ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Προσομοίωση και βελτιστοποίηση συστήματος διαχείρισης έξυπνης αποθήκης με αυτόνομα ρομποτικά οχήματα

ΣΠΥΡΙΔΩΝ ΑΘ. ΠΑΠΑΣΥΚΙΩΤΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιρίτας Νικόλαος
Επίκουρος Καθηγητής

Λαμία 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Προσομοίωση και βελτιστοποίηση συστήματος διαχείρισης έξυπνης αποθήκης με αυτόνομα ρομποτικά οχήματα

ΣΠΥΡΙΔΩΝ ΑΘ. ΠΑΠΑΣΥΚΙΩΤΗΣ

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Simulation and Optimization of AMRs Management System for Warehouse Environments

SPYRIDON ATH. PAPASYKIOTIS

ΥΠΕΥΘΥΝΟΣ

Τζιρίτας Νικόλαος
Επίκουρος Καθηγητής

FINAL THESIS

ADVISOR

Tziritas Nikolaos
Assistant Professor

Lamia 2024

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 26/3/2024

Ο Δηλών

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Καθώς η ανάγκη για ταχύτερη και αποτελεσματικότερη αποθήκευση και ανάκτηση φορτίων συνεχίζει να αυξάνεται, η αυτοματοποίηση αποθηκών γίνεται όλο και πιο απαραίτητη. Τα αυτόνομα Ρομποτικά οχήματα (AMR) αναδύονται ως μια πολλά υποσχόμενη λύση λόγω της ικανότητάς τους να πλοηγούνται και να εκτελούν διάφορες εργασίες εντός των αποθηκών. Η εργασία αυτή έχει ως στόχο να διερευνήσει τα οφέλη και τις προκλήσεις της χρήσης των αυτόνομων ρομποτικών οχημάτων για την αυτοματοποίηση αποθηκών, καθώς και να αναπτύξει ένα αλγόριθμο που επιτρέπει την δυναμική πλοήγηση των AMR σε ένα περιβάλλον έξυπνης αποθήκης.

Επιπλέον, η εργασία εμβαθύνει στα πλεονεκτήματα των αυτόνομων ρομποτικών οχημάτων σε δυναμικά περιβάλλοντα αποθηκών, σε αντίθεση με τα AS/RS και AGVs, και διερευνά τον τρόπο με τον οποίο παράγονται τα μονοπάτια κίνησης των οχημάτων από τους αλγορίθμους A*, D*, D* Lite και DRRT.

Για να αποκτήσουμε εικόνα για τη λειτουργία των αυτόνομων ρομποτικών οχημάτων σε μια αποθήκη, αναπτύσσεται ένας αλγόριθμος εύρεσης και επίλυσης πιθανών συγκρούσεων πάνω στα υπολογισμένα μονοπάτια. Ο αλγόριθμος αυτός εκτελείται ως πείραμα σε ένα δισδιάστατο χώρο, αξιολογεί τις επιδόσεις των παραπάνω αλγορίθμων σχεδιασμού μονοπατιού (A*, D*, D* Lite, DRRT), σε διάφορα σενάρια και συλλέγει στατιστικά για την περαιτέρω διερεύνηση και εξαγωγή συμπερασμάτων. Στη εκτέλεση αυτή ενσωματώνονται οι ξεχωριστές διατάξεις αποθηκών που δημιουργήθηκαν.

Εν κατακλείδι, η παρούσα εργασία παρέχει μια διεξοδική εξέταση για πιθανά οφέλη των αυτόνομων ρομποτικών οχημάτων στην αυτοματοποίηση δυναμικών αποθηκών και προσφέρει μια πρακτική προσέγγιση για την μελέτη της συμπεριφοράς των αλγορίθμων σχεδιασμού μονοπατιού σε αποθήκες με πολλαπλά αυτόνομα ρομποτικά οχήματα.

ABSTRACT

As the need for faster and more efficient storage and retrieval of shipments continues to grow, warehouse automation is becoming increasingly necessary. Autonomous Mobile Robots (AMR) are emerging as a promising solution due to their ability to navigate and perform various tasks within warehouses. This thesis aims to investigate the benefits and challenges of using autonomous robotic vehicles for warehouse automation, as well as to develop an algorithm that enables the dynamic navigation of AMRs in a smart warehouse environment.

In addition, the paper delves into the advantages of autonomous robotic vehicles in dynamic warehouse environments, as opposed to AS/RS and AGVs, and explores how vehicle motion paths are generated by the A*, D*, D* Lite and DRRT algorithms.

To gain insight into the operation of autonomous robotic vehicles in a warehouse, an algorithm for finding and resolving potential conflicts on the computed paths is developed. This algorithm is executed as an experiment in a two-dimensional space, evaluates the performance of the above path planning algorithms (A*, D*, D* Lite, DRRT) in different scenarios and collects statistics for further investigation and inference. The separate warehouse layouts created are integrated into this execution.

In conclusion, this thesis provides a thorough examination of the potential benefits of autonomous robotic vehicles for the automation of dynamic warehouses and offers a practical approach to study the behavior of path planning algorithms in warehouses with multiple autonomous mobile robots.

Table of Contents

ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

1.1 Ιστορική εξέλιξη της αυτοματοποίησης των αποθηκών

Η αυτοματοποίηση αποθηκών έχει μακρά ιστορία που εκτείνεται πάνω από έναν αιώνα. Η πρώτη μορφή αυτοματοποίησης αποθηκών ξεκίνησε στα τέλη του 19ου αιώνα, με την εισαγωγή μηχανικών ιμάντων μεταφοράς και περονοφόρων ανυψωτικών μηχανημάτων. Τα μηχανήματα αυτά βοήθησαν στην αυτοματοποίηση της μετακίνησης των εμπορευμάτων εντός της αποθήκης και επέτρεψαν μεγαλύτερη αποτελεσματικότητα στην αποθήκευση και ανάκτηση των προϊόντων.

Τις δεκαετίες του 1960 και 1970, η μηχανοργάνωση άρχισε να μεταμορφώνει τις λειτουργίες της αποθήκης. Τα συστήματα υπολογιστών χρησιμοποιήθηκαν για τη διαχείριση των υφιστάμενων αποθεμάτων, την παρακολούθηση της κίνησης των προϊόντων και τη βελτιστοποίηση της αξιοποίησης του αποθηκευτικού χώρου. Αυτό έκανε τις αποθήκες πιο αποτελεσματικές και να μείωσε το κόστος λειτουργίας τους.

Η χρήση ρομπότ στις αποθήκες άρχισε να αυξάνεται τις δεκαετίες του 1980 και του 1990. Τα ρομπότ χρησιμοποιήθηκαν για την αυτοματοποίηση εργασιών όπως η τοποθέτηση των φορτίων σε παλέτες μεταφοράς, η συλλογή παραγγελιών και ο χειρισμός υλικών. Αυτό μείωσε την ανάγκη για ανθρώπινη εργασία και αύξησε την ταχύτητα και την ακρίβεια των εργασιών της αποθήκης.

Η δεκαετία του 1990 αποτέλεσε σημαντικό ορόσημο στην εξέλιξη της αυτοματοποίησης αποθηκών καθώς ενσωματώθηκαν τα αυτοματοποιημένα καθοδηγούμενα οχήματα (AGV) στις λειτουργίες της αποθήκης, μαζί με άλλες τεχνολογίες αυτοματισμού, όπως τα (AS/RS) που προϋπήρχαν. Τα οχήματα AGV συμπλήρωσαν τα συστήματα AS/RS παρέχοντας ευέλικτη και αυτόματη μεταφορά εμπορευμάτων εντός της αποθήκης.

Τα συστήματα AS/RS άρχισαν να εμφανίζονται τη δεκαετία του 1980 και συνεχίζουν να χρησιμοποιούνται και σήμερα. Μπορούν να αποθηκεύουν και να ανακτούν προϊόντα γρήγορα και αποτελεσματικά, γεγονός που αυξάνει την παραγωγικότητα.

Τα τελευταία χρόνια, έχουν γίνει όλο και πιο δημοφιλή στην αυτοματοποίηση αποθηκών τα αυτόνομα ρομποτικά οχήματα (AMR), τα οποία μπορούν να κινούνται αυτόνομα σε γνωστά καθώς και άγνωστα περιβάλλοντα και να εκτελούν εργασίες όπως η συλλογή και η μεταφορά προϊόντων. Τα AMR βελτιώνουν την αποδοτικότητα μίας αποθήκης. Επίσης μειώνουν το κόστος της αυτοματοποίησης, περιορίζουν την ανάγκη για ανθρώπινη εργασία και αυξάνουν την ταχύτητα και την ακρίβεια της ανάκτησης και αποθήκευσης των προϊόντων.

Εν τέλει, η εξέλιξη της αυτοματοποίησης των αποθηκών καθοδηγείται από την ανάγκη για όλο και μεγαλύτερη αποδοτικότητα, ταχύτητα και ακρίβεια στις λειτουργίες της αποθήκης. Καθώς η τεχνολογία συνεχίζει να εξελίσσεται, μπορούμε να αναμένουμε περαιτέρω εξελίξεις στην αυτοματοποίηση αποθηκών που θα συνεχίσουν να αναβαθμίζουν τον κλάδο. Η ενσωμάτωση τεχνολογιών αυτοματισμού, όπως τα συστήματα AS/RS, τα οχήματα AGV και AMR καθώς και η τεχνητή νοημοσύνη αλλάζει τον τρόπο λειτουργίας των αποθηκών και βοηθά τις επιχειρήσεις να επιτύχουν τους στόχους τους.

1.2 Προκλήσεις κατά την εξέλιξη της αυτοματοποίησης αποθηκών

Στα πρώτα στάδια της αυτοματοποίησης των αποθηκών, οι τεχνολογικές δυνατότητες ήταν περιορισμένες. Οι διαθέσιμες τεχνολογίες συχνά δεν διέθεταν την απαιτούμενη πολυπλοκότητα για σύνθετες λειτουργίες. Τα ζητήματα συμβατότητας μεταξύ υλικού και λογισμικού, οι περιορισμοί στην επεξεργαστική ισχύ και η πολυπλοκότητα στην ανάπτυξη λογισμικού αποτελούσαν σημαντικές προκλήσεις. Η υπέρβαση αυτών των εμποδίων απαιτούσε εξελίξεις στην ταχύτητα των υπολογιστικών κυκλωμάτων, την τεχνολογία αισθητήρων και τους αλγορίθμους διαχείρισης των πόρων της αποθήκης.

Εξίσου σημαντικό εμπόδιο για την ευρεία υιοθέτηση της αυτοματοποίησης αποθηκών αποτέλεσε το υψηλό κόστος υλοποίησης. Το κόστος απόκτησης και εγκατάστασης υλικού αυτοματισμού, ανάπτυξης ή παραμετροποίησης λογισμικού και τροποποίησης της υπάρχουσας υποδομής ήταν σημαντικό. Αυτό καθιστούσε τις λύσεις αυτοματισμού οικονομικά απαγορευτικές για πολλές μικρές και μεσαίες επιχειρήσεις. Ωστόσο, με την πρόοδο της τεχνολογίας και την επίτευξη οικονομιών κλίμακας, το κόστος που σχετίζεται με την αυτοματοποίηση μειώθηκε, καθιστώντας την πιο προσιτή σε ένα ευρύτερο φάσμα επιχειρήσεων.

1.3 Σύγχρονες τεχνικές αυτοματοποίησης αποθηκών

Οι σύγχρονες αποθήκες εξοπλίζονται με όλο και περισσότερες τεχνικές αυτοματοποίησης για τη βελτιστοποίηση των λειτουργιών και τη αύξηση της αποδοτικότητας. Αυτές οι καινοτόμες τεχνικές αλλάζουν τις παραδοσιακές διαδικασίες αποθήκευσης και προσφέρουν σημαντικά οφέλη.

Ο ρομποτικός αυτοματισμός χρησιμοποιείται επίσης για εργασίες όπως η συλλογή, η συσκευασία και η διαλογή. Με την χρήση ρομπότ βελτιώνεται η ακρίβεια, η ταχύτητα και μειώνονται τα ανθρώπινα λάθη, επιτρέποντας στους ανθρώπινο δυναμικό να επικεντρωθεί σε δραστηριότητες υψηλότερης σημασίας.

Τα αυτόματα καθοδηγούμενα οχήματα (AGV) και τα αυτόνομα ρομποτικά οχήματα (AMR) μεταφέρουν τα εμπορεύματα, βελτιστοποιώντας τη ροή κίνησης των φορτίων, ενισχύοντας την ασφάλεια και ελαχιστοποιώντας τον χρόνο αδράνειας της λειτουργίας των αποθηκών. Τα αυτοματοποιημένα συστήματα αποθήκευσης και ανάκτησης (AS/RS) μεγιστοποιούν τη χρήση του χώρου, αυξάνουν την αποθηκευτική ικανότητα και βελτιώνουν την ταχύτητα και την ακρίβεια εκτέλεσης των παραγγελιών.

Τα λογισμικά διαχείρισης αποθήκης (WMS) και διαχείρισης αποθεμάτων ενσωματώνουν και αυτοματοποιούν λειτουργίες όπως η παρακολούθηση αποθεμάτων, η διαχείριση παραγγελιών και παρέχουν εποπτεία σε πραγματικό χρόνο. Επίσης καθιστούν εφικτή τη λήψη αποφάσεων βάσει των δεδομένων που συλλέγουν.

Η τεχνητή νοημοσύνη (AI) και η μηχανική μάθηση βελτιστοποιούν τις ροές εργασίας μέσω της ανάλυσης δεδομένων και προβλέπουν πρότυπα ζήτησης. Επιτρέπουν τη δυναμική κατανομή πόρων, ενισχύοντας τη λειτουργική αποδοτικότητα και μειώνοντας το κόστος.

Συνοψίζοντας, η υιοθέτηση τεχνικών αυτοματοποίησης στις αποθήκες, συμπεριλαμβανομένων των AS/RS, AGV, AMR, WMS και της τεχνητής νοημοσύνης, βελτιώνει σημαντικά την ακρίβεια και την αποδοτικότητα και δίνει τη δυνατότητα στις αποθήκες να παραμείνουν ανταγωνιστικές και να ανταποκριθούν στις εξελισσόμενες απαιτήσεις του σύγχρονου επιχειρηματικού περιβάλλοντος.

ΚΕΦΑΛΑΙΟ 2 Συστήματα αυτοματοποίησης αποθηκών

2.1 Αυτοματοποιημένα συστήματα αποθήκευσης και ανάκτησης (AS/RS)

Τα AS/RS είναι ένα αυτοματοποιημένο σύστημα αποθήκευσης και ανάκτησης προϊόντων. Πρόκειται για ένα σύστημα ελεγχόμενο από υπολογιστικό κύκλωμα που χρησιμοποιείται σε αποθήκες και κέντρα διανομής για την αποτελεσματική διαχείριση των αποθεμάτων. Είναι σχεδιασμένα να χειρίζονται ένα ευρύ φάσμα προϊόντων, συμπεριλαμβανομένων παλετών, κιβωτίων, δοχείων και μεμονωμένων αντικειμένων.

2.1.1 Δομικά μέρη ενός AS/RS συστήματος

Μηχανές αποθήκευσης και ανάκτησης (SRM):

Τα SRM είναι ρομποτικές μονάδες υπεύθυνες για την ανάκτηση και την αποθήκευση αντικειμένων από καθορισμένα σημεία της αποθήκης. Υπάρχουν διάφοροι τύποι SRM, συμπεριλαμβανομένων γερανών στοίβαξης, συστημάτων μεταφοράς και ρομποτικών βραχιόνων. Τα μηχανήματα αυτά είναι εξοπλισμένα με ειδικές λαβές καθώς επίσης και ιμάντες μεταφοράς για τον χειρισμό διαφόρων τύπων φορτίων.

Δομές και ράφια αποθήκευσης:

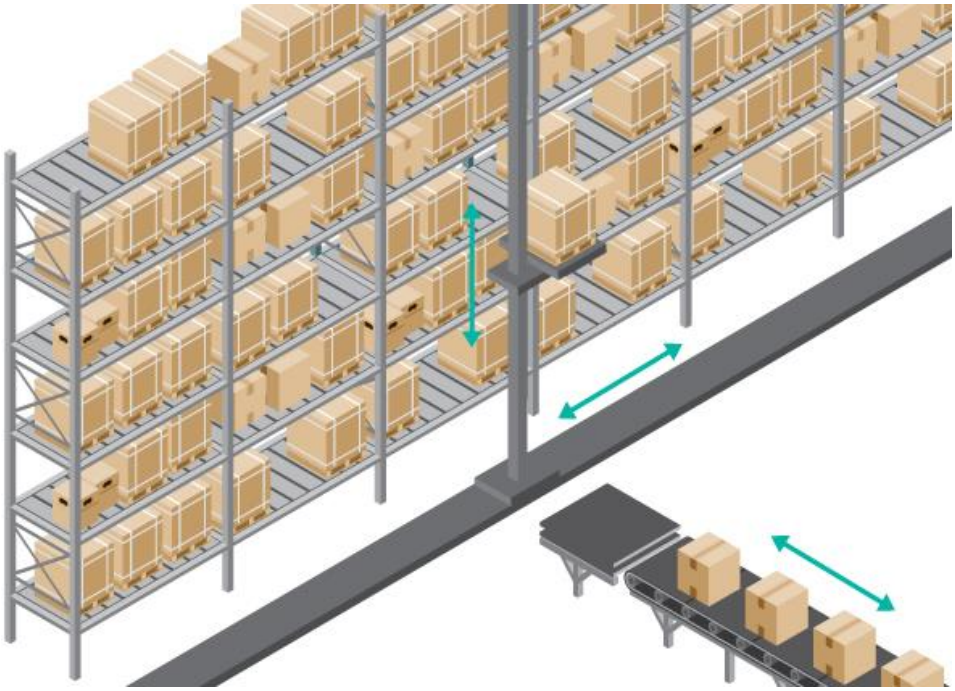
Τα συστήματα AS/RS χρησιμοποιούν εξειδικευμένες δομές αποθήκευσης και ράφια για τη καλύτερη χρήση του χώρου. Οι δομές αυτές περιλαμβάνουν υψηλά ράφια, μονάδες κατακόρυφης ανύψωσης (VLM). Έχουν σχεδιαστεί για να φιλοξενούν διαφορετικούς τύπους εμπορευμάτων, είτε πρόκειται για φορτία σε παλέτες, είτε για δοχεία ή μεμονωμένα αντικείμενα.

Λογισμικό και σύστημα ελέγχου:

Το λογισμικό και το σύστημα ελέγχου ενεργούν ως εγκέφαλος του AS/RS, συντονίζοντας τις λειτουργίες των διαφόρων εξαρτημάτων και εξασφαλίζοντας την ομαλή ροή των υλικών. Το λογισμικό αυτό ενσωματώνεται με συστήματα διαχείρισης αποθηκών (WMS) ή συστήματα προγραμματισμού επιχειρησιακών πόρων (ERP) για την παραλαβή και την επεξεργασία παραγγελιών, την παρακολούθηση αποθεμάτων, τη βελτιστοποίηση διαδρομών και την παρακολούθηση της απόδοσης του συστήματος.

Αισθητήρες και μέτρα ασφαλείας:

Τα συστήματα AS/RS διαθέτουν διάφορους αισθητήρες και μέτρα ασφαλείας για να διασφαλίζουν αποτελεσματικές και ασφαλείς λειτουργίες. Σε αυτά περιλαμβάνονται αισθητήρες προσέγγισης για την ανίχνευση της παρουσίας αντικειμένων, δικλείδες ασφαλείας για την πρόληψη ατυχημάτων και συστήματα αποφυγής συγκρούσεων μεταξύ κινούμενων εξαρτημάτων ή αντικειμένων που εμποδίζουν.



Εικόνα 2.1 - Automated Storage & Retrieval System (https://www.posital.com)

2.1.2 Τρόπος λειτουργίας ενός συστήματος AS/RS

Τα συστήματα AS/RS εκτελούν διάφορες βασικές λειτουργίες για την αυτοματοποίηση και τη βελτιστοποίηση των εργασιών αποθήκευσης. Παρακάτω παρουσιάζονται οι πιο σημαντικές λειτουργίες που επιτυγχάνουν:

Παραλαβή και αποθήκευση εμπορευμάτων:

Κατά την παραλαβή εμπορευμάτων, τα συστήματα AS/RS προσδιορίζουν και κατανέμουν αυτόματα θέσεις αποθήκευσης βάσει προκαθορισμένων κανόνων ή αλγορίθμων. Τα SRM ανακτούν τα αντικείμενα και στην συνέχεια τα τοποθετούν σε άλλα σημεία εντός της αποθήκης.

Διαχείριση και παρακολούθηση αποθεμάτων:

Τα συστήματα AS/RS ενσωματώνουν λογισμικό διαχείρισης αποθεμάτων για να παρέχουν τη δυνατότητα ακριβούς παρακολούθηση των αποθεμάτων σε πραγματικό χρόνο. Τα συστήματα καταγράφουν και ενημερώνουν τα επίπεδα αποθεμάτων, τις τοποθεσίες και άλλες σχετικές πληροφορίες, επιτρέποντας τον πλήρη έλεγχο και την εποπτεία τους.

Συλλογή και εκτέλεση παραγγελιών:

Τα συστήματα AS/RS βελτιώνουν τη διαδικασία συλλογής παραγγελιών με την ενοποίηση πολλαπλών παραγγελιών σε μία μόνο διαδρομή. Το λογισμικό ελέγχου καθορίζει την πιο αποτελεσματική διαδρομή για τα SRM για την ανάκτηση των προϊόντων, ελαχιστοποιώντας τον χρόνο μετακίνησης. Αυτό έχει θετικό αποτέλεσμα στην ακρίβεια των παραγγελιών, μειώνοντας τα σφάλματα συλλογής και βελτιώνοντας τα ποσοστά εκπλήρωσης τους.

2.1.3 Πλεονεκτήματα ενός συστήματος AS/RS

Η χρήση ενός προηγμένου συστήματος διαχείρισης φορτίων AS/RS, το οποίο διαθέτει αυτοματοποιημένες τεχνολογίες για την αποθήκευση και ανάκτηση των αποθεμάτων, προσφέρει σημαντικά πλεονεκτήματα στην λειτουργία και τη διαχείριση μιας αποθήκης. Τα πιο σημαντικά από αυτά είναι:

Καλύτερη αξιοποίηση του διαθέσιμου αποθηκευτικού χώρου:

Ένα από τα κύρια πλεονεκτήματα ενός συστήματος AS/RS είναι η ικανότητά του να αυξάνει τη χρήση του διαθέσιμου αποθηκευτικού χώρου. Με την αξιοποίηση του κάθετου χώρου μέσω υψηλών ραφιών και την χρήση προηγμένων αλγορίθμων για τον προσδιορισμό της καλύτερης τοποθέτησης των αντικειμένων, το σύστημα περιορίζει τη σπατάλη χώρου και ελαχιστοποιεί την ανάγκη για επιπλέον αποθηκευτική χωρητικότητα.

Βελτιωμένη διαχείριση αποθεμάτων:

Τα συστήματα AS/RS παρέχουν δυνατότητα παρακολούθησης σε πραγματικό χρόνο και ακριβή έλεγχο των αποθεμάτων. Η αυτοματοποιημένη παρακολούθηση και ο έλεγχος των επιπέδων αποθεμάτων προσφέρουν την ικανότητα να προβλέπουν με ακρίβεια τη ζήτηση, να βελτιώνουν τις διεργασίες ανεφοδιασμού, ελαχιστοποιώντας τις περιπτώσεις εξάντλησης αποθεμάτων καθώς και ανεφοδιασμού με περίσσια φορτία. Με τον τρόπο αυτο διασφαλίζεται ότι τα σωστά προϊόντα είναι διαθέσιμα όταν και όπου χρειάζονται. Ετσι λοιπόν η αυτοματοποίηση των διαδικασιών αποθήκευσης και ανάκτησης αποτρέπουν τα λάθη που σχετίζονται με ανθρώπινες αποφάσεις και με τον χειροκίνητο χειρισμό των φορτίων. Αυτό έχει ως αποτέλεσμα υψηλότερα ποσοστά ακρίβειας παραγγελιών και μειώνει την πιθανότητα αποστολής λανθασμένων ή ελλιπών προϊόντων.

2.2 Αυτοματοποιημένα καθοδηγούμενα οχήματα (AGV)

Τα αυτόματα καθοδηγούμενα οχήματα (AGV) είναι φορητά ρομποτικά οχήματα που έχουν προγραμματιστεί για να διανύουν συγκεκριμένες διαδρομές που προσδιορίζονται από γραμμές, καλώδια ή χρησιμοποιούν διάφορες μεθόδους πλοήγησης, όπως κάμερες όρασης, μαγνήτες, λέιζερ κλπ. Χρησιμοποιούνται κυρίως σε βιομηχανικά περιβάλλοντα, και παίζουν καθοριστικό ρόλο στη μετακίνηση βαρέων φορτίων σε μεγάλες εγκαταστάσεις, όπως εργοστάσια ή αποθήκες.

2.2.1 Δομικά μέρη ενός AGV

Πλαίσιο:

Το πλαίσιο κατασκευής παρέχει στήριξη για όλα τα άλλα εξαρτήματα και στεγάζει το σύστημα πρόωσης του οχήματος, τις μπαταρίες και τα συστήματα ελέγχου. Το πλαίσιο σχεδιάζεται για σταθερότητα, ανθεκτικότητα και ικανότητα μεταφοράς φορτίου, ανάλογα με τη συγκεκριμένη εφαρμογή.

Τροχοί :

Τα AGV χρησιμοποιούν τροχούς για την μετακίνηση. Η φορά της κίνησης των τροχών ενός AGV είναι διπλή, και η στροφή τους γίνεται προς οποιαδήποτε κατεύθυνση, δίνοντας

τους τη δυνατότητα να ελίσσονται σε στενούς χώρους. Η επιλογή του τύπου των τροχών εξαρτάται από παράγοντες όπως η χωρητικότητα τους, το έδαφος και η απαιτούμενη ευελιξία.

Περόνες:

Τα AGV είναι εξοπλισμένα με συστήματα περονών ικανά να ανυψώνουν, να μετακινούν και να στοιβάζουν παλέτες ή κιβώτια. Οι περόνες τους μπορούν να ρυθμίζονται ως προς το πλάτος και το ύψος τους, προκειμένου να προσαρμόζονται σε διαφορετικά μεγέθη φορτίου.

Συστήματα κίνησης:

Τα AGV χρησιμοποιούν για την κίνηση τους κατα κύριο λόγο ηλεκτροκινητήρες. Οι ηλεκτρικοί κινητήρες χρησιμοποιούνται λόγω της αποδοτικότητάς τους, της ακρίβειας που προσφέρουν στην λειτουργία τους και της ευκολίας ενσωμάτωσης σε συστήματα ελέγχου.

Αισθητήρες:

Τα AGV είναι εξοπλισμένα με διάφορους αισθητήρες, για να αντιλαμβάνονται το περιβάλλον τους και να λαμβάνουν ενημερωμένες αποφάσεις πλοήγησης. Αυτό μπορεί να περιλαμβάνει αισθητήρες λέιζερ (LiDAR), αισθητήρες υπερήχων και κάμερες. Αυτοί οι αισθητήρες επιτρέπουν στα AGV να ανιχνεύουν εμπόδια, να αναγνωρίζουν σημεία αναφοράς ώστε να εξασφαλίζουν ασφαλή μετακίνηση.

Συστήματα ελέγχου:

Τα AGV ενσωματώνουν εξελιγμένα συστήματα ελέγχου, συμπεριλαμβανομένων ενσωματωμένων υπολογιστών και μικροεπεξεργαστών, για την επεξεργασία των δεδομένων των αισθητήρων και την εκτέλεση εντολών κίνησης. Τα συστήματα αυτά παρέχουν τις δυνατότητες νοημοσύνης και λήψης αποφάσεων που απαιτούνται για την αυτόματη λειτουργία.

Τεχνολογίες πλοήγησης:

Τα AGV χρησιμοποιούν διάφορες τεχνολογίες όδευσης για τον προσδιορισμό της θέσης τους και την εκτέλεση προκαθορισμένων διαδρομών. Αυτό μπορεί να περιλαμβάνει μετακίνησης καθοδηγούμενη με λέιζερ, όπου το AGV ακολουθεί τις αντανακλάσεις λέιζερ από ανακλαστικές λωρίδες. Συνήθως επιτυγχάνεται με την παρακολούθηση διαδρομών μαγνητικής ταινίας η οποία είναι ενσωματωμένη στο δάπεδο. Η πλοήγηση με βάση την όραση βασίζεται σε κάμερες και αλγορίθμους επεξεργασίας εικόνας για τον εντοπισμό σημείων αναφοράς ή μοτίβων στο δάπεδο.

Μπαταρίες και συστήματα φόρτισης:

Τα AGV τροφοδοτούνται συνήθως από επαναφορτιζόμενες μπαταρίες. Αυτές οι μπαταρίες παρέχουν την απαραίτητη ηλεκτρική ενέργεια για τη λειτουργία του συστήματος προώθησης του AGV, των αισθητήρων και των συστημάτων ελέγχου. Η φόρτιση επιτυγχάνεται σε καθορισμένους σταθμούς φόρτισης εντός της εγκατάστασης. Οι μπαταρίες των οχημάτων, επαναφορτίζονται αυτόματα, κατά τη διάρκεια περιόδων αδράνειας, εξασφαλίζοντας συνεχή λειτουργία χωρίς χειροκίνητη παρέμβαση.

Συστήματα διαχείρισης ενέργειας:

Τα AGV ενσωματώνουν συστήματα διαχείρισης ενέργειας για τη βελτιστοποίηση της κατανάλωσης ενέργειας και την παράταση της διάρκειας ζωής των μπαταριών. Τα συστήματα αυτά παρακολουθούν τα επίπεδα της μπαταρίας, εκτιμούν τις ενεργειακές απαιτήσεις για διάφορες εργασίες και χρησιμοποιούν στρατηγικές εξοικονόμησης ενέργειας, όπως ο έξυπνος σχεδιασμός διαδρομών, με σκοπό τη μεγιστοποίηση της απόδοσης.

2.2.2 Εφαρμογές των AGV σε αποθήκες

Συλλογή παραγγελιών:

Τα AGV απλοποιούν τη διαδικασία συλλογής παραγγελιών στις αποθήκες, καθώς οδηγούνται αυτόνομα στις καθορισμένες θέσεις και συλλέγουν τα απαιτούμενα είδη. Έτσι, εξαλείφεται η ανάγκη για χειροκίνητα καρότσια συλλογής, μειώνοντας το κόστος εργασίας, τα λάθη και το χρόνο της συλλογής. Τα AGV μπορούν να ακολουθήσουν τις πιο αποφελείς διαδρομές, συλλέγοντας πολλά αντικείμενα ταυτόχρονα και εξασφαλίζοντας ταχεία εκπλήρωση των παραγγελιών.

Διαχείριση αποθεμάτων:

Τα AGV διαδραματίζουν κρίσιμο ρόλο στη διαχείριση των αποθεμάτων στις αποθήκες. Μπορούν να σαρώνουν γραμμωτούς κώδικες ή ετικέτες RFID σε προϊόντα στα ράφια, παρέχοντας δεδομένα σε πραγματικό χρόνο σχετικά με τα επίπεδα αποθεμάτων. Τα AGVs που διαθέτουν συστήματα διαχείρισης αποθήκης (WMS) ή λογισμικό διαχείρισης αποθεμάτων επιτρέπουν την ακριβή παρακολούθηση των εμπορευμάτων, μειώνοντας τις απώλειες και αυξάνουν την συνολική ακρίβεια.

Οργάνωση της διάταξης αποθήκης:

Τα AGV συμβάλλουν στην καλύτερη οργάνωση της αποθήκης. Μπορούν να περιηγηθούν σε στενούς χώρους, διαδρόμους και συστήματα ραφιών, επιτρέποντας διαμορφώσεις αποθήκευσης υψηλής πυκνότητας. Τα AGV επιτρέπουν την αποτελεσματική αξιοποίηση του χώρου, εξαλείφουν τον αναξιοποίητο χώρο των διαδρόμων και βοηθούν στην αναδιαμόρφωση της διάταξης της αποθήκης για καλύτερη ροή εργασιών και αυξημένη παραγωγικότητα.

Διασύνδεση με αυτοματοποιημένα συστήματα αποθήκευσης και ανάκτησης (AS/RS):

Τα AGV διασυνδέονται συχνά με συστήματα AS/RS για την αυτοματοποίηση της μετακίνησης των εμπορευμάτων μεταξύ του χώρου αποθήκευσης και του χώρου συλλογής. Τα AGV μεταφέρουν αντικείμενα από το AS/RS στους σταθμούς συλλογής ή στις περιοχές στάσης, εξασφαλίζοντας την απρόσκοπτη ροή των φορτίων και μειώνοντας τον χειρωνακτικό χειρισμό.

Συνεργασία AGV και ανθρώπου:

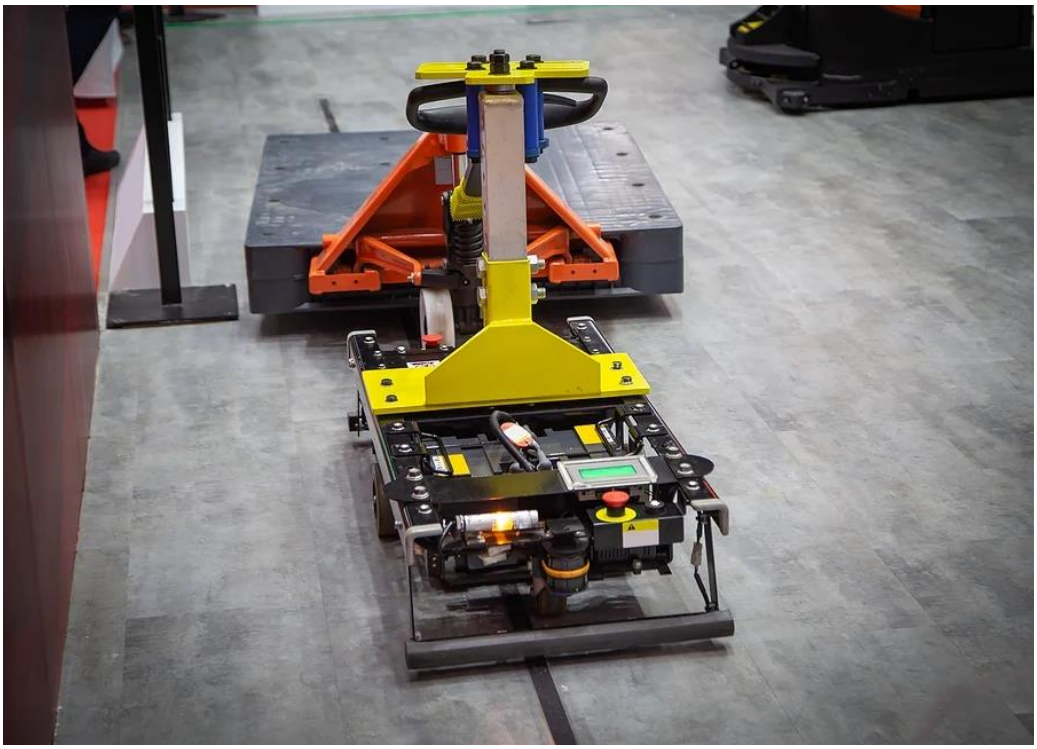
Η συνεργασία μεταξύ των AGV και του ανθρώπινου δυναμικού στις αποθήκες είναι σημαντική στην λειτουργία της αποθήκης. Αξιοποιώντας τα πλεονεκτήματα και των δύο πλευρών, μπορεί να επιτευχθεί στις αποθήκες μια ισορροπία που συμβάλλει στη αύξηση της παραγωγικότητας και της αποδοτικότητας. Η ορθή κατανομή των εργασιών, σε συνδυασμό με τον αποτελεσματικό συντονισμό, επιτρέπει στα AGVs να χειρίζονται επαναλαμβανόμενες

και απαιτητικές εργασίες, ενώ το προσωπικό επικεντρώνεται σε σύνθετες διαδικασίες λήψης αποφάσεων και σε λειτουργίες που απαιτούν προσαρμογή. Αυτή η συνεργασία όχι μόνο ενισχύει την παραγωγικότητα, αλλά και προσφέρει περισσότερη ασφάλεια μέσω της εφαρμογής κανόνων ασφαλείας και διαύλων επικοινωνίας.

2.2.3 Σύγκριση AS/RS και AGV

Τα συστήματα AS/RS περιλαμβάνουν συνήθως δομές ραφιών πολλαπλών επιπέδων, επιτρέποντας την αποτελεσματική κατακόρυφη αποθήκευση. Οι γερανοί στοίβαξης κινούνται κατακόρυφα κατά μήκος των ραφιών, αποκτώντας πρόσβαση στις θέσεις αποθήκευσης για την ανάκτηση ή την αποθήκευση αντικειμένων. Τα ράφια είναι συχνά σχεδιασμένα για να φιλοξενούν παλέτες ή κιβώτια διαφόρων μεγεθών. Στο σύστημα περιλαμβάνονται ιμάντες για τη μεταφορά αντικειμένων μεταξύ των γερανών και άλλων περιοχών της αποθήκης. Τα συστήματα AS/RS είναι γενικά σταθερά στο σχεδιασμό τους και απαιτούν μια ειδική υποδομή για την υποστήριξη της λειτουργίας τους.

Τα AGVs είναι σχεδιασμένα για να λειτουργούν στο δάπεδο της αποθήκης. Μπορούν να μετακινούνται σε προκαθορισμένες διαδρομές χρησιμοποιώντας διάφορα συστήματα, όπως καθοδήγηση με λέιζερ, μαγνητική ή επαγωγική καθοδήγηση ή συστήματα που βασίζονται στην όραση. Τα AGV είναι εξοπλισμένα με ενσωματωμένους αισθητήρες που βοηθούν στον εντοπισμό εμποδίων και εξασφαλίζουν ασφαλή μετακίνηση. Σε αντίθεση με τα AS/RS, τα AGV δεν απαιτούν σταθερή υποδομή, όπως ράφια ή μεταφορείς. Είναι ευέλικτα στην κίνησή τους και μπορούν εύκολα να ελιχθούν σε στενούς διαδρόμους και περιορισμένους χώρους. Έχουν την ικανότητα να προσαρμόζονται σε αλλαγές στη διάταξη της αποθήκης καθώς επίσης και στις διάφορες λειτουργικές απαιτήσεις που μπορεί να παρουσιαστούν.



Εικόνα 2.2 – Automated Guided Vehicle (https://control.com)

2.3 Αυτόνομα ρομποτικά οχήματα (AMR)

2.3.1 Ορισμός

Τα αρχικά AMR αναφέρονται στα αυτόνομα ρομποτικά οχήματα (Autonomous Mobile Robots). Πρόκειται για ρομποτικά οχήματα τα οποία έχουν σχεδιαστεί προκειμένου να λειτουργούν αυτόνομα και να εκτελούν διάφορες εργασίες σε συγκεκριμένο περιβάλλον (βιομηχανικό, αποθηκευτικό κλπ.). Ομοίως με τα AGV, τα AMR είναι εξοπλισμένα με αισθητήρες, κάμερες και τεχνολογίες καθοδήγησης που τους επιτρέπουν να αντιλαμβάνονται το περιβάλλον τους, να λαμβάνουν αποφάσεις και να κινούνται αυτόνομα χωρίς την ανάγκη συνεχούς ανθρώπινης παρέμβασης. Κινούνται μέσα στο περιβάλλον τους αποφεύγοντας τα εμπόδια και προσαρμόζοντας δυναμικά τις διαδρομές τους σε μεταβαλλόμενες συνθήκες. Τα οχήματα AMR είναι συνήθως συνδεδεμένα με ένα κεντρικό σύστημα ελέγχου που αναθέτει εργασίες, παρακολουθεί τα αποθέματα και βελτιστοποιεί την κίνηση των ρομπότ εντός της εγκατάστασης. Αυτό επιτρέπει τον αποτελεσματικό συντονισμό και την ενσωμάτωση των AMR με άλλες λειτουργίες στο χώρο της μιας αποθήκης.

Εικόνα 2.3 – Autonomous Mobile Robot (https://www.designworldonline.com)

2.3.2 Σύγκριση AMR με AGV

Ανίχνευση και αντίληψη του περιβάλλοντος χώρου:

Η αντίληψη του χώρου τόσο στα αυτόματα όσο και στα αυτόνομα ρομποτικά οχήματα περιλαμβάνει την επεξεργασία των δεδομένων που συλλέγονται από τους αισθητήρες για την κατανόηση και την ερμηνεία του περιβάλλοντος. Τα AMR χρησιμοποιούν προηγμένους αλγορίθμους και τεχνικές μηχανικής μάθησης για να εξάγουν σημαντικές πληροφορίες αξιοποιώντας δεδομένα των αισθητήρων. Αυτό περιλαμβάνει τον εντοπισμό εμποδίων, την αναγνώριση αντικειμένων ή σημείων αναφοράς, την κατανόηση της διάταξης της αποθήκης και τη λήψη αποφάσεων.

Τα AGV, από την άλλη πλευρά, βασίζονται συνήθως σε απλούστερα συστήματα ανίχνευσης σε σύγκριση με τα AMR. Τα AGV χρησιμοποιούν συνήθως μαγνητικές ταινίες ή καλώδια ενσωματωμένα στο δάπεδο, γεγονός που περιορίζει τις δυνατότητες αντίληψής τους. Ακολουθούν προκαθορισμένα μονοπάτια ή διαδρομές και οι αισθητήρες τους επικεντρώνονται κυρίως στην τήρηση αυτών των προκαθορισμένων διαδρομών, στην ανίχνευση σημείων αναφοράς και στην αποφυγή συγκρούσεων με τη χρήση βασικών αισθητήρων απόστασης.

Αντιθέτως, τα AMR έχουν πιο δυναμικό και προσαρμοστικό χαρακτήρα. Μπορούν να αντιλαμβάνονται και να ερμηνεύουν το περιβάλλον τους σε πραγματικό χρόνο. Η δυνατότητα αυτή επιτυγχάνεται με την χρήση τεχνολογιών όπως ο ταυτόχρονος εντοπισμός και χαρτογράφηση (SLAM). Συνδυάζοντας δεδομένα από αισθητήρες LIDAR, άλλους αισθητήρες ή κάμερες μαζί με πληροφορίες οδομετρίας, τα AMR μπορούν να δημιουργήσουν έναν χάρτη, ενώ παράλληλα προσδιορίζουν τη θέση τους σε αυτόν.

Ακρίβεια προσδιορισμού της τοποθεσίας:

Ορισμένα AGV χρησιμοποιούν τεχνικές ανίχνευσης σημείων αναφοράς για τον προσδιορισμό της θέσης τους στο περιβάλλον που κινούνται. Αυτό περιλαμβάνει την αναγνώριση προκαθορισμένων σημάνσεων που τοποθετούνται στρατηγικά σε όλη την εγκατάσταση. Τα AGV χρησιμοποιούν αισθητήρες όπως κάμερες ή σαρωτές λέιζερ για να ανιχνεύσουν και να αναγνωρίσουν αυτές τις σημάνσεις, γεγονός που τους επιτρέπει να εκτιμήσουν την ακριβή θέση τους στο χώρο.

Τα AMR χρησιμοποιούν αλγορίθμους εντοπισμού για να εκτιμήσουν την ακριβή θέση και τον προσανατολισμό τους στο περιβάλλον. Αυτό μπορεί να επιτευχθεί μέσω τεχνικών όπως feature matching, scan matching ή μεθόδων όπως τα particle filters. Συγκρίνοντας τα δεδομένα των αισθητήρων με υπάρχοντες χάρτες, τα AMR μπορούν να προσδιορίσουν με ακρίβεια τη θέση τους.

Σχεδιασμός διαδρομής και δημιουργία τροχιάς:

Τα AMR χρησιμοποιούν αλγορίθμους σχεδιασμού διαδρομής για να καθορίσουν τη βέλτιστη διαδρομή από την τρέχουσα θέση τους στον επιθυμητό προορισμό. Αυτοί οι αλγόριθμοι λαμβάνουν υπόψη παράγοντες όπως τα εμπόδια, οι λειτουργικοί περιορισμοί και τα κριτήρια βελτιστοποίησης (π.χ. συντομότερη διαδρομή, ενεργειακή απόδοση). Στη συνέχεια, οι αλγόριθμοι δημιουργίας τροχιάς παράγουν τις συγκεκριμένες εντολές κίνησης που απαιτούνται για την ακριβή εκτέλεση της προγραμματισμένης διαδρομής.

Όπως έχει ήδη αναφερθεί τα AGV περιηγούνται στο περιβάλλον ακολουθώντας προκαθορισμένες διαδρομές. Ακολουθούν αυτές τις σταθερές διαδρομές και βασίζονται στην ανατροφοδότηση από αισθητήρες επί του οχήματος, όπως αισθητήρες προσέγγισης ή

αισθητήρες εντοπισμού γραμμών, ώστε να παραμένουν σε τροχιά και να εντοπίζουν εμπόδια ή ενδείξεις κατά μήκος της διαδρομής.

Συνεργασία με τον άνθρωπο και μέτρα ασφαλείας:

Τα AMR και τα AGV συνεργάζονται με τους ανθρώπους για την καλύτερη και ασφαλέστερη λειτουργία μιας αποθήκης. Τα AMR προσφέρουν ευελιξία, προσαρμοστικότητα και άμεση αλληλεπίδραση, επιτρέποντας τη δυναμική συνεργασία σε διάφορες καταστάσεις. Έχουν σχεδιαστεί για να μετακινούνται αυτόνομα, να προσαρμόζουν τις διαδρομές σε πραγματικό χρόνο και να ανταποκρίνονται στις αλλαγές στο περιβάλλον. Στα AMR μπορούν να ανατεθούν συγκεκριμένοι ρόλοι και καθήκοντα, τα οποία λειτουργούν σε συντονισμό με τους ανθρώπινο δυναμικό, μέσω κεντρικών συστημάτων ελέγχου. Κατά αυτόν τον τρόπο εξασφαλίζεται η αποτελεσματική κατανομή και ο συντονισμός εργασιών, επιτρέποντας την ομαλή συνεργασία.

Τα AMR ενσωματώνουν διεπαφές ανθρώπου-ρομπότ, όπως οθόνες αφής, φωνητικές εντολές, για να διευκολύνουν την αποτελεσματική επικοινωνία. Η ενημέρωση για την κατάσταση των οχημάτων σε πραγματικό χρόνο, η κατανομή εργασιών και οι μηχανισμοί αξιολόγησης ενισχύουν τη συνεργασία μεταξύ των AMR και των ανθρώπων. Βασικό μέλημα είναι η ασφάλεια, για το λόγο αυτό τα AMR είναι εξοπλισμένα με συστήματα ανίχνευσης και αποφυγής συγκρούσεων για την πρόληψη ατυχημάτων και τη διατήρηση ασφαλών αποστάσεων από το ανθρώπινο προσωπικό.

Τα μέτρα ασφαλείας των AGV περιλαμβάνουν φυσικά όρια, ζώνες περιορισμένης πρόσβασης και προειδοποιητικά σήματα για την ελαχιστοποίηση του κινδύνου προσκρούσεων μεταξύ AGV και ανθρώπων. Οι αισθητήρες προσέγγισης και τα συστήματα όρασης επιτρέπουν στα AGVs να ανιχνεύουν εμπόδια και ανθρώπους, πραγματοποιώντας τις κατάλληλες ενέργειες για την αποφυγή ατυχημάτων. Συχνά απαιτείται ανθρώπινη επίβλεψη και παρέμβαση για την επιτήρηση των λειτουργιών ενός AGV. Οι χειριστές παρακολουθούν τις δραστηριότητες του AGV, αντιμετωπίζουν λειτουργικά ζητήματα και παρεμβαίνουν όταν είναι απαραίτητο. Εξασφαλίζουν την ορθή λειτουργία των AGV, εκτελούν εργασίες συντήρησης και χειρίζονται σφάλματα που ενδέχεται να προκύψουν κατά τη διάρκεια των εργασιών. Η ανθρώπινη επίβλεψη διασφαλίζει σε μεγαλύτερο βαθμό την ασφαλή λειτουργία των AGV σε συνεργασία με τους ανθρώπους.

Αρχική επένδυση:

Η αρχική επένδυση τοποθέτησης οχημάτων AMR σε μια αποθήκη είναι συνήθως ακριβότερη σε σύγκριση με τα AGV. Τα οχήματα AMR είναι εξοπλισμένα με προηγμένες τεχνολογίες, όπως αισθητήρες, υπολογιστές επί του οχήματος και συστήματα πλοήγησης, τα οποία συμβάλλουν στο υψηλότερο αρχικό τους κόστος. Επιπλέον, τα AMR συχνά απαιτούν παραμετροποίηση και ενσωμάτωση με τα υπάρχοντα συστήματα αυτοματισμού της αποθήκης, γεγονός που αυξάνει την αρχική επένδυση.

Τα AGV, από την άλλη πλευρά, έχουν γενικά χαμηλότερο αρχικό κόστος σε σύγκριση με τα AMR. Το κόστος των AGV ποικίλλει ανάλογα με τα χαρακτηριστικά τους όπως το μέγεθος, η χωρητικότητα ωφέλιμου φορτίου, η τεχνολογία πλοήγησης και οι επιπλέον δυνατότητες που διαθέτουν. Ενώ τα AGV απαιτούν επίσης παραμετροποίηση και ενσωμάτωση, το συνολικό κόστος είναι συχνά χαμηλότερο λόγω του απλούστερου σχεδιασμού τους.

Απαιτήσεις υποδομής:

Τα AMR δεν απαιτούν γενικά σημαντικές τροποποιήσεις υποδομής εντός της αποθήκης επειδή λειτουργούν αυτόνομα χρησιμοποιώντας ενσωματωμένους αισθητήρες, αλγόριθμους χαρτογράφησης και δυνατότητες αποφυγής εμποδίων. Αυτό εξαλείφει την ανάγκη για σταθερή υποδομή πλοήγησης, όπως μαγνητικές ταινίες ή αισθητήρες ενσωματωμένοι στο δάπεδο, με αποτέλεσμα την πιθανή μείωση του κόστους.

Αντίθετα, τα AGV συνήθως βασίζονται σε τροποποιήσεις της υποδομής αυτής για να λειτουργήσουν αποτελεσματικά. Οι τροποποιήσεις αυτές αυξάνουν το αρχικό κόστος σε σύγκριση με τα AMR.

Επεκτασιμότητα:

Το χαρακτηριστικό της επεκτασιμότητας είναι κοινό και για τους δύο τύπους των οχημάτων. Τα AMR προσφέρουν ευελιξία στην αύξηση των οχημάτων, καθώς επιπλέον ρομπότ μπορούν εύκολα να ενσωματωθούν στο υπάρχον σύστημα. Η επεκτασιμότητα αυτή επιτρέπει στις επιχειρήσεις να ξεκινήσουν με μικρότερο αριθμό ρομπότ και να αυξήσουν σταδιακά το στόλο τους, ανάλογα με τις ανάγκες, με μειωμένο κόστος αναβάθμισης της αποθήκης. Τα AGV προσφέρουν και αυτά τη δυνατότητα της επεκτασιμότητας. Η αναβάθμιση αυτή όμως σε σύγκριση με την αναβάθμιση που προσφέρουν τα AMR είναι μεν οικονομική αλλά περισσότερο περιορισμένη. Αυτό οφείλεται στην σταθερή διαμόρφωση της αποθήκης τα προκαθορισμένα μονοπάτια κίνησης.

Συμπερασματικά, μολονότι η αρχική επένδυση σε AMR σε μια αποθήκη είναι γενικά υψηλότερη από τα AGV, τα AMR προσφέρουν πλεονεκτήματα όπως η ευελιξία, η προσαρμοστικότητα και η επεκτασιμότητα χωρίς την ανάγκη εκτεταμένων τροποποιήσεων της υποδομής. Τα AGV έχουν μεν χαμηλότερο κόστος αλλά απαιτούν σταθερή υποδομή, συστήματα πλοήγησης εγκατεστημένα στο χώρο της αποθήκης και η επεκτασιμότητα τους έχει όριο.

ΚΕΦΑΛΑΙΟ 3 Πλοήγηση οχημάτων AMR σε δυναμικά περιβάλλοντα αποθήκης

3.1 Περιγραφή δυναμικού περιβάλλοντος αποθήκης

Με τον όρο δυναμικό περιβάλλον αποθήκης εννοούμε μια εγκατάσταση που υφίσταται συχνές αλλαγές στη διάταξη, τις εργασίες, τα αποθέματα ή άλλους συντελεστές που επηρεάζουν την καθημερινή της λειτουργία. Οι αλλαγές αυτές μπορεί να οφείλονται σε πλήθος παραγόντων όπως οι απαιτήσεις για αποθήκευση διαφορετικών προϊόντων ανάλογα με την εποχή, οι εξελισσόμενες απαιτήσεις των πελατών και η ανάγκη προσαρμογής σε νέα προϊόντα. Μια δυναμική αποθήκη διακρίνεται από την ικανότητά της να προσαρμόζεται γρήγορα και αποτελεσματικά σε αυτές τις αλλαγές, ώστε να διασφαλίζεται η βέλτιστη παραγωγικότητα και η ικανοποίηση των απαιτήσεων.

3.1.1 Παράγοντες που ορίζουν μια αποθήκη ως δυναμική

Μεταβαλλόμενη διάταξη:

Οι δυναμικές αποθήκες υφίστανται συχνά τροποποιήσεις στη διάταξη τους. Αυτό μπορεί να περιλαμβάνει την αναδιαμόρφωση των ραφιών αποθήκευσης, την προσαρμογή των συστημάτων μεταφοράς και την δημιουργία νέων περιοχών συλλογής με σκοπό την εξυπηρέτηση των μεταβαλλόμενων λειτουργικών απαιτήσεων.

Μεταβλητά αποθέματα:

Για τις αποθήκες που φιλοξενούν ένα ευρύ φάσμα προϊόντων ή προϊόντα των οποίων η ζήτηση δεν είναι σταθερή, υπάρχει η ανάγκη να προσαρμόζονται συνεχώς οι τεχνικές αποθήκευσης και ανάκτησης των αποθεμάτων τους. Αυτό απαιτεί αποτελεσματικά συστήματα μεταφοράς αποθεμάτων, ευέλικτα και με δυνατότητα προσαρμογής όσο αφορά τις διαφορετικές διαστάσεις των προϊόντων και τις ταχύτητες μετακίνησης τους.

Κυμαινόμενη ζήτηση:

Στις δυναμικές αποθήκες, η ζήτηση προϊόντων μπορεί να είναι εξαιρετικά ευμετάβλητη λόγω παραγόντων όπως η εποχικότητα, οι προσφορές ή οι τάσεις της αγοράς. Για το λόγο αυτό, πρέπει προσαρμόζονται κατάλληλα οι εργασίες και η κατανομή των πόρων της αποθήκης, έτσι ώστε να ανταποκρίνονται στα μεταβαλλόμενα μοτίβα ζήτησης. Παράδειγμα αυτής της προσαρμογής αποτελεί η αυξομείωση του εργατικού δυναμικού, του τεχνολογικού εξοπλισμού καθώς και των αποθηκευτικών δυνατοτήτων.

Εξελισσόμενη τεχνολογία:

Η εισαγωγή νέων τεχνολογιών, όπως οι πλατφόρμες ηλεκτρονικού εμπορίου, η ρομποτική ή τα συστήματα διαχείρισης αποθήκης, δημιουργούν την ανάγκη να μετασχηματιστούν γρήγορα οι λειτουργίες της αποθήκης. Στις δυναμικές αποθήκες πρέπει να ενσωματώνονται οι αναδυόμενες τεχνολογίες για να βελτιώνεται η παραγωγικότητά τους και να διατηρείται το ανταγωνιστικό τους πλεονέκτημα.

3.1.2 Προκλήσεις στην αυτοματοποίηση δυναμικών αποθηκών

Επεκτασιμότητα:

Οι μεταβαλλόμενες απαιτήσεις στις δυναμικές αποθήκες απαιτούν λύσεις αυτοματισμού που μπορούν εύκολα να ενσωματωθούν. Είναι ιδιαίτερα σημαντικό η ενσωμάτωση αυτή να πραγματοποιηθεί χωρίς να διαταραχθούν σημαντικά οι λειτουργίες της αποθήκης που ήδη εκτελούνται. Η δυνατότητα προσθήκης ή αφαίρεσης εξοπλισμού αυτοματισμού αν και δεν είναι εύκολη διαδικασία είναι αναγκαία για την ικανοποίηση των μεταβαλλόμενων απαιτήσεων.

Λήψη αποφάσεων σε πραγματικό χρόνο:

Οι δυναμικές αποθήκες απαιτούν συστήματα αυτοματισμού που μπορούν να λαμβάνουν αποφάσεις σε πραγματικό χρόνο με βάση τις συνθήκες που επικρατούν. Για να επιτευχθεί κάτι τέτοιο είναι απαραίτητη η χρήση ευφυών αλγορίθμων, προηγμένων αισθητήρων και δυνατοτήτων ανάλυσης δεδομένων για τη βελτιστοποίηση των εργασιών, την κατανομή των πόρων και την άμεση ανταπόκριση στις μεταβαλλόμενες προτεραιότητες.

Συνεργασία ρομποτικών οχημάτων με τον άνθρωπο:

Η συνεργασία μεταξύ ανθρώπων και ρομπότ σε δυναμικές αποθήκες αποτελεί ουσιώδη παράγοντα για την επίτευξη ομαλής και αποδοτικής επιχειρησιακής λειτουργίας. Τα ρομποτικά οχήματα προσφέρουν αυξημένη ταχύτητα, ακρίβεια και αντοχή στις εργασίες της αποθήκης. Ο ανθρώπινος παράγοντας συμβάλλει στο έλεγχο και τη λήψη συνθέτων αποφάσεων επίλυσης τυχών προβλημάτων. Ωστόσο προκειμένου να συνδυαστούν τα πλεονεκτήματα και των δύο είναι απαραίτητη η πιστή τήρηση κανόνων λειτουργίας, όπως ο διαχωρισμός των περιοχών εργασίας των ανθρώπων και λειτουργίας των ρομποτικών οχημάτων. Έτσι επηρεάζεται όσο τον δυνατόν λιγότερο η απόδοση των ρομπότ από ανθρώπινη δραστηριότητα και αυξάνεται η ασφάλεια του εργασιακού χώρου. Εντούτοις, η επιτυχής συνεργασία μεταξύ ανθρώπων και ρομπότ απαιτεί την αντιμετώπιση προκλήσεων όπως τα επικοινωνιακά χάσματα, η κατανομή εργασιών και η διαφοροποίηση των αρμοδιοτήτων.

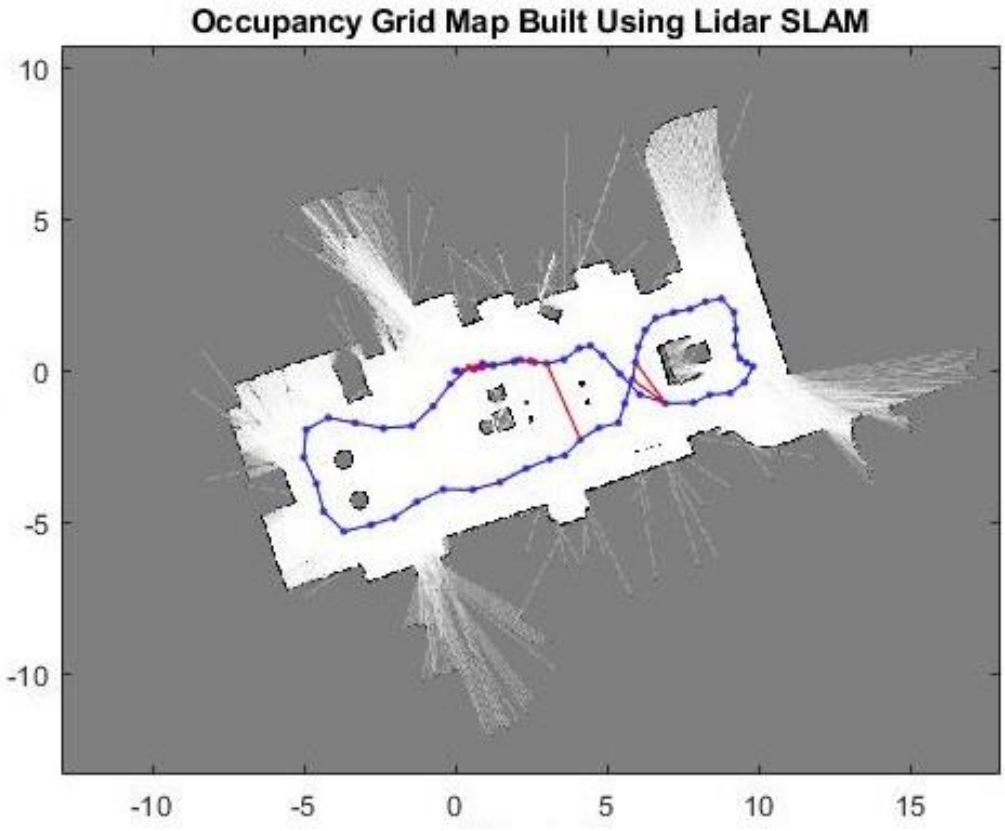
Η αντιμετώπιση αυτών των προκλήσεων απαιτεί μια ολοκληρωμένη προσέγγιση που συνδυάζει ευφυείς τεχνολογίες αυτοματισμού, στρατηγικό σχεδιασμό και συνεχή έλεγχο. Με την ικανοποίηση αυτών των απαιτήσεων, η αυτοματοποίηση αποθηκών μπορεί να συμβάλει στην αύξηση της παραγωγικότητας και της ακρίβειας.

3.2 Καταλληλότητα AMR για δυναμικές αποθήκες

Τα αυτόνομα ρομποτικά οχήματα χρησιμοποιούν μια ποικιλία αλγορίθμων και λογισμικών για να λειτουργούν αποτελεσματικά σε δυναμικά περιβάλλοντα. Η χρήση τους καθιστά εφικτή την αντιμετώπιση των προκλήσεων που χαρακτηρίζουν τις δυναμικές αποθήκες όπως η ανάγκη για αυτοεντοπισμό, η αντίληψη των μεταβολών στο περιβάλλον καθώς και η αποφυγή των εμποδίων αλλά και του ανθρωπίνου δυναμικού που εργάζεται και κινείται στον ίδιο χώρο με τα AMR.

Ταυτόχρονος εντοπισμός και χαρτογράφηση (SLAM):

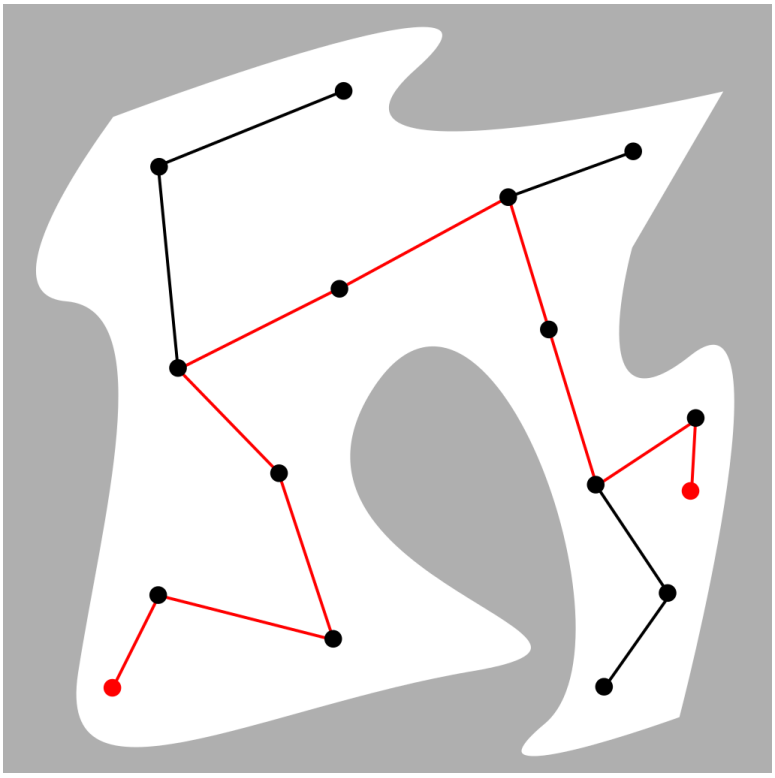
Ο ακριβής αυτοεντοπισμός μέσα σε μια δυναμική αποθήκη επιτυγχάνεται μέσω της εφαρμογής αλγορίθμων SLAM. Αυτοί οι αλγόριθμοι, οι οποίοι βασίζονται σε μοντέλα πιθανοτήτων, επιτρέπουν στα AMR να κατασκευάζουν ταυτόχρονα έναν χάρτη του περιβάλλοντός τους, ενώ παράλληλα εκτιμούν τη δική τους θέση μέσα σε αυτόν τον χάρτη. Οι αλγόριθμοι EKF-SLAM (Extended Kalman Filter SLAM) και FastSLAM είναι εναλλακτικές εκδοχές που χρησιμοποιούνται επίσης για τον αυτοεντοπισμό. Η συγχώνευση των δεδομένων των αισθητήρων και των πληροφοριών οδομετρίας έχει ως αποτέλεσμα την ενημέρωση του χάρτη σε πραγματικό χρόνο, εξασφαλίζοντας ακριβή εντοπισμό με δυνατότητα προσαρμογής, ακόμη και σε περιβάλλοντα που χαρακτηρίζονται από μεταβολές στη διάταξη τους.



Εικόνα 3.1 – Lidar SLAM Map Example (https://www.mathworks.com)

Σχεδιασμός μονοπατιού και προσαρμογή σε πραγματικό χρόνο.

Ο σχεδιασμός της πορείας σε δυναμικές αποθήκες απαιτεί συνεχή προσαρμογή. Τα AMR χρησιμοποιούν προηγμένους αλγορίθμους σχεδιασμού διαδρομών που επανασχεδιάζουν δυναμικά τις διαδρομές λαμβάνοντας υπόψη τις αλλαγές στο περιβάλλον σε πραγματικό χρόνο. Οι αλγόριθμοι LPA* (Lifelong Planning A-star), D* (Dynamic A-star) και Dynamic RRT (Dynamic Rapidly-exploring Random Trees) είναι παραδείγματα μεθόδων σχεδιασμού διαδρομών που χρησιμοποιούνται. Αυτοί οι αλγόριθμοι ενσωματώνουν παράγοντες όπως οι μεταβαλλόμενες θέσεις των εργαζομένων και οι διαστάσεις των προϊόντων, εξασφαλίζοντας την επιλογή της πιο αποτελεσματικής και χωρίς συγκρούσεις διαδρομής. Η δυνατότητα δυναμικής προσαρμογής των διαδρομών τους είναι καθοριστικής σημασίας για την ελαχιστοποίηση των δυσλειτουργιών, την ενίσχυση της επιχειρησιακής απόδοσης και της ασφάλειας.



Εικόνα 3.2 – Path Planning Example (https://en.wikipedia.org)

Ανίχνευση και παρακολούθηση εμποδίων:

Για την πλοήγηση ανάμεσα σε κινούμενους εργαζόμενους και απρόβλεπτες μεταφορές προϊόντων, τα AMR βασίζονται σε αλγορίθμους ανίχνευσης και εντοπισμού εμποδίων. Αυτοί οι αλγόριθμοι αξιοποιούν τα δεδομένα από τους αισθητήρες για τον εντοπισμό και την παρακολούθηση των κινήσεων των δυναμικών εμποδίων εντός του χώρου εργασίας. Συχνά χρησιμοποιούνται αλγόριθμοι εντοπισμού αντικειμένων, όπως το φίλτρο Kalman, το φίλτρο σωματιδίων ή νευρωνικά δίκτυα. Με τη συνεχή παρακολούθηση και πρόβλεψη της κίνησης των δυναμικών εμποδίων, τα AMR μπορούν να σχεδιάζουν τις διαδρομές τους ώστε να αποφεύγουν με ασφάλεια τις συγκρούσεις και να προσαρμόζονται στο μεταβαλλόμενο περιβάλλον.

ΚΕΦΑΛΑΙΟ 4 Related Work

Στο συγκεκριμένο κεφάλαιο περιγράφονται οι αλγόριθμοι σχεδιασμού μονοπατιού που αξιοποιήθηκαν από τον αλγόριθμο εύρεσης και επίλυσης πιθανών συγκρούσεων (Κεφάλαιο 6). Ειδικότερα παρουσιάζεται ο τρόπος λειτουργίας και ο ψευδοκώδικας του κάθε αλγορίθμου. Τέλος πραγματοποιείται μια συνοπτική αναφορά στην απόδοση των αλγορίθμων αναδεικνύοντας τόσο τα ισχυρά όσο και τα αδύναμα σημεία τους. Στην τελευταία ενότητα παρουσιάζονται επίσης τα παραγόμενα μονοπάτια και τα στατιστικά της εκτέλεσης των παραπάνω αλγορίθμων, για ένα μόνο όχημα, επιτρέποντας την άμεση σύγκριση τους.

4.1 Αλγόριθμος A* (A-star)

4.1.1 Περιγραφή αλγορίθμου και τρόπος λειτουργίας

Ο αλγόριθμος A* (A-star) [7] είναι μια ευρέως χρησιμοποιούμενη και αποτελεσματική μέθοδος στον τομέα της τεχνητής νοημοσύνης και της επιστήμης των υπολογιστών για την επίλυση προβλημάτων εύρεσης μονοπατιών και διάσχισης γράφων. Ο αλγόριθμος αναπτύχθηκε από τους Peter Hart, Nils Nilsson και Bertram Raphael το 1968 και συνδυάζει στοιχεία του αλγορίθμου του Dijkstra και ευρετικές μεθόδους αναζήτησης για την εύρεση της συντομότερης διαδρομής μεταξύ δύο κόμβων ή σημείων σε ένα γράφο.

Τρόπος λειτουργίας:

Στον πυρήνα του, ο αλγόριθμος A* πραγματοποιεί μια συστηματική εξερεύνηση ενός γράφου εξετάζοντας πιθανές διαδρομές από έναν αρχικό κόμβο προς έναν κόμβο-στόχο, ενώ παρακολουθεί το κόστος που σχετίζεται με κάθε διαδρομή. Διατηρεί δύο λίστες, την ανοιχτή λίστα (OPEN) και την κλειστή λίστα (CLOSED), για να παρακολουθεί τους κόμβους που έχουν επισκεφθεί και εκείνους που δεν έχουν εξερευνηθεί ακόμη. Η ανοικτή λίστα αποθηκεύει κόμβους που είναι υποψήφιοι για εξερεύνηση, ενώ η κλειστή λίστα περιέχει κόμβους που έχουν ήδη αξιολογηθεί.

4.1.2 Ψευδοκώδικας

Ο κόμβος-στόχος συμβολίζεται ως node_goal και ο κόμβος-αφετηρία με node_start. Διατηρούμε δύο λίστες τις OPEN και CLOSE. Η OPEN αποτελείται από κόμβους που έχουν επισκεφθεί αλλά δεν έχουν επεκταθεί (δηλαδή δεν έχουν εξερευνηθεί ακόμη οι διάδοχοι). Αυτή είναι η λίστα των εν αναμονή αναζητήσεων. Η λίστα CLOSE αποτελείται από κόμβους που έχουν επισκεφθεί και επεκταθεί (δηλαδή, οι κόμβοι διάδοχοι έχουν ήδη εξερευνηθεί και συμπεριληφθεί στη λίστα OPEN).

```
1  Put node_start in the OPEN list with f(node_start) = h(node_start) (initialization)
2  while the OPEN list is not empty {
3    Take from the open list the node node_current with the lowest
4         f(node_current) = g(node_current) + h(node_current)
5    if node_current is node_goal we have found the solution; break
6    Generate each state node_successor that come after node_current
7    for each node_successor of node_current {
8      Set successor_current_cost = g(node_current) + w(node_current, node_successor)
9      if node_successor is in the OPEN list {
10       if g(node_successor) ≤ successor_current_cost continue (to line 20)
11     } else if node_successor is in the CLOSED list {
12       if g(node_successor) ≤ successor_current_cost continue (to line 20)
13       Move node_successor from the CLOSED list to the OPEN list
14     } else {
15       Add node_successor to the OPEN list
16       Set h(node_successor) to be the heuristic distance to node_goal
17    }
18     Set g(node_successor) = successor_current_cost
19     Set the parent of node_successor to node_current
20   }
21   Add node_current to the CLOSED list
22 }
23 if(node_current != node_goal) exit with error (the OPEN list is empty)
```

Εικόνα 4.1 – A Pseudocode (https://mat.uab.cat)*

4.1.3 Αξιολόγηση κόμβων

Η αποτελεσματικότητα του αλγορίθμου A* πηγάζει από τη χρήση μιας ευρετικής συνάρτησης, η οποία παρέχει μια εκτίμηση του κόστους από έναν δεδομένο κόμβο έως τον κόμβο-στόχο. Η χρήση αυτής της ευρετικής συνάρτησης διαχωρίζει τον αλγόριθμο A* από άλλους αλγορίθμους εξερεύνησης κόμβων όπως ο αλγόριθμος Dijkstra. Ενώ ο αλγόριθμος Dijkstra εξερευνά όλες τις πιθανές διαδρομές από τον αρχικό κόμβο έως τον κόμβο-στόχο με βάση μόνο το πραγματικό κόστος επίτευξης κάθε κόμβου, ο A* αξιολογεί τους κόμβους με βάση έναν συνδυασμό του πραγματικού κόστους (γνωστό ως g-value) και του εκτιμώμενου κόστους για την επίτευξη του στόχου από τον εν λόγω κόμβο (γνωστό ως h-value). Το συνολικό κόστος μιας διαδρομής υπολογίζεται ως το άθροισμα της τιμής g και της τιμής h. Η ευρετική συνάρτηση καθοδηγεί τη διαδικασία αναζήτησης, επιτρέποντας στον αλγόριθμο να δίνει προτεραιότητα στα μονοπάτια που είναι πιο πιθανό να οδηγήσουν στο στόχο.

4.1.4 Απόδοση αλγορίθμου

Συνδυάζοντας έξυπνα τα πλεονεκτήματα τόσο του αλγορίθμου Dijkstra όσο και της ευρετικής συνάρτησης, ο αλγόριθμος A* είναι ικανός να βρίσκει αποτελεσματικά τη συντομότερη διαδρομή, ανάμεσα σε δυο επιλεγμένους κόμβους, σε γράφους με μεγάλο αριθμό κόμβων και ακμών. Αξίζει να σημειωθεί ότι ο αλγόριθμος A* δεν είναι δυναμικός, το οποίο σημαίνει ότι δεν διαθέτει την δυνατότητα να προσαρμόζει το μονοπάτι που έχει υπολογίσει αρχικά. Κατά την εκτέλεση του αλγορίθμου εύρεσης και επίλυσης πιθανών συγκρούσεων, θα αξιοποιήσουμε ένα τέχνασμα που επιτρέπει στον A* να λειτουργήσει δυναμικά, σε ένα περιβάλλον αποθήκης με πολλά ρομποτικά οχήματα, έτσι ώστε το τελικό μονοπάτι που προκύπτει να μην οδηγεί σε συγκρούσεις μεταξύ των οχημάτων.

4.2 Αλγόριθμος D* (Dynamic A-star)

4.2.1 Περιγραφή αλγορίθμου και τρόπος λειτουργίας

Ο αλγόριθμος D* (D-star) [11] είναι ένας ευρέως γνωστός αλγόριθμος σχεδιασμού μονοπατιού στον τομέα της ρομποτικής και της τεχνητής νοημοσύνης. Ο D* αναπτύχθηκε από τον A. Stenz το 1994 ως βελτιστοποίηση του αρχικού αλγορίθμου A* και παρέχει στα αυτόνομα ρομποτικά οχήματα τη δυνατότητα να προσαρμόζουν δυναμικά τις προγραμματισμένες διαδρομές τους σε πραγματικό χρόνο με βάση τα απρόβλεπτα εμπόδια.

Στην ουσία, ο αλγόριθμος D* λειτουργεί με επαναληπτική ενημέρωση των εκτιμήσεων κόστους για κάθε κόμβο, σε ένα γράφο που αναπαριστά το περιβάλλον. Αυτό επιτυγχάνεται μέσω μιας διαδικασίας σχεδιασμού διαδρομής από τον κόμβο-στόχο προς τα πίσω, δηλαδή προς την τρέχουσα θέση του ρομπότ και στη συνέχεια επαναληπτικής βελτίωσης της διαδρομής με βάση τις ενημερωμένες εκτιμήσεις κόστους.

Το βασικό χαρακτηριστικό που διαφοροποιεί τον D* από τους παραδοσιακούς αλγορίθμους σχεδιασμού διαδρομής είναι η ικανότητά του να χειρίζεται δυναμικά εμπόδια. Αυτό επιτυγχάνεται μέσω μιας διαδικασίας αναπροσαρμογής μονοπατιού, κατά την οποία ο αλγόριθμος διορθώνει συνεχώς το σχεδιασμένο μονοπάτι κάθε φορά που προκύπτουν νέα δεδομένα.

Τρόπος λειτουργίας:

Ο αλγόριθμος ξεκινά με την αρχικοποίηση των εκτιμήσεων κόστους-στόχου για κάθε κόμβο του γράφου. Συνήθως, οι εκτιμήσεις αυτές αρχικά λαμβάνουν υψηλές τιμές, υποδεικνύοντας ότι το ρομπότ δεν έχει βρει ακόμη εφικτή διαδρομή.

Όσο αφορά τον σχεδιασμό διαδρομής ο αλγόριθμος χρησιμοποιεί τις τρέχουσες εκτιμήσεις κόστους-στόχου για να σχεδιάσει μια διαδρομή από την τρέχουσα θέση του στον κόμβο-στόχο χρησιμοποιώντας μια ευρετική συνάρτηση όπως ο A*. Αυτή η αρχική διαδρομή χρησιμεύει ως βάση για τις επόμενες επαναλήψεις. Όταν ο αλγόριθμος λαμβάνει νέες πληροφορίες για εμπόδια, οι εκτιμήσεις κόστους προς τον κόμβο στόχο ενημερώνονται αναλόγως. Αυτό μπορεί να περιλαμβάνει τον επανυπολογισμό του κόστους διέλευσης από ορισμένους κόμβους ή την προσαρμογή των εκτιμήσεων με βάση τα εκάστοτε δεδομένα.

Κάθε φορά που το περιβάλλον αλλάζει ή λαμβάνονται νέες πληροφορίες, ο αλγόριθμος ξεκινά μια διαδικασία επανασχεδιασμού του μονοπατιού σε πραγματικό χρόνο. Αυτό περιλαμβάνει την επαναληπτική βελτίωση της διαδρομής με βάση τις ενημερωμένες εκτιμήσεις κόστους μέχρι να βρεθεί μια βέλτιστη ή σχεδόν βέλτιστη διαδρομή.

4.2.3 Ψευδοκώδικας

```
Function: PROCESS-STATE ()
L1   X = MIN-STATE( )
L2   if X = NULL then return –1
L3   k_old = GET-KMIN( ); DELETE(X)
L4   if k_old < h(X) then
L5     for each neighbor Y of X:
L6       if h(Y) ≤ k_old and h(X) > h(Y) + c(Y, X) then
L7         b(X) = Y; h(X) = h(Y) + c(Y, X)
L8   if k_old = h(X) then
L9     for each neighbor Y of X:
L10      if t(Y) = NEW or
L11        (b(Y) = X and h(Y) ≠ h(X) + c(X, Y)) or
L12        (b(Y) ≠ X and h(Y) > h(X) + c(X, Y)) then
L13        b(Y) = X; INSERT(Y, h(X) + c(X, Y))
L14  else
L15    for each neighbor Y of X:
L16      if t(Y) = NEW or
L17        (b(Y) = X and h(Y) ≠ h(X) + c(X, Y)) then
L18        b(Y) = X; INSERT(Y, h(X) + c(X, Y))
L19      else
L20        if b(Y) ≠ X and h(Y) > h(X) + c(X, Y) then
L21          INSERT(X, h(X))
L22        else
L23          if b(Y) ≠ X and h(X) > h(Y) + c(Y, X) and
L24            t(Y) = CLOSED and h(Y) > k_old then
L25            INSERT(Y, h(Y))
L26  return GET-KMIN( )
```

```
Function: MODIFY-COST (X, Y, cval)
L1   c(X, Y) = cval
L2   if t(X) = CLOSED then INSERT(X, h(X))
L3   return GET-KMIN( )
```

Εικόνα 4.2 – D Pseudocode [11]*

PROCESS-STATE: Αυτή η συνάρτηση υπολογίζει το βέλτιστο κόστος διαδρομής προς το στόχο. Αρχικά, όλες οι καταστάσεις ορίζονται ως " NEW". Η ευρετική συνάρτηση h(G) μηδενίζεται και η κατάσταση του κόμβου-στόχου G τοποθετείται στη λίστα OPEN. Η συνάρτηση PROCESS-STATE καλείται επανειλημμένα έως ότου είτε η τρέχουσα κατάσταση X του ρομπότ αφαιρεθεί από τη λίστα OPEN (υποδεικνύοντας την ολοκλήρωση του υπολογισμού της διαδρομής), είτε επιστραφεί η τιμή -1 που υποδεικνύει ότι η ακολουθία {X} δεν μπορεί να υπολογιστεί. Στη συνέχεια, αλγόριθμος ακολουθεί τα backpointers στην υπολογισμένη ακολουθία {X} μέχρι να φτάσει στο στόχο ή να ανιχνεύσει ένα σφάλμα στη συνάρτηση κόστους τόξου c(°) (όπως η συνάντηση ενός εμποδίου).

MODIFY-COST: Αυτή η συνάρτηση χρησιμοποιείται για να ενημερώνει τη συνάρτηση κόστους τόξου c(°) και κατά συνέπεια να ενημερωθούν οι επηρεαζόμενες καταστάσεις στη λίστα OPEN. Όταν ανιχνεύεται σφάλμα στη συνάρτηση κόστους τόξου στην κατάσταση Y, καλείται η συνάρτηση MODIFY-COST για να διορθώσει τη συνάρτηση κόστους και να ενημερώσει τη λίστα με τις επηρεαζόμενες καταστάσεις. Με τον τρόπο αυτό διασφαλίζεται ότι ο αλγόριθμος προσαρμόζεται στις αλλαγές των δεδομένων.

Ο αλγόριθμος επίσης χρησιμοποιεί διάφορες ενσωματωμένες μεθόδους, όπως η MIN-STATE (για την εύρεση της κατάστασης με την ελάχιστη τιμή k(°) στη λίστα OPEN), η GET-KMIN (για την ανάκτηση της ελάχιστης τιμής k από τη λίστα OPEN), η DELETE(X) (για την αφαίρεση της κατάστασης X από τη λίστα OPEN) και η INSERT(X, h) (για τον υπολογισμό και την εισαγωγή της κατάστασης X στη λίστα OPEN ταξινομημένη με βάση την τιμή k(°)).

4.2.4 Απόδοση αλγορίθμου

Ένα από τα βασικά πλεονεκτήματα του αλγορίθμου D* είναι η ικανότητά του να χειρίζεται αποτελεσματικά δυναμικά περιβάλλοντα και άγνωστο έδαφος, προσεγγίζοντας το βέλτιστο μονοπάτι και σε πολλές περιπτώσεις επιτυγχάνει να βρει το βέλτιστο μονοπάτι. Με τη συνεχή ενημέρωση της προγραμματισμένης διαδρομής βάσει πληροφοριών σε πραγματικό χρόνο, ο D* επιτρέπει στα ρομποτικά οχήματα να περιηγούνται σε πολύπλοκα περιβάλλοντα με μεγαλύτερη ευελιξία και αξιοπιστία σε σύγκριση με τους παραδοσιακούς αλγορίθμους σχεδιασμού διαδρομής που δεν προσφέρουν αυτή την δυνατότητα.

4.3 Αλγόριθμος D* Lite (Dynamic A-star Lite)

4.3.1 Περιγραφή αλγορίθμου και τρόπος λειτουργίας

Ο αλγόριθμος D* Lite [12] είναι μια βελτιστοποίηση του αλγορίθμου D* (Dynamic A*), που έχει σχεδιαστεί για να παρέχει αποδοτικό σχεδιασμό μονοπατιών σε δυναμικά περιβάλλοντα, ελαχιστοποιώντας παράλληλα το υπολογιστικό κόστος. Αναπτυγμένος από τους Sven Koenig και Maxim Likhachev το 2002, ο D* Lite προσφέρει μια βελτιωμένη προσέγγιση για δυναμικό επανασχεδιασμό, καθιστώντας τον κατάλληλο για εφαρμογές πραγματικού χρόνου, όπως η αυτόνομη πλοήγηση.

Ο D* Lite λειτουργεί με επαναληπτική ενημέρωση των εκτιμήσεων κόστους-στόχου για κάθε κόμβο σε μια αναπαράσταση του περιβάλλοντος σε γράφο. Διατηρεί μια ουρά προτεραιότητας των κόμβων προς επέκταση, με προτεραιότητες που βασίζονται στις τρέχουσες εκτιμήσεις κόστους.

Τρόπος λειτουργίας:

Ο αλγόριθμος D* Lite ξεκινά με την αναπαράσταση του περιβάλλοντος σε γράφο και δημιουργεί μια ουρά προτεραιότητας με αρχικό κόμβο την κατάσταση εκκίνησης. Στη συνέχεια ενημερώνει τις εκτιμήσεις κόστους-στόχου με τη διάδοση των αλλαγών από την κατάσταση στόχου προς την αρχική κατάσταση, επεκτείνοντας τους κόμβους με τη σειρά των προτεραιοτήτων τους στην ουρά. Μετά από κάθε βήμα επέκτασης, ο D* Lite ενημερώνει τις προτεραιότητες των κόμβων με βάση τις πρόσφατα υπολογισμένες εκτιμήσεις κόστους, καθοδηγούμενος από ευρετικές συναρτήσεις για την ιεράρχηση των πιο υποσχόμενων μονοπατιών. Όταν συμβαίνουν σημαντικές αλλαγές στο περιβάλλον, ο αλγόριθμος ενημερώνει μόνο τους επηρεαζόμενους κόμβους, ελαχιστοποιώντας την υπολογιστική επιβάρυνση κατά τη βελτίωση του μονοπατιού.

Ο D* Lite χρησιμοποιεί απλοποιημένες ουρές προτεραιότητας, σταδιακές ενημερώσεις του γράφου και ευρετικές εκτιμήσεις για τη βελτίωση της αποτελεσματικότητας και τη μείωση της επιβάρυνσης υπολογισμού κατά τη διάδοση και τον επανασχεδιασμό.

4.3.2 Ψευδοκώδικας

```
procedure CalculateKey(s)
{01"} return [min(g(s), rhs(s)) + h(s_start, s) + k_m; min(g(s), rhs(s))];
procedure Initialize()
{02"} U = ∅;
{03"} k_m = 0;
{04"} for all s ∈ S rhs(s) = g(s) = ∞;
{05"} rhs(s_goal) = 0;
{06"} U.Insert(s_goal, [h(s_start, s_goal); 0]);
procedure UpdateVertex(u)
{07"} if (g(u) ≠ rhs(u) AND u ∈ U) U.Update(u, CalculateKey(u));
{08"} else if (g(u) ≠ rhs(u) AND u ∉ U) U.Insert(u, CalculateKey(u));
{09"} else if (g(u) = rhs(u) AND u ∈ U) U.Remove(u);
procedure ComputeShortestPath()
{10"} while (U.TopKey() < CalculateKey(s_start) OR rhs(s_start) > g(s_start))
{11"}   u = U.Top();
{12"}   k_old = U.TopKey();
{13"}   k_new = CalculateKey(u));
{14"}   if(k_old < k_new)
{15"}     U.Update(u, k_new);
{16"}   else if (g(u) > rhs(u))
{17"}     g(u) = rhs(u);
{18"}     U.Remove(u);
{19"}     for all s ∈ Pred(u)
{20"}       if (s ≠ s_goal) rhs(s) = min(rhs(s), c(s, u) + g(u));
{21"}       UpdateVertex(s);
{22"}   else
{23"}     g_old = g(u);
{24"}     g(u) = ∞;
{25"}     for all s ∈ Pred(u) ∪ {u}
{26"}       if (rhs(s) = c(s, u) + g_old)
{27"}         if (s ≠ s_goal) rhs(s) = min_{s'∈Succ(s)} (c(s, s') + g(s'));
{28"}       UpdateVertex(s);
```

Εικόνα 4.2.1 – D Lite Pseudocode [12]*

```
procedure Main()
{29"} s_last = s_start;
{30"} Initialize();
{31"} ComputeShortestPath();
{32"} while (s_start ≠ s_goal)
{33"}   /* if (rhs(s_start) = ∞) then there is no known path */
{34"}   s_start = arg min_{s'∈Succ(s_start)} (c(s_start, s') + g(s'));
{35"}   Move to s_start;
{36"}   Scan graph for changed edge costs;
{37"}   if any edge costs changed
{38"}     k_m = k_m + h(s_last, s_start);
{39"}     s_last = s_start;
{40"}     for all directed edges (u, v) with changed edge costs
{41"}       c_old = c(u, v);
{42"}       Update the edge cost c(u, v);
{43"}       if (c_old > c(u, v))
{44"}         if (u ≠ s_goal) rhs(u) = min(rhs(u), c(u, v) + g(v));
{45"}       else if (rhs(u) = c_old + g(v))
{46"}         if (u ≠ s_goal) rhs(u) = min_{s'∈Succ(u)} (c(u, s') + g(s'));
{47"}       UpdateVertex(u);
{48"}     ComputeShortestPath();
```

Εικόνα 4.2.2 – D Lite Pseudocode [12]*

4.3.3 Σύγκριση με τον αλγόριθμο D*

Η κύρια διαφορά μεταξύ του D* και του D* Lite έγκειται στην προσέγγισή τους για τον δυναμικό επανασχεδιασμό και την υπολογιστική πολυπλοκότητα. Ενώ και οι δύο αλγόριθμοι μοιράζονται το στόχο της δημιουργίας αποτελεσματικού σχεδιασμού μονοπατιών σε δυναμικά περιβάλλοντα, ο D* Lite απλοποιεί τη διαδικασία αυτή χρησιμοποιώντας βελτιστοποιήσεις για τη μείωση του υπολογιστικού κόστους. Συγκεκριμένα, ο D* Lite χρησιμοποιεί μια απλοποιημένη δομή ουράς προτεραιότητας και ευρετικές εκτιμήσεις για την επιτάχυνση της διαδικασίας επανασχεδιασμού, εστιάζοντας στην ενημέρωση μόνο του υποσυνόλου του μονοπατιού που επηρεάζεται από τις αλλαγές στο περιβάλλον. Αντιθέτως, ο D* προσφέρει μια πιο ολοκληρωμένη προσέγγιση για τον δυναμικό επανασχεδιασμό, επιτρέποντας ακριβείς προσαρμογές της σχεδιασμένης διαδρομής, αλλά απαιτώντας δυνητικά ισχυρότερους υπολογιστικούς πόρους.

4.3.4 Απόδοση

Όσο αφορά την απόδοση, ο D* Lite στις περισσότερες περιπτώσεις παρουσιάζει ταχύτερους χρόνους επανασχεδιασμού και χαμηλότερο υπολογιστικό κόστος σε σύγκριση με τον D*, καθιστώντας τον πιο κατάλληλο για εφαρμογές πραγματικού χρόνου όπου η ταχεία προσαρμογή στις μεταβαλλόμενες συνθήκες είναι απαραίτητη. Ωστόσο, αυτή η αποδοτικότητα αποβαίνει εις βάρος κάποιας ακρίβειας στο σχεδιασμό διαδρομής, καθώς ο D* Lite μπορεί να θυσιάσει τη ακρίβεια για χάρη της ταχύτητας σε ορισμένα σενάρια. Η επιλογή μεταξύ του D* και του D* Lite εξαρτάται κάθε φορά από τις συγκεκριμένες απαιτήσεις, εξισορροπώντας την ανάγκη για ακρίβεια με τους διαθέσιμους υπολογιστικούς πόρους και την απόδοση σε πραγματικό χρόνο.

4.4 Αλγόριθμος RRT (Rapidly-Exploring-Random trees)

4.4.1 Περιγραφή αλγορίθμου και τρόπος λειτουργίας

Ο αλγόριθμος Rapidly-exploring Random Tree (RRT) [13] είναι μια διαδεδομένη μέθοδος, βασισμένη στις πιθανότητες, που χρησιμοποιείται στη ρομποτική και τον σχεδιασμό διαδρομών για την αποτελεσματική εξερεύνηση σύνθετων περιβαλλόντων. Ο RRT αναπτύχθηκε από τον Steven M. LaValle το 1998 και είναι ιδιαίτερα κατάλληλος για προβλήματα όπου ο χώρος είναι πολύπλοκος και υπάρχουν σύνθετα εμπόδια. Η βασική ιδέα πίσω από τον αλγόριθμο RRT είναι η σταδιακή ανάπτυξη ενός δέντρου με ρίζα μια αρχική κατάσταση προς μια επιθυμητή κατάσταση-στόχο, χρησιμοποιώντας τυχαία δειγματοληψία για την εξερεύνηση του χώρου. Η διαδικασία αυτή περιλαμβάνει δύο βασικά βήματα την επέκταση των κόμβων και την σύνδεση τους, όταν αυτό είναι επιτρεπτό.

Σε κάθε επανάληψη, δημιουργείται τυχαία, στο χώρο που ερευνάται ένας νέος κόμβος. Αυτός ο κόμβος προστίθεται στη συνέχεια στην υπάρχουσα δενδρική δομή συνδέοντάς τον με τον πλησιέστερο υπάρχοντα κόμβο του δέντρου. Η σύνδεση γίνεται κατά μήκος μιας εφικτής διαδρομής που αποφεύγει συγκρούσεις με εμπόδια του περιβάλλοντος. Αφού προστεθεί ο νέος κόμβος στο δέντρο, χρησιμοποιείται μια στρατηγική τοπικού σχεδιασμού για την προσπάθεια σύνδεσης του νεοπροστιθέμενου κόμβου με τους γειτονικούς του κόμβους στο δέντρο. Εάν βρεθεί ένα έγκυρο μονοπάτι, αυτό ενσωματώνεται στο δέντρο. Αυτό το βήμα σύνδεσης συμβάλλει στην αποτελεσματικότερη εξερεύνηση του χώρου. Ο αλγόριθμος RRT συνεχίζει την επανάληψη μεταξύ των βημάτων επέκτασης και σύνδεσης έως ότου ικανοποιηθεί μια καθορισμένη συνθήκη τερματισμού. Η συνθήκη αυτή συχνά ορίζεται είτε ως η επίτευξη ενός προκαθορισμένου αριθμού επαναλήψεων είτε όταν ο στόχος είναι αρκετά κοντά σε έναν από τους κόμβους του δέντρου.

4.4.2 Απόδοση αλγορίθμου

Τα βασικά χαρακτηριστικά και πλεονεκτήματα του αλγορίθμου RRT περιλαμβάνουν την πιθανολογική πληρότητα, την αποτελεσματικότητα στην εξερεύνηση πολύπλοκων χώρων και την προσαρμοστικότητα σε διαφορετικά προβλήματα.

Παρά τα πλεονεκτήματά του, ο RRT παρουσιάζει αδυναμίες, όπως η δυσκολία εύρεσης βέλτιστων λύσεων και η πιθανότητα μη αποδοτικής εξερεύνησης σε ορισμένα σενάρια. Παρ' όλα αυτά, παραμένει ένας θεμελιώδης αλγόριθμος στον τομέα της ρομποτικής και του σχεδιασμού μονοπατιών, με πολυάριθμες επεκτάσεις και παραλλαγές που έχουν αναπτυχθεί για την αντιμετώπιση συγκεκριμένων προκλήσεων. Ένα παράδειγμα παραλλαγής του RRT είναι ο αλγόριθμος Dynamic RRT οποίος είναι εμπνευσμένος από την οικογένεια των ντετερμινιστικών αλγορίθμων επανασχεδιασμού μονοπατιού star. Η παραλλαγή αυτή συνδυάζει τις στοχαστικές ιδιότητες του RRT και τη δυνατότητα για δυναμική αναπροσαρμογή του σχεδιασμένου δέντρου.

4.5 Αλγόριθμος DRRT (Dynamic Rapidly-Exploring-Random trees)

4.5.1 Περιγραφή αλγορίθμου και τρόπος λειτουργίας

Ο αλγόριθμος Dynamic Rapidly-exploring Random Trees (DRRT) [15] , ο οποίος προτάθηκε από τους D.Ferguson, N.Kalra και A.Stentz το 2006, είναι μια μέθοδος για προσαρμοστικό επανασχεδιασμό μονοπατιού σε δυναμικά περιβάλλοντα, προσφέροντας την ικανότητα πλοήγησης αυτόνομων ρομποτικών οχημάτων. Ενσωματώνει έννοιες τόσο από ντετερμινιστικούς αλγορίθμους αναπροσαρμογής, όπως ο D*, όσο και από στοχαστικούς αλγορίθμους όπως ο RRT. Ο DRRT αλλάζει δυναμικά τη δομή του δέντρου ώστε να προσαρμόζεται στις αλλαγές του περιβάλλοντος, επισημαίνοντας και αφαιρώντας τα άκυρα τμήματα του δέντρου και στη συνέχεια αναπτύσσοντάς το, εκ νέου, προς τον κόμβο-στόχο. Αυτή η διαδικασία εξασφαλίζει ότι το δέντρο που προκύπτει παραμένει έγκυρο, ενώ προσαρμόζεται αποτελεσματικά στις περιβαλλοντικές αλλαγές.

Τρόπος λειτουργίας:

Ο αλγόριθμος ξεκινά με ένα αρχικό δέντρο που παράγεται από τον RRT ορίζοντας ένα σημείο έναρξης και ένα σημείο τερματισμού. Καθώς το περιβάλλον αλλάζει (π.χ. κίνηση των ρομποτικών οχημάτων εντός της αποθήκης), ο αλγόριθμος ανιχνεύει αυτές τις αλλαγές και εντοπίζει και επισημαίνει τμήματα του υπάρχοντος δέντρου που καθίστανται άκυρα λόγω των αλλαγών. Αυτό συνήθως περιλαμβάνει τον εντοπισμό κόμβων και ακμών που τέμνονται με νεοεισαχθέντα εμπόδια. Στη συνέχεια, ο αλγόριθμος «κλαδεύει» το δέντρο αφαιρώντας όλα τα άκυρα μέρη.

Αυτό το βήμα εξασφαλίζει ότι μόνο τα έγκυρα τμήματα του δέντρου παραμένουν άθικτα. Μετά το «κλάδεμα», ο αλγόριθμος ανανεώνει το δέντρο για να αποκαταστήσει τη διαδρομή προς τον τελικό κόμβο. Η διαδικασία ανανέωσης περιλαμβάνει την επέκταση του δέντρου από τους εναπομείναντες έγκυρους κόμβους έως ότου ο τελικός κόμβος είναι και πάλι προσβάσιμος.

Για τη διευκόλυνση του αποτελεσματικού επανασχεδιασμού, ιδίως σε σενάρια που περιλαμβάνουν αυτόνομα ρομποτικά οχήματα, ο αλγόριθμος τροποποιεί την κατεύθυνση ανάπτυξης του δέντρου. Αντί να αναπτύσσεται από τον αρχικό κόμβο προς τον στόχο, αναπτύσσεται από τον στόχο προς την τρέχουσα θέση του οχήματος. Αυτή η αντιστροφή της κατεύθυνσης ανάπτυξης επιτρέπει την επαναχρησιμοποίηση των έγκυρων τμημάτων του υπάρχοντος δέντρου, όταν αλλάζει η θέση του ρομπότ ή εμφανίζονται νέα εμπόδια, ελαχιστοποιώντας την ανάγκη για πλήρη αναπαραγωγή του δέντρου.

4.5.2 Ψευδοκώδικας

```
Main()
1   q_start = q_goal; q_goal = q_robot;
2   InitRRT();
3   GrowRRT();
4   while (q_goal ≠ q_start)
5     q_goal = Parent(q_goal);
6     Move to q_goal and check for new obstacles;
7     if any new obstacles are observed
8       for each new obstacle o
9         InvalidateNodes(o);
10      if solution path contains an invalid node
11        RegrowRRT();
```

Εικόνα 4.5.1 – DRRT Pseudocode [15]

```
RegrowRRT()
1   TrimRRT();
2   GrowRRT();

TrimRRT()
3   S = ∅; i = 1;
4   while (i < T.size())
5     q_i = T.node(i); q_p = Parent(q_i);
6     if (q_p.flag = INVALID)
7       q_i.flag = INVALID;
8     if (q_i.flag ≠ INVALID)
9       S = S ∪ {q_i};
10    i = i + 1;
11  T = CreateTreeFromNodes(S);

InvalidateNodes(obstacle)
12  E = FindAffectedEdges(obstacle);
13  for each edge e ∈ E
14    q_e = ChildEndpointNode(e);
15    q_e.flag = INVALID;
```

Εικόνα 4.5.2 – DRRT Pseudocode [15]

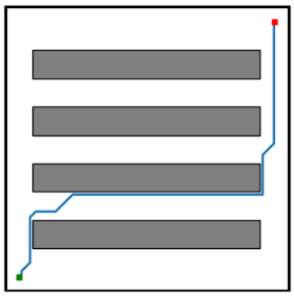
4.5.3 Απόδοση

Ανάλογα με τη φύση των αλλαγών στο περιβάλλον, ο αλγόριθμος μπορεί να προσαρμόσει τη στρατηγική ανάπτυξης του δέντρου εξερεύνησης ώστε να επικεντρωθεί σε περιοχές που επηρεάζονται από τις αλλαγές αυτές. Αυτή η προσαρμοστική προσέγγιση ενισχύει την αποτελεσματικότητα δίνοντας προτεραιότητα στην εξερεύνηση στις προαναφερόμενες περιοχές. Ο αλγόριθμος DRRT μπορεί να χρησιμοποιηθεί ως στρατηγική επανασχεδιασμού μονοπατιού σε πραγματικό χρόνο. Παρακολουθεί συνεχώς το περιβάλλον, προσαρμόζει δυναμικά τη δομή του δέντρου και τη στρατηγική ανάπτυξης του για να διασφαλίσει αποτελεσματική πλοήγηση σε άγνωστα ή δυναμικά περιβάλλοντα.

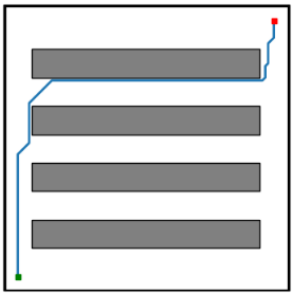
4.6 Σύγκριση αλγορίθμων για ένα AMR

4.6.1 Σχεδιασμένα μονοπάτια

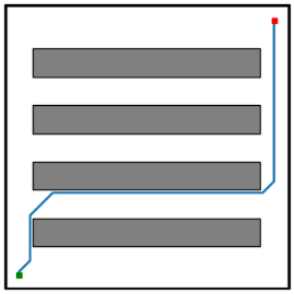
Warehouse 1:



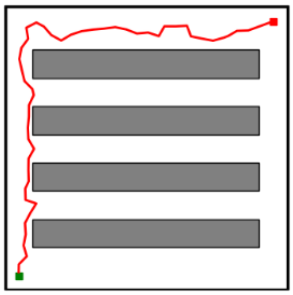
Εικόνα 4.6.1 A* Path



Εικόνα 4.6.2 D* Path

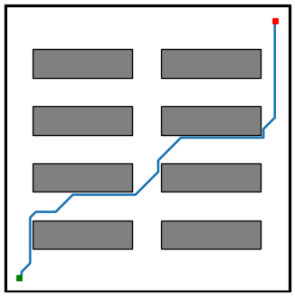


Εικόνα 4.6.3 D* Lite Path

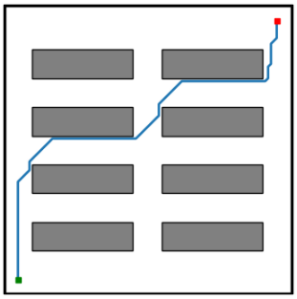


Εικόνα 4.6.4 DRRT Path

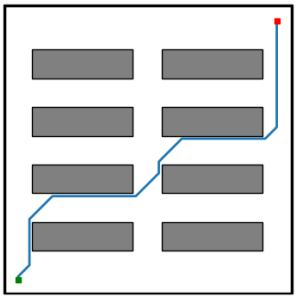
Warehouse 2:



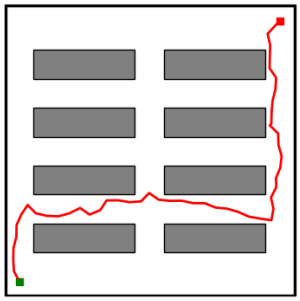
Εικόνα 4.6.5 A* Path



Εικόνα 4.6.6 D* Path

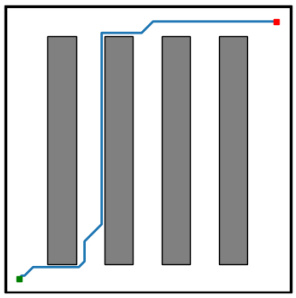


Εικόνα 4.6.7 D* Lite Path

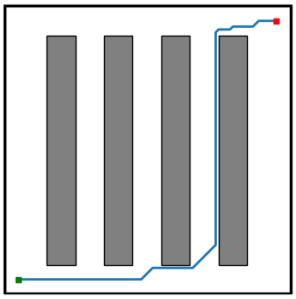


Εικόνα 4.6.8 DRRT Path

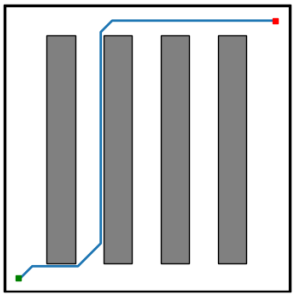
Warehouse 3:



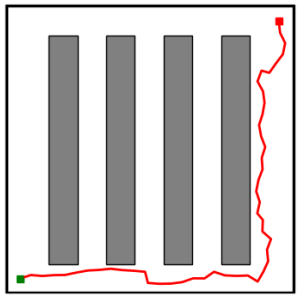
Εικόνα 4.6.9 A* Path



Εικόνα 4.6.10 D* Path

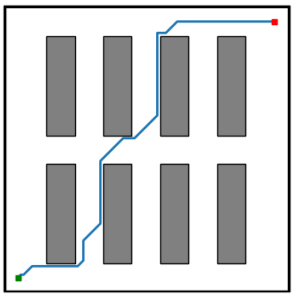


Εικόνα 4.6.11 D* Lite Path

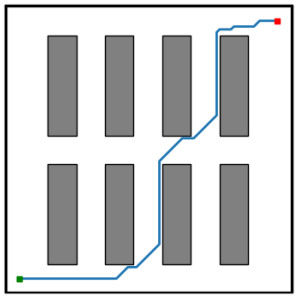


Εικόνα 4.6.12 DRRT Path

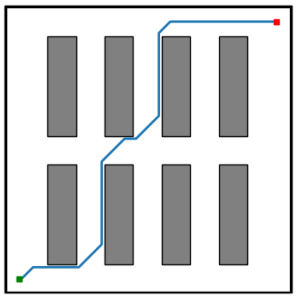
Warehouse 4:



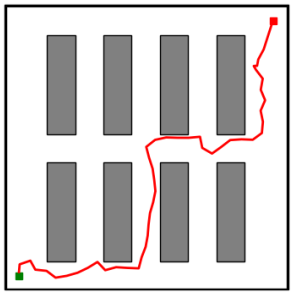
Εικόνα 4.6.13 A* Path



Εικόνα 4.6.14 D* Path

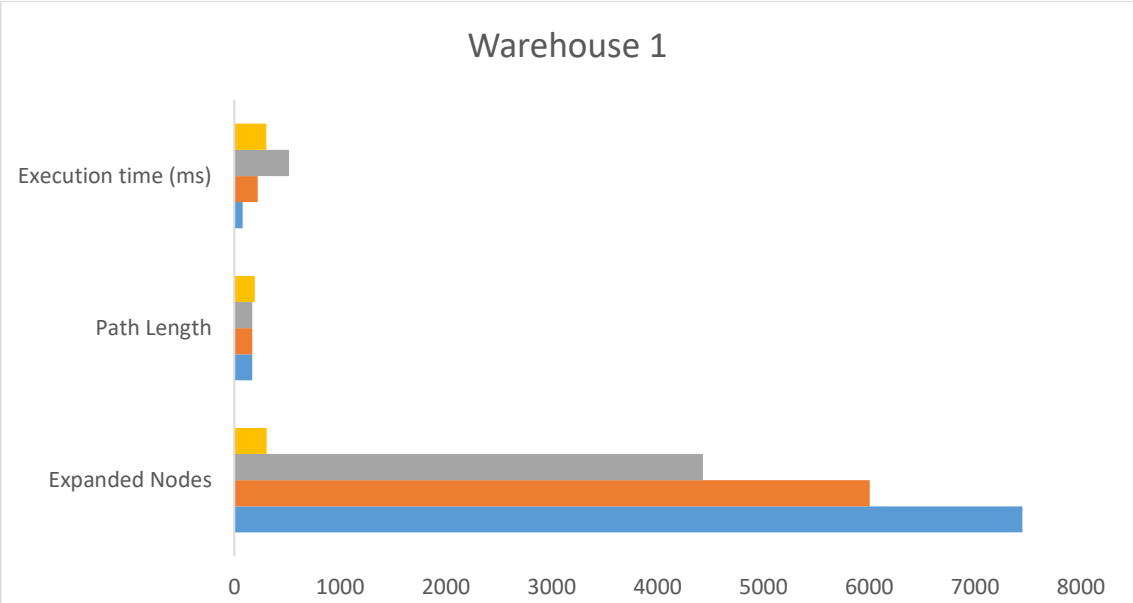


Εικόνα 4.6.15 D* Lite Path



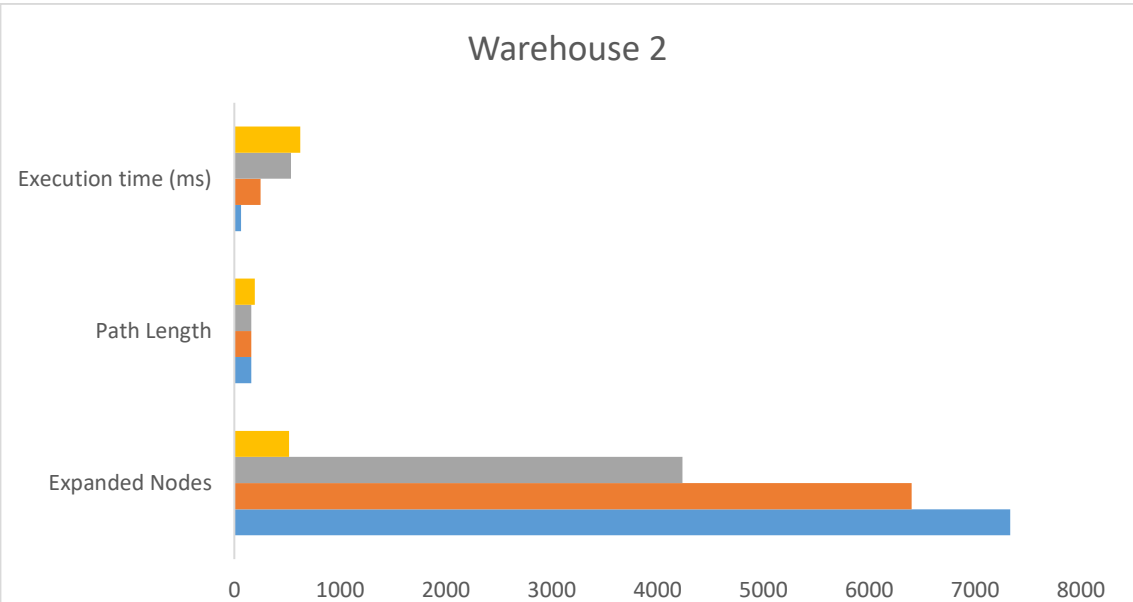
Εικόνα 4.6.16 DRRT Path

4.6.2 Διαγράμματα και στατιστικά



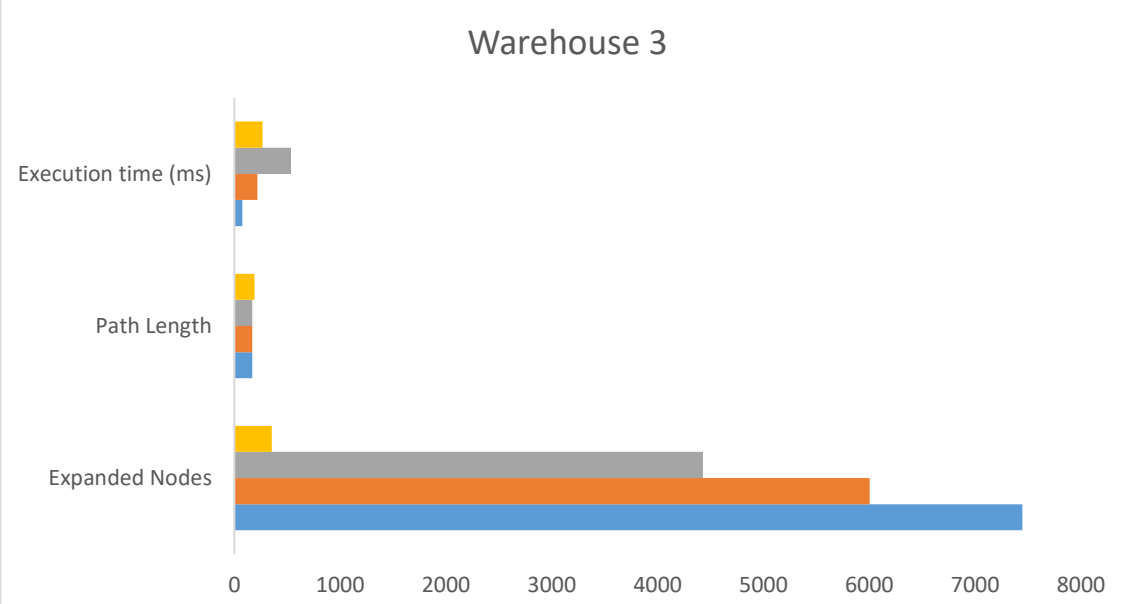
	Expanded Nodes	Path Length	Execution time (ms)
DRRT	305	194,552	301,41
D* Lite	4429	170,627	516,71
D*	6004	170,627	222,22
A*	7448	170,627	78,11

Εικόνα 4.6.17 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 1



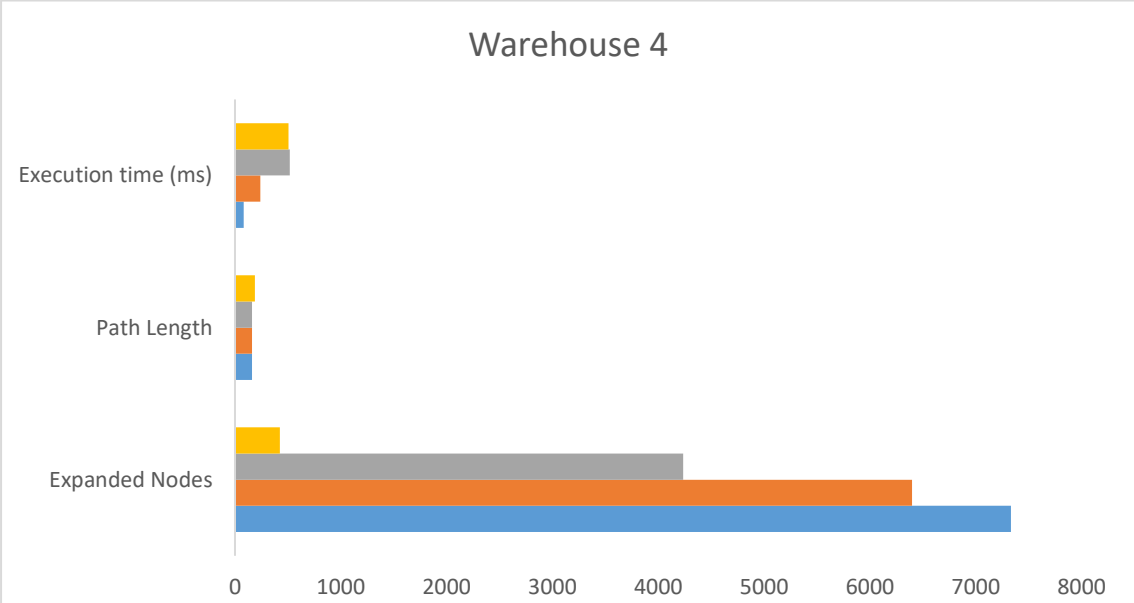
	Expanded Nodes	Path Length	Execution time (ms)
DRRT	516	193,907	624,46
D* Lite	4236	161,255	536,06
D*	6400	161,255	246,86
A*	7334	161,255	62,48

Εικόνα 4.6.18 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 2



	Expanded Nodes	Path Length	Execution time (ms)
DRRT	354	190,049	265,09
D* Lite	4429	170,627	534,77
D*	6005	170,627	216,44
A*	7448	170,627	75,66

Εικόνα 4.6.19 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 3



	Expanded Nodes	Path Length	Execution time (ms)
DRRT	424	187,298	505,18
D* Lite	4236	161,255	516,48
D*	6400	161,255	238,38
A*	7334	161,255	83,05

Εικόνα 4.6.20 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 4

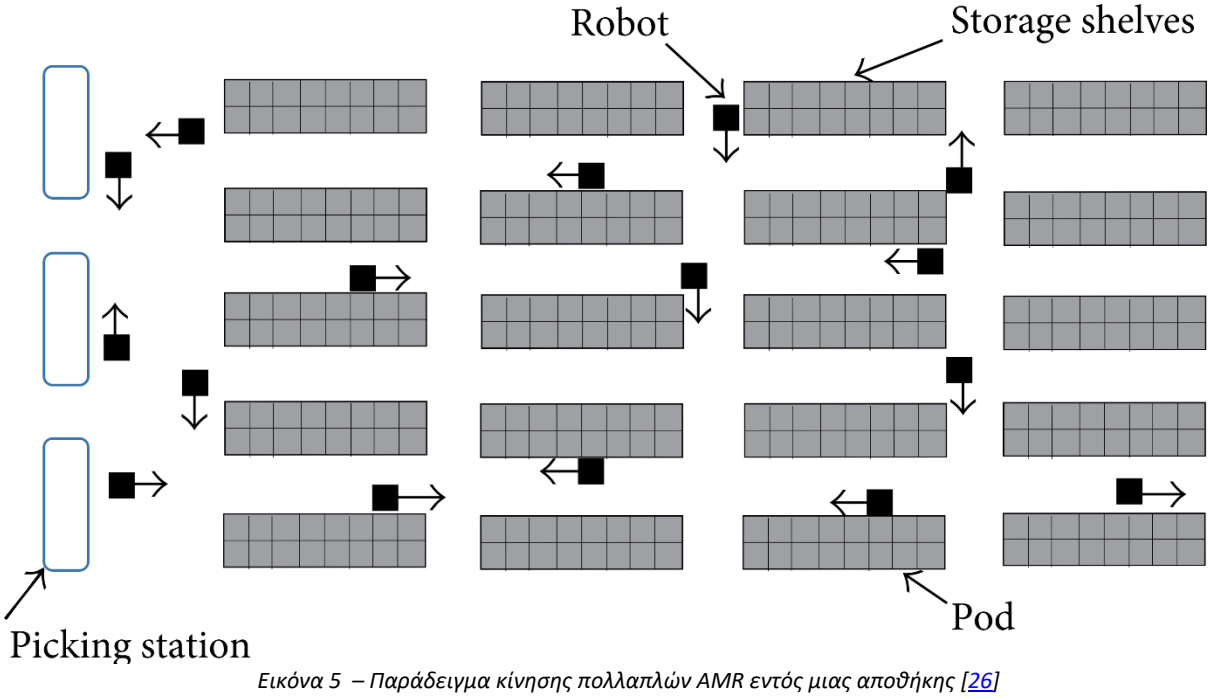
ΚΕΦΑΛΑΙΟ 5 Περιγραφή του προβλήματος

Το πρόβλημα που καλείται να αντιμετωπίσει η συγκεκριμένη εργασία είναι ο συντονισμός ενός στόλου που αποτελείται από ένα πλήθος αυτόνομων ρομποτικών οχημάτων τα οποία λειτουργούν μέσα σε έναν οριοθετημένο δισδιάστατο χώρο εργασίας, ο οποίος περιλαμβάνει εμπόδια. Πιο συγκεκριμένα προσπαθούμε να εντοπίσουμε και να αποτρέψουμε τις πιθανές συγκρούσεις, για να δημιουργήσουμε ασφαλή μονοπάτια, εξαλείφοντας την πιθανότητα αποτυχίας επίτευξης του στόχου για κάποιο όχημα.

Στην προσπάθεια μας να λύσουμε το εν λόγο πρόβλημα, εστιάζουμε στο σχεδιασμό ενός αλγορίθμου πρόβλεψης πιθανών συγκρούσεων. Ο αλγόριθμος αυτός εξοπλίζει τα οχήματα με τη δυνατότητα αυτόνομης μετακίνησης έτσι ώστε να μπορούν να εκτελέσουν με επιτυχία τις εργασίες που τους έχουν ανατεθεί. Ο χώρος πλοήγησης κάθε οχήματος περιλαμβάνει όλες τις πιθανές τιμές (x, y), εντός του σταθερού συστήματος συντεταγμένων, με εξαίρεση εκείνες οι οποίες είναι κατειλημμένες από εμπόδια.

Σε κάθε όχημα ανατίθεται μία εργασία. Η εκτέλεση της εργασίας αυτής περιλαμβάνει τη μετάβαση του οχήματος από τον κόμβο εκκίνησης στον κόμβο στόχο. Η μετάβαση πραγματοποιείται σε δύο διαδοχικές φάσεις: (i) τον προσδιορισμό μιας αρχικής διαδρομής χωρίς εμπόδια, προς τη θέση του στόχου και (ii) την συντονισμένη κίνηση του οχήματος με άλλα οχήματα κατά μήκος αυτής της διαδρομής εξασφαλίζοντας κίνηση χωρίς συγκρούσεις και χωρίς αδιέξοδα.

Θεωρούμε ότι κάθε όχημα είναι εξοπλισμένο με χάρτη του χώρου πλοήγησης και σύστημα εντοπισμού που παρέχει ακριβή την τοποθεσία και την διαδρομή όλων των οχημάτων. Ο αστάθμητος παράγοντας της εμφάνισης δυναμικών εμποδίων, όπως πχ. ένα παρατημένο φορτίο, δεν εξετάζεται στο συγκεκριμένο πρόβλημα. Ωστόσο, μια απόσταση ασφαλείας τόσο από τα στατικά εμπόδια όσο και από τα άλλα οχήματα ενσωματώνεται στον σχεδιασμό μονοπατιού.



Εικόνα 5 – Παράδειγμα κίνησης πολλαπλών AMR εντός μιας αποθήκης [26]

ΚΕΦΑΛΑΙΟ 6 Αλγόριθμος εύρεσης και επίλυσης συγκρούσεων

Έχοντας περιγράψει το πρόβλημα, μπορούμε να προχωρήσουμε σε επίλυση προτείνοντας ένα αλγόριθμο πρόβλεψης και αποτροπής των πιθανών συγκρούσεων μεταξύ των οχημάτων και στην συνέχεια να περιγράψουμε αναλυτικά τον τρόπο εκτέλεσης του. Παρακάτω παρουσιάζεται η εκτέλεση του κώδικα σταδιακά, με σκοπό την πλήρη κατανόηση της φιλοσοφίας του συγκεκριμένου αλγορίθμου. Τα βήματα εκτέλεσης είναι ίδια για κάθε αλγόριθμο εύρεσης μονοπατιού, που ενσωματώνεται, με μικρές διαφοροποιήσεις οι οποίες απαιτούνται για την ομαλή εκτέλεση του.

6.1 Πρώτη φάση εκτέλεσης

Δημιουργία οχημάτων :

Το πρώτο στάδιο της εκτέλεσης του αλγορίθμου είναι οι δημιουργία των ρομποτικών οχημάτων. Το σύστημα διαχείρισης το οποίο θα εξάγει τα τελικά μονοπάτια κίνησης διαθέτει την πληροφορία της θέσης όλων των ρομποτικών οχημάτων εντός της αποθήκης.

Εύρεση αρχικού μονοπατιού:

Το επόμενο βήμα της προσομοίωσης απαιτεί την δημιουργία του αρχικού μονοπατιού το οποίο θα ελεγχθεί αργότερα από τον αλγόριθμο έτσι ώστε να επισημανθούν πιθανές συγκρούσεις με άλλα οχήματα και να επιλυθούν δυναμικά. Για την δημιουργία των αρχικών μονοπατιών χρησιμοποιούνται οι αλγόριθμοι A*, D*, D* Lite καθώς και DRRT, οι οποίοι περιγράφονται στο κεφάλαιο (4).

Από τις παραμέτρους του αλγορίθμου DRRT, η μεταβλητή «step length» είναι το μήκος βήματος που χρησιμοποιείται στον αλγόριθμο. Καθορίζει τη μέγιστη απόσταση που μπορεί να διανυθεί σε ένα μόνο βήμα από έναν κόμβο προς έναν άλλο. Στον αλγόριθμο, αυτό το μήκος βήματος χρησιμοποιείται για τον περιορισμό της απόστασης μεταξύ των κόμβων κατά την επέκταση του δέντρου.

Η παράμετρος «goal sampling rate» ελέγχει την πιθανότητα επιλογής του κόμβου-στόχου απευθείας ως τυχαίο υποψήφιο κόμβο κατά τη διάρκεια της επέκτασης του δέντρου. Μια υψηλότερη τιμή του «goal sampling rate» αυξάνει την πιθανότητα επιλογής του κόμβου-στόχου, στρέφοντας την αναζήτηση προς την θέση του και κάνοντας τον αλγόριθμο να συμπεριφέρεται περισσότερο σαν μια αναζήτηση «best-first».

Η παράμετρος «waypoint sampling rate» ρυθμίζει την πιθανότητα επιλογής ενός προηγούμενου έγκυρου κόμβου κατά τη διάρκεια της διαδικασίας σχεδιασμού. Αυτή η πιθανότητα επηρεάζει την προτίμηση για την επαναχρησιμοποίηση επιτυχημένων λύσεων και μπορεί να βελτιώσει την αποτελεσματικότητα του αλγορίθμου καθοδηγώντας την ανάπτυξη του δέντρου προς γνωστές καλές διαδρομές. Η προσαρμογή της τιμής του «waypoint sampling rate» επιτρέπει ένα συμβιβασμό μεταξύ της εξερεύνησης νέων μονοπατιών και της εκμετάλλευσης προηγούμενων επιτυχημένων λύσεων.

Τέλος η παράμετρος «max iterations» καθορίζει το μέγιστο αριθμό επαναλήψεων εκτέλεσης του αλγορίθμου.

Σε αυτό το σημείο πρέπει να επισημανθεί ότι τα μονοπάτια που εξάγονται από τους αλγορίθμους της οικογένειας star περιέχουν κόμβους με συντεταγμένες (x, y) όπου η μεταξύ τους απόσταση είναι σταθερή.

Στην περίπτωση του DRRT, η απόσταση των κόμβων που περιέχονται στο μονοπάτι που παράγεται εξαρτάται από τις αρχικές παραμέτρους που δίνονται στον αλγόριθμο. Οι κόμβοι αυτοί δεν είναι τοποθετημένοι πάνω στο πλέγμα εφόσον η δειγματοληψία γίνεται στην περιοχή γύρω από τον κόμβο με ακτίνα ίση με την τιμή της μεταβλητής “step-length”. Συνεπώς η ακτίνα της περιοχής καθορίζει και την μέγιστη απόσταση μεταξύ των κόμβων του μονοπατιού. Αντίθετα με τα μονοπάτια των άλλων αλγορίθμων, το μονοπάτι που παράγει ο DRRT δημιουργεί πρόβλημα στον συντονισμό κίνησης και τον έλεγχο πιθανών συγκρούσεων. Μια πιθανή λύση στο πρόβλημα αυτό θα ήταν η τροποποίηση του αλγορίθμου έτσι ώστε να πραγματοποιεί την δειγματοληψία των υποψήφιων κόμβων πάνω σε πλέγμα. Επίσης θα έπρεπε η ακτίνα εξερεύνησης να είναι σταθερή. Οι δυο ενέργειες αυτές περιορίζουν σημαντικά την απόδοση του αλγορίθμου με αποτέλεσμα αρκετές φορές να μην ήταν καν εφικτή η εύρεση μονοπατιού εντός των προκαθορισμένων επαναλήψεων εκτέλεσης.

Η προτεινομένη λύση στο πρόβλημα αυτό είναι η προσθήκη σημείων, στο μονοπάτι, με τη χρήση γραμμικής παρεμβολής (linear interpolation), έτσι ώστε η απόσταση μεταξύ όλων των σημείων να προσεγγίζει την τιμή της απόστασης μεταξύ των σημείων του μονοπατιού που εξάγουν οι αλγόριθμοι star. Παρακάτω παρουσιάζεται ο ψευδοκώδικας που υπολογίζει το πλήθος και τις θέσεις των νέων σημείων που παρεμβάλλονται μεταξύ δυο διαδοχικών σημείων του μονοπατιού που παράγει ο DRRT. Η τεχνική αυτή εφαρμόζεται για όλα τα σημεία του μονοπατιού. Οι τιμές αυτές υπολογίζονται δυναμικά και έχουν ικανοποιητικά αποτελέσματα για κάθε τιμή της ακτίνας εξερεύνησης του DRRT.

Ψευδοκώδικας INTERPOLATE-POINTS

Function INTERPOLATE-POINTS ((x1,y1),(x2, y2), value, t)

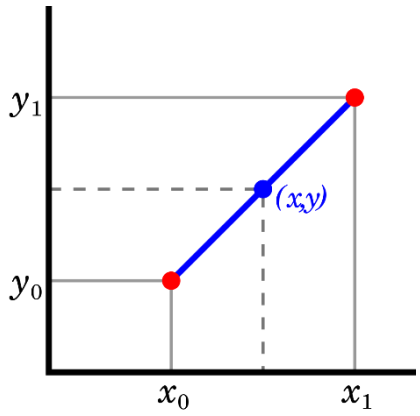
1. *t:* minimum distance threshold of interpolated points

// Calculate the differences in x and y coordinates
2. *dx = x2 - x1*
3. *dy = y2 - y1*

// Calculate the Euclidean distance between the points
4. *distance = square-root(dx^2 + dy^2)*

// Check if the distance between the points is less than a threshold
5. IF *distance < t* THEN
    // Return the endpoint as a single interpolated point
6. return (x2, y2)
7. k = distance/value
// Determine the number of intermediate points to be interpolated
8. IF *k - floor(k) <= 0.5* THEN
9. *spaces = floor(k)*
10. ELSE *spaces = floor(distance) + 1*

// Calculate the increment value for evenly spacing the points
11. increment = distance / spaces



Εικόνα 6.1 – Linear Interpolation (https://en.wikipedia.org)

// Initialize an empty list to store the interpolated points
12. Interpolated points = list

// Iterate over the line segment to calculate the intermediate points and calculate the coordinates of the interpolated point
13. FOR *i = 1* to *spaces – 1* :
14 *x = x1 + (x2 - x1) * (i * increment / distance)*
15. *y = y1 + (y2 - y1) * (i * increment / distance*)

// Append the interpolated point to the list
16. i*nterpolated points ADD (x, y)*
17. *interpolated points ADD (x2, y2)*

// Add the endpoint as the final interpolated point
18. return *interpolated points*

6.2 Δεύτερη φάση εκτέλεσης

Συντονισμένη κίνηση του στόλου οχημάτων AMR:
Το δεύτερο στάδιο του αλγορίθμου είναι ο συντονισμός της κίνησης των ρομποτικών οχημάτων πάνω στα αρχικά σχεδιασμένα μονοπάτια, η εύρεση πιθανών συγκρούσεων και η αντιμετώπιση τους, έτσι ώστε κάθε όχημα να διαθέτει ένα μονοπάτι που του επιτρέπει την κίνηση εντός της αποθήκης, από το σημείο έναρξης ως το σημείο τερματισμού, χωρίς να παρεμποδίζεται από τα άλλα οχήματα. Η διαδικασία αυτή είναι ένας επαναληπτικός έλεγχος των θέσεων (συντεταγμένων) των ρομποτικών οχημάτων. Μια επανάληψη αποτελεί την μονάδα χρόνου της προσομοίωσης. Ο δείκτης των επαναλήψεων παρέχει πρόσβαση σε όλες τις θέσεις των οχημάτων για οποιαδήποτε χρονική στιγμή της προσομοίωσης. Θεωρούμε ότι η ταχύτητα κίνησης των οχημάτων είναι ένας κόμβος (x, y) ανά επανάληψη. Για το μονοπάτι που παράγει ο αλγόριθμος DRRT η ταχύτητα είναι πάλι ένας κόμβος ανά επανάληψη με τη διαφορά ότι οι κόμβοι απέχουν περίπου την ίδια απόσταση. Αυτό συμβαίνει διότι στο τελικό μονοπάτι κίνησης που παράγει ο DRRT έχουν προστεθεί κόμβοι (σημεία) έτσι ώστε η απόσταση των διαδοχικών κόμβων να προσεγγίζει την απόσταση των κόμβων των μονοπατιών που σχεδιάστηκαν σε πλέγμα από τους A*, D*, D* Lite. Ο λόγος που εφαρμόζεται αυτή η στρατηγική είναι να προσεγγιστεί η ταχύτητα κίνησης στα μονοπάτια των προηγουμένων αλγορίθμων, χωρίς να μεταβάλλεται το συνολικό κόστος (μήκος), έτσι ώστε τα στατιστικά που εξάγονται να μπορούν να συσχετιστούν και να γίνει εφικτή η σύγκριση της απόδοσης όλων των αλγορίθμων σχεδιασμού μονοπατιού που χρησιμοποιούνται.

Προσδιορισμός θέσεων:
Το πρώτο βήμα του ελέγχου πιθανών συγκρούσεων είναι η εύρεση της θέσης του κάθε οχήματος κάθε χρονική στιγμή. Η εύρεση της θέσης επιτυγχάνεται με τη χρήση του δείκτη της επανάληψης. Χρησιμοποιούμε τον δείκτη αυτό και ανατρέχουμε στη λίστα που περιλαμβάνει τα σημεία του μονοπατιού, για κάθε όχημα και επιστρέφουμε την τιμή της λίστας που αντιστοιχεί σε αυτό τον δείκτη. Παρόλα τα παραπάνω, τα μονοπάτια των προσομοιωμένων AMR ενδέχεται να διαφέρουν σε μήκος. Για το λόγο αυτό απαιτείται ιδιαίτερη προσοχή στην ανάκτηση της θέσης των οχημάτων. Την χρονική στιγμή που ένα όχημα εκτελεί την διαδρομή του υπάρχει πιθανότητα ένα άλλο όχημα να βρίσκεται ήδη στον

τερματισμό. Στην περίπτωση αυτή πραγματοποιείται σχετικός έλεγχος για το αν ο δείκτης της επανάληψης είναι μεγαλύτερος από τον δείκτη του τελευταίου σημείου της λίστας που αντιπροσωπεύει το μονοπάτι ενός οχήματος. Σε περίπτωση που αυτό αληθεύει το πρόβλημα λύνεται με την χρήση του τελευταίου στοιχείου της λίστας, για το συγκεκριμένο όχημα, το οποίο έχει ολοκληρώσει τη διαδρομή του.

Έλεγχος πιθανών συγκρούσεων:
Σε κάθε χρονική στιγμή (επανάληψη) της εκτέλεσης του αλγορίθμου γίνεται έλεγχος πιθανών συγκρούσεων μεταξύ των οχημάτων. Ο έλεγχος αυτός προσδιορίζει τις θέσεις των οχημάτων σύμφωνα με την παραπάνω διαδικασία και ελέγχει όλους τους συνδυασμούς των θέσεων συγκρίνοντας δυο τιμές ανά επανάληψη. Με αυτό τον τρόπο ελέγχονται οι πιθανές συγκρούσεις για όλα τα οχήματα την συγκεκριμένη χρονική στιγμή. Σημαντική παρατήρηση αποτελεί το γεγονός ότι η σειρά ελέγχου δεν επηρεάζει το αποτέλεσμα της εκτέλεσης του αλγορίθμου εφόσον το όχημα που θα αναπροσαρμόσει το μονοπάτι του επιλέγεται με μια συγκεκριμένη συνθήκη. Επιπλέον κατά της διάρκεια της αναζήτησης πιθανών συγκρούσεων ελέγχονται όλοι οι συνδυασμοί των οχημάτων, όπως πχ. το πρώτο όχημα με το τελευταίο, στην αρχή του ελέγχου, άλλα και αντίστροφα το τελευταίο με το πρώτο. Αυτός ο έλεγχος εφαρμόζεται διότι είναι πιθανό να έχει μεσολαβήσει κάποια αναγνώριση πιθανής σύγκρουσης με άλλο όχημα σε προηγούμενο έλεγχο και κάποιο από τα δυο αυτά οχήματα να έχει προσαρμόσει το μονοπάτι του στις καινούργιες συνθήκες.
Η αναγνώριση πιθανών συγκρούσεων επιτυγχάνεται με μια συνάρτηση οι οποία ελέγχει τις τιμές (x, y) για δυο οχήματα. Πιο συγκεκριμένα ελέγχεται αν ένα όχημα βρίσκεται στην γειτονική περιοχή του άλλου. Αν η παραπάνω συνθήκη αληθεύει, θεωρούμε πως υπάρχει πιθανή σύγκρουση μεταξύ των οχημάτων που την ικανοποιούν και στην συνέχεια καλείται ο αλγόριθμος σχεδιασμού μονοπατιού που χρησιμοποιήθηκε αρχικά, με σκοπό τον επανασχεδιασμό του μονοπατιού και κατ' επέκταση την επίλυση της σύγκρουσης.
Η επιλογή τους οχήματος που θα αναδιαμορφώσει το μονοπάτι του γίνεται με βάση την θέση του οχήματος εκείνη την χρονική στιγμή. Πιο συγκεκριμένα επιλέγεται το όχημα το οποίο βρίσκεται πιο κοντά στον κόμβο τερματισμού. Σε περίπτωση που τα δύο οχήματα ισαπέχουν από τον κόμβο τερματισμού, επιλέγεται ένα τυχαία.

Επανασχεδιασμός μονοπατιού:
Όσο αφορά τον επανασχεδιασμό του μονοπατιού, στα σενάρια που γίνεται χρήση δυναμικών αλγορίθμων, όπως ο D*, D* Lite και ο DRRT μπορούμε εύκολα να επιλύσουμε τις συγκρούσεις εκμεταλλευόμενοι τις δυναμικές ιδιότητες τους. Αρκεί απλώς να ενημερώσουμε την λίστα των εμποδίων για κάθε όχημα, προσθέτοντας σε αυτή τις θέσεις όλων των άλλων οχημάτων την συγκεκριμένη χρονική στιγμή. Ο λόγος που προσθέτουμε ως εμπόδιο τις θέσεις όλων των άλλων οχημάτων είναι το γεγονός ότι κάποιο όχημα το οποίο δεν εμποδίζει αρχικά μπορεί να βρίσκεται ιδιαίτερα κοντά στο όχημα που έχει επιλεχθεί να αναπροσαρμόσει το μονοπάτι του, με αποτέλεσμα αν δεν ληφθεί υπόψιν η θέση του συγκεκριμένου οχήματος, το νέο μονοπάτι που θα δημιουργηθεί για το αρχικό όχημα, να οδηγεί σε άμεση σύγκρουση με το γειτονικό του.
Στην περίπτωση που γίνεται χρήση του μη δυναμικού αλγορίθμου A* η διαδικασία επίλυσης των συγκρούσεων και επανασχεδιασμού του μονοπατιού αλλάζει πλήρως. Εφόσον ο αλγόριθμος A* δεν διαθέτει την δυνατότητα να προσαρμόζει δυναμικά το μονοπάτι του σύμφωνα με τις αλλαγές στο περιβάλλον του, επινοήθηκε ένα τέχνασμα που παραβλέπει αυτή την αδυναμία και δημιουργεί ένα καινούργιο μονοπάτι αξιοποιώντας το παλιό

μονοπάτι έως την στιγμή που εμφανίστηκε το εμπόδιο. Πιο συγκεκριμένα σε περίπτωση που ικανοποιείται η συνθήκη σύγκρουσης, υπολογίζεται ο αμέσως προηγούμενος από τον τρέχοντα κόμβο και εκτελείται εκ νέου ο αλγόριθμος A* αφού προσθέσουμε στην λίστα των στατικών εμποδίων όλες τις θέσεις των άλλων οχημάτων εκείνη την χρονική στιγμή. Στην συνέχεια ο προηγούμενος κόμβος δίνεται ως κόμβος έναρξης στον αλγόριθμο. Ο κόμβος τερματισμού παραμένει ο ίδιος. Το τέχνασμα αυτό παράγει ένα μονοπάτι χωρίς συγκρούσεις. Η απόδοση του είναι ιδιαίτερα ικανοποιητική εφόσον ο A* χρησιμοποιεί μια ευρετική συνάρτηση που του επιτρέπει να εστιάζει στην αναζήτηση περιοχής όσο πιο κοντά στον κόμβο τερματισμού είναι εφικτό. Επομένως δεν ελέγχει πολλούς από τους ήδη εξερευνημένους κόμβους, παρά μόνο αν βρίσκεται πολύ μακριά από τον τερματισμό. Ακόμη και σε αυτή την περίπτωση το πλήθος των κόμβων που ελέγχονται παραπάνω από μια φορά είναι μικρό.

Σε κάθε εκτέλεση των αλγορίθμων σχεδιασμού μονοπατιού, τόσο στην αρχή όσο και κατά τη εκτέλεση του αλγορίθμου, συλλέγονται και τα σχετικά στατιστικά τα οποία θα αξιοποιηθούν για την σύγκριση της απόδοσης των αλγορίθμων σχεδιασμού μονοπατιού, σε συγκεκριμένα περιβάλλοντα αποθηκών. Η αναλυτική περιγραφή των στατιστικών πραγματοποιείται στο κεφάλαιο 7.

6.3 Ψευδοκώδικας:

Function MAIN ():

```
        //Initialize variables
1.      obs: list of environment obstacles coordinate (global variable)
2       md: minimum allowed distance from an obstacle (global variable)
3.      n: number of robots (global variable)
4.      p: list of paths, initially EMPTY (global variable)
5.      r: list of robots containing (start, goal) as a pair of (x,y) coordinates (global variable)
6.      sn: total steps of the execution, initially 0

7.       FOR i=0 till i=n
8.              r[i] = (start,goal)
9.              CalculatePath(start, goal, p[i], obs)      // Calculates a path using one
                                                           // of the path planning algorithms (A*, D*,
                                                           //D* Lite, DRRT)
10.             sn = MAX (sn, LENGTH of p[i])

11.     FOR iteration=0 till iteration=sn:
12.             DetectResolveCollisions(iteration)
```

Function DetectResolveCollisions(iteration):

```
1.  FOR i=0 till i=n:
2.      FOR j=0 till j=n:
3.          IF i equals j THEN CONTINUE
4.          ELSE robot1 = CurrentCoordinates(p[i], iteration)
5.              robot2 = CurrentCoordinates(p[j], iteration)
6.          IF CheckCollision(robot1, robot2) returns True:
7.              ri = LENGTH(p[i])-iteration
8.              rj = LENGTH(p[j])-iteration
9.              IF ri AND rj <=0:
10.                 CONTINUE
11.             index = SelectRobot(ri, rj, i, j)
12.             RecalculatePath(index, iteration)
```

Function CurrentCoordinates(path, iteration):

```
1.  last = LENTGTH(path)-1
2.  pos = MINIMUM (iteration, last)
3.  (x, y) = path[pos]
4.  Return (x, y)
```

Function CheckCollision(robot1, robot2):

```
1.  IF Euclidian Distance of robot1 and robot2 <= md        //md: minimum allowed distance
2.      Return TRUE
```

Function SelectRobot(ri, rj, i, j):

```
1.  IF ri>0 AND rj>0:
2.      IF ri<rj:
3.          return i
4.      ELSE IF ri>rj:
5.          return j
6.      ELSE return RANDOM (i, j)
7.  ELSE IF ri>0 return i
8.  ELSE return j
```

Function RecalculatePath(index, iteration):

```
1.    goal = p[index][LAST]
2.    IF Planning Algorithm is A*:
3.        start = p[index][iteration-1]      // If A* is selected we set as start node the one
                                             // before collision
4.    ELSE start = p[index][FIRST]
5.    w = list of other robots current position, initialized as EMPTY
6.    FOR i=0 till i=n:
7.        IF p[i] ≠ p[index]
8.            cp = CurrentCoordinates(p[i], iterartion)
9.            w ADD cp
10.   w = w + obs      // obs contains the environment obstacle coordinates
11.   CalculatePath(start, goal, p[index], w)    // Calculate new path using w as obstacle list
```

Περιγραφή της εκτέλεσης του ψευδοκώδικα:

Εκτέλεση της συνάρτησης **MAIN**:

1. Αρχικοποίηση μεταβλητών:

   obs: λίστα με τις συντεταγμένες των εμποδίων του περιβάλλοντος (μεταβλητη global)
   md: ελάχιστη επιτρεπτή απόσταση από ένα εμπόδιο (μεταβλητή global)
   n: πλήθος ρομποτικών οχημάτων (μεταβλητή global)
   p: λίστα με τα μονοπάτια, η οποία αρχικά είναι κενή (μεταβλητή global)
   r: λίστα με τα ρομποτικά οχήματα η οποία περιέχει τους κόμβους (start, goal) ως ζεύγος συντεταγμένων (x, y) (μεταβλητή global)
   sn: συνολικές επαναλήψεις της εκτέλεσης του αλγορίθμου, αρχικά έχει την τιμή 0

2. Εύρεση μονοπατιού: Στη γραμμή 9 της συνάρτησης main καλείται η συνάρτηση CalculatePath η οποία υπολογίζει τα αρχικά μονοπάτια με τη χρήση των αλγορίθμων A*, D*, D* Lite και DRRT.

3. Αριθμός επαναλήψεων: Υπολογισμός του αριθμού των επαναλήψεων της εκτέλεσης του αλγορίθμου και αποθήκευση στη μεταβλητή sn (γραμμή 10). Ο αριθμός αυτός ισούται με το πλήθος των κόμβων στο μεγαλύτερο μονοπάτι.

4. Επαναληπτική εκτέλεση του αλγορίθμου εύρεσης και επίλυση συγκρούσεων sn φορές: Στη γραμμή 12 εκτελείται η συνάρτηση DetectResolveCollisions. Η συνάρτηση αυτή δέχεται ως όρισμα τη μεταβλητή iteration η οποία είναι η χρονική μονάδα της εκτέλεσης του αλγορίθμου. Η μεταβλητή αυτή αξιοποιείται για τον προσδιορισμό της θέσης των ρομποτικών οχημάτων ανά πάσα χρονική στιγμή.

Εκτέλεση της συνάρτησης **DetectResolveCollisions**:

1. Προσδιορισμός της θέσης των οχημάτων: Στις γραμμές 1 και 2 της συνάρτησης, πραγματοποιείται εύρεση της θέσης των ρομποτικών οχημάτων μέσω εμφωλευμένης επανάληψης. Σε κάθε επανάληψη ελέγχονται 2 οχήματα. Η εύρεση αυτή πραγματοποιείται μέσω της συνάρτησης CurrentCoordinates, η οποία δέχεται ως όρισμα τη λίστα με το μονοπάτι ενός οχήματος (γραμμές 4 και 5) και χρησιμοποιεί τη μεταβλητή iteration ως δείκτη της λίστας για να επιστρέψει τη θέση που αντιστοιχεί σε εκείνη τη χρονική στιγμή. Αν η μεταβλητή iteration έχει τιμή μεγαλύτερη από την τιμή του δείκτη του τελευταίου στοιχείου (LAST), επιστρέφεται το τελευταίο στοιχείο της λίστας. Η συνάρτηση εκτελείται για τα 2 οχήματα που ελέγχονται και αποθηκεύει τις θέσεις τους σε 2 μεταβλητές της robot1 και robot2 αντίστοιχα.

2. Έλεγχος πιθανών συγκρούσεων: Ο έλεγχος πιθανών συγκρούσεων μεταξύ των ρομποτικών οχημάτων πραγματοποιείται με τη χρήση της συνάρτησης CheckCollision στη γραμμή 6. Η συνάρτηση αυτή δέχεται ως ορίσματα τις μεταβλητές με τις θέσεις των οχημάτων (robot1 και robot2) και επιστρέφει μια μεταβλητή boolean με τιμή true εάν η ευκλείδεια απόσταση μεταξύ των παραπάνω θέσεων είναι μικρότερη της μεταβλητής md (ελάχιστη επιτρεπτή απόσταση).

3. Μεταβλητές ri και rj: Στις γραμμές 7 και 8 της συνάρτησης υπολογίζεται το πλήθος των κόμβων που απομένουν ώστε τα δύο ρομποτικά οχήματα που ελέγχονται, να ολοκληρώσουν τη διαδρομή τους και αποθηκεύεται στις μεταβλητές ri και rj αντίστοιχα. Στην γραμμή 9 της συνάρτησης DetectResolveCollisions ελέγχεται αν οι παραπάνω μεταβλητές έχουν και οι δύο αρνητική τιμή. Στην περίπτωση που εντοπισθεί ότι έχουν αρνητική τιμή σημαίνει ότι και τα 2 οχήματα έχουν τερματίσει και δίνεται εντολή στην συνάρτηση να συνεχίσει με τον έλεγχο του επόμενου συνδυασμού οχημάτων.

4. Επιλογή οχήματος: Στη γραμμή 11 γίνεται κλήση της συνάρτησης SelectRobot η οποία δέχεται ως όρισμα τις μεταβλητές ri και rj καθώς και τους δείκτες της εμφωλευμένης επανάληψης και καθορίζει το όχημα που θα επανασχεδιάσει το μονοπάτι του. Η συνάρτηση SelectRobot συγκρίνει τις τιμές ri και rj και επιστρέφει τον δείκτη που αντιστοιχεί στο όχημα με τη μικρότερη τιμή, η οποία αποθηκεύεται στη μεταβλητή index. Έτσι επιστρέφεται ο δείκτης του οχήματος που βρίσκεται πιο κοντά στον τερματισμό. Αν ένα όχημα έχει ήδη τερματίσει τότε επιλέγεται το άλλο ενώ αν τα 2 οχήματα ισαπέχουν από τον κόμβο τερματισμού τους, επιλέγεται ένα τυχαία (γραμμή 6 της SelectRobot).

5. Επανασχεδιασμός μονοπατιού: Στο τέλος της συνάρτησης εύρεσης και επίλυσης συγκρούσεων καλείται η συνάρτηση RecalculatePath η οποία δέχεται ως ορίσματα τις μεταβλητές iteration και index. Η συνάρτηση αυτή αποθηκεύει σε μια λίστα w τις θέσεις των υπολοίπων ρομποτικών οχημάτων (γραμμές 6 έως 9 της Recalculate Path), προσθέτει σε αυτή τη λίστα τα στατικά εμπόδια και με τη χρήση της συνάρτησης CalculatePath επανασχεδιάζει δυναμικά το μονοπάτι του οχήματος που έχει επιλεχθεί σύμφωνα με την μεταβλητή index (γραμμή 11 της RecalculatePath).

Στον παραπάνω ψευδοκώδικα η συνάρτηση CalculatePath είναι υπεύθυνη για τον σχεδιασμό μονοπατιού από τον κόμβο start στον κόμβο goal. Θεωρούμε ότι σε κάθε σενάριο επιλέγεται και από ένας αλγόριθμος εκ των A*, D*, D* Lite και DRRT. Αν γίνει επιλογή των δυναμικών αλγορίθμων η συνάρτηση δέχεται ως παράμετρο το προηγούμενο μονοπάτι και το αναμορφώνει δυναμικά σύμφωνα με τις θέσεις των άλλων οχημάτων. Στην περίπτωση του A*, ο αλγόριθμος υπολογίζει εκ νέου ένα μονοπάτι αυτή τη φορά με κόμβο αφετηρίας τον αμέσως προηγούμενο κόμβο πριν την πιθανή σύγκρουση (γραμμές 2,3 της RecalculatePath) και ενώνει το προηγούμενο μονοπάτι με το καινούργιο.

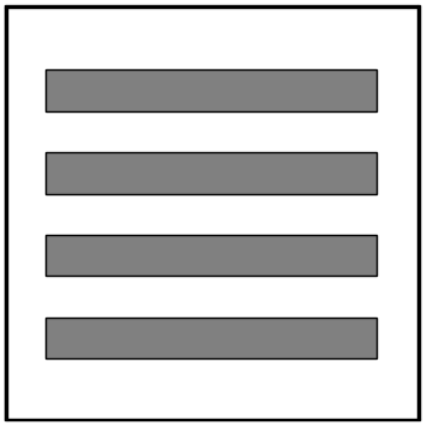
ΚΕΦΑΛΑΙΟ 7 Πειραματική ανάλυση

7.1 Παράμετροι της εκτέλεσης του αλγορίθμου

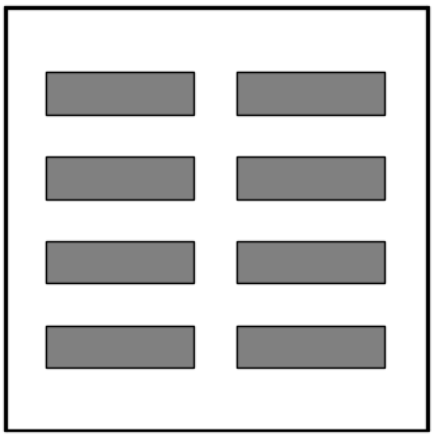
Δημιουργούμε ένα πλήθος Ν αυτόνομων οχημάτων με την προσθήκη των συντεταγμένων (x, y) του κόμβου έναρξης και του κόμβου τερματισμού. Οι δυο κόμβοι αντιπροσωπεύουν δυο σημεία στην αποθήκη. Ο κόμβος έναρξης αποτελεί την αρχική θέση του οχήματος, ενώ ο κόμβος τερματισμού την τελική. Κάθε προσομοιωμένο όχημα διαθέτει επίσης ένα μοναδικό αναγνωριστικό ID καθώς και μία λίστα με όλα τα εμπόδια εντός της αποθήκης. Όσο αφορά τις διαστάσεις των οχημάτων δεν ορίζουμε κάποια τιμή τόσο στην περίπτωση που γίνεται χρήση των A*, D*, D* Lite, η οποία πραγματοποιείται σε περιβάλλον πλέγματος (grid) με μέγεθος κελιού 1x1, όσο και στην εκτέλεση με τον αλγόριθμο DRRT όπου τα οχήματα οι συντεταγμένες των θέσεων παίρνουν και δεκαδικές τιμές (x, y) στον δισδιάστατο χώρο της αποθήκης. Οι επιτρεπτές κινήσεις που μπορεί να εκτελέσει ένα όχημα είναι: οριζόντια, κάθετα και διαγώνια. Για να εντοπίσουμε μια πιθανή σύγκρουση μεταξύ των οχημάτων ορίζουμε την ακτίνα της γειτονικής περιοχής ενός οχήματος ίση με 0,5 . Η ελάχιστη τιμή της απόστασης των οχημάτων από τα στατικά εμπόδια είναι ίση με ένα. Τέλος αρχικές παράμετροι για τον αλγόριθμο DRRT είναι «step length: 4», «goal sampling rate: 0.1», «waypoint sampling rate: 0.7» και «max iterations: 6000».

7.2 Περιγραφή διάταξης αποθηκών (2D)

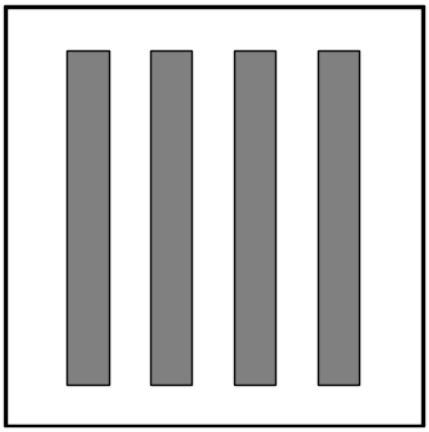
Προκειμένου να γίνει εκτεταμένη σύγκριση της απόδοσης των παραπάνω αλγορίθμων, σε χώρους με πληθώρα αυτόνομων ρομποτικών οχημάτων, δημιουργήθηκαν τέσσερα διαφορετικά περιβάλλοντα αποθηκών, στα οποία και θα εκτελεστεί το πείραμα, για κάθε αλγόριθμο και με διαφορετικό πλήθος των οχημάτων. Όλες οι αποθήκες έχουν μέγεθος 100x100 και περιλαμβάνουν ορθογώνια στατικά εμπόδια τα οποία προσομοιώνουν τα ράφια αποθήκευσης που συνήθως συναντώνται στις εμπορικές αποθήκες. Οι αποθήκες που εξετάστηκαν παρουσιάζονται στις παρακάτω εικόνες:



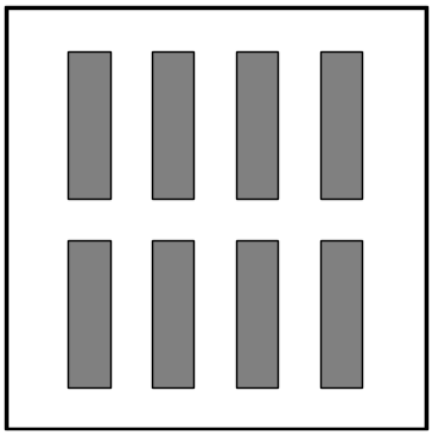
Εικόνα 7.2.1 - Warehouse 1



Εικόνα 7.2.2 - Warehouse 2



Εικόνα 7.2.3 - Warehouse 3



Εικόνα 7.2.4 - Warehouse 4

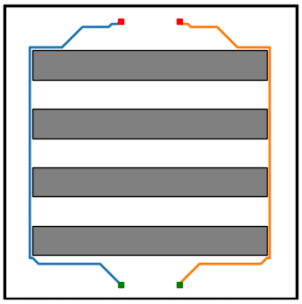
Η αλλαγή της διάταξης είναι ιδιαίτερα σημαντική στην απόδοση των αλγορίθμων σχεδιασμού μονοπατιού εφόσον καθορίζει τις συντεταγμένες των επιτρεπόμενων χώρων κίνησης. Μια αλλαγή στην διάταξη της αποθήκης μπορεί να οδηγήσει σε φαινόμενα όπως η δημιουργία στενών διαδρόμων ή ακόμη και συγκέντρωση πολλών οχημάτων σε ένα συγκεκριμένο σημείο της αποθήκης, εξαιτίας της προτίμησης αυτού τους σημείου, από τους αλγορίθμους σχεδιασμού μονοπατιού, λόγω του χαμηλού συνολικού κόστους (μήκος διαδρομής). Το φαινόμενο αυτό εμφανίζεται στις εικόνες 7.3.48 και 7.3.58, οι οποίες παρουσιάζουν το μονοπάτι του αλγορίθμου D* Lite, στο σενάριο με τα 6 οχήματα, στις αποθήκες 2 και 4 αντίστοιχα.

7.3 Σχεδιασμένα μονοπάτια και διαγράμματα

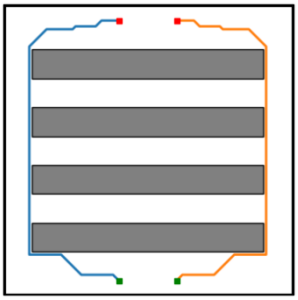
Παρακάτω παρουσιάζονται τα σχεδιασμένα μονοπάτια καθώς και τα στατιστικά που συλλέγονται από την εκτέλεση του αλγορίθμου με δυο, τέσσερα και έξι ρομποτικά οχήματα. Τα στατιστικά αυτά περιλαμβάνουν το πλήθος των εξερευνημένων κόμβων, το μήκος του τελικού μονοπατιού, τον χρόνο υπολογισμού του μονοπατιού, συμπεριλαμβανομένου του χρόνου επανασχεδιασμού, καθώς και το πλήθος των πιθανών συγκρούσεων που διαγνώστηκαν για κάθε όχημα. Για να αναδειχθεί καλύτερα η απόδοση του κάθε αλγορίθμου, σε κάθε περιβάλλον αποθήκης, δημιουργήθηκαν γραφήματα που παρουσιάζουν με συνδυαστικό τρόπο τα στατιστικά της προσομοίωσης. Πιο συγκεκριμένα παρουσιάζεται ο μέσος όρος του αθροίσματος των παραπάνω τιμών για κάθε ρομποτικό όχημα και οι τιμές αυτές συγκρίνονται μεταξύ τους για κάθε αλγόριθμο. Για πρακτικούς λόγους οι επιμέρους τιμές των στατιστικών κάθε οχήματος της προσομοίωσης για κάθε αποθήκη, σε όλα τα σενάρια που εξετάστηκαν τοποθετούνται σε κατάλληλους πίνακες στο τέλος της εργασίας προκειμένου να είναι άμεσα προσβάσιμες.

7.3.1 Σενάριο Ι – Δύο Ρομποτικά οχήματα

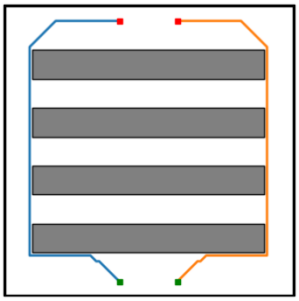
Warehouse 1, 2 Robots:



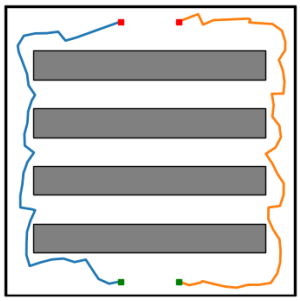
Εικόνα 7.3.1 A Path*



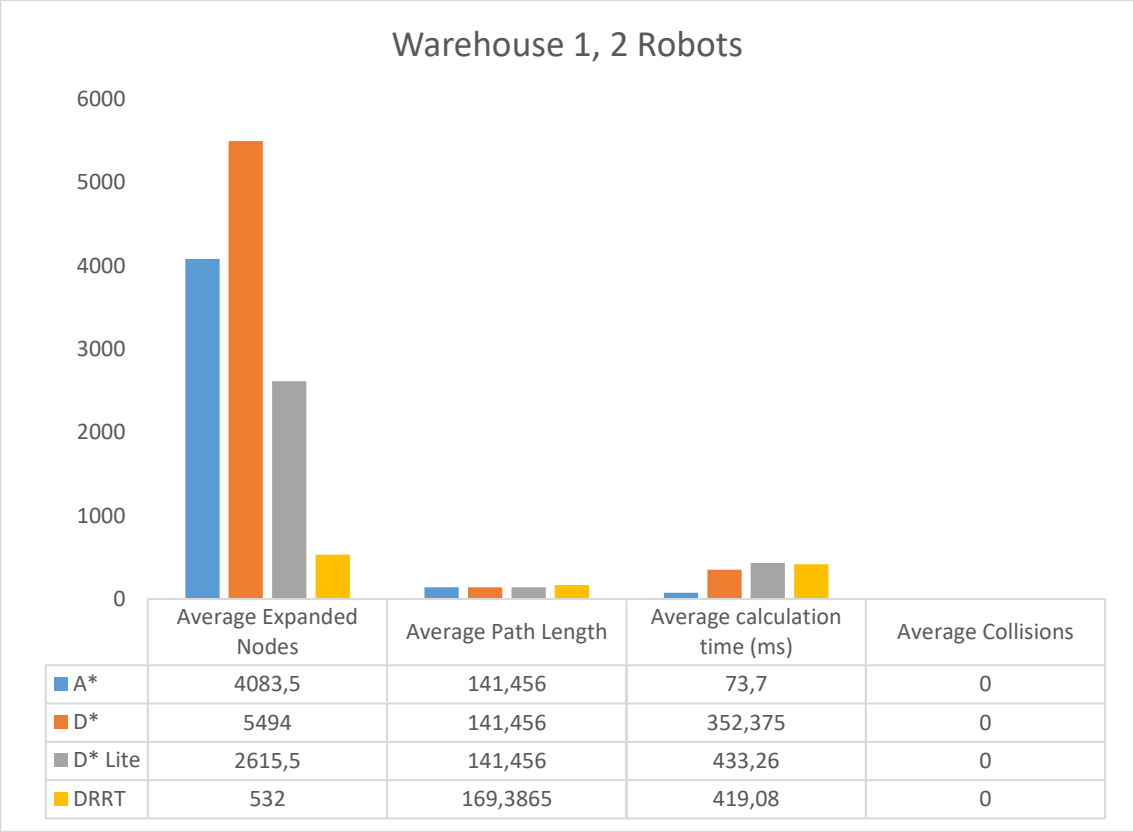
Εικόνα 7.3.2 D Path*



*Εικόνα 7.3.3 D*Lite Path*



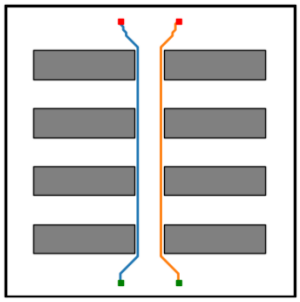
Εικόνα 7.3.4 DRRT Path



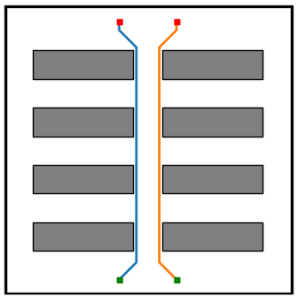
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	4083,5	141,456	73,7	0
D*	5494	141,456	352,375	0
D* Lite	2615,5	141,456	433,26	0
DRRT	532	169,3865	419,08	0

Εικόνα 7.3.5 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 1

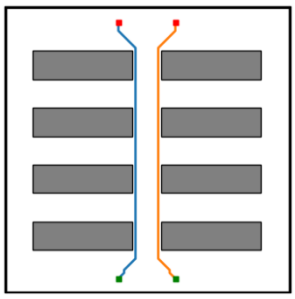
Warehouse 2, 2 Robots:



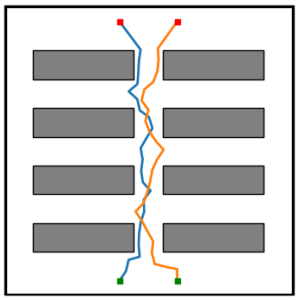
Εικόνα 7.3.6 A Path*



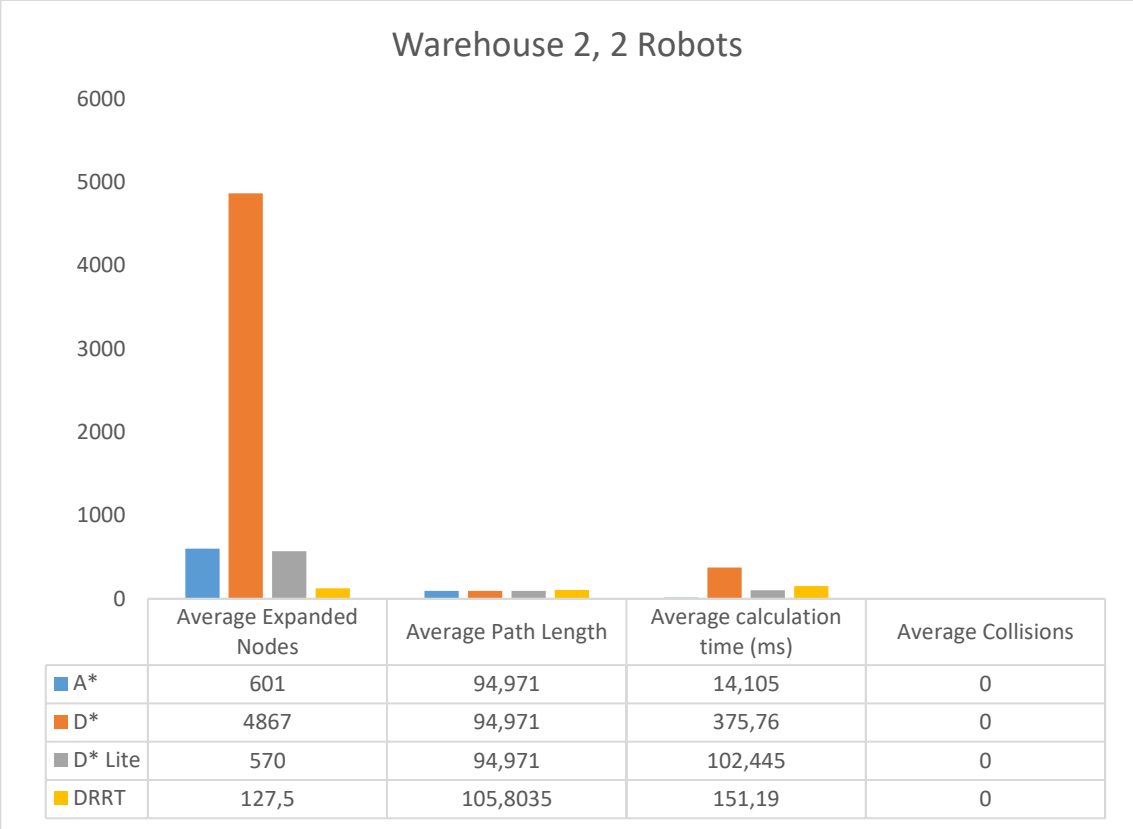
Εικόνα 7.3.7 D Path*



*Εικόνα 7.3.8 D*Lite Path*



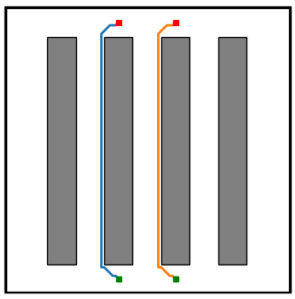
Εικόνα 7.3.9 DRRT Path



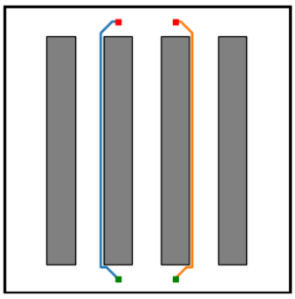
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	601	94,971	14,105	0
D*	4867	94,971	375,76	0
D* Lite	570	94,971	102,445	0
DRRT	127,5	105,8035	151,19	0

Εικόνα 7.3.10 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 2

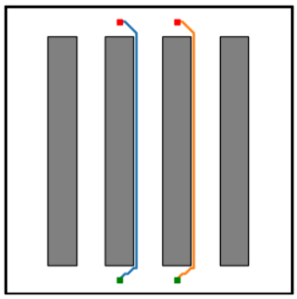
Warehouse 3, 2 Robots:



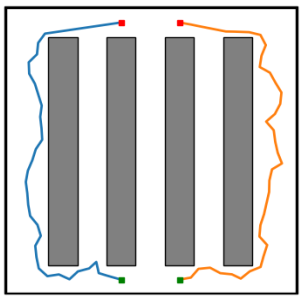
Εικόνα 7.3.11 A Path*



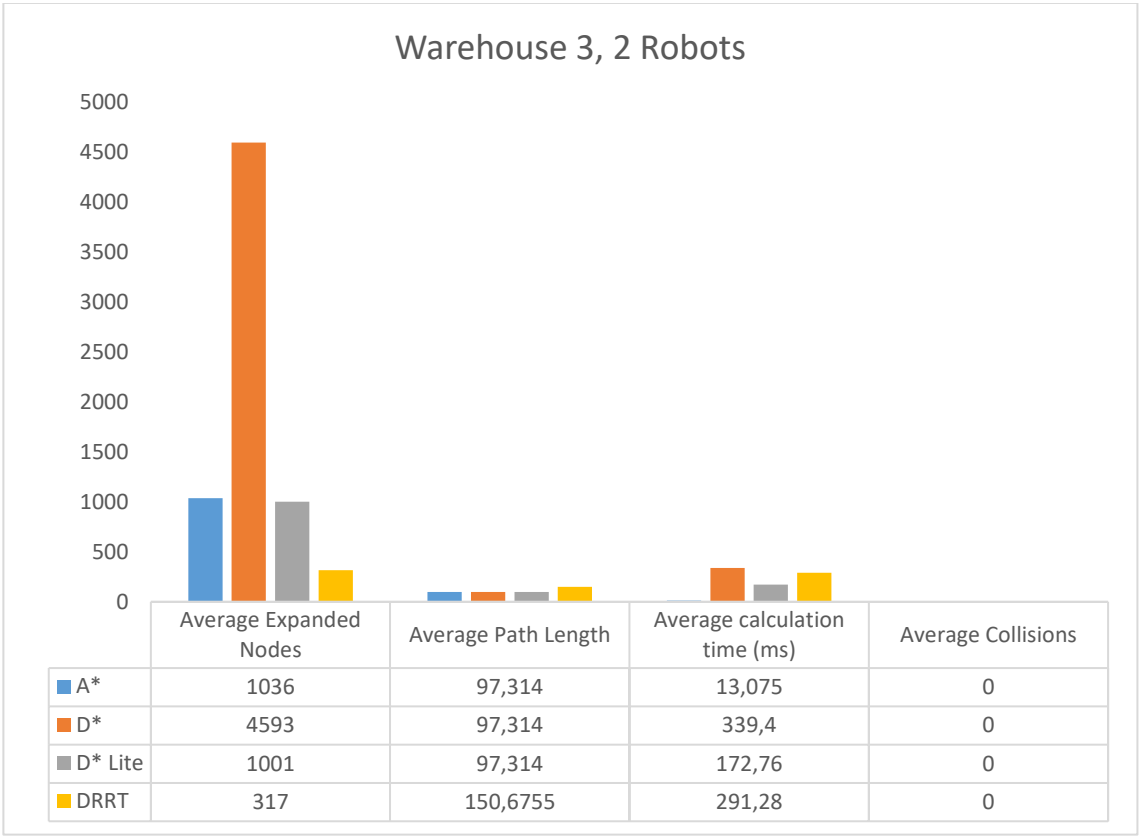
Εικόνα 7.3.12 D Path*



*Εικόνα 7.3.13 D*Lite Path*



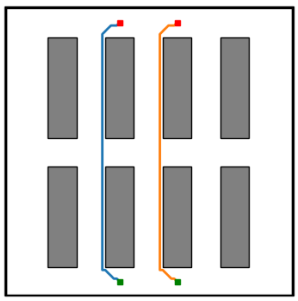
Εικόνα 7.3.14 DRRT Path



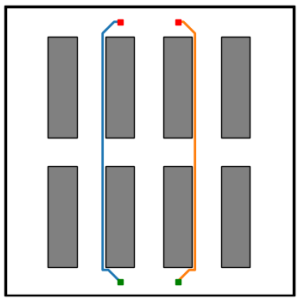
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	1036	97,314	13,075	0
D*	4593	97,314	339,4	0
D* Lite	1001	97,314	172,76	0
DRRT	317	150,6755	291,28	0

Εικόνα 7.3.15 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 3

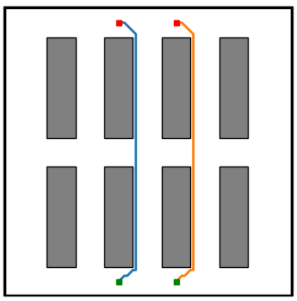
Warehouse 4, 2 Robots:



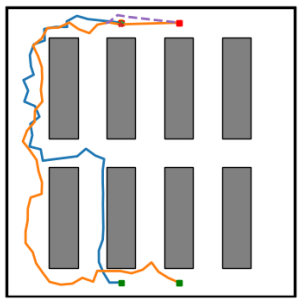
Εικόνα 7.3.16 A Path*



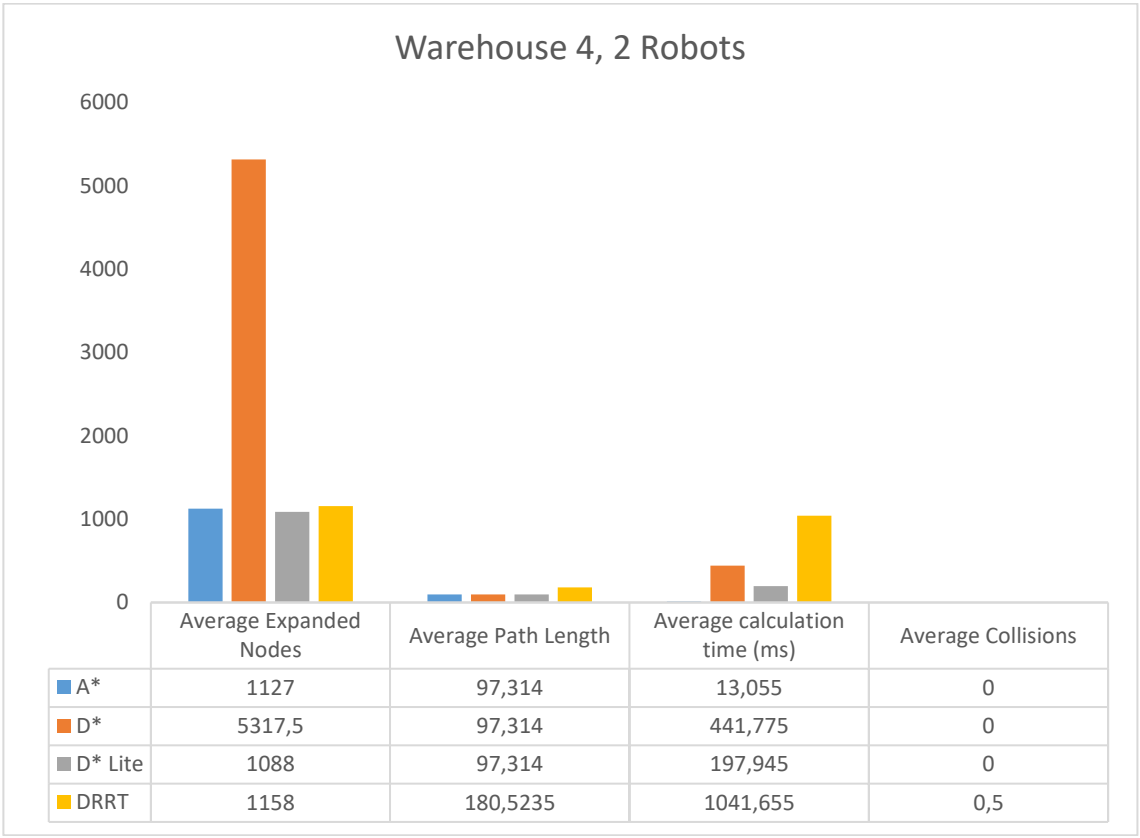
Εικόνα 7.3.17 D Path*



*Εικόνα 7.3.18 D*Lite Path*



Εικόνα 7.3.19 DRRT Path

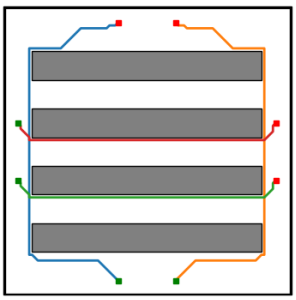


	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	1127	97,314	13,055	0
D*	5317,5	97,314	441,775	0
D* Lite	1088	97,314	197,945	0
DRRT	1158	180,5235	1041,655	0,5

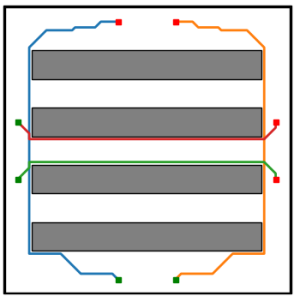
Εικόνα 7.3.20 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 4

7.3.2 Σενάριο ΙΙ – Τέσσερα Ρομποτικά οχήματα

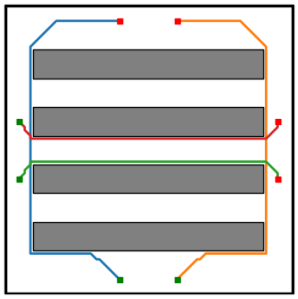
Warehouse 1, 4 Robots:



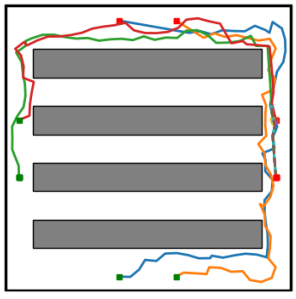
Εικόνα 7.3.21 A Path*



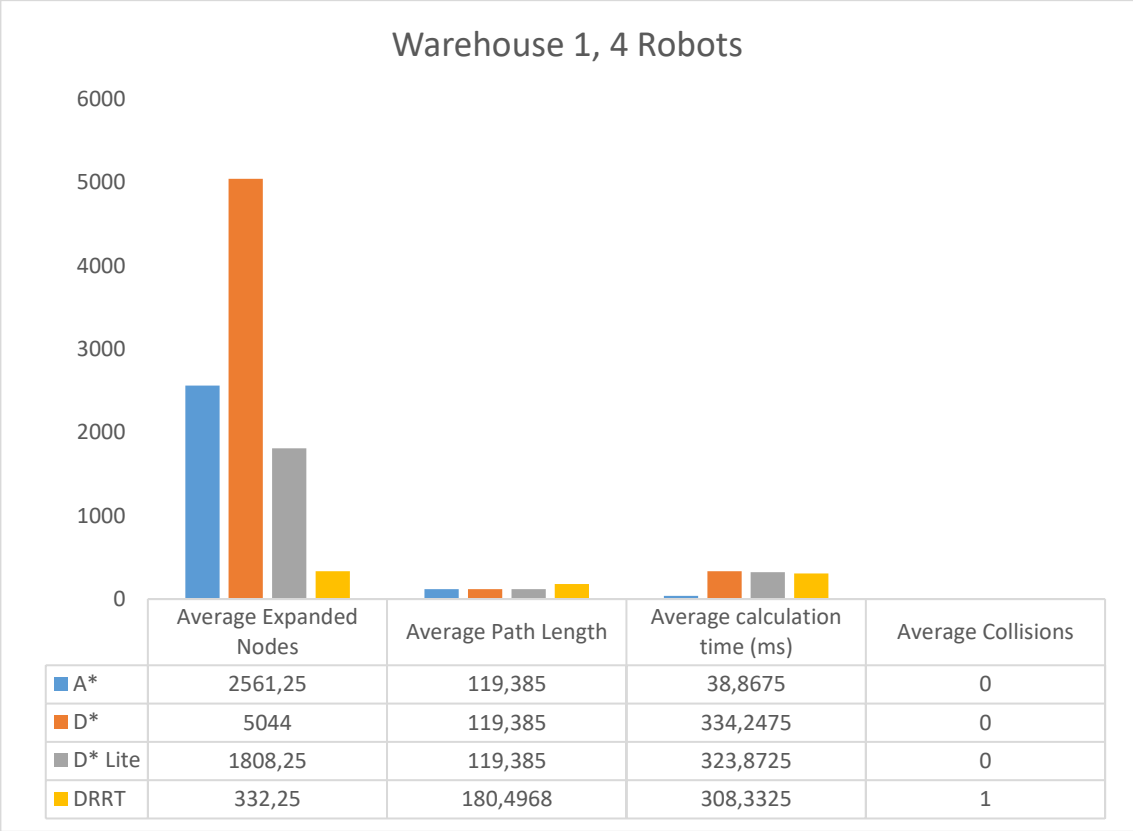
Εικόνα 7.3.22 D Path*



*Εικόνα 7.3.23 D*Lite Path*



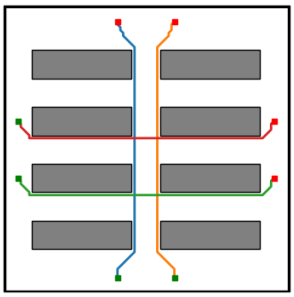
Εικόνα 7.3.24 DRRT Path



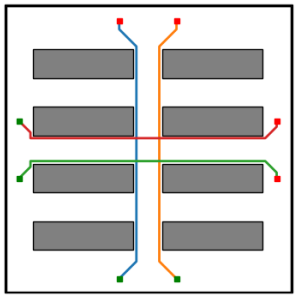
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	2561,25	119,385	38,8675	0
D*	5044	119,385	334,2475	0
D* Lite	1808,25	119,385	323,8725	0
DRRT	332,25	180,4968	308,3325	1

Εικόνα 7.3.25 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 1

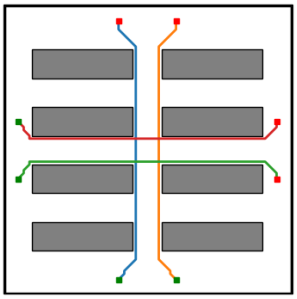
Warehouse 2, 4 Robots:



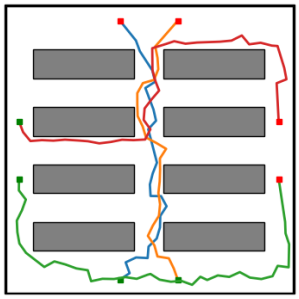
Εικόνα 7.3.26 A Path*



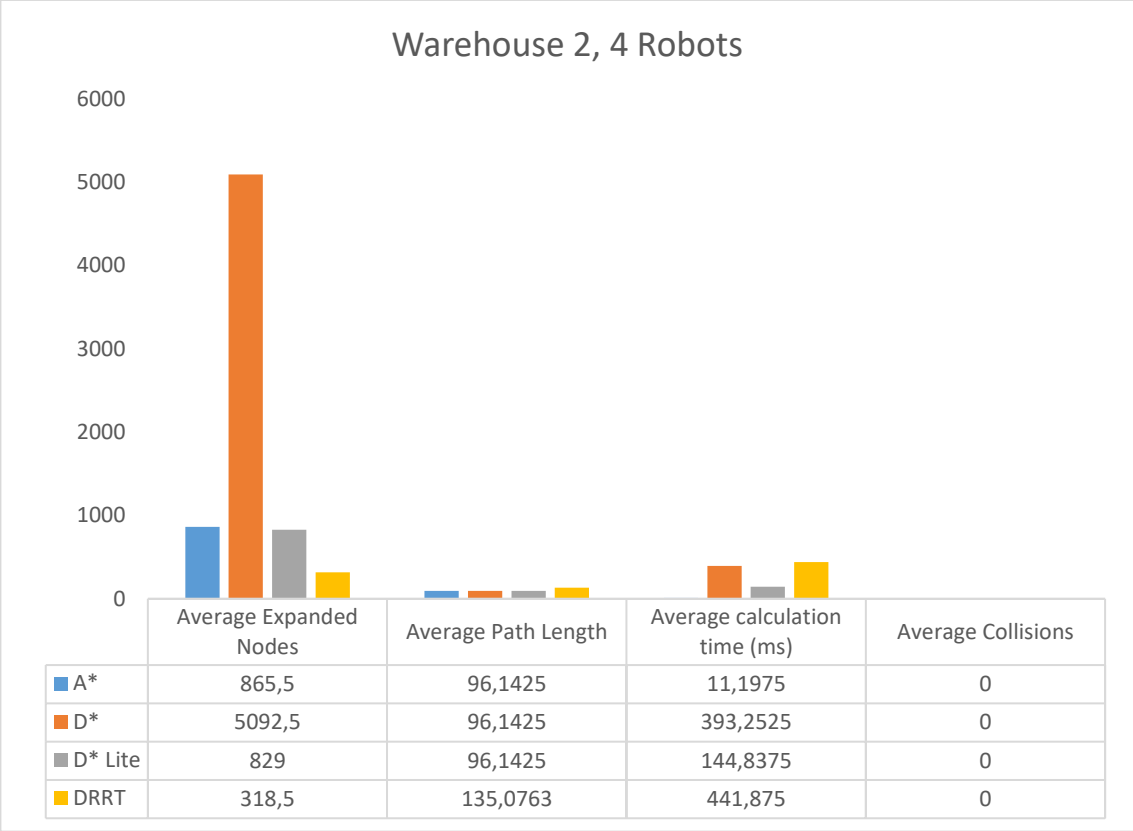
Εικόνα 7.3.27 D Path*



*Εικόνα 7.3.28 D*Lite Path*



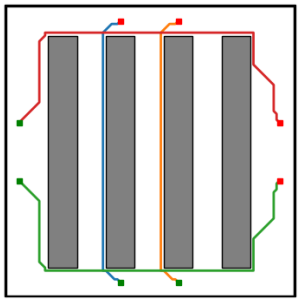
Εικόνα 7.3.29 DRRT Path



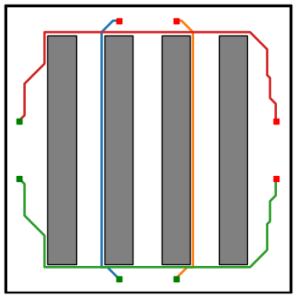
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	865,5	96,1425	11,1975	0
D*	5092,5	96,1425	393,2525	0
D* Lite	829	96,1425	144,8375	0
DRRT	318,5	135,0763	441,875	0

Εικόνα 7.3.30 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 2

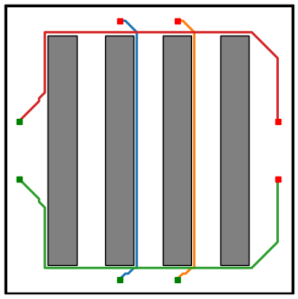
Warehouse 3, 4 Robots:



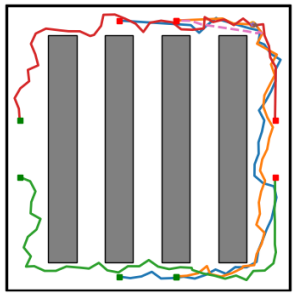
Εικόνα 7.3.31 A Path*



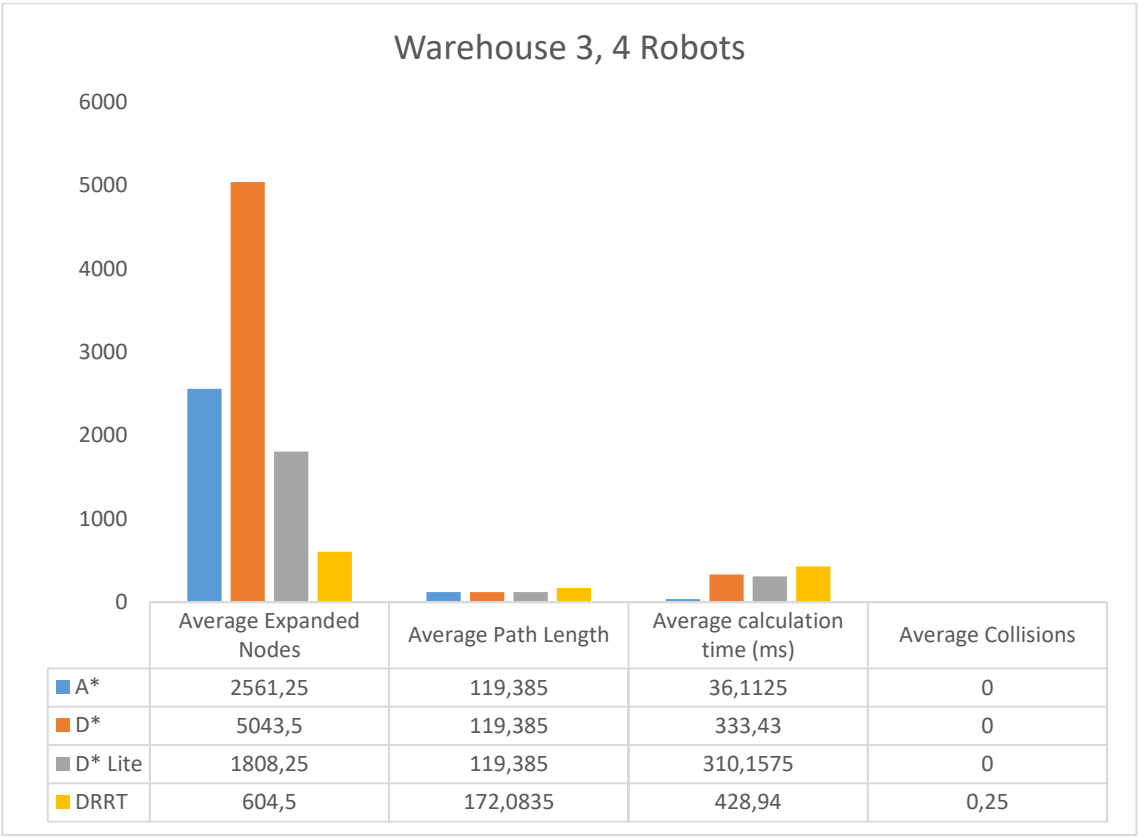
Εικόνα 7.3.32 D Path*



*Εικόνα 7.3.33 D*Lite Path*



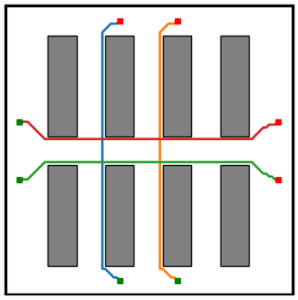
Εικόνα 7.3.34 DRRT Path



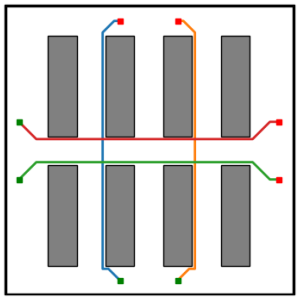
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	2561,25	119,385	36,1125	0
D*	5043,5	119,385	333,43	0
D* Lite	1808,25	119,385	310,1575	0
DRRT	604,5	172,0835	428,94	0,25

Εικόνα 7.3.35 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 3

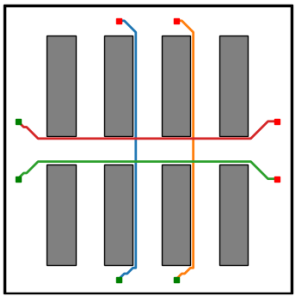
Warehouse 4, 4 Robots:



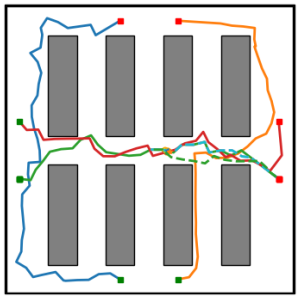
Εικόνα 7.3.36 A Path*



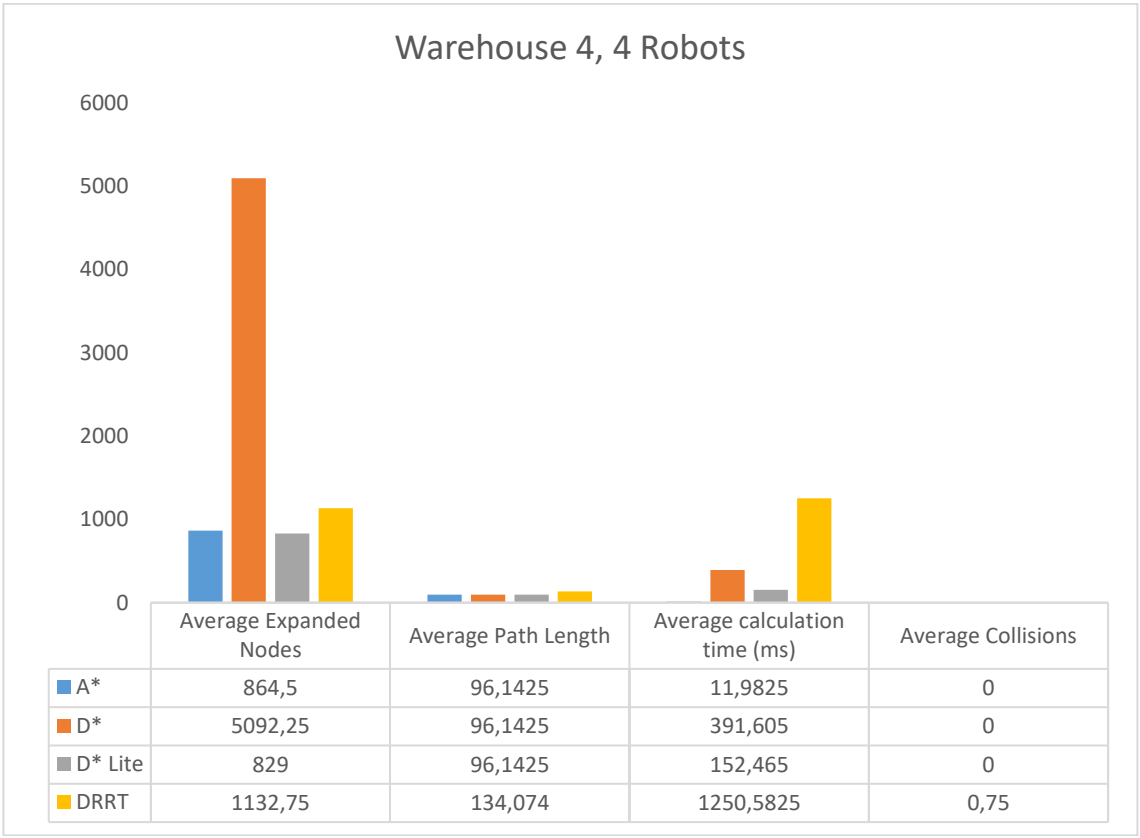
Εικόνα 7.3.37 D Path*



*Εικόνα 7.3.38 D*Lite Path*



Εικόνα 7.3.39 DRRT Path

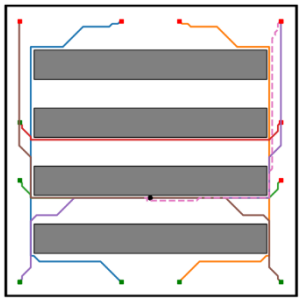


	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	864,5	96,1425	11,9825	0
D*	5092,25	96,1425	391,605	0
D* Lite	829	96,1425	152,465	0
DRRT	1132,75	134,074	1250,5825	0,75

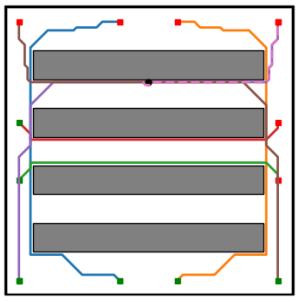
Εικόνα 7.3.40 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 4

7.3.3 Σενάριο III – Έξι Ρομποτικά οχήματα

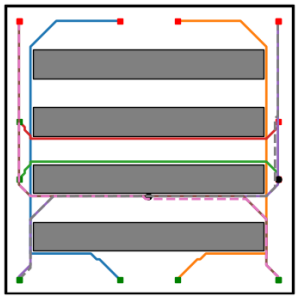
Warehouse 1, 6 Robots:



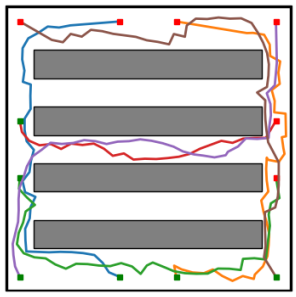
Εικόνα 7.3.41 A Path*



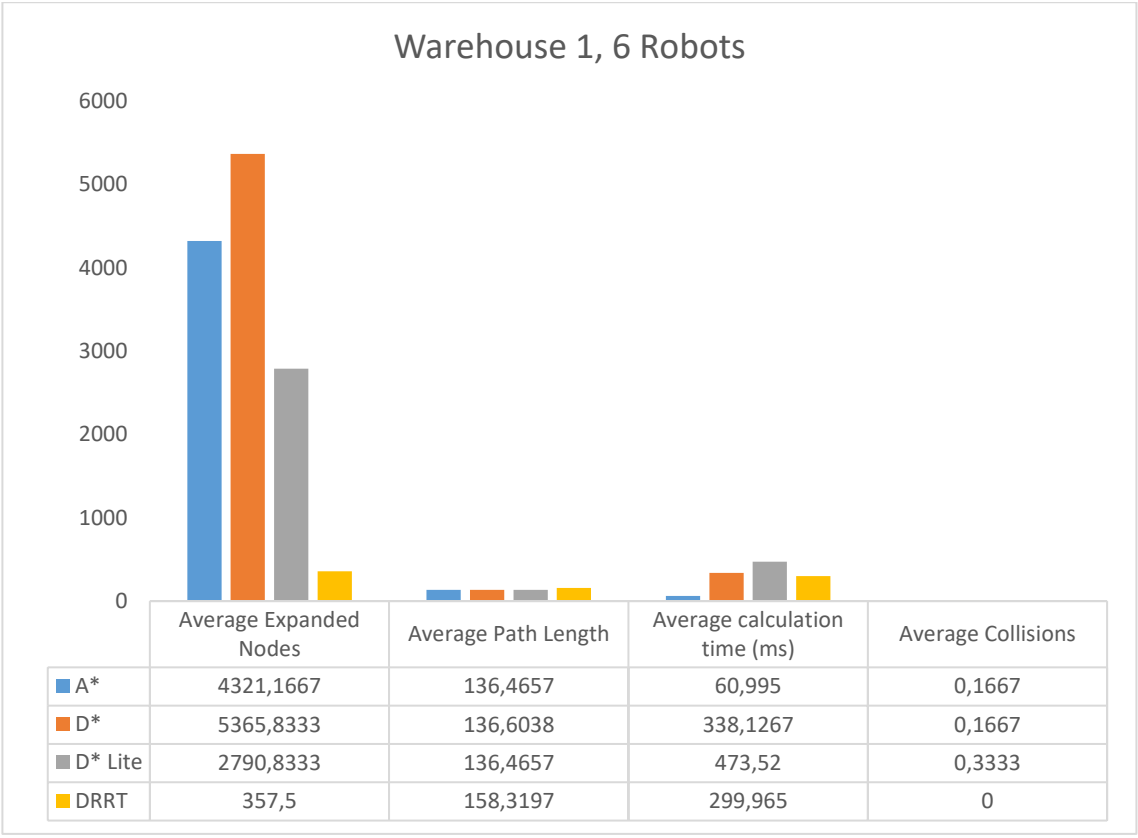
Εικόνα 7.3.42 D Path*



*Εικόνα 7.3.43 D*Lite Path*



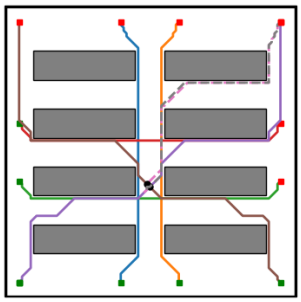
Εικόνα 7.3.44 DRRT Path



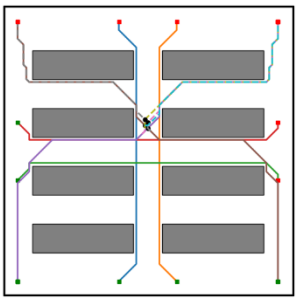
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	4321,1667	136,4657	60,995	0,1667
D*	5365,8333	136,6038	338,1267	0,1667
D* Lite	2790,8333	136,4657	473,52	0,3333
DRRT	357,5	158,3197	299,965	0

Εικόνα 7.3.45 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 1

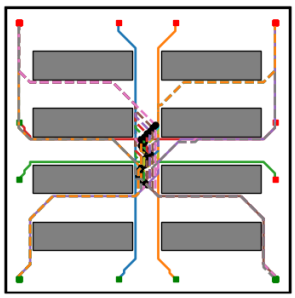
Warehouse 2, 6 Robots:



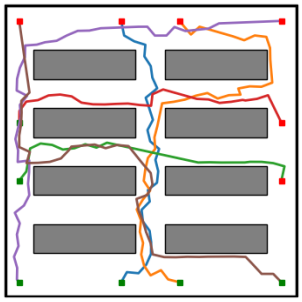
Εικόνα 7.3.46 A Path*



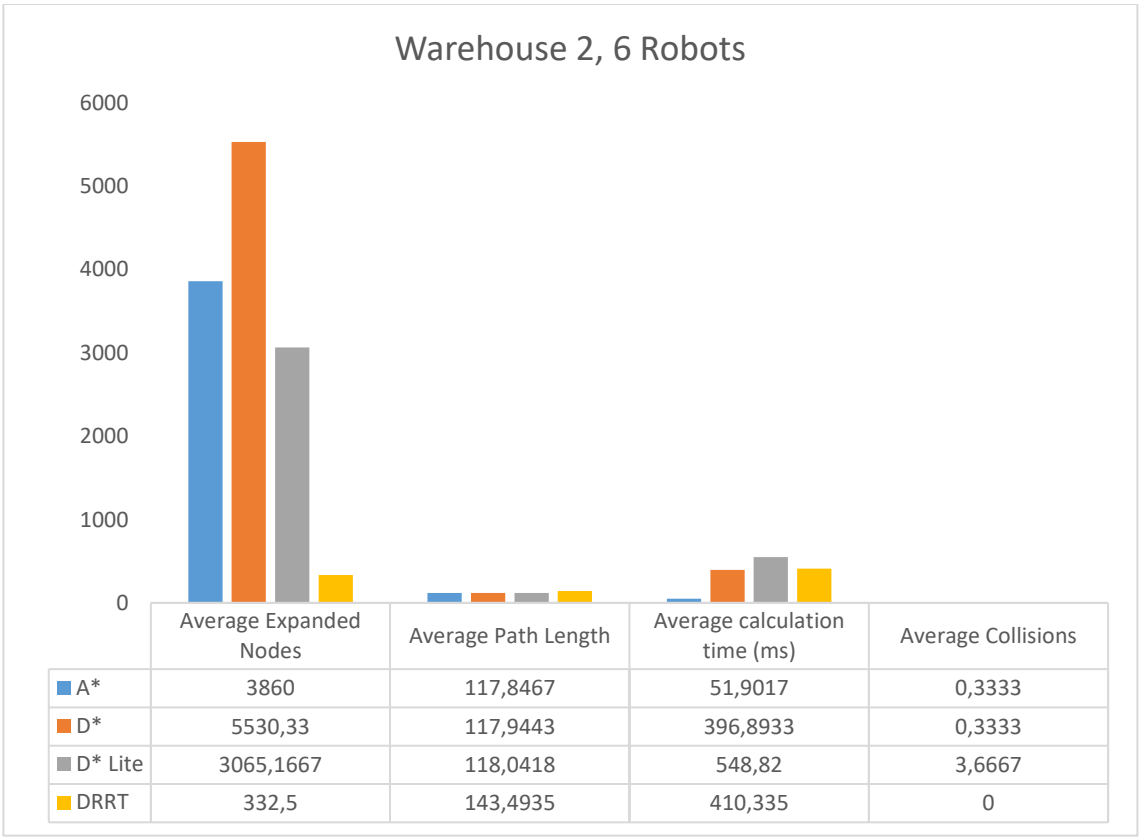
Εικόνα 7.3.47 D Path*



*Εικόνα 7.3.48 D*Lite Path*



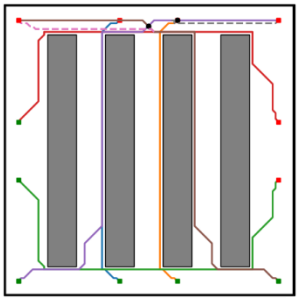
Εικόνα 7.3.49 DRRT Path



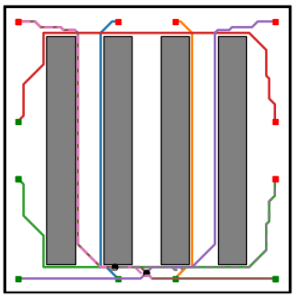
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	3860	117,8467	51,9017	0,3333
D*	5530,33	117,9443	396,8933	0,3333
D* Lite	3065,1667	118,0418	548,82	3,6667
DRRT	332,5	143,4935	410,335	0

Εικόνα 7.3.50 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 2

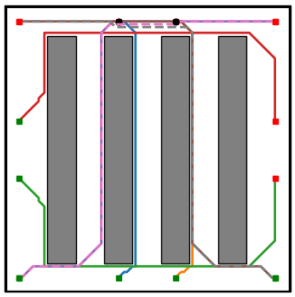
Warehouse 3, 6 Robots:



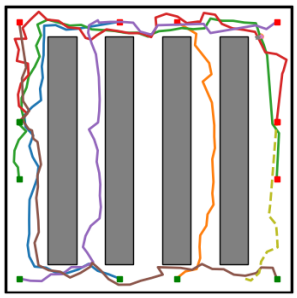
Εικόνα 7.3.51 A Path*



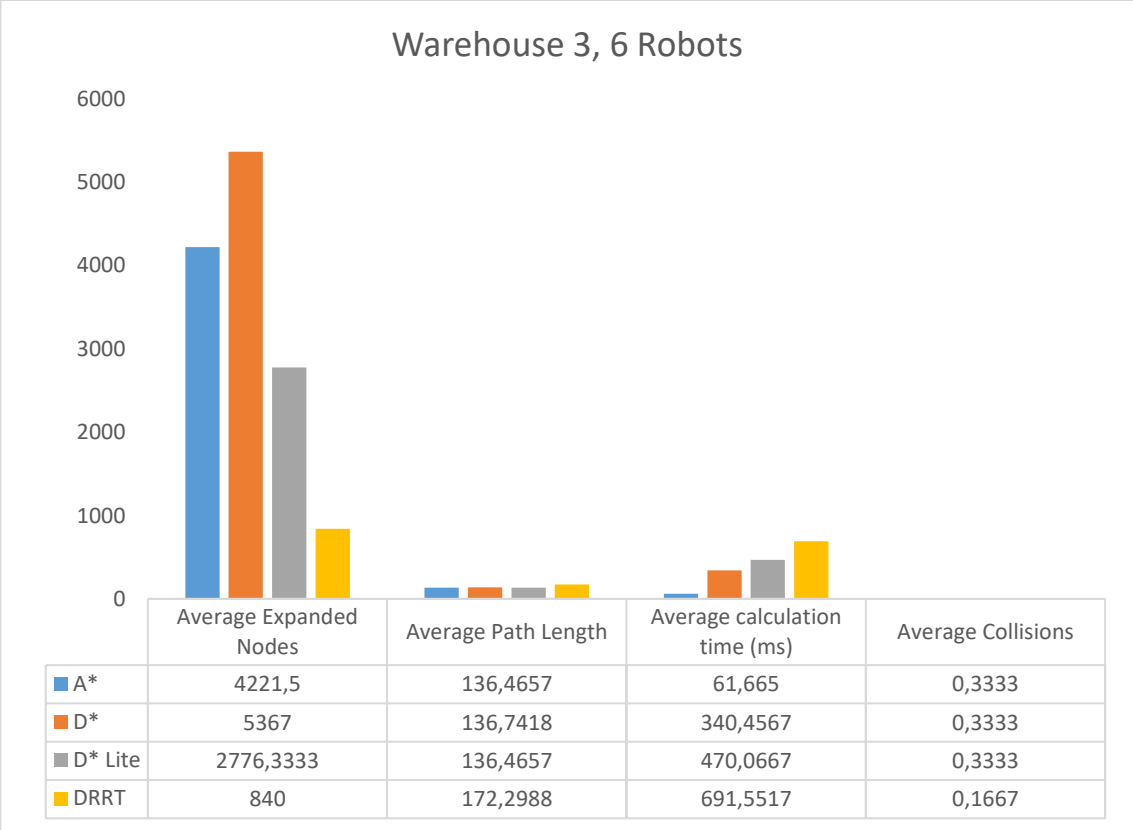
Εικόνα 7.3.52 D Path*



*Εικόνα 7.3.53 D*Lite Path*



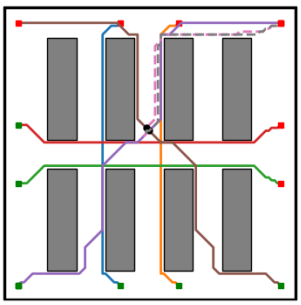
Εικόνα 7.3.54 DRRT Path



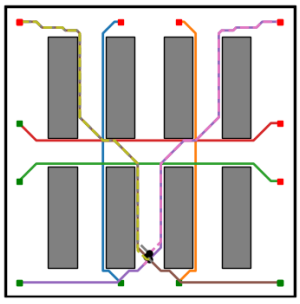
	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	4221,5	136,4657	61,665	0,3333
D*	5367	136,7418	340,4567	0,3333
D* Lite	2776,3333	136,4657	470,0667	0,3333
DRRT	840	172,2988	691,5517	0,1667

Εικόνα 7.3.55 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 3

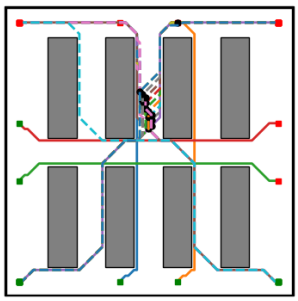
Warehouse 4, 6 Robots:



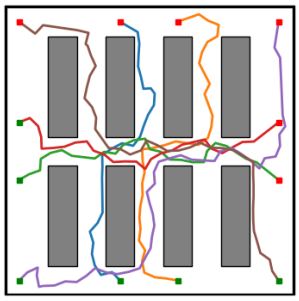
Εικόνα 7.3.56 A Path*



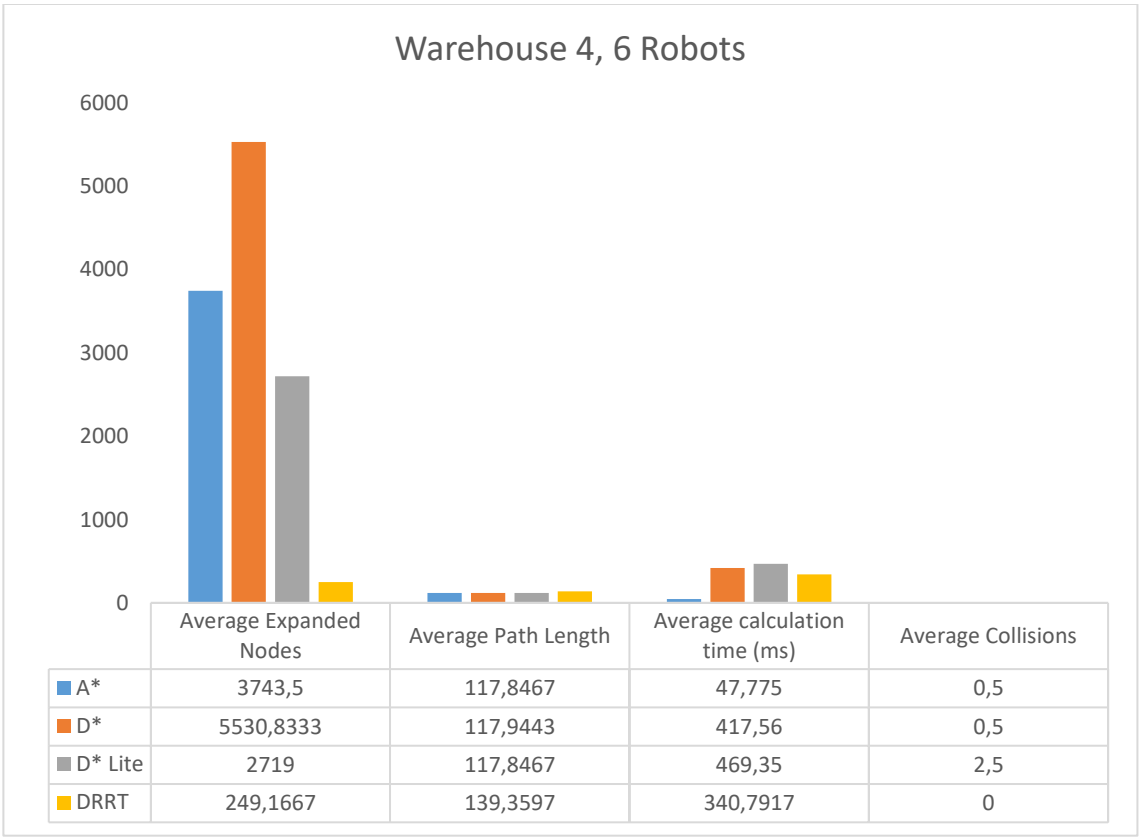
Εικόνα 7.3.57 D Path*



*Εικόνα 7.3.58 D*Lite Path*



Εικόνα 7.3.59 DRRT Path



	Average Expanded Nodes	Average Path Length	Average calculation time (ms)	Average Collisions
A*	3743,5	117,8467	47,775	0,5
D*	5530,8333	117,9443	417,56	0,5
D* Lite	2719	117,8467	469,35	2,5
DRRT	249,1667	139,3597	340,7917	0

Εικόνα 7.3.60 Συγκεντρωτικό διάγραμμα στατιστικών εκτέλεσης των αλγορίθμων, Αποθήκη 4

7.4 Ανάλυση διαγραμμάτων και στατιστικών

Σενάριο I - Δύο Ρομποτικά οχήματα:

Στο σενάριο με τα δύο ρομποτικά οχήματα την καλύτερη επίδοση σχετικά με τον χρόνο υπολογισμού του μονοπατιού εμφάνισε ο A*, ενώ το συνολικό μήκος των μονοπατιών είναι ίδιο για όλους τους αλγορίθμους της οικογένειας star (εικόνες 7.3.5, 7.3.10, 7.3.15, 7.3.20). Το μήκος αυτό είναι το βέλτιστο για όλες τις αποθήκες, παρόλο που τα μονοπάτια δεν είναι ακριβώς ίδια, όπως για παράδειγμα φαίνεται στις εικόνες 7.3.1 και 7.3.2. Το μεγαλύτερο πλήθος εξερευνημένων κόμβων εμφάνιση με διαφορά ο αλγόριθμος D* και στις τέσσερις αποθήκες, ενώ το μικρότερο ο αλγόριθμος DRRT, με εξαίρεση την αποθήκη 4 (εικόνα 7.3.19). Ο αλγόριθμος D* Lite εμφάνισε το δεύτερο κατά σειρά μικρότερο πλήθος εξερευνημένων κόμβων, με εξαίρεση την αποθήκη 4 όπως φαίνεται στην εικόνα 7.3.20 και το δεύτερο καλύτερο χρόνο υπολογισμού εκτός από την αποθήκη 1 όπου η απόδοση του ήταν η χειρότερη σε θέμα χρόνου καθώς χρειάστηκε περίπου 433 ms για να σχεδιάσει τα μονοπάτια. Ο χρόνος υπολογισμού του αλγόριθμου D* είναι ο μεγαλύτερος στις αποθήκες 1 έως 3. Όσον αφορά την αποθήκη 4 το μεγαλύτερο χρόνο υπολογισμού εμφάνισε ο αλγόριθμος DRRT επειδή είναι ο μόνος αλγόριθμος για τον οποίο στα αρχικά σχεδιασμένα μονοπάτια βρέθηκε μια πιθανή σύγκρουση όπως φαίνεται στην εικόνα 7.3.20. Η πιθανή αυτή η σύγκρουση δημιουργεί την ανάγκη για επανασχεδιασμό του μονοπατιού και κατ' επέκταση αυξάνει το συνολικό χρόνο εκτέλεσης του αλγορίθμου. Η εμφάνιση μόνο μιας πιθανής σύγκρουσης οφείλεται στο γεγονός ότι στο σενάριο αυτό σχεδιάζονται μονοπάτια για 2 ρομποτικά οχήματα, αριθμός που περιορίζει την πιθανότητα εμφάνισης περισσότερων συγκρούσεων. Όπως θα φανεί στα αποτελέσματα από τα παρακάτω σενάρια η αύξηση του πλήθους των ρομποτικών οχημάτων αυξάνει και την πιθανότητα εμφάνισης συγκρούσεων, όπως είναι λογικό. Τέλος ο DRRT είχε σε όλες τις αποθήκες το μεγαλύτερο συνολικό μήκος μονοπατιών με σημαντική η διαφορά από αυτά των αλγορίθμων star, με εξαίρεση την αποθήκη 1 όπου οι τιμές έχουν μικρή απόκλιση όπως φαίνεται στην εικόνα 7.3.5.

Σενάριο II – Τέσσερα Ρομποτικά οχήματα:

Σύμφωνα με τα αποτελέσματα της εκτέλεσης του αλγορίθμου εύρεσης και επίλυσης συγκρούσεων για το σενάριο με τα τέσσερα ρομποτικά οχήματα προκύπτει πάλι, ότι ο αλγόριθμος A* έχει με διαφορά το μικρότερο χρόνο εκτέλεσης. Επίσης ομοίως με το προηγούμενο σενάριο το συνολικό μήκος των τελικών μονοπατιών είναι ίδιο για τους αλγορίθμους star σε κάθε αποθήκη. Αυτό συμβαίνει επειδή και σε αυτή την περίπτωση στα αρχικά σχεδιασμένα μονοπάτια από τους αλγόριθμους αυτούς δεν εντοπίστηκαν πιθανές συγκρούσεις. Όσο αφορά το πλήθος των εξερευνημένων κόμβων, την εκτενέστερη αναζήτηση πραγματοποίησε και πάλι ο αλγόριθμος D* όπως προκύπτει από τις εικόνες 7.3.25, 7.3.30, 7.3.35, 7.3.40 . Με εξαίρεση την αποθήκη 4 (εικόνα 7.3.39), ο DRRT είχε το μικρότερο πλήθος εξερευνημένων κόμβων. Στην αποθήκη 4 λόγω των πολλών συγκρούσεων που εντοπίστηκαν, συγκεκριμένα τρεις στον αριθμό (εικόνα 7.3.40), το μέγεθος της αναζήτησης αυξήθηκε σημαντικά. Έτσι σε αυτή την περίπτωση ο αλγόριθμος D* Lite αναδείχθηκε ως εκείνος με τη μικρότερη αναζήτηση καθώς και με τον καλύτερο χρόνο υπολογισμού, μετά τον A* φυσικά. Ο D* Lite είχε το δεύτερο καλύτερο χρόνο υπολογισμού στις αποθήκες 2, 3 και 4. Ενδιαφέρον έχει το μονοπάτι που σχεδιάστηκε από τον DRRT για την αποθήκη 1 (εικόνα 7.3.24), όπου παρά τον εντοπισμό 4 πιθανών συγκρούσεων το πλήθος των εξερευνημένων κόμβων είναι το χαμηλότερο και

ο χρόνος υπολογισμού ο δεύτερος καλύτερος. Παρόλα αυτά ακόμα είχε τα μεγαλύτερα τελικά μονοπάτια και τον χειρότερο χρόνο υπολογισμού σε όλες τις άλλες αποθήκες με αποκορύφωμα την αποθήκη 4 όπου, όπως φαίνεται στην εικόνα 7.3.40, ο χρόνος υπολογισμού που χρειάστηκε ήταν περίπου 100 φορές μεγαλύτερος από εκείνο του αλγορίθμου A*.

Σενάριο ΙΙΙ - Έξι ρομποτικά οχήματα:

Η πρώτη παρατήρηση στα αποτελέσματα, η οποία είναι και βασική διαφορά με τα προηγούμενα σενάρια, σχετίζεται με το πλήθος των πιθανών συγκρούσεων που εντοπίστηκαν στα σχεδιασμένα μονοπάτια. Πιο συγκεκριμένα όλοι οι αλγόριθμοι star εμφάνισαν πιθανές συγκρούσεις στα μονοπάτια τους, με τον D* Lite να έχει τις περισσότερες. Με εξαίρεση την αποθήκη 3 (εικόνα 7.3.53) όπου DRRT είχε το μεγαλύτερο χρόνο υπολογισμού, ο αλγόριθμος D* Lite είχε την χειρότερη απόδοση σε θέμα χρόνου, όπως φαίνεται στις εικόνες 7.3.45, 7.3.50, 7.3.60, γεγονός το οποίο οφείλεται στις συγκρούσεις που αναφέρονται παραπάνω, ενώ ο A* συνεχίζει να διατηρεί την πρώτη θέση με το χαμηλότερο χρόνο υπολογισμού σε όλες αποθήκες. Στο σενάριο με τα έξι ρομποτικά οχήματα η απόδοση του αλγορίθμου DRRT ξεχώρισε, με μόνο μια σύγκρουση στην αποθήκη 3 (εικόνα 7.3.55). Οι αλγόριθμοι A* και D* εμφάνισαν παρόμοια συμπεριφορά ως προς το πλήθος των συγκρούσεων που εντοπίστηκαν στα μονοπάτια τους. Όσο αφορά την έκταση της αναζήτησης, όπως και στα προηγούμενα σενάρια, τους περισσότερους κόμβους εξερεύνησε κατά την εκτέλεση του ο αλγόριθμος D* σε όλες τις αποθήκες. Σε αυτό το σημείο γίνεται εύκολα αντιληπτό ότι ο τρόπος λειτουργίας του συγκεκριμένου αλγόριθμου θα έχει πάντα ως αποτέλεσμα τη μεγαλύτερη εξερεύνηση του χώρου. Ωστόσο αυτό δεν μεταφράζεται πάντα στον υψηλότερο χρόνο υπολογισμού, εφόσον στις αποθήκες 2 και 3, ο D* χρειάστηκε τον δεύτερο κατά σειρά μικρότερο χρόνο υπολογισμού. Ο αλγόριθμος DRRT είχε το μικρότερο πλήθος εξερευνημένων κόμβων και σε αυτό το σενάριο. Τέλος όσον αφορά τα μήκη των τελικών μονοπατιών οι αλγόριθμοι star έδειξαν σχεδόν την ίδια συμπεριφορά ενώ ο αλγόριθμος DRRT δημιούργησε τα μεγαλύτερα μονοπάτια και στις 4 αποθήκες, χωρίς όμως να έχουν σημαντική απόκλιση από τα μονοπάτια των άλλων αλγορίθμων.

7.5 Συνδυαστική αξιολόγηση

Εξετάζοντας τα διαγράμματα και τα στατιστικά από όλα τα παραπάνω σενάρια, παρατηρούμε ότι ο αλγόριθμος A* υπολογίζει πάντα το βέλτιστο μονοπάτι, ενώ τα μήκη μονοπατιού που παράγουν οι αλγόριθμοι D* και D* Lite κυμαίνονται κοντά στο βέλτιστο και σε πολλές περιπτώσεις το πετυχαίνουν. Ο αλγόριθμος DRRT δημιούργησε σε όλα τα σενάρια το μεγαλύτερο μονοπάτι. Το γεγονός αυτό οφείλεται στην τυχαία δειγματοληψία των κόμβων εντός του περιβάλλοντος.

Όσο αφορά το πλήθος των εξερευνημένων κόμβων, την καλύτερη απόδοση εμφανίζει ο DRRT το οποίο οφείλεται στον τρόπο λειτουργίας του. Επομένως έχει νόημα να συγκρίνουμε τους αλγορίθμους της οικογένειας star μεταξύ τους. Από τη σύγκριση αυτή ξεχωρίζει ο αλγόριθμος D* Lite, ο οποίος πραγματοποίησε την λιγότερη εξερεύνηση , ενώ ο αμέσως επόμενος είναι ο A* ο οποίος απέδωσε ικανοποιητικά παρόλο που στερείται την ικανότητα δυναμικού επανασχεδιασμού του παραγόμενου μονοπατιού. Τη μεγαλύτερη εξερεύνηση πραγματοποίησε ο D*.

Το επόμενο στατιστικό που θα εξετάσουμε είναι το πλήθος των πιθανών συγκρούσεων που εντοπίστηκαν. Ο λόγος που δίνουμε μεγαλύτερη έμφαση σε αυτή την τιμή, σε σχέση με τον χρόνο υπολογισμού που απαιτήθηκε, είναι επειδή, τόσο στην περίπτωση του A* όπου ο αλγόριθμος εκτελείται εκ νέου με νέα αφετηρία, όσο και στην περίπτωση των δυναμικών αλγορίθμων, το πλήθος των επανασχεδιασμών καθορίζει άμεσα τον απαιτούμενο χρόνο υπολογισμού. Σε όλα τα σενάρια και περιβάλλοντα οι αλγόριθμοι παρουσίασαν παρόμοια αποτελέσματα, με εξαίρεση το σενάριο με τα έξι ρομποτικά οχήματα στην αποθήκη 2. Σε αυτό το σενάριο, παρατηρήθηκαν 22 πιθανές συγκρούσεις, στα μονοπάτια που δημιουργήθηκαν από τον αλγόριθμο D* Lite, ο οποίος προφανώς είχε και τον μεγαλύτερο χρόνο υπολογισμού σε αυτό το πείραμα.

Σε όλα τα σενάρια οι χρόνοι εκτέλεσης των δυναμικών αλγορίθμων έχουν παραπλήσιες τιμές με εξαίρεση πολύ συγκεκριμένες περιπτώσεις. Παρόλα αυτά τον μικρότερο χρόνο εκτέλεσης εμφάνισε με διαφορά ο αλγόριθμος A*. Αυτό οφείλεται στο τέχνασμα που υλοποιήθηκε το οποίο εμφανίζει παρόμοιο τρόπο λειτουργίας με τον αλγόριθμο D* Lite. Ο A* ωστόσο δεν προσπαθεί να επιδιορθώσει το μονοπάτι και ενδιαφέρεται μόνο για την επίτευξη του τελικού κόμβου. Σημαντικό ρόλο έχει επίσης η συνθήκη επιλογής του οχήματος που θα επαναπροσδιορίσει το μονοπάτι του, σύμφωνα με την οποία επιλέγεται το όχημα που είναι πιο κοντά στον στόχο του.

Τέλος όσο αφορά την επιλογή διάταξης αποθήκης καταλήγουμε στο συμπέρασμα πως δεν υπάρχει ιδανική επιλογή, εφόσον κάθε διάταξη επηρεάζει σημαντικά τα μήκη των τελικών μονοπατιών καθώς επίσης και το πλήθος των πιθανών συγκρούσεων. Ωστόσο ιδιαίτερο ενδιαφέρον εμφανίζει η αποθήκη 4 (Εικόνα 7.2.4) στο σενάριο με τα έξι ρομποτικά οχήματα, η οποία κατά μέσο όρο έχει τα συντομότερα τελικά μονοπάτια με ικανοποιητικούς χρόνους υπολογισμού, σε σύγκριση πάντα με τις υπόλοιπες αποθήκες.

ΚΕΦΑΛΑΙΟ 8 Συμπεράσματα

Σύμφωνα με τα αποτελέσματα που προέκυψαν από την εκτέλεση του αλγορίθμου εύρεσης και επίλυσης πιθανών συγκρούσεων για διάφορα σενάρια του πειράματος, σε διαφορετικές αποθήκες όπως αναλυτικά περιγράφονται στο κεφάλαιο 7, συμπεραίνουμε αρχικά ότι όλοι οι αλγόριθμοι που χρησιμοποιήθηκαν (A*, D*, D* Lite, DRRT) κατάφεραν να σχεδιάσουν τελικό μονοπάτι για όλα τα ρομποτικά οχήματα.

Από τα στατιστικά που συγκεντρώθηκαν, διαπιστώθηκαν τα πλεονεκτήματα και τα μειονεκτήματα εκάστου αλγορίθμου. Περιγράφοντας συνοπτικά, ο αλγόριθμος A* υπερέχει των άλλων, όσο αφορά το χρόνο εύρεσης μονοπατιού. Η απόδοση σχετικά με το πλήθος πιθανών συγκρούσεων είναι χαμηλή για τον D* Lite, ενώ οι υπόλοιποι αλγόριθμοι παρουσιάζουν παρόμοια εικόνα. Για το συνολικό μήκος των τελικών μονοπατιών παρατηρούμε παρόμοιες τιμές για όλους τους αλγόριθμους με εξαίρεση τον DRRT ο οποίος παράγει μεγαλύτερα μονοπάτια, χωρίς όμως σημαντική απόκλιση από τους άλλους.

Εν κατακλείδι είναι γεγονός ότι τα αυτόνομα ρομποτικά οχήματα (AMR) είναι ιδιαίτερα σημαντικά για την αυτοματοποίηση των αποθηκών ιδίως όταν το περιβάλλον είναι δυναμικό. Η δυνατότητα τους να προσαρμόζουν δυναμικά το μονοπάτι τους με χρήση αλγορίθμων όπως αυτοί που εξετάστηκαν στην παρούσα εργασία συμβάλει σημαντικά στην αυτόνομη πλοήγηση τους.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. S. A. Soundattikar, V. R. Naik and C. V. Adake, "Component Handing Automated Guided Vehicle–A Cyber Physical System Case Study," *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India, 2022, pp. 1-6, doi: 10.1109/ICAECT54875.2022.9808022

2. S. G. M. Hossain, H. Jamil, M. Y. Ali and M. Zahurul Haq, "Automated guided vehicles for industrial logistics - Development of intelligent prototypes using appropriate technology," *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Singapore, 2010, pp. 237-241, doi: 10.1109/ICCAE.2010.5451466.

3. Tikwayo, Lihle N., and Tebello N. D. Mathaba. 2023. "Applications of Industry 4.0 Technologies in Warehouse Management: A Systematic Literature Review" *Logistics* 7, no. 2: 24. https://doi.org/10.3390/logistics7020024.

4. J. Fernandes, F.J.G. Silva, R.D.S.G. Campilho, G.F.L. Pinto, A. Baptista, Intralogistics and industry 4.0: designing a novel shuttle with picking system, Procedia Manufacturing, Volume 38, 2019, Pages 1801-1832, ISSN 2351-9789, https://doi.org/10.1016/j.promfg.2020.01.078.

5. R. Inam *et al*., "Risk Assessment for Human-Robot Collaboration in an automated warehouse scenario," *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Turin, Italy, 2018, pp. 743-751, doi: 10.1109/ETFA.2018.8502466.

6. Anbalagan Loganathan, Nur Syazreen Ahmad, A systematic review on recent advances in autonomous mobile robot navigation, Engineering Science and Technology, an International Journal, volume 40, 2023, 101343, ISSN 2215-0986,

https://doi.org/10.1016/j.jestch.2023.101343.

7. P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, July 1968, doi: 10.1109/TSSC.1968.300136.

8. Yan, Yumeng. (2023). Research on the A Star Algorithm for Finding Shortest Path. Highlights in Science, Engineering and Technology. 46. 154-161. 10.54097/hset.v46i.7697.

9. Rina Dechter and Judea Pearl. 1985. Generalized best-first search strategies and the optimality of A*. J. ACM 32, 3 (July 1985), 505–536. https://doi.org/10.1145/3828.3830.

10. Li, Pei & Xie, Cong & Guo, Xin. (2023). Multi-robots dynamic path planning based on improved A* algorithm. 10.21203/rs.3.rs-3421061/v1.

11. A. Stentz, "Optimal and efficient path planning for partially-known environments," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, USA, 1994, pp. 3310-3317 vol.4, doi: 10.1109/ROBOT.1994.351061.

12. Koenig, Sven & Likhachev, Maxim & Liu, Yaxin & Furcy, David. (2004). Incremental Heuristic Search in AI.. AI Magazine. 25. 99-112.+

13. LaValle, Steven M.. “Rapidly-exploring random trees : a new tool for path planning.” The annual research report (1998): n. pag.

14. Koenig, Sven & Likhachev, Maxim. (2005). Fast replanning for navigation in unknown terrain. Robotics, IEEE Transactions on. 21. 354 - 363. 10.1109/TRO.2004.838026.

15. D. Ferguson, N. Kalra and A. Stentz, "Replanning with RRTs," *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, FL, USA, 2006, pp. 1243-1248, doi: 10.1109/ROBOT.2006.1641879.

16. Sven Koenig, Maxim Likhachev, David Furcy, Lifelong Planning A∗, Artificial Intelligence, Volume 155, Issues 1–2, 2004, Pages 93-146, ISSN 0004-3702, https://doi.org/10.1016/j.artint.2003.12.001.

17. J. -H. Oh, J. -H. Park and J. -T. Lim, "Centralized Decoupled Path Planning Algorithm for Multiple Robots Using the Temporary Goal Configurations," *2011 Third International Conference on Computational Intelligence, Modelling & Simulation*, Langkawi, Malaysia, 2011, pp. 206-209, doi: 10.1109/CIMSim.2011.43.

18. W. Hong, Y. Tian and Y. Xu, "The Research of Dynamic Path Planning for Centralized Vehicle Navigation," 2007 IEEE International Conference on Automation and Logistics, Jinan, China, 2007, pp. 1198-1202, doi: 10.1109/ICAL.2007.4338751.

19. A. Bolu and Ö. Korçak, "Path Planning for Multiple Mobile Robots in Smart Warehouse," 2019 7th International Conference on Control, Mechatronics and Automation (ICCMA), Delft, Netherlands, 2019, pp. 144-150, doi: 10.1109/ICCMA46720.2019.8988635.

20. K.J.C. Fransen, J.A.W.M. van Eekelen, A. Pogromsky, M.A.A. Boon, I.J.B.F. Adan, A dynamic path planning approach for dense, large, grid-based automated guided vehicle systems, Computers &

Operations Research, Volume 123, 2020, 105046, ISSN 0305-0548, https://doi.org/10.1016/j.cor.2020.105046.

21. M. De Ryck, M. Versteyhe, K. Shariatmadar, Resource management in decentralized industrial Automated Guided Vehicle systems, Journal of Manufacturing Systems, Volume 54, 2020, Pages 204-214, ISSN 0278-6125, https://doi.org/10.1016/j.jmsy.2019.11.003.

22. M. De Ryck, M. Versteyhe, F. Debrouwere, Automated guided vehicle systems, state-of-the-art control algorithms and techniques, Journal of Manufacturing Systems, Volume 54, 2020, Pages 152-173, ISSN 0278-6125, https://doi.org/10.1016/j.jmsy.2019.12.002.

23. Ivica Draganjac, Tamara Petrović, Damjan Miklić, Zdenko Kovačić, Juraj Oršulić, Highly-scalable traffic management of autonomous industrial transportation systems, Robotics and Computer-Integrated Manufacturing, Volume 63, 2020, 101915, ISSN 0736-5845, https://doi.org/10.1016/j.rcim.2019.101915.

24. A. Krnjak, I. Draganjac, S. Bogdan, T. Petrović, D. Miklić and Z. Kovačić, "Decentralized control of free ranging AGVs in warehouse environments," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015, pp. 2034-2041, doi: 10.1109/ICRA.2015.7139465.

25. Parker, Lynne. (2009). Multiple Mobile Robot Teams, Path Planning and Motion Coordination in. 10.1007/978-0-387-30440-3_344.

26. Zhou, Luowei, et al. “A Balanced Heuristic Mechanism for Multirobot Task Allocation of Intelligent Warehouses.” Mathematical Problems in Engineering, vol. 2014, 2014, pp. 1–10, https://doi.org/10.1155/2014/380480.

27. Giuseppe Fragapane, René de Koster, Fabio Sgarbossa, Jan Ola Strandhagen, Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda, European Journal of Operational Research, Volume 294, Issue 2, 2021, Pages 405-426, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2021.01.019.

ΔΙΑΔΙΚΤΥΑΚΕΣ ΠΗΓΕΣ

1. https://www.posital.com/pt/industrias/material-handling/as-rs-system/as_rs_system.php
2. https://control.com/technical-articles/an-overview-of-automatic-guided-vehicles/
3. https://www.designworldonline.com/slam-technology-improves-amr-autonomy/
4. https://www.mathworks.com/discovery/slam.html
5. https://en.wikipedia.org/wiki/Motion_planning
6. https://mat.uab.cat/~alseda/MasterOpt/AStar-Algorithm.pdf
7. https://en.wikipedia.org/wiki/Linear_interpolation

ΑΝΑΛΥΤΙΚΑ ΣΤΑΤΙΣΤΙΚΑ

Σενάριο Ι – Δύο Ρομποτικά οχήματα

A:*

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4144	141,456	63,18	0
Robot 2	4023	141,456	84,22	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	600	94,971	19,21	0
Robot 2	602	94,971	9	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1036	97,314	13,99	0
Robot 2	1036	97,314	12,16	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1127	97,314	15,9	0
Robot 2	1127	97,314	10,21	0

D:*

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	5495	141,456	361,81	0
Robot 2	5493	141,456	342,94	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4875	94,971	385,16	0
Robot 2	4859	94,971	366,36	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4604	97,314	348,74	0
Robot 2	4582	97,314	330,06	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	5331	97,314	449,24	0
Robot 2	5304	97,314	434,31	0

D Lite:*

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	2651	141,456	441,46	0
Robot 2	2580	141,456	425,06	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	570	94,971	99,5	0
Robot 2	570	94,971	105,39	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1001	97,314	175,79	0
Robot 2	1001	97,314	169,73	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1088	97,314	191,39	0
Robot 2	1088	97,314	204,5	0

DRRT:

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	416	161,827	298,68	0
Robot 2	648	176,946	539,48	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	162	104,166	203,92	0
Robot 2	93	107,441	98,46	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	424	149,725	366,11	0
Robot 2	210	151,626	216,45	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	859	159,707	1075,07	0
Robot 2	1457	201,34	1008,24	1

Σενάριο ΙΙ – Τέσσερα Ρομποτικά οχήματα

A:*

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4144	141,456	47,11	0
Robot 2	4023	141,456	69,2	0
Robot 3	1039	97,314	21,49	0
Robot 4	1039	97,314	17,67	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	600	94,971	7,98	0
Robot 2	602	94,971	8,82	0
Robot 3	1130	97,314	14,83	0
Robot 4	1130	97,314	13,16	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1036	97,314	13,96	0
Robot 2	1036	97,314	12,97	0
Robot 3	4144	141,456	67,19	0
Robot 4	4029	141,456	50,33	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1127	97,314	28,3	0
Robot 2	1127	97,314	10,45	0
Robot 3	602	94,971	7,98	0
Robot 4	602	94,971	1,2	0

D:*

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	5495	141,456	361,52	0
Robot 2	5493	141,456	340,54	0
Robot 3	4606	97,314	318,8	0
Robot 4	4582	97,314	316,13	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4875	94,971	373,84	0
Robot 2	4859	94,971	358,53	0
Robot 3	5332	97,314	422,45	0
Robot 4	5304	97,314	418,19	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4604	97,314	343,74	0
Robot 2	4582	97,314	317,53	0
Robot 3	5495	141,456	340,04	0
Robot 4	5493	141,456	332,41	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	5331	97,314	443,94	0
Robot 2	5304	97,314	415,74	0
Robot 3	4875	94,971	357,24	0

Robot 4	4859	94,971	349,5	0

D Lite:*

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	2651	141,456	465,74	0
Robot 2	2580	141,456	461,72	0
Robot 3	1001	97,314	183,73	0
Robot 4	1001	97,314	184,3	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	570	94,971	90,77	0
Robot 2	570	94,971	98,73	0
Robot 3	1088	97,314	189,37	0
Robot 4	1088	97,314	200,48	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1001	97,314	173,27	0
Robot 2	1001	97,314	172,93	0
Robot 3	2651	141,456	463,22	0
Robot 4	2580	141,456	431,21	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1088	97,314	194,5	0
Robot 2	1088	97,314	196,74	0
Robot 3	570	94,971	105,46	0
Robot 4	570	94,971	113,16	0

DRRT:

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	449	207,329	360,88	0
Robot 2	383	169,311	343,2	0
Robot 3	272	188,952	289,13	4
Robot 4	225	156,395	240,12	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	120	112,456	155,63	0
Robot 2	199	107,859	281,32	0
Robot 3	602	161,949	910,41	0
Robot 4	353	158,041	420,14	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	678	199,582	601,86	0
Robot 2	1192	155,191	632,38	1
Robot 3	225	166,149	216,23	0
Robot 4	323	167,412	265,29	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	350	165,597	500,87	0
Robot 2	318	141,687	536,55	0
Robot 3	3342	105,16	3186,93	3
Robot 4	521	123,852	777,98	0

Σενάριο ΙΙΙ – Έξι Ρομποτικά οχήματα

A:*

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisionss
Robot 1	4144	141,456	55,05	0
Robot 2	4023	141,456	68,22	0
Robot 3	1039	97,314	24,08	0
Robot 4	1039	97,314	12,41	0
Robot 5	8317	170,627	102,25	1
Robot 6	7365	170,627	103,96	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	600	94,971	16,82	0
Robot 2	602	94,971	8	0
Robot 3	1130	97,314	13,74	0
Robot 4	1130	97,314	13,43	0
Robot 5	12435	161,255	161,43	2
Robot 6	7263	161,255	97,99	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1036	97,314	26,26	0
Robot 2	1036	97,314	13,61	0
Robot 3	4144	141,456	60,16	0
Robot 4	4029	141,456	52,95	0
Robot 5	7555	170,627	99,59	1
Robot 6	7529	170,627	117,42	1

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1127	97,314	25,56	0
Robot 2	1127	97,314	11,89	0
Robot 3	602	94,971	7,01	0

Robot 4	602	94,971	1,12	0
Robot 5	10203	161,255	136,5	2
Robot 6	8800	161,255	104,57	1

D:*

Warehouse 1	Total Visited Nodes	Path Length	Execution time (ms)	Collisionss
Robot 1	5495	141,456	347,02	0
Robot 2	5493	141,456	346,98	0
Robot 3	4606	97,314	326,78	0
Robot 4	4582	97,314	308,96	0
Robot 5	6008	171,456	360,96	1
Robot 6	6011	170,627	338,06	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4875	94,971	380,13	0
Robot 2	4859	94,971	366,33	0
Robot 3	5332	97,314	428,09	0
Robot 4	5304	97,314	401,72	0
Robot 5	6406	161,841	404,04	2
Robot 6	6406	161,255	401,05	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	4604	97,314	340,04	0
Robot 2	4582	97,314	324,75	0
Robot 3	5503	143,113	338,46	1
Robot 4	5493	141,456	344,78	0
Robot 5	6005	170,627	345,89	0
Robot 6	6015	170,627	348,82	1

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	5331	97,314	438,93	0
Robot 2	5304	97,314	432,14	0
Robot 3	4875	94,971	400,01	0
Robot 4	4859	94,971	385,52	0
Robot 5	6402	161,255	414,55	1
Robot 6	6414	161,841	434,21	2

D Lite:*

Warehouse 1	Total Visited Nodes	Path Length	Execution time (ms)	Collisionss

Robot 1	2651	141,456	447,63	0
Robot 2	2580	141,456	439,41	0
Robot 3	1001	97,314	186,14	0
Robot 4	1001	97,314	180,2	0
Robot 5	5006	170,627	838,57	2
Robot 6	4506	170,627	749,17	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	570	94,971	88,17	0
Robot 2	570	94,971	106,24	0
Robot 3	1088	97,314	210,36	0
Robot 4	1088	97,314	183,1	0
Robot 5	6167	161,255	964,97	9
Robot 6	8908	162,426	1740,08	13

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1001	97,314	179,77	0
Robot 2	1001	97,314	166,5	0
Robot 3	2651	141,456	454,24	0
Robot 4	2580	141,456	429,09	0
Robot 5	4655	170,627	785,13	1
Robot 6	4770	170,627	805,67	1

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	1088	97,314	179,1	0
Robot 2	1088	97,314	188,21	0
Robot 3	570	94,971	90,65	0
Robot 4	570	94,971	102,83	0
Robot 5	6616	161,255	1111,01	9
Robot 6	6382	161,255	1144,3	6

DRRT:

Warehouse 1	Expanded Nodes	Path Length	Execution time (ms)	Collisionss
Robot 1	457	152,881	363,53	0
Robot 2	599	170,41	484,04	0
Robot 3	325	156,876	317,54	0
Robot 4	119	102,978	84,15	0
Robot 5	378	174,939	316,92	0
Robot 6	267	191,834	233,61	0

Warehouse 2	Expanded Nodes	Path Length	Execution time (ms)	Collisions

Robot 1	562	110,278	738,49	0
Robot 2	273	168,39	327,61	0
Robot 3	451	106,886	536,31	0
Robot 4	124	107,437	174,29	0
Robot 5	252	185,6	291,53	0
Robot 6	333	182,37	393,78	0

Warehouse 3	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	165	149,104	176,27	0
Robot 2	318	104,108	273,86	0
Robot 3	378	197,031	290,75	0
Robot 4	3376	204,023	2668,75	1
Robot 5	270	189,906	218,36	0
Robot 6	533	189,621	521,32	0

Warehouse 4	Expanded Nodes	Path Length	Execution time (ms)	Collisions
Robot 1	554	115,949	792,05	0
Robot 2	167	139,647	253,17	0
Robot 3	122	109,709	170,91	0
Robot 4	229	110,847	276,48	0
Robot 5	213	188,602	292,2	0
Robot 6	210	171,404	259,94	0