ίση με 30.

Στην πέμπτη και τελευταία στήλη του πίνακα Α υπάρχει η μεταβλητή “time difference” (y), η οποία δείχνει τον χρόνο που χρειάστηκε το όχημα για να διασχίσει το αντίστοιχο τμήμα του δρόμου, το οποίο μας δείχνει η μεταβλητή “length”. Ο χρόνος που αντιστοιχεί για το πρώτο τμήμα του δρόμου των 600 μέτρων, είναι ο χρόνος που έκανε το όχημα από τη στιγμή που μπήκε στο δρόμο μέχρι και τη στιγμή που πέρασε τον πρώτο φωτεινό σηματοδότη. Ο χρόνος που αντιστοιχεί στο δεύτερο τμήμα του δρόμου, των 200 μέτρων, είναι ο χρόνος από τη στιγμή που πέρασε τον πρώτο φωτεινό σηματοδότη έως και τη χρονική στιγμή που πέρασε το δεύτερο. Ο χρόνος που αντιστοιχεί στο τρίτο τμήμα του δρόμου, των 100 μέτρων, είναι ο χρόνος από τη στιγμή που πέρασε το δεύτερο φωτεινό σηματοδότη έως και τη χρονική στιγμή που πέρασε τον τρίτο.

Όπως παρατηρούμε στο σχήμα 8, ο κάθε πίνακας A_i αποτελείται από εξακόσιες γραμμές και τις πέντε στήλες που παρουσιάστηκαν πρωτίτερα. Οι γραμμές αντιστοιχούν στα οχήματα που βρέθηκαν στο δρόμο στη διάρκεια της προσομοίωσης. Μέσα από τις προσομοιώσεις βρέθηκε ότι τα οχήματα που πέρασαν από το δρόμο σε κάθε περίπτωση ήταν λιγότερα από διακόσια. Οι πρώτες διακόσιες γραμμές αντιστοιχούν στα οχήματα που πέρασαν μέσα από το πρώτο τμήμα του δρόμου, οι επόμενες διακόσιες στα οχήματα που βρέθηκαν στο δεύτερο τμήμα του δρόμου και οι τελευταίες διακόσιες στα οχήματα που βρέθηκαν στο τρίτο τμήμα του δρόμου. Ο αριθμός διακόσια επιλέχθηκε για λόγους ασφαλείας, ούτως ώστε εάν άλλαζαν κάποιες μεταβλητές να ήταν δυνατή η κατασκευή των πινάκων A_i με τις λιγότερες δυνατές ενέργειες.

	length	flow sum	occupancy sum	green time	Time difference	
Οχήματα i=600						Οχήματα του πρώτου τμήματος του δρόμου i=200
						Οχήματα του δεύτερου τμήματος του δρόμου i=200
						Οχήματα του τρίτου τμήματος του δρόμου i=200

Σχήμα 8. Η μορφή του κάθε τμήματος του πίνακα A για κάθε περίπτωση και κάθε τιμή της μεταβλητής alpha

Αφού κατασκευαστούν οι είκοσι αυτοί πίνακες, ανακατεύονται και τοποθετούνται τυχαία ο ένας κάτω από τον άλλο. Έτσι σχηματίζεται ο πίνακας A, ο οποίος έχει διαστάσεις $A(12000,5)$.

3.3.3 Εύρεση Σφάλματος

Ο πίνακας των δεδομένων (A) χωρίστηκε σε δύο μέρη, το δείγμα εκπαίδευσης (training set) και το δείγμα αξιολόγησης (test set). Οι 10000 πρώτες γραμμές του πίνακα συνιστούν το δείγμα εκπαίδευσης και οι υπόλοιπες 2000 το δείγμα αξιολόγησης. Το δείγμα εκπαίδευσης είναι εκείνο στο οποίο υπολογίζονται οι τιμές των βαρών w_i , όπου $i=1,2,\dots,14$. Στο δείγμα αξιολόγησης, αντικαθίστανται οι τιμές των βαρών που υπολογίστηκαν στην εκτιμήτρια συνάρτηση \hat{y} . Μέσω του σφάλματος είμαστε σε θέση να συμπεράνουμε κατά πόσο τα βάρη που βρέθηκαν στο δείγμα εκπαίδευσης είναι ικανοποιητικά για τα δεδομένα του δείγματος αξιολόγησης.

Το πρώτο σφάλμα που βρίσκουμε, όταν ο αλγόριθμος υπολογίζει τις τιμές των βαρών ονομάζεται Μέσο Τετραγωνικό Σφάλμα (Mean Square Error – MSE). Έπειτα, έχοντας βρει τις τιμές των βαρών, τις αντικαθιστούμε στην εκτιμήτρια συνάρτηση \hat{y} και βρίσκουμε ένα δεύτερο σφάλμα για τα δεδομένα του δείγματος εκπαίδευσης που ονομάζεται Μέσο Τετραγωνικό Σφάλμα 2 (Mean Square Error 2 – MSE 2). Τέλος, αντικαθιστώντας και πάλι τα ίδια βάρη στην εκτιμήτρια συνάρτηση, βρίσκουμε το σφάλμα για τα δεδομένα του δείγματος αξιολόγησης. Με αυτόν τον τρόπο είμαστε σε θέση να δούμε αν τα βάρη που υπολογίστηκαν μπορούν να δώσουν μια καλή προσέγγιση του εκτιμώμενου χρόνου διαδρομής \hat{y} . Αυτό το σφάλμα το ονομάζουμε Μέσο Τετραγωνικό Σφάλμα 3 (Mean Square Error 3 – MSE 3).

Το καθένα από τα τρία σφάλματα προκύπτει από τη νόρμα της διαφοράς $y - \hat{y}$. Δηλαδή, $E = norm(y - \hat{y})$ ή αλλιώς:

$$E = \sqrt{\sum_i^m (y_i - \hat{y}_i)^2} \quad (3.20)$$

Ο προσεγγιστικός αλγόριθμος θα πρέπει να εξάγει όσο το δυνατόν μικρότερο σφάλμα εκτίμησης του χρόνου διαδρομής. Δηλαδή, τα βάρη των όρων της συνάρτησης εκτίμησης θα πρέπει να είναι τέτοια ώστε το σφάλμα να είναι το ελάχιστο δυνατόν. Η διαδικασία εύρεσης των βαρών και των τριών σφαλμάτων πραγματοποιήθηκε για τα τρία σενάρια των t προηγούμενων χρονικών στιγμών, όπως αναφέρθηκαν στην υποενότητα 3.3.2., όπου $t=50$, $t=100$ και $t=300$.

Ακολουθεί η μεθοδολογία που χρησιμοποιήθηκε για την εύρεση των βαρών w_i και τα σφάλματα που προέκυπταν σε κάθε δοκιμή για το σενάριο των $t=50$ προηγούμενων χρονικών στιγμών. Η διαδικασία εύρεσης των βαρών για τα υπόλοιπα σενάρια είναι η ίδια.

Αφού υπολογίστηκαν τα βάρη w_i (όπου $i = 1, 2, \dots, 14$) και βρέθηκαν τα τρία σφάλματα που προαναφέρθηκαν, πραγματοποιήθηκαν δοκιμές εύρεσης των σημαντικότερων βαρών w_i , εκείνων δηλαδή που επηρεάζουν περισσότερο τα σφάλματα MSE 2 και

MSE 3 και ειδικότερα του MSE 3. Το σφάλμα αυτό είναι το σημαντικότερο γιατί στην περίπτωση που είναι μικρό, αυτό σημαίνει ότι ο προσεγγιστικός αλγόριθμος που κατασκευάστηκε έχει μάθει από τα δεδομένα του δείγματος εκπαίδευσης (training set) και μπορεί να ανταποκριθεί αποτελεσματικά στα δεδομένα του δείγματος αξιολόγησης (test set).

Αρχικά, υπολογίστηκαν τα τρία σφάλματα χρησιμοποιώντας όλα τα βάρη. Τα σφάλματα που προέκυψαν, όπως φαίνεται και στον πίνακα 6 ήταν αντίστοιχα $1,9079 \times 10^4$, $2,5216 \times 10^4$ και $2,7965 \times 10^4$. Έπειτα, αφαιρώντας το μικρότερο από τα βάρη, w_2 , υπολογίστηκαν ξανά τα σφάλματα. Όπως παρατηρούμε στον πίνακα τα αποτελέσματα ήταν παρόμοια. Αυτό σημαίνει ότι ο συντελεστής που είχε το βάρος w_2 , δεν ήταν τόσο σημαντικός στον υπολογισμό του σφάλματος. Αφαιρώντας με όμοιο τρόπο τα βάρη w_i , προέκυψε ότι το μικρότερο σφάλμα RMS Error 3 προκύπτει για το συνδυασμό βαρών w_1 , w_5 , w_6 και w_{11} . Σε αυτήν την περίπτωση το σφάλμα λαμβάνει τιμή $2,7108 \times 10^{-4}$ και οι τιμές των βαρών, όπως παρουσιάζονται στον πίνακα 7, είναι ίσες με $w_1=0,04644$, $w_5=0,032625$, $w_6=0,017632$ και $w_{11}=0,047641$. Επίσης, στον ίδιο πίνακα παρουσιάζονται οι τιμές που λάμβαναν τα βάρη κάθε φορά που κάποιο αφαιρείτο. Τα βάρη που λείπουν σε κάθε συνδυασμό βαρών (σειρά) λαμβάνουν τιμές μηδέν και είναι αυτά που έχουν εξαιρεθεί από τον υπολογισμό της εκτιμήτριας συνάρτησης. Στις τρεις τελευταίες γραμμές του πίνακα 6 παρουσιάζονται τρεις διαφορετικοί συνδυασμοί βαρών που δοκιμάστηκαν, για να επαληθευτεί ότι ο συνδυασμός βαρών w_1 , w_5 , w_6 και w_{11} είναι όντως αυτός που δίνει τα μικρότερα σφάλματα.

Πίνακας 6. Συνδυασμοί βαρών της συνάρτησης \hat{y} και αντίστοιχα σφάλματα

Περιπτώσεις γαμμα=0,0000000000001	RMS Error	RMS Error 2	RMS Error 3
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1,9079	2,5216	2,7965
1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1,9079	2,5218	2,7965
1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	1,9079	2,5216	2,7966
1, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14	1,9079	2,5216	2,797
1, 5, 6, 11	2,4272	2,4271	2,7108
1, 5, 11	2,4874	2,4873	2,7686
1, 11	2,5078	2,5078	2,7879
1	44.444	4,4444	5,1414
1, 4, 11	2,5071	2,5071	2,7873
1, 11, 14	2,4867	2,4867	2,7771
1, 5, 6	2,8763	2,8762	3,2062

Πίνακας 7. Οι τιμές που λάμβαναν τα βάρη όταν κάθε φορά αφαιρείτο κάποιο από την εκτιμήτρια συνάρτηση

w1	0,02075700	0,020757	0,020757	0,020757	0,04644	0,067046	0,068296	0,2555
w2	0,00004406	0	0	0	0	0	0	0
w3	0,00018551	0,00018551	0	0	0	0	0	0
w4	0,00029877	0,00029877	0,0002988	0,00029881	0	0	0	0
w5	0,01480300	0,014803	0,014803	0,014803	0,032625	0,034941	0	0
w6	0,05515800	0,055158	0,055158	0,055158	0,017632	0	0	0
w7	0,00367110	0,0036711	0,003671	0,0036709	0	0	0	0
w8	0,00020138	0,00020138	0,00020141	0	0	0	0	0
w9	0,00040479	0,00040479	0,00040481	0,00040483	0	0	0	0
w10	0,00384610	0,0038461	0,0038469	0,0038476	0	0	0	0
w11	0,05187500	0,051875	0,051875	0,051875	0,047641	0,065679	0,066368	0
w12	0,00603690	0,006037	0,0060377	0,0060385	0	0	0	0
w13	-0,00071545	-0,00071545	-0,00071544	-0,00071544	0	0	0	0
w14	0,00199360	0,0019936	0,0019936	0,0019937	0	0	0	0

Η ίδια διαδικασία ακολουθήθηκε και για τις περιπτώσεις των $t=100$ αλλά και $t=300$ προηγούμενων χρονικών στιγμών. Στο επόμενο κεφάλαιο παρουσιάζονται τα αποτελέσματα που προέκυψαν για τα σφάλματα, μετά από την διαδικασία επιλογής των βαρών κάθε περίπτωσης. Οι τρεις αυτές περιπτώσεις των t προηγούμενων στιγμών

συγκρίνονται μεταξύ τους και κατασκευάζεται ένας βελτιστοποιημένος προσεγγιστικός αλγόριθμος για την καλύτερη περίπτωση.

4 ΑΠΟΤΕΛΕΣΜΑΤΑ

4.1 Εισαγωγή

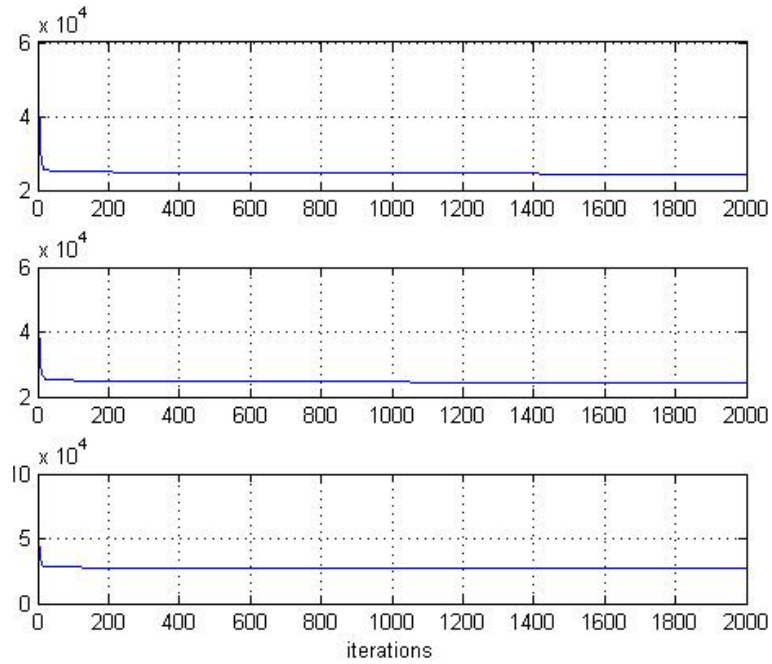
Στο κεφάλαιο 4 παρουσιάζονται τα αποτελέσματα που προέκυψαν σχετικά με την εκτίμηση του χρόνου διαδρομής των οχημάτων σε τρία τμήματα του υπό μελέτη δρόμου. Πιο συγκεκριμένα, από την αρχή του δρόμου έως και το πρώτο φωτεινό σηματοδότη, από το δεύτερο φωτεινό σηματοδότη έως και τον τρίτο και τέλος από τον τρίτο έως και τον τέταρτο.

Αρχικά, παρουσιάζονται τα τρία σφάλματα Mean Square Error (MSE), Mean Square Error 2 (MSE 2) και Mean Square Error 3 (MSE 3) που προέκυψαν για τρία διαφορετικά σενάρια t προηγούμενων χρονικών στιγμών. Το πρώτο σενάριο είναι για $t=50$, το δεύτερο για $t=100$ και το τρίτο για $t=300$. Τα σφάλματα συγκρίνονται μεταξύ τους και επιλέγεται το σενάριο όπου εμφανίζεται το μικρότερο σφάλμα.

Στη συνέχεια, παρουσιάζονται οι γραφικές παραστάσεις των εκτιμώμενων χρόνων \hat{y} και των πραγματικών χρόνων διαδρομής y για το σενάριο που επιλέχθηκε πρωτύτερα. Τα δεδομένα χωρίζονται στις τέσσερις περιπτώσεις χρόνου πράσινου/κόκκινου, οι οποίες παρουσιάστηκαν στο δεύτερο κεφάλαιο, και δίνονται οι γραφικές παραστάσεις των εκτιμώμενων χρόνων \hat{y} και των πραγματικών χρόνων διαδρομής y . Εν κατακλείδι, επιλέγεται η περίπτωση για την οποία ο προσεγγιστικός αλγόριθμος που κατασκευάστηκε δίνει τα καλύτερα αποτελέσματα.

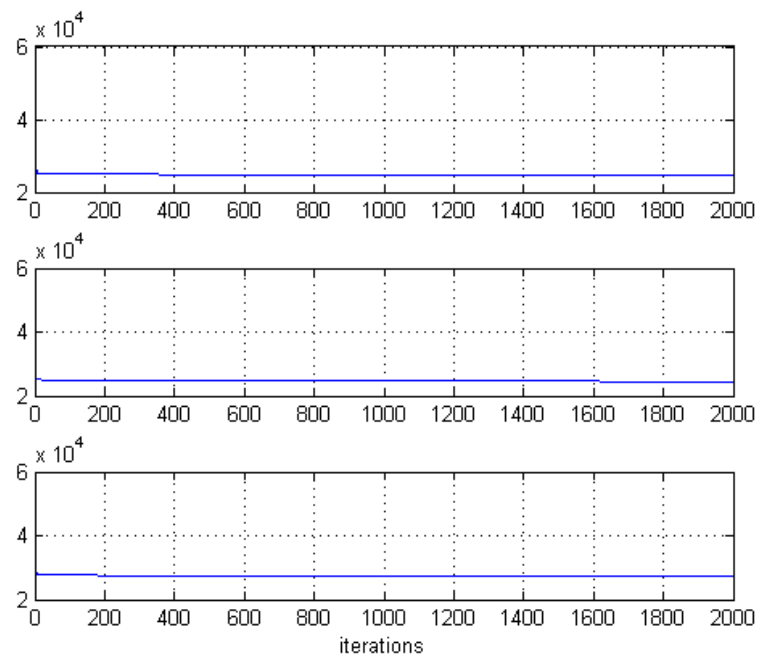
4.2 Συγκριτικά αποτελέσματα για τα σφάλματα

Για $t=50$, οι γραφικές παραστάσεις των τριών σφαλμάτων για 2000 επαναλήψεις, απεικονίζονται στο σχήμα 9. Στην πρώτη γραφική απεικονίζεται το MSE, στη δεύτερη το MSE 2 και στην τρίτη το MSE 3.



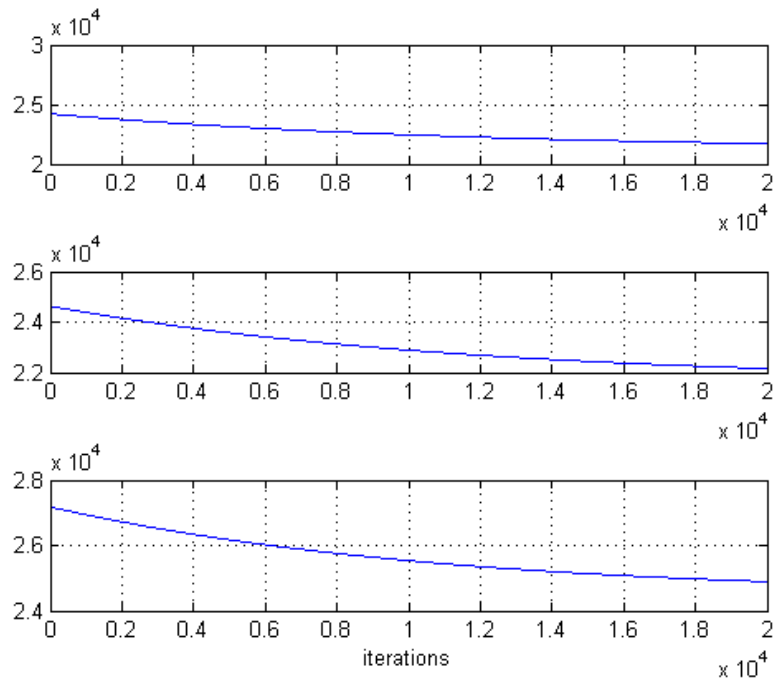
Σχήμα 9. Σφάλματα για $t=50$ προηγούμενες χρονικές στιγμές

Για $t=100$, οι γραφικές παραστάσεις των τριών σφαλμάτων για δύο χιλιάδες επαναλήψεις απεικονίζονται στο σχήμα 10. Στην πρώτη γραφική απεικονίζεται το MSE, στη δεύτερη το MSE 2 και στην τρίτη το MSE 3.



Σχήμα 10. Γραφικές παραστάσεις των τριών σφαλμάτων για $t=100$ προηγούμενες χρονικές στιγμές

Για $t=300$, οι γραφικές παραστάσεις των τριών σφαλμάτων για είκοσι χιλιάδες επαναλήψεις απεικονίζονται στο σχήμα 11. Στην πρώτη γραφική απεικονίζεται το MSE, στη δεύτερη το MSE 2 και στην τρίτη το MSE 3.



Σχήμα 11. Γραφικές παραστάσεις των τριών σφαλμάτων για $t=300$ προηγούμενες χρονικές στιγμές

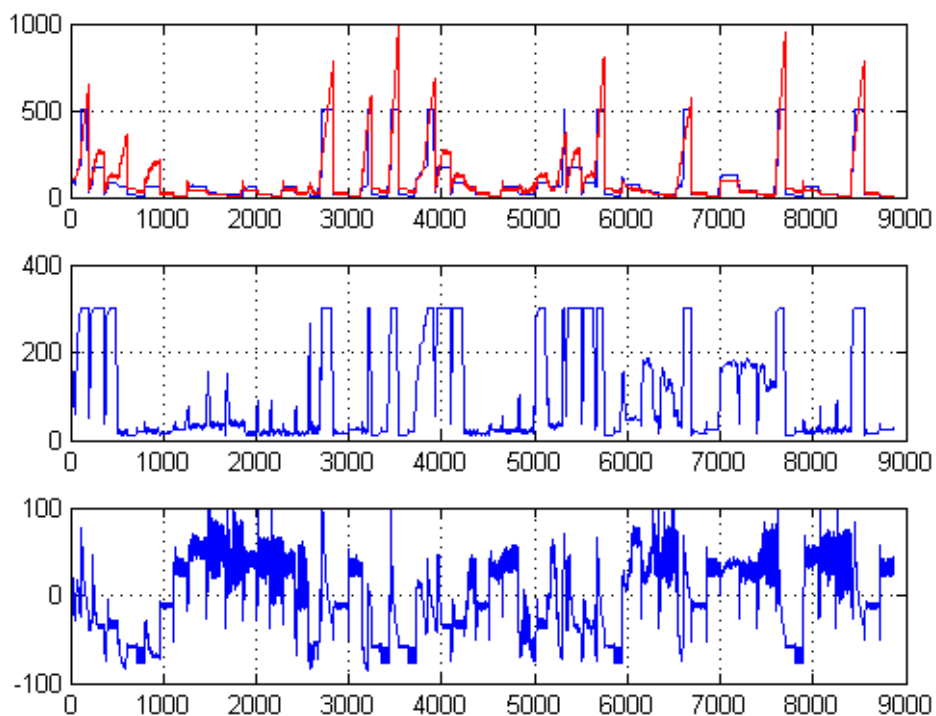
Συγκρίνοντας τα αποτελέσματα παρατηρούμε ότι τα σφάλματα για $t=300$ είναι μικρότερα από τα τρία σενάρια που επιλέχθηκαν. Συνεπώς, επιλέγεται το σενάριο $t=300$ για την εκτίμηση του χρόνου που χρειάζονται τα οχήματα να διασχίσουν το καθένα από τα τρία τμήματα του δρόμου.

Πίνακας 8. Συγκριτικός πίνακας των σφαλμάτων για τα τρία διαφορετικά σενάρια

t	MSE ($\times 10^4$)	MSE 2 ($\times 10^4$)	MSE 3 ($\times 10^4$)
-50	2,4272	2,4271	2,7108
-100	2,4464	2,4465	2,7143
-300	2,1716	2,2159	2,4894

4.3 Εκτιμώμενος χρόνος

Ο εκτιμώμενος χρόνος που υπολογίζεται είναι για την περίπτωση $t=300$. Οι εκτιμήσεις των υπολοίπων περιπτώσεων μπορούν να βρεθούν στο παράρτημα Β της παρούσης εργασίας. Στο σχήμα 12 παρατηρούμε ότι υπάρχουν τρεις γραφικές συναρτήσεις. Στην πρώτη απεικονίζεται ο εκτιμώμενος χρόνος \hat{y} και ο χρόνος y . Με το κόκκινο χρώμα απεικονίζεται ο πραγματικός χρόνος y . Στη δεύτερη γραφική παράσταση απεικονίζονται οι τιμές της μεταβλητής occupancy sum και στην τρίτη γραφική παράσταση απεικονίζεται το ποσοστιαίο (%) λάθος εκτίμησης.



Σχήμα 12. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης

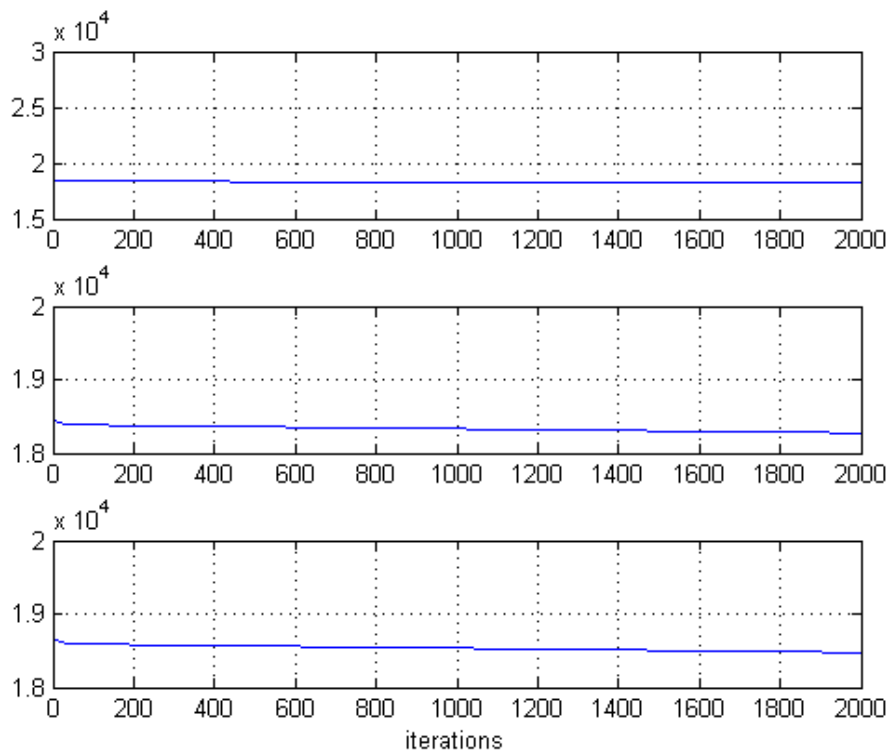
Παρατηρούμε ότι η εκτίμηση του χρόνου δεν είναι πολύ καλή για περιπτώσεις όπου οι τιμές της μεταβλητής occupancy sum είναι πολύ μικρές και πολύ μεγάλες. Πιο

συγκεκριμένα, η εκτίμηση δεν είναι πολύ καλή για τιμές της μεταβλητής occupancy sum μικρότερες του 5 και σε μερικές περιπτώσεις για τιμές μεγαλύτερες του 40.

Στη συνέχεια θα παρουσιαστεί ο εκτιμώμενος χρόνος για κάθε περίπτωση πράσινου/κόκκινου φαναριού των φωτεινών σηματοδοτών, όπως αναπτύχθηκαν στο κεφάλαιο 3.

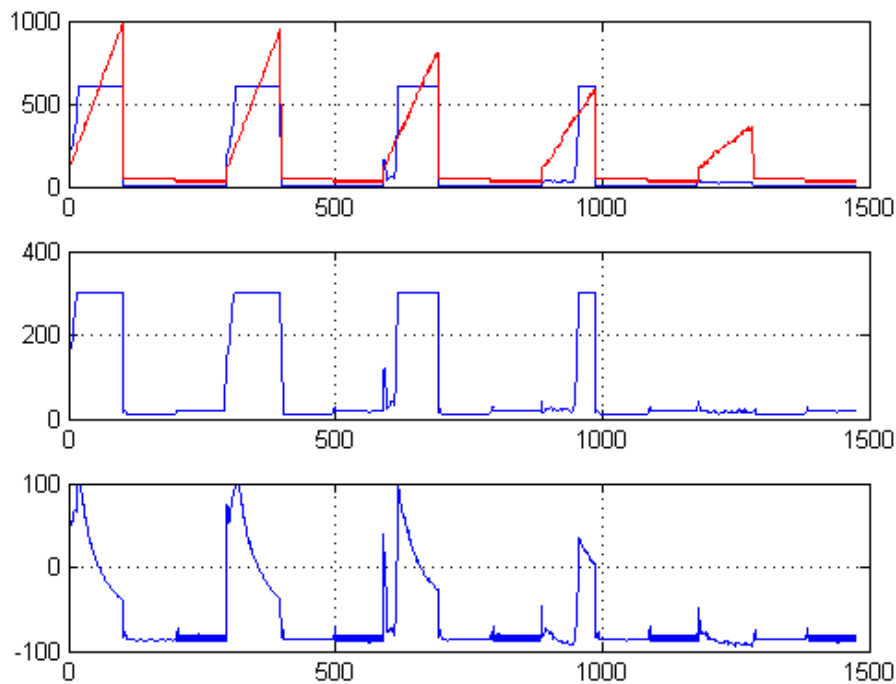
4.3.1 Περίπτωση Α

Η περίπτωση Α είναι εκείνη όπου δημιουργείται μποτιλιάρισμα στο δρόμο. Ο χρόνος που διαρκεί το κόκκινο φανάρι στους τρεις φωτεινούς σηματοδότες είναι ίδιος και ίσος με 70. Ο χρόνος που διαρκεί το πράσινο φανάρι είναι 70. Στο σχήμα 13 απεικονίζονται οι γραφικές παραστάσεις των σφαλμάτων της πρώτης μόνο περίπτωσης. Ο πίνακας των δεδομένων Α έχει διαστάσεις Α(3000, 5). Το training set των δεδομένων είναι οι πρώτες 2500 γραμμές και αντίστοιχα το test set οι 500 τελευταίες.



Σχήμα 13. Γραφικές παραστάσεις των σφαλμάτων για την πρώτη μόνο περίπτωση

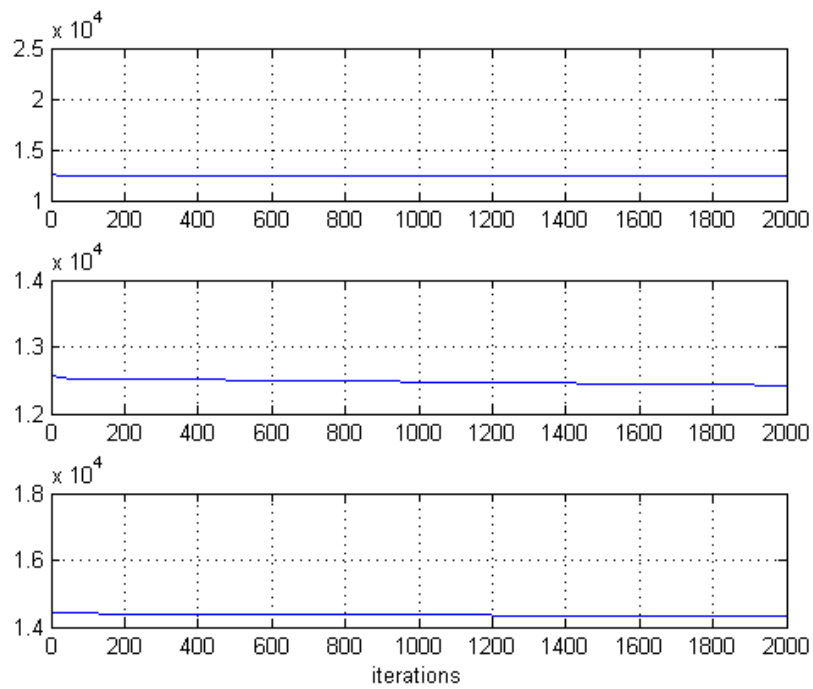
Στο σχήμα 14 απεικονίζονται οι γραφικές παραστάσεις του εκτιμώμενου χρόνου \hat{y} και του χρόνου που έχει πραγματικού χρόνου y , η μεταβλητή occupancy sum και το ποσοστιαίο (%) λάθος εκτίμησης αντίστοιχα. Στην πρώτη γραφική παράσταση με κόκκινο χρώμα απεικονίζεται ο πραγματικός χρόνος y .



Σχήμα 14. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης

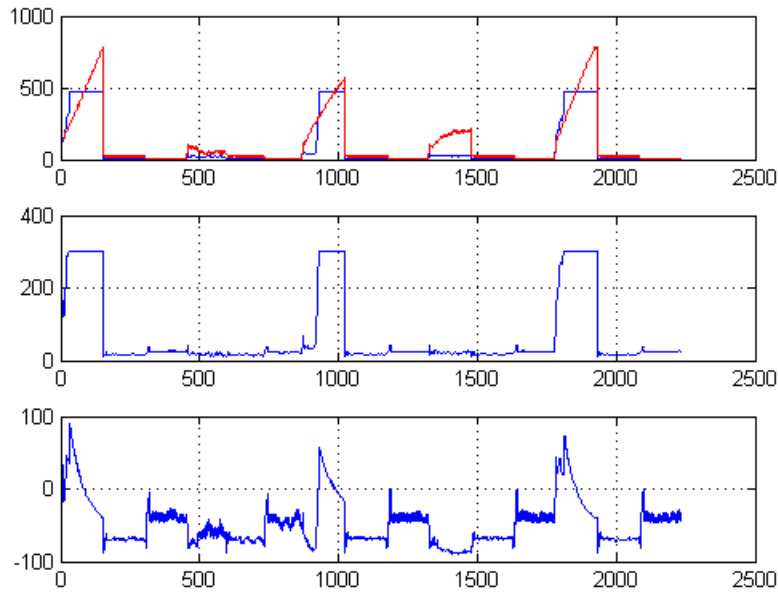
4.3.2 Περίπτωση Β

Στην περίπτωση Β οι χρόνοι που διαρκεί το πράσινο φανάρι αυξάνουν από το πρώτο μέχρι και το τρίτο φανάρι. Πιο συγκεκριμένα, ο χρόνος που διαρκεί το πράσινο φανάρι στον πρώτο φωτεινό σηματοδότη είναι ίσος με 30 και ο χρόνος που διαρκεί το κόκκινο ίσος με 60. Ο χρόνος που διαρκεί το πράσινο φανάρι στο δεύτερο και τον τρίτο φωτεινό σηματοδότη είναι 40 και 70 αντίστοιχα. Οπότε, ο χρόνος που διαρκεί το κόκκινο φανάρι είναι 50 και 20 αντίστοιχα. Ο πίνακας των δεδομένων Α έχει και σε αυτήν την περίπτωση διαστάσεις Α(3000, 5). Το training set των δεδομένων είναι οι πρώτες 2500 γραμμές και αντίστοιχα το test set οι 500 τελευταίες.



Σχήμα 15. Γραφικές παραστάσεις των σφαλμάτων για τη δεύτερη μόνο περίπτωση

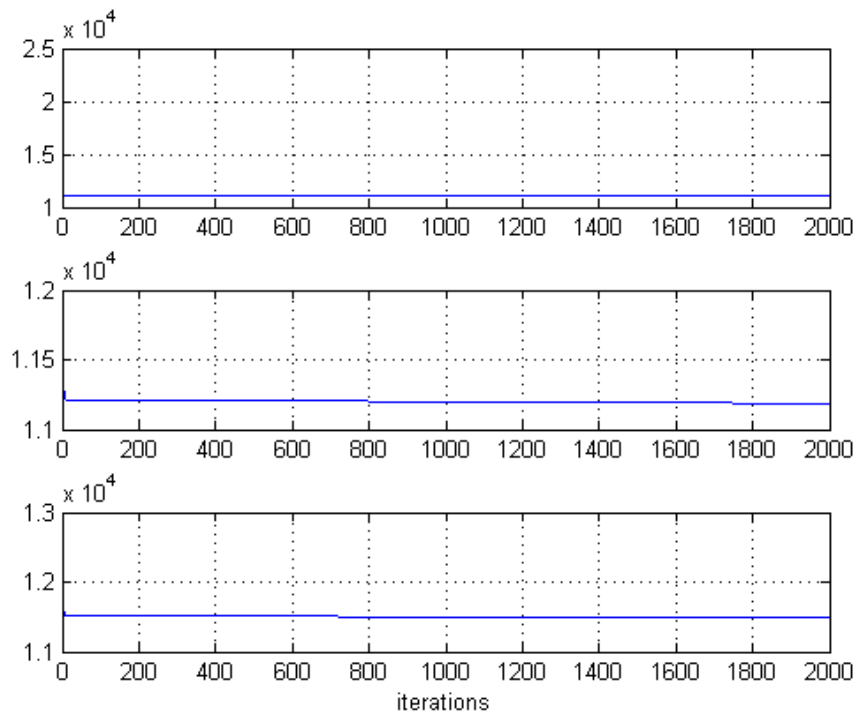
Στο σχήμα 16 απεικονίζονται οι γραφικές παραστάσεις του εκτιμώμενου χρόνου \hat{y} και του χρόνου που έχει πραγματικού χρόνου y , η μεταβλητή occupancy sum και το ποσοστιαίο (%) λάθος εκτίμησης αντίστοιχα. Στην πρώτη γραφική παράσταση με κόκκινο χρώμα απεικονίζεται ο πραγματικός χρόνος y .



Σχήμα 16. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης

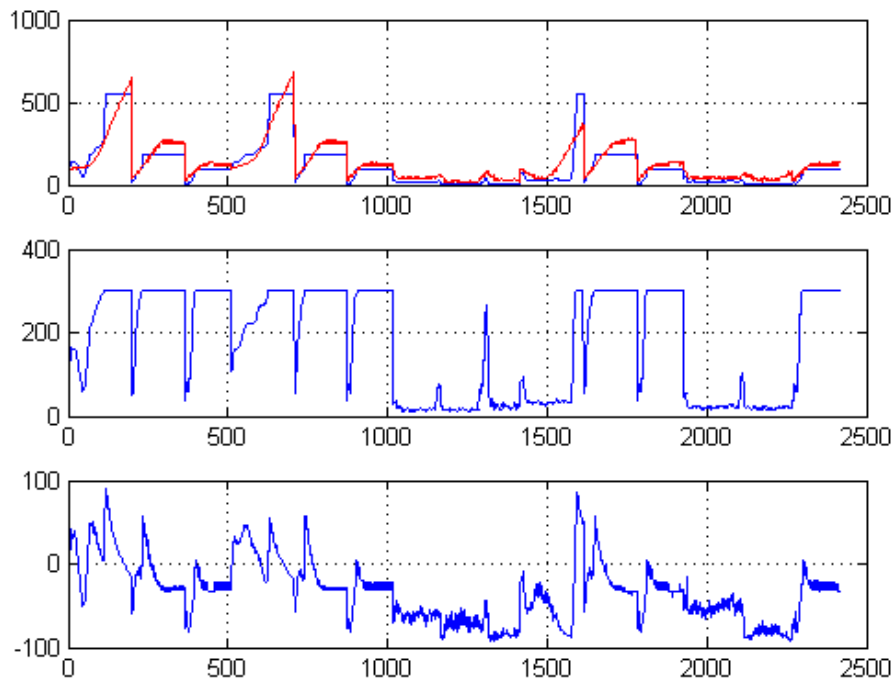
4.3.3 Περίπτωση Γ

Η περίπτωση Γ είναι το αντίστροφο της περίπτωσης Β. Οι χρόνοι που διαρκεί το πράσινο φανάρι στους τρεις φωτεινούς σηματοδότες μικραίνουν σταδιακά από τον πρώτο έως και τον τρίτο φωτεινό σηματοδότη. Πιο συγκεκριμένα, ο χρόνος που διαρκεί το πράσινο φανάρι στον πρώτο φωτεινό σηματοδότη είναι ίσος με 70 και ο χρόνος που διαρκεί το κόκκινο ίσος με 20. Ο χρόνος που διαρκεί το πράσινο φανάρι στο δεύτερο και τον τρίτο φωτεινό σηματοδότη είναι 40 και 30 αντίστοιχα. Οπότε, ο χρόνος που διαρκεί το κόκκινο φανάρι είναι 50 και 60 αντίστοιχα. Ο πίνακας των δεδομένων Α έχει και σε αυτήν την περίπτωση διαστάσεις $A(3000, 5)$. Το training set των δεδομένων είναι οι πρώτες 2500 γραμμές και αντίστοιχα το test set οι 500 τελευταίες.



Σχήμα 17. Γραφικές παραστάσεις των σφαλμάτων για την τρίτη μόνο περίπτωση

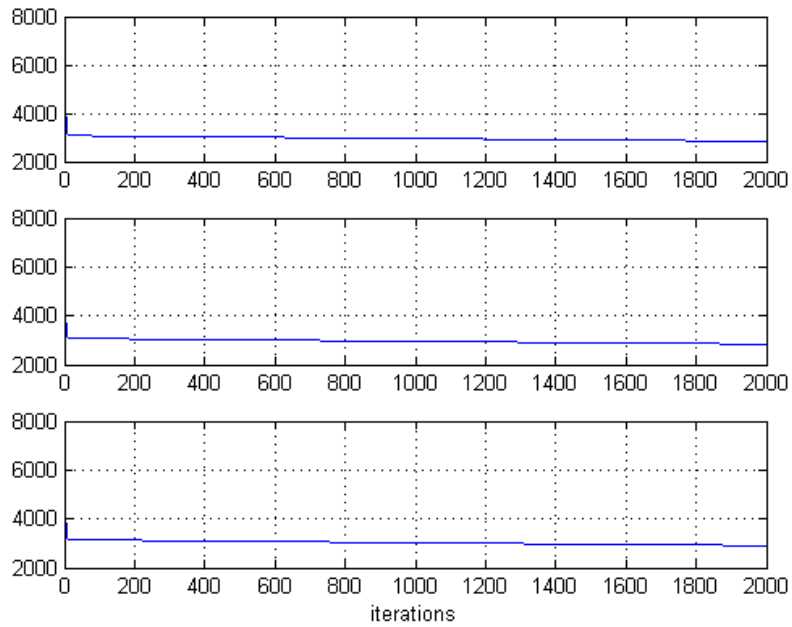
Στο σχήμα 18 απεικονίζονται οι γραφικές παραστάσεις του εκτιμώμενου χρόνου \hat{y} και του χρόνου που έχει πραγματικού χρόνου y , η μεταβλητή occupancy sum και το ποσοστιαίο (%) λάθος εκτίμησης αντίστοιχα. Στην πρώτη γραφική παράσταση με κόκκινο χρώμα απεικονίζεται ο πραγματικός χρόνος y .



Σχήμα 18. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης

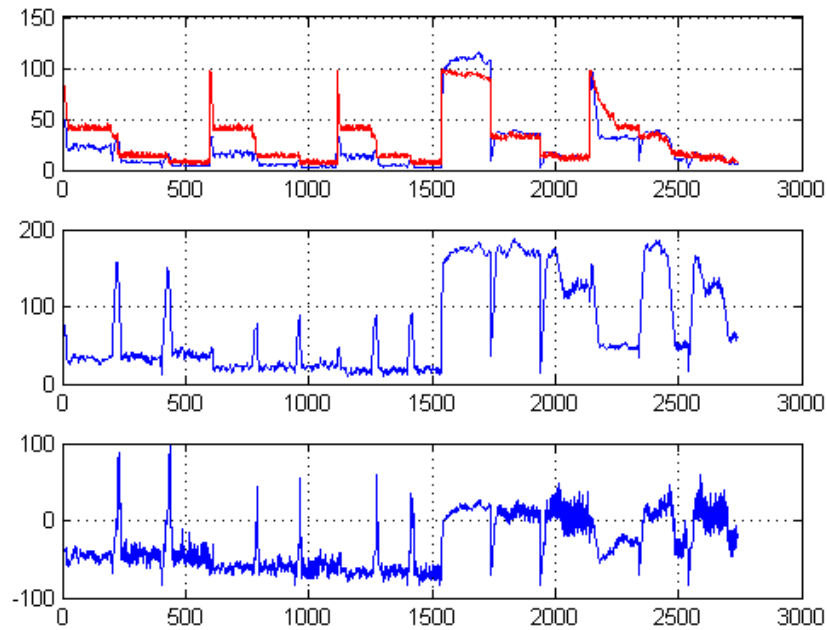
4.3.4 Περίπτωση Δ

Στην περίπτωση Δ, ο χρόνος που διαρκεί το πράσινο φανάρι είναι πολύ μεγαλύτερος από το χρόνο που διαρκεί το κόκκινο και ίδιος και για τους τρεις φωτεινούς σηματοδότες. Πιο συγκεκριμένα, ο χρόνος που διαρκεί το πράσινο φανάρι και στους τρεις φωτεινούς σηματοδότες είναι ίσος με 80 και άρα ο χρόνος του κόκκινου είναι ίσος με 10. Έτσι, η κυκλοφοριακή ροή γίνεται όσο το δυνατόν γρηγορότερη αφού τα οχήματα δεν αναγκάζονται να σταματούν στο υπό μελέτη τμήμα του δρόμου. Ο πίνακας των δεδομένων A έχει και σε αυτήν την περίπτωση διαστάσεις $A(3000, 5)$. Το training set των δεδομένων είναι οι πρώτες 2500 γραμμές και αντίστοιχα το test set οι 500 τελευταίες.



Σχήμα 19. Γραφικές παραστάσεις των σφαλμάτων για την τέταρτη μόνο περίπτωση

Στο σχήμα 20 απεικονίζονται οι γραφικές παραστάσεις του εκτιμώμενου χρόνου \hat{y} και του χρόνου που έχει πραγματικού χρόνου y , η μεταβλητή occupancy sum και το ποσοστιαίο (%) λάθος εκτίμησης αντίστοιχα. Στην πρώτη γραφική παράσταση με κόκκινο χρώμα απεικονίζεται ο πραγματικός χρόνος y .



Σχήμα 20. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης

4.3.5 Σύγκριση των τεσσάρων περιπτώσεων

Αναφορικά με τα σφάλματα που προέκυψαν σε κάθε περίπτωση, παρατηρούμε μέσα από τον συγκριτικό πίνακα 9 ότι το μικρότερο σφάλμα εμφανίζεται στην περίπτωση Δ όπου τα οχήματα σταματούν για λιγότερο χρόνο στους φωτεινούς σηματοδότες από τις άλλες περιπτώσεις. Ο χρόνος πρασίνου των φωτεινών σηματοδοτών σε αυτήν την περίπτωση είναι 80 και ο χρόνος του κόκκινου είναι 10.

Πίνακας 9. Συγκριτικός πίνακας σφαλμάτων για τις τέσσερις περιπτώσεις

Περιπτώσεις	MSE ($\times 10^4$)	MSE 2 ($\times 10^4$)	MSE 3 ($\times 10^4$)
A	1,8245	1,8274	1,8475
B	1,2363	1,2426	1,4317
Γ	1,1058	1,1187	1,1487
Δ	0,2841	0,2842	0,2909

4.4 Βελτιωμένη έκδοση προσεγγιστικού αλγορίθμου

Συγκρίνοντας τις τέσσερις περιπτώσεις διαφορετικών συνδυασμών χρόνων πράσινου και κόκκινου στους φωτεινούς σηματοδότες, προέκυψε ότι η περίπτωση Δ εμφανίζει τα μικρότερα σφάλματα. Επιδιώχθηκε η κατασκευή ενός βελτιωμένου προσεγγιστικού αλγορίθμου ούτως ώστε να προκύψουν ακόμη μικρότερα σφάλματα.

Παρατηρήθηκε ότι η μεταβλητή occupancy sum είναι αυτή που ευθύνεται για τα σφάλματα που προκύπτουν και έτσι επιλέχθηκε ο προσεγγιστικός αλγόριθμος να χρησιμοποιήσει αρχικά τα δεδομένα του πίνακα A για τα οποία η μεταβλητή παίρνει τιμές μικρότερες του 60 και έπειτα για τα δεδομένα για τα οποία η μεταβλητή παίρνει τιμές μεγαλύτερες και ίσες του 60.

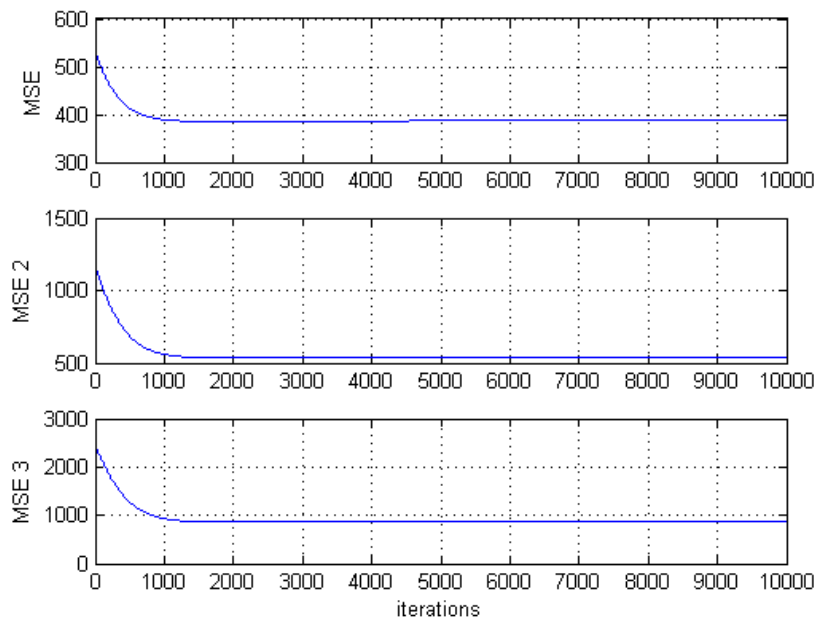
Ακολουθήθηκε η ίδια μεθοδολογία, όπως και στις τέσσερις περιπτώσεις A, B, Γ και Δ. Αρχικά, υπολογίστηκαν όλα τα βάρη της συνάρτησης του εκτιμώμενου χρόνου \hat{y} . Έπειτα, αφαιρέθηκε το βάρος w_i με την μικρότερη τιμή, το οποίο επηρεάζει λιγότερο από τα υπόλοιπα την εκτίμηση του χρόνου. Ακολουθώντας αυτήν την επαναληπτική διαδικασία και ελέγχοντας κάθε φορά τις τιμές των σφαλμάτων που προκύπτουν, καταλήγουμε στην ομάδα βαρών w_i , όπου $i = 7, 10, 11, 13, 14$.

Πιο αναλυτικά στο πίνακα 10 παρατηρούμε τις τιμές που λάμβαναν τα βάρη w_i κάθε φορά. Πάντα για την περίπτωση όπου οι τιμές της μεταβλητής occupancy sum είναι μικρότερες του 60.

Πίνακας 10. Οι τιμές που λάμβαναν τα βάρη όταν κάθε φορά αφαιρείτο κάποιο από την εκτιμήτρια συνάρτηση

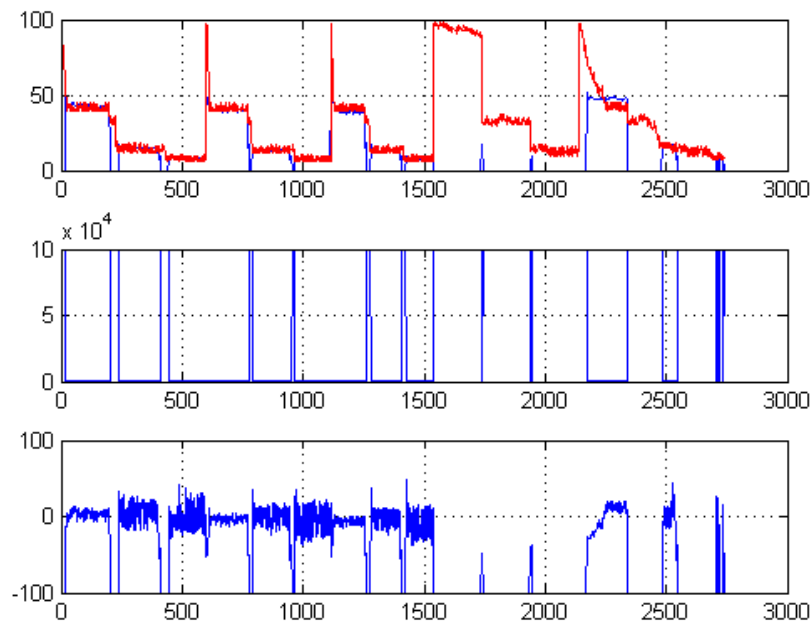
	$\times 10^{-6}$									
w1	27,728	27,728	27,728	27,729	27,729	27,729	27,729	27,729	27,73	0
w2	0,0147	0	0	0	0	0	0	0	0	0
w3	0,7353	0,7353	0,7353	0,7354	0	0	0	0	0	0
w4	6,1265	6,1265	6,1265	6,1266	6,1266	6,1267	6,1267	0	0	0
w5	5,2188	5,2188	5,2188	5,2189	5,2189	5,2189	0	0	0	0
w6	0,626	0,626	0,626	0	0	0	0	0	0	0
w7	2218,3	2218,3	2218,3	2218,3	2218,3	2218,3	2218,3	2218,3	2218,4	2218,6
w8	0,2894	0,2894	0	0	0	0	0	0	0	0
w9	1,1753	1,1753	1,1753	1,1753	1,1753	0	0	0	0	0
w10	58,823	58,823	58,823	58,831	58,831	58,831	58,831	58,837	58,878	58,875
w11	81,825	81,825	81,825	81,828	81,828	81,828	81,828	81,828	81,829	81,832
w12	23,148	23,148	23,148	23,149	23,149	23,149	23,149	23,149	0	0
w13	50,076	50,076	50,076	50,083	50,083	50,083	50,083	50,083	50,089	50,095
w14	417,51	417,51	417,51	417,51	417,51	417,51	417,52	417,52	417,52	417,55

Στο σχήμα 21 παρατηρούμε τα σφάλματα που προέκυψαν για τιμές της μεταβλητής occupancy sum μικρότερες του 60, για 10000 επαναλήψεις και για σύνολο βαρών w_i , όπου $i = 7,10,11,13,14$.



Σχήμα 21. Γραφικές παραστάσεις των σφαλμάτων για occupancy sum ίσο και μικρότερο του 60

Στο σχήμα 22 απεικονίζονται οι γραφικές παραστάσεις του εκτιμώμενου χρόνου \hat{y} και του χρόνου που έχει πραγματικού χρόνου y , η μεταβλητή occupancy sum και το ποσοστιαίο (%) λάθος εκτίμησης αντίστοιχα. Στην πρώτη γραφική παράσταση με κόκκινο χρώμα απεικονίζεται ο πραγματικός χρόνος y .



Σχήμα 22. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης

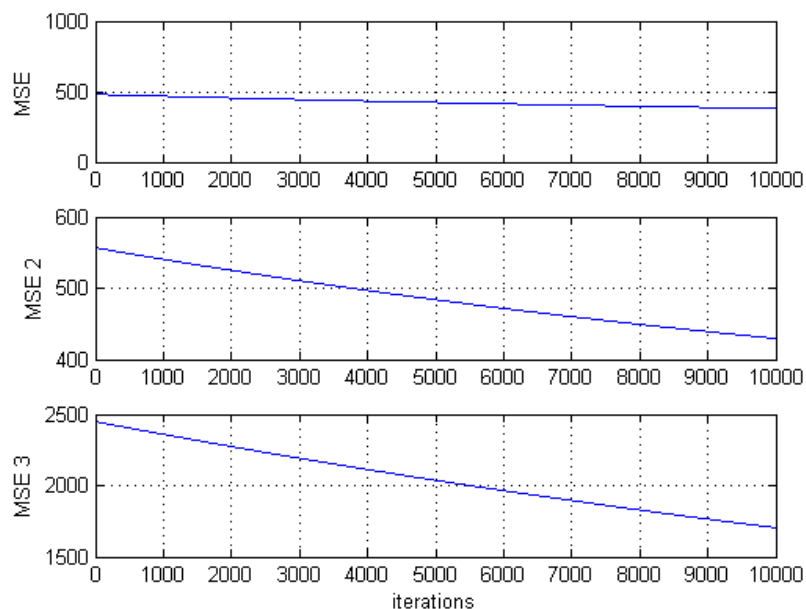
Ομοίως, για την περίπτωση όπου η μεταβλητή occupancy sum παίρνει τιμές μεγαλύτερες και ίσες της τιμής 60, ακολουθώντας την ίδια επαναληπτική διαδικασία και ελέγχοντας κάθε φορά τις τιμές των σφαλμάτων που προκύπτουν, καταλήγουμε στην ομάδα βαρών w_i , όπου $i = 7, 10, 13$.

Πιο αναλυτικά στο πίνακα 11 παρατηρούμε τις τιμές που λάμβαναν τα βάρη w_i κάθε φορά.

Πίνακας 11. Οι τιμές που λάμβαναν τα βάρη όταν κάθε φορά αφαιρείτο κάποιο από την εκτιμήτρια συνάρτηση

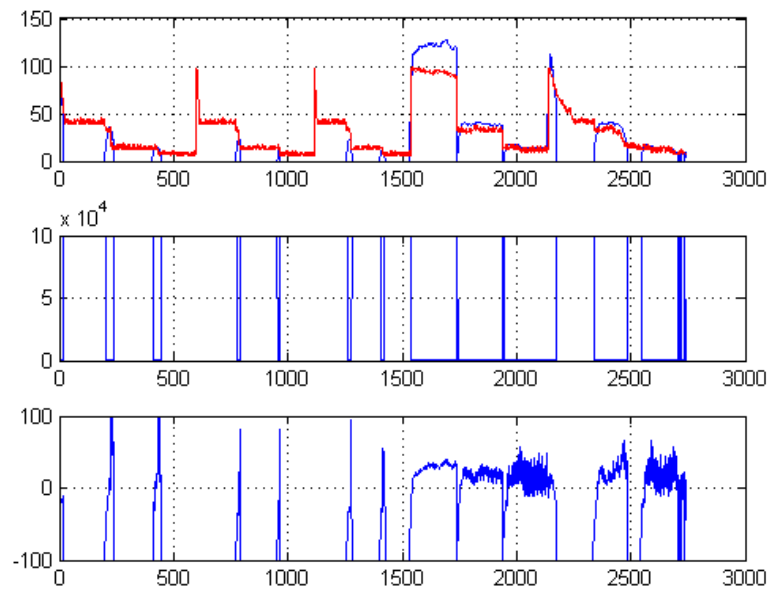
	$\times 10^{-6}$											
w1	2,8669	2,8669	2,8669	2,8669	2,8669	2,8669	2,867	2,867	2,867	0	0	0
w2	0,0004	0	0	0	0	0	0	0	0	0	0	0
w3	-1,106	-1,106	-1,106	-1,106	-1,106	-1,106	-1,1061	-1,1061	0	0	0	0
w4	-0,06	-0,06	-0,06	-0,06	0	0	0	0	0	0	0	0
w5	0,2629	0,2629	0,2629	0,2629	0,2629	0	0	0	0	0	0	0
w6	0,7792	0,7792	0,7792	0,7792	0,7792	0,7792	0	0	0	0	0	0
w7	229,35	229,35	229,35	229,35	229,35	229,35	229,36	229,36	229,36	229,36	229,37	229,39
w8	-0,0107	-0,0107	0	0	0	0	0	0	0	0	0	0
w9	0,0339	0,0339	0,0339	0	0	0	0	0	0	0	0	0
w10	-88,482	-88,482	-88,482	-88,482	-88,482	-88,482	-88,491	-88,491	-88,491	-88,491	-88,493	-88,501
w11	10,398	10,398	10,398	10,398	10,398	10,398	10,398	10,398	10,398	10,398	0	0
w12	-0,854	-0,854	-0,854	-0,854	-0,854	-0,854	-0,8543	0	0	0	0	0
w13	62,332	62,332	62,332	62,332	62,332	62,332	62,343	62,343	62,343	62,343	62,349	62,359
w14	21,032	21,032	21,032	21,032	21,032	21,032	21,033	21,033	21,033	21,033	21,034	0

Στο σχήμα 23 παρατηρούμε τα σφάλματα που προέκυψαν για τιμές της μεταβλητής occupancy sum μικρότερες του 60, για 10000 επαναλήψεις και για σύνολο βαρών w_i , όπου $i = 7,10,13$.



Σχήμα 23. Γραφικές παραστάσεις των σφαλμάτων για occupancy sum μεγαλύτερο και ίσο του 60

Στο σχήμα 24 απεικονίζονται οι γραφικές παραστάσεις του εκτιμώμενου χρόνου \hat{y} και του χρόνου που έχει πραγματικού χρόνου y , η μεταβλητή occupancy sum και το ποσοστιαίο (%) λάθος εκτίμησης αντίστοιχα. Στην πρώτη γραφική παράσταση με κόκκινο χρώμα απεικονίζεται ο πραγματικός χρόνος y .



Σχήμα 24. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης

Στον πίνακα 12 απεικονίζονται τα σφάλματα που προέκυψαν μετά το βελτιστοποιημένο προσεγγιστικό αλγόριθμο και πριν αυτόν.

Πίνακας 12. Συγκριτικός πίνακας σφαλμάτων, όπως προέκυψαν με τον βελτιστοποιημένο προσεγγιστικό αλγόριθμο και χωρίς αυτόν

Περίπτωση	MSE ($\times 10^4$)	MSE 2 ($\times 10^4$)	MSE 3 ($\times 10^4$)
Occupancy sum < 60	0,0385	0,0535	0,0868
Occupancy sum > 60	0,03807	0,0429	0,1705
Δ	0,2841	0,2842	0,2909

5 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

5.1 Συμπεράσματα

Ανακεφαλαιώνεται ότι στην παρούσα εργασία κατασκευάστηκε ένας προσεγγιστικός αλγόριθμος με στόχο την εκτίμηση του χρόνου διαδρομής οχημάτων σε έναν εικονικό δρόμο μονής κυκλοφορίας όπου βρίσκονται εγκαταστημένοι τρεις φωτεινοί σηματοδότες.

Επιλέχθηκαν τέσσερα σενάρια διαφορετικών χρόνων πράσινου/κόκκινου για τον καθένα φωτεινό σηματοδότη, από τα οποία ο προσεγγιστικός αλγόριθμος παρουσίαζε καλύτερα αποτελέσματα για την περίπτωση Δ. Σε εκείνη την περίπτωση οι χρόνοι κόκκινου για τον κάθε φωτεινό σηματοδότη ήταν 10, και εκείνος του πράσινου ήταν 80, με άθροισμα στο 90.

Για την εξαγωγή ακόμα καλύτερων αποτελεσμάτων, αποφασίστηκε η βελτιστοποίηση του προσεγγιστικού αλγορίθμου. Παρατηρήθηκε ότι το σφάλμα το σφάλμα μεγάλωνε πολύ στην περίπτωση που η μεταβλητή “occupancy sum” λάμβανε τιμές μικρότερες του 3 και μεγαλύτερες του 40. Αποφασίστηκε να χωριστούν τα δεδομένα σε δύο κατηγορίες και να πραγματοποιηθεί η ίδια μεθοδολογία σε καθεμιά από αυτές. Η μια κατηγορία αφορούσε τα δεδομένα που η μεταβλητή “occupancy sum” λάμβανε τιμές μικρότερες και ίσες του 60 και η άλλη εκείνη που λάμβανε τιμές μεγαλύτερες του 60.

Μετά από την εφαρμογή της μεθοδολογίας στον προσεγγιστικό αλγόριθμο καταλήξαμε σε ένα σύνολο βαρών για κάθε περίπτωση για τα οποία τα αποτελέσματα που προκύπτουν είναι πολύ καλύτερα. Σημειώνεται η μεταβλητή “occupancy sum” αφορά το άθροισμα των τελευταίων $t=300$ μετρήσεων για κάθε όχημα, όπου t είναι το βήμα της προσομοίωσης.

5.2 Μελλοντικές επεκτάσεις

Όσον αφορά τις μελλοντικές ενέργειες που θα μπορούσαν να πραγματοποιηθούν στην παρούσα εργασία αυτές θα ήταν η κατασκευή ενός άλλου προσεγγιστικού αλγορίθμου που θα διορθώνει αυτήν ακριβώς την αύξηση του σφάλματος με ακόμη μεγαλύτερη επιτυχία από τον βελτιωμένο προσεγγιστικό αλγόριθμο που κατασκευάστηκε.

Επίσης, θα μπορούσε εάν κριθεί επιθυμητό, να χρησιμοποιηθεί μεγαλύτερη λεπτομέρεια στο μικροσκοπικό μοντέλο προσομοίωσης.

ΠΑΡΑΡΤΗΜΑ Α

Κώδικας

Μικροσκοπικό μοντέλο προσομοίωσης (sim2.m)

```

clear all
close all
N=1000;          %xronos prosomoiosis
Tsim=0.25;      %vima prosomoiosis
i=0;
g=2;
a_max=1.5;
a_min=-6;
L=0.7;
D=1;
sss=zeros(3,1);

Sensitivity_Detector=4;          % eyaisthisia tou detector

length_vehicle=5;                %mikos oximatos
x=zeros(1000,N);                 %arxikopoiisi pinaka theseon oximaton
y=5*rand(1000,N);                %arxikopoiisi pinaka taxititon oximaton
a=zeros(1000,N);                 %arxikopoiisi pinaka epitaxinseon oximaton
occ=zeros(N,3);                  % occupancy measurements
flow=zeros(N,3);                 % flow measurements
delta_min=zeros(1000,N);
ydi=16.5*ones(1000,N);           %arxikopoiisi pinaka epithimiton taxititon oximaton
t_last_entrance=0;
t_entrance=2;
alpha=1;

X_tf_1=600;                       %me ti sixnotita tha anavosvinei to proto fanari
g1=30;
r1=60;
det_loc(1,1)=X_tf_1-300;          %H THESI TOU PROTOU DETECTOR

X_tf_2=800;                       %me ti sixnotita tha anavosvinei to deutero fanari
g2=60;
r2=30;
det_loc(2,1)=X_tf_2-100;         %H THESI TOU DEUTEROU DETECTOR

X_tf_3=900;                       %me ti sixnotita tha anavosvinei to trito fanari
g3=80;
r3=10;
det_loc(3,1)=X_tf_3-50;          %H THESI TOU TRITOU DETECTOR
for t=2:1:N
    if rem(t,g1+r1)<=g1
        status1(t)=1;            %prasino
    else
        status1(t)=0;            %kokkino
    end

```

```

if rem(t,g2+r2)<=g2
    status2(t)=1;           %prasino
else
    status2(t)=0;           %kokkino
end
if rem(t,g3+r3)<=g3
    status3(t)=1;           %prasino
else
    status3(t)=0;           %kokkino
end

if i==0
    t_last_entrance=t;
    t_entrance=alpha*(round(5*rand(1,1))+1);
    i=i+1;
end

if (i>0) & ((t_last_entrance+t_entrance<=t) & (x(i,t-1)>length_vehicle+2*D))
    t_last_entrance=t;
    t_entrance=alpha*(round(5*rand(1,1))+1);
    i=i+1;
end

for k=1:i
    if (x(k,t-1) > X_tf_1-50) & (x(k,t-1) < X_tf_1) & (status1(t-1)==0)
        delta(k,t)=min(X_tf_1-x(k,t-1)-D,5);
    elseif (x(k,t-1) > X_tf_2-50) & (x(k,t-1) < X_tf_2) & (status2(t-1)==0)
        delta(k,t)=min(X_tf_2-x(k,t-1)-D,5);
    elseif (x(k,t-1) > X_tf_3-50) & (x(k,t-1) < X_tf_3) & (status3(t-1)==0)
        delta(k,t)=min(X_tf_3-x(k,t-1)-D,5);
    else
        if (k==1)
            delta(k,t)=10;
        else
            delta(k,t)=x(k-1,t-1)-x(k,t-1)-length_vehicle;
        end
    end
    if ((x(k,t-1) > X_tf_1-5) & (x(k,t-1) < X_tf_1) & (status1(t-1)==0)) %
        delta(k,t)=1010101;
    end
    if ((x(k,t-1) > X_tf_2-5) & (x(k,t-1) < X_tf_2) & (status2(t-1)==0))
        delta(k,t)=1010101;
    end
    if ((x(k,t-1) > X_tf_3-5) & (x(k,t-1) < X_tf_3) & (status3(t-1)==0))
        delta(k,t)=1010101;
    end
    if k>1 & (y(k-1,t)<0.1 & x(k-1,t-1)-x(k,t-1)-length_vehicle<1)
        delta(k,t)=1010101;
    end
    if delta(k,t)<length_vehicle-D
        delta(k,t)=length_vehicle-D;
    end

ydi(k,t-1)=L*(delta(k,t)-D);
if ydi(k,t-1)>16.5
    ydi(k,t-1)=16.5;
end
end

```

```

yd=ydi(k,t-1);
a(k,t-1)=g*(ydi(k,t-1)-y(k,t-1)); %Ipologismos epitaxisis oximatos
if a(k,t-1)>=a_max
    a(k,t-1)=a_max;
elseif a(k,t-1)<=a_min
    a(k,t-1)=a_min;
end
yy=y(k,t-1)+a(k,t-1)*Tsim; %Ipologismos taxititas oximatos
xx=x(k,t-1)+Tsim*y(k,t-1)+0.5*Tsim^2*a(k,t-1); %Ipologismos thesis oximatos
if delta(k,t)==1010101
    yy=0;
    xx=x(k,t-1);
end
y(k,t)=yy;
x(k,t)=xx;

% occupancy calculation
for ii=1:3
    if(min(size(find(abs(det_loc(ii,1)-x(:,t))<Sensitivity_Detector)))>0)
        occ(t,ii)=1;
    else
        occ(t,ii)=0;
    end
end

end

if x(k,t)>1000
    x(k,t)=1500;
    %out_of_road(k,:)=k;
    %out_of_road(:,t)=t
    y(k,t)=30;
    a(k,t)=3;

end

end
end

for t=2:1:N
    for ii=1:3
        if occ(t,ii)==1 & occ(t-1,ii)==0
            flow(t,ii)=1;
            sss(ii,1)=1;
        end
        if sss(ii,1)>0 & occ(t,ii)==1
            sss(ii,1)=sss(ii,1)+1;
        end
        if sss(ii,1)>8
            flow(t,ii)=1;
            sss(ii,1)=0;
        end
    end
end
end

plot(x')
hold
plot(100*status1)
hold

```

```
grid
```

```
figure
plot(occ(:,1))      %TIPONEI TIN OCCUPANCY GIA TON PROTO DETECTOR
hold;
plot(flow(:,1), 'r') %TIPONEI TO FLOW GIA TON PROTO DETECTOR (me kokkino xroma)
hold
axis([0 N 0 1.5])
grid
```

```
figure
plot(occ(:,2))      %TIPONEI TIN OCCUPANCY GIA TON DEUTERO DETECTOR
hold;
plot(flow(:,2), 'r') %TIPONEI TO FLOW GIA TON DEUTERO DETECTOR (me kokkino xroma)
hold
axis([0 N 0 1.5])
grid
```

```
figure
plot(occ(:,3))      %TIPONEI TIN OCCUPANCY GIA TON TRITO DETECTOR
hold;
plot(flow(:,3), 'r') %TIPONEI TO FLOW GIA TON TRITO DETECTOR (me kokkino xroma)
hold
axis([0 N 0 1.5])
grid
```

Συμπληρωματικοί υπολογισμοί (sim3.m)

```
clear all
close all
load alpha10-2070-2070-2070
N=5000;
time_pass=zeros(4,1000);      %arxikopoiisi pinaka time_pass
time_difference=zeros(3,1000); %arxikopoiisi pinaka xronikon diaforon
flow_sum=zeros(1000,3);
occ_sum=zeros(1000,3);
for i=1:1000,                  % i einai o arithmos ton oximatou
    for j=1:N,                  % j einai to vima tis prosomoiosis
        if time_pass(1,i)==0 & x(i,j)>0 %apothikevei ti xroniki stigmi pou mpainei to oxima sto dromo sti
            %proti grammi tou pinaka
                time_pass(1,i)=j;
            end
            if time_pass(2,i)==0 & x(i,j)>600 %sti deuteri grammi tou pinaka apothikevei ti xroniki stigmi pou to
            %oxima perna to proto fanari
                time_pass(2,i)=j;
            end
            if time_pass(3,i)==0 & x(i,j)>800 %sti triti grammi tou pinaka apothikevei ti xroniki stigmi pou to
            %oxima perna to deutero fanari
                time_pass(3,i)=j;
            end
            if time_pass(4,i)==0 & x(i,j)>900 %sti triti grammi tou pinaka apothikevei ti xroniki stigmi pou to
            %oxima perna to trito fanari
                time_pass(4,i)=j;
            end
        end
    end
end
```



```

    end
end

for i=1:1000
    time_difference(1,i)=time_pass(2,i)-time_pass(1,i); %ipologizontai oi xronikes diafores
    time_difference(2,i)=time_pass(3,i)-time_pass(2,i);
    time_difference(3,i)=time_pass(4,i)-time_pass(3,i);
end

for i=1:1000
    for link_id=1:3
        if time_pass(link_id+1,i)>50
            flow_sum(i,link_id)=sum(flow(time_pass(link_id+1,i)-50:time_pass(link_id+1,i),link_id));
            occ_sum(i,link_id)=sum(occ(time_pass(link_id+1,i)-50:time_pass(link_id+1,i),link_id));
        else
            flow_sum(i,link_id)=99999;
            occ_sum(i,link_id)=99999;
        end
    end
end
end
save alpha10-2070-2070-2070

```

Κατασκευή πίνακα δεδομένων A (calculate A c.m)

```

clear all
close all

length_1=600;
length_2=200;
length_3=100;
num_cars=200;

A1=zeros(num_cars*3,5);
A2=zeros(num_cars*3,5);
A3=zeros(num_cars*3,5);
A4=zeros(num_cars*3,5);
A5=zeros(num_cars*3,5);
A6=zeros(num_cars*3,5);
A7=zeros(num_cars*3,5);
A8=zeros(num_cars*3,5);
A9=zeros(num_cars*3,5);
A10=zeros(num_cars*3,5);
A11=zeros(num_cars*3,5);
A12=zeros(num_cars*3,5);
A13=zeros(num_cars*3,5);
A14=zeros(num_cars*3,5);
A15=zeros(num_cars*3,5);
A16=zeros(num_cars*3,5);
A17=zeros(num_cars*3,5);
A18=zeros(num_cars*3,5);
A19=zeros(num_cars*3,5);
A20=zeros(num_cars*3,5);

w=zeros(14,1);
gamma=0.00000000001;

```

```
load alpha1-2070-2070-2070
```

```
for k=1:num_cars
    A1(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A1(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A1(k,2)=99999;
    else
        A1(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A1(k,3)=occ_sum(k,1);
    else
        A1(k,3)=99999;
    end
    A1(k,4)=20;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A1(k,5)=time_difference(1,k);
    else
        A1(k,5)=99999;
    end
end

for k=num_cars+1:num_cars*2
    A1(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A1(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A1(k,2)=99999;
    else
        A1(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A1(k,3)=occ_sum(k-num_cars,2);
    else
        A1(k,3)=99999;
    end
    A1(k,4)=20;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A1(k,5)=time_difference(2,k-num_cars);
    else
        A1(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A1(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A1(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A1(k,2)=99999;
    else
        A1(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
```

```

    A1(k,3)=occ_sum((k-num_cars*2),3);
else
    A1(k,3)=99999;
end
A1(k,4)=20;
if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
    A1(k,5)=time_difference(3,k-(num_cars*2));
else
    A1(k,5)=99999;
end
end

load alpha3-2070-2070-2070

for k=1:num_cars
    A2(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A2(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A2(k,2)=99999;
    else
        A2(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A2(k,3)=occ_sum(k,1);
    else
        A2(k,3)=99999;
    end
    A2(k,4)=20;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A2(k,5)=time_difference(1,k);
    else
        A2(k,5)=99999;
    end
end
end

for k=num_cars+1:num_cars*2
    A2(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A2(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A2(k,2)=99999;
    else
        A2(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A2(k,3)=occ_sum(k-num_cars,2);
    else
        A2(k,3)=99999;
    end
    A2(k,4)=20;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A2(k,5)=time_difference(2,k-num_cars);
    else
        A2(k,5)=99999;
    end
end
end

```

```

for k=(num_cars*2)+1:num_cars*3
    A2(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A2(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A2(k,2)=99999;
    else
        A2(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A2(k,3)=occ_sum((k-num_cars*2),3);
    else
        A2(k,3)=99999;
    end
    A2(k,4)=20;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A2(k,5)=time_difference(3,k-(num_cars*2));
    else
        A2(k,5)=99999;
    end
end

```

```
load alpha5-2070-2070-2070
```

```

for k=1:num_cars
    A3(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A3(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A3(k,2)=99999;
    else
        A3(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A3(k,3)=occ_sum(k,1);
    else
        A3(k,3)=99999;
    end
    A3(k,4)=20;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A3(k,5)=time_difference(1,k);
    else
        A3(k,5)=99999;
    end
end

```

```

for k=num_cars+1:num_cars*2
    A3(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A3(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A3(k,2)=99999;
    else
        A3(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999

```

```

    A3(k,3)=occ_sum(k-num_cars,2);
else
    A3(k,3)=99999;
end
A3(k,4)=20;
if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
    A3(k,5)=time_difference(2,k-num_cars);
else
    A3(k,5)=99999;
end
end

for k=(num_cars*2)+1:num_cars*3
    A3(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A3(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A3(k,2)=99999;
    else
        A3(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A3(k,3)=occ_sum((k-num_cars*2),3);
    else
        A3(k,3)=99999;
    end
    A3(k,4)=20;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A3(k,5)=time_difference(3,k-(num_cars*2));
    else
        A3(k,5)=99999;
    end
end
end

load alpha8-2070-2070-2070

for k=1:num_cars
    A4(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A4(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A4(k,2)=99999;
    else
        A4(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A4(k,3)=occ_sum(k,1);
    else
        A4(k,3)=99999;
    end
    A4(k,4)=20;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A4(k,5)=time_difference(1,k);
    else
        A4(k,5)=99999;
    end
end
end

```

```

for k=num_cars+1:num_cars*2
    A4(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A4(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A4(k,2)=99999;
    else
        A4(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A4(k,3)=occ_sum(k-num_cars,2);
    else
        A4(k,3)=99999;
    end
    A4(k,4)=20;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A4(k,5)=time_difference(2,k-num_cars);
    else
        A4(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A4(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A4(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A4(k,2)=99999;
    else
        A4(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A4(k,3)=occ_sum((k-num_cars*2),3);
    else
        A4(k,3)=99999;
    end
    A4(k,4)=20;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A4(k,5)=time_difference(3,k-(num_cars*2));
    else
        A4(k,5)=99999;
    end
end

load alpha10-2070-2070-2070

for k=1:num_cars
    A5(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A5(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A5(k,2)=99999;
    else
        A5(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999

```

```

    A5(k,3)=occ_sum(k,1);
else
    A5(k,3)=99999;
end
A5(k,4)=20;
if time_difference(1,k)>0 && time_difference(1,k)~=99999
    A5(k,5)=time_difference(1,k);
else
    A5(k,5)=99999;
end
end

for k=num_cars+1:num_cars*2
    A5(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A5(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A5(k,2)=99999;
    else
        A5(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A5(k,3)=occ_sum(k-num_cars,2);
    else
        A5(k,3)=99999;
    end
    A5(k,4)=20;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A5(k,5)=time_difference(2,k-num_cars);
    else
        A5(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A5(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A5(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A5(k,2)=99999;
    else
        A5(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A5(k,3)=occ_sum((k-num_cars*2),3);
    else
        A5(k,3)=99999;
    end
    A5(k,4)=20;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A5(k,5)=time_difference(3,k-(num_cars*2));
    else
        A5(k,5)=99999;
    end
end

load alpha1-3060-4050-7020

```

```

for k=1:num_cars
    A6(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A6(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A6(k,2)=99999;
    else
        A6(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A6(k,3)=occ_sum(k,1);
    else
        A6(k,3)=99999;
    end
    A6(k,4)=30;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A6(k,5)=time_difference(1,k);
    else
        A6(k,5)=99999;
    end
end

for k=num_cars+1:num_cars*2
    A6(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A6(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A6(k,2)=99999;
    else
        A6(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A6(k,3)=occ_sum(k-num_cars,2);
    else
        A6(k,3)=99999;
    end
    A6(k,4)=40;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A6(k,5)=time_difference(2,k-num_cars);
    else
        A6(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A6(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A6(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A6(k,2)=99999;
    else
        A6(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A6(k,3)=occ_sum((k-num_cars*2),3);
    else

```



```

    A6(k,3)=99999;
end
A6(k,4)=70;
if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
    A6(k,5)=time_difference(3,k-(num_cars*2));
else
    A6(k,5)=99999;
end
end

```

```
load alpha3-3060-4050-7020
```

```

for k=1:num_cars
    A7(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A7(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A7(k,2)=99999;
    else
        A7(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A7(k,3)=occ_sum(k,1);
    else
        A7(k,3)=99999;
    end
    A7(k,4)=30;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A7(k,5)=time_difference(1,k);
    else
        A7(k,5)=99999;
    end
end
end

```

```

for k=num_cars+1:num_cars*2
    A7(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A7(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A7(k,2)=99999;
    else
        A7(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A7(k,3)=occ_sum(k-num_cars,2);
    else
        A7(k,3)=99999;
    end
    A7(k,4)=40;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A7(k,5)=time_difference(2,k-num_cars);
    else
        A7(k,5)=99999;
    end
end
end

```

```
for k=(num_cars*2)+1:num_cars*3
```

```

A7(k,1)=length_3;
if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
    A7(k,2)=1/flow_sum((k-num_cars*2),3);
elseif flow_sum(k-(num_cars*2),3)==99999
    A7(k,2)=99999;
else
    A7(k,2)=1;
end
if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
    A7(k,3)=occ_sum((k-num_cars*2),3);
else
    A7(k,3)=99999;
end
A7(k,4)=70;
if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
    A7(k,5)=time_difference(3,k-(num_cars*2));
else
    A7(k,5)=99999;
end
end

```

```
load alpha5-3060-4050-7020
```

```

for k=1:num_cars
    A8(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A8(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A8(k,2)=99999;
    else
        A8(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A8(k,3)=occ_sum(k,1);
    else
        A8(k,3)=99999;
    end
    A8(k,4)=30;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A8(k,5)=time_difference(1,k);
    else
        A8(k,5)=99999;
    end
end
end

```

```

for k=num_cars+1:num_cars*2
    A8(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A8(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A8(k,2)=99999;
    else
        A8(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A8(k,3)=occ_sum(k-num_cars,2);
    else

```

```

    A8(k,3)=99999;
end
A8(k,4)=40;
if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
    A8(k,5)=time_difference(2,k-num_cars);
else
    A8(k,5)=99999;
end
end

for k=(num_cars*2)+1:num_cars*3
    A8(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A8(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A8(k,2)=99999;
    else
        A8(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A8(k,3)=occ_sum((k-num_cars*2),3);
    else
        A8(k,3)=99999;
    end
    A8(k,4)=70;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A8(k,5)=time_difference(3,k-(num_cars*2));
    else
        A8(k,5)=99999;
    end
end
end

load alpha8-3060-4050-7020

for k=1:num_cars
    A9(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A9(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A9(k,2)=99999;
    else
        A9(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A9(k,3)=occ_sum(k,1);
    else
        A9(k,3)=99999;
    end
    A9(k,4)=30;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A9(k,5)=time_difference(1,k);
    else
        A9(k,5)=99999;
    end
end
end

for k=num_cars+1:num_cars*2

```

```

A9(k,1)=length_2;
if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
    A9(k,2)=1/(flow_sum(k-num_cars,2));
elseif flow_sum(k-num_cars,2)==99999
    A9(k,2)=99999;
else
    A9(k,2)=1;
end
if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
    A9(k,3)=occ_sum(k-num_cars,2);
else
    A9(k,3)=99999;
end
A9(k,4)=40;
if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
    A9(k,5)=time_difference(2,k-num_cars);
else
    A9(k,5)=99999;
end
end

for k=(num_cars*2)+1:num_cars*3
    A9(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A9(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A9(k,2)=99999;
    else
        A9(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A9(k,3)=occ_sum((k-num_cars*2),3);
    else
        A9(k,3)=99999;
    end
    A9(k,4)=70;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A9(k,5)=time_difference(3,k-(num_cars*2));
    else
        A9(k,5)=99999;
    end
end

load alpha10-3060-4050-7020

for k=1:num_cars
    A10(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A10(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A10(k,2)=99999;
    else
        A10(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A10(k,3)=occ_sum(k,1);
    else

```

```

    A10(k,3)=99999;
end
A10(k,4)=30;
if time_difference(1,k)>0 && time_difference(1,k)~=99999
    A10(k,5)=time_difference(1,k);
else
    A10(k,5)=99999;
end
end

for k=num_cars+1:num_cars*2
    A10(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A10(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A10(k,2)=99999;
    else
        A10(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A10(k,3)=occ_sum(k-num_cars,2);
    else
        A10(k,3)=99999;
    end
    A10(k,4)=40;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A10(k,5)=time_difference(2,k-num_cars);
    else
        A10(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A10(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A10(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A10(k,2)=99999;
    else
        A10(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A10(k,3)=occ_sum((k-num_cars*2),3);
    else
        A10(k,3)=99999;
    end
    A10(k,4)=70;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A10(k,5)=time_difference(3,k-(num_cars*2));
    else
        A10(k,5)=99999;
    end
end

load alpha1-7020-4050-3060

for k=1:num_cars

```

```

A11(k,1)=length_1;
if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
    A11(k,2)=1/flow_sum(k,1);
elseif flow_sum(k,1)==99999
    A11(k,2)=99999;
else
    A11(k,2)=1;
end
if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
    A11(k,3)=occ_sum(k,1);
else
    A11(k,3)=99999;
end
A11(k,4)=70;
if time_difference(1,k)>0 && time_difference(1,k)~=99999
    A11(k,5)=time_difference(1,k);
else
    A11(k,5)=99999;
end
end

for k=num_cars+1:num_cars*2
    A11(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A11(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A11(k,2)=99999;
    else
        A11(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A11(k,3)=occ_sum(k-num_cars,2);
    else
        A11(k,3)=99999;
    end
    A11(k,4)=40;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A11(k,5)=time_difference(2,k-num_cars);
    else
        A11(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A11(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A11(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A11(k,2)=99999;
    else
        A11(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A11(k,3)=occ_sum((k-num_cars*2),3);
    else
        A11(k,3)=99999;
    end
end

```

```

A11(k,4)=30;
if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
    A11(k,5)=time_difference(3,k-(num_cars*2));
else
    A11(k,5)=99999;
end
end

load alpha3-7020-4050-3060

for k=1:num_cars
    A12(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A12(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A12(k,2)=99999;
    else
        A12(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A12(k,3)=occ_sum(k,1);
    else
        A12(k,3)=99999;
    end
    A12(k,4)=70;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A12(k,5)=time_difference(1,k);
    else
        A12(k,5)=99999;
    end
end

for k=num_cars+1:num_cars*2
    A12(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A12(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A12(k,2)=99999;
    else
        A12(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A12(k,3)=occ_sum(k-num_cars,2);
    else
        A12(k,3)=99999;
    end
    A12(k,4)=40;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A12(k,5)=time_difference(2,k-num_cars);
    else
        A12(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A12(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999

```

```

    A12(k,2)=1/flow_sum((k-num_cars*2),3);
elseif flow_sum(k-(num_cars*2),3)==99999
    A12(k,2)=99999;
else
    A12(k,2)=1;
end
if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
    A12(k,3)=occ_sum((k-num_cars*2),3);
else
    A12(k,3)=99999;
end
A12(k,4)=30;
if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
    A12(k,5)=time_difference(3,k-(num_cars*2));
else
    A12(k,5)=99999;
end
end

```

```
load alpha5-7020-4050-3060
```

```

for k=1:num_cars
    A13(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A13(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A13(k,2)=99999;
    else
        A13(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A13(k,3)=occ_sum(k,1);
    else
        A13(k,3)=99999;
    end
    A13(k,4)=70;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A13(k,5)=time_difference(1,k);
    else
        A13(k,5)=99999;
    end
end
end

```

```

for k=num_cars+1:num_cars*2
    A13(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A13(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A13(k,2)=99999;
    else
        A13(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A13(k,3)=occ_sum(k-num_cars,2);
    else
        A13(k,3)=99999;
    end
end

```



```

A13(k,4)=40;
if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
    A13(k,5)=time_difference(2,k-num_cars);
else
    A13(k,5)=99999;
end
end

for k=(num_cars*2)+1:num_cars*3
    A13(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A13(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A13(k,2)=99999;
    else
        A13(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A13(k,3)=occ_sum((k-num_cars*2),3);
    else
        A13(k,3)=99999;
    end
    A13(k,4)=30;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A13(k,5)=time_difference(3,k-(num_cars*2));
    else
        A13(k,5)=99999;
    end
end

load alpha8-7020-4050-3060

for k=1:num_cars
    A14(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A14(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A14(k,2)=99999;
    else
        A14(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A14(k,3)=occ_sum(k,1);
    else
        A14(k,3)=99999;
    end
    A14(k,4)=70;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A14(k,5)=time_difference(1,k);
    else
        A14(k,5)=99999;
    end
end

for k=num_cars+1:num_cars*2
    A14(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999

```

```

    A14(k,2)=1/(flow_sum(k-num_cars,2));
elseif flow_sum(k-num_cars,2)==99999
    A14(k,2)=99999;
else
    A14(k,2)=1;
end
if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
    A14(k,3)=occ_sum(k-num_cars,2);
else
    A14(k,3)=99999;
end
A14(k,4)=40;
if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
    A14(k,5)=time_difference(2,k-num_cars);
else
    A14(k,5)=99999;
end
end

for k=(num_cars*2)+1:num_cars*3
    A14(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A14(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A14(k,2)=99999;
    else
        A14(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A14(k,3)=occ_sum((k-num_cars*2),3);
    else
        A14(k,3)=99999;
    end
    A14(k,4)=30;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A14(k,5)=time_difference(3,k-(num_cars*2));
    else
        A14(k,5)=99999;
    end
end

load alpha10-7020-4050-3060

for k=1:num_cars
    A15(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A15(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A15(k,2)=99999;
    else
        A15(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A15(k,3)=occ_sum(k,1);
    else
        A15(k,3)=99999;
    end
end

```

```

A15(k,4)=70;
if time_difference(1,k)>0 && time_difference(1,k)~=99999
    A15(k,5)=time_difference(1,k);
else
    A15(k,5)=99999;
end
end

for k=num_cars+1:num_cars*2
    A15(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A15(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A15(k,2)=99999;
    else
        A15(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A15(k,3)=occ_sum(k-num_cars,2);
    else
        A15(k,3)=99999;
    end
    A15(k,4)=40;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A15(k,5)=time_difference(2,k-num_cars);
    else
        A15(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A15(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A15(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A15(k,2)=99999;
    else
        A15(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A15(k,3)=occ_sum((k-num_cars*2),3);
    else
        A15(k,3)=99999;
    end
    A15(k,4)=30;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A15(k,5)=time_difference(3,k-(num_cars*2));
    else
        A15(k,5)=99999;
    end
end

load alpha1-8010-8010-8010

for k=1:num_cars
    A16(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999

```

```

    A16(k,2)=1/flow_sum(k,1);
elseif flow_sum(k,1)==99999
    A16(k,2)=99999;
else
    A16(k,2)=1;
end
if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
    A16(k,3)=occ_sum(k,1);
else
    A16(k,3)=99999;
end
A16(k,4)=80;
if time_difference(1,k)>0 && time_difference(1,k)~=99999
    A16(k,5)=time_difference(1,k);
else
    A16(k,5)=99999;
end
end

for k=num_cars+1:num_cars*2
    A16(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A16(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A16(k,2)=99999;
    else
        A16(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A16(k,3)=occ_sum(k-num_cars,2);
    else
        A16(k,3)=99999;
    end
    A16(k,4)=80;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A16(k,5)=time_difference(2,k-num_cars);
    else
        A16(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A16(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A16(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A16(k,2)=99999;
    else
        A16(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A16(k,3)=occ_sum((k-num_cars*2),3);
    else
        A16(k,3)=99999;
    end
    A16(k,4)=80;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999

```

```

    A16(k,5)=time_difference(3,k-(num_cars*2));
else
    A16(k,5)=99999;
end
end
end

load alpha3-8010-8010-8010

for k=1:num_cars
    A17(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A17(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A17(k,2)=99999;
    else
        A17(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A17(k,3)=occ_sum(k,1);
    else
        A17(k,3)=99999;
    end
    A17(k,4)=80;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A17(k,5)=time_difference(1,k);
    else
        A17(k,5)=99999;
    end
end
end

for k=num_cars+1:num_cars*2
    A17(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A17(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A17(k,2)=99999;
    else
        A17(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A17(k,3)=occ_sum(k-num_cars,2);
    else
        A17(k,3)=99999;
    end
    A17(k,4)=80;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A17(k,5)=time_difference(2,k-num_cars);
    else
        A17(k,5)=99999;
    end
end
end

for k=(num_cars*2)+1:num_cars*3
    A17(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A17(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999

```

```

    A17(k,2)=99999;
else
    A17(k,2)=1;
end
if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
    A17(k,3)=occ_sum((k-num_cars*2),3);
else
    A17(k,3)=99999;
end
A17(k,4)=80;
if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
    A17(k,5)=time_difference(3,k-(num_cars*2));
else
    A17(k,5)=99999;
end
end

```

```
load alpha5-8010-8010-8010
```

```

for k=1:num_cars
    A18(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A18(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A18(k,2)=99999;
    else
        A18(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A18(k,3)=occ_sum(k,1);
    else
        A18(k,3)=99999;
    end
    A18(k,4)=80;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A18(k,5)=time_difference(1,k);
    else
        A18(k,5)=99999;
    end
end
end

```

```

for k=num_cars+1:num_cars*2
    A18(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A18(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A18(k,2)=99999;
    else
        A18(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A18(k,3)=occ_sum(k-num_cars,2);
    else
        A18(k,3)=99999;
    end
    A18(k,4)=80;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999

```

```

    A18(k,5)=time_difference(2,k-num_cars);
else
    A18(k,5)=99999;
end
end

for k=(num_cars*2)+1:num_cars*3
    A18(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A18(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A18(k,2)=99999;
    else
        A18(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A18(k,3)=occ_sum((k-num_cars*2),3);
    else
        A18(k,3)=99999;
    end
    A18(k,4)=80;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A18(k,5)=time_difference(3,k-(num_cars*2));
    else
        A18(k,5)=99999;
    end
end

load alpha8-8010-8010-8010

for k=1:num_cars
    A19(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A19(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A19(k,2)=99999;
    else
        A19(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A19(k,3)=occ_sum(k,1);
    else
        A19(k,3)=99999;
    end
    A19(k,4)=80;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999
        A19(k,5)=time_difference(1,k);
    else
        A19(k,5)=99999;
    end
end

for k=num_cars+1:num_cars*2
    A19(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A19(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999

```

```

    A19(k,2)=99999;
else
    A19(k,2)=1;
end
if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
    A19(k,3)=occ_sum(k-num_cars,2);
else
    A19(k,3)=99999;
end
A19(k,4)=80;
if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
    A19(k,5)=time_difference(2,k-num_cars);
else
    A19(k,5)=99999;
end
end

for k=(num_cars*2)+1:num_cars*3
    A19(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A19(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A19(k,2)=99999;
    else
        A19(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A19(k,3)=occ_sum((k-num_cars*2),3);
    else
        A19(k,3)=99999;
    end
    A19(k,4)=80;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A19(k,5)=time_difference(3,k-(num_cars*2));
    else
        A19(k,5)=99999;
    end
end

load alpha10-8010-8010-8010

for k=1:num_cars
    A20(k,1)=length_1;
    if flow_sum(k,1)>0 && flow_sum(k,1)~=99999
        A20(k,2)=1/flow_sum(k,1);
    elseif flow_sum(k,1)==99999
        A20(k,2)=99999;
    else
        A20(k,2)=1;
    end
    if occ_sum(k,1)>=0 && occ_sum(k,1)~=99999
        A20(k,3)=occ_sum(k,1);
    else
        A20(k,3)=99999;
    end
    A20(k,4)=80;
    if time_difference(1,k)>0 && time_difference(1,k)~=99999

```



```

    A20(k,5)=time_difference(1,k);
else
    A20(k,5)=99999;
end
end

for k=num_cars+1:num_cars*2
    A20(k,1)=length_2;
    if flow_sum(k-num_cars,2)>0 && flow_sum(k-num_cars,2)~=99999
        A20(k,2)=1/(flow_sum(k-num_cars,2));
    elseif flow_sum(k-num_cars,2)==99999
        A20(k,2)=99999;
    else
        A20(k,2)=1;
    end
    if occ_sum(k-num_cars,2)>=0 && occ_sum(k-num_cars,2)~=99999
        A20(k,3)=occ_sum(k-num_cars,2);
    else
        A20(k,3)=99999;
    end
    A20(k,4)=80;
    if time_difference(2,k-num_cars)>0 && time_difference(2,k-num_cars)~=99999
        A20(k,5)=time_difference(2,k-num_cars);
    else
        A20(k,5)=99999;
    end
end

for k=(num_cars*2)+1:num_cars*3
    A20(k,1)=length_3;
    if flow_sum(k-(num_cars*2),3)>0 && flow_sum(k-(num_cars*2),3)~=99999
        A20(k,2)=1/flow_sum((k-num_cars*2),3);
    elseif flow_sum(k-(num_cars*2),3)==99999
        A20(k,2)=99999;
    else
        A20(k,2)=1;
    end
    if occ_sum(k-(num_cars*2),3)>=0 && occ_sum(k-(num_cars*2),3)~=99999
        A20(k,3)=occ_sum((k-num_cars*2),3);
    else
        A20(k,3)=99999;
    end
    A20(k,4)=80;
    if time_difference(3,k-(num_cars*2))>0 && time_difference(3,k-(num_cars*2))~=99999
        A20(k,5)=time_difference(3,k-(num_cars*2));
    else
        A20(k,5)=99999;
    end
end

A=[A12;A5;A9;A18;A20;A15;A7;A4;A1;A11;A10;A14;A13;A3;A17;A8;A16;A2;A19;A6];

save matrix_A_c

```

Προσεγγιστικός Αλγόριθμος – Υπολογισμός Σφαλμάτων

(calculate E C.m)

```
clear all
close all

load matrix_A_c

gamma=0.000000000000001;
for jj=1:2000,
e=zeros(50000,1);
for i=1:length(A)-2000
    if A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));

    e(i,1)=(A(i,5)-psi);

    w2(1,1)=w(1,1)+gamma*e(i,1)*A(i,1);
    w2(2,1)=w(2,1)+gamma*e(i,1)*A(i,2);
    w2(3,1)=w(3,1)+gamma*e(i,1)*A(i,3);
    w2(4,1)=w(4,1)+gamma*e(i,1)*A(i,4);
    w2(5,1)=w(5,1)+gamma*e(i,1)*A(i,1)*A(i,2);
    w2(6,1)=w(6,1)+gamma*e(i,1)*A(i,1)*A(i,3);
    w2(7,1)=w(7,1)+gamma*e(i,1)*A(i,1)*A(i,4);
    w2(8,1)=w(8,1)+gamma*e(i,1)*A(i,2)*A(i,3);
    w2(9,1)=w(9,1)+gamma*e(i,1)*A(i,2)*A(i,4);
    w2(10,1)=w(10,1)+gamma*e(i,1)*A(i,3)*A(i,4);
    w2(11,1)=w(11,1)+gamma*e(i,1)*A(i,1)*A(i,2)*A(i,3);
    w2(12,1)=w(12,1)+gamma*e(i,1)*A(i,2)*A(i,3)*A(i,4);
    w2(13,1)=w(13,1)+gamma*e(i,1)*A(i,1)*A(i,3)*A(i,4);
    w2(14,1)=w(14,1)+gamma*e(i,1)*A(i,1)*A(i,2)*A(i,4);

    w(:,1)=w2(:,1);
end

end

E(jj,1)=norm(e);

for i=1:length(A)-2000
    if A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));

    e(i,1)=(A(i,5)-psi);
    PSI(i,1)=psi;

end

end
```

```

end
end
E2(jj,1)=norm(e);
for i=length(A)-2000+1:length(A)
if A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));
e(i,1)=(A(i,5)-psi);
PSI2(i,1)=psi;

end
E3(jj,1)=norm(e);
end
end
subplot(3,1,1), plot(E)
grid
%axis([0 5000 200 600])
ylabel('RMS error')

subplot(3,1,2), plot(E2)
grid
%axis([0 5000 200 600])
ylabel('RMS error2')

subplot(3,1,3), plot(E3)
grid
%axis([0 5000 200 600])
ylabel('RMS error3')

xlabel('iterations')

```

Κατασκευή σχήματος 3-plots

```

close all
clear all
load A.dat
load PSI.dat

ij=find(A(:,5)~=99999);
%ij=find(A(601:1200,5)~=99999);

figure
subplot(3,1,1), plot(PSI(ij,1)/4)
hold
subplot(3,1,1), plot(A(ij,5)/4, 'r')
hold
grid

subplot(3,1,2), plot(A(ij,3))
grid

subplot(3,1,3), plot(100*(PSI(ij,1)-A(ij,5))./A(ij,5))
grid

```

```
axis([0 3000 -100 100])
```

Βελτιωμένος προσεγγιστικός αλγόριθμος (occupancy sum ≤ 60)

(calculate swsto plot.m)

```
clear all
close all

load matrix_A_case_D

% Xreiazotan na proste0ei kai to parakatw:
for i=1:length(A)
    if A(i,3)>60
        A(i,3)=99999;
    end
end
%
gamma=0.00000000000001;
for jj=1:1000,
e=zeros(50000,1);
for i=1:length(A)-2000
    if A(i,3)>=0 && A(i,3)<=60 && A(i,3)~=99999 && A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));

    e(i,1)=(A(i,5)-psi);

    w(1,1)=w(1,1)+gamma*e(i,1)*A(i,1);
    w(2,1)=w(2,1)+gamma*e(i,1)*A(i,2);
    w(3,1)=w(3,1)+gamma*e(i,1)*A(i,3);
    w(4,1)=w(4,1)+gamma*e(i,1)*A(i,4);
    w(5,1)=w(5,1)+gamma*e(i,1)*A(i,1)*A(i,2);
    w(6,1)=w(6,1)+gamma*e(i,1)*A(i,1)*A(i,3);
    w(7,1)=w(7,1)+gamma*e(i,1)*A(i,1)*A(i,4);
    w(8,1)=w(8,1)+gamma*e(i,1)*A(i,2)*A(i,3);
    w(9,1)=w(9,1)+gamma*e(i,1)*A(i,2)*A(i,4);
    w(10,1)=w(10,1)+gamma*e(i,1)*A(i,3)*A(i,4);
    w(11,1)=w(11,1)+gamma*e(i,1)*A(i,1)*A(i,2)*A(i,3);
    w(12,1)=w(12,1)+gamma*e(i,1)*A(i,2)*A(i,3)*A(i,4);
    w(13,1)=w(13,1)+gamma*e(i,1)*A(i,1)*A(i,3)*A(i,4);
    w(14,1)=w(14,1)+gamma*e(i,1)*A(i,1)*A(i,2)*A(i,4);

    % w

end

end
```

```

E(jj,1)=norm(e);

for i=1:length(A)-2000
    if A(i,3)>=0 && A(i,3)<=60 && A(i,3)~=99999 && A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));
    e(i,1)=(A(i,5)-psi);
    PSI(i,1)=psi;

    end
end
E2(jj,1)=norm(e);
for i=length(A)-2000+1:length(A)
    if A(i,3)>=0 && A(i,3)<=60 && A(i,3)~=99999 && A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));
    e(i,1)=(A(i,5)-psi);
    PSI(i,1)=psi;

    end
end
E3(jj,1)=norm(e);
end

jj
end
subplot(3,1,1), plot(E)
grid
%axis([0 5000 200 600])
ylabel('MSE')

subplot(3,1,2), plot(E2)
grid
%axis([0 5000 200 600])
ylabel('MSE 2')

subplot(3,1,3), plot(E3)
grid
%axis([0 5000 200 600])
ylabel('MSE 3')

xlabel('iterations')

save A.dat A -ascii
save PSI.dat PSI -ascii

```

(calculate swsto plot 3-plots.m)

```
close all
clear all
load A.dat
load PSI.dat

ij=find(A(:,5)~=99999);

figure
subplot(3,1,1), plot(PSI(ij,1)/4)
hold
subplot(3,1,1), plot(A(ij,5)/4, 'r')
hold
grid

subplot(3,1,2), plot(A(ij,3))
grid

subplot(3,1,3), plot(100*(PSI(ij,1)-A(ij,5))./A(ij,5))
grid
axis([0 3000 -100 100])
```

Βελτιωμένος προσεγγιστικός αλγόριθμος (occupancy sum > 60)

(calculate swsto plot.m)

```
clear all
close all

load matrix_A_case_D

for i=1:length(A)
    if A(i,3)<61
        A(i,3)=99999;
    end
end

gamma=0.000000000000001;
for jj=1:1000,
    e=zeros(50000,1);
    for i=1:length(A)-2000
        if A(i,3)>=0 && A(i,3)>60 && A(i,3)~=99999 && A(i,5)>0 && A(i,5)~=99999
            psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2)+w(3,1)*A(i,3)+w(4,1)*A(i,4)+w(5,1)*A(i,1)*A(i,2)+w(6,1)*A(i,1)*
            A(i,3)+w(7,1)*A(i,1)*A(i,4)+w(8,1)*A(i,2)*A(i,3)+w(9,1)*A(i,2)*A(i,4)+w(10,1)*A(i,3)*A(i,4)+w(11,
            1)*A(i,1)*A(i,2)*A(i,3)+w(12,1)*A(i,2)*A(i,3)*A(i,4)+w(13,1)*A(i,1)*A(i,3)*A(i,4)+w(14,1)*A(i,1)*A(i,2)
            )*A(i,4);

            e(i,1)=(A(i,5)-psi);

            w(1,1)=w(1,1)+gamma*e(i,1)*A(i,1);
            w(2,1)=w(2,1)+gamma*e(i,1)*A(i,2);
            w(3,1)=w(3,1)+gamma*e(i,1)*A(i,3);
```

```

w(4,1)=w(4,1)+gamma*e(i,1)*A(i,4);
w(5,1)=w(5,1)+gamma*e(i,1)*A(i,1)*A(i,2);
w(6,1)=w(6,1)+gamma*e(i,1)*A(i,1)*A(i,3);
w(7,1)=w(7,1)+gamma*e(i,1)*A(i,1)*A(i,4);
w(8,1)=w(8,1)+gamma*e(i,1)*A(i,2)*A(i,3);
w(9,1)=w(9,1)+gamma*e(i,1)*A(i,2)*A(i,4);
w(10,1)=w(10,1)+gamma*e(i,1)*A(i,3)*A(i,4);
w(11,1)=w(11,1)+gamma*e(i,1)*A(i,1)*A(i,2)*A(i,3);
w(12,1)=w(12,1)+gamma*e(i,1)*A(i,2)*A(i,3)*A(i,4);
w(13,1)=w(13,1)+gamma*e(i,1)*A(i,1)*A(i,3)*A(i,4);
w(14,1)=w(14,1)+gamma*e(i,1)*A(i,1)*A(i,2)*A(i,4);

% w

end

end

E(jj,1)=norm(e);

for i=1:length(A)-2000
    if A(i,3)>=0 && A(i,3)>60 && A(i,3)~=99999 && A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));
    e(i,1)=(A(i,5)-psi);
    PSI(i,1)=psi;

    end

end
E2(jj,1)=norm(e);
for i=length(A)-2000+1:length(A)
    if A(i,3)>=0 && A(i,3)>60 && A(i,3)~=99999 && A(i,5)>0 && A(i,5)~=99999

psi=(w(1,1)*A(i,1))+w(2,1)*A(i,2))+w(3,1)*A(i,3))+w(4,1)*A(i,4))+w(5,1)*A(i,1)*A(i,2))+w(6,1)*A(i,1)*
A(i,3))+w(7,1)*A(i,1)*A(i,4))+w(8,1)*A(i,2)*A(i,3))+w(9,1)*A(i,2)*A(i,4))+w(10,1)*A(i,3)*A(i,4))+w(11,
1)*A(i,1)*A(i,2)*A(i,3))+w(12,1)*A(i,2)*A(i,3)*A(i,4))+w(13,1)*A(i,1)*A(i,3)*A(i,4))+w(14,1)*A(i,1)*A(i,2
)*A(i,4));
    e(i,1)=(A(i,5)-psi);
    PSI(i,1)=psi;

    end

end
E3(jj,1)=norm(e);
end

jj
end
subplot(3,1,1), plot(E)
grid
%axis([0 5000 200 600])
ylabel('MSE')

subplot(3,1,2), plot(E2)
grid
%axis([0 5000 200 600])

```

```

ylabel('MSE 2')

subplot(3,1,3), plot(E3)
grid
%axis([0 5000 200 600])
ylabel('MSE 3')

xlabel('iterations')

save A.dat A -ascii
save PSI.dat PSI -ascii

```

(calculate swsto plot 3-plots.m)

```

close all
clear all
load A.dat
load PSI.dat

ij=find(A(:,5)~=99999);
%ij=find(A(601:1200,5)~=99999);

figure
subplot(3,1,1), plot(PSI(ij,1)/4)
hold
subplot(3,1,1), plot(A(ij,5)/4, 'r')
hold
grid

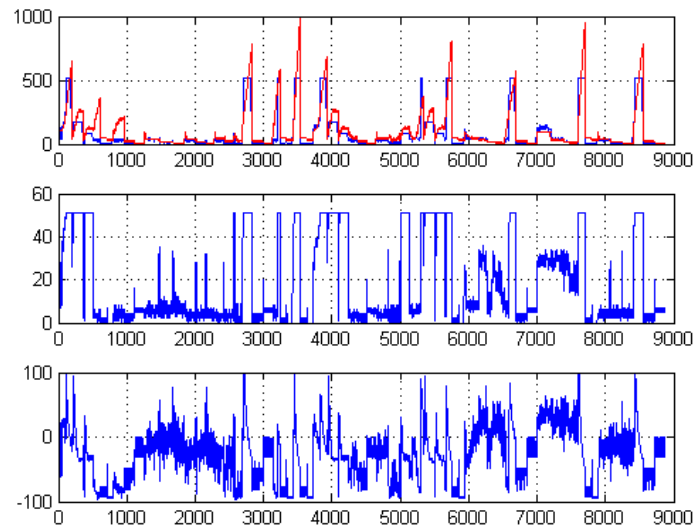
subplot(3,1,2), plot(A(ij,3))
grid

subplot(3,1,3), plot(100*(PSI(ij,1)-A(ij,5))./A(ij,5))
grid
axis([0 3000 -100 100])

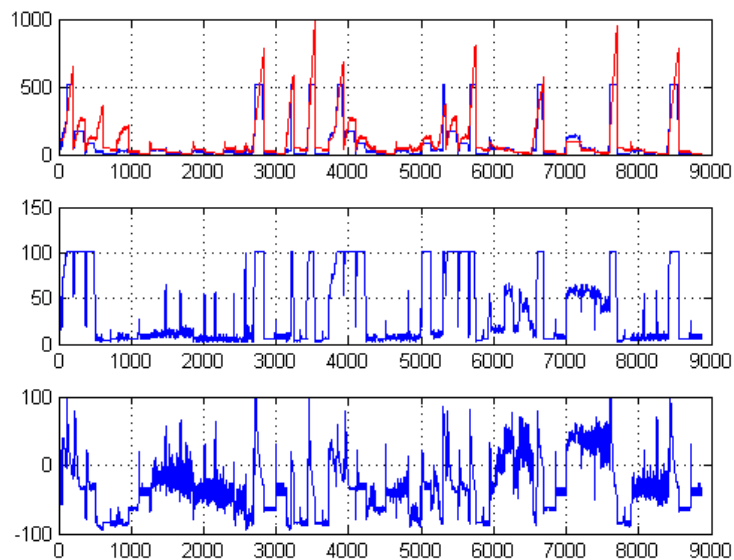
```


ΠΑΡΑΡΤΗΜΑ Β

ΣΧΗΜΑΤΑ



Σχήμα 25. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης για τις $t=50$ προηγούμενες χρονικές στιγμές



Σχήμα 26. Γραφικές παραστάσεις του χρόνου και του εκτιμώμενου χρόνου, της μεταβλητής occupancy sum και του ποσοστιαίου λάθους εκτίμησης για τις $t=100$ προηγούμενες χρονικές στιγμές

ΒΙΒΛΙΟΓΡΑΦΙΑ

Αρβανιτάκη Α., «Μακροοικονομική πρόβλεψη με χρήση γραμμικών μοντέλων και τεχνητών νευρωνικών δικτύων», Τμήμα Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης, 2005

Γαλανόπουλος Γ., «Ανάπτυξη μικροσκοπικού μοντέλου προσομοίωσης κυκλοφορίας για την εκτίμηση της χωρικής κατάληψης», Τμήμα Μηχανικών Παραγωγής και Διοίκησης, Πολυτεχνείο Κρήτης, 2002

Κουικόγλου Β, «Προσομοίωση – Σημειώσεις Μαθήματος», Πολυτεχνείο Κρήτης, 2002

Bellemans T, De Schutter B. And De Moor B., “Models for traffic control”, Journal A, vol 43, no 3-4, pp. 13-22, 2002

Bertini R., Lindgren R. and Tantiyanugulchai, “Application of PARAMICS simulation at a diamond interchange”, Oregon department of transportation, 2002

Cayford R. and Merritt G., “Automatic Diagnostics of loop detectors and the data collection system in the Berkeley Highway lab”, 2004

Coifman B., “Estimating travel times and vehicle trajectories on freeways using dual loop detectors”, 2000

Coifman B., Dhoorjaty S. and Lee Z. “Estimating median velocity instead of mean velocity at single loop detectors”, 2001

Gupte S., Masoud O., Martin R. and Papanikolopoulos N, “Detection and classification of vehicles”, IEEE Transactions on intelligent transportation systems, vol. 3, No 1, 2002

May D. A., Cayford R., Coifman B. and Merritt G., “Loop detector data collection and travel time measurement in the Berkeley Highway Laboratory”, California PATH Research Report, University of California, Berkeley, 2003

Miska M, Muller T. and Van Zuylen H., “A driving behavior model for microscopic online simulation based on remote sensing and equipped vehicle data”, 2005

Miska, Muller & van Zuylen, 2004, MiOS – A Microscopic Online Simulator, CD-ROM Proceedings of the ITS World Conference 2004, Nagoya, Japan

Owen L. and McHale G., “Traffic flow simulation using CORSIM”, Proceedings of the 2000 Winter Simulation Conference, 2000

Potter T., “The evolution of inductive loop detector technology”

Rothery R.W., “Car following models”, Civil Engineering Department, University of Texas, 2002

Vigos G. “Real time estimation of vehicle-count within signalized links”, thesis submitted in partial fulfillment of the requirements for the degree of Master of Operational Research, Dept. of Production Engineering and Management, Technical University of Crete, 2006

Vitsaxakis I., “Use of automatic incident detection within coordinated traffic signal time determination”, Dept. of Production Engineering and Management, Technical University of Crete, 1997

Wang Y. and Nihan N., “Can single-loop detectors do the work of dual-loop detectors?”, Department of Civil and Environmental Engineering, University of Washington