

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική εργασία για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση  
του Διπλώματος Μηχανολόγου Μηχανικού

**ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ  
ΑΝΑΘΕΣΗΣ ΠΛΟΙΩΝ ΣΕ ΠΡΟΒΛΗΤΕΣ ΣΕ ΕΝΑ  
ΕΜΠΟΡΕΥΜΑΤΙΚΟ ΛΙΜΑΝΙ**



της φοιτήτριας Μούκα Βενετίας

Επιβλέπων Καθηγητής: κ. Γεώργιος Κ.Δ. Σαχαρίδης

ΒΟΛΟΣ, 2014



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ  
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 12851/1  
Ημερ. Εισ.: 08-09-2014  
Δωρεά: Συγγραφέα  
Ταξιθετικός Κωδικός: ΠΤ – ΜΜ  
2014  
ΜΟΥ

**Μέλη τριμελούς εξεταστικής επιτροπής της διπλωματικής:**

Πρώτος εξεταστής  
(Επιβλέπων)

κ. Γιώργος Κ.Δ. Σαχαρίδης  
Λέκτορας Επιχειρησιακής Έρευνας και Βιομηχανικής  
Διοίκησης, Τμήμα Μηχανολόγων Μηχανικών,  
Πανεπιστήμιο Θεσσαλίας

Δεύτερος εξεταστής

κ. Γιώργος Κοζανίδης  
Επίκουρος Καθηγητής Μεθόδων Βελτιστοποίησης  
Συστημάτων Παραγωγής/Υπηρεσιών, Τμήμα  
Μηχανολόγων Μηχανικών, Πανεπιστήμιο  
Θεσσαλίας

Τρίτος εξεταστής

κ. Δημήτρης Παντελής  
Επίκουρος Καθηγητής Στοχαστικών Προτύπων  
Επιχειρησιακής Έρευνας στη Βιομηχανική  
Διοίκηση, Τμήμα Μηχανολόγων Μηχανικών,  
Πανεπιστήμιο Θεσσαλίας

## **Ευχαριστίες**

Πρώτα απ' όλα, θέλω να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, κ. Γιώργιο Σαχαρίδη, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου. Οφείλω ευχαριστίες και στα υπόλοιπα μέλη της εξεταστικής επιτροπής της διπλωματικής εργασίας μου, Καθηγητές κ. Γιώργιο Κοζανίδη και κ. Δημήτρη Παντελή για την συμμετοχή τους και την ανάγνωση της εργασίας μου. Επίσης, ευχαριστώ τον συνάδελφο Κωνσταντίνο Καπούλα για τη συνδρομή του στις μοντελοποιήσεις της ενότητας 2.1 του Κεφαλαίου 2. Τέλος, ευχαριστώ την συνάδελφο Ευτυχία Κωσταρέλου για τη πολύτιμη βοήθεια της στον προγραμματισμό με CPLEX.

Μούκα Βενετία

## Περίληψη

Ο προγραμματισμός λιμένων αναφέρεται στην ανάθεση των εμπορευματικών πλοίων σε κατάλληλες θέσεις ελλιμενισμού (προβλήτες), κατά μήκος της αποβάθρας, για την εξυπηρέτησή τους. Στη παρούσα διπλωματική εργασία, αρχικά, περιγράφεται το σύνολο των διαδικασιών και η λήψη των αποφάσεων που πραγματοποιούνται σε ένα εμπορευματικό λιμάνι, προκειμένου να επιτευχθεί ο σωστός προγραμματισμός για το πού και το πότε τα πλοία θα πρέπει να προσαράξουν για την αποτελεσματική και γρήγορη εξυπηρέτησή τους. Στη συνέχεια, αναπτύσσονται ορισμένοι τρόποι μοντελοποίησης του προβλήματος ανάθεσης πλοίων σε προβλήτες, που αφορούν διακριτή διάταξη αποβάθρας, χρόνο συνεχή, αλλά και διακριτό και στοχεύουν κυρίως στην ελαχιστοποίηση του συνολικού χρόνου εξυπηρέτησης των πλοίων. Τέλος, με χρήση παραδειγμάτων τα μοντέλα αυτά επιλύονται με τη βοήθεια της CPLEX και παρουσιάζονται τα αριθμητικά τους αποτελέσματα αλλά και συγκρίσεις μεταξύ αυτών.

## Πίνακας Περιεχομένων

<b>Κεφάλαιο 1 Περιγραφή διαδικασιών σε ένα εμπορευματικό λιμάνι .....</b>	<b>10</b>
1.1 Πράσινη διαχείριση στους τερματικούς σταθμούς.....	11
1.2 Η «πράσινη διαχείριση» ως πηγή συγκριτικού πλεονεκτήματος .....	11
1.3 Η περίπτωση του Hutchison Port Holdings (HPH).....	12
1.4 Οι φιλικές προς το περιβάλλον δραστηριότητες του HIT .....	12
1.5 Από την κατανάλωση καυσίμου στις εκπομπές των πλοίων.....	15
1.6 Σύστημα αποφάσεων για τις εργασίες σε έναν τερματικό σταθμό εμπορευματοκιβωτίων .....	16
1.7 Εμπορευματοκιβώτια, μέρη αποθήκευσης, γερανοί ναυπηγείου .....	17
1.8 Εισερχόμενα - εξερχόμενα εμπορευματοκιβώτια και εξοπλισμός που χρησιμοποιείται σε έναν τερματικό σταθμό .....	18
1.9 Εργασίες στις αυλές αποθήκευσης.....	19
1.9.1 Οι λειτουργίες ενός τερματικού σταθμού εμπορευματοκιβωτίων .....	19
1.9.2 Οι ροές των εξερχόμενων και εισερχόμενων κιβωτίων .....	19
1.9.3 Οργάνωση πλοίου που μεταφέρει εμπορευματοκιβώτια .....	20
1.9.4 Γερανογέφυρες ελαστικοφόρες (RTGCs) και οι λειτουργίες τους .....	21
1.10 Αποφάσεις για τις καθημερινές λειτουργίες .....	22
1.11 Το λιμάνι του Χονγκ Κονγκ.....	24
1.12 Η εισαγωγή δεδομένων και η περίοδος προγραμματισμού για τη λήψη αποφάσεων .....	25
1.13 Πηγές διακοπής των εργασιών σε έναν τερματικό σταθμό .....	27
1.14 Κρίσιμα προβλήματα βελτιστοποίησης.....	28
1.14.1 Πρόβλημα κατανομής πλοίων σε προβλήτες (berth allocation problem).....	28
1.14.2 Πρόβλημα ανάθεσης γερανών αποβάθρας (quay crane assignment problem).....	32
1.14.3 Πρόβλημα προγραμματισμού των γερανών αποβάθρας (quay crane scheduling problem) .....	34
1.15 Πρόβλημα στοιβάγματος εμπορευματοκιβωτίων .....	37

1.16	Πρόβλημα δρομολόγησης και προγραμματισμού πλοίων που μεταφέρουν υγροποιημένο φυσικό αέριο .....	38
1.17	Ο ρόλος της ταχύτητας στην ενεργειακή απόδοση των θαλάσσιων μεταφορών .....	40
<b>Κεφάλαιο 2 Περιγραφή και μοντελοποίηση του προβλήματος ανάθεσης (BAP) .....</b>		<b>42</b>
2.1	Παρουσίαση πρώτης μοντελοποίησης του προβλήματος ανάθεσης πλοίων .....	42
2.2	Περίπτωση διακριτού χρόνου στο πρόβλημα ανάθεσης πλοίων .....	47
2.3	Γενικά συμπεράσματα από την παρουσίαση των τριών μοντελοποιήσεων .....	50
<b>Κεφάλαιο 3 Αριθμητικά αποτελέσματα .....</b>		<b>51</b>
3.1	Παρουσίαση παραδειγμάτων .....	51
3.2	Παρουσίαση αριθμητικών αποτελεσμάτων .....	53
3.3	Προσθήκη περιορισμών για την βελτίωση των τριών μοντελοποιήσεων (Valid inequalities) .....	58
3.4	Υπολογισμός και συγκριτικά αποτελέσματα του χρόνου αναμονής .....	63
3.5	Συμπεράσματα από τη παρουσίαση των αριθμητικών αποτελεσμάτων και συγκρίσεων .....	68
<b>Κεφάλαιο 4 Συμπεράσματα .....</b>		<b>69</b>
<b>Παράρτημα .....</b>		<b>70</b>
<b>Βιβλιογραφία .....</b>		<b>118</b>

## Λίστα ακρωνύμων

HPH=Hutchison Port Holdings.....	12
HIT=Hong Kong International Terminals.....	12-14
EPD=Environmental Product Declarations.....	12,13
IPCC=Intergovernmental Panel on Climate Change.....	15
IMO=International Maritime Organization.....	15
MPEC=Minor Planet Electronic Circular.....	15
MARPOL=Marine Pollution.....	15
TEU=Twenty-foot Equivalent Unit.....	17,24,25
RTGCs= Rubber Tyred Gantry Cranes.....	17-19,21,22,24
RMGCs=Rail Mounted Gantry Cranes.....	18,24
QCs=Quay Cranes.....	19,20,25,32,34,36
XTs=External Trucks.....	19,24
ITs=Internal Trucks.....	19,20,24
COSCO=China Ocean Shipping Company.....	24
OBAP=Operational Berth Allocation Problem.....	28
TBAP=Tactical Berth Allocation Problem.....	28
QCAP=Quay Crane Assignment Problem.....	33
BAP=Berth Allocation Problem.....	33,34,42
QCSP=Quay Crane Scheduling Problem.....	34-36
LNG=Liquefied Natural Gas.....	37,38
CSP=Container Stacking Problem.....	35
DBAP=Discrete Berth Allocation Problem.....	40



## Λίστα εικόνων

Εικόνα 1: Περιβαλλοντική οργανωτική δομή του HIT .....	14
Εικόνα 2: Η αυλή αποθήκευσης σε έναν τερματικό σταθμό.....	16
Εικόνα 3: Μια ελαστικοφόρα γερανογέφυρα (RTGC).....	17
Εικόνα 4: Εξυπηρέτηση ενός πλοίου από τρεις γερανούς αποβάθρας .....	18
Εικόνα 5: Η ροή των εξερχόμενων εμπορευματοκιβωτίων .....	20
Εικόνα 6: Προβλήματα προγραμματισμού σε έναν τερματικό σταθμό εμπορευματοκιβωτίων .....	23
Εικόνα 7: Περιπτώσεις διάταξης προβλητών .....	29
Εικόνα 8: Παρουσίαση των χρονικών διαστημάτων σε προβλήτες .....	31
Εικόνα 9: Ανάθεση γερανών στα πλοία .....	33
Εικόνα 10: Γερανοί αποβάθρας εξυπηρετούν τα πλοία που έχουν προσαράξει στις προβλήτες .....	35
Εικόνα 11: Διαδοχικές κινήσεις προγραμματισμού των παραθαλάσσιων επιχειρήσεων .....	37
Εικόνα 12: Οι εκπομπές διοξειδίου του άνθρακα από τη ναυτιλία σε σχέση με τις συνολικές παγκόσμιες εκπομπές για το 2007.....	41

## Λίστα πινάκων

Πίνακας 1: Αριθμός πλοίων και προβλητών που χρησιμοποιήθηκαν για την επίλυση των τριών μοντέλων .....	52
Πίνακας 2: Παρουσίαση αριθμητικών παραδειγμάτων για τις τρεις μοντελοποιήσεις .....	54
Πίνακας 3: Σύγκριση χρόνων επίλυσης μεταξύ 2ης και 3ης μοντελοποίησης με $(CPU_2 - CPU_3)/CPU_3$ .....	55
Πίνακας 4: Σύγκριση χρόνων επίλυσης μεταξύ 1ης και 3ης μοντελοποίησης με $(CPU_1 - CPU_3)/CPU_3$ .....	56
Πίνακας 5: Σύγκριση χρόνων επίλυσης μεταξύ 1ης και 2ης μοντελοποίησης με $(CPU_1 - CPU_2)/CPU_2$ .....	57
Πίνακας 6: Παρουσίαση αριθμητικών αποτελεσμάτων για τη 1η μοντελοποίηση με την προσθήκη του περιορισμού (28) .....	60
Πίνακας 7: Παρουσίαση αριθμητικών αποτελεσμάτων για τη 2η μοντελοποίηση με την προσθήκη του περιορισμού (29) .....	61
Πίνακας 8: Παρουσίαση αριθμητικών αποτελεσμάτων για τη 3η μοντελοποίηση με την προσθήκη του περιορισμού (30) .....	62
Πίνακας 9: Χρόνος αναμονής (WT) που χρησιμοποιήθηκε για τα παραδείγματα των τριών μοντελοποιήσεων .....	64
Πίνακας 10: Σύγκριση των πραγματικών με τους θεωρητικούς χρόνους αναμονής που υπολογίστηκαν για τη 1η μοντελοποίηση .....	65
Πίνακας 11: Σύγκριση των πραγματικών με τους θεωρητικούς χρόνους αναμονής που υπολογίστηκαν για τη 2η μοντελοποίηση .....	66
Πίνακας 12: Σύγκριση των πραγματικών με τους θεωρητικούς χρόνους αναμονής που υπολογίστηκαν για τη 3η μοντελοποίηση .....	67

## 1. ΠΕΡΙΓΡΑΦΗ ΔΙΑΔΙΚΑΣΙΩΝ ΣΕ ΕΝΑ ΕΜΠΟΡΕΥΜΑΤΙΚΟ ΛΙΜΑΝΙ

Αρχικά, στην παρούσα διπλωματική εργασία γίνεται λόγος για τις αποφάσεις που πρέπει να παρθούν σε ένα τερματικό σταθμό για το σύνολο των εργασιών που πραγματοποιούνται, λαμβάνοντας υπόψη τις επιπτώσεις που μπορεί να έχουν στο φυσικό περιβάλλον. Οι εργασίες αυτές σχετίζονται με τον εξοπλισμό που θα χρησιμοποιηθεί για την εξυπηρέτηση των πλοίων, τη σωστή λειτουργία των γερανών για το φόρτωμα και το ξεφόρτωμα, τα εμπορεύματα και πως αυτά μεταφέρονται από και προς τους χώρους αποθήκευσης. Επιπλέον, γίνεται αναφορά στους παράγοντες που επηρεάζουν και μπορούν να προκαλέσουν διακοπή αυτών των λειτουργιών. Τέλος, παρουσιάζονται τα κρίσιμα προβλήματα βελτιστοποίησης που προκύπτουν σε ένα τερματικό σταθμό και πως αυτά επιδρούν στην αποδοτικότητα του σταθμού.

### 1.1. «Πράσινη διαχείριση» στους τερματικούς σταθμούς

Η βιώσιμη οικονομία είναι μια έννοια που προκαλεί την ανησυχία τόσο της κοινωνίας όσο και της βιομηχανίας. Μελέτες για τον ρυθμό εξάντλησης των φυσικών πόρων μπορούν να χρονολογηθούν τουλάχιστον από τη δεκαετία του 1970. Στη σημερινή εποχή, η πρόκληση για τις βιομηχανίες είναι να υποστηρίζουν την οικονομική ανάπτυξη, ενώ παράλληλα να εξασφαλίζουν τη προστασία του περιβάλλοντος. Λόγω του αυξημένου ενδιαφέροντος για την αειφόρο ανάπτυξη, αποκαλούμενη «πράσινη διαχείριση», δημιουργείται το ζήτημα ελαχιστοποίησης των αρνητικών επιπτώσεων στο φυσικό περιβάλλον και τίθεται σημαντικό θέμα για τις επιχειρήσεις να επιτύχουν τα κέρδη τους, τηρώντας τις δεσμεύσεις που απαιτούνται για την προστασία του περιβάλλοντος. Σε έναν τερματικό σταθμό εμπορευματοκιβωτίων, ο διαχειριστής επιλέγεται ανάλογα με την περίπτωση για να τονίσει την εφαρμογή του προτεινόμενου μοντέλου «πράσινης διαχείρισης». Οι τερματικοί σταθμοί είναι κόμβοι που συνδέονται με άλλα μέσα χερσαίας μεταφοράς, όπως είναι οι αυτοκινητόδρομοι, σιδηρόδρομοι και άλλες εσωτερικές πλωτές μεταφορές (Lun et al., 2008). Επιπλέον, αποτελούν μια διεπαφή μεταξύ των τομέων της παραγωγής και της κατανάλωσης, προσελκύοντας το ενδιαφέρον των ανθρώπων της ναυτιλίας και της μεταφοράς (Song et al., 2010; Ugbooma et al., 2009). Οι διαχειριστές των τερματικών σταθμών χειρίζονται τις δραστηριότητες τόσο από τη λήψη των διαφόρων εμπορευματοκιβωτίων μέχρι και τη φόρτωση τους στα πλοία, όσο από την αποστολή μέχρι και την εκφόρτωση τους από τα πλοία. Οι ίδιοι αναλαμβάνουν και μια σειρά από δραστηριότητες σχεδιασμού, όπως ο

προγραμματισμός της προβλήτας, ο τρόπος που στοιβάζονται τα πλοία και η οργάνωση της αυλής στο λιμάνι.

## **1.2. Η «πράσινη διαχείριση» ως πηγή συγκριτικού πλεονεκτήματος**

Η έγκριση και η υιοθέτηση ενός μοντέλου πράσινης διαχείρισης είναι σημαντικές για πολλούς λόγους. Επικεντρώνεται στον προσδιορισμό των βέλτιστων πρακτικών και παράλληλα στοχεύει στη μείωση των αρνητικών επιπτώσεων των δραστηριοτήτων των επιχειρήσεων στο φυσικό περιβάλλον συμβάλλοντας έτσι στην βελτίωση της αποδοτικότητας των επιχειρήσεων. Ένα μοντέλο «πράσινης διαχείρισης» είναι η συλλογή και η εφαρμογή των εσωτερικών προσπαθειών κατά τον σχεδιασμό και τον προγραμματισμό της επιχείρησης. Το μοντέλο αυτό αποτελείται από την πολιτική και το σύνολο των διαδικασιών της επιχείρησης που απαιτούνται για την αξιολόγηση των περιβαλλοντικών της επιπτώσεων και καθορίζει τους περιβαλλοντικούς στόχους, εφαρμόζει τις περιβαλλοντικές λειτουργίες, συντονίζει την επίτευξη των στόχων και υφίσταται ανασκόπηση από τη διοίκηση. Έχει χρησιμοποιηθεί για την έγκριση του μοντέλου μια προσέγγιση πέντε βημάτων (Darnall and Edwards, 2006). Το πρώτο βήμα για την υιοθέτηση του μοντέλου είναι η εξασφάλιση της δέσμευσης για μια υπεύθυνη περιβαλλοντική διαχείριση. Με την υποστήριξη της ανώτερης διοίκησης, οι περιβαλλοντικές δεσμεύσεις ανανεώνονται κάθε φορά για τη συνεχή βελτίωση της πρόληψης της ρύπανσης και τη συμμόρφωση στη νομοθεσία του περιβάλλοντος (Starkey, 1998). Το δεύτερο βήμα είναι να αξιολογηθούν οι λειτουργίες των επιχειρήσεων και να καθοριστούν οι στόχοι. Κατά το στάδιο αυτό, τίθενται οι προτεραιότητες που έχουν οι επιχειρήσεις και λαμβάνονται αποφάσεις σχετικά με την πραγματοποίηση της περιβαλλοντικής πολιτικής σε δράση (Netherwood, 1998). Το τρίτο στάδιο του μοντέλου περιλαμβάνει, για την υλοποίηση των περιβαλλοντικών στόχων της επιχείρησης, τη δημιουργία μιας δομής διαχείρισης και την επικοινωνία με τους επιχειρηματικούς εταίρους. Στο τέταρτο βήμα, περιλαμβάνεται η παρακολούθηση και η λήψη διορθωτικών μέτρων, εφόσον αυτό είναι αναγκαίο, συμπερασματικά το στάδιο αυτό είναι ζωτικής σημασίας για τη συνεχή βελτίωση του περιβάλλοντος. Το τελικό στάδιο αποτελεί μια ανασκόπηση της διαχείρισης για την παροχή συστάσεων, κρίσιμων εκτιμήσεων και νέων περιβαλλοντικών ανησυχιών.

Τα κόστη διαρθρώνονται κατά τη διάρκεια των πέντε βημάτων. Επειδή οι επιχειρήσεις πρέπει να αναλάβουν εκτεταμένες εσωτερικές αξιολογήσεις, την κατάρτιση των εργαζομένων και την ανάπτυξη σχεδίων, απαιτούνται πόροι για την ρύθμιση αυτών των στόχων. Το μοντέλο της «πράσινης διαχείρισης» συντελεί στη διαβεβαίωση ότι οι πρακτικές διαχείρισής των επιχειρήσεων συμμορφώνονται στους περιβαλλοντικούς κανονισμούς και στον έλεγχο των εσωτερικών τους λειτουργιών με τη συμμετοχή των εργαζομένων στα περιβαλλοντικά θέματα, παρακολουθώντας τη βελτίωση του περιβάλλοντος και αυξάνοντας τις γνώσεις τους

σχετικά με τις δραστηριότητες τους. Όλες αυτές οι δράσεις μπορούν να βοηθήσουν τις επιχειρήσεις στην επίτευξη μεγαλύτερης αποδοτικότητας. Δεδομένου ότι αυτές οι δραστηριότητες αναπτύσσονται με τις γνώσεις βασισμένες στις δεξιότητες και είναι δύσκολο για τους ανταγωνιστές να τις μιμηθούν, το μοντέλο «πράσινης διαχείρισης» δημιουργεί ευκαιρίες απόκτησης ανταγωνιστικού πλεονεκτήματος (Hart, 1995). Η υιοθέτηση του μοντέλου ενθαρρύνει τις επιχειρήσεις να χρησιμοποιήσουν περιβαλλοντικές στρατηγικές που είναι περισσότερο εξελιγμένες και είναι βασισμένες στις δικές τους αρχές προστασίας του περιβάλλοντος. Επιπλέον, βοηθάει στην εξάλειψη λειτουργικών διαδικασιών επικίνδυνων για το περιβάλλον και στον επανασχεδιασμό υφιστάμενων συστημάτων λειτουργίας για τη μείωση των επιπτώσεων στον κύκλο ζωής (Hart, 1995).

### **1.3. Η περίπτωση του Hutchison Port Holdings (HPH)**

Στη μελέτη αυτή χρησιμοποιείται το HPH για την απεικόνιση της εφαρμογής του μοντέλου σε πραγματικές συνθήκες. Το Hutchison Port Holdings είναι παγκοσμίως ο μεγαλύτερος φορέας τερματικών σταθμών και απασχολεί συνολικά 300 προβλήτες σε 49 λιμάνια τα οποία εκτείνονται σε 25 χώρες σε ολόκληρο τον κόσμο. Η ιστορία του Hutchison Port Holdings ξεκίνησε το 1866. Το 1969, ιδρύθηκε η ναυαρχίδα λειτουργίας του HPH, το Hong Kong International Terminals (HIT). Βάση της λειτουργικής ικανότητας του HIT, το Hutchison Port Holdings έχει επεκταθεί σε ολόκληρο τον κόσμο και έχει γίνει ο κορυφαίος φορέας εκμετάλλευσης τερματικών σταθμών παγκοσμίως. Το Hong Kong International Terminals (HIT) αποτελεί τη ναυαρχίδα λειτουργίας για τον όμιλο HPHGroup, τον κορυφαίο επενδυτή και επιχειρηματία λιμανιού παγκοσμίως. Οι πρωτοποριακές τεχνικές διαχείρισης τερματικών και η τεχνολογία αιχμής έχουν μετατρέψει το HIT σε κέντρο αριστείας για τον όμιλο HPHGroup.

### **1.4. Οι φιλικές προς το περιβάλλον δραστηριότητες του HIT**

Το Hong Kong International Terminals (HIT) συμμετέχει ενεργά στη μείωση της ρύπανσης και προωθεί την υιοθέτηση του μοντέλου της «πράσινης διαχείρισης» σε ολόκληρη την επιχειρηματική του κοινότητα. Στις αρχές του 2007, ξεκίνησε ένα πρόγραμμα περιβαλλοντικής προστασίας (EPD) για να ανταποκριθεί στις δεσμεύσεις του βάσει ενός χάρτη «καθαρού αέρα» του Χονγκ Κονγκ και να μειώσει τις εκπομπές διοξειδίου του θείου το οποίο αποτελεί αιτία δημιουργίας όξινης βροχής και αιθαλομίχλης. Ο χάρτης, ο οποίος καταρτίστηκε σύμφωνα με διεθνώς αναγνωρισμένα πρωτόκολλα, δεσμεύει τις επιχειρήσεις ώστε να μειώσουν τη κατανάλωση ενέργειας των δραστηριοτήτων τους. Η ποσοτικοποίηση και η μείωση αυτής της κατανάλωσης συντελεί στον περιορισμό εκπομπών αερίων, που οφείλονται για το φαινόμενο του θερμοκηπίου. Για το λόγο αυτό, το HIT αναζητά τρόπους

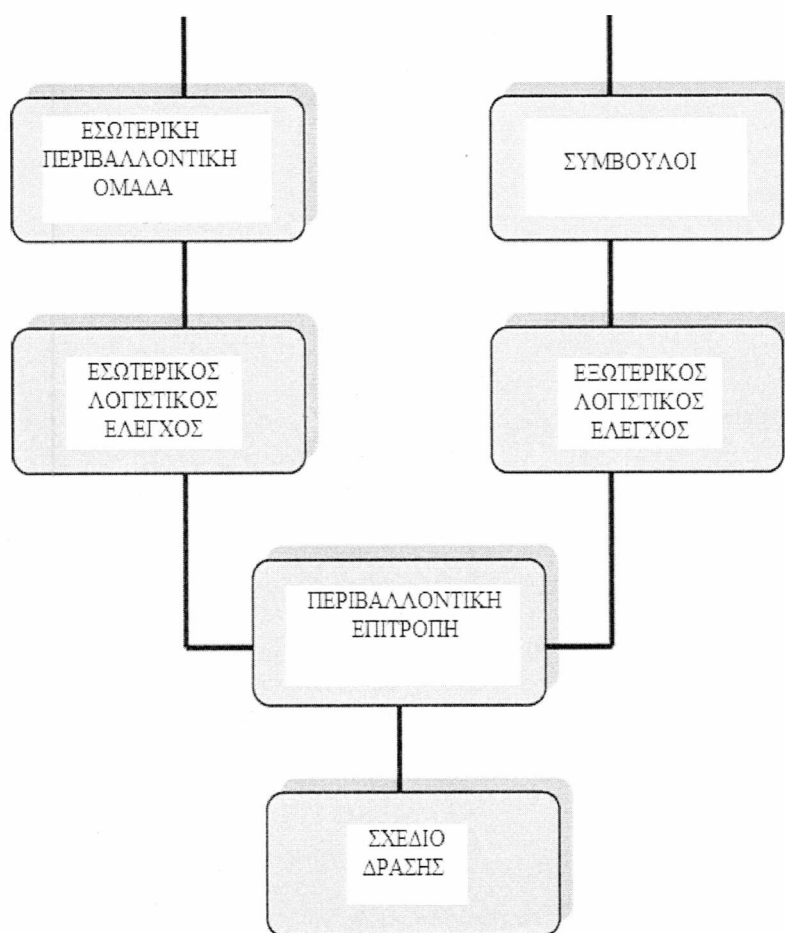
τόσο για τη μείωση των εκπομπών διοξειδίου του άνθρακα, όσο και για τη βελτίωση της ενεργειακής απόδοσης. Το HIT υποστηρίζει το πρόγραμμα περιβαλλοντικής προστασίας (EPD) επειδή συμβάλλει στην αειφόρο ανάπτυξη του Χονγκ Κονγκ σε συνδυασμό με τη μείωση των αρνητικών επιπτώσεων για το περιβάλλον. Για την εφαρμογή του προγράμματος, το HIT έχει τοποθετήσει στις κτιριακές του εγκαταστάσεις ειδικό φωτισμό για εξοικονόμηση ενέργειας και ηλιακούς θερμοσίφωνες. Επιπλέον, έχει ρυθμίσει την θερμοκρασία των γραφείων για να διατηρήσει ένα άνετο άλλα ενεργειακά αποδοτικό κλίμα εργασίας.

Ένα χαρακτηριστικό παράδειγμα της επίτευξης του HIT κατά την εφαρμογή του μοντέλου πράσινης διαχείρισης είναι η μείωση των εκπομπών του διοξειδίου του θείου κατά 90% το 2008. Αυτό έχει πραγματοποιηθεί με την εναλλαγή του βιομηχανικού πετρελαίου diesel, που χρησιμοποιούσαν οι ελαστικοφόρες γερανογέφυρες, από ένα πετρέλαιο diesel χαμηλής περιεκτικότητας σε θείο, το οποίο είναι ένα καθαρότερο καύσιμο με 1% περιεκτικότητα σε θείο. Το HIT, για την υιοθέτηση του μοντέλου, σχεδιάζει οχήματα με λειτουργίες φιλικές προς το περιβάλλον όπως είναι μπροστινοί φορτωτές, εσωτερικά φορτηγά και περνοφόρα οχήματα. Παρόλο που αυτές οι «πράσινες» πρωτοβουλίες δημιουργούν μια επιπλέον οικονομική επιβάρυνση λόγω της τροποποίησης και μετατροπής του εξοπλισμού, η επένδυση σε νέο εξοπλισμό μπορεί να επιφέρει λειτουργικές αποταμιεύσεις. Ένα χαρακτηριστικό παράδειγμα αποτελούν οι ελαστικοφόρες γερανογέφυρες, οι οποίες λειτουργούν με ηλεκτρική ενέργεια, γεγονός που επιφέρει μείωση του κόστους συντήρησης του κινητήρα περίπου 90% και ελάττωση στα κόστη καυσίμων 65%. Το Hutchison Port Holdings έχει αναπτύξει τις ηλεκτρονικές επικοινωνίες προκειμένου να επιτυγχάνεται η σύνδεση με τους πελάτες και τους συνεργάτες του.

Τα έγγραφα μεταφοράς των εμπορευματοκιβωτίων μετατρέπονται σε ηλεκτρονική μορφή για τη μείωση της χρήσης χαρτιού και για τη βελτίωση της αποδοτικότητας της επιχείρησης. Δεδομένα αποστολών όπως είναι η κράτηση στοιχείων εξαγωγών, οδηγίες για την αποθήκευση, πληροφορίες για τη φόρτωση και την εκφόρτωση των εμπορευματοκιβωτίων, η παράδοση των άδειων κιβωτίων και η συλλογή των δεδομένων, παραχωρούνται στους συντάκτες και τους πελάτες ηλεκτρονικά. Το σύστημα διορισμού φορτωτών είναι ένα σύστημα προγραμματισμού του HIT για τη συλλογή των εμπορευματοκιβωτίων από τα φορτηγά. Το σύστημα επιτρέπει τους συνεργάτες να επικοινωνήσουν με το HIT μέσω ήχου κλήσης τηλεφώνου ή μέσω διαδικτύου και ενισχύει τόσο την αποδοτικότητα του κόστους, όσο και την εξυπηρέτηση πελατών, διασφαλίζοντας την ταχεία ανάκαμψη. Επιπλέον, ευνοεί τη ροή της κυκλοφορίας μέσω του προγραμματισμού των αφίξεων προκειμένου να αποφεύγεται η κυκλοφοριακή συμφόρηση στην περιοχή του λιμανιού. Η επαλήθευση της ταυτότητας των πλοίων που ρυμουλκούνται πραγματοποιείται με τη χρήση «bar code». Αυτό μειώνει τη γραφική εργασία, ενισχύει την ασφάλεια του τερματικού σταθμού, επεκτείνει τις

διασυνδέσεις με τους φορείς αυτών των πλοίων και ωφελεί στη διοργάνωση των κινήσεων των πλοίων.

Το Hong Kong International Terminals (HIT) για την υιοθέτηση του μοντέλου «πράσινης διαχείρισης» έχει ξεκινήσει την ανακύκλωση υλικών όπως το χαρτί και το πολυστερένιο. Επίσης, επηρεάζει τους συνεργάτες του να ζητάνε από τους προμηθευτές τους να χρησιμοποιούν φιλικά προς το περιβάλλον υλικά και να είναι υπεύθυνοι απορρίπτοντας τις επικίνδυνες ουσίες. Για την εξάλειψη της ανάγκης που έχουν τα μέλη προσωπικού και οι πωλητές να κυκλοφορούν και να εκτυπώνουν τα διάφορα έγγραφα σε χαρτί αναπτύχθηκε μια πλατφόρμα ηλεκτρονικών προμηθειών από το τμήμα προμηθειών του HIT. Η χρήση της πλατφόρμας αυτής έχει μειώσει τη κατανάλωση περισσότερων από 250.000 φύλλων χαρτιού τον χρόνο. Η εικόνα 1 απεικονίζει το επίσημο οργανωτικό πλαίσιο του HIT (ποιες περιβαλλοντικές εργασίες προσδιορίζονται και ελέγχονται).



Εικόνα 1. Περιβαλλοντική οργανωτική δομή του HIT. Πηγή: Y.H.V. Lun (2011)

### 1.5. Από την κατανάλωση καυσίμου στις εκπομπές των πλοίων

Σε αυτό το τμήμα θα ζητηθεί ο υπολογισμός των εκπομπών των πλοίων με βάση τα καύσιμα που καταναλώνονται. Όπως έχει αναφερθεί και προηγουμένως, οι θαλάσσιες μεταφορές έχουν συμβάλει σημαντικά στην οικονομική ανάπτυξη του κόσμου. Ταυτόχρονα, τα πλοία εκπέμπουν όλο και περισσότερους ρύπους όπως είναι το διοξείδιο του άνθρακα  $CO_2$ , οξείδια του θείου  $SO_x$ , οξείδια του αζώτου  $NO_x$  και αιωρούμενα σωματίδια PM στο περιβάλλον. Στο συνολικό όγκο εκπομπών το μερίδιο των θαλάσσιων εκπομπών σίγουρα θα αυξηθεί, επειδή αυστηρά πρότυπα εκπομπών και προδιαγραφές καυσίμων επιβάλλονται στους υπόλοιπους τρόπους μεταφοράς (οδικές, σιδηροδρομικές μεταφορές, κλπ.) και η μεγάλη διάρκεια ζωής των πλοίων επιβραδύνει τον κύκλο εργασιών των τεχνολογιών τους. Επιπλέον, οι θαλάσσιες εκπομπές έχουν αγνοηθεί σε μεγάλο βαθμό από εθνικά προγράμματα ποιότητας του αέρα από τότε που αυτές παράγονται σε διεθνή ύδατα. Στη μεθοδολογία υπολογισμού της Διακυβερνητικής Επιτροπής για την αλλαγή του κλίματος (IPCC), οι εκπομπές των πλοίων δεν έχουν αποδοθεί σε χώρες λόγω της διαφοράς του τόπου όπου τα καύσιμα αγοράζονται και του τόπου όπου οι εκπομπές παράγονται (IPCC, 1997).

Με την αρχή του πρωτοκόλλου του Κιότο, ο Διεθνής Ναυτιλιακός Οργανισμός (IMO) έχει κάνει μια μεγάλη προσπάθεια για τη μείωση των εκπομπών αυτών. Το Παράρτημα VI της MARPOL υιοθετήθηκε από τον Οργανισμό το 1997 και τέθηκε σε ισχύ στις 19 Μαΐου του 2005. Το Παράρτημα αυτό είναι η κύρια ρύθμιση για την πρόληψη της ατμοσφαιρικής ρύπανσης από τα πλοία και οι τελευταίες σημαντικές τροποποιήσεις του εγκρίθηκαν στην 58<sup>η</sup> σύνοδο (6-10 Οκτωβρίου, 2008) της Επιτροπής Προστασίας Θαλάσσιου Περιβάλλοντος (MPEC). Το αναθεωρημένο Παράρτημα καλύπτει τα θέματα από τους κανονισμούς σχετικά με τις προδιαγραφές των καυσίμων και τις τεχνολογίες κινητήρων στους μηχανισμούς της αγοράς (IMO, 2008b). Επιπλέον, ο Οργανισμός ενημερώνεται για τη σημασία της μείωσης των εκπομπών μέσω των διευθυντικών μέτρων στο επιχειρησιακό επίπεδο, όπως είναι ο έλεγχος της ταχύτητας, η διαχείριση του στόλου και η βελτίωση στις διαδικασίες διακίνησης των φορτίων (IMO, 2008a).

Η απουσία μεθοδολογιών αναφοράς και προγραμμάτων παρακολούθησης καθιστά δύσκολο τον ακριβή υπολογισμό των εκπομπών των πλοίων. Γενικότερα, οι εκπομπές υπολογίζονται κατά προσέγγιση με την κατανάλωση καυσίμου και με παράγοντες των εκπομπών:

$$\text{vessel emissions (g)} = \text{fuel consumption (kg)} \times \text{emission factor (g/kg-fuel)}.$$

Λόγω της απουσίας εξοπλισμού μέτρησης των συντελεστών εκπομπής καυσίμου, οι παράγοντες αυτοί επιλέγονται πάντα με τη γενική έννοια. Η κατανάλωση καυσίμου για την πλεύση είναι κυρίαρχη σε σύγκριση με εκείνη για τις περιόδους ελλιμενισμού. Οι εκπομπές  $CO_2$  κατά την πλεύση αντιπροσωπεύουν το 74% του συνολικού όγκου εκπομπών διοξειδίου



του άνθρακα για τα πλοία, τα οξείδια του αζώτου και του θείου περίπου το 79% (Schrooten et al., 2008).

### **1.6. Σύστημα αποφάσεων για τις εργασίες σε έναν τερματικό σταθμό εμπορευματοκιβωτίων**

Σε ένα λιμάνι ένας τερματικός σταθμός εμπορευματοκιβωτίων είναι το μέρος όπου τα πλοία που μεταφέρουν τα εμπορευματοκιβώτια αγκυροβολούν σε προβλήτες, ξεφορτώνουν τα εισερχόμενα κιβώτια (γεμάτα με φορτίο ή άδεια) και φορτώνουν τα εξαγόμενα. Για την προσωρινή τοποθέτηση των εμπορευματοκιβωτίων, οι τερματικοί σταθμοί διαθέτουν αυλές αποθήκευσης (εικόνα 2).

Το μεγαλύτερο μέρος του φορτίου το οποίο μεταφέρεται από τα πλοία παγκοσμίως στη σημερινή εποχή μπορεί να ταξινομηθεί σε δύο τύπους:

- Τεράστιες ποσότητες εμπορευμάτων ,όπως είναι το αργό πετρέλαιο, ο άνθρακας, μεταλλεύματα, κόκκοι, κλπ, που έχουν αποσταλεί με εξειδικευμένα πλοία μεταφοράς τα οποία ονομάζονται πλοία μεταφοράς φορτίου χύδην;
- Ποικιλία προϊόντων τα οποία είναι συσκευασμένα σε κανονικού μεγέθους εμπορευματοκιβώτια από χάλυβα και αποστέλλονται με πλοία.



**Εικόνα 2.** Η αυλή αποθήκευσης σε έναν τερματικό σταθμό. Πηγή: K.G. Murty et al.(2005)

### 1.7. Εμπορευματοκιβώτια, μέρη αποθήκευσης, γερανοί ναυπηγείου

Τα εμπορευματοκιβώτια είναι κουτιά από χάλυβα με διαστάσεις (όλες οι μετρήσεις είναι σε πόδια)  $20 \times 8 \times 8.5$  ή  $20 \times 8 \times 9.5$  (ονομάζονται «20-ft containers»),  $40 \times 8 \times 8.5$  και  $40 \times 8 \times 9.5$  (ονομάζονται «40-ft containers»), ή εξειδικευμένα ελαφρώς μεγαλύτερου μεγέθους κουτιά. Για τη μέτρηση της απόδοσης ενός τερματικού, της χωρητικότητας του πλοίου, κλπ, χρησιμοποιείται συνήθως μια ενιαία μονάδα, TEU (ισοδύναμη μονάδα με 20 πόδια), με κάθε εμπορευματοκιβώτιο που είναι 40 πόδια ή μεγαλύτερο να υπολογίζεται ως δύο TEUs.

Η αυλή αποθήκευσης σε ένα τερματικό σταθμό χωρίζεται σε ορθογώνιες περιοχές που ονομάζονται μπλοκ αποθήκευσης. Ένα τυπικό μπλοκ έχει χώρους με επτά σειρές ή λωρίδες, έξι από τις οποίες χρησιμοποιούνται για την αποθήκευση των εμπορευματοκιβωτίων και η έβδομη προορίζεται για τη διέλευση των φορτηγών. Κάθε σειρά αποτελείται συνήθως από στοίβες εμπορευματοκιβωτίων που είναι 20 πόδια και πάνω, τα οποία τοποθετούνται κατά μήκος και το ένα πάνω στο άλλο, ενώ για την αποθήκευση των εμπορευματοκιβωτίων που είναι 40 πόδια, χρησιμοποιούνται χώροι για δύο στοίβες 20 ποδιών.



Εικόνα 3. Μια ελαστικοφόρα γερανογέφυρα (RTGC). Πηγή: K.G. Murty et al.(2005)

Η τοποθέτηση ή η ανάκτηση των εμπορευματοκιβωτίων από τις στοίβες πραγματοποιείται με γερανούς που ονομάζονται γερανοί ναυπηγείου. Οι ελαστικοφόρες γερανογέφυρες (RTGCs) είναι οι πιο συνηθισμένες και είναι γερανογέφυρες οι οποίες στέκονται σε δύο σειρές ελαστικών, κινούνται με αυτά και εκτείνονται μεταξύ των ελαστικών στις επτά λωρίδες των μπλοκ αποθήκευσης (εικόνα 3). Η γέφυρα διαθέτει μια σπάτουλα που λαμβάνει το κιβώτιο και μπορεί να μετακινηθεί στις επτά σειρές σε όλο το πλάτος του μπλοκ αλλά και σε όλο το μήκος του με τη βοήθεια των ελαστικών. Οι γερανογέφυρες αυτές μπορούν να σηκώσουν ή να τοποθετήσουν τα κουτιά στις στοίβες ή πάνω στα φορτηγά που διέρχονται από την έβδομη σειρά.

Το ύψος των γερανών (RTGCs) καθορίζεται από τον αριθμό των κουτιών που μπορούν να τοποθετούνται κατακόρυφα σε μια στοίβα. Υπάρχουν γερανογέφυρες με ύψος πέντε επιπέδων όπου σε μία στοίβα τοποθετούνται τέσσερα εμπορευματοκιβώτια και το 5<sup>ο</sup> επίπεδο χρησιμοποιείται για τη μεταφορά των εμπορευματοκιβωτίων κατά πλάτος του μπλοκ και οι καινούριες γερανογέφυρες με ύψος έξι επιπέδων όπου το 6<sup>ο</sup> επίπεδο χρησιμοποιείται για τη μεταφορά και τα άλλα για την αποθήκευση πέντε κιβωτίων.

Μερικά μπλοκ αποθήκευσης εξυπηρετούνται από γερανογέφυρες με σιδηροτροχιές (RMGCs), οι οποίες είναι γερανογέφυρες με 13 σειρές μεταξύ των ποδιών τους και μεγαλύτερο ύψος αποθήκευσης (έξι επίπεδα). Οι ελαστικοφόρες γερανογέφυρες προσφέρουν μεγαλύτερη ευελιξία σε σχέση με αυτές, διότι μπορούν και μεταφέρονται από μπλοκ σε μπλοκ και για το λόγο αυτό η χρήση των πρώτων είναι περισσότερο συνηθισμένη στις αυλές αποθήκευσης για τη διακίνηση των εμπορευματοκιβωτίων.

### **1.8. Εισερχόμενα - εξερχόμενα εμπορευματοκιβώτια και εξοπλισμός που χρησιμοποιείται σε έναν τερματικό σταθμό**



**Εικόνα 4. Εξυπηρέτηση ενός πλοίου από τρεις γερανούς αποβάθρας. Πηγή: K.G. Murty et al.(2005)**

Ένα εξερχόμενο εμπορευματοκιβώτιο αποστέλλεται από τον πελάτη ενός τερματικού σταθμού διαμέσου του λιμανιού σε ένα άλλο λιμάνι οπουδήποτε στον κόσμο και ένα εισερχόμενο κιβώτιο είναι αυτό που μεταφέρεται με πλοίο από ένα λιμάνι για να εκφορτωθεί σε ένα άλλο και να αποθηκευτεί προσωρινά μέχρι ο πελάτης, για τον οποίο προορίζεται, να το πάρει. Οι πελάτες με δικά τους φορτηγά, τα οποία ονομάζονται εξωτερικά, φέρνουν στους τερματικούς σταθμούς τα εξερχόμενα εμπορευματοκιβώτια και παραλαμβάνουν τα εισερχόμενα. Μέσα στο τερματικό σταθμό υπάρχουν φορτηγά για τη μετακίνηση των εμπορευματοκιβωτίων, τα λεγόμενα εσωτερικά. Η φόρτωση και η εκφόρτωση των

εμπορευματοκιβωτίων από τα πλοία πραγματοποιείται με τη βοήθεια μεγάλων γερανών (εικόνα 4) που ονομάζονται γερανοί αποβάθρας (QCs).

## **1.9. Εργασίες στις αυλές αποθήκευσης**

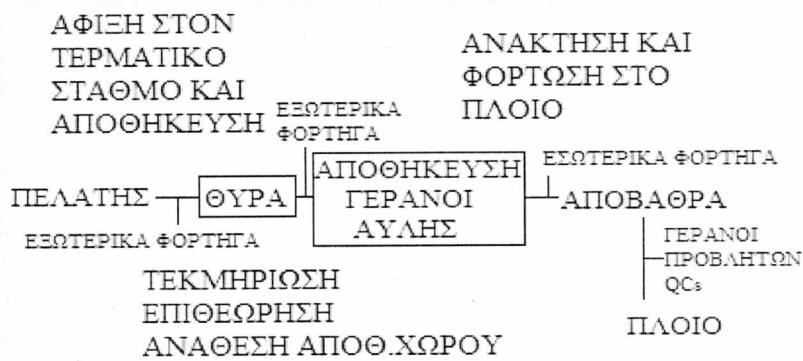
### **1.9.1. Οι λειτουργίες ενός τερματικού σταθμού εμπορευματοκιβωτίων**

Οι κύριες λειτουργίες ενός τερματικού σταθμού είναι οι εξής:

- Να λαμβάνονται τα εξερχόμενα εμπορευματοκιβώτια από τους φορτωτές και να τοποθετούνται στα πλοία ή να ξεφορτώνονται τα εισερχόμενα εμπορευματοκιβώτια από τα πλοία προκειμένου αυτά να τα παραλάβουν οι πελάτες;
- Η προσωρινή αποθήκευση των εμπορευματοκιβωτίων.

### **1.9.2. Οι ροές των εξερχόμενων και εισερχόμενων κιβωτίων**

Τα εξωτερικά φορτηγά (XTs) τα οποία ανήκουν στους πελάτες μεταφέρουν τα εξερχόμενα εμπορευματοκιβώτια και εισέρχονται στο τερματικό σταθμό μέσω μιας θύρας (Terminal Gatehouse) όπου κάθε κιβώτιο και η τεκμηρίωση του ελέγχονται. Στη συνέχεια, το εξωτερικό φορτηγό προχωράει με εντολή της θύρας στο μπλοκ αποθήκευσης, όπου το εμπορευματοκιβώτιο θα παραμείνει μέχρι να φτάσει το πλοίο στο οποίο θα φορτωθεί. Το εμπορευματοκιβώτιο μετακινείται από το φορτηγό στη θέση αποθήκευσης μέσω ελαστικοφόρας γερανογέφυρας (RTGC). Η γερανογέφυρα, όταν έρθει η στιγμή της φόρτωσης, μετακινεί το κιβώτιο και το τοποθετεί στο εσωτερικό φορτηγό (IT), το οποίο με τη σειρά του μεταφέρει το εμπορευματοκιβώτιο στους γεραμούς αποβάθρας(QCs) για τη φόρτωση του στο πλοίο. Η ροή των εξερχόμενων εμπορευματοκιβωτίων παρουσιάζεται στην εικόνα 5 και τα εισερχόμενα κιβώτια έχουν την αντίθετη ροή.



Εικόνα 5. Η ροή των εξερχόμενων εμπορευματοκιβωτίων. Πηγή: K.G. Murty et al.(2005)

### 1.9.3. Οργάνωση πλοίου που μεταφέρει εμπορευματοκιβώτια

Σε ένα πλοίο, το οποίο προορίζεται για τη μεταφορά εμπορευματοκιβωτίων, τα κιβώτια είναι στοιβαγμένα κατά μήκος. Οι καταπακτές είναι τα τμήματα στα οποία διαιρείται το πλοίο σε όλο του το μήκος. Κάθε ένα από αυτά τα τμήματα μπορεί να φιλοξενήσει δύο εμπορευματοκιβώτια των 20 ποδιών ή ένα των 40 ποδιών και στην περίπτωση ενός μεγάλου πλοίου, στις καταπακτές μπορούν να είναι στοιβαγμένα σε μια σειρά κατά το πλάτος του καραβιού έως και 20 κιβώτια. Ανάλογα με το συνολικό μήκος του πλοίου, τα τμήματα μπορούν να είναι παραπάνω από 20 σε αριθμό. Για τη γρήγορη φόρτωση και εκφόρτωση, συνήθως τα εμπορευματοκιβώτια ανατίθενται σε καταπακτές των λιμένων στα οποία έχει γίνει κλήση από το πλοίο κατά το ταξίδι (5 με 10 λιμάνια συνήθως καλούνται). Όταν το πλοίο φτάσει και αγκυροβολήσει στην προβλήτα, για τη μεταφορά των εμπορευματοκιβωτίων μεγάλος αριθμός γερανών (QCs) εργάζονται ταυτόχρονα και σε διαφορετικό τμήμα. Την ίδια χρονική στιγμή, δεν γίνεται σε διπλάνες καταπακτές να δουλεύουν γερανοί διαφορετικού είδους και κάθε γερανός είναι ευρύτερος από το τμήμα που εξυπηρετεί.

Ένας γερανός διαθέτει τέσσερα πόδια. Αυτά είναι διατεταγμένα σε δύο σειρές των δύο ποδιών η κάθε μια και ο χώρος που υπάρχει μεταξύ των δύο σειρών διαιρείται σε πέντε λωρίδες για φορτηγά. Ένα σύνολο από εσωτερικά φορτηγά (ITs) είτε αφαιρούν τα κιβώτια που εκφορτώνονται από τους γερανούς είτε μεταφέρουν τα εμπορευματοκιβώτια στους γερανούς για να φορτωθούν στα πλοία. Το φορτηγό που εξυπηρετεί έναν γερανό, πάντα παρατάσσεται σε μία από τις πέντε λωρίδες οι οποίες προορίζονται αποκλειστικά για τα εσωτερικά φορτηγά που εξυπηρετούν γερανούς αποβάθρας. Σε ένα πλοίο, τρεις ή τέσσερις

γερανοί λειτουργούν ταυτόχρονα και αυτό συνήθως καθορίζεται από τον αριθμό των λωρίδων.

#### **1.9.4. Γερανογέφυρες ελαστικοφόρες (RTGCs) και οι λειτουργίες τους**

Οι ελαστικοφόρες γερανογέφυρες (RTGCs) αποτελούν ακριβό εξοπλισμό και η σωστή χρησιμοποίησή τους είναι κρίσιμη για την αποτελεσματικότητα των διαδικασιών χειρισμού των εμπορευματοκιβωτίων στις αυλές αποθήκευσης. Οι γερανογέφυρες αυτές συντελούν στην ανάκτηση των εμπορευματοκιβωτίων που είναι αποθηκευμένα, την τοποθέτησή τους στα εξωτερικά/εσωτερικά φορτηγά ή μεταφέρουν τα κιβώτια από τα φορτηγά στους χώρους αποθήκευσης. Τα φορτηγά (εσωτερικά και εξωτερικά) μπαίνουν σε ουρά στη λωρίδα που προορίζεται για το πέρασμα τους όταν φτάνουν στο μπλοκ μέχρι η γερανογέφυρα να τα εξυπηρετήσει. Για το λόγο αυτό, εάν οι γερανογέφυρες δεν εκτελούν αποτελεσματικά το έργο τους, υπάρχει περίπτωση να δημιουργηθεί κυκλοφοριακή συμφόρηση μέσα στο ίδιο το μπλοκ αλλά και κοντά στον δρόμο του. Επιπλέον, όταν οι γερανογέφυρες εξυπηρετούν αργά τα εσωτερικά φορτηγά, οι γερανοί αποβάθρας αναγκάζονται να τα περιμένουν με αποτέλεσμα την καθυστέρηση στην φόρτωση/εκφόρτωση του πλοίου.

Υπάρχουν δύο ειδών διαδικασίες ανάκτησης των εμπορευματοκιβωτίων στις γερανογέφυρες (RTGCs):

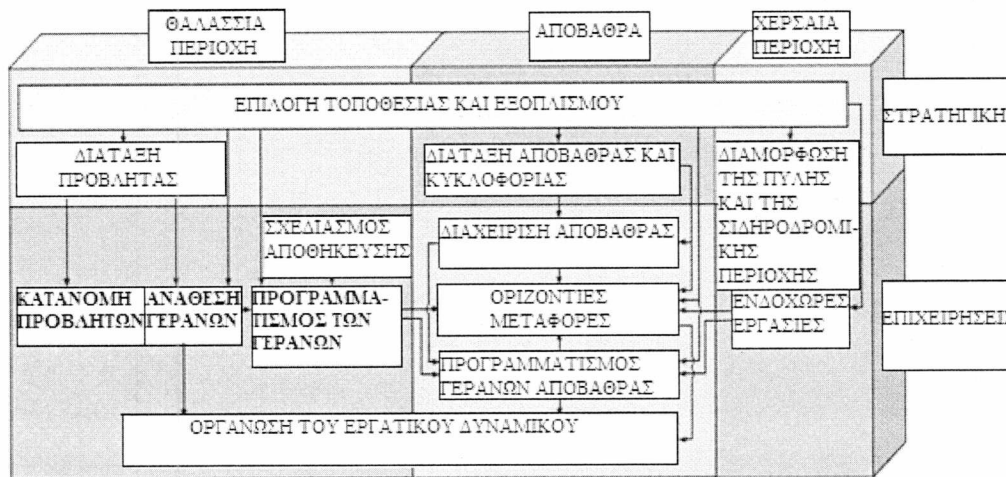
- *Η παραγωγική μεταφορά:* όταν το εμπορευματοκιβώτιο μεταφέρεται άμεσα από τον χώρο που είναι αποθηκευμένο σε ένα εξωτερικό ή εσωτερικό φορτηγό που περιμένει για την παραλαβή, αυτή είναι μια παραγωγική κίνηση;
- *Η μη παραγωγική μεταφορά:* όταν ένα εμπορευματοκιβώτιο είναι αναγκαίο να μετακινηθεί για την ανάκτηση ενός αλλού το οποίο είναι αποθηκευμένο κάτω από αυτό στην ίδια στοίβα, αυτή είναι μια μη παραγωγική κίνηση. Η μετακίνηση των εμπορευματοκιβωτίων για τη συλλογή του σωστού πραγματοποιείται ξεχωριστά για το καθένα.

Ο αριθμός αυτών των κινήσεων ανασχηματισμού εξαρτάται από τη στρατηγική που χρησιμοποιείται για τη κατανομή των θέσεων αποθήκευσης. Αν δύο μπλοκ αποθήκευσης βρίσκονται δίπλα κατά πλάτος, τότε η γερανογέφυρα (RTGC) μπορεί να μετακινηθεί μεταξύ τους χωρίς καμία περιστροφική κίνηση. Για τη μετακίνηση κατά μήκος δύο διπλανών μπλοκ η γερανογέφυρα πρέπει να έρθει στον δρόμο στο ένα άκρο του μπλοκ, να κάνει στροφή 90 μοιρών με τη βοήθεια των τροχών της, να μετακινηθεί στο δρόμο που είναι παράλληλος στη γραμμή πλάτους στη σωστή θέση για το διπλανό μπλοκ, να κάνει άλλη μια στροφή 90 μοιρών και να εισέλθει στο μπλοκ. Οι στροφές των 90 μοιρών λαμβάνουν επιπλέον χρόνο και εμποδίζεται η κυκλοφορία όταν η γερανογέφυρα βρίσκεται πάνω στον δρόμο.

### 1.10. Αποφάσεις για τις καθημερινές λειτουργίες

- Κατανομή των θέσεων ελλιμενισμού των αφικνούμενων πλοίων στις προβλήτες με σκοπό την ελαχιστοποίηση του χρόνου αναμονής των καραβιών, του κόστους λιμανιού, τη δυσαρέσκεια για τη σειρά προτεραιότητας της αγκυροβόλησης αλλά και τη μεγιστοποίηση της αξιοποίησης των γερανών και των προβλητών;
- Η ανάπτυξη ενός σχεδίου για την κατανομή και τον προγραμματισμό του χρονοδιαγράμματος για τις λειτουργίες των γερανών στα πλοία που έχουν προσαράξει. Αυτό επηρεάζει τον χρόνο ολοκλήρωσης των πλοίων και το ποσοστό απόδοσης του τερματικού σταθμού;
- Στην ιδανική περίπτωση, όλοι οι πελάτες, καλώντας νωρίτερα, κάνουν κρατήσεις για το χρόνο όπου θα έρθουν να παραλάβουν τα εισερχόμενα εμπορευματοκιβώτια που τους ανήκουν ή θα μεταφέρουν τα εξερχόμενα κιβώτια. Έτσι, αναπτύσσονται απλοί κανόνες αποφάσεων «online» για τη δημιουργία ραντεβού, για την ελαχιστοποίηση του χρόνου αναμονής των εξωτερικών φορτηγών και της κυκλοφοριακής συμφόρησης στο οδικό δίκτυο;
- Η δρομολόγηση των εσωτερικών και εξωτερικών φορτηγών στον τερματικό σταθμό αποσκοπεί στην ελαχιστοποίηση της συμφόρησης στους δρόμους;
- Η ανάπτυξη μιας «online» διαδικασίας για την αποστολή των εξωτερικών φορτηγών που φτάνουν στη θύρα του τερματικού σταθμού (Terminal Gatehouse) με τα εξερχόμενα εμπορευματοκιβώτια και των εσωτερικών φορτηγών που μεταφέρουν τα εισερχόμενα κιβώτια, με σκοπό την ελαχιστοποίηση της συμφόρησης στα μπλοκ και στους δρόμους;
- Εκχώρηση αποθηκευτικών χώρων στα εμπορευματοκιβώτια που φτάνουν για την ελαχιστοποίηση του ανασχηματισμού του όγκου και της συμφόρησης στους δρόμους;
- Ο καθορισμός του αριθμού των γερανογεφυρών (RTGCs) που πρέπει να δουλεύουν σε κάθε μπλοκ και της στιγμής που πρέπει να γίνεται η μετακίνησή τους από το ένα τετράγωνο στο άλλο, επηρεάζει τον χρόνο που τα πλοία μένουν στο λιμάνι και τους χρόνους αναμονής των γερανών και των εξωτερικών/εσωτερικών φορτηγών;
- Ο καθορισμός του αριθμού των εσωτερικών φορτηγών τα οποία ανατίθενται σε κάθε γερανό που βρίσκεται σε λειτουργία, ελαχιστοποιεί τον χρόνο για την ολοκλήρωση ενός πλοίου και τους χρόνους αναμονής των εσωτερικών φορτηγών και των γερανών;
- Η πρόσληψη εσωτερικών φορτηγών τα οποία ανταποκρίνονται στις απαιτήσεις που προκύπτουν σε κάθε διάστημα ίσων ωρών (κάθε 12ωρο) την ημέρα, ελαχιστοποιεί

τον συνολικό αριθμό των μισθωμένων φορητών κάθε μέρα και μεγιστοποιεί την αξιοποίησή τους.



Εικόνα 6. Προβλήματα προγραμματισμού σε έναν τερματικό σταθμό εμπορευματοκιβωτίων.

Πηγή: C. Bierwirth, F. Meisel (2010)

Για κάθε εμπορευματοκιβώτιο πρέπει να λαμβάνονται διαφορετικές αποφάσεις καθώς το καθένα περνάει από διαφορετικά στάδια σε έναν τερματικό σταθμό. Λίγες εβδομάδες πριν την πραγματική άφιξη του πλοίου ο τερματικός σταθμός λαμβάνει πληροφορίες για αυτό. Με δεδομένο το φόρτος εργασίας του πλοίου, ο σταθμός πραγματοποιεί προσωρινές κατανομές της προβλήτας και των γερανών που απαιτούνται. Τα εξωτερικά φορητά, έχοντας κλείσει ραντεβού, αρχίζουν τη μεταφορά των εξερχόμενων εμπορευματοκιβωτίων για το πλοίο λίγες μέρες πριν τη πραγματική του άφιξη. Ανάλογα με τη κατάσταση που επικρατεί στην αυλή του σταθμού κατά την άφιξη των κιβωτίων, αυτά αποστέλλονται στους χώρους αποθήκευσης. Τα εμπορευματοκιβώτια κατανέμονται στο μπλοκ και μετά στοιβάζονται με στόχο την ελάττωση του αριθμού των ανασχηματισμών που θα χρειαστεί να γίνουν.

Το πραγματικό φορτίο εργασίας και ο πιθανός χρόνος άφιξης του πλοίου γίνονται γνωστά μια ημέρα πριν την πραγματική του άφιξη. Με αυτές τις πληροφορίες καθορίζονται η θέση ελλιμενισμού του πλοίου στη προβλήτα αλλά και οι γερανοί που θα χρησιμοποιηθούν και με τις ενημερωμένες απαιτήσεις των εργασιών, οι οποίες πρέπει να πραγματοποιηθούν, ο σταθμός προσδιορίζει τον αριθμό των εσωτερικών φορητών που πρέπει να προσληφθούν την ακόλουθη ημέρα και διαμορφώνει ένα σχέδιο ανάθεσης των φορητών στους γεραμούς. Κατά τη διάρκεια των πραγματικών εργασιών, ο σταθμός παίρνει αποφάσεις πραγματικού χρόνου για την κατανομή των εσωτερικών φορητών στους γεραμούς αποβάθρας, τη κίνηση των γερανών μεταξύ των μπλοκ και τη δρομολόγηση των εσωτερικών φορητών μεταξύ των



γερανών και των μπλοκ. Όλες οι άλλες αποφάσεις που είναι προγραμματισμένες και αναφέρονται παραπάνω υπόκεινται σε αλλαγές, σύμφωνα με τον πραγματικό χρόνο των δραστηριοτήτων που πραγματοποιούνται στον τερματικό σταθμό.

### **1.11. Το λιμάνι του Χονγκ Κονγκ**

Το Χονγκ Κονγκ αποτελεί το πιο πολυσύχναστο λιμάνι εμπορευματοκιβωτίων στον κόσμο. Η ναυτιλία εμπορευματοκιβωτίων είναι η ψυχή της οικονομίας του Χονγκ Κονγκ. Το Χονγκ Κονγκ είναι το κύριο λιμάνι εισόδου στην περιοχή της νότιας Κίνας. Ο όγκος των κιβωτίων που μεταφέρονται μέσω του Χονγκ Κονγκ αυξάνεται ετησίως 10% από το 1986. Η διακίνηση μέσα στο 2000 ανήλθε στα 18,1 εκατομμύρια TEUs. Η ένταση της κίνησης των εμπορευματοκιβωτίων εκτιμάται να είναι επτά φορές εκείνης που επικρατεί στη Νέα Υόρκη, ενώ η στενότητα του χώρου στις αυλές αποθήκευσης του Χονγκ Κονγκ καθιστά πολύ δύσκολη τη διατήρηση υψηλής ποιότητας εξυπηρέτησης.

Το λιμάνι εμπορευματοκιβωτίων στο Χονγκ Κονγκ αυτή τη στιγμή τρέχει από τέσσερις φορείς εκμετάλλευσης τερματικών σταθμών που είναι οι εξής: Hong Kong International Terminals Limited, Modern Terminals Limited, CSX World Terminals Hong Kong Limited και COSCOHIT Terminals (Hong Kong) Limited. Οι καθημερινές δραστηριότητες αυτών των εταιριών είναι αρκετά πολύπλοκες. Υπάρχουν περίπου 65 γερανοί, 200 ελαστικοφόρες γερανογέφυρες (RTGCs), 30 γερανογέφυρες με σιδηροτροχιές (RMGCs) και 300 εσωτερικά φορητά (ITs) συγκεντρωμένα στις αυλές αποθήκευσης με συνολική (όλους τους τερματικούς σταθμούς) έκταση των 220 ha. Οι αυλές αποθήκευσης είναι χωρισμένες σε τμήματα τα λεγόμενα μπλοκ, με ικανότητα αποθήκευσης περίπου 600 TEUs το καθένα αν εξυπηρετείται από μια ελαστικοφόρα γερανογέφυρα (RTGC) και 1300 TEUs αν εξυπηρετείται από γερανογέφυρα με σιδηροτροχιές (RMGC). Η πυκνότητα του στοιβάγματος είναι περίπου 720 TEUs ανά εκτάριο, μεταξύ των υψηλότερων στον κόσμο. Το Χονγκ Κονγκ διαθέτει 18 προβλήτες που αποτελούν μαζί μια αποβάθρα μήκους 6 χιλιομέτρων. Όλοι οι τερματικοί σταθμοί στο Χονγκ Κονγκ επεξεργάζονται από κοινού 35 πλοία κατά μέσο όρο την ημέρα, με μια ροή εμπορευματοκιβωτίων των 32.000 TEUs μεταξύ των προβλητών και των χώρων αποθήκευσης. Υπάρχουν γύρω στα 30.000 εξωτερικά φορητά (XTs) τα οποία μεταφέρουν τα εξαγόμενα κιβώτια ή συλλέγουν τα εισαγόμενα από αυτούς τους σταθμούς κάθε μέρα.

Από τις ακόλουθες συγκρίσεις μπορεί να φανεί η ανώτερη απόδοση των τερματικών σταθμών του Χονγκ Κονγκ. Το 2000, 6,6 εκατομμύρια TEUs διακινήθηκαν από δύο τερματικούς του Χονγκ Κονγκ, τους Hong Kong International Terminals (HIT) και COSCO-HIT, χρησιμοποιώντας 122 εκτάρια γης. Το 2001, ο πιο πολυσύχναστος τερματικός σταθμός στην Ευρώπη ο οποίος είναι ο Delta Terminal και βρίσκεται στην Ολλανδία, διακίνησε 2,5 TEUs,

χρησιμοποιώντας 280 εκτάρια γης καθώς και το λιμάνι του Long Beach, ένα από τα πιο πολυσύχναστα λιμάνια εμπορευματοκιβωτίων στις Ηνωμένες Πολιτείες, διακίνησε 4.6 εκατομμύρια TEUs, χρησιμοποιώντας 295 εκτάρια γης. Κατά μέσο όρο ένας γερανός αποβάθρας (QC) στο Χονγκ Κονγκ μπορεί να χειριστεί πάνω από 30 εμπορευματοκιβώτια την ώρα. Επιπλέον, οι τερματικοί σταθμοί του Χονγκ Κονγκ, προκειμένου να διατηρήσουν τα χρονικά περιθώρια παραμονής των πλοίων στις προβλήτες, είναι πρόθυμοι να επενδύσουν στην απόκτηση ενός επαρκούς αριθμού γερανών (QCs). Για το λόγο αυτό, οι τερματικοί σταθμοί του Χονγκ Κονγκ θεωρούνται πλέον οι πιο αποδοτικοί στον κόσμο.

### **1.12. Η εισαγωγή δεδομένων και η περίοδος προγραμματισμού για τη λήψη αποφάσεων**

Οι τερματικοί σταθμοί εμπορευματοκιβωτίων λειτουργούν τρεις βάρδιες των 8 ωρών κάθε μέρα. Υπάρχουν δύο σημαντικά ζητήματα για την επιλογή και την οργάνωση της κατάλληλης περιόδου προγραμματισμού. Ένα από αυτά αποτελεί η δυνατή εκτίμηση για το συνολικό φόρτο εργασίας κατά την περίοδο με λογική ακρίβεια. Το δεύτερο ζήτημα είναι η ομοιόμορφη κατανομή του συνολικού φόρτου εργασίας με τη πάροδο του χρόνου κατά τη διάρκεια της περιόδου. Οι διαχειριστές έχουν διαπιστώσει ότι μια περίοδος των τεσσάρων ωρών είναι αποτελεσματικότερη στην επαρκή ικανοποίηση των δύο παραπάνω παραγόντων. Με μικρότερες χρονικές περιόδους, η ακρίβεια της εκτίμησης του συνολικού φόρτου εργασίας δεν είναι ικανοποιητική, ενώ με μεγαλύτερες χρονικές περιόδους, η κατανομή του φόρτου εργασίας με την πάροδο του χρόνου τείνει να γίνει άνιση. Οι τερματικοί σταθμοί οργανώνουν τις εργασίες με περίοδο προγραμματισμού 4 ωρών. Μια μέρα χωρίζεται σε 6 περιόδους προγραμματισμού:

- 00:00 - 4:00;
- 4:00 - 8:00;
- 8:00 - 12:00;
- 12:00 - 16:00;
- 16:00 - 20:00;
- 20:00 - 00:00.

Υπάρχουν δραστηριότητες που εκτείνονται σε περισσότερες από μία χρονικές περιόδους και είναι μοιρασμένες στα διαστήματα αυτά ανάλογα με το χρόνο που απαιτείται για την ολοκλήρωσή τους. Προς το τέλος κάθε περιόδου, οι σταθμοί χρησιμοποιούν τις τελευταίες διαθέσιμες πληροφορίες για την επόμενη περίοδο, προκειμένου να κατασκευάσουν ένα σχέδιο για τις δραστηριότητες αυτής της περιόδου.

Οι τερματικοί σταθμοί του Hong Kong διαθέτουν προηγμένα συστήματα πληροφοριών που διατηρούν τις πληροφορίες σχετικά με τις ροές των εμπορευματοκιβωτίων σε πραγματικούς χρόνους και τις καταστάσεις των διάφορων πόρων. Οι πληροφορίες αυτές είναι απαραίτητες τόσο για την ασφάλεια των κιβωτίων και λόγους τελωνειακούς όσο και για την παροχή αξιόπιστων δεδομένων εισόδου για τη λήψη αποφάσεων.

Για τις αποφάσεις υπάρχουν δύο τύποι δεδομένων εισόδου: τα στατικά και τα δυναμικά. Τα στατικά δεδομένα είναι οι παράμετροι του συστήματος που γενικά διατηρούνται σταθερά σε όλη τη διάρκεια οποιουδήποτε ορίζοντα προγραμματισμού. Χαρακτηριστικά παραδείγματα στατικών δεδομένων αποτελούν ο αριθμός των γερανών της αποβάθρας και της αυλής, το δίκτυο των δρόμων στην αυλή του σταθμού, ο αριθμός και η χωρητικότητα των μπλοκ. Όποτε αυτά χρειάζονται είναι εύκολη η πρόσβαση στα αρχεία της βάσης δεδομένων. Τα δυναμικά δεδομένα αντανακλούν την κατάσταση των μεταβλητών (οι οποίες αλλάζουν με τις διάφορες δραστηριότητες), ενημερώνονται σε πραγματικούς χρόνους και οι τρέχουσες τιμές τους είναι διαθέσιμες από το σύστημα πληροφοριών. Τέτοιου είδους δεδομένα αποτελούν οι τρέχουσες θέσεις εξοπλισμού, ο αριθμός των κιβωτίων που είναι αποθηκευμένα στα μπλοκ, η θέση αποθήκευσης του κάθε κιβωτίου.

Οι αλλαγές στα δυναμικά δεδομένα προκαλούνται από τις αφίξεις των εμπορευματοκιβωτίων στους τερματικούς σταθμούς αλλά και τις αναχωρήσεις τους από αυτούς. Από την πλευρά της πύλης, οι χρόνοι άφιξης και αναχώρησης του εμπορευματοκιβωτίου καθορίζονται με τα ραντεβού που κλείνουν οι πελάτες για την παραλαβή ή την αποστολή των κιβωτίων. Στην αποβάθρα, πριν φτάσει το πλοίο, πληροφορίες σχετικά με την εκτιμώμενη ώρα άφιξης, ο αριθμός των εμπορευματοκιβωτίων που θα εκφορτωθούν στον τερματικό σταθμό του Hong Kong και οι ακριβείς τους θέσεις στο πλοίο, αποστέλλονται ηλεκτρονικά από τον μεταφορέα στον τερματικό σταθμό. Ο μεταφορέας δίνει επίσης πληροφορίες για τη φόρτωση των κιβωτίων στο πλοίο και τη καταπακτή στην οποία θα έπρεπε να πάει το καθένα από αυτά, βάσει το λιμάνι προορισμού, χωρίς να περιορίζεται η ακριβής θέση στην καταπακτή για κάθε ξεχωριστό κιβώτιο. Με τις πληροφορίες όλων των μεταφορέων, ο διαχειριστής των δραστηριοτήτων του τερματικού σταθμού κατασκευάζει ένα σχέδιο εργασίας για την φόρτωση και την εκφόρτωση χρησιμοποιώντας ένα λογισμικό ειδικό για αυτόν τον σκοπό. Η ώρα άφιξης του πλοίου και το σχέδιο φόρτωσης/εκφόρτωσης καθορίζει τους χρόνους άφιξης των εισερχόμενων κιβωτίων από το πλοίο και τις ώρες αναχώρησης των εξερχόμενων κιβωτίων στο πλοίο. Με τα σύγχρονα μέσα επικοινωνίας, αρκετά ακριβή δεδομένα που σχετίζονται με τα πλοία είναι διαθέσιμα για την οργάνωση και τη λήψη των αποφάσεων σχεδιασμού. Τα δεδομένα που αφορούν την πύλη δεν είναι πάντοτε γνωστά εκ των προτέρων με ακρίβεια επειδή κάποιοι πελάτες δεν κλείνουν ραντεβού πριν την παραλαβή ή την παράδοση των κιβωτίων.

### **1.13. Πηγές διακοπής των εργασιών σε έναν τερματικό σταθμό**

Σε έναν τερματικό σταθμό εντοπίζονται ορισμένα βασικά ζητήματα και πηγές διακοπής των λειτουργιών. Ειδικότερα, οι καθυστερήσεις που προκύπτουν σε μια προβλήτα έχουν σαν αποτέλεσμα τους υψηλούς χρόνους αναμονής των πλοίων στις προβλήτες, στις οποίες και παραμένουν αγκυροβολημένα για μεγαλύτερο χρονικό διάστημα από το κανονικό. Αυτές οι καθυστερήσεις μπορούν να αποδοθούν σε:

- Μη διαθεσιμότητα των προβλητών λόγω της συμφόρησης των εισερχόμενων πλοίων;
- Μη διαθεσιμότητα του απαιτούμενου αριθμού και του τύπου του εξοπλισμού στον επιθυμητό χρόνο, είτε επειδή ο εξοπλισμός αυτός χρησιμοποιείται σε άλλες εργασίες, είτε λόγω της απρόσμενης διακοπής της λειτουργίας του εξοπλισμού με αποτέλεσμα να διαταράσσεται το χρονοδιάγραμμα των εργασιών;
- Αβεβαιότητα στην άφιξη των φορτηγών που είναι υπεύθυνα για την παραλαβή και την παράδοση των φορτίων.

Η ομαλή λειτουργία ενός λιμανιού μπορεί να διαταραχθεί από έναν μεγάλο αριθμό αβεβαιοτήτων που μπορούν να προκύψουν κατά τη διάρκεια των εργασιών, όπως είναι διάφορα μηχανικά προβλήματα, η αβεβαιότητα των καιρικών συνθηκών, κλπ, και απαιτούν τη γρήγορη δράση σε πραγματικό χρόνο για την πρόληψη των ζημιών όσο γίνεται στο ελάχιστο επίπεδο. Μερικά από αυτά τα κοινά προβλήματα διακοπής είναι:

- η αβεβαιότητα των πληροφοριών;
- η αλλαγή του εκτιμώμενου χρόνου άφιξης των πλοίων;
- η ποικιλία των συνθηκών του προϊόντος;
- οι συνθήκες του πλοίου;
- τα φορτηγά μεταφορών;
- οι βλάβες;
- οι καιρικές συνθήκες;
- η αξιοπιστία εξοπλισμών;
- η αλλαγή της θέσης φόρτωσης/εκφόρτωσης του πλοίου με περιστροφή;
- οι αλλαγές στα φορτία, τους προμηθευτές και τους εμπόρους.

Ο κύριος στόχος του σχεδιασμού ενός πλάνου των εργασιών των τερματικών σταθμών είναι να ελαχιστοποιήσει το λειτουργικό κόστος, ενώ παράλληλα να μεγιστοποιήσει την αξιοπιστία του συστήματος. Ειδικότερα, ο σκοπός είναι να ελαχιστοποιηθεί ο αριθμός των κενών θέσεων που υπάρχουν για τα πλοία στους τερματικούς σταθμούς, διασφαλίζοντας ταυτόχρονα ότι η παροχή υπηρεσιών στα πλοία είναι σύμφωνη με ευρέως αποδεκτά πρότυπα.

## 1.14. Κρίσιμα προβλήματα βελτιστοποίησης

### 1.14.1. Πρόβλημα κατανομής πλοίων σε προβλήτες (berth allocation problem)

Σε αυτό το πρόβλημα δίνεται η διάταξη των προβλητών ενός τερματικού σταθμού εμπορευματοκιβωτίων, μαζί με ένα σύνολο σκαφών τα οποία πρέπει να εξυπηρετηθούν μέσα σε ένα προγραμματισμένο χρονικό ορίζοντα. Για κάθε σκάφος πρέπει να παρέχονται πληροφορίες για το μήκος του, την αναμενόμενη ώρα άφιξης, τον προβλεπόμενο χρόνο εξυπηρέτησης, το βάθος του νερού για το οποίο το πλοίο μπορεί να επιπλέει. Όλα τα πλοία θα πρέπει να έχουν αγκυροβολήσει εντός των ορίων της προβλήτας και δεν επιτρέπεται την ίδια χρονική στιγμή να καταλαμβάνουν τον ίδιο χώρο στην προβλήτα.

Λειτουργικό πρόβλημα κατανομής (OBAP): Αποτελείται από την ανάθεση και τον προγραμματισμό των πλοίων στις θέσεις ελλιμενισμού κατά μήκος της αποβάθρας, με στόχο την ελαχιστοποίηση του χρόνου ολοκλήρωσης των πλοίων. Το πρόβλημα αυτό καλύπτει συνήθως χρονικό ορίζοντα το πολύ μία εβδομάδα, λόγω της αβεβαιότητας στους χρόνους των θαλάσσιων ταξιδιών.

Τακτικό πρόβλημα κατανομής (TBAP): Σε αυτό το πρόβλημα ο χρονικός ορίζοντας είναι μεγαλύτερος και το βασικό κίνητρο είναι η υποστήριξη αποφάσεων που λαμβάνονται από τους διαχειριστές των τερματικών σταθμών κατά τη διαδικασία των διαπραγματεύσεων με ναυτιλιακές εταιρίες. Κατά τη διάρκεια αυτής της διαδικασίας, οι διαχειριστές θα πρέπει να αξιολογούν το αντίκτυπο της ανάθεσης ορισμένων λειτουργικών πόρων στην επίδοση των τερματικών σταθμών.

Τα δύο παραπάνω προβλήματα ασχολούνται με την ανάθεση και τον προγραμματισμό των πλοίων σε θέσεις ελλιμενισμού, δηλαδή λαμβάνονται αποφάσεις για το πού και το πότε τα πλοία θα πρέπει να προσαράξουν. Επιπλέον, στοχεύουν στην εξισορρόπηση του κόστους των τερματικών σταθμών και στην ποιότητα των υπηρεσιών. Στο τακτικό πρόβλημα κατανομής (TBAP) η ποιότητα της εξυπηρέτησης εξαρτάται από τις διαπραγματεύσεις μεταξύ του τερματικού και των ναυτιλιακών εταιριών όσον αφορά τους πόρους του σταθμού. Η υψηλότερη ποιότητα εξυπηρέτησης συμβαίνει όταν ο τερματικός σταθμός μπορεί να ανταπεξέλθει στα αιτήματα των εταιριών σε σχέση με τους αναμενόμενους χρόνους ελλιμενισμού, τους ανατιθέμενους γερανούς και τους αναμενόμενους χρόνους εξυπηρέτησης (handling times).

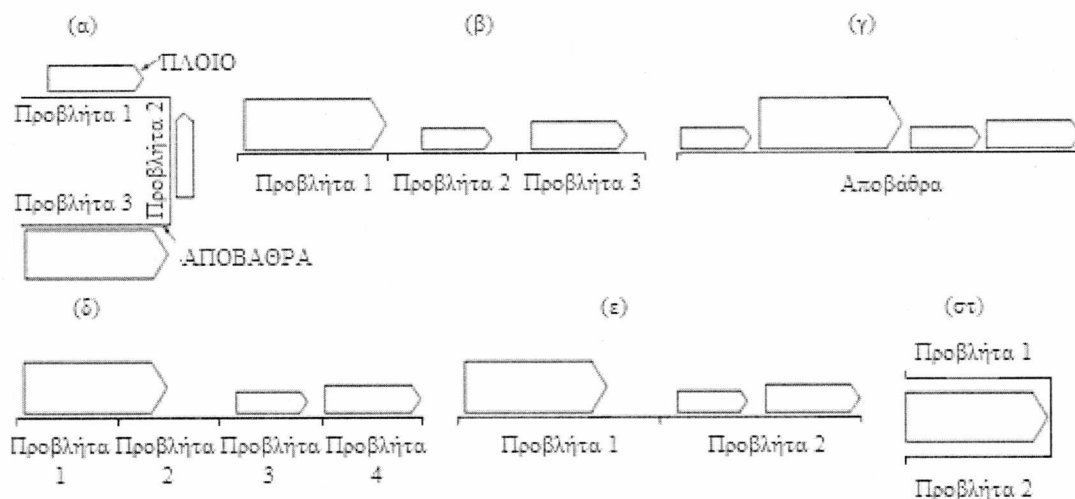
Σε ένα πρόβλημα κατανομής πλοίων σε προβλήτες (berth allocation problem) υπάρχουν οι παρακάτω περιπτώσεις:

*Διακριτή διάταξη προβλήτας:* Η αποβάθρα είναι χωρισμένη σε μια σειρά από ενότητες, που ονομάζονται προβλήτες. Μόνο ένα πλοίο μπορεί να εξυπηρετηθεί σε μία μόνο προβλήτα κάθε χρονική στιγμή. Αυτός ο διαχωρισμός μπορεί να γίνεται είτε ακολουθώντας το σχέδιο

κατασκευής της προβλήτας (εικόνα 7α) ή οργανώνεται έτσι ώστε να διευκολύνεται το πρόβλημα σχεδιασμού και προγραμματισμού (εικόνα 7β).

*Συνεχής διάταξη προβλήτας:* Δεν υπάρχει διαχωρισμός της αποβάθρας σε μέρη, δηλαδή τα πλοία μπορούν να προσαράξουν σε αυθαίρετες θέσεις εντός των ορίων της αποβάθρας (εικόνα 7γ). Η οργάνωση της προβλήτας με συνεχή διάταξη και η καλύτερη αξιοποίηση του χώρου της είναι περισσότερο περίπλοκη σε σχέση με προβλήτες που έχουν διακριτή διάταξη.

*Υβριδική διάταξη προβλήτας:* Όπως και στη περίπτωση των διακριτών, η προβλήτα είναι χωρισμένη σε θέσεις ελλιμενισμού, αλλά τα μεγάλα πλοία μπορούν να καταλαμβάνουν περισσότερες από μία αποβάθρες (εικόνα 7δ), ενώ τα μικρά πλοία μπορούν να μοιραστούν την αποβάθρα (εικόνα 7ε). Εάν δύο αντίθετες θέσεις ελλιμενισμού υφίστανται, είναι δυνατόν η ύπαρξη μιας αποβάθρας με εσοχή, έτσι ώστε να εξυπηρετήσει ένα μεγάλο πλοίο και από τις δύο πλευρές του (εικόνα 7στ).



Εικόνα 7. Περιπτώσεις διάταξης προβλητών. Πηγή: C. Bierwirth, F. Meisel (2010)

Χρονικοί περιορισμοί μπορούν να περιορίσουν τους χρόνους αγκυροβόλησης και αναχώρησης των πλοίων. Υπάρχουν οι παρακάτω περιπτώσεις (Imai et al. (2001)):

- Στατική άφιξη: Δε δίνονται οι χρόνοι άφιξης των πλοίων και στην περίπτωση αυτή θεωρείται ότι τα πλοία περιμένουν ήδη στο λιμάνι και μπορούν να προσαράξουν αμέσως. Επίσης, υπάρχει περίπτωση περιορισμού στους χρόνους ελλιμενισμού των πλοίων ανάλογα με τους χρόνους άφιξης. Για το λόγο αυτό, το πλοίο μπορεί να πληρώσει κάποιο κόστος προκειμένου να προσαράξει στην προβλήτα νωρίτερα από την αναμενόμενη ώρα άφιξης;

## Κεφάλαιο 1: Περιγραφή διαδικασιών σε ένα εμπορευματικό λιμάνι

- Δυναμική άφιξη: Στην περίπτωση αυτή, ενώ δίνονται σταθεροί χρόνοι άφιξης για τα πλοία, αυτά δεν μπορούν να αγκυροβολήσουν πριν από τους αναμενόμενους χρόνους άφιξης.

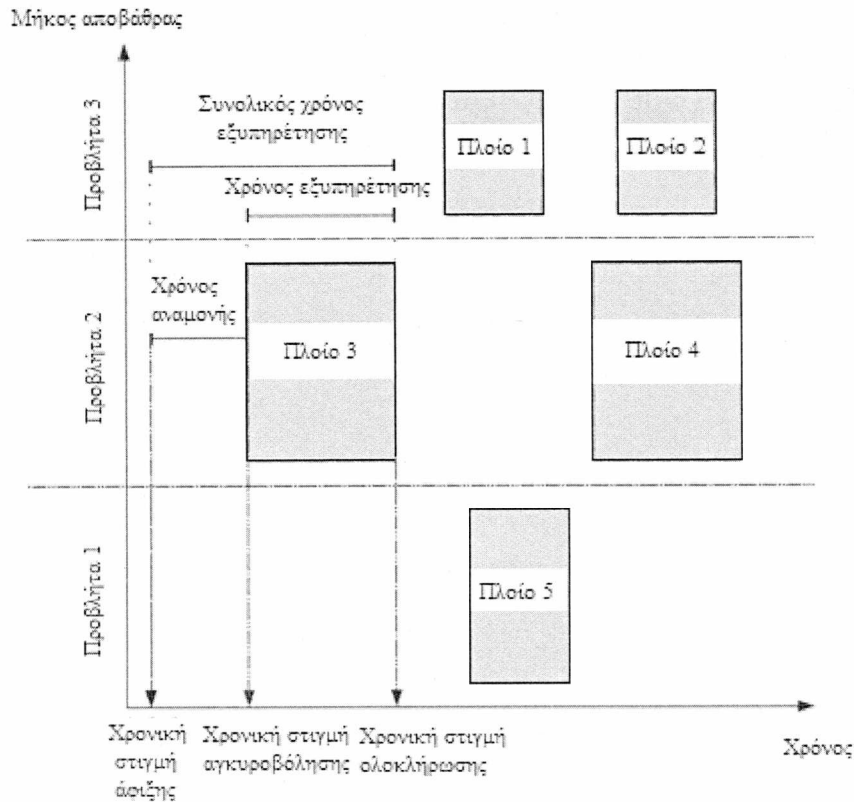
Οι αφίξεις των πλοίων μπορούν να εξεταστούν και περαιτέρω ως ντετερμινιστικές και στοχαστικές.

- Ντετερμινιστική άφιξη: Στην περίπτωση αυτή δίνονται σταθερές αναμενόμενες τιμές για τους χρόνους άφιξης;
- Στοχαστική άφιξη: Μπορεί να δοθεί μια κατανομή για τους χρόνους άφιξης των πλοίων για να υπολογιστεί η αβεβαιότητα που προκύπτει στις αφίξεις.

Ο καθορισμός των τελευταίων ωρών αναχώρησης των πλοίων μπορεί να πραγματοποιηθεί προκειμένου να διατηρηθεί το πρόγραμμα τακτικών γραμμών. Στην περίπτωση της δυναμικής άφιξης, ολόκληρη η εξυπηρέτηση του σκάφους πρέπει να εκτελείται εντός του καθορισμένου χρονικού παραθύρου.

Οι χρόνοι εξυπηρέτησης των πλοίων (handling times) στα περισσότερα μοντέλα είναι ντετερμινιστικοί και διακρίνονται στους εξής:

- I. Σε αυτούς που έχουν καθοριστεί, είναι δηλαδή γνωστοί εκ των προτέρων και θεωρούνται αμετάβλητοι;
- II. Σε αυτούς που εξαρτώνται από τις θέσεις αγκυροβόλησης των πλοίων;
- III. Σε εκείνους που εξαρτώνται από τον αριθμό των γερανών που εξυπηρετούν τα καράβια;
- IV. Σε εκείνους που εξαρτώνται από τα χρονοδιαγράμματα των εργασιών των γερανών που έχουν ανατεθεί στα πλοία;
- V. Σε αυτούς που αποτελούν συνδυασμό των II, III, IV;
- VI. Σε εκείνους που θεωρούνται στοχαστικοί, λαμβάνοντας υπόψη την αβεβαιότητα που μπορεί να προκύψει στους χρόνους αυτούς λόγω απρόβλεπτων διαταραχών, όπως είναι η βλάβη του εξοπλισμού, η μη διαθεσιμότητα του εξοπλισμού, κλπ.



Εικόνα 8. Παρουσίαση των χρονικών διαστημάτων σε προβλήτες. Πηγή: Steenken et al.(2004)

Σύμφωνα με τα παραπάνω, υπάρχουν τρεις παραδοχές για το πρόβλημα προγραμματισμού προβλητών σε ένα τερματικό σταθμό και είναι ο διακριτός ή συνεχής χώρος ελλιμενισμού, η στατική ή δυναμική άφιξη του πλοίου και ο στατικός ή δυναμικός χρόνος εξυπηρέτησης. Όπως αναφέρθηκε και προηγουμένως, στο διακριτό πρόβλημα η αποβάθρα χωρίζεται σε ένα σύνολο προβλητών ενώ, στο συνεχές το πλοίο μπορεί να αγκυροβολήσει οπουδήποτε κατά μήκος της αποβάθρας. Στο πρόβλημα στατικής άφιξης όλα τα πλοία βρίσκονται στο λιμάνι τη στιγμή που αρχίζει ο προγραμματισμός εξυπηρέτησης και στο πρόβλημα δυναμικής άφιξης δεν μπορούν να προσαράξουν στις προβλήτες πριν από τους χρόνους άφιξης που έχουν δοθεί. Τέλος, στο πρόβλημα με το στατικό χρόνο εξυπηρέτησης, αυτός ο χρόνος θεωρείται δεδομένος και στο πρόβλημα δυναμικού χρόνου εξυπηρέτησης αποτελεί μεταβλητή. Για να είναι η μοντελοποίηση του προβλήματος πιο κοντά στις πραγματικές συνθήκες μπορούν να ληφθούν υπόψη και κάποιοι τεχνικοί περιορισμοί όπως είναι η απόσταση ασφαλείας μεταξύ του πλοίου και το τέλος της προβλήτας. Ένας τέτοιος περιορισμός μπορεί να συμβάλει και στη μείωση του υπολογιστικού χρόνου της ακριβούς λύσης.

Η ανάπτυξη και η εφαρμογή μια ποικιλίας πολιτικών που σχετίζονται με τον προγραμματισμό της προβλήτας οφείλεται στον έντονο ανταγωνισμό μεταξύ των τερματικών σταθμών και στην καλύτερη αξιοποίηση των πόρων. Βασικοί παράγοντες που παίζουν ρόλο



στον καλύτερο προγραμματισμό είναι ο τύπος και η λειτουργικότητα του λιμένα, το μέγεθος του λιμένα, η περιοχή που αυτός βρίσκεται και αν υπάρχουν κοντά άλλοι τερματικοί σταθμοί, οι συμφωνίες που έχουν πραγματοποιηθεί μεταξύ των μεταφορέων και του τερματικού. Επιπλέον, το πόσο ανταγωνιστικό είναι ένα λιμάνι εξαρτάται από διάφορους παράγοντες όπως, ο χρόνος που απαιτείται για να ολοκληρωθεί η εξυπηρέτηση ενός πλοίου σε συνδυασμό με τη γρήγορη φόρτωση ή ξεφόρτωση του. Ο γενικός στόχος του προγραμματισμού της προβλήτας είναι η παροχή γρήγορης και αξιόπιστης εξυπηρέτησης των πλοίων, η αύξηση των εσόδων, η αύξηση της ανταγωνιστικότητας του τερματικού σταθμού και η αποτελεσματική ικανοποίηση των πελατών. Με τη βελτίωση ενός από τους παραπάνω στόχους σε έναν τερματικό σταθμό συχνά προκαλείται μείωση στην απόδοση των υπολοίπων. Για παράδειγμα, τα πλοία που διακινούν μεγάλο όγκο εμπορευμάτων συνήθως ζητούν υψηλότερη προτεραιότητα σε σχέση με τα μικρότερα πλοία, με αποτέλεσμα να μειώνεται η παραγωγικότητα της προβλήτας. Αν δοθεί προτεραιότητα στα πλοία με μικρότερο όγκο φορτίου, τότε προκαλείται η δυσaréσκεια των πελατών γιατί τα πλοία με τον μεγαλύτερο όγκο αναγκάζονται να περιμένουν όμως, ο συνολικός χρόνος εξυπηρέτησης των πλοίων μειώνεται. Έτσι, οι φορείς που σχετίζονται με την οργάνωση των τερματικών σταθμών πραγματοποιούν διάφορες συμφωνίες σχετικές με τη προτεραιότητα των πλοίων προκειμένου να αποφύγουν τη δυσaréσκεια των πελατών και να διατηρήσουν την αποδοτικότητα των σταθμών αυξάνοντας την ανταγωνιστικότητά τους.

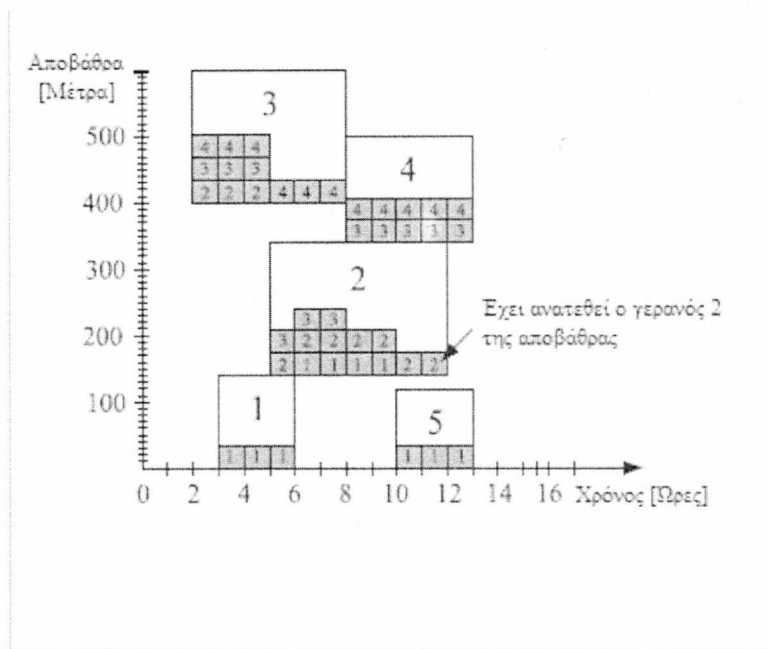
Υπάρχουν μοντέλα τα οποία στοχεύουν στην ελαχιστοποίηση του αθροίσματος του χρόνου αναμονής και του χρόνου εξυπηρέτησης των σκαφών (χρόνοι παραμονής στο λιμάνι). Περαιτέρω στόχοι είναι, η ελαχιστοποίηση του φόρτου εργασίας των τερματικών σταθμών και η ελαχιστοποίηση του αριθμού των πλοίων, τα οποία δεν θα εξυπηρετούνται και θα απορρίπτονται από τον τερματικό σταθμό. Η απόδοση του προγραμματισμού των προβλητών συχνά μετριέται από την άποψη του κόστους, η οποία επιτρέπει τον συνδυασμό διαφορετικών στόχων σε μια συνολική συνάρτηση κόστους.

#### **1.14.2. Πρόβλημα ανάθεσης γερανών αποβάθρας (quay crane assignment problem)**

Σε αυτό το πρόβλημα δίνεται ένα εφικτό σχέδιο προβλητών και ένα σύνολο πανομοιότυπων γερανών (QCs), οι οποίοι είναι διαθέσιμοι για την υπηρεσία. Για κάθε πλοίο το οποίο περιλαμβάνεται στο σχέδιο, είναι γνωστός ο όγκος εμπορευματοκιβωτίων που πρέπει να φορτωθούν και να εκφορτωθούν, καθώς και ο μέγιστος αριθμός γερανών που επιτρέπεται να το εξυπηρετήσει ταυτόχρονα. Οι γερανοί πρέπει να παρατάσσονται κατά μήκος της αποβάθρας, δεν είναι σε θέση να περνούν ο ένας τον άλλον και μπορούν να μετακινηθούν σε κάθε πλοίο. Το πρόβλημα είναι η σωστή ανάθεση των γερανών στα πλοία, έτσι ώστε όλες οι

απαιτούμενες μεταφορές των εμπορευματοκιβωτίων να πραγματοποιηθούν με τον καλύτερο δυνατό τρόπο. Τα προβλήματα ανάθεσης των γερανών (QCAP) και κατανομής των πλοίων σε προβλήτες (BAP) είναι αλληλένδετα, επειδή η επίλυση του πρώτου μπορεί να έχει ισχυρή επίδραση στους χρόνους εξυπηρέτησης (handling times) των πλοίων. Μόνο στην περίπτωση των προβλητών με διακριτή διάταξη, κάθε μία από αυτές κατέχει ένα σύνολο ειδικών γερανών και για το λόγο αυτό, δεν είναι απαραίτητη η ανάθεση των γερανών στα πλοία. Παρακάτω παρουσιάζονται οι συγκεκριμένες απαιτήσεις οι οποίες μπορούν να προκύψουν σε ένα πρόβλημα ανάθεσης γερανών (QCAP):

- I. Είτε ένα σύνολο συγκεκριμένων γερανών ή ένας αριθμός γερανών ανατίθεται σε ένα πλοίο;
- II. Ο αριθμός των γερανών που ανατίθενται σε ένα πλοίο είναι αμετάβλητος σε όλη τη διαδικασία εξυπηρέτησης, η οποία αναφέρεται ως χρονικά αναλλοίωτη ανάθεση(εικόνα 9, πλοία 1,4 και 5). Αντίθετα, μια χρονικά μεταβλητή ανάθεση γερανών πραγματοποιείται σε ορισμένες προσεγγίσεις(εικόνα 9,πλοία 2 και 3);



Εικόνα 9. Ανάθεση γερανών στα πλοία. Πηγή: C. Bierwirth, F. Meisel (2010)

- III. Ο ελάχιστος αριθμός γερανών που εξυπηρετεί το πλοίο σε όλη τη διάρκεια του χρόνου εξυπηρέτησης (handling time) μπορεί να καθοριστεί. Αυτός ο αριθμός είναι συνήθως μέρος της σύμβασης μεταξύ του ιδιοκτήτη του πλοίου και του διαχειριστή του τερματικού σταθμού.

Η ικανότητα του γερανού καθώς και ο όγκος φόρτωσης και εκφόρτωσης των εμπορευματοκιβωτίων μετριούνται συνήθως με βάση τις ώρες του γερανού αποβάθρας. Λόγω του γεγονότος ότι η παρεμβολή των γερανών μπορεί να παρεμποδίσει την αποτελεσματική χρήση των ανατιθέμενων ωρών, τουλάχιστον για τον όγκο φόρτωσης/εκφόρτωσης αυτό δεν είναι ένα απαιτούμενο μέτρο. Τυπικοί στόχοι του προβλήματος ανάθεσης είναι η ελαχιστοποίηση των απωλειών παραγωγικότητας των γερανών με τη μείωση του αριθμού των γερανών που ανατίθενται στα πλοία και τη μείωση των χρονικών στιγμών που οι γερανοί μετακινούνται.

Το πρόβλημα αυτό έχει αντίκτυπο στο πρόβλημα κατανομής (BAP). Με την υπόθεση ότι για ένα πλοίο απαιτούνται έξι βάρδιες εργασιών από τους γεραμούς Αποβάθρας (QCs), ο πελάτης είναι πρόθυμος να δεχτεί, είτε ένα εντατικό προφίλ των τριών γερανών που θα εργαστούν σε δύο βάρδιες, είτε ένα μεγαλύτερο σε διάρκεια, όπου θα εργαστούν δύο γερανοί σε τρεις βάρδιες. Το πιο γρήγορο προφίλ θα είναι πιθανότητα και πιο ικανοποιητικό για τον πελάτη, λόγω του μικρότερου χρόνου εξυπηρέτησης (handling time), ενώ το πιο αργό θα ασκήσει λιγότερη πίεση σχετικά με τη διαθεσιμότητα των γερανών. Ωστόσο, το πρόβλημα είναι πιο περίπλοκο επειδή αν η διαθεσιμότητα των γερανών δεν αποτελεί περιοριστικό παράγοντα, τότε ένας γρήγορος χρόνος εξυπηρέτησης είναι πλεονέκτημα για τον τερματικό σταθμό εξαιτίας της αύξησης της διαθεσιμότητας των προβλητών.

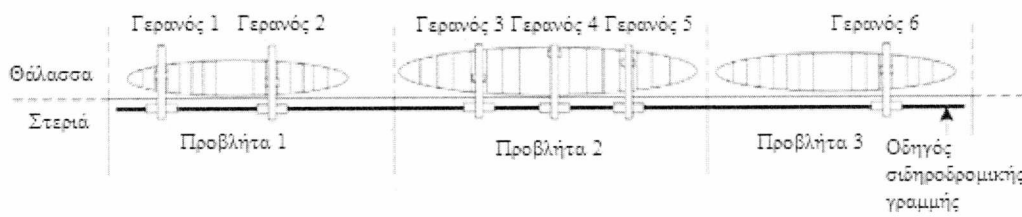
#### **1.14.3. Πρόβλημα προγραμματισμού των γερανών αποβάθρας (quay crane scheduling problem)**

Στο πρόβλημα αυτό (QCSP) λαμβάνεται υπόψη ένα σύνολο εργασιών, που αντιπροσωπεύει τις εργασίες φόρτωσης και εκφόρτωσης των πλοίων, καθώς και το σύνολο των ανατιθέμενων γερανών (QCs). Για τη διασφάλιση του γεγονότος ότι η εκφόρτωση προηγείται της φόρτωσης, αλλά και για τη τήρηση ενός ορισμένου σχεδίου για το στοίβαγμα των εμπορευματοκιβωτίων, δημιουργούνται κάποιες σχέσεις προτεραιότητας μεταξύ των καθηκόντων. Κάθε εργασία πρέπει να επεξεργάζεται από έναν γερανό (QC) μία φορά, ενώ ένας γεραμός μπορεί να επεξεργαστεί το πολύ μία εργασία σε μία χρονική στιγμή. Συνήθως, επιδιώκεται η ελαχιστοποίηση της χρονικής διάρκειας της λειτουργίας των γερανών (QCs) επειδή αντιπροσωπεύει τον συνολικό χρόνο εξυπηρέτησης (Handling Time) των πλοίων. Στο συγκεκριμένο πρόβλημα, υπάρχουν περιορισμοί προτεραιότητας και οι γερανοί αποτελούν παράλληλες πανομοιότυπες μηχανές.

Για την αποφυγή, κατά τη διέλευση των γερανών, να συμπέσουν μεταξύ τους, απαιτείται κάποιος χωρικός περιορισμός, ο οποίος δεν περιέχεται σε προβλήματα προγραμματισμού μηχανής. Για τον λόγο αυτό, περιλαμβάνονται σε ορισμένα εξελεγμένα μοντέλα του προβλήματος αυτού (QCSP) όρια ασφαλείας μεταξύ των γειτονικών γερανών. Τα καθήκοντα

των εργασιών για τους γερανούς ορίζονται ανάλογα με περιοχές του κόλπου ή μόνο τους κόλπους, καθώς και ανάλογα με τα στοιβαγμένα εμπορευματοκιβώτια, τις ομάδες που αυτά αποτελούν ή μεμονωμένα το καθένα:

- Περιοχές κόλπου: Μια εργασία αποτελείται απ' όλες τις διαδικασίες φόρτωσης και εκφόρτωσης εμπορευματοκιβωτίων σε μια συγκεκριμένη περιοχή του κόλπου;
- Όρμοι: Μια εργασία αποτελείται απ' όλες τις διαδικασίες φόρτωσης και εκφόρτωσης σε έναν κόλπο;
- Ομάδες: Μια εργασία αναφέρεται σε ομάδα εμπορευματοκιβωτίων τα οποία είναι αποθηκευμένα σε παρακείμενες υποδοχές του κόλπου. Τα εμπορευματοκιβώτια τα οποία ανήκουν σε κοινή ομάδα ή είναι παρόμοια μεταξύ τους έχουν συνήθως τον ίδιο προορισμό.



Εικόνα 10. Γερανοί αποβάθρας εξυπηρετούν τα πλοία που έχουν προσαράξει στις προβλήτες.

Πηγή: X.-I. Han et al.(2010)

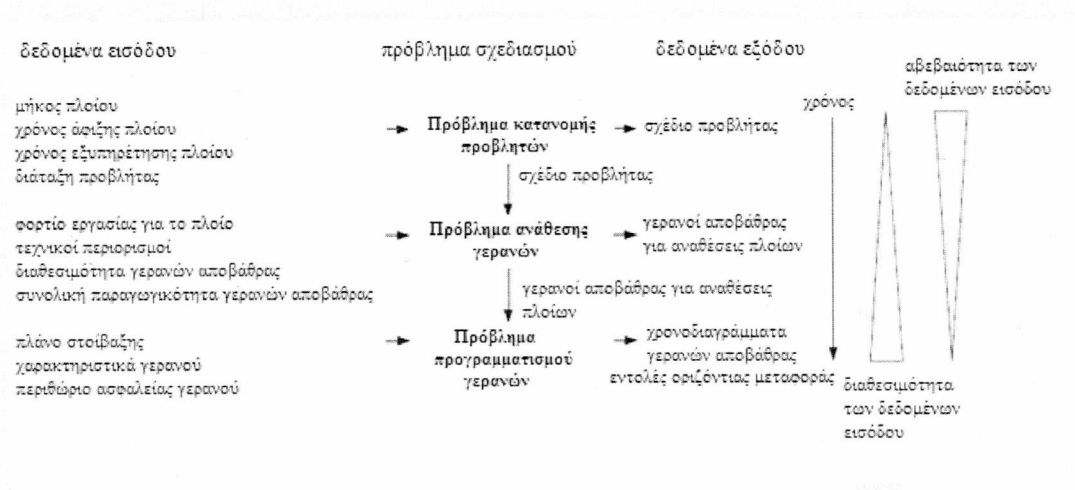
Προκειμένου κάθε περιοχή να εξυπηρετείται από έναν γερανό αποβάθρας, πραγματοποιείται η διαίρεση του φόρτου εργασίας για κάθε πλοίο σε περιοχές κόλπου. Αν οι περιοχές αυτές δεν αλληλεπικαλύπτονται, η παρεμβολή γερανών αποφεύγεται εντελώς. Συχνά, δεν είναι δυνατή η επαρκής κατανομή του φόρτου εργασίας των γερανών. Στην περίπτωση που οι εργασίες πραγματοποιούνται στον κόλπο και όχι σε περιοχές αυτού, ενδέχεται να είναι εφικτή η καλύτερη κατανομή του φόρτου εργασίας, αλλά η επίλυση του προβλήματος γίνεται περισσότερο πολύπλοκη λόγω του απαιτούμενου περιορισμού που υπάρχει για την αποφυγή της συνάντησης των γερανών μεταξύ τους. Επιπλέον, η πολυπλοκότητα του προβλήματος εξακολουθεί να είναι μέτρια, επειδή ανάλογα με το μέγεθος του πλοίου οριοθετούνται και τα καθήκοντα των εργασιών που πρέπει να πραγματοποιηθούν. Όταν το πλοίο είναι μεγάλο και κάθε εμπορευματοκιβώτιο φορτώνεται ή εκφορτώνεται ξεχωριστά το καθένα, το πρόβλημα (QCSP) μπορεί να γίνει ανεξέλεγκτο.

Παρακάτω παρουσιάζονται κάποιες ιδιότητες του γερανού για τον πιο λεπτομερή καθορισμό των λειτουργιών του:

- Χρόνοι ετοιμότητας: Κάθε χρόνος ετοιμότητας χρησιμοποιείται για να ορίσει τη λειτουργία του γερανού που είναι η συντομότερη δυνατή;
- Χρονικά παράθυρα: Τα χρονικά παράθυρα μπορούν να χρησιμοποιηθούν για κάθε έναν γερανό ή και περισσότερους, για τον προσδιορισμό των χρονικών διαστημάτων, κατά τη διάρκεια των οποίων ο γερανός είναι διαθέσιμος να εξυπηρετήσει το ανατιθέμενο πλοίο. Αυτά τα χρονικά παράθυρα είναι συχνά αποτέλεσμα μιας χρονικής μεταβλητής των αναθέσεων του γερανού;
- Θέσεις: Οι προβλεπόμενες αρχικές και τελικές θέσεις του γερανού;
- Χρόνοι διέλευσης: Η ταχύτητα της κίνησης του γερανού δίνεται σε σχέση με τον χρόνο που απαιτείται για να μετακινηθεί μεταξύ των όρμων.

Όταν η προβλήτα είναι κατασκευασμένη με εσοχή ο χρόνος που απαιτείται για την φόρτωση και την εκφόρτωση των πλοίων από τους γερανούς της αποβάθρας (QCs) μειώνεται. Το πρόβλημα (QCSP) σε αυτή την περίπτωση διαθέτει τα εξής μοναδικά χαρακτηριστικά:

- I. Σε προβλήτα που διαθέτει εσοχή, οι γερανοί της αποβάθρας είναι σε θέση να χειριστούν ένα πλοίο ταυτόχρονα και από τις δύο του πλευρές. Λόγω του σχεδιασμού των γερανών (ο βραχίονας ενός γερανού μπορεί να σηκωθεί, όταν πρόκειται να περάσει τους γερανούς στην απέναντι πλευρά), οι γερανοί της αποβάθρας που βρίσκονται σε διαφορετικές πλευρές της εσοχής είναι απαλλαγμένοι από τους περιορισμούς που υπάρχουν εξαιτίας της περίπτωσης τρακαρίσματος μεταξύ τους. Ωστόσο, η ανύψωση του βραχίονα ενός γερανού θα εισάγει έναν επιπλέον χρόνο αδράνειας για τον γερανό;
- II. Ο χρόνος διεκπεραίωσης μιας εργασίας (χρόνος εξυπηρέτησης, handling time), συμπεριλαμβανομένου του χρόνου επεξεργασίας και του χρόνου μεταφοράς, εξαρτάται από τον γερανό στον οποίο έχει ανατεθεί.



Εικόνα 11. Διαδοχικές κινήσεις προγραμματισμού των παραθαλάσσιων επιχειρήσεων. Πηγή: C. Bierwirth, F. Meisel (2010)

### 1.15. Πρόβλημα στοιβάγματος εμπορευματοκιβωτίων

Το φυσικό αέριο είναι μια πηγή ενέργειας ζωτικής σημασίας στη παγκόσμια παροχή ενέργειας. Μεταξύ άλλων, χρησιμοποιείται για την παραγωγή ηλεκτρικής ενέργειας, για θέρμανση και ως καύσιμο στα οχήματα. Το φυσικό αέριο είναι ευρέως διαδεδομένο σε σύγκριση με άλλες εναλλακτικές λύσεις λόγω της ιδιότητας του να δίνει καθαρότερη καύση και χαμηλές εκπομπές σε ρύπους. Ένας τρόπος μεταφοράς του από τον τόπο παραγωγής στους καταναλωτές είναι η μετατροπή του σε υγροποιημένο φυσικό αέριο (LNG), ακολουθούμενη από τη θαλάσσια μεταφορά με ειδικά πλοία στους καταναλωτές. Με αυτό τον τρόπο, το φυσικό αέριο μπορεί να παραδοθεί από μια μονάδα παραγωγής στους καταναλωτές σε οποιαδήποτε γωνία του κόσμου.

Όπως έχει αναφερθεί και προηγουμένως, ένα από τα μέτρα απόδοσης στους τερματικούς σταθμούς είναι ο χρόνος που δαπανάται από τα πλοία στις αποβάθρες των λιμανιών. Αυτός ο χρόνος αποτελείται κυρίως από τον χρόνο φόρτωσης και εκφόρτωσης των εμπορευματοκιβωτίων. Προκειμένου να μειωθεί ο χρόνος φόρτωσης, είναι απαραίτητο να διατηρηθούν τα εξαγόμενα κιβώτια χωρίς ανασχηματισμούς μέσω των μεθοδευμένων τρόπων οργάνωσης και εργασιών. Αυτό το πρόβλημα είναι γνωστό ως πρόβλημα στοιβάγματος εμπορευματοκιβωτίων (CSP). Για να υπάρξει πρόσβαση σε ένα κιβώτιο, το οποίο δεν βρίσκεται στην κορυφή της στοιβάς του, τα εμπορευματοκιβώτια που είναι τοποθετημένα πάνω πρέπει να μετακινηθούν. Αυτό συμβαίνει επειδή άλλα πλοία έχουν εκφορτώσει προηγουμένως και τα εμπορευματοκιβώτια στοιβάζονται με λάθος σειρά λόγω της έλλειψης ακριβών πληροφοριών. Επιπρόσθετα, αυτή η μετακίνηση των κιβωτίων μειώνει την

παραγωγικότητα των γερανών. Για την μεγιστοποίηση της αποτελεσματικότητας αυτής της διαδικασίας προκύπτουν οι παρακάτω απαιτήσεις:

- Κάθε εισερχόμενο εμπορευματοκιβώτιο θα πρέπει να διαθέτει μια θέση στη στοίβα η οποία να είναι ελεύθερη να καλυφθεί κατά της στιγμή της άφιξης;
- Κάθε εξερχόμενο εμπορευματοκιβώτιο θα πρέπει να είναι εύκολα προσβάσιμο και κατά προτίμηση κοντά στη θέση εκφόρτωσής του, κατά τη στιγμή της αποχώρησής του.

Επιπλέον, υπάρχουν και κάποιοι άλλοι περιορισμοί που πρέπει να ληφθούν υπόψη σχετικά με τις περιοχές των εμπορευματοκιβωτίων όπως τα επικίνδυνα κιβώτια θα πρέπει να στοιβάζονται ξεχωριστά διατηρώντας μια ελάχιστη απόσταση, οι μικρές διαφορές στο ύψος της αποβάθρας και του κόλπου. Στους τερματικούς σταθμούς η λειτουργία φόρτωσης για τα εξαγόμενα εμπορευματοκιβώτια έχει προσχεδιαστεί από τους υπεύθυνους προγραμματισμού του φορτίου. Γενικότερα, υπάρχουν δύο διαφορετικές προσεγγίσεις τους προβλήματος ανάλογα με το πότε θα πρέπει να γίνει η βελτιστοποίηση:

- Η ελαχιστοποίηση του αριθμού των αλλαγών θέσεων κατά τη διάρκεια συλλογής των εμπορευματοκιβωτίων;
- Να υπάρχει μια επιθυμητή διάταξη για τον κόλπο πριν ξεκινήσει η διαδικασία συλλογής προκειμένου να ελαχιστοποιηθεί ο αριθμός των μετακινήσεων κατά τη διάρκεια αυτής της διαδικασίας.

### **1.16. Πρόβλημα δρομολόγησης και προγραμματισμού πλοίων που μεταφέρουν υγροποιημένο φυσικό αέριο**

Το πρόβλημα δρομολόγησης και προγραμματισμού των πλοίων που μεταφέρουν υγροποιημένο φυσικό αέριο (LNG) είναι ένα πραγματικό πρόβλημα σχεδιασμού που αντιμετωπίζεται από έναν από τους μεγαλύτερους παγκοσμίως παραγωγούς υγροποιημένου φυσικού αερίου. Η ετήσια δυναμικότητα παραγωγής για τον παραγωγό ανέρχεται σε 42 εκατομμύρια τόνους. Ο παραγωγός υγροποιημένου φυσικού αερίου έχει συμβατική υποχρέωση να μεταφέρει το αέριο αυτό από το λιμάνι παραγωγής σε πελάτες που βρίσκονται σε όλο τον κόσμο. Κάθε χρόνο, ο παραγωγός πρέπει να δημιουργήσει και να παρουσιάσει ένα ετήσιο πρόγραμμα παράδοσης στους πελάτες, που καθορίζει πότε αυτοί θα λαμβάνουν αποστολές υγροποιημένου φυσικού αερίου στη διάρκεια του έτους (συμπεριλαμβανομένου του χρόνου παράδοσης, από ποιο πλοίο και τη ποσότητα του αερίου).

Οι μακροπρόθεσμες συμβάσεις αναφέρουν πόσο υγροποιημένο φυσικό αέριο πρόκειται να παραδοθεί σε κάθε πελάτη κατά τη διάρκεια του έτους. Οι πραγματικές ημερομηνίες

παράδοσης πρέπει να συμφωνηθούν με μία διαδικασία όπου ο παραγωγός του υγροποιημένου φυσικού αερίου θα δημιουργήσει ένα αρχικό ετήσιο πρόγραμμα παράδοσης με κάποιες ημερομηνίες που έχουν καθοριστεί κατά τη διαδικασία τη δημιουργίας του τελικού σχεδίου.

Για τη μεταφορά του υγροποιημένου φυσικού αερίου από το λιμάνι παραγωγής στους πελάτες, ο παραγωγός ελέγχει έναν ετερογενή στόλο με πλοία, τα οποία διαθέτουν ποικίλες δυνατότητες και ικανότητες φόρτωσης, καθώς και ταχύτητες πλεύσης. Ο στόλος αυτός καθορίζεται κατά τη διάρκεια προγραμματισμού του χρονικού ορίζοντα και ορισμένα από τα πλοία ικανοποιούν συγκεκριμένες συμβάσεις παράδοσης και μπορούν να χρησιμοποιηθούν μόνο για τις υπηρεσίες συνόλου των πελατών.

Όλες οι παραδόσεις υγροποιημένου φυσικού αερίου πραγματοποιούνται από πλοία τα οποία είναι γεμάτα από αυτό, δεδομένου ότι δεν είναι οικονομικά ευεργετικό να επισκέπτονται παραπάνω από έναν πελάτη σε ένα ταξίδι πριν επιστρέψουν στο λιμάνι παραγωγής. Αυτό δημιουργεί ένα απλό δομικό δίκτυο με ένα λιμάνι επιλογής, πολλά λιμάνια παράδοσης και πλήρη πλοία μεταφοράς. Κάθε αποστολή υγροποιημένου φυσικού αερίου θα συνιστάται από ένα ταξίδι μετ' επιστροφής από το λιμάνι παραγωγής σε έναν πελάτη και πίσω στο λιμάνι παραγωγής.

Ένα πλοίο πλήρες φορτωμένο αντιπροσωπεύει ένα φορτίο. Με βάση το μέσο όρο των ικανοτήτων φόρτωσης των πλοίων, ο παραγωγός αρχικά υπολογίζει πόσα φορτία πρέπει να παραδοθούν σε κάθε πελάτη κατά τη διάρκεια του έτους και καθορίζει ένα χρονικό διάστημα για το πότε τα φορτία θα πρέπει να συλλεχθούν στο λιμάνι παραγωγής, δεδομένου των προδιαγραφών που υπάρχουν στις συμβάσεις των πελατών. Η διαδικασία αυτή είναι αποδεκτή όταν η δυνατότητα φόρτωσης για τα πλοία που μπορούν να επισκεφτούν ένα συγκεκριμένο υποσύνολο των πελατών διαφέρει σε μικρό βαθμό (λιγότερο από 10% διαφορά ανάμεσα στη μικρότερη και μεγαλύτερη χωρητικότητα του πλοίου). Ο παραγωγός μπορεί να προβεί σε υπό ή υπέρ παράδοση στον ετήσιο συμβατικό όγκο για την παράδοση στον κάθε πελάτη, που επιτρέπει τα πλοία με διαφορετική ικανότητα φόρτωσης να επισκεφτούν τους πελάτες. Φθηνότερα πλοία προτιμώνται όταν πραγματοποιείται υπό παράδοση.

Στο λιμάνι παραγωγής η χωρητικότητα για τον ελλιμενισμό των πλοίων είναι περιορισμένη. Από τα πλοία που μπορούν να πάνε στις διαθέσιμες προβλήτες μόνο αυτά μπορούν να συλλέξουν το φορτίο σε μια δεδομένη ημέρα. Επιπλέον, υπάρχει περιορισμένη ικανότητα απογραφής του υγροποιημένου φυσικού αερίου, απαιτώντας έτσι τα επίπεδα αποθεμάτων του αερίου να είναι ανά πάσα στιγμή εντός των ελάχιστων και μέγιστων επιπέδων. Συνήθως, η παραγωγή του υγροποιημένου αερίου είναι υψηλότερη από τους όγκους που έχουν καθοριστεί για παράδοση στους πελάτες. Τα συγκεντρωτικά φορτία πωλούνται στην ελεύθερη αγορά. Αυτά συλλέγονται από πλοία τα οποία δεν ανήκουν στον στόλο του παραγωγού και έτσι επηρεάζεται μόνο η διαθεσιμότητα των προβλητών για ελλιμενισμό και τα επίπεδα των αποθεμάτων σε υγροποιημένο φυσικό αέριο.

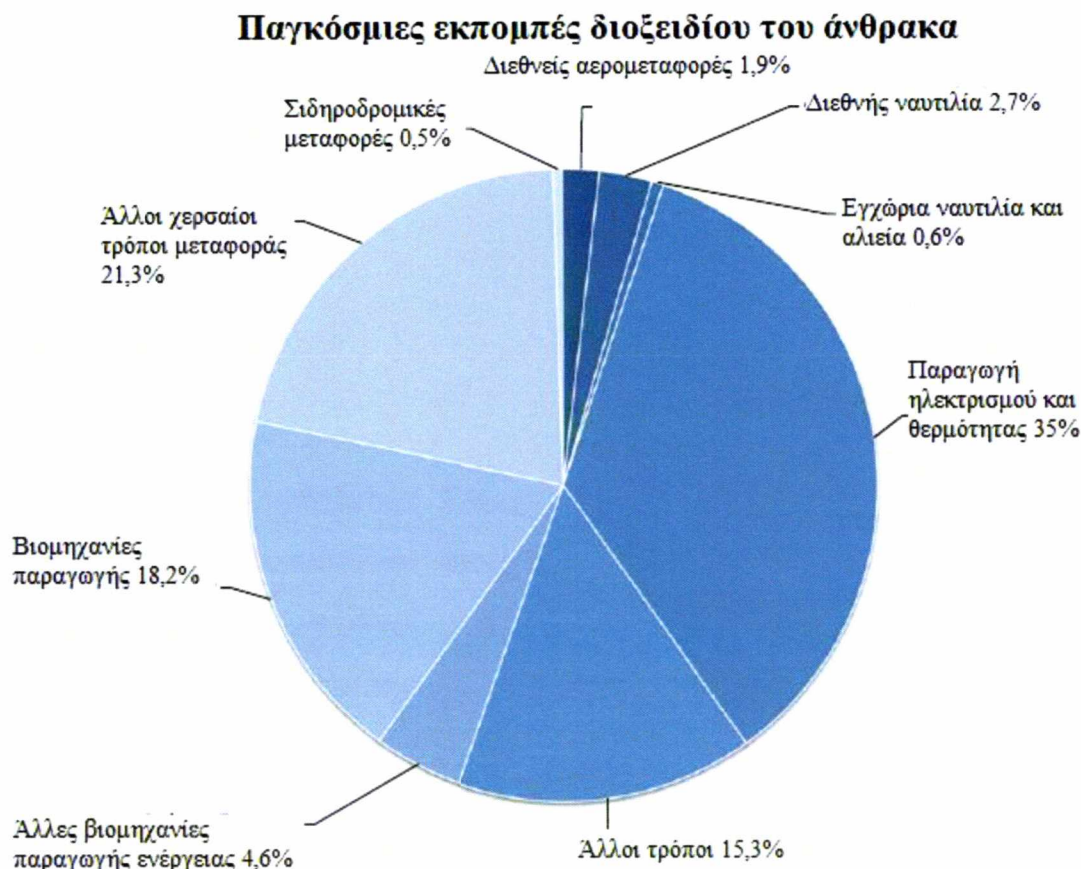


Το πρόβλημα δρομολόγησης και προγραμματισμού πλοίων που μεταφέρουν το υγροποιημένο φυσικό αέριο με τη δημιουργία του ετήσιου προγράμματος παράδοσης στοχεύει στην ελαχιστοποίηση του κόστους μεταφοράς όλων των φορτίων στους πελάτες μέσα στα καθορισμένα χρονικά διαστήματα, ενώ την ίδια στιγμή διασφαλίζεται η διαθεσιμότητα της προβλήτας και δεν παραβιάζονται οι περιορισμοί στα επίπεδα των αποθεμάτων στο λιμάνι παραγωγής.

### **1.17. Ο ρόλος της ταχύτητας στην ενεργειακή απόδοση των θαλάσσιων μεταφορών**

Η ταχύτητα είναι μια βασική μεταβλητή στις θαλάσσιες μεταφορές. Τα πλοία ταξιδεύουν πιο αργά σε σχέση με τα άλλα μέσα και επειδή ταξίδια μεγάλων αποστάσεων μπορεί να διαρκέσουν από 1 έως 2 μήνες, τα οφέλη από μια υψηλότερη ταχύτητα μπορεί να είναι σημαντικά. Υψηλότερη ταχύτητα συνεπάγεται την ταχύτερη παράδοση των εμπορευμάτων στους παραλήπτες τους, το χαμηλότερο κόστος αποθήκευσης στην αποβάθρα και την αύξηση του εμπορίου/ανταλλαγής αποθεμάτων ανά μονάδα χρόνου. Η ανάγκη για υψηλότερες ταχύτητες στη ναυτιλία δημιουργείται κυρίως από την ισχυρή ανάπτυξη του παγκόσμιου εμπορίου και έγινε εφικτή χάρη στις σημαντικές τεχνολογικές εξελίξεις στον τομέα των θαλάσσιων μεταφορών, όπως είναι ο σχεδιασμός του πλοίου, η υδροδυναμική απόδοση των πλοίων, η απόδοση των κινητήρων. Συμπερασματικά, οι εξελίξεις στα συστήματα διακίνησης των εμπορευματοκιβωτίων και διαχείρισης της εφοδιαστικής αλυσίδας και λειτουργιών συνέβαλε στη γρήγορη μεταφορά.

Η αύξηση των τιμών των καυσίμων δημιουργεί προβλήματα στις συνθήκες αγοράς και στα περιβαλλοντικά ζητήματα, όπως είναι οι εκπομπές στον αέρα από τα πλοία με καινούριες προοπτικές στη ταχύτητα που πλέουν. Συνεπώς, η γρήγορη πλεύση μπορεί να μην είναι και η καλύτερη επιλογή, καθώς ένα πλοίο που πηγαίνει γρηγορότερα θα εκπέμπει και περισσότερα καυσαέρια στο περιβάλλον. Ωστόσο, η μείωση της ταχύτητας μπορεί να έχει κάποιες μη ευεργετικές προεκτάσεις. Για παράδειγμα, περισσότερα πλοία θα χρειαστούν για την παραγωγή της ίδιας αποδοτικότητας της μεταφοράς και αυτό με τη σειρά του συνεπάγεται οικονομικές και περιβαλλοντικές δαπάνες. Επιπλέον, τα κόστη αποθήκευσης των φορτίων θα αυξηθούν εξαιτίας του αυξημένου χρόνου διέλευσής τους. Αυτά τα κόστη αποθήκευσης είναι ανάλογα με την αξία των φορτίων και αν ένα πλοίο μεταφέρει εμπορευματοκιβώτια υψηλής αξίας, η πλεύση με χαμηλότερη ταχύτητα μπορεί να συνεπάγεται σημαντικό κόστος για τον αποστολέα. Μια ακόμα επίπτωση της μείωσης της ταχύτητας είναι ότι οι αποστολείς θα προτιμήσουν τους χερσαίους τρόπους μεταφοράς και αυτό θα έχει ως αποτέλεσμα την αύξηση στις εκπομπές των αερίων.



Εικόνα 22. Οι εκπομπές διοξειδίου του άνθρακα από τη ναυτιλία σε σχέση με τις συνολικές παγκόσμιες εκπομπές για το 2007. Πηγή: IMO (2009)

Συμπερασματικά, η σωστή λειτουργία ενός τερματικού σταθμού και η καλύτερη εξυπηρέτηση των πλοίων που μεταφέρουν τα εμπορεύματα βασίζεται στην αποτελεσματική λήψη αποφάσεων που αφορούν τις εργασίες που πρέπει να γίνουν στο λιμάνι αλλά και την κατάλληλη αντιμετώπιση των παραγόντων που επηρεάζουν πολλές φορές αυτές τις εργασίες. Με την κατάλληλη οργάνωση του εξοπλισμού, του χώρου αποθήκευσης των κιβωτίων και την ανάπτυξη χρονοδιαγράμματος είναι δυνατόν να επιτευχθεί ο κατάλληλος προγραμματισμός και τα πλοία να εξυπηρετηθούν με τον καλύτερο δυνατό τρόπο. Στην παρακάτω ενότητα, γίνεται αναφορά στο πρόβλημα ανάθεσης πλοίων στις προβλήτες, ένα από τα βασικά προβλήματα που απασχολεί έναν τερματικό σταθμό και αναφέρθηκε παραπάνω.

## 2. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΑΝΑΘΕΣΗΣ (BAP)

Ξεκινώντας, στην ενότητα 2.1, πραγματοποιείται η περιγραφή του πρώτου μοντέλου το οποίο προτάθηκε από τον Imai et al. (2001) και η βελτιωμένη επέκταση του από τους Monaco & Sammarra (2007). Στην ενότητα 2.2, περιγράφεται το πρόβλημα ανάθεσης πλοίων με χρόνο διακριτό (DBAP) που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας. Τέλος, στην ενότητα 2.3 παρουσιάζονται τα συμπεράσματα από τη πρώτη εικόνα περιγραφής των τριών μοντελοποιήσεων.

### 2.1 Παρουσίαση πρώτης μοντελοποίησης του προβλήματος ανάθεσης πλοίων

Οι μεταβλητές απόφασης διέπουν την ανάθεση των πλοίων σε προβλήτες, καθώς και τη σειρά προτεραιότητας με την οποία τα πλοία θα εξυπηρετηθούν σε αυτές. Κάθε μεταβλητή απόφασης είναι μια δυαδική μεταβλητή της μορφής  $x_{ikp}$  η οποία δηλώνει ότι το πλοίο  $i$  θα εξυπηρετηθεί στην προβλήτα  $k$  με σειρά προτεραιότητας  $p$ . Ας υποθέσουμε ότι έχουμε μια σειρά από θέσεις ελλιμενισμού  $M$  (με  $|M| = m$ ), ένα σύνολο από πλοία  $N$  (με  $|N| = n$ ) που θέλουν να αγκυροβολήσουν και ένα σύνολο με σειρές εξυπηρέτησης  $P$  (με  $|P| = n$ ). Περαιτέρω υποθέτουμε ότι ο χρόνος εξυπηρέτησης (handling time) του πλοίου  $i$  στην προβλήτα  $k$  είναι  $h_{ik}$ , ο χρόνος άφιξης του πλοίου  $i$  δίνεται από το  $a_i$  και το  $s_k$  ορίζει τον χρόνο στον οποίο η προβλήτα  $k$  γίνεται διαθέσιμη για τον προγραμματισμό της κατανομής. Υπάρχει περίπτωση να μη φτάνουν όλα τα πλοία στην προβλήτα ανάθεσης τους πριν το  $s_k$  και για το λόγο αυτό, ένα σύνολο  $W_k$  είναι απαραίτητο για να αναφέρει το σύνολο των πλοίων που ικανοποιούν  $a_i \geq s_k$ . Επιπλέον, πρέπει να οριστεί  $P_{(p)} = \{q \in P : q < p\}$ , το οποίο δίνει το σύνολο των σειρών εξυπηρέτησης πριν από το  $p$  και η μεταβλητή απόφασης  $y_{ikp}$  που δίνει τον χρόνο αδράνειας στην προβλήτα  $k$  μεταξύ της αναχώρησης του πλοίου με σειρά προτεραιότητας  $(p-1)$  και της άφιξης του πλοίου με σειρά  $p$ , αν το πλοίο  $i$  είναι το πλοίο προς εξυπηρέτηση με σειρά  $p$ . Σύμφωνα με τον Cordeau et al. (2005), το αρχικό μοντέλο αγνοεί το γεγονός ότι κάθε προβλήτα κάποια χρονική στιγμή  $e_k$  κλείνει και δεν εξυπηρετεί. Για τον λόγο αυτό παρατηρείται η προσθήκη ενός επιπλέον περιορισμού, που λαμβάνει υπόψη το χρονικό παράθυρο λειτουργίας της προβλήτας.

## Κεφάλαιο 2: Περιγραφή και μοντελοποίηση του προβλήματος ανάθεσης (BAP)

### Δείκτες

$i, l \in N$  ( $|N| = n$ )

$k \in M$  ( $|M| = m$ )

$p \in P$  ( $|P| = n$ )

Ο αριθμός των πλοίων με  $i \neq l$

Ο αριθμός των προβλητών

Η σειρά εξυπηρέτησης των πλοίων

### Δεδομένα

$h_{ik}$

Ο χρόνος εξυπηρέτησης του πλοίου  $i$  στην προβλήτα  $k$

$a_i$

Ο χρόνος άφιξης του πλοίου  $i$

$s_k$

Ο χρόνος που η προβλήτα  $k$  γίνεται διαθέσιμη

$e_k$

Ο χρόνος ολοκλήρωσης της εξυπηρέτησης στη προβλήτα  $k$

### Μεταβλητές απόφασης

$x_{ikp}$

Ισούται με 1 όταν το πλοίο  $i$  εξυπηρετηθεί στην προβλήτα  $k$  με σειρά προτεραιότητας  $p$ , αλλιώς 0.

$y_{ikp}$

Δίνει τον χρόνο που η προβλήτα  $k$  μένει ανενεργή, μεταξύ της αναχώρησης του πλοίου με σειρά εξυπηρέτησης  $p-1$  και της άφιξης του πλοίου με σειρά  $p$ .

Το μοντέλο έχει την εξής μορφή σύμφωνα με τον Imai et al. (2001) και παρουσιάζεται παρακάτω:

$$\text{Min} \sum_{k \in M} \sum_{i \in N} \sum_{p \in P} (h_{ik} + s_k - a_i) x_{ikp} + \sum_{k \in M} \sum_{i \in W_k} \sum_{p \in P} y_{ikp} \quad (1)$$

### Περιορισμοί

$$\sum_{k \in M} \sum_{p \in P} x_{ikp} = 1$$

$$\forall i \in N, \quad (2)$$

$$\sum_{i \in N} x_{ikp} \leq 1 \quad \forall k \in M, p \in P, \quad (3)$$

$$\sum_{i \in N} \sum_{q \in P(p)} (h_{ik} x_{ikq} + y_{ikq}) + y_{ikp} \geq (a_i - s_k) x_{ikp} \quad \forall i \in W_k, p \in P, k \in M, \quad (4)$$

$$\sum_{i \in N} \sum_{p \in P} (h_{ik} x_{ikp} + y_{ikp}) \leq e_k - s_k \quad \forall k \in M, \quad (5)$$

$$x_{ikp} \in \{0,1\} \quad \forall i \in N, p \in P, k \in M, \quad (6)$$

$$y_{ikp} \geq 0 \quad \forall i \in N, p \in P, k \in M, \quad (7)$$

Η αντικειμενική συνάρτηση δίνεται από την (1) και ελαχιστοποιεί τον συνολικό χρόνο αναμονής και χρόνο εξυπηρέτησης του κάθε πλοίου. Οι περιορισμοί (2) και (3) βεβαιώνουν ότι κάθε πλοίο εξυπηρετείται σε μία προβλήτα και κάθε προβλήτα μπορεί να εξυπηρετηθεί το πολύ ένα πλοίο τη φορά αντίστοιχα. Ο περιορισμός (4) δείχνει ότι για κάθε πλοίο  $i$ , το οποίο φτάνει μετά το άνοιγμα της προβλήτας, ο χρόνος στον οποίο μπορεί να αρχίσει να εξυπηρετείται δεν μπορεί να είναι μικρότερος από  $a_i - s_k$ . Συμπερασματικά, αν το πλοίο  $i$  είναι αυτό που εξυπηρετείται με σειρά προτεραιότητας  $p$  στην προβλήτα  $k$  και ο χρόνος άφιξης του είναι μικρότερος από τον χρόνο ολοκλήρωσης της εξυπηρέτησης του πλοίου με σειρά προτεραιότητας  $p-1$ , τότε  $y_{ikp} = 0$ . Με τον περιορισμό (5) εξασφαλίζεται ότι η εξυπηρέτηση ενός πλοίου πραγματοποιείται ανάμεσα στο χρονικό παράθυρο λειτουργίας  $[e_k, s_k]$  της προβλήτας. Οι περιορισμοί (6) και (7) δηλώνουν ότι η μεταβλητή  $x_{ikp}$  παίρνει δυαδικές τιμές και η  $y_{ikp}$  δηλώνει ότι ο χρόνος αδράνειας της προβλήτας είναι μη αρνητικός.

Το βελτιωμένο μοντέλο σύμφωνα με τους Monaco & Sammarra (2007) δείχνει ότι η αρχική μοντελοποίηση του Imai et al. (2001) μπορεί να ενισχυθεί λαμβάνοντας υπόψη ότι οι ανενεργοί χρόνοι είναι ανεξάρτητοι από το ποιο πλοίο εξυπηρετείται τελικά στη προβλήτα  $k$  με σειρά  $p$ . Έτσι η μεταβλητή απόφασης  $y_{ikp}$  μετατρέπεται σε  $y_{kp}$  και δηλώνει τον χρόνο

## Κεφάλαιο 2: Περιγραφή και μοντελοποίηση του προβλήματος ανάθεσης (BAP)

αδράνειας μεταξύ του ξεκινήματος της εξυπηρέτησης με σειρά  $p$  και της ολοκλήρωσης της εξυπηρέτησης  $p-1$  στην προβλήτα  $k$ . Επομένως, με τις παραπάνω αναφορές προκύπτει το βελτιωμένο μοντέλο της ανάθεσης πλοίων σε προβλήτες και έχει τα παρακάτω χαρακτηριστικά:

Δείκτες

$i \in N$ ( $ N  = n$ )	Ο αριθμός των πλοίων
$j \in N$ ( $ N  = n$ )	Ο αριθμός των πλοίων με $j \neq i$
$k \in M$ ( $ M  = m$ )	Ο αριθμός των προβλητών
$p \in P$ ( $ P  = n$ )	Η σειρά εξυπηρέτησης των πλοίων
$l \in P_{(p)}$ ( $P_{(p)} = \{l \in P : l < p\}$ )	Η σειρά εξυπηρέτησης των πλοίων πριν από αυτά με σειρά $p$

Δεδομένα

$h_{ik}$	Ο χρόνος εξυπηρέτησης του πλοίου $i$ στην προβλήτα $k$
$a_i$	Ο χρόνος άφιξης του πλοίου $i$
$s_k$	Ο χρόνος που η προβλήτα $k$ γίνεται διαθέσιμη
$e_k$	Ο χρόνος στον οποίο η προβλήτα $k$ σταματάει τη λειτουργία της

Μεταβλητές απόφασης

$x_{ikp}$	Ισούται με 1 όταν το πλοίο $i$ εξυπηρετηθεί στην προβλήτα $k$ με σειρά προτεραιότητας $p$ , αλλιώς 0.
$y_{kp}$	Ανενεργός χρόνος ανάμεσα στην έναρξη της $p$ εξυπηρέτησης και της ολοκλήρωσης της $p-1$ στην προβλήτα $k$ .

Η μορφή του μοντέλου παρουσιάζεται παρακάτω:

$$\text{Min} \sum_{k \in M} \sum_{i \in N} \sum_{p \in P} (h_{ik} + s_k - a_i) x_{ikp} + \sum_{k \in M} \sum_{p \in P} y_{kp} \quad (8)$$

Περιορισμοί

$$\sum_{k \in M} \sum_{p \in P} x_{ikp} = 1 \quad \forall i \in N, \quad (9)$$

$$\sum_{i \in N} x_{ikp} \leq 1 \quad \forall k \in M, p \in P, \quad (10)$$

$$\sum_{i \in W_k} (a_i - s_k) x_{ikp} - \sum_{l \in P(p)} (y_{lk} + \sum_{j \in N} h_{jk} x_{jkl}) - y_{kp} \leq 0 \quad \forall p \in P, k \in M, \quad (11)$$

$$\sum_{i \in N} \sum_{p \in P} (h_{ik} x_{ikp} + y_{kp}) \leq e_k - s_k \quad \forall k \in M, \quad (12)$$

$$x_{ikp} \in \{0,1\} \quad \forall i \in N, p \in P, k \in M, \quad (13)$$

$$y_{kp} \geq 0 \quad \forall p \in P, k \in M, \quad (14)$$

Η αντικειμενική συνάρτηση δίνεται από την (8), είναι αντίστοιχη της αντικειμενικής (1) της πρώτης μοντελοποίησης και ελαχιστοποιεί τον συνολικό χρόνο αναμονής και χρόνο εξυπηρέτησης του κάθε πλοίου. Οι περιορισμοί (9) και (10) βεβαιώνουν ότι κάθε πλοίο εξυπηρετείται σε μία προβλήτα και κάθε προβλήτα μπορεί να εξυπηρετηθεί το πολύ ένα πλοίο τη φορά αντίστοιχα, όπως εξασφαλίζεται και στην πρώτη μοντελοποίηση με τους περιορισμούς (2) και (3). Ο περιορισμός (11) εξασφαλίζει ότι πρώτα γίνεται διαθέσιμη η προβλήτα  $k$  και μετά εξυπηρετείται το πλοίο με σειρά προτεραιότητας  $p$  και ο περιορισμός (12) ότι η εξυπηρέτηση ενός πλοίου πραγματοποιείται ανάμεσα στο χρονικό παράθυρο λειτουργίας  $[e_k, s_k]$  της προβλήτας. Οι περιορισμοί (11) και (12) είναι αντίστοιχοι των περιορισμών (4) και (5), που παρουσιάζονται παραπάνω. Οι περιορισμοί (13) και (14) δηλώνουν ότι η μεταβλητή  $x_{ikp}$  παίρνει δυαδικές τιμές και η  $y_{kp}$  δηλώνει ότι ο χρόνος αδράνειας της προβλήτας είναι μη αρνητικός.

## 2.2 Περίπτωση διακριτού χρόνου στο πρόβλημα ανάθεσης πλοίων

Παρακάτω παρουσιάζεται ένα μοντέλο προγραμματισμού πλοίων σε προβλήτες όπου ο χρόνος τώρα είναι διακριτός και έχει τη μορφή ενός δείκτη στο πρόβλημα. Αναλυτικά, η μορφοποίηση έχει τα παρακάτω χαρακτηριστικά:

Δείκτες

$i \in N$ ( $ N  = n$ )	Ο αριθμός των πλοίων
$j \in M$ ( $ M  = m$ )	Ο αριθμός των προβλητών
$t, p \in F$ ( $ F  = f$ )	Δηλώνει την χρονική περίοδο
$l \in N$ ( $ N  = n$ )	Ο αριθμός των πλοίων με $l \neq i$
$q \in F_{(t)}$ ( $F_{(t)} = \{q \in F : q < t\}$ )	Η χρονική στιγμή εξυπηρέτησης πριν από την $t$
$T$	Ο συνολικός χρονικός ορίζοντας

Δεδομένα

$h_{ij}$	Ο χρόνος εξυπηρέτησης του πλοίου $i$ στην προβλήτα $j$
$a_i$	Ο χρόνος άφιξης του πλοίου $i$
$s_j$	Χρονική στιγμή κατά την οποία η προβλήτα $j$ γίνεται διαθέσιμη
$e_j$	Χρονική στιγμή κατά την οποία η προβλήτα $j$ σταματάει να λειτουργεί
$M$	Μεγάλος σταθερός αριθμός

Μεταβλητές απόφασης

$X_{ijt}$	Ισούται με 1 αν το πλοίο $i$ πάει στην προβλήτα $j$ την χρονική στιγμή $t$ , αλλιώς 0.
$Y_{ijt}$	Ισούται με 1 αν το πλοίο $i$ εξυπηρετηθεί στην προβλήτα $j$ την χρονική στιγμή $t$ , αλλιώς 0.
$I_{jt}$	Ανενεργός χρόνος στην προβλήτα $j$ μεταξύ της ολοκλήρωσης της εξυπηρέτησης την χρονική στιγμή $t$ και την έναρξη της εξυπηρέτησης την $t'$ χρονική στιγμή.

Η μοντελοποίηση του προβλήματος είναι η εξής:



$$\text{Min} \sum_{i \in N} \sum_{j \in M} \sum_{t \in F} (h_{ij} + s_j - a_i) X_{ijt} + \sum_{j \in M} \sum_{t \in F} I_{jt} \quad (15)$$

Περιορισμοί

$$\sum_{j \in M} \sum_{t \in F} X_{ijt} = 1 \quad \forall i \in N, (16)$$

$$\sum_{j \in M} Y_{ijt} \leq 1 \quad \forall i \in N, t \in F, (17)$$

$$\sum_{i \in N} Y_{ijt} \leq 1 \quad \forall j \in M, t \in F, (18)$$

$$\sum_{i \in N} (a_i - s_j) X_{ijt} - \sum_{q \in F(t)} (I_{jq} + \sum_{l \in N} h_{lj} X_{ljq}) \leq I_{jt} \quad \forall j \in M, t \in F, (19)$$

$$\sum_{i \in N} \sum_{t \in F} (h_{ij} X_{ijt} + I_{jt}) \leq e_j - s_j \quad \forall j \in M, (20)$$

$$\sum_{i \in N} \sum_{t < a_i \in F} \sum_{j \in M} X_{ijt} + Y_{ijt} = 0 \quad (21)$$

$$\sum_{p=0}^{h_{ij}} Y_{ijt+p} - X_{ijt} (h_{ij} + M) + M \geq 0 \quad \forall i \in N, j \in M, t < T - h_{ij} \in F, (22)$$

$$- \sum_{p=0}^{h_{ij}} Y_{ijt+p} + X_{ijt} (h_{ij} - M) + M \geq 0 \quad \forall i \in N, j \in M, t < T - h_{ij} \in F, (23)$$

$$Y_{ijt} - X_{ijt} \geq 0 \quad \forall i \in N, j \in M, t \in F, (24)$$

$$X_{ijt} + Y_{ijt-1} - Y_{ijt} \geq 0 \quad \forall i \in N, j \in M, t > 1 \in F, (25)$$

$$X_{ijt}, Y_{ijt} \in \{0,1\} \quad \forall i \in N, j \in M, t \in F, (26)$$

$$I_{jt} \geq 0 \quad \forall j \in M, t \in F, (27)$$

Η αντικειμενική συνάρτηση (15) ελαχιστοποιεί τον συνολικό χρόνο εξυπηρέτησης (χρόνος αναμονής και χρόνος ξεφορτώματος) των πλοίων όπως, αντίστοιχα και η (1) στη πρώτη μοντελοποίηση που παρουσιάζεται στην ενότητα 2.1. Ο περιορισμός (16), ο οποίος αντιστοιχεί στον περιορισμό (2), δηλώνει ότι κάθε πλοίο ανά πάσα χρονική στιγμή δεν μπορεί να εξυπηρετηθεί σε περισσότερες από μια προβλήτες και ο περιορισμός (17) είναι συμπληρωματικός του (16) και εγγυάται ότι ένα πλοίο  $i$  μια χρονική στιγμή  $t$  μπορεί να εξυπηρετείται το πολύ σε μια προβλήτα  $j$ . Αντίστοιχος του περιορισμού (3) είναι ο (18) και εξασφαλίζει ότι τα πλοία  $i$  που εξυπηρετούνται στην προβλήτα  $j$  τη χρονική στιγμή  $t$  δεν μπορεί να είναι περισσότερα από ένα. Με τον περιορισμό (19) εξασφαλίζεται ότι για να εξυπηρετηθεί ένα πλοίο τη χρονική στιγμή  $t$  θα πρέπει πρώτα να γίνει διαθέσιμη η προβλήτα  $j$  και με τον (20) δηλώνεται ότι η εξυπηρέτηση των πλοίων πρέπει να πραγματοποιηθεί στο χρονικό παράθυρο κατά το οποίο η προβλήτα λειτουργεί, όπως επιτυγχάνεται και στην πρώτη μοντελοποίηση με τους περιορισμούς (4) και (5) αντίστοιχα. Έκτος από του παραπάνω περιορισμούς είναι αναγκαίο να προστεθούν ακόμα πέντε περιορισμοί. Ο περιορισμός (21) εγγυάται ότι δεν είναι δυνατόν να πραγματοποιηθεί η εξυπηρέτηση ενός πλοίου στην προβλήτα πριν από τη χρονική στιγμή άφιξης αυτού. Με τους περιορισμούς (22) και (23) εξασφαλίζεται ότι όταν ανατεθεί ένα πλοίο  $i$  στη προβλήτα  $j$  θα πρέπει να εξυπηρετείται για  $h_{ij}$ . Ικανοποίηση της ύπαρξης εξυπηρέτησης επιτυγχάνεται με τον περιορισμό (24) και με τον περιορισμό (25) γίνεται ορισμός της αρχής της εξυπηρέτησης του πλοίου. Τέλος, οι περιορισμοί (26) και (27) δηλώνουν ότι οι μεταβλητές  $X_{ijt}$  και  $Y_{ijt}$

παίρνουν δυαδικές τιμές και η μεταβλητή  $I_{jt}$  δηλώνει ότι ο χρόνος αδράνειας της προβλήτας είναι μη αρνητικός.

### **2.3 Γενικά συμπεράσματα από την παρουσίαση των τριών μοντελοποιήσεων**

Συμπερασματικά, οι δύο αρχικές μοντελοποιήσεις που παρουσιάζονται στην ενότητα 2.1 του κεφαλαίου, δείχνουν με ποια σειρά θα ανατεθούν τα πλοία στις προβλήτες που είναι διαθέσιμες προκειμένου να επιτευχθεί η αποτελεσματικότερη εξυπηρέτησή τους ενώ, η τρίτη μοντελοποίηση που αναπτύσσεται στα πλαίσια της παρούσας διπλωματικής εργασίας στην ενότητα 2.2, παρουσιάζει την σειρά ανάθεσης των πλοίων στις προβλήτες αλλά και τη χρονική διάρκεια που θα παραμείνουν σε αυτές μέχρι να εξυπηρετηθούν. Στην επόμενη ενότητα, οι τρεις μοντελοποιήσεις επιλύονται και γίνεται σύγκριση των αποτελεσμάτων τους.

### 3. ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Τα μοντέλα που παρουσιάζονται στο κεφάλαιο 2 επιλύονται αριθμητικά σε αυτήν την ενότητα. Η επίλυση όλων των παρακάτω παραδειγμάτων έγινε στο εργαστήριο Οργάνωσης Παραγωγής και Βιομηχανικής Διοίκησης του τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας. Τα χαρακτηριστικά του υπολογιστή που χρησιμοποιήθηκε ήταν τα εξής:

- Επεξεργαστής: Intel(R) Core(TM) i5-3330 CPU @3.00GHz,
- Εγκατεστημένη μνήμη: 16.00 GB,
- Λογισμικό: Windows 7 Professional 64-bit.

Χρησιμοποιήθηκαν οι βιβλιοθήκες CPLEX ILOG IBM σε γλώσσα προγραμματισμού C++ και σε γραφικό περιβάλλον Microsoft Visual Studio 2010.

#### 3.1 Παρουσίαση παραδειγμάτων

Στις τρεις μοντελοποιήσεις δεν υπάρχει αντιστοίχιση του χρόνου άφιξης με το χρονικό διάστημα  $t$  και έτσι, υπάρχει ένα σύνολο λύσεων μέσα από το οποίο μπορούν να επιλέξουν. Ειδικότερα, στη μοντελοποίηση με το διακριτό χρόνο υπάρχει περίπτωση μια προβλήτα  $j$  να είναι άδεια και το πλοίο  $i$  να περιμένει χωρίς να εξυπηρετείται. Πολλαπλασιάζοντας τους όρους  $(i+1)$ ,  $(j+1)$  και  $(t+1)$  με τη μεταβλητή απόφασης  $X_{ijt}$  το πλοίο αρχίζει να εξυπηρετείται όσο πιο γρήγορα γίνεται, δηλαδή παίρνει τη τιμή 1 με όσο το δυνατό πιο μικρό  $t$ . Όμως, πρωταρχικός στόχος είναι η ελαχιστοποίηση των όρων  $(h_{ij} + s_j - a_i)X_{ijt}$  και  $I_{jt}$ , οπότε πολλαπλασιάζουμε τους όρους  $(i+1)$ ,  $(j+1)$  και  $(t+1)$  με έναν συντελεστή βάρους έτσι ώστε να δίνεται λιγότερο βάρος στην ελαχιστοποίηση των τελευταίων.

Παρακάτω παρουσιάζονται για τα τρία μοντέλα, που αναφέρθηκαν στις ενότητες 2.1 και 2.2 του κεφαλαίου 2, ορισμένα από τα αριθμητικά δεδομένα που χρησιμοποιήθηκαν. Αναλυτικότερα, οι τρεις μοντελοποιήσεις επιλύθηκαν 21 φορές η καθεμία. Τα παραδείγματα 1-21 αντιστοιχούν το καθένα σε έναν καθορισμένο αριθμό πλοίων ( $N$ ) και προβλητών ( $K$ ), με  $(N,K)$  να παίρνουν τις τιμές που δηλώνονται στον Πίνακα 1.

	Αριθμός πλοίων (N)	Αριθμός προβλητών (K)
Παράδειγμα 1	12	3
Παράδειγμα 2	25	3
Παράδειγμα 3	25	6
Παράδειγμα 4	35	6
Παράδειγμα 5	40	8
Παράδειγμα 6	45	10
Παράδειγμα 7	50	4
Παράδειγμα 8	58	13
Παράδειγμα 9	60	12
Παράδειγμα 10	60	13
Παράδειγμα 11	60	15
Παράδειγμα 12	62	10
Παράδειγμα 13	62	12
Παράδειγμα 14	65	11
Παράδειγμα 15	65	15
Παράδειγμα 16	67	10
Παράδειγμα 17	68	10
Παράδειγμα 18	70	11
Παράδειγμα 19	100	30
Παράδειγμα 20	105	40
Παράδειγμα 21	108	40

Πίνακας 1. Αριθμός πλοίων και προβλητών που χρησιμοποιήθηκαν για την επίλυση των τριών μοντέλων.

### 3.2 Παρουσίαση αριθμητικών αποτελεσμάτων

Στην ενότητα αυτή παρουσιάζονται τα αριθμητικά αποτελέσματα από την επίλυση των τριών μοντελοποιήσεων με τη χρήση της `crplex`. Η 1<sup>η</sup> μοντελοποίηση αντιστοιχεί στη μορφοποίηση του προβλήματος ανάθεσης πλοίων με χρόνο συνεχή, η 2<sup>η</sup> μοντελοποίηση στο βελτιωμένο πρόβλημα ανάθεσης με χρόνο συνεχή και η 3<sup>η</sup>, στη μορφοποίηση του προβλήματος με χρόνο διακριτό. Τα τρία προβλήματα αποδίδουν την ίδια τιμή της αντικειμενικής και διαφέρουν στον χρόνο επίλυσης τους. Τα αριθμητικά αποτελέσματα που προέκυψαν παρουσιάζονται αναλυτικά στον Πίνακα 2 και για τα 21 παραδείγματα που αναπτύχθηκαν.

Σύμφωνα με τα αποτελέσματα του Πίνακα 2, για τα παραδείγματα 1 έως 18 δεν υπάρχει σημαντική διαφορά στον χρόνο επίλυσης των τριών μοντελοποιήσεων. Στα παραδείγματα 19-21 το πρώτο μοντέλο δεν είναι δυνατό να επιλυθεί λόγω προβλημάτων μνήμης του υπολογιστή και οι άλλες δύο μοντελοποιήσεις δίνουν αποτελέσματα με την 2<sup>η</sup> μοντελοποίηση να είναι σε ποσοστό 0.98 καλύτερη από τη 3<sup>η</sup> στα παραδείγματα 19, 20 και 21.

Στον Πίνακα 3 γίνεται σύγκριση στους χρόνους επίλυσης της 2<sup>ης</sup> και 3<sup>ης</sup> μοντελοποίησης και παρουσιάζονται οι μικρότερες και μεγαλύτερες διαφορές στους χρόνους που χρειάζονται για την απόδοση της λύσης. Παρατηρείται ότι, και στα 21 παραδείγματα, παρουσιάζεται αύξηση του χρόνου επίλυσης του 3<sup>ου</sup> σε σύγκριση με αυτόν του 2<sup>ου</sup>. Επιπλέον, από τις μικρότερες και μεγαλύτερες διαφορές, που παρουσιάζονται στον Πίνακα 3, είναι το ποσοστό 0.88 και 0.992 αντίστοιχα, όπου και εδώ ο χρόνος επίλυσης της 3<sup>ης</sup> μοντελοποίησης αυξάνεται σε σχέση με τη 2<sup>η</sup>. Συμπερασματικά, με την αύξηση του αριθμού των πλοίων και των προβλητών, που εισάγονται για την επίλυση των δύο μοντελοποιήσεων, προκύπτει ότι το 2<sup>ο</sup> μοντέλο ανάθεσης είναι καλύτερο ως προς τον χρόνο επίλυσης του σε σχέση με το 3<sup>ο</sup> μοντέλο διακριτού χρόνου.

Στον Πίνακα 4, παρουσιάζονται οι διαφορές που εμφανίζονται στους χρόνους επίλυσης της 3<sup>ης</sup> μοντελοποίησης συγκριτικά με την 1<sup>η</sup>. Συγκεκριμένα, στα παραδείγματα 1 έως 18, οι δύο μοντελοποιήσεις αποδίδουν το αποτέλεσμα σε λίγα μόλις δευτερόλεπτα, με την 1<sup>η</sup> να είναι σε ελάχιστο βαθμό καλύτερη στον χρόνο επίλυσης. Η μικρότερη διαφορά είναι σε ποσοστό 0.863 και συναντάται στο παράδειγμα 18, όπου παρατηρείται μια μικρή αύξηση του χρόνου επίλυσης της 3<sup>ης</sup> μοντελοποίησης. Για μεγαλύτερο αριθμό πλοίων και προβλητών είναι εμφανές ότι, μόνο η 3<sup>η</sup> μοντελοποίηση αποδίδει λύση και σε χρονικό διάστημα ορισμένων λεπτών.

	1 <sup>η</sup> μοντελοποίηση	2 <sup>η</sup> μοντελοποίηση	3 <sup>η</sup> μοντελοποίηση	
	Χρόνος επίλυσης (sec.)			Τιμή αντικειμενικής συνάρτησης
Παράδειγμα 1 (12,3)	0.03	0.03	0.27	67
Παράδειγμα 2 (25,3)	0.05	0.05	2.37	201
Παράδειγμα 3 (25,6)	0.05	0.05	4.90	92
Παράδειγμα 4 (35,6)	0.06	0.08	6.37	352
Παράδειγμα 5 (40,8)	0.09	0.11	12.51	331
Παράδειγμα 6 (45-10)	0.11	0.17	17.32	232
Παράδειγμα 7 (50,4)	0.75	0.09	8.47	426
Παράδειγμα 8 (58-13)	0.33	0.37	25.75	250
Παράδειγμα 9 (60-12)	0.66	0.34	17.33	250
Παράδειγμα 10 (60,13)	2.40	0.37	19.47	250
Παράδειγμα 11 (60,15)	1.64	2.50	35.80	209
Παράδειγμα 12 (62,10)	0.55	0.31	26.47	260
Παράδειγμα 13 (62,12)	0.86	0.38	40.82	207
Παράδειγμα 14 (65,11)	0.42	0.39	35.58	286
Παράδειγμα 15 (65,15)	3.39	0.53	47.49	250
Παράδειγμα 16 (67,10)	0.33	0.34	23.56	281
Παράδειγμα 17 (68,10)	1.11	0.36	20.28	238
Παράδειγμα 18 (70,11)	3.74	0.45	27.27	305
Παράδειγμα 19 (100,30)	-	4.16	553.23	330
Παράδειγμα 20 (105,40)	-	5.27	146.06	324
Παράδειγμα 21 (108,40)	-	6.76	541.46	335
<b>Μέση τιμή</b>				<b>244.81</b>

Πίνακας 2. Παρουσίαση αριθμητικών παραδειγμάτων για τις τρεις μοντελοποιήσεις.

Κεφάλαιο 3: Αριθμητικά αποτελέσματα

Αριθμητικό παράδειγμα	Χρόνος επίλυσης της 2 <sup>ης</sup> μοντελοποίησης (CPU <sub>2</sub> )	Χρόνος επίλυσης της 3 <sup>ης</sup> μοντελοποίησης (CPU <sub>3</sub> )	Διαφορά (%)
1	0.03	0.27	-0.88
2	0.05	2.37	-0.97
3	0.05	4.90	-0.989
4	0.08	6.37	-0.987
5	0.11	12.51	-0.991
6	0.17	17.32	-0.99
7	0.09	8.47	-0.989
8	0.37	25.75	-0.986
9	0.34	17.33	-0.98
10	0.37	19.47	-0.98
11	2.50	35.80	-0.93
12	0.31	26.47	-0.988
13	0.38	40.82	-0.99
14	0.39	35.58	-0.989
15	0.53	47.49	-0.988
16	0.34	23.56	-0.986
17	0.36	20.28	-0.982
18	0.45	27.27	-0.983
19	4.16	553.23	-0.992
20	5.27	146.06	-0.964
21	6.76	541.46	-0.988
<b>Μέση τιμή</b>			<b>-0.977</b>

Πίνακας 3. Σύγκριση χρόνων επίλυσης μεταξύ 2ης και 3ης μοντελοποίησης με  $(CPU_2 - CPU_3)/CPU_3$ .



Αριθμητικό παράδειγμα	Χρόνος επίλυσης της 1 <sup>ης</sup> μοντελοποίησης (CPU <sub>1</sub> )	Χρόνος επίλυσης της 3 <sup>ης</sup> μοντελοποίησης (CPU <sub>3</sub> )	Διαφορά (%)
1	0.03	0.27	-0.888
2	0.05	2.37	-0.979
3	0.05	4.90	-0.989
4	0.06	6.37	-0.99
5	0.09	12.51	-0.993
6	0.11	17.32	-0.994
7	0.75	8.47	-0.912
8	0.33	25.75	-0.987
9	0.66	17.33	-0.962
10	2.40	19.47	-0.877
11	1.64	35.80	-0.954
12	0.55	26.47	-0.979
13	0.86	40.82	-0.979
14	0.42	35.58	-0.988
15	3.39	47.49	-0.928
16	0.33	23.56	-0.986
17	1.11	20.28	-0.945
18	3.74	27.27	-0.863
19	-	553.23	-
20	-	146.06	-
21	-	541.46	-
<b>Μέση τιμή</b>			<b>-0.818</b>

Πίνακας 4. Σύγκριση χρόνων επίλυσης μεταξύ 1ης και 3ης μοντελοποίησης με  $(CPU_1 - CPU_3)/CPU_3$ .

Αριθμητικό παράδειγμα	Χρόνος επίλυσης της 1 <sup>ης</sup> μοντελοποίησης (CPU <sub>1</sub> )	Χρόνος επίλυσης της 2 <sup>ης</sup> μοντελοποίησης (CPU <sub>2</sub> )	Διαφορά (%)
1	0.03	0.03	0
2	0.05	0.05	0
3	0.05	0.05	0
4	0.06	0.08	-0.25
5	0.09	0.11	-0.182
6	0.11	0.17	-0.353
7	0.75	0.09	7.33
8	0.33	0.37	-0.108
9	0.66	0.34	0.941
10	2.40	0.37	5.486
11	1.64	2.50	-0.344
12	0.55	0.31	0.774
13	0.86	0.38	1.263
14	0.42	0.39	0.07
15	3.39	0.53	5.396
16	0.33	0.34	-0.03
17	1.11	0.36	2.08
18	3.74	0.45	7.31
19	-	4.16	-
20	-	5.27	-
21	-	6.76	-
<b>Μέση τιμή</b>			<b>1.39</b>

Πίνακας 5. Σύγκριση χρόνων επίλυσης μεταξύ 1ης και 2ης μοντελοποίησης με  $(CPU_1 - CPU_2) / CPU_2$ .

Από τη σύγκριση της 1<sup>ης</sup> και της 2<sup>ης</sup> μοντελοποίησης προκύπτουν τα αποτελέσματα των διαφορών τους, που παρουσιάζονται στον Πίνακα 5. Οι μεγαλύτερες διαφορές στους χρόνους είναι σε ποσοστά 7.33, 5.486, 7.31 και εμφανίζονται στα παραδείγματα 7,10 και 18 όπου, αυξάνεται ο χρόνος επίλυσης του 1<sup>ου</sup> μοντέλου. Από τις μικρότερες διαφορές που παρατηρούνται είναι σε ποσοστό 0 και 0.03 με μια μικρή αύξηση στον χρόνο επίλυσης του 2<sup>ου</sup> μοντέλου. Τέλος, στα τελευταία παραδείγματα 19-21 παρατηρείται ότι, είναι δυνατό να επιλυθεί μόνο η 2<sup>η</sup> μοντελοποίηση και σε χρονικό διάστημα ορισμένων δευτερολέπτων.

Συμπερασματικά, για έναν ορισμένο αριθμό πλοίων και προβλητών τα τρία μοντέλα είναι εξίσου ικανοποιητικά στους χρόνους επίλυσης τους για την απόδοση της λύσης αλλά, με την αύξηση των δεδομένων που εισάγονται, το 1<sup>ο</sup> μοντέλο είναι αδύνατον να επιλυθεί, ενώ το 2<sup>ο</sup> και 3<sup>ο</sup> μοντέλο λύνονται άμεσα.

### 3.3 Προσθήκη περιορισμών για την βελτίωση των τριών μοντελοποιήσεων (Valid inequalities)

Σε αυτή την ενότητα, παρουσιάζεται ένας επιπλέον περιορισμός, ο οποίος προστίθεται και στις τρεις μορφοποιήσεις της παρούσας εργασίας και σχετίζεται με τον χρόνο αναμονής των πλοίων (Waiting Time). Ο περιορισμός αυτός διαφοροποιείται αντίστοιχα για τις τρεις μοντελοποιήσεις και είναι ο εξής:

$$\sum_{j \in N} \sum_{m \in P(p)} (h_{jk}x_{jkm} + y_{jkm}) + \sum_{i \in N} (s_k - a_i)x_{ikp} \leq WT \quad \forall k \in M, (p > 1) \in P, (28)$$

$$\sum_{j \in N} \sum_{m \in P(p)} (h_{jk}x_{jkm} + y_{mk}) + \sum_{i \in N} (s_k - a_i)x_{ikp} \leq WT \quad \forall k \in M, (p > 1) \in P, (29)$$

$$\sum_{k \in N} \sum_{m \in F(t)} (h_{kj}x_{kjm} + I_{jm}) + \sum_{i \in N} (s_j - a_i)x_{ijt} \leq WT \quad \forall j \in M, (t > 1) \in F, (30)$$

Οι περιορισμοί (28) και (29) δηλώνουν ότι, ένα πλοίο, που θα έρθει να εξυπηρετηθεί με σειρά προτεραιότητας  $p$  σε μία προβλήτα  $k$ , θα πρέπει να περιμένει πρώτα να εξυπηρετηθεί το πλοίο, που υπάρχει με μικρότερη σειρά προτεραιότητας στην ίδια προβλήτα  $k$ . Επιπλέον, ο

περιορισμός (30) εξασφαλίζει ότι, όταν η προβλήτα  $j$  είναι κατειλημμένη και εξυπηρετεί ένα πλοίο  $k$  κάποια συγκεκριμένη χρονική στιγμή  $m$ , τότε το πλοίο  $i$  που θα φτάσει κάποια από τις επόμενες χρονικές στιγμές  $t$  στην ίδια προβλήτα, θα πρέπει να περιμένει να εξυπηρετηθεί το ήδη υπάρχον πλοίο.

Στη συνέχεια, στους Πίνακες 6,7 και 8 παρουσιάζονται τα αριθμητικά αποτελέσματα και οι χρόνοι επίλυσης για τις τρεις μορφοποιήσεις, που αναπτύχθηκαν στις ενότητες 2.1 και 2.2, με την προσθήκη του περιορισμού (28),(29) και (30), αντίστοιχα. Για την 1<sup>η</sup> μοντελοποίηση, στα παραδείγματα 1-9 του Πίνακα 6, η διαφορά στον χρόνο επίλυσης είναι αμελητέα. Στα παραδείγματα 7,10,13,14,15 και 18 παρατηρείται μείωση στον χρόνο που χρειάζεται να επιλυθεί η 1<sup>η</sup> μορφοποίηση με την προσθήκη του περιορισμού (28). Το μεγαλύτερο ποσοστό μείωσης του χρόνου επίλυσης που συναντάται είναι 7.33 και στα παραδείγματα 1 και 2 δε παρατηρείται κάποια διαφορά. Αύξηση του χρόνου επίλυσης παρουσιάζεται στα παραδείγματα 3,4,5,6,8,9,11,12,16 και 17 ενώ, τα υπόλοιπα παραδείγματα που αναπτύσσονται είναι αδύνατον να επιλυθούν. Στον Πίνακα 7, οι χρόνοι, που απαιτούνται για την επίλυση της 2<sup>ης</sup> μοντελοποίησης με τη προσθήκη του περιορισμού (29), αυξάνονται αλλά δεν διαφέρουν σημαντικά από τους αρχικούς χρόνους της. Οι σημαντικότερες διαφορές που παρατηρούνται είναι στα παραδείγματα 20 και 21 σε ποσοστό 0.9 και 0.854 αντίστοιχα. Τέλος, για την 3<sup>η</sup> μοντελοποίηση, σημαντικές διαφορές παρουσιάζονται στα παραδείγματα 14 και 20 του Πίνακα 8, όπου παρατηρείται αύξηση του χρόνου επίλυσης της 3<sup>ης</sup> μοντελοποίησης με τον περιορισμό (30) και στα παραδείγματα 1 και 5 πραγματοποιείται μείωση αυτού του χρόνου.

Κεφάλαιο 3: Αριθμητικά αποτελέσματα

	1 <sup>η</sup> μοντελοποίηση	1 <sup>η</sup> μοντελοποίηση (Με περιορισμό (22))	
	Χρόνος επίλυσης (sec.)		Διαφορά (%)
<b>Παράδειγμα 1</b> (12,3)	0.03	0.03	0
<b>Παράδειγμα 2</b> (25,3)	0.05	0.05	0
<b>Παράδειγμα 3</b> (25,6)	0.05	0.06	-0.167
<b>Παράδειγμα 4</b> (35,6)	0.06	0.08	-0.25
<b>Παράδειγμα 5</b> (40,8)	0.09	0.14	-0.357
<b>Παράδειγμα 6</b> (45,10)	0.11	0.33	-0.667
<b>Παράδειγμα 7</b> (50,4)	0.75	0.09	7.33
<b>Παράδειγμα 8</b> (58,13)	0.33	1.06	-0.688
<b>Παράδειγμα 9</b> (60,12)	0.66	1.08	-0.388
<b>Παράδειγμα 10</b> (60,13)	2.40	1.30	0.846
<b>Παράδειγμα 11</b> (60,15)	1.64	3.15	-0.48
<b>Παράδειγμα 12</b> (62,10)	0.55	0.91	-0.395
<b>Παράδειγμα 13</b> (62,12)	0.86	0.79	0.08
<b>Παράδειγμα 14</b> (65,11)	0.42	0.37	0.135
<b>Παράδειγμα 15</b> (65,15)	3.39	2.85	0.189
<b>Παράδειγμα 16</b> (67,10)	0.33	1.08	-0.694
<b>Παράδειγμα 17</b> (68,10)	1.11	1.20	-0.07
<b>Παράδειγμα 18</b> (70,11)	3.74	3.03	0.234
<b>Παράδειγμα 19</b> (100,30)	-	-	-
<b>Παράδειγμα 20</b> (105,40)	-	-	-
<b>Παράδειγμα 21</b> (108,40)	-	-	-
<b>Μέση τιμή</b>			<b>0.258</b>

Πίνακας 6. Παρουσίαση αριθμητικών αποτελεσμάτων για τη 1η μοντελοποίηση με την προσθήκη του περιορισμού (28).

	2 <sup>η</sup> μοντελοποίηση	2 <sup>η</sup> μοντελοποίηση (Με περιορισμό (23))	
	Χρόνος επίλυσης (sec.)		Τιμή αντικειμενικής συνάρτησης
Παράδειγμα 1 (12,3)	0.03	0.05	-0.4
Παράδειγμα 2 (25,3)	0.05	0.09	-0.44
Παράδειγμα 3 (25,6)	0.05	0.09	-0.44
Παράδειγμα 4 (35,6)	0.08	0.11	-0.272
Παράδειγμα 5 (40,8)	0.11	0.23	-0.521
Παράδειγμα 6 (45,10)	0.17	0.58	-0.7
Παράδειγμα 7 (50,4)	0.09	0.14	-0.357
Παράδειγμα 8 (58,13)	0.37	1.48	-0.75
Παράδειγμα 9 (60,12)	0.34	1.64	-0.793
Παράδειγμα 10 (60,13)	0.37	1.89	-0.8
Παράδειγμα 11 (60,15)	2.50	4.27	-0.414
Παράδειγμα 12 (62,10)	0.31	1.53	-0.797
Παράδειγμα 13 (62,12)	0.38	1.08	-0.648
Παράδειγμα 14 (65,11)	0.39	1.70	-0.77
Παράδειγμα 15 (65,15)	0.53	2.54	-0.79
Παράδειγμα 16 (67,10)	0.34	1.72	-0.8
Παράδειγμα 17 (68,10)	0.36	1.39	-0.741
Παράδειγμα 18 (70,11)	0.45	2.14	-0.789
Παράδειγμα 19 (100,30)	4.16	19.53	-0.787
Παράδειγμα 20 (105,40)	5.27	53.90	-0.9
Παράδειγμα 21 (108,40)	6.76	46.35	-0.854
<b>Μέση τιμή</b>			<b>0.655</b>

Πίνακας 7. Παρουσίαση αριθμητικών αποτελεσμάτων για τη 2η μοντελοποίηση με την προσθήκη του περιορισμού (29).

	3 <sup>η</sup> μοντελοποίηση	3 <sup>η</sup> μοντελοποίηση (Με περιορισμό (24))	
	Χρόνος επίλυσης (sec.)		Τιμή αντικειμενικής συνάρτησης
Παράδειγμα 1 (12,3)	0.27	0.26	0.04
Παράδειγμα 2 (25,3)	2.37	2.5	-0.052
Παράδειγμα 3 (25,6)	4.90	7.22	-0.32
Παράδειγμα 4 (35,6)	6.37	6.38	-0.002
Παράδειγμα 5 (40,8)	12.51	12.40	0.008
Παράδειγμα 6 (45,10)	17.32	27.52	-0.37
Παράδειγμα 7 (50,4)	8.47	8.47	0
Παράδειγμα 8 (58,13)	25.75	49.34	-0.478
Παράδειγμα 9 (60,12)	17.33	37.14	-0.53
Παράδειγμα 10 (60,13)	19.47	36.11	-0.46
Παράδειγμα 11 (60,15)	35.80	62.87	-0.43
Παράδειγμα 12 (62,10)	26.47	32.53	-0.186
Παράδειγμα 13 (62,12)	40.82	47.88	-0.147
Παράδειγμα 14 (65,11)	35.58	122.34	-0.709
Παράδειγμα 15 (65,15)	47.49	79.93	-0.405
Παράδειγμα 16 (67,10)	23.56	58.36	-0.596
Παράδειγμα 17 (68,10)	20.28	65.58	-0.69
Παράδειγμα 18 (70,11)	27.27	60.45	-0.548
Παράδειγμα 19 (100,30)	553.23	554.74	-0.003
Παράδειγμα 20 (105,40)	146.06	627.25	-0.767
Παράδειγμα 21 (108,40)	541.46	-	-
Μέση τιμή			<b>-6.645</b>

Πίνακας 8. Παρουσίαση αριθμητικών αποτελεσμάτων για τη 3η μοντελοποίηση με την προσθήκη του περιορισμού (30).

Προσθέτοντας στα τρία προβλήματα τον αντίστοιχο περιορισμό (valid inequality) υπάρχει η περίπτωση, ο χώρος λύσεων να μειώνεται με αποτέλεσμα τη μείωση των ακραίων σημείων που αντιπροσωπεύουν τις εφικτές λύσεις και κατ' επέκταση του χρόνου επίλυσης των προβλημάτων. Επιπλέον, είναι δυνατόν με τη προσθήκη του περιορισμού αυτού να αυξάνεται ο χρόνος επίλυσης των προβλημάτων διότι δημιουργούνται περισσότερα ακραία σημεία και επομένως να αυξάνονται οι εφικτές τους λύσεις.

#### **3.4 Υπολογισμός και συγκριτικά αποτελέσματα του χρόνου αναμονής**

Στον παρακάτω πίνακα, παρουσιάζονται οι τιμές που υπολογίστηκαν για τον χρόνο αναμονής (WT) και για τα 21 παραδείγματα που αναπτύχθηκαν στην ενότητα 3.1. Για τις τρεις μοντελοποιήσεις αυτές οι τιμές είναι ίδιες.

Οι παρακάτω χρόνοι αναμονής δεν είναι πραγματικοί και έχουν υπολογιστεί για το κάθε παράδειγμα ξεχωριστά, τοποθετώντας τα πλοία στις προβλήτες ανάλογα με τον χρόνο άφιξης τους και τον χρόνο που γίνονται διαθέσιμες οι προβλήτες για πρώτη φορά. Έτσι, τα επόμενα πλοία που θα εξυπηρετηθούν θα περιμένουν όσο χρειαστεί για να τελειώσουν τα πρώτα πλοία που ανατέθηκαν στις προβλήτες. Αυτή η διαδικασία πραγματοποιείται μέχρι να εξυπηρετηθούν όλα τα πλοία και υπολογίζεται ο θεωρητικός συνολικός χρόνος αναμονής. Παρακάτω στους Πίνακες 10,11 και 12 παρουσιάζονται οι τιμές του χρόνου αναμονής που υπολογίστηκαν για τις τρεις μοντελοποιήσεις και γίνονται συγκρίσεις με τις θεωρητικές τιμές.



	Αριθμός πλοίων (N)	Αριθμός προβλητών (K)	Χρόνος αναμονής (WT)
Παράδειγμα 1	12	3	77
Παράδειγμα 2	25	3	185
Παράδειγμα 3	25	6	75
Παράδειγμα 4	35	6	303
Παράδειγμα 5	40	8	288
Παράδειγμα 6	45	10	223
Παράδειγμα 7	50	4	394
Παράδειγμα 8	58	13	244
Παράδειγμα 9	60	12	211
Παράδειγμα 10	60	13	195
Παράδειγμα 11	60	15	188
Παράδειγμα 12	62	10	230
Παράδειγμα 13	62	12	199
Παράδειγμα 14	65	11	263
Παράδειγμα 15	65	15	248
Παράδειγμα 16	67	10	248
Παράδειγμα 17	68	10	230
Παράδειγμα 18	70	11	286
Παράδειγμα 19	100	30	256
Παράδειγμα 20	105	40	216
Παράδειγμα 21	108	40	225

Πίνακας 9. Χρόνος αναμονής (WT) που χρησιμοποιήθηκε για τα παραδείγματα των τριών μοντελοποιήσεων.

1 <sup>η</sup> μοντελοποίηση	Θεωρητικός χρόνος αναμονής	Πραγματικός χρόνος αναμονής	Διαφορά (%)	Πραγματικός χρόνος αναμονής με περιορισμό (28)	Διαφορά (%)
Παράδειγμα 1	77	45	0.71	45	0.71
Παράδειγμα 2	185	126	0.47	126	0.47
Παράδειγμα 3	75	16	3.68	16	4.77
Παράδειγμα 4	303	236	0.28	236	0.28
Παράδειγμα 5	288	205	0.4	205	0.4
Παράδειγμα 6	223	90	1.47	90	1.47
Παράδειγμα 7	394	263	0.49	263	0.49
Παράδειγμα 8	244	71	2.43	70	2.48
Παράδειγμα 9	211	65	2.24	65	2.24
Παράδειγμα 10	195	65	2	65	2
Παράδειγμα 11	188	26	6.23	26	6.23
Παράδειγμα 12	230	67	2.43	67	2.43
Παράδειγμα 13	199	14	13.2	8	23.8
Παράδειγμα 14	263	72	2.65	72	2.65
Παράδειγμα 15	248	49	4.06	48	4.16
Παράδειγμα 16	248	73	2.39	73	2.39
Παράδειγμα 17	230	19	11.10	23	9
Παράδειγμα 18	286	74	2.86	74	2.86
Παράδειγμα 19	256	-	-	-	-
Παράδειγμα 20	216	-	-	-	-
Παράδειγμα 21	225	-	-	-	-

Πίνακας 10. Σύγκριση των πραγματικών με τους θεωρητικούς χρόνους αναμονής που υπολογίστηκαν για τη 1η μοντελοποίηση.

2 <sup>η</sup> μοντελοποίηση	Θεωρητικός χρόνος αναμονής	Πραγματικός χρόνος αναμονής	Διαφορά (%)	Πραγματικός χρόνος αναμονής με περιορισμό (29)	Διαφορά (%)
Παράδειγμα 1	77	45	0.71	45	0.71
Παράδειγμα 2	185	126	0.47	126	0.47
Παράδειγμα 3	75	16	3.68	16	4.77
Παράδειγμα 4	303	236	0.28	236	0.28
Παράδειγμα 5	288	205	0.4	205	0.4
Παράδειγμα 6	223	90	1.47	90	1.47
Παράδειγμα 7	394	263	0.49	263	0.49
Παράδειγμα 8	244	71	2.43	70	2.48
Παράδειγμα 9	211	65	2.24	65	2.24
Παράδειγμα 10	195	65	2	65	2
Παράδειγμα 11	188	26	6.23	26	6.23
Παράδειγμα 12	230	67	2.43	67	2.43
Παράδειγμα 13	199	14	13.2	8	23.8
Παράδειγμα 14	263	72	2.65	72	2.65
Παράδειγμα 15	248	49	4.06	48	4.16
Παράδειγμα 16	248	73	2.39	73	2.39
Παράδειγμα 17	230	19	11.10	23	9
Παράδειγμα 18	286	74	2.86	74	2.86
Παράδειγμα 19	256	108	1.37	108	1.37
Παράδειγμα 20	216	115	0.87	115	0.87
Παράδειγμα 21	225	118	0.90	118	0.90

Πίνακας 11. Σύγκριση των πραγματικών με τους θεωρητικούς χρόνους αναμονής που υπολογίστηκαν για τη 2η μοντελοποίηση.

3 <sup>η</sup> μοντελοποίηση	Θεωρητικός χρόνος αναμονής	Πραγματικός χρόνος αναμονής	Πραγματικός χρόνος αναμονής με περιορισμό (30)	Διαφορά (%)
Παράδειγμα 1	77	45	45	0.71
Παράδειγμα 2	185	126	126	0.47
Παράδειγμα 3	75	13	13	4.77
Παράδειγμα 4	303	236	236	0.28
Παράδειγμα 5	288	205	205	0.4
Παράδειγμα 6	223	90	90	1.47
Παράδειγμα 7	394	263	263	0.49
Παράδειγμα 8	244	74	74	2.29
Παράδειγμα 9	211	65	65	2.24
Παράδειγμα 10	195	65	65	2
Παράδειγμα 11	188	26	26	6.23
Παράδειγμα 12	230	67	67	2.43
Παράδειγμα 13	199	13	13	14.3
Παράδειγμα 14	263	72	72	2.65
Παράδειγμα 15	248	43	43	4.76
Παράδειγμα 16	248	73	73	2.39
Παράδειγμα 17	230	28	28	7.21
Παράδειγμα 18	286	74	74	2.86
Παράδειγμα 19	256	108	108	1.37
Παράδειγμα 20	216	115	115	0.87
Παράδειγμα 21	225	118	-	-

Πίνακας 12. Σύγκριση των πραγματικών με τους θεωρητικούς χρόνους αναμονής που υπολογίστηκαν για τη 3η μοντελοποίηση.

Για την 1<sup>η</sup> και 2<sup>η</sup> μοντελοποίηση, με την προσθήκη των περιορισμών (28) και (29) αντίστοιχα, παρατηρείται στα παραδείγματα 8,13 και 15 μείωση του πραγματικού χρόνου αναμονής. Αύξηση αυτού του χρόνου παρουσιάζεται μόνο στο παράδειγμα 17 ενώ, στα υπόλοιπα παραδείγματα επικρατεί ο ίδιος χρόνος αναμονής. Στη 3<sup>η</sup> μοντελοποίηση, με τη προσθήκη του περιορισμού (30) δεν αλλάζει ο πραγματικός χρόνος αναμονής. Συγκρίνοντας τους πραγματικούς χρόνους των τριών μοντελοποιήσεων προκύπτει ότι, για τα παραδείγματα 3 και 15 ο χρόνος αναμονής της 3<sup>ης</sup> μοντελοποίησης είναι μικρότερος από τους αντίστοιχους της 1<sup>ης</sup> και της 2<sup>ης</sup> και για τα παραδείγματα 8 και 17 είναι μεγαλύτερος. Στο παράδειγμα 13, αρχικά, ο χρόνος αναμονής της 3<sup>ης</sup> μοντελοποίησης είναι μικρότερος αλλά, με τη προσθήκη των περιορισμών (28) και (29) στις δύο πρώτες μοντελοποιήσεις ο χρόνος αναμονής μειώνεται σε σχέση με αυτόν της 3<sup>ης</sup> μοντελοποίησης.

### **3.5 Συμπεράσματα από τη παρουσίαση των αριθμητικών αποτελεσμάτων και συγκρίσεων**

Συμπερασματικά, η 1<sup>η</sup> και 2<sup>η</sup> μοντελοποίηση αποδίδουν σε μερικά παραδείγματα μικρότερο χρόνο αναμονής των πλοίων σε σχέση με τη 3<sup>η</sup> μοντελοποίηση. Τέλος, οι τρεις μοντελοποιήσεις δίνουν εξίσου ικανοποιητικά αποτελέσματα αλλά, με την προσθήκη των περιορισμών (28),(29) και (30) είναι φανερό ότι, μόνο η 2<sup>η</sup> μοντελοποίηση μπορεί να επιλυθεί και για τα 21 παραδείγματα που αναπτύχθηκαν και να αποδώσει λύση σε ορισμένα δευτερόλεπτα.

## 4. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκε η περιγραφή, η επίλυση και η σύγκριση τριών διαφορετικών μοντελοποιήσεων του προβλήματος ανάθεσης πλοίων σε προβλήτες. Το πρόβλημα ανάθεσης των πλοίων στις προβλήτες για την αποτελεσματικότερη εξυπηρέτηση τους αποτελεί ένα από τα βασικότερα προβλήματα που απασχολεί έναν τερματικό σταθμό. Για όλα τα παραδείγματα που αναπτύχθηκαν στην ενότητα 3, η δεύτερη και τρίτη μοντελοποίηση είναι σημαντικά αποτελεσματικές αποδίδοντας τη λύση από ορισμένα δευτερόλεπτα έως και μερικά λεπτά. Η πρώτη μοντελοποίηση δεν είναι δυνατόν να επιλυθεί με την εισαγωγή μεγαλύτερων δεδομένων. Επιπλέον, στα τρία μοντέλα προστέθηκε ακόμα ένας σημαντικός περιορισμός που σχετίζεται με τον χρόνο αναμονής των πλοίων με αποτέλεσμα, μόνο η δεύτερη μοντελοποίηση να αποδώσει λύση για όλα τα παραδείγματα κ σε χρόνο ορισμένων δευτερολέπτων. Εξίσου, όμως ικανοποιητικά αποτελέσματα έδωσε σε όλες τις περιπτώσεις και το τρίτο μοντέλο που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας και αποδίδει εκτός από τη σειρά και το χρόνο που θα παραμείνει το πλοίο στη προβλήτα μέχρι να εξυπηρετηθεί. Έτσι, οι δύο τελευταίες μοντελοποιήσεις αποτελούν έναν σωστό και γρήγορο τρόπο επίλυσης του προβλήματος ανάθεσης, με τη δεύτερη να είναι σε ελάχιστο βαθμό καλύτερη από τη τρίτη.

## ΠΑΡΑΡΤΗΜΑ

Παρακάτω παρουσιάζονται οι κώδικες που δημιουργήθηκαν για την επίλυση των τριών μοντελοποιήσεων.

*1<sup>η</sup> μοντελοποίηση (αριθμός πλοίων→12-αριθμός προβλητών→3)*

```
#include <ilcplex/ilocplex.h>
```

```
ILOSTLBEGIN
```

```
int i,p,k,l,q,n;
```

```
const int imax=12;
```

```
const int nmax=12;
```

```
const int kmax=3;
```

```
const int pmax=12;
```

```
const int qmax=12;
```

```
const int lmax=12;
```

```
//float capmax=100000;
```

```
//float capmin=0;
```

```
int lastl;
```

```
int
```

```
main (int argc, char **argv)
```

```
{
```

```
double H[imax][kmax];
```

```
double A[imax];
```

```
double S[kmax];
```

```
double B[kmax];
```

```
for (i=0;i<imax;i++){
```

```
    for (k=0;k<kmax;k++){
```

```
        H[i][k]=0;
```

```
    }  
}  
  
H[0][0]=1;  
H[0][1]=3;  
H[0][2]=1;  
H[1][0]=1;  
H[1][1]=1;  
H[1][2]=2;  
H[2][0]=2;  
H[2][1]=2;  
H[2][2]=2;  
H[3][0]=1;  
H[3][1]=1;  
H[3][2]=1;  
H[4][0]=4;  
H[4][1]=1;  
H[4][2]=1;  
  
H[5][0]=3;  
H[5][1]=2;  
H[5][2]=1;  
H[6][0]=1;  
H[6][1]=1;  
H[6][2]=1;  
H[7][0]=5;  
H[7][1]=5;
```



H[7][2]=4;

H[8][0]=8;

H[8][1]=8;

H[8][2]=3;

H[9][0]=1;

H[9][1]=6;

H[9][2]=1;

H[10][0]=1;

H[10][1]=1;

H[10][2]=1;

H[11][0]=7;

H[11][1]=7;

H[11][2]=5;

for (i=0;i<imax;i++){

    A[i]=0;

    }

A[0]=2;

A[1]=1;

A[2]=3;

A[3]=1;

A[4]=3;

A[5]=4;

A[6]=1;

A[7]=2;

A[8]=1;

```

A[9]=6;
A[10]=1;
A[11]=2;

for (k=0;k<kmax;k++){
    S[k]=0;
}

S[0]=6;
S[1]=6;
S[2]=6;

for (k=0;k<kmax;k++){
    B[k]=0;
}

B[0]=100;
B[1]=100;
B[2]=100;

IloEnv env;

try {

    IloModel model (env);

typedef IloArray<IloNumArray> IloNumMatrix2x2;
typedef IloArray<IloNumMatrix2x2> IloNumMatrix3x3;
typedef IloArray<IloNumMatrix3x3> IloNumMatrix4x4;

```

```

typedef IloArray<IloNumVarArray> IloNumVarMatrix2x2;

typedef IloArray<IloNumVarMatrix2x2> IloNumVarMatrix3x3;

typedef IloArray<IloNumVarMatrix3x3> IloNumVarMatrix4x4;

typedef IloArray<IloRangeArray> IloRangeMatrix2x2;

typedef IloArray<IloRangeMatrix2x2> IloRangeMatrix3x3;

typedef IloArray<IloRangeMatrix3x3> IloRangeMatrix4x4;

IloCplex cplex(env);

//----- METAVLHTH APOFASHS Y (continuous)-----

IloNumVarMatrix3x3 Yikp(env,0);

for (i=0;i<imax;i++){

    IloNumVarMatrix2x2 Ykp(env,0);

    for (k=0;k<kmax;k++){

        IloNumVarArray Yp(env,0);

        for (p=0;p<pmax;p++){

            char Quantite_Decharge[70];

            sprintf(Quantite_Decharge,"Yikp(i%d,k%d,p%d)",i,k,p);

            IloNumVar Y(env,0,100000,ILOFLOAT,Quantite_Decharge);

            Yp.add(Y);

        }

        Ykp.add(Yp);

    }

    Yikp.add(Ykp);
}

```

```

    }

cout<<1<<endl;

//----- METAVLHTH APOFASHS X (binary)-----

IloNumVarMatrix3x3 Xipk(env,0);

for (i=0;i<imax;i++){

    IloNumVarMatrix2x2 Xpk(env,0);

    for (p=0;p<pmax;p++){

        IloNumVarArray Xk(env,0);

        for (k=0;k<kmax;k++){

            char Chargement[70];

            sprintf(Chargement,"Xipk(i%d,p%d,k%d)",i,p,k);

            IloNumVar X(env,0,1,ILOINT,Chargement);

            Xk.add(X);

        }

        Xpk.add(Xk);

    }

    Xipk.add(Xpk);

}

cout<<2<<endl;

//-----

//-----PERIORISMOI-----

//-----

//----- PERIORISMOS 2 -----

```

```

IloRangeArray SumXi(env,0);

for (i=0;i<imax;i++){

    IloExpr expr(env,0);

    for (k=0;k<kmax;k++){

        for (p=0;p<pmax;p++){

            expr+=Xipk[i][p][k];

        }

    }

    char Quantite_Charge[60];

    sprintf(Quantite_Charge,"SumXi(i%d)",i);

    float LB=1,UB=1;

    IloRange SumX(env,LB,expr,UB,Quantite_Charge);

    model.add(SumX);

    SumXi.add(SumX);

    expr.end();

}

cout<<3<<endl;

//-----
//----- PERIORISMOS 3 -----

IloRangeMatrix2x2 SumXkp(env,0);

for (k=0;k<kmax;k++){

    IloRangeArray SumXp(env,0);

    for (p=0;p<pmax;p++){

        IloExpr expr(env);

        for (i=0;i<imax;i++){

            expr+=Xipk[i][p][k];


```

```

    }

    char Quantite_Decharge[60];

    sprintf(Quantite_Decharge,"SumXkp(k%d,p%d)",k,p);

    float LB=0,UB=1;

    IloRange SumX(env,LB,expr,UB,Quantite_Decharge);

    model.add(SumX);

    SumXp.add(SumX);

    expr.end();

}

SumXkp.add(SumXp);

}

cout<<<endl;

//-----
//----- PERIORISMOS 4 -----

IloRangeMatrix3x3 Y_4kp(env,0);

for (i=0;i<imax;i++){

    IloRangeMatrix2x2 Y_4kp(env,0);

    for (k=0;k<kmax;k++){

        IloRangeArray Y_4p(env,0);

        for (p=0;p<pmax;p++){

            IloExpr expr(env);

            IloExpr expr1(env);

            for (l=0;l<imax;l++){

                if(l!=i){

                    for (q=0;q<pmax;q++){

```

```

        if(q<p) {
            expr1+=H[l][k]*Xipk[l][q][k]+Yikp[l][k][q];
        }
    }
}

expr=expr1+Yikp[i][k][p]-(A[i]-S[k])*Xipk[i][p][k];

char Equoition_equilibre[60];

sprintf(Equoition_equilibre,"Y_4ikp(i%d,k%d,p%d)",i,k,p);

float LB=0,UB=IloInfinity;

IloRange Y_4(env,LB,expr,UB,Equoition_equilibre);

model.add(Y_4);

Y_4p.add(Y_4);

expr.end();

expr1.end();

}

Y_4kp.add(Y_4p);

}

Y_4ikp.add(Y_4kp);

}

```

```

cout<<5<<endl;

//-----
//----- PERIORISMOS 5 -----

IloRangeArray Y_7k(env,0);

for (k=0;k<kmax;k++){

    IloExpr expr(env);

        IloExpr expr1(env);

        for (i=0;i<imax;i++){

            for (p=0;p<pmax;p++){

                expr1+=-(H[i][k]*Xipk[i][p][k] + Yikp[i][k][p]);

            }

        }

        expr=(B[k]-S[k])+expr1;

        char Equoition_equilibre[60];

        sprintf(Equoition_equilibre,"Y_7k(k%d)",k);

        float LB=0,UB=IloInfinity;

        IloRange Y_7(env,LB,expr,UB,Equoition_equilibre);

        model.add(Y_7);

        Y_7k.add(Y_7);

        expr.end();

        expr1.end();

```



```

    }

cout<<7<<endl;

//-----
//-----
//-----ANTIKEIMENIKH-----
//-----

IloExpr expr1(env);

for (k=0;k<kmax;k++){
    for (i=0;i<imax;i++){
        for (p=0;p<pmax;p++){
            expr1+=(H[i][k] + S[k] - A[i])*Xipk[i][p][k];
        }
    }
}

for (k=0;k<kmax;k++){
    for (i=0;i<imax;i++){
        for (p=0;p<pmax;p++){
            expr1+=Yikp[i][k][p];
        }
    }
}

```

```

model.add(IloMinimize(env, expr1));

expr1.end();

cplex.extract(model);

cplex.exportModel("berth.lp");

cplex.solve();

//IloNumMatrix4x4 vXizjt(env);

//cplex.getValues(vXizjt, Xizjt);

    if (!cplex.solve ()) {

        env.error() << "Failed to optimize LP." << endl;

        throw(-1);

    }

    env.out() << "Solution status = " << cplex.getStatus() << endl;

    env.out() << "Solution value = " << cplex.getObjValue() << endl;

for (i=0; i<imax; i++) {

    for (p=0; p<pmax; p++) {

        for (k=0; k<kmax; k++) {

            float g = cplex.getValue(Xipk[i][p][k]);

            if (g!=0) cout << "Xipk" << "(" << i << ", " << p << ", " << k << ")" << "=" << g << endl;

        }

    }

}

```

```

for (i=0;i<imax;i++){
    for (k=0;k<kmax;k++){
        for (p=0;p<pmax;p++){
            float g = cplex.getValue(Yikp[i][k][p]);
            if(g!=0) cout<<"Yikp"<<(" "<<i<<","<<k<<","<<p<<")<<=" "<<g<<endl;
        }
    }
}

//cout<<"-----";

//cout<<gg;

//cout<<endl;

}

catch ( IloException& e){
    cerr << "concert exception caught:"<<e<<endl;
}

catch (...){
    cerr<<"Unknown exception caught" <<endl;
}

env.end();

return 0;

} //End main

```

### Ο περιορισμός (28)

```

IloRangeMatrix2x2 Okp(env,0);

for (k=0;k<kmax;k++){

    IloRangeArray Op(env,0);

    for (p=0;p<pmax;p++){

```

```

        if(p>1){
            IloExpr expr(env,0);
IloExpr expr1(env,0);
            IloExpr expr2(env,0);
            for (h=0;h<pmax;h++){
                if (h<p) {
                    for (j=0;j<imax;j++){
                        if (j!=i) {
                            expr+=H[j][k]*Xipk[j][h][k]+Yikp[j][k][h];
                        }
                    }
                }
            }
            for (i=0;i<imax;i++){
                expr1+=(S[k]-A[i])*Xipk[i][p][k];
            }
            expr2=expr+expr1;

            char Okp[60];

            sprintf(Okp,"Okp(k%d,p%d)",k,p);

            float LB=0,UB=77;

            IloRange O(env,LB,expr2,UB,Okp);

            expr.end();

            expr1.end();

            expr2.end();

            model.add(O);

```

```

        Op.add(O);
    }
}

Okp.add(Op);
}

```

## 2<sup>η</sup> μοντελοποίηση

```

#include <ilcplex/ilocplex.h>

ILOSTLBEGIN

int i,p,k,l,j,n;

const int imax=12;

const int nmax=12;

const int kmax=3;

const int pmax=12;

const int lmax=12;

const int jmax=12;

//float capmax=100000;

//float capmin=0;

int lastl;

int

main (int argc, char **argv)

{

double H[imax][kmax];

double A[imax];

double S[kmax];

double B[kmax];

for (i=0;i<imax;i++){

    for (k=0;k<kmax;k++){

```

```
        H[i][k]=0;
    }
}
H[0][0]=1;
H[0][1]=3;
H[0][2]=1;
H[1][0]=1;
H[1][1]=1;
H[1][2]=2;
H[2][0]=2;
H[2][1]=2;
H[2][2]=2;
H[3][0]=1;
H[3][1]=1;
H[3][2]=1;
H[4][0]=4;
H[4][1]=1;
H[4][2]=1;
H[5][0]=3;
H[5][1]=2;
H[5][2]=1;
H[6][0]=1;
H[6][1]=1;
H[6][2]=1;
H[7][0]=5;
H[7][1]=5;
H[7][2]=4;
```

```
H[8][0]=8;
H[8][1]=8;
H[8][2]=3;
H[9][0]=1;
H[9][1]=6;
H[9][2]=1;
H[10][0]=1;
H[10][1]=1;
H[10][2]=1;
H[11][0]=7;
H[11][1]=7;
H[11][2]=5;
for (i=0;i<imax;i++){
```

```
    A[i]=0;
```

```
}
```

```
A[0]=2;
```

```
A[1]=1;
```

```
A[2]=3;
```

```
A[3]=1;
```

```
A[4]=3;
```

```
A[5]=4;
```

```
A[6]=1;
```

```
A[7]=2;
```

```
A[8]=1;
```

```
A[9]=6;
```

```
A[10]=1;
```

```
A[11]=2;
```

```

for (k=0;k<kmax;k++){
    S[k]=0;
}

S[0]=6;
S[1]=6;
S[2]=6;
for (k=0;k<kmax;k++){
    B[k]=0;
}

B[0]=100;
B[1]=100;
B[2]=100;

IloEnv env;
try {

    IloModel model (env);

typedef IloArray<IloNumArray> IloNumMatrix2x2;
typedef IloArray<IloNumMatrix2x2> IloNumMatrix3x3;
typedef IloArray<IloNumMatrix3x3> IloNumMatrix4x4;

typedef IloArray<IloNumVarArray> IloNumVarMatrix2x2;
typedef IloArray<IloNumVarMatrix2x2> IloNumVarMatrix3x3;
typedef IloArray<IloNumVarMatrix3x3> IloNumVarMatrix4x4;

typedef IloArray<IloRangeArray> IloRangeMatrix2x2;
typedef IloArray<IloRangeMatrix2x2> IloRangeMatrix3x3;

```



```

typedef IloArray<IloRangeMatrix3x3> IloRangeMatrix4x4;

IloCplex cplex(env);

//----- METAVLHTH APOFASHS Y (continuous)-----

IloNumVarMatrix2x2 Ykp(env,0);

for (k=0;k<kmax;k++){

    IloNumVarArray Yp(env,0);

    for (p=0;p<pmax;p++){

        char Quantite_Decharge[70];

        sprintf(Quantite_Decharge,"Ykp(k%d,p%d)",k,p);

        IloNumVar Y(env,0,100000,ILOFLOAT,Quantite_Decharge);

        Yp.add(Y);

    }

    Ykp.add(Yp);

}

cout<<1<<endl;

//----- METAVLHTH APOFASHS X (binary)-----

IloNumVarMatrix3x3 Xkip(env,0);

for (k=0;k<kmax;k++){

    IloNumVarMatrix2x2 Xip(env,0);

    for (i=0;i<imax;i++){

        IloNumVarArray Xp(env,0);

        for (p=0;p<pmax;p++){

            char Chargement[70];

            sprintf(Chargement,"Xkip(k%d,i%d,p%d)",k,i,p);

            IloNumVar X(env,0,1,ILOINT,Chargement);

            Xp.add(X);

```

```

    }

    Xip.add(Xp);

}

Xkip.add(Xip);

}

cout<<2<<endl;

//-----
//-----PERIORISMOI-----
//-----
//----- PERIORISMOS 2 -----

IloRangeArray SumXi(env,0);

for (i=0;i<imax;i++){

    IloExpr expr(env,0);

    for (k=0;k<kmax;k++){

        for (p=0;p<pmax;p++){

            expr+=Xkip[k][i][p];

        }

    }

    char Quantite_Charge[60];

    sprintf(Quantite_Charge,"SumXi(i%d)",i);

    float LB=1,UB=1;

    IloRange SumX(env,LB,expr,UB,Quantite_Charge);

    model.add(SumX);

    SumXi.add(SumX);

```

```

        expr.end();
    }

cout<<3<<endl;

//-----
//----- PERIORISMOS 3 -----

IloRangeMatrix2x2 SumXkp(env,0);

for (k=0;k<kmax;k++){

    IloRangeArray SumXp(env,0);

    for (p=0;p<pmax;p++){

        IloExpr expr(env);

        for (i=0;i<imax;i++){

            expr+=Xkip[k][i][p];

        }

        char Quantite_Decharge[60];

        sprintf(Quantite_Decharge,"SumXkp(k%d,p%d)",k,p);

        float LB=0,UB=1;

        IloRange SumX(env,LB,expr,UB,Quantite_Decharge);

        model.add(SumX);

        SumXp.add(SumX);

        expr.end();

    }

    SumXkp.add(SumXp);

}

cout<<4<<endl;

//-----
//----- PERIORISMOS 4 -----

```

```

IloRangeMatrix2x2 Y_4kp(env,0);

for (k=0;k<kmax;k++){

    IloRangeArray Y_4p(env,0);

    for (p=0;p<pmax;p++){

        IloExpr expr(env);

        IloExpr expr1(env);

        IloExpr expr2(env);

        IloExpr expr3(env);

        for (l=0;l<pmax;l++){

            if(l<p){

                for (j=0;j<imax;j++){

                    if (j!=i) {

                        expr1+=H[j][k]*Xkip[k][j][l];

                    }

                }

                expr+=Ykp[k][l]+expr1;

            }

        }

        for (i=0;i<imax;i++){

            expr2+= (A[i]-S[k])*Xkip[k][i][p];

        }

        expr3=(expr2)-(expr)-Ykp[k][p];
    }
}

```

```

char Equoition_equilibre[60];

sprintf(Equoition_equilibre,"Y_4kp(k%d,p%d)",k,p);

float LB=-IloInfinity,UB=0;

IloRange Y_4(env,LB,expr3,UB,Equoition_equilibre);

model.add(Y_4);

Y_4p.add(Y_4);

expr.end();

expr1.end();

expr2.end();

expr3.end();

}

Y_4kp.add(Y_4p);

}

cout<<5<<endl;

//----- PERIORISMOS 5 -----

IloRangeArray Y_7k(env,0);

for (k=0;k<kmax;k++){

    IloExpr expr(env);

        IloExpr expr1(env);

```

```

for (i=0;i<imax;i++){
    for (p=0;p<pmax;p++){
        expr1+=(H[i][k]*Xkip[k][i][p] + Ykp[k][p]);
    }
}

expr=(B[k]-S[k])+expr1;

char Equoition_equilibre[60];

sprintf(Equoition_equilibre,"Y_7k(k%d)",k);

float LB=0,UB=IloInfinity;

IloRange Y_7(env, LB, expr, UB, Equoition_equilibre);

model.add(Y_7);

Y_7k.add(Y_7);

expr.end();

expr1.end();

}

```

```
cout<<6<<endl;
```

```

//-----
//-----
//-----ANTIKEIMENIKH-----
//-----

```

```

IloExpr expr1(env);

for (k=0;k<kmax;k++){
    for (i=0;i<imax;i++){

```

```

        for (p=0;p<pmax;p++){
            expr1+=(H[i][k] + S[k] - A[i])*Xkip[k][i][p];
        }
    }
}

for (k=0;k<kmax;k++){
    for (p=0;p<pmax;p++){
        expr1+=Ykp[k][p];
    }
}

model.add(IloMinimize(env, expr1));

expr1.end();

cplex.extract(model);

cplex.exportModel("berth.lp");

cplex.solve();

//IloNumMatrix4x4 vXizjt(env);

//cplex.getValues(vXizjt, Xizjt);

if (!cplex.solve ()) {
    env.error()<<"Failed to optimize LP."<<endl;
    throw(-1);
}

```

```

env.out()<<<"Solution status = " <<cplex.getStatus()<<<endl;

env.out()<<<"Solution value = " <<cplex.getObjValue()<<<endl;

for (k=0;k<kmax;k++){

    for (i=0;i<imax;i++){

        for (p=0;p<pmax;p++){

            float g = cplex.getValue(Xkip[k][i][p]);

            if(g!=0) cout<<<"Xkip"<<<(" "<<k<<<","<<i<<<","<<p<<<")<<< "="<<g<<<endl;

        }

    }

}

for (k=0;k<kmax;k++){

    for (p=0;p<pmax;p++){

        float g = cplex.getValue(Ykp[k][p]);

        if(g!=0) cout<<<"Ykp"<<<(" "<<k<<<","<<p<<<")<<< "="<<g<<<endl;

    }

}

//cout<<<"-----";

//cout<<<gg;

//cout<<<endl;

}

catch ( IloException& e){

    cerr << "concert exception caught:"<<<e<<<endl;

}

catch (...){

    cerr<<<"Unknown exception caught" <<<endl;

```



```

}

env.end();

return 0;

} //End main

```

Ο περιορισμός (29)

```

IloRangeMatrix2x2 Okp(env,0);

for (k=0;k<kmax;k++){

    IloRangeArray Op(env,0);

    for (p=0;p<pmax;p++){

        if (p>1){

            IloExpr expr(env,0);

            IloExpr expr1(env,0);

            IloExpr expr2(env,0);

            for (h=0;h<pmax;h++){

                if (h<p) {

                    for (q=0;q<imax;q++){

                        if (q!=i) {

                            expr+=H[q][k]*Xkip[k][q][h]+Ykp[k][h];

                        }

                    }

                }

            }

            for (i=0;i<imax;i++){

                expr1+=(S[k]-A[i])*Xkip[k][i][p];

            }

        }

    }

}

```

```

        expr2=expr+expr1;

        char Okp[60];

        sprintf(Okp,"Okp(k%d,p%d)",k,p);

        float LB=0,UB=77;

        IloRange O(env,LB,expr2,UB,Okp);

        expr.end();

        expr1.end();

        expr2.end();

        model.add(O);

        Op.add(O);

    }

}

Okp.add(Op);

}

```

### 3η μοντελοποίηση

```

#include <ilcplex/ilocplex.h>

ILOSTLBEGIN

int

main (int argc, char **argv)

{

    IloEnv env;

    typedef IloArray<IloNumArray> IloNumMatrix2x2;

    typedef IloArray<IloNumMatrix2x2> IloNumMatrix3x3;

    typedef IloArray<IloNumMatrix3x3> IloNumMatrix4x4;

    typedef IloArray<IloNumMatrix4x4> IloNumMatrix5x5;

    typedef IloArray<IloNumMatrix5x5> IloNumMatrix6x6;

```

```
typedef IloArray<IloNumVarArray> IloNumVarMatrix2x2;  
typedef IloArray<IloNumVarMatrix2x2> IloNumVarMatrix3x3;  
typedef IloArray<IloNumVarMatrix3x3> IloNumVarMatrix4x4;  
typedef IloArray<IloNumVarMatrix4x4> IloNumVarMatrix5x5;  
typedef IloArray<IloNumVarMatrix5x5> IloNumVarMatrix6x6;
```

```
typedef IloArray<IloRangeArray> IloRangeMatrix2x2;  
typedef IloArray<IloRangeMatrix2x2> IloRangeMatrix3x3;  
typedef IloArray<IloRangeMatrix3x3> IloRangeMatrix4x4;  
typedef IloArray<IloRangeMatrix4x4> IloRangeMatrix5x5;  
typedef IloArray<IloRangeMatrix5x5> IloRangeMatrix6x6;
```

```
int i,j,t,l,q,n;
```

```
const int imax=12;
```

```
const int lmax=12;
```

```
const int qmax=100;
```

```
const int tmax=100;
```

```
const int nmax=100;
```

```
const int jmax=3;
```

```
int loop;
```

```
IloModel modelSlave (env);
```

```
IloModel modelMaster (env);
```

```
IloCplex cplexSlave(env);
```

```
IloCplex cplexMaster(env);
```

```
////////// DATA //////////////////////////////////////
```

```
double M=1000;
```

```
double sm=0.001;
```

```
int H[imax][jmax];
```

```
for (i=0;i<imax;i++){
```

```
    for (j=0;j<jmax;j++){
```

```
        H[i][j]=0;
```

```
    }
```

```
}
```

```
H[0][0]=1;
```

```
H[0][1]=3;
```

```
H[0][2]=1;
```

```
H[1][0]=1;
```

```
H[1][1]=1;
```

```
H[1][2]=2;
```

```
H[2][0]=2;
```

```
H[2][1]=2;
```

```
H[2][2]=2;
```

```
H[3][0]=1;
```

```
H[3][1]=1;
```

```
H[3][2]=1;
```

```
H[4][0]=4;
```

```
H[4][1]=1;
```

```
H[4][2]=1;
```

```
H[5][0]=3;
```

```
H[5][1]=2;
```

```
H[5][2]=1;
```

```
H[6][0]=1;
H[6][1]=1;
H[6][2]=1;
H[7][0]=5;
H[7][1]=5;
H[7][2]=4;
H[8][0]=8;
H[8][1]=8;
H[8][2]=3;
H[9][0]=1;
H[9][1]=6;
H[9][2]=1;
H[10][0]=1;
H[10][1]=1;
H[10][2]=1;
H[11][0]=7;
H[11][1]=7;
H[11][2]=5;
int A[imax];
for (i=0;i<imax;i++){
    A[i]=0;
}
A[0]=2;
A[1]=1;
A[2]=3;
A[3]=1;
A[4]=3;
```

```

A[5]=4;

A[6]=1;

A[7]=2;

A[8]=1;

A[9]=6;

A[10]=1;

A[11]=2;

int S[jmax];

for (j=0;j<jmax;j++){

                S[j]=0;

                }

S[0]=6;

S[1]=6;

S[2]=6;

int B[jmax];

for (j=0;j<jmax;j++){

                B[j]=0;

                }

B[0]=100;

B[1]=100;

B[2]=100;

//-----

//-----

//-----

//-----

//----- Variable de Decision -----

```

```

IloNumVarMatrix3x3 SCijt(env,0);

for (i=0;i<imax;i++){

    IloNumVarMatrix2x2 SCjt(env,0);

    for (j=0;j<jmax;j++){

        IloNumVarArray SCt(env,0);

        for (t=0;t<tmax;t++){

            char SetupUnload_i_j_t[70];

            sprintf(SetupUnload_i_j_t,"SC(i%d,j%d,t%d)",i,j,t);

            IloNumVar SC(env,0,1,ILOINT,SetupUnload_i_j_t);

            SCt.add(SC);

        }

        SCjt.add(SCt);

    }

    SCijt.add(SCjt);

}

```

```

cout<<1<<endl;

```

```

//-----

```

```

IloNumVarMatrix3x3 CCijt(env,0);

for (i=0;i<imax;i++){

    IloNumVarMatrix2x2 CCjt(env,0);

    for (j=0;j<jmax;j++){

        IloNumVarArray CCt(env,0);

        for (t=0;t<tmax;t++){

            char Unload_i_j_t[70];

            sprintf(Unload_i_j_t,"C(i%d,j%d,t%d)",i,j,t);

            IloNumVar CC(env,0,1,ILOINT,Unload_i_j_t);

```

```

        CCt.add(CC);
    }
    CCjt.add(CCt);
}
CCijt.add(CCjt);
}
cout<<1<<endl;

//-----METAVLHTH APOFASHS I (continuous)-----

IloNumVarMatrix2x2 Ijt(env,0);
for (j=0;j<jmax;j++){
    IloNumVarArray It(env,0);
    for (t=0;t<tmax;t++){
        char Chargement[70];
        sprintf(Chargement,"Ijt(j%d,t%d)",j,t);
        IloNumVar I(env,0,1000000,ILOFLOAT,Chargement);
        It.add(I);
    }
    Ijt.add(It);
}

cout<<1<<endl;

//----- Constraints -----

IloRangeMatrix2x2 Con2it(env,0);
for (i=0;i<imax;i++){
    IloRangeArray Con2t(env,0);

```



```

    for (t=0;t<tmax;t++){
        IloExpr expr(env,0);
    for (j=0;j<jmax;j++){
        expr+=CCijt[i][j][t];
    }

        char Con_2[60];

        sprintf(Con_2,"Con_2it(i%d,t%d)",i,t);

        float LB=0,UB=1;

        IloRange Con2(env,LB,expr,UB,Con_2);

        modelSlave.add(Con2);

        Con2t.add(Con2);

        expr.end();

    }

    Con2it.add(Con2t);
}

cout<<<<endl;

//-----

IloRangeArray Con3i(env,0);

for (i=0;i<imax;i++){

    IloExpr expr(env,0);

    for (t=0;t<tmax;t++){

        for (j=0;j<jmax;j++){

            expr+=SCijt[i][j][t];

        }

    }

}

char Con_3[60];

```

```

sprintf(Con_3,"Con_3i(i%d)",i);

float LB=1,UB=1;

IloRange Con3(env,LB,expr,UB,Con_3);

modelSlave.add(Con3);

Con3i.add(Con3);

expr.end();

}

cout<<3<<endl;

//-----

IloRangeMatrix2x2 Con6jt(env,0);

for (j=0;j<jmax;j++){

    IloRangeArray Con6t(env,0);

    for (t=0;t<tmax;t++){

        IloExpr expr(env,0);

        for (i=0;i<imax;i++){

            expr+=CCijt[i][j][t];

        }

        char Con_6[60];

        sprintf(Con_6,"Con_6jt(j%d,t%d)",j,t);

        float LB=0,UB=1;

        IloRange Con6(env,LB,expr,UB,Con_6);

        modelSlave.add(Con6);

        Con6t.add(Con6);

        expr.end();

    }

}

```

```

Con6jt.add(Con6t);

}

cout<<4<<endl;

//-----

IloExpr expr9(env,0);

for (i=0;i<imax;i++){

    for (t=0;t<A[i];t++){

        for (j=0;j<jmax;j++){

            expr9+=SCijt[i][j][t]+CCijt[i][j][t];

        }

    }

}

float LB=0,UB=0;

IloRange Con9(env,LB,expr9,UB);

modelSlave.add(Con9);

expr9.end();

cout<<5<<endl;

//-----

IloRangeMatrix3x3 Con10ijt(env,0);

for (i=0;i<imax;i++){

    IloRangeMatrix2x2 Con10jt(env,0);

    for (j=0;j<jmax;j++){

        IloRangeArray Con10t(env,0);

        for (t=0;t<tmax;t++){

            IloExpr expr(env,0);

            expr+=CCijt[i][j][t]-SCijt[i][j][t];

```

```

char Con_10[60];

sprintf(Con_10,"Con_10ijt(i%d,j%d,t%d)",i,j,t);

float LB=0,UB=IloInfinity;

IloRange Con10(env,LB,expr,UB,Con_10);

modelSlave.add(Con10);

Con10t.add(Con10);

expr.end();

    }

Con10jt.add(Con10t);

}

Con10ijt.add(Con10jt);

}

cout<<6<<endl;

//-----

IloRangeMatrix3x3 Con11ijt(env,0);

for (i=0;i<imax;i++){

    IloRangeMatrix2x2 Con11jt(env,0);

    for (j=0;j<jmax;j++){

        IloRangeArray Con11t(env,0);

        for (t=0;t<tmax-H[i][j];t++){

            //if (t<tmax-H[i][j]){

                IloExpr expr(env,0);

                for (n=0;n<H[i][j];n++){

                    expr+=CCijt[i][j][t+n];

                }

                expr+=-SCijt[i][j][t]*(H[i][j]+M)+M;

```

```

char Con_11[60];

sprintf(Con_11,"Con_11ijt(i%d,j%d,t%d)",i,j,t);

float LB=0,UB=IloInfinity;

IloRange Con11(env,LB,expr,UB,Con_11);

modelSlave.add(Con11);

Con11t.add(Con11);

expr.end();

//}

    }

Con11jt.add(Con11t);

}

Con11ijt.add(Con11jt);

}

cout<<7<<endl;

//-----

IloRangeMatrix3x3 Con21ijt(env,0);

for (i=0;i<imax;i++){

    IloRangeMatrix2x2 Con21jt(env,0);

    for (j=0;j<jmax;j++){

        IloRangeArray Con21t(env,0);

        for (t=0;t<tmax-H[i][j];t++){

            //if (t<tmax-H[i][j]){

                IloExpr expr(env,0);

                for (n=0;n<H[i][j];n++){

                    expr+=-CCijt[i][j][t+n];

```

```

    }

    expr+=SCijt[i][j][t]*(H[i][j]-M)+M;

    char Con_21[60];

    sprintf(Con_21,"Con_21ijt(i%d,j%d,t%d)",i,j,t);

    float LB=0,UB=IloInfinity;

    IloRange Con21(env,LB,expr,UB,Con_21);

    modelSlave.add(Con21);

    Con21t.add(Con21);

    expr.end();

    //    }

    }

    Con21jt.add(Con21t);

}

Con21ijt.add(Con21jt);

}

cout<<8<<endl;

//-----

IloRangeMatrix3x3 Con5ijt(env,0);

for (i=0;i<imax;i++){

    IloRangeMatrix2x2 Con5jt(env,0);

    for (j=0;j<jmax;j++){

        IloRangeArray Con5t(env,0);

        for (t=0;t<tmax;t++){

            if (t==0){

                IloExpr expr(env,0);

```

```

expr+=SCijt[i][j][t]-CCijt[i][j][t];

        char Con_5[60];

        sprintf(Con_5,"Con_5ijt(i%d,j%d,t%d)",i,j,t);

        float LB=0,UB=IloInfinity;

        IloRange Con5(env,LB,expr,UB,Con_5);

        modelSlave.add(Con5);

        Con5t.add(Con5);

        expr.end();

    }

    if (t>0){

        IloExpr expr2(env,0);

expr2+=CCijt[i][j][t-1]+SCijt[i][j][t]-CCijt[i][j][t];

        char Con_5[60];

        sprintf(Con_5,"Con_5ijt(i%d,j%d,t%d)",i,j,t);

        float LB=0,UB=IloInfinity;

        IloRange Con5(env,LB,expr2,UB,Con_5);

        modelSlave.add(Con5);

        Con5t.add(Con5);

        expr2.end();

    }

    }

    Con5jt.add(Con5t);

}

Con5ijt.add(Con5jt);

}

```

```
cout<<9<<endl;
```

```
//-----
```

```
IloRangeMatrix2x2 Ujt(env,0);
```

```
for (j=0;j<jmax;j++){
```

```
    IloRangeArray Ut(env,0);
```

```
    for (t=0;t<tmax;t++){
```

```
        IloExpr expr(env);
```

```
        IloExpr expr1(env);
```

```
        IloExpr expr2(env);
```

```
        IloExpr expr3(env);
```

```
        for (q=0;q<tmax;q++){
```

```
            if(q<t){
```

```
                for (l=0;l<imax;l++){
```

```
                    if (l!=i) {
```

```
                        expr1 += H[l][j]*SCijt[l][j][q];
```

```
                    }
```

```
                }
```

```
                expr += Ijt[j][q] + expr1;
```

```
            }
```

```
        }
```

```
        for (i=0;i<imax;i++){
```

```
            expr2 += (A[i]-S[j])*SCijt[i][j][t]; }
```



```

        expr3=(expr2)-(expr)-Ijt[j][t];

        char Equoition_equilibre[60];

        sprintf(Equoition_equilibre,"Ujt(j%d,t%d)",j,t);

        float LB=-IloInfinity,UB=0;

        IloRange U(env,LB,expr3,UB,Equoition_equilibre);

        modelSlave.add(U);

        Ut.add(U);

        expr.end();

        expr1.end();

        expr2.end();

        expr3.end();

    }

    Ujt.add(Ut);

}

cout<<10<<endl;

//----- PERIORISMOS 4 -----

IloRangeArray Y_7j(env,0);

for (j=0;j<jmax;j++){

    IloExpr expr(env);

        IloExpr expr1(env);

        for (i=0;i<imax;i++){

            for (t=0;t<tmax;t++){

```

```

        expr1+=(H[i][j]*SCijt[i][j][t] + Ijt[j][t]);
    }
}

expr=(B[j]-S[j])+expr1;

char Equoition_equilibre[60];

sprintf(Equoition_equilibre,"Y_7j(j%d)",j);

float LB=0,UB=IloInfinity;

IloRange Y_7(env,LB,expr,UB,Equoition_equilibre);

modelSlave.add(Y_7);

Y_7j.add(Y_7);

expr.end();

expr1.end();
}

cout<<11<<endl;

IloExpr expr1(env);

for (i=0;i<imax;i++){

    for (j=0;j<jmax;j++){

        for (t=0;t<tmax;t++){

            expr1+=(H[i][j]+S[j]-
A[i]+sm*(i+1)*(j+1)*(t+1))*SCijt[i][j][t];//+000.1*t*SCijt[i][j][t];

        }

    }

}

for (j=0;j<jmax;j++){

    for (t=0;t<tmax;t++){

        expr1+=Ijt[j][t];
    }
}

```



```

venia=venia+(H[i][j]+S[j]-A[i])*g;

venia2=venia2+(S[j]-A[i])*g;

if(g>=0.001) cout<<"SCijt"<<"("<<i<<","<<j<<","<<t<<")"<<="<<g<<endl;

        }

    }

}

cout<<"obj="<<venia;

cout<<"\n";

cout<<"dif="<<venia2;

cout<<"\n";

cout<<"_____ "<<endl;

for (i=0;i<imax;i++){

    for (j=0;j<jmax;j++){

        for (t=0;t<tmax;t++){

float gg = cplexSlave.getValue(CCijt[i][j][t]);

if(gg>=0.001) cout<<"Cijt"<<"("<<i<<","<<j<<","<<t<<")"<<="<<gg<<endl;

        }

    }

}

for (j=0;j<jmax;j++){

    for (t=0;t<tmax;t++){

        float g = cplexSlave.getValue(Ijt[j][t]);

        if(g!=0)

            cout<<"Ijt"<<"("<<j<<","<<t<<")"<<="<<g<<endl;

    }

}

```

```

} //end of first try(try of primal slave 1)

catch ( IloException& e){

    cerr << "concert exception caught:"<<e<<endl;

}

catch (...){

    cerr<<"Unknown exception caught" <<endl;

}

env.end();

return 0;

} //End main

```

### Ο περιορισμός (30)

```

IloRangeMatrix2x2 Ojt(env,0);

for (j=0;j<jmax;j++){

    IloRangeArray Ot(env,0);

    for (t=0;t<tmax;t++){

        if (t>1){

            IloExpr expr(env,0);

            IloExpr expr1(env,0);

            IloExpr expr2(env,0);

            for (h=0;h<tmax;h++){

                if (h<t) {

                    for (k=0;k<imax;k++){

                        if (k!=i) {

```

```

        expr+=H[k][j]*SCijt[k][j][h]+Ijt[j][h];
    }
}
}

for (i=0;i<imax;i++){
    expr1+=(S[j]-A[i])*SCijt[i][j][t];
}

    expr2=expr+expr1;

    char Ojt[60];

    sprintf(Ojt,"Ojt(j%d,t%d)",j,t);

    float LB=0,UB=77;

    IloRange O(env,LB,expr2,UB,Ojt);

    expr.end();

    expr1.end();

    expr2.end();

    modelSlave.add(O);

    Ot.add(O);
}
}

    Ojt.add(Ot);
}

```

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J. and Lusby, R., 2009. Models for the Discrete Berth Allocation Problem: A Computational Comparison.
2. Murty, K.G., Liu, J., Wan, Y. and Linn, R., 2003. A decision support system for operations in a container terminal. *Decision Support Systems*, 39 (2005), 309– 332.
3. Bierwirth, C. and Meisel, F., 2009. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202 (2010), 615–627.
4. Lun, Y.H.V., 2010. Green management practices and firm performance: A case of container terminal Operations. *Resources, Conservation and Recycling*, 55 (2011), 559–566.
5. Sharif, O. and Huynh, N., 2012. Storage space allocation at marine container terminals using ant-based control. *Expert Systems with Applications*, (2012).
6. Chen, J.H., Lee, D.H. and Cao, J.X., 2011. Heuristics for quay crane scheduling at indented berth. *Transportation Research Part E*, 47 (2011,) 1005–1020.
7. Han, X.-l., Lu, Z.-q. and Xi, L.-f., 2010. A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research*, 207 (2010), 1327–1340.
8. Du, Y., Chen, Q., Quan, X., Long, L. and Fung, R.Y.K., 2011. Berth allocation considering fuel consumption and vessel emissions. *Transportation Research Part E*, 47 (2011), 1021–1037.
9. Giallombardo, G., Moccia, L., Salani, M. and Vacca, I., 2009. Modeling and solving the Tactical Berth Allocation Problem. *Transportation Research Part B*, 44 (2010), 232–245.
10. Salido, M.A., Rodriguez-Molins, M. and Barber, F., 2010. Integrated intelligent techniques for remarshaling and berthing in maritime terminals. *Advanced Engineering Informatics*, 25 (2011), 435–451.
11. Halvorsen-Weare, E.E., Fagerholt, K. and Rönnqvist, M., 2012. Vessel routing and scheduling under uncertainty in the liquefied natural gas business. *Computers & Industrial Engineering*, 64 (2013), 290–301.
12. Psaraftis, H.N. and Kontovas, C.A., 2012. Speed models for energy-efficient maritime transportation: A taxonomy and survey. *Transportation Research Part C*, 26 (2013), 331–351.

13. Hansen P, Oguz C, Mladenovic N (2008) Variable neighborhood search for minimum cost berth allocation. *Eur J Oper Res* 191 (3):636–649. doi:10.1016/j.ejor.2006.12.057
14. Imai A, Nishimura E, Papadimitriou S (2001) The dynamic berth allocation problem for a container port. *Transp Res Part B* 35:401–417. doi:10.1016/S0191-2615(99)00057-0
15. Imai A, Nishimura E, Papadimitriou S (2003) Berth allocation with service priority. *Transp Res Part B* 37:437–457. doi:10.1016/S0191-2615(02)00023-1
16. Monaco FM, Samara M (2007) The berth allocation problem: a strong formulation solved by a lagrangean approach. *Transport Sci* 41(2):265–280. doi:10.1287/trsc.1060.0171
17. Guan Y, Cheung RK (2004) The berth allocation problem: models and solution methods. *OR-Spectrum* 26(1):75–92. doi:10.1007/s00291-003-0140-8
18. Guan Y, Xiao W-Q, Cheung RK, Li C-L (2002) A multiprocessor task scheduling model for berth allocation: heuristic and worst case analysis. *Oper Res Lett* 30(5):343–350. doi:10.1016/S0167-6377(02)00147-5
19. Kim KH, Moon KC (2003) Berth scheduling by simulated annealing. *Transp Res Part B* 37:541–560. doi:10.1016/S0191-2615(02)00027-9
20. Moorthy R, Teo C-P (2006) Berth management in container terminal: the template design problem. *OR-Spectrum* 28(4):495–518. doi:10.1007/s00291-006-0036-5
21. Hooker NJ (1998) Constraint satisfaction methods for generating valid cuts. Available Online as: <http://citeseer.ist.psu.edu/163317.html>.
22. Imai A, Chen HC, Nishimura E, Papadimitriou S (2008) The simultaneous berth and quay crane allocation problem. *Transp Res Part E* 44(5):900–920. doi:10.1016/j.tre.2007.03.003
23. Park MY, Kim HK (2003) A scheduling method for berth and quay cranes. *Oper Res Spectr* 25(1):1–23. doi:10.1007/s00291-002-0109-z Springer
24. Theofanis S, Boile M, Golias MM (2009) Container terminal berth planning: critical review of research approaches and practical challenges. *J Transp Res Rec*



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΒΙΒΛΙΟΘΗΚΗ



004000121282

