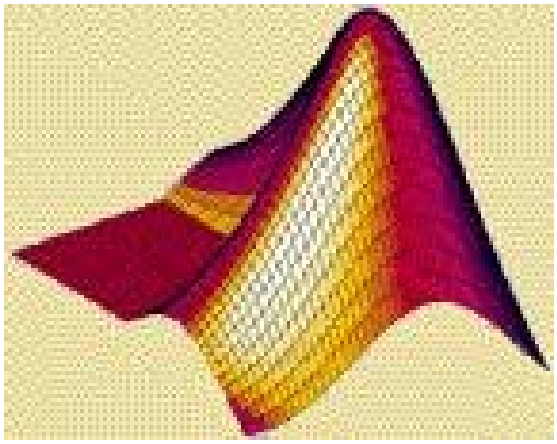
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων : Η. Κοσματόπουλος

‘‘ Τεχνολογίες Αυτόματων Τερματικών Σταθμών Σε Λιμάνια: *Προσομοίωση & Βελτιστοποίηση* ’’



Εμμανουήλ Ν. Ειρήνη
Α.Μ. 1999010022

ΧΑΝΙΑ, 2005

η εργασία αυτή είναι αφιερωμένη στον πατέρα μου,

Νίκο.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΡΟΛΟΓΟΣ

Η συγκεκριμένη εργασία αναφέρεται στην αυτοματοποίηση των τερματικών σταθμών λιμανιών για τη διαχείριση containers. Είναι γεγονός, ότι ολοένα και περισσότερο ποσοστό των θαλάσσιων μεταφερόμενων φορτίων, παγκοσμίως, μεταφέρεται με containers. Συνεπώς, οι μελλοντικές απαιτήσεις καθιστούν απαραίτητη την αυτοματοποίηση των τερματικών σταθμών, ώστε να αυξηθεί ικανοποιητικά η απόδοσή τους.

Αρχικά, πραγματοποιείται μια ανασκόπηση των τεχνολογιών, που υπάρχουν για τη διεκπεραίωση των εργασιών σε τερματικούς σταθμούς φόρτωσης/ εκφόρτωσης, περιγράφονται αυτόματες τεχνολογίες όπως είναι τα AGVs (Automated Guided Vehicles) και στη συνέχεια αναπτύσσεται ένα ρεαλιστικό μοντέλο προσομοίωσης ενός αυτοματοποιημένου τερματικού σταθμού, με τη βοήθεια του οποίου εφαρμόζουμε και αξιολογούμε απλούς αλγόριθμους ανάθεσης. Ένας από αυτούς τους αλγόριθμους είναι ο αλγόριθμος τυχαίας ανάθεσης. Επίσης, με βάση τη διαίσθηση ελέγχουμε και αλγόριθμους βασισμένους στη χρονική απόσταση των οχημάτων. Οι αλγόριθμοι αναπτύσσονται σε Matlab και οι κώδικες όλων των σεναρίων προσομοίωσης του μοντέλου του τερματικού σταθμού παρατίθενται στο *παράρτημα*.

ΕΙΣΑΓΩΓΗ

Η διακίνηση αγαθών με τη χρήση containers, πρωτοεμφανίστηκε στις αρχές του 1960. Τα πλοία της πρώτης γενιάς είχαν χωρητικότητα περίπου 400 TEU (Twenty-foot container Equivalent Unit). Την τελευταία δεκαετία παρατηρήθηκε μια αξιοσημείωτη αύξηση στα φορτία που μεταφέρονται με containers. Κατά μέσο όρο, ο όγκος αυξήθηκε κατά 6% ετησίως στις Η.Π.Α. , 1,5% στον Καναδά και 10% στον υπόλοιπο κόσμο. Αυτή η τάση είναι πολύ πιθανό να συνεχιστεί και την επόμενη δεκαετία. Υπολογίζεται ότι μέχρι το 2010, το 90% του παγκόσμιου θαλάσσιου φορτίου θα μεταφέρεται με containers.

Αυτή η αύξηση οφείλεται στη δυνατή οικονομία των Η.Π.Α. και των ασιατικών χωρών, στην εξάλειψη των διεθνών εμπορικών συνόρων και στα μεταβατικά φαινόμενα της παγκόσμιας παραγωγής και κατανάλωσης. Σημαντικά διευκόλυναν επίσης, την αύξηση αυτή, οι τεχνολογικές εξελίξεις στις θαλάσσιες μεταφορές. Η αναβάθμιση του εξοπλισμού διαχείρισης των containers (γερανοί, φορτηγά) και η εισαγωγή μεγαλύτερων πλοίων μεταφοράς τους, ικανά να μεταφέρουν 3.000 μονάδες, συνέβαλε στο να αυξηθεί ουσιαστικά η παραγωγικότητα. Η αυξημένη ταχύτητα του εξοπλισμού διαχείρισης, είχε σαν αποτέλεσμα να μειωθεί ο χρόνος παραμονής ενός πλοίου στο λιμάνι για φόρτωση/ εκφόρτωση, ενώ η αύξηση του μεγέθους των πλοίων μεταφοράς containers συνέβαλε στη μείωση του μέσου κόστους μεταφοράς. Παράλληλα, η σύνδεση σιδηροδρόμων με λιμάνια επέφερε την αύξηση της παραγωγικότητας των τερματικών σταθμών και τη μείωση της κυκλοφοριακής συμφόρησης από τα φορτηγά. Πιο συγκεκριμένα, το φορτίο κάθε τρένου με διπλό αποθηκευτικό χώρο containers ισοδυναμεί με φορτίο 200 φορτηγών. Επίσης, η αυτοματοποίηση των πυλών των τερματικών σταθμών οδήγησε σε παραπέρα ελάφρυνση της κυκλοφοριακής συμφόρησης, παρέχοντας ανεμπόδιστη είσοδο στα φορτηγά. Οι τελευταίες εξελίξεις σε συνδυασμό με τις εξελίξεις της τεχνολογίας πληροφοριών και της διοίκησης της πληροφορίας, που σχετίζεται με τη ροή του φορτίου, οδήγησαν σε ταχύτερη διαβίβαση των containers από το ένα μεταφορικό μέσο στο άλλο.

Η συνεχής τάση για μείωση του μέσου μεταφορικού κόστους στις θαλάσσιες διαδρομές έχει ήδη οδηγήσει στην ανάπτυξη ταχύτερων, μεγαλύτερων και βαθύτερων πλοίων, που είναι γνωστά ως Megaships και έχουν τη δυνατότητα να μεταφέρουν πάνω από 8.000 containers. Το κόστος ενός τέτοιου πλοίου μπορεί κάλλιστα να υπερβεί τα 250.000.000 € και γι' αυτό, για μια τόσο υψηλή επένδυση απαιτείται ο χρόνος παραμονής του πλοίου στο λιμάνι να είναι ο μικρότερος δυνατός. Μικρός χρόνος παραμονής στο λιμάνι, σημαίνει γρηγορότερη φόρτωση/ εκφόρτωση, με τερματικούς σταθμούς ικανούς να λαμβάνουν και να διαχειρίζονται μεγάλο αριθμό containers σε μικρή χρονική περίοδο. Αυτές οι μεγάλες μαζικές αφίξεις containers θα αποτελέσουν ουσιαστική επιβάρυνση, στο κοντινό μέλλον, για όλες τις λειτουργίες των τερματικών σταθμών.

Τα Megaships, λόγω των υψηλών απαιτήσεών τους για γρήγορους και μεγάλους τερματικούς σταθμούς, θα κινούνται σε θαλάσσιες οδούς μεταξύ κομβικών λιμανιών, ενώ τα μικρότερα πλοία θα συνδέουν τα μικρότερα λιμάνια με τα κομβικά. Μέσα στην επόμενη δεκαετία θα καθοριστεί το ποια λιμάνια θα γίνουν κομβικά. Είναι φανερό ότι μόνο αυτά τα λιμάνια, που επενδύουν σε υποδομή και σε εκσυγχρονισμό των λειτουργιών τους, με στόχο την αύξηση της αποδοτικότητας των τερματικών

τους σταθμών, θα έχουν τη δυνατότητα να συμμετάσχουν στη νέα οικονομική πραγματικότητα, που θα δημιουργήσουν τα Megaships.

Αναμφισβήτητα, αυτές οι αλλαγές αποτελούν μια ουσιαστική πρόκληση για τα λιμάνια και την προσπάθειά τους να ανταγωνιστούν τη διεθνή αγορά. Για να το πετύχουν αυτό θα πρέπει να δοθεί περαιτέρω έμφαση στην αύξηση της αποδοτικότητάς τους και τη γρήγορη διαβίβαση των containers από τα πλοία στα υπόλοιπα μέσα μεταφοράς (φορτηγά, τρένα) και το αντίστροφο. Απαραίτητη προϋπόθεση βέβαια, για κάθε λιμάνι αποτελεί η επένδυση σε εξοπλισμό διαχείρισης containers. Η επένδυση αυτή περιλαμβάνει από την αγορά μεγαλύτερων και γρηγορότερων γερανών, μέχρι και την ανάπτυξη ευφυών συστημάτων ελέγχου που θα είναι ικανά να εντοπίζουν containers και να παρακολουθούν την απόδοση του εξοπλισμού μέσω ενσωματωμένων συσκευών του παγκόσμιου συστήματος εντοπισμού (GPS). Γενικότερα, είναι απαραίτητο για τη διοίκηση των λιμανιών και των τερματικών σταθμών να αναπτύξουν νέες τεχνικές και μεθόδους για την επιτάχυνση των λειτουργικών διαδικασιών.
Σήμερα, υπάρχουν πολλές τεχνολογίες, που σχετίζονται με τις εργασίες σε ένα τερματικό σταθμό containers. Η πιο ακριβής εκτίμηση των τεχνολογιών αυτών είναι η υλοποίησή τους σε ένα περιβάλλον τερματικού σταθμού containers σε πλήρη κλίμακα και η συλλογή στοιχείων κατά τη διάρκεια των εργασιών. Αυτό ωστόσο, δεν είναι δυνατό καθώς, μερικές από αυτές τις τεχνολογίες βρίσκονται σε σχεδιαστικό στάδιο, ενώ άλλες χρειάζεται να εκτιμηθούν ή να ξεπεραστούν κάποια εμπόδια, που καθιστούν αδύνατη την εφαρμογή τους. Είναι απαραίτητη λοιπόν, η εκτίμηση τέτοιων συστημάτων με τη χρήση μοντέλων και προσομοιώσεων, ώστε να υπολογιστεί η αποδοτικότητά τους και να εξεταστεί η δυνατότητα εφαρμογής τους.

Στην παρούσα διπλωματική εργασία εξετάζουμε αλγόριθμους με τη βοήθεια προσομοίωσης για τη βελτιστοποίηση της διαδικασίας φόρτωσης/ εκφόρτωσης. Το πρόβλημα του σχεδιασμού αυτής της διαδικασίας, περιλαμβάνει την εύρεση θέσης για το προς αποθήκευση container, έτσι ώστε, να αποθηκευτεί και να μπορεί να ανακτηθεί όσο το δυνατόν γρηγορότερα, την εύρεση container για ανάθεση του οχήματος, καθώς και την εύρεση γερανού που θα εξυπηρετήσει γρηγορότερα το όχημα.

Κεφάλαιο 1

1. Συστήματα αποθήκευσης και ανάκτησης

Από την εμφάνιση των μεταφορών με containers και έπειτα, έχουν υπάρξει αλλαγές στον τρόπο, με τον οποίο το φορτίο μετακινείται από τον αποθηκευτικό χώρο στο πλοίο και από τον αποθηκευτικό χώρο στις πύλες εισόδου/ εξόδου . Στην περίπτωση των containers, η μετακίνηση από και προς τον αποθηκευτικό χώρο γίνεται με τη χρήση χειροκίνητων μηχανημάτων, όπως φορτηγά, γερανογέφυρες με ελαστικά, τρένλερ, διασκελισμένους γερανούς, κ.α.

Ωστόσο, πολλές ενδιαφέρουσες εξελίξεις βρίσκονται καθ' οδών και έχουν ως στόχο τη μεγιστοποίηση της απόδοσης τέτοιων συστημάτων. Αυτές οι εξελίξεις περιλαμβάνουν τη χρήση αυτοματοποιημένων κατευθυνόμενων οχημάτων (AGVs – Automated Guided Vehicles), συστημάτων αυτόματης αποθήκευσης containers, αυτοματοποιημένων γερανών κ.α. Στην παρούσα εργασία θα ασχοληθούμε κυρίως με τα AGVs .

Η διαδικασία μεταφοράς των containers, διαμέσου του τερματικού είναι η εξής:

Τα containers προς εισαγωγή:

1. περνάνε την πύλη,
2. αποθηκεύονται στον τερματικό σταθμό,
3. ανακτώνται από τον τερματικό σταθμό και έρχονται στο πλοίο,
4. φορτώνονται στο πλοίο.

Τα container προς εξαγωγή:

1. ξεφορτώνονται από το πλοίο,
2. αποθηκεύονται στον τερματικό,
3. ανακτώνται από τον τερματικό,
4. περνάνε τη πύλη.

Σε ορισμένες περιπτώσεις, η αποθήκευση στον τερματικό μπορεί να παραληφθεί τόσο για τα εξαγόμενα όσο και για τα εισαγόμενα containers.

1.1 Κριτήρια απόδοσης και παράγοντες για την αξιολόγηση του εξοπλισμού φόρτωσης/ εκφόρτωσης

Σίγουρα, το περισσότερο διαδεδομένο μέτρο της απόδοσης του εξοπλισμού φόρτωσης/ εκφόρτωσης αποτελεί ο μέσος χρόνος φόρτωσης, μετρούμενος σε κινήσεις ανά ώρα. Ο μέσος κύκλος, ή κίνηση, ορίζεται ως ο μέσος χρόνος, που απαιτείται από τον εξοπλισμό φόρτωσης/ εκφόρτωσης, για να παραλάβει ένα container, να το σηκώσει από τη θέση του, να το ξεφορτώσει και να ξαναγυρίσει στην αρχική του θέση έτοιμο να παραλάβει ένα άλλο container. Οι κινήσεις ανά ώρα μπορούν να χρησιμοποιηθούν είτε για να εκτιμηθεί η απόδοση ενός μόνο μηχανήματος φόρτωσης/ εκφόρτωσης, είτε για να εκτιμηθεί η παραγωγικότητα του τερματικού σταθμού. Στην τελευταία περίπτωση, η απόδοση του τερματικού μετριέται σε κινήσεις ανά ώρα ανά γερανό πλοίου. Η απόδοση του τερματικού είναι συνήθως, χαμηλότερη από την απόδοση που μπορεί να επιτευχθεί, όταν οι γερανοί του πλοίου δουλεύουν στο μέγιστο της απόδοσής τους. Για παράδειγμα, γερανοί πλοίου ικανοί για 45 κινήσεις ανά ώρα, μπορούν να οδηγήσουν σε απόδοση τερματικού με 30 κινήσεις ανά ώρα ανά γερανό πλοίου, εξαιτίας των καθυστερήσεων στη μεταφορά των containers από και προς το τερματικό ουρών, που σχηματίζονται στους γερανούς του πλοίου κτλ. Ωστόσο, η απόδοση του τερματικού δεν μπορεί να υπερβεί τη μέση απόδοση του ταχύτερου γερανού του πλοίου. Ένα άλλο μέτρο, που χρησιμοποιείται για την αξιολόγηση της αποδοτικότητας του τερματικού, είναι το κόστος ανά κίνηση, που ορίζεται ως το συνολικό κόστος, συμπεριλαμβανομένης της αγοράς του εξοπλισμού, της ανάπτυξης του τερματικού, της συντήρησης του εξοπλισμού και του εργατικού κόστους προς τις μέσες κινήσεις των γερανών του πλοίου.

Επιπλέον παράγοντες, που λαμβάνονται υπόψη για την αξιολόγηση της απόδοσης του εξοπλισμού φόρτωσης/ εκφόρτωσης αποτελούν:

- Η δυνατότητα για υψηλότερη παραγωγικότητα, π.χ. η δυνατότητα αύξησης της παραγωγικότητας με κατάλληλη τροποποίηση του εξοπλισμού.
- Η επεκτασιμότητα των διατάξεων του εξοπλισμού, ώστε να καλύψουν όλο το χώρο του τερματικού.
- Η δυνατότητα αυτοματοποίησης.
- Περιβαλλοντικά οφέλη.
- Η ικανότητα εξυπηρέτησης σιδηροδρομικού δικτύου.
- Η ασφάλεια.
- Το κατασκευαστικό κόστος.
- Το κόστος εξοπλισμού.
- Το κόστος συντήρησης.
- Το κόστος εργατικού δυναμικού.
- Η αξιοπιστία.

Στη συνέχεια της αναφοράς μας, χρησιμοποιούμε τις κινήσεις ανά ώρα ως το βασικότερο κριτήριο μέτρησης της απόδοσης, καθώς είναι ευκολότερη η ποσοτικοποίησή του σε σχέση με άλλα κριτήρια, ενώ σίγουρα αποτελεί ένα αξιόπιστο μέτρο σύγκρισης.

1.2 Εξοπλισμός φόρτωσης/ εκφόρτωσης για το πλοίο

Ο γερανός είναι το κύριο μηχάνημα, που χρησιμοποιείται για τη φόρτωση/ εκφόρτωση του πλοίου. Οι γερανοί μπορεί να βρίσκονται είτε στο μόλο, είτε ενσωματωμένοι στο ίδιο το πλοίο. Έχει διαπιστωθεί, ότι παρόλο που οι γερανοί, που είναι ενσωματωμένοι στο πλοίο, παρέχουν ευελιξία όσον αφορά στις εγκαταστάσεις και τον εξοπλισμό του λιμανιού, οι γερανοί μόλου έχουν καλύτερη απόδοση. Επιπλέον, οι ενσωματωμένοι στο πλοίο γερανοί αυξάνουν σημαντικά το βάρος και συνεπώς, το λειτουργικό κόστος του τελευταίου. Ωστόσο, αυτού του τύπου οι γερανοί αποτελούν τη βέλτιστη λύση για πλοία που βρίσκονται σε λιμάνια, τα οποία δεν είναι πολύ καλά εξοπλισμένα, κάτι που παρατηρείται συχνά σε υποανάπτυκτες χώρες.

Παρακάτω παρουσιάζονται μερικοί βασικοί τύποι γερανών και τα τεχνικά χαρακτηριστικά τους.

1. Γερανοί ενσωματωμένοι στο μόλο.

Η πλειοψηφία των σύγχρονων γερανών ενσωματωμένων στο μόλο, γερανογέφυρες πάνω σε ράγες (αν και μερικές φορές χρησιμοποιούνται γερανογέφυρες με ελαστικά), χρησιμοποιούνται κυρίως για φόρτωση/ εκφόρτωση containers. Αυτοί οι γερανοί είναι ικανοί να προσεγγίσουν οποιοδήποτε σημείο στο κύτος του πλοίου και η απόδοσή τους κυμαίνεται μεταξύ 25 με 35 κινήσεις ανά ώρα.

Τύπος: Συμβατικές και τροποποιημένες γερανογέφυρες Α- πλαισίου (Conventional and modified A-frame Cranes)

Εικόνα 1.2.1: **Συμβατική γερανογέφυρα**

Σύντομη περιγραφή: Η συμβατική και η τροποποιημένη γερανογέφυρα Α- πλαισίου είναι η πιο κοινή αξιόπιστη χρησιμοποιούμενη γέφυρα. Το όνομά της προέρχεται από το Α που σχηματίζεται όταν παρατηρηθεί από κάτω. Η τροποποιημένη γερανογέφυρα

Α- πλαισίου είναι είτε χαμηλότερου προφίλ ή με αρθρωτή βάση. Οι περισσότερες γερανογέφυρες αυτού του είδους έχουν 15 m προέκταση προς τα πίσω, 30.5 m άνοιγμα και 44 – 49 m προέκταση προς τα εμπρός. Συνήθως, εξυπηρετούν πλοία post- Panamax φάρδους 16 containers, αλλά μπορούν να εξυπηρετήσουν και αυτά που έχουν φάρδος 18 containers με τη χρήση επέκτασης. Η απόδοσή τους κυμαίνεται από 20 έως 35 κινήσεις ανά ώρα. Το κόστος τους υπολογίζεται μεταξύ 5 με 7 εκατομμύρια ευρώ, αλλά αυτό μπορεί να αυξηθεί μέχρι και 50% για τις τροποποιημένες γερανογέφυρες.

Απόδοση: Τα χαρακτηριστικά της απόδοσης των συμβατικών και τροποποιημένων γερανογεφυρών συνοψίζονται παρακάτω:

Κινήσεις ανά ώρα	20-35
Προέκταση προς τα πίσω (m)	~15
Προέκταση προς τα εμπρός (m)	44-49
Πλοία που εξυπηρετούν	16-18 Post- Panamax, σχεδιάζονται και για 21
Κόστος (σε εκατομμύρια ευρώ)	5-7 (συμβατικές Α- πλαισίου), 7-11 (τροποποιημένες)
Κύκλοι λειτουργίας	Μονός και διπλός
Κατάσταση ανάπτυξης	Χρησιμοποιείται ήδη

Τύπος: Megacranes

Σύντομη περιγραφή: Με την εμφάνιση πλοίων μεγαλύτερων από τα post- Panamax, τα Megaships, έγινε απαραίτητη η κατασκευή μεγαλύτερων γερανών. Αυτοί οι γερανοί (Megacranes) χαρακτηρίζονται από τις μεγαλύτερες προεκτάσεις προς τα εμπρός και προς τα πίσω, από την ικανότητα μεγαλύτερης ανύψωσης και από το συνολικό ύψος. Η μεγαλύτερη πρόκληση στο σχεδιασμό τους είναι η διατήρηση, αν όχι η αύξηση της παραγωγικότητας, σε σχέση με μικρότερους γερανούς. Υπάρχουν δύο σχεδιαστικές προσεγγίσεις για να επιτευχθεί αυτό. Η πρώτη χρησιμοποιεί ένα άκαμπτο σκελετό, ώστε να αποφευχθούν οι αποκλίσεις και οι δονήσεις του γερανού, που οφείλονται στην κίνησή του. Η δεύτερη προσέγγιση χρησιμοποιεί μία δομή παρόμοια με τις υπάρχουσες (όσον αφορά την ακαμψία) και προηγμένα συστήματα ελέγχου, ώστε να ελεγχθεί το φορτίο και να εξομαλυνθούν οι αποκλίσεις και οι δονήσεις. Εφόσον δίνεται έμφαση πρώτα στο μέγεθος του γερανού και έπειτα στη διατήρηση της απόδοσής του, εκτιμάται ότι αυτή θα κυμαίνεται μεταξύ 40 με 50 κινήσεις ανά ώρα. Η δομή των Megacranes είναι ανάλογη με τους συμβατικούς γερανούς. Βέβαια, το αποτέλεσμα του αυξημένου μεγέθους είναι μια βαρύτερη κατασκευή.

Απόδοση: Η απόδοση των Megacranes και άλλα χαρακτηριστικά τους συνοψίζονται στον επόμενο πίνακα.

Κινήσεις ανά ώρα	40-50
Προέκταση προς τα πίσω (m)	20
Βάρος (τόνοι)	1150-1300 (30% βαρύτεροι από ένα συμβατικό γερανό)
Ύψος (m)	~70
Προέκταση προς τα εμπρός (m)	51
Πλοία που εξυπηρετούνται	Megaships (>7.000 TEU)
Κόστος	Δεν υπάρχουν στοιχεία
Κύκλοι λειτουργίας	Μονός και διπλός
Κατάσταση ανάπτυξης	Σχεδιαστικό στάδιο

2. Γερανοί ενσωματωμένοι στο πλοίο

Εικόνα 1.2.2: Γερανοί ενσωματωμένοι στο πλοίο

***Εικόνα 1.2.3:* Περιστρεφόμενοι γερανοί ενσωματωμένοι στο πλοίο**

Οι γερανοί που είναι ενσωματωμένοι στο πλοίο παρέχουν ευελιξία, όσον αφορά τις εγκαταστάσεις και τον εξοπλισμό του λιμανιού, αλλά δεν επιτυγχάνουν την υψηλή απόδοση των γερανών που είναι ενσωματωμένοι στο μόλο.

Τύπος: Γερανογέφυρες ενσωματωμένες στο πλοίο (Ship- mounted Gantry Cranes)

Σύντομη περιγραφή: Υπάρχουν τρεις κύριοι διαφορετικοί τύποι τέτοιων γερανών: ο τύπος Α, ο τύπος Β και ο περιστρεφόμενος. Ο τύπος Α ή Α- πλαισίου είναι παρόμοιος με το γερανό Α- πλαισίου, που είναι ενσωματωμένος στο μόλο με τα δύο του πόδια στις άκρες του πλοίου. Μια κινητή άκρη βρίσκεται στο πάνω μέρος της γέφυρας, που σχηματίζεται από τα δύο πόδια και η γέφυρα προεκτείνεται από τα άκρα του πλοίου.

Ο τύπος Β είναι παρόμοιος με τον τύπο Α, με τη διαφορά ότι έχει ένα πόδι σε κάθε άκρη αντί για δύο. Το κύριο πλεονέκτημα αυτού του τύπου είναι ότι μπορεί να συνδυαστεί, ώστε να λειτουργεί με ακόμα ένα γερανό Β τύπου σε παράλληλη λειτουργία. Αυτό είναι εξαιρετικής σημασίας όταν χρειάζεται να διαχειριστούν μεγάλα ή βαριά φορτία.

Ο περιστρεφόμενος γερανός είναι παρόμοιος με τον Β τύπου στο χαρακτηριστικό της μιας μόνο γέφυρας, που υποστηρίζεται από μονά πόδια (εικόνα 1.2.3). Ο γερανός αυτός δεν χρησιμοποιεί προεκτάσεις εκτός πλοίου, με αποτέλεσμα να ελαττώνεται το βάρος του. Τέτοιου τύπου γερανοί είναι ιδιαίτερα χρήσιμοι όταν το κατάστρωμα του πλοίου είναι στενό ή όταν απαιτείται ένας γερανός μικρού βάρους.

3. Ημιαυτόματοι και αυτόματοι γερανοί

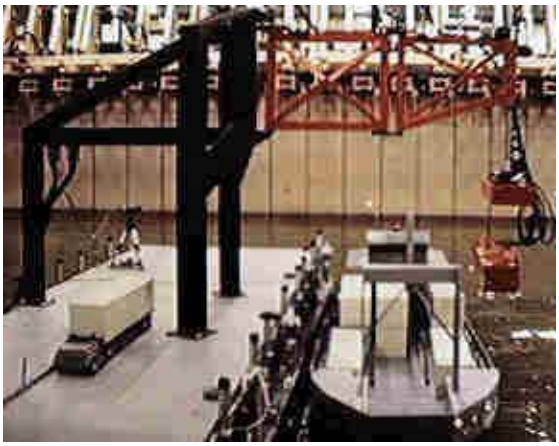
Παρόλο, που υπάρχουν πολλοί διαφορετικοί τρόποι, για να ταξινομηθούν οι γερανοί (π.χ. μηχανολογικά και σχεδιαστικά χαρακτηριστικά, βάρος, απόδοση κ.α.), σε αυτή την παράγραφο ταξινομούνται ανάλογα με το βαθμό αυτοματοποίησής τους. Έτσι, ορίζονται τρεις διαφορετικοί τύποι: οι συμβατικοί χειροκίνητοι γερανοί, οι ημιαυτόματοι και οι αυτόματοι γερανοί. Οι συμβατικοί χειροκίνητοι γερανοί είναι αυτοί, η απόδοση των οποίων βασίζεται κυρίως στις ικανότητες του χειριστή. Οι ημιαυτόματοι είναι αυτοί που είναι εξοπλισμένοι με προηγμένα συστήματα ελέγχου και αισθητήρες, που υποβοηθούν το χειριστή έτσι ώστε, να επιτυγχάνεται καλύτερη απόδοση. Για παράδειγμα, το σύστημα ελέγχου ενός ημιαυτόματου γερανού μπορεί να ελέγξει την ταχύτητα των επιμέρους εξαρτημάτων του γερανού προς αποφυγή των ταλαντώσεων και ακριβή τοποθέτηση της τελικής θέσης της αρπάγης. Αυτό είναι ιδιαίτερα σημαντικό καθώς, σύμφωνα με μια πρόσφατη μελέτη το 30% της κίνησης ενός χειροκίνητου γερανού καταναλώνεται στην εξάλειψη ταλάντωσης και στην ακριβή τοποθέτηση της αρπάγης. Οι αυτόματοι γερανοί είναι εξοπλισμένοι με προηγμένα και ευφυή συστήματα ελέγχου και δεν απαιτούν χειριστή.

<u>Τύπος:</u> Γερανοί με συστήματα μείωσης της ταλάντωσης (Cranes with Anti-Sway Systems)

<u>Σύντομη περιγραφή:</u> Αυτοί οι γερανοί είναι εφοδιασμένοι με ειδικά συστήματα ελέγχου για εξάλειψη των ταλαντώσεων. Ένας υπολογιστής διαβάζει την ταχύτητα και τις εντολές θέσης του χειριστή από το χειριστήριο και στέλνει κατάλληλα τροποποιημένες εντολές στη μηχανή του γερανού ώστε να ελεγχθούν οι ταλαντώσεις, χωρίς να αφαιρείται ο έλεγχος από το χειριστή. Ο υπολογιστής μετράει την επιτάχυνση και επιβράδυνση του κινητού μέρους του γερανού ώστε να ταυτίζεται με την περίοδο του εκκρεμούς και το φορτίο να μην ταλαντώνεται στο τέλος της κίνησης. Επίσης, με παρόμοιο τρόπο ελέγχονται και οι κινήσεις του γερανού, ώστε να τοποθετηθεί με ακρίβεια η αρπάγη στην επιθυμητή θέση. Το σύστημα μείωσης της ταλάντωσης Wagner, που έχει εφαρμοστεί στο διεθνές τερματικό σταθμό της Βιρτζίνια, είναι ένα παράδειγμα τέτοιου συστήματος. Λειτουργεί σε τέσσερις διαφορετικές καταστάσεις : (1) χειροκίνητη μείωση της ταλάντωσης, όπου ο υπολογιστής διαβάζει την επιθυμητή ταχύτητα του χειριστή και παρέχει τις κατάλληλες εντολές στη μηχανή, ώστε να επιτευχθεί αυτή η ταχύτητα χωρίς ταλαντώσεις, (2) η κατάσταση μεγάλης μετατόπισης, κατά την οποία η αρπάγη κινείται κατά μήκος της γέφυρας με τη μέγιστη ταχύτητα, προσεγγίζοντας μια θέση που έχει οριστεί από το χειριστή, (3) η κατάσταση μικρής μετατόπισης, κατά την οποία η αρπάγη μετακινείται σε απόσταση ίση με το πλάτος ενός container προς ή

από το μόλο, (4) η κατάσταση απενεργοποίησης του συστήματος μείωσης της ταλάντωσης, κατά την οποία απενεργοποιούνται τα συστήματα ελέγχου για να γίνουν πολύ ακριβείς κινήσεις (π.χ. μετατόπιση 1m).Τα συστήματα μείωσης των ταλαντώσεων επιδέχονται και αμφισβήτησης. Οι περισσότεροι χειριστές γερανών έχουν μεγάλη εξειδίκευση και είναι περήφανοι για την ανεπτυγμένη ικανότητά τους να δουλεύουν αποδοτικά. Σύμφωνα με αρκετούς χειριστές και συντηρητές γερανών, που ρωτήθηκαν από την August Design, μερικοί τύποι συστημάτων μείωσης της ταλάντωσης είναι διασπαστικοί, με την έννοια ότι αφαιρούν τον έλεγχο από το χειριστή απρόβλεπτα πολλές φορές. Σε αυτές τις περιπτώσεις, ο χειριστής προσπαθεί να κάνει μία κίνηση και το σύστημα αντιδρά μετακινώντας το φορτίο διαφορετικά, από ότι θα περίμενε ο χειριστής. Ο χειριστής αντιλαμβάνεται ότι κάτι έχει χαλάσει στο σύστημα ελέγχου του γερανού. Είναι σύνηθες, οι συσκευές μείωσης των ταλαντώσεων να είναι μόνιμα απενεργοποιημένες για να ικανοποιείται ο χειριστής. Είναι δυνατόν, με την πάροδο του χρόνου, ο χειριστής να εξοικειωθεί σε μεγαλύτερο βαθμό με αυτά τα συστήματα, όμως οι πιέσεις σε αυτό το χώρο εργασίας δεν παρέχουν τον απαραίτητο χρόνο.

Τύπος: Αυτόματος γερανός της August Design



Εικόνα 1.2.4: **Γερανός ρομπότ από την August Design**

Σύντομη περιγραφή: Η August Design ανέπτυξε ένα μοντέλο αυτόματου γερανού για container υπό κλίμακα 1:10. Σε κλίμακα 1:1 ο γερανός ρομπότ θα είχε προέκταση προς τα εμπρός 43*m* και ικανότητα ανύψωσης 46*m*. Ο γερανός είναι στην ουσία ένας μεγάλος συνδυασμός ενός ρομπότ SCARA, μιας συσκευής ανύψωσης και μιας αρπάγης 6 βαθμών ελευθερίας (εικόνα 1.2.4). Ο ελαφρός βραχίονας SCARA είναι άκαμπτος οριζόντια, με δύο αρθρώσεις. Είναι ικανός να περιστρέφεται γύρω από τις αρθρώσεις «ώμου» και «αγκώνα». Η συσκευή ανύψωσης μεταφέρει την αρπάγη κατακόρυφα και εξαλείφει τις ταλαντώσεις λόγω των διπλών συρματόσχοινων. Χρησιμοποιείται μηχανική όραση για να εντοπίζεται αυτόματα ο κινούμενος στόχος και να καθοδηγείται αυτόματα η αρπάγη στην κατάλληλη θέση, για να σηκώσει ή να αφήσει το container. Το σύστημα επίσης, παρέχει τη δυνατότητα τηλεχειρισμού από ανθρώπινο χειριστή, ο οποίος μπορεί να επέμβει στο βρόχο ελέγχου.

Έχει υπολογιστεί ότι ένας τέτοιος γερανός μπορεί να φορτώσει ή να ξεφορτώσει ένα πλοίο με μέσο ρυθμό 75 containers περίπου, ανά ώρα. Πέρα από την αυξημένη παραγωγικότητα ο γερανός ρομπότ έχει και άλλα πλεονεκτήματα:

- Μπορεί να λειτουργήσει αποδοτικά με δυνατό αέρα, εξαιτίας της στιβαρής κατασκευής του και την έλλειψη μακριών ταλαντωμένων συρματόσχοινων.
- Μπορεί να λειτουργήσει σε άσχημη θαλασσοταραχή, χρησιμοποιώντας την 6 βαθμών ελευθερίας αρπάγη και προσαρμόζοντάς την στην κίνηση του πλοίου.
- Η χρήση μηχανικής όρασης μειώνει το επίπεδο επίβλεψης.

- Λόγω της μεγάλης παραγωγικότητας του γερανού αναμένεται ότι θα χρησιμοποιείται τόσο στους μόλους όσο και πάνω στα πλοία.

Απόδοση: Στον παρακάτω πίνακα συνοψίζονται τα τεχνικά χαρακτηριστικά του γερανού ρομπότ από την August Design.

Κινήσεις ανά ώρα	75
Ανύψωση (m)	46
Προέκταση προς τα εμπρός (m)	43
Πλοία που εξυπηρετούνται	16-18 container φάρδους post-Panamax
Κόστος (εκατομμύρια ευρώ)	10.8
Κύκλοι λειτουργίας	Μονός και διπλός
Κατάσταση ανάπτυξης	Σχεδιαστικό στάδιο

1.3 Εξοπλισμός φόρτωσης/ εκφόρτωσης για τον τερματικό σταθμό

Στο συγκεκριμένο χωρίο, παρουσιάζονται τα διάφορα είδη εξοπλισμού που χρησιμοποιούνται για τη φόρτωση/ εκφόρτωση των containers μέσα στον τερματικό σταθμό.

Τύπος: Γερανογέφυρα με ελαστικά (Rubber- Tired Gantry-RTG)

Σύντομη περιγραφή: Αυτού του τύπου ο γερανός (εικόνες 1.3.2 και 1.3.3), αναφέρεται στους μετακινούμενους μέσα στο τερματικό γερανούς, που χρησιμοποιούνται για την μετακίνηση containers. Μπορούν επίσης, να χρησιμοποιηθούν για τη φόρτωση/ εκφόρτωση containers σε/ από βαγόνια σιδηροδρόμου.

***Εικόνα 1.3.1:* Διασκελισμένος γερανός**

Τύπος: Διασκελισμένος γερανός (Straddle Carrier)

Σύντομη περιγραφή: Τέτοιοι γερανοί (εικόνα 1.3.1) χρησιμοποιούνται για να στοιβάζονται containers μέσα στο τερματικό. Είναι χειροκίνητοι και τυπικά ικανοί να ανυψώσουν ένα container πάνω από 3 έως 5 άλλα containers.

Εικόνα 1.3.2: **Γερανογέφυρα με ελαστικά**

Εικόνα 1.3.3: **Γερανογέφυρα με ελαστικά μεγάλου ανοίγματος**

Τύπος: Γερανογέφυρα (Bridge crane)

Σύντομη περιγραφή: Η γερανογέφυρα αποτελείται από μία κατασκευή, που μοιάζει με γέφυρα και μια αρπάγη, όπως φαίνεται παρακάτω (εικόνα 1.3.4). Είναι ένας γερανός πάνω σε ένα ζεύγος παράλληλων ανυψωμένων ραγών, με την ικανότητα να μετακινεί ένα φορτίο παράλληλα και κάθετα σε αυτές τις ράγες. Η λειτουργία αυτού του γερανού περιορίζεται στην περιοχή που σχηματίζεται από αυτές τις δύο ράγες.

Εικόνα 1.3.4: **Γερανογέφυρα με ράγες**

Στον παρακάτω πίνακα δίνονται οι χρόνοι, που απαιτούνται για φόρτωση/ εκφόρτωση containers από τους γερανούς, μέσα στον τερματικό σταθμό.

Τύπος	**Χρόνος φόρτωσης/ εκφόρτωσης**			**Υποθέσεις**
	Καλύτερη περίπτωση	**Χειρότερη περίπτωση**	**Μέσος χρόνος**	
Γερανογέφυρα με ελαστικά	30 sec	150 sec	50 sec	Το όχημα είναι ευθυγραμμισμένο κάτω από το γερανό
Διασκελισμένος γερανός	20 sec	60 sec	30 sec	Το όχημα βρίσκεται 12m μακριά από το container
Γερανογέφυρα	35 sec	145 sec	50 sec	Το όχημα βρίσκεται 12m μακριά από το container

ΚΕΦΑΛΑΙΟ 2

2. Τεχνολογίες containers

2.1 Εισαγωγή

Εικόνα 2.1.1: **Containers γενικού φορτίου σε τερματικό σταθμό**

Η χρήση των containers αυξήθηκε σημαντικά μετά το τέλος του πόλεμου της Κορέας το 1960. Σήμερα για το 70% των φορτίων, που μεταφέρονται δια θαλάσσης, χρησιμοποιούνται containers. Ανάμεσα στα πλεονεκτήματα της μεθόδου είναι ότι τα containers έχουν τυποποιημένη φόρμα και απαιτούνται σχετικά λίγα είδη εξοπλισμού για τη διαχείριση τους, ανεξάρτητα με το είδος του φορτίου μέσα σε αυτά. Με τη χρήση τους, πολλά μικρά δέματα συνδυάζονται σε μια μεγαλύτερη μονάδα, απλοποιώντας το χειρισμό της και μειώνοντας το κόστος διαχείρισής της. Επιπρόσθετα, τα containers προσφέρουν πολύ μεγάλη ασφάλεια στα εσωτερικά τους φορτία, τόσο κατά τη μεταφορά, όσο και κατά την αποθήκευσή τους. Αυτό είναι αρκετά σημαντικό, όταν το φορτίο περιέχει εκρηκτικό η εύφλεκτο υλικό ή όταν τα μεταφερόμενα προϊόντα πρέπει να διατηρούνται σε συγκεκριμένη θερμοκρασία.

Απέναντι βέβαια σε όλες τις παραπάνω ευκολίες, τις οποίες προσφέρουν τα containers, πρέπει να παρατεθούν και το ακριβό κόστος αγοράς, αλλά και συντήρησής τους. Το κόστος αυτό μάλιστα, ανεβαίνει κατά πολύ λαμβάνοντας υπόψη το γεγονός, ότι χρειάζεται ένας μεγάλος αριθμός από αυτά για να επιτευχθεί ένα αποδοτικό επίπεδο υπηρεσίας. Αν το εμπόριο μεταξύ δυο περιοχών είναι μεταβλητό, ο αριθμός των άδειων, και γι' αυτό μη κερδοφόρων, containers μπορεί να αυξηθεί δραστικά στην περιοχή που μειονεκτεί στην εξαγωγή. Γι' αυτό, η χρήση τους είναι ιδανική ανάμεσα σε δυο περιοχές όπου η ροή των προϊόντων είναι αμφίδρομα ίση.

2.2 Τύποι containers

2.2.1 Container Γενικού Φορτίου

Container γενικού φορτίου *(general cargo container),* είναι κάθε container, που δεν πρόκειται να χρησιμοποιηθεί για αερομεταφορά και που δεν έχει αρχικά κατασκευαστεί για τη μεταφορά φορτίων, όπως υγρών, αερίων ή χύμα φορτίων. Τα containers γενικού φορτίου χωρίζονται σε containers γενικού σκοπού (general purpose), ειδικού σκοπού (specific purpose) και ειδικού φορτίου containers (specific cargo containers).

Γενικού σκοπού containers

Τα containers γενικού σκοπού είναι ανεπηρέαστα από τις καιρικές συνθήκες, με άκαμπτη οροφή, πάτωμα και πλαϊνά τοιχώματα. Τα containers αυτά είναι κατάλληλα για τη μεταφορά ποικίλων φορτίων. Η πλειοψηφία αυτού του είδους των containers έχουν διαστάσεις 20 και 40 πόδια μήκος και 8.6 ύψος. Το 1992 παρουσιάστηκαν 3.160.000 20άρια containers γενικού σκοπού, αντιπροσωπεύοντας το 49% του συνολικού αριθμού παγκοσμίως, και 1.650.000 40άρια containers γενικού σκοπού, αντιπροσωπεύοντας το 26%. Ένας άλλος τύπος containers, που έχει σημαντικό ποσοστό στο συνολικό αριθμό παγκοσμίως, είναι αυτά που έχουν ύψος 9.6 πόδια και λέγονται containers υψηλού κύβου (high cube containers). Αυτού του είδους τα 20άρια και 40άρια containers γενικού σκοπού αντιπροσώπευαν το 7% του συνολικού αριθμού (457.000) το 1992. Τα containers υψηλού κύβου παρέχουν επιπλέον, όγκο για ελαφρύ, ογκώδες φορτίο για περισσότερη άνεση και απόδοση. Επίσης, υπάρχουν τα 45άρια (μήκος) υψηλού κύβου containers. Στην πραγματικότητα, τα περισσότερα 45αρια είναι υψηλού κύβου. Τα containers γενικού σκοπού κατασκευάζονται από αλουμίνιο ή ατσάλι. Τα containers αλουμινίου έχουν λίγο μεγαλύτερη χωρητικότητα από τα ατσάλινα, ενώ τα ατσάλινα έχουν μεγαλύτερο ωφέλιμο φορτίο από τα αλουμινένια.

Ειδικού σκοπού containers
Τα containers ειδικού σκοπού έχουν ειδικά κατασκευαστικά χαρακτηριστικά. Οι διαστάσεις τους είναι συνήθως συμβατές με αυτά του γενικού σκοπού, έτσι ώστε να χωράνε στον αποθηκευτικό χώρο ενός συμβατικού πλοίου container και να διαχειρίζονται από συμβατικό εξοπλισμό φόρτωσης/ εκφόρτωσης. Στη συνέχεια, περιγράφονται διάφοροι τύποι containers ειδικού σκοπού.

- **Κλειστά με αεραγωγούς/ αεριζόμενα containers.**

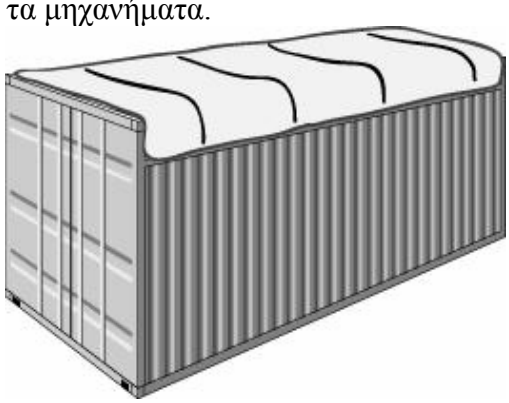
  Τα κλειστά με αεραγωγούς/ αεριζόμενα containers (closed vented/ ventilated containers) είναι παρόμοια με τα containers γενικού σκοπού, αλλά είναι σχεδιασμένα ώστε να επιτρέπουν την ανταλλαγή αέρα μεταξύ του εσωτερικού του container και της εξωτερικής ατμόσφαιρας. Τα containers με αεραγωγούς έχουν παθητικούς αεραγωγούς στο πάνω μέρος του χώρου του φορτίου. Τα αεριζόμενα containers ωστόσο, έχουν σύστημα αερισμού σχεδιασμένο να επιταχύνει και να αυξάνει ομοιόμορφα τον αέρα μέσα σε αυτά.

- **Containers ανοιχτής οροφής**

  Τα containers ανοιχτής οροφής (open top containers) (εικόνες 2.2.1.1 και 2.2.1.2) είναι παρόμοια από όλες τις απόψεις με τα γενικού σκοπού containers, εκτός του ότι δεν έχουν άκαμπτη οροφή. Η οροφή τους μπορεί να είναι

εύκαμπτη και κινούμενη ή αφαιρούμενη. Το περισσότερο χρησιμοποιούμενο υλικό για την κατασκευή της είναι το πολυβινυλοχλωρίδιο (PVC). Τα containers ανοιχτής οροφής επιτρέπουν τη φόρτωση φορτίου από πάνω, οπότε και είναι κατάλληλα για χύμα φορτίο, όπως είναι τα μηχανήματα.

Εικόνα 2.2.1.1: **Container ανοιχτής οροφής**



Εικόνα 2.2.1.2: **Σχεδίαση container ανοιχτής οροφής**

- **Containers με βάση πλατφόρμας**

Τα containers με βάση πλατφόρμας (platform-based containers) (εικόνα 2.2.1.3) δεν έχουν πλευρικά τοιχώματα, αλλά έχουν παρόμοια βάση με αυτή ενός κανονικού container. Είναι κατάλληλα για βαριά φορτία ή φορτία που πρέπει να φορτωθούν από πάνω ή από το πλάι, όπως σωλήνες και μηχανήματα. Υπάρχουν τρεις τύποι containers με βάση πλατφόρμας: (α) τα containers με βάση πλατφόρμας με πλήρη υπερδομή, τα οποία έχουν μία μόνιμη διαμήκη κατασκευή μεταξύ των άκρων της οροφής, (β) τα containers με βάση πλατφόρμας με ατελή υπερδομή, τα οποία δεν έχουν μόνιμη διαμήκη κατασκευή μεταξύ των άκρων της οροφής και (γ) τα containers με βάση πλατφόρμας με ατελή υπερδομή και σκελετό που διπλώνει.

Εικόνα 2.2.1.3: **Container με βάση πλατφόρμας**

Εικόνα 2.2.1.4: **Container πλατφόρμα**

- **Containers πλατφόρμες**

Τα containers πλατφόρμες (platform containers) (εικόνα 2.2.1.4) είναι πλατφόρμες, που δεν έχουν υπερδομή, αλλά έχουν τις ίδιες διαστάσεις βάσης με ένα container εφοδιασμένο με πάνω και κάτω γωνίες.

Containers ειδικού φορτίου

Τα containers ειδικού φορτίου (specific cargo containers) προορίζονται εξαρχής για μεταφορά συγκεκριμένων κατηγοριών φορτίου. Παρόμοια με τα containers ειδικού σκοπού, οι διαστάσεις τους είναι όμοιες με αυτές των γενικού σκοπού containers. Τα κυριότερα είδη των containers ειδικού φορτίου είναι:

- **Θερμικά containers**: Τα θερμικά containers (thermal containers) είναι φτιαγμένα με μονωτικά τοιχώματα, πόρτα, πάτωμα και οροφή, ώστε να μειώνεται η μετάδοση θερμότητας μεταξύ του container και του εξωτερικού περιβάλλοντος.
- **Μονωμένα containers**: Τα μονωμένα containers (insulated containers) έχουν μονωμένους τοίχους, πόρτα, πάτωμα και οροφή, αλλά δεν έχουν συσκευές ψύξης ή θέρμανσης.
- **Καταψυχόμενα containers**: Τα καταψυχόμενα containers (refrigerated containers) χρησιμοποιούν πάγο, ξηρό πάγο ή υγροποιημένα αέρια για ψύξη. Αυτά τα containers δεν έχουν εξωτερική πηγή ενέργειας ή κάποια πηγή καυσίμου.
- **Μηχανικά καταψυχόμενα containers**:Τα μηχανικά καταψυχόμενα containers (mechanically refrigerated containers) (εικόνα 2.2.1.5) έχουν μια ψυκτική μονάδα, όπως ένας μηχανικός συμπιεστής ή μια μονάδα απορρόφησης.
- **Θερμαινόμενα containers**: Τα θερμαινόμενα containers (heated containers) (εικόνα 2.2.1.6) είναι παρόμοια με τα μηχανικά καταψυχόμενα, αλλά έχουν μια μονάδα παραγωγής θερμότητας αντί μια ψυκτική μονάδα.
- **Θερμαινόμενα και καταψυχόμενα containers**: Τα θερμαινόμενα και καταψυχόμενα containers (refrigerated and heated containers) έχουν τόσο ψυκτική μονάδα, όσο και μονάδα παραγωγής θερμότητας.

Εικόνα 2.2.1.5: **Μηχανικά καταψυχόμενα containers**

Εικόνα 2.2.1.6: **Θερμαινόμενα containers**

2.2.2 Άλλοι τύποι containers

Υπάρχουν τρεις κύριοι τύποι containers, που δεν ανήκουν στην κατηγορία γενικού φορτίου:

- **Containers ντεπόζιτα:** Τα containers ντεπόζιτα (tank containers) (εικόνα 2.2.2.1) έχουν δυο βασικά στοιχεία, το ή τα ντεπόζιτα και το πλαίσιο. Αυτά τα containers χρησιμοποιούνται κατά βάση για την μεταφορά υγροποιημένων φορτίων.
- **Containers χύμα φορτίου:** Τα containers χύμα φορτίου (dry bulk containers) χρησιμοποιούνται για την μεταφορά στερεών, σε χύμα μορφή, χωρίς συσκευασία. Αποτελούνται από μια κατασκευή μεταφοράς φορτίου ασφαλισμένη καλά μέσα σε ένα πλαίσιο.

- **Containers ονομαστικού φορτίου:** Τα containers ονομαστικού φορτίου (named cargo containers) αποτελούνται από αρκετά είδη containers, όπως είναι τα containers αυτοκινήτων, τα containers ζώων φάρμας κτλ.

Εικόνα 2.2.2.1: **Container ντεπόζιτο**

2.3 Διαστασιολόγηση των containers και τυποποίησή τους

Η χρήση των τυποποιήσεων των containers συνεχίζει να είναι από τους πιο σημαντικούς παράγοντες, που επηρεάζουν την εξέλιξη αυτού του είδους των μεταφορών. Ο παγκόσμιος στόλος containers αυξήθηκε τέσσερις φορές κατά την περίοδο 1980-1983, με το 90% από αυτά να είναι 20άρια και 40άρια με 8.6 πόδια ύψος.

Παρά το γεγονός, ότι η πλειοψηφία των containers είναι 20άρια και 40άρια, το θέμα της τυποποίησής τους είναι ακόμη ανοιχτό. Αρκετοί παράγοντες, όπως η εξέλιξη γερανών, που είναι ικανοί να χειρίζονται μεγαλύτερα και βαρύτερα containers σε σχέση με τους παλιότερους γερανούς, καθώς και η πρόοδος στους σιδηροδρόμους και στην αυτοκινητοβιομηχανία, καθιστούν δυνατό το σχεδιασμό μεγαλύτερων, από τα τυποποιημένα μεγέθη, containers. Τα μεγαλύτερα containers είναι σε πολλές περιπτώσεις προτιμότερα, αφού μπορούν να μεταφέρουν μεγαλύτερους όγκους και βαρύτερα φορτία, αυξάνοντας το ποσοστό του συνολικού όγκου και βάρους που διαχειρίζεται το ένα container.

Ανάμεσα στα containers, που δεν έχουν μήκος 20΄ και 40΄ και ύψος 8.6΄ πόδια, τα πιο δημοφιλή είναι αυτά που έχουν 9.6΄ πόδια ύψος. Τα containers αυτού του ύψους κερδίζουν συνεχώς έδαφος και το μερίδιό τους στο συνολικό αριθμό παγκόσμια, ήταν το 1998 μεγαλύτερο του 7%, ενώ το 1988 ήταν μικρότερο του 3%. Πολλά από τα 9.6΄ ύψους είναι 45΄ πόδια μήκους και ένας μεγάλος αριθμός από αυτά είναι containers με βάση πλατφόρμα ή καταψυχόμενα. Άλλες κατηγορίες containers περιλαμβάνουν:

- 48΄ ποδών containers, τα οποία χρησιμοποιούνται περισσότερο για εγχώριους σκοπούς.

- 53΄ ποδών containers. Αυτά τα containers τείνουν να αναπτύσσονται σε συγκεκριμένες εφαρμογές και κατασκευάζονται κύρια για την εξυπηρέτηση των ιδιαίτερων απαιτήσεων των πελατών.

- Containers διαφόρων μηκών και με πλάτος 8.2 πόδια, χρησιμοποιούνται στην Ευρώπη σε συνδυασμό με προϊόντα στοιβαγμένα σε παλέτες. Υπολογίζεται ότι ένα ισοδύναμο των 200.000 TEU's βρίσκεται αυτή τη στιγμή σε χρήση. Ανάμεσά τους, τα containers πλάτους παλέτας (cellular pallet-wide containers)

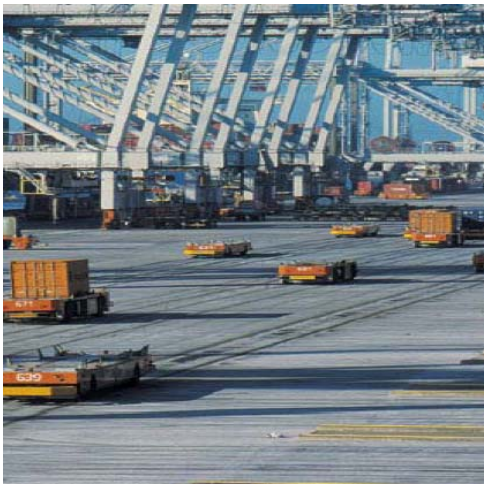
χρησιμοποιούνται στο εμπόριο μικρών αποστάσεων. Το κύριο χαρακτηριστικό αυτών των containers είναι ότι μπορούν να μεταφερθούν από ένα συμβατικό πλοίο container και μπορούν να αποθηκεύσουν παλέτες με πλάτος 3.11 πόδια.

Τα μη τυποποιημένα containers χρησιμοποιούνται, σε γενικές γραμμές, σε συγκεκριμένο διεθνές εμπόριο, όπου επικρατούν οι μεγάλοι όγκοι και η χαμηλή πυκνότητα φορτίου. Το μερίδιο τέτοιων containers, στο συνολικό αριθμό παγκοσμίως, είναι αμελητέο και δεν υπάρχει ενιαία χρήση τέτοιων containers. Μόλις το ένα τρίτο των λιμανιών χρησιμοποιούσαν τέτοια containers το 1998, ενώ ο συνολικός αριθμός τους αντιπροσώπευε το 3.8% των containers παγκοσμίως. Επιπλέον, η χρήση των περισσότερων containers αυτού του είδους φθίνει, ενώ πολλά λιμάνια εγκατέλειψαν τη χρήση τους. Ωστόσο, υπάρχουν παραδείγματα λιμανιών και τερματικών σταθμών, που αρχικά αποφάσισαν να μη δέχονται τέτοια containers και στην πορεία αναγκάστηκαν να αλλάξουν την απόφασή τους, καθώς και τον εξοπλισμό τους, ώστε να μπορούν να διαχειριστούν μη τυποποιημένα containers. Ο λόγος γι' αυτό ήταν ότι δεν ήθελαν να χάσουν μερίδιο της αγοράς, δεδομένου ότι αρκετοί πελάτες χρησιμοποιούσαν μη τυποποιημένα containers.

ΚΕΦΑΛΑΙΟ 3

3. Αυτοματοποιημένα κατευθυνόμενα οχήματα για εφαρμογές λιμανιών

3.1 Εισαγωγή



Εικόνα 3.1.1: **Αυτοματοποιημένα κατευθυνόμενα οχήματα**

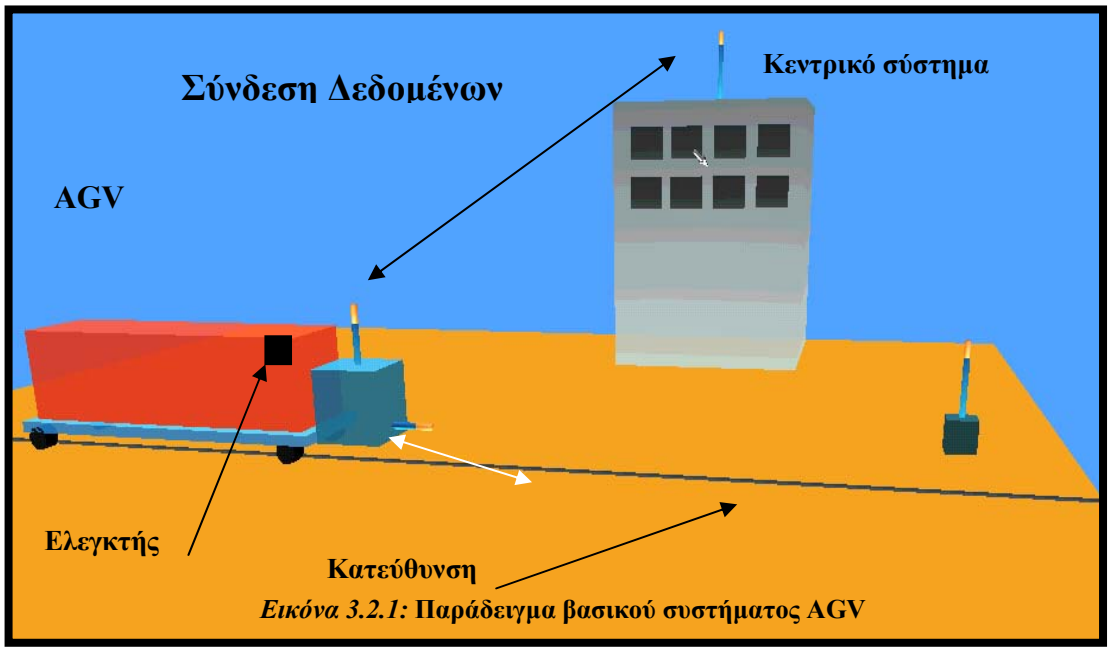
Οι πιο ενδιαφέρουσες εξελίξεις στον εξοπλισμό χειρισμού φορτίων είναι αυτές που σχετίζονται με τα αυτοματοποιημένα κατευθυνόμενα οχήματα (AGVs). Τα AGVs την τελευταία δεκαετία παρουσίασαν μια εκρηκτική ανάπτυξη, κυρίως στον τομέα της βιομηχανικής αυτοματοποίησης. Οι προοπτικές των AGVs προέρχονται από την μεγάλη ελεγξιμότητα και το βαθμό, με τον οποίο μπορούν τα τελευταία να εκτελέσουν τις ίδιες εργασίες, που τώρα απαιτούν τη χρήση σημαντικού εργατικού δυναμικού. Στην περίπτωση της διαχείρισης containers, τα AGVs είναι κατάλληλα για να αλληλεπιδράσουν με μια αυτοματοποιημένη μονάδα αποθήκευσης και ανάκτησης. Προς το παρόν, υπάρχουν εφαρμογές των AGVs σε λιμάνια, όπως στο Ρότερνταμ, στις Κάτω Χώρες και στην Αγγλία. Προκαταρκτικά βήματα βρίσκονται σε εξέλιξη στα λιμάνια της Σιγκαπούρης και στο Καοσίουνγκ στην Ταϊβάν.

Το σύστημα του AGV (εικόνα 3.2.1) αποτελείται από το όχημα, έναν ελεγκτή πάνω στο όχημα, μια σύνδεση δεδομένων με ένα κεντρικό σύστημα διαχείρισης και ένα σύστημα πλοήγησης. Τα οχήματα κινούνται με ηλεκτρική ενέργεια και είναι κατασκευασμένα με ετοιμοπαράδοτα υλικά. Ο ελεγκτής πάνω στο όχημα διαχειρίζεται την ώθηση, το τιμόνι, τα φρένα και άλλες λειτουργίες του οχήματος. Το σύστημα διαχείρισης ασχολείται με την ανάθεση, δρομολόγηση και τον έλεγχο της κυκλοφορίας. Το σύστημα πλοήγησης χρησιμοποιείται για την καθοδήγηση του οχήματος στον προορισμό του.

3.2 Έλεγχος και σύστημα διαχείρισης AGV

Καθώς το AGV λειτουργεί σε συγκεκριμένο περιβάλλον, απαιτεί έναν ελεγκτή ενσωματωμένο στο όχημα και/ ή έναν κεντρικό υπολογιστή για να συντονίζει τις κινήσεις του σε σχέση με άλλα μηχανήματα διαχείρισης φορτίου ή άλλα AGVs. Γι' αυτό, πρέπει να δούμε το σύστημα του AGV σαν ένα εποπτικό σύστημα, το οποίο έχει ιεραρχική δομή.

Το χαμηλότερο επίπεδο σ' αυτή την ιεραρχία ανήκει στον ενσωματωμένο ελεγκτή του οχήματος, ο οποίος αποτελείται από το σύστημα ελέγχου οδήγησης (π.χ. μηχανή, μετάδοση, φρένα, κ.α.) και το σύστημα πλοήγησης. Το υψηλότερου επιπέδου σύστημα ελέγχου είναι υπεύθυνο για τη διαχείριση και την αλληλεπίδραση της θέσης των οχημάτων. Αποτελείται από το σχεδιασμό, τον προγραμματισμό και την εκτέλεση των κινήσεων.



Εικόνα 3.2.1: **Παράδειγμα βασικού συστήματος AGV**

3.2.1 Συστήματα πλοήγησης

Τα συστήματα πλοήγησης των AGVs εξελίσσονται από συστήματα καθοδήγησης με καλώδιο σε συστήματα ελεύθερης εμβέλειας. Μέχρι πρόσφατα, τα AGVs ακολουθούσαν ένα καλώδιο οδήγησης (εικόνα 3.2.1) ή μια οπτικά ορατή γραμμή, βαμμένη ή φτιαγμένη από ταινία, στο έδαφος. Σήμερα τα AGVs χρησιμοποιούν νέες μεθόδους . Παρακάτω παρουσιάζονται μερικές τεχνολογίες, που χρησιμοποιούνται ή θα μπορούσαν να χρησιμοποιηθούν από τα AGVs για καθοδήγηση και έλεγχο.

1. *Σύστημα Καθοδήγησης με Καλώδιο.* Η πλειοψηφία των αυτοματοποιημένων οχημάτων, που χρησιμοποιούνται σήμερα, είναι σχεδιασμένα να ακολουθούν ένα καλώδιο, που είναι θαμμένο στο έδαφος. Ένα εναλλασσόμενο ρεύμα παράγει ένα ηλεκτρομαγνητικό πεδίο γύρω από το καλώδιο. Τα διερχόμενα οχήματα ανιχνεύουν το μαγνητικό πεδίο, χρησιμοποιώντας δύο μαγνητικά πηνία, που βρίσκονται πλευρικά στο όχημα. Η διαφορά μεταξύ των τάσεων αυτών των πηνίων χρησιμοποιείται για να βγει η εντολή ελέγχου για τον μηχανισμό στροφής του οχήματος. Ανάμεσα στα τεχνικά πλεονεκτήματα είναι ότι η καθοδήγηση με καλώδιο είναι καλά δοκιμασμένη και σχετικά απλή. Ωστόσο, είναι ακριβή η συντήρηση, δύσκολη η εγκατάσταση, χρειάζεται υποδομή και δεν είναι ευέλικτη μέθοδος, καθώς απαιτείται η παρουσία καλωδίου για οποιοδήποτε προορισμό χρειαστεί το όχημα να προσεγγίσει.
2. *Μαγνητικά Συστήματα.* Παρόμοια με το σύστημα καθοδήγησης με καλώδιο, η θεμελιώδης αρχή καθοδήγησης είναι να παρέχεις μια διαδρομή στο έδαφος, εύκολη για ένα όχημα να την ακολουθήσει. Οι μαγνητικές ράβδοι (συστήματα) είναι αξιόπιστες και μπορούν να χρησιμοποιηθούν σε όλες τις καιρικές συνθήκες, αλλά δεν είναι ευέλικτες, είναι ευαίσθητες σε μαγνητικές παρεμβολές και μπορούν να παρουσιάσουν προβλήματα στην άσφαλτο.
3. *Πλέγμα Ελεύθερης Εμβέλειας.* Αυτό το σύστημα χρησιμοποιεί ένα πλέγμα, σημειωμένο στο δάπεδο με πομπούς στους κόμβους για να μεταδίδουν σήματα,

που περιέχουν μια μοναδική «ταυτότητα» και πληροφορίες θέσης. Κάθε φορά που ένα όχημα συναντά μια γραμμή του πλέγματος, συγκρίνει αυτήν την πληροφορία με την υπολογισμένη θέση του και κάνει αυτόματα τη διόρθωση. Αυτό το σύστημα πλοήγησης, εφαρμοσμένο στο Ρότενταρμ, είναι ευέλικτο και έχει ακρίβεια ± 3cm. Είναι δύσκολο να εγκατασταθεί και οι πομποί είναι δυνατό να μετακινηθούν με την κακοκαιρία.

4. *Καθοδήγηση με Λέιζερ*. Το όχημα χρησιμοποιεί ακτίνες λέιζερ και ανακλαστήρες για να υπολογίσει την απόστασή του από καθορισμένα σημεία, χρησιμοποιώντας τριγωνισμό. Ένα ελάχιστο τριών στόχων πρέπει να ανιχνεύεται, κάθε στιγμή, κατά τη διάρκεια της κίνησης. Κανονικά πρέπει να υπάρχουν πάντα πέντε ορατοί στόχοι. Το όχημα μπορεί τότε να χρησιμοποιήσει τον ενσωματωμένο χάρτη για να προσδιορίσει και να διορθώσει τη θέση του. Αυτό το σύστημα είναι ευέλικτο, ακριβές και απαιτεί μικρό κόστος υποδομής. Ωστόσο, επηρεάζεται από τον καιρό, χρειάζεται μεγάλο αριθμό ανακλαστήρων και μεγάλο χρόνο αρχικού στησίματος.
5. *Ραντάρ Κυμάτων - Millimeter Wave Radar (MMWR)*. Ένα περιστρεφόμενο MMWR ανιχνεύει την παρουσία ραδιοφάρων καθοδήγησης σε γνωστές θέσεις, για να προσδιορίσει τη θέση του οχήματος. Οι παρατηρήσεις των ραδιοφάρων κατεύθυνσης επεξεργάζονται για συνεχή ενημέρωση της θέσης του οχήματος. Το MMWR είναι ακριβείας μέχρι ± 10cm, αλλά είναι ακριβό χρειάζεται μεγάλο χρόνο αρχικού στησίματος, καθώς και μεγάλο αριθμό ανακλαστήρων.
6. *Διαφορικό GPS*. Το σύστημα διαφορικού GPS (DGPS) μπορεί να χρησιμοποιηθεί για πλοήγηση μέσα στον τερματικό σταθμό, παρόλα μερικά κρίσιμα ζητήματα, που έχουν να κάνουν με τη δορυφορική κάλυψη και την αξιοπιστία του σήματος για εμπορικές εργασίες AGV στα λιμάνια. Ένα τέτοιο σύστημα είναι ακριβείας μέχρι ± 5cm, έχει σχετικά μικρό κόστος εγκατάστασης και απαιτεί μερικές μετατροπές στον τερματικό σταθμό.
7. *Υπολογισμός Θέσης από Αναμέτρηση*. Ο υπολογισμός θέσης από αναμέτρηση είναι μία μέθοδος ελεύθερης διαδρομής, που καθοδηγεί υπολογίζοντας σχετικές θέσεις. Σε αυτήν την μέθοδο, ανιχνευτές κίνησης πάνω στο όχημα ανιχνεύουν με ακρίβεια την κατεύθυνση του οχήματος και την ταχύτητά του. Δεδομένου μιας γνωστής αρχικής θέσης, η ολοκλήρωση των πληροφοριών ταχύτητας με το χρόνο επιτρέπει τον υπολογισμό της θέσης του οχήματος. Επειδή τα λάθη συσσωρεύονται, όσο αυξάνει η απόσταση που ταξιδεύει το όχημα, αυτά τα συστήματα γίνονται ανακριβή και αναξιόπιστα, εκτός και αν η θέση του οχήματος διορθώνεται περιοδικά με άλλα μέσα. Αυτή η μέθοδος είναι απλή, ευέλικτη, χαμηλού κόστους και εύκολη να υλοποιηθεί σε πραγματικό χρόνο.
8. *Μηχανική Όραση*. Αυτό το σύστημα χρησιμοποιεί επεξεργασία εικόνας για να παρέχει καθοδήγηση. Τα συστήματα μηχανικής όρασης χρειάζονται μικρή ή καθόλου υποδομή. Έχει βρεθεί ότι παρέχουν εξαιρετικές πληροφορίες θέσης για καθοδήγηση οχήματος και μπορούν να ρυθμιστούν έτσι, ώστε να εκτελούν πολλές διαφορετικές εργασίες (από διατήρηση του ρεύματος, μέχρι αποφυγή συγκρούσεων, αναγνώριση οδικών σημάτων κ.α.). Δε λείπουν βέβαια, και τα μειονεκτήματα, όπως είναι το υψηλό κόστος, η πολυπλοκότητα και έμφυτοι περιορισμοί του βασικού αισθητήρα (κάμερα), που μπορεί να παρέχει πληροφορίες μόνο για ό,τι είναι ορατό άμεσα.
9. *Μαγνητικά καρφιά.* Μικροί μαγνήτες με τη μορφή στερεού κυλίνδρου, διαμέτρου ενός νομίσματος και μήκους περίπου 10 εκατοστών, τοποθετούνται σε τρύπες στη μέση ενός δρόμου με απόσταση μεταξύ τους ενός μέτρου ή και λιγότερο. Μαγνητόμετρα , που τοποθετούνται στο κάτω μέρος του οχήματος αισθάνονται το μαγνητικό πεδίο και παρέχουν μια εκτίμηση της απόκλισης από τη μέση του

δρόμου, ώστε να διορθωθεί η πορεία. Πληροφορίες, όπως η γεωμετρία του δρόμου, θέσεις κρίσιμων κλίσεων, αλλαγή λωρίδας, μπορούν να κωδικοποιηθούν αλλάζοντας την πολικότητα των μαγνητών. Οι Path και Caltrans, που τοποθέτησαν μαγνητικά καρφιά σε ένα κομμάτι της εθνικής οδού I-15 για να επιδείξουν την αυτοματοποιημένη οδήγηση οχημάτων, χρησιμοποίησαν 10 οχήματα με ταχύτητες μεγαλύτερες από 100 χιλιόμετρα/ ώρα τον Αύγουστο του 1997. Η χρήση των μαγνητικών καρφιών, για την καθοδήγηση των AGVs, αναμένεται να είναι ευκολότερη λόγω των μικρών ταχυτήτων.

Εικόνα 3.2.1.1: **AGV με καλώδιο**

Εικόνα 3.2.1.2: **AGV με LASER**

3.2.2 Σχεδιασμός

Ο σχεδιασμός έχει να κάνει με την επιλογή ενός οχήματος και τον προσδιορισμό της πορείας του. Ο σχεδιασμός αναφέρεται συχνά, σαν δρομολόγηση και ανάθεση όσον αφορά τα AGVs. Ανάθεση είναι η διαδικασία της επιλογής και ανάθεσης εργασίας στα οχήματα. Δυο περιπτώσεις είναι δυνατές, όταν υπάρχει ζήτηση για ανάθεση μιας εργασίας σε ένα όχημα: εργασία με βάση σταθμούς εργασίας και εργασία με βάση τα οχήματα. Στην εργασία με βάση τα οχήματα, ένα όχημα επιλέγεται από μια ομάδα ανταγωνιστικών αδρανών οχημάτων. Διαφορετικοί κανόνες μπορούν να εφαρμοστούν για να δοθούν προτεραιότητες στα οχήματα. Τέτοιοι κανόνες είναι ο κανόνας του τυχαίου οχήματος, του κοντινότερου οχήματος ή του λιγότερου χρησιμοποιούμενου οχήματος. Στην εργασία με βάση τους σταθμούς εργασίας, ένας σταθμός εργασίας θα ανατεθεί σε ένα όχημα από μια ομάδα ανταγωνιστικών σταθμών. Υπάρχουν αρκετοί κανόνες για την ανάθεση σταθμού σε ένα όχημα. Τέτοιοι είναι ο κανόνας τυχαίου σταθμού εργασίας, μικρότερου χρόνου ταξιδιού/ απόσταση, μέγιστης εξερχόμενης σειράς και ο τροποποιημένος κανόνας first come first served. Η δρομολόγηση είναι η επιλογή συγκεκριμένων διαδρομών για τα οχήματα, ώστε να φτάσουν στους προορισμούς τους. Ένα σύστημα με AGVs μπορεί να μοντελοποιηθεί σαν γράφημα, που αποτελείται από κόμβους, οι οποίοι συνδέονται από μια ομάδα τόξων. Δεδομένου της θέσης ενός AGV και του προορισμού του, ο διαχειριστής των διαδρομών μπορεί να βρει την ακολουθία των κόμβων, που περιγράφουν τη διαδρομή του οχήματος.

3.2.3 Προγραμματισμός

Ο προγραμματισμός περιλαμβάνει το συνδυασμό όλων των διαδρομών των οχημάτων, σε μια συνολική ακολουθία με επιμέρους τμήματα τις προτεραιότητες των οχημάτων. Με άλλα λόγια, ο σχεδιασμός είναι υπεύθυνος να αναλύει τις διαδρομές των οχημάτων σε μικρότερα τμήματα, ενώ ο προγραμματισμός είναι υπεύθυνος να ορίζει την ακολουθία της πρόσβασης των οχημάτων σε κάθε τμήμα. Η λειτουργία του προγραμματισμού είναι επίσης υπεύθυνη, για να αποτρέπει συγκρούσεις ή αποκλεισμούς των AGVs και να δημιουργεί/ ενημερώνει τους αναμενόμενους χρόνους εκκίνησης και άφιξης για τις επιλεγμένες διαδρομές.

3.2.4 Εκτέλεση

Η εκτέλεση παρέχει τη διασύνδεση του φυσικού συστήματος (σύστημα ελέγχου οδήγησης και πλοήγηση) με το σύστημα ελέγχου υψηλότερου επιπέδου. Για το λόγο αυτό, είναι υπεύθυνο για την αλληλεπίδραση των ελεγκτών των οχημάτων, των διαδικασιών εκκίνησης και κλεισίματος του συστήματος, για την έκδοση εντολών για ανατεθείσες εργασίες και για την παρακολούθηση και ανίχνευση λαθών και διόρθωσή τους.

3.3 Εμπορικές δραστηριότητες

Τα AGVs πραγματοποιούν μια αρχική εισβολή, για εφαρμογές σε λιμάνια, σε πολλά μέρη του κόσμου. Βρέθηκε ότι έγιναν σημαντικές προσπάθειες από αμερικάνικες εταιρίες, στο να προωθήσουν τα AGVs για εφαρμογές σε λιμάνια, αλλά χωρίς αποτέλεσμα. Παραπέρα, αυτές οι εταιρίες είχαν μικρή ή καθόλου επιτυχία στην προώθηση των προϊόντων τους σε ξένες χώρες, κυρίως για πολιτικούς λόγους. Η πιο γνωστή αμερικάνικη εταιρία σ' αυτόν τον τομέα, η Mentor AGV, κατασκεύασε, δοκίμασε και προσπάθησε δυναμικά να μπει στην αγορά των AGVs για εφαρμογές λιμανιών, αλλά μείωσε τις προσπάθειές της για να κυνηγήσει πιο αποτελεσματικά αυτήν την αναπτυσσόμενη αγορά. Σημερινά δείγματα της τεχνολογίας των AGVs βρίσκονται σε συστήματα λιμανιών του Ρότενταρμ, της Αγγλίας, της Σιγκαπούρης, του Καβασάκι και του Καοσίουνγκ.

Λιμάνι Ρότερνταμ

Εικόνα 3.3.1: **τερματικό Deltaport**

Το τερματικό Deltaport στο Ρότερνταμ, όπως φαίνεται και στη διπλανή εικόνα, χρησιμοποιεί AGVs στη μεταφορά containers από τον αποθηκευτικό χώρο (που εξυπηρετείται από γερανούς οροφής σε ράγες) στους γερανούς του πλοίου. Το συνολικό μέγεθος του στόλου είναι 105 οχήματα. Ηλεκτρονικές συσκευές προσδιορισμού θέσης ευθυγραμμίζουν τους γερανούς οροφής με τα οχήματα, για τη γρήγορη φόρτωση του πλοίου. Ένας ενσωματωμένος υπολογιστής σε κάθε AGV επικοινωνεί με το κέντρο ελέγχου για να γίνει δυνατή η πλοήγηση προς κάθε σημείο μέσα στον τερματικό σταθμό.

Λιμάνι Thamesport

Το Thamesport είναι ένας τερματικός σταθμός containers στη βορειοανατολική Αγγλία και είναι πιθανά το πρώτο παγκόσμια, πλήρως αυτοματοποιημένο λιμάνι. Κατασκευασμένο το 1992, το Thamesport χρησιμοποιεί πλήρως αυτοματοποιημένο εξοπλισμό αποθήκευσης (γερανούς οροφής) και δοκιμάζει πρωτότυπα AGVs, κατασκευασμένα από την Terberg Benschop (Κάτω Χώρες) τα τελευταία έξι χρόνια.

Λιμάνι της Σιγκαπούρης

Εικόνα 3.3.2: **Τερματικό Σιγκαπούρης**

Η αρχή του λιμανιού της Σιγκαπούρης έκανε ένα συμβόλαιο για 5 πρωτότυπα AGVs στο Kamag και στο Mitsui στα τέλη του 1994. Σε συζητήσεις με αντιπροσώπους της Mentor AGVs, της μόνης αμερικανικής εταιρίας, που ενεργά επιδιώκει αυτήν την αγορά, βρέθηκε ότι ενδιαφέρονταν τελικά για παραγγελίες περίπου 1.000 με 2.000 AGVs. Στην εικόνα 3.3.2 φαίνονται τα AGVs, τα οποία είναι σχεδιασμένα να επιταχύνουν από 0 έως 5 μέτρα το δευτερόλεπτο σε λιγότερο από 10 δεύτερα, με μέγιστη ταχύτητα 7 μέτρα το δευτερόλεπτο. Επίσης, μπορούν να μεταφέρουν 20-feet ή 40-feet containers.

Λιμάνι του Καβασάκι

Το λιμάνι του Καβασάκι αγόρασε ένα δοκιμαστικό AGV από την Mitsui (γιαπωνέζικη εταιρία) και ένα δοκιμαστικό από μια γερμανική εταιρία, τη Schueulre.

Λιμάνι του Καοσίουνγκ

Το λιμάνι του Καοσίουνγκ έχει δεσμευτεί να αυτοματοποιήσει τους τερματικούς του σταθμούς, ώστε να μπορεί να καλύψει τις απαιτήσεις των αναμενόμενων μελλοντικών αυξήσεων. Στην τελική αναφορά από το Πρόγραμμα Ανάπτυξης του Διεθνούς Λιμανιού του Καοσίουνγκ, προτείνεται ότι το τερματικό του πρέπει να αυτοματοποιηθεί πλήρως για να διαχειρίζεται τα containers. Είναι επίσης γνωστό, ότι το λιμάνι του Καοσίουνγκ εξετάζει πολλές τεχνολογίες αυτοματοποίησης, αν και περισσότερες λεπτομέρειες είναι σπάνιες.

Λιμάνι της Βιρτζίνια και Mentor AGVs

Η Mentor AGVs, η οποία συνεργάστηκε με την Raytheon, προσπαθεί να εξασφαλίσει τη χρηματοδότηση για την ανάπτυξη ενός πρωτότυπου συστήματος AGVs στο λιμάνι της ενδοχώρας της Βιρτζίνια (Virginia Inland Port). Η χρηματοδότηση θα προερχόταν από το υπουργείο αμύνης των Η.Π.Α., παρόλο που η διεθνής εταιρική συνεργασία είναι ανοιχτή και σε εναλλακτικές χρηματοδοτικές προτάσεις. Η Mentor έχει ήδη αναπτύξει ένα λειτουργικό πρωτότυπο και ισχυρίζεται ότι έχει χτίσει το μεγαλύτερο AGV στον κόσμο, για λογαριασμό της Boeing Co.

3.4 Μελλοντικές τάσεις

Τα AGVs επηρεάζονται σε μικρό βαθμό από τις παρεμβάσεις του περιβάλλοντος. Τα οχήματα χωρίς οδηγό μπορούν να δουλεύουν σε οποιοδήποτε καιρό ή άλλες δύσκολες συνθήκες, 24 ώρες τη μέρα. Επίσης, τα AGVs μπορούν να μεταφέρουν φορτία με μεγάλη ακρίβεια και απόδοση και μπορούν να διεκπεραιώσουν εργασίες, που θεωρούνται επικίνδυνες για τον άνθρωπο. Τέλος, η ασφάλεια είναι ευκολότερο να διατηρηθεί σε έναν αυτοματοποιημένο τερματικό σταθμό.

Τα AGVs προσφέρουν μια ασφαλή, αξιόπιστη εναλλακτική για τους ανθρώπους, χειριστές σε περιβάλλοντα, όπου οι απαιτήσεις μεταφοράς υλικού είναι καλά ορισμένες και σταθερές. Οι διαδρομές, ωστόσο, είναι δύσκολο και ακριβό να αλλάξουν, ειδικά στις περιπτώσεις καθοδήγησης με καλώδιο. Σε δυναμικά περιβάλλοντα τα οχήματα με αμετάβλητες διαδρομές μπορεί να μην είναι και τόσο πρακτικά. Για το λόγο αυτό, μια νέα γενιά οχημάτων διαχείρισης φορτίων αναπτύσσεται, η οποία είναι πιο αυτόνομη από τα AGVs του παρελθόντος. Αυτές οι μηχανές δεν απαιτούν μια διαδρομή καθοδήγησης μόνιμα εγκατεστημένη, αλλά βρίσκουν τη σωστή διαδρομή με μέσα, όπως οι εσωτερικοί χάρτες και η περιοδική ανίχνευση σημαδιών στο χώρο, η θέση των οποίων είναι γνωστή. Είναι αναμενόμενο ότι με την πάροδο του χρόνου, πολύπλοκα και «έξυπνα» συστήματα ελέγχου θα ενσωματωθούν στα AGVs και θα επιτρέπουν περισσότερο αυτόνομες εργασίες και καλύτερη απόδοση.

ΚΕΦΑΛΑΙΟ 4

4. Περιγραφή του περιβάλλοντος του τερματικού σταθμού.

4.1 Διάταξη του τερματικού σταθμού και συμβάσεις για τις κινήσεις των AGVs

Στη συγκεκριμένη μελέτη προτείνεται το μοντέλο ενός τερματικού σταθμού containers, που χρησιμοποιεί εξελιγμένους γερανούς και AGVs για την αυτοματοποιημένη φόρτωση και εκφόρτωση των containers. Το σχέδιο του αυτοματοποιημένου τερματικού, που χρησιμοποιείται στην προσομοίωση, φαίνεται στην εικόνα 4.1.1. Οι διαστάσεις του είναι 170m πλάτος και 570m μήκος (96.900m^2).

Το τερματικό αποτελείται από 4 κύρια κομμάτια. Κάθε κομμάτι έχει 4 αποθηκευτικές στοίβες και κάθε στοίβα έχει 10 στήλες και 5 γραμμές. Υποθέτουμε ότι κάθε κελί στοίβας μπορεί να χρησιμοποιηθεί για την αποθήκευση τριών containers διαστάσεων 40΄x 8,6΄ ή έξι containers διαστάσεων 20΄ x 8,6΄ . Οπότε, η χωρητικότητα του τερματικού είναι: 4x4x50x3=2400 FEUs (Forty foot Equivalent Unit).

Κάθε στοίβα χωρίζεται από την άλλη με ένα δρόμο, στον οποίο θα αναφερόμαστε σαν ***δρόμο εργασίας.*** Ένας διπλός δρόμος χωρίζει τα κύρια κομμάτια μεταξύ τους και θα αναφερόμαστε σε αυτόν σαν ***δρόμο διέλευσης***. Επίσης, δρόμοι διέλευσης είναι και οι δρόμοι δεξιά και αριστερά του τερματικού, καθώς και οι δρόμοι προσκείμενοι στην πύλη και στους γερανούς του πλοίου.

- Για να εμποδίσουμε την κυκλοφοριακή συμφόρηση και το μπλοκάρισμα υποθέτουμε ότι οι δρόμοι εργασίας χρησιμοποιούνται μόνο για δραστηριότητες αποθήκευσης και ανάκτησης containers, ενώ οι δρόμοι διέλευσης χρησιμοποιούνται για να προσεγγιστεί ένας δρόμος εργασίας, η πύλη και οι γερανοί του πλοίου.
- Επίσης, υποθέτουμε ότι οι δρόμοι εργασίας είναι μονόδρομοι και έχουν κατεύθυνση προς τα δεξιά όταν βρισκόμαστε δεξιά από τον μεσαίο δρόμο διέλευσης και αριστερά στην αντίθετη περίπτωση. Αυτή η σύμβαση γίνεται για δυο λόγους :

  (i):Ο πρώτος λόγος είναι ότι με αυτόν τον τρόπο γνωρίζουμε εκ των προτέρων τη διαδρομή που θα ακολουθήσει το όχημα για να προσεγγίσει ένα συγκεκριμένο container. Αυτή η διαδρομή είναι συνεχής και το όχημα δε χρειάζεται να κάνει επιτόπου στροφή μέχρι το τέλος της κίνησης.

  (ii): Ο δεύτερος λόγος είναι η ασφάλεια μέσα στον τερματικό σταθμό, που είναι ένας πολύ σημαντικός παράγοντας. Εάν συμβεί μια βλάβη σε ένα δρόμο εργασίας, είναι δυνατόν να σταματήσουν όλες οι δραστηριότητες σε αυτόν τον δρόμο και να προσεγγιστεί το όχημα από επισκευαστικό συνεργείο, χωρίς να υπάρχει κίνδυνος ατυχήματος από άλλα κινούμενα οχήματα.
- Όσον αφορά τους κατακόρυφους δρόμους διέλευσης, πρέπει να πούμε ότι είναι διπλής κατεύθυνσης και τα AGVs κινούνται στο δεξί ρεύμα όταν ανεβαίνουν και στο αριστερό όταν κατεβαίνουν.
- Υποθέτουμε ότι κάθε στοίβα εξυπηρετείται από μόνο ένα γερανό και αυτός ο γερανός φορτώνει/ εκφορτώνει τα AGVs από και προς το δρόμο εργασίας, που βρίσκεται από κάτω από τη στοίβα. Στον τερματικό σταθμό υπάρχουν επίσης, τέσσερις γερανοί πλοίου. Οι γερανοί στις στοίβες είναι υπεύθυνοι για τη

φόρτωση/ εκφόρτωση των AGVs, ενώ οι γερανοί πλοίου είναι υπεύθυνοι για τη φόρτωση/ εκφόρτωση containers προς/ από το πλοίο και για τη φόρτωση/ εκφόρτωση προς/ από τα AGVs. Οι δύο γερανοί πλοίου, που βρίσκονται στα αριστερά εξυπηρετούν τα AGVs που φορτώνουν/ εκφορτώνουν containers στο αριστερό μέρος του τερματικού (αριστερά από τον μεσαίο δρόμο διέλευσης). Το αντίστοιχο συμβαίνει για τους δυο γερανούς δεξιά.

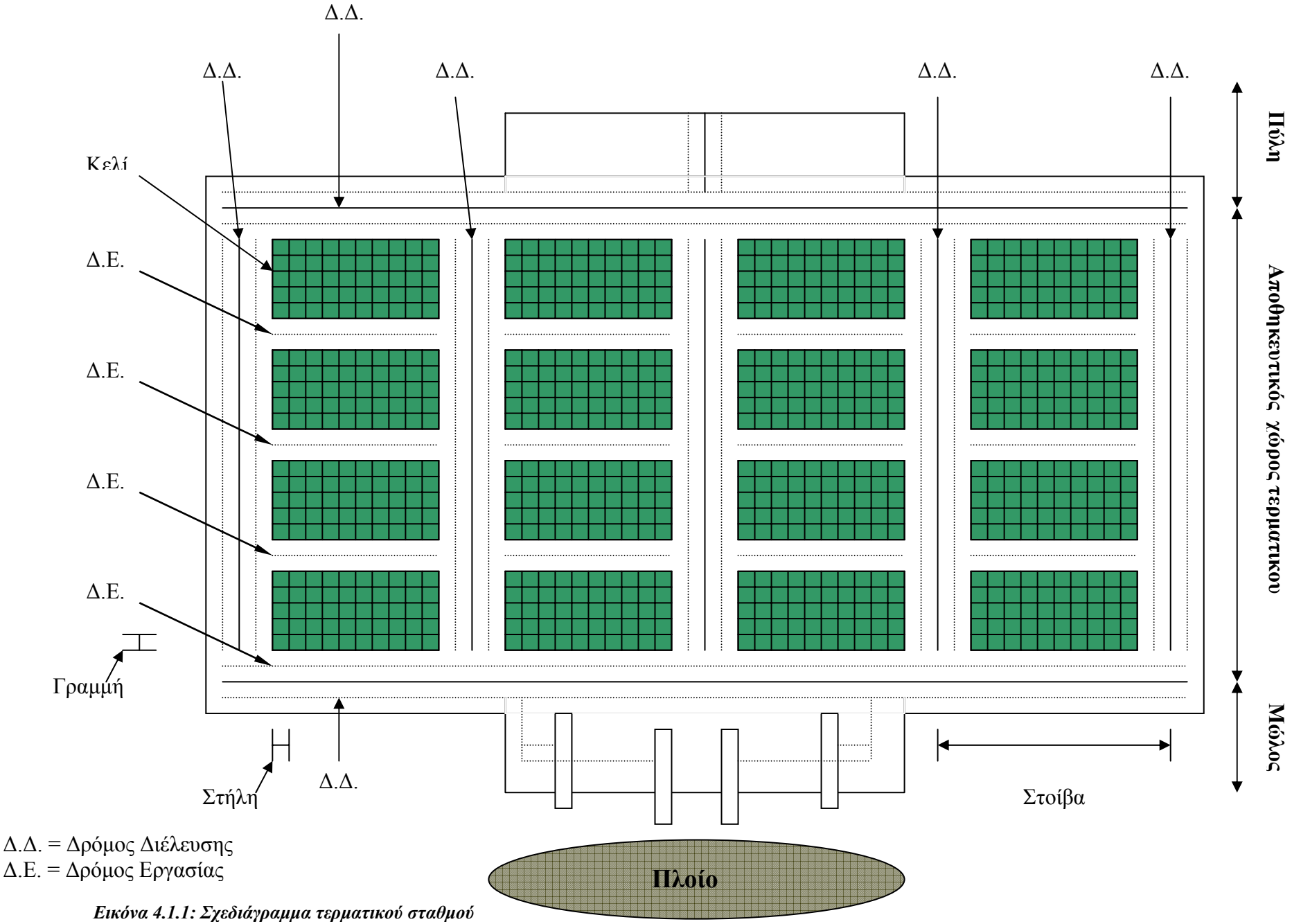
- Υπάρχουν δυο ουρές μπροστά στους γερανούς. Μια ουρά για τους δυο γερανούς δεξιά και άλλη μια για αυτούς αριστερά. Κάθε AGV εξυπηρετείται σύμφωνα με τη σειρά άφιξης στην ουρά και επιλέγει το γερανό, που είτε δεν εξυπηρετεί κανένα AGV ή ολοκληρώνει γρηγορότερα την εξυπηρέτηση προηγούμενου οχήματος.

Οι παραπάνω κανόνες για τις κινήσεις των AGVs δεν είναι μοναδικοί. Βασίζονται στη διαίσθηση και μπορεί να μην εξασφαλίζουν τις βέλτιστες συνθήκες λειτουργίας.

Στην ανάλυση και εκτίμηση του τερματικού θεωρούμε τα παρακάτω χαρακτηριστικά για τον εξοπλισμό του:

- Για τον μέσο χρόνο, που χρειάζονται οι γερανοί του τερματικού για φόρτωση/ εκφόρτωση, λάβαμε την τιμή *100sec* για την ευκολότερη διεξαγωγή συμπερασμάτων, καθώς και την τιμή *900sec* για περισσότερο ρεαλιστικά αποτελέσματα.
- Μέσος χρόνος για τους γερανούς στο μόλο για φόρτωση/ εκφόρτωση: 48 sec που αντιστοιχεί σε 75 κινήσεις την ώρα. Θεωρούμε ότι έχουμε αρκετά γρήγορους γερανούς στο μόλο, γιατί δεν θέλουμε να έχουμε καθυστερήσεις εξαιτίας αυτών, αλλά να ελέγξουμε τις εργασίες μέσα στον τερματικό σταθμό.
- Ταχύτητα για τα φορτωμένα AGVs: 8.28 χιλιόμετρα/ ώρα.
- Ταχύτητα για τα άδεια AGVs: 16.56 χιλιόμετρα/ ώρα.

Ακολουθεί σχεδιάγραμμα του αυτοματοποιημένου τερματικού σταθμού.



Δ.Δ. = Δρόμος Διέλευσης
Δ.Ε. = Δρόμος Εργασίας

Εικόνα 4.1.1: Σχεδιάγραμμα τερματικού σταθμού

4.2 Έλεγχος για την κίνηση των AGVs

Ο έλεγχος για την κίνηση των AGVs πρέπει να αποτρέπει κάθε πιθανή σύγκρουση μεταξύ των οχημάτων μέσα στον τερματικό σταθμό, καθώς και την περίπτωση να βρεθεί ένα ή περισσότερα οχήματα σε αδιέξοδο.

Σύγκρουση μεταξύ δύο η περισσοτέρων AGVs έχουμε στις εξής περιπτώσεις:

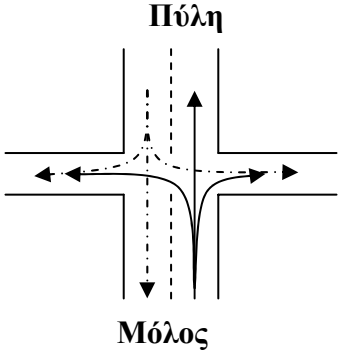
1. Όταν το AGV βρίσκει μπροστά του άλλο AGV, που κινείται με μικρότερη ταχύτητα ή είναι σταματημένο.

Τέτοιες συγκρούσεις είναι δυνατόν να συμβούν, επειδή τα φορτωμένα AGVs έχουν μικρότερη ταχύτητα από τα ξεφόρτωτα. Επίσης, κάποιο AGV, κατά την πορεία του, μπορεί να συναντήσει ένα άλλο AGV που φορτώνει ή ξεφορτώνει container μέσα στον τερματικό σταθμό. Σε αυτήν την περίπτωση, το AGV που κινείται με μεγαλύτερη ταχύτητα, όταν πλησιάσει και περάσει μια απόσταση ασφαλείας (π.χ. 3m) επιβραδύνει και αποκτάει την ταχύτητα του μπροστινού AGV.

2. Άφιξη σε διασταύρωση από διαφορετικούς δρόμους ταυτόχρονα

Σε αυτή την περίπτωση, δεδομένου ότι γνωρίζουμε εκ των προτέρων τις πιθανές κατευθύνσεις των AGVs σε κάθε διασταύρωση, ξεχωρίζουμε τρεις υποπεριπτώσεις: (*i*)την περίπτωση των διασταυρώσεων στον κεντρικό δρόμο διέλευσης, (*ii*)την περίπτωση των διασταυρώσεων αριστερά από τον κεντρικό δρόμο διέλευσης και (*iii*)την περίπτωση των διασταυρώσεων στα δεξιά του. Αυτή η λογική ελέγχου, παρότι είναι περίπλοκη, μας δίνει τη δυνατότητα να δίνουμε προτεραιότητα, κάθε φορά, στο όχημα που είναι ξεφόρτωτο και άρα πιο γρήγορο. Με τον τρόπο αυτό, ελαχιστοποιούμε την πιθανότητα να καθυστερεί ένα ξεφόρτωτο όχημα πίσω από ένα φορτωμένο.

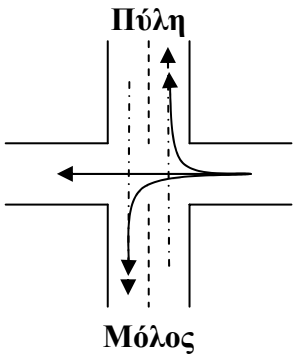
Με αυτή τη λογική, για την πρώτη περίπτωση, η οποία φαίνεται και στην εικόνα 4.2.1, προτεραιότητα έχουν τα οχήματα που έρχονται από τον μόλο (καθώς αυτά είναι σίγουρα άδεια).



Εικόνα 4.2.1: **Πιθανές κατευθύνσεις των AGVs σε διασταύρωση του Κ.Δ.Δ.**

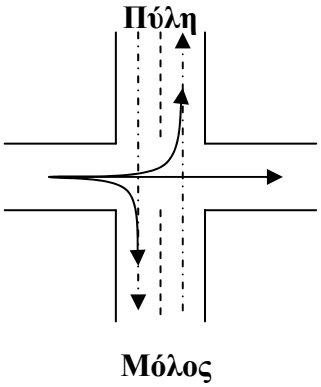
Στη δεύτερη περίπτωση, για τις διασταυρώσεις στα αριστερά του κεντρικού δρόμου διέλευσης, εικόνα 4.2.2, προτεραιότητα έχουν οι κατακόρυφες κατευθύνσεις. Εδώ δε

δίνεται προτεραιότητα στα AGVs, που κινούνται πιο γρήγορα, αλλά ωστόσο δεν υπάρχει περίπτωση να «κολλήσει» ένα γρήγορο AGV πίσω από ένα αργό.



Εικόνα 4.2.2: **Πιθανές κατευθύνσεις των AGVs σε διασταύρωση αριστερά του Κ.Δ.Δ.**

Στην τρίτη περίπτωση, για τις διασταυρώσεις δεξιά του κεντρικού δρόμου διέλευσης, δίνεται προτεραιότητα στις κατακόρυφες κατευθύνσεις, όμοια με την παραπάνω περίπτωση.



Εικόνα 4.2.3: **Πιθανές κατευθύνσεις των AGVs σε διασταύρωση δεξιά του Κ.Δ.Δ.**

4.3 Παράδειγμα λειτουργίας αλγορίθμου

Έχοντας εξετάσει τις συμβάσεις για τις κινήσεις των AGVs μέσα στον τερματικό σταθμό ακολουθεί ένα απλό παράδειγμα που δείχνει τη βασική λειτουργία του αλγορίθμου που χρησιμοποιήθηκε.

Στην εικόνα 4.3.1, που ακολουθεί, είναι χαραγμένη με βέλη η πορεία που θα ακολουθήσει το AGV, στο οποίο έχει γίνει ανάθεση για να ανακτήσει ένα τυχαίο container, που για το συγκεκριμένο παράδειγμα βρίσκεται στη θέση με κόκκινο χρώμα και να το φέρει στο χώρο φόρτωσης του καραβιού.

Θα περιγραφούν δύο περιπτώσεις. Στην πρώτη, δε θα ληφθούν υπόψη τυχόν επιπλέον καθυστερήσεις, πέραν αυτών λόγο των διασταυρώσεων και της αναμονής στην ουρά

μπροστά στους γερανούς του λιμανιού. Ενώ στη δεύτερη, θα συμπεριληφθούν και καθυστερήσεις που οφείλονται σε συνάντηση:

i. Με σταματημένο AGV(το οποίο βρίσκεται σε διαδικασία φόρτωσης).

ii. Με φορτωμένο προπορευόμενο AGV, που κινείται πιο αργά.

Στην πρώτη περίπτωση η αρχική θέση του AGV είναι το κέντρο των γερανών του μόλου. Πιο αναλυτικά για το AGV γίνεται προσομοίωση στις παρακάτω ενέργειες:

- Κινείται στον κατακόρυφο άξονα μέχρι το ύψος που είναι τοποθετημένο το container προς ανάκτηση.
- Κινείται στον οριζόντιο άξονα μέχρι την ακριβή θέση που βρίσκεται το container.
- Γίνεται φόρτωση του container.
- Κινείται στον οριζόντιο άξονα μέχρι να φτάσει στον αριστερό δρόμο διέλευσης.
- Ακολουθεί κατακόρυφη πορεία κατεβαίνοντας στο επίπεδο των γερανών του λιμανιού.
- Και τελικά κινείται οριζόντια προς τους γερανούς, όπου και ξεφορτώνεται το container.

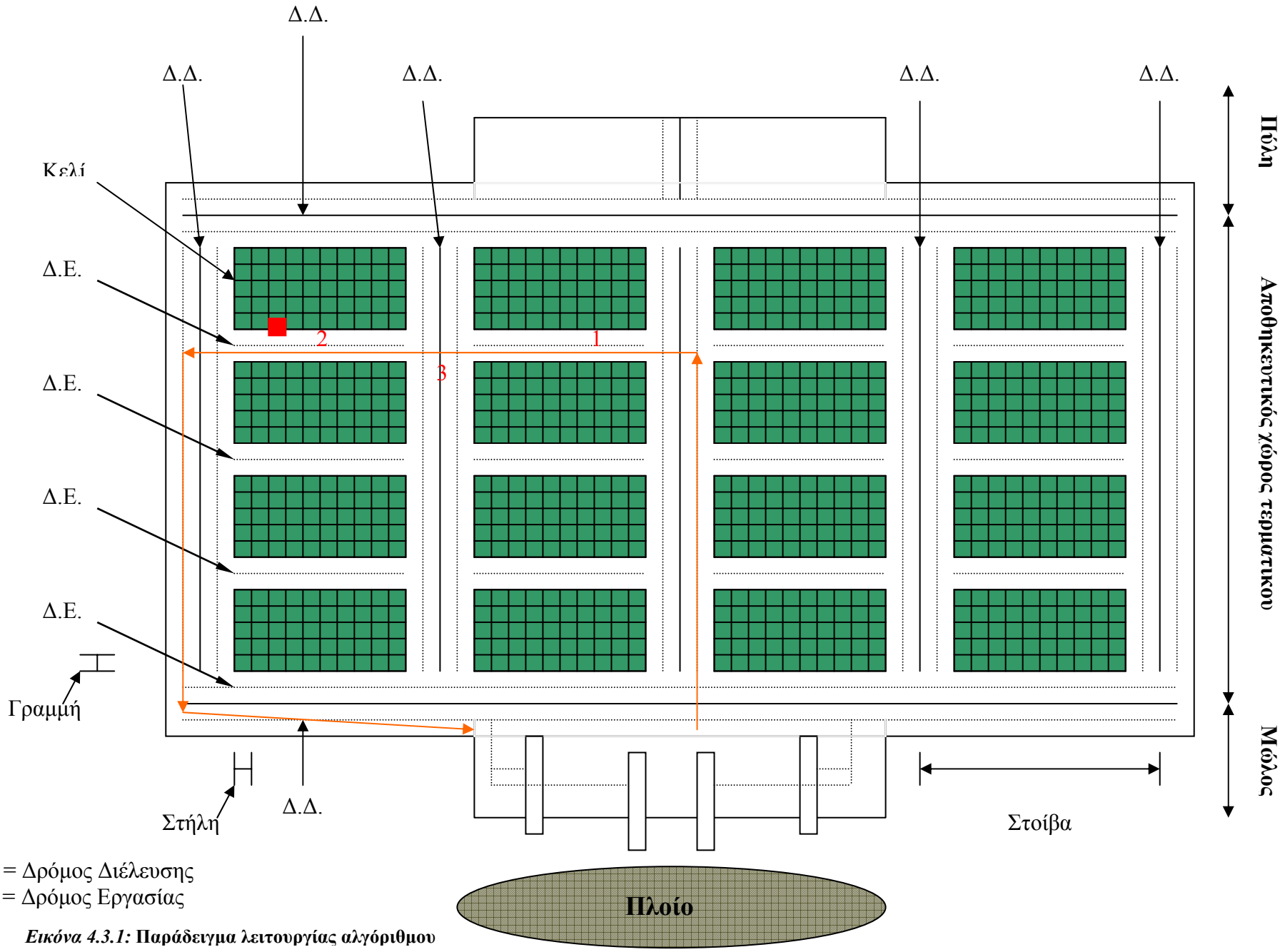
Στη δεύτερη περίπτωση οι κινήσεις και οι διαδικασίες, που ακολουθούν τα AGVs είναι ίδιες με την προηγούμενη. Η μόνη διαφοροποίηση είναι οι δύο περιπτώσεις συναντήσεων, που προαναφέρθηκαν και στο συγκεκριμένο παράδειγμα συμβαίνουν στα **σημεία 1**και **2,** όπως είναι σημειωμένα στην εικόνα 4.3.1. Τα AGVs, για λόγους συντομίας, τα διαχωρίζουμε σε :

AGV(i): αυτά που εισέρχονται στον τερματικό σταθμό και

AGV(j): αυτά που ήδη κινούνται εντός του σταθμού.

Το AGV(i) θα συναντήσει στο δρόμο του το προπορευόμενο AGV(j) στο σημείο **1**, είτε όταν το δεύτερο φορτώνει, είτε όταν αυτό είναι φορτωμένο και κατευθύνεται προς το λιμάνι. Αυτό σημαίνει ότι το AGV(i) θα καθυστερήσει όσο χρόνο χρειάζεται για να ολοκληρώσει τη φόρτωσή του το AGV(j). Επιπλέον, καθυστερεί ακολουθώντας το με ταχύτητα μισή από αυτή που θα είχε αν ακολουθούσε την πορεία του χωρίς εμπόδιο. Αυτό συμβαίνει μέχρι το AGV(j) να συναντήσει το δρόμο διέλευσης (**σημείο 3**), οπότε και θα φύγει προς τα λιμάνι και θα σταματήσει να το εμποδίζει.

Μια αντίστοιχη διαδικασία συμβαίνει όταν το συναντήσει στο **σημείο 2.** Σε αυτή την περίπτωση, δε θα χρειαστεί το AGV(i) να ακολουθεί το AGV(j), μέχρι το δεύτερο να φτάσει σε δρόμο εργασίας, γιατί θα έχει φτάσει ήδη στον προορισμό του και θα αρχίσει τη φόρτωση του container που του έχει ανατεθεί .Οι ίδιες καθυστερήσεις υφίστανται συμμετρικά για τις στοίβες του τερματικού που βρίσκονται στα δεξιά του μεσαίου δρόμου διέλευσης.



Δ.Δ. = Δρόμος Διέλευσης
Δ.Ε. = Δρόμος Εργασίας

Εικόνα 4.3.1: **Παράδειγμα λειτουργίας αλγόριθμου**

ΚΕΦΑΛΑΙΟ 5

5. Σενάρια προσομοίωσης

Στο χωρίο αυτό, εφαρμόζονται διάφορα σενάρια για το μοντέλο της προσομοίωσης, που περιγράφηκε σε προηγούμενο κεφάλαιο. Θα εξεταστούν δύο περιπτώσεις. Κατά την πρώτη περίπτωση, θεωρούμε τον τερματικό σταθμό γεμάτο με containers και γι' αυτό χρησιμοποιούνται μόνο AGVs και καθόλου φορτηγά. Κατά τη δεύτερη περίπτωση, θεωρούμε τον τερματικό σταθμό μερικώς συμπληρωμένο με containers, συγκεκριμένα με 450 containers στον αριθμό. Στη δεύτερη περίπτωση, τα φορτηγά θα μπορούσαν να χρησιμοποιηθούν σε συνδυασμό με τα AGVs, για τη συμπλήρωση των άδειων κελιών του τερματικού απ' έξω. Το μοντέλο, που έχουμε κατασκευάσει, έχει τη δυνατότητα να προσομοιώσει παράλληλα τις εργασίες των φορτηγών και των AGVs, καθώς και τη διαδικασία διαχείρισης των εισερχόμενων φορτηγών από την πύλη, η οποία είναι παρόμοια με την εργασία των γερανών του μόλου (αριθμός εξυπηρετούντων, ουρές ,χρόνοι εξυπηρέτησης). Ωστόσο, εκτελώντας προσομοιώσεις και για την περίπτωση εισαγωγής containers από την πύλη με φορτηγά, παρατηρήσαμε ότι αυξάνει κατά πολύ ο υπολογιστικός φόρτος, χωρίς ουσιαστική μεταβολή των τελικών αποτελεσμάτων. Έτσι, συνεχίστηκαν οι προσομοιώσεις και για τη δεύτερη περίπτωση, μόνο με τη χρήση AGVs και καθόλου φορτηγών.

Σενάριο Α: πολιτική τυχαίας ανάθεσης

Κατά το σενάριο αυτό, χρησιμοποιείται η πολιτική τυχαίας ανάθεσης **(Random)** των AGVs σε containers, που σημαίνει ότι τα τελευταία βρίσκονται αποθηκευμένα τυχαία μέσα στη γυάρδα και όχι σε συγκεκριμένες θέσεις. Αυτή η πολιτική εφαρμόστηκε πρώτη φορά από την εταιρία August Design-Sealand.

Όταν γίνεται ανάθεση σε ένα AGV ένα τυχαίο container, γνωρίζουμε τις συντεταγμένες του συγκεκριμένου container μέσα στη γυάρδα και επομένως και τη **διαδρομή,** που θα πρέπει να ακολουθήσει το AGV για να φτάσει ως εκεί. Για την τυχαία ανάθεση containers, προστέθηκε ένας πίνακας storage[1,160], του οποίου όλα τα στοιχεία έχουν την τιμή 15, υποθέτοντας ότι σε κάθε κελί της εκάστοτε στοίβας τοποθετούνται τρία containers. Ο τελευταίος πίνακας αντιστοιχεί σε όλα τα containers, 2400 τον αριθμό, που είναι αρχικά τοποθετημένα στον τερματικό σταθμό, θεωρώντας ότι είναι γεμάτος. Κάθε θέση του πίνακα είναι αντίστοιχη με ένα κελί της γυάρδας. Για τη δεύτερη περίπτωση, μερικής συμπλήρωσης της γυάρδας, τα πρώτα 30 στοιχεία του πίνακα storage παίρνουν την τιμή 15, ενώ όλα τα υπόλοιπα στοιχεία του συμπληρώνονται με την τιμή μηδέν.

Κάθε φορά που επιλέγεται τυχαία ένα container, πραγματοποιείται έλεγχος στο αντίστοιχο στοιχείο του πίνακα για τιμή μεγαλύτερη του μηδενός, που σημαίνει ότι υπάρχει container στην αντίστοιχη θέση της γυάρδας. Αν υπάρχει, τότε μειώνεται κατά ένα αυτή η τιμή και συνεχίζει ο αλγόριθμος. Σε αντίθετη περίπτωση, επιλέγεται τυχαία το επόμενο container. Αυτή η διαδικασία συνεχίζεται έως ότου γίνουν όλα τα στοιχεία του πίνακα μηδέν, οπότε θα έχει αδειάσει ο τερματικός σταθμός και θα τερματίσει ο αλγόριθμος. Ο **χρόνος** που χρειάζεται για να ολοκληρωθεί η παραπάνω διαδικασία είναι το βασικό μέγεθος, για το οποίο εφαρμόστηκε το παραπάνω και τα ακόλουθα σενάρια με σκοπό τη βελτιστοποίηση του.

»»»

Στην προσπάθειά μας να βελτιστοποιήσουμε την απόδοση του τερματικού, χωρίς να αυξήσουμε τον αριθμό των AGVs (καθώς η αξία αγοράς τους είναι αρκετά υψηλή), χρησιμοποιήσαμε δύο μη ρεαλιστικά σενάρια κατά τα οποία δεν ελέγχαμε τις συγκρούσεις μεταξύ των οχημάτων. Τα AGVs κινούνταν ελεύθερα και στην περίπτωση που συναντούσαν στην πορεία τους άλλο όχημα, διασταυρώνονταν χωρίς σύγκρουση. Παρατηρήσαμε, ότι για μικρό αριθμό AGVs, όπου οι συγκρούσεις είναι εκ των πραγμάτων λίγες, η απόδοση του τερματικού σταθμού ήταν παρόμοια με αυτή του σεναρίου Α(πολιτική τυχαίας ανάθεσης). Όμως, καθώς αυξανόταν ο αριθμός των AGVs, αυξανόταν και το περιθώριο βελτίωσης της απόδοσης σε σχέση με το σενάριο Α. Έτσι, βασιζόμενοι στη διαίσθηση, προσπαθήσαμε να βρούμε μια άλλη πολιτική ανάθεσης για τα AGVs, που θα μείωνε τις συγκρούσεις και συνεπώς θα αύξανε την απόδοση. Σύμφωνα πάλι με τα αποτελέσματα του σεναρίου Α, παρατηρήσαμε ότι οι συγκρούσεις μεταξύ των AGVs συμβαίνουν κυρίως λόγω της αναμονής στις ουρές των γερανών και όχι λόγω των κινήσεων μέσα στον τερματικό σταθμό. Έτσι, προσπαθήσαμε να ανιχνεύσουμε εκ των προτέρων τις πιθανές συγκρούσεις στις ουρές και να μειώσουμε το μέγεθος των τελευταίων με κατάλληλη ανάθεση των AGVs.

Γνωρίζοντας τα παραπάνω στοιχεία, μπορούμε να υπολογίσουμε το χρόνο T_{2i}, που θα χρειαστεί ένα AGV για να ολοκληρώσει τη **διαδρομή** του**,** συμπεριλαμβανομένου και του χρόνου φόρτωσης, μέχρι να φτάσει στην ουρά του γερανού. Για τον υπολογισμό του χρόνου T_2 θεωρούμε ότι δεν υπάρχουν συγκρούσεις εντός του τερματικού σταθμού. Στη συνέχεια γίνεται σύγκριση με τους αντίστοιχους χρόνους T_{2j} των υπολοίπων AGVs, που κινούνται στον τερματικό σταθμό.

Επίσης, υπολογίζονται οι χρόνοι καθυστέρησης T_1 των AGVs, που οφείλονται σε συνάντηση/ σύγκρουση (όπως προαναφέρθηκαν σε παράδειγμα) με:

- σταματημένο AGV(το οποίο βρίσκεται σε διαδικασία φόρτωσης).
- φορτωμένο προπορευόμενο AGV, που κινείται πιο αργά (βλ. εικόνα 4.3.1, σημεία 1, 2).

Η πρώτη συνάντηση έχει σαν συνέπεια τη δεύτερη και επηρεάζουν το συνολικό χρόνο ολοκλήρωσης της διαδρομής.

Η απόλυτη διαφορά αυτών των χρόνων

$$T_3 = \left| T_{2i} - T_{2j} \right|$$

μας δίνει τη χρονική απόσταση του οχήματος *i*(που εισέρχεται στον τερματικό σταθμό) από τα οχήματα *j* (που ήδη κινούνται εντός του σταθμού). Στην περίπτωση που λαμβάνονται υπόψη οι συγκρούσεις ο χρόνος T_3 γίνεται:

$$T_4 = \left| (T_{2i} - T_{2j}) + T_1 \right| \quad .$$

Με βάση τα παραπάνω έχουμε τα ακόλουθα σενάρια:

Σενάριο Α1: ελαχιστοποίηση της ουράς στους γερανούς ($\min T_3 \geq 24$ ***)***

Στο σενάριο A_1 θέλουμε ο ελάχιστος από τους χρόνους T_3 να είναι ίσος ή μεγαλύτερος από το χρόνο που χρειάζεται ένας γερανός για να εξυπηρετήσει ένα AGV. Με αυτή τη συνθήκη προσπαθούμε να εξαλείψουμε το σχηματισμό ουράς στο σημείο φόρτωσης του καραβιού. Αν ο ελάχιστος από τους χρόνους T_3 δεν ικανοποιεί τη συνθήκη αυτή, γίνεται ανάθεση στο AGV ένα άλλο τυχαίο container και υπολογίζονται ξανά όλοι οι χρόνοι από την αρχή, έως ότου ικανοποιηθεί η συνθήκη. Με τον τρόπο αυτό αναμένεται να μειωθεί το μέγεθος της ουράς και να επιτευχθεί καλύτερη απόδοση του τερματικού σταθμού.

Σενάριο Α2: ελαχιστοποίηση του χρόνου αδράνειας των γερανών ($\min T_3 \leq 24$)

Στο σενάριο A_2 θέλουμε ο ελάχιστος από τους χρόνους T_3 να είναι ίσος ή μικρότερος από το χρόνο που χρειάζεται ένας γερανός για να εξυπηρετήσει ένα AGV. Με τον τρόπο αυτό, ναι μεν δεν μειώνεται το μέγεθος της ουράς, αλλά γίνεται προσπάθεια να υπάρχει συνεχής τροφοδοσία των γερανών, ώστε να μειωθεί ο χρόνος αδράνειας των γερανών και να αυξηθεί η απόδοση του τερματικού σταθμού.

»»»

Τα σενάρια A_1 και A_2 αποτελούν μη ρεαλιστικά σενάρια, όπως και προαναφέρθηκε. Ακολουθούν τα σενάρια A_3 και A_4, τα οποία εκφράζουν και το πραγματικό μοντέλο του τερματικού.

Σενάριο A_3: ελαχιστοποίηση της ουράς στους γερανούς, λαμβάνοντας υπόψη τις καθυστερήσεις **(** $\min T_4 \geq 24$ **)**

Το σενάριο A_3 ακολουθεί την ίδια λογική με το σενάριο A_1, με τη διαφορά ότι σε αυτή την περίπτωση, στο χρόνο T_3 υπολογίζονται και οι καθυστερήσεις λόγω των συγκρούσεων, T_1.

Σενάριο A_4: ελαχιστοποίηση του χρόνου αδράνειας των γερανών, λαμβάνοντας υπόψη τις καθυστερήσεις. **(** $\min T_4 \leq 24$ **)**

Το σενάριο A_4 ακολουθεί την ίδια λογική με το σενάριο A_2, με τη διαφορά ότι σε αυτή την περίπτωση στο χρόνο T_3 υπολογίζονται και οι καθυστερήσεις λόγω των συγκρούσεων, T_1.

»»»

Όλα τα παραπάνω σενάρια, Α, A_1, A_2, A_3, A_4, εφαρμόζονται τόσο για την περίπτωση γεμάτου τερματικού σταθμού, όσο και για την περίπτωση μερικής συμπλήρωσής του με 450 containers.

Μια άλλη παράμετρος, που διαφοροποιήθηκε, είναι ο χρόνος που χρειάζονται για φόρτωση οι γερανοί του τερματικού σταθμού. Αυτή η παράμετρος χρησιμοποιήθηκε για να εξεταστεί ο αλγόριθμος σε πιο ρεαλιστικές συνθήκες, δηλαδή πιο αργούς γερανούς τερματικού. Έτσι λοιπόν, εκτελέστηκαν προσομοιώσεις για 100sec και 900sec.

ΚΕΦΑΛΑΙΟ 6

6. Κριτήρια αξιολόγησης, αποτελέσματα

6.1 Κριτήρια αξιολόγησης

Το βασικότερο κριτήριο αξιολόγησης, που χρησιμοποιείται, είναι ο συνολικός χρόνος που χρειάζεται για να ολοκληρωθεί μία προσομοίωση. Άλλα κριτήρια αξιολόγησης, που χρησιμοποιήθηκαν, είναι οι χρόνοι αδράνειας των AGVs, καθώς και των γερανών του μόλου.

Χρόνος αδράνειας των AGVs ορίζεται το ποσοστό επί του συνολικού χρόνου, που ένα AGV καθυστερούσε λόγω σύγκρουσης, ή λόγω αναμονής στην ουρά των γερανών του μόλου, δηλαδή:

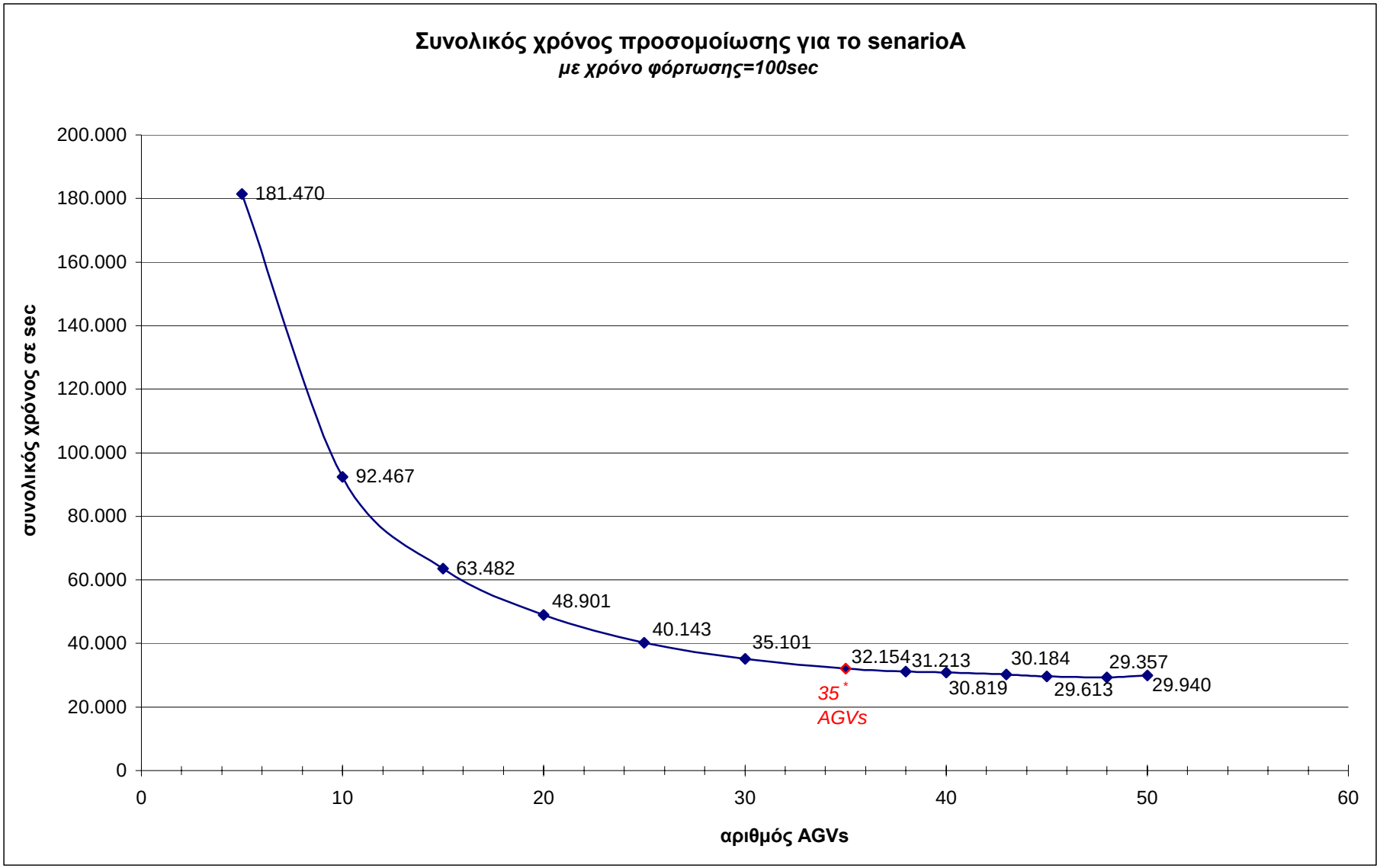
$$X.A.AGV = \frac{\text{Χρόνος Καθυστέρησης Από Συγκρούσεις} + \text{Χρόνος Αναμονής Σε Ουρά}}{\text{Συνολικός Χρόνος [ώρες]}}$$

Όμοια, χρόνος αδράνειας, για τους γερανούς του μόλου, ορίζεται το ποσοστό επί του συνολικού χρόνου, που οι γερανοί δεν εξυπηρετούν κάποιο όχημα, δηλαδή:

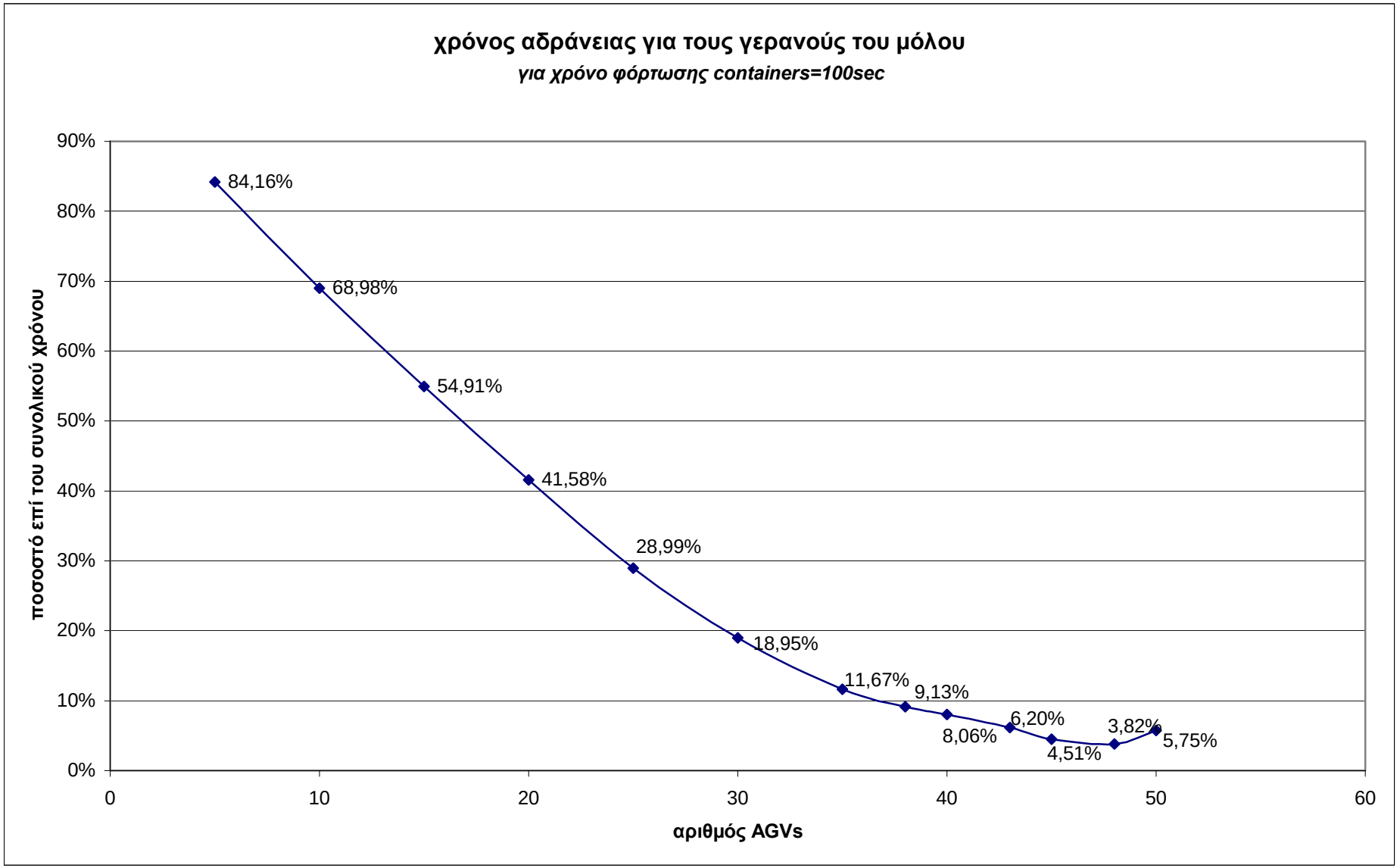
$$X.A.\text{Γερανών} = \frac{\text{Χρόνος Μη Λειτουργίας Γερανών}}{\text{Συνολικός Χρόνος [ώρες]}}$$

6.2 Αποτελέσματα & αξιολόγηση

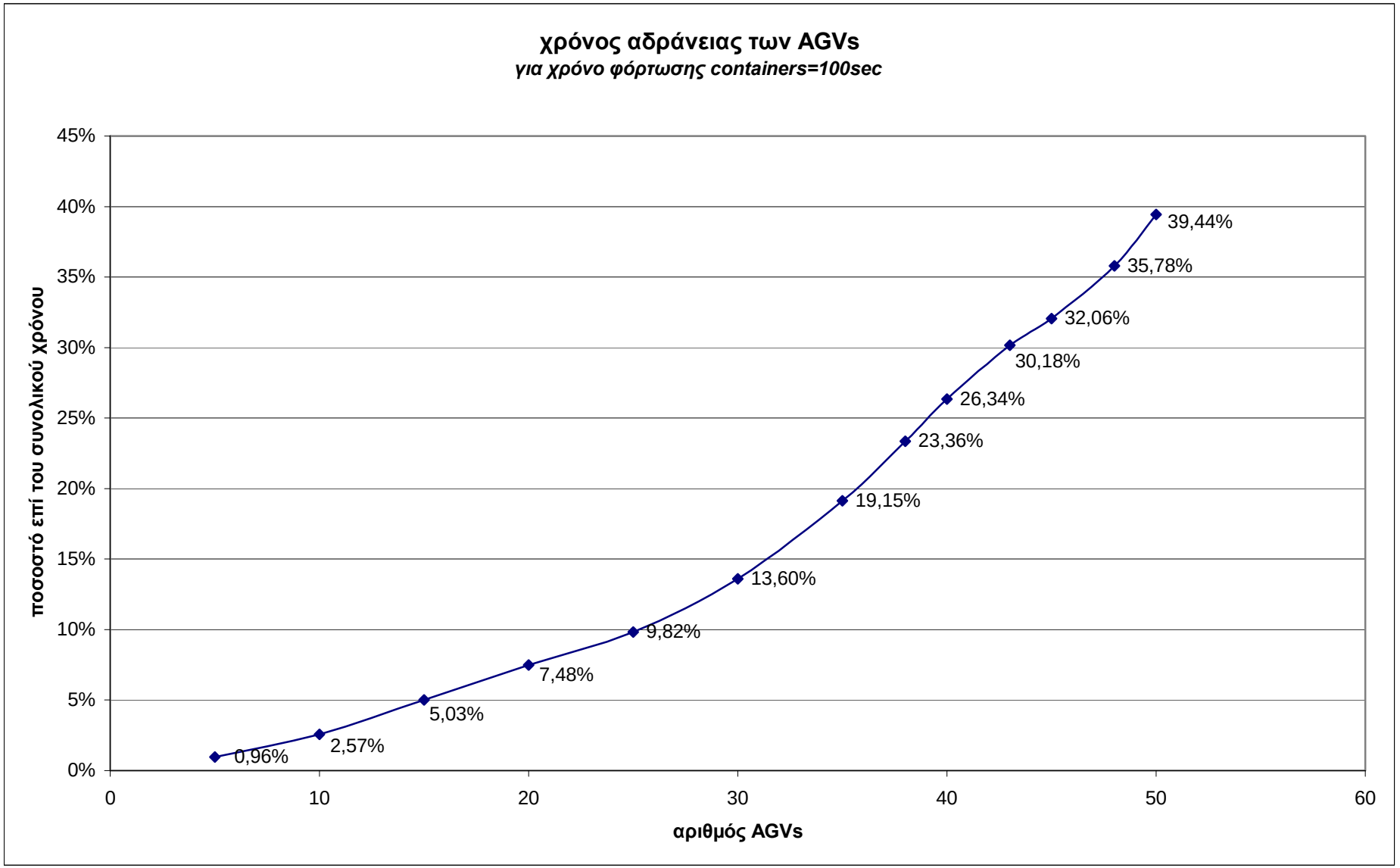
Κατά το σενάριο Α, πραγματοποιήθηκαν συνολικά 13 προσομοιώσεις (AGVs = 5,10,15,20,25,30,35,38,40,43,45,48,50) για την περίπτωση γεμάτου τερματικού σταθμού. Τα αποτελέσματα των προσομοιώσεων αυτών, για το συνολικό χρόνο προσομοίωσης και τους χρόνους αδράνειας των γερανών του μόλου και των AGV's, περιγράφονται στα *γραφήματα 1*, *2* και *3*. Όπως φαίνεται και στο *γράφημα 1*, ο συνολικός χρόνος προσομοίωσης μειώνεται με την αύξηση του αριθμού των AGVs. Έτσι, παρατηρούμε ότι για 5 AGVs ο αλγόριθμος απαιτεί 181.470sec (ή 50,41ώρες) για την ολοκλήρωση της προσομοίωσης. Ενώ για 50 AGVs απαιτούνται μόλις 29.940sec (ή 8,32ώρες). Από τις προσομοιώσεις, που εκτελέστηκαν κατά το σενάριο Α, διαπιστώθηκε ότι για μικρό αριθμό AGVs δεν είχαμε περιθώρια βελτίωσης της απόδοσης του τερματικού. Εμφανή διαφοροποίηση των αποτελεσμάτων μας, είχαμε μετά τα 35 AGVs. Γι' αυτό το λόγο, για τα υπόλοιπα σενάρια επιλέχτηκαν σαν αρχή τα 30 AGVs και ο αριθμός τους αυξάνεται ανά 5 μέχρι τα 50 AGVs. Στο *γράφημα 2* παρατηρούμε, όπως άλλωστε είναι αναμενόμενο, ότι με την αύξηση του αριθμού των AGVs ο χρόνος αδράνειας των γερανών του μόλου μειώνεται σημαντικά, καθώς τροφοδοτούνται συνέχεια με containers. Θα μπορούσε λοιπόν, να διεξαχθεί το συμπέρασμα ότι ο βέλτιστος αριθμός AGVs είναι ο μεγαλύτερος δυνατός, καθώς επιθυμούμε τη γρηγορότερη εξυπηρέτηση των πλοίων. Αυτό όμως, θα απαιτούσε μία υπέρογκη δαπάνη εξοπλισμού των τερματικών σταθμών των λιμανιών με AGVs, το κόστος των οποίων είναι αρκετά υψηλό. Όπως φαίνεται και από το *γράφημα 3* άλλωστε, ο χρόνος αδράνειας των AGVs αυξάνεται σημαντικά με την αύξηση του αριθμού τους. Συνεπώς, δεν είναι σωστό να χρησιμοποιήσουμε σαν κριτήριο αξιολόγησης μόνο τη γρηγορότερη εξυπηρέτηση των πλοίων, αλλά και τις δαπάνες που χρειάζεται το λιμάνι για τον εξοπλισμό του τερματικού του σταθμού.



Γράφημα 1: **Συνολικός χρόνος προσομοίωσης συναρτήσει του αριθμού AGVs για το σενάριο Α (περίπτωση γεμάτου τερματικού)**

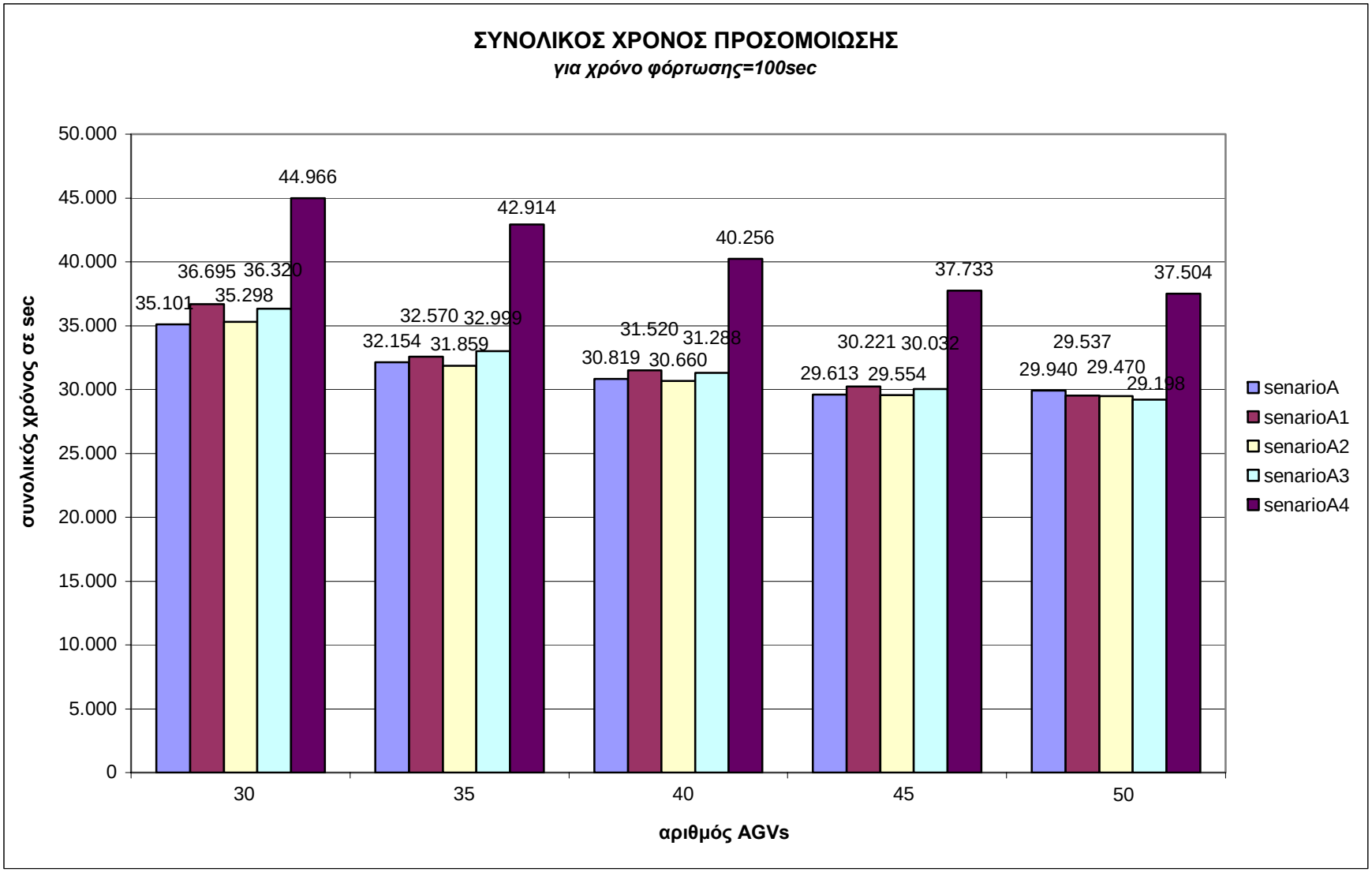


Γράφημα 2: **Χρόνος αδράνειας γερανών μόλου συναρτήσει αριθμού AGVs για το σενάριο Α (περίπτωση γεμάτου τερματικού)**

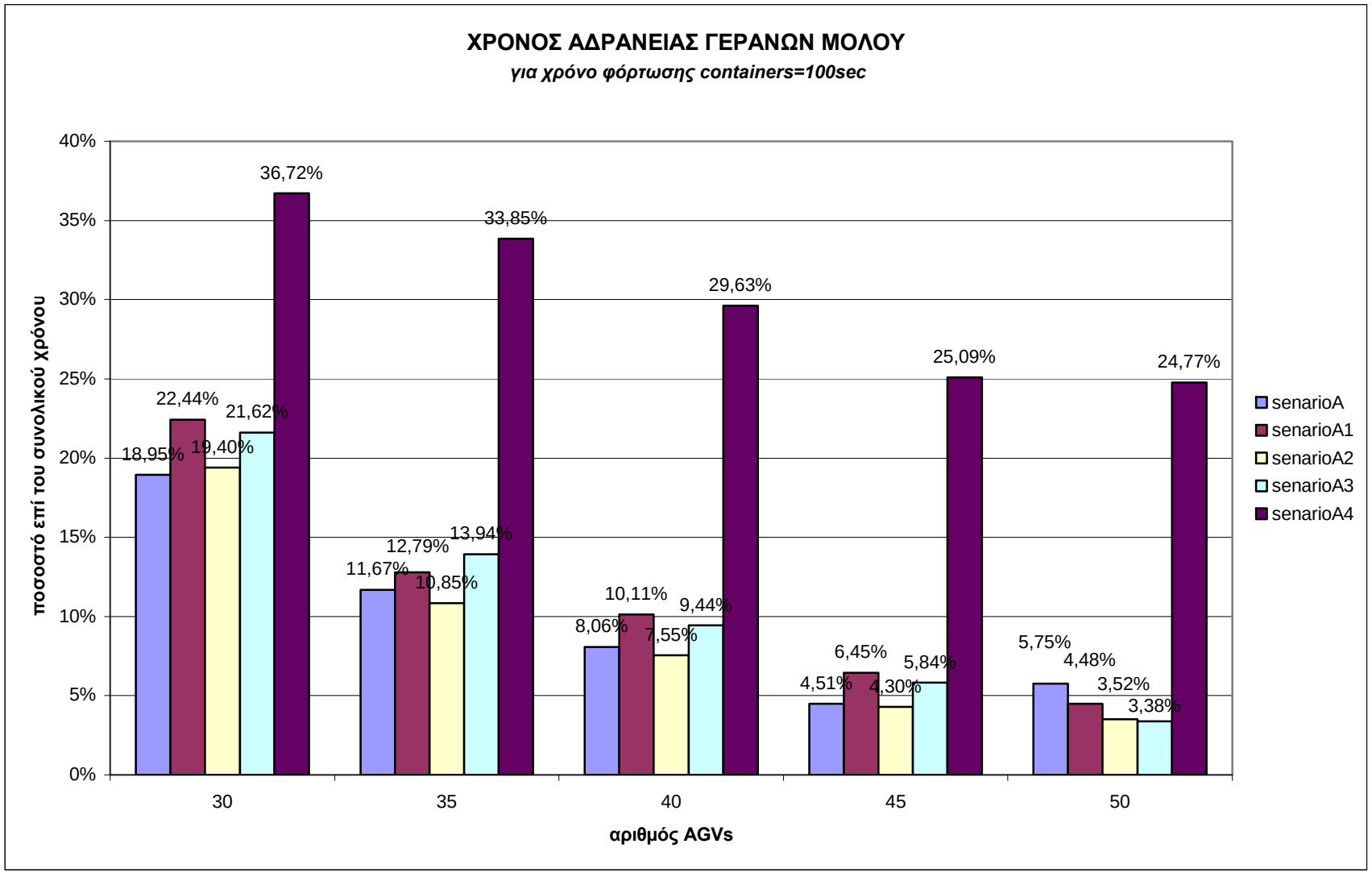


Γράφημα 3: **Χρόνος αδράνειας AGVs συναρτήσει αριθμού AGVs για το σενάριο Α(περίπτωση γεμάτου τερματικού)**

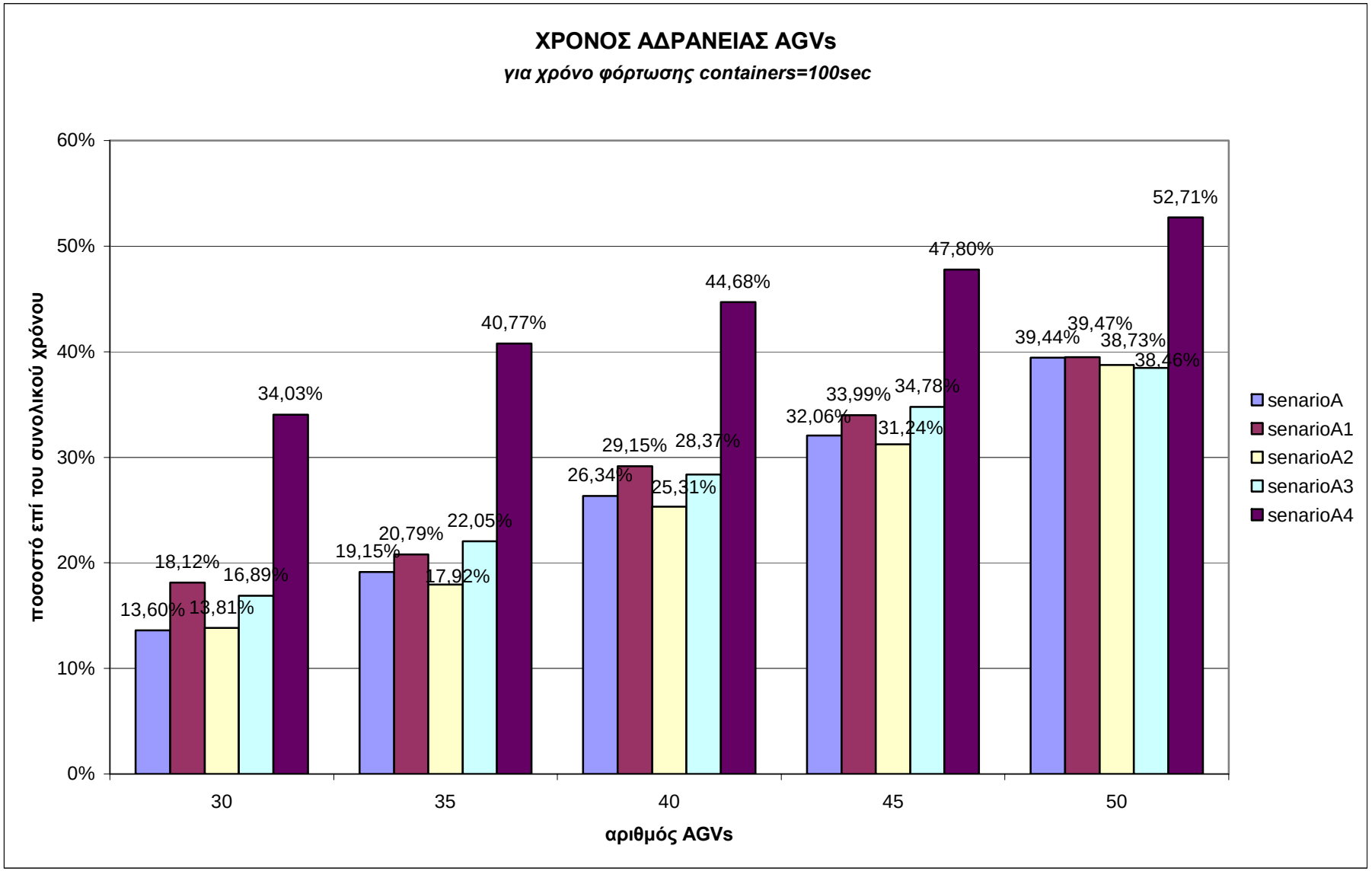
Παρακάτω παρουσιάζονται τα συγκριτικά αποτελέσματα των προσομοιώσεων για την περίπτωση του γεμάτου τερματικού σταθμού και για χρόνο φόρτωσης 100sec. Στο *γράφημα 4* παρουσιάζεται ο συνολικός χρόνος προσομοίωσης συναρτήσει του αριθμού των AGVs. Παρατηρούμε ότι τα σενάρια Α, A_1, A_2 και A_3, έχουν συγκρίσιμους συνολικούς χρόνους προσομοίωσης, ενώ το σενάριο A_4 παρουσιάζει για όλους τους αριθμούς των AGVs πολύ μεγάλους χρόνους. Σύμφωνα με αυτό το κριτήριο λοιπόν, απορρίπτεται το σενάριο A_4, σύμφωνα με το οποίο προσπαθούμε να μειώσουμε το χρόνο αδράνειας των γερανών, ενώ το βέλτιστο σενάριο, με χαμηλούς συνολικούς χρόνους προσομοίωσης για όλους τους αριθμούς των AGVs, φαίνεται να είναι το σενάριο Α(πολιτική τυχαίας ανάθεσης. Στο *γράφημα 5* παρουσιάζεται ο χρόνος αδράνειας των γερανών του μόλου συναρτήσει του αριθμού των AGVs. Στο γράφημα αυτό, τα καλύτερα αποτελέσματα δίνουν τα σενάρια Α (πολιτική τυχαίας ανάθεσης) και A_2 (προσπάθεια μείωσης του χρόνου αδράνειας των γερανών του μόλου, μη ρεαλιστικό σενάριο) με πολύ μικρές διαφορές στα ποσοστά χρόνων. Επειδή όμως, το σενάριο A_2 δε λαμβάνει υπόψη τις καθυστερήσεις λόγω των συγκρούσεων και συνεπώς είναι ένα μη ρεαλιστικό σενάριο, το καλύτερο σενάριο, σύμφωνα και με αυτό το κριτήριο, είναι και πάλι το Α. Ενώ, χειρότερο είναι το σενάριο A_4 με πολύ υψηλά ποσοστά χρόνων. Στο *γράφημα 6* παρουσιάζεται ο χρόνος αδράνειας των AGVs συναρτήσει του αριθμού των AGVs. Και στο γράφημα αυτό, τα καλύτερα αποτελέσματα, με πολύ μικρή διαφορά, δίνουν τα σενάρια Α και A_2, ενώ το χειρότερο σενάριο είναι το A_4.



Γράφημα 4: **Συγκριτικά αποτελέσματα συνολικού χρόνου προσομοίωσης συναρτήσει του αριθμού AGVs (περίπτωση γεμάτου τερματικού σταθμού)**



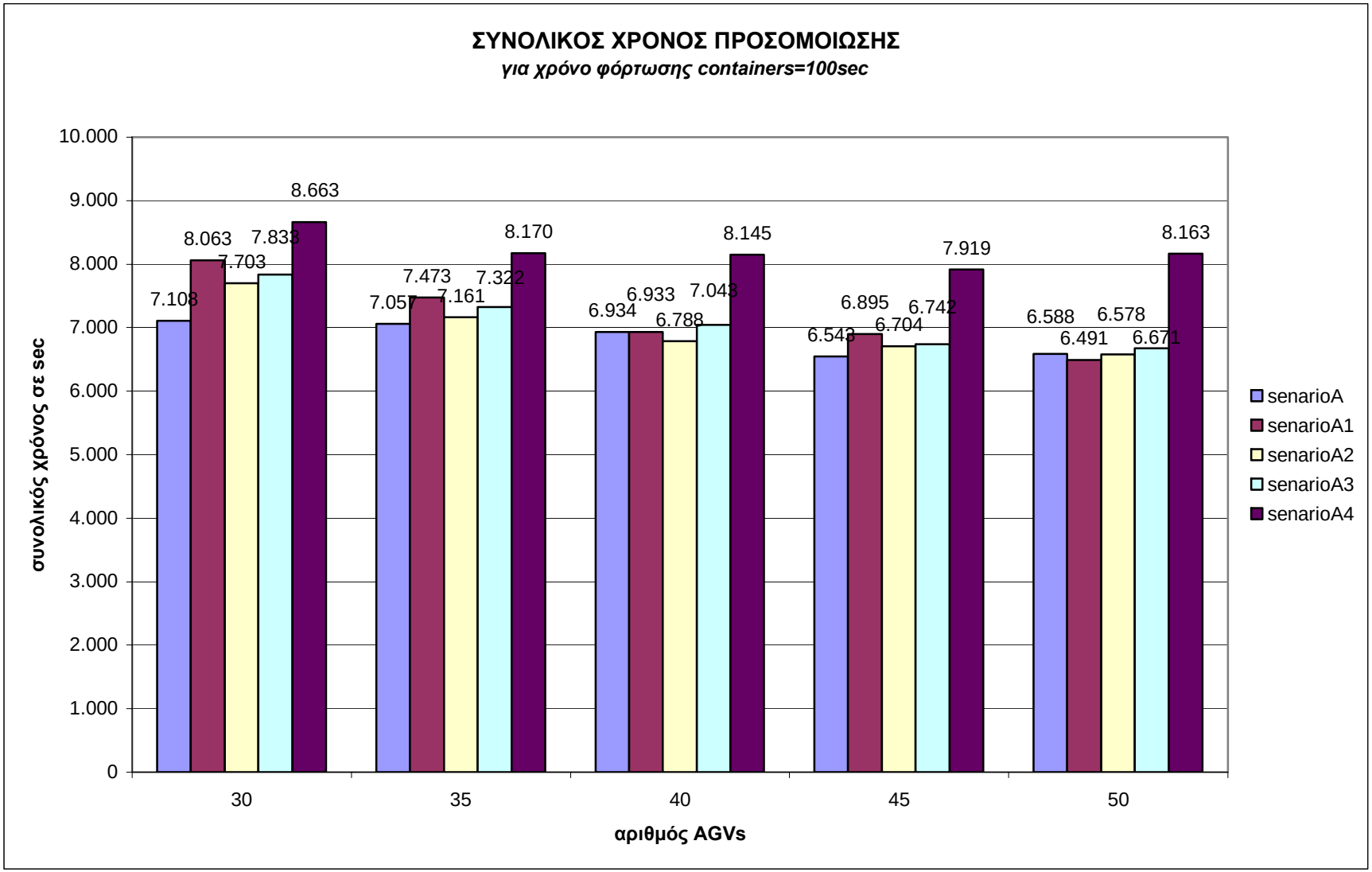
Γράφημα 5: **Συγκριτικά αποτελέσματα χρόνου αδράνειας γερανών συναρτήσει του αριθμού AGVs (περίπτωση γεμάτου τερματικού σταθμού)**



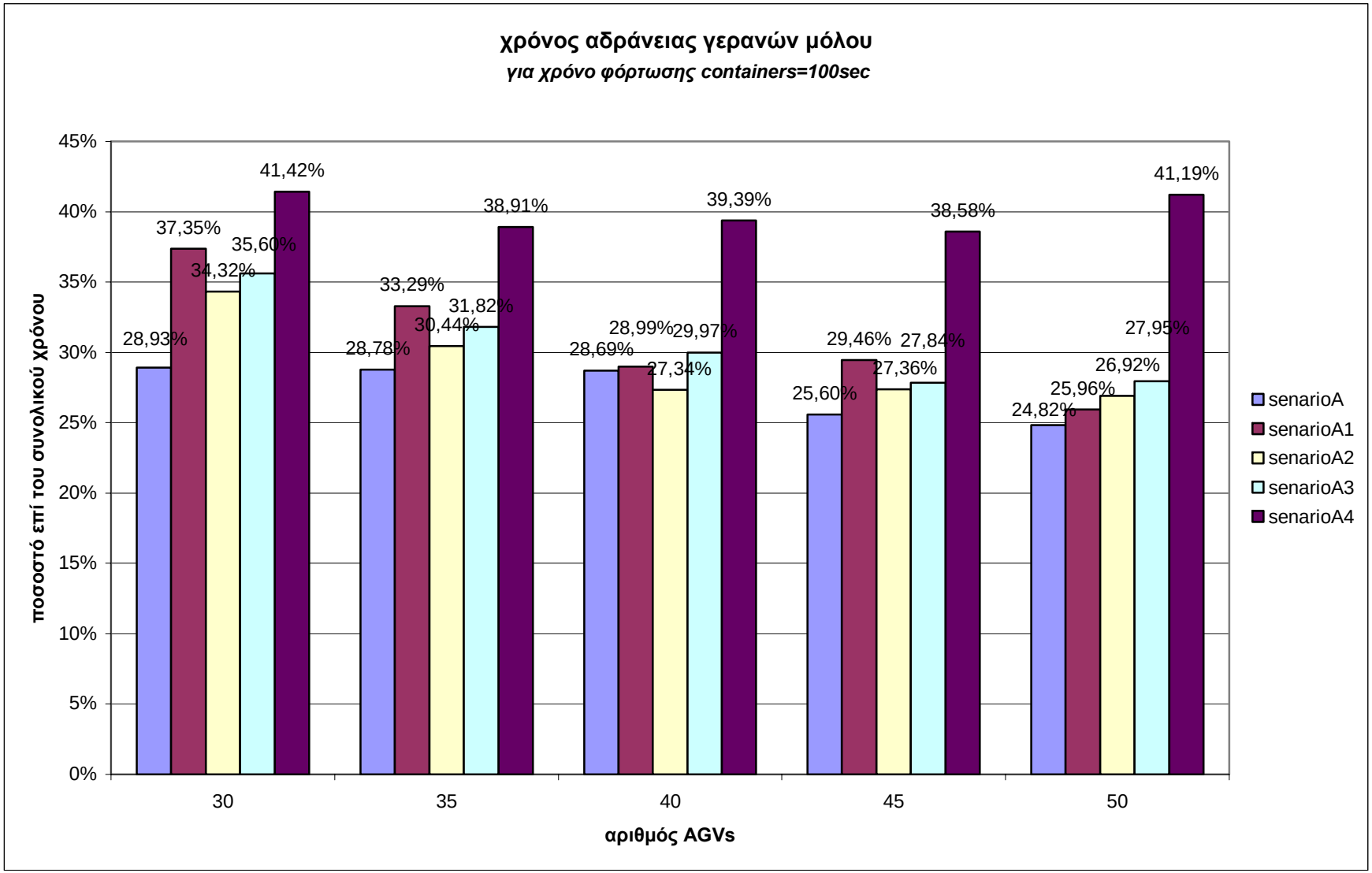
Γράφημα 6: **Συγκριτικά αποτελέσματα χρόνου αδράνειας AGVs συναρτήσει του αριθμού των AGVs (περίπτωση γεμάτου τερματικού σταθμού)**

Παρακάτω παρουσιάζονται τα αποτελέσματα για την περίπτωση που ο τερματικός σταθμός περιέχει 450 containers. Για την περίπτωση αυτή, εκτελούνται προσομοιώσεις για χρόνο φόρτωσης εντός της γυάρδας 100sec (*γραφήματα 7*και *8*) και για 900sec (*γραφήματα 9* και *10*). Στο *γράφημα 7* παρουσιάζονται τα αποτελέσματα όλων των σεναρίων για το συνολικό χρόνο προσομοίωσης συναρτήσει του αριθμού των AGVs. Για 30, 35 και για 45 AGVs το σενάριο Α παρουσιάζει το μικρότερο συνολικό χρόνο προσομοίωσης από τα υπόλοιπα σενάρια. Για 40 AGVs το μικρότερο χρόνο προσομοίωσης παρουσιάζει το σενάριο A_2 (6.788sec), ο οποίος βέβαια είναι πολύ κοντά με αυτόν του σεναρίου Α (6.934sec). Για 50 AGVs το μικρότερο χρόνο προσομοίωσης παρουσιάζει το σενάριο A_1(6.491sec), ο οποίος είναι πολύ κοντά με το χρόνο του σεναρίου Α(6.588sec), αλλά και με του σεναρίου A_2(6.578sec). Παρατηρούμε επίσης, ότι και σε αυτήν την περίπτωση το σενάριο A_4 απαιτεί πολύ μεγάλους χρόνους προσομοίωσης συγκριτικά με τα υπόλοιπα σενάρια. Το *γράφημα 8* αναπαριστά το ποσοστό του χρόνου αδράνειας των γερανών του μόλου επί του συνολικού χρόνου συναρτήσει του αριθμού των AGVs. Για 30, 35, 45 και 50 AGVs το σενάριο Α παρουσιάζει τους μικρότερους χρόνους αδράνειας των γερανών του μόλου, με αρκετή διαφορά επί των ποσοστών των χρόνων των υπολοίπων σεναρίων, ενώ για 40 AGVs χαμηλούς, αλλά και συγκρίσιμους χρόνους, παρουσιάζουν τα σενάρια A_2 (27,34%) και Α (28,69%). Πολύ μεγάλα ποσοστά χρόνων, για κάθε αριθμό AGVs, παρουσιάζει το σενάριο A_4.

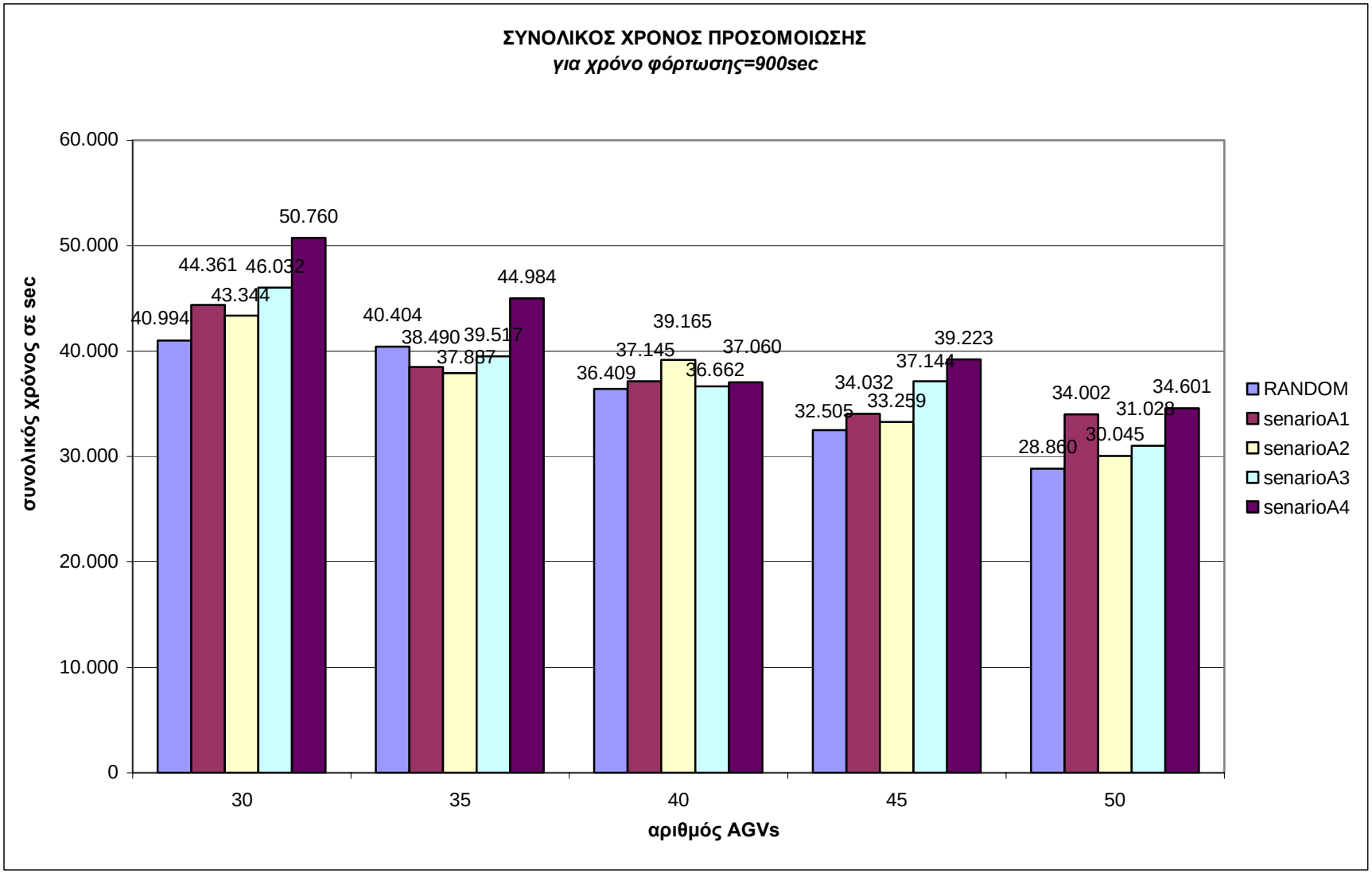
Τα *γραφήματα 9* και *10* αναπαριστούν τα αποτελέσματα των προσομοιώσεων για χρόνο φόρτωσης containers 900sec. Στο *γράφημα 9* παρουσιάζονται οι συνολικοί χρόνοι προσομοίωσης συναρτήσει του αριθμού των AGVs. Για 30, 40, 45 και 50 AGVs το σενάριο Α παρουσιάζει τους μικρότερους χρόνους προσομοίωσης. Αξιοσημείωτη είναι στο γράφημα αυτό, η μείωση της διαφοράς του χρόνου προσομοίωσης του σεναρίου A_4 με τα υπόλοιπα σενάρια, αν και σε αυτήν την περίπτωση οι χρόνοι προσομοίωσης του προαναφερόμενου σεναρίου είναι οι υψηλότεροι για κάθε αριθμό AGVs. Στο *γράφημα 10* παρουσιάζονται τα ποσοστά των χρόνων αδράνειας των γερανών του μόλου επί του συνολικού χρόνου προσομοίωσης, συναρτήσει του αριθμού των AGVs. Παρατηρούμε ότι όλοι οι χρόνοι είναι συγκρίσιμοι, αν και όπως και στις περισσότερες των περιπτώσεων το σενάριο Α παρουσιάζει τα χαμηλότερα ποσοστά χρόνων, ενώ το σενάριο A_4 τα υψηλότερα.



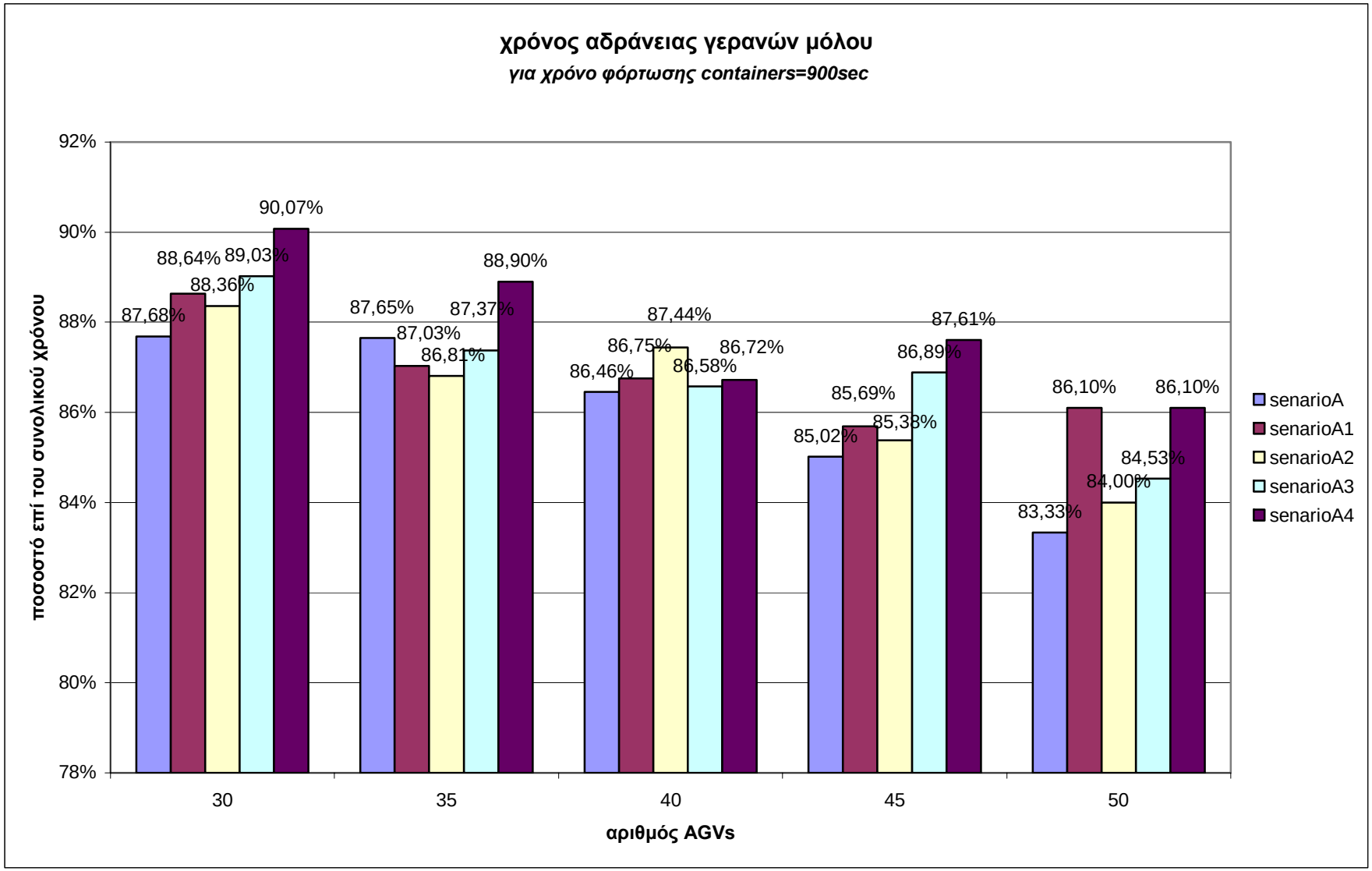
Γράφημα 7: **Συγκριτικά αποτελέσματα συνολικού χρόνου προσομοίωσης συναρτήσει του αριθμού AGVs (περίπτωση μερικής συμπλήρωσης τερματικού, χρόνος φόρτωσης containers=100sec)**



Γράφημα 8: **Συγκριτικά αποτελέσματα χρόνου αδράνειας γερανών συναρτήσει του αριθμού AGVs (περίπτωση μερικής συμπλήρωσης τερματικού, χρόνος φόρτωσης containers=900sec)**



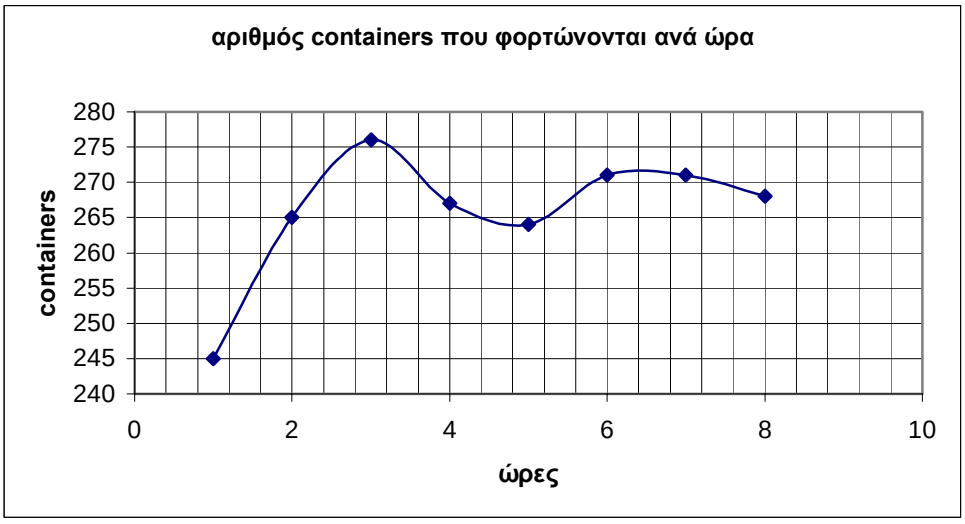
Γράφημα 9: **Συγκριτικά αποτελέσματα συνολικού χρόνου προσομοίωσης συναρτήσει του αριθμού AGVs (περίπτωση μερικής συμπλήρωσης του τερματικού, χρόνος φόρτωσης containers=900sec)**



Γράφημα 10: **Συγκριτικά αποτελέσματα χρόνου αδράνειας γερανών συναρτήσει του αριθμού AGVs (περίπτωση μερικής συμπλήρωσης τερματικού, χρόνος φόρτωσης containers=900sec)**

6.3 Επίλογος

Το μοντέλο που χρησιμοποιήθηκε και στο οποίο εφαρμόστηκαν τα παραπάνω σενάρια ανταποκρίνεται σε ρεαλιστικές συνθήκες. Παρόλο, που το αναμενόμενο θα ήταν να πάρουμε καλύτερους χρόνους από τα σενάρια κατά τα οποία μελετούσαμε τη χρονική απόσταση των οχημάτων, παρατηρούμε ότι ο αλγόριθμος τυχαίας ανάθεσης είναι σχεδόν σε όλες τις περιπτώσεις καλύτερος από όλα τα σενάρια που εφαρμόστηκαν. Σύμφωνα λοιπόν, με το σενάριο Α, μελετώντας το διάγραμμα συνολικού χρόνου, συμπεραίνουμε ότι ο βέλτιστος αριθμός AGVs για τη γυάρδα μας είναι τα 35 AGVs. Πράγματι, όπως φαίνεται και στο διάγραμμα έχουμε θεαματική μείωση του συνολικού χρόνου προσομοίωσης αυξάνοντας τον αριθμό των AGVs μέχρι τα 35. Από τα 35 AGVs και έπειτα δεν παρατηρείται σημαντική βελτίωση του χρόνου και άρα δεν συμφέρει η αγορά περισσότερων από τα τελευταία, μια και η τιμή τους παραμένει στα ύψη. Στο ίδιο συμπέρασμα καταλήγουμε μελετώντας και το διάγραμμα των χρόνων αδράνειας των γερανών του μόλου. Μέχρι τα 35 AGVs έχουμε σταδιακή μείωση των χρόνων αδράνειας των γερανών του μόλου, ενώ έπειτα τα 35 AGVs τα περιθώρια βελτίωσης ελαττώνονται. Στο *γράφημα 11* φαίνεται ο αριθμός containers που φορτώνονται ανά ώρα στον τερματικό σταθμό, σύμφωνα με το σενάριο Α και για 35 AGVs.



Γράφημα 11: **Αριθμός containers που φορτώνονται ανά ώρα, σύμφωνα με το σενάριο Α**

Από τα διαγράμματα της προηγούμενης παραγράφου παρατηρούμε επίσης, ότι για μικρό χρόνο φόρτωσης *(100sec)* το σενάριο Α δίνει σχεδόν για όλες τις περιπτώσεις τους καλύτερους χρόνους και μάλιστα με αρκετή διαφορά από τα υπόλοιπα σενάρια. Για την περίπτωση όμως, πιο αργών γερανών *(χρόνος φόρτωσης = 900 sec)*, η διαφορά των χρόνων μεταξύ των χρησιμοποιούμενων σεναρίων μειώνεται σημαντικά, με το σενάριο A_2 να παρουσιάζει πολύ ανταγωνιστικούς χρόνους με την πολιτική τυχαίας ανάθεσης. Αυτό εξηγείται από το γεγονός ότι τα σενάρια A_1 έως A_4, λόγω των περιορισμών και των προϋποθέσεων, που περιέχουν, όσον αφορά τους χρόνους και τις καθυστερήσεις, χρειάζονται να έχουν μεγάλο πεδίο δράσης. Για την εξεταζόμενη περίπτωση όμως, η χωρητικότητα του τερματικού σταθμού αποδεικνύεται ότι δεν επαρκεί για τα προαναφερόμενα σενάρια, με αποτέλεσμα ο εκάστοτε αλγόριθμος να μη βρίσκει πάντα τη βέλτιστη λύση και να κάνει αναθέσεις αναγκαστικά, που είτε δεν είναι βέλτιστες, ή καθυστερεί αρκετά περιμένοντας να ικανοποιηθούν όλοι οι περιορισμοί του. Έτσι, τα αποτελέσματα των τελικών χρόνων δεν είναι τα αναμενόμενα , αλλά τα χειρότερα, όπως συμβαίνει με το σενάριο A_4.

ΠΑΡΑΡΤΗΜΑ

% tel_terminal

```
%eisagogi diastaseon terminal
disp('dose arithmo container kata ton x axona pollaplasio tou 10')
a=input('arithmos container=')/10;
disp('dose arithmo container kata ton y axona pollaplasio tou 5')
b=input('arithmos container=')/5;
%Block1 coordinates
Block1=[
   0 33.55
   0 27.45
   0 21.35
   0 15.25
   0 9.15
   0 0
   12.2 0
   24.4 0
   36.6 0
   48.8 0
   61 0
   73.2 0
   85.4 0
   97.6 0
   109.8 0
   122 0
   134.2 0
   134.2 9.15
   134.2 15.25
   134.2 21.35
   134.2 27.45
   134.2 33.55
];
%generating coordinate matrix kata ton x
Block_x=[
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
   146.4 0
```

```
  146.4 0
  146.4 0
];
%generating coordinate matrix kata ton y
Block_y=[
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
  0 42.7
];
for i = 1:a
  eval(['Block' num2str(i) ' = Block1+(i-1)*Block_x']);
  for j = 2:b
    eval(['Block' num2str((j-1)*a+i) ' = Block1+(i-1)*Block_x+(j-1)*Block_y']);
  end
end
%generating cranes coordinates
crane=[
  (fix((a+1)/2))*134.2+(fix((a+1)/2)-1)*12.2 -4*12.2
  (fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2 -4*12.2
];
x_crane=crane(:,1);
y_crane=crane(:,2);
hold on;
plot(x_crane,y_crane,'b+:');
%generating entrance coordinates
entrance=[
  (fix((a+1)/2))*134.2+(fix((a+1)/2)-1)*12.2 (b*5*6.1+(b-1)*12.2+2*12.2)
  (fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2 (b*5*6.1+(b-1)*12.2+2*12.2)
];
x_entrance=entrance(:,1);
y_entrance=entrance(:,2);
hold on;
plot(x_entrance,y_entrance,'b+:');
```

```
%generating crossing coordinate matrix
crossing=[
  0 0
  134.2 0
];
for i=2:a*b
        eval(['crossing=[crossing;Block' num2str(i) '(6,:);Block' num2str(i) '(17,:)]']);
end
x_crossing=crossing(:,1);
y_crossing=crossing(:,2);
hold on;
plot(x_crossing,y_crossing,'b+:');

%generating container coordinate matrix
for i=1:a*b
  for j=7:16
    eval(['container=[container;Block' num2str(i) '(j,:)]']);;
  end
end
x_container=container(:,1);
y_container=container(:,2);
hold on;
plot(x_container,y_container,'rx:');
```

% senario A

```
a=4;
b=4;
no_of_agv=35;
no_of_trucks=0;
dt=0.25;
hold_time=0;
Time_to_load_container=100;%sec
Time_to_load_crane=48;%sec
Time_to_load_gate=48;%sec

load_time_crane=zeros(1,4);
load_time=zeros(1,no_of_trucks+no_of_agv);
Dir_agv=zeros(2,no_of_trucks+no_of_agv);
Dir_agv(2,:)=-1;
Load_agv=zeros(1,no_of_trucks+no_of_agv);
%destination = 1 (gate) -1 (container)
Dest=zeros(1,no_of_trucks+no_of_agv);
Dest(:)=-1;
%assigned = 0 (not assigned) 1 (assigned)
Assign=zeros(1,no_of_trucks+no_of_agv);
x_cont=zeros(1,no_of_trucks+no_of_agv);
y_cont=zeros(1,no_of_trucks+no_of_agv);
x_agv=zeros(1,no_of_trucks+no_of_agv);
y_agv=zeros(1,no_of_trucks+no_of_agv);
x_dest=zeros(1,no_of_trucks+no_of_agv);
x_entrance=zeros(1,no_of_trucks+no_of_agv);
x_origin=zeros(1,no_of_trucks+no_of_agv);
y_dest=zeros(1,no_of_trucks+no_of_agv);
y_entrance=zeros(1,no_of_trucks+no_of_agv);
y_origin=zeros(1,no_of_trucks+no_of_agv);
Velocity=zeros(1,no_of_trucks+no_of_agv);
col=zeros(1,no_of_trucks+no_of_agv);
turn_col=zeros(1,no_of_trucks+no_of_agv);
queue_col=zeros(1,no_of_trucks+no_of_agv);
length_col=2;
length_alarm=2;
y_collision=zeros(1,no_of_trucks+no_of_agv);
x_collision=zeros(1,no_of_trucks+no_of_agv);
x_gate=[265.6 275.6 285.6 295.6];
y_gate=[185.5 188 188 185.5];
x_cranes=[277.8 287.8 297.8 307.8];
y_cranes=[-51.3 -53.8 -53.8 -51.3];
go_to_crane=zeros(4,no_of_agv);
idle_crane=zeros(1,4);
idle_agv=zeros(1,no_of_agv+no_of_trucks);
crane=zeros(1,4);
gate=zeros(1,4);
containers_loaded=0;
containers_unloaded=0;
containers_per_crane=zeros(1,4);
k=0;
```

```
Distance=zeros(1,no_of_trucks+no_of_agv);


N=14400*10;

containers=zeros(1,N/14400);


% IRINI addition

storage=(1:160);
for i=1:160
   storage(i)=15;
end

% End of IRINI

% IRINI addition


loop=0;

while sum(storage) >0

 loop=loop+1;


 % End of IRINI

  for i=1:no_of_trucks+no_of_agv
    %statements for trucks
    if i<=no_of_trucks
      if Assign(i)==0
        %random epilogi container
        rand_cont=1+fix(a*b*10*rand);
        x_cont(i)=container(rand_cont,1);
        y_cont(i)=container(rand_cont,2);
        x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2)-1)*12.2;
        y_gate_center=(b*5*6.1+(b-1)*12.2+2*12.2);
        x_agv(i)=x_gate_center;
        y_agv(i)=y_gate_center;
        Load_agv(i)=+1;
        load_time(i)=0;
        Assign(i)=1;
      end
      if Dest(i)==-1
        x_origin(i)=x_gate_center;
        y_origin(i)=y_gate_center;
        x_dest(i)=x_cont(i);
        y_dest(i)=y_cont(i);
      end
      if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
```

```
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=0;
  y_dest(i)=y_cont(i);
end
if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=146.4;
  y_dest(i)=y_cont(i);
end
if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<439.2
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=439.2;
  y_dest(i)=y_cont(i);
end
if Dest(i)==1 & 439.2<x_cont(i) & x_cont(i)<585.6
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=585.6;
  y_dest(i)=y_cont(i);
end
if Dest(i)==-1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;%m/sec
  end
  if y_agv(i)>y_dest(i)
    Dir_agv(:,i)=[
      0
      -1
    ];

  else
    y_agv(i)=y_dest(i);  %diorthotiki kinisi
    if abs(x_agv(i)-x_dest(i))>1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    elseif load_time(i)<Time_to_load_container
      x_agv(i)=x_dest(i);          %diorthotiki kinisi
      Velocity(i)=0;
      load_time(i)=load_time(i)+dt;
    else
      Dest(i)=1;
      x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
      load_time(i)=0;
      Load_agv(i)=0;
    end
```

```
  end
elseif Dest(i)==1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;%m/sec
  end
  if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
    Dir_agv(:,i)=[
      sign(x_dest(i)-x_origin(i))
      0
    ];
  end
  if y_agv(i)<183 & abs(x_agv(i)-x_dest(i))<1.2
    x_agv(i)=x_dest(i);  %diorthotiki kinisi
    Dir_agv(:,i)=[
      0
      1
    ];
  end
  if abs(x_agv(i)-x_gate_center)>1.2 & abs(y_agv(i)-183)<1.2
    y_agv(i)=183; %diorthotiki kinisi
    Dir_agv(:,i)=[
      sign(x_gate_center-x_origin(i))
      0
    ];
  end
  if abs(y_gate_center-y_agv(i))<1.2 & abs(x_agv(i)-x_gate_center)<25
    if x_agv(i)-x_gate_center<0
      if gate(1)==0
        y_agv(i)=185.5;
        Dir_agv(:,i)=[
          1
          0
        ];
      elseif gate(2)==0
        y_agv(i)=188;
        Dir_agv(:,i)=[
          1
          0
        ];
      else
        Velocity(i)=0;
      end
    else
      if gate(4)==0
        y_agv(i)=185.5;
        Dir_agv(:,i)=[
          -1
          0
        ];
      elseif crane(3)==0
```

```
            y_agv(i)=188;
            Dir_agv(:,i)=[
              -1
              0
            ];
          else
            Velocity(i)=0;
          end
        end
      end
      for j=1:4
        if abs(x_agv(i)-x_gate(j))<1.2 & abs(y_agv(i)-y_gate(j))<1.2
          x_agv(i)=x_gate(j);
          gate(j)=1;
          Velocity(i)=0;
          if load_time(i)<Time_to_load_gate
            load_time(i)=load_time(i)+dt;
          else
            x_agv(i)=x_gate_center;
            y_agv(i)=y_gate_center;
            load_time(i)=0;
            gate(j)=0;
            containers_unloaded=containers_unloaded+1;
            Dest(i)=-1;
            Assign(i)=0;
          end
        end
      end
    end
    %statements for agv's
  else
    if Assign(i)==0
      %random epilogi container
%       rand_cont=1+fix(a*b*10*rand);

% IRINI ADDITION

counter=160*15;
while counter>0
          rand_cont=1+fix(a*b*10*rand);
           if storage(rand_cont)>0
                storage(rand_cont)=storage(rand_cont)-1;
                counter=0;

             else
                                        counter=counter-1;
          end
        end

 sum(storage)

% END OF IRINI ADDITION
```

```
    x_cont(i)=container(rand_cont,1);
    y_cont(i)=container(rand_cont,2);
    x_crane_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
    y_crane_center=-4*12.2;
    x_agv(i)=x_crane_center;
    y_agv(i)=y_crane_center;

    Load_agv(i)=0;
    load_time(i)=0;
    Assign(i)=1;
  end
  if Dest(i)==-1
    x_origin(i)=x_crane_center;
    y_origin(i)=y_crane_center;
    x_dest(i)=x_cont(i);
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=-12.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=134.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<427
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=427;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 427<x_cont(i) & x_cont(i)<573.4
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=573.4;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==-1 & Assign(i)==1
    if Load_agv(i)==+1
      Velocity(i)=2.3;%m/sec
    else
      Velocity(i)=4.6;%m/sec
    end
    if y_agv(i)<y_dest(i)
      Dir_agv(:,i)=[
        0
```

```
        1
      ];

    else
      y_agv(i)=y_dest(i);  %diorthotiki kinisi
      if abs(x_agv(i)-x_dest(i))>1.2
        Dir_agv(:,i)=[
          sign(x_dest(i)-x_origin(i))
          0
        ];
      elseif load_time(i)<Time_to_load_container

        x_agv(i)=x_dest(i);          %diorthotiki kinisi
        Velocity(i)=0;
        load_time(i)=load_time(i)+dt;
      else
        Dest(i)=-Dest(i);
        load_time(i)=0;
        Load_agv(i)=1;
      end
    end
  elseif Dest(i)==1 & Assign(i)==1
    if Load_agv(i)==+1
      Velocity(i)=2.3;%m/sec
    else
      Velocity(i)=4.6;
    end
    if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    end
    if y_agv(i)>-48.8 & abs(x_agv(i)-x_dest(i))<1.2
      x_agv(i)=x_dest(i);  %diorthotiki kinisi
      Dir_agv(:,i)=[
        0
        -1
      ];
    end
    if abs(x_agv(i)-x_crane_center)>1.2 & abs(y_agv(i)+48.8)<1.2
      y_agv(i)=-48.8; %diorthotiki kinisi
      Dir_agv(:,i)=[
        sign(x_crane_center-x_origin(i))
        0
      ];
    end
    if y_agv(i)==y_crane_center & abs(x_agv(i)-x_crane_center)<25
      if x_agv(i)-x_crane_center<0
        if crane(1)==0 | (load_time_crane(1)>44.25 & go_to_crane(1,:)==0)
          crane(1)=1;
          go_to_crane(1,i)=1;
```

```
        y_agv(i)=-51.3;
        Dir_agv(:,i)=[
          1
          0
        ];
      elseif crane(2)==0 | (load_time_crane(2)>40 & go_to_crane(2,:)==0)
        crane(2)=1;
        go_to_crane(2,i)=1;
        y_agv(i)=-53.8;
        Dir_agv(:,i)=[
          1
          0
        ];
      else
        x_agv(i)=x_crane_center-25;
        Velocity(i)=0;
        idle_agv(i)=idle_agv(i)+1;
      end
    else
      if crane(4)==0 | (load_time_crane(4)>44.25 & go_to_crane(4,:)==0)
        crane(4)=1;
        go_to_crane(4,i)=1;
        y_agv(i)=-51.3;
        Dir_agv(:,i)=[
          -1
          0
        ];
      elseif crane(3)==0 | (load_time_crane(3)>40 & go_to_crane(3,:)==0)
        crane(3)=1;
        go_to_crane(3,i)=1;
        y_agv(i)=-53.8;
        Dir_agv(:,i)=[
          -1
          0
        ];
      else
        x_agv(i)=x_crane_center+25;
        Velocity(i)=0;
        idle_agv(i)=idle_agv(i)+1;
      end
    end
  end
  for j=1:4
    if abs(x_agv(i)-x_cranes(j))<1.2 & y_agv(i)==y_cranes(j)
      x_agv(i)=x_cranes(j);
      go_to_crane(j,i)=0;
      Velocity(i)=0;
      if load_time(i)<Time_to_load_crane
        load_time(i)=load_time(i)+dt;
        load_time_crane(j)=load_time(i);
        hold_time=hold_time+dt;
      else
```

```
            x_agv(i)=x_crane_center;
            y_agv(i)=y_crane_center;
            load_time(i)=0;
            load_time_crane(j)=0;
            crane(j)=0;
            containers_loaded=containers_loaded+1;
            containers_per_crane(j)=containers_per_crane(j)+1;
            Dest(i)=-1;
            Assign(i)=0;
          end
        end
      end
    end
end


%check for collision ston y

if Dir_agv(2,i)==1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if y_agv(j)-y_agv(i)<length_col & y_agv(j)-y_agv(i)>0 & x_agv(j)==x_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end
if Dir_agv(2,i)==-1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if y_agv(i)-y_agv(j)<length_col & y_agv(i)-y_agv(j)>0 & x_agv(j)==x_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end

%check for collision ston x

if Dir_agv(1,i)==-1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if x_agv(i)-x_agv(j)<length_col & x_agv(i)-x_agv(j)>0 & y_agv(j)==y_agv(i)
        Velocity(i)=Velocity(j);
```

```
            col(i)=1+col(i);
            if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
              queue_col(i)=queue_col(i)+1;
            end

            if Velocity(j)==0
              idle_agv(i)=idle_agv(i)+1;
            end
          end
        end
      end
    end
    if Dir_agv(1,i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          if x_agv(j)-x_agv(i)<length_col & x_agv(j)-x_agv(i)>0 & y_agv(j)==y_agv(i)
            Velocity(i)=Velocity(j);
            col(i)=1+col(i);
            if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
              queue_col(i)=queue_col(i)+1;
            end

            if Velocity(j)==0
              idle_agv(i)=idle_agv(i)+1;
            end
          end
        end
      end
    end
    %elegxos strofis
    if abs(y_dest(i)-y_agv(i))<length_alarm & y_dest(i)~=y_agv(i) & Dest(i)==-1
      y_collision(i)=1;
      x_collision(i)=0;
    elseif abs(x_dest(i)-x_agv(i))<length_alarm & x_dest(i)~=x_agv(i) & Dest(i)==+1
      y_collision(i)=0;
      x_collision(i)=1;
    end
    %gia ton kentriko dromo
    if y_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na katevainei kai na thelei na stripsei aristera
          if abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i)&
Dir_agv(2,i)==-1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)>(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          %periptosi na anevainei kai na thelei na stripsei aristera
          elseif abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i) &
Dir_agv(2,i)==1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)<(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(j)=0;
```

```
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
    %gia tis ipoloipes diastayroseis
    if x_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na kateythinetai aristera
          if Dir_agv(1,i)==-1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          elseif Dir_agv(1,i)==-1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
          %periptosi na kateythinetai dexia
          if Dir_agv(1,i)==1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=turn_col(i)+1;
          elseif Dir_agv(1,i)==1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end




    x_agv(i)=x_agv(i)+dt*Dir_agv(1,i)*Velocity(i);
    y_agv(i)=y_agv(i)+dt*Dir_agv(2,i)*Velocity(i);

Distance(i)=Distance(i)+abs(dt*Dir_agv(1,i)*Velocity(i))+abs(dt*Dir_agv(2,i)*Velocity(i));







    x_collision(i)=0;
    y_collision(i)=0;
  end
```

```
  if mod(loop,1440)==0
    loop
  end


  if mod(loop,14400)==0
    loop
    k=k+1;
    containers(k)=containers_loaded
    containers_loaded=0;
    hold_time1=hold_time;

  end
  for j=1:4
    if load_time_crane(j)==0
      idle_crane(j)=idle_crane(j)+1;
    end
  end
end
average=sum(containers)/(loop*dt*4)
idle_rate_cranes=((sum(idle_crane)/4)/loop)*100
idle_rate_agv=((sum(idle_agv)/no_of_agv)/loop)*100
%IRINI ADDITION
total_time=loop*dt
containers
% IRINI ADDITION
```

% SENARIO A1

```
a=4;
b=4;
```

```
no_of_agv=30;
no_of_trucks=0;
dt=0.25;
hold_time=0;
Time_to_load_container=100;%sec
Time_to_load_crane=48;%sec
Time_to_load_gate=48;%sec

load_time_crane=zeros(1,4);
load_time=zeros(1,no_of_trucks+no_of_agv);
Dir_agv=zeros(2,no_of_trucks+no_of_agv);
Dir_agv(2,:)=-1;
Load_agv=zeros(1,no_of_trucks+no_of_agv);
%destination = 1 (gate) -1 (container)
Dest=zeros(1,no_of_trucks+no_of_agv);
Dest(:)=-1;
%assigned = 0 (not assigned) 1 (assigned)
Assign=zeros(1,no_of_trucks+no_of_agv);
x_cont=zeros(1,no_of_trucks+no_of_agv);
y_cont=zeros(1,no_of_trucks+no_of_agv);
x_agv=zeros(1,no_of_trucks+no_of_agv);
y_agv=zeros(1,no_of_trucks+no_of_agv);
x_dest=zeros(1,no_of_trucks+no_of_agv);
x_entrance=zeros(1,no_of_trucks+no_of_agv);
x_origin=zeros(1,no_of_trucks+no_of_agv);
y_dest=zeros(1,no_of_trucks+no_of_agv);
y_entrance=zeros(1,no_of_trucks+no_of_agv);
y_origin=zeros(1,no_of_trucks+no_of_agv);
Velocity=zeros(1,no_of_trucks+no_of_agv);
col=zeros(1,no_of_trucks+no_of_agv);
turn_col=zeros(1,no_of_trucks+no_of_agv);
queue_col=zeros(1,no_of_trucks+no_of_agv);
length_col=2;
length_alarm=2;
y_collision=zeros(1,no_of_trucks+no_of_agv);
x_collision=zeros(1,no_of_trucks+no_of_agv);
x_gate=[265.6 275.6 285.6 295.6];
y_gate=[185.5 188 188 185.5];
x_cranes=[277.8 287.8 297.8 307.8];
y_cranes=[-51.3 -53.8 -53.8 -51.3];
go_to_crane=zeros(4,no_of_agv);
idle_crane=zeros(1,4);
idle_agv=zeros(1,no_of_agv+no_of_trucks);
crane=zeros(1,4);
gate=zeros(1,4);
containers_loaded=0;
containers_unloaded=0;
containers_per_crane=zeros(1,4);
k=0;
counter=0;
Tcounter=0;
counter2=0;
```

```
Distance=zeros(1,no_of_trucks+no_of_agv);
pinakas_elegxoy=[1
  2
  3
  4
  5
  6
];
%diafora xronon mexri to destination tou i
T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
%xronos gia na paei to i apo arxi se telos xoris collision
T2=zeros(no_of_agv+no_of_trucks);
%diafora xronon gia na min exoume collision sto queue
T3=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
for i=1:no_of_agv+no_of_trucks
  for j=1:no_of_agv+no_of_trucks
    T3(i,j)=18;
  end
end
%xronos poy exei ginei anathesi sto i se loops(to collision lambanetai ipopsin)
T4=zeros(no_of_agv+no_of_trucks);
l=1;
ll=0;

N=14400*10;

% IRINI addition

storage=(1:160);
for i=1:160
   storage(i)=15;
end

% End of IRINI


containers=zeros(1,N/14400);


% IRINI addition

loop=0;

while sum(storage) >0

 loop=loop+1;

 % End of IRINI


  for i=1:no_of_trucks+no_of_agv
    %statements for trucks
```

```
if i<=no_of_trucks
  if Assign(i)==0
    %random epilogi container
    rand_cont=1+fix(a*b*10*rand);
    x_cont(i)=container(rand_cont,1);
    y_cont(i)=container(rand_cont,2);
    x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2)-1)*12.2;
    y_gate_center=(b*5*6.1+(b-1)*12.2+2*12.2);
    x_agv(i)=x_gate_center;
    y_agv(i)=y_gate_center;
    Load_agv(i)=+1;
    load_time(i)=0;
    Assign(i)=1;
  end
  if Dest(i)==-1
    x_origin(i)=x_gate_center;
    y_origin(i)=y_gate_center;
    x_dest(i)=x_cont(i);
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=0;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=146.4;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<439.2
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=439.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 439.2<x_cont(i) & x_cont(i)<585.6
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=585.6;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==-1 & Assign(i)==1
    if Load_agv(i)==+1
      Velocity(i)=2.3;%m/sec
    else
      Velocity(i)=4.6;%m/sec
    end
    if y_agv(i)>y_dest(i)
      Dir_agv(:,i)=[
```

```
        0
        -1
      ];

    else
      y_agv(i)=y_dest(i);  %diorthotiki kinisi
      if abs(x_agv(i)-x_dest(i))>1.2
        Dir_agv(:,i)=[
          sign(x_dest(i)-x_origin(i))
          0
        ];
      elseif load_time(i)<Time_to_load_container
        x_agv(i)=x_dest(i);          %diorthotiki kinisi
        Velocity(i)=0;
        load_time(i)=load_time(i)+dt;
      else
        Dest(i)=1;
        x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
        load_time(i)=0;
        Load_agv(i)=0;
      end
    end
  elseif Dest(i)==1 & Assign(i)==1
    if Load_agv(i)==+1
      Velocity(i)=2.3;%m/sec
    else
      Velocity(i)=4.6;%m/sec
    end
    if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    end
    if y_agv(i)<183 & abs(x_agv(i)-x_dest(i))<1.2
      x_agv(i)=x_dest(i);  %diorthotiki kinisi
      Dir_agv(:,i)=[
        0
        1
      ];
    end
    if abs(x_agv(i)-x_gate_center)>1.2 & abs(y_agv(i)-183)<1.2
      y_agv(i)=183; %diorthotiki kinisi
      Dir_agv(:,i)=[
        sign(x_gate_center-x_origin(i))
        0
      ];
    end
    if abs(y_gate_center-y_agv(i))<1.2 & abs(x_agv(i)-x_gate_center)<25
      if x_agv(i)-x_gate_center<0
        if gate(1)==0
          y_agv(i)=185.5;
```

```
            Dir_agv(:,i)=[
              1
              0
            ];
          elseif gate(2)==0
            y_agv(i)=188;
            Dir_agv(:,i)=[
              1
              0
            ];
          else
            Velocity(i)=0;
          end
        else
          if gate(4)==0
            y_agv(i)=185.5;
            Dir_agv(:,i)=[
              -1
              0
            ];
          elseif crane(3)==0
            y_agv(i)=188;
            Dir_agv(:,i)=[
              -1
              0
            ];
          else
            Velocity(i)=0;
          end
        end
      end
      for j=1:4
        if abs(x_agv(i)-x_gate(j))<1.2 & abs(y_agv(i)-y_gate(j))<1.2
          x_agv(i)=x_gate(j);
          gate(j)=1;
          Velocity(i)=0;
          if load_time(i)<Time_to_load_gate
            load_time(i)=load_time(i)+dt;
          else
            x_agv(i)=x_gate_center;
            y_agv(i)=y_gate_center;
            load_time(i)=0;
            gate(j)=0;
            containers_unloaded=containers_unloaded+1;
            Dest(i)=-1;
            Assign(i)=0;
          end
        end
      end
    end
    %statements for agv's
  else
```

```
        if Assign(i)==1
          T4(i,1)=T4(i,1)+1;
        end
        if loop>1350
          ll=0;
        end
        if loop>ll*50
          while Assign(i)==0
          counter=counter+1;
          %random epilogi container
          % rand_cont=1+fix(a*b*10*rand);
% IRINI ADDITION
         kcounter=160*15;
                  while kcounter>0
               rand_cont=1+fix(a*b*10*rand);
                if storage(rand_cont)>0
                     kcounter=0;

                  else
                                          kcounter=kcounter-1;
                    end
           end
loop
 sum(storage)


% END OF IRINI ADDITION
          x_cont(i)=container(rand_cont,1);
          y_cont(i)=container(rand_cont,2);
          x_crane_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
          y_crane_center=-4*12.2;
          x_agv(i)=x_crane_center;
          y_agv(i)=y_crane_center;
          Load_agv(i)=0;
          load_time(i)=0;
          pinakas_elegxoy(l)=i;
          if x_cont(i)<134.2
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)+12.2+280))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>134.2 & x_cont(i)<280.6
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)-134.2+133.6))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>292.8 & x_cont(i)<427
            T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+427+109.2))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>427
```

```
            T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+573.4+255.6))/2.3)+Time_to_load_container;
          end
          T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
          for ii=1:no_of_agv+no_of_trucks
            for jj=1:no_of_agv+no_of_trucks
              T3(ii,jj)=24;
            end
          end

          for j=1:no_of_agv+no_of_trucks

            if j~=i & j~=0
              if Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)<x_crane_center & x_cont(i)<x_crane_center
                if x_dest(j)-x_cont(i)<0
                 T1(i,j)=1;
                elseif x_cont(i)<146 & x_dest(j)>146
                  T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-146)/2.3)+Time_to_load_container-
load_time(j));
                else
                  T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
                end
              elseif Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)>x_crane_center & x_cont(i)>x_crane_center
                if x_dest(j)-x_cont(i)>0
                  T1(i,j)=1;
                elseif x_cont(i)>439 & x_dest(j)<439
                  T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+439)/2.3)+Time_to_load_container-
load_time(j));
                else
                  T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
                end
              end


              %IRINI add

              if T1(i,j)<=0
                 T1(i,j)=0;
              end

              %IRINI end
```

```
        if x_cont(i)<x_crane_center & x_dest(j)<x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if x_cont(i)>x_crane_center & x_dest(j)>x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if T3(i,j)<24
          Tcounter=Tcounter+T3(i,j);
          counter2=counter2+1;
          end
      end
    end




    %prakatv ginetai i epilogi sinthikis T3>24 i T3<24
    if min(T3(i,:))>=24 | counter2>(160*15)
      counter2=0;
      Assign(i)=1;
      storage(rand_cont)=storage(rand_cont)-1;
      ll=ll+1;
      counter=0;
      l=l+1;
      if l==7
        l=1;
      end
    end
    end
  end
  if Dest(i)==-1
    x_origin(i)=x_crane_center;
    y_origin(i)=y_crane_center;
    x_dest(i)=x_cont(i);
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=-12.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=134.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<427
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=427;
```

```
  y_dest(i)=y_cont(i);
end
if Dest(i)==1 & 427<x_cont(i) & x_cont(i)<573.4
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=573.4;
  y_dest(i)=y_cont(i);
end
if Dest(i)==-1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;%m/sec
  end
  if y_agv(i)<y_dest(i)
    Dir_agv(:,i)=[
      0
      1
    ];

  else
    y_agv(i)=y_dest(i);  %diorthotiki kinisi
    if abs(x_agv(i)-x_dest(i))>1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    elseif load_time(i)<Time_to_load_container

      x_agv(i)=x_dest(i);          %diorthotiki kinisi
      Velocity(i)=0;
      load_time(i)=load_time(i)+dt;
    else
      Dest(i)=-Dest(i);
      load_time(i)=0;
      Load_agv(i)=1;
    end
  end
elseif Dest(i)==1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;
  end
  if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
    Dir_agv(:,i)=[
      sign(x_dest(i)-x_origin(i))
      0
    ];
  end
  if y_agv(i)>-48.8 & abs(x_agv(i)-x_dest(i))<1.2
    x_agv(i)=x_dest(i);  %diorthotiki kinisi
```

```
  Dir_agv(:,i)=[
    0
    -1
  ];
end
if abs(x_agv(i)-x_crane_center)>1.2 & abs(y_agv(i)+48.8)<1.2
  y_agv(i)=-48.8; %diorthotiki kinisi
  Dir_agv(:,i)=[
    sign(x_crane_center-x_origin(i))
    0
  ];
end
if y_agv(i)==y_crane_center & abs(x_agv(i)-x_crane_center)<25
  if x_agv(i)-x_crane_center<0
    if crane(1)==0 | (load_time_crane(1)>44.25 & go_to_crane(1,:)==0)
      crane(1)=1;
      go_to_crane(1,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        1
        0
      ];
    elseif crane(2)==0 | (load_time_crane(2)>40 & go_to_crane(2,:)==0)
      crane(2)=1;
      go_to_crane(2,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        1
        0
      ];
    else
      x_agv(i)=x_crane_center-25;
      Velocity(i)=0;
      idle_agv(i)=idle_agv(i)+1;
    end
  else
    if crane(4)==0 | (load_time_crane(4)>44.25 & go_to_crane(4,:)==0)
      crane(4)=1;
      go_to_crane(4,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        -1
        0
      ];
    elseif crane(3)==0 | (load_time_crane(3)>40 & go_to_crane(3,:)==0)
      crane(3)=1;
      go_to_crane(3,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        -1
        0
      ];
```

```
          else
            x_agv(i)=x_crane_center+25;
            Velocity(i)=0;
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
      for j=1:4
        if abs(x_agv(i)-x_cranes(j))<1.2 & y_agv(i)==y_cranes(j)
          x_agv(i)=x_cranes(j);
          go_to_crane(j,i)=0;
          Velocity(i)=0;
          if load_time(i)<Time_to_load_crane
            load_time(i)=load_time(i)+dt;
            load_time_crane(j)=load_time(i);
            hold_time=hold_time+dt;
          else
            x_agv(i)=x_crane_center;
            y_agv(i)=y_crane_center;
            load_time(i)=0;
            load_time_crane(j)=0;
            crane(j)=0;
            containers_loaded=containers_loaded+1;
            containers_per_crane(j)=containers_per_crane(j)+1;
            Dest(i)=-1;
            Assign(i)=0;
            T4(i,1)=0;
          end
        end
      end
    end
  end
end


%check for collision ston y

if Dir_agv(2,i)==1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if y_agv(j)-y_agv(i)<length_col & y_agv(j)-y_agv(i)>0 & x_agv(j)==x_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end
if Dir_agv(2,i)==-1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
```

```
        if y_agv(i)-y_agv(j)<length_col & y_agv(i)-y_agv(j)>0 & x_agv(j)==x_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end

  %check for collision ston x

  if Dir_agv(1,i)==-1
    for j=1:no_of_trucks+no_of_agv
      if j~=i
        if x_agv(i)-x_agv(j)<length_col & x_agv(i)-x_agv(j)>0 & y_agv(j)==y_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
            queue_col(i)=queue_col(i)+1;
          end

          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end
  if Dir_agv(1,i)==1
    for j=1:no_of_trucks+no_of_agv
      if j~=i
        if x_agv(j)-x_agv(i)<length_col & x_agv(j)-x_agv(i)>0 & y_agv(j)==y_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
            queue_col(i)=queue_col(i)+1;
          end

          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end
  %elegxos strofis
  if abs(y_dest(i)-y_agv(i))<length_alarm & y_dest(i)~=y_agv(i) & Dest(i)==-1
    y_collision(i)=1;
    x_collision(i)=0;
  elseif abs(x_dest(i)-x_agv(i))<length_alarm & x_dest(i)~=x_agv(i) & Dest(i)==+1
```

```
      y_collision(i)=0;
      x_collision(i)=1;
    end
    %gia ton kentriko dromo
    if y_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na katevainei kai na thelei na stripsei aristera
          if abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i)&
Dir_agv(2,i)==-1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)>(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          %periptosi na anevainei kai na thelei na stripsei aristera
          elseif abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i) &
Dir_agv(2,i)==1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)<(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(j)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
    %gia tis ipoloipes diastayroseis
    if x_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na kateythinetai aristera
          if Dir_agv(1,i)==-1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          elseif Dir_agv(1,i)==-1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
          %periptosi na kateythinetai dexia
          if Dir_agv(1,i)==1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=turn_col(i)+1;
          elseif Dir_agv(1,i)==1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
```

```
    x_agv(i)=x_agv(i)+dt*Dir_agv(1,i)*Velocity(i);
    y_agv(i)=y_agv(i)+dt*Dir_agv(2,i)*Velocity(i);

Distance(i)=Distance(i)+abs(dt*Dir_agv(1,i)*Velocity(i))+abs(dt*Dir_agv(2,i)*Velocity(i));









    x_collision(i)=0;
    y_collision(i)=0;
  end



  if mod(loop,1440)==0
    loop
  end


  if mod(loop,14400)==0
    loop
    k=k+1;
    containers(k)=containers_loaded
    containers_loaded=0;
    hold_time1=hold_time;

  end
  for j=1:4
    if load_time_crane(j)==0
      idle_crane(j)=idle_crane(j)+1;
    end
  end
end
average=sum(containers)/(loop*dt*4)
idle_rate_cranes=((sum(idle_crane)/4)/loop)*100
idle_rate_agv=((sum(idle_agv)/no_of_agv)/loop)*100
%IRINI ADDITION
total_time=loop*dt
containers

save senarioA1AGV30

% IRINI ADDITION
```

% SENARIO A2

```
a=4;
b=4;
no_of_agv=30;
no_of_trucks=0;
dt=0.25;
hold_time=0;
Time_to_load_container=100;%sec
Time_to_load_crane=48;%sec
Time_to_load_gate=48;%sec

load_time_crane=zeros(1,4);
load_time=zeros(1,no_of_trucks+no_of_agv);
Dir_agv=zeros(2,no_of_trucks+no_of_agv);
Dir_agv(2,:)=-1;
Load_agv=zeros(1,no_of_trucks+no_of_agv);
%destination = 1 (gate) -1 (container)
Dest=zeros(1,no_of_trucks+no_of_agv);
Dest(:)=-1;
%assigned = 0 (not assigned) 1 (assigned)
Assign=zeros(1,no_of_trucks+no_of_agv);
x_cont=zeros(1,no_of_trucks+no_of_agv);
y_cont=zeros(1,no_of_trucks+no_of_agv);
x_agv=zeros(1,no_of_trucks+no_of_agv);
y_agv=zeros(1,no_of_trucks+no_of_agv);
x_dest=zeros(1,no_of_trucks+no_of_agv);
x_entrance=zeros(1,no_of_trucks+no_of_agv);
x_origin=zeros(1,no_of_trucks+no_of_agv);
y_dest=zeros(1,no_of_trucks+no_of_agv);
y_entrance=zeros(1,no_of_trucks+no_of_agv);
y_origin=zeros(1,no_of_trucks+no_of_agv);
Velocity=zeros(1,no_of_trucks+no_of_agv);
col=zeros(1,no_of_trucks+no_of_agv);
turn_col=zeros(1,no_of_trucks+no_of_agv);
queue_col=zeros(1,no_of_trucks+no_of_agv);
length_col=2;
length_alarm=2;
y_collision=zeros(1,no_of_trucks+no_of_agv);
x_collision=zeros(1,no_of_trucks+no_of_agv);
x_gate=[265.6 275.6 285.6 295.6];
y_gate=[185.5 188 188 185.5];
x_cranes=[277.8 287.8 297.8 307.8];
y_cranes=[-51.3 -53.8 -53.8 -51.3];
go_to_crane=zeros(4,no_of_agv);
idle_crane=zeros(1,4);
idle_agv=zeros(1,no_of_agv+no_of_trucks);
crane=zeros(1,4);
gate=zeros(1,4);
containers_loaded=0;
containers_unloaded=0;
containers_per_crane=zeros(1,4);
k=0;
counter=0;
```

```
Tcounter=0;
counter2=0;
Distance=zeros(1,no_of_trucks+no_of_agv);
pinakas_elegxoy=[1
  2
  3
  4
  5
  6
];
%diafora xronon mexri to destination tou i
T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
%xronos gia na paei to i apo arxi se telos xoris collision
T2=zeros(no_of_agv+no_of_trucks);
%diafora xronon gia na min exoume collision sto queue
T3=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
for i=1:no_of_agv+no_of_trucks
  for j=1:no_of_agv+no_of_trucks
    T3(i,j)=18;
  end
end
%xronos poy exei ginei anathesi sto i se loops(to collision lambanetai ipopsin)
T4=zeros(no_of_agv+no_of_trucks);
l=1;
ll=0;

N=14400*10;

% IRINI addition

storage=(1:160);
for i=1:160
   storage(i)=15;
end

% End of IRINI


containers=zeros(1,N/14400);


% IRINI addition

loop=0;

while sum(storage) >0

 loop=loop+1;

 % End of IRINI
```

```
for i=1:no_of_trucks+no_of_agv
  %statements for trucks
  if i<=no_of_trucks
    if Assign(i)==0
      %random epilogi container
      rand_cont=1+fix(a*b*10*rand);
      x_cont(i)=container(rand_cont,1);
      y_cont(i)=container(rand_cont,2);
      x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2)-1)*12.2;
      y_gate_center=(b*5*6.1+(b-1)*12.2+2*12.2);
      x_agv(i)=x_gate_center;
      y_agv(i)=y_gate_center;
      Load_agv(i)=+1;
      load_time(i)=0;
      Assign(i)=1;
    end
    if Dest(i)==-1
      x_origin(i)=x_gate_center;
      y_origin(i)=y_gate_center;
      x_dest(i)=x_cont(i);
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=0;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=146.4;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<439.2
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=439.2;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 439.2<x_cont(i) & x_cont(i)<585.6
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=585.6;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==-1 & Assign(i)==1
      if Load_agv(i)==+1
        Velocity(i)=2.3;%m/sec
      else
        Velocity(i)=4.6;%m/sec
      end
```

```
    if y_agv(i)>y_dest(i)
      Dir_agv(:,i)=[
        0
        -1
      ];

    else
      y_agv(i)=y_dest(i);  %diorthotiki kinisi
      if abs(x_agv(i)-x_dest(i))>1.2
        Dir_agv(:,i)=[
          sign(x_dest(i)-x_origin(i))
          0
        ];
      elseif load_time(i)<Time_to_load_container
        x_agv(i)=x_dest(i);          %diorthotiki kinisi
        Velocity(i)=0;
        load_time(i)=load_time(i)+dt;
      else
        Dest(i)=1;
        x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
        load_time(i)=0;
        Load_agv(i)=0;
      end
    end
  elseif Dest(i)==1 & Assign(i)==1
    if Load_agv(i)==+1
      Velocity(i)=2.3;%m/sec
    else
      Velocity(i)=4.6;%m/sec
    end
    if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    end
    if y_agv(i)<183 & abs(x_agv(i)-x_dest(i))<1.2
      x_agv(i)=x_dest(i);  %diorthotiki kinisi
      Dir_agv(:,i)=[
        0
        1
      ];
    end
    if abs(x_agv(i)-x_gate_center)>1.2 & abs(y_agv(i)-183)<1.2
      y_agv(i)=183; %diorthotiki kinisi
      Dir_agv(:,i)=[
        sign(x_gate_center-x_origin(i))
        0
      ];
    end
    if abs(y_gate_center-y_agv(i))<1.2 & abs(x_agv(i)-x_gate_center)<25
      if x_agv(i)-x_gate_center<0
```

```
        if gate(1)==0
          y_agv(i)=185.5;
          Dir_agv(:,i)=[
            1
            0
          ];
        elseif gate(2)==0
          y_agv(i)=188;
          Dir_agv(:,i)=[
            1
            0
          ];
        else
          Velocity(i)=0;
        end
      else
        if gate(4)==0
          y_agv(i)=185.5;
          Dir_agv(:,i)=[
            -1
            0
          ];
        elseif crane(3)==0
          y_agv(i)=188;
          Dir_agv(:,i)=[
            -1
            0
          ];
        else
          Velocity(i)=0;
        end
      end
    end
    for j=1:4
      if abs(x_agv(i)-x_gate(j))<1.2 & abs(y_agv(i)-y_gate(j))<1.2
        x_agv(i)=x_gate(j);
        gate(j)=1;
        Velocity(i)=0;
        if load_time(i)<Time_to_load_gate
          load_time(i)=load_time(i)+dt;
        else
          x_agv(i)=x_gate_center;
          y_agv(i)=y_gate_center;
          load_time(i)=0;
          gate(j)=0;
          containers_unloaded=containers_unloaded+1;
          Dest(i)=-1;
          Assign(i)=0;
        end
      end
    end
  end
```

```
        %statements for agv's
      else
        if Assign(i)==1
          T4(i,1)=T4(i,1)+1;
        end
        if loop>1350
          ll=0;
        end
        if loop>ll*50
          while Assign(i)==0
          counter=counter+1;
          %random epilogi container
          % rand_cont=1+fix(a*b*10*rand);
% IRINI ADDITION
         kcounter=160*15;
                  while kcounter>0
               rand_cont=1+fix(a*b*10*rand);
                if storage(rand_cont)>0
                     kcounter=0;

                else
                                          kcounter=kcounter-1;
                   end
           end
loop
 sum(storage)


% END OF IRINI ADDITION
          x_cont(i)=container(rand_cont,1);
          y_cont(i)=container(rand_cont,2);
          x_crane_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
          y_crane_center=-4*12.2;
          x_agv(i)=x_crane_center;
          y_agv(i)=y_crane_center;
          Load_agv(i)=0;
          load_time(i)=0;
          pinakas_elegxoy(l)=i;
          if x_cont(i)<134.2
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)+12.2+280))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>134.2 & x_cont(i)<280.6
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)-134.2+133.6))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>292.8 & x_cont(i)<427
            T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+427+109.2))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>427
```

```
            T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+573.4+255.6))/2.3)+Time_to_load_container;
          end
          T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
          for ii=1:no_of_agv+no_of_trucks
            for jj=1:no_of_agv+no_of_trucks
              T3(ii,jj)=24;
            end
          end

          for j=1:no_of_agv+no_of_trucks

            if j~=i & j~=0
              if Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)<x_crane_center & x_cont(i)<x_crane_center
                if x_dest(j)-x_cont(i)<0
                 T1(i,j)=1;
                elseif x_cont(i)<146 & x_dest(j)>146
                  T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-146)/2.3)+Time_to_load_container-
load_time(j));
                else
                  T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
                end
              elseif Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)>x_crane_center & x_cont(i)>x_crane_center
                if x_dest(j)-x_cont(i)>0
                  T1(i,j)=1;
                elseif x_cont(i)>439 & x_dest(j)<439
                  T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+439)/2.3)+Time_to_load_container-
load_time(j));
                else
                  T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
                end
              end


              %IRINI add

              if T1(i,j)<=0
                 T1(i,j)=0;
              end

              %IRINI end
```

```
        if x_cont(i)<x_crane_center & x_dest(j)<x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if x_cont(i)>x_crane_center & x_dest(j)>x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if T3(i,j)>24
          Tcounter=Tcounter+T3(i,j);
          counter2=counter2+1;
          end
      end
    end




    %prakatv ginetai i epilogi sinthikis T3>24 i T3<24
    if min(T3(i,:))<=24 | counter2>(160*15)
      counter2=0;
      Assign(i)=1;
      storage(rand_cont)=storage(rand_cont)-1;
      ll=ll+1;
      counter=0;
      l=l+1;
      if l==7
        l=1;
      end
    end
    end
  end
  if Dest(i)==-1
    x_origin(i)=x_crane_center;
    y_origin(i)=y_crane_center;
    x_dest(i)=x_cont(i);
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=-12.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=134.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<427
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=427;
```

```
  y_dest(i)=y_cont(i);
end
if Dest(i)==1 & 427<x_cont(i) & x_cont(i)<573.4
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=573.4;
  y_dest(i)=y_cont(i);
end
if Dest(i)==-1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;%m/sec
  end
  if y_agv(i)<y_dest(i)
    Dir_agv(:,i)=[
      0
      1
    ];

  else
    y_agv(i)=y_dest(i);  %diorthotiki kinisi
    if abs(x_agv(i)-x_dest(i))>1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    elseif load_time(i)<Time_to_load_container

      x_agv(i)=x_dest(i);          %diorthotiki kinisi
      Velocity(i)=0;
      load_time(i)=load_time(i)+dt;
    else
      Dest(i)=-Dest(i);
      load_time(i)=0;
      Load_agv(i)=1;
    end
  end
elseif Dest(i)==1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;
  end
  if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
    Dir_agv(:,i)=[
      sign(x_dest(i)-x_origin(i))
      0
    ];
  end
  if y_agv(i)>-48.8 & abs(x_agv(i)-x_dest(i))<1.2
    x_agv(i)=x_dest(i);  %diorthotiki kinisi
```

```
  Dir_agv(:,i)=[
    0
    -1
  ];
end
if abs(x_agv(i)-x_crane_center)>1.2 & abs(y_agv(i)+48.8)<1.2
  y_agv(i)=-48.8; %diorthotiki kinisi
  Dir_agv(:,i)=[
    sign(x_crane_center-x_origin(i))
    0
  ];
end
if y_agv(i)==y_crane_center & abs(x_agv(i)-x_crane_center)<25
  if x_agv(i)-x_crane_center<0
    if crane(1)==0 | (load_time_crane(1)>44.25 & go_to_crane(1,:)==0)
      crane(1)=1;
      go_to_crane(1,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        1
        0
      ];
    elseif crane(2)==0 | (load_time_crane(2)>40 & go_to_crane(2,:)==0)
      crane(2)=1;
      go_to_crane(2,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        1
        0
      ];
    else
      x_agv(i)=x_crane_center-25;
      Velocity(i)=0;
      idle_agv(i)=idle_agv(i)+1;
    end
  else
    if crane(4)==0 | (load_time_crane(4)>44.25 & go_to_crane(4,:)==0)
      crane(4)=1;
      go_to_crane(4,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        -1
        0
      ];
    elseif crane(3)==0 | (load_time_crane(3)>40 & go_to_crane(3,:)==0)
      crane(3)=1;
      go_to_crane(3,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        -1
        0
      ];
```

```
          else
            x_agv(i)=x_crane_center+25;
            Velocity(i)=0;
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
      for j=1:4
        if abs(x_agv(i)-x_cranes(j))<1.2 & y_agv(i)==y_cranes(j)
          x_agv(i)=x_cranes(j);
          go_to_crane(j,i)=0;
          Velocity(i)=0;
          if load_time(i)<Time_to_load_crane
            load_time(i)=load_time(i)+dt;
            load_time_crane(j)=load_time(i);
            hold_time=hold_time+dt;
          else
            x_agv(i)=x_crane_center;
            y_agv(i)=y_crane_center;
            load_time(i)=0;
            load_time_crane(j)=0;
            crane(j)=0;
            containers_loaded=containers_loaded+1;
            containers_per_crane(j)=containers_per_crane(j)+1;
            Dest(i)=-1;
            Assign(i)=0;
            T4(i,1)=0;
          end
        end
      end
    end
  end
end


%check for collision ston y

if Dir_agv(2,i)==1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if y_agv(j)-y_agv(i)<length_col & y_agv(j)-y_agv(i)>0 & x_agv(j)==x_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end
if Dir_agv(2,i)==-1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
```

```
        if y_agv(i)-y_agv(j)<length_col & y_agv(i)-y_agv(j)>0 & x_agv(j)==x_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end

  %check for collision ston x

  if Dir_agv(1,i)==-1
    for j=1:no_of_trucks+no_of_agv
      if j~=i
        if x_agv(i)-x_agv(j)<length_col & x_agv(i)-x_agv(j)>0 & y_agv(j)==y_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
            queue_col(i)=queue_col(i)+1;
          end

          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end
  if Dir_agv(1,i)==1
    for j=1:no_of_trucks+no_of_agv
      if j~=i
        if x_agv(j)-x_agv(i)<length_col & x_agv(j)-x_agv(i)>0 & y_agv(j)==y_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
            queue_col(i)=queue_col(i)+1;
          end

          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end
  %elegxos strofis
  if abs(y_dest(i)-y_agv(i))<length_alarm & y_dest(i)~=y_agv(i) & Dest(i)==-1
    y_collision(i)=1;
    x_collision(i)=0;
  elseif abs(x_dest(i)-x_agv(i))<length_alarm & x_dest(i)~=x_agv(i) & Dest(i)==+1
```

```
      y_collision(i)=0;
      x_collision(i)=1;
    end
    %gia ton kentriko dromo
    if y_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na katevainei kai na thelei na stripsei aristera
          if abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i)&
Dir_agv(2,i)==-1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)>(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          %periptosi na anevainei kai na thelei na stripsei aristera
          elseif abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i) &
Dir_agv(2,i)==1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)<(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(j)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
    %gia tis ipoloipes diastayroseis
    if x_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na kateythinetai aristera
          if Dir_agv(1,i)==-1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          elseif Dir_agv(1,i)==-1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
          %periptosi na kateythinetai dexia
          if Dir_agv(1,i)==1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=turn_col(i)+1;
          elseif Dir_agv(1,i)==1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
```

```
    x_agv(i)=x_agv(i)+dt*Dir_agv(1,i)*Velocity(i);
    y_agv(i)=y_agv(i)+dt*Dir_agv(2,i)*Velocity(i);

Distance(i)=Distance(i)+abs(dt*Dir_agv(1,i)*Velocity(i))+abs(dt*Dir_agv(2,i)*Velocity(i));









    x_collision(i)=0;
    y_collision(i)=0;
  end



  if mod(loop,1440)==0
    loop
  end


  if mod(loop,14400)==0
    loop
    k=k+1;
    containers(k)=containers_loaded
    containers_loaded=0;
    hold_time1=hold_time;

  end
  for j=1:4
    if load_time_crane(j)==0
      idle_crane(j)=idle_crane(j)+1;
    end
  end
end
average=sum(containers)/(loop*dt*4)
idle_rate_cranes=((sum(idle_crane)/4)/loop)*100
idle_rate_agv=((sum(idle_agv)/no_of_agv)/loop)*100
%IRINI ADDITION
total_time=loop*dt
containers

save senarioA2AGV30

% IRINI ADDITION
```

% SENARIO A3

```
a=4;
b=4;
no_of_agv=30;
no_of_trucks=0;
dt=0.25;
hold_time=0;
Time_to_load_container=100;%sec
Time_to_load_crane=48;%sec
Time_to_load_gate=48;%sec

load_time_crane=zeros(1,4);
load_time=zeros(1,no_of_trucks+no_of_agv);
Dir_agv=zeros(2,no_of_trucks+no_of_agv);
Dir_agv(2,:)=-1;
Load_agv=zeros(1,no_of_trucks+no_of_agv);
%destination = 1 (gate) -1 (container)
Dest=zeros(1,no_of_trucks+no_of_agv);
Dest(:)=-1;
%assigned = 0 (not assigned) 1 (assigned)
Assign=zeros(1,no_of_trucks+no_of_agv);
x_cont=zeros(1,no_of_trucks+no_of_agv);
y_cont=zeros(1,no_of_trucks+no_of_agv);
x_agv=zeros(1,no_of_trucks+no_of_agv);
y_agv=zeros(1,no_of_trucks+no_of_agv);
x_dest=zeros(1,no_of_trucks+no_of_agv);
x_entrance=zeros(1,no_of_trucks+no_of_agv);
x_origin=zeros(1,no_of_trucks+no_of_agv);
y_dest=zeros(1,no_of_trucks+no_of_agv);
y_entrance=zeros(1,no_of_trucks+no_of_agv);
y_origin=zeros(1,no_of_trucks+no_of_agv);
Velocity=zeros(1,no_of_trucks+no_of_agv);
col=zeros(1,no_of_trucks+no_of_agv);
turn_col=zeros(1,no_of_trucks+no_of_agv);
queue_col=zeros(1,no_of_trucks+no_of_agv);
length_col=2;
length_alarm=2;
y_collision=zeros(1,no_of_trucks+no_of_agv);
x_collision=zeros(1,no_of_trucks+no_of_agv);
x_gate=[265.6 275.6 285.6 295.6];
y_gate=[185.5 188 188 185.5];
x_cranes=[277.8 287.8 297.8 307.8];
y_cranes=[-51.3 -53.8 -53.8 -51.3];
go_to_crane=zeros(4,no_of_agv);
idle_crane=zeros(1,4);
idle_agv=zeros(1,no_of_agv+no_of_trucks);
crane=zeros(1,4);
gate=zeros(1,4);
containers_loaded=0;
containers_unloaded=0;
containers_per_crane=zeros(1,4);
k=0;
```

```
counter=0;
Tcounter=0;
counter2=0;
Distance=zeros(1,no_of_trucks+no_of_agv);
pinakas_elegxoy=[1
   2
   3
   4
   5
   6
];
%diafora xronon mexri to destination tou i
T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
%xronos gia na paei to i apo arxi se telos xoris collision
T2=zeros(no_of_agv+no_of_trucks);
%diafora xronon gia na min exoume collision sto queue
T3=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
for i=1:no_of_agv+no_of_trucks
   for j=1:no_of_agv+no_of_trucks
      T3(i,j)=18;
   end
end
%xronos poy exei ginei anathesi sto i se loops(to collision lambanetai ipopsin)
T4=zeros(no_of_agv+no_of_trucks);
l=1;
ll=0;

N=14400*10;

% IRINI addition

storage=(1:160);
for i=1:160
    storage(i)=15;
end

% End of IRINI


containers=zeros(1,N/14400);


% IRINI addition

loop=0;

while sum(storage) >0

 loop=loop+1;

 % End of IRINI
```

```
for i=1:no_of_trucks+no_of_agv
  %statements for trucks
  if i<=no_of_trucks
    if Assign(i)==0
      %random epilogi container
      rand_cont=1+fix(a*b*10*rand);
      x_cont(i)=container(rand_cont,1);
      y_cont(i)=container(rand_cont,2);
      x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2)-1)*12.2;
      y_gate_center=(b*5*6.1+(b-1)*12.2+2*12.2);
      x_agv(i)=x_gate_center;
      y_agv(i)=y_gate_center;
      Load_agv(i)=+1;
      load_time(i)=0;
      Assign(i)=1;
    end
    if Dest(i)==-1
      x_origin(i)=x_gate_center;
      y_origin(i)=y_gate_center;
      x_dest(i)=x_cont(i);
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=0;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=146.4;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<439.2
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=439.2;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 439.2<x_cont(i) & x_cont(i)<585.6
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=585.6;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==-1 & Assign(i)==1
      if Load_agv(i)==+1
        Velocity(i)=2.3;%m/sec
      else
        Velocity(i)=4.6;%m/sec
```

```
    end
    if y_agv(i)>y_dest(i)
      Dir_agv(:,i)=[
        0
        -1
      ];

    else
      y_agv(i)=y_dest(i);  %diorthotiki kinisi
      if abs(x_agv(i)-x_dest(i))>1.2
        Dir_agv(:,i)=[
          sign(x_dest(i)-x_origin(i))
          0
        ];
      elseif load_time(i)<Time_to_load_container
        x_agv(i)=x_dest(i);          %diorthotiki kinisi
        Velocity(i)=0;
        load_time(i)=load_time(i)+dt;
      else
        Dest(i)=1;
        x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
        load_time(i)=0;
        Load_agv(i)=0;
      end
    end
  elseif Dest(i)==1 & Assign(i)==1
    if Load_agv(i)==+1
      Velocity(i)=2.3;%m/sec
    else
      Velocity(i)=4.6;%m/sec
    end
    if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    end
    if y_agv(i)<183 & abs(x_agv(i)-x_dest(i))<1.2
      x_agv(i)=x_dest(i);  %diorthotiki kinisi
      Dir_agv(:,i)=[
        0
        1
      ];
    end
    if abs(x_agv(i)-x_gate_center)>1.2 & abs(y_agv(i)-183)<1.2
      y_agv(i)=183; %diorthotiki kinisi
      Dir_agv(:,i)=[
        sign(x_gate_center-x_origin(i))
        0
      ];
    end
    if abs(y_gate_center-y_agv(i))<1.2 & abs(x_agv(i)-x_gate_center)<25
```

```
  if x_agv(i)-x_gate_center<0
    if gate(1)==0
      y_agv(i)=185.5;
      Dir_agv(:,i)=[
        1
        0
      ];
    elseif gate(2)==0
      y_agv(i)=188;
      Dir_agv(:,i)=[
        1
        0
      ];
    else
      Velocity(i)=0;
    end
  else
    if gate(4)==0
      y_agv(i)=185.5;
      Dir_agv(:,i)=[
        -1
        0
      ];
    elseif crane(3)==0
      y_agv(i)=188;
      Dir_agv(:,i)=[
        -1
        0
      ];
    else
      Velocity(i)=0;
    end
  end
end
for j=1:4
  if abs(x_agv(i)-x_gate(j))<1.2 & abs(y_agv(i)-y_gate(j))<1.2
    x_agv(i)=x_gate(j);
    gate(j)=1;
    Velocity(i)=0;
    if load_time(i)<Time_to_load_gate
      load_time(i)=load_time(i)+dt;
    else
      x_agv(i)=x_gate_center;
      y_agv(i)=y_gate_center;
      load_time(i)=0;
      gate(j)=0;
      containers_unloaded=containers_unloaded+1;
      Dest(i)=-1;
      Assign(i)=0;
    end
  end
end
```

```
        end
        %statements for agv's
      else
        if Assign(i)==1
          T4(i,1)=T4(i,1)+1;
        end
        if loop>1350
          ll=0;
        end
        if loop>ll*50
          while Assign(i)==0
          counter=counter+1;
          %random epilogi container
          % rand_cont=1+fix(a*b*10*rand);
% IRINI ADDITION
          kcounter=160*15;
                  while kcounter>0
               rand_cont=1+fix(a*b*10*rand);
                if storage(rand_cont)>0
                     kcounter=0;

                  else
                                          kcounter=kcounter-1;
                    end
           end
loop
 sum(storage)


% END OF IRINI ADDITION
          x_cont(i)=container(rand_cont,1);
          y_cont(i)=container(rand_cont,2);
          x_crane_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
          y_crane_center=-4*12.2;
          x_agv(i)=x_crane_center;
          y_agv(i)=y_crane_center;
          Load_agv(i)=0;
          load_time(i)=0;
          pinakas_elegxoy(l)=i;
          if x_cont(i)<134.2
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)+12.2+280))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>134.2 & x_cont(i)<280.6
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)-134.2+133.6))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>292.8 & x_cont(i)<427
            T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+427+109.2))/2.3)+Time_to_load_container;
          end
```

```
        if x_cont(i)>427
          T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+573.4+255.6))/2.3)+Time_to_load_container;
        end
        T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
        for ii=1:no_of_agv+no_of_trucks
          for jj=1:no_of_agv+no_of_trucks
            T3(ii,jj)=24;
          end
        end

        for j=1:no_of_agv+no_of_trucks

          if j~=i & j~=0
            if Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)<x_crane_center & x_cont(i)<x_crane_center
              if x_dest(j)-x_cont(i)<0
               T1(i,j)=1;
              elseif x_cont(i)<146 & x_dest(j)>146
                T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-146)/2.3)+Time_to_load_container-
load_time(j));
              else
                T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
              end
            elseif Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)>x_crane_center & x_cont(i)>x_crane_center
              if x_dest(j)-x_cont(i)>0
                T1(i,j)=1;
              elseif x_cont(i)>439 & x_dest(j)<439
                T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+439)/2.3)+Time_to_load_container-
load_time(j));
              else
                T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
              end
            end


            %IRINI add

            if T1(i,j)<=0
               T1(i,j)=0;
            end

            %IRINI end
```

```
        if x_cont(i)<x_crane_center & x_dest(j)<x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if x_cont(i)>x_crane_center & x_dest(j)>x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if (T3(i,j)+T1(i,j))<24
          Tcounter=Tcounter+T3(i,j);
          counter2=counter2+1;
          end
      end
    end




    %prakatv ginetai i epilogi sinthikis T3>24 i T3<24
    if min(T3(i,:)+T1(i,:))>=24 | counter2>(160*15)
      counter2=0;
      Assign(i)=1;
      storage(rand_cont)=storage(rand_cont)-1;
      ll=ll+1;
      counter=0;
      l=l+1;
      if l==7
        l=1;
      end
    end
    end
  end
  if Dest(i)==-1
    x_origin(i)=x_crane_center;
    y_origin(i)=y_crane_center;
    x_dest(i)=x_cont(i);
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=-12.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=134.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<427
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
```

```
  x_dest(i)=427;
  y_dest(i)=y_cont(i);
end
if Dest(i)==1 & 427<x_cont(i) & x_cont(i)<573.4
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=573.4;
  y_dest(i)=y_cont(i);
end
if Dest(i)==-1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;%m/sec
  end
  if y_agv(i)<y_dest(i)
    Dir_agv(:,i)=[
      0
      1
    ];

  else
    y_agv(i)=y_dest(i);  %diorthotiki kinisi
    if abs(x_agv(i)-x_dest(i))>1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    elseif load_time(i)<Time_to_load_container

      x_agv(i)=x_dest(i);          %diorthotiki kinisi
      Velocity(i)=0;
      load_time(i)=load_time(i)+dt;
    else
      Dest(i)=-Dest(i);
      load_time(i)=0;
      Load_agv(i)=1;
    end
  end
elseif Dest(i)==1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;
  end
  if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
    Dir_agv(:,i)=[
      sign(x_dest(i)-x_origin(i))
      0
    ];
  end
  if y_agv(i)>-48.8 & abs(x_agv(i)-x_dest(i))<1.2
```

```
  x_agv(i)=x_dest(i);  %diorthotiki kinisi
  Dir_agv(:,i)=[
    0
    -1
  ];
end
if abs(x_agv(i)-x_crane_center)>1.2 & abs(y_agv(i)+48.8)<1.2
  y_agv(i)=-48.8; %diorthotiki kinisi
  Dir_agv(:,i)=[
    sign(x_crane_center-x_origin(i))
    0
  ];
end
if y_agv(i)==y_crane_center & abs(x_agv(i)-x_crane_center)<25
  if x_agv(i)-x_crane_center<0
    if crane(1)==0 | (load_time_crane(1)>44.25 & go_to_crane(1,:)==0)
      crane(1)=1;
      go_to_crane(1,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        1
        0
      ];
    elseif crane(2)==0 | (load_time_crane(2)>40 & go_to_crane(2,:)==0)
      crane(2)=1;
      go_to_crane(2,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        1
        0
      ];
    else
      x_agv(i)=x_crane_center-25;
      Velocity(i)=0;
      idle_agv(i)=idle_agv(i)+1;
    end
  else
    if crane(4)==0 | (load_time_crane(4)>44.25 & go_to_crane(4,:)==0)
      crane(4)=1;
      go_to_crane(4,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        -1
        0
      ];
    elseif crane(3)==0 | (load_time_crane(3)>40 & go_to_crane(3,:)==0)
      crane(3)=1;
      go_to_crane(3,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        -1
        0
```

```
            ];
          else
            x_agv(i)=x_crane_center+25;
            Velocity(i)=0;
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
      for j=1:4
        if abs(x_agv(i)-x_cranes(j))<1.2 & y_agv(i)==y_cranes(j)
          x_agv(i)=x_cranes(j);
          go_to_crane(j,i)=0;
          Velocity(i)=0;
          if load_time(i)<Time_to_load_crane
            load_time(i)=load_time(i)+dt;
            load_time_crane(j)=load_time(i);
            hold_time=hold_time+dt;
          else
            x_agv(i)=x_crane_center;
            y_agv(i)=y_crane_center;
            load_time(i)=0;
            load_time_crane(j)=0;
            crane(j)=0;
            containers_loaded=containers_loaded+1;
            containers_per_crane(j)=containers_per_crane(j)+1;
            Dest(i)=-1;
            Assign(i)=0;
            T4(i,1)=0;
          end
        end
      end
    end
  end
end


%check for collision ston y

if Dir_agv(2,i)==1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if y_agv(j)-y_agv(i)<length_col & y_agv(j)-y_agv(i)>0 & x_agv(j)==x_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end
if Dir_agv(2,i)==-1
  for j=1:no_of_trucks+no_of_agv
```

```
      if j~=i
        if y_agv(i)-y_agv(j)<length_col & y_agv(i)-y_agv(j)>0 & x_agv(j)==x_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end

  %check for collision ston x

  if Dir_agv(1,i)==-1
    for j=1:no_of_trucks+no_of_agv
      if j~=i
        if x_agv(i)-x_agv(j)<length_col & x_agv(i)-x_agv(j)>0 & y_agv(j)==y_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
            queue_col(i)=queue_col(i)+1;
          end

          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end
  if Dir_agv(1,i)==1
    for j=1:no_of_trucks+no_of_agv
      if j~=i
        if x_agv(j)-x_agv(i)<length_col & x_agv(j)-x_agv(i)>0 & y_agv(j)==y_agv(i)
          Velocity(i)=Velocity(j);
          col(i)=1+col(i);
          if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
            queue_col(i)=queue_col(i)+1;
          end

          if Velocity(j)==0
            idle_agv(i)=idle_agv(i)+1;
          end
        end
      end
    end
  end
  %elegxos strofis
  if abs(y_dest(i)-y_agv(i))<length_alarm & y_dest(i)~=y_agv(i) & Dest(i)==-1
    y_collision(i)=1;
    x_collision(i)=0;
```

```
    elseif abs(x_dest(i)-x_agv(i))<length_alarm & x_dest(i)~=x_agv(i) & Dest(i)==+1
      y_collision(i)=0;
      x_collision(i)=1;
    end
    %gia ton kentriko dromo
    if y_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na katevainei kai na thelei na stripsei aristera
          if abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i)&
Dir_agv(2,i)==-1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)>(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          %periptosi na anevainei kai na thelei na stripsei aristera
          elseif abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i) &
Dir_agv(2,i)==1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)<(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(j)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
    %gia tis ipoloipes diastayroseis
    if x_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na kateythinetai aristera
          if Dir_agv(1,i)==-1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          elseif Dir_agv(1,i)==-1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
          %periptosi na kateythinetai dexia
          if Dir_agv(1,i)==1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=turn_col(i)+1;
          elseif Dir_agv(1,i)==1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
```

```
    x_agv(i)=x_agv(i)+dt*Dir_agv(1,i)*Velocity(i);
    y_agv(i)=y_agv(i)+dt*Dir_agv(2,i)*Velocity(i);

Distance(i)=Distance(i)+abs(dt*Dir_agv(1,i)*Velocity(i))+abs(dt*Dir_agv(2,i)*Velocity(i));









    x_collision(i)=0;
    y_collision(i)=0;
  end



  if mod(loop,1440)==0
    loop
  end


  if mod(loop,14400)==0
    loop
    k=k+1;
    containers(k)=containers_loaded
    containers_loaded=0;
    hold_time1=hold_time;

  end
  for j=1:4
    if load_time_crane(j)==0
      idle_crane(j)=idle_crane(j)+1;
    end
  end
end
average=sum(containers)/(loop*dt*4)
idle_rate_cranes=((sum(idle_crane)/4)/loop)*100
idle_rate_agv=((sum(idle_agv)/no_of_agv)/loop)*100
%IRINI ADDITION
total_time=loop*dt
containers

save senarioA3AGV30

% IRINI ADDITION
```

% SENARIO A4

```
a=4;
b=4;
no_of_agv=30;
no_of_trucks=0;
dt=0.25;
hold_time=0;
Time_to_load_container=100;%sec
Time_to_load_crane=48;%sec
Time_to_load_gate=48;%sec

load_time_crane=zeros(1,4);
load_time=zeros(1,no_of_trucks+no_of_agv);
Dir_agv=zeros(2,no_of_trucks+no_of_agv);
Dir_agv(2,:)=-1;
Load_agv=zeros(1,no_of_trucks+no_of_agv);
%destination = 1 (gate) -1 (container)
Dest=zeros(1,no_of_trucks+no_of_agv);
Dest(:)=-1;
%assigned = 0 (not assigned) 1 (assigned)
Assign=zeros(1,no_of_trucks+no_of_agv);
x_cont=zeros(1,no_of_trucks+no_of_agv);
y_cont=zeros(1,no_of_trucks+no_of_agv);
x_agv=zeros(1,no_of_trucks+no_of_agv);
y_agv=zeros(1,no_of_trucks+no_of_agv);
x_dest=zeros(1,no_of_trucks+no_of_agv);
x_entrance=zeros(1,no_of_trucks+no_of_agv);
x_origin=zeros(1,no_of_trucks+no_of_agv);
y_dest=zeros(1,no_of_trucks+no_of_agv);
y_entrance=zeros(1,no_of_trucks+no_of_agv);
y_origin=zeros(1,no_of_trucks+no_of_agv);
Velocity=zeros(1,no_of_trucks+no_of_agv);
col=zeros(1,no_of_trucks+no_of_agv);
turn_col=zeros(1,no_of_trucks+no_of_agv);
queue_col=zeros(1,no_of_trucks+no_of_agv);
length_col=2;
length_alarm=2;
y_collision=zeros(1,no_of_trucks+no_of_agv);
x_collision=zeros(1,no_of_trucks+no_of_agv);
x_gate=[265.6 275.6 285.6 295.6];
y_gate=[185.5 188 188 185.5];
x_cranes=[277.8 287.8 297.8 307.8];
y_cranes=[-51.3 -53.8 -53.8 -51.3];
go_to_crane=zeros(4,no_of_agv);
idle_crane=zeros(1,4);
idle_agv=zeros(1,no_of_agv+no_of_trucks);
crane=zeros(1,4);
gate=zeros(1,4);
containers_loaded=0;
containers_unloaded=0;
containers_per_crane=zeros(1,4);
k=0;
```

```
counter=0;
Tcounter=0;
counter2=0;
Distance=zeros(1,no_of_trucks+no_of_agv);
pinakas_elegxoy=[1
   2
   3
   4
   5
   6
];
%diafora xronon mexri to destination tou i
T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
%xronos gia na paei to i apo arxi se telos xoris collision
T2=zeros(no_of_agv+no_of_trucks);
%diafora xronon gia na min exoume collision sto queue
T3=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
for i=1:no_of_agv+no_of_trucks
   for j=1:no_of_agv+no_of_trucks
      T3(i,j)=18;
   end
end
%xronos poy exei ginei anathesi sto i se loops(to collision lambanetai ipopsin)
T4=zeros(no_of_agv+no_of_trucks);
l=1;
ll=0;

N=14400*10;

% IRINI addition

storage=(1:160);
for i=1:160
    storage(i)=15;
end

% End of IRINI


containers=zeros(1,N/14400);


% IRINI addition

loop=0;

while sum(storage) >0

 loop=loop+1;

 % End of IRINI
```

```
for i=1:no_of_trucks+no_of_agv
  %statements for trucks
  if i<=no_of_trucks
    if Assign(i)==0
      %random epilogi container
      rand_cont=1+fix(a*b*10*rand);
      x_cont(i)=container(rand_cont,1);
      y_cont(i)=container(rand_cont,2);
      x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2)-1)*12.2;
      y_gate_center=(b*5*6.1+(b-1)*12.2+2*12.2);
      x_agv(i)=x_gate_center;
      y_agv(i)=y_gate_center;
      Load_agv(i)=+1;
      load_time(i)=0;
      Assign(i)=1;
    end
    if Dest(i)==-1
      x_origin(i)=x_gate_center;
      y_origin(i)=y_gate_center;
      x_dest(i)=x_cont(i);
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=0;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=146.4;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<439.2
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=439.2;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==1 & 439.2<x_cont(i) & x_cont(i)<585.6
      x_origin(i)=x_cont(i);
      y_origin(i)=y_cont(i);
      x_dest(i)=585.6;
      y_dest(i)=y_cont(i);
    end
    if Dest(i)==-1 & Assign(i)==1
      if Load_agv(i)==+1
        Velocity(i)=2.3;%m/sec
      else
        Velocity(i)=4.6;%m/sec
```

```
    end
    if y_agv(i)>y_dest(i)
      Dir_agv(:,i)=[
        0
        -1
      ];

    else
      y_agv(i)=y_dest(i);  %diorthotiki kinisi
      if abs(x_agv(i)-x_dest(i))>1.2
        Dir_agv(:,i)=[
          sign(x_dest(i)-x_origin(i))
          0
        ];
      elseif load_time(i)<Time_to_load_container
        x_agv(i)=x_dest(i);          %diorthotiki kinisi
        Velocity(i)=0;
        load_time(i)=load_time(i)+dt;
      else
        Dest(i)=1;
        x_gate_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
        load_time(i)=0;
        Load_agv(i)=0;
      end
    end
  elseif Dest(i)==1 & Assign(i)==1
    if Load_agv(i)==+1
      Velocity(i)=2.3;%m/sec
    else
      Velocity(i)=4.6;%m/sec
    end
    if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    end
    if y_agv(i)<183 & abs(x_agv(i)-x_dest(i))<1.2
      x_agv(i)=x_dest(i);  %diorthotiki kinisi
      Dir_agv(:,i)=[
        0
        1
      ];
    end
    if abs(x_agv(i)-x_gate_center)>1.2 & abs(y_agv(i)-183)<1.2
      y_agv(i)=183; %diorthotiki kinisi
      Dir_agv(:,i)=[
        sign(x_gate_center-x_origin(i))
        0
      ];
    end
    if abs(y_gate_center-y_agv(i))<1.2 & abs(x_agv(i)-x_gate_center)<25
```

```
  if x_agv(i)-x_gate_center<0
    if gate(1)==0
      y_agv(i)=185.5;
      Dir_agv(:,i)=[
        1
        0
      ];
    elseif gate(2)==0
      y_agv(i)=188;
      Dir_agv(:,i)=[
        1
        0
      ];
    else
      Velocity(i)=0;
    end
  else
    if gate(4)==0
      y_agv(i)=185.5;
      Dir_agv(:,i)=[
        -1
        0
      ];
    elseif crane(3)==0
      y_agv(i)=188;
      Dir_agv(:,i)=[
        -1
        0
      ];
    else
      Velocity(i)=0;
    end
  end
end
for j=1:4
  if abs(x_agv(i)-x_gate(j))<1.2 & abs(y_agv(i)-y_gate(j))<1.2
    x_agv(i)=x_gate(j);
    gate(j)=1;
    Velocity(i)=0;
    if load_time(i)<Time_to_load_gate
      load_time(i)=load_time(i)+dt;
    else
      x_agv(i)=x_gate_center;
      y_agv(i)=y_gate_center;
      load_time(i)=0;
      gate(j)=0;
      containers_unloaded=containers_unloaded+1;
      Dest(i)=-1;
      Assign(i)=0;
    end
  end
end
```

```
        end
        %statements for agv's
      else
        if Assign(i)==1
          T4(i,1)=T4(i,1)+1;
        end
        if loop>1350
          ll=0;
        end
        if loop>ll*50
          while Assign(i)==0
            counter=counter+1;

          %random epilogi container
          % rand_cont=1+fix(a*b*10*rand);
% IRINI ADDITION
         kcounter=160*15;
                  while kcounter>0
              rand_cont=1+fix(a*b*10*rand);
               if storage(rand_cont)>0
                    kcounter=0;

                else
                                          kcounter=kcounter-1;
                   end
           end
loop
 sum(storage)


% END OF IRINI ADDITION
          x_cont(i)=container(rand_cont,1);
          y_cont(i)=container(rand_cont,2);
          x_crane_center=(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
          y_crane_center=-4*12.2;
          x_agv(i)=x_crane_center;
          y_agv(i)=y_crane_center;
          Load_agv(i)=0;
          load_time(i)=0;
          pinakas_elegxoy(l)=i;
          if x_cont(i)<134.2
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)+12.2+280))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>134.2 & x_cont(i)<280.6
            T2(i,1)=(((y_cont(i)-y_crane_center)+(x_crane_center-x_cont(i)))/4.6)+(((y_cont(i)-
y_crane_center)+(x_cont(i)-134.2+133.6))/2.3)+Time_to_load_container;
          end
          if x_cont(i)>292.8 & x_cont(i)<427
            T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+427+109.2))/2.3)+Time_to_load_container;
```

```
        end
        if x_cont(i)>427
          T2(i,1)=(((y_cont(i)-y_crane_center)+(-
x_crane_center+x_cont(i)))/4.6)+(((y_cont(i)-y_crane_center)+(-
x_cont(i)+573.4+255.6))/2.3)+Time_to_load_container;
        end
        T1=ones(no_of_agv+no_of_trucks,no_of_agv+no_of_trucks);
        for ii=1:no_of_agv+no_of_trucks
          for jj=1:no_of_agv+no_of_trucks
            T3(ii,jj)=24;
          end
        end

        for j=1:no_of_agv+no_of_trucks

          if j~=i & j~=0
            if Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)<x_crane_center & x_cont(i)<x_crane_center
              if x_dest(j)-x_cont(i)<0
               T1(i,j)=1;
              elseif x_cont(i)<146 & x_dest(j)>146
                T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-146)/2.3)+Time_to_load_container-
load_time(j));
              else
                T1(i,j)=(y_cont(i)-y_crane_center+x_crane_center-x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)+x_agv(j)-x_dest(j))/4.6)+((x_dest(j)-x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
              end
            elseif Dest(j)==-1 & Assign(j)~=0 & y_dest(j)==y_cont(i) &
x_dest(j)>x_crane_center & x_cont(i)>x_crane_center
              if x_dest(j)-x_cont(i)>0
                T1(i,j)=1;
              elseif x_cont(i)>439 & x_dest(j)<439
                T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+439)/2.3)+Time_to_load_container-
load_time(j));
              else
                T1(i,j)=(y_cont(i)-y_crane_center-x_crane_center+x_cont(i))/4.6-(((y_dest(j)-
y_agv(j)-x_agv(j)+x_dest(j))/4.6)+((-x_dest(j)+x_cont(i))/2.3)+Time_to_load_container-
load_time(j));
              end
            end


            %IRINI add

            if T1(i,j)<=0
               T1(i,j)=0;
            end

            %IRINI end
```

```
        if x_cont(i)<x_crane_center & x_dest(j)<x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if x_cont(i)>x_crane_center & x_dest(j)>x_crane_center
          T3(i,j)=abs(T2(i,1)-(T2(j,1)-0.25*T4(j,1))-T1(i,j));
        end
        if (T3(i,j)+T1(i,j))>24
          Tcounter=Tcounter+T3(i,j);
          counter2=counter2+1;
          end
      end
    end




    %prakatv ginetai i epilogi sinthikis T3>24 i T3<24
    if min(T3(i,:)+T1(i,:))<=24 | counter2>(160*15)
      counter2=0;
      Assign(i)=1;
      storage(rand_cont)=storage(rand_cont)-1;
      ll=ll+1;
      counter=0;
      l=l+1;
      if l==7
        l=1;
      end
    end
    end
  end
  if Dest(i)==-1
    x_origin(i)=x_crane_center;
    y_origin(i)=y_crane_center;
    x_dest(i)=x_cont(i);
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 0<x_cont(i) & x_cont(i)<134.2
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=-12.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 134.2<x_cont(i) & x_cont(i)<280.6
    x_origin(i)=x_cont(i);
    y_origin(i)=y_cont(i);
    x_dest(i)=134.2;
    y_dest(i)=y_cont(i);
  end
  if Dest(i)==1 & 280.6<x_cont(i) & x_cont(i)<427
    x_origin(i)=x_cont(i);
```

```
  y_origin(i)=y_cont(i);
  x_dest(i)=427;
  y_dest(i)=y_cont(i);
end
if Dest(i)==1 & 427<x_cont(i) & x_cont(i)<573.4
  x_origin(i)=x_cont(i);
  y_origin(i)=y_cont(i);
  x_dest(i)=573.4;
  y_dest(i)=y_cont(i);
end
if Dest(i)==-1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;%m/sec
  end
  if y_agv(i)<y_dest(i)
    Dir_agv(:,i)=[
      0
      1
    ];

  else
    y_agv(i)=y_dest(i);  %diorthotiki kinisi
    if abs(x_agv(i)-x_dest(i))>1.2
      Dir_agv(:,i)=[
        sign(x_dest(i)-x_origin(i))
        0
      ];
    elseif load_time(i)<Time_to_load_container

      x_agv(i)=x_dest(i);          %diorthotiki kinisi
      Velocity(i)=0;
      load_time(i)=load_time(i)+dt;
    else
      Dest(i)=-Dest(i);
      load_time(i)=0;
      Load_agv(i)=1;
    end
  end
elseif Dest(i)==1 & Assign(i)==1
  if Load_agv(i)==+1
    Velocity(i)=2.3;%m/sec
  else
    Velocity(i)=4.6;
  end
  if abs(x_agv(i)-x_dest(i))>1.2 & abs(y_agv(i)-y_cont(i))<1.2
    Dir_agv(:,i)=[
      sign(x_dest(i)-x_origin(i))
      0
    ];
  end
```

```
if y_agv(i)>-48.8 & abs(x_agv(i)-x_dest(i))<1.2
  x_agv(i)=x_dest(i);  %diorthotiki kinisi
  Dir_agv(:,i)=[
    0
    -1
  ];
end
if abs(x_agv(i)-x_crane_center)>1.2 & abs(y_agv(i)+48.8)<1.2
  y_agv(i)=-48.8; %diorthotiki kinisi
  Dir_agv(:,i)=[
    sign(x_crane_center-x_origin(i))
    0
  ];
end
if y_agv(i)==y_crane_center & abs(x_agv(i)-x_crane_center)<25
  if x_agv(i)-x_crane_center<0
    if crane(1)==0 | (load_time_crane(1)>44.25 & go_to_crane(1,:)==0)
      crane(1)=1;
      go_to_crane(1,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        1
        0
      ];
    elseif crane(2)==0 | (load_time_crane(2)>40 & go_to_crane(2,:)==0)
      crane(2)=1;
      go_to_crane(2,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        1
        0
      ];
    else
      x_agv(i)=x_crane_center-25;
      Velocity(i)=0;
      idle_agv(i)=idle_agv(i)+1;
    end
  else
    if crane(4)==0 | (load_time_crane(4)>44.25 & go_to_crane(4,:)==0)
      crane(4)=1;
      go_to_crane(4,i)=1;
      y_agv(i)=-51.3;
      Dir_agv(:,i)=[
        -1
        0
      ];
    elseif crane(3)==0 | (load_time_crane(3)>40 & go_to_crane(3,:)==0)
      crane(3)=1;
      go_to_crane(3,i)=1;
      y_agv(i)=-53.8;
      Dir_agv(:,i)=[
        -1
```

```
            0
          ];
        else
          x_agv(i)=x_crane_center+25;
          Velocity(i)=0;
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
    for j=1:4
      if abs(x_agv(i)-x_cranes(j))<1.2 & y_agv(i)==y_cranes(j)
        x_agv(i)=x_cranes(j);
        go_to_crane(j,i)=0;
        Velocity(i)=0;
        if load_time(i)<Time_to_load_crane
          load_time(i)=load_time(i)+dt;
          load_time_crane(j)=load_time(i);
          hold_time=hold_time+dt;
        else
          x_agv(i)=x_crane_center;
          y_agv(i)=y_crane_center;
          load_time(i)=0;
          load_time_crane(j)=0;
          crane(j)=0;
          containers_loaded=containers_loaded+1;
          containers_per_crane(j)=containers_per_crane(j)+1;
          Dest(i)=-1;
          Assign(i)=0;
          T4(i,1)=0;
        end
      end
    end
  end
end


%check for collision ston y

if Dir_agv(2,i)==1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if y_agv(j)-y_agv(i)<length_col & y_agv(j)-y_agv(i)>0 & x_agv(j)==x_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end
if Dir_agv(2,i)==-1
```

```
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if y_agv(i)-y_agv(j)<length_col & y_agv(i)-y_agv(j)>0 & x_agv(j)==x_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end

%check for collision ston x

if Dir_agv(1,i)==-1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if x_agv(i)-x_agv(j)<length_col & x_agv(i)-x_agv(j)>0 & y_agv(j)==y_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
          queue_col(i)=queue_col(i)+1;
        end

        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end
if Dir_agv(1,i)==1
  for j=1:no_of_trucks+no_of_agv
    if j~=i
      if x_agv(j)-x_agv(i)<length_col & x_agv(j)-x_agv(i)>0 & y_agv(j)==y_agv(i)
        Velocity(i)=Velocity(j);
        col(i)=1+col(i);
        if y_agv(i)==-51.3 | y_agv(i)==-53.8 | y_agv(i)==-48.8
          queue_col(i)=queue_col(i)+1;
        end

        if Velocity(j)==0
          idle_agv(i)=idle_agv(i)+1;
        end
      end
    end
  end
end
%elegxos strofis
if abs(y_dest(i)-y_agv(i))<length_alarm & y_dest(i)~=y_agv(i) & Dest(i)==-1
  y_collision(i)=1;
```

```
      x_collision(i)=0;
    elseif abs(x_dest(i)-x_agv(i))<length_alarm & x_dest(i)~=x_agv(i) & Dest(i)==+1
      y_collision(i)=0;
      x_collision(i)=1;
    end
    %gia ton kentriko dromo
    if y_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na katevainei kai na thelei na stripsei aristera
          if abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i)&
Dir_agv(2,i)==-1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)>(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          %periptosi na anevainei kai na thelei na stripsei aristera
          elseif abs(y_agv(j)-y_agv(i))<length_col & Dir_agv(2,j)==-Dir_agv(2,i) &
Dir_agv(2,i)==1 & abs(x_agv(j)-x_agv(i))<13 &
x_dest(i)<(fix((a+1)/2))*134.2+(fix((a+1)/2))*12.2;
            Velocity(j)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
    %gia tis ipoloipes diastayroseis
    if x_collision(i)==1
      for j=1:no_of_trucks+no_of_agv
        if j~=i
          %periptosi na kateythinetai aristera
          if Dir_agv(1,i)==-1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          elseif Dir_agv(1,i)==-1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
          %periptosi na kateythinetai dexia
          if Dir_agv(1,i)==1 & Load_agv(i)==0 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<13 & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=turn_col(i)+1;
          elseif Dir_agv(1,i)==1 & Load_agv(i)==1 & abs(y_agv(j)-y_dest(i))<length_col &
abs(x_agv(j)-x_dest(i))<length_col & Dir_agv(2,j)~=0
            Velocity(i)=0;
            turn_col(i)=1+turn_col(i);
          end
        end
      end
    end
```

```
    x_agv(i)=x_agv(i)+dt*Dir_agv(1,i)*Velocity(i);
    y_agv(i)=y_agv(i)+dt*Dir_agv(2,i)*Velocity(i);

Distance(i)=Distance(i)+abs(dt*Dir_agv(1,i)*Velocity(i))+abs(dt*Dir_agv(2,i)*Velocity(i));




    x_collision(i)=0;
    y_collision(i)=0;
  end



  if mod(loop,1440)==0
    loop
  end

  if mod(loop,14400)==0
    loop
    k=k+1;
    containers(k)=containers_loaded
    containers_loaded=0;
    hold_time1=hold_time;

  end
  for j=1:4
    if load_time_crane(j)==0
      idle_crane(j)=idle_crane(j)+1;
    end
  end
end
average=sum(containers)/(loop*dt*4)
idle_rate_cranes=((sum(idle_crane)/4)/loop)*100
idle_rate_agv=((sum(idle_agv)/no_of_agv)/loop)*100
%IRINI ADDITION
total_time=loop*dt
containers
save senarioA4AGV30
% IRINI ADDITION
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] US Department of Transportation, Office of Intermodalism, "The impact of changes in ship design on transportation infrastructure and operations", February 1998.
[2] C. I. Liu, H. Jula, and P. Ioannou, "Automated guided vehicle system for two container yard layouts", submitted to *Transport Research Part-C.*
[3] C. I. Liu, H. Jula, K. Vukadinovic and P. Ioannou, "Comparing different technologies for container movement in marine container terminals", in *Proceedings of the third IEEE International Conference on ITS,* 2000, pp. 488-493
[4] P. A. Ioannou,E. B. Kosmatopoulos, H. Jula, A. Collinge, C. I. Liu, A. Asef-Vaziri, E. Dougherty, "Cargo Handling Technologies Final Report", submitted to the Center for Commercial Deployment of Transportation Technologies.
[5] PRS Inc, "Sealift technology development program. Assesment of cargo handling technology", Dot/Marad Tech. Rep. MA-RD-840-93003, August 1993.
[6] M. A. Jordan, "Dockside container cranes", *Proc. Of Ports'95,* March 1995.
[7] A. Bhimani, C. Morris and S. Karasuda, "Dockside container crane design for the 21st century", *Container Efficiency Conference,* Singapore, March 1996.
[8] URL:http://www.wagner.com
[9] URL:http://www.august-design.com
[10] URL:http://www.kyamk.fi
[11] United Nations Conference on Trade and Development, UNCTAD/SDD/MT/2, October 1992.
[12] C. I. Liu, H. Jula, and P. Ioannou, K. Vukadinovic, "Automated Guided Vehicle System for two Container Yard Layouts".
[13] C. I. Liu, H. Jula, and P. Ioannou, "Design, Simulation and Evaluation of Automated Container Terminals"/
[14] C. I. Liu, H. Jula, and P. Ioannou, E. B. Kosmatopoulos, "Container Terminals using Automated Shuttles Driven by Linear Motors".
[15] Towery S.A., et al., "Planning for maximum efficiency at Norfolk International Terminals", JWD report, AAPA, Tampa, Florida, 1996
[16] Larsen R., and Moses J., "AVCS for ports: An automation study for Norfolk International Terminals".
[17] C. I. Liu, H. Jula, K. Vukadinovic, P. Ioannou, H. Pourmohammadi, "Advanced material handling", Technical Report, Center for Advanced Transportation Technologies, University of Southern California, Oct 2000.