

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

**Προσομοίωση κεντρικής αποθήκης διανομής  
προϊόντων και βελτίωση της χωροθέτησης και  
διακίνησης εμπορευμάτων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Μιλτιάδης Μυλερός**

Εξεταστική επιτροπή:  
Βασίλης Κουϊκόγλου, επιβλέπων  
Ευστράτιος Ιωαννίδης  
Γεώργιος Τσιναράκης

Χανιά, 2022

© 2022 Μιλτιάδης Μυλερός

## Περιεχόμενα

1	Εισαγωγή.....	4
1.1	Αντικείμενο και μεθοδολογία της διπλωματικής εργασίας .....	4
1.2	Περιγραφή μελετώμενου συστήματος .....	5
1.3	Πεδίο εφαρμογής και σκοπός της εν λόγω διπλωματικής εργασίας .....	6
2	Κεντρικές Αποθήκες Διανομής Προϊόντων.....	7
2.1	Εισαγωγή.....	7
2.2	Το κέντρο διανομής και η χρησιμότητά του.....	7
2.3	Διαδικασίες στο εσωτερικό της Αποθήκης.....	8
2.3.1	Παραλαβή .....	9
2.3.2	Τακτοποίηση .....	9
2.3.3	Διαλογή παραγγελιών .....	9
2.3.4	Έλεγχος και συσκευασία.....	9
2.3.5	Αποστολή.....	10
2.4	Λογισμικό διαχείρισης εφοδιαστικής αλυσίδας.....	10
2.5	Συστήματα Διαχείρισης Αποθηκών .....	11
2.6	Αποθήκευση και εξοπλισμός διαχείρισης.....	12
2.6.1	Αποθήκευση.....	12
2.6.2	Εξοπλισμός διαχείρισης .....	14
3	Βασικές Λειτουργίες και Συνιστώσες του Μελετώμενου Κέντρου Διανομής .....	17
3.1	Χώρος.....	17
3.2	Προϊόντα.....	20
3.3	Συνθήκες αποθήκευσης προϊόντων .....	24
3.4	Εξυπηρετητές .....	25
4	Προσομοίωση του Συστήματος .....	27
4.1	Εισαγωγή.....	27
4.2	Περιγραφή .....	27
4.3	Μοντέλο διακριτών γεγονότων .....	29
4.4	Παραμετροποίηση .....	32
4.4.1	Αφίξεις.....	32
4.4.2	Χρόνος εξυπηρέτησης.....	32
4.5	Αλγόριθμος βελτίωσης .....	36
4.5.1	Συνάρτηση findlocation.....	36
5	Αριθμητικά Αποτελέσματα.....	39
5.1	Υπολογιστικός χρόνος εκτέλεσης.....	39
5.2	Μέτρα απόδοσης.....	39
6	Συμπεράσματα.....	4241
7	Βιβλιογραφία.....	43
	Παράρτημα Α: Κώδικες στη Matlab.....	44
	SIM (αλγόριθμος προσομοίωσης υπάρχοντος μοντέλου) .....	44
	SIM2 (αλγόριθμος προσομοίωσης μοντέλου που εξετάζουμε).....	48
	Συνάρτηση PAL.....	54
	Συνάρτηση findstime.....	56

Συνάρτηση findlocation.....	57
-----------------------------	----

# 1 Εισαγωγή

## 1.1 Αντικείμενο και μεθοδολογία της διπλωματικής εργασίας

Στην εργασία αυτή παρουσιάζεται το σύστημα διαχείρισης, αποθήκευσης και παραγγελιοληψίας μίας κεντρικής αποθήκης, αξιολογείται η λειτουργία του μέσω ενός αλγορίθμου προσομοίωσης και εξετάζεται κατά πόσο μπορούν να βελτιωθούν τα μέτρα απόδοσης της αποθήκης με κατάλληλη κατανομή των προϊόντων εντός αυτής.

Οι αποθήκες είναι χώροι όπου προϊόντα διαφόρων τύπων φθάνουν από τους προμηθευτές, εκφορτώνονται και αποθηκεύονται προσωρινά και αργότερα διανέμονται σε πελάτες για να ικανοποιήσουν αντίστοιχες παραγγελίες τους. Στις κεντρικές αποθήκες όπως αυτή που εξετάζεται εδώ, οι πελάτες είναι αποθέτες, δηλαδή αποκεντρωμένες και ανεξάρτητες αποθήκες, κάθε μία με το δικό της πελατολόγιο.

Η εκτέλεση των παραγγελιών καθώς και η αποθήκευση των προϊόντων που καταφθάνουν λαμβάνει χώρα σε σταθμούς διαλογής και περατώνεται από ανθρώπινες οντότητες, τους διαλογείς (pickers), οι οποίοι αποτελούν σημαντικό παράγοντα για την ορθή λειτουργία του αποθηκευτικού συστήματος.

Προσομοίωση είναι η περιγραφή με τη βοήθεια υπολογιστή της εξέλιξης της λειτουργίας ενός συστήματος. Οι αλγόριθμοι προσομοίωσης αποτελούν σημαντικά εργαλεία για τη μελέτη συστημάτων, γιατί παρέχουν την δυνατότητα πειραματισμών με τις παραμέτρους του συστήματος με στόχο τον βέλτιστο σχεδιασμό τους. Η προσομοίωση μπορεί να περιγράψει με λεπτομέρεια αρκετά συστήματα τα οποία επιδέχονται θεωρητικής ανάλυσης μόνον μετά από χονδροειδείς απλουστευτικές παραδοχές. Ένας ακόμα λόγος που η προσομοίωση αποτελεί εργαλείο για τον μελετητή είναι πως τις περισσότερες φορές η δοκιμή ενός νέου σχεδίου λειτουργίας σε πραγματικό σύστημα απαιτεί χρόνο και έχει υψηλό κόστος και αρνητικό αντίκτυπο στη λειτουργία του συστήματος. Για παράδειγμα οι πειραματισμοί στον τρόπο



λειτουργίας μίας κεντρικής αποθήκης μπορεί να οδηγήσουν σε καθυστερήσεις στη διάθεση του προϊόντος στον πελάτη και σε υποβάθμιση της ποιότητάς του λόγω κακής διαχείρισης.

Στην εργασία αυτή αναπτύσσονται αλγόριθμοι προσομοίωσης διακριτών γεγονότων. Οι αλγόριθμοι αυτοί καταγράφουν την εξέλιξη του συστήματος, υπολογίζοντας τον χρόνο στον οποίο θα σημειωθεί η επόμενη σημαντική μεταβολή κατάστασης κάθε συνιστώσας του. Για παράδειγμα για την συνιστώσα που περιγράφει ένα υποσύστημα μεταφοράς σημειώνεται η τρέχουσα κατάσταση (πχ. υλικά που μεταφέρονται, θέση-αφετηρία, προορισμός) και υπολογίζεται ο χρόνος επόμενου γεγονότος (πχ. της άφιξης στον προορισμό). Αν οι χρόνοι μετακίνησης προϊόντων, άφιξης παραγγελιών και τροφοδοσίας είναι τυχαίοι, τότε χρησιμοποιούνται για τους υπολογισμούς οι λεγόμενες γεννήτριες τυχαίων αριθμών, αλλιώς χρησιμοποιούνται πρότυποι χρόνοι (παράμετροι του συστήματος). Το ίδιο επαναλαμβάνεται για κάθε συνιστώσα και η προσομοίωση ξεκινά τη στιγμή 0. Ο αλγόριθμος συνοψίζεται σε τρία βήματα:

- 1) Αποθηκεύονται στη μνήμη του υπολογιστή ο χρόνος επόμενου γεγονότος για κάθε συνιστώσα και η τρέχουσα κατάστασή της. Υπολογίζεται το συντομότερο γεγονός και σημειώνεται η συνιστώσα στην οποία θα συμβεί το γεγονός αυτό.
- 2) Ακολούθως εκτελείται ένα χρονικό άλμα στον χρόνο εμφάνισης του συντομότερου γεγονότος. Υπολογίζονται η νέα κατάσταση και οι χρόνοι επόμενων γεγονότων για όσες συνιστώσες επηρεάζονται από την εμφάνιση του συγκεκριμένου γεγονότος.
- 3) Η αλληλουχία βημάτων 1-2 επαναλαμβάνεται (με το νέο συντομότερο γεγονός, νέο χρονικό άλμα, νέα κατάσταση, κλπ) μέχρις ότου διανυθεί ένα προκαθορισμένο χρονικό διάστημα προσομοίωσης της λειτουργίας του συστήματος, οπότε τερματίζεται η προσομοίωση.

Με το πέρας της προσομοίωσης, ο υπολογιστής εξάγει στατιστικές πληροφορίες με τη μορφή δεικτών ή μέτρων απόδοσης του συστήματος, όπως η μέση καθυστέρηση στην ικανοποίηση παραγγελιών, ποσοστά χρόνων αδράνειας ή απασχόλησης προσωπικού και εξοπλισμού, το κόστος λειτουργίας κλπ.

## 1.2 Περιγραφή μελετώμενου συστήματος

Το σύστημα που εξετάζουμε αποτελείται από τις εγκαταστάσεις μίας κεντρικής αποθήκης (με καθορισμένο σχεδιασμό και διαρρύθμιση), το απόθεμα με τις ιδιότητές του (συνθήκες αποθήκευσης, μεγέθη κ.ά.), και τις κινήσεις υλικού που κλήθηκε να πραγματοποιήσει το ανθρώπινο δυναμικό εντός αυτής σε συγκεκριμένο χρονικό διάστημα. Το σύνολο των παραπάνω πληροφοριών παραχωρήθηκε από τον όμιλο FDL Group που δραστηριοποιείται για πάνω από 20 χρόνια στον κλάδο του εφοδιασμού (αποθήκευσης, συντήρησης και διανομής) για τρίτους (3PL – 3<sup>rd</sup> Party Logistics) και αφορά χρονικό διάστημα 5 ημερών, από 16 έως 20 Απριλίου 2018.

Τα προϊόντα είναι κατά κύριο λόγο τρόφιμα με διαφορετικές ανάγκες αποθήκευσης και η αποθήκη αποτελείται από κλιματιζόμενους και μη χώρους. Κάθε μονάδα προϊόντος προορίζεται για έναν από τους συνολικά έξι (6) αποθέτες που τροφοδοτεί η κεντρική αποθήκη, οι οποίοι μπορεί να έχουν διαφορετική ή και παρόμοια παρουσία στην αγορά, πράγμα που μπορεί να δημιουργήσει ανάλογες απαιτήσεις στη διαχείριση των προϊόντων τους. Για παράδειγμα, ένας από τους αποθέτες έχει την απαίτηση τα προϊόντα του να μην αποθηκεύονται στους ίδιους χώρους με αυτά των ανταγωνιστών του.

Οι κινήσεις που καλείται να πραγματοποιήσει το έμπυχο δυναμικό μέσα σε αυτό το σύστημα, είναι η παραλαβή και αποθεματοποίηση των προϊόντων που καταφθάνουν καθημερινά στην αποθήκη και η διαλογή ολόκληρων παλετών, κιβωτίων ή ακόμα και μεμονωμένων τεμαχίων για

την αποστολή μίας παραγγελίας. Για την επίτευξη αυτού του στόχου, υπάρχουν μέσα όπως ηλεκτρικοί παλετοφόροι και ανυψωτικά μηχανήματα για τη συλλογή και μεταφορά μεγάλων φορτίων. Ακόμα υπάρχουν μικρότερες ομάδες ανθρώπων που ασχολούνται αποκλειστικά με την διαλογή προϊόντων κατά παρτίδες ή τεμάχια σε συγκεκριμένους σταθμούς.



### 1.3 Πεδίο εφαρμογής και σκοπός της εν λόγω διπλωματικής εργασίας

Η εργασία εστιάζει στις βασικές λειτουργίες που λαμβάνουν χώρα στο εσωτερικό της κεντρικής αποθήκης και παρουσιάζει δύο αλγόριθμους προσομοίωσης διακριτών γεγονότων. Έναν με βάση τον τρόπο που ήδη λειτουργεί το σύστημα και έναν βασισμένο σε ένα μοντέλο αποθήκης τριών ταχυτήτων και δοκιμαστικές παραμετροποιήσεις με στόχο την βελτίωση των μέτρων απόδοσης στην διαλογή παλετών. Το μοντέλο τριών ταχυτήτων περιλαμβάνει μία ομαδοποίηση θέσεων και προϊόντων, ανάλογα με την απόστασή τους από τα σημεία διαλογής και ανάλογα με την κινητικότητα που έχουν εντός της αποθήκης αντίστοιχα. Η ιδέα αυτού του μοντέλου προήλθε από το βιβλίο των Bartholdi III and Hackman (2016).

Τα αποτελέσματα θα παρουσιαστούν, θα αναλυθούν και η υπεροχή του του μοντέλου έναντι της τρέχουσας πολιτικής είτε θα επιβεβαιωθεί είτε θα απορριφθεί. Ωστόσο, αυτό που είναι σημαντικό είναι η δομή, οι παραμετροποιήσεις και η χρησιμότητα ενός τέτοιου αλγορίθμου στη διαδικασία του πειραματισμού πολλαπλών σεναρίων με σκοπό τη ορθή λήψη αποφάσεων.

## 2 Κεντρικές Αποθήκες Διανομής Προϊόντων

### 2.1 Εισαγωγή

Μια εφοδιαστική αλυσίδα (supply chain) ή αλλιώς δίκτυο εφοδιαστικής (logistics network) αποτελείται από όλα τα στάδια που εμπλέκονται, έμμεσα ή άμεσα, στην ικανοποίηση των απαιτήσεων του πελάτη (Μαρινάκης και Μυγδαλάς 2008), είτε αναφερόμαστε σε προϊόν, είτε σε παροχή υπηρεσιών. Επειδή ο στόχος κάθε εταιρίας είναι η κερδοφορία και συγχρόνως η ικανοποίηση των όλο και αυξανόμενων, λόγω του ανταγωνισμού, απαιτήσεων του πελάτη, η διαχείριση και η διοίκηση μιας εφοδιαστικής αλυσίδας χρήζει επιστημονικής έρευνας ως προς τη λήψη αποφάσεων και τη βελτιστοποίησή της. Ορισμένες από τις αποφάσεις που θα κληθεί να πάρει ο αποφασίζων για ένα τέτοιο σύστημα είναι οι εξής:

- Εξυπηρέτηση πελατών ως προς την ποιότητα του τελικού προϊόντος (customer service)
- Υπηρεσίες μετά την πώληση και χορήγηση ανταλλακτικών (parts and service Support)
- Συσκευασία και μεταποίηση έτοιμων προϊόντων.
- Τροφοδοσία υλικών και μέσων για τη διαχείριση προϊόντων (procurement)
- Διαχείριση πρώτων υλών για την υποστήριξη της παραγωγής (material handling)
- Διαχείριση αποθηκών και κέντρων διανομής (warehousing and distribution center management)
- Στρατηγική τοποθέτηση αποθηκών και μονάδων παραγωγής στο χάρτη.
- Σχεδιασμός και διαχείριση διανομών (distribution network management).
- Μηχανογράφηση και ηλεκτρονική υποστήριξη σε όλο το μήκος της αλυσίδας.
- Εκπαίδευση προσωπικού κ.α.

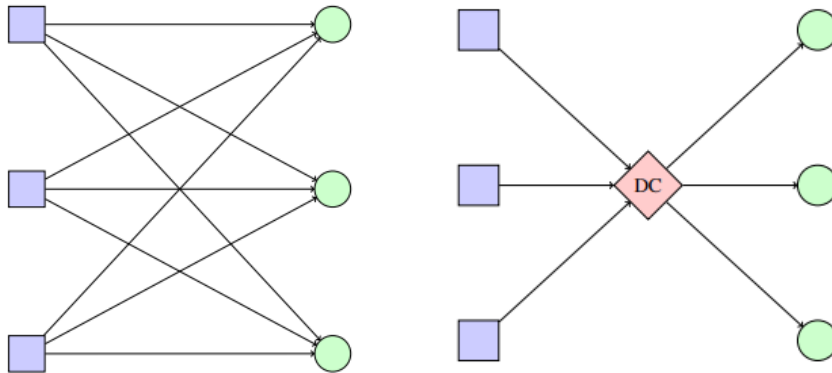
Στην εργασία θα εστιάσουμε στη διαχείριση αποθηκών και κέντρων διανομής και στις διεργασίες που γίνονται εντός αυτών.

### 2.2 Το κέντρο διανομής και η χρησιμότητά του

Όπως έχει αναφερθεί, μία αποθήκη-κέντρο διανομής, εν συντομία DC (distribution center), αποτελεί το τμήμα της εφοδιαστικής αλυσίδας όπου σε οργανωμένες εγκαταστάσεις αποθηκεύονται προσωρινά διάφορα υλικά και προϊόντα μέχρις ότου χρειαστεί να ανακαλεστούν είτε για την παραγωγή, είτε για την διάθεσή τους στην αγορά.

Η χρησιμότητα ενός κέντρου διανομής διαφαίνεται αν σκεφτεί κανείς το πλήθος των σχέσεων που υπάρχει μεταξύ παραγωγής και ενδιάμεσου παραλήπτη ενός προϊόντος. Για παράδειγμα, σε ένα σύστημα με  $m$  προμηθευτές και  $n$  καταστήματα-αποθέτες, χωρίς την ύπαρξη κέντρου διανομής, οι απαιτούμενες διαδρομές για την τροφοδοσία όλων των καταστημάτων είναι  $m \times n$ . Αντίθετα, αν υπάρχει μεταξύ αυτών ένα κέντρο διανομής, οι διαδρομές αυτές είναι  $m + n$ . Οι δύο αναπαραστάσεις φαίνονται στο Σχ. 2.1.





Σχήμα 2.1 Παρουσίαση συστημάτων με και χωρίς ύπαρξη κέντρου διανομής  
(Bartholdi III and Hackman 1998-2016)

Οι αποθήκες λοιπόν βρίσκονται σε κάθε στάδιο της εφοδιαστικής αλυσίδας όπου τα προϊόντα παύουν να κινούνται και αποθηκεύονται μέχρι την μετακίνησή τους στο επόμενο στάδιο ή την διανομή τους στους τελικούς καταναλωτές. Τα προϊόντα αυτά δεσμεύουν εκτός από χώρο και χρόνο προκειμένου να τα διαχειριστεί το ανθρώπινο δυναμικό. Η επιλογή και το πλήθος των μέσων διαλογής, της χωροθέτησης και της διαρρύθμισης του χώρου ανάλογα με τις ανάγκες του αποθέτη επηρεάζουν σημαντικά τους χρόνους διαχείρισης των προϊόντων και στη συνέχεια την απόδοση ενός κέντρου διανομής.

### 2.3 Διαδικασίες στο εσωτερικό της Αποθήκης

Μία αποθήκη αναδιοργανώνει και ανασυσκευάζει τα προϊόντα που διαχειρίζεται. Το προϊόν καταφθάνει συσκευασμένο σε παλέτες και φεύγει ανασυσκευασμένο σε παλέτες ή και μικρότερες συσκευασίες (πχ κιβώτιο, κυτίο, τεμάχιο). Είναι σημαντικό λοιπόν για τη δυναμικότητα μίας αποθήκης να βελτιστοποιήσει τέτοιες διεργασίες, ιδιαίτερα σε κέντρα που διαχειρίζονται μεγάλο αριθμό παραγγελιών σε τεμάχια τα οποία βρίσκονται σε όλο και μεγαλύτερες συσκευασίες (πχ σοκολάτες). Παρόλο που οι αποθήκες και τα κέντρα διανομής εξυπηρετούν πελάτες διαφορετικής δυναμικότητας, οι διεργασίες που λαμβάνουν χώρα εντός αυτών έχουν αρκετές ομοιότητες όσον αφορά τη ροή των προϊόντων. Ένας αρχικός διαχωρισμός των διεργασιών γίνεται κατά εισερχόμενα και εξερχόμενα προϊόντα και ακολουθούν επί μέρους διαδικασίες που φαίνονται παρακάτω και περιγράφονται λεπτομερώς στη συνέχεια:

- Διεργασίες Εισερχομένων
  - Παραλαβή
  - Τακτοποίηση εντός αποθήκης
- Διεργασίες Εξερχομένων
  - Διαλογή παραγγελιών
  - Έλεγχος, συσκευασία και αποστολή

### 2.3.1 Παραλαβή

Η παραλαβή ξεκινά μόλις καταφθάσει το όχημα με τις παλέτες στην αποθήκη. Εκείνη τη στιγμή, πραγματοποιείται η εκφόρτωση του φορτηγού και, στην περίπτωση του κέντρου DC που εξετάζεται, τοποθέτηση των προϊόντων στο χώρο παραλαβών μέσω ηλεκτρικών παλετοφόρων προκειμένου να πραγματοποιηθεί έλεγχος όσον αφορά την ποσότητα και την κατάσταση του προϊόντος που παραλήφθηκε. Τέλος, τοποθετείται σε κάθε παλέτα ετικέτα με μοναδικό κωδικό (SSCC - Serial Shipping Container Code) που αντιστοιχίζεται με τα χαρακτηριστικά του προϊόντος. Από αυτή τη στιγμή και έπειτα, η παλέτα αναφέρεται και κυκλοφορεί εντός της αποθήκης με αυτόν τον κωδικό.

### 2.3.2 Τακτοποίηση

Στη συνέχεια, κατάλληλα περνοφόρα ανυψωτικά οχήματα αναλαμβάνουν να πάρουν τις παραληφθείσες παλέτες και να τις τοποθετήσουν σε κατάλληλα διαμορφωμένες θέσεις που τηρούν τις προδιαγραφές του προϊόντος με βάση τις συνθήκες αποθήκευσης (ΣΑ) κάθε θέσης. Την επιλογή της θέσης εντός της αποθήκης μαζί με οποιοδήποτε υπολογισμό και ενημέρωση της βάσης δεδομένων την αναλαμβάνουν ειδικά λογισμικά διαχείρισης αποθηκών (WMS - Warehouse Management Systems) τα οποία θα περιγράψουμε αργότερα. Η μετακίνηση υλικών, για λόγους τακτοποίησης, στους χώρους της αποθήκης συμβάλλει περίπου στο 15% του συνολικού κόστους λειτουργίας της αποθήκης (Frazelle 1996).

### 2.3.3 Διαλογή παραγγελιών

Με την άφιξη μιας παραγγελίας στο σύστημα, η αποθήκη κάνει άμεσα έλεγχο στα αποθέματά της ως προς τη διαθεσιμότητα ή όχι των προϊόντων για αποστολή. Στη συνέχεια, μέσω του WMS, εκτυπώνονται λίστες διαλογής (pick lists) για την καθοδήγηση των εργαζομένων στις κατάλληλες χωροθέσεις για τη διαλογή. Η διαλογή ανάλογα με το μέγεθος της συσκευασίας ανατίθεται σε διαφορετικές ομάδες εργαζομένων με διαφορετικά μέσα (Full Pallet Pick, Standard Pick, Micro Pick). Οι διαλογείς με τη σειρά τους, αναλαμβάνουν τη λίστα και μεταβαίνουν στις θέσεις που χρειάζεται για να εξαγουν το προϊόν από την κάθε χωροθέση και να το ετοιμάσουν για αποστολή. Για τη σωστή ενημέρωση του αποθέματος, οι διαλογείς φέρουν ειδικό scanner όπου δίνουν πληροφορίες στο WMS για τις τελικές τους κινήσεις μέσω των SSCC για την αποφυγή σφαλμάτων. Έχει παρατηρηθεί πως η διαδικασία της διαλογής αποτελεί περίπου το 55% του κόστους λειτουργίας μίας αποθήκης και η κατανομή του χρόνου της είναι η εξής (Frazelle 1996):

Διεργασία	% Ποσοστό χρόνου της διαλογής
Μετάβαση από και προς τη Χωροθέση	55%
Αναζήτηση SSCC	15%
Αφαίρεση από τη θέση	10%
Λοιπές διεργασίες (Χαρτιά κλπ.)	20%

### 2.3.4 Έλεγχος και συσκευασία

Με την περάτωση της διαλογής των προϊόντων της λίστας, η παραγγελία έχει πλέον πάρει μια φυσική μορφή και μεταφέρεται στο χώρο της αποστολής. Εκεί, μέσω του προσωπικού, επιβεβαιώνεται η σωστή διαλογή των προϊόντων προς αποστολή και η παραγγελία

συσκευάζεται συνήθως με μεμβράνη προκειμένου να σταθεροποιηθεί η κάθε σύμικτη πλέον παλέτα αρχικά για την ασφαλή φόρτωση της εντός του φορτηγού και στη συνέχεια για οποιαδήποτε μετακίνησή τους χώρους του πελάτη.

### 2.3.5 Αποστολή

Τέλος, η παραγγελία αποστέλλεται, φεύγει από τις εγκαταστάσεις της αποθήκης και μεταβαίνει στον αμέσως επόμενο κλάδο της εφοδιαστικής αλυσίδας, τη Διανομή και Δρομολόγηση των προϊόντων μέχρι να φτάσουν στον πελάτη.



## 2.4 Λογισμικό διαχείρισης εφοδιαστικής αλυσίδας

Με την αύξηση της πολυπλοκότητας των συστημάτων Logistics, δημιουργήθηκε η ανάγκη λογισμικών Η/Υ. Τέτοιου είδους λογισμικά ασχολούνται κυρίως με τη μηχανογράφηση κάθε είδους πληροφορίας που θα λειτουργήσει σαν εργαλείο για τη σωστή διοίκηση του συστήματος. Τα πιο βασικά από αυτά είναι τα εξής:

- **Logistics Management System (LMS)** για τη διαχείριση των παραγγελιών που εισέρχονται στο σύστημα, την επαφή με το πελατολόγιο και τους προμηθευτές.
- **Warehouse Management System (WMS)** για τη διαχείριση μίας ή περισσότερων αποθηκών όσον αφορά το απόθεμα, τις πληροφορίες κάθε προϊόντος (μέγεθος, βάρος, χωροθέση, συνθήκες αποθήκευσης κ.α) και την αξιολόγησή τους ως προς την αποδοτικότητά τους.
- **Transportation Management System (TMS)** για τη δημιουργία και διαχείριση του πλάνου διανομής, τη ζωντανή παρακολούθηση των φορτηγών και την ενημέρωση των πελατών από το διάστημα που φύγει από ένα κέντρο διανομής το προϊόν έως ότου φτάσει στα χέρια τους.

Άλλα πακέτα λογισμικού χρησιμοποιούνται για τη **διαχείριση προσωπικού (HR)**, τη **διαχείριση σε οικονομικό επίπεδο (Finance)** κλπ.



Σχήμα 2.2 Ολοκληρωμένο πακέτο διαχείρισης συστημάτων Logistics

Logistician – DOTLiNK IT/IS SERVICES

## 2.5 Συστήματα Διαχείρισης Αποθηκών

Τα συστήματα διαχείρισης αποθηκών (warehouse management systems, WMS) είναι σύνθετα πακέτα λογισμικών υποστήριξης που εστιάζουν στον έλεγχο των αποθεμάτων, τις χωροθέσεις για αποθήκευση και το εργατικό δυναμικό προκειμένου να διασφαλιστεί η έγκαιρη διαλογή των παραγγελιών, η συσκευασία της και η αποστολή της στον πελάτη. Τέτοια λογισμικά μπορούν να δώσουν πληροφορίες μεμονωμένα για κάθε προϊόν όπως τις διαστάσεις του, τις ιδιαιτερότητές του, τον τύπο συσκευασίας του, τις συνθήκες αποθήκευσής του κ.α. Ακόμα μπορούν να δώσουν πληροφορίες για κάθε θέση αποθήκευσης, όπως διαστάσεις, αντοχές και συνθήκες αποθήκευσης θέσεως. Με αυτόν τον τρόπο τα WMS δημιουργούν τις προϋποθέσεις για ομαλή και γρήγορη ροή των προϊόντων τόσο ως εισερχόμενα (με την επιλογή κατάλληλων θέσεων για αποθήκευση), όσο και ως εξερχόμενα (άμεση δημιουργία pick lists). Αξιοσημείωτο είναι να αναφέρουμε πως υπάρχουν λογισμικά WMS που συνεργάζονται με πακέτα Voice Picking, όπου ο διαλογέας ενημερώνεται ηχητικά και αλληλεπιδρά με το πρόγραμμα με σκοπό την διευκόλυνση του κατά τη διάρκεια της διαδικασίας και την εξοικονόμηση πόρων και χρόνου.

Μερικές από τις τυπικές επιλογές ενός τέτοιου συστήματος είναι οι εξής (Bartholdi III and Hackman 2016, βλέπε και Σχ. 2.3):

- Παραλαβή
- Έλεγχος ποιότητας
- Τακτοποίηση προϊόντος
- Εύρεση πληροφοριών θέσης
- Διαχείριση σειράς εργασιών
- Διαλογή προϊόντων

- Συσκευασία και οριστικοποίηση παραγγελίας
- Αποστολή
- Προγραμματισμός έλευσης φορτηγών



Σχήμα 2.3 Παράδειγμα διεργασιών μεταξύ των οποίων εμπλέκονται λογισμικά WMS

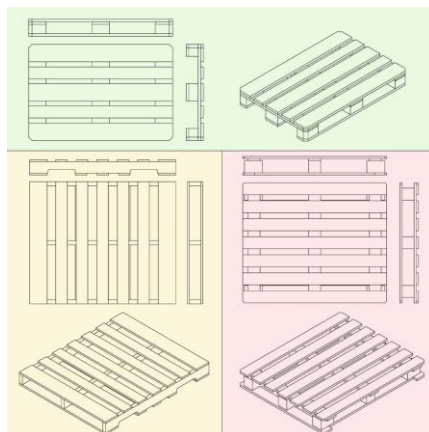
## 2.6 Αποθήκευση και εξοπλισμός διαχείρισης

### 2.6.1 Αποθήκευση

Το υψηλό κόστος των προϊόντων και η χρήση τους σε υψηλούς ρυθμούς δημιούργησαν την ανάγκη για τυποποιήσεις και αξιόπιστα συστήματα αποθήκευσης και μεταχείρισης των προϊόντων. Σκοπός ήταν η όλο και πιο γρήγορη και συγχρόνως ασφαλής μετακίνησή τους μέχρι τον τελικό αποδέκτη (πελάτη). Τέτοια συστήματα είναι για παράδειγμα ειδικά διαμορφωμένες ραφιέρες με μεταβαλλόμενα ύψη όπου ανάλογα τις ανάγκες των προϊόντων ως προς τις διαστάσεις τους μπορούμε να αυξομειώσουμε το διάκενο μεταξύ των ραφιών και να εξασφαλίσουμε τον απαραίτητο χώρο. Αυτό ως αποτέλεσμα την μέγιστη δυνατή εκμετάλλευση του όγκου του κτιρίου σε θέσεις. Σημαντικό είναι επίσης να αναφέρουμε πως για τη σωστή και ασφαλή διαχείριση των προϊόντων από περνοφόρα οχήματα εσωτερικού χώρου, υπάρχουν τυποποιήσεις όσον αφορά τις διαστάσεις και τις αντοχές των παλετών, πράγμα που διευκολύνει καθ' όλη τη διάρκεια της μεταφοράς και της διακίνησης του προϊόντος εντός της αποθήκης, όπως για παράδειγμα η χωροθέτησή τους εντός των φορτηγών κατά την φόρτωσή του δίνοντας άμεσα την πληροφορία για το μέγιστο πλήθος παλετών που μπορεί να «φιλοξενήσει» ένα φορτηγό.

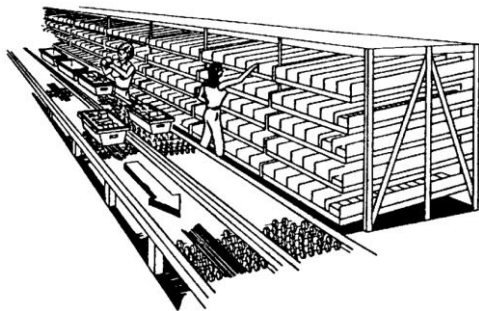


Σχήμα 2.4 Παράδειγμα  
συστήματος ραφιών  
αποθήκευσης



Σχήμα 2.5 Διαφορετικές  
διαστάσεις παλετών.

Άλλη μορφή βέλτιστης διαχείρισης των προϊόντων, αλλά σε αυτήν την περίπτωση σε κιβωτιακό ή και τεμαχιακό επίπεδο, είναι η χρήση κεκλιμένων ραφιών. Τα ράφια αυτά επιτρέπουν στον διαλογέα την εύκολη εξαγωγή του προϊόντος από αυτά ενώ με τη βοήθεια της βαρύτητας κυλά στη θέση του το αμέσως από πίσω, γεγονός που διευκολύνει σε μεγάλο βαθμό την εξαγωγή του επόμενου κιβωτίου ή μικρότερης συσκευασίας αναφοράς όταν αυτό απαιτηθεί. Επίσης δεν πρέπει να ξεχνάμε το μεγάλο αριθμό ωρών που απαιτεί η διαλογή τεμαχίων, όπως για παράδειγμα ένα (1) στύλο, ένας (1) σκληρός δίσκος Η/Υ κ.α. Το μικρό τους μέγεθος και πολλές φορές το υψηλό τους κόστος σε συνδυασμό με την υψηλή ζήτησή τους κατά μονάδες, απαιτούν αυξημένη παραγωγικότητα και ασφάλεια για την αποφυγή της φθοράς τους. Η φθορά θα δημιουργήσει ανάγκες για περισσότερες διαδικασίες όπως αντικατάσταση στον πελάτη και επιπλέον κινήσεις εντός αποθήκης όπως για παράδειγμα δημιουργία εγγράφων καταστροφής, παύση λειτουργίας έως ότου ο χώρος επανέλθει στην αρχική του κατάσταση, αν αυτό απαιτείται, και άλλα. Για την ορθή λοιπόν διαχείριση τεμαχίων, υπάρχουν λύσεις ολοκληρωμένα συστήματα σταθμών διαλογής που επιτρέπουν στον εργαζόμενο να πραγματοποιεί μικρές κινήσεις προκειμένου να ολοκληρωθεί μια παραγγελία σε χώρους ειδικά διαμορφωμένους για αυτό το είδος διαλογής. Τα συστήματα αυτά ποικίλουν ανάλογα το κόστος, την ταχύτητα και τις διευκολύνσεις που προσφέρουν στον εργαζόμενο. Μερικά από αυτά τα συστήματα αποτελούνται από θυρίδες σε μικρές μεταξύ τους αποστάσεις που επιτρέπουν την εξαγωγή του προϊόντος σε ελάχιστα δευτερόλεπτα χωρίς να χρειάζεται να γίνουν ιδιαίτερες μετακινήσεις από προϊόν σε προϊόν. Οι θυρίδες αυτές γεμίζουν με τη διαδικασία της ανατροφοδότησης από το περιεχόμενο των κιβωτίων. Επικουρικά μπορεί να λειτουργήσει και ένα σύστημα ενός ή περισσότερων κινούμενων διαδρόμων, όπου σε συνεργασία με λογισμικό εσωτερικής δρομολόγησης, ο διαλογέας παραμένει σταθερός και εναποθέτει σε διερχόμενα κυτία μόνο προϊόντα τα οποία βρίσκονται σε ένα εύρος γύρω από τη θέση του. Με αυτόν τον τρόπο αποφεύγονται οι άσκοπες κινήσεις ενός διαλογέα από κατά τον κύκλο της εξαγωγής προϊόντων. Πάνω στον ίδιο άξονα, κινούνται ρομποτικές τεχνολογίες όπου διατηρούν τον διαλογέα σταθερό και τον προσεγγίζουν ρομποτικά οχήματα όπου φέρουν κατάλληλη «βιβλιοθήκη» με προϊόντα.



Σχήμα 2.6 Κεκλιμένα ράφια για τεμαχιακή διαλογή με χρήση διαδρόμων (Bartholdi III and Hackman 2016)



Σχήμα 2.7 Ρομποτικά οχήματα από τις αποθήκες της Amazon

Τέλος, είναι σημαντικό να σχολιάσουμε πως υπάρχει ανάγκη για ύπαρξη ισορροπίας μεταξύ σταθμών διαλογής και ραφιών αποθήκευσης παλετών διότι οι πρώτοι απαιτούν κατάληψη μεγάλων χώρων εντός της αποθήκης και όπου συχνά δεν επιτρέπεται η διέλευση περνοφόρων οχημάτων ή η εγκατάσταση ραφιών. Έτσι η επιλογή οφείλει να γίνεται με τέτοιο τρόπο ώστε να συυπολογίζονται οι ανάγκες της αγοράς και της αποθήκης.

### 2.6.2 Εξοπλισμός διαχείρισης

Ο εξοπλισμός διαχείρισης αποτελείται κυρίως από scanners και περνοφόρα οχήματα. Για τα scanners έχουμε αναφερθεί σε προηγούμενη ενότητα (WMS), ενώ για τα περνοφόρα πρέπει να γνωρίζουμε ότι κατατάσσονται σε τρεις βασικές κατηγορίες:

- Χειροκίνητοι τροχήλατοι γρύλοι
- Ηλεκτροκίνητοι τροχήλατοι γρύλοι
- Περνοφόρα ανυψωτικά

Οι τροχήλατοι γρύλοι είναι τύποι μηχανημάτων για τη μεταφορά φορτίων πάνω σε παλέτα. Κινούνται αποκλειστικά σε επίπεδο πάτωμα και διαχωρίζονται σε χειροκίνητους και μηχανικούς. Το φορτίο κάθεται πάνω στην περόνη ή οποία είναι σχεδιασμένη να σύρεται κάτω από την παλέτα μέσω των διακένων της με τη βοήθεια μηχανικού ή ηλεκτρικού γρύλου που βρίσκεται στον ίδιο άξονα με τις ρόδες και το τιμόνι που κατευθύνει τον παλετοφόρο. Το τιμόνι χρησιμεύει ακόμα και για το χειρισμό του γρύλου (ανύψωση-καθίζηση της περόνης) και έχει φρένο. Πλέον εντός αποθηκών, οι χειροκίνητοι παλετοφόροι χρησιμοποιούνται κυρίως από τους διαλογείς τεμαχίων (micro picking) και οι ηλεκτροκίνητοι από αυτούς των κιβωτίων (standard picking),

την φόρτωση και την εκφόρτωση φορτηγών όπου απαιτούνται μεγαλύτερες και συντομότερες μετακινήσεις.



Σχήμα 2.8 Χειροκίνητος Τροχήλατος Γρύλος –  
Toyota Material Handling Greece

Σχήμα 2.9 Ηλεκτροκίνητος Τροχήλατος  
Γρύλος – Toyota Material Handling Greece



Ακόμα, ανυψωτικά περionoφόρα οχήματα (reach trucks) ονομάζονται τα μεγαλύτερα, επίσης ηλεκτροκίνητα μηχανήματα που έχουν τη δυνατότητα να ανεβάσουν με την περόνη τους φορτία σε μεγάλο ύψος (έως και 13m). Χρησιμοποιούνται κυρίως για την τακτοποίηση παλετών εντός αποθήκης σε ράφια και την εξαγωγή τους από αυτά όταν αυτό ζητηθεί και η χρήση τους είναι αναγκαία σε κάθε αποθήκη από ειδικά εκπαιδευμένο προσωπικό.

Είναι σημαντικό να αναφέρουμε πως, ενώ υπάρχουν περionoφόρα οχήματα εσωτερικής καύσης, η χρήση τους δεν επιτρέπεται σε κλειστούς χώρους, άρα και σε αποθήκες, για την ασφάλεια των εργαζομένων από τα καυσαέρια που εκλύουν.



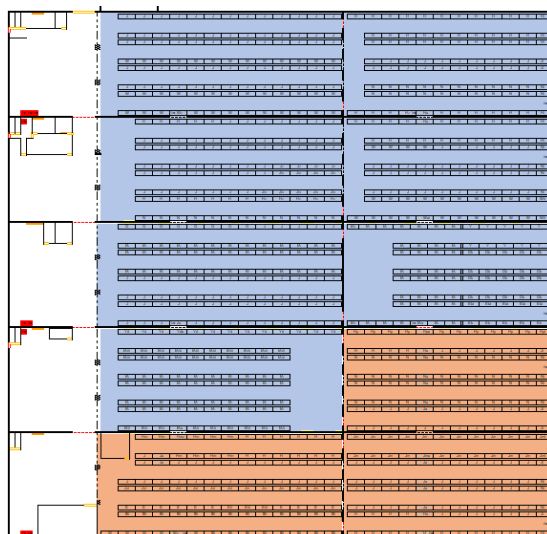


*Σχήμα 2.10 Reach Truck – Toyota Material Handling Greece*

### 3 Βασικές Λειτουργίες και Συνιστώσες του Μελετώμενου Κέντρου Διανομής

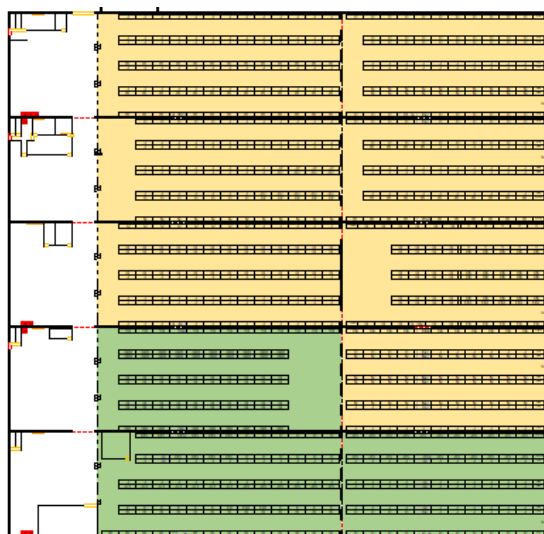
#### 3.1 Χώρος

Το υπό εξέταση κέντρο διανομής δραστηριοποιείται στον κλάδο των 3PL (εφοδιασμού προς τρίτους), όπου ο διαχειριστής του κέντρου αναλαμβάνει την αποθήκευση, τη διαλογή και τέλος τη διανομή των προϊόντων για λογαριασμό τρίτων (Αποθέτες) προς το πελατολόγιό τους. Αυτό σημαίνει πως την αποθήκη τη «μοιράζονται» συνήθως παραπάνω από ένας αποθέτες, στη συγκεκριμένη περίπτωση έξι (6). Ο στεγασμένος χώρος είναι περίπου 10000 m<sup>2</sup> και αποτελείται από δέκα (10) δωμάτια, εκ των οποίων επτά βρίσκονται σε κατάσταση κλιματισμού και 3 από αυτά όχι, για την εξυπηρέτηση ξηρών φορτίων και οι χώροι εκφόρτωσης βρίσκονται αριστερά με άσπρο χρώμα όπως βλέπουμε στο σκαρίφημα του Σχ. 3.1.



Σχήμα 3.1  
Άποψη του  
κέντρου  
διανομής ως  
προς τα  
κλιματιζόμενα  
και μη  
δωμάτια. Με  
γαλάζιο  
(επάνω), οι  
κλιματιζόμενοι  
χώροι.

Ακόμα, στο συγκεκριμένο κέντρο διανομής αντιμετωπίζουμε την ιδιαιτερότητα πως από τους έξι αποθέτες, ο «Αποθέτης 3» έχει κάνει συμφωνία με τη διοίκηση του συστήματος ώστε να δεσμευθούν συγκεκριμένοι χώροι εντός της αποθήκης προκειμένου να μην έρχονται τα προϊόντα του σε επαφή με αυτά των άλλων αποθετών για δικούς του λόγους. Αυτό έχει ως αποτέλεσμα, ο «Αποθέτης 3» να έχει δύο χώρους ξηρού φορτίου και έναν με κλιματισμό και οι υπόλοιποι τέσσερις αποθέτες να μοιράζονται έξι (6) χώρους κλιματισμού και έναν (1) ξηρού φορτίου. Παρόλα αυτά, το προσωπικό δεν διαχωρίζεται για την περάτωση των παραγγελιών κάθε αποθέτη.



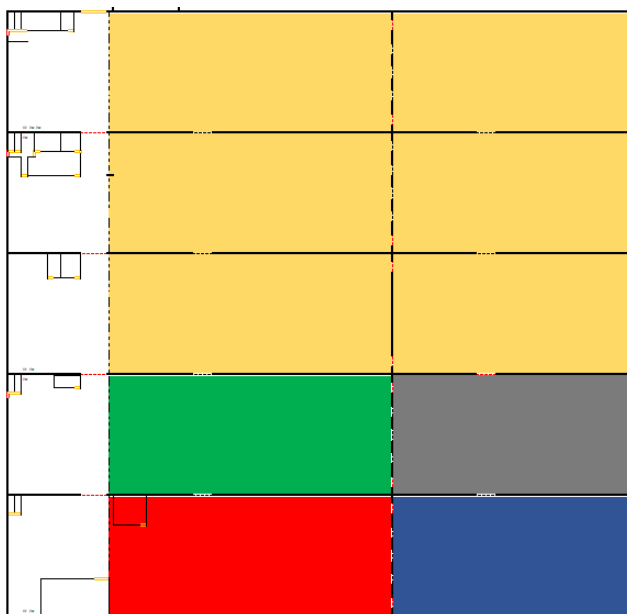
Σχήμα 3.2  
Άποψη του  
κέντρου ως  
προς τους  
αποθέτες ανά  
χώρο. Με  
πράσινο  
χρώμα (κάτω),  
χώροι  
δεσμευμένοι  
για τον  
«Αποθέτη 3».

Τέλος, το δίκτυο μεταφοράς εντός της αποθήκης, ορίζεται από τα φυσικά εμπόδια (επιπρόσθετοι χώροι κλπ.), τις γέφυρες που μας επιτρέπουν μετακίνηση από διάδρομο σε διάδρομο και τις διαθέσιμες πόρτες για τη μετακίνηση σε άλλους χώρους της αποθήκης.



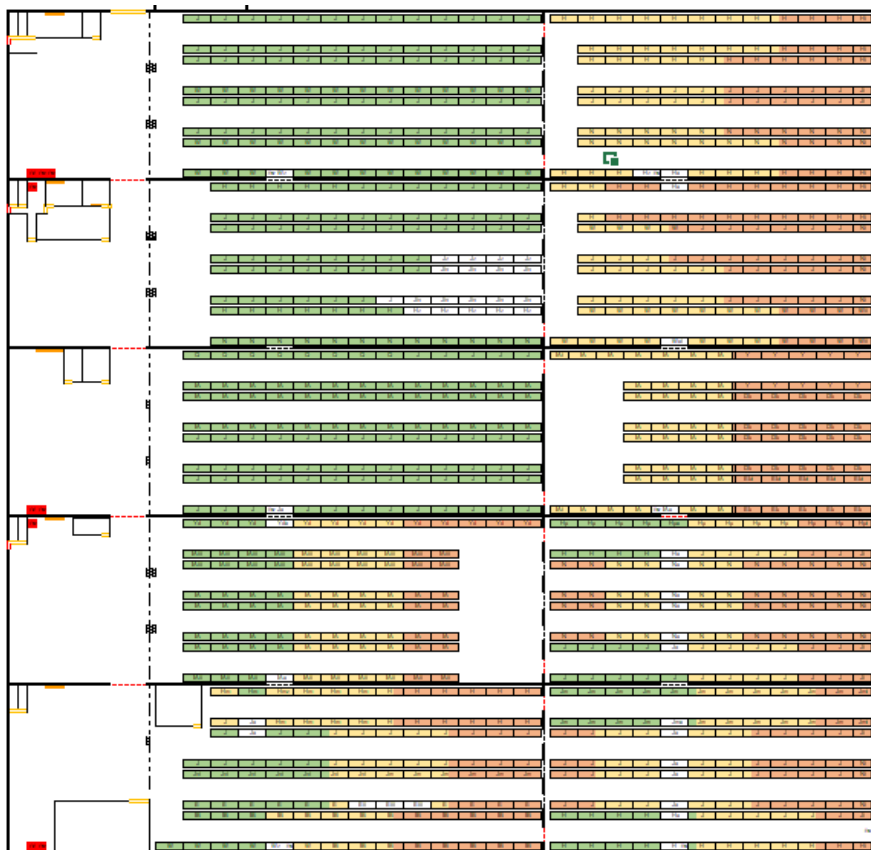
Σχήμα 3.3 Άποψη της αποθήκης ως προς τις διαθέσιμες διόδους διέλευσης

Οι πληροφορίες των τριών παραπάνω σχεδίων, μας βοήθησαν να δημιουργήσουμε ξεχωριστές ζώνες ανάλογα με την κινητικότητα των προϊόντων εντός αποθήκης. Για τη δημιουργία αυτών των «ομάδων θέσεων», πρέπει πρώτα να ξεχωρίζουμε ποια δωμάτια λειτουργούν αυτόνομα ανάλογα με τον αποθέτη και τις συνθήκες αποθήκευσής τους. Ο διαχωρισμός αυτός είναι ο παρακάτω και όπου ο μπλε και ο κόκκινος χώρος δεν ανήκουν στην ίδια ομάδα για λόγους συνθηκών αποθήκευσης προϊόντων που θα μελετήσουμε αργότερα.



Σχήμα 3.4 Ομάδες δωματίων της αποθήκης.

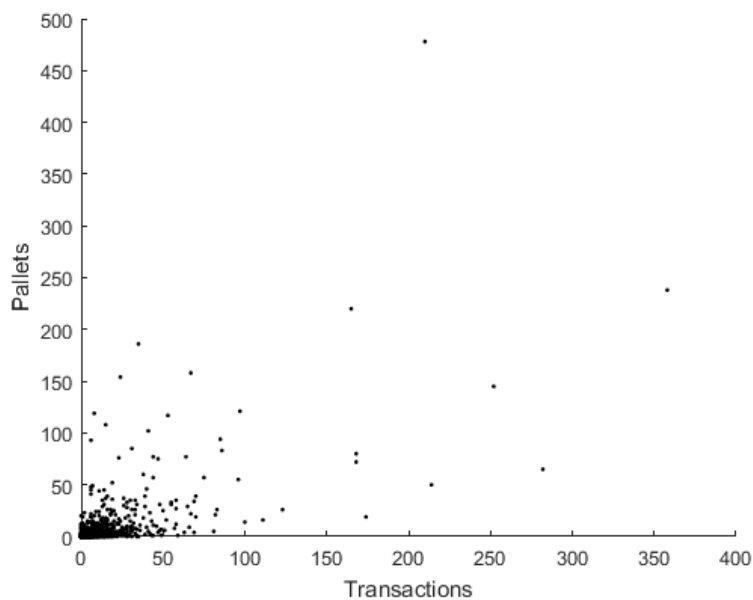
Στη συνέχεια, μέσω αλγορίθμου στη Matlab και με τη βοήθεια γραφημάτων με κόμβους και τόξα, υπολογίσαμε τις αποστάσεις όλων των χωροθέσεων της αποθήκης προς τη νοητή γραμμή που ενώνει όλες τις ράμπες για παραλαβή και αποστολή με χρήση της συνάρτησης `shortestpath(G,s,t)` η οποία βρίσκει τη βέλτιστη διαδρομή σε ένα γράφημα  $G$  από έναν κόμβο  $s$  σε κόμβο  $t$ . Από αυτή τη λίστα θέσεων εξαιρέσαμε όλες τις θέσεις `picking` όπου δεν μπορούν να φιλοξενήσουν παλέτες για απόθεση. Γνωρίζοντας πλέον την απόσταση σε δύο άξονες, συνεχίζουμε για την εύρεση του χρόνου προσέγγισης κάθε θέσης. Για να πραγματοποιηθεί αυτό, χρησιμοποιούμε τις προδιαγραφές ενός reach truck που υπάρχει στην αγορά όσον αφορά τις ταχύτητές του σε οριζόντιο επίπεδο, σε ανύψωση-καθίζηση και προσέγγιση παλέτας. Τα αποτελέσματα δημιουργούν το παρακάτω σχήμα όπου με πράσινο χρώμα είναι οι θέσεις με την ευκολότερη πρόσβαση, με κίτρινο αυτές με την αμέσως πιο δύσκολη πρόσβαση και με κόκκινο αυτές με τη δυσκολότερη. Παρουσιάζουμε μόνο το δεύτερο επίπεδο ραφιών. Τα επόμενα επίπεδα, είναι σχεδόν όμοια και συγκλίνουν προς το επίπεδο του τοίχου των θυρών.



Σχήμα 3.5 Άποψη αποθήκης ως προς τις ζώνες που δημιουργούνται ανά ομάδα δωματίων

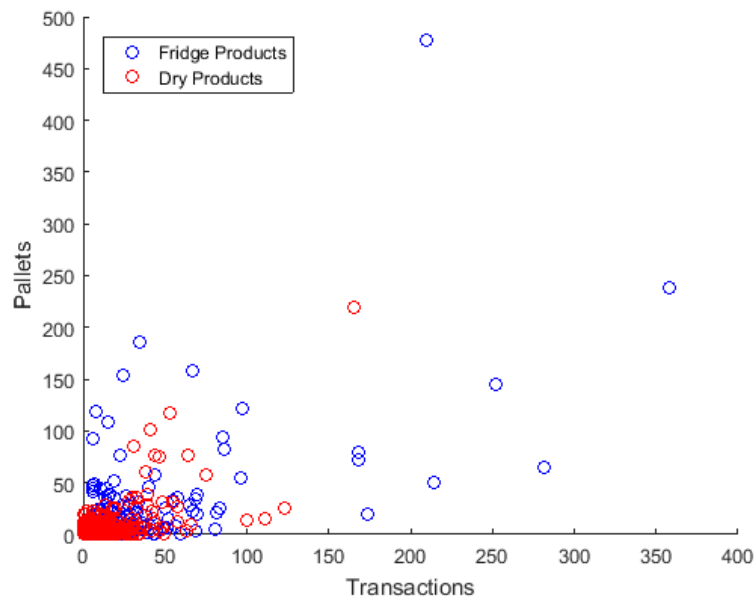
## 3.2 Προϊόντα

Αφού δημιουργήσαμε ομάδες χωροθέσεων εντός αποθήκης, πρέπει να δημιουργήσουμε και ομάδες προϊόντων προκειμένου να έχουμε μία άποψη της κινητικότητάς τους στο σύστημα που μελετάμε. Προϋπόθεση για να γίνει αυτό είναι να γνωρίζουμε για κάθε ένα προϊόν, πόσες φορές έχει ζητηθεί να πραγματοποιηθεί κίνηση για αυτό (παραλαβή – ενδοδιακίνηση/διαλογή παλέτας) και πόσες χωροθέσεις καταλαμβάνει αυτό στο ήδη υπάρχον απόθεμα (παλετοθέσεις). Από αυτό προκύπτει ένα διάγραμμα όπου στον οριζόντιο άξονα θα έχουμε το πλήθος των κινήσεων των προϊόντων και στον κάθετο, το πλήθος των κατειλημμένων θέσεων από αυτά.



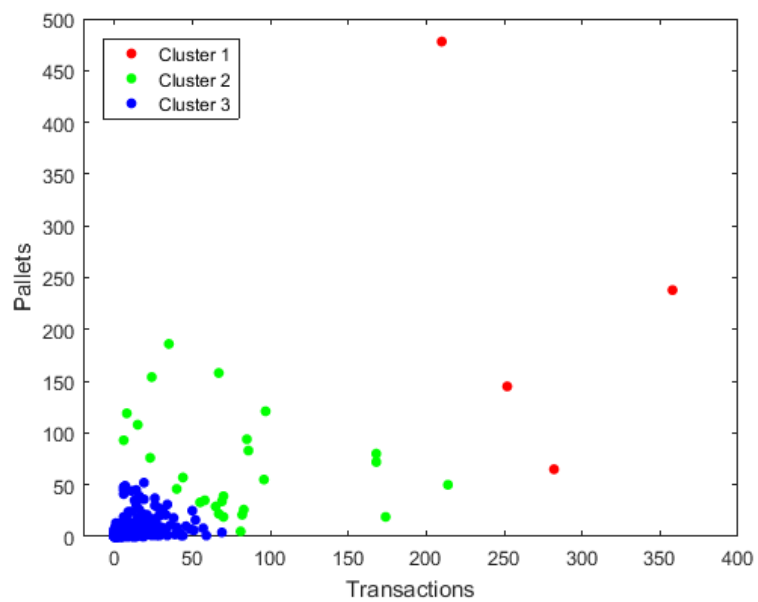
Σχήμα 3.6 Κατανομή Προϊόντων ως προς τις κινήσεις τους (transactions) και τις χωροθέσεις που καταλαμβάνουν εντός αποθήκης (pallets)

Προκειμένου να κατηγοριοποιήσουμε τα προϊόντα, θα χρησιμοποιήσουμε τον αλγόριθμο συσταδοποίησης (clustering) *k-means*. Ο εν λόγω αλγόριθμος έχει ως στόχο να διαχωρίσει η παρατηρήσεις ενός χώρου  $m$  διαστάσεων σε  $k$  ομάδες, έτσι ώστε κάθε παρατήρηση να ανήκει στη συστάδα (cluster) με το κοντινότερο μέσο, το οποίο χρησιμεύει ως ένα χαρακτηριστικό δείγμα της συστάδας. Παρόλα αυτά πρέπει να γίνει επιπλέον διαχωρισμός των προϊόντων σε κλιματιζόμενα και μη (ξηρό φορτίο), διότι απευθύνονται σε χώρους με διαφορετικό μέγεθος και συνθήκες αποθήκευσης και τα δεύτερα δείχνουν πως έχουν και μικρότερη κινητικότητα στο διάγραμμα.

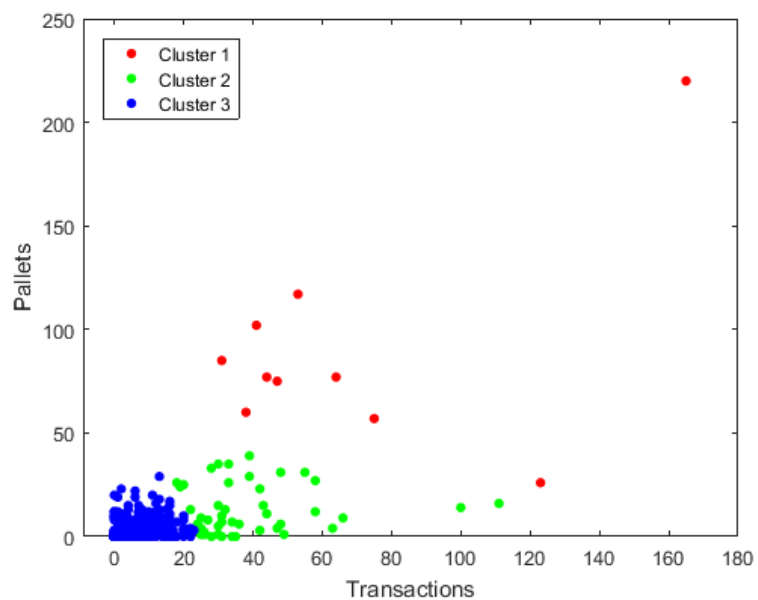


Σχήμα 3.7 Διάγραμμα Προϊόντων ως προς τους χώρους αποθήκευσης

Στη συνέχεια θα τρέξουμε δύο φορές τον αλγόριθμο k-means, μια για κάθε ευρύτερη ομάδα προϊόντων (ψυχρό και ξηρό φορτίο) προκειμένου να δημιουργήσουμε τρεις ομάδες με προϊόντα που κινούνται με μεγάλη, μέτρια και μικρή ταχύτητα εντός της αποθήκης ανάλογα με τον τρόπο αποθήκευσής τους όπως βλέπουμε παρακάτω. Σκοπός είναι να δημιουργήσουμε τις προϋποθέσεις έτσι ώστε τα ήδη αποθηκευμένα προϊόντα να βρίσκονται σε κατάλληλες χωροθέσεις με βάση τις συστάδες θέσεων-προϊόντων και οι κινήσεις που γίνονται με τα Reach Trucks να βασίζονται σε αυτή τη λογική.

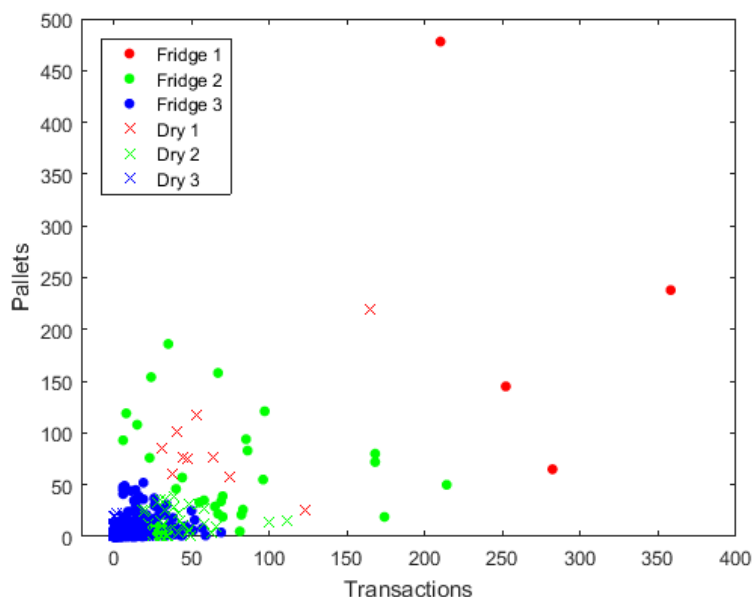


Σχήμα 3.8 Διάγραμμα συστάδων προϊόντων ψυγείου



Σχήμα 3.9 Διάγραμμα συστάδων προϊόντων εκτός ψυγείου





Σχήμα 3.10 Διάγραμμα συστάδων προϊόντων για ψυγείο (Fridge) και ξηρή (Dry) αποθήκευση

### 3.3 Συνθήκες αποθήκευσης προϊόντων

Τα προϊόντα που διαχειρίζεται ή αποθήκη είναι κατά κύριο λόγο τρόφιμα όπου κατά τη διαλογή επιλέγονται κατά σειρά παλαιότητας – FIFO (First In First Out) και όπως προαναφέραμε, ο αρχικός διαχωρισμός έχει γίνει βάσει ψυχρού και ξηρού φορτίου. Παρακάτω θα δούμε τις υποκατηγορίες που τα χαρακτηρίζουν προκειμένου να γίνει κατάλληλη συντήρηση και τήρηση προδιαγραφών από τον αποθέτη.

Κατηγορίες Συνθ. Απ	Ψυχρό Φορτίο	Ξηρό Φορτίο
Υποκατηγορίες Συνθ. Απ	02 - Κλιματισμός	01 - Κανονικές Συνθήκες
		05 - Καν.Συνθ. (Γυάλινη Συσκευασία)
		06 - Καν.Συνθ. (Ιδιαίτερης Οσμής)
		07 - Καν.Συνθ. (Ελεύθερο Οσμών)
		08 - Καν. Συνθ. (Τσίχλα)
		99 - Γενικές Συνθήκες

Σχήμα 3.11 Υποκατηγορίες Συνθηκών Αποθήκευσης Προϊόντων

Για τα προϊόντα με τα χαρακτηριστικά «01 - Κανονικές Συνθήκες», «02 - Κλιματισμός» και «99 - Γενικές Συνθήκες» είναι εύκολο να φανταστούμε τα είδη τους και τον τρόπο μεταχείρισης που χρήζουν. Αντίθετα, τα προϊόντα με γυάλινη συσκευασία πρέπει να αποθηκεύονται με τρόπο τέτοιο ώστε στα από κάτω τους ράφια να βρίσκονται όμοιά τους. Ο λόγος είναι για την αποφυγή εκτεταμένης καταστροφής προϊόντων με συσκευασία χαμηλότερης αντοχής σε περίπτωση

καταστροφής της πρώτης κατά τη διάρκεια κάποιας διαδικασίας. Επίσης, οι περιορισμοί οσμών και τσίχλας αφορούν την τη γειτνίαση με προϊόντα που μπορούν να αλλοιώσουν τα χαρακτηριστικά άλλων. Όπως για παράδειγμα η τσίχλα που είναι ένα προϊόν όπου η ποιότητά του σχετίζεται άμεσα με τη γεύση της και την οσμή της και η συσκευασία της είναι ευτελής (περιτύλιγμα με πολύ λεπτά υλικά). Αυτά την καθιστούν άμεσα ένα ευαίσθητο προϊόν που αδυνατεί να αποθηκευτεί μαζί με προϊόντα με ιδιαίτερη οσμή όπως για παράδειγμα οι σκυλοτροφές. Επομένως, για κάποια προϊόντα μπορούν να γίνουν κάποιες παραχωρήσεις και να αποθηκευθούν σε θέσεις με διαφορετικές συνθήκες αποθήκευσης και για κάποια όχι.

01 Κανονικές Συνθήκες	-	02 Κλιματισμός	-	05 Καν.Συνθ. (Γυάλινη Συσκευασία)	-	06 Καν.Συνθ. (Ιδιαίτερης Οσμής)	-	07 Καν.Συνθ. (Ελεύθερο Οσμών)	-	08 Καν.Συνθ. (Τσίχλα)	-	01 Κανονικές Συνθήκες
02 Κλιματισμός	-	-	-	01 Κανονικές Συνθήκες	-	-	-	06 Καν.Συνθ. (Ιδιαίτερης Οσμής)	-	-	-	02 Κλιματισμός
07 Καν.Συνθ. (Ελεύθερο Οσμών)	-	-	-	02 Κλιματισμός	-	-	-	01 Κανονικές Συνθήκες	-	-	-	05 Καν.Συνθ. (Γυάλινη Συσκευασία)
												06 Καν.Συνθ. (Ιδιαίτερης Οσμής)
												07 Καν.Συνθ. (Ελεύθερο Οσμών)
												08 Καν.Συνθ. (Τσίχλα)
												99 Γενικές Συνθήκες

Σχήμα 3.12 Περιορισμοί και υποχωρήσεις λόγω συνθηκών αποθήκευσης

### 3.4 Εξυπηρετητές

Η συγκεκριμένη αποθήκη, λόγω της φύσης των προϊόντων της λειτουργεί περισσότερο με τεμαχιακή και κιβωτιακή διαλογή. Παρόλα αυτά, λόγω του μεγάλου όγκου και βάρους των προϊόντων που παραλαμβάνονται και κινούνται καθημερινά, η ανάγκη για χρήση ανυψωτικών μηχανημάτων κρίνεται επιτακτική. Έτσι, η επιχείρηση, προκειμένου να περατώνονται οι καθημερινές παραγγελίες, έχει προβεί στον εξής σχεδιασμό.

- Οι εργασίες να κατανέμονται σε δύο (2) βάρδιες
- Οι διαλογείς του micro picking να είναι οκτώ (8) και του κιβωτιακού δεκατέσσερις (14) με τρεις (3) από κάθε ομάδα να μπορούν να κάνουν και τις δύο εργασίες σε κάθε βάρδια
- Οι χειριστές των reach trucks να είναι πέντε (5) σε κάθε βάρδια

Επειδή στην παρούσα εργασία εστιάζουμε στην ανάπτυξη ενός αλγορίθμου προσομοίωσης και τη βελτιστοποίηση των μεγάλων κινήσεων εντός της αποθήκης, θα χρειαστούμε στοιχεία για Reach Trucks, όπως ταχύτητα κίνησης/ανύψωσης κλπ. Πρέπει λοιπόν να κάνουμε μια επιλογή, τέτοια ώστε να καλύπτονται οι ανάγκες της αποθήκης σε βάρη και μέγιστο ύψος ανύψωσης και στην προκειμένη περίπτωση θα επιλέξουμε πέντε (5) Toyota Reach Trucks BT Reflex των 1.6 τόνων.



Ταχύτητα κίνησης με/χωρίς φορτίο	km/h	10.0/10.0
Ταχύτητα ανύψωσης με/χωρίς φορτίο	m/s	0.28/0.43
Ταχύτητα καθίζησης με/χωρίς φορτίο	m/s	0.52/0.52
Ταχύτητα προσέγγισης με/χωρίς φορτίο	m/s	0.20/0.20

Σχήμα 3.14 Εικόνες και τεχνικά χαρακτηριστικά BT Reflex 1.6t -  
Toyota Material Handling Greece

## 4 Προσομοίωση του Συστήματος

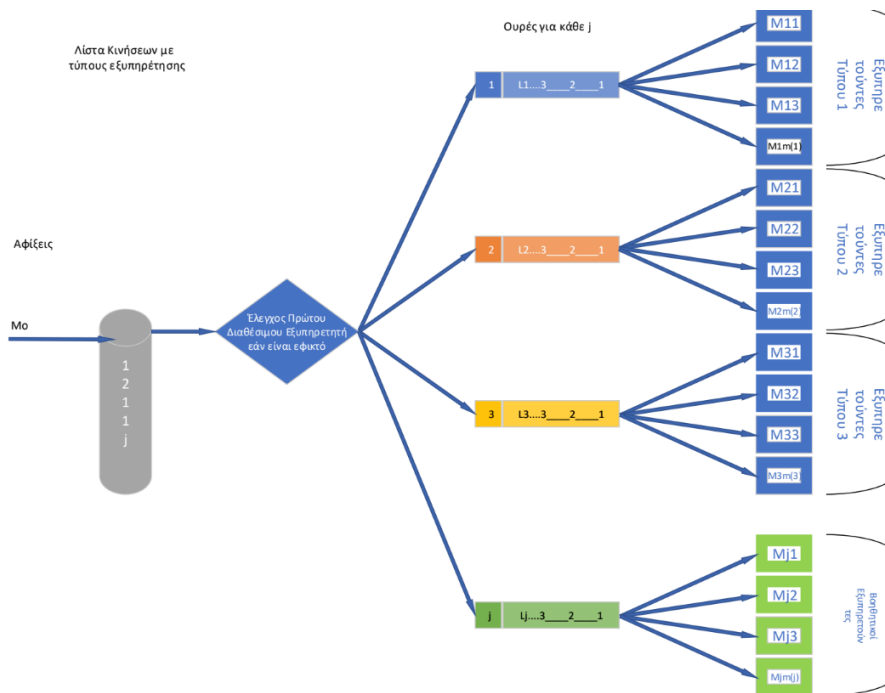
### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό θα αναφερθούμε στον αλγόριθμο προσομοίωσης διακριτών γεγονότων που αναπτύξαμε και στις λεπτομέρειες παραμετροποίησής του στα δεδομένα του συστήματος που μελετάμε.

Τα μοντέλα διακριτών γεγονότων επιτρέπουν την περιγραφή πολύπλοκων συστημάτων με πολλές παραμέτρους και δίνουν τη δυνατότητα για άμεσο πειραματισμό πάνω σε αυτά. Είναι πολύ σημαντικό για μία επιχείρηση να μπορεί να εξετάζει διάφορα σενάρια και πιθανούς τρόπους λειτουργίας των συστημάτων της χωρίς να υπάρχει κίνδυνος να μείνει πίσω στην παραγωγή ή να προβεί σε άσκοπη χρήση για δοκιμές που δεν είναι πάντα επιτυχείς και μπορεί να επιβαρύνουν με υψηλό κόστος.

### 4.2 Περιγραφή

Εξετάζουμε ένα σύστημα αναμονής με αφίξεις και εξυπηρέτηση με ένα μηχανισμό  $M_0$  ο οποίος "γεννά" οντότητες που εισέρχονται στο σύστημα διαδοχικά και παίρνουν το χαρακτηριστικό του τύπου εργασίας από την πρώτη γραμμή της λίστας εργασιών κάθε φορά. Στη συνέχεια, εξετάζεται εάν υπάρχει διαθέσιμος εξυπηρετητής (με μπλε χρώμα στο σχεδιάγραμμα) για την συγκεκριμένη εργασία. Εάν πράγματι υπάρχει, η άφιξη μεταφέρεται σε αυτόν και τον δεσμεύει μέχρις ότου περατωθεί ο χρόνος εξυπηρέτησης. Εάν πάλι δεν υπάρχει διαθέσιμος εξυπηρετητής, εξετάζεται η περίπτωση του βοηθητικού (με πράσινο χρώμα στο σχεδιάγραμμα) με τον ίδιο τρόπο και τέλος εάν είναι όλοι απασχολημένοι, άφιξη παραμένει στην ουρά του ικανού εξυπηρετητή που θα είναι νωρίτερα διαθέσιμος. Κάθε ουρά ανά κατηγορία εργασίας μπορεί να έχει διαφορετική χωρητικότητα και κάθε κατηγορία εξυπηρετητών, διαφορετικό πλήθος. Στη συγκεκριμένη περίπτωση, οι ουρές αναμονής δεν έχουν όριο και κάθε εργασία μπορεί να εκτελεστεί και από κάποιον βοηθητικό εξυπηρετητή.



Σχήμα 4.1 Σύστημα αναμονής με  $m$  τύπους εξυπηρέτησης, ουράς και διαφορετικά πλήθη εξυπηρετούντων ανά τύπο

Υποθέτουμε ότι οι προτεραιότητες σε κάθε τύπο εξυπηρέτησης καθορίζονται από το πρωτόκολλο First Come First Served (FCFS): όταν ένας εξυπηρετών είναι διαθέσιμος για νέα εξυπηρέτηση, αναλαμβάνει τον πελάτη που είχε φτάσει νωρίτερα στο σύστημα και έχει τον μεγαλύτερο χρόνο αναμονής στην ουρά.

Όταν η ουρά αναμονής αδειάσει, οι εξυπηρετούντες που τελειώνουν τους προηγούμενους πελάτες τους μένουν αδρανείς.

Ο αλγόριθμος ακολουθεί τα εξής βήματα:

**Βήμα 1:** Μόλις γίνει η άφιξη στο σύστημα, καθορίζεται ο τύπος εξυπηρετητή και εφόσον είναι διαθέσιμος κάποιος από αυτούς, υπολογίζεται ο χρόνος εξυπηρέτησης και στη συνέχεια αποθηκεύεται σε έναν πίνακα  $T_{ij}$  το επόμενο γεγονός όπου  $i$  είναι ο εξυπηρετών και  $j$  ο τύπος εξυπηρέτησης. Αν δεν υπάρχει διαθέσιμος εξυπηρετών και υπάρχει η επιλογή του βοηθητικού τύπου εξυπηρέτησης επιλέγεται αυτός.

**Βήμα 2:** Ο αλγόριθμος βρίσκει την μικρότερη τιμή του πίνακα  $T$  (επόμενο γεγονός), αποθηκεύει τη θέση του  $i, j$  (αριθμό εξυπηρετούντος και τύπο αντίστοιχα), την τιμή του και κάνει ένα χρονικό άλμα στην τιμή αυτή καθώς ενημερώνει τις μεταβλητές του όσον αφορά τους δείκτες απόδοσης και τους μετρητές του.

**Βήμα 3:** Εάν δεν υπάρχει κανένας διαθέσιμος, η άφιξη μετατοπίζεται στην ουρά αυτού (βασικού ή βοηθητικού τύπου) που θα είναι πιο γρήγορα διαθέσιμος και αποθηκεύεται μαζί της

η γραμμή που βρίσκεται στη λίστα κινήσεων για να μπορεί να ανακαλεστεί και να υπολογιστεί εκ νέου ο χρόνος εξυπηρέτησης.

Ο αλγόριθμος επαναλαμβάνεται από το Βήμα 1 έως ότου οι όλες οι αφίξεις εξυπηρετηθούν.

### 4.3 Μοντέλο διακριτών γεγονότων

Περιγράφεται στη συνέχεια ένα μοντέλο προσομοίωσης διακριτών γεγονότων του συστήματος αναμονής που προσεγγίζει τη λειτουργία της αποθήκης. Το σύστημα αναμονής έχει διαφορετικούς τύπους πελατών και διάφορους τύπους εξυπηρετούντων με διαφορετικούς χρόνους εξυπηρέτησης.

Γεγονότα συστήματος είναι:

Γεγονός 0<sub>j</sub>: άφιξη μίας κίνησης στο σύστημα που θα εξυπηρετηθεί με τον τύπο εξυπηρέτησης j

Γεγονός i<sub>j</sub>: αναχώρηση πελάτη από εξυπηρετούντα i=1,...,m του τύπου εξυπηρέτησης j

Ορίζουμε επίσης τις εξής μεταβλητές:

t = τρέχων χρόνος (ρολόι προσομοίωσης – το «τώρα»)

L<sub>j</sub> = αριθμός κινήσεων τύπου j σε αναμονή

τ<sub>j</sub> = χρόνος που συνέβη η προηγούμενη μεταβολή του L<sub>j</sub> (για να βρούμε το εμβαδόν του L(j))

N<sub>qj</sub> = εμβαδόν του αποθέματος L<sub>j</sub> για κάθε τύπο εξυπηρέτησης j, ενώ στο τέλος της προσομοίωσης διαφείνεται με τον χρόνο και δίνει το μέσο απόθεμα

T<sub>ij</sub> = χρόνος επόμενου γεγονότος στον εξυπηρετούντα ij με i=1,...,m και τύπο εξυπηρέτησης j=1,...,πλήθος διαφορετικών τύπων εξυπηρέτησης

x<sub>ij</sub> = διάρκεια εξυπηρέτησης

s<sub>ij</sub> = κατάσταση εξυπηρετούντα ij με i=1,...,m και τύπο εξυπηρέτησης j=1,...,πλήθος διαφορετικών τύπων εξυπηρέτησης

TH<sub>ij</sub> = πλήθος κινήσεων που αναχώρησαν από τον ij με i=1,...,m και τύπο εξυπηρέτησης j=1,...,πλήθος διαφορετικών τύπων εξυπηρέτησης

TH<sub>0j</sub> = πλήθος πελατών που εισήλθαν στην εξυπηρέτηση τύπου j

A<sub>j</sub> = πλήθος αδρανών εξυπηρετούντων τύπου j, με A<sub>j</sub> ≤ m<sub>j</sub>

Λ<sub>j</sub> = λίστα αδρανών εξυπηρετούντων j

Όταν s<sub>ikj</sub>=0 σημαίνει ότι ο εξυπηρετών S<sub>ikj</sub> είναι αδρανής και, επειδή δεν έχει προγραμματιστεί το τέλος της επόμενης εξυπηρέτησης για αυτόν, θέτουμε T<sub>ikj</sub>=∞. Όταν αργότερα κάποιος πελάτης ενεργοποιήσει τον S<sub>ikj</sub>, τότε ξεκινά η εξυπηρέτηση και θέτουμε s<sub>ikj</sub>=1 και T<sub>ikj</sub>=t+x<sub>ikj</sub>.

Όταν το σύστημα λειτουργεί θα υπάρχουν περίοδοι όπου όλοι οι εξυπηρετούντες είναι απασχολημένοι και περίοδοι όπου κάποιοι αδρανούν. Η λίστα Λ<sub>j</sub> περιέχει A<sub>j</sub> ≥ 0 στοιχεία. Όταν δεν είναι άδεια, το στοιχείο Λ<sub>1j</sub> είναι ο δείκτης του εξυπηρετούντα τύπου j που πρώτος από όλους τους A<sub>j</sub> είχε παραμείνει αδρανής, το Λ<sub>2j</sub> είναι ο δείκτης του δεύτερου κοκ. Όταν ο εξυπηρετών S<sub>ikj</sub> μένει αδρανής, τότε προσθέτουμε τον δείκτη k στην τελευταία θέση της λίστας τύπου j ως εξής:

$$A_j = A_j + 1$$

$$\Lambda_{Aj} = k$$

Όταν γίνεται άφιξη και υπάρχουν αδρανείς εξυπηρετούντες, το Λ<sub>1j</sub> είναι ο δείκτης εξυπηρετητή στον οποίο θα πάει ο «πελάτης» που μόλις έφτασε. Τότε το Λ<sub>1j</sub> αφαιρείται από τη λίστα και τα υπόλοιπα A<sub>j</sub> - 1 στοιχεία της λίστας j ανεβαίνουν μία θέση. Επίσης κρατάμε σε μία

μεταβλητή  $k$  για να γνωρίζουμε ποιος εξυπηρετών θα αναλάβει την εξυπηρέτηση. Οι εντολές είναι:

$k = \Lambda_{ij}$   
Για  $p=1, \dots, A_j - 1$   
 $\Lambda_{pj} = \Lambda_{pj+1}$   
Επόμενο  $p$   
 $A_j = A_j - 1$

Για απλούστευση υποθέτουμε ότι στην αρχή το σύστημα είναι άδειο. Ο αλγόριθμος διακριτών γεγονότων είναι ο ακόλουθος:

1. Ανάγνωση δεδομένων

$J$ =πλήθος διαφορετικών τύπων εξυπηρέτησης  
 $m(j)$ =αριθμός εξυπηρετούντων τύπου  $j$ ,  $j=1, \dots, J$   
Λίστα κινήσεων

2. Αρχικές τιμές

Χρόνος:  $t=0$

Γραμμή Λίστας που εξετάζουμε:  $LK=1$

Για  $j=1, \dots, J$

Ουρές άδειες:  $L_j=0$

Εμβασμών του  $L_j$ :  $Nq_j=0$

Γεννήτρια χρόνου άφιξης (πρώτου) πελάτη:  $TH_0=0$

Όλοι οι εξυπηρετούντες ελεύθεροι:  $A_j=m_j$

Για  $k=1, \dots, m_j$

Εξυπηρετήσεις:  $TH_{kj}=0$

Κατάσταση εξυπηρετούντος:  $s_{kj}=0$

Επόμενη αναχώρηση:  $T_{kj}=\infty$

Ο εξυπηρετών καταγράφεται στη λίστα αδρανών εξυπηρετούντων:  $\Lambda_{kj}=k$

Επόμενο  $k$

Αφίξεις:  $TH_0=0$

Επόμενο  $j$

Επαναλαμβάνουμε τα βήματα 3,4:

3. Χρόνος επόμενου γεγονότος ή περάτωση προσομοίωσης

$t=\infty$  (στην αρχή δεν έχει προσδιοριστεί)

Για  $n=1, \dots, J$  , για κάθε διαφορετικό τύπο εξυπηρέτησης

Για  $k=1, \dots, m_n$  , για κάθε εξυπηρετούντα τύπου  $n$

Αν  $T_{kn} < t$  , αναζήτηση μικρότερου στοιχείου στον πίνακα  $T$  και αποθήκευσή του

$i=k$

$j=n$

$t=T_{kn}$

Τέλος  $A_n$

Επόμενο  $k$

Επόμενο  $n$

Αν έχουν φθάσει και εξυπηρετηθεί όλες οι παραγγελίες, δηλαδή  $\sum TH_0 =$  (πλήθος γραμμών  $LK$ ) και  $\sum s_{ij}=0$  για  $i=1, \dots, m$  και  $j=1, \dots, J$  τότε εξάγονται τα αποτελέσματα:

$TSIM$  = χρόνος τελευταίου γεγονότος (μπορεί να βρεθεί με 2 τρόπους, ένας είναι στο βήμα 4.)

Για  $j=1,...,J$

Ενημερώνουμε το εμβαδόν του  $L_j$ :  $N_{qj}=N_{qj}+L_j(TSIM-\tau_j)$

Μέσος ρυθμός στις ουρές:  $N_{qj}= N_{qj}/ TSIM$

Για  $k=0,..., m_j$

Μέσος ρυθμός του  $k_j$ :  $TH_{kj}= TH_{kj}/ TSIM$

Επόμενο  $k$

Επόμενο  $j$

Εκτύπωση αποτελεσμάτων

Περατώνουμε την εκτέλεση του προγράμματος (διακοπή επαναλήψεων 3,4)

ΤέλοςΑν

4. Εκτέλεση επόμενου γεγονότος  $i$  τύπου  $j$  στιγμή  $t$

$TSIM = t$  , πιθανός χρόνος περάτωσης της προσομοίωσης αν αυτό το γεγονός είναι τελευταίο

Αν  $i=0$

$j$ =τύπος εξυπηρετούντος $_{AK}$

$js$ =τύπος βοηθητικού εξυπηρετούντος $_{AK}$

Αν  $T_{0j}>T_{0js}$

Αναλαμβάνει ο βοηθητικός εξυπηρετητής σε περίπτωση που είναι ενωρίτερα διαθέσιμος από τον πρώτο:  $j=js$

ΤέλοςΑν

Αν  $A_j>0$  (ο πελάτης πηγαίνει στην εξυπηρέτηση τύπου  $j$  και η ουρά μένει άδεια)

$TH_{0j}= TH_{0j}+1$

Ο εξυπηρετών  $\Lambda_{1j}$  πρόκειται να ενεργοποιηθεί:  $k=\Lambda_{1j}$

Ο  $\Lambda_{1j}$  αφαιρείται από τη λίστα  $\Lambda$  με τον τρόπο που προαναφέραμε

Ο εξυπηρετών  $k$  τύπου  $j$  ενεργοποιείται:

$s_{kj}=1$

$T_{kj}=t+x_{kj}$  (χρόνος εξυπηρέτησης)

Αλλιώς

Η ουρά θα αυξηθεί, για αυτό ενημερώνουμε το εμβαδόν:  $N_{qj}=N_{qj}+L_j(t-\tau_j)$

$L_j=L_j+1$

$\tau_j=t$

ΤέλοςΑν

Επόμενη γραμμή στη Λίστα Κινήσεων:  $\Lambda K=\Lambda K+1$

Αν  $\Lambda K<(\text{πλήθος γραμμών } \Lambda K)$

$T_{0j}= t+x_{0j}$  (χρόνος επόμενης άφιξης)

Αλλιώς

$T_{0j}= \infty$  (σταματάνε οι αφίξεις)

ΤέλοςΑν

Αλλιώς (αν  $i>0$ , που σημαίνει ότι αναχωρεί πελάτης από τον  $M_{ij}$ )

$TH_{ij}= TH_{ij}+1$

Αν  $L_j=0$

$s_{ij}=0$

$T_{ij}=\infty$

Προσθέτουμε το δείκτη  $i$  στο τέλος της λίστας  $\Lambda_i$ :

$A_j= A_j+1$

$\Lambda_{Aj}=i$

Αλλιώς (η ουρά μειώνεται και ο  $M_{ij}$  ξεκινά νέα εξυπηρέτηση)

Ενημερώνουμε το εμβαδόν:  $N_{qj}=N_{qj}+L_j(t-\tau_j)$



$$L_j = L_j - 1$$

$$\tau_j = t$$

$$T_{ij} = t + x_{ij} \text{ (χρόνος εξυπηρέτησης)}$$

ΤέλοςΑν

ΤέλοςΑν

5. Επανάληψη από το βήμα 3.

## 4.4 Παραμετροποίηση

### 4.4.1 Αφίξεις

Στο σύστημα που μελετάμε, όλες οι αφίξεις πραγματοποιούνται στην αρχή της ημέρας:

Αν  $LK < (\text{πλήθος γραμμών } LK)$

$$T_{0j} = t \text{ (χρόνος επόμενης άφιξης)}$$

Αλλιώς

$$T_{0j} = \infty \text{ (σταματάνε οι αφίξεις)}$$

ΤέλοςΑν

### 4.4.2 Χρόνος εξυπηρέτησης

Στο σύστημα που μελετάμε, δεν χρησιμοποιούμε γεννήτριες τυχαίων αριθμών γιατί οι χρόνοι εξυπηρέτησης εξαρτώνται από τη θέση του κάθε προϊόντος αν αναφερόμαστε σε κινήσεις Reach Truck και από το πλήθος των τεμαχίων ή των κιβωτίων αν αναφερόμαστε στις υπόλοιπες κατηγορίες picking. Για το λόγο αυτό, όταν μία άφιξη πηγαίνει στην ουρά, πρέπει να αποθηκεύσουμε μαζί της και τη γραμμή της λίστας που βρίσκεται η κίνηση αυτή ώστε να μπορεί να υπολογιστεί εκ νέου ο χρόνος εξυπηρέτησης όταν αυτό χρειαστεί. Αυτό μπορούμε να το επιτύχουμε δημιουργώντας έναν κενό πίνακα που θα ονομάσουμε QAK όπου σε κάθε του στήλη βρίσκονται οι αριθμοί των γραμμών από τη λίστα κινήσεων που εκκρεμούν.

Επομένως κάνουμε τις εξής αλλαγές:

4. Εκτέλεση επόμενου γεγονότος  $i$  τύπου  $j$  τη στιγμή  $t$

Αν  $i=0$

....

Αν  $A_j > 0$  (ο πελάτης πηγαίνει στην εξυπηρέτηση τύπου  $j$  και η ουρά μένει άδεια)

...

Αλλιώς

Η ουρά θα αυξηθεί, για αυτό ενημερώνουμε το εμβαδόν:  $Nq_j = Nq_j + L_j(t - \tau_j)$

$$L_j = L_j + 1$$

$$QAK(L_{ij}) = LK$$

$$\tau_j = t$$

ΤέλοςΑν

...

Αλλιώς (αν  $i > 0$ , που σημαίνει ότι αναχωρεί πελάτης από τον  $M_{ij}$ )

$TH_{ij} = TH_{ij} + 1$

Αν  $L_j = 0$

...

Αλλιώς (η ουρά μειώνεται και ο  $M_{ij}$  ξεκινά νέα εξυπηρέτηση)

Ενημερώνουμε το εμβαδόν:  $N_{qj} = N_{qj} + L_j(t - \tau_j)$

$L_j = L_j - 1$

$\tau_j = t$

$\Delta Kp = Q\Delta K(1, j)$

Ο  $\Delta Kp_{1j}$  αφαιρείται από τη λίστα  $\Delta Kp_{j1}$

$T_{ij} = t + x_{ij}$  (χρόνος εξυπηρέτησης)

ΤέλοςΑν

ΤέλοςΑν

5.Επανάληψη από το βήμα 3.

Για τον υπολογισμό του χρόνου εξυπηρέτησης θα χρειαστούμε την απόσταση που διανύει ο εξυπηρετών εντός της αποθήκης, και τα ύψη που προσεγγίζει σε κάθε θέση. Αυτά, σε συνδυασμό με τα τεχνικά χαρακτηριστικά κάθε μηχανήματος μας δίνουν τον χρόνο εξυπηρέτησης για κάθε κίνηση.

TravelSpeed=2.78; %m/s  
LiftSpeed=0.43; %m/s  
LoweringSpeed=0.52; %m/s  
ReachSpeed=0.2; %m/s  
WalkSpeed=0.83; %m/s

Η απόσταση εξαρτάται από την μεταξύ απόσταση των δύο θέσεων και από την τελευταία θέση που βρισκόταν ο εξυπηρετών πριν από αυτή την κίνηση. Οι τελευταίες θέσεις κάθε εξυπηρετητή βρίσκονται αποθηκευμένες σε έναν πίνακα με όνομα LastLoc και κάθε φορά που τρέχουμε την συνάρτηση της εύρεσης χρόνου δίνουμε σαν δεδομένα το γράφημα (H) που έχουμε προαναφέρει, την τελευταία θέση της προηγούμενης κίνησης (LastLoc) και τις δύο νέες θέσεις (Loc1, Loc2) με τον ακόλουθο τρόπο:

dist= shortestpath(H,LastLoc,Loc1)+shortestpath(H,Loc1,Loc2)

Στη μεταβλητή High αποθηκεύουμε τα ύψη που θα χρειαστεί να προσεγγίσει το ReachTruck και για τις θέσεις. Οπότε ο τελικός χρόνος εξυπηρέτησης είναι:

stime=6/ReachSpeed+dist/TravelSpeed+High/LiftSpeed+High/LoweringSpeed+5;

Πιο Αναλυτικά:

6/ReachSpeed : Σε μία κίνηση απαιτούνται κινήσεις 4\*1.5m για την εισαγωγή και εξαγωγή της περόνης του Reach Truck στην παλέτα. Μονάδες m/(m/s)=s

dist/TravelSpeed: Ο χρόνος που απαιτείται για να καλυφθεί η απόσταση στο επίπεδο. Μονάδες m/(m/s)=s

High/LoweringSpeed: Ο χρόνος που απαιτείται για την ανύψωση και την καθίζηση της περόνης καθ'όλη τη διάρκεια της κίνησης. Μονάδες m/(m/s)=s

Στο τέλος του τύπου προσθέτουμε +5 δευτερόλεπτα για το χρόνο που χρειάζεται ο οδηγός του Reach Truck για να σκαννάρει το προϊόν και να ενημερώσει το απόθεμα.

Για τη διαδικασία των micro και Standard picking θα υπολογίσουμε με βάση την ποσότητα (q) που ζητείται σε κάθε κίνηση.

Για την εξαγωγή κάθε μονάδας τεμαχιακού προϊόντος απαιτείται ένα (1) δευτερόλεπτο και ο χρόνος μέχρι ο διαλογέας να προσεγγίσει το ράφι από την προηγούμενή του θέση.

dist= shortestpath(H,LastLoc,Loc1)

stime=q+dist/WalkSpeed

ενώ για το κιβωτιακό η εξαγωγή απαιτεί δύο (2) δευτερόλεπτα και έχουμε:

dist= shortestpath(H,LastLoc,Loc1)

stime=2\*q+dist/WalkSpeed

Formatted: Greek

Formatted: Greek

Formatted: Greek

Οι παραπάνω ενέργειες πραγματοποιούνται με τις συναρτήσεις

- **PAL**, για την εύρεση της βέλτιστης διαδρομής μεταξύ δύο θέσεων και το ύψος τους από το επίπεδο και
  - ο **findstime** όπου με τα παραπάνω δεδομένα και τις τεχνικές προδιαγραφές των ανυψωτικών μηχανημάτων, βρίσκουμε τον αναμενόμενο χρόνο περάτωσης της κάθε εργασίας.

Ως δεδομένο λαμβάνουμε και το σύνολο τόξων και κόμβων που έχουμε ορίσει για όλες τις αποστάσεις στο επίπεδο με βάση τους διαδρόμους και τα φατνώματα που έχουμε για την αποθήκη.

## 4.5 Αλγόριθμος βελτίωσης

Στη συνέχεια, για να μπορέσουμε να εκτελέσουμε τον αλγόριθμο της προσομοίωσης του σεναρίου που θέλουμε να εξετάσουμε, θα χρειαστεί να δουλέψουμε σε έναν χώρο ο οποίος λειτουργεί ήδη κάτω από τις προϋποθέσεις του νέου μας συστήματος ειδικά οι δείκτες παραγωγής που θα εμφανιστούν δεν θα αφορούν μια μόνιμη κατάσταση, όπου είναι και το ζητούμενο, αλλά τη μεταβατική περίοδο από την πραγματικότητα στο σενάριο που μελετάμε.

Προκειμένου να προσπεράσουμε το παραπάνω κώλυμα δημιουργήσαμε έναν αλγόριθμο (Nstock.m) ο οποίος αποθέτει εκ νέου όλες τις ποσότητες, με τρόπο τέτοιο ώστε να λαμβάνονται υπόψη οι συστάδες που έχουμε ορίσει και οι συνθήκες αποθήκευσης που διέπουν τα προϊόντα και τις θέσεις. Ο αλγόριθμος αυτός (Nstock.m), όπως και αυτός της βελτίωσης λειτουργούν με τη χρήση της συνάρτησης findlocation.

### 4.5.1 Συνάρτηση findlocation

Μία συνάρτηση η οποία δέχεται σαν ορίσματα:

- Λίστα με διαθέσιμες θέσεις αποθήκευσης και τα χαρακτηριστικά τους
  - Συστάδα πρόσβασης
  - Όγκος
  - Συνθήκες Αποθήκευσης
  - Αποθέτης
  - Δωμάτιο εντός της αποθήκης
- Λίστα προϊόντων με τα χαρακτηριστικά τους
  - Συστάδα ταχυκινήσις
  - Όγκος
  - Συνθήκες Αποθήκευσης
  - Αποθέτης
- Προτεραιότητες αποθήκευσης όσον αφορά
  - Τα Δωμάτια ανά αποθέτη
  - Τις επιτρεπτές εναλλακτικές συνθήκες αποθήκευσης των θέσεων που μπορούμε να χρησιμοποιήσουμε για να αποθηκεύσουμε ένα προϊόν
- Το προϊόν προς εξέταση
- Τον αποθέτη του εν λόγω προϊόντος

Στη συνέχεια δημιουργεί μια λίστα με τις πιθανές θέσεις αποθήκευσης και επιλέγει αυτή με τον ελάχιστο όγκο με σκοπό την εξοικονόμηση μεγαλύτερων θέσεων για την απόθεση προϊόντων με μεγαλύτερου όγκου. Επειδή το εμβαδόν της παλέτας είναι σταθερό (1,2μ×0,8μ) από εδώ και στο εξής θα μελετάμε τις θέσεις και τα προϊόντα βάσει ύψους.

Στην περίπτωση που δεν βρεθεί θέση με τις αρχικές προδιαγραφές του εξεταζόμενου προϊόντος μπορούμε να μεταβούμε σε νέα κριτήρια, τόσο σε διαφορετικές συστάδες όσο και σε διαφορετικές συνθήκες αποθήκευσης τις οποίες έχουμε ορίσει ως επιτρεπτές στα ορίσματα της συνάρτησης (πχ ένα προϊόν γενικών συνθηκών, που δεν αντιμετωπίζει ποιοτικά θέματα, να μπορεί να μπει σε μία θέση κλιματισμού). Τέλος, αν αποτύχει η αναζήτηση με τα παραπάνω κριτήρια και υπάρξει υπερχείλιση λόγω έλλειψης διαθέσιμων θέσεων, αποθηκεύουμε αναγκαστικά τα προϊόντα στο χώρο του άλλου αποθέτη.

Ορίσματα Συνάρτησης

Προϊόν

MAT: Προϊόν προς αποθήκευση

ΣΑ: Συνθήκη αποθήκευσης προϊόντος

ΥΨΟΣ: Ύψος Προϊόντος πάνω στην Παλέτα

Δ: Λίστα Δωματίων που μπορεί να αποθηκευτεί ένα Προϊόν

c: Συστάδα που ανήκει το Προϊόν (c)

DEP: Αποθέτης Προϊόντος

Δεδομένα

Stock: Λίστα διαθέσιμων θέσεων για αποθήκευση

Στήλες Λίστας Stock(Αναφέρονται στα χαρακτηριστικά της χωροθέσης και ΟΧΙ του προϊόντος):

[Χωροθέση Δωμάτιο ΣΑ\_αρχική Ύψος c ΣΑ\_τελική]

Λ: Λίστα επιτρεπόμενων Συνθηκών αποθήκευσης

Βοηθητικές Μεταβλητές

ΘΕΣΗ: Χωροθέση που εξετάζουμε

c1: Συστάδα για αναδρομή

c\_αρχ: Αρχική συστάδα Προϊόντος

k: Δείκτης λίστας Δ

BPEΘΗΚΕ: Έλεγχος εύρεσης θέσης (0 ή 1)

A:Πίνακας Διαθέσιμων θέσεων που αποτελούν επιλογές για το προϊόν

j: Δείκτης λίστας A

Αρχικοποίηση τιμών

BPEΘΗΚΕ=0

k=1

A=[0 0 0 0 0 0]

c\_αρχ=c

c1=c

Αλγόριθμος Επιλογής Χωροθέσης

Αν DEP=='DEP0003'

Επιλογή δεσμευμένων χώρων από τον Αποθέτη3: Δ=[4A 5A 5B]

Αλλιώς

Δ=[1A 1B 2A 2B 3A 3B 4B]

ΤέλοςΑν

Όσο (c>0 και c<4 και c1>0 και c1<4 και k<(πλήθος γραμμών Λ)+1 και BPEΘΗΚΕ==0)

Για i από 1 έως (πλήθος γραμμών Stock)

ΘΕΣΗ=Stock(i,1)

Αν (Stock(i,2) ανήκει στο Δ) και (Stock(i,3) ανήκει στο Λ) και (Stock(i,4)<ΥΨΟΣ) και (Stock(i,5)==c)

A(j,:)=Stock(i,:)

j=j+1

ΤέλοςΑν

Επόμενο i

Αν A(1,1)==0

Αν k<(πλήθος γραμμών Λ)

```

        k=k+1 (επόμενη ΣΑ)
ΑλλιώςΑν c<3
    c=c+1 (επόμενη συστάδα)
    k=1(αρχικοποίηση ΣΑ)
ΑλλιώςΑν c>1 και c1>1
    c1=c1-1 (κάνουμε αναδρομή στις συστάδες)
    c=c1
    k=1(αρχικοποίηση ΣΑ)
Αλλιώς
    Αν DEP=='DEP0003'
        Δ=[1A 1B 2A 2B 3A 3B 4B] (αναγκαστική επιλογή υπόλοιπων δωματίων)
    Αλλιώς
        Δ=[4A 5A 5B]
    ΤέλοςΑν
    c=c_αρχ (αρχικοποίηση c)
    c1=c (αρχικοποίηση c1)
    k=1(αρχικοποίηση ΣΑ)
    ΤέλοςΑν
    j=1
    A=[0 0 0 0 0 0]
Αλλιώς
    BPEΘΗΚΕ=1
ΤέλοςΑν
ΤέλοςΌσο
Αν BPEΘΗΚΕ==1
    Ταξινόμησε τον Α με βάση το Ύψος (στήλη 4, ΥΨΟΣ)
    Χωροθέση=A(1,1)
    Δωμάτιο=A(1,2)
    ΣΑ_τελική=A(1,6)
Αλλιώς
    Εκτύπωσε "Δέν βρέθηκε θέση για το προϊόν"
ΤέλοςΑν

```

Με έντονα γράμματα στο τέλος έχουμε τα αποτελέσματα που δίνει η συνάρτηση στο πρόγραμμα προκειμένου να ενημερώσει το απόθεμα.

Εφόσον βρεθεί ικανή θέση για την αποθήκευση του αποθέματος, θα χρειαστούμε την απόσταση που θα διανύσει ο εξυπηρετών προκειμένου να την προσεγγίσει και στη συνέχεια να υπολογίσουμε τον χρόνο εξυπηρέτησης, επομένως συνεχίζουμε όπως και πριν με τις συναρτήσεις PAL και findstime.

## 5 Αριθμητικά Αποτελέσματα

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα αποτελέσματα των δύο προσομοιώσεων που πραγματοποιήσαμε και θα τα αναλύσουμε και θα τα συγκρίνουμε ως προς τα μέτρα απόδοσής τους. Η πρώτη προσομοίωση, που αφορά τη λειτουργία του υφιστάμενου συστήματος, περατώθηκε σε σημαντικά μικρότερο χρονικό διάστημα σε σχέση με τη δεύτερη όπου απαιτήθηκε επανειλημμένα η χρήση συναρτήσεων.

### 5.1 Υπολογιστικός χρόνος εκτέλεσης

	Επαναλήψεις	Συνολικός Χρόνος	Χρόνος Συνάρτησης
body	1	04:53.9	00:19.1
findtime	19654	03:55.5	00:01.3
PAL	19060	03:44.8	01:47.0

Σχήμα 5.1 Χρόνος Περάτωσης Πρώτης Προσομοίωσης

	Επαναλήψεις	Συνολικός Χρόνος	Χρόνος Συνάρτησης
body	1	34:22.9	01:57.4
findlocation	1636	17:41.6	06:42.2
findtime	49083	12:06.3	00:04.0
PAL	47604	11:34.7	05:31.3

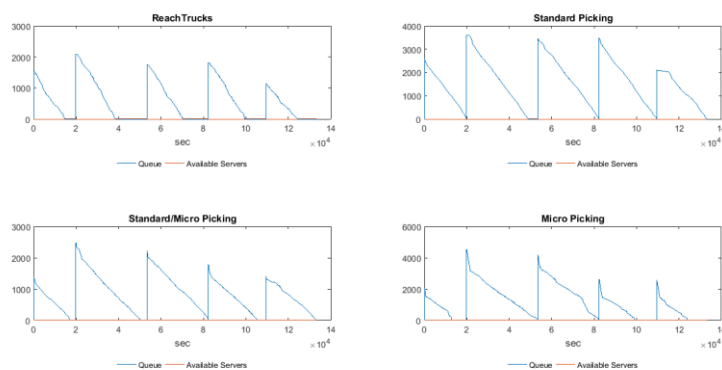
Σχήμα 5.2 Χρόνος Περάτωσης Δεύτερης Προσομοίωσης

Ονομάζοντας body τον αλγόριθμο προσομοίωσης και findlocation, findtime και PAL τις συναρτήσεις που δημιουργήσαμε, παρατηρούμε πως ο δεύτερος αλγόριθμος (του προτεινόμενου σεναρίου λειτουργίας) χρειάζεται πολύ περισσότερο χρόνο για να περατωθεί, πράγμα αναμενόμενο δεδομένης της συνεχούς χρήσης συναρτήσεων για την εύρεση κατάλληλης θέσης προϊόντος με βάση τις συνθήκες αποθήκευσης του και τις ομάδες που τα έχουμε καταναίμει.

### 5.2 Μέτρα απόδοσης

Τα μέτρα απόδοσης που μπορούμε να έχουμε από τον παραπάνω αλγόριθμο σχετίζονται με το μέσο πλήθος πελατών στην ουρά του κάθε εξυπηρετούντος, τους διαθέσιμους εξυπηρετούντες σε κάθε στιγμή και το μέσο ρυθμό εξυπηρέτησης από αυτόν. Παρόλα αυτά, στην δικιά μας εφαρμογή, όλες οι αφίξεις πραγματοποιούνται στην αρχή της κάθε ημέρας με αποτέλεσμα να μην έχουμε εικόνα για τις ουρές που δημιουργούνται καθώς για κάθε ημέρα έχουν φθίνουσα πορεία μέχρι να φτάσουν στο μηδέν και να επανέλθουν οι διαθέσιμοι εξυπηρετούντες στο αρχικό τους πλήθος.



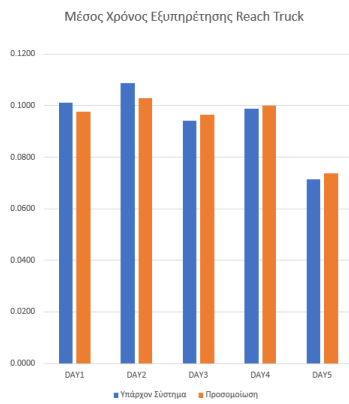


Σχήμα 5.1 Πλήθος σε ουρές και διαθέσιμους εξυπηρετητές κατά τη διάρκεια της προσομοίωσης

Αντίθετα, όσον αφορά τον μέσο ρυθμό εξυπηρέτησης, παρατηρούμε πως τις πρώτες δύο ημέρες το υπάρχον σύστημα δείχνει να υπερτερεί από το δικό μας σενάριο, πράγμα που αντιστρέφεται στις επόμενες ημέρες.

	Υπάρχον Σύστημα	Προσομοίωση
Ημέρα 1	0.1010	0.0977
Ημέρα 2	0.1087	0.1030
Ημέρα 3	0.0942	0.0965
Ημέρα 4	0.0988	0.1000
Ημέρα 5	0.0715	0.0736

Σχήμα 5.2 Μέσος ρυθμός εξυπηρέτησης Reach Trucks κάθε ημέρα.



Σχήμα 5.3 Σύγκριση μέσων ρυθμών εξυπηρέτησης ανυψωτικών μηχανημάτων.

## 6 Συμπεράσματα

Στην παρούσα εργασία μελετήθηκε η λειτουργία μίας κεντρικής αποθήκης διανομής προϊόντων και, με κατάλληλη χρήση των δεδομένων που μας παραχωρήθηκαν από την διοίκηση της αποθήκης, προτάθηκε ένα μοντέλο χωροθέτησης των προϊόντων ανά είδος και αναπτύχθηκε ένα μοντέλο προσομοίωσης του συστήματος. Η χωροθέτηση έγινε χρησιμοποιώντας στατικά δεδομένα απόστασης κάθε αποθηκευτικής θέσης από την είσοδο και έξοδο της αποθήκης και δυναμικά δεδομένα που αφορούν συχνότητες και ποσότητες τροφοδοσίας και ζήτησης προϊόντων, τα οποία κατηγοριοποιήθηκαν σε τρεις κύριες ομάδες με βάση την κινητικότητα τους. Συνδυάζοντας τις πληροφορίες αυτές με ένα κριτήριο αποτελεσματικής χρήσης του συστήματος διακίνησης υλικών, με στόχο την εξοικονόμηση πόρων και χρόνου, έγινε η αντιστοίχιση κάθε τύπου προϊόντος σε κάποιες αποθηκευτικές θέσεις μέσα στην αποθήκη. Στη συνέχεια, έγινε ανακατανομή της αντιστοιχίας προϊόντων σε θέσεις μέσα στο κέντρο σύμφωνα με ειδικές συνθήκες συντήρησής τους και με βάση τις τρεις κατηγορίες κινητικότητας προκειμένου να τηρηθούν κάποιοι πρόσθετοι λειτουργικοί περιορισμοί.

Ακολούθως αναπτύχθηκαν δύο αλγόριθμοι προσομοίωσης, ένας για τον υπάρχοντα τρόπο λειτουργίας της κεντρικής αποθήκης και ένας με το μοντέλο τριών ταχυτήτων. Από τη σύγκριση των αποτελεσμάτων των δύο αλγορίθμων παρατηρούμε πως οι διαφορές μεταξύ των δύο συστημάτων δεν είναι μεγάλες. Αυτό οφείλεται στην πολυπλοκότητα εύρεσης κατάλληλης θέσης αποθήκευσης για κάθε προϊόν. Πιο αναλυτικά, ο διαχωρισμός των προϊόντων ανάλογα με την κινητικότητά τους εντός της αποθήκης με σκοπό την ευκολότερη εξαγωγή τους σε περιπτώσεις ανατροφοδοσίας και διαλογής, δεν οδηγεί σε σημαντικές διαφοροποιήσεις έναντι της υπάρχουσας κατάστασης διότι οι περιορισμοί των συνθηκών αποθήκευσης είναι πολύ ισχυροί και το ποσοστό πληρότητας της αποθήκης είναι υψηλό και δεν επιτρέπουν στα προϊόντα να αποθηκεύονται με βάση την κινητικότητα. Ακόμα, το κέντρο αποθήκευσης και διανομής που μελετάμε λειτουργεί κυρίως με παραγγελίες επιπέδου κιβωτίων ή και μικρότερες οπότε το «κέρδος» που θα έχουμε -λόγω διαφορετικής χωροθέτησης- στον χρόνο συλλογής και απόθεσης μιας ολόκληρης παλέτας είναι πολύ μικρό.

Σε κάθε περίπτωση, ο αλγόριθμος που έχουμε δημιουργήσει αποτελεί, με τις κατάλληλες παραμετροποιήσεις, ένα εργαλείο για μελλοντικούς πειραματισμούς με αλλαγές-βελτιώσεις στους χρόνους περάτωσης εργασιών και το διαθέσιμο πλήθος των εξυπηρετούντων για την εύρυθμη λειτουργία του και τέλος τον προγραμματισμό των αναγκών λειτουργίας του συστήματος. Μερικά εφαρμογές προσομοιώσεων σε τέτοιου είδους κεντρικές αποθήκες είναι η μελέτη για το αν ένας χώρος έχει τη δυνατότητα να φιλοξενήσει έναν νέο αποθέτη με τα προϊόντα του, αν το ανθρώπινο δυναμικό μπορεί να καλύψει τις ανάγκες σε περιόδους αυξημένης ζήτησης των προϊόντων και πιο γενικά, στο σχεδιασμό μίας κεντρικής αποθήκης από το μηδέν όσον αφορά την εύρεση κατάλληλου χώρου και την επιλογή πλήθους εργαζομένων.

## 7 Βιβλιογραφία

Κουϊκόγλου Β., Κωνσταντάς Δ., 2016. *Προσομοίωση Συστημάτων Διακριτών Γεγονότων*. Εκδόσεις Δίσιγμα.

Μαρινάκης Ι., Μυγδαλάς Α., 2008. *Σχεδιασμός και Βελτιστοποίηση της Εφοδιαστικής Αλυσίδας*, Θεσσαλονίκη: Εκδόσεις Σοφία

Bartholdi J. J., III, Hackman T. S., 2019. *Warehouse & Distribution Science, Release 0.98.1*. Available: <https://www.warehouse-science.com/>

Kłodawski M., Jacyna M., Lewczuk K., Wasiak M., 2017. "The issues of selection warehouse process strategies," *Procedia Engineering*, vol. 187, 451-457

Malaguti E., Nannicini G., Thomopulos D., 2018. "Optimizing allocation in a warehouse network," *Electronic Notes in Discrete Mathematics*, vol. 64, 195-204

## Παράρτημα Α: Κώδικες στη Matlab

### SIM (αλγόριθμος προσομοίωσης υπάρχοντος μοντέλου)

```
clc
clear all
[num,txt,stock]=xlsread('Stock.xlsx');
[num,txt,locations]=xlsread('Locations.xlsx');
[num,txt,transactions]=xlsread('TransTest.xlsx');
[highs,frametypes]=xlsread('frames.xlsx');
[num,txt,LastLoc]=xlsread('LastLoc.xlsx');
load('WarehouseMapf.mat');
DAYS=[2 7666;7667 20423;20424 32067;32068 41842;41843 49089];
t=0; %xronos
GT=[];
DTHmean=[];
DTH=[];

for hh=1:5 %days
m=[5 11 6 5]; %server qnty
%RT KP BP MP
A=m;%available servers
check(1:4)=0;

Tzero(1:size(m,2))=0;%xronos epomenou gegonotos
L=zeros(1,size(m,2)); %oura
Nq(1:size(m,2))=0; %meso L
pointer=zeros(1,size(m,2));

t_el(1:size(m,2))=0; %xronos pou sinevi i proigoumeni metavoli tou L(i)

%voithitikoi pinakes%
vima=1;
Time(vima)=0;
J=[0 A L];
%-----%

%-----Servers-----%
for j=1:size(m,2)
    for k=1:max(m)
        if k<m(j)+1
            TH(k,j)=0;
            s(k,j)=0;
            T(k,j)=inf;
            ListAX(k,j)=k;
        else
            T(k,j)=NaN;
            s(k,j)=NaN;
            TH(k,j)=NaN;
        end
    end
end
end
THzero(1:size(m,2))=0;
THall=0;
List=DAYS(hh,1);
Listfirst=List;
Listbr=DAYS(hh,2);%numel(transactions(:,1));
while sum(TH(:),'omitnan')<Listbr-Listfirst
```

```

THall*100/(Listbr-Listfirst)
%3.epomeno gegonos
t=inf;
if List<Listbr
    %%anathesi
    d=strsplit(transactions{List,2});
    d=str2num(d{1});
    if d==1 %paralavi
        j=1;
        js=j;
    elseif d==2 %endodiakinisi
        j=1;
        js=j;
    elseif d==3
        if isequal(transactions{List,3},'Micro')
            j=4;
            js=3; %3
        elseif isequal(transactions{List,3},'Standard')
            j=2;
            js=3; %3
        elseif isequal(transactions{List,3},'Full Pallet Picking')
            j=1;
            js=j;
        elseif isequal(transactions{List,3},'Part of Pallet')
            j=2;
            js=3; %3
        end
    elseif d==4
        j=1 ;
        js=j;
    elseif d==6
        j=2;
        js=3; %3
    else
        j=2;
        js=3;
    end
end
[minNum, minIndex] = min(T(:)); %evresi epomenou gegonotos se olo ton
pinaka T
[row, col] = ind2sub(size(T), minIndex);
if List<Listbr
    if Tzero(j)<t
        r=rand;
        %ean einai apasxolimenos Second Type Server
        if rand>0.77
            j=js;
        end
        F(j)=0;
        t=Tzero(j);
    end
    if minNum<t %
        j=col;
        F(j)=row;
        t=minNum;
    end
else
    j=col;
    F(j)=row;
    t=minNum;
end
end

```

```

%4.ektelesi epomenou gegonotos

if F(j)==0
    if A(j)>0
        THzero(j)=THzero(j)+1;
        k=ListAX(1,j);
        ListAX(:,j)=[ListAX(2:end,j);0];
        s(k,j)=1;
        A(j)=A(j)-1;

[stime,tel_thes]=findstime(transactions(List,:),locations,H,highs,frametype
s,LastLoc{k,j});
        T(k,j)=t+stime; %%xronos
        LastLoc{k,j}=tel_thes;

    else
        THzero(j)=THzero(j)+1;
        Nq(j)=Nq(j)+L(j)*(t-t_el(j));
        L(j)=L(j)+1; %oura
        pointer(j)=pointer(j)+1;
        Qlist(pointer(j),j)=List;
        t_el(j)=t; %%na dw

    end
    List=List+1;
    if List<Listbr
%        Tzero(j)=min(T(j,:),);
        Tzero(j)=t; %%1.69;
    else
        Tzero(:)=inf

    end

%
else %feygei apo to systima
    TH(F(j),j)=TH(F(j),j)+1;
    if L(j)==0
        s(F(j),j)=0;
        T(F(j),j)=inf;
        A(j)=A(j)+1;
        ListAX(A(j),j)=F(j);

    else
        Nq(j)=Nq(j)+L(j)*(t-t_el(j));
        L(j)=L(j)-1;
        t_el(j)=t;
        listp=Qlist(1,j);
        Qlist(:,j)=[Qlist(2:end,j);0];
        pointer(j)=pointer(j)-1;

%
[stime,tel_thes]=findstime(transactions,txt,listp,highs,frametypes,LastLoc{
F(j),j});

[stime,tel_thes]=findstime(transactions(listp,:),locations,H,highs,frametyp
es,LastLoc{F(j),j});
        T(F(j),j)=t+stime; %%xronous
        LastLoc{F(j),j}=tel_thes;

    end

end

```

```

if t>0
if ismember(t,J(:,1),'rows')
row=find(J(:,1)==t);
for j=2:5
if A(j-1)>J(row,j)
J(row,j)=A(j-1);
end
end
for j=6:9
if L(j-5)>J(row,j)
J(row,j)=L(j-5);
end
end
else
J(size(J,1)+1,:)= [t A L];
J=sortrows(J,1);
end

end

if t>0
for i=1:4
if L(i)==0 && check(i)==0 && A(i)==m(i)
ENDS(i)=t;
check(i)=1;
end

end

end

% B(vima+1,:)=L;
% D(vima+1,:)=A;
% Time(vima+1)=t;
% vima=vima+1;
THall=sum(TH(:),'omitnan');
end
DENDS(hh,:)=ENDS;
if hh==1
er=0;
else
er=max(GT(:,1));
end
THzeroall=sum(THzero(:));
THall=sum(TH(:),'omitnan')
vima
sum(THzeroall(:),'omitnan')-sum(TH(:),'omitnan')
TSIM=max(J(:,1));
for i=1:4
Ngmean(i)=Nq(i)/ENDS(i);
THmean(:,i)=[THzero(i)/ENDS(i);TH(:,i)/ENDS(i)];
frt(i)=(16*60*60-ENDS(i))/2;
end
GT=[GT;J(:,1)+er J(:,2:end)];
DTHmean=[DTHmean;THmean];
DTH=[DTH;TH];
end

```



```

THzeroall=sum(THzero(:));
THall=sum(TH(:),'omitnan')
vima
sum(THzeroall(:),'omitnan')-sum(TH(:),'omitnan')
TSIM=max(GT(:,1));
for i=1:4
    Nqmean(i)=Nq(i)/ENDS(i);
    THmean(:,i)=[THzero(i)/ENDS(i);TH(:,i)/ENDS(i)];
    frt(i)=(16*60*60-ENDS(i))/2;
end

```

## SIM2 (αλγόριθμος προσομοίωσης μοντέλου που εξετάζουμε)

```

[num,txt,stock]=xlsread('NstockFreeAll.xlsx');
[num,txt,locations]=xlsread('Locations.xlsx');
[num,txt,transactions]=xlsread('TransTest.xlsx');
[highs,frametypes]=xlsread('frames.xlsx');
[num,txt,LastLoc]=xlsread('LastLoc.xlsx');
% [num,txt,Stock]=xlsread('Zones/StockClusters.xlsx');
[num,txt,MATS]=xlsread('MatData/Data.xlsx');
[prnum,prttxt,PRIOR]=xlsread('Zones/thadoume.xlsx');
[num,txt,StockLocs]=xlsread('StockLocsFreeAll.xlsx');
[num,txt,StockLocsDat]=xlsread('Zones/StockClusters.xlsx');
load('WarehouseMapff.mat');
StockLocs=StockLocs(3:end,:);

DAYS=[2 7666;7667 20423;20424 32067;32068 41842;41843 49089];
t=0; %xronos
GT=[];
DTHmean=[];
DTH=[];
for hh=1:2
m=[5 11 6 5];
%RT KP BP MP
A=m;
check(1:4)=0;
Nq(1:size(m,2))=0;
Tzero(1:size(m,2))=0;
L=zeros(1,size(m,2)); %oura
pointer=zeros(1,size(m,2));
List=DAYS(hh,1);
Listfirst=List;
Listbr=DAYS(hh,2);
t_el(1:size(m,2))=0;
vima=1;
Time(vima)=0;
J=[0 A L];
%-----Servers-----%
for j=1:size(m,2)
    for k=1:max(m)
        if k<m(j)+1
            TH(k,j)=0;
            s(k,j)=0;
            T(k,j)=inf;
            ListAX(k,j)=k;

```

```

        else
            T(k,j)=NaN;
            s(k,j)=NaN;
            TH(k,j)=NaN;
        end
    end
end
THzero(1:size(m,2))=0;
THall=0;
while sum(TH(:),'omitnan')<Listbr-Listfirst

    i=THall*100/(Listbr-Listfirst);
    sprintf('%d %-.2f ',hh,i)
    %3.epomeno gegonos
    t=inf;
    if List<Listbr
        d=strsplit(transactions{List,2});
        d=str2num(d{1});
        if d==1 %paralavi
            j=1;
            js=j;
        elseif d==2 %endodiakinisi
            j=1;
            js=j;
        elseif d==3
            if isequal(transactions{List,3},'Micro')
                j=4;
                js=3; %3
            elseif isequal(transactions{List,3},'Standard')
                j=2;
                js=3; %3
            elseif isequal(transactions{List,3},'Full Pallet Picking')
                j=1;
                js=j;
            elseif isequal(transactions{List,3},'Part of Pallet')
                j=2;
                js=3; %3
            end
        elseif d==4
            j=1 ;
            js=j;
        elseif d==6
            j=2;
            js=3; %3
        else
            j=2;
            js=3;
        end
    end
    [minNum, minIndex] = min(T(:));
    [row, col] = ind2sub(size(T), minIndex);
    if List<Listbr
        if Tzero(j)<t
            r=rand;
            if rand>0.77
                j=js;
            end
            F(j)=0;
            t=Tzero(j);
        end
    end
    if minNum<t

```

```

        j=col;
        F(j)=row;
        t=minNum;
    end
else
    j=col;
    F(j)=row;
    t=minNum;
end
%4.ektelesi epomenou gegonotos
if F(j)==0
    if A(j)>0
        THzero(j)=THzero(j)+1;
        k=ListAX(1,j);
        ListAX(:,j)=[ListAX(2:end,j);0];
        s(k,j)=1;
        A(j)=A(j)-1;
        DEP=transactions(List,1);
        SSCC=transactions(List,15);
        MAT= transactions(List,7);
        QNT=transactions(List,8);
        SYSK=transactions(List,9);
        EXP=transactions(List,11);
        BATCH=transactions(List,12);
        PALTYPE=transactions(List,20);
        OLD=transactions(List,22);
        QLTY=transactions(List,21);
        XAR=transactions(List,23);
        KATAST=transactions(List,24);
        if j==1
            Loc1=transactions(List,13);
            Loc2=transactions(List,14);
            if strcmp(transactions(List,2), '02 - Ενδοδιακίνηση')
                PARALAVI='not sure';
                if strcmp(Loc1(3), 'R') && ~strcmp(Loc2(3), 'R')
%Απόθεση μετά από Παραλαβή
                    [Loc,Room,SA_final,SA,cluster_final] =
findlocation( StockLocs,MATS,PRIOR,MAT,DEP );
                    k1=find(strcmp(StockLocs(:,1),Loc));
                    if strcmp(SA, '05 - Καν.Συνθ. (Γυάλινη
Συσκευασία)')%Αλλαγή Τύπου θέσης Αποθήκευσης λόγω γυάλινης συσκευασίας
                        Lev=Loc(8:9);
                        for i=str2num(Lev):-10:10
newLoc=strcat(Loc(1:7),num2str(i),Loc(10:end))
                                if
max(ismember(locations(:,1),newLoc))==1
n=find(strcmp(locations(:,1),newLoc));
                                locations{n,5}='05 - Καν.Συνθ.
(Γυάλινη Συσκευασία)';
                                    end
                                end
                                end
                                Loc2=Loc;
                                end
                                stock(end+1,:)={DEP Loc2 Room 'Παλετοθέση' SSCC
'No' MAT QNT SYSK EXP BATCH PALTYPE OLD QLTY XAR KATAST PARALAVI SA_final
};
                                k1=find(strcmp(StockLocs(:,1),Loc2));
                                StockLocs(k1,:)=[];

```

```

        if ~strcmp(Loc1(3),'R') && ~strcmp(Loc2(3),'R')
            k1=find(strcmp(stock(:,5),SSCC),1);
            if QNT<stock{k1,8}
                stock{k1,8}=stock{k1,8}-QNT;
            else
                stock([k1,:])=[];
                k1=find(strcmp(StockLocsDat(:,1),Loc1));
                StockLocs=[StockLocs;StockLocsDat(k1,:)]
            end
            StockLocs=sortrows(StockLocs,10);
        end
        end
        transactions(List,13)=Loc1;
        transactions(List,14)=Loc2;
    end
    if strcmp(transactions(List,2),'03 - Picking') ||
strcmp(transactions(List,2),'04 - Αναρροφοδοσ[α]')
        if strcmp(Loc1(3),'R') %&& ~strcmp(Loc2(3),'R')
            Loc=Loc1;
        else
            k1=find(strcmp(stock(:,5),SSCC),1);
            Loc=stock{k1,2};
            if QNT<stock{k1,8}
                stock{k1,8}=stock{k1,8}-QNT;
            else
                stock([k1,:])=[];
                k1=find(strcmp(StockLocsDat(:,1),Loc));
                StockLocs=[StockLocs;StockLocsDat(k1,:)]
            end
            StockLocs=sortrows(StockLocs,10);
        end
        end
        %diagrafi apo stock an einai picking kai allagi
        %thesis an einai anatrof
        transactions(List,13)=Loc;
        transactions(List,14)=Loc2;
    end
end

[stime,tel_thes]=findtime(transactions(List,:),locations,H,highs,frametype
s,LastLoc{k,j});
    T(k,j)=t+stime;%%%xronos
    LastLoc{k,j}=tel_thes;

    else
        THzero(j)=THzero(j)+1;
        Nq(j)=Nq(j)+L(j)*(t-t_el(j));
        L(j)=L(j)+1; %oura
        pointer(j)=pointer(j)+1;
        Qlist(pointer(j),j)=List;
        t_el(j)=t; %%%na dw
    end
    List=List+1;
    if List<Listbr
        %
        Tzero(j)=min(T(j,:),);
        Tzero(j)=t;
    else
        Tzero(:)=inf;
    end
else %feygei apo to systima

```

```

TH(F(j),j)=TH(F(j),j)+1;
if L(j)==0
    s(F(j),j)=0;
    T(F(j),j)=inf;
    A(j)=A(j)+1;
    ListAX(A(j),j)=F(j);
else
    Nq(j)=Nq(j)+L(j)*(t-t_el(j));
    L(j)=L(j)-1;
    t_el(j)=t;
    listp=Qlist(1,j);
    Qlist(:,j)=[Qlist(2:end,j);0];
    pointer(j)=pointer(j)-1;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%5
    DEP=transactions(listp,1);
    SSCC=transactions(listp,15);
    MAT= transactions(listp,7);
    QNT=transactions(listp,8);
    SYSK=transactions(listp,9);
    EXP=transactions(listp,11);
    BATCH=transactions(listp,12);
    PALTYPE=transactions(listp,20);
    OLD=transactions(listp,22);
    QLTY=transactions(listp,21);
    XAR=transactions(listp,23);
    KATAST=transactions(listp,24);
    if j==1
        Loc1=transactions(listp,13);
        Loc2=transactions(listp,14);
        if strcmp(transactions(listp,2),'02 - Ενδοδιακίνηση')
            PARALAVI='not sure';
            if strcmp(Loc1(3),'R') && ~strcmp(Loc2(3),'R')
                [Loc,Room,SA_final,SA] = findlocation(
StockLocs,MATS,PRIOR,MAT,DEP );
                k1=find(strcmp(StockLocs(:,1),Loc));
                if strcmp(SA,'05 - Καν.Συνθ. (Γυάλινη
Συσκευασία)')
                    Lev=Loc(8:9);
                    for i=str2num(Lev):-10:10
newLoc=strcat(Loc(1:7),num2str(i),Loc(10:end))
                        if
max(ismember(locations(:,1),newLoc))==1
n=find(strcmp(locations(:,1),newLoc));
                        locations{n,5}='05 - Καν.Συνθ.
(Γυάλινη Συσκευασία)';
                                end
                                end
                                end
                                Loc2=Loc;
                                end
                                stock(end+1,:)= {DEP Loc2 Room 'Παλετοθήκη' SSCC
'No' MAT QNT SYSK EXP BATCH PALTYPE OLD QLTY XAR KATAST PARALAVI SA_final
};
                                k1=find(strcmp(StockLocs(:,1),Loc2));
                                StockLocs(k1,:)=[];
                                if ~strcmp(Loc1(3),'R') && ~strcmp(Loc2(3),'R')
                                    k1=find(strcmp(stock(:,5),SSCC),1);
                                    if QNT<stock{k1,8}
                                        stock{k1,8}=stock{k1,8}-QNT;

```

```

        else
            stock([k1,:])=[];
            k1=find(strcmp(StockLocsDat(:,1),Loc1));
            StockLocs=[StockLocs; [StockLocsDat(k1,:)
StockLocsDat(k1,3)]];

            StockLocs=sortrows(StockLocs,10);
        end
        end
        transactions(listp,13)=Loc1;
        transactions(listp,14)=Loc2;
    end
    if strcmp(transactions(listp,2),'03 - Picking') ||
strcmp(transactions(listp,2),'04 - Ανατροφοδότηση')
        if strcmp(Loc1(3),'R') || ~strcmp(Loc1(3),'N')
            Loc=Loc1;
        else
            k1=find(strcmp(stock(:,5),SSCC),1);
            Loc=stock{k1,2};
            if QNT<stock{k1,8}
                stock{k1,8}=stock{k1,8}-QNT;
            else
                stock([k1,:])=[];
                k1=find(strcmp(StockLocs(:,1),Loc));
                StockLocs=[StockLocs; [StockLocsDat(k1,:)
StockLocsDat(k1,3)]];
                StockLocs=sortrows(StockLocs,10);
            end
        end
        end

        %diagrafi apo stock an einai picking kai allagi
        %thesis an einai anatrof
        transactions(listp,13)=Loc;
        transactions(listp,14)=Loc2;
    end
end

%
[stime,tel_thes]=findtime(transactions,txt,listp,highs,frametypes,LastLoc{
F(j),j});
    end

[stime,tel_thes]=findtime(transactions(listp,:),locations,H,highs,frametypes,LastLoc{F(j),j});
T(F(j),j)=t+stime; %%xronous
LastLoc{F(j),j}=tel_thes;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if t>0
if ismember(t,J(:,1),'rows')
    row=find(J(:,1)==t);
    for j=2:5
        if A(j-1)>J(row,j)
            J(row,j)=A(j-1);
        end
    end
    for j=6:9
        if L(j-5)>J(row,j)
            J(row,j)=L(j-5);
        end
    end
end

```

```

        end
    else
        J(size(J,1)+1,:)= [t A L];
        J=sortrows(J,1);
    end

end
if t>0
    for i=1:4
        if L(i)==0 && check(i)==0 && A(i)==m(i)
            ENDS(i)=t;
            check(i)=1;
        end
    end

    end

    THall=sum(TH(:),'omitnan');
end
DENDS(hh,:)=ENDS;
if hh==1
    er=0;
else
    er=max(GT(:,1));
end
THzeroall=sum(THzero(:));
THall=sum(TH(:),'omitnan')
vima
sum(THzeroall(:),'omitnan')-sum(TH(:),'omitnan')
TSIM=max(J(:,1));
for i=1:4
    Ngmean(i)=Nq(i)/ENDS(i);
    THmean(:,i)=[THzero(i)/ENDS(i);TH(:,i)/ENDS(i)];
    frt(i)=(16*60*60-ENDS(i))/2;
end
GT=[GT;J(:,1)+er J(:,2:end)];
DTHmean=[DTHmean;THmean];
DTH=[DTH;TH];
end

```

## Συνάρτηση PAL

```

function
[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th)
a=5.69; %apostasi metaksy diadromwn
frame_width=2.75;
apostasiaporafi=1.95;
pallw=0.8;
dim_bet_pall=(frame_width-3*pallw)/4;
Location1=transaction{13};
Location2=transaction{14};
if Location1(3)=='R'
    loc1=Location1(3:5);
    High1=0;
else
    if Location1(5)=='1' || Location1(5)=='2'
        cor='a';
    end
end

```

```

elseif Location1(5)=='3' || Location1(5)=='4'
    cor='b';
elseif Location1(5)=='5' || Location1(5)=='6'
    cor='c';
else
    cor='d';
end
if Location1(11)>3
    pal1=3;
else
    pal1=Location1(11);
end
loc1=strcat(Location1(4),cor,Location1(6:7),num2str(pal1));
%ipsi
i=find(strcmp(Locations,Location1));
frame_t1=Locations{i,9};
f1=frame_t1(4:end);
j=find(strcmp(frametypes,f1));
if mod(str2num(Location1(8:9)),10)==0
    Lev1=str2num(Location1(8));
else
    Lev1=str2num(Location1(9));
end
if Lev1==1
    High1=0;
else
    High1=sum(highs(j,1:Lev1-1));
end
end
if Location2(3)=='R'
    loc2=Location2(3:5);
    High2=0;
else
    if Location2(5)=='1' || Location2(5)=='2'
        cor='a';
    elseif Location2(5)=='3' || Location2(5)=='4'
        cor='b';
    elseif Location2(5)=='5' || Location2(5)=='6'
        cor='c';
    else
        cor='d';
    end
    if Location2(11)>3
        pal2=3;
    else
        pal2=Location2(11);
    end
    loc2=strcat(Location2(4),cor,Location2(6:7),num2str(pal2));
    i=find(strcmp(Locations,Location2));
    frame_t2=Locations{i,9};
    f2=frame_t2(4:end);
    j=find(strcmp(frametypes,f2));
    if mod(str2num(Location2(8:9)),10)==0
        Lev2=str2num(Location2(8));
    else
        Lev2=str2num(Location2(9));
    end
    if Lev2==1
        High2=0;
    else
        High2=sum(highs(j,1:Lev2-1));
    end
end

```



```

    end
end
[P,dist1] = shortestpath(H,from_th,loc1);
[P,dist2] = shortestpath(H,loc1,loc2);
High=High1+High2;
if isequal(transaction{3},'Micro')
    dist=dist1;
    LastLoc=loc1;
elseif isequal(transaction{3},'Standard')
    dist=dist1;
    LastLoc=loc1;
elseif isequal(transaction{3},'Part of Pallet')
    dist=dist1;
    LastLoc=loc1;
else
    dist=dist1+dist2;
    LastLoc=loc2;
end
end

```

## Συνάρτηση findstime

```

function [stime,LastLoc] =
findstime(transaction,Locations,H,highs,frametypes,from_th)
a=5.69; %απόσταση μεταξύ διαδρόμων
frame_width=2.75; %πλάτος φατνώματος(ραφιού)
apostasiaporafi=1.95;
pallw=0.8;
dim_bet_pall=(frame_width-3*pallw)/4;
Location1=transaction{13}; %θέση 1
Location2=transaction{14}; %θέση 2
%ταχύτητες μετακίνησης%
TravelSpeed=2.78; %m/s; Ταχύτητα Ανυψωτικού Περονοφόρου
LiftSpeed=0.43; %m/s Ανύψωση
LoweringSpeed=0.52; %m/s Κατάβαση
ReachSpeed=0.2; %m/s Προσέγγιση
WalkSpeed=0.83; %m/s Ταχύτητα Ανθρώπου
BTSpeed=1.67; %m/s Ταχύτητα Ηλ.Παλετοφόρου
%-----%
d=strsplit(transaction{2}); %Τύπος Κίνησης
d=str2num(d{1}); %Αριθμός Τύπου Κίνησης
q=transaction{8}; %Ποσότητα Συσκευασίας Κίνησης
q2=transaction{10}; %Ποσότητα Κίνησης σε Κιβώτια
if q2<q
    q=q2; %Μετατροπή σε κιβώτια εάν αυτό είναι
απάραικτο
end
if d==1 %Παραλαβή
    Locstion2=transaction{14};
    loc2=Location2(3:5); %θέση προς μετακίνηση
    [P,dist] = shortestpath(H,from_th,loc2); %Εύρεση απόστασης από
προηγούμενη θέση εξυπηρετούντα
    stime=5+dist/TravelSpeed; %Χρονος διεκπεραίωσης
εντολής
    LastLoc=loc2; %Αποθήκευση τελευταίας
θέσης εξυπηρετούντα
elseif d==2 %Ενδοδιακίνηση Παλλετών

```

```

[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th);
%Συνάρτηση εύρεσης μετακίνησης στο επίπεδο και καθ' ύψος

stime=3/ReachSpeed+dist/TravelSpeed+High/LiftSpeed+High/LoweringSpeed+3;
%Χρονος διεκπεραίωσης εντολής
elseif d==3
    if isequal(transaction{3},'Micro')%Micro Picking

[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th);
    stime=q+dist/WalkSpeed;
    elseif isequal(transaction{3},'Standard')

[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th);
    stime=3*q+dist/WalkSpeed;
    elseif isequal(transaction{3},'Full Pallet Picking')

[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th);
stime=6/ReachSpeed+dist/TravelSpeed+High/LiftSpeed+High/LoweringSpeed+5;
    elseif isequal(transaction{3},'Part of Pallet')

[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th);
    stime=5*q+dist/WalkSpeed;
    end
    elseif d==4 %Ανατροφοδοσία Θέσης Picking

[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th);
stime=6/ReachSpeed+dist/TravelSpeed+High/LiftSpeed+High/LoweringSpeed+5;
    elseif d==6 %Αναπλήρωση Θέσης Micro Picking

[dist,High,LastLoc]=PAL(transaction,Locations,H,highs,frametypes,from_th);
    stime=5+dist/BTSpeed;
    else %Ακυρο
        stime=0;
        LastLoc=from_th;
    end
end
end

```

## Συνάρτηση findlocation

%Εύρεση κατάλληλης θέσης προς Αποθήκευση

%Ορίσματα:

%StockLocs	Διαθέσιμες θέσεις προς αποθήκευση
%MATS	Λίστα Προϊόνων με Χαρακτηριστικά
%PRIOR	Λίστα προτεραιότητας Συνθηκών Αποθήκευσης ανά προϊόν
%MAT	Προϊόν που εξετάζουμε με χαρακτηριστικά
%DEF	Αποθέτης Προϊόντος που εξετάζουμε

%Αποτελέσματα:

%Location	Θέση Προς Αποθήκευση
%Room_final	Θάλαμος Αποθήκευσης
%SA_final	Τελικές Συνθήκες Αποθήκευσης
%SA	Πρώτη Επιλογή Θέσης Αποθήκευσης
%cluster	Συστάδα Προϊόντος με βάση την ταχυκίνησή του

```

%Λοιπές Μεταβλητές
%

function [Location,Room_final,SA_final,SA,cluster] = findlocation(
StockLocs,MATS,PRIOR,MAT,DEP )
    k=find(strcmp(MATS(:,1),MAT));
    SA=MATS{k,2}; %Συνθήκες Αποθήκευσης προϊόντος
    HIGH=MATS{k,3}*1000; %Ύψος Προϊόντος
    cluster=MATS{k,4}; %Συστάδα που ανοίκει το Προϊόν
    newcluster=cluster; %Συστάδα θέσης που εξετάζεται

    %Επιλογή δωματίων προς εξέταση με βάση τον αποθέτη
    if strcmp(DEP,'DEP0003')
        k=find(strcmp(PRIOR(:,1),DEP));
    else
        k=2;
    end
    Rooms=PRIOR(k,2:end); %Δωμάτια που θα εξεταστούν
    Rooms(cellfun(@(Rooms) any(isnan(Rooms)),Rooms)) = [];

    %Επιλογή συνθήκες αποθήκευσης θέσεων που θα εξεταστούν βάσει προϊόντος
    k=find(strcmp(PRIOR(:,1),SA));
    SAPR=PRIOR(k,2:end);
    SAPR(cellfun(@(SAPR) any(isnan(SAPR)),SAPR)) = [];

    j=1;
    found=0;
    A=num2cell(zeros(1,size(StockLocs,2))); %Πίνακας θέσεων που
    μπορούμε να αποθηκεύσουμε
    k=1;
    while newcluster<4 && k<size(SAPR,2)+1 && found==0 && cluster<4 &&
    cluster>0

        %εύρεση όλων των πιθανών θέσεων βάσει των κριτηρίων που εξετάζουμε
        for i=2:size(StockLocs,1)
            if strcmp(StockLocs{i,5},'3 - Παλετοθέση') &&
ismember(StockLocs{i,2},Rooms) && StockLocs{i,9}==cluster % &&
StockLocs{i,4}>HIGH && strcmp(StockLocs{i,3},SAPR{k})
                A(j,:)=StockLocs(i,:);
                j=j+1;
            end
        end

        %Αλλαγή κριτηρίων σε περίπτωση που δεν έχουμε διαθέσιμες θέσεις
        if A{1,1}==0;
            if k<size(SAPR,2) %Επόμενη δυνατή Συνθήκη Αποθήκευσης
                k=k+1;
            elseif cluster<3 %Επόμενη συστάδα θέσης
                cluster=cluster+1;
                k=1;
            elseif cluster>1 && newcluster>1 %προηγούμενη συστάδα
                newcluster=newcluster-1;
                cluster=newcluster;
                k=1;
            else %Αν όλα αποτύχουν έχουμε δεν υπάρχει καμία θέση και
            αναγκαστικά πηγαίνουμε στους άλλους χώρους.
                if strcmp(DEP,'DEP0003')

```

```

        k=2;
    else
        k=1;
    end
    Rooms=PRIOR(k,2:end);
    Rooms(cellfun(@(Rooms) any(isnan(Rooms)),Rooms)) = [];
    k=find(strcmp(MATS(:,1),MAT));
    cluster=MATS{k,4};
    newcluster=cluster;
    k=1;
end
j=1;
clear A
A=num2cell(zeros(1,size(StockLocs,2)));
else
    found=1;
end
end
if found==1
    AB=sortrows(A,4); %ταξινόμηση με βάση το ύψος σε αύξουσα σειρά για
    εξοικονόμηση χώρου και επιλογή της μικρότερης δυνατής θέσης που χωράει το
    προϊόν

    Location=AB{1,1};
    Room_final=AB{1,2};
    SA_final=AB{1,3};

    clear AB
else
    sprintf('Didnt found Location for %c%c%c%c%c%c, Sorry :/',MAT)
end
end
end

```