ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Έλεγχος της οδικής κυκλοφορίας με χρήση αλγορίθμων ενισχυτικής μάθησης (Reinforcement Learning)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ηλίας Κ. Αλκίδης

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Έλεγχος της οδικής κυκλοφορίας με χρήση αλγορίθμων ενισχυτικής μάθησης (Reinforcement Learning)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ηλίας Κ. Αλκίδης

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30η Οκτωβρίου 2014.

.....	.....	.....
Ανδρέας-Γεώργιος Σταφυλοπάτης	Στέφανος Κόλλιας	Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

.....

Ηλίας Κ. Αλκίδης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ηλίας Κ. Αλκίδης, 2014.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Το κυκλοφοριακό ζήτημα είναι μείζονος σημασίας στις σύγχρονες αστικές κοινωνίες. Η δραματική αύξηση του αριθμού των ανθρώπων και των οχημάτων τις τελευταίες δεκαετίες στα αστικά κέντρα έχουν οδηγήσει σε καθημερινά φαινόμενα κυκλοφοριακής συμφόρησης.

Στην παρούσα εργασία πραγματοποιήθηκε μια εκτενής μελέτη του προβλήματος της διαχείρισης της οδικής κυκλοφορίας και των μεθόδων που εφαρμόζονται για την επίλυσή του. Χρησιμοποιήθηκε ο αλγόριθμος Q-learning της ενισχυτικής μάθησης στην ε-greedy παραλλαγή του με σκοπό να εκπαιδευτεί κατάλληλα ένας πράκτορας και να ελέγξει μια διασταύρωση τεσσάρων κατευθύνσεων. Σαν σημείο αναφοράς χρησιμοποιήθηκαν δύο άλλες μέθοδοι ελέγχου της ίδιας διασταύρωσης – ένας αλγόριθμος σταθερού χρονισμού και ένας προσαρμοστικός. Η λύση που προτάθηκε από την εργασία αυτή παρουσίασε σημαντικές βελτιώσεις στην απόδοση σε σχέση και με τις άλλες δύο μεθόδους, δείχνοντας ότι ευφυείς τεχνικές και, ειδικότερα, της ενισχυτικής μάθησης είναι ιδιαίτερα κατάλληλες για την επίλυση προβλημάτων όπως το κυκλοφοριακό.

Λέξεις Κλειδιά

Ενισχυτική Μάθηση, Έλεγχος Οδικής Κυκλοφορίας, προσομοίωση

Abstract

Traffic control is of great importance in modern urban societies. The dramatic increase of population and vehicles during the last decades has led to daily occurrences of traffic congestion.

In this thesis was conducted a thorough research of the issue of traffic control management and of the methods that are deployed for its solution. We used the Q-learning algorithm of reinforcement learning theory in its ε-greedy variation to effectively train an agent in controlling a four way intersection. As a baseline were used two conventional methods of control for the same intersection – a fixed time algorithm and an adaptive one. The solution that was proposed in this thesis has provided substantial improvements in performance in comparison to the other two methods, showing that intelligent techniques and specifically reinforcement learning are very well suited to solve problems such as traffic control.

KeyWords

Reinforcement learning, traffic control, Q-learning, congestion, simulation

Ευχαριστίες

Για την εκπόνηση της παρούσας εργασίας ευχαριστώ θερμά τον καθηγητή μου κ. Σταφυλοπάτη Ανδρέα που με ενέπνευσε μέσω της εκπαιδευτικής διαδικασίας να καταπιαστώ με το τόσο ενδιαφέρον θέμα των ευφυών τεχνικών. Αναντικατάστατη ήταν η βοήθεια του επιβλέποντα ερευνητή Δρ. Γιώργου Σιόλα, οποίος με ιδιαίτερη υπομονή με συμβούλευε και βοηθούσε σε όλα τα στάδια της εργασίας αυτής. Τέλος, ευχαριστώ την οικογένεια μου και τους φίλους μου για την υπομονή τους και την στήριξη που μου παρείχαν καθ' όλη την διάρκεια της φοίτησης μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΩΝ

ΚΕΦΑΛΑΙΟ 1ο – ΕΙΣΑΓΩΓΗ

1.1 Σημασία προβλήματος κυκλοφοριακής συμφόρησης

Το κυκλοφοριακό ζήτημα είναι μείζονος σημασίας στις σύγχρονες αστικές κοινωνίες. Η δραματική αύξηση του αριθμού των ανθρώπων και των οχημάτων τις τελευταίες δεκαετίες στα αστικά κέντρα έχουν οδηγήσει σε καθημερινά φαινόμενα κυκλοφοριακής συμφόρησης. Μερικά παραδείγματα που τονίζουν την σημασία του φαινομένου αυτού και την αναγκαιότητα της βέλτιστης επίλυσής του είναι τα ακόλουθα:

- Σε κατάσταση κυκλοφοριακής συμφόρησης σε κύριους οδικούς άξονες της Αττικής έχει παρατηρηθεί έως και πενταπλασιασμός του αναμενόμενου χρόνου διαδρομής (29η Έκθεση Λειτουργίας Κέντρου Διαχείρισης Κυκλοφορίας[1])
- Σύμφωνα με την Ευρωπαϊκή Επιτροπή[2] τα τελευταία στοιχεία, μεταξύ άλλων, δείχνουν πως η συμφόρηση στα αστικά κέντρα κοστίζει στην Ευρώπη περίπου το 1% του ΑΕΠ της ετησίως. Στα πλαίσια μιας οικογένειας αντίστοιχα το ποσοστό του εισοδήματος που αναφέρεται στις μετακινήσεις ανέρχεται στο 13,2%.
- Σε μεγάλες πόλεις ο χρόνος που αφιερώνεται στις μετακινήσεις από τους πολίτες είναι πολύ μεγάλος. Για παράδειγμα, στο Λονδίνο[3] το 20% των μετακινούμενων διαθέτουν πάνω από 2 ώρες ημερησίως για την πρόσβαση στην εργασία τους. Αντίστοιχα, στην Γερμανία[4] το 37% ξοδεύει πάνω από μια ώρα.

[1] http://www.patt.gov.gr/main/attachments2/11224_29_ekthesi_diax_kikloforias.pdf
[2] http://ec.europa.eu/transport/strategies/facts-and-figures/all-themes/index_en.htm
[3] http://www.tfl.gov.uk/assets/downloads/corporate/Travel-in-London-report-1.pdf
[4] http://www.mobilitaet-in-deutschland.de/engl%202008/index.htm

- Σύμφωνα με το White Paper του 2011[5] της Ευρωπαϊκής Επιτροπής, το αυξανόμενο επίπεδο της κυκλοφοριακής συμφόρησης στην Ευρώπη έχει αρνητικές συνέπειες για το περιβάλλον καθώς προκαλεί αυξημένη μόλυνση του αέρα και ηχορύπανση. Οδηγεί σε υψηλότερη κατανάλωση καυσίμων καθώς έχει αποδειχθεί πως η κατανάλωση αυξάνεται κατά περίπου 30% σε συνθήκες υψηλής συμφόρησης.

Από τα παραπάνω, συνεπάγεται εύκολα πως οι συνέπειες της κυκλοφοριακής συμφόρησης αυτής αγγίζουν πολλούς διαφορετικούς τομείς της σύγχρονης πραγματικότητας. Η ταλαιπωρία των μετακινούμενων, η αύξηση της παραγωγής ρύπων (CO, NO2, NO, SO2 κ.α.) και ηχορύπανσης, το υψηλό κόστος που συνεπάγεται και γενικότερα η υποβάθμιση του βιοτικού επιπέδου των κατοίκων των μεγάλων πόλεων είναι συνοπτικά κάποια από τα κυριότερα φαινόμενα που θα βελτιωθούν με την ανάπτυξη αποτελεσματικών μεθόδων για την καταπολέμηση του προβλήματος.

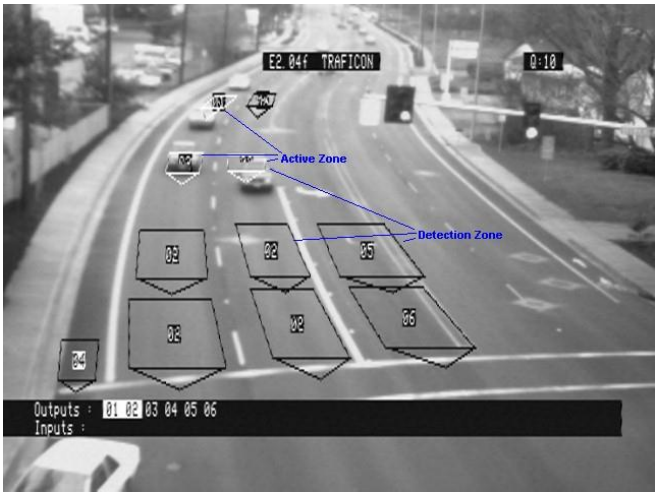
Εικόνα 1 – Αριστερά: Συμφόρηση στο Σάο Πάολο (Βραζιλία), Δεξιά: Έντονο μποτιλιάρισμα (gridlock)

[5] http://ec.europa.eu/transport/themes/strategies/doc/2011_white_paper/white_paper_2011_ia_full_en.pdf

1.2 Επιστημονικοί τομείς για την επίλυση της κυκλ. συμφόρησης

Ιδανικά, για την επίλυση τέτοιων φαινομένων θα έπρεπε να αναπτυχθεί πιο σύνθετο και επεκταμένο οδικό δίκτυο ικανό να εξυπηρετήσει την αυξημένη ζήτηση και να έχει χωρητικότητα τέτοια που να αποφεύγεται η δημιουργία ουρών και καθυστερήσεων. Παρόλα αυτά, τα οδικά δίκτυα σήμερα έχουν αγγίξει τα όρια των δυνατοτήτων επέκτασης τους λόγω χωροταξικών παραγόντων και δραματικά μεγάλου κόστους που θα προϋπέθετε ο επανασχεδιασμός τους. Ως εκ τούτου έχουν δημιουργηθεί αρκετές ερευνητικές κατευθύνσεις που επιχειρούν να βελτιώσουν την υπάρχουσα κατάσταση. Τέτοιες κατευθύνσεις είναι για παράδειγμα:

- Η πρόβλεψη της κίνησης (Clark, 2003), (Min, 2011) η οποία μπορεί να παρέχει εγκαίρως πληροφορίες για την βέλτιστη διαχείριση της αναμενόμενης ζήτησης στο οδικό δίκτυο. Σε αυτόν τον τομέα είναι πολύ δημοφιλείς οι ευφυείς τεχνικές για την προσέγγιση του προβλήματος.
- Μέθοδοι ανακατεύθυνσης της κίνησης των οχημάτων (traffic re-routing) που αποσκοπούν στην μείωση στον χρόνο ταξιδιού των οχημάτων μέσα στο οδικό δίκτυο (Pan, 2013).
- Προχωρημένες τεχνικές επεξεργασίας εικόνας και βίντεο ώστε να μπορούν ταχέως να εξάγονται συμπεράσματα από εναέριες κάμερες παρακολούθησης της κυκλοφορίας για σημεία συμφόρησης, περιπτώσεις ατυχημάτων, χαρακτηριστικά της οδικής κίνησης (μέση ταχύτητα, χρόνος αναμονής) κ.α. (Kamijo, 2000), (Coifman, 1998), (Somasundaram, 2013)



Εικόνα 2 - Οπτική Αναγνώριση Οχημάτων

Το πιο ευρέως αναπτυγμένο επιστημονικό πεδίο όμως αναφορικά με την επίλυση του κυκλοφοριακού ζητήματος είναι ίσως ο έλεγχος της οδικής κυκλοφορίας μέσω της υπάρχουσας υποδομής, που είναι κυρίως οι φωτεινοί σηματοδότες (φανάρια). Η ρύθμιση των φαναριών απαιτεί συνήθως εκτεταμένη μελέτη καθώς η αλληλεπίδραση μεταξύ επιλογών σε διαφορετικά σημεία του δικτύου, αν και όχι προφανής, είναι ιδιαίτερα έντονη. Παρόλα αυτά σε μια μεγάλη πλειοψηφία των διασταυρώσεων χρησιμοποιείται μια πολύ απλή μέθοδος ελέγχου η οποία αποτελείται από μια διαδοχή πράσινου σήματος σταθερού μήκους και περιόδου για κάθε ρεύμα που ανταγωνίζεται για την χρήση μιας διασταύρωσης. Αν και οι περίοδοι και τα μήκη αυτά είναι προϊόν μελέτης, δεν δύνανται να εξασφαλίσουν την ομαλή διευθέτηση της κυκλοφορίας, αφού η ζήτηση των οχημάτων σε διάφορους δρόμους κάθε άλλο παρά σταθερή είναι. Μέσα στην πάροδο μίας μέρας υπάρχουν διαστήματα υψηλής συγκέντρωσης οχημάτων σε διαφορετικές περιοχές του οδικού δικτύου αλλά και έκτακτα συμβάντα (ατυχήματα, έργα, εκδηλώσεις κ.α.). Συνεπώς απαιτείται ο σχεδιασμός συστημάτων που να είναι σε θέση να προσαρμόζονται στις αλλαγές του περιβάλλοντος και να καθορίζουν την βέλτιστη ρύθμιση των φαναριών με στόχο την βελτιστοποίηση της απόδοσης του δικτύου σε τοπικό επίπεδο (στην υπό έλεγχο διασταύρωση) αλλά και στο σύνολό του.

1.3. Η προσέγγιση της παρούσας εργασίας

Στην εργασία αυτή θα χρησιμοποιηθεί ως βασικός τρόπος για την προσέγγιση αυτού του ζητήματος μια μέθοδος της τεχνητής νοημοσύνης, η ενισχυτική μάθηση (Reinforcement Learning). Θα χρησιμοποιηθεί ένας ευφυής πράκτορας (agent) οποίος θα λαμβάνει πληροφορίες για το μήκος των ουρών οχημάτων στα ρεύματα που οδηγούν σε μία διασταύρωση και θα επιχειρεί να επιλέξει την καλύτερη στρατηγική αποφάσεων για να εξυπηρετήσει βέλτιστα το σύνολο των οχημάτων, βελτιστοποιώντας μια δοσμένη παράμετρο. Τέτοιες παράμετροι είναι η ελαχιστοποίηση του χρόνου μέσου αναμονής των οχημάτων, η ελαχιστοποίηση των εκπεμπόμενων ρύπων, η μεγιστοποίηση της μέσης ταχύτητας των οχημάτων, η ελαχιστοποίηση του μέσου χρόνου «ταξιδιού» κ.α.

Το πλαίσιο στο οποίο θα διεξαχθεί το σύνολο των πειραμάτων είναι ο προσομοιωτής κυκλοφορίας SUMO (Simulation of Urban Mobility) το οποίο είναι ένας προσομοιωτής ανοιχτού κώδικα που έχει αναπτυχθεί από Ινστιτούτο Συστημάτων Μετακίνησης (Institution of Transportation Systems) του Γερμανικού Κέντρου Αεροδιαστημικής (German Aerospace Center).

Το 2ο Κεφάλαιο θα αναλύσει τις σημαντικότερες από τις ήδη υπάρχουσες τεχνικές ελέγχου κυκλοφορίας που υπάγονται είτε σε παραδοσιακές προσεγγίσεις, είτε σε πιο σύγχρονες που αξιοποιούν μεθόδους τεχνητής νοημοσύνης. Το 3ο Κεφάλαιο θα αποτελέσει μια σύντομη ανασκόπηση τη θεωρίας της Ενισχυτικής Μάθησης (στο εξής RL - Reinforcement Learning) και τους διαφορετικούς αλγορίθμους που την συνιστούν. Το 4ο Κεφάλαιο θα αναφέρει μερικές από τις πιο σημαντικές προσεγγίσεις του ζητήματος του ελέγχου οδικής κυκλοφορίας με μεθόδους της ενισχυτικής μάθησης. Το 5ο Κεφάλαιο θα παρουσιάσει μια συνοπτική ανασκόπηση στην αναγκαιότητα προσομοίωσης των συγκοινωνιακών συστημάτων, στα μοντέλα που χρησιμοποιούνται καθώς και διάφορα προγράμματα που είναι διαθέσιμα για την πραγματοποίησή της. Λόγω του ότι το σύνολο της εργασίας αυτής βασίστηκε στις

δυνατότητες του προσομοιωτή SUMO το 6ο Κεφάλαιο θα αναλύσει τον τρόπο λειτουργίας του και τις βασικότερες από αυτές τις δυνατότητες. Θα ακολουθήσει το 7ο Κεφάλαιο με την υλοποίηση και τα πειράματα που πραγματοποιήθηκαν και το 8ο Κεφάλαιο με τα συμπεράσματα και τις προτάσεις για επέκταση της έρευνας.

ΚΕΦΑΛΑΙΟ 2ο – ΕΛΕΓΧΟΣ ΟΔΙΚΗΣ ΚΥΚΛΟΦΟΡΙΑΣ

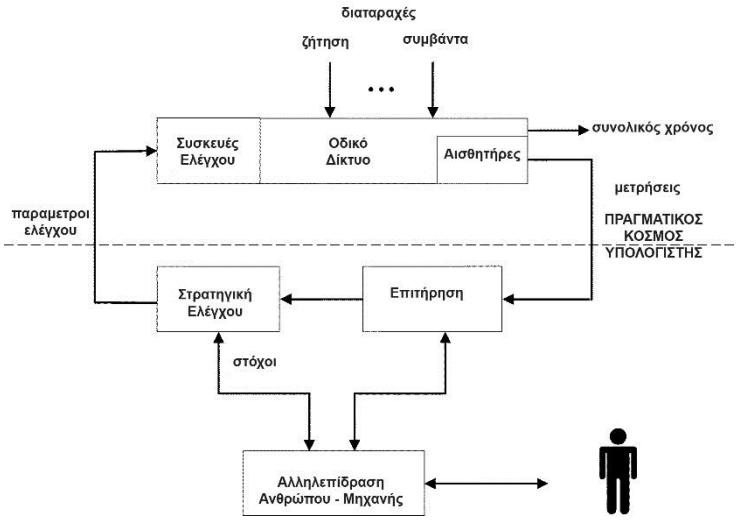
Το πρόβλημα του ελέγχου της οδικής κυκλοφορίας συνήθως συνοψίζεται στην εύρεση της βέλτιστης πολιτικής ρύθμισης των φαναριών στις υπό εξέταση διασταυρώσεις, αν και υπάρχουν και άλλα μέσα που μπορούν να χρησιμοποιηθούν για αυτόν τον σκοπό, όπως οι φωτεινές επιγραφές με μηνύματα προς τους οδηγούς, η τοποθέτηση τροχονόμων σε κατάλληλα σημεία κ.α. Εν τούτοις το μεγαλύτερο ενδιαφέρον και το μέσο που έχει την μεγαλύτερη και αμεσότερη επίδραση στην απόδοση του οδικού δικτύου είναι η χρήση φωτεινών σηματοδοτών.

Ο έλεγχος των φαναριών και η προσπάθεια να βρεθεί μια βέλτιστη πολιτική απόφασης (policy) για κάθε κατάσταση του οδικού δικτύου είναι ένα ιδιαίτερα δύσβατο έργο για κάποιους αναπόφευκτους λόγους. Αρχικά το πρόβλημα αυτό χαρακτηρίζεται από την στοχαστική του φύση και την μερική-παρατηρησιμότητα του (partially observable environment). Στην πράξη χρησιμοποιούνται για την εκτίμηση των χαρακτηριστικών του δικτύου συσκευές που τοποθετούνται στο οδόστρωμα σε συγκεκριμένες θέσεις (inductive loop detectors) ή εναέριες κάμερες (aerial detectors) που συγκεντρώνουν στοιχεία όπως το πλήθος των οχημάτων, η ταχύτητά τους, το είδος τους κλπ. Τα στοιχεία αυτά όμως είναι θορυβώδη και προσεγγιστικά και αυτό καθιστά ακόμα ένα εμπόδιο στην αποτελεσματική εκτίμηση των ακριβών μεγεθών.

Εικόνα 3 - Αριστερά: Inductive Loop Detector, Δεξιά: Εναέρια Κάμερα Διαχείρισης Κυκλοφορίας

Η στοχαστική φύση του προβλήματος συνίσταται στο ότι η κίνηση των οχημάτων εξαρτάται σε μεγάλο βαθμό από την συμπεριφορά των οδηγών, η οποία πολλές φορές είναι απρόβλεπτη και η μοντελοποίηση της παράγει προσεγγιστικά αποτελέσματα. Τέλος, ατυχήματα, παράνομο παρκάρισμα, κακές καιρικές συνθήκες και άλλα έκτακτα συμβάντα μπορούν να προκαλέσουν πολύ γρήγορα δραματικές αλλαγές στις καταστάσεις του δικτύου, με αποτέλεσμα να απαιτείται ταχύτατη προσαρμογή των συστημάτων ελέγχου κυκλοφορίας στα νέα δεδομένα. Αναποτελεσματική ρύθμιση της κυκλοφορίας παράγει έντονα φαινόμενα συμφόρησης, η οποία με την σειρά της οδηγεί το δίκτυο σε χαμηλή απόδοση και ολοένα αυξανόμενη συμφόρηση με ακόμα ταχύτερους ρυθμούς (Papageorgiou, 2003).

Το βασικό μοντέλο του βρόχου ελέγχου της κυκλοφορίας που χρησιμοποιείται και από τις παραδοσιακές προσεγγίσεις αλλά και από τις πιο σύγχρονες είναι το ακόλουθο:



Εικόνα 4 - Μοντέλο του βρόχου ελέγχου της κυκλοφορίας (Papageorgiou, 2003)

Οι παράμετροι ελέγχου αναφέρονται σε τιμές που σχετίζονται με τα φανάρια, τα μηνύματα προς τους οδηγούς στις φωτεινές επιγραφές, την ανακατεύθυνση της κίνησης μέσω παρακάμψεων κ.α. Αντίστοιχα η έξοδος του δικτύου είναι η μετρική που χρησιμοποιείται για την εκτίμηση της απόδοσής του, όπως για παράδειγμα ο συνολικός χρόνος παραμονής των οχημάτων στο δίκτυο. Η στρατηγική ελέγχου είναι το τμήμα εκείνο που αποφασίζει τις παραμέτρους ελέγχου με βάση τις επεξεργασμένες από το τμήμα της επιτήρησης μετρήσεις. Αυτές οι μετρήσεις μπορεί να περιλαμβάνουν μεγέθη όπως μήκη ουρών, αριθμό σταματημένων οχημάτων, μέση ταχύτητα κ.α.

Η κύρια διαφορά των τεχνικών τεχνητής νοημοσύνης από τις υπόλοιπες είναι πως το κομμάτι της στρατηγικής ελέγχου υλοποιείται με αλγορίθμους της Τ.Ν. όπως τα ασαφή συστήματα, τα νευρωνικά δίκτυα, οι γενετικοί αλγόριθμοι αλλά και το Reinforcement Learning (RL) που είναι και το αντικείμενο της παρούσας εργασίας.

2.1 Κλασσικές Τεχνικές Ελέγχου Οδικής Κυκλοφορίας

2.1.1 MAXBAND

Το MAXBAND ως σύστημα ελέγχου μιας αρτηρίας δύο κατευθύνσεων πρωτοαναπτύχθηκε από τον Little (Little, 1966) και είχε ως βασικό στόχο την ρύθμιση των φαναριών κατά μήκος της αρτηρίας με τέτοιο τρόπο ώστε τα διερχόμενα οχήματα να κινούνται χωρίς να εμποδίζονται από κόκκινους σηματοδότες. Αυτό επιτυγχάνεται με την κατάλληλη ρύθμιση της καθυστέρησης (offset) κάθε φαναριού ώστε τα οχήματα που κινούνται στις προβλεπόμενες ταχύτητες να συναντούν το λεγόμενο «πράσινο ρεύμα» (green wave). Επέκταση της μεθόδου αυτής είναι το MULTIBAND (N. H. Gartner, 1991), (Gartner, 1996) το οποίο λαμβάνει υπόψη και παραμέτρους όπως αριστερές στροφές και διαφορετικό εύρος ζώνης για διαφορετικά σημεία της αρτηρίας.

2.1.2 TRANSYT

Το TRANSYT (TRAffic Network StudY Tool) αναπτύχθηκε αρχικά από τον Robertson (Robertson, 1969) και είναι ένα από τα πιο ευρέως χρησιμοποιούμενα off-line συστήματα διαχείρισης οδικής κυκλοφορίας (Papageorgiou, 2003). Χρησιμοποιεί κατά κύριο λόγο ευριστικές μεθόδους όπως τον αλγόριθμο hill-climbing για να βελτιστοποιήσει τις παραμέτρους του οδικού δικτύου συνολικά. Συγκεκριμένα λαμβάνει ως είσοδο ιστορικά δεδομένα κίνησης για το δίκτυο και προσπαθεί να βρει τον καλύτερο συνδυασμό τμηματοποίησης του χρόνου πρασίνου (green time) κάθε οδικού ρεύματος αλλά και της καθυστέρησης (offset) μεταξύ διαδοχικών διασταυρώσεων. Μπορούν επίσης να βελτιστοποιηθούν και άλλες παράμετροι του δικτύου με τον ίδιο τρόπο, όπως η μέση κατανάλωση καυσίμου των οχημάτων και οι εκπομπές περιβαλλοντολογικών ρύπων. Κάποια από τα μειονεκτήματα του συστήματος αυτού είναι ότι οι λύσεις στις οποίες συγκλίνει δεν είναι εγγυημένα οι

βέλτιστες λόγω των ευριστικών τεχνικών που χρησιμοποιούνται, καθώς και το γεγονός ότι δεν μπορούν να αντιμετωπιστούν καλά περιπτώσεις όπου έχει επέλθει κορεσμός στο οδικό δίκτυο.

2.1.3 SCOOT

Το SCOOT (Split, Cycle and Offset Optimization Technique)[6] ανήκει επίσης στα πιο διαδεδομένα και ευρέως χρησιμοποιούμενα συστήματα on-line διαχείρισης της οδικής κυκλοφορίας καθώς έχει χρησιμοποιηθεί σε πάνω από 200 πόλεις σε 14 χώρες παράγοντας σημαντικά αποτελέσματα μειωμένης κυκλοφοριακής συμφόρησης και καθυστέρησης των οχημάτων. Η βασική του λειτουργία συνοψίζεται στην πραγματικού-χρόνου λήψη δεδομένων για την είσοδο οχημάτων σε κάθε ρεύμα μιας διασταύρωσης, μετατροπή των δεδομένων αυτών σε εκτιμώμενες ροές σε μελλοντική χρονική στιγμή που τα οχήματα αυτά θα πλησιάζουν τις ήδη υπάρχουσες ουρές και χρήση τριών συστημάτων βελτιστοποίησης του χρόνου πρασίνου, της καθυστέρησης αλλά και του συνολικού κύκλου (cycle time) των φαναριών της εν λόγω διασταύρωσης. Η μέθοδος για αυτές τις βελτιστοποιήσεις είναι παρόμοια με αυτή που χρησιμοποιείται στο TRANSYT με την διαφορά ότι χρησιμοποιούνται δεδομένα πραγματικού χρόνου αντί ιστορικά δεδομένα. Τα αποτελέσματα εφαρμογής του συστήματος SCOOT έχουν υπάρξει πολύ σημαντικά με σημαντικές βελτιώσεις σε διάφορες κυκλοφοριακές παραμέτρους. Ορισμένα παραδείγματα[7] είναι τα εξής:

- Στο Λονδίνο παρατηρήθηκε μείωση 8% στον μέσο χρόνο ταξιδιού των αυτοκινήτων, 19% στην μέση καθυστέρηση και 5% στο μέσο αριθμό στάσεων
- Στο Πεκίνο το σύστημα διαχείρισης της κυκλοφορίας συντονίζει και την κίνηση των ποδηλάτων. Τα αποτελέσματα κατά τις ώρες αιχμής της χρήσης ποδηλάτου (07:00-08:00) είναι 41% μείωση στην καθυστέρηση και 26% μείωση στον αριθμό

[6] http://www.scoot-utc.com/documents/1_SCOOT-UTC.pdf
[7] http://www.scoot-utc.com/GeneralResults.php?menu=Results

των στάσεων, ποσοστά που μετατρέπονται σε 32% και 33% αντίστοιχα για τις ώρες αιχμής των αυτοκινήτων (08:00-09:00)

- Στο Τορόντο οι μειώσεις καθυστέρησης και αριθμού στάσεων ήταν 17% και 22% αντίστοιχα, ενώ υπολογίστηκε μια μείωση της μέσης κατανάλωσης καυσίμου 5,7%.

Ένα μειονέκτημα της τεχνικής αυτής είναι ότι οι αισθητήρες που χρησιμοποιούνται για τις μετρήσεις των οχημάτων τοποθετούνται συνήθως 100 – 300 μέτρα πριν την διασταύρωση με συνέπεια να μην μπορούν να προβλεφθούν εγκαίρως φαινόμενα συμφόρησης (Richter, 2006).

2.2 Τεχνικές Τεχνητής Νοημοσύνης για τον Έλεγχο Οδικής Κυκλοφορίας

2.2.1 Ασαφής Λογική

Σε αρκετές περιπτώσεις έχουν χρησιμοποιηθεί τεχνικές ασαφούς λογικής για τον έλεγχο της κυκλοφορίας σε μια διασταύρωση (Yulianto, 2003) αλλά και περισσότερων διαδοχικών διασταυρώσεων (Bien, 1995) με πολύ καλά αποτελέσματα. Χρησιμοποιούνται ασαφή σύνολα για να αναπαραστήσουν την ροή των οχημάτων και να ληφθούν αποφάσεις μέσω κανόνων για τον χρόνο πρασίνου. Σε πιο σύνθετα δίκτυα, οι ασαφείς ελεγκτές χρησιμοποιούν και πληροφορίες από την προηγούμενη και την επόμενη διασταύρωση που ελέγχουν ώστε να επιτύχουν μία καλύτερη απόδοση στο σύνολο του εξεταζόμενου οδικού δικτύου (Bien, 1995). Αν και η τεχνική αυτή αποδεικνύεται πιο προσαρμοστική από τα φανάρια σταθερού χρονισμού παρουσιάζεται δυσκολία στον αποτελεσματικό έλεγχο μεγαλύτερων δικτύων καθώς

αυτά παρουσιάζουν πολυπλοκότητα που είναι πολύ δύσκολο να αναπαρασταθεί από ποιοτικές μεταβλητές (Liu, 2007).

2.2.2 Νευρωνικά Δίκτυα

Τα Νευρωνικά δίκτυα χρησιμοποιούνται κυρίως σε συνδυασμό με ασαφή λογική ή άλλες τεχνικές καθώς έχουν το πλεονέκτημα της μη-γραμμικότητας, της αυτό-οργάνωσης και αυτό-βελτίωσης. Ένα παράδειγμα χρήσης τους είναι το σύστημα PRODYN (Liu, 2007) όπου το νευρωνικό δίκτυο χρησιμοποιείται για να εξάγει τους κανόνες της ασαφούς λογικής και να βελτιώσει την ακρίβεια του ασαφούς ελεγκτή. Συγκεκριμένα χρησιμοποιούνται δύο διαφορετικά νευρωνικά δίκτυα, τα οποία εναλλάξ εκπαιδεύονται και εξάγουν συμπεράσματα κατά τη διάρκεια της εκπαίδευσης, ενώ στην πορεία αναλαμβάνουν χρέη ελεγκτή του δικτύου. Αν και τα νευρωνικά δίκτυα έχουν χρησιμοποιηθεί και σε άλλες περιπτώσεις για την επίλυση του προβλήματος του ελέγχου της κυκλοφορίας, παρατηρείται δυσκολία να γενικεύσουν σε μεγάλα δίκτυα ώστε να εφαρμοστούν σε πραγματικές συνθήκες (Minoarivelo, 2009).

2.2.3 Λοιπές Τεχνικές

Κατά καιρούς έχουν χρησιμοποιηθεί πολλές ακόμα τεχνικές της τεχνητής νοημοσύνης για το πρόβλημα του ελέγχου της οδικής κυκλοφορίας. Κάποιες από αυτές είναι τα Συστήματα Αυτό-Οργάνωσης (Gershenson C. , 2005) (Gershenson C. R., 2012) (Wang, 2013) και οι Γενετικοί Αλγόριθμοι (Ceylan, 2004).

Το αυτοοαργανούμενο σύστημα χρησιμοποιεί ένα αποκεντρωμένο σχήμα βελτιστοποίησης, το οποίο επιτρέπει τον συνολικό συντονισμό των ροών των οχημάτων στο οδικό δίκτυο. Σε κάθε διασταύρωση του δικτύου τα φανάρια ελέγχονται από έναν πράκτορα που βασίζει τις αποφάσεις του στα δεδομένα των μετρήσεων που

λαμβάνει. Η αποκεντρωμένη τοπική βελτιστοποίηση παρέχει επαρκεί συντονισμό μεταξύ των διαφορετικών πρακτόρων και επιτυγχάνει φαινόμενα όπως πράσινο κύμα (green wave). Το αυτοοργανούμενο σύστημα μπορεί να προσαρμοστεί σε πραγματικού χρόνου δεδομένα κίνησης χωρίς την χρήση προαποφασισμένων τακτικών διαχείρισης της κυκλοφορίας που έχουν εξαχθεί από ιστορικά δεδομένα (Płaczek, 2014).

Η πρώτη χρήση γενετικών αλγορίθμων για την βελτιστοποίηση των κυκλοφοριακών σημάτων έγινε από τους Foy et al. (Foy, 1992) στην μελέτη του οποίου χρησιμοποιήθηκε ένα δίκτυο τεσσάρων διασταυρώσεων με σταθερές ροές οχημάτων. Ως ρητές μεταβλητές απόφασης θεωρήθηκαν οι χρονισμοί των πράσινων σημάτων και ο συνολικός χρόνος κύκλου των φαναριών (κοινός για όλες τις διασταυρώσεις), ενώ έμμεσες μεταβλητές αποτέλεσαν οι καθυστερήσεις (offsets) μεταξύ των διασταυρώσεων. Τα αποτελέσματα έδειξαν βελτίωση με την χρήση των γενετικών αλγορίθμων αν και δεν συγκρίθηκαν με τα τότε υπάρχοντα εργαλεία βελτιστοποίησης της οδικής κυκλοφορίας. Μια πιο σύγχρονη μελέτη έγινε από τους Teklu et al. (Teklu, 2007) στην οποία χρησιμοποιείται σαν συνάρτηση ικανότητας (fitness function) ο συνολικός χρόνος ταξιδιού μέσα στο δίκτυο. Η μέθοδος εφαρμόστηκε σε μια μελέτη πάνω στην πόλη Chester του Ηνωμένου Βασιλείου και τα αποτελέσματα έδειξαν βελτίωση στην απόδοση του δικτύου με την χρήση του γενετικού αλγορίθμου σε σύγκριση με μεθόδους που δεν υλοποιούν ανακατεύθυνση (rerouting) της κίνησης. Ιδιαίτερη βελτίωση παρουσιάστηκε σε καταστάσεις υψηλής συμφόρησης γεγονός ιδιαίτερα χρήσιμο, αφού αυτές οι καταστάσεις είναι το σημείο στο οποίο μειονεκτούν σημαντικά οι κλασσικές τεχνικές.

Αυτές οι τεχνικές έχουν παράγει σημαντικά αποτελέσματα και παρουσιάζουν πολλά πλεονεκτήματα έναντι των παραδοσιακών, όπως ότι δεν απαιτούν την χρήση μοντέλου για το οδικό δίκτυο.

ΚΕΦΑΛΑΙΟ 3ο – ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ (REINFORCEMENT LEARNING)

Η ενισχυτική μάθηση είναι μια μέθοδος της τεχνητής νοημοσύνης που απαντάται πολύ συχνά σε πλήθος προβλημάτων. Η βασική ιδέα στην οποία βασίζεται πηγάζει σε ένα μεγάλο ποσοστό από την ανθρώπινη συμπεριφορά. Η επιβράβευση ή η αποθάρρυνση ύστερα από μία ενέργεια διαμορφώνει σταδιακά πεποιθήσεις του ανθρώπου για το ποια ενέργεια είναι «καλή» και «αποδοτική» ή «κακή» και πρέπει να αποφεύγεται.

Με το ίδιο σκεπτικό ορίζεται ως πράκτορας το πρόγραμμα εκείνο το οποίο αξιολογεί την κατάσταση στην οποία βρίσκεται (state) και αποφασίζει να ενεργήσει (action) με τον τρόπο που θα του αποδώσει το μέγιστο προσδοκώμενο όφελος στο μέλλον (expected reward). Αφού εκτελεστεί η ενέργεια, ο πράκτορας λαμβάνει κάποιο reward (ή penalty) ανάλογα με το πόσο καλό ήταν το action που επέλεξε, δεδομένης της κατάστασης που βρισκόταν, και προσαρμόζει αναλόγως την προσδοκία του για το όφελος της συγκεκριμένης ενέργειας στο μέλλον.

Το μεγάλο πλεονέκτημα αυτής της προσέγγισης είναι ότι δεν απαιτείται επίβλεψη από κάποιον παρατηρητή και η συνεχής τροφοδοσία του πράκτορα με παραδείγματα σωστών ενεργειών για κάθε κατάσταση στην οποία μπορεί τυχόν να βρεθεί. Αυτό που επαρκεί είναι μία συνάρτηση επιβράβευσης (reward function) η οποία θα καθορίζει πόσο καλά απέδωσε ο πράκτορας με βάση κάποια προκαθορισμένα κριτήρια. Παραδείγματα τέτοιων συναρτήσεων στο πρόβλημα του ελέγχου της οδικής κυκλοφορίας είναι η μέση ταχύτητα των οχημάτων του δικτύου, το μέσο μήκος ουρών στα ρεύματα που οδηγούν σε μια διασταύρωση (penalty) κ.α.

Πιο αναλυτικά οι μεθόδων για το reinforcement learning απαρτίζονται συνήθως από τα ακόλουθα στοιχεία:

- S : Ένα σύνολο καταστάσεων (States)
- A : Ένα σύνολο ενεργειών (Actions)
- $T : S \times A \rightarrow \Pi(S)$ Μία συνάρτηση μετάβασης (transition function) που υπολογίζει την πιθανότητα μετάβασης από ένα state πραγματοποιώντας ένα action σε ένα νέο state
- $R : S \times A \rightarrow \mathbb{R}$ Μία συνάρτηση επιβράβευσης (reward function) που υπολογίζει το reward για κάθε συνδυασμό state, action
- $\pi : S \rightarrow A$ Μια πολιτική η οποία είναι μια συνάρτηση που υποδεικνύει πιο action πρέπει να γίνει σε κάθε state

Ο κύριος στόχος του reinforcement learning είναι να υπολογιστεί η βέλτιστη πολιτική π* η οποία και μεγιστοποιεί τα συνολικά rewards κατά την λειτουργία του agent.

3.1 MDP (Markov Decision Process)

Σε μια μεγάλη πλειοψηφία των περιπτώσεων θεωρείται ότι το περιβάλλον είναι Μαρκοβιανό το οποίο σημαίνει ότι ισχύει η ιδιότητα του Markov. Αυτό σημαίνει ότι η μετάβαση στο επόμενο state ύστερα από την πραγματοποίηση ενός action εξαρτάται μόνο από το action αυτό και το παρόν state στο οποίο βρίσκεται ο agent. Δεν υπάρχει δηλαδή εξάρτηση από τα προηγούμενα states στα οποία βρέθηκε ο agent. Αυτή η ιδιότητα, ακόμα και όταν ισχύει κατά προσέγγιση, απλοποιεί πολύ το μοντέλο του περιβάλλοντος και διευκολύνει τους υπολογισμούς με σχετικά απλούς αλγορίθμους.

Η ιδιότητα αυτή μπορεί να γραφτεί ως εξής:

$$P\{S_{t+1} = s', r_{t+1} = r' | S_t, a_t, r_t, S_{t-1}, a_{t-1}, r_{t-1}, \dots\} = P\{S_{t+1} = s', r_{t+1} = r' | S_t, a_t, r_t,\} \quad (1)$$

Όπου το πρώτο μέλος της εξίσωσης εκφράζει την πιθανότητα να μεταβεί ο agent στο state s' και να λάβει reward r' δεδομένου ότι την χρονική στιγμή t βρίσκεται στο state S_t, πραγματοποιεί το action a_t και λαμβάνει reward r_t, την χρονική στιγμή t-1 βρισκόταν στο state S_{t-1} πραγματοποίησε το action a_{t-1} και έλαβε reward r_{t-1} κ.ο.κ. ενώ το δεύτερο μέλος υποδηλώνει ότι αυτή η πιθανότητα είναι ίση με το να μεταβεί ο agent στο state s' και να λάβει reward r' δεδομένου απλά ότι την χρονική στιγμή t βρίσκεται στο state S_t, πραγματοποιεί το action a_t και λαμβάνει reward r_t.

Σύμφωνα με τα παραπάνω ένα MDP μπορεί να οριστεί ως τετράδα (S,A,T,R) όπως αυτά ορίστηκαν παραπάνω.

3.1.1 Επανάληψη Αξιών (Value Iteration)

Ως αξία ή χρησιμότητα ενός state ορίζουμε την αναμενόμενη συνολική ανταμοιβή του πράκτορα ενώ βρίσκεται σε αυτό το state, αν επιλέξει actions με βάση την πολιτική που ακολουθεί την στιγμή αυτή. Επειδή στα περισσότερα προβλήματα ο χρονικός ορίζοντας δράσης είναι άπειρος (infinite horizon MDP) η συνολική αυτή πιθανότατα θα απειρίζεται. Για αυτόν τον λόγο σταθμίζουμε τα μελλοντικά rewards με έναν παράγοντα γ^k όπου k η χρονική απόσταση από την παρούσα στιγμή. Προφανώς, για να αποφευχθεί ο απειρισμός οφείλει να ισχύει $\gamma < 1$. Έτσι για την συνάρτηση αξίας έχουμε:

$$U(s) = E\left[\sum_{k=0}^{\infty} \gamma^k r_k \mid \pi,\ s_o = s\right] \qquad (2)$$

Ορίζουμε λοιπόν την συνάρτηση επιβράβευσης (reward function) ως εξής:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \qquad (3)$$

Συνεπώς έχουμε:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} = r_{t+1} + \sum_{k=1}^{\infty} \gamma^k r_{t+1+k} =$$

$$= r_{t+1} + \sum_{k=0}^{\infty} \gamma^{k+1} r_{t+1+k+1} = r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+1+k+1}$$

$$= r_{t+1} + \gamma R_{t+1} \qquad (4)$$

όπου παρατηρούμε την αναδρομική φύση της συνάρτησης επιβράβευσης.

Σύμφωνα με τα παραπάνω, η εξίσωση (2) γίνεται:

$$U(s) = E[R_t \mid \pi,\ s_o = s] = E[r_{t+1} + \gamma R_{t+1} \mid \pi,\ s_o = s]$$

$$= E[r_{t+1} \mid \pi,\ s_o = s] + \gamma E[R_{t+1} \mid \pi,\ s_o = s]$$

$$R(s, \pi(s)) + \gamma E[R_{t+1} \mid \pi,\ s_o = s] \qquad (5)$$

αφού το R(s,π(s)) είναι το reward που λαμβάνει ο πράκτορας στο state s ακολουθώντας το action που του επιβάλει η πολιτική π.

Για τον υπολογισμό του δεύτερου προσθετέου του δεύτερου μέλους της εξίσωσης (5) πρέπει να βρεθεί η αναμενόμενη ανταμοιβή κατά την επόμενη κατάσταση στην οποία θα βρεθεί ο πράκτορας. Για αυτόν τον σκοπό θα χρησιμοποιήσουμε την συνάρτηση μετάβασης T(s,a,s') που υποδηλώνει την πιθανότητα να μεταβεί ο πράκτορας σε μια επόμενη κατάσταση δεδομένης της παρούσας κατάστασης του και ενός action. Έτσι έχουμε:

$$E[R_{t+1} \mid \pi,\ s_o = s] = \max_a \sum_{s'} T(s,a,s')\, E[R_{t+1} \mid \pi,\ s_o = s']$$

αφού υποθέτουμε πως ο πράκτορας θα επιλέξει το action με την μέγιστη αναμενόμενη ανταμοιβή. Συνεπώς η (5) γίνεται:

$$U(s) = R(s,\pi(s)) + \gamma \max_a \sum_{s'} T(s,a,s')\, U(s') \qquad (6)$$

Η εξίσωση (6) ονομάζεται εξίσωση Bellman και σε αυτήν βασίζεται η μέθοδος της επανάληψης αξιών. Με βάση τα παραπάνω, η εύρεση της πολιτικής π* η οποία μεγιστοποιεί τις αναμενόμενες ανταμοιβές παίρνει την μορφή:

$$\pi^*(s) = \operatorname*{argmax}_a \left[R(s,a) + \gamma \sum_{s'} T(s,a,s')\, U(s') \right] \qquad (7)$$

Για να μπορέσει να υπολογιστεί η πολιτική αυτή πρέπει να είναι γνωστές οι τιμές U(s), πρόβλημα το οποίο είναι κατ' ουσίαν ένα σύστημα ν-εξισώσεων με ν-αγνώστους. Το πρόβλημα στη διαδικασία αυτή είναι ότι ο τελεστής max είναι μη γραμμικός και συνεπώς το σύστημα είναι μη γραμμικό και η επίλυση με μεθόδους της γραμμικής

άλγεβρας δεν ενδείκνυται. Συνεπώς χρησιμοποιείται η ακόλουθη επαναληπτική διαδικασία.

Αλγόριθμος 1: Επανάληψη Αξιών (Value Iteration)

Input: Το σύνολο καταστάσεων **S** (states), το σύνολο ενεργειών **A** (actions), η συνάρτηση ανταμοιβής **R(s,a)**, η συνάρτηση μετάβασης **T(s,a,s')**, ο παράγοντας προεξόφλησης (discount rate) **γ** και **ε** ένας πολύ μικρός αριθμός

$U_0(s) = 0, \forall s \in S$ (ή κάποια άλλη αρχικοποίηση)

While $|U_k(s) - U_{k-1}(s)| < \varepsilon, \ \forall s \in S$ **Do**
  For all s in S Do
    For all a in A Do
      $U_{k+1}(s) = \max_a [R(s,a) + \gamma \sum_{s'} T(s,a,s')\, U_k(s')]$
    End_For
  End_For
End_While

return $U(s)$

Όταν η μεταβολή των τιμών της συνάρτησης χρησιμότητας γίνει επαρκώς μικρή, λέμε ότι ο αλγόριθμος έχει συγκλίνει και χρησιμοποιώντας την εξίσωση (7) μπορούμε να εξάγουμε την βέλτιστη πολιτική σύμφωνα με την οποία ο πράκτορας θα ενεργεί με την μέγιστη δυνατή ανταμοιβή.

3.2 SARSA (State-Action-Reward, State-Action)

Η παραπάνω ανάλυση, αν και ιδιαίτερα χρήσιμη, έχει έναν σημαντικό περιορισμό. Απαιτεί γνώση του μοντέλου του περιβάλλοντος, δηλαδή την συνάρτηση μετάβασης T. Αρκετές φορές όμως τα προβλήματα τα οποία καλείται να λύσει η μέθοδος της ενισχυτικής μάθησης – και ο έλεγχος της οδικής κυκλοφορίας είναι ένα από αυτά – δεν

δίνουν την πληροφορία αυτή. Συγκεκριμένα στο πρόβλημα της εργασίας αυτής, δεν είναι εκ των προτέρων γνωστό πως θα επιδράσει το οδικό δίκτυο μία απόφαση για αλλαγή ρύθμισης του φωτεινού σηματοδότη και συνεπώς δεν μπορούμε να γνωρίζουμε την κατανομή πιθανότητας για την επόμενη κατάσταση στην οποία θα βρεθεί.

Για τέτοιες περιπτώσεις είναι χρήσιμη μία διαφορετική προσέγγιση, η οποία αξιοποιεί τις εμπειρίες που λαμβάνει ο πράκτορας καθώς δρα (on-line), για να εκπαιδευτεί και να βρει την βέλτιστη πολιτική.

Ένας αλγόριθμος που λειτουργεί με αυτόν τον τρόπο είναι ο SARSA. Ο τρόπος που λειτουργεί είναι ο ακόλουθος: ο πράκτορας βρίσκεται στο state s, πραγματοποιεί action a, λαμβάνει reward r, μεταβαίνει στο state s' όπου θα πραγματοποιήσει action α' με βάση την πολιτική που ακολουθεί. Η πεντάδα (s, a, r, s, a') είναι που δίνει και το όνομά της στον αλγόριθμο. Η τιμή που υπολογίζεται σε κάθε βήμα είναι η αξία ενεργειών σε κάθε κατάσταση (state – action value function Q(s, a)) αντί της συνάρτησης χρησιμότητας που αναλύθηκε στο 3.1. Ο υπολογισμός της γίνεται ως εξής:

Αλγόριθμος 2: SARSA

Input: Το σύνολο καταστάσεων **S** (states), το σύνολο ενεργειών **A** (actions), ο παράγοντας προεξόφλησης (discount rate) **γ** και **α** ο ρυθμός μάθησης

$Q(s,a) = random, \forall s \in S$ (ή κάποια άλλη αρχικοποίηση ανάλογα με το πρόβλημα)
Repeat (για κάθε επεισόδιο):
Επίλεξε ένα αρχικό state s
Επίλεξε ένα action a με βάση την πολιτική που εξάγεται από το Q (π.χ. μέγιστη αξία)
Repeat Until s είναι τερματικό state
Πραγματοποίησε το action a και παρατήρησε το reward r και το επόμενο state s'
Επίλεξε ένα action a' με βάση την πολιτική που εξάγεται από το Q (π.χ. μέγιστη αξία)
$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$
$s \leftarrow s', a \leftarrow a'$

Return $Q(s,a)$

Σε κάθε βήμα, το σφάλμα $r + \gamma Q(s', a') - Q(s, a)$ σταθμίζεται με τον παράγοντα α και ενημερώνει την τιμή $Q(s, a)$. Ο παράγοντας α ονομάζεται ρυθμός μάθησης επειδή όσο μεγαλύτερη είναι η τιμή του, τόσο πιο δραστικά αλλάζουν οι τιμές της συνάρτησης Q. Μία τακτική που συχνά ακολουθείται είναι η παράμετρος αυτή να μειώνεται καθώς η διαδικασία εκπαίδευσης προχωράει και θεωρούμε πως η γνώση των τιμών Q γίνεται σταδιακά αξιόπιστη. Ο αλγόριθμος SARSA μαθαίνει την συνάρτηση Q με βάση την πολιτική που ακολουθεί και στην συνέχεια ενημερώνει την πολιτική αυτή με τις νέες τιμές που υπολόγισε. Με αυτόν τον τρόπο προσεγγίζει την βέλτιστη πολιτική. Ακριβώς επειδή η πολιτική που ακολουθεί ο πράκτορας επηρεάζει τον υπολογισμό των τιμών της συνάρτησης Q , λέγεται πως ο αλγόριθμος SARSA είναι μία on-policy μέθοδος.

3.3 Q-Learning

Ένας από τους αλγορίθμους που έχει παίξει καθοριστικό ρόλο στον τομέα της ενισχυτικής μάθησης είναι ο αλγόριθμος Q-learning ο οποίος προτάθηκε από τον Watkins το 1989 (Watkins, 1992). Η κυριότερη διαφορά του από τον SARSA είναι πως η πολιτική η οποία ακολουθεί ο πράκτορας σε κάθε βήμα δεν επηρεάζει τον υπολογισμό της συνάρτησης Q. Για αυτό το Q-learning αναφέρεται και ως on-line, off-policy μάθηση. Η ενημέρωση των τιμών Q πραγματοποιείται ως εξής:

Αλγόριθμος 3: Q-Learning

Input: Το σύνολο καταστάσεων **S** (states), το σύνολο ενεργειών **A** (actions), ο παράγοντας προεξόφλησης (discount rate) **γ** και **α** ο ρυθμός μάθησης

$Q(s, a) = random, \forall s \in S$ (ή κάποια άλλη αρχικοποίηση ανάλογα με το πρόβλημα)
Repeat (για κάθε επεισόδιο):
  Επίλεξε ένα αρχικό state s
  Repeat Until s είναι τερματικό state
    Πραγματοποίησε το action a και παρατήρησε το reward r και το επόμενο state s'

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$
$$s \leftarrow s'$$

Return $Q(s,a)$

Σε κάθε βήμα ο αλγόριθμος ενημερώνει τις τιμές Q με το σφάλμα

$$r + \gamma \max_{a'} Q(s',a') - Q(s,a)$$

στο οποίο έχει υπολογίζεται το προσδοκώμενο μελλοντικό όφελος με βάση το action a' που οδηγεί στην μεγιστοποίηση του, ανεξαρτήτως αν εν τέλει ο πράκτορας θα πραγματοποιήσει την ενέργεια αυτή. Αυτό διευκολύνει την σύγκλιση στην βέλτιστη πολιτική, καθώς ο πράκτορας μπορεί να εξερευνεί actions που οδηγούν σε κακά rewardsχωρίς να επηρεάζει αυτό τους υπολογισμούς της τιμής Q για το state από το οποίο ξεκίνησε.

Οι βασικές παράμετροι που παίζουν σημαντικό ρόλο στο πόσο καλά θα αποδώσει ο αλγόριθμος είναι ο ρυθμός μάθησης α, ο παράγοντας προεξόφλησης γ και η αρχικοποίηση των τιμών Q.

Ο ρυθμός μάθησης (learning rate) α μπορεί να πάρει τιμές στο διάστημα [0, 1]. Η τιμή 0 συνεπάγεται πως οι τιμές Q παραμένουν σταθερές και δεν ενημερώνονται κατά το τρέξιμο του αλγορίθμου. Η τιμή 1 αντίθετα κάνει τον πράκτορα να λαμβάνει υπόψη του μόνο τις πιο πρόσφατες παρατηρήσεις αγνοώντας το παρελθόν.

Ο παράγοντας προεξόφλησης (discount rate) καθορίζει σε ποιόν βαθμό οι μελλοντικές ανταμοιβές λαμβάνονται υπόψη κατά την επιλογή ενός action. Για να είναι εγγυημένη η σύγκλιση η τιμή του πρέπει να είναι μικρότερη του 1. Αν η τιμή του πλησιάζει την μονάδα τότε ο πράκτορας επιλέγει ενέργειες που θα του αποφέρουν μεγάλες ανταμοιβές στο μέλλον. Σε αντίθετη περίπτωση, αν λάβει τιμή 0, τότε ο πράκτορας ονομάζεται μυωπικός, καθώς υπολογίζει μόνο την ανταμοιβή που αναμένει να λάβει στο επόμενο βήμα και δεν ενδιαφέρεται για τα μελλοντικά rewards.

Τέλος, η αρχικοποίηση των τιμών Q μπορεί να παίξει καθοριστικό ρόλο στην ταχύτητα σύγκλισης του αλγορίθμου. Μία συνηθισμένη τακτική είναι η χρήση «αισιόδοξων» αρχικών τιμών. Αυτό έχει ως αποτέλεσμα ο πράκτορας να θέλει να επιλέξει actions τα οποία δεν έχει διαλέξει στο παρελθόν καθώς η αναμενόμενη ανταμοιβή θα είναι υψηλή. Μια μέθοδος να μην παραμένουν υψηλές αρχικές εκτιμήσεις για μεγάλο χρονικό διάστημα, είναι κατά την πρώτη ενημέρωση της κάθε τιμής Q να μην χρησιμοποιείται ο κανόνας του αλγορίθμου αλλά να χρησιμοποιείται το reward που λήφθηκε ως μια νέα αρχική τιμή.

3.3.1 Exploration vs. Exploitation

Ένα άλλο πολύ σημαντικό ζήτημα που εγείρεται στην μέθοδο του Q-learning είναι η εύρεση μίας ισορροπίας ανάμεσα στην εξερεύνηση νέων actions (exploration) και στην εκμετάλλευση υψηλών rewards από την ήδη εξασφαλισμένη γνώση (exploitation). Αυτό επιτυγχάνεται κάνοντας τον πράκτορα να δρα μη-βέλτιστα κατά ένα ποσοστό των επαναλήψεων χωρίς να επιλέγει το action με το μεγαλύτερο αναμενόμενο reward. Υπάρχουν αρκετοί τρόποι να εξασφαλιστεί η μη-βέλτιστη συμπεριφορά όπως τυχαία επιλογή, επιλογή με πιθανότητα ανάλογη των υπολογισμένων τιμών Q ή τον αριθμό επισκέψεων της παρούσας κατάστασης.

3.3.1.1 ε-Greedy

Η μέθοδος ε-greedy, όπου ε είναι μια παράμετρος που παίρνει τιμές από το [0, 1] και υποδηλώνει τη πιθανότητα να επιλεγεί ένα τυχαίο action σε κάθε βήμα εκτέλεσης, είναι η πιο απλή από τις προαναφερθείσες. Σε κάθε επανάληψη του αλγορίθμου επιλέγεται η βέλτιστη ενέργεια με πιθανότητα 1 – ε. Αλλιώς ο πράκτορας επιλέγει τυχαία μία από όλες τις διαθέσιμες ενέργειες. Ο αλγόριθμος του Q-learning με την επιλογή αυτή διαμορφώνεται ακολούθως:

Αλγόριθμος 4: ε-greedy Q-Learning

Input: Το σύνολο καταστάσεων **S** (states), το σύνολο ενεργειών **A** (actions), ο παράγοντας προεξόφλησης (discount rate) **γ**, ο ρυθμός μάθησης **α** και η πιθανότητα τυχαίας επιλογής **ε**

$Q(s,a) = random, \forall s \in S$ (ή κάποια άλλη αρχικοποίηση ανάλογα με το πρόβλημα)
Repeat (για κάθε επεισόδιο):
  Επίλεξε ένα αρχικό state s
  Repeat Until s είναι τερματικό state
    Επίλεξε τυχαίο αριθμό p από Ομοιόμορφη_Κατανομή(0, 1)
    If p < ε
      $a = \underset{a'}{\text{argmax}}\, Q(s,a')$
    Else
      $a = random(Q(s,a))$

    Πραγματοποίησε το action a και παρατήρησε το reward r και το επόμενο state s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$
    $s \leftarrow s'$
Return $Q(s,a)$

Η παραπάνω μέθοδος εξερεύνησης των άγνωστων actions ονομάζεται και τυχαία εξερεύνηση (random exploration), με την έννοια ότι κατά την φάση της τυχαίας επιλογής ενέργειας ο πράκτορας ισοπίθανα μπορεί να πραγματοποιήσει οποιοδήποτε action είναι διαθέσιμο από την κατάσταση στην οποία βρίσκεται.

3.3.1.2 Utility-driven Exploration

Μία άλλη μέθοδος είναι η «οδηγούμενη από την χρησιμότητα εξερεύνηση» (Utility-driven Exploration). Σε αυτή την παραλλαγή, η πιθανότητα να επιλεγεί μία ενέργεια είναι ανάλογη της τιμής Q που έχει υπολογιστεί μέχρι εκείνη την στιγμή για εκείνη την ενέργεια. Συχνά για αυτόν τον σκοπό χρησιμοποιείται η κατανομή Boltzmann όπου η πιθανότητα να επιλεγεί ένα action **α** είναι (Thrun, 1992):

$$P(s,a) = \frac{e^{Q(s,a)\tau^{-1}}}{\sum_{a' in\ A} e^{Q(s,a')\tau^{-1}}} \quad (8)$$

Ο συντελεστής **τ** ονομάζεται θερμοκρασία και όσο η τιμή πλησιάζει στο 0 ο πράκτορας τείνει να επιλέξει την ενέργεια **α** με την μεγαλύτερη τιμή Q. Αντίθετα όταν το **τ** τείνει στο άπειρο, η κατανομή πλησιάζει την ομοιόμορφη κατανομή, και συνεπώς ο πράκτορας πραγματοποιεί αποκλειστικά εξερεύνηση.

Αυτή η μέθοδος εξασφαλίζει την παράλληλη εξερεύνηση και εκμετάλλευση καθ' όλη την διάρκεια της εκπαίδευσης (Thrun, 1992). Ο αλγόριθμος του Q-learning τροποποιείται ακολούθως:

Αλγόριθμος 5: Q-Learning with Utility-driven Exploration Probability Distributions

---

Input: Το σύνολο καταστάσεων **S** (states), το σύνολο ενεργειών **A** (actions), ο παράγοντας προεξόφλησης (discount rate) **γ**, ο ρυθμός μάθησης **α** και η θερμοκρασία **τ**

$Q(s,a) = random, \forall s \in S$ (ή κάποια άλλη αρχικοποίηση ανάλογα με το πρόβλημα)
Αρχικοποίηση των τιμών P για κάθε (s, a): $P(s,a) = \frac{e^{Q(s,a)\tau^{-1}}}{\sum_{a' in\ A} e^{Q(s,a')\tau^{-1}}}$
Repeat (για κάθε επεισόδιο):
  Επίλεξε ένα αρχικό state s
  Repeat Until s είναι τερματικό state
    Επίλεξε μια ενέργεια α με βάση την κατανομή πιθανότητας P
    Πραγματοποίησε το action a και παρατήρησε το reward r και το επόμενο state s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$
    Ενημέρωση των τιμών P για κάθε (s, a): $P(s,a) = \frac{e^{Q(s,a)\tau^{-1}}}{\sum_{a' in\ A} e^{Q(s,a')\tau^{-1}}}$
    $s \leftarrow s'$
Return $Q(s,a)$

3.3.1.3 Εξερεύνηση με μετρητή

Μια διαφορετική προσέγγιση που επιχειρεί να εξασφαλίσει την αποτελεσματική εξερεύνηση όλου του χώρου καταστάσεων είναι να χρησιμοποιηθεί ένας μετρητής c(s) για κάθε κατάσταση που να μετράει τον αριθμό των φορών που ο πράκτορας την έχει επισκεφθεί. Χρησιμοποιώντας έναν γραμμικό συνδυασμό του μετρητή αυτού και των τιμών Q λαμβάνουμε μια νέα μετρική για τον υπολογισμό της αξίας κάθε ζεύγους (s, a):

$$Q'(s,a) = b \cdot Q(s,a) + \frac{c(s)}{\sum_{s'} T(s,a,s')c(s')} \quad (9)$$

Ο συντελεστής b καθορίζει την βαρύτητα της τιμής Q στον υπολογισμό της νέας συνάρτησης αξίας. Ο παρονομαστής του κλάσματος υποδηλώνει την αναμενόμενη τιμή του μετρητή στην κατάσταση που θα μεταβεί ο πράκτορας πραγματοποιώντας την ενέργεια α. Σε κάθε επανάληψη του αλγορίθμου επιλέγεται η ενέργεια που μεγιστοποιεί την τιμή Q'. Αυτό εποπτικά φαίνεται στον αλγόριθμο που ακολουθεί:

Αλγόριθμος 6: Q-Learning with Counter-based Exploration

Input: Το σύνολο καταστάσεων **S** (states), το σύνολο ενεργειών **A** (actions), ο παράγοντας προεξόφλησης (discount rate) **γ**, ο ρυθμός μάθησης **α** και ο συντελεστής exploitation **b**

$Q(s,a) = random, \forall s \in S$ (ή κάποια άλλη αρχικοποίηση ανάλογα με το πρόβλημα)
$Q'(s,a) = Q(s,a)$
$c(s) = 0, \quad \forall s \in S$
Repeat (για κάθε επεισόδιο):
  Επίλεξε ένα αρχικό state s
  $c(s) = c(s) + 1$
  Repeat Until s είναι τερματικό state
    $a = \underset{a'}{\text{argmax}}\, Q'(s,a')$
    Πραγματοποίησε το action a και παρατήρησε το reward r και το επόμενο state s'
    $c(s') = c(s') + 1$

$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$

Ενημέρωση των τιμών Q’ για κάθε (s, a): $Q'(s,a) = b \cdot Q(s,a) + \frac{c(s)}{\sum_{s'} T(s,a,s')c(s')}$

$s \leftarrow s'$

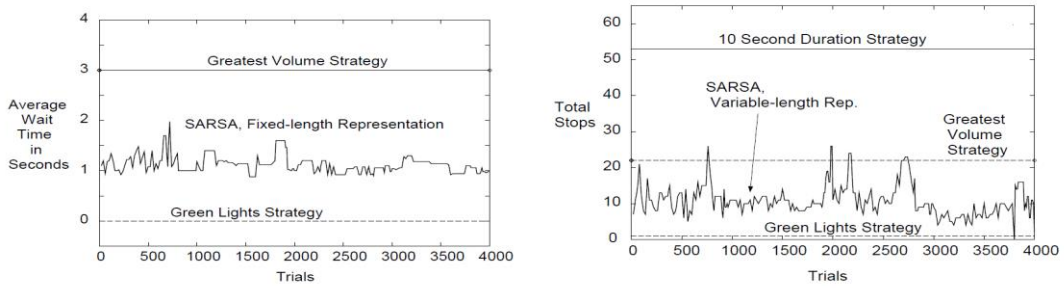
Return $Q(s,a)$

ΚΕΦΑΛΑΙΟ 4ο – ΕΝΙΣΧΥΤΙΚΗ ΜΑΘΗΣΗ ΣΤΟΝ ΕΛΕΓΧΟ ΤΗΣ ΟΔΙΚΗΣ ΚΥΚΛΟΦΟΡΙΑΣ

Η πρώτη χρήση της ενισχυτικής μάθησης για την επίλυση του προβλήματος του ελέγχου της οδικής κυκλοφορίας πραγματοποιήθηκε από τους Thorp και Anderson (T. L. Anderson, 1996) οι οποίοι χρησιμοποίησαν τον αλγόριθμο SARSA της ενισχυτικής μάθησης για να ελέγξουν την κυκλοφορία γύρω από μία απομονωμένη διασταύρωση. Ως αναπαράσταση της κατάστασης (state) χρησιμοποιήθηκαν τρεις διαφορετικές προσεγγίσεις:

- Ο αριθμός των εισερχόμενων οχημάτων στα ρεύματα της διασταύρωσης κβαντισμένος σε 10 επίπεδα.
- Η σχετική απόσταση των οχημάτων από την διασταύρωση κβαντισμένη σε 8 ισομήκη τμήματα.
- Η σχετική απόσταση των οχημάτων από την διασταύρωση κβαντισμένη σε 4 αυξανόμενου μήκους τμήματα.

Τα αποτελέσματα που εξήχθηκαν παρουσίασαν σημαντική βελτίωση σε σχέση με την χρήση φαναριών σταθερής ρύθμισης, ακόμα και αν αυτή η ρύθμιση είναι η βέλτιστη δυνατή. Παρακάτω συμπεριλαμβάνεται το γράφημα απόδοσης του αλγορίθμου SARSA:



Εικόνα 5 - Απόδοση του αλγορίθμου SARSA (T. L. Anderson, 1996)

Μια άλλη προσέγγιση του ζητήματος είναι με την χρήση Q-learning σε μία μεμονωμένη διασταύρωση (Abdulhai, 2003). Ως κατάσταση του δικτύου θεωρήθηκε το μήκος των ουρών στα εισερχόμενα στην διασταύρωση ρεύματα, καθώς και η παρούσα διάρκεια των φάσεων του φαναριού. Ως συνάρτηση επιβράβευσης (ποινής στην προκειμένη περίπτωση) θεωρήθηκε η μέση καθυστέρηση των οχημάτων ανάμεσα στις διαδοχικές αποφάσεις του ευφυούς πράκτορα. Το σύστημα χρησιμοποιώντας αυτά τα δεδομένα λαμβάνει αποφάσεις για την επέκταση της παρούσας φάσης, ή την αλλαγή στην επόμενη. Τα αποτελέσματα της μελέτης έδειξαν σημαντική υπεροχή της μεθόδου σε σχέση με την χρήση σταθερού χρονισμού σηματοδότη.

Στις παραπάνω θεωρήσεις ως ευφυής πράκτορας χρησιμοποιείται ο ελεγκτής του φωτεινού σηματοδότη της διασταύρωσης. Μια διαφορετική θεώρηση είναι να χρησιμοποιηθεί ένα πολυπρακτορικό σύστημα (Multi-agent Reinforcement Learning) (Steingrover, 2005) το οποίο βασίζει τις καταστάσεις του στα οχήματα και όχι στα φανάρια. Συγκεκριμένα κάθε πράκτορας αναφέρεται σε ένα όχημα που βρίσκεται στο δίκτυο και εκτιμά τον χρόνο αναμονής του γύρω από την διασταύρωση. Συνδυάζοντας τις καταστάσεις αυτές, μπορεί το σύστημα να λάβει αποφάσεις για τον καταμερισμό του χρόνου πρασίνου στα ανταγωνιστικά ρεύματα. Ένα πλεονέκτημα αυτής της μεθόδου είναι ότι αποφεύγεται ο πολύ μεγάλος χώρος καταστάσεων που συνοδεύει την μέθοδο που βασίζεται στις καταστάσεις με βάση το φανάρι.

Μια παράμετρος που καθιστά σχετικά δύσκολη την χρήση reinforcement learning για την επίλυση ενός προβλήματος ελέγχου κυκλοφορίας, είναι η μη στασιμότητα του περιβάλλοντος. Αλλαγές στις ροές των οχημάτων προκαλούν διαφορετικές συμπεριφορές στο δίκτυο ακόμα και στην ίδια κατάσταση. Αυτό το ζήτημα μπορεί να επιλυθεί με χρήση «συμφραζόμενων» (contexts) (de Oliveira, 2006). Το σύστημα καλείται να εντοπίσει αρχικά το context στο οποίο βρίσκεται και στην συνέχεια αντιμετωπίζει το περιβάλλον ως στάσιμο. Η εκπαίδευση πραγματοποιείται για κάθε context σε χωριστά πλαίσια και έτσι καταφέρνει να αποδώσει καλύτερα από συστήματα που αντιμετωπίζουν το πρόβλημα ως στάσιμο.

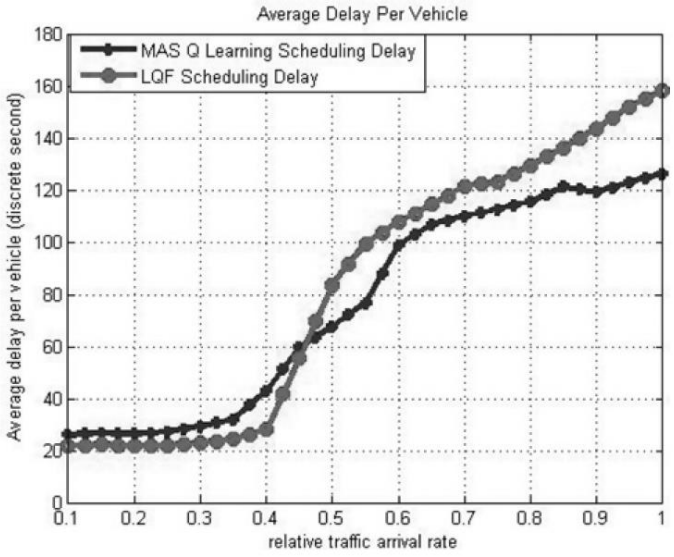
Μια πιο πρόσφατη μελέτη (Prashanth, 2011) προσπαθεί να λύσει το πρόβλημα του μεγάλου χώρου κατάστασης-ενεργειών χρησιμοποιώντας, σε συνδυασμό με την μέθοδο της ενισχυτικής μάθησης, συναρτησιακή προσέγγιση (function approximation). Για να αποφευχθεί η εκθετική αύξηση που παρατηρείται στον χώρο καταστάσεων και ενεργειών σε ένα μεγάλο δίκτυο χρησιμοποιήθηκε αναπαράσταση των καταστάσεων βασισμένη σε χαρακτηριστικά (feature-based state representation). Η σύγκριση της μεθόδου αυτής έδειξε σημαντικά πλεονεκτήματα σε σχέση με άλλες τεχνικές.

Οι Kuyer et al. (Kuyer, 2008) έδωσαν ιδιαίτερη έμφαση στην βελτιστοποίηση ενός μεγάλου δικτύου, σε αντίθεση με την πλειονότητα της ήδη υπάρχουσας βιβλιογραφίας που ασχολείτο με την τοπική βελτιστοποίηση μιας διασταύρωσης. Συμπεριέλαβαν στο πολυπρακτορικό τους σύστημα ρητά τον συντονισμό μεταξύ γειτονικών φαναριών, ο οποίος επιτυγχάνεται μέσω του αλγορίθμου max-plus ο οποίος εκτιμάει την βέλτιστη συντονισμένη ενέργεια στέλνοντας βελτιστοποιημένα μηνύματα μεταξύ των συντονιζόμενων πρακτόρων. Τα αποτελέσματα έδειξαν σημαντικά οφέλη σε σχέση με άλλες υλοποιήσεις, ειδικά σε περιπτώσεις υψηλής συμφόρησης όπου φάνηκε ότι η ανάγκη συντονισμού των πρακτόρων είναι καθοριστική.

Παρόμοια προσέγγιση με πολυπρακτορικό σύστημα πραγματοποίησαν οι Salkham et al. (Salkham, 2008) όπου χρησιμοποιήθηκε ένα προσαρμοστικό μοντέλο χρονοπρογραμματισμού εκ περιτροπής (Adaptive Round Robin) το οποίο βελτιστοποιήθηκε μέσω συντονισμένης ενισχυτικής μάθησης. Ο πράκτορας κάθε διασταύρωσης βελτιστοποίησε τους χρονισμούς των φάσεων του φαναριού βάση του μοτίβου της κίνησης σε συντονισμό με τους γειτονικούς πράκτορες. Σε σύγκριση με τον αλγόριθμο εξισορρόπησης κορεσμού (saturation balancing algorithm) βρέθηκε βελτίωση απόδοσης περίπου 57% σε μια μεγάλης κλίμακας προσομοίωση για το κέντρο του Δουβλίνου.

Ένας συνδυασμός Reinforcement-Learning με την συμβατική τεχνική LQF (Longest Queue First), η οποία δίνει πράσινο σήμα στην ουρά μεγαλύτερου μήκους, προτάθηκε από τους Arel et al.. Αντικείμενο μελέτης τους ήταν ένα σύστημα πέντε

διασταυρώσεων σε σχηματισμό «σταυρού» με τις τέσσερεις περιφερειακές διασταυρώσεις να ελέγχονται από τον συμβατικό αλγόριθμο LQF και μόνο την μεσαία να υιοθετεί ευφυή πράκτορα. Σύμφωνα με τα αποτελέσματά τους ο αλγόριθμος κατάφερε να υπολογίσει τις μη γραμμικές σχέσεις που οι συμβατικές τεχνικές αδυνατούν και οι προσομοιώσεις που πραγματοποιήθηκαν έδειξαν χαμηλότερες καθυστερήσεις στα οχήματα, ειδικά για μέτριες και υψηλές συνθήκες κίνησης. Ενδεικτικά τα ευρήματά τους συνοψίζονται στο παρακάτω διάγραμμα που δείχνει την μέση καθυστέρηση των οχημάτων για διαφορετικούς ρυθμούς άφιξης οχημάτων στη διασταύρωση.



Εικόνα 6 - (Arel, 2010)

ΚΕΦΑΛΑΙΟ 5ο –ΠΡΟΣΟΜΟΙΩΤΕΣ ΟΔΙΚΗΣ ΚΥΚΛΟΦΟΡΙΑΣ (TRAFFIC SIMULATORS)

5.1 Η αναγκαιότητα της προσομοίωσης στις Συγκοινωνίες

Η προσομοίωση ορίζεται ως η δυναμική απεικόνιση ενός τμήματος του πραγματικού κόσμου η οποία επιτυγχάνεται με την δημιουργία ενός υπολογιστικού μοντέλου και μετακινώντας το μέσα στον χρόνο (Drew, 1968). Η πρώτη χρήση προσομοίωσης της οδικής κυκλοφορίας σε υπολογιστή πραγματοποιήθηκε με την διπλωματική εργασία του D.L. Gerlough στο πανεπιστήμιο της California, Los Angeles το 1955 με θέμα «Προσομοίωση της κίνησης σε υπολογιστή γενικού σκοπού διακριτών μεταβλητών» (Kallberg, 1971). Έκτοτε έχει γίνει αντικείμενο συστηματικής μελέτης και έχουν παραχθεί διάφορα μοντέλα που προσεγγίζουν την πραγματικότητα σε μεγάλο βαθμό και επιτρέπουν την δοκιμή της πληθώρας των αλγορίθμων που προτείνονται για την επίλυση του προβλήματος του ελέγχου της οδικής κυκλοφορίας.

Η αποτελεσματική μετακίνηση των ανθρώπων και των προϊόντων στο οδικό δίκτυο είναι ένα σύνθετο και εντυπωσιακό πρόβλημα. Τα συστήματα συγκοινωνιών προσδιορίζονται από ένα σύνολο χαρακτηριστικών που τα κάνει δύσκολο να αναλυθούν, να ελεγχθούν και να βελτιστοποιηθούν. Τα συστήματα αυτά συνήθως καλύπτουν ευρεία πεδία φυσικής, ο αριθμός των συμμετεχόντων είναι υψηλός, οι στόχοι των συμμετεχόντων πολλές φορές έρχονται σε αντίφαση μεταξύ τους ή με τον ελεγκτή του ευρύτερου συστήματος και υπάρχουν πολλοί απρόβλεπτοι παράγοντες. Επιπρόσθετα τα συστήματα οδικών συγκοινωνιών είναι εγγενώς δυναμικά αφού ο αριθμός των συνιστωσών του συστήματος μεταβάλλεται με τον χρόνο και με μεγάλο επίπεδο τυχαιότητας. Ο ιδιαίτερα υψηλός αριθμός ενεργών συμμετεχόντων στο σύστημα την ίδια χρονική στιγμή συνεπάγεται έναν πολύ υψηλό αριθμό ταυτόχρονων αλληλεπιδράσεων. Τέτοιες αλληλεπιδράσεις μπορεί να είναι είτε μεταξύ ανθρώπων

είτε μεταξύ ανθρώπων και μηχανών (οχημάτων, ενδείξεις ελέγχου οδικής κυκλοφορίας) οι οποίες μάλιστα είναι διέπονται από προσεγγιστικούς «νόμους» και κανόνες, αφού, για παράδειγμα, οι αντιδράσεις των οδηγών κυριεύονται από ένα πλήθος αστάθμητων παραγόντων (κούραση, ψυχολογία, αντίληψη) και όχι από απόλυτους κανόνες.

Για όλους τους παραπάνω λόγους, τα συστήματα συγκοινωνιών αποτελούν έναν πρόσφορο τομέα για έρευνα και σχεδιασμό βασισμένο στην προσομοίωση καθώς αναλυτικά εργαλεία δεν μπορούν να εξασφαλίσουν αξιόπιστα και λεπτομερή αποτελέσματα.

5.2 Κατηγορίες Προσομοίωσης οδικής κυκλοφορίας

Το προγράμματα συγκοινωνιακής προσομοίωσης χωρίζονται σε τρεις βασικές κατηγορίες αναφορικά με το επίπεδο λεπτομέρειας της μοντελοποίησης (μακροσκοπική, μεσοσκοπική, και μικροσκοπική) καθώς και σε διακριτού ή συνεχούς χρόνου και χώρου. Η πιο λεπτομερής προσέγγιση, η μικροσκοπική, προβλέπει σε κάθε βήμα της προσομοίωσης την κατάσταση κάθε στοιχείου της (όχημα, πεζός, οδική κατάσταση, φανάρια κλπ.) με ιδιαίτερη έμφαση στις επιμέρους ταχύτητες και θέσεις των οχημάτων. Στην αντίθετη άκρη του φάσματος βρίσκεται η μακροσκοπική προσέγγιση η οποία συναθροίζει πληροφορίες και επεξεργάζεται τιμές όπως οι ροές και η πυκνότητα οχημάτων. Το μεσοσκοπικό μοντέλο δανείζεται στοιχεία και από τις δύο προηγούμενες προσεγγίσεις και προσπαθεί να καλύψει το κενό ανάμεσά τους περιγράφοντας τις οντότητες της προσομοίωσης σε μεγάλη λεπτομέρεια, ενώ την συμπεριφορά τους και τις αλληλεπιδράσεις τους τις περιγράφει σε χαμηλότερο επίπεδο λεπτομέρειας (Ratrout, 2009). Ένα πιο πρόσφατο μοντέλο που επιχειρεί να αναλύσει σε ακόμα μεγαλύτερη λεπτομέρεια τις συνιστώσες του συστήματος είναι το νανοσκοπικό (Ni, 2003). Σε αυτό, μοντελοποιείται σε μεγάλο βαθμό και ο οδηγός και η συμπεριφορά του και τα οχήματα. Έτσι οι οδηγοί θεωρούνται συστήματα που

δέχονται ερεθίσματα και παράγουν εντολές οδήγησης ενώ τα οχήματα δέχονται τις εντολές αυτές και συμπεριφέρονται αναλόγως. Πολλές φορές διαφορετικές προσεγγίσεις συνυπάρχουν σε μία προσομοίωση, ώστε να αξιοποιηθούν τα πλεονεκτήματά τους στον έπακρο βαθμό.

5.3 Μικροσκοπικά Προγράμματα Προσομοίωσης (Microscopic simulators)

Στην παρούσα εργασία έγινε χρήση προσομοιωτή μικροσκοπικής προσέγγισης και ως εκ τούτου θα αναφερθούν κάποια από τα πιο γνωστά εργαλεία σε αυτήν την κατηγορία προγραμμάτων. Το πρόγραμμα προσομοίωσης SUMO στο οποίο υλοποιήθηκε το σύστημα ελέγχου θα αναλυθεί διεξοδικά στο επόμενο κεφάλαιο και συνεπώς παραλείπεται.

Ο αριθμός των διαθέσιμων προγραμμάτων για προσομοίωση της οδικής κυκλοφορίας είναι πολύ ψηλός και διαρκώς αυξάνεται. Μια ενδεικτική λίστα γνωστών προγραμμάτων και των αντίστοιχων φορέων που τα αναπτύσσουν είναι η ακόλουθη[8]:

AIMSUN2	Universitat Politècnica de Catalunya, Barcelona	Spain
ANATOLL	ISIS and Centre d'Etudes Techniques de l'Equipement	France
AUTOBAHN	Benz Consult - GmbH	Germany
CASIMIR	Institut National de Recherche sur les Transports et la Sécurité	France
CORSIM	Federal Highway Administration	USA
DRACULA	Institute for Transport Studies, University of Leeds	UK
FLEXSYT II	Ministry of Transport	Netherlands

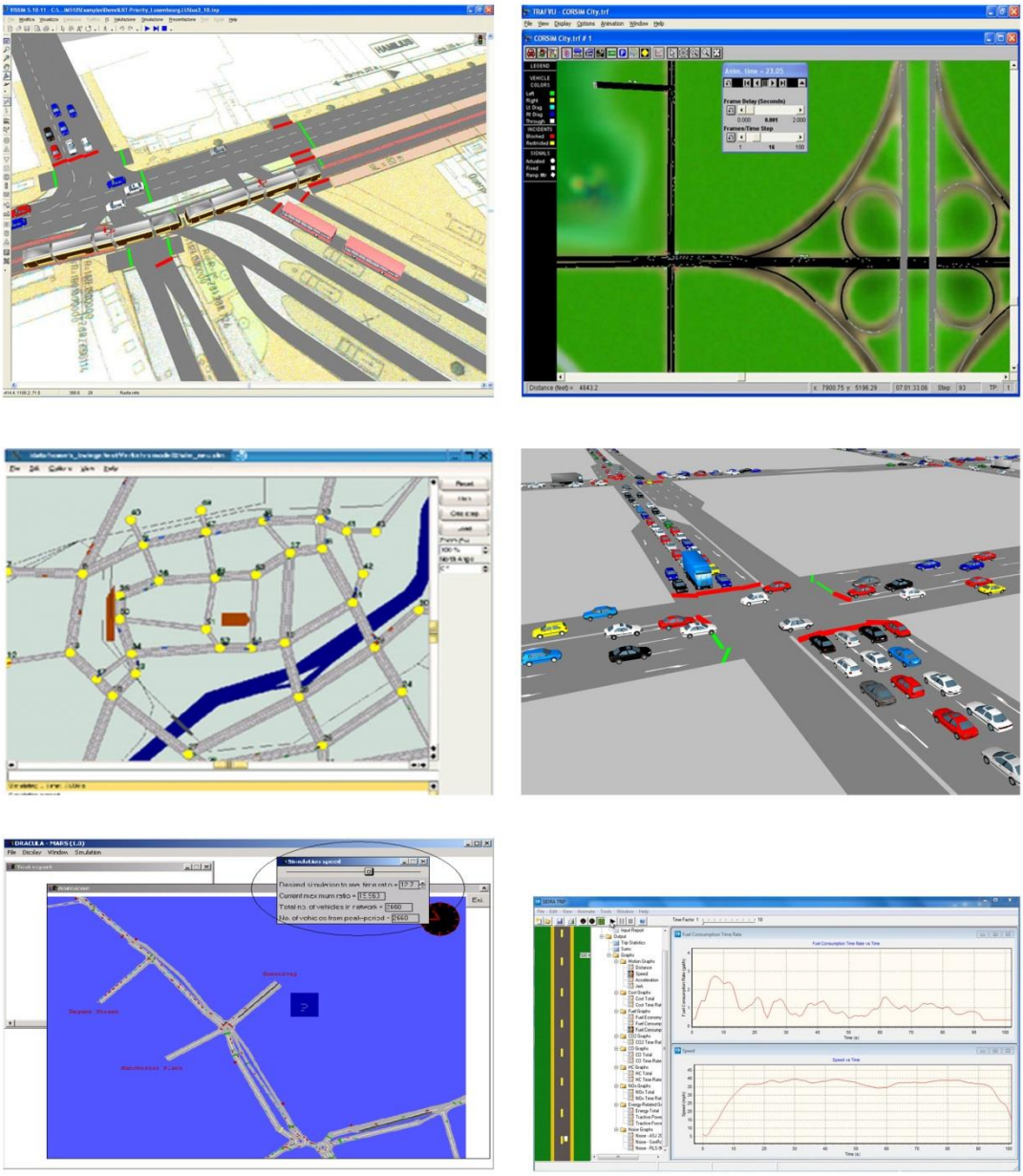
[8] http://www.its.leeds.ac.uk/projects/smartest/deliv3.html

FREEVU	University of Waterloo, Department of Civil Engineering	Canada
FRESIM	Federal Highway Administration	USA
HUTSIM	Helsinki University of Technology	Finland
INTEGRATION	Queen's University, Transportation Research Group	Canada
MELROSE	Mitsubishi Electric Corporation	Japan
MICROSIM	Centre of parallel computing (ZPR), University of Cologne	Germany
MICSTRAN	National Research Institute of Police Science	Japan
MITSIM	Massachusetts Institute of Technology	USA
MIXIC	Netherlands Organisation for Applied Scientific Research - TNO	Netherlands
NEMIS	Mizar Automazione, Turin	Italy
NETSIM	Federal Highway Administration	USA
PADSIM	Nottingham Trent University - NTU	UK
PARAMICS	The Edinburgh Parallel Computing Centre and SIAS Ltd	UK
PHAROS	Institute for simulation and training	USA
PLANSIM-T	Centre of parallel computing (ZPR), University of Cologne	Germany
SHIVA	Robotics Institute - CMU	USA
SIGSIM	University of Newcastle	UK
SIMDAC	ONERA - Centre d'Etudes et de Recherche de Toulouse	France
SIMNET	Technical University Berlin	Germany
SISTM	Transport Research Laboratory, Crowthorne	UK
SITRA-B+	ONERA - Centre d'Etudes et de Recherche de Toulouse	France
SITRAS	University of New South Wales, School of Civil Engineering	Australia
TRANSIMS	Los Alamos National Laboratory	USA
THOREAU	The MITRE Corporation	USA
VISSIM	PTV System Software and Consulting GMBH	

Τα περισσότερα από τα μικροσκοπικά προγράμματα προσομοίωσης παρέχουν ένα σύνολο δυνατοτήτων που μεταξύ άλλων περιλαμβάνει:

- Πολυτροπική (multimodal) προσομοίωση που περιλαμβάνει ταυτόχρονα στο ίδιο σύστημα μοντελοποίηση διαφόρων τύπων οχημάτων (αυτοκίνητα, λεωφορεία, μοτοσυκλέτες κλπ) αλλά και πεζούς.
- Υποστήριξη μεγάλων δικτύων ή και προσομοίωση ολόκληρων πόλεων με μόνο περιορισμό τους πόρους του υπολογιστή στον οποίο διεξάγεται.
- Γραφικό περιβάλλον που συνεισφέρει στην οπτικοποίηση της πληροφορίας και στην γρήγορη αξιολόγηση της επίδοσης του συστήματος
- Σχεδίαση οδικών δικτύων ή εισαγωγή τους από άλλες εφαρμογές ή χάρτες
- Μεταβαλλόμενη ταχύτητα προσομοίωσης
- Υπολογισμός στατιστικών των οχημάτων όπως μέση ταχύτητα και αριθμός στάσεων αλλά και προσεγγιστικός υπολογισμός της κατανάλωσης καυσίμου και των εκπομπών ρύπων.

Ακολουθούν ορισμένα στιγμιότυπα οθόνης από μερικά από τα προγράμματα εξομοίωσης που προσφέρουν γραφική αναπαράσταση του οδικού δικτύου:



Εικόνα 7 - Γραφικά Περιβάλλοντα Προσομοίωσης

ΚΕΦΑΛΑΙΟ 6ο – SUMO (SIMULATION of URBAN MOBILITY)

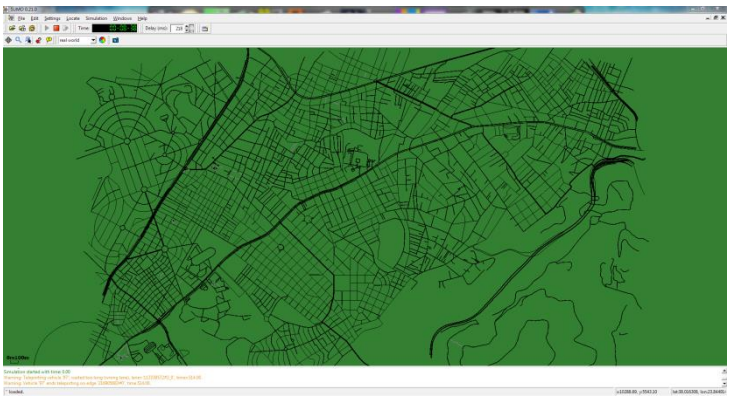
Η παρούσα εργασία πραγματοποιήθηκε με χρήση του προγράμματος προσομοίωσης οδικής κυκλοφορίας SUMO[9] το όνομα του οποίου προέρχεται από τα αρχικά των λέξεων Simulation of Urban MObility (Προσομοίωση Αστικής Κινητικότητας). Ο προσομοιωτής αυτός αναπτύσσεται κυρίως από εργαζόμενους στο Ινστιτούτο Συστημάτων Μετακίνησης (Institution of Transportation Systems) του Γερμανικού Κέντρου Αεροδιαστημικής (German Aerospace Center) από το 2001. Χρησιμοποιεί το μικροσκοπικό μοντέλο της κίνησης και χρησιμοποιεί συνεχείς μεταβλητές για την αναπαράσταση του χώρου και διακριτή για την αναπαράσταση του χρόνου (το βήμα μπορεί να καθοριστεί από τον χρήστη). Δύναται να αναπαραστήσει διαφόρων ειδών οχήματα, αστικές συγκοινωνίες και πεζούς και για κάθε όχημα που προσομοιώνεται μπορεί να επιλεχθεί ένα πλήθος παραμέτρων (μήκος οχήματος, μέγιστη επιτάχυνση/επιβράδυνση, μέγιστη ταχύτητα, αντίδραση του οδηγού κλπ) κάνοντας την προσομοίωση να παράγει πολύ ρεαλιστικά αποτελέσματα και εφαρμόσιμα στον πραγματικό κόσμο.

Μέσα στο πλήθος των δυνατοτήτων του είναι και ο έλεγχος των φαναριών στις διασταυρώσεις του οδικού δικτύου, έλεγχος που μπορεί να γίνει είτε online – δηλαδή κατά την εκτέλεση της προσομοίωσης, είτε offline με φόρτωση ρυθμίσεων των φαναριών κατά την εκκίνηση της. Ο online έλεγχος της προσομοίωσης είναι και ένα από τα πιο σημαντικά χαρακτηριστικά του προγράμματος καθώς υλοποιείται άμεσα με την διεπαφή TraCI (Traffic Control Interface) η οποία δίνει την δυνατότητα στον χρήστη μέσω μίας TCP client-server αρχιτεκτονικής να λαμβάνει και να τροποποιεί

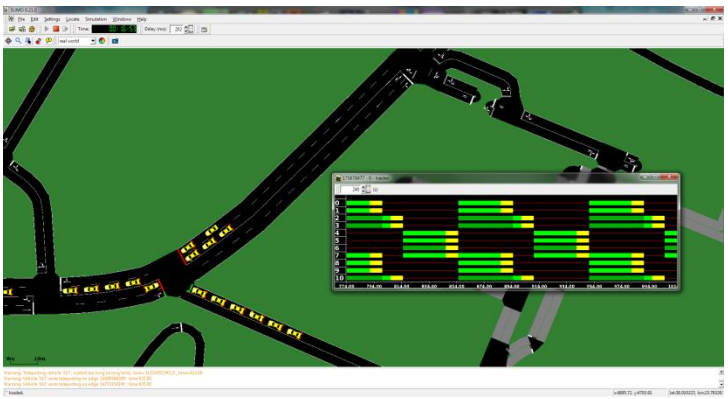
[9] http://dlr.de/ts/sumo

τιμές της προσομοίωσης κατά την εκτέλεσή της. Με χρήση της διεπαφής TraCI πραγματοποιήθηκαν και τα πειράματα που αναλύονται στην παρούσα εργασία.

Το SUMO παρέχει επιπλέον ένα αναλυτικό γραφικό περιβάλλον προσομοίωσης που δίνει αρκετές πληροφορίες σχετικά με την εξέλιξη των πειραμάτων και διευκολύνει την κατανόηση και εποπτεία των αποτελεσμάτων. Δίνεται μάλιστα η δυνατότητα στον χρήστη να επιλέξει και τον βαθμό της λεπτομέρειας της γραφικής αναπαράστασης, ανάλογα με την επεξεργαστική ισχύ που διαθέτει.



Εικόνα 8 - Προσομοίωση μεγάλου χάρτη (ΒΑ Προάστια Αθήνας)



Εικόνα 9 - Κοντινό στιγμιότυπο διασταύρωσης

Ακολουθεί μία εκτενής ανάλυση του τρόπου λειτουργίας του SUMO με λεπτομέρειες σχετικά με τον τρόπο ορισμού δικτύων, οχημάτων καθώς και του τρόπου ελέγχου των φαναριών.

6.1 Ορισμός δικτύου στο SUMO

Τα οδικά δίκτυα στο SUMO κωδικοποιούνται ως .xml αρχεία. Η δημιουργία των αρχείων αυτόν πραγματοποιείται με τρεις πιθανούς τρόπους:

- με την χρήση του εργαλείου netconvert που παρέχεται και μετατρέπει αρχεία .xml με επιμέρους ορισμούς σχετικά με τους κόμβους, τις ακμές και άλλες λεπτομέρειες του δικτύου
- τυχαία με το εργαλείο netgenerate
- με εισαγωγή δικτύων είτε από άλλα προγράμματα προσομοίωσης (Vissim, Matsim κ.α.) ή από χάρτες OpenStreetMap

Ο τρόπος που δίνει τον μέγιστο έλεγχο στις παραμέτρους του δικτύου και προτιμήθηκε για την εκπόνηση της εργασίας αυτής είναι ο πρώτος, αν και δοκιμάστηκαν και οι άλλοι δύο. Στις παραπάνω εικόνες (Εικόνα 8, Εικόνα 9) φαίνεται ένα παράδειγμα εισαγωγής δικτύου από τους χάρτες του OpenStreetMaps.

6.1.1 Ορισμός κόμβων δικτύου

Η έννοια του κόμβου στο SUMO χρησιμοποιείται για να αναπαραστήσει διασταυρώσεις είτε σημεία εκκίνησης ή τερματισμού δρόμων. Η διαφοροποίηση των παραπάνω γίνεται αυτόματα από το πρόγραμμα netconvert που διαπιστώνει τον αριθμό των προσκείμενων ακμών σε κάθε κόμβο. Οι κόμβοι ορίζονται σε αρχείο με κατάληξη *.nod.xml* με τις εξής παραμέτρους:

- *id*: Η ονομασία του κόμβου. Μπορούν να χρησιμοποιηθούν αριθμοί ή αλφαριθμητικές ονομασίες.
- *x*: Η x-συντεταγμένη του κόμβου στο επίπεδο της προσομοίωσης
- *y*: Η y-συντεταγμένη του κόμβου στο επίπεδο της προσομοίωσης
- *type:* Ο τύπος του κόμβου. Εδώ επιλέγεται πώς θα επιλέγεται μια διασταύρωση:
  - "traffic_light": αν μια διασταύρωση θα ελέγχεται με φανάρια
  - "priority": αν προηγούνται οχήματα από δρόμο με μεγαλύτερη προτεραιότητα
  - "right_before_left": αν ισχύει η εκ των δεξιών προτεραιότητα

Ένα παράδειγμα ορισμού κόμβων από την υλοποίηση που θα αναλυθεί στο επόμενο κεφάλαιο είναι το ακόλουθο:

```
<?xml version="1.0" encoding="UTF-8"?>
<nodes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/nodes_file.xsd">
    <node id="0" x="0.0" y="0.0" type="traffic_light"/>
<node id="1" x="-500.0" y="0.0" type="priority"/>
<node id="2" x="+500.0" y="0.0" type="priority"/>
<node id="3" x="0.0" y="-500.0" type="priority"/>
<node id="4" x="0.0" y="+500.0" type="priority"/>
</nodes>
```

6.1.2 Ορισμός ακμών δικτύου

Οι ακμές του δικτύου αναπαριστούν τους δρόμους που θα υλοποιηθούν στην προσομοίωση. Κάθε κατεύθυνση ενός δρόμου αντιστοιχεί σε διαφορετική ακμή και συνεπώς ένας δρόμος δύο κατευθύνσεων ανάμεσα σε δύο διασταυρώσεις υλοποιείται με δύο ακμές αντίθετης κατεύθυνσης. Οι ορισμοί γίνονται σε ένα αρχείο με κατάληξη *.edg.xml* με τις ακόλουθες παραμέτρους:

- *id*: Το όνομα της ακμής

- from: Το όνομα του κόμβου από το αρχείο κόμβων από τον οποίο θα ξεκινάει η ακμή
- *to*: Το όνομα του κόμβου από το αρχείο κόμβων στον οποίο θα τερματίζει η ακμή
- *type*: Το όνομα του αρχείου τύπου της ακμής που περιλαμβάνει όλες τις πληροφορίες που αναφέρονται στην συνέχεις προκειμένου να αποφευχθεί η άσκοπη επανάληψη (αναλύεται στην επόμενη υποενότητα)
- *numLanes*: Ο αριθμός των λωρίδων κυκλοφορίας στην ακμή
- *speed*: Η μέγιστη ταχύτητα που επιτρέπεται στις λωρίδες της ακμής (σε m/s)
- *priority*: Ένας ακέραιος αριθμός που υποδεικνύει την προτεραιότητα της ακμής. Σε διασταυρώσεις που δεν ελέγχονται από φανάρια (τύπου "priority") τα οχήματα που κινούνται σε ακμές με χαμηλότερο priority σταματούν και περιμένουν να περάσουν τα οχήματα με υψηλότερη τιμή στο πεδίο αυτό.
- *allow/disallow*: Μια λίστα με τους τύπους των οχημάτων που επιτρέπονται/απαγορεύονται στην ακμή
- *width*: Το πλάτος κάθε λωρίδας (σε m)

Το αρχείο ακμών της υλοποίησής μας είναι το ακόλουθο. Παράμετροι που δεν ήταν σημαντικές, δεν αναφέρονται ρητά και εξάγονται αυτόματα από το πρόγραμμα δημιουργίας του δικτύου:

```
<?xml version="1.0" encoding="UTF-8"?>
<edges xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/edges_file.xsd">
<edge id="1i"  from="1" to="0" priority="1" numLanes="1" speed="16.666" />
<edge id="1o" from="0" to="1" priority="1" numLanes="1" speed="16.666" />

<edge id="2i"  from="2" to="0" priority="1" numLanes="1" speed="16.666" />
<edge id="2o" from="0" to="2" priority="1" numLanes="1" speed="16.666" />

<edge id="3i"  from="3" to="0" priority="1" numLanes="1" speed="16.666" />
<edge id="3o" from="0" to="3" priority="1" numLanes="1" speed="16.666" />

<edge id="4i"  from="4" to="0" priority="1" numLanes="1" speed="16.666" />
```

```
<edge id="4o" from="0" to="4" priority="1" numLanes="1" speed="16.666" />
</edges>
```

6.1.3 Ορισμός τύπων

Όπως φαίνεται στον παραπάνω κώδικα, ένα μεγάλο μέρος της πληροφορίας στους ορισμούς της ακμής επαναλαμβάνεται. Για να διευκολυνθεί η δημιουργία ακμών με ίδια χαρακτηριστικά, χρησιμοποιείται το αρχείο τύπου το οποίο φέρει τις παραμέτρους *numLanes, speed, priority, allow, disallow, width* μαζί με μία παράμετρο *id* που υποδηλώνει το όνομα του τύπου της ακμής. Το αρχείο έχει κατάληξη *.typ.xml* και το παραπάνω παράδειγμα θα μπορούσε να μετατραπεί ως εξής:

```
<?xml version="1.0" encoding="UTF-8"?>
<edges xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/edges_file.xsd">
<edge id="1i"  from="1" to="0" type="a"  />
<edge id="1o" from="0" to="1" type="a"  />

<edge id="2i"  from="2" to="0" type="a"  />
<edge id="2o" from="0" to="2" type="a"  />
<edge id="3i"  from="3" to="0"type="a"  />
<edge id="3o" from="0" to="3" type="a"  />

<edge id="4i"  from="4" to="0" type="a"  />
<edge id="4o" from="0" to="4" type="a"  />
</edges>



<?xml version="1.0" encoding="UTF-8"?>
<types xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/edges_file.xsd">
<type id="a"  priority="1" numLanes="1" speed="16.666" />
</types>
```

6.1.4 Ορισμός διασυνδέσεων

Το τελευταίο πολύ βασικό στοιχείο της δημιουργίας ενός δικτύου είναι το αρχείο των διασυνδέσεων. Σε αυτό το αρχείο περιλαμβάνονται οι πληροφορίες για το σε ποιες ακμές μπορεί να μεταβεί ένα όχημα αφού διανύσει την τρέχουσα ακμή. Ο ορισμός θέλει προσοχή γιατί αν οι ακμές δεν πρόσκεινται στην ίδια διασταύρωση, ή δεν είναι συνέχεια η μία της άλλης θα δοθεί σφάλμα από το πρόγραμμα δημιουργίας του δικτύου. Το αρχείο έχει κατάληξη .con.xml και περιλαμβάνει τις ακόλουθες παραμέτρους:

- from: Το όνομα της ακμής αναχώρησης του οχήματος
- to: Το όνομα της ακμής άφιξης του οχήματος
- fromLane: Ο αριθμός της λωρίδας αναχώρησης του οχήματος
- toLane: Ο αριθμός της λωρίδας άφιξης του οχήματος

Οι διασυνδέσεις της διασταύρωσης που χρησιμοποιήθηκε στα πειράματα φαίνεται παρακάτω:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<connections xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/connections_file.xsd">
    <connection from="1i" to="2o" fromLane="0" toLane="0"/>
    <connection from="2i" to="1o" fromLane="0" toLane="0"/>
    <connection from="3i" to="4o" fromLane="0" toLane="0"/>
    <connection from="4i" to="3o" fromLane="0" toLane="0"/>
</connections>
```

6.1.5 Δημιουργία δικτύου

Η δημιουργία του δικτύου με χρήση των παραπάνω τεσσάρων αρχείων γίνεται με χρήση του εργαλείου netconvert το οποίο το τρέχουμε από την γραμμή εντολών με είσοδο ένα αρχείο που παρέχει πληροφορίες για τα αρχεία nod, edg, typ και con. Το αρχείο αυτό έχει κατάληξη .netccfg και παρουσιάζεται παρακάτω:

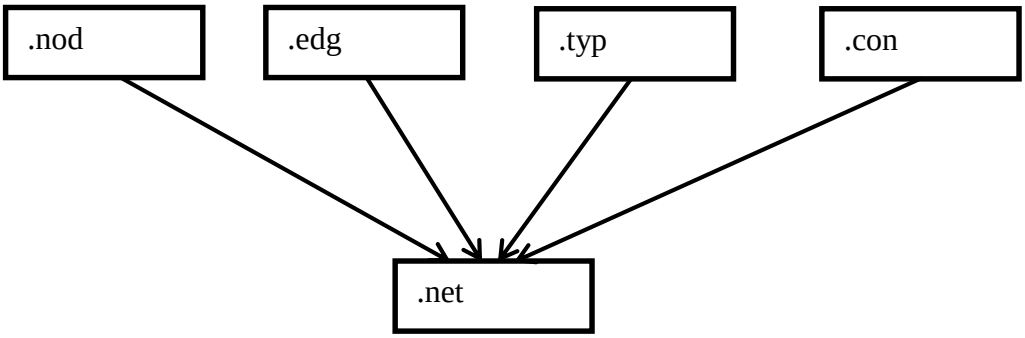
```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://sumo.sf.net/xsd/netconvertConfiguration.x
sd">
   <input>
      <node-files value="cross.nod.xml"/>
      <edge-files value="cross.edg.xml"/>
      <connection-files value="cross.con.xml"/>
      <type-files value="cross.typ.xml"/>
   </input>
   <output>
      <output-file value="cross.net.xml"/>
   </output>
   <report>
      <verbose value="true"/>
   </report>
</configuration>
```

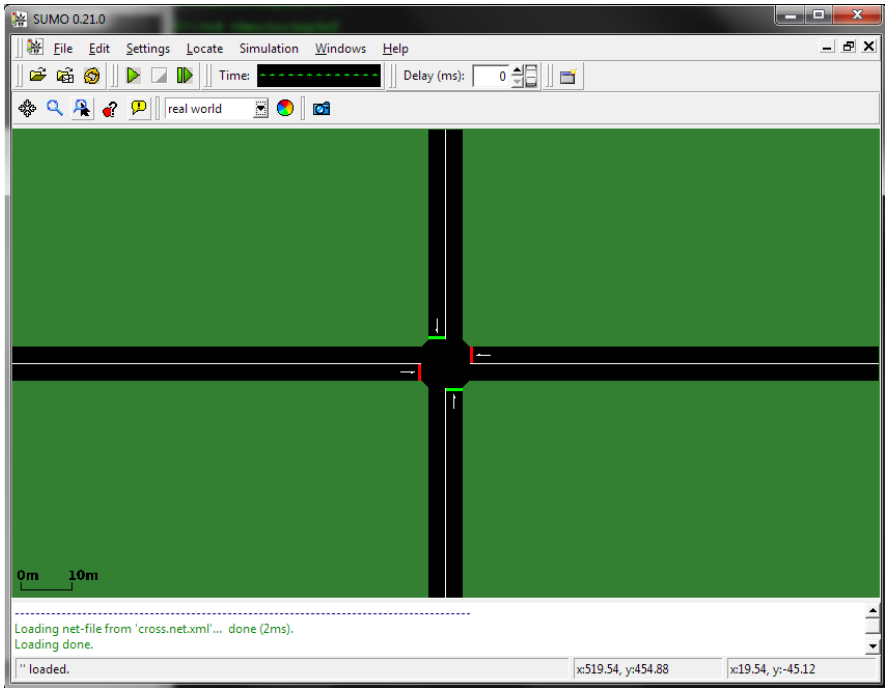
Τρέχοντας το πρόγραμμα σε ένα τερματικό, λαμβάνουμε το ακόλουθο επιβεβαιωτικό μήνυμα με κάποια χαρακτηριστικά του δικτύου:

```
C:\thesissingle2\data>netconvert -c cross.netccfg
Loading configuration... done.
Parsing nodes from 'cross.nod.xml'... done.
Parsing edges from 'cross.edg.xml'... done.
Parsing connections from 'cross.con.xml'... done.
 Import done:
   5 nodes loaded.
   8 edges loaded.
Removing self-loops... done.
Removing empty nodes... done.
   0 nodes removed.
Moving network to origin... done.
Joining similar edges... done.
Computing turning directions... done.
Sorting nodes' edges... done.
Computing node shapes... done.
Computing edge shapes... done.
Computing node types... done.
Computing priorities... done.
Computing approached edges... done.
Computing approaching lanes... done.
Dividing of lanes on approached lanes... done.
Processing turnarounds... done.
Rechecking of lane endings... done.
Assigning nodes to traffic lights... done.
Computing traffic light control information... done.
Computing node logics... done.
Computing traffic light logics... done.
 1 traffic light(s) computed.
Building inner edges... done.
-----------------------------------------------------
Summary:
 Node type statistics:
  Unregulated junctions       : 0
  Priority junctions          : 5
  Right-before-left junctions : 0
 Network boundaries:
  Original boundary  : -500.00,-500.00,500.00,500.00
  Applied offset     : 500.00,500.00
  Converted boundary : 0.00,0.00,1000.00,1000.00
-----------------------------------------------------
Success.
```

Εικόνα 10 - Επιτυχής δημιουργία δικτύου με το netconvert

Ο τρόπος λειτουργίας του netconvert παρουσιάζεται εποπτικά στο παρακάτω διάγραμμα. Ακολουθεί ένα στιγμιότυπο οθόνης από το δίκτυο που δημιουργήθηκε με τα παραπάνω αρχεία:





Εικόνα 11 - Διασταύρωση ελεγχόμενη από φανάρι στο SUMO

6.2 Δημιουργία ζήτησης

Την δημιουργία του δικτύου ακολουθεί ο ορισμός της ροής των οχημάτων, ή αλλιώς της ζήτησης, που θα εισαχθούν στην προσομοίωση. Για αυτό το σκοπό, το sumo προβλέπει μερικές διαφορετικές μεθόδους που παρέχουν διαφορετικό επίπεδο λεπτομέρειας. Στην παρούσα μέθοδο χρησιμοποιήθηκε η πρώτη μέθοδος η οποία, αν και πιο κουραστική, παρείχε μεγαλύτερη ελευθερία στον έλεγχο της κατανομής των οχημάτων και στην άμεση αλλαγή των παραμέτρων ανάλογα με τις ανάγκες του εκάστοτε πειράματος. Για λόγους πληρότητας θα αναφερθούν και οι πιο σημαντικές από τις υπόλοιπες μεθόδους.

6.2.1 Χειροκίνητα - με αρχεία .xml

Η χειροκίνητη μέθοδος περιλαμβάνει την στοιχειοθέτηση ενός xml αρχείου με κατάληξη .rou.xml το οποίο περιλαμβάνει πληροφορίες για τον τύπο των οχημάτων, τις διαδρομές που είναι επιτρεπτό να ακολουθήσουν, αλλά και κάθε όχημα χωριστά με την ακριβή στιγμή εισαγωγής του στη προσομοίωση. Προφανώς το σύνολο των οχημάτων δεν ορίστηκαν «χειροκίνητα» αλλά με την βοήθεια ενός script σε python το οποίο παρήγαγε καταχωρήσεις οχημάτων στο εν λόγω αρχείο με βάσει ορισμένες προδιαγραφές τυχαιότητας. Λεπτομέρειες θα αναλυθούν στο επόμενο κεφάλαιο.

Για τους τύπους τον οχημάτων παρέχονται οι ακόλουθες παράμετροι:

- id: Το όνομα του τύπου του οχήματος
- accel: Η μέγιστη επιτάχυνση του οχήματος (σε m/s2)
- decel: Η μέγιστη επιβράδυνση του οχήματος (σε m/s2)
- sigma: Η ατέλεια του οδηγού (από 0 έως 1 με προεπιλογή 0.5)
- tau: Ο χρόνος αντίδρασης του οδηγού
- length: Το μήκος του οχήματος

- minGap: Το ελάχιστο κενό διάστημα που πρέπει να διατηρείται από το προπορευόμενο όχημα
- maxSpeed: Η μέγιστη ταχύτητα του οχήματος (σε m/s)
- color: Το χρώμα των οχημάτων αυτού του τύπου
- guiShape: Το σχήμα που θα χρησιμοποιηθεί για την αναπαράσταση του τύπου
- width: Το πλάτος του οχήματος

Οι παράμετροι αυτές δεν είναι όλες υποχρεωτικό να οριστούν ρητά. Όποιες παραλείπονται από τον ορισμό λαμβάνουν κάποιες προεπιλεγμένες τιμές που έχουν οριστεί από το SUMO. Ένα παράδειγμα υλοποίησης κάποιων τύπων οχημάτων ακολουθεί:

```
<vType id="typeWE" accel="0.8" decel="4.5" sigma="0.5" length="10"
minGap="2.5" maxSpeed="16.67" guiShape="bus"/>
 <vType id="typeNS" accel="0.8" decel="4.5" sigma="0.5" length="5"
minGap="2.5" maxSpeed="25" guiShape="passenger"/>
```

Το δεύτερο τμήμα του αρχείου αναφέρεται στις διαδρομές που μπορεί να πραγματοποιήσει κάθε όχημα. Κάθε διαδρομή απαρτίζεται από τα εξής πεδία:

- id: Το όνομα της διαδρομής
- edges: Η λίστα των ακμών από τις οποίες απαρτίζεται μια διαδρομή με την σειρά με την οποία διανύονται. Οι ακμές αναφέρονται με το id τους όπως αυτό έχει οριστεί στο αρχείο .edg
- color: Το χρώμα της διαδρομής (προαιρετικά)

Ένα παράδειγμα για τα παραπάνω είναι το ακόλουθο:

```
<route id="right" edges="1i 2o" />
<route id="left" edges="2i 1o" />
<route id="down" edges="4i 3o" />
<route id="up" edges="3i 4o" />
```

Με βάση τους παραπάνω ορισμούς, εισάγονται καταχωρήσεις για τα οχήματα της προσομοίωσης με τις εξής παραμέτρους:

- id: Το όνομα του οχήματος – ιδανικά ο αύξων αριθμός του
- type: ο τύπος του οχήματος από αυτούς που ορίστηκαν παραπάνω
- route: το id της διαδρομής από τους παραπάνω ορισμούς
- color: το χρώμα του οχήματος
- depart: Η χρονική στιγμή που εισάγεται το όχημα στο δίκτυο
- departLane: Η λωρίδα στην οποία εισάγεται. Υπάρχουν επιλογές για τυχαία επιλογή λωρίδας, η άλλα κριτήρια
- departPos: Η θέση στην λωρίδα όπου θα τοποθετηθεί το όχημα
- departSpeed: Η ταχύτητα αναχώρησης

Τα παραπάνω φαίνονται συνοπτικά στους ορισμούς των εξής τριών οχημάτων:

```
<vehicle id="1" type="typeNS" route="up" depart="1" color="1,0,0"
departPos="free" departSpeed="max"/>
<vehicle id="2" type="typeWE" route="right" depart="4" departPos="free"
departSpeed="max"/>
<vehicle id="3" type="typeWE" route="left" depart="6" departPos="free"
departSpeed="max"/>
```

6.2.2 Με ορισμό διαδρομών

Ένας από τους πιο αυτοματοποιημένους τρόπους είναι με τον ορισμό κάποιας οικογένειας διαδρομών, και την χρήση του προγράμματος DUAROUTER το οποίο παράγει το αρχείο rou.xml που παρουσιάστηκε στο 6.2.1. Τα στοιχεία που πρέπει να οριστούν είναι τα ακόλουθα:

- id: Το όνομα των οχημάτων που θα παραχθούν με βάση αυτή τη διαδρομή. Αν δεν δοθεί χρησιμοποιείται ένας αύξων αριθμός.
- depart: Η χρονική στιγμή κατά την οποία αναχωρεί το πρώτο όχημα
- from: Η ακμή αφετηρίας της διαδρομής
- to: Η ακμή τερματισμού της διαδρομής

- period: Η χρονική καθυστέρηση μέχρι την δημιουργία του επόμενου οχήματος αυτής της διαδρομής.
- repno: Ο συνολικός αριθμός οχημάτων που θα δημιουργηθούν σε αυτή τη διαδρομή.
- departLane, departPos, departSpeed: όπως παραπάνω
-

Ένα παράδειγμα αυτής της υλοποίησης είναι το εξής

```
<tripdef id="Route1" depart="50" from="1i" to="2o" period="10" repno="2000" />
```

6.2.3 Με ορισμό ροών

Μία παρόμοια μέθοδος ορισμού οχημάτων είναι μέσω δηλώσεων ροών. Σε αυτή την περίπτωση ορίζονται η αρχή και το τέλος του διαστήματος κατά το οποίο θα εισαχθούν τα οχήματα, το πλήθος τους αλλά και αντίστοιχα τα πεδία id, from, to όπως παραπάνω. Αυτά τα στοιχεία δίνονται επίσης με την μορφή αρχείου στο πρόγραμμα DUAROUTER για να δημιουργηθεί το αρχείο .rou.xml

```
<interval begin="0" end="5000">
<flow id="carNS" from="1i" to="2o" no="1000"/>
</interval>
```

Ο παραπάνω ορισμός παράγει οχήματα με ροή 1000/5000 = 0.2 οχήματα το δευτερόλεπτο τα οποία εισάγονται ομοιόμορφα στην προσομοίωση κατά την διάρκεια των 5000 χρονικών βημάτων. Αυτή η ομοιομορφία αυτής και του τρόπου που παρουσιάστηκε στο 6.2.2 είναι μη ρεαλιστική και για αυτό προτιμήθηκε ο τρόπος του 6.2.1.

6.2.4 Με ορισμό λόγου αλλαγής κατεύθυνσης (Junction turning ratio)

Μια δυνατότητα που προσφέρει το SUMO που εξασφαλίζει μία πιο ρεαλιστική προσέγγιση είναι με έναν συνδυασμό του ορισμού ροών που αναφέρθηκε παραπάνω, και του ορισμού του ποσοστού των οχημάτων της ροής που ενδέχεται να αλλάξουν πορεία σε κάθε διασταύρωση.

```
<interval begin="0" end="3600">
    <fromEdge id="1i ">
      <toEdge id="2o" probability="0.2"/>
      <toEdge id="3o" probability="0.5"/>
      <toEdge id="4o" probability="0.3"/>
    </fromEdge>
```

Το παραπάνω δηλώνει πως στο χρονικό διάστημα από 0 έως 3600 τα οχήματα που εξέρχονται από την ακμή 1i έχουν 20% πιθανότητα να εισέλθουν στην ακμή 2o, 50% στην 3o, και 30% στην 4o. Για τον ορισμό του αριθμού των οχημάτων που εξέρχονται από την 1i χρησιμοποιούνται ορισμοί ροών όπως στο 6.2.3 με την διαφορά ότι παραλείπεται το πεδίο “to” αφού δεν είναι προκαθορισμένη η ακμή άφιξης του κάθε οχήματος.

Αφού συνταχθεί το αρχείο με τις ροές και τους λόγους ανακατεύθυνσης εισάγεται στο πρόγραμμα JTRROUTER το οποίο παράγει το ζητούμενο αρχείο .rou.xml.

6.2.5 Τυχαία

Η πιο γρήγορη επιλογή για την παραγωγή κίνησης στο δίκτυο είναι να χρησιμοποιηθεί το πρόγραμμα DUAROUTER χωρίς ορισμούς για ροές και διαδρομές ώστε να παραχθεί ένα σύνολο τυχαίας κίνησης σε όλες τις πιθανές διαδρομές του δικτύου. Η μέθοδος αυτή δεν εξάγει ρεαλιστικά αποτελέσματα, αλλά ενδείκνυται για γρήγορη δοκιμή του υπό εξέταση οδικού δικτύου.

6.3 TraCI

Το TraCI (Traffic Control Interface) είναι η διεπαφή που επιτρέπει στον χρήστη να έχει άμεση πρόσβαση στο πρόγραμμα προσομοίωσης, να ανακτά από αυτό τιμές για και πληροφορίες για όλα τα αντικείμενα που προσομοιώνονται καθώς και να μεταβάλει τιμές και την συμπεριφορά τους κατά την εκτέλεση. Το TraCI χρησιμοποιεί μία αρχιτεκτονική client-server τύπου TCP, δίνοντας τον ρόλο του server στο SUMO και τον ρόλο του client σε πρόγραμμα που μπορεί να συντάξει ο χρήστης (συνήθως σε γλώσσα Python ή Java). Στο μενού αυτό λειτουργίας του SUMO η προσομοίωση δεν εκτελείται ανεξάρτητα, αλλά περιμένει σε κάθε βήμα για οδηγίες από τον client.

Αρχικά για να θέσουμε το SUMO στη λειτουργία αυτή, πρέπει να προσθέσουμε στις παραμέτρους εκκίνησης την επιλογή: --remote-port <INT> όπου στο <INT> ορίζεται η θύρα (port) που θα επικοινωνεί ο server με τον client.

Το πακέτο του SUMO έχει έτοιμη μια εκτενή βιβλιοθήκη μεθόδων σε Python που χρησιμεύουν για την ανταλλαγή δεδομένων μεταξύ του χρήστη και της προσομοίωσης. Παρακάτω ακολουθεί μια συνοπτική αναφορά σε μερικές από τις συναντήσεις που φάνηκαν πιο χρήσιμες κατά την εκπόνηση της εργασίας αυτής.

- traci.lane.setMaxSpeed(laneID, speed): Παρακάμπτεται η τιμή που δόθηκε ως μέγιστη ταχύτητα της λωρίδας laneID κατά την δημιουργία του δικτύου και ορίζεται ως μέγιστη η speed.
- traci.trafficlights.setCompleteRedYellowGreenDefinition(tlsID, tls) : Ορίζεται ένα καινούργιο πρόγραμμα για το φανάρι με αναγνωριστικό tlsID. Το πρόγραμμα βρίσκεται στη μεταβλητή tls.
- traci.trafficlights.setPhase(tlsID,index): Ορίζει την φάση του τρέχοντος προγράμματος με δείκτη index ως την ενεργή φάση.
- traci.trafficlights.setPhaseDuration(tlsID,phaseDuration): Ορίζει την υπολειπόμενη διάρκεια της τρέχουσας φάσης.

- traci.trafficlights.setRedYellowGreenState(tlsID,state): «Εξαναγκάζει» το φανάρι με αναγνωριστικό tlsID να μεταβεί στην κατάσταση state ανεξαρτήτως του προγράμματος και της φάσης στην οποία βρίσκεται. Το state αποτελείται από ένα σύνολο χαρακτήρων που υποδεικνύει σε τι κατάσταση θα βρίσκεται κάθε ρεύμα που ελέγχεται από φωτεινό σηματοδότη. Για παράδειγμα, το state = 'RGRG' σημαίνει ότι οι φωτεινοί σηματοδότες 0 και 2 θα έχουν χρώμα κόκκινο και οι 1 και 3 χρώμα πράσινο.

Παράλληλα υπάρχουν πολλές μέθοδοι για τον έλεγχο σχεδόν όλων των παραμέτρων των οχημάτων αλλά και του γραφικού περιβάλλοντος του προγράμματος προσομοίωσης. Λεπτομέρειες για την ανάκτηση δεδομένων μέσω του TraCI θα δοθούν στην ακόλουθη παράγραφο.

6.4 Έξοδος της προσομοίωσης – Συλλογή δεδομένων

Το SUMO παρέχει στον χρήστη μέσω της διεπαφής TraCI μια πληθώρα επιλογών για την ανάκτηση αποτελεσμάτων και δεδομένων κατά την διάρκεια της προσομοίωσης. Αυτές οι επιλογές θα μπορούσαν να χωριστούν στις επόμενες δύο κατηγορίες

6.4.1 Με χρήση ανιχνευτών

Οι ανιχνευτές που προσφέρει το Sumo είναι τριών τύπων:

E1: ανιχνευτές επαγωγικού βρόχου (induction loop detectors) που τοποθετούνται σε μια λωρίδα του οδικού δικτύου σε μια ορισμένη θέση.

E2: ανιχνευτές περιοχής (areal detectors) οι οποίοι τοποθετούνται σε μία λωρίδα όπως οι E1 με την διαφορά όμως ότι χαρακτηρίζονται από το μήκος του τμήματος της λωρίδας που παρακολουθούν.

E3: ανιχνευτές πολλαπλών προελεύσεων/πολλαπλών προορισμών (multiorigin/multidestination detectors) οι οποίοι είναι πολλοί ανιχνευτές τύπου E1 στους οποίους το ίδιο το SUMO επεξεργάζεται και συσσωρεύει αποτελέσματα για να δώσει πιο χρήσιμα και άμεσα αξιοποιήσιμα δεδομένα σχετικά με την κατάσταση της κίνησης σε έναν συνδυασμό λωρίδων.

Κυρίως χρησιμοποιήθηκαν τα δύο πρώτα είδη ανιχνευτών καθώς τα δεδομένα εξόδου τους ήταν σε πιο ακατέργαστη μορφή και συνεπώς πιο εύκολα διαχειρίσιμα από τους αλγορίθμους. Για να οριστούν οι ανιχνευτές μέσα στο οδικό δίκτυο οφείλουν να συμπεριληφθούν σε ένα χωριστό αρχείο το οποίο δίνεται σαν όρισμα κατά την εκκίνηση του SUMO. Μέσα στο αρχείο αυτό δίνονται εγγραφές για κάθε ανιχνευτή ανάλογα με το είδος του.

Για τους ανιχνευτές E1 έχουμε τα ακόλουθα στοιχεία:

- id: Το όνομα του ανιχνευτή.
- lane: Το όνομα της λωρίδας που θα τοποθετηθεί ο ανιχνευτής.
- pos: Η θέση σε μέτρα από την αφετηρία της λωρίδας στην οποία θα τοποθετηθεί ο ανιχνευτής.
- freq: Ο χρόνος σε δευτερόλεπτα κατά τον οποίο ο ανιχνευτής θα συλλέγει δεδομένα ανάμεσα σε δύο εξόδους του.

Αντίστοιχα για τους ανιχνευτές E2 δίνονται οι ακόλουθες παράμετροι:

- id: Το όνομα του ανιχνευτή.
- lane: Το όνομα της λωρίδας που θα τοποθετηθεί ο ανιχνευτής.
- pos: Η θέση σε μέτρα από την αφετηρία της λωρίδας στην οποία θα τοποθετηθεί ο ανιχνευτής.
- length: Το μήκος της λωρίδας κατά το οποίο ο ανιχνευτής συλλέγει δεδομένα.

- freq: Ο χρόνος σε δευτερόλεπτα κατά τον οποίο ο ανιχνευτής θα συλλέγει δεδομένα ανάμεσα σε δύο εξόδους του.
- speedThreshold: Το όριο της ταχύτητας σε m/s κάτω από το οποίο θα υπολογίζεται πως ένα όχημα είναι σταματημένο (halting vehicle).
- timeThreshold: Το όριο του χρόνου σε s κάτω από το οποίο θα υπολογίζεται πως ένα όχημα είναι σταματημένο (halting vehicle).
- jamThreshold: Η μέγιστη απόσταση σε m που πρέπει να έχει ένα όχημα από το επόμενο σταματημένο όχημα, ώστε να θεωρηθεί κομμάτι της σταματημένης ουράς .

Με βάση τα παραπάνω έχουμε πρόσβαση με την βοήθεια του TraCI σε πολλές διαφορετικές τιμές που αφορούν την προσομοίωση. Ακολουθούν ορισμένες από αυτές, που χρησιμοποιήθηκαν εκτενέστερα:

- traci.inductionloop.getLastStepMeanSpeed(loopID): Επιστρέφεται η μέση ταχύτητα σε m/s των οχημάτων που διέσχισαν τον συγκεκριμένο ανιχνευτή βρόχου κατά την τελευταία χρονική περίοδο μέτρησης.
- traci.inductionloop.getLastStepOccupancy(loopID): Επιστρέφεται το ποσοστό χρόνου κατά το οποίο ο ανιχνευτής ήταν κατειλημμένος από όχημα την τελευταία χρονική περίοδο μέτρησης.
- traci.inductionloop.getLastStepVehicleIDs(loopID): Επιστρέφεται μια λίστα με τα αναγνωριστικά όλων τον οχημάτων που διέσχισαν τον ανιχνευτή την τελευταία χρονική περίοδο μέτρησης.
- traci.inductionloop.getLastStepVehicleNumber(loopID): Επιστρέφεται ο αριθμός των οχημάτων που διέσχισαν τον ανιχνευτή την τελευταία χρονική περίοδο μέτρησης.
- traci.areal.getJamLengthMeters(detID): Επιστρέφεται το μήκος της ουράς των ακινητοποιημένων οχημάτων την προηγούμενη χρονική περίοδο μέτρησης.

- traci.areal.getJamLengthVehicle(detID): Επιστρέφεται ο αριθμός των οχημάτων που συμμετείχαν σε μποτιλιάρισμα στην προηγούμενη χρονική περίοδο μέτρησης.

6.4.3 Απευθείας από την προσομοίωση

Εκτός από την χρήση ανιχνευτών για την απόκτηση πληροφοριών για την πορεία της προσομοίωσης το TraCI δίνει και μία σειρά άλλων συναρτήσεων που στοχεύουν απευθείας στην ανάκτηση τέτοιων δεδομένων. Αν και ο αριθμός των συναρτήσεων αυτών είναι πολύ μεγάλος, παρακάτω θα αναφερθούν αυτές που φάνηκαν ιδιαιτέρως πιο χρήσιμες για τους σκοπούς της εργασίας αυτής.

- traci.edge.getLastStepHaltingNumber(edgeID): Επιστρέφεται ο αριθμός των σταματημένων οχημάτων στην λωρίδα με αναγνωριστικό edgeID κατά το τελευταίο βήμα της προσομοίωσης. Σταματημένο θεωρείται ένα όχημα με ταχύτητα μικρότερη του 0.1m/s
- traci.edge.getLastStepMeanSpeed(edgeID): Επιστρέφεται η μέση ταχύτητα των οχημάτων στην λωρίδα με αναγνωριστικό edgeID κατά το τελευταίο βήμα της προσομοίωσης.
- traci.edge.getLastStepOccupancy(edgeID): Επιστρέφεται η ποσοστιαία κατάληψη της επιφάνειας της λωρίδας edgeID κατά το τελευταίο βήμα της προσομοίωσης.
- traci.edge.getLastStepVehicleIDs(edgeID): Επιστρέφεται η λίστα των αναγνωριστικών όλων των οχημάτων που βρισκόντουσαν στην λωρίδα edgeID κατά το τελευταίο βήμα της προσομοίωσης.
- traci.edge.getLastStepVehicleNumber(edgeID): Επιστρέφεται ο συνολικός αριθμός των οχημάτων που βρισκόντουσαν στην λωρίδα edgeID κατά το τελευταίο βήμα της προσομοίωσης.

- traci.simulation.getDepartedIDList(): Επιστρέφεται μία λίστα που περιέχει τα αναγνωριστικά όλων των οχημάτων που εισήλθαν στην προσομοίωση την τελευταία χρονική στιγμή.
- traci.simulation.getDepartedNumber(): Επιστρέφεται ο αριθμός των οχημάτων που εισήλθαν στην προσομοίωση την τελευταία χρονική στιγμή.
- traci.simulation.getArrivedIDList():Επιστρέφεται μία λίστα που περιέχει τα αναγνωριστικά όλων των οχημάτων που εξήλθαν από την προσομοίωση την τελευταία χρονική στιγμή.
- traci.simulation.getArrivedNumber():Επιστρέφεται ο αριθμός των οχημάτων που εξήλθαν από την προσομοίωση την τελευταία χρονική στιγμή

ΚΕΦΑΛΑΙΟ 7ο – ΠΕΙΡΑΜΑΤΙΚΟ ΚΟΜΜΑΤΙ

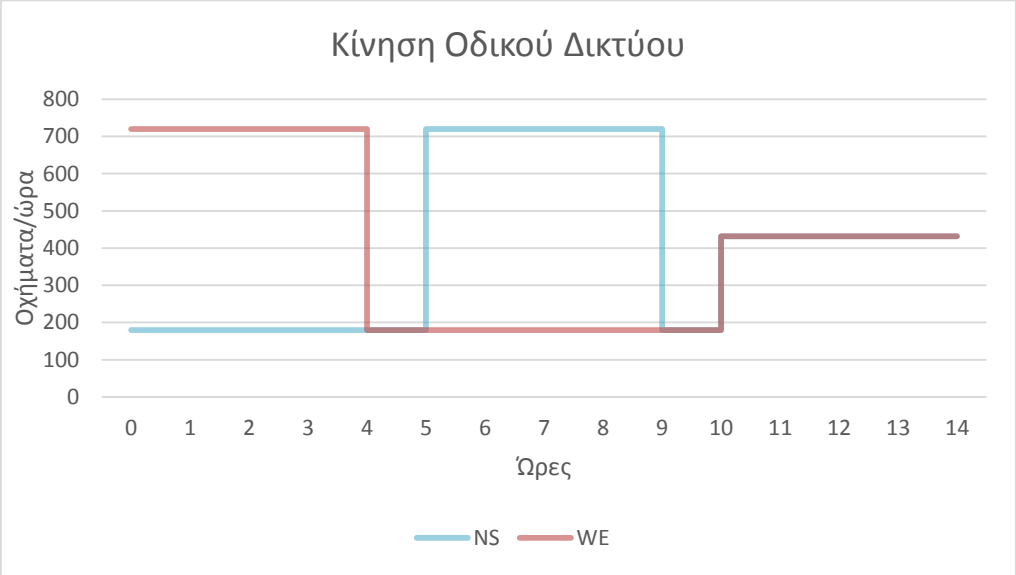
Το πειραματικό κομμάτι της παρούσας εργασίας στράφηκε κυρίως γύρω από το θέμα του ελέγχου μίας διασταύρωσης διπλής κατεύθυνσης όπως φαίνεται στην Εικόνα 11. Η διαδικασία για την δημιουργία του δικτύου, καθώς και οι μέθοδοι για την άντληση αποτελεσμάτων και πληροφοριών της προσομοίωσης αναλύθηκαν διεξοδικά στο κεφάλαιο 6. Κομμάτια του κώδικα που παρατέθηκαν έχουν ληφθεί από τα αντίστοιχα αρχεία που χρησιμοποιήθηκαν για την δημιουργία του εν λόγω οδικού δικτύου.

Ο αλγόριθμος που χρησιμοποιήθηκε ήταν ο ε-greedy Q-learning αλγόριθμος όπως αναλύθηκε στο κεφάλαιο 3.3. Για να συγκριθεί η απόδοση του αλγορίθμου χρησιμοποιήθηκαν άλλες δύο μέθοδοι ελέγχου της διασταύρωσης. Η πρώτη, και πιο απλή, ήταν τα φανάρια σταθερού κύκλου σύμφωνα με την οποία τα δύο ανταγωνιστικά ρεύματα αποκτούν πράσινο σήμα με σταθερό ρυθμό κάθε 30 δευτερόλεπτα. Αυτή η μέθοδος, αν και ιδιαίτερα απλοϊκή, εκφράζει ένα μεγάλο ποσοστό της ρύθμισης των διασταυρώσεων στον πραγματικό κόσμο.

Η δεύτερη υλοποίηση που χρησιμοποιήθηκε για να συγκριθεί με τον αλγόριθμο Qlearning χρησιμοποιεί μια άμεσα προσαρμοστική λογική (adaptive), σύμφωνα με την οποία δίνεται προτεραιότητα στο ρεύμα με την μεγαλύτερη ουρά ακινητοποιημένων οχημάτων. Συγκεκριμένα όταν ο λόγος της ουράς που αναφέρεται στα ρεύματα της διασταύρωσης με κόκκινο σήμα προς την ουρά των ρευμάτων που έχουν πράσινο σήμα ξεπεράσει μια κρίσιμη τιμή, αλλάζουν τα σήματα του φωτεινού σηματοδότη. Για να διασφαλιστεί η εύρυθμη λειτουργία του ελεγκτή και να αποφευχθούν φαινόμενα αστάθειας, ο έλεγχος αυτός πραγματοποιείται κάθε 5 δευτερόλεπτα και ορίζεται μια μέγιστη διάρκεια πρασίνου για μία κατεύθυνση 45 δευτερολέπτων καθώς και μία ελάχιστη 15 δευτερολέπτων.

7.1 Μοτίβο παραγόμενης κίνησης (traffic generation)

Επιλέχθηκε για λόγους ρεαλισμού η κίνηση που θα παραχθεί για την εκπαίδευση του ευφυούς πράκτορα, καθώς και για τον έλεγχο της επίδοσης και των τριών συστημάτων να μην είναι σταθερή αλλά να ακολουθεί ένα μοτίβο εναλλαγής υψηλής, μεσαίας και χαμηλής κίνησης στους δύο άξονες κίνησης. Ακολουθεί διάγραμμα που δείχνει την εναλλαγή της κίνησης κατά την μία εποχή εκπαίδευσης του πράκτορα:



Εικόνα 12 - Διάγραμμα κίνησης

Η «ημερήσια» κίνηση χωρίζεται σε τρείς κυρίως φάσεις:

- 0-4 ώρες: Υψηλή κίνηση 720 οχημάτων/ώρα ανά λωρίδα στον οριζόντιο άξονα (WestEast-WE) και χαμηλή 180 οχημάτων/ώρα ανά λωρίδα στον κάθετο (NorstSouth-NS).
- 5-9 ώρες: Υψηλή κίνηση 720 οχημάτων/ώρα ανά λωρίδα στον κάθετο άξονα και χαμηλή 180 οχημάτων/ώρα ανά λωρίδα στον οριζόντιο.
- 10-14 ώρες: Μέτρια κίνηση 430 οχημάτων/ώρα ανά λωρίδα και στους δύο άξονες.

Ανάμεσα στις φάσεις αυτές παρεμβλήθηκε 1 ώρα (εικονικού χρόνου) χαμηλής κίνησης και στους δύο άξονες, ώστε να αποφορτιστεί το δίκτυο από τυχόν συσσωρευμένη κίνηση και να γίνει πιο διακριτός ο διαχωρισμός των φάσεων. Παρόλα αυτά παρατηρήθηκε ότι δεν ήταν αναγκαίο να γίνει κάτι τέτοιο για την εκπαίδευση του συστήματος αλλά κυρίως εξυπηρέτησε στην εκσφαλμάτωση και ερμηνεία των ενδιάμεσων αποτελεσμάτων.

7.2 Χώρος Καταστάσεων – Ενεργειών (State-Space, Action-Space)

Ένα από τα πιο δύσκολα ζητήματα που πολύ συχνά έχουν να αντιμετωπίσουν συστήματα ενισχυτικής μάθησης είναι η κατάλληλη επιλογή του χώρου καταστάσεων αλλά και των ενεργειών που θα πραγματοποιεί ο πράκτορας. Η δυσκολία στο συγκεκριμένο πρόβλημα αυξάνεται δεδομένου ότι οι μεταβλητές που υποδηλώνουν την κατάσταση του οδικού δικτύου αλλά και την απόδοσή του είναι συνεχείς. Το ίδιο ισχύει και για τις πιθανές ενέργειες που στο συγκεκριμένο πρόβλημα είναι η κατάλληλη επιλογή της διάρκειας των πράσινων σημάτων στα ανταγωνιστικά ρεύματα της διασταύρωσης. Αντικείμενο πειραματισμού και δοκιμών υπήρξε συνεπώς ο τρόπος κβαντοποίησης των συνεχών μεταβλητών σε διακριτές με γνώμονα την απόδοση του δικτύου. Παρατηρήθηκε, όπως ήταν αναμενόμενο, πως για πολύ λίγα επίπεδα κβάντισης η απόδοση του δικτύου ήταν χαμηλή λόγω του ότι εγγενώς διαφορετικές καταστάσεις κατηγοριοποιούνταν στο ίδιο κβαντισμένο επίπεδο ενώ, αντίθετα, πολύ υψηλός αριθμός επιπέδων στερούσε από τον πράκτορα την ικανότητα γενίκευσης και αποτελεσματικής εκπαίδευσης προκαλώντας επιπρόσθετα και το πρόβλημα του πολύ μεγάλου χώρου καταστάσεων.

Αν και στην βιβλιογραφία που μελετήθηκε παρατηρήθηκε ως συχνότερη επιλογή ο αριθμός των ακινητοποιημένων οχημάτων ανά λωρίδα ή η ποσοστιαία κατάληψη (occupancy) του εκάστοτε ρεύματος, αυτή η επιλογή είναι επαρκής μόνο σε συνθήκες

χαμηλής κίνησης στο υπό μελέτη δίκτυο. Ένα παράδειγμα που κάνει τον παραπάνω ισχυρισμό πιο σαφή είναι το ακόλουθο:

Έστω η κατάσταση (1,1) που σηματοδοτεί χαμηλό αριθμό ακινητοποιημένων οχημάτων και στον NS άξονα αλλά και στον WE. Αν υποθέσουμε πως η κίνηση και στους δύο άξονες είναι περίπου ίδια, έστω 100 οχήματα/ώρα, τότε εύλογα προκύπτει πως μία λογική ενέργεια για τον πράκτορα είναι να θέσει τα φανάρια σε ισοκατανεμημένη λειτουργεία (για παράδειγμα 30 δευτερόλεπτα πράσινου χρόνου για το NS ρεύμα ακολουθούμενα από 30 δευτερόλεπτα για το WE ρεύμα). Αν αυτή η πολιτική εφαρμοστεί στην ίδια κατάσταση αλλά σε διαφορετική κατάσταση κίνησης (για παράδειγμα 500 οχήματα/ώρα στον NS άξονα και 100 οχήματα/ώρα στον WE) σύντομα θα δημιουργηθεί μποτιλιάρισμα και από την κατάσταση (1,1) ο πράκτορας θα βρεθεί σε πιο δυσμενή κατάσταση όπως (3,1) ή (4,1).

Αυτό το πρόβλημα αναφέρουν και οι Oliveira et al (de Oliveira, 2006) και επιλέγουν να το επιλύσουν με χρήση συμφραζόμενων (context). Συγκεκριμένα, για διαφορετικές καταστάσεις κίνησης ο πράκτορας μαθαίνει διαφορετικό μοντέλο και πολιτική. Στην δική μας υλοποίηση δοκιμάστηκε μια άλλη προσέγγιση όπου η κατάσταση της κίνησης στο οδικό δίκτυο συμπεριλήφθηκε σαν μεταβλητή κατάστασης. Έτσι ο πράκτορας μαθαίνει μεν ένα μόνο μοντέλο, αλλά αποφεύγονται τα φαινόμενα aliasing των καταστάσεων που αναφέρθηκαν παραπάνω. Συνοπτικά οι καταστάσεις κωδικοποιούνται ως εξής:

$[(TF_{NS}, TF_{WE}), (Occ_{NS}, Occ_{WE})]$ όπου:

TF_{NS}: Η ροή των οχημάτων (traffic flow) στον κάθετο άξονα

TF_{WE}: Η ροή των οχημάτων στον οριζόντιο άξονα

Occ_{NS}: Η ποσοστιαία κάλυψη (occupancy) της επιφάνειας του NS άξονα

Occ_{WE}: Η ποσοστιαία κάλυψη της επιφάνειας του WE άξονα

Ως ενέργειες (actions) του πράκτορα επιλέχθηκαν ορισμένοι χρονισμοί των φαναριών. Αρχικά υποθέσαμε σταθερό κύκλο φαναριού 60 δευτερολέπτων, γεγονός που ανταποκρίνεται στον τρόπο λειτουργίας των φαναριών στον πραγματικό κόσμο καθώς ο σταθερός κύκλος εξασφαλίζει τον ομαλό συντονισμό διαδοχικών διασταυρώσεων μεγαλύτερων δικτύων. Επιπρόσθετα, όπως και στο adaptive σύστημα, ορίστηκε ως ελάχιστη διάρκεια πράσινου χρόνου τα 15 δευτερόλεπτα και ως μέγιστη τα 45. Από αυτά προέκυψαν οι εξής επτά πιθανές ενέργειες του πράκτορα:

$[T_{NS}, T_{WE}]$:(15s, 45s), (20s, 30s), (25s, 35s), (30s, 30s), (35s, 25s), (40s, 20s), (45s, 15s)

Το πρόβλημα του κβαντισμού των καταστάσεων οδήγησε σε ένα σύνολο δοκιμών και αναπροσαρμογών, όπου βρέθηκε η βέλτιστη συμπεριφορά και καμπύλη μάθησης για τον ακόλουθο συνδυασμό:

Flow discretized states = [(0, 0.07), (0.07, 0.14), (0.14, 0.3)]

Occupancy discretized states = [(0, 0.05), (0.05, 0.1), (0.1, 0.15), (0.15, 0.2), (0.20, 0.5), (0.5, 1)]

Ο συνδυασμός τον παραπάνω δίνει έναν χώρο καταστάσεων μήκους 3 x 6 = 18 και συνεπώς ο πίνακας με τις τιμές Q έχει 18states x 7actions = 126 καταχωρήσεις.

7.3 Παράμετροι και Αποτελέσματα Πειραμάτων

Μερικές παράμετροι που έπρεπε να ληφθούν υπόψιν κατά την διάρκεια της εκπαίδευσης ήταν το χρονικό διάστημα ανάμεσα σε διαδοχικές αποφάσεις του πράκτορα t_{ag}, ο ρυθμός μάθησης α, ο παράγοντας προεξόφλησης γ, ο λόγος εξερεύνησης ε καθώς και η διάρκεια εξερεύνησης. Βρέθηκε πως τα βέλτιστα αποτελέσματα παρουσιάστηκαν με τις ακόλουθες επιλογές:

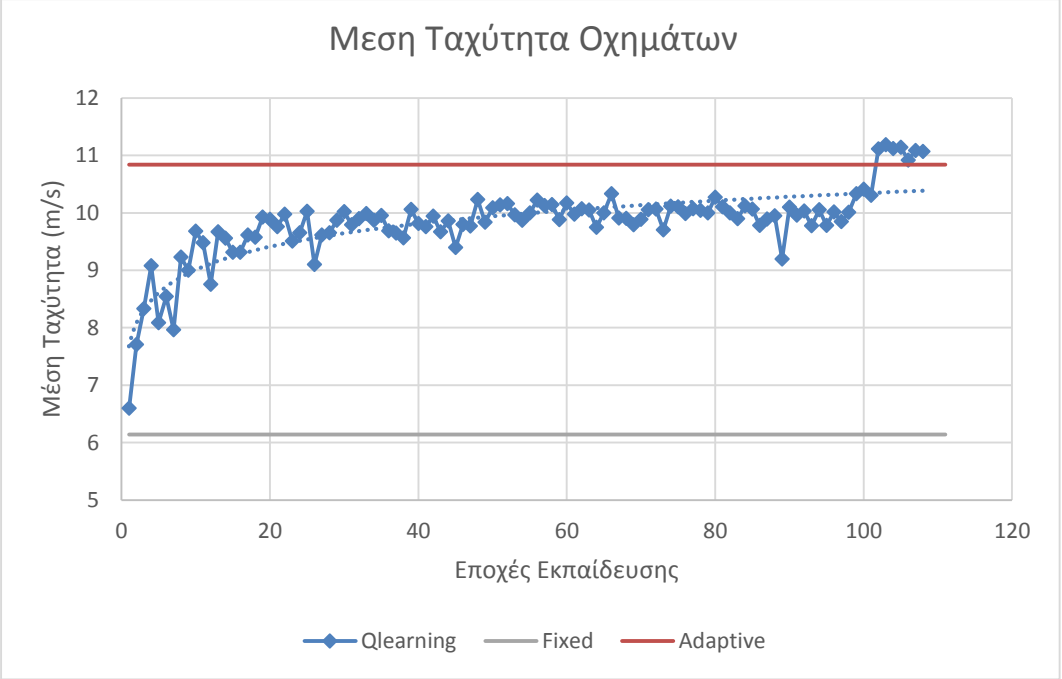
t_{ag} = 5min
α = 0.1

$\gamma = 0.1$

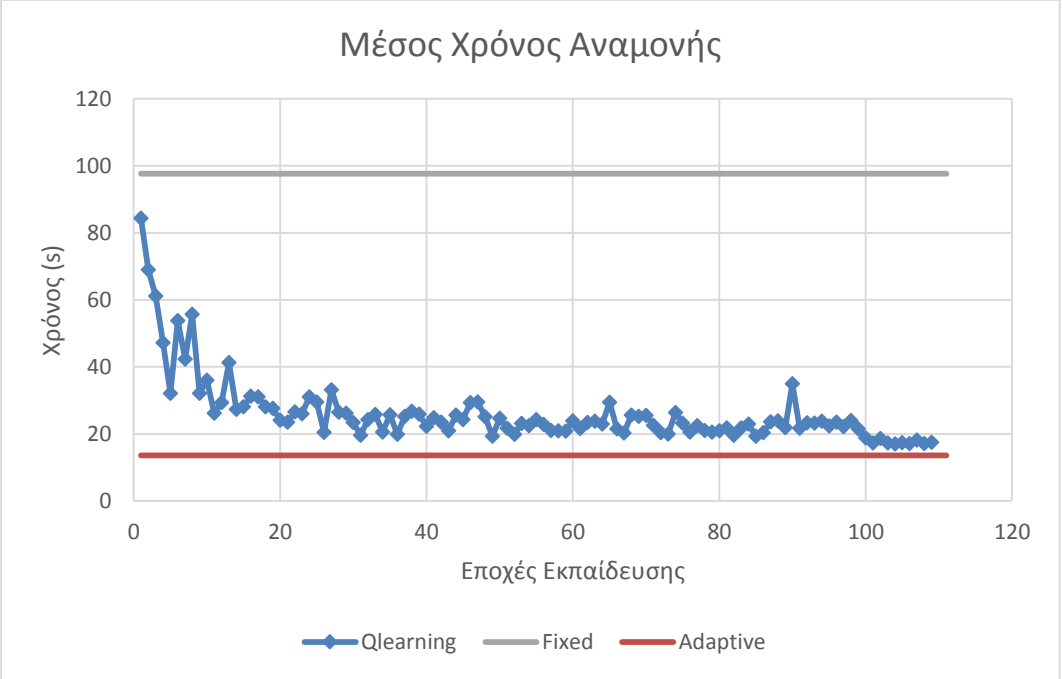
$\varepsilon = 0.1$

$t_{exploration} = 100$ εποχές

Ως εκτίμηση της απόδοσης του δικτύου χρησιμοποιήθηκαν δύο μεταβλητές: Η μέση ταχύτητα των οχημάτων κατά την κίνησή τους στο οδικό δίκτυο της προσομοίωσης καθώς και ο μέσος χρόνος αναμονής (ταχύτητα κάτω από 0.1m/s) σε δευτερόλεπτα. Μετά το πέρας των 100 εποχών εκπαίδευσης, ο πράκτορας επιλέχθηκε να πραγματοποιεί αποφάσεις κάθε 1 λεπτό, δηλαδή για κάθε κύκλο του φαναριού. Δοκιμάστηκε να εκπαιδευτεί το σύστημα με αυτό το διάστημα αποφάσεων εξαρχής (δηλαδή με $t_{ag} = 1$min) αλλά τα αποτελέσματα ήταν ελαφρώς υποδεέστερα. Μια πιθανή αιτία είναι πως η γρήγορη εναλλαγή αποφάσεων του πράκτορα σε συνδυασμό με την στοχαστική φύση της κίνησης κατέστησε δύσκολη την αποτελεσματικό συμπερασμό της αναμενόμενης επιβράβευσης για κάθε ενέργεια. Ακολουθεί το διάγραμμα της απόδοσης του δικτύου κατά την εκπαίδευσή του για τις παραπάνω παραμέτρους σε σύγκριση με την απόδοση των δύο άλλων συστημάτων ελέγχου της διασταύρωσης (σταθερού χρονισμού, adaptive):

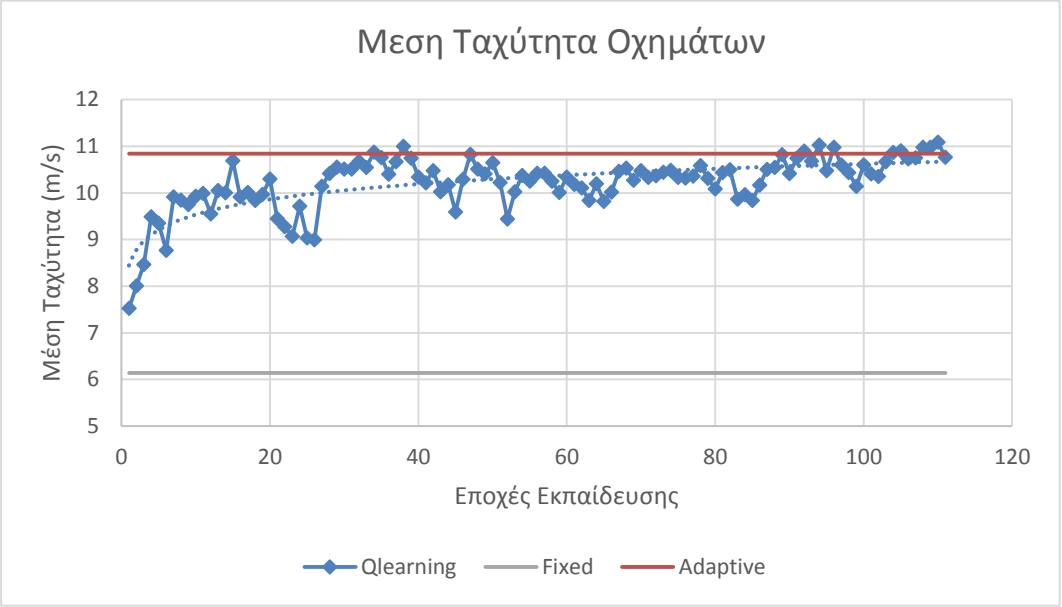


Εικόνα 13 - Μέση ταχύτητα οχημάτων (5min)

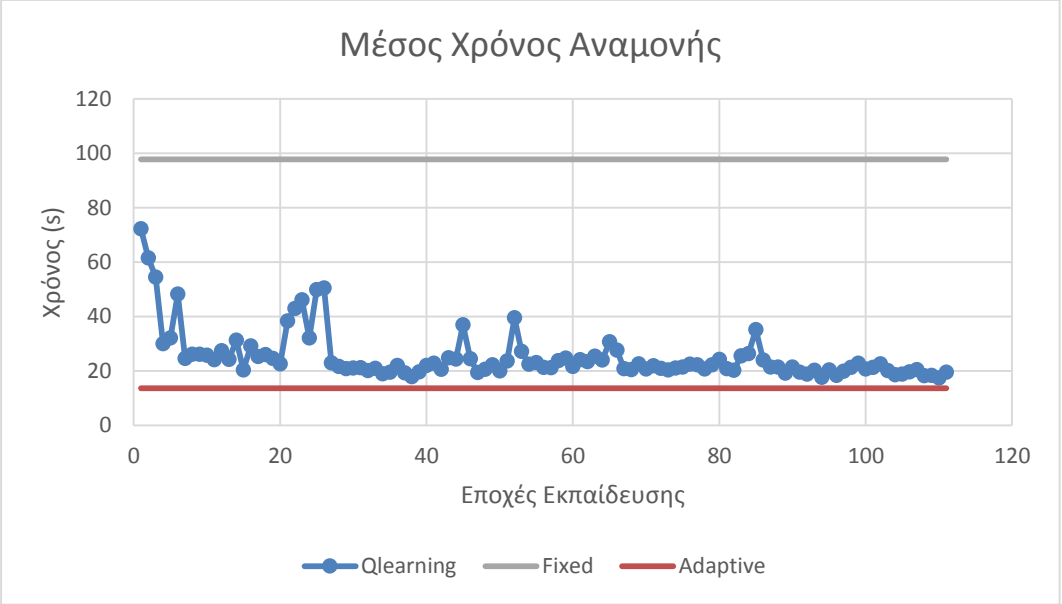


Εικόνα 14 - Μέσος χρόνος αναμονής (5min)

Ακολουθούν για λόγους σύγκρισης και τα αντίστοιχα διαγράμματα, όπου η εκπαίδευση πραγματοποιήθηκε εξαρχής με χρόνο 1 λεπτό ανάμεσα στις αποφάσεις του ευφυούς πράκτορα:



Εικόνα 15 - Μέση ταχύτητα οχημάτων (1min)



Εικόνα 16 - Μέσος χρόνος αναμονής (1min)

Με βάση τα διαγράμματα που παρατέθηκαν στο προηγούμενο κεφάλαιο, παρατηρούμε πως το σύστημα ελέγχου της διασταύρωσης με χρήση του αλγορίθμου Q-learning καταφέρνει να αποδώσει ιδιαίτερα καλά. Παρουσιάζει βελτίωση 82% σε σχέση με τα φανάρια σταθερού χρονισμού (fixed) και 3.2% σε σχέση με τον προσαρμοστικό αλγόριθμο (adaptive) αναφορικά με την μέση ταχύτητα κίνησης των οχημάτων. Αναφορικά με τον μέσο χρόνο αναμονής των οχημάτων παρατηρείται σε σύγκριση με τον έλεγχο σταθερού χρονισμού ότι απαιτείται περίπου 6 φορές λιγότερος χρόνος. Ο προσαρμοστικός αλγόριθμος φαίνεται να επιτυγχάνει χαμηλότερους χρόνους αναμονής αν και υστερεί στην μέση ταχύτητα. Αυτό συμβαίνει καθώς ο τρόπος λειτουργίας του εξαναγκάζει τα οχήματα σε κίνηση με χαμηλή ταχύτητα για περισσότερο χρόνο σε αντίθεση με τον αλγόριθμο Q-learning που «προτιμάει» να διατηρήσει τα οχήματα ακινητοποιημένα για λίγο μεγαλύτερο χρονικό διάστημα και να τους επιτρέψει να κινηθούν τελικά με μεγαλύτερη μέση ταχύτητα. Αυτή η πρακτική θα μπορούσε να έχει και θετικές συνέπειες στην μέση κατανάλωση των οχημάτων και στους εκπεμπόμενους ρύπους καθώς και στην ταλαιπωρία των οδηγών.

Τα διαγράμματα που παρουσιάζουν την απόδοση του Q-learning για εκπαίδευση με μικρότερα διαστήματα απόφασης (1 λεπτό) δείχνουν παρόμοια αποτελέσματα με ελαφρώς υποδεέστερη επίδοση.

ΚΕΦΑΛΑΙΟ 8ο – ΣΥΜΠΕΡΑΣΜΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΕΡΕΥΝΑ

Στην παρούσα εργασία πραγματοποιήθηκε μια εκτενής μελέτη του προβλήματος της διαχείρισης της οδικής κυκλοφορίας και των μεθόδων που εφαρμόζονται για την επίλυσή του. Χρησιμοποιήθηκε ο αλγόριθμος Q-learning της ενισχυτικής μάθησης στην ε-greedy παραλλαγή του με σκοπό να εκπαιδευτεί κατάλληλα ένας πράκτορας και να ελέγξει μια διασταύρωση τεσσάρων κατευθύνσεων. Σαν σημείο αναφοράς χρησιμοποιήθηκαν δύο άλλες μέθοδοι ελέγχου της ίδιας διασταύρωσης – ένας αλγόριθμος σταθερού χρονισμού και ένας προσαρμοστικός. Η λύση που προτάθηκε από την εργασία αυτή παρουσίασε σημαντικές βελτιώσεις στην απόδοση σε σχέση και με τις άλλες δύο μεθόδους, δείχνοντας ότι ευφυείς τεχνικές και, ειδικότερα, της ενισχυτικής μάθησης είναι ιδιαίτερα κατάλληλες για την επίλυση προβλημάτων όπως το κυκλοφοριακό.

Ένα επόμενο βήμα στην κατεύθυνση αυτή είναι να χρησιμοποιηθεί ο παραπάνω αλγόριθμος για τον έλεγχο διασταυρώσεων μεγαλύτερων δικτύων ώστε να επιτευχθεί η πιο ρεαλιστική αξιολόγηση της απόδοσής του σε σενάρια πραγματικού κόσμου. Εν συνεχεία μπορούν οι αλγόριθμοι που θα εξαχθούν να δοκιμαστούν σε προσομοιώσεις υπαρκτών οδικών δικτύων και να συγκριθεί η απόδοσή τους σε σχέση με τα ήδη χρησιμοποιούμενα συστήματα (SCOOT, TRANSYT κ.α.). Στην περίπτωση που βρεθούν επαρκώς αποδοτικές τεχνικές, θα μπορούσε πιλοτικά να εφαρμοστεί σε κάποια αστική περιοχή ένας αλγόριθμος ενισχυτικής μάθησης για να παρατηρηθεί κατά πόσο η απόδοσή του σε πραγματικές συνθήκες συνάδει με τα αποτελέσματα των προσομοιώσεων και να πραγματοποιηθούν τυχόν βελτιστοποιήσεις.

Παράλληλα ο τομέας της διαχείρισης της οδικής κυκλοφορίας είναι πολύ πρόσφορος για την δοκιμή και άλλων ευφυών τεχνικών ή συνδυασμό τους. Μια λογική επέκτασης της εργασίας αυτής θα ήταν να χρησιμοποιηθεί ένα νευρωνικό δίκτυο το οποίο θα επεξεργάζεται διάφορα στοιχεία της εξόδου του δικτύου (ουρές, σταθμευμένα

οχήματα, μέση ταχύτητα κλπ.) και θα αποφασίζει για το state το οποίο θα χρησιμοποιήσει ο αλγόριθμος του Q-learning ως τρέχον. Έτσι θα μπορούσε να διορθωθεί το πρόβλημα της a priori επιλογής του συνόλου των καταστάσεων που συναντήθηκε στην παρούσα εργασία. Ακόμη, θα μπορούσε να δοκιμαστεί ένας συνδυασμός της μεθόδου που προτάθηκε στην εργασία αυτή, με ένα συντονιζόμενο πολυπρακτορικό σύστημα το οποίο θα αναθέτει έναν πράκτορα σε κάθε όχημα για την εξαγωγή της κατάστασης του δικτύου και στην συνέχεια θα πραγματοποιεί την επεξεργασία του συνόλου των δεδομένων μέσω ενός νευρωνικού δικτύου και θα πραγματοποιεί τις τελικές αποφάσεις μέσω ενός πράκτορα - συντονιστή βασισμένο στον αλγόριθμο Q-learning. Για τον αλγόριθμο Q-learning καθ' αυτό θα μπορούσαν να δοκιμαστούν ένα σύνολο διαφορετικών τεχνικών εξερεύνησης (exploration). Κατά την εκπόνηση της παρούσας εργασίας χρησιμοποιήθηκε κατ' εξοχήν η λογική της «άπληστης» προσέγγισης (ε-greedy) αλλά ίσως θα μπορέσουν να αποδειχτούν πιο αποδοτικές για την ταχεία εκπαίδευση του πράκτορα, άλλες τεχνικές που αναφέρθηκαν στο κεφάλαιο 3.3.1 .

ΒΙΒΛΙΟΓΡΑΦΙΑ

Abdulhai, B. P. (2003). Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering, 129(3),* σσ. 278-285.

Arel, I. L. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *Intelligent Transport Systems, IET, 4(2),* σσ. 128-135.

Bien, K. L. (1995). *A Corner Matching Algorithm Using Fuzzy Logic.* Ανάκτηση από www.bioele.nuee.nagoya-u.ac.jpwsc1papersfileslee.ps.gz

Ceylan, H. &. (2004). Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing. *Transportation Research Part B: Methodological, 38(4),* σσ. 329-342.

Clark, S. (2003). Traffic prediction using multivariate nonparametric regression. *Journal of transportation engineering, 129(2),* σσ. 161-168.

Coifman, B. B. (1998). A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies, 6(4),* σσ. 271-288.

de Oliveira, D. B. (2006, December). Reinforcement Learning based Control of Traffic Lights in Non-stationary Environments: A Case Study in a Microscopic Simulator. *EUMAS*.

Drew, D. (1968). *Traffic flow theory and control.* New York: McGraw-Hill.

Foy, M. D. (1992). Signal timing determination using genetic algorithms. *Transportation Research Record, (1365)*.

Gartner, C. S. (1996). MULTIBAND-96 A program for variable bandwidth progression optimization of multiarterial traffic networks. *U.S. Dept. Transp., Washington, DC, Transp. Res. Record 1554*.

Gershenson, C. (2005). Self-organizing traffic lights. *Complex Systems 16(1)*, σσ. 29-53.

Gershenson, C. R. (2012). Self-organizing traffic lights at multiple-street intersections. *Complexity, 17(4)*, σσ. 23-39.

Institute of Transportation Systems. (n.d.). Ανάκτηση από SUMO - Simulation of Urban MObility : dlr.de/ts/sumo

Kallberg, H. (1971). *Traffic simulation.* Helsinki University of Technology, Transportation Engineering. Espoo.

Kamijo, S. M. (2000). Traffic monitoring and accident detection at intersections. *Intelligent Transportation Systems, IEEE Transactions on, 1(2)*, σσ. 108-118.

Kuyer, L. W. (2008). Multiagent reinforcement learning for urban traffic control using coordination graphs. *Machine Learning and Knowledge Discovery in Databases*, σσ. 656-671.

Little, J. D. (1966). The synchronization of traffic signals by mixedinteger-linear-programming. *Oper. Res., vol. 14*, σσ. pp. 568–594.

Liu, Z. (2007). A survey of intelligence methods in urban traffic signal control. *IJCSNS International Journal of Computer Science and Network Security, 7(7)*, σσ. 105-112.

Min, W. W. (2011). Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies, 19(4)*, σσ. 606-616.

Minoarivelo, O. H. (2009). Application of Markov Decision Processes to the Control of a Traffic Intersection. University of Barcelona, Spain.

N. H. Gartner, S. F. (1991). A multiband approach to arterial traffic signal optimization. *Transp. Res.*, σσ. 55–74.

Ni, D. (2003). 2DSIM: a prototype of nanoscopic traffic simulation. . *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, σσ. 47-52.

Pan, J. P. (2013). Proactive vehicular traffic rerouting for lower travel time. *Vehicular Technology, IEEE Transactions on, 62(8)*, σσ. 3551-3568.

Papageorgiou, M. a. (2003). 'Review of road traffic control strategies.',, 91 (12). pp. 2043-2067. *Proceedings of the IEEE., 91 (12)*, σσ. 2043-2067.

Płaczek, B. (2014). A self-organizing system for urban traffic control based on predictive interval microscopic mode. *Engineering Applications of Artificial Intelligence, 34*, σσ. 75-84.

Prashanth, L. A. (2011). Reinforcement learning with function approximation for traffic signal control. *Intelligent Transportation Systems, IEEE Transactions on, 12(2)*, σσ. 412-421.

Ratrout, N. T. (2009). A comparative analysis of currently used microscopic and macroscopic traffic simulation software. *The Arabian Journal for Science and Engineering, 34(1B)*, σσ. 121-133.

Richter, S. (2006). Learning traffic control - towards practical traffic control using policy gradients. *Albert-Ludwigs-Universitat Freiburg, Tech. Rep*.

Robertson, D. I. (1969). TRANSYT method for area traffic control. *Traffic Engineering & Control, 10276–281*.

Salkham, A. A. (2008). A collaborative reinforcement learning approach to urban traffic control optimization. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, σσ. 560-566.

SCOOT-UTC. (n.d.). Ανάκτηση από http://www.scoot-utc.com/documents/1_SCOOT-UTC.pdf.

Somasundaram, G. S. (2013). Classification and Counting of Composite Objects in Traffic Scenes Using Global and Local Image Analysis. *Intelligent Transportation Systems, IEEE Transactions on, 14(1)*, σσ. 69-81.

Steingrover, M. S. (2005, October). Reinforcement Learning of Traffic Light Controllers Adapting to Traffic Congestion. *BNAIC* , σσ. 216-223.

T. L. Anderson, C. W. (1996). Traffic Light Control Using SARSA with Three State Representations,. *Tech. report, Colorado State University, Computer Science Department*.

Teklu, F. S. (2007). A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer‐Aided Civil and Infrastructure Engineering, 22(1)*, σσ. 31-43.

Thrun, S. B. (1992). Efficient exploration in reinforcement learning. *Technical report CMU-CS-92-102*. School of Computer Science, Carnegie-Mellon University.

Wang, H. L. (2013). Self-organized traffic signal coordinated control based on interactive and distributed subarea. *Applied Mechanics and Materials, 241*, σσ. 2031-2037.

Watkins, C. J. (1992). Q-learning. *Machine learning, 8(3-4)*, σσ. 279-292.

Yulianto, B. (2003). Application of fuzzy logic to traffic signal control under mixed traffic conditions. *Traffic Engineering and Control, 44(9)*, σσ. 332-335.