

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΜΗΝΟΥΣ ΣΩΜΑΤΙΔΙΩΝ ΓΙΑ ΤΟ  
ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΚΑΙ ΧΡΟΝΙΚΟΥ  
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΦΟΡΤΗΓΩΝ ΠΛΟΙΩΝ ΒΙΟΜΗΧΑΝΙΚΟΥ  
ΦΟΡΤΙΟΥ ΚΑΙ ΕΛΕΥΘΕΡΩΝ ΦΟΡΤΗΓΩΝ ΠΛΟΙΩΝ ΞΗΡΟΥ  
ΦΟΡΤΙΟΥ

ΦΟΙΒΟΣ ΑΠΟΣΤΟΛΟΣ ΜΑΡΓΩΝΗΣ



**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
TECHNICAL UNIVERSITY OF CRETE

Διπλωματική Εργασία  
Σχολή Μηχανικών Παραγωγής και Διοίκησης  
Πολυτεχνείο Κρήτης

Επιβλέπων: Δρ. Ιωάννης Μαρινάκης

Χανιά, Οκτώβριος 2021

Φοίβος Απόστολος Μαργώνης: Βελτιστοποίηση Σμίνους Σωματιδίων  
για το πρόβλημα δρομολόγησης και χρονικού προγραμματισμού  
φορτηγών πλοίων βιομηχανικού φορτίου και ελεύθερων φορτηγών  
πλοίων ξηρού φορτίου, Διπλωματική Εργασία © Οκτώβριος 2021

## ΠΕΡΙΛΗΨΗ

---

Σύγχρονες μελέτες στον χώρο της ναυτιλιακής εφοδιαστικής αλυσίδας παρουσιάζουν ένα γενικό πρόβλημα δρομολόγησης πλοίων, μαζί με ένα σύνολο λύσεων βασισμένο σε πραγματικά δεδομένα, ως σημείο αναφοράς. Ένας ετερογενής στόλος πλοίων αναλαμβάνει την παραλαβή και παράδοση ενός συνόλου προϊόντων, αποσκοπώντας στην εξυπηρέτησή τους με βέλτιστη σειρά, την κατάλληλη ανάθεση εμπορευμάτων σε πλοία και στο σωστό χρονικό περιθώριο. Ταυτόχρονα πρέπει να λαμβάνεται υπ' όψη η μέγιστη χωρητικότητα του κάθε πλοίου, ο χρόνος και η θέση από την οποία ξεκινάει να είναι διαθέσιμο όπως και περιορισμοί προτεραιότητας και συμβατότητας. Στόχος της παρούσας διπλωματικής εργασίας είναι να κατασκευαστεί μια μέθοδος ικανή να ελαχιστοποιεί το κόστος εξυπηρέτησης των εμπορευμάτων, στον ελάχιστο δυνατό χρόνο, ανεξάρτητα από το μέγεθος και την πολυπλοκότητα των δεδομένων. Για να αντιμετωπιστεί αυτή η πρόκληση, μία καινοτόμος δομή του αλγορίθμου Βελτιστοποίησης Σμήνους Σωματιδίων, προσαρμοσμένη για εφαρμογή σε διακριτά προβλήματα, συνδυάζεται με μια πρωτοποριακή παραλλαγή ενός μεθευρετικού αλγορίθμου κατασκευής αρχικών λύσεων και ένα σύνολο τροποποιημένων τοπικών αναζητήσεων. Η λειτουργία του αναλύεται ενδελεχώς στα επόμενα κεφάλαια. Ο προτεινόμενος αλγόριθμος συγκρίνεται με τον καλύτερο και αποδοτικότερο έως τώρα αλγόριθμο πάνω σε ένα ευρύ φάσμα ρεαλιστικών δεδομένων.

## ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

---

Ναυτιλιακή Εφοδιαστική Αλυσίδα, Δρομολόγηση Πλοίων, Βελτιστοποίηση Σμήνους Σωματιδίων, Industrial and Tramp Ship Routing.



*If I have seen further,  
it is by standing on the shoulders of Giants.*

— Isaac Newton

## ΕΥΧΑΡΙΣΤΙΕΣ

---

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. *Ιωάννη Μαρινάκη* για την ανεκτίμητη βοήθεια και καθοδήγηση που μου παρείχε καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας. Θα ήθελα να ευχαριστήσω ιδιαίτερα τον φίλο και συνεργάτη κ. *Ελευθέριο Τσακιράκη* για την αμέριστη βοήθεια και υποστήριξη του. Επιπλέον, θέλω να ευχαριστήσω τους φίλους συμφοιτητές μου για τις πολλές ευχάριστες εκπαιδευτικές ώρες που περάσαμε μαζί. Τέλος, το μεγαλύτερο ευχαριστώ οφείλω στη οικογένειά μου για την αδιάκοπη στήριξη που μου παρείχαν όλα αυτά τα χρόνια.



## ΠΕΡΙΕΧΟΜΕΝΑ

---

|                             |  |
|-----------------------------|--|
| <b>I ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ</b> |  |
| 1                           | ΕΙΣΑΓΩΓΗ 3                               |
| 2                           | ΕΝ ΠΛΩ ΕΦΟΔΙΑΣΤΙΚΗ ΑΛΥΣΙΔΑ 5             |
| 2.1                         | Χαρακτηριστικά πλοίων 5                  |
| 2.2                         | Τύποι ναυτιλιακών υπηρεσιών 6            |
| 2.3                         | Χαρακτηριστικά φορτίων 6                 |
| 2.4                         | Γεωγραφικά χαρακτηριστικά 7              |
| 3                           | ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ ΠΡΟΒΛΗΜΑΤΩΝ 9   |
| 3.1                         | Προβλήματα δρομολόγησης οχημάτων (VRP) 9 |
| 3.1.1                       | Το πρόβλημα του TSP 9                    |
| 3.1.2                       | Το πρόβλημα του CVRP 9                   |
| 3.1.3                       | Το πρόβλημα του VRP-TW 9                 |
| 3.1.4                       | Το πρόβλημα του VRP-PDPTW 10             |
| 3.2                         | Το πρόβλημα του ITSRSRSP 10              |
| 4                           | ΚΑΤΗΓΟΡΙΕΣ ΑΛΓΟΡΙΘΜΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ 15 |
| 4.1                         | Ευρετικοί Αλγόριθμοι 15                  |
| 4.1.1                       | Αλγόριθμοι απληστίας 16                  |
| 4.1.2                       | Αλγόριθμοι τοπικής αναζήτησης 17         |
| 4.2                         | Μεθευρετικοί Αλγόριθμοι 17               |
| 4.2.1                       | Τροχιακοί 18                             |
| 4.2.2                       | Πληθυσμιακοί 19                          |
| <b>II ΠΡΑΚΤΙΚΗ ΕΠΙΛΥΣΗ</b>  |  |
| 5                           | ΑΝΑΛΥΣΗ ΤΟΥ ACP SO 27                    |
| 5.1                         | Σχηματισμός αρχικών λύσεων 27            |
| 5.2                         | Τοπική Αναζήτηση 30                      |
| 5.2.1                       | 1-0 Relocate 30                          |
| 5.2.2                       | 1-1 Intraexchange 31                     |
| 5.2.3                       | 1-Y Interexchange 31                     |
| 5.2.4                       | Επιλογή κατάλληλης γειτονιάς 32          |
| 5.2.5                       | Μηχανισμός Διχοτόμησης 32                |
| 5.3                         | Ο αλγόριθμος ACP SO 33                   |
| 5.3.1                       | Δομή αλγορίθμου 33                       |
| 5.3.2                       | Επιπλέον βελτίωση λύσης 36               |
| 6                           | ΑΠΟΤΕΛΕΣΜΑΤΑ 39                          |
| 6.1                         | Παράδειγμα 45                            |
| 7                           | ΣΥΜΠΕΡΑΣΜΑΤΑ 49                          |
| <b>III APPENDIX</b>         |  |
| A                           | APPENDIX 53                              |
| BIBΛΙΟΓΡΑΦΙΑ 57             |  |

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

---

|            |   |    |
|------------|---|----|
| Εικόνα 2.1 | Χαρακτηριστικά προσφοράς και ζήτησης της εν πλώ εφοδιαστικής αλυσίδας.  | 7  |
| Εικόνα 3.1 | Λύση πλήρους φορτίου υπερατλαντικών ταξιδιών  | 13 |
| Εικόνα 3.2 | Λύση μικτού φορτίου υπερατλαντικών ταξιδιών   | 13 |
| Εικόνα 4.1 | Η κίνηση ενός σωματιδίου προς την καλύτερη λύση του σμήνους $g^*$ και προς την καλύτερη λύση $x_i^*$ για το σωματίδιο $i$ | 21 |
| Εικόνα 6.1 | Παγκόσμιος χάρτης με τα λιμάνια που εξυπηρετούν πλοία υπερατλαντικών ταξιδιών   | 40 |
| Εικόνα 6.2 | Κατανομή των εμπορευμάτων στα πλοία βάσει της αρχικής λύσης   | 45 |
| Εικόνα 6.3 | Κατανομή των εμπορευμάτων στα πλοία βάσει της ολικά βέλτιστης λύσης   | 46 |
| Εικόνα 6.4 | Σύγκριση κόστους ανά λύση   | 47 |
| Εικόνα Α'1 | Παράδειγμα για "Μικρά"  | 53 |
| Εικόνα Α'2 | Παράδειγμα για "Μεσαία"   | 53 |
| Εικόνα Α'3 | Παράδειγμα για "Μεγάλα"   | 55 |
| Εικόνα Α'4 | Παράδειγμα για "Πολύ μεγάλα"  | 56 |

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

---

|             |   |    |
|-------------|---|----|
| Πίνακας 6.1 | Υποκατηγορίες δεδομένων                           | 40 |
| Πίνακας 6.2 | Αποτελέσματα υποκατηγορίας "Μικρά"                | 41 |
| Πίνακας 6.3 | Αποτελέσματα υποκατηγορίας "Μεσαία"               | 42 |
| Πίνακας 6.4 | Αποτελέσματα υποκατηγορίας "Μεγάλα"               | 43 |
| Πίνακας 6.5 | Αποτελέσματα υποκατηγορίας "Πολύ μεγάλα"          | 44 |
| Πίνακας Α'1 | Αποτελέσματα έρευνας για την αξιοπιστία του ACPSO | 54 |

## ΚΑΤΑΛΟΓΟΣ ΑΛΓΟΡΙΘΜΩΝ

---

|             |  |    |
|-------------|--|----|
| Algorithm 1 | Ο γενικός αλγόριθμος GRASP             | 19 |
| Algorithm 2 | Ο γενικός αλγόριθμος του κλασσικού PSO | 22 |
| Algorithm 3 | Ο αλγόριθμος ACPSO                     | 34 |



## AKRONYMIA

---

ACPSO Adaptive Constriction PSO

CPSO Constriction PSO

GRASP Greedy Randomized Adaptive Search Procedure

ITSRSP Industrial and Tramp Ship Routing and Scheduling Problem

LS Local Search

PSO Particle Swarm Optimization

RCL Restricted Candidate List

WEGRASP Weight Enhanced GRASP



Μέρος Ι

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ



Στην σύγχρονη εποχή όπου κυριαρχεί η παγκοσμιοποίηση, αποδοτικά και οικονομικά μέσα μεταφοράς που συνδέουν τη διεθνή εφοδιαστική αλυσίδα, αποτελούν την κινητήριο δύναμη της οικονομικής ανάπτυξης και ευημερίας. Με το 80 τοις εκατό των εμπορευμάτων κατ' όγκο να μεταφέρονται μέσω θαλάσσης και να διαχειρίζονται από λιμάνια παγκοσμίως, η σπουδαιότητα της οικονομικής στρατηγικής της εν πλώ εφοδιαστικής αλυσίδας ως υποκινητή του διεθνές εμπορίου, είναι αδιαμφισβήτητη. Η εμπορική ανταγωνιστικότητα όλων των χωρών, αναπτυσσόμενων ή αναπτυσσόμενων, βασίζεται σε μεγάλο βαθμό στην αποτελεσματική πρόσβαση σε διεθνείς ναυτιλιακές υπηρεσίες και λιμενικά δίκτυα.

Στις αρχές του 2020, ο συνολικός παγκόσμιος στόλος αριθμούσε 98.140 εμπορικά πλοία, με συνολική χωρητικότητα που ξεπερνά τους 2 δισεκατομμύρια τόνους. Το 2019 ο διεθνής εμπορικός στόλος αυξήθηκε κατά 4,1 τοις εκατό, αντιστοιχώντας στη μεγαλύτερη άνοδο πριν το 2014. Τα πλοία υγροποιημένου αερίου (Liquefied Gas Carriers) γνώρισαν τη μεγαλύτερη ανάπτυξη, ακολουθούμενα από τα Tankers, τα ξηρού φορτίου (bulk carriers) και τέλος τα Container πλοία [24].

Η πανδημία του κορονοϊού (COVID-19) ανέδειξε την αλληλεξάρτηση των εθνών, και έφερε στο προσκήνιο τη σημασία της εν πλώ εφοδιαστικής αλυσίδας ως απαραίτητος τομέας για τη συνεχή παροχή απαραίτητων εφοδίων, σε καθεστώς κρίσης. Η υγειονομική και οικονομική κρίση που προκλήθηκε από την πανδημία, ανέτρεψε τα δεδομένα για τις ναυτιλιακές εταιρείες, καθώς εκτιμάται ότι ο όγκος του διεθνές εμπορίου θα πέσει κατά 4,1 τοις εκατό. Λαμβάνοντας υπόψη προηγούμενες οικονομικές κρίσεις, όπως αυτή του 2008, έχει αποδειχθεί ότι η μείωση της ζήτησης ναυτιλιακών υπηρεσιών, οδηγούν στην υποτίμηση των ναύλων [23]. Συνεπώς, για να παραμείνει κερδοφόρα μια ναυτιλιακή εταιρεία στα σύγχρονα δεδομένα, ο αποτελεσματικός προγραμματισμός της εφοδιαστικής της αλυσίδας, κρίζεται πιο σημαντικός από ποτέ.

Τα είδη ναυτιλιακών υπηρεσιών χωρίζονται σε Liner, Industrial και Tramp shipping [5]. Η παρούσα διπλωματική εργασία, επικεντρώνεται στον χρονικό προγραμματισμό και στη δρομολόγηση των εμπορευμάτων για τις υπηρεσίες του Industrial και Tramp, διότι πρακτικά οι προκλήσεις που καλούνται να αντιμετωπίσουν παρουσιάζουν αρκετές ομοιότητες. Ο στόχος του Industrial and Tramp Ship Routing and Scheduling Problem (ITSRSP) είναι διπλός. Αφενός, εστιάζεται στην ελαχιστοποίηση του κόστους λειτουργίας των πλοίων, και αφετέρου στη μεγιστοποίηση των εσόδων από την κατάλληλη επιλογή των εξυπηρετηθέντων εμπορευμάτων.

Στην πράξη, το πρόβλημα του [ITSRSP](#) βρίσκει περισσότερο εφαρμογή στις μεταφορές ακατέργαστων πετρελαίων και ξηρών φορτίων, με αυτά να αποτελούν παραπάνω από το 60 τοις εκατό του συνολικού φορτίου που μεταφέρεται σε διεθνή ύδατα. Η κατηγορία που αντιπροσωπεύει καλύτερα το παραπάνω σενάριο, και επιλέχθηκε ως το επίκεντρο της εν λόγω έρευνας, περιλαμβάνει υπερατλαντικά ταξίδια (Deepsea) πλήρους φορτίου (Full Load) που διεξάγονται από έναν ετερογενή στόλο πλοίων.

Για την επίλυση του προβλήματος [ITSRSP](#), εξαιτίας της πολυπλοκότητας των περιορισμών που παρουσιάζει, απαιτεί την εφαρμογή και υλοποίηση μεθοδολογιών βελτιστοποίησης από την κατηγορία των μεθευρετικών αλγορίθμων. Το κύριο δομικό τους στοιχείο αποτελεί ο εμπνευσμένος από τη φύση, αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων - Particle Swarm Optimization ([PSO](#)). Στα κεφάλαια που ακολουθούν θα εξεταστούν οι πολυσύνθετες λεπτομέρειες που χαρακτηρίζουν το πρόβλημα του [ITSRSP](#) και τον αλγόριθμο του [PSO](#), καθώς και οι απαραίτητες τροποποιήσεις των μεθόδων για την επίλυση του προαναφερθέντος προβλήματος.

## 2.1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΠΛΟΙΩΝ

Στον χώρο της ναυτιλίας υπάρχουν πολλές κατηγορίες πλοίων που ξεχωρίζουν από το μέγεθος και τη χρήση τους [4]. Το μέγεθος ενός πλοίου σχετίζεται με το ωφέλιμο βάρος ή τον όγκο των εμπορευμάτων που είναι ικανό να φορτώσει. Για παράδειγμα, ένα εμπορεύμα με χαμηλό βάρος κατ' όγκο, γεμίζει το πλοίο πριν φτάσει το ανώτατο όριο βάρους σε κιλά που μπορεί να φορτώσει. Τα πλοία χωρίζονται και σε διαφορετικούς τύπους:

- Τάνκερς (Tankers) είναι τα πλοία που σχεδιάστηκαν για τη μαζική μεταφορά εμπορευμάτων υγρής μορφής. Τα μεγαλύτερου μεγέθους Τάνκερς συνήθως μεταφέρουν ακατέργαστο πετρέλαιο, ενώ τα μικρότερα, παράγωγα πετρελαίου, χημικά και άλλα ρευστά.
- Μπάλκ κάριερ (Bulk carriers) τα οποία μεταφέρουν μαζικά εμπορεύματα ξηρού τύπου όπως σίδηρο, άνθρακα, σιτηρά, βωξίτη, αλουμίνια, φωσφορικό άλας και άλλα μέταλλα.
- Πλοία υγροποιημένου αερίου (Liquefied Gas Carriers) τα οποία μεταφέρουν ψυγμένα αέρια υπό υψηλή πίεση.
- Πλοία εμπορευματοκιβωτίων (Container Ships) τα οποία μεταφέρουν μεταλλικά εμπορευματοκιβώτια στα οποία αποθηκεύονται πακεταρισμένα εμπορεύματα.
- Γενικού φορτίου (General Cargo) τα οποία μεταφέρουν στα αμπάρια και στο κατάστρωμα φορτία παντός τύπου, συνήθως συσκευασμένα. Καθώς όμως η μεταχείριση γενικού φορτίου είναι κοπιώδης και χρονοβόρα, τα τελευταία πενήντα χρόνια επικράτησαν σταδιακά τα εμπορευματοκιβώτια, μειώνοντας έτσι τον χρόνο εκφόρτωσης από μέρες σε ώρες.
- Πλοία "ψυγεία" (Refrigerated vessels - reefers) είναι σχεδιασμένα για τη μεταφορά εμπορευμάτων που απαιτούν χαμηλές ή ελεγχόμενες θερμοκρασίες όπως ψάρια, κρέας, φρούτα, αλλά είναι ικανά να μεταφέρουν και φορτία γενικού τύπου.
- Roll-on-Roll-off (Ro-Ro) πλοία, τα οποία έχουν ειδικές ράμπες για να μπορούν φορτηγά και οχήματα να οδηγούν από και προς αυτό.
- Τέλος, υπάρχουν και συνδυασμοί των παραπάνω τύπων όπως γενικού τύπου με ψυχανκικά μέρη ή επιβατικά με Ro-Ro.

## 2.2 ΤΥΠΟΙ ΝΑΥΤΙΛΙΑΚΩΝ ΥΠΗΡΕΣΙΩΝ

Τρεις είναι οι βασικοί τύποι λειτουργίας των εμπορικών πλοίων: Liner (δρομολόγια γραμμής), Tramp (ελεύθερου προγράμματος) και Industrial (βιομηχανικά). Τα Liner λειτουργούν βάση ενός σταθερού προγράμματος, παρόμοιου με τον τρόπο λειτουργίας ενός λεωφορείου, και τα είδη πλοίων που συνήθως χρησιμοποιούν είναι εμπορευματοκιβωτίων και γενικού φορτίου. Τα Tramp λειτουργούν βάση των διαθέσιμων εμπορευμάτων, όπως θα λειτουργούσε ένα ταξί. Συνήθως ένα πλοίο αυτού του είδους λειτουργεί με συμβόλαιο ναύλωσης. Το συμβόλαιο καθορίζει τις ποσότητες εμπορευμάτων που πρέπει να μεταφερθούν μεταξύ συγκεκριμένων λιμανιών, μέσα σε ένα προκαθορισμένο χρονικό περιθώριο, για ένα συγκεκριμένο αντίτιμο ανά μονάδα φορτίου. Τα είδη πλοίων που συνήθως χρησιμοποιούν είναι τάνκερ και μπάλκ κάριερ ξηρού φορτίου. Ο στόχος ενός αποφασίζων για τη λειτουργία ενός Tramp ή Liner δρομολογίου είναι η μεγιστοποίηση των κερδών ανά μονάδα χρόνου. Αντιθέτως, μια εταιρία που σχεδιάζει Industrial δρομολόγια, καλείται συνήθως να μεταφέρει τα δικά της εμπορεύματα από πλοία που η ίδια ελέγχει. Τα πλοία που χρησιμοποιούνται μπορεί να της ανήκουν ή μπορεί να έχει γίνει ναύλωση κάποιων Tramp πλοίων για να καλυφθούν πιθανά κενά. Στόχος είναι η ελαχιστοποίηση του κόστους λειτουργίας. Τα εμπορεύματα που συνήθως μεταφέρουν είναι υπέρογκα ρευστά και μαζικά ξηρά φορτία.

Στην περίπτωση που μια εταιρία οποιουδήποτε τύπου έχει λιγότερα πλοία απ' ό,τι χρειάζεται για να εκπληρώσει τα απαραίτητα δρομολόγια, έχει τη δυνατότητα να ναυλώσει επιπλέον πλοία. Εταιρείες τύπου Tramp και Liner, μπορεί να επιλέξουν να μη μεταφέρουν κάποια πλεονάζοντα φορτία μειώνοντας τα δυνητικά τους έσοδα, οι Industrial εταιρίες όμως είναι αναγκασμένες να μεταφέρουν όλα τους τα εμπορεύματα. Στην αντίθετη περίπτωση, αν υπάρχει πλεόνασμα πλοίων, μπορούν να διατεθούν για ναύλωση σε άλλες εταιρείες, να μείνουν αδρανή ή ακόμα και να πουληθούν.

Οι Industrial εταιρίες συνήθως αποσκοπούν στη μείωση του ρίσκου και γι' αυτό τείνουν να μη ναυλώνουν τα δικά τους πλοία, να αναθέτουν λιγότερα πλοία στον στόλο τους απ' ό,τι θα συνιστούσε η προβλεπόμενη μακροπρόθεσμη ζήτηση και να καλύπτουν τα βραχυπρόθεσμα κενά τους με ναύλωση Tramp πλοίων από άλλες εταιρείες. Εποχιακή ζήτηση και άλλοι παράγοντες αβεβαιότητας όπως κόστη ναύλωσης και αγοράς πλοίων, επηρεάζουν το μέγεθος του κάθε στόλου.

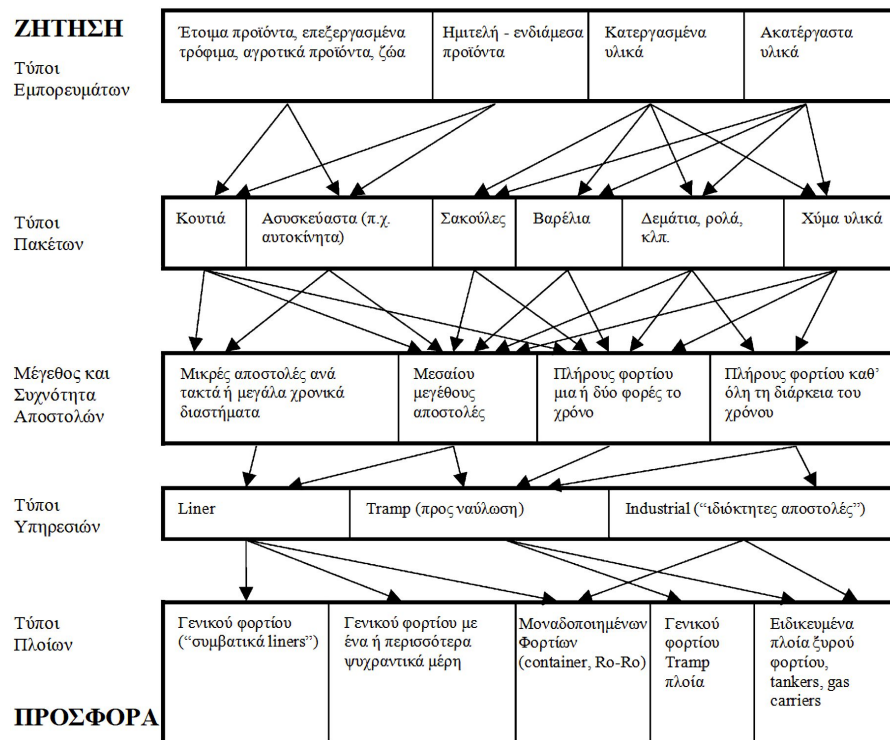
## 2.3 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΦΟΡΤΙΩΝ

Τα πλοία μεταφέρουν μια πληθώρα διαφορετικών εμπορευμάτων. Αυτά μπορεί να είναι παρασκευασμένα αγαθά, επεξεργασμένα ή μη τρόφιμα, ζώα, βιομηχανικός εξοπλισμός, κατεργασμένα ή ακατέργαστα υλικά. Μπορεί να είναι ασυσκευάστα, συσκευασμένα



σε κουτιά, σακούλες, βαρέλια, ή να μεταφέρονται χονδρικώς. Κάποιες φορές τα εμπορεύματα ομαδοποιούνται ως τυποποιημένες μονάδες σε παλέτες, εμπορευματοκιβώτια (containers) ή ρυμουλκούμενα (trailers). Γενικά για να επιτευχθεί αποδοτικότερη μεταχείρισή τους, προτιμάται η μεταφορά τους σε μεγαλύτερες ποσότητες ή χονδρικώς. Όπως αναφέρθηκε και παραπάνω, τα εμπορεύματα που χρειαζόντουσαν χρονοβόρο και κοπιώδη χειρισμό σταδιακά ομαδοποιήθηκαν σε εμπορευματοκιβώτια, επιτρέποντάς τη φόρτωση - εκφόρτωσή τους από γερανούς, μειώνοντας τον χρόνο, το κόστος και τις μικροκλοπές. Το μήκος ενός τυπικού εμπορευματοκιβωτίου είναι 20ft ή 40ft (1ft = 33cm) . Ένα 20ft. εμπορευματοκιβώτιο μπορεί να χωρέσει περίπου 28 τόνους ή 1000ft πόδια όγκο φορτίου.

Τα φορτωμένα εμπορεύματα συνήθως χρειάζονται κατάλληλη ταξινόμηση ώστε να διασφαλιστεί η σταθερότητα του πλοίου κατά το ταξίδι, κατάλληλη θερμοκρασία ή ειδική μεταχείριση. Διαφορετικά φορτία μπορεί να έχουν διαφορετικές πυκνότητες και έτσι το πλοίο μπορεί να γεμίσει είτε λόγω βάρους, όγκου, είτε βάση κάποιας άλλης ειδικής μετρικής.



Εικόνα 2.1: Χαρακτηριστικά προσφοράς και ζήτησης της εν πλω εφοδιαστικής αλυσίδας.

## 2.4 ΓΕΩΓΡΑΦΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Τα ταξίδια των πλοίων μπορούν να κατηγοριοποιηθούν ανάλογα με τα γεωγραφικά τους χαρακτηριστικά ως βαθιάς (deep-sea) ή

ρηχής θάλασσας (short-sea), παρακτιακά (coastal) ή χερσαίων καναλιών (inland waterways). Οικονομίες κλίμακας στα ναυτιλιακά ενθαρρύνουν τη χρήση μεγαλύτερων πλοίων για μεταφορές βαθιάς θάλασσας ανάμεσα σε ηπείρους, ενώ μικρότερα πλοία χρησιμοποιούνται σε ρηχές και παραθαλάσσιες περιοχές.

### 3.1 ΠΡΟΒΛΗΜΑΤΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ (VRP)

#### 3.1.1 Το πρόβλημα του πλανόδιου πωλητή (TSP)

Οι Μαρινάκης, Μυγδαλάς και Μαρινάκη [2] ορίζουν το πρόβλημα του πλανόδιου πωλητή (Traveling Salesman Problem) ως την “εύρεση της συντομότερης (σε χρόνο, απόσταση ή άλλο κόστος) διαδρομής για ένα όχημα (ή πωλητή) με αφετηρία κάποιο σημείο, π.χ. ένα κέντρο διανομής, και επιστροφή στο ίδιο σημείο αφού επισκεφθεί έναν προκαθορισμένο αριθμό πελατών ακριβώς μία φορά τον καθένα”. Το αποτέλεσμα είναι ένας κύκλος που διέρχεται από του κόμβους που αντιστοιχούν στο σημείο αφετηρίας και στους πελάτες ακριβώς μία φορά.

#### 3.1.2 Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων (CVRP)

Το περιορισμένης χωρητικότητας πρόβλημα δρομολόγησης οχημάτων (Capacitated Vehicle Routing Problem) ή απλούστερα το πρόβλημα δρομολόγησης Οχημάτων (Vehicle Routing Problem - (VRP)) [2] αποτελεί επέκταση του προβλήματος του πλανόδιου πωλητή στην περίπτωση που ένα μόνο όχημα δεν αρκεί για να επισκεφθεί όλους τους πελάτες. Αυτό μπορεί να οφείλεται στο ότι η συνολική ζήτηση των πελατών υπερβαίνει την δυναμικότητα του οχήματος, η ζήτηση πρέπει να ικανοποιηθεί μέσα σε ορισμένα χρονικά περιθώρια που είναι αδύνατον να επιτευχθούν από ένα μόνο όχημα, η ζήτηση διαφόρων πελατών αφορά διαφορετικά προϊόντα και αυτά δεν μπορούν να αναμειχθούν. Σε αυτήν την περίπτωση, αναζητούνται πολλαπλοί υπόκυκλοι (κυκλικές διαδρομές), όπου ο καθένας επισκέπτεται ένα υποσύνολο των πελατών. Κάθε πελάτης πρέπει να εξυπηρετείται ακριβώς μια φορά και αυστηρά από μόνο έναν κύκλο. Απώτερος στόχος του προβλήματος συνεχίζει να είναι η ελαχιστοποίηση του κόστους των κυκλικών διαδρομών που εξυπηρετούν τους πελάτες.

#### 3.1.3 Το πρόβλημα δρομολόγησης οχημάτων με δεδομένα χρονικά περιθώρια (VRPTW)

Στο πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα [1] κάθε πελάτης έχει ένα φορτίο που πρέπει να εξυπηρετηθεί και ο πελάτης καθορίζει μια χρονική περίοδο (time window - χρονικό παράθυρο ή περιθώριο)  $[a_i, b_i]$  στην οποία η εξυπηρέτηση πρέπει

να πραγματοποιηθεί. Η εξυπηρέτηση για κάθε πελάτη πρέπει να ξεκινήσει μέσα στο χρονικό παράθυρο που μπορεί ο πελάτης να εξυπηρετηθεί, και το όχημα πρέπει να παραμείνει στην τοποθεσία που βρίσκεται ο πελάτης για χρόνο  $s_i$ . Αν κάποιο όχημα φτάσει σε κάποιον πελάτη νωρίτερα από τον προκαθορισμένο χρόνο, το όχημα πρέπει να παραμείνει στην τοποθεσία του πελάτη μέχρι να ξεκινήσει το χρονικό παράθυρο  $a_i$ . Υπάρχουν δύο ειδών χρονικά παράθυρα. Από τη μία, τα χαλαρά παράθυρα, όπου αν ένα όχημα φτάσει σε έναν πελάτη κάποια χρονική στιγμή εκτός του χρονικού παραθύρου μπορεί να ξεκινήσει την εξυπηρέτηση του εκείνη τη στιγμή, συνήθως καταβάλλοντας κάποιο πρόστιμο, και από την άλλη, τα σκληρά χρονικά παράθυρα που δεν επιτρέπουν να φτάσει το όχημα στον πελάτη μετά από τον βραδύτερο χρόνο εξυπηρέτησης  $b_i$ . Ο σκοπός είναι να βρεθεί ένα σύνολο από διαδρομές για τα οχήματα, όπου κάθε διαδρομή ξεκινάει και τελειώνει στην αποθήκη, εξυπηρετεί ένα υποσύνολο από τους πελάτες χωρίς να παραβιάζεται η χωρητικότητα και οι περιορισμοί από τα χρονικά παράθυρα καθώς ελαχιστοποιείται το συνολικό μήκος των διαδρομών.

### 3.1.4 Το πρόβλημα δρομολόγησης οχημάτων μέσα σε χρονικά περιθώρια και με διανομή και παραλαβή προϊόντων (VRP-PDPTW)

Το πρόβλημα δρομολόγησης οχημάτων με χρονικά παράθυρα επεκτείνεται [1] με την προσθήκη δύο μεταβλητών  $d_i$  και  $p_i$  που αναλογούν στις ποσότητες που πρέπει να διανεμηθούν και να παραληφθούν από τον πελάτη  $i$ , κατά την διάρκεια μιας διαδρομής. Επίσης, για κάθε πελάτη ένας κόμβος  $O_i$  δηλώνει το σημείο από το οποίο πρέπει να γίνει η παραλαβή του προϊόντος που θα διανεμηθεί στον πελάτη, και ένας κόμβος  $D_i$  δηλώνει τον προορισμό του προϊόντος που παραλήφθηκε από τον πελάτη. Πολύ σημαντικός περιορισμός της διαδικασίας αυτής είναι η παραλαβή προϊόντων από έναν πελάτη να έπεται της διανομής, έτσι η συνολική φόρτωση ενός οχήματος όταν φτάσει στον πελάτη  $i$  είναι ίση με την αρχική φόρτωση του οχήματος μείον το άθροισμα των προϊόντων που παρέδωσε στους πελάτες πριν τον  $i$  συν το άθροισμα των προϊόντων που παρέλαβε από τους πελάτες πριν τον  $i$ .

## 3.2 TO INDUSTRIAL AND TRAMP SHIP ROUTING AND SCHEDULING PROBLEM (ITSRSP)

Το [ITSRSP](#) επεκτείνει το πρόβλημα διανομής και παραλαβής με χρονικά παράθυρα (PDPTW) προσθέτοντας ετερογένεια στον στόλο, περιορισμούς συμβατότητας ανάμεσα στα πλοία και τα εμπορεύματα, διαφορετικές αρχικές θέσεις και χρόνους έναρξης για κάθε πλοίο, και ευελιξία εξυπηρέτησης με πρόστιμα.

Το πρόβλημα του **ITSRSP** μπορεί να παρασταθεί ως ένας πλήρης γράφος  $G = (N, A)$ , όπου  $N$  είναι η ένωση των κόμβων φόρτωσης  $P = 1, \dots, n$ , των κόμβων εκφόρτωσης  $D = n + 1, \dots, 2n$ , και των αρχικών θέσεων  $o(v) = o1, \dots, ov$ . Μια εταιρία τύπου Industrial ή Tramp, έχει στη διάθεσή της ένα στόλο  $V$  πλοίων  $V = 1, \dots, v$  με χωρητικότητα  $K_v$ , και  $n$  εμπορεύματα είναι διαθέσιμα προς μεταφορά. Κάθε εμπόρευμα  $i \in 1, \dots, n$  χαρακτηρίζεται από ένα βάρος  $Q_i$ . Το πλοίο φορτώνει ( $Q_i \geq 0$ ) το εμπόρευμα από το λιμάνι φόρτωσης  $i \in P$  και πρέπει να το μεταφέρει στο αντίστοιχο λιμάνι εκφόρτωσης  $n + i \in D$  και να το ξεφορτώσει ( $Q_{n+i} = -Q_i$ ). Το βάρος του πλοίου  $v$ , αφού εξυπηρετήσει έναν κόμβο  $i$ , συμβολίζεται με  $l_{iv}$ . Κάθε κόμβος  $i \in P \cup D$  συνδέεται με ένα αυστηρό χρονικό παράθυρο επιτρεπτών επισκέψεων  $[a_i, b_i]$ . Η χρονική στιγμή που ξεκινάει η εξυπηρέτηση στο κόμβο  $i$  από το πλοίο  $v$ , αντιπροσωπεύεται από τη μεταβλητή  $t_{iv}$ . Κάθε πλοίο  $v \in V$  γίνεται διαθέσιμο τη χρονική στιγμή  $s_{ov}^d$ , στη θέση  $o(v)$ . Μπορεί να διασχίσει κάθε τόξο  $(i, j) \in A_v$  με κόστος  $C_{ijv}$  (συμπεριλαμβανομένων των καυσίμων και το κόστος καναλιών) και διάρκεια  $T_{ijv}$ . Για κάθε επίσκεψη ενός πλοίου  $v \in V$  σε ένα κόμβο  $i \in P \cup D$ , υπάρχει ένα ανάλογο κόστος εξυπηρέτησης  $S_{iv}^c \geq 0$  και διάρκεια  $S_{iv}^D \geq 0$ . Είναι πιθανό κάποια εμπορεύματα να είναι ασύμβατα με κάποια πλοία. Για κάθε  $i \in 1, \dots, n$  και  $v \in V$ , ένας πίνακας Μπούλ  $I^{iv}$  καθορίζει αν το εμπόρευμα  $i$  μπορεί να φορτωθεί από το πλοίο  $v$ . Έστω λοιπόν ότι  $N_v$  είναι το σύνολο των κόμβων που μπορεί να επισκεφτεί ένα πλοίο,  $N_v^P = N^P \cap N_v$  και  $N_v^D = N^D \cap N_v$  είναι οι εφικτοί κόμβοι φόρτωσης και εκφόρτωσης για το πλοίο  $v$ , αντίστοιχα. Μία δυαδική μεταβλητή ροής  $x_{ijv}$  δηλώνει αν το πλοίο  $v$  θα μετακινηθεί κατευθείαν από τον κόμβο  $i$  στον κόμβο  $j$ , ενώ η δυαδική μεταβλητή  $y_i$  δηλώνει αν το εμπόρευμα  $i$  εξυπηρετείται από τον στόλο της εταιρίας ή όχι. Τέλος, ένα πρόστιμο  $C_i^S$  πληρώνεται, αν το εμπόρευμα  $i$  δε μεταφερθεί από τον στόλο. Το πρόστιμο αντιστοιχεί στο έλλειμμα (ή κόστος ναύλωσης) λόγω της αδυναμίας μεταφοράς ενός εμπορεύματος.

Ο στόχος του **ITSRSP** είναι η διαμόρφωση διαδρομών που ελαχιστοποιούν το άθροισμα του κόστους ταξιδιού τους, και ταυτόχρονα το πιθανό πρόστιμο που πρέπει να πληρωθεί στην περίπτωση που γίνει ναύλωση επιπλέον πλοίων ή εμπορεύματα δε μεταφερθούν. Οι διαδρομές ξεκινούν από τα αντίστοιχα σημεία έναρξης αλλά δεν έχουν προκαθορισμένο σημείο τερματισμού  $d(v)$ , καθώς τα πλοία λειτουργούν αδιάκοπα. Κάθε διαδρομή πρέπει να είναι εφικτή ως προς του περιορισμούς. Δηλαδή, η φόρτωση - εκφόρτωση των εμπορευμάτων μπορεί να γίνει αυστηρά και μόνο εντός των χρονικών τους παραθύρων και δεν μπορούν να εξυπηρετηθούν από ασύμβατα πλοία. Επιπλέον, ο περιορισμός “ζεύγους” διασφαλίζει πως κάθε ζευγάρι ( $i \in P, n + i \in D$ ) θα ανήκει στην ίδια διαδρομή, και ο περιορισμός “προτεραιότητας” πως κάθε διαδικασία φόρτωσης  $i \in P$  θα συμβεί πριν την εκφόρτωση  $n + i \in D$ .

Η μαθηματική μοντελοποίηση του προβλήματος είναι η ακόλουθη:

$$\min \sum_{v \in V} \sum_{(i,j) \in A_v} C_{ijv} x_{ijv} + \sum_{i \in N^P} C_i^S y_i \quad (3.1)$$

υπό τους περιορισμούς

$$\sum_{v \in V} \sum_{j \in N_v} x_{ijv} + y_i = 1, \quad i \in N^P \quad (3.2)$$

$$\sum_{j \in N_v} x_{o(v)jv} = 1, \quad v \in V \quad (3.3)$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{jiv} = 0, \quad v \in V, i \in N_v \setminus \{o(v), d(v)\} \quad (3.4)$$

$$\sum_{j \in N_v} x_{jd(v)v} = 1, \quad v \in V \quad (3.5)$$

$$l_{iv} + Q_j - l_{jv} \leq K_v(1 - x_{ijv}), \quad v \in V, j \in N_v^P, (i, j) \in A_v \quad (3.6)$$

$$l_{iv} - Q_j - l_{(n+j)v} \leq K_v(1 - x_{i(j+n)v}), \quad v \in V, j \in N_v^P, (i, n+j) \in A_v \quad (3.7)$$

$$0 \leq l_{iv} \leq K_v, \quad v \in V, i \in N_v^P \quad (3.8)$$

$$t_{iv} + T_{ijv} - t_{jv} \leq (b_i + T_{ijv})(1 - x_{ijv}), \quad v \in V, (i, j) \in A_v \quad (3.9)$$

$$\sum_{j \in N_v} x_{ijv} - \sum_{j \in N_v} x_{(n+i)jv} = 0, \quad v \in V, i \in N_v^P \quad (3.10)$$

$$t_{iv} + T_{i(n+i)v} - t_{(n+i)v} \leq 0, \quad v \in V, i \in N_v^P \quad (3.11)$$

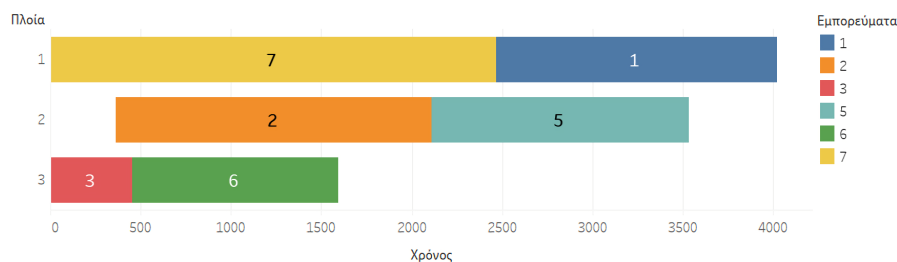
$$a_i \leq t_{iv} \leq b_i, \quad v \in V, i \in N_v \quad (3.12)$$

$$y_i \in \{0, 1\}, \quad i \in N^C \quad (3.13)$$

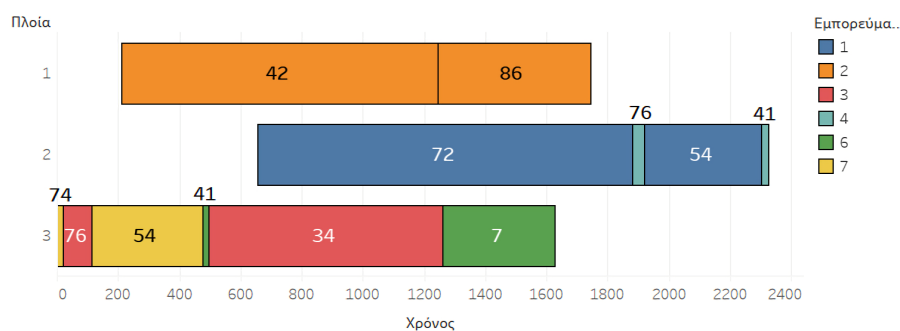
$$x_{ijv} \in \{0, 1\}, \quad v \in V, (i, j) \in A_v \quad (3.14)$$

Η αντικειμενική συνάρτηση (3.1) ελαχιστοποιεί τα κόστη λειτουργίας του στόλου και ταυτόχρονα τα πρόστιμα που πιθανώς πρέπει να πληρωθούν. Ο περιορισμός (3.2) εξασφαλίζει ότι όλα τα εμπορεύματα, είτε θα εξυπηρετηθούν από τον στόλο είτε από κάποια άλλη εταιρία. Οι κινήσεις των πλοίων καθορίζονται από τους περιορισμούς (3.3) - (3.5), διασφαλίζοντας ότι όλα τα πλοία θα φύγουν από τις αρχικές τους θέσεις, όλα τα πλοία που φτάνουν σε ένα λιμάνι μπορούν να φύγουν από αυτό, και όλα τα πλοία φτάνουν στον τελικό τους προορισμό αντίστοιχα. Το φορτίο του πλοίου κατά τη φόρτωση και εκφόρτωση, ελέγχεται από τους περιορισμούς (3.6) και (3.7) αντίστοιχα. Ο περιορισμός (3.8) διασφαλίζει πως το βάρος ενός εμπορεύματος δεν ξεπερνάει τη χωρητικότητα του πλοίου που προσπαθεί να το φορτώσει. Ο περιορισμός (3.9) εξασφαλίζει πως ο χρόνος έναρξης εξυπηρέτησης είναι εφικτός αναφορικά με τις διάρκειες των μεταφορών. Οι εξισώσεις (3.10) και (3.11) διασφαλίζουν τους περιορισμούς “ζεύγους” και “προτεραιότητας” αντίστοιχα. Τα χρονικά παράθυρα ελέγχονται από τον περιορισμό (3.12). Τέλος, όλες οι μεταβλητές ροής και εξυπηρέτησης πρέπει να παίρνουν τιμές 0 ή 1, όπως φαίνεται από τους περιορισμούς (3.13) και (3.14).

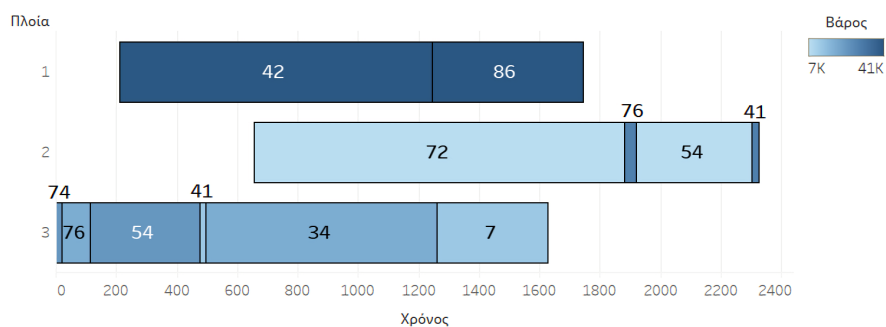
**ΠΑΡΑΔΕΙΓΜΑ** Στην εικόνα 3.1 παρουσιάζεται μια λύση<sup>1</sup> πλήρους φορτίου (full load) για υπερατλαντικά ταξίδια (deep sea), που σχεδιάστηκε βάσει της παραπάνω μοντελοποίησης. Σε αυτή την κατηγορία, το πλοίο γεμίζει πλήρως με την φόρτωση ενός εμπορεύματος, και άρα η εκφόρτωσή του πρέπει να γίνει αμέσως μετά την φόρτωση. Το μέγεθος του στόλου ανέρχεται στα τρία πλοία τα οποία γίνονται διαθέσιμα για αξιοποίηση σε διαφορετικές χρονικές στιγμές. Σε αυτά κατανέμονται τα 7 διαθέσιμα εμπορεύματα και το πλάτος κάθε εμπορεύματος δηλώνει τον χρόνο εξυπηρέτησής του. Από εδώ και πέρα η ανάλυση του προβλήματος πλήρους φορτίου θα γίνεται σε επίπεδο φορτίου, για λόγους ευκολίας και σαφήνειας.



Εικόνα 3.1: Λύση πλήρους φορτίου υπερατλαντικών ταξιδιών



(α') Κατανομή εμπορευμάτων στα πλοία



(β') Βάρος εμπορευμάτων

Εικόνα 3.2: Λύση μικτού φορτίου υπερατλαντικών ταξιδιών

<sup>1</sup> DEEPSEA\_FUN\_C8\_V3\_HE2 με απόκλιση 0%



Στην εικόνα 3.2 παρατίθεται για σύγκριση μια λύση<sup>2</sup> του μικτού φορτίου (mixed load) υπερατλαντικών ταξιδιών (deep sea). Σε αυτή την κατηγορία, ένα πλοίο μπορεί να έχει φορτωμένο ανα πάσα στιγμή πάνω από ένα εμπορεύμα. Στην υπο-εικόνα 3.2α' αναπαρίστανται τα εμπορεύματα που εξυπηρετεί το κάθε πλοίο με διαφορετικό χρώμα, και ο αριθμός που περικλείουν αντιστοιχεί στα λιμάνια φορτοεκφόρτωσης. Η πρώτη φορά που εμφανίζεται ένα χρώμα, δηλώνει την φόρτωση του εμπορεύματος και η δεύτερη την εκφόρτωσή του. Στην υπο-εικόνα 3.2β', η ένταση του χρώματος υποδηλώνει το βάρος του αντίστοιχου φορτίου. Η ανάλυση του προβλήματος μικτού φορτίου πρέπει να γίνεται αναγκαστικά σε επίπεδο λιμανιού. Αξίζει να σημειωθεί πως, απαραίτητη προϋπόθεση για την επίτευξη βέλτιστου κόστους και στις δύο παραπάνω λύσεις, ήταν η ανάθεση ενός εμπορεύματος προς ναύλωση.

---

2 DEEPSEA\_MUN\_C7\_V3\_HE\_2 με απόκλιση 0.0038%



## ΚΑΤΗΓΟΡΙΕΣ ΑΛΓΟΡΙΘΜΩΝ ΚΑΙ ΜΕΘΟΔΟΛΟΓΙΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Κάθε πρόβλημα συνδυαστικής βελτιστοποίησης περιλαμβάνει ένα πεπερασμένο αριθμό εναλλακτικών. Δεδομένου δηλαδή, ενός συνόλου επιλογών  $E = \{1, \dots, n\}$  και μιας αντικειμενικής συνάρτησης  $f : 2^E \rightarrow \mathbb{R}$ , το σύνολο εφικτών λύσεων  $X \subseteq 2^E$  είναι πεπερασμένο. Στην περίπτωση ενός προβλήματος ελαχιστοποίησης, γίνεται αναζήτηση για μία βέλτιστη λύση  $x^* \in X$  τέτοια ώστε  $f(x^*) \leq f(x)$  (ή  $f(x^*) \geq f(x)$  για πρόβλημα μεγιστοποίησης),  $\forall x \in X$ .

Θεωρητικά, αφού το σύνολο των εφικτών λύσεων είναι πεπερασμένο, κάθε πρόβλημα συνδυαστικής βελτιστοποίησης θα μπορούσε να λυθεί από έναν ακριβή (Exact) αλγόριθμο που απλά απαριθμεί όλα τα στοιχεία στο  $X$  και επιλέγει ένα από αυτά που οδηγεί στο βέλτιστο κόστος της αντικειμενικής συνάρτησης. Δυστυχώς, αφού ο αριθμός των εφικτών εναλλακτικών λύσεων  $|X|$  συνήθως αυξάνεται με εκθετικό ρυθμό σε σχέση με το μέγεθος του προς επίλυση προβλήματος, μια τόσο απλή μέθοδος δεν είναι αποδοτική και σίγουρα όχι εφαρμόσιμη στην επίλυση πραγματικών προβλημάτων πρακτικού ενδιαφέροντος. [8]

### 4.1 ΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Οι Ευρετικοί, είναι μια αρκετά διαδεδομένη κατηγορία αλγορίθμων. Χρησιμοποιούν κάποια γνώση για το προς επίλυση πρόβλημα και με τη μέθοδο δοκιμής και σφάλματος προσπαθούν να παράξουν μια εφικτή λύση. Σε αντίθεση με τους ακριβείς αλγορίθμους, η ποιότητα της λύσης δεν εγγυάται ότι θα είναι η βέλτιστη. Όσο όμως αυξάνει το μέγεθος του προβλήματος, γίνεται ολοένα και δυσκολότερη η ακριβής επίλυσή του. Για να επιλύσουμε προβλήματα αυτής της μορφής συχνά καταφεύγουμε σε αυτές τις τεχνικές που μας οδηγούν σε ικανοποιητικά, σχεδόν βέλτητα αποτελέσματα σε λογικά χρονικά περιθώρια.

Γιατί και τότε χρησιμοποιούνται οι ευρετικοί αλγόριθμοι[26]:

1. Ανακριβή ή περιορισμένα δεδομένα χρησιμοποιούνται για τον υπολογισμό του μοντέλου.
2. Χρησιμοποιείται ένα απλουστευμένο, μη αντιπροσωπευτικό του πραγματικού προβλήματος, μοντέλο. Και στις δύο περιπτώσεις (1) & (2), μια γρήγορη υποβέλτιστη λύση, είναι πολύ πιο δελεαστική απέναντι σε ένα πολύπλοκο και χρονοβόρο μοντέλο.

3. Δεν υπάρχει διαθέσιμο αξιόπιστο μοντέλο. Πιθανά, μια μικρή εταιρία να μην έχει πρόσβαση σε προηγμένα προγράμματα βελτιστοποίησης.
4. Το υπάρχον ακριβές μοντέλο είναι υπολογιστικά απαγορευτικό εξαιτίας χρονικών και αποθηκευτικών απαιτήσεων. Πραγματικά προβλήματα πολλαπλής πολυπλοκότητας μπορεί να καταστήσουν την εύρεση βέλτιστης ή έστω εφικτής λύσης, σε λογικά χρονικά περιθώρια, αδύνατη. Αντιθέτως, ένας απλός ευρετικός, μπορεί τουλάχιστον να βγάλει εφικτές λύσεις, χωρίς υψηλές χρονικές και αποθηκευτικές απαιτήσεις.
5. Για να βελτιωθεί η απόδοση πολυπλοκότερων αλγορίθμων, ευρετικοί αλγόριθμοι αξιοποιούνται για την κατασκευή αρχικών λύσεων ή την καθοδήγηση της αναζήτησης και μείωσης των πιθανών εναλλακτικών λύσεων (π.χ. branch-and-bound).
6. Διαρκής ανάγκη επίλυσης του ίδιου προβλήματος σε πραγματικό χρόνο που μακροπρόθεσμα, η ταχύτητα ενός ευρετικού, μπορεί να οδηγήσει σε σημαντική εξοικονόμηση χρόνου.
7. Η παραχθείσα λύση μπορεί να είναι ικανοποιητική για τον αποφασίζων, αν ανατρέπει τα προϋπάρχοντα δεδομένα.
8. Είναι τόσο απλοί που ακόμα και ένας ανειδίκευτος χρήστης μπορεί να τους κατανοήσει, να τους εμπιστευτεί και άρα να εφαρμόσει τις προτεινόμενες αλλαγές με μεγαλύτερη ευκολία.
9. Άλλοι περιορισμοί πόρων, όπως χρόνος, κεφάλαιο, διαθέσιμες εργατοώρες και εξειδίκευση.

#### 4.1.1 Αλγόριθμοι απληστίας

Στους άπληστους (Greedy) ή αλλιώς κατασκευαστικούς αλγορίθμους, κάποιος ξεκινάει από το μηδέν (κενή λύση) και εισάγοντας ένα στοιχείο τη φορά προσπαθεί να κατασκευάσει μια πλήρη και εφικτή λύση. Πιο αναλυτικά, σχεδόν σε κάθε πρόβλημα βελτιστοποίησης υπάρχει ένα σύνολο  $n$  στοιχείων - εισόδων (π.χ. Πελάτες προς εξυπηρέτηση, δουλειές προς εκχώρηση σε μηχανές) από το οποίο πρέπει να επιλεγεί ένα υποσύνολο που να ικανοποιεί κάποιους περιορισμούς και ταυτόχρονα να ελαχιστοποιεί ή μεγιστοποιεί μια αντικειμενική συνάρτηση η οποία συνήθως μεταφράζεται σε κέρδος. Σε κάθε επανάληψη ο αλγόριθμος ξεκινάει με μία μερική λύση  $x_1, x_2, x_3, \dots, x_k$  και προκειμένου να την ολοκληρώσει σε μια πλήρη  $x_1, x_2, x_3, \dots, x_k, \dots, x_{k+1}, \dots, x_n$  ( $n > k$ ) επιλέγει μεταξύ όλων των υποψήφιων στοιχείων  $y$  εκείνο που (α) διασφαλίζει την τελική εφικτότητα της λύσης και (β) επικρατεί των υπολοίπων αναφορικά με ένα κριτήριο που δεν ταυτίζεται απαραίτητα με το αντικειμενικό [1].

Οι άπληστοι αλγόριθμοι είναι αρκετά διαδεδομένοι χάρη στην απλή τους δομή και μειωμένη πολυπλοκότητα σε σχέση με άλλους

επαναληπτικούς αλγορίθμους. Η εφαρμογή τους όμως σε πολυσύνθετα πραγματικά προβλήματα, συνήθως απαιτεί πάρα πολύ μεγάλο χρόνο γιατί είναι μυωπικοί, δηλαδή βλέπουν μόνο μπροστά.

#### 4.1.2 Αλγόριθμοι τοπικής αναζήτησης

Οι αλγόριθμοι τοπικής αναζήτησης - Local Search (LS) προσπαθούν από μια αρχική εφικτή λύση, να τη βελτιώσουν με κάποια μέθοδο αναζήτησης στη γειτονιά της. Παρά την απλή τους μορφή, έχουν γνωρίσει εκπληκτική εφαρμογή σε μεγάλο αριθμό προβλημάτων συνδυαστικής βελτιστοποίησης. Πιο συγκεκριμένα, η τοπική αναζήτηση ορίζεται από τους Μαρινάκης και Μυγδαλάς [1] ως εξής:

“Δοθέντος ενός προβλήματος ελαχιστοποίησης  $(F, c)$ , όπου  $F$  είναι ένα εφικτό σύνολο και  $c$  είναι το κόστος, επιλέγουμε τη γειτονιά

$$N : F \rightarrow 2^F$$

στην οποία εφαρμόζεται αναζήτηση για να βρεθεί καλύτερη λύση από την αρχική αν υπάρχει κάποιο σημείο  $s \in N$  που να έχει μικρότερο κόστος από το κόστος που έχουμε αυτή τη στιγμή. Για όσο χρονικό διάστημα μια καλύτερη λύση υπάρχει, αντικαθιστούμε την υπάρχουσα και συνεχίζουμε την αναζήτηση μέχρι το σημείο που θα βρούμε κάποιο τοπικό ελάχιστο και η λύση που έχουμε δε θα βελτιώνεται παραπάνω.”

Παρά τα πλεονεκτήματα που περιγράφηκαν μέχρι τώρα, η αποκλειστική χρήση ευρετικών αλγορίθμων συνεπάγεται και δυνητικά προβλήματα. Όσο αυξάνεται η πολυπλοκότητα και το μέγεθος του προβλήματος μειώνεται η πιθανότητα εύρεσης βέλτιστης ή ακόμα και υποβέλτιστης λύσης. Για μια εταιρία, η μείωση στην ποιότητα της λύσης, μεταφράζεται σε άσκοπες δαπάνες που μακροπρόθεσμα θα οδηγήσουν στην απώλεια του ανταγωνιστικού πλεονεκτήματος.

## 4.2 ΜΕΘΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Για να καταπολεμηθούν οι αδυναμίες των ευρετικών αλγορίθμων, μέθοδοι τοπικής αναζήτησης συνδυάστηκαν με υψηλότερου επιπέδου στρατηγικές για να σχεδιάσουν διαδικασίες με τη δυνατότητα να ξεφεύγουν από τοπικά ελάχιστα. Εξού και το “Μετά” στον όρο “Μεθευρετικός”. Τα τελευταία 30 χρόνια οι μεθευρετικοί αλγόριθμοι γνώρισαν μεγάλη δημοσιότητα, χάρη στη δυνατότητά τους να παράγουν ικανοποιητικές λύσεις για προβλήματα μεγάλης πολυπλοκότητας σε λογικούς χρόνους. Έχουν χρησιμοποιηθεί σε ευρύ

φάσμα προβλημάτων σε διάφορους κλάδους όπως της Μηχανικής, Ρομποτικής, Βιοϊατρικής, Μηχανικής Μάθησης, Εφοδιαστικής Αλυσίδας, αποδεικνύοντας την αποτελεσματικότητά τους.

Είναι εμφανές, από την πληθώρα δημοσιευμένων ερευνών, πως η αποτελεσματική σχεδίαση ενός μεθευρετικού αλγορίθμου μπορεί να αποκληθεί και τέχνη, απαιτώντας υψηλή διαίσθηση εκ' μέρους του ερευνητή. Η σωστή, όμως, επιλογή των παραμέτρων και της δομής του μεθευρετικού, αδιαμφισβήτητα θα επηρεάσουν την απόδοση του στην προσέγγιση και ποιότητα της λύσης. [15]

Τα δύο κυριότερα δομικά στοιχεία των μεθευρετικών είναι η διάχυση (diversification) και η εντατικοποίηση (intensification). Διάχυση, είναι η παραγωγή ποικιλόμορφων λύσεων, για να εξερευνηθούν νέες περιοχές του χώρου αναζήτησης λύσεων. Εντατικοποίηση, είναι η επικέντρωση της αναζήτησης γύρω από ένα τοπικό βέλτιστο, εκμεταλλευόμενοι τη γνώση ύπαρξης μιας καλής λύσης σε αυτή τη γειτονιά. Ενισχύοντας την υπεροχή των καλύτερων λύσεων, εξασφαλίζεται η σύγκλιση σε βέλτιστο, ενώ η διάχυση, μέσω τυχαίων διαταραχών αποτρέπει τον εγκλωβισμό σε τοπικά ελάχιστα και αυξάνει την ποικιλία των λύσεων. Σωστός συνδυασμός των δύο αυτών κομβικών στοιχείων, συνήθως διαβεβαιώνει ότι η προσέγγιση της ολικά βέλτιστης λύσης είναι επιτεύξιμη.

Οι μεθευρετικοί μπορούν να χωριστούν σε δύο γενικές υποκατηγορίες αλγορίθμων, ανάλογα με τον αριθμό των πρακτόρων τους. Στους τροχιακούς (trajectory based) και στους πληθυσμιακούς (population based).

#### 4.2.1 Τροχιακοί (trajectory based)

Στους τροχιακούς, για παράδειγμα, ανήκει ο αλγόριθμος προσομοιωμένης ανόπτωσης (simulated annealing) γιατί χρησιμοποιεί μόνο έναν πράκτορα (λύση) που κινείται στον χώρο των λύσεων βηματικά. Κάθε φορά επιλέγεται μια βελτιωμένη λύση ή μία λίγο χειρότερη βάση κάποιας πιθανότητας. Τα βήματα - κινήσεις του σχηματίζουν μια τροχιά στον χώρο των λύσεων, με μη-μηδενική πιθανότητα να προσεγγίσουν το ολικό βέλτιστο [25].

Στη συνέχεια παρουσιάζεται αναλυτικά ο τροχιακός μεθευρετικός αλγόριθμος άπληστης τυχαιοποιημένης προσαρμοστικής αναζήτησης - Greedy Randomized Adaptive Search Procedure (GRASP) στη γενική του μορφή, παραλλαγή του οποίου χρησιμοποιήθηκε στη διαδικασία επίλυσης του [ITSRSP](#).

##### 4.2.1.1 Άπληστη Τυχαιοποιημένη Προσαρμοστική Αναζήτηση

Ο μεθευρετικός αλγόριθμος [GRASP](#) είναι μία πολυεναρκτήρια επαναληπτική μέθοδος, στην οποία κάθε επανάληψη αποτελείται από δύο φάσεις: την κατασκευαστική (construction phase) και την τοπική αναζήτηση (local search phase).

Στην πρώτη φάση, ο αλγόριθμος ξεκινάει με μια άδεια λύση και μέσω μιας επαναληπτικής διαδικασίας, ένα στοιχείο επιλέγεται τη

φορά, μέχρι να καταλήξει στον σχηματισμό πλήρους λύσης. Σε κάθε επανάληψη, η επιλογή του επόμενου στοιχείου προς εισαγωγή, καθορίζεται από την κατάταξη όλων των υποψηφίων στοιχείων (π.χ. Τα στοιχεία που είναι εφικτό αυτή τη στιγμή να εισαχθούν στη μερική λύση, αναφορικά με την κατάσταση του συστήματος) σε μια λίστα  $C$  βάση μιας συνάρτησης απληστίας  $g : C \rightarrow \mathbb{R}$ . Αυτή η συνάρτηση υπολογίζει το (μυωπικό) όφελος επιλογής κάθε στοιχείου. Ο αλγόριθμος χαρακτηρίζεται προσαρμοστικός, γιατί τα συσχετιζόμενα προνόμια του κάθε στοιχείου ανανεώνονται σε κάθε επανάληψη της κατασκευαστικής φάσης ώστε να αντικατοπτρίζουν τις μεταβολές που προκλήθηκαν από την επιλογή των προηγούμενων στοιχείων. Τυχαιοποιημένος, χαρακτηρίζεται γιατί επιλέγεται τυχαία ένα στοιχείο από τη λίστα επικρατέστερων υποψηφίων και όχι απαραίτητα το καλύτερο. Η λίστα αυτή ονομάζεται λίστα περιορισμού υποψηφίων - Restricted Candidate List (**RCL**). Αυτή η τεχνική επιλογής ενισχύει την ανάδυση διαφορετικών αρχικών λύσεων σε κάθε επανάληψη του **GRASP**, χωρίς απαραίτητα να αλλοιώνει το Άπληστο Προσαρμοστικό μέρος της μεθόδου.

Ωστόσο, όπως και σε πολλές ντετερμινιστικές μεθόδους, οι λύσεις της πρώτης φάσης του **GRASP** δεν είναι σίγουρο ότι θα βρίσκονται σε τοπικά βέλτιστα. Η δεύτερη φάση, της τοπικής αναζήτησης, επιχειρεί να βελτιώσει τις αρχικές κατασκευαστικές λύσεις. Σε κάθε επανάληψη μια τοπική αναζήτηση εξερευνεί τη γειτονιά της υπάρχουσας λύσης και προσοδευτικά την αντικαθιστά με μία βελτιωμένη. Η επαναληπτική διαδικασία τερματίζεται όταν δεν μπορεί να εντοπιστεί καλύτερη λύση στη γειτονιά της. Αν αυτή είναι καλύτερη από όλες τις προηγούμενες λύσεις του **GRASP**, επιλέγεται ως η ολικά επικρατέστερη [15].

---

**Algorithm 1:** Ο γενικός αλγόριθμος GRASP
 

---

Result:  $s^*$

```

1  $c(s^*) = \infty$ 
2 while (κριτήριο τερματισμού) do
3   Κατασκεύασε αρχική λύση  $s$ 
4    $s \leftarrow$  Τοπική αναζήτηση
5   if  $c(s) < c(s^*)$  then
6      $s^* = s$ 
```

---

#### 4.2.2 Πληθυσμιακοί (population based) αλγόριθμοι εμπνευσμένοι από τη φύση

Τα τελευταία χρόνια υπάρχει η τάση να χρησιμοποιούνται αλγόριθμοι εμπνευσμένοι από τη φύση, χάρις στην ευελιξία και αποτελεσματικότητά τους. Σε αυτή την κατηγορία κατά βάση χρησιμοποιείται ένα σύνολο πρακτόρων που λειτουργούν αξιοποιώντας ένα μηχανισμό συνεργασίας ή ανταγωνισμού. Για παράδειγμα, οι γενετι-



κοί (genetic algorithms) είναι πληθυσμιακοί, αφού χρησιμοποιούν ένα πλήθος ατόμων (individuals). Αντίστοιχα και στη νοημοσύνη σμήνους (swarm intelligence) αλγόριθμοι όπως, η βελτιστοποίηση σμήνους σωματιδίων (PSO), ο αλγόριθμος της πυγολαμπίδας (FA) και του κούκου (CS) χρησιμοποιούν πολλαπλούς πράκτορες ή σωματίδια [25].

Το όνομά τους το οφείλουν στη φυσική διεργασία που προσπαθούν να μιμηθούν. Ο αλγόριθμος βελτιστοποίησης αποικίας μυρμηγκιών (ACO) π.χ. προσπαθεί να προσομοιώσει τη συμπεριφορά πραγματικών μυρμηγκιών κατά τη διαδικασία εύρεσης της τροφής τους. Ο αλγόριθμος βελτιστοποίησης ζευγαρώματος μελισσών (HBMO), προσομοιώνει τη διαδικασία ζευγαρώματος της βασιλίσσας των μελισσών στην κυνέλη.

#### 4.2.2.1 Υπολογιστική, Τεχνητή και Νοημοσύνη Σμήνους

Υπολογιστική νοημοσύνη είναι η μελέτη της σχεδίασης ευφυών πρακτόρων. Πράκτορας είναι οτιδήποτε δρα μέσα σε ένα περιβάλλον. Ως πράκτορας θα μπορούσε να χαρακτηριστεί ένα ζώο, ένας άνθρωπος, ένας θερμοστάτης, ένα αεροπλάνο, μια οργάνωση ή μία κοινωνία. Οι δράσεις του είναι κατάλληλες για τις συνθήκες του περιβάλλοντός του και τον σκοπό του, και είναι ευέλικτος ως προς τις αλλαγές τους. Μαθαίνει από τις εμπειρίες του και κάνει κατάλληλες επιλογές δεδομένης της περιορισμένης αντίληψης και υπολογιστικής του ικανότητας. Ο κύριος λοιπόν στόχος της επιστήμης της υπολογιστικής νοημοσύνης, είναι να κατανοήσει τις αρχές που καθιστούν τη νοήμονα συμπεριφορά δυνατή σε φυσικά ή τεχνητά συστήματα [20].

Η τεχνητή νοημοσύνη, προσπαθεί να συνδυάσει αυτή την ευφυΐα με την υπολογιστική ικανότητα των μηχανών (H/Y), για να σχεδιάσει προγράμματα ικανά να μιμηθούν την πολύπλοκη ανθρώπινη σκέψη. Στόχος, η επίλυση πολυσύνθετων προβλημάτων και υποστήριξη της διαδικασίας λήψης αποφάσεων [6].

Νοημοσύνη σμήνους, είναι η συλλογική συμπεριφορά ενός αποκεντροποιημένου, αυτοοργανωμένου, φυσικού ή τεχνητού συστήματος. Τυπικά ένα τέτοιο σύστημα αποτελείται από ένα πληθυσμό απλών πρακτόρων, που αλληλεπιδρούν τοπικά μεταξύ τους και με το εξωτερικό τους περιβάλλον. Οι πράκτορες ακολουθούν απλούς κανόνες, και παρά την έλλειψη κεντρικής αρχής που να καθορίζει την ατομική τους συμπεριφορά, τοπικές, μερικώς τυχαίες αλληλεπιδράσεις οδηγούν στην ανάδυση “ευφυών” συνολικών συμπεριφορών στο επίπεδο του σμήνους, που παραμένουν άγνωστες στους υποκείμενους πράκτορες [3].

#### 4.2.2.2 Βελτιστοποίηση σμήνους σωματιδίων (PSO)

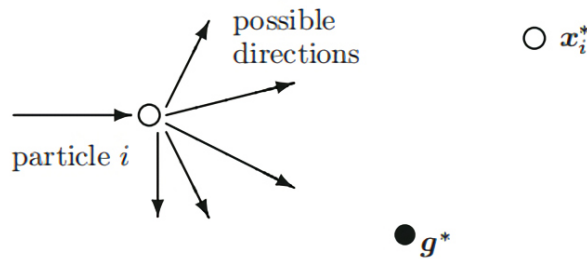
Ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων - Particle Swarm Optimization (PSO) προτάθηκε από τους Kennedy και Eberhart το 1995. Η έμπνευσή του προήλθε από την παρατήρηση της συ-

μπεριφοράς των πουλιών και των ψαριών στην φύση. Από τότε, κατάφερε να προκαλέσει έντονο ενδιαφέρον στους ερευνητές και να σχηματίσει ένα διαρκώς επεκτεινόμενο πεδίο. Μέχρι σήμερα, έχουν αναπτυχθεί τουλάχιστον δύο δεκάδες παραλλαγές του PSO. Υβριδικές παραλλαγές στις οποίες συνδυάζεται με άλλους αλγόριθμους, εξετάζονται συνεχώς. Χάρης την απλή του μορφή και γρήγορη προσέγγιση του βέλτιστου, βρίσκει εφαρμογή σχεδόν σε όλους τους τομείς της συνδυαστικής βελτιστοποίησης, αποδεικνύοντας την αποτελεσματικότητά και αποδοτικότητά του.

Αρκετοί αλγόριθμοι, όπως ο αλγόριθμος των μυρμηγκιών και της πυγολαμπίδας χρησιμοποιούν τη συμπεριφορά της νοημοσύνης σμήνους. Σε αυτή την κατηγορία ανήκει και ο PSO. Αντί όμως να χρησιμοποιεί μεταλλάξεις (γενετικοί) ή φερομόνη (μυρμήγκια), χρησιμοποιεί τυχαιότητα πραγματικών αριθμών και καθολική (global) επικοινωνία μεταξύ των σωματιδίων του σμήνους. Συνεπώς, είναι πιο εύκολη η υλοποίησή του, διότι δεν είναι αναγκαία η κωδικοποίηση και αποκωδικοποίηση των παραμέτρων του προβλήματος σε δυαδική μορφή, όπως στους γενετικούς.

Αρκετοί από τους νεότερους αλγόριθμους, που είναι βασισμένοι στη νοημοσύνη σμήνους, έχουν πάρει έμπνευση από διάφορες πηγές αλλά, σχεδόν όλοι έχουν ομοιότητες με κάποια από τα στοιχεία του PSO. Για το λόγο αυτό, μπορεί να ειπωθεί ότι ο PSO πρωτοπόρησε στη διαμόρφωση της νοημοσύνης σμήνους.

**ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΟΥ** Ο αλγόριθμος του PSO εξερευνεί τον χώρο λύσεων μιας αντικειμενικής συνάρτησης  $f$  προσαρμόζοντας τις τροχιές των πρακτόρων του, τα αποκαλούμενα σωματίδια. Η θέση  $x$  ενός σωματιδίου  $i$ , αντιπροσωπεύει μία πιθανή λύση του προβλήματος, και η απόδοσή της εκτιμάται βάση της προκαθορισμένης  $f(x)$ . Η ταχύτητα  $v$  εκφράζει τις αλλαγές που πρέπει να γίνουν για να μετακινηθεί από τη μία θέση στην άλλη. Η κίνησή του μέσα στο σμήνος βασίζεται σε δύο κύρια δομικά στοιχεία, ένα στοχαστικό και ένα ντετερμινιστικό. Καθώς κινείται προς τη δικά του κατεύθυνση, το σωματίδιο έλκεται από τη θέση του καλύτερου σωματιδίου ολόκληρου του σμήνους  $g^*$  και την καλύτερη θέση  $x^*$  που είχε βρει το ίδιο έως τώρα.



Εικόνα 4.1: Η κίνηση ενός σωματιδίου προς την καλύτερη λύση του σμήνους  $g^*$  και προς την καλύτερη λύση  $x_i^*$  για το σωματίδιο  $i$

Όταν ένα σωματίδιο βρει μία θέση καλύτερη από κάθε θέση που είχε στο παρελθόν, επικρατεί ως η νέα βέλτιστη  $x^*$  του σωματιδίου. Κάθε σωματίδιο θυμάται τη δικιά του καλύτερη θέση. Σκοπός είναι να βρεθεί η καλύτερη θέση  $g^*$  ανάμεσα σε όλα τα σωματίδια του σμήνους, μέχρι το αποτέλεσμα της αντικειμενικής συνάρτησης να σταματήσει να βελτιώνεται ή μέχρι να ολοκληρωθεί ένα σύνολο επαναλήψεων. Έτσι καλλιεργείται ο συνεργατικός παράγοντας, αφού τα μέλη της ομάδας συνεργάζονται μεταξύ τους για την κατασκευή νέων λύσεων. Μία τυπική μορφή του κλασσικού PSO παρουσιάζεται στον ψευδοκώδικα 2.

---

Algorithm 2: Ο γενικός αλγόριθμος του κλασσικού PSO

---

Result: Βέλτιστη  $g^*$  και  $f(g^*)$   
1 Αντικειμενική συνάρτηση  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$   
2 Αρχικοποίησε θέσεις  $x_i$  και ταχύτητες πολυπλοκότητας  $v_i \forall$  σωματίδιο  
3 Βρες  $g^*$  από  $\min\{f(x_1), \dots, f(x_n)\} (t = 0)$   
4 while (κριτήριο τερματισμού) do  
5     forall Σωματίδιο do  
6         Υπολόγισε νέα ταχύτητα  $v_{ij}^{t+1}$  από εξίσωση (4.1)  
7         Υπολόγισε νέα θέση  $x_{ij}^{t+1}$  από εξίσωση (4.2)  
8         Αξιολόγησε θέση  $f(x_{ij}^{t+1})$   
9         Βρες το καινούριο  $x^*$   
10     Βρες το καινούριο  $g^*$   
11     Ανανέωσε  $t = t + 1$  (μετρητής επανάληψης)

---

Πιο αναλυτικά, ο αλγόριθμος ξεκινάει με τον καθορισμό  $n$  σωματιδίων. Στο κάθε σωματίδιο  $i$  δίνεται ένα τυχαίο (ή μη) διάνυσμα θέσης  $x_{ij}$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, d$  ( $d$  ο αριθμός των διαστάσεων - κόμβων) και μια αρχικά μηδενική ταχύτητα  $v_{ij}$ . Στην πρώτη επανάληψη, η καλύτερη θέση που είχε ποτέ το σωματίδιο  $x_{ij}^{*(0)}$ , ταυτίζεται με την τωρινή του θέση  $x_{ij}$ . Έπειτα, υπολογίζεται η ολική καλύτερη θέση  $g^* \approx \min\{f(x_{ij})\}$ , όπου όλα τα σωματίδια αξιολογούνται από την αντικειμενική συνάρτηση  $f(x_{ij})$  και ως  $g^*$  επιλέγεται αυτή με το καλύτερο κόστος. Σε κάθε επανάληψη  $t$  οι εξισώσεις ταχύτητας και θέσης υπολογίζονται ως εξής:

$$v_{ij}^{t+1} = v_{ij}^t + \alpha \epsilon_1 [g^* - x_{ij}^t] + \beta \epsilon_2 [x_{ij}^{*(t)} - x_{ij}^t], \quad (4.1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (4.2)$$

όπου  $\epsilon_1$  και  $\epsilon_2$  είναι δύο τυχαία διανύσματα ομοιόμορφα κατανεμημένα στο διάστημα  $[0, 1]$ . Η παράμετρος  $\alpha$  αναφέρεται ως κοινωνικός όρος (social component), και ποσοτικοποιεί την επιρροή που επιθυμούμε να ασκήσει η  $g^*$  στο σωματίδιο  $i$ . Αντίστοιχα, η  $\beta$  ονομάζεται γνωστικός (cognitive component), και δηλώνει την επιρροή



της μνήμης  $x^*$  του σωματιδίου  $i$  στη νέα του πορεία. Γενικά λειτουργούν σαν μαθησιακοί παράγοντες ή συντελεστές επιτάχυνσης και τυπικά παίρνουν τιμές  $\alpha \approx \beta \approx 2$ .

Υπάρχουν πολλές παραλλαγές του κλασσικού PSO που στοχεύουν στην επέκταση των δυνατοτήτων του. Μια από τις πιο διαδεδομένες, είναι η επιβολή μίας συνάρτησης βάρους αδράνειας  $\theta(t)$  στην ταχύτητα του σωματιδίου, με τη συνάρτηση ταχύτητας να μετατρέπεται σε:

$$v_{ij}^{t+1} = \theta v_{ij}^t + \alpha \epsilon_1 [g^* - x_{ij}^t] + \beta \epsilon_2 [x_{ij}^{*(t)} - x_{ij}^t] \quad (4.3)$$

όπου  $\theta$  παίρνει τιμές από 0 έως 1. Στην πιο απλή της μορφή η αδράνεια παίρνει μια σταθερή τιμή, τυπικά από 0.5 έως 0.9. Με τη μέθοδο αυτή, ουσιαστικά εισάγεται μια εικονική μάζα για να σταθεροποιήσει την κίνηση των σωματιδίων και συνεπώς αναμένεται πως ο αλγόριθμος θα συγκλίνει γρηγορότερα στη βέλτιστη λύση.



Μέρος II

ΠΡΑΚΤΙΚΗ ΕΠΙΛΥΣΗ



## 5.1 ΣΧΗΜΑΤΙΣΜΟΣ ΑΡΧΙΚΩΝ ΛΥΣΕΩΝ

Προκειμένου να λειτουργήσει ο κλασικός αλγόριθμος του PSO, χρειάζεται ολοκληρωμένες λύσεις ίδιου μήκους για κάθε σωματίδιό του. Συνήθως οι αρχικές λύσεις αρχικοποιούνται με τυχαίο τρόπο. Στο συγκεκριμένο πρόβλημα όμως, περιορισμοί όπως η ετερογένεια και τα χρονικά παράθυρα, μπορούν να οδηγήσουν σε κακές ή ακόμα και μη-εφικτές αρχικές λύσεις. Είναι λοιπόν αναγκαίο να βρεθεί ένας πιο συστηματικός μηχανισμός για τη σχεδίαση των αρχικών λύσεων. Για τον λόγο αυτό, σχεδιάστηκε μια νέα παραλλαγή του [GRASP](#), προσαρμοσμένη στις απαιτήσεις και ιδιομορφίες του προβλήματος. Συνδυάζει *μεταβλητές* και *βάρη απόφασης* για τον σχηματισμό αρχικών λύσεων και στη συνέχεια θα αναφέρεται ως Weight Enhanced GRASP ([WEGRASP](#))

*Ανάλυση του WEGRASP*

Σε κάθε επανάληψη της κατασκευαστικής φάσης του αλγορίθμου, επιλέγεται τυχαία ένα πλοίο με σκοπό να του ανατεθεί ένα φορτίο προς εξυπηρέτηση. Με το πέρας της κατασκευαστικής φάσης, έχει δημιουργηθεί μία αρχική λύση που αντιπροσωπεύει την αρχική θέση ενός σωματιδίου. Ο [WEGRASP](#) καλείται να δώσει μία αρχική εφικτή λύση για κάθε σωματίδιο. Η δεύτερή του φάση, αυτή της τοπικής αναζήτησης, εκτελείται σαν υποδιαδικασία του PSO και παρουσιάζεται αναλυτικότερα στη συνέχεια.

Το κρίσιμότερο στάδιο στη σχεδίαση ενός [GRASP](#), είναι η κατάλληλη διαμόρφωση της [RCL](#) και ο τρόπος επιλογής του επικρατέστερου από αυτή. Στην παραλλαγή του [WEGRASP](#), γίνεται χρήση τεσσάρων μεταβλητών απόφασης που φέρουν κάποια γνώση για την παρούσα κατάσταση του συστήματος, και με κατάλληλο συνδυασμό διαμορφώνεται η [RCL](#).

Συνήθως, σε κάθε αλγόριθμο απληστίας που εφαρμόζεται σε πρόβλημα με χρονικά παράθυρα, ο κόμβος που θα επιλεγεί για να συμπληρώσει τη μερική λύση, θα είναι ο κοντινότερος σύμφωνα με τον απαιτούμενο χρόνο επίσκεψής του. Έτσι και εδώ, μια από τις μεταβλητές που θα καθορίσουν ποιο φορτίο θα εξυπηρετηθεί στη συνέχεια από το επιλεγθέν πλοίο, θα πρέπει να συνυπολογίζει τον χρόνο που χρειάζεται το πλοίο για να φτάσει στο λιμάνι φόρτωσης. Αν το πλοίο φτάσει νωρίτερα από τον ενωρίτερο χρόνο επιτρεπτής φόρτωσης  $a_i$ , είναι αναγκασμένο να περιμένει. Όσο περιμένει, το πλοίο θεωρείται αδρανές. Η αδράνεια είναι ανεπιθύμητη, καθώς ο χρόνος ενός πλοίου είναι παραγωγικός μόνο όταν εκτελεί κάποια

*Μεταβλητή  
χρόνου*

διεργασία (μετακίνηση - φόρτωση - εκφόρτωση). Ο χρόνος φόρτωσης ενός εμπορεύματος από κάποιο πλοίο, είναι διαφορετικός για κάθε εμπόρευμα και διαφορετικός για κάθε πλοίο, εξαιτίας της ετερογένειας. Ο χρόνος εκφόρτωσης του εμπορεύματος, είναι σχεδόν ίδιος με τον χρόνο φόρτωσής του, με αμελητέα διαφορά. Η εξίσωση που συνυπολογίζει τις παραπάνω παραμέτρους σχηματίζεται ως εξής:

$$TT = \max(t_0 + T_{ijv}, a) + LTC_i^v \quad (5.1)$$

με  $t_0$  η χρονική στιγμή που ξεκινάει το δρομολόγιο,  $T_{ijv}$  η διάρκεια του δρομολογίου και  $LTC_i^v$  ο χρόνος φόρτωσης του εμπορεύματος  $i$  από το πλοίο  $v$ . Συνολικά υπολογίζει τη χρονική στιγμή στην οποία θα έχει ολοκληρωθεί η φόρτωση του εμπορεύματος  $i$  από το πλοίο  $v$ . Η εξίσωση εφαρμόζεται για κάθε εφικτή εναλλακτική επιλογή, εξαιρουμένων δηλαδή των εμπορευμάτων που έχουν ήδη εξυπηρετηθεί, ή δεν μπορούν να εξυπηρετηθούν από αυτό το πλοίο, επειδή παραβιάζουν περιορισμούς παραθύρων ή συμβατότητας. Τα αποτελέσματα των υποψήφιων εμπορευμάτων αποθηκεύονται σε μία λίστα, ταξινομούνται με αύξουσα σειρά και βαθμολογούνται ανάλογα με την κατάταξή τους στη λίστα. Δηλαδή, στη λίστα πρώτο είναι το εμπόρευμα  $i$  που έχει τον μικρότερο χρόνο  $TT$  με βαθμολογία 1 και τελευταίο το εμπόρευμα με τον μεγαλύτερο χρόνο και βαθμολογία ίση με το πλήθος των εναλλακτικών. Με τον συνδυασμό αυτών των παραμέτρων, αφενός τιμωρείται η αδράνεια και αφετέρου επιβραβεύεται η επιλογή εμπορεύματος με μικρότερο χρόνο εξυπηρέτησης.

Μεταβλητή  
κόστους

Κάθε δρομολόγιο  $(i, j) \in A_v$  σχετίζεται με ένα κόστος  $C_{ijv}$ . Το κόστος ενός δρομολογίου  $i, j$  διαφέρει από πλοίο σε πλοίο, αλλά για το  $v \in V$  το κόστος οποιουδήποτε δρομολογίου είναι ανάλογο με τη χρονική διάρκεια του. Όμοια με το χρόνο εξυπηρέτησης, το κόστος εξυπηρέτησης  $LC_i^v$  διαφέρει για κάθε εμπόρευμα και για κάθε πλοίο. Η εξίσωση κόστους σχηματίζεται ως:

$$TC = C_{ijv} + LC_i^v, \quad (5.2)$$

Η λίστα υπολογίζεται ομοίως και ταξινομείται με αύξουσα σειρά. Συνεπώς, η χρήση του  $LC_i^v$  αναδεικνύει τα εμπορεύματα που συμφέρει περισσότερο να φορτωθούν από το πλοίο  $v$ . Το κόστος δρομολογίου, αντιθέτως, λειτουργεί σαν ισοσταθμιστικός παράγοντας μεταξύ του κόστους  $C_{ijv}$  και της διάρκειας του δρομολογίου  $T_{ijv}$  (μεταβλητές ανάλογες μεταξύ τους), καθιστώντας την απόφαση λιγότερο μυωπική.

Μεταβλητή  
ποινή

Κάθε εμπόρευμα συνδέεται με μια ποινή μη εξυπηρέτησης  $C_i^S$  διαφορετικού μεγέθους, που πληρώνεται αν το εμπόρευμα  $i$  δε μεταφερθεί από τον στόλο. Μια λίστα σχηματίζεται με τις ποινές των υποψήφιων εμπορευμάτων και ταξινομείται με φθίνουσα σειρά. Συμπεριλαμβάνοντας το  $C_i^S$  στη διαδικασία απόφασης, το σύστημα

έχει την ικανότητα να δώσει προτεραιότητα στα εμπορεύματα με υψηλότερη ποινή και άρα να μειώσει δυνητικά το τελικό κόστος της αντικειμενικής συνάρτησης.

Συνήθως, κάθε εμπόρευμα έχει ένα πλήθος εναλλακτικών πλοίων που μπορούν να το φορτώσουν. Ανάλογα όμως το μέγεθος του στόλου και το πλήθος των εμπορευμάτων, είναι πιθανό να παρουσιαστούν εμπορεύματα που να μπορούν να φορτωθούν μόνο από ένα πλοίο, εξαιτίας του περιορισμού συμβατότητας. Τα εμπορεύματα αυτά τα ονομάζουμε “μοναδικά”.

Μία δυαδική μεταβλητή  $u$  καθορίζει με τιμή 1 αν το εμπόρευμα  $i$  μπορεί να φορτωθεί από το πλοίο  $v$  ή όχι με τιμή 0. Συνολικά η λίστα  $U$  αναδεικνύει ποια από τα υποψήφια εμπορεύματα είναι μοναδικά για το πλοίο  $v$ . Η χρήση της μεταβλητής μοναδικότητας είναι διπλή.

Πρώτον, στον σχηματισμό διαδρομής με περιορισμό χρονικών παραθύρων, η επιλογή του επόμενου στοιχείου εξαρτάται από τη χρονική στιγμή  $t_0$  στην οποία βρίσκεται τώρα το πλοίο. Η  $t_0$  διαμορφώθηκε από τις παρελθούσες επιλογές εμπορευμάτων. Συνεπώς, κάθε επιλογή που γίνεται επηρεάζει τις μελλοντικές εναλλακτικές. Το μοναδικό εμπόρευμα έχει δύο εναλλακτικές. Είτε θα εξυπηρετηθεί από αυτό το πλοίο, είτε θα ναυλωθεί. Εκμεταλλευόμενοι αυτή τη γνώση, μπορούμε να του δώσουμε προτεραιότητα. Έτσι, θα επηρεάσει την υπόλοιπη διαδρομή και θα μειώσει το πιθανό υπολογιστικό κόστος διόρθωσής της σε επόμενα στάδια του αλγορίθμου. Δεύτερον, αν το μοναδικό εμπόρευμα δεν εξυπηρετηθεί από το πλοίο του, η ναύλωση του γίνεται αναπόφευκτη και άρα αυτόματα αυξάνεται το κόστος της αντικειμενικής συνάρτησης.

Από τις παραπάνω μεταβλητές, αποδεικνύεται πειραματικά πως δε συμβάλλουν όλες με την ίδια βαρύτητα στην ελαχιστοποίηση της αντικειμενικής συνάρτησης. Ένας συντελεστής σημαντικότητας  $w_i$  εισάγεται για κάθε μεταβλητή. Συνδυαστικά, λειτουργούν για να διορθώσουν την ισορροπία των μεταβλητών. Οι συντελεστές  $w_1, w_2, w_3, w_4$  αντιστοιχούν στη βαρύτητα που αποδίδεται στο χρόνο, κόστος, ποινή και μοναδικότητα, αντίστοιχα. Εμπειρικά, οι σχέσεις τους ορίστηκαν ως:  $w_1 \approx w_4 \approx 2$  και  $w_2 \approx w_3 \approx 1$ .

Οι μεταβλητές χρόνου και μοναδικότητας χαρακτηρίζονται ως υψηλότερης βαρύτητας γιατί, συμβάλλουν στο σχηματισμό διαδρομών με περισσότερα εμπορεύματα, μειώνοντας ταυτόχρονα τα εμπορεύματα για ναύλωση. Στις μεταβλητές κόστους και ποινής δίνεται μικρότερη βαρύτητα, ώστε να μην απορρυθμίζουν τις πιο σημαντικές μεταβλητές, αλλά αρκετή ώστε να μπορούν να εντείνουν τις διαφορές μεταξύ των υποψηφίων, για να επιλεγεί ευκολότερα ο πραγματικά καλύτερος.

Σε αυτό το στάδιο, η λίστα κάθε μεταβλητής έχει υπολογιστεί, οι εναλλακτικές σε κάθε λίστα έχουν ταξινομηθεί και βαθμολογηθεί (βαθμολογία = θέση στη λίστα), και το βάρος κάθε μεταβλητής έχει πολλαπλασιαστεί με τη λίστα της. Η τελική βαθμολογία κάθε εναλλακτικής, υπολογίζεται ως το άθροισμα των βαθμολογιών της, στις λίστες. Το αποτέλεσμα είναι η **RCL** λίστα με τη συνολική βαθ-

*Μεταβλητή  
μοναδικότητας*

*Βαρύτητα  
μεταβλητών*

*Σύνθεση  
μεταβλητών*

μολογία κάθε εναλλακτικής. Ένα διάνυσμα “ $\epsilon$ ” με μήκος ίσο με το πλήθος των εναλλακτικών και τυχαίες τιμές στο εύρος  $[0, 1]$  εφαρμόζεται στην τελική λίστα, για να αποφευχθούν ισοπαλίες και να ενισχυθεί η τυχαιότητα. Από τη λίστα [RCL](#) επιλέγεται το εμπόρευμα με τη μικρότερη βαθμολογία. Η τελική εξίσωση που σχηματίζει τη [RCL](#) αποτυπώνεται ως:

$$RCL = (w_1 TT + w_2 TC + w_3 P - w_4 U) + \epsilon \quad (5.3)$$

Για παράδειγμα, το πλοίο μπορεί να φορτώσει το εμπόρευμα  $i = 3$ . Μετά την εφαρμογή των συντελεστών βάρους, το εμπόρευμα 3 στη λίστα χρόνου μπορεί να βρίσκεται 3ο, στη λίστα κόστους 1ο και ποινης 6ο. Η συνολική βαθμολογία του υπολογίζεται ως 10. Αν το εμπόρευμα είναι “μοναδικό” και μπορεί να φορτωθεί μόνο από αυτό το πλοίο, αφαιρείται η μεταβλητή μοναδικότητας από τη βαθμολογία του. Τέλος, μια ομοιόμορφα κατανεμημένη τιμή  $\epsilon_i$  προστίθεται στη βαθμολογία. Μικρότερη βαθμολογία ισούται με πιο επιθυμητή εναλλακτική.

Συνθήκη  
τερματισμού

Ένα πλοίο μπορεί να φορτώσει κάποιο συμβατό εμπόρευμα  $i$ , όσο η βραδύτερη χρονική στιγμή εξυπηρέτησης  $b_i$  του εμπορεύματος, είναι μεγαλύτερη από τη χρονική στιγμή  $TT_i^v$  στην οποία το  $v$  θα έχει ολοκληρώσει την εξυπηρέτησή του  $i$ . Αν δεν υπάρχει μη-κατανεμημένο εμπόρευμα που μπορεί να φορτώσει, το πλοίο σταματάει να συμμετέχει στη διαδικασία επιλογής. Όταν η λίστα με τα πλοία που μπορούν να φορτώσουν εμπορεύματα αδειάσει ή όλα τα εμπορεύματα κατανεμηθούν στον στόλο, ο αλγόριθμος τερματίζεται.

## 5.2 ΤΟΠΙΚΗ ΑΝΑΖΗΤΗΣΗ

Ένα σύνολο τριών διαφορετικών τροποποιημένων μεθοδολογιών τοπικής αναζήτησης, αναπτύχθηκε για το πρόβλημα του [ITSRSP](#). Οι [LS](#) εφαρμόζονται σε ολοκληρωμένες εφικτές διαδρομές  $\hat{x}_i$ , και παίζουν καταλυτικό ρόλο στην διαδικασία της εντατικοποίησης. Σε κάθε επανάληψη μιας μεθόδου, μια μικρή διαταραχή προκαλείται στο σύστημα. Αν το αποτέλεσμα της είναι η μείωση του κόστους της αντικειμενικής συνάρτησης, η διαταραχή αποδέχεται ως η νέα βελτιωμένη λύση. Διαφορετικά απορρίπτεται και το σύστημα επαναφέρεται στην προηγούμενη σταθερή του κατάσταση.

### 5.2.1 1-0 Relocate

Η κλασσική μέθοδος Relocate είναι η πιο απλή και βασική μέθοδος επανατοποθέτησης ενός κόμβου. Τυχαία επιλέγεται ένα μη-κενό πλοίο και από αυτό, ένα εμπόρευμα αφαιρείται από την τρέχουσα θέση του και εισάγεται σε διαφορετική θέση, σε διαφορετικό συμβατό πλοίο.



Η κίνηση της επανατοποθέτησης  $m^{reloc}$  μπορεί να αποσυντεθεί σε δύο μερικές κινήσεις [9]. Έστω  $k$  το προς μετακίνηση εμπόρευμα,  $i$  το εμπόρευμα που προηγείται και  $j$  αυτό που έπεται. Η διαδικασία ξεκινάει με την απελευθέρωση  $p^{free}$  του εμπορεύματος  $k$  από το πλοίο  $s$ . Αποκόπτονται οι ακμές  $(D_i, P_k)$  και  $(D_k, P_j)$  που συνδέουν το  $k$  σε αυτή τη διαδρομή, και εισέρχεται η ακμή  $(D_i, P_j)$  για να επιδιορθωθεί η διαδρομή. Στη συνέχεια εφαρμόζεται η διαδικασία εισαγωγής  $p^{ins}$  σε πλοίο  $r$  συμβατό με το  $k$  ( $I_{kr} = 1$ ). Τώρα αποκόπτονται οι ακμές  $(D_i, P_j)$  που συνδέουν τα νέα εμπορεύματα  $i$  &  $j$ , ανάμεσα στα οποία θα εισαχθεί το  $k$ , και επανασυνδέονται με τις ακμές  $(D_i, P_k)$  και  $(D_k, P_j)$ . Η ακμή  $(P_k, D_k)$  παραμένει άθικτη. Οι εξοικονομήσεις των μερικών κινήσεων υπολογίζονται από τις ακόλουθες εξισώσεις:

$$g(p^{free}) = C_{D_i P_k s} + C_{D_k P_j s} + S_{P_k s}^c + S_{D_k s}^c - C_{D_i P_j s} \quad (5.4)$$

$$g(p^{ins}) = C_{D_i P_j r} - C_{D_i P_k r} - C_{D_k P_j r} - S_{P_k r}^c - S_{D_k r}^c \quad (5.5)$$

με  $C$  το κόστος δρομολογίου και  $S$  το κόστος φόρτωσης - εκφόρτωσης. Αν η συνολική εξοικονόμηση  $g(m^{reloc}) = g(p^{free}) + g(p^{ins}) > 0$ , η νέα λύση αντικαθιστά την υπάρχουσα, αλλιώς απορρίπτεται και η διαδικασία ξεκινάει από την αρχή.

### 5.2.2 1-1 Intraexchange

Επειδή ο στόχος του [ITSRSP](#) είναι διπλός, δύο διαφορετικοί πίνακες χρησιμοποιούνται που αντικατοπτρίζουν την κάθε πτυχή του προβλήματος. Ένας πίνακας αναπαριστά τις διαδρομές  $\hat{x}_i$ , τα εμπορεύματα δηλαδή που θα εξυπηρετηθούν από το κάθε πλοίο ξεχωριστά, και ο άλλος  $CC$  περιέχει τα μη εξυπηρετημένα από τον στόλο εμπορεύματα.

Το πρόθεμα “Intra” υποδεικνύει ότι η εμβέλεια της τοπικής αναζήτησης είναι “ενδο-διαδρομική”. Δηλαδή, η μέθοδος εφαρμόζεται εσωτερικά, μόνο στον πρώτο πίνακα. Τυχαία επιλέγονται δύο μη-κενά πλοία  $p$ . Αν για καθένα από αυτά τα πλοία υπάρχει τουλάχιστον ένα φορτωμένο εμπόρευμα που να μπορεί να εξυπηρετηθεί από το άλλο πλοίο ( $\forall v \in p \exists I_{x_v v} \cap I_{x_v(p \ominus v)}$ ), επιλέγεται ένα εμπόρευμα από κάθε διαδρομή και εφαρμόζεται η μερική κίνηση απελευθέρωσης  $p^{free}$  και εισάγεται με  $p^{ins}$  στο αντίθετο πλοίο. Ομοίως, η συνολική εξοικονόμηση  $g(m^{intra}) = g(p_i^{free}) + g(p_i^{ins}) + g(p_j^{free}) + g(p_j^{ins})$  θα καθορίσει αν η διαταραχή θα επικρατήσει ή αν θα απορριφθεί.

### 5.2.3 1-Y Interexchange

Μέχρι στιγμής, υπάρχει ένας μηχανισμός  $m^{reloc}$  που μετακινεί εμπορεύματα και δημιουργεί κατάλληλο διαθέσιμο χρονικό περιθώριο για την πιθανή εξυπηρέτηση εμπορεύματος ή εμπορευμάτων εκτός της εφικτής λύσης. Αυτή τη λειτουργία αναλαμβάνει η [LS](#) της μεθόδου Interexchange. Εδώ το πρόθεμα “Inter” δηλώνει ότι η

τοπική αναζήτηση διευρύνεται, ώστε να συμπεριλαμβάνει την λίστα με τα μη εξυπηρετημένα εμπορεύματα. Η μεταβλητή  $Y$  παίρνει τιμές 0 ή 1, και τη μετατρέπει σε λειτουργία δύο ταχυτήτων.

Αρχικά, με  $Y = 0$ , η μέθοδος επιλέγει ένα τυχαίο εμπόρευμα  $k$  από τη δεύτερη “εξωτερική” λίστα και το εισαγάγει  $p^{ins}$  στη διαδρομή ενός συμβατού πλοίου  $v$ . Αν, μετά την εισαγωγή, η διαδρομή παραμένει εφικτή, αποδέχεται ως η νέα βελτιωμένη  $\hat{x}_v$  καθώς το κόστος ναύλωσης  $C_i^S$  είναι πάντα πολύ μεγαλύτερο από το κόστος εισαγωγής  $g(p^{ins})$ . Διαφορετικά, η λειτουργία αλλάζει σε  $Y = 1$ , και επιχειρείται ανταλλαγή (swap) του  $k$  με ένα εμπόρευμα  $i$  στη  $\hat{x}_v$ . Αν μετά την ανταλλαγή το συνολικό κόστος της αντικειμενικής συνάρτησης μειωθεί, γιατί το εξαγόμενο εμπόρευμα είχε μικρότερο κόστος ναύλωσης από το εισαγμένο, η αλλαγή επικρατεί.

#### 5.2.4 Επιλογή κατάλληλης γειτονιάς

Για κάθε υποψήφια κίνηση εισαγωγής  $p^{ins}$  ενός εμπορεύματος  $i \in \{1, \dots, n\}$  σε πλοίο με  $n$  φορτωμένα εμπορεύματα, υπάρχουν  $n + 1$  εναλλακτικές θέσεις εισαγωγής. Τρία είναι τα πιθανά αποτελέσματα ανάλογα με τη θέση που επιλέχθηκε. Πρώτον, η θέση που επιλέχθηκε ήταν η κατάλληλη και το εμπόρευμα κατάφερε να εισαχθεί και η διαδρομή να παραμένει εφικτή. Δεύτερον, το εμπόρευμα εισήχθη σε θέση όπου ο χρόνος εξυπηρέτησης του  $TT_i$  είναι μεγαλύτερος από τον βραδύτερο επιτρεπτό χρόνο φόρτωσής του  $b_i$  και άρα το εμπόρευμα δεν μπορεί να εξυπηρετηθεί. Τρίτον, το  $i$  εισήχθη σε θέση όπου ο ενωρίτερος χρόνος φόρτωσης  $a_i$  είναι μεγαλύτερος από αυτόν του επόμενου εμπορεύματος  $a_{i+1}$ , με αποτέλεσμα να αυξάνεται η αδράνεια του πλοίου, όλες οι εξυπηρετήσεις να καθυστερούν και άρα η επισκευή της διαδρομής να καθίσταται αδύνατη.

Από την παραπάνω συμπεριφορά του συστήματος, και έπειτα από πειραματική εξακρίβωση, γίνεται η παραδοχή ότι, η σωστή θέση εισαγωγής  $p^{ins}$  ενός εμπορεύματος είναι εκείνη στην οποία ισχύει  $a_{i-1} < a_i < a_{i+1}$ . Με αυτό τον τρόπο, οι  $n + 1$  πιθανές εναλλακτικές ελαττώθηκαν σε 1 βέβαιη θέση. Συνεπώς, οι άσκοποι έλεγχοι περιορίζονται, και οι υπολογισμοί των [LS](#) επιταχύνονται σημαντικά.

#### 5.2.5 Μηχανισμός Διχοτόμησης

Όπως εδραιώθηκε προηγουμένως, κάθε εμπόρευμα που επιλέγεται να εισαχθεί σε μια διαδρομή επηρεάζει τις επόμενες επιλογές. Αυτή η ιδιομορφία έκανε εμφανή τα δύο ακόλουθα προβλήματα. Πρώτον, η εισαγωγή  $p^{ins}$  ενός κόμβου σε μια διαδρομή μπορεί να επηρεάσει έναν μεταγενέστερο κόμβο και η διαδρομή να μετατραπεί σε μη εφικτή, με αποτέλεσμα μια πιθανά καλή κίνηση να απορρίπτεται. Δεύτερον, παρατηρήθηκε πως κάποια εμπορεύματα μπορούν να φορτωθούν μόνο με μια συγκεκριμένη αλληλουχία, α-

νεξαρτήτως πλοίου. Για παράδειγμα, το εμπόρευμα  $i = 3$  μπορεί να φορτωθεί μόνο μετά το εμπόρευμα  $i = 7$ . Άρα και η μετακίνηση  $m$  τους πρέπει να γίνει σε “πακέτα”. Το πρόβλημα που προκύπτει τώρα, είναι πως δεν είναι γνωστό αφενός αν χρειάζεται η μετακίνηση σε πακέτα και αφετέρου το “μέγεθος” των πακέτων (πόσα δηλαδή εμπορεύματα πρέπει να μετακινηθούν σε μία κίνηση  $m$ ). Ένας απλός αλλά συνάμα δυναμικός μηχανισμός διχοτόμησης της διαδρομής σχεδιάστηκε για την αντιμετώπιση αυτού του φαινομένου.

Αρχικά, γίνεται η παραδοχή ότι, η επισκευή μιας διαδρομής μετά από μία κίνηση  $p^{ins}$  θα είναι ανέφικτη. Γιαυτό, η κίνηση της εισαγωγής τροποποιείται ώστε να μην επιχειρεί την επισκευή της ακμής  $(D_k, P_j)$  με το ακόλουθο εμπόρευμα. Αντιθέτως, όλα τα εμπορεύματα, μετά από αυτό που εισήχθη, αφαιρούνται από τη διαδρομή και προστίθενται στην εξωτερική λίστα. Σε αυτό το στάδιο, ένας τροποποιημένος **WEGRASP** αντλεί εμπορεύματα από την εξωτερική λίστα και επιδιορθώνει τη μερική λύση με στοχευμένο τρόπο, εξοικονομώντας έτσι υπολογιστικό χρόνο (που διαφορετικά θα επιβάρυνε τον αλγόριθμο, αν γινόταν χρήση του 1-Y Interexchange για επιδιόρθωση).

Προνόμια που μπορούν να αποδοθούν στη μέθοδο είναι και τα ακόλουθα. Πρώτον, λειτουργεί και σαν μέθοδος εισαγωγής, αφού μπορεί να προσθέσει στη διαδρομή εμπορεύματα που προηγουμένως άνηκαν στην εξωτερική λίστα. Δεύτερον, με την δραστική της συμπεριφορά, έχει τη δυνατότητα να αποφεύγει τοπικά ελάχιστα, χωρίς να απαιτείται ο συνδυασμός περίπλοκων αλγορίθμων ή μεγαλύτερος αριθμός επαναλήψεων του ACPSO, εξοικονομώντας μακροπρόθεσμα ακόμα περισσότερο υπολογιστικό χρόνο.

### 5.3 Ο ΑΛΓΟΡΙΘΜΟΣ ACPSO

Στο κεφάλαιο 4 αναλύθηκε η λειτουργία του κλασσικού PSO μαζί με τα χαρακτηριστικά που τον ξεχώρισαν για την επιλογή του ως το βασικό αλγόριθμο βελτιστοποίησης, για το πρόβλημα του **ITSRSP**. Η εφαρμογή του όπως παρουσιάστηκε όμως, απαιτεί προβλήματα που μοντελοποιούνται με συνεχείς μεταβλητές. Σε αυτό το κεφάλαιο αρχικά, παρατίθεται η συνοπτική μορφή του αλγορίθμου ως σημείο αναφοράς, και έπειτα αναλύονται λεπτομερώς οι τροποποιήσεις που ήταν απαραίτητες για να μετατρέψουν τον κλασσικό PSO στον Adaptive Constriction PSO (**ACPSO**), που είναι πλέον ικανός να αντιμετωπίσει διακριτά προβλήματα δρομολόγησης και χρονικού προγραμματισμού.

#### 5.3.1 Δομή αλγορίθμου

Ο αλγόριθμος ξεκινάει με τον ορισμό  $\phi$  σωματιδίων που θα αξιοποιηθούν για να γίνει η αναζήτηση της λύσης, και όλα μαζί τα σωματίδια αποτελούν ένα σμήνος. Κάθε σωματίδιο  $i$  χρειάζεται μια

*Αρχικοποίηση*

## Algorithm 3: Ο αλγόριθμος ACP SO

---

Result: Βέλτιστη  $g^*$  και  $f(g^*)$

```

1 Αρχικοποίησε παραμέτρους και σμίνη
2 forall Σμίνος do
3   Αρχικοποίησε σωματίδια
4   Βρες  $g^*$  από  $\min\{f(x_{1j}), \dots, f(x_{nj})\}$ 
5   Κωδικοποίησε θέσεις σε συνεχή μορφή με εξίσωση (5.6)
6   while επανάληψη χωρίς εναλλαγή του κυρίαρχου
       σωματιδίου  $\leq It_{msa}$  do
7     forall Σωματίδιο do
8       Υπολόγισε νέα ταχύτητα  $v_{ij}^{t+1}$  με εξίσωση (5.9)
9       Υπολόγισε νέα θέση  $x_{ij}^{t+1}$  με εξίσωση (4.2)
10       $x_{ij}^{t+1} \leftarrow$  Αποκωδικοποίησε σε διακριτή μορφή
11       $x_{ij}^{t+1} \leftarrow$  Επιδιόρθωσε
12       $x_{ij}^{t+1} \leftarrow$  Τοπικές Αναζητήσεις
13      Αξιολόγησε θέση  $f(x_{ij}^{t+1})$ 
14       $x_{ij}^{t+1} \leftarrow$  Κωδικοποίησε σε συνεχή μορφή
15      if  $f(x_{ij}^{t+1}) < f(x_{ij}^*)$  then
16        | Ανανέωσε  $x_{ij}^*$  και  $f(x_{ij}^*)$   $\triangleright$  σωματίδιο βελτιώθηκε
17      else if επανάληψη χωρίς βελτίωση του σωματιδίου
            $\geq It_{mpa}$  then
18        | Διέγραψε σωματίδιο
19        | Κατασκεύασε νέο σωματίδιο
20      else
21        | Κάνε επανέναρξη θέσης και ταχύτητας
22    Βρες το κυρίαρχο σωματίδιο του σμίνους  $g^*$ 
23  while επανάληψη χωρίς βελτίωση του κυρίαρχου
       σωματιδίου  $\leq It_{mka}$  do
24     $g^* \leftarrow$  Τοπικές Αναζητήσεις
25  Αποθήκευσε  $g^*$  και  $f(g^*)$  σμίνους

```

---

θέση  $x_{ij}$  και μια ταχύτητα  $v_{ij}$ , με  $j : \{1, \dots, d\}$  και  $d$  το πλήθος των διαθέσιμων εμπορευμάτων. Η ταχύτητα αρχικοποιείται ως ένα μηδενικό διάνυσμα  $v_{ij} = 0_{1,d}$ . Για την αρχικοποίηση της θέσης  $x_{ij}$  εφαρμόζεται ο αλγόριθμος του [WEGRASP](#). Ο [WEGRASP](#) παράγει εφικτές διαδρομές  $\hat{x}_v$  για κάθε πλοίο. Οι  $\hat{x}_v$  ενώνονται μεταξύ τους και σχηματίζουν μια πλήρη λύση, τη γνωστή και ως Giant Route (GR). Οι  $GR_i$  μεταξύ τους όμως, έκτος από την αλληλουχία, μπορεί να διαφέρουν και στο πλήθος των εμπορευμάτων που περιέχουν (ανάλογα με το μέγεθος της λίστας ναύλωσης -  $CC_i$ ). Εδώ παρουσιάζεται και η πρώτη ιδιομορφία του PSO. Προκειμένου να εφαρμοστούν οι εξισώσεις του, όλα τα σωματίδια πρέπει να έχουν τις ίδιες διαστάσεις. Για να αντιμετωπιστεί το πρόβλημα αυτό, η θέση κάθε σωματιδίου πλέον διορθώνεται ως  $x_{ij} = GR_i \cup CC_i$ . Έ-

τσι, η αλληλουχία των εμπορευμάτων διατηρείται για να μπορεί στη συνέχεια να διαιρεθεί σε ξεχωριστές διαδρομές πλοίων, και ταυτόχρονα διασφαλίζεται ότι οι διαστάσεις όλων των σωματιδίων θα είναι  $1 \times d$ . Η διαδικασία αρχικοποίησης συνεχίζει, όπως περιγράφεται και στον κλασσικό PSO, με τον υπολογισμό της αντικειμενικής συνάρτησης  $f(x_{ij}) \forall i$  και της βέλτιστης προσωπικής θέσης  $x_{ij}^{*(0)} = x_{ij}$ . Έπειτα, η καλύτερη λύση - θέση - όλου του σμήνους  $g^*$  αναδεικνύεται ως η  $\min f(x_{ij}^{*(0)})$ . Το σωματίδιο που έχει την καλύτερη θέση  $g^*$ , θεωρείται ως το κυρίαρχο σωματίδιο του σμήνους.

Όπως εδραιώθηκε νωρίτερα, η λειτουργία του κλασσικού PSO απαιτεί συνεχείς τιμές. Οι θέσεις όμως των σωματιδίων, αναπαρίστανται σε διακριτή μορφή. Μια μεθοδολογία είναι πλέον απαραίτητη για να κωδικοποιήσει τις διακριτές  $x_{ij}$  σε συνεχείς [12], και η εξίσωση που χρησιμοποιείται είναι η ακόλουθη:

$$x_{ij} = \frac{x_{ij}}{d}, \quad \forall j \in \{1, \dots, d\} \quad (5.6)$$

Σε αυτό το στάδιο, ξεκινάει η κύρια επαναληπτική διαδικασία του PSO. Ένας εσωτερικός βρόχος υπολογίζει, για κάθε σωματίδιο, τη νέα ταχύτητα  $v_{ij}^{t+1}$  και θέση  $x_{ij}^{t+1}$  αντλώντας και συνθέτοντας πληροφορία από τη μνήμη του και τη μνήμη καλύτερης θέσης ολόκληρου του σμήνους. Η θέση που παρήχθη, πρέπει να αποκωδικοποιηθεί τώρα από συνεχής σε διακριτή μορφή, για να αξιολογηθεί η ποιότητά της. Αυτό επιτυγχάνεται με βαθμολόγηση του κάθε στοιχείου  $j$  στο  $x_{ij}^{t+1}$  ανάλογα με το μέγεθος της τιμής του. Η μικρότερη τιμή, για παράδειγμα, θα αντιστοιχιστεί στο εμπόρευμα 1, ενώ η μεγαλύτερη τιμή στο εμπόρευμα  $d$ .

Τώρα, η νέα θέση  $x_{ij}^{t+1}$  αντιπροσωπεύει την προτεινόμενη αλληλουχία εξυπηρέτησης των εμπορευμάτων. Ο PSO δε διαθέτει κάποιο μηχανισμό ελέγχου των περιορισμών του [ITSRSP](#) και άρα οι λύσεις που προτείνει θα είναι σχεδόν πάντα μη-εφικτές. Ένας απλός και γρήγορος ευρετικός αλγόριθμος, δέχεται σαν όρισμα την προτεινόμενη αλληλουχία  $x_{ij}^{t+1}$  και την επιδιορθώνει, ανακατατάσσοντας τα εμπορεύματα στα πλοία μέχρι να σχηματιστούν εφικτές διαδρομές  $\hat{x}_v$ .

*Κωδικοποίηση, αποκωδικοποίηση και επιδιόρθωση.* Κάθε επεξεργασία που επιδέχεται η λύση για να έρθει στα μέτρα μας, της κοστίζει και λίγη πληροφορία. Γιαυτό, η εξίσωση θέσης σπάνια θα δίνει μια βελτιωμένη λύση. Αυτό που καταφέρει όμως, είναι να στέλνει τα σωματίδια σε γειτονίες που θεωρεί ότι κρύβονται καλύτερες λύσεις. Αξιοποιώντας αυτή τη γνώση, μια σειρά τοπικών αναζητήσεων [LS](#) εφαρμόζεται σε κάθε  $x_{ij}^{t+1}$ , για να αναζητήσουν τη γειτονιά της και να ανακαλύψουν τη βελτιωμένη λύση. Έτσι, ενισχύεται άμεσα η εντατικοποίηση του αλγορίθμου.

Η νέα, επιδιορθωμένη πλέον λύση, υποβάλλεται σε αξιολόγηση από την αντικειμενική συνάρτηση. Αν  $f(x_{ij}^{t+1}) < f(x_{ij}^*)$ , μια καλύτε-

Κωδικοποίηση  
θέσεων

Αποκωδικοποίηση  
θέσεων

Επιδιόρθωση

Τοπική  
Αναζήτηση

Αξιολόγηση



ρη λύση βρέθηκε και αντικαθίσταται η προσωπική βέλτιστη λύση  $x_{ij}^* = x_{ij}^{t+1}$ .

Με την ολοκλήρωση του εσωτερικού βρόγχου, γίνεται έλεγχος για την ανανέωση του κυρίαρχου σωματιδίου του σμήνους  $g^*$ , και ο αλγόριθμος προχωρά στην επόμενη επανάληψη, μέχρι να ικανοποιηθεί η συνθήκη τερματισμού του.

### 5.3.2 Επιπλέον βελτίωση λύσης

#### Επανέναρξη

Η πρώτη τροποποίηση που έγινε, ήταν η εφαρμογή της μεθόδου επανέναρξης θέσης  $x$  και ταχύτητας  $v$ . Έστω ότι ένα σωματίδιο μετακινείται σε νέα γειτονιά με την εξίσωση θέσης, και εφαρμόζει μια σειρά τοπικών αναζητήσεων σε αυτή. Αν, με την ολοκλήρωση των αναζητήσεων, δεν καταφέρει να βρει μια βελτιωμένη λύση  $f(x_{ij}^{t+1}) < f(x^*)$ , διατρέχει τον κίνδυνο να παγιδευτεί σε μια υποβέλτιστη γειτονιά. Γιαυτό, το σωματίδιο γυρνάει πίσω στην καλύτερη θέση που έχει βρει έως τώρα  $x^*$  και η ταχύτητά του μηδενίζεται. Έτσι, σε επόμενες επαναλήψεις, έχει την ικανότητα να κινηθεί προς διαφορετικές κατευθύνσεις και να εξερευνήσει νέες γειτονιές.

#### Ανανέωση του πληθυσμού

Ακόμα και με την επανέναρξη, είναι αρκετά πιθανό τα σωματίδια να συγκλίνουν κάποια στιγμή σε τοπικά ελάχιστα και να είναι αδύνατο να βελτιωθούν. Για να αποφευχθεί αυτό το σενάριο, πρέπει να αυξηθεί η διάχυση, προκειμένου να εξερευνηθούν νέες περιοχές του χώρου λύσεων. Έτσι, γεννιέται η ανάγκη ανανέωσης του πληθυσμού.

Πλέον, κάθε σωματίδιο έχει μία ηλικία. Όσο το σωματίδιο παραμένει παραγωγικό, καταφέρει δηλαδή να βελτιώσει τη λύση του  $x^*$ , παραμένει νέο. Κάθε φορά που το σωματίδιο δεν καταφέρει να βελτιωθεί, η ηλικία του αυξάνεται κατά ένα χρόνο. Όταν φτάσει σε μια προκαθορισμένη ηλικία  $It_{mpa}$ , το σωματίδιο διαγράφεται, και ένα καινούριο δημιουργείται για να καλύψει τη θέση του στο σμήνος. Το καινούργιο σωματίδιο δεν είναι απόγονος του παλιού. Δηλαδή, δεν κληρονομεί τη μνήμη ταχύτητας και βέλτιστης θέσης, για να αποφευχθούν προκαταλήψεις (bias) που πιθανόν θα το παγιδέψουν στις ίδιες γειτονιές.

Η δημιουργία των σωματιδίων γίνεται με έναν τροποποιημένο [WEGRASP](#). Σε αυτή την παραλλαγή, μια ομοιόμορφη τυχαία μεταβλητή  $\gamma \in [-\frac{1}{10}, \frac{1}{10}]$  εφαρμόζεται στα βάρη  $w$  της συνάρτησης του [WEGRASP](#) για να ενισχύσει τη διάχυση των σωματιδίων.

#### Νέα εξίσωση ταχύτητας PSO

Πλέον, ο PSO αναλαμβάνει τον κρίσιμο ρόλο της κατεύθυνσης των νέων σωματιδίων προς τη σωστή μεριά. Έτσι, αυτά εκπαιδεύ-

ονται γρηγορότερα, συγκλίνουν σε βελτιωμένες λύσεις ευκολότερα, χωρίς να χρειάζονται πολύ περισσότερες τοπικές αναζητήσεις για να φτάσουν σε ένα εξίσου καλό επίπεδο ποιότητας λύσης με αυτή του κυρίαρχου σωματιδίου.

Έπειτα από εκτενή πειραματισμό, διαπιστώθηκε πως η κατάλληλη ισορροπία διάχυσης - εντατικοποίησης, απαραίτητη για την εκπαίδευση των νέων σωματιδίων, επιτεύχθηκε από την ακόλουθη παραλλαγή του Constriction PSO (**CPSO**):

$$v_{ij}^{t+1} = \chi(v_{ij}^t + \alpha\epsilon_1[g^* - x_{ij}^t] + \beta\epsilon_2[x_{ij}^{*(t)} - x_{ij}^t]), \quad (5.7)$$

$$\chi = \frac{2}{|2 - c - \sqrt{c^2 - 4c}|} \text{ και } c = \alpha + \beta, c > 4 \quad (5.8)$$

Η παραπάνω εξίσωση χρησιμοποιεί έναν παράγοντα περιορισμού (constriction factor)  $\chi$  για να μετριάσει τις ταχύτητες των σωματιδίων. Παρόλα αυτά, με τη μέθοδο της επανέναρξης τα σωματίδια βρίσκονται μονίμως στη βέλτιστη θέση τους  $x^*$ . Συνεπώς, ο γνωστικός παράγοντας ( $\beta\epsilon_2[x_{ij}^{*(t)} - x_{ij}^t]$ ) της εξίσωσης (5.7) θα είναι πάντα μηδέν. Αφού αυτός ο παράγοντας δε συμβάλει στους υπολογισμούς, μπορεί να παραληφθεί για να αποφευχθούν άσκοποι υπολογισμοί, που σε βάθος χρόνου καθυστερούν τον αλγόριθμο. Η εξίσωση ταχύτητας που προκύπτει είναι:

$$v_{ij}^{t+1} = \chi(v_{ij}^t + \alpha\epsilon_1[g^* - x_{ij}^t]) \quad (5.9)$$

Με αυτή την κίνηση όμως, αποσταθεροποιήθηκε η εξίσωση υπολογισμού του  $\chi$ , γιατί το  $\beta$  δεν υπάρχει πια. Γενικά, ο κοινωνικός παράγοντας  $\alpha$  πρέπει να έχει αρκετή βαρύτητα για να έλκονται τα σωματίδια σε ικανοποιητικό βαθμό από το κυρίαρχο σωματίδιο. Αλλά, αν αυξηθεί το  $\alpha$  σε 4 ώστε να ικανοποιείται η συνθήκη του  $c$ , τα σωματίδια θα ταξιδεύουν απότομα και θα παγιδεύονται στην ίδια γειτονιά με αυτόν. Η καλύτερη ισορροπία που επιτεύχθηκε ήταν με ανάθεση σταθερής τιμής στον παράγοντα περιορισμού  $\chi = 0.642$  και στον κοινωνικό παράγοντα  $\alpha = 2.2$ .

Εξίσου απαραίτητη ήταν και η αναδιαμόρφωση του  $\epsilon_1$  από τυχαίο διάνυσμα σε τυχαίο αριθμό, καθώς προκαλούσε σημαντικές διαταραχές στην αναπαράσταση των προτεινόμενων διαδρομών και καθιστούσε την ικανοποιητική επιδιόρθωση τους αδύνατη. Πλέον η λειτουργία του  $\epsilon$  είναι να αυξομειώνει την επιρροή του κυρίαρχου σωματιδίου για να ενισχύεται η διάχυση ή η εντατικοποίηση.

#### Διαχείριση Βέλτιστου Σωματιδίου

Όπως και τα σωματίδια, έτσι και όλο το σμήνος έχει μια ηλικία. Ουσιαστικά, τα σωματίδια μέσα στις επαναλήψεις του PSO, συναγωνίζονται για την ανακάλυψη της καλύτερης λύσης. Κάθε φορά που βρίσκεται ένα νέο κυρίαρχο σωματίδιο, η ζωή του σμήνους ανανεώνεται. Όσο το σμήνος δεν παράγει ικανούς ανταγωνιστές, δηλαδή δε βρίσκεται νέα βελτιωμένη λύση από κάποιο σωματίδιο εκτός

του κυρίαρχου σωματιδίου, η ηλικία του σμήνους αυξάνεται. Όταν το σμήνος φτάσει σε μια συγκεκριμένη ηλικία  $It_{msa}$ , το κυρίαρχο σωματίδιο αισθάνεται ότι το σμήνος δεν είναι πλέον παραγωγικό. Αποφασίζει, λοιπόν, να το αφήσει και να τρέξει να αναζητήσει μόνος του τη γειτονιά του.

Πρακτικά, όταν ο αλγόριθμος φτάσει σε αυτό το σημείο, είναι βέβαιο ότι έχει συγκλίνει σε κάποιο ακρότατο, από το οποίο δεν υπάρχει διαφυγή. Σταματάει να δημιουργεί νέα σωματίδια και να τα εκπαιδεύει με τη βοήθεια του PSO, για να μη δαπανάται υπολογιστική δύναμη σε άσκοπες αναζητήσεις. Πλέον η διαδικασία εστιάζεται στην εξερεύνηση της γειτονίας της καλύτερης λύσης  $g^*$  με χρήση τοπικών αναζητήσεων. Ακόμα και εδώ, όσο το κυρίαρχο σωματίδιο παραμένει παραγωγικό ( $f(g^{*t+1}) < f(g^{*t})$ ), συνεχίζει να αναζητά βελτιωμένες λύσεις, μέχρι να φτάσει μια ηλικία  $It_{mka}$  και ο αλγόριθμος να τερματιστεί.

### *Πολλαπλά Σμήνη και Παράλληλα*

Ανεξαρτήτως του βαθμού διάχυσης που προστίθεται στο σύστημα, εφόσον ο αλγόριθμος δεν ανήκει στην οικογένεια των ακριβών αλγορίθμων, η πιθανότητα απόκτησης της ολικά βέλτιστης λύσης είναι αβέβαιη. Όσο καλή και να είναι η ισορροπία διάχυσης - εντατικοποίησης που επιτεύχθηκε, υπάρχει ακόμα ο κίνδυνος, η τροχιά που θα ακολουθήσει το σμήνος μέσα στον χώρο λύσεων να το παγιδεύσουν σε μια υποβέλτιστη γειτονιά.

Ο πιο εύκολος και γρήγορος τρόπος αποφυγής αυτής της έκβασης είναι η χρήση πολλαπλών σμηνών. Έτσι, αυξάνεται η διάχυση χωρίς να ελλοιώνεται η εντατικοποίηση. Σε κάθε σμήνος, ο αλγόριθμος ξεκινάει με εντελώς διαφορετικές αρχικές λύσεις, βρίσκει νέα κυρίαρχα σωματίδια και ακολουθεί διαφορετικές τροχιές. Επισκέπτεται νέες γειτονιές και του δίνεται η δυνατότητα να ανακαλύψει λύσεις που διαφορετικά θα του ήταν αδύνατο.

Τα σμήνη εξετάζονται σειριακά ή παράλληλα. Είναι ανεξάρτητα μεταξύ τους, αφενός για να αποφευχθεί κάποια περιπλοκή στον υπολογισμό τους, και αφετέρου για να είναι δυνατός ο παράλληλος υπολογισμός τους. Σε υπολογιστικά συστήματα υψηλότερων επιδόσεων, με απαραίτητη προϋπόθεση την αρκετή διαθέσιμη μνήμη τυχαίας προσπέλασης (RAM), είναι δυνατό να κατανεμηθεί ο υπολογισμός κάθε σμήνους σε ένα πυρήνα του Η/Υ με αντιστοιχία ένα προς ένα. Έτσι, ο συνολικός χρόνος εκτέλεσης ολόκληρου του αλγορίθμου, υποπολλαπλασιάζεται με τον αριθμό των διαθέσιμων πυρήνων, επιταχύνοντας σημαντικά την εκτέλεσή του.



## ΑΠΟΤΕΛΕΣΜΑΤΑ

Το 2014 μια μεγάλη ομάδα ρεαλιστικών παραδειγμάτων<sup>1</sup>, βασισμένα σε πραγματικά δεδομένα, έγινε διαθέσιμη στην επιστημονική κοινότητα από τους Hemmati et al. [10], για το πρόβλημα του *ITSRSP*. Εξήντα (60) από αυτά επιλέχθηκαν για να εξεταστεί η αποτελεσματικότητα του *ACPSO*. Τα παραδείγματα μοντελοποιούν το πρόβλημα για εμπορεύματα πλήρους φορτίου (full load) και υπερατλαντικά ταξίδια (deep sea). Τα προς εξυπηρέτηση εμπορεύματα κυμαίνονται από 8 έως 100, ο στόλος αποτελείται από 3 έως 50 πλοία, ενώ τα χαρακτηριστικά των εμπορευμάτων και η ετερογένεια των πλοίων διαφέρουν για κάθε παράδειγμα. Συνεπώς, το φάσμα των δεδομένων είναι αρκετά ευρύ ώστε να εξεταστεί η εφικτότητα του πρωταρχικού μας στόχου.

Τα προβλήματα δρομολόγησης και χρονικού προγραμματισμού που καλούνται να αντιμετωπίσουν οι περισσότερες ναυτιλιακές εταιρείες σε καθημερινή βάση, εμπλέκουν το πολύ λίγες δεκάδες πλοία. Παρόλα αυτά, σε σπάνιες περιπτώσεις, όπως για παράδειγμα σε συγχώνευση εταιρειών ή συντονισμένη οργάνωση στόλων, προκύπτουν προβλήματα πολύ μεγαλύτερων διαστάσεων. Το 2020, οι Homsi et al. [11] κατασκεύασαν μια εξίσου ρεαλιστική σειρά παραδειγμάτων<sup>2</sup>, που αντικατοπτρίζουν σενάρια τέτοιου τύπου. Οκτώ εκ' των παραπάνω παραδειγμάτων, επιλέχθηκαν για να πιέσουμε τον *ACPSO* στα άκρα και να εξεταστεί η επεκτασιμότητά του. Τα παραδείγματα ανήκουν στην ίδια κατηγορία (deep sea with full loads), τα προς εξυπηρέτηση εμπορεύματα κυμαίνονται από 110 έως 200, και ο στόλος αποτελείται από 52 έως 89 πλοία.

Αποσκοπώντας στη λεπτομερέστερη ανάλυσή τους, το σύνολο των παραπάνω παραδειγμάτων, χωρίστηκε σε μικρά, μεσαία, μεγάλα και πολύ μεγάλα, ανάλογα με τον αριθμό των διαθέσιμων εμπορευμάτων.

Στους πίνακες που ακολουθούν, συγκρίνονται τα αποτελέσματα του *ACPSO* με τις υπάρχουσες βέλτιστες λύσεις<sup>3</sup> από την βιβλιογραφία. Η πρώτη στήλη του πίνακα, περιγράφει τα χαρακτηριστικά του κάθε παραδείγματος. Τα εμπορεύματα αντιπροσωπεύονται από το πρόθεμα C, τα πλοία από το V και η παραλλαγή της ετερογένειας από το HE. Στην τέταρτη στήλη παρουσιάζεται η ποιότητα της λύσης, που υπολογίζεται ως  $ER = \frac{ACPSO-Optimal}{Optimal} \%$ . Στην

1 Τα αρχεία του 2014 είναι διαθέσιμα στην ιστοσελίδα:

<https://home.himolde.no/hvattum/benchmarks/>

2 Τα αρχεία του 2020 είναι διαθέσιμα στην ιστοσελίδα: <https://w1.cirrelet.ca/~vidalt/en/VRP-resources.html>

3 Οι έως σήμερα βέλτιστες λύσεις, επιτεύχθηκαν από τους Homsi, Martinelli, Vidal και Fagerholt [11] το 2020



Εικόνα 6.1: Παγκόσμιος χάρτης με τα λιμάνια που εξυπηρετούν πλοία υπερατλαντικών ταξιδιών

| ΜΕΓΕΘΟΣ     | ΕΜΠΟΡΕΥΜΑΤΑ | ΣΤΟΛΟΣ  |
|-------------|-------------|---------|
| Μικρά       | 8 - 16      | 3 - 6   |
| Μεσαία      | 17 - 35     | 6 - 13  |
| Μεγάλα      | 50 - 100    | 20 - 50 |
| Πολύ μεγάλα | 110 - 200   | 52 - 89 |

Πίνακας 6.1: Υποκατηγορίες δεδομένων

#### Αποτελέσματα

τελευταία στήλη, με T συμβολίζεται ο υπολογιστικός χρόνος απόκτησης του αποτελέσματος σε λεπτά.

**ΜΙΚΡΑ** Στα παραδείγματα μικρού μεγέθους (βλ. [Πίνακα 6.2](#)) εμφανίζονται σημαντικά αποτελέσματα. Επιπρόσθετα, ο [WEGRASP](#) κατάφερε στην πρώτη του φάση, να κατασκευάσει την ολικά βέλτιστη λύση για ορισμένα παραδείγματα, είτε να βρει μια σχεδόν βέλτιστη για τα υπόλοιπα, σε κλάσματα του δευτερολέπτου. Σε δεκατέσσερα (14) από τα είκοσι (20), ο [ACPSO](#) καταφέρνει να βρει την ολικά βέλτιστη λύση. Κατά μέσο όρο χρειάζεται λιγότερο από 3 δευτερόλεπτα για να φτάσει την καλύτερη λύση και καταφέρνει να επιλύσει όλα τα παραδείγματα με μέση απόκλιση 1,69%.

**ΜΕΣΑΙΑ** Στου μεσαίου μεγέθους (βλ. [Πίνακα 6.3](#)), ο περιορισμός της ετερογένειας ξεκινάει να αναδεικνύει την επιρροή του. Παρατηρώντας πιο προσεκτικά την υποομάδα των C25\_V7, διακρίνεται το παράδοξο σενάριο, σε κάποια παραδείγματα να έχει επιτευχθεί το ολικά βέλτιστο, ενώ σε άλλο να υπάρχει σημαντική απόκλιση. Παρά ταύτα, συνολικά επιτυγχάνεται απόκλιση μόλις 1,51% σε περίπου ένα λεπτό.

| ΑΡΧΕΙΟ             | OPTIMAL  | ACPSO    | ER    | T    |
|--------------------|----------|----------|-------|------|
| DS_FUN_C8_V3_HE_1  | 9584863  | 9584863  | 0,00  | 0,00 |
| DS_FUN_C8_V3_HE_2  | 9369654  | 10106054 | 7,86  | 0,00 |
| DS_FUN_C8_V3_HE_3  | 4596681  | 4596681  | 0,00  | 0,01 |
| DS_FUN_C8_V3_HE_4  | 6899730  | 6899730  | 0,00  | 0,00 |
| DS_FUN_C8_V3_HE_5  | 6815253  | 8262007  | 21,23 | 0,02 |
| DS_FUN_C11_V4_HE_1 | 34854819 | 34854819 | 0,00  | 0,00 |
| DS_FUN_C11_V4_HE_2 | 25454434 | 25759174 | 1,20  | 0,05 |
| DS_FUN_C11_V4_HE_3 | 29627143 | 29627143 | 0,00  | 0,14 |
| DS_FUN_C11_V4_HE_4 | 33111680 | 33156278 | 0,13  | 0,01 |
| DS_FUN_C11_V4_HE_5 | 28175914 | 28175914 | 0,00  | 0,00 |
| DS_FUN_C13_V5_HE_1 | 11629005 | 11629005 | 0,00  | 0,02 |
| DS_FUN_C13_V5_HE_2 | 11820655 | 11820655 | 0,00  | 0,00 |
| DS_FUN_C13_V5_HE_3 | 9992593  | 9992593  | 0,00  | 0,01 |
| DS_FUN_C13_V5_HE_4 | 12819619 | 13135553 | 2,46  | 0,01 |
| DS_FUN_C13_V5_HE_5 | 10534892 | 10534892 | 0,00  | 0,01 |
| DS_FUN_C16_V6_HE_1 | 51127590 | 51127590 | 0,00  | 0,21 |
| DS_FUN_C16_V6_HE_2 | 44342796 | 44342796 | 0,00  | 0,11 |
| DS_FUN_C16_V6_HE_3 | 45391842 | 45391842 | 0,00  | 0,00 |
| DS_FUN_C16_V6_HE_4 | 39687114 | 40037355 | 0,88  | 0,11 |
| DS_FUN_C16_V6_HE_5 | 42855603 | 42855603 | 0,00  | 0,12 |
| Average            |          |          | 1,69  | 0,04 |

Πίνακας 6.2: Αποτελέσματα υποκατηγορίας "Μικρά"

**ΜΕΓΑΛΑ** Τα αποτελέσματα των παραδειγμάτων μεγάλου μεγέθους (βλ. Πίνακα 6.4) ενισχύουν την παραδοχή ότι, η ετερογένεια αποτελεί τον ισχυρότερο αποσταθεροποιητικό παράγοντα. Το ίδιο μοτίβο παρατηρείται στην ποιότητα των λύσεων, με τις αποκλίσεις να διακυμαίνονται ανά τα παραδείγματα συνεχώς. Από αυτό το σημείο και έπειτα, ο **WEGRASP** απέδειξε την αναγκαιότητά του, καθώς άλλες απλούστερες μέθοδοι κατασκευής αρχικών λύσεων που εξετάστηκαν, δεν κατάφεραν να επιτύχουν το επιθυμητό επίπεδο ποιότητας και διασποράς, απαραίτητο για την επιτάχυνση της επίλυσης. Τα δεδομένα εδώ, έχουν αυξηθεί σημαντικά, και ακόμα ο **ACPSO** καταφέρνει μέσα σε λιγότερο από 2 λεπτά να επιτύχει μέση απόκλιση 4,62%.

**ΠΟΛΥ ΜΕΓΑΛΑ** Στην τελευταία κατηγορία (βλ. Πίνακα 6.5), το μέγεθος των δεδομένων έχει αυξηθεί τόσο πολύ που αγγίζει τα όρια του πλασματικού. Για μία ακόμα φορά, η μεγαλύτερη απόκλιση, αντί να εμφανιστεί στο υπερμεγέθες παράδειγμα των διακοσίων

| ΑΡΧΕΙΟ              | OPTIMAL  | ACPSO    | ER   | T    |
|---------------------|----------|----------|------|------|
| DS_FUN_C17_V13_HE_1 | 17316720 | 17316720 | 0,00 | 0,38 |
| DS_FUN_C17_V13_HE_2 | 12194861 | 12226871 | 0,26 | 0,81 |
| DS_FUN_C17_V13_HE_3 | 12091554 | 12119704 | 0,23 | 0,46 |
| DS_FUN_C17_V13_HE_4 | 12847653 | 12870749 | 0,18 | 0,55 |
| DS_FUN_C17_V13_HE_5 | 13213406 | 13213513 | 0,00 | 1,66 |
| DS_FUN_C20_V6_HE_1  | 16406738 | 16406738 | 0,00 | 0,28 |
| DS_FUN_C20_V6_HE_2  | 16079401 | 16079401 | 0,00 | 0,11 |
| DS_FUN_C20_V6_HE_3  | 17342200 | 17347580 | 0,03 | 0,02 |
| DS_FUN_C20_V6_HE_4  | 16529748 | 16567762 | 0,23 | 0,26 |
| DS_FUN_C20_V6_HE_5  | 17449378 | 17648163 | 1,14 | 0,09 |
| DS_FUN_C25_V7_HE_1  | 22773158 | 22799562 | 0,12 | 0,12 |
| DS_FUN_C25_V7_HE_2  | 20206329 | 20377734 | 0,85 | 0,69 |
| DS_FUN_C25_V7_HE_3  | 19108952 | 19108952 | 0,00 | 0,17 |
| DS_FUN_C25_V7_HE_4  | 22668675 | 22668675 | 0,00 | 0,17 |
| DS_FUN_C25_V7_HE_5  | 23036603 | 23922190 | 3,84 | 0,14 |
| DS_FUN_C35_V13_HE_1 | 86951609 | 90031391 | 3,54 | 5,45 |
| DS_FUN_C35_V13_HE_2 | 83422071 | 85899158 | 2,97 | 3,76 |
| DS_FUN_C35_V13_HE_3 | 83898591 | 88187646 | 5,11 | 3,01 |
| DS_FUN_C35_V13_HE_4 | 91970481 | 96468791 | 4,89 | 0,55 |
| DS_FUN_C35_V13_HE_5 | 91123040 | 97394320 | 6,88 | 1,80 |
| Average             |          |          | 1,51 | 1,02 |

Πίνακας 6.3: Αποτελέσματα υποκατηγορίας "Μεσαία"

εμπορευμάτων, εμφανίζεται στο δεύτερο μικρότερο. Ο **ACPSO** ανταπεξέρχεται αξιοπρεπώς στις αυξημένες απαιτήσεις, καθώς χρειάζεται περίπου 5 λεπτά για να βρει την καλύτερη λύση, και καταφέρνει να επιλύσει όλα τα παραδείγματα με μέση απόκλιση 5,48%. Εδώ οφείλει να τονιστεί πως, από ένα σημείο της εκτέλεσης και έπειτα, οι διαδρομές γίνονται τόσο ευαίσθητες σε νέες μεταβολές, που οι εξισώσεις του PSO δεν καταφέρνουν να οδηγήσουν σε νέες γειτονιές και πρακτικά είναι περιττές. Συνεπώς, χωρίς την ύπαρξη του μηχανισμού "Διαχείριση βέλτιστου σωματιδίου", τέτοια επίπεδα ποιότητας - χρόνου, θα ήταν αδύνατα.

Τιμές  
παραμέτρων

Η επίτευξη των παραπάνω αποτελεσμάτων, έγινε έπειτα από εκτενέστατο πειραματισμό με τα δομικά στοιχεία και τις παραμέτρους του αλγορίθμου. Οι τιμές των παραμέτρων που αποδείχθηκαν ως οι αποδοτικότερες παρατίθενται στη συνέχεια. Αρχικά, 10 σμήνη με 10 σωματίδια το καθένα κρίθηκαν ικανά για την αποδοτική εξερεύνηση του χώρου λύσεων. Με λιγότερα σμήνη και σωματίδια θυσιάζεται η διάχυση, ενώ με περισσότερα, η ταχύτη-

| ΑΡΧΕΙΟ               | OPTIMAL   | ACPSO     | ER    | T    |
|----------------------|-----------|-----------|-------|------|
| DS_FUN_C50_V20_HE_1  | 41310946  | 41988576  | 1,64  | 0,90 |
| DS_FUN_C50_V20_HE_2  | 37784994  | 38311676  | 1,39  | 1,90 |
| DS_FUN_C50_V20_HE_3  | 39841724  | 40372174  | 1,33  | 1,30 |
| DS_FUN_C50_V20_HE_4  | 43941098  | 44235878  | 0,67  | 1,31 |
| DS_FUN_C50_V20_HE_5  | 41947437  | 42449467  | 1,20  | 1,71 |
| DS_FUN_C70_V30_HE_1  | 142679953 | 152258642 | 6,71  | 1,29 |
| DS_FUN_C70_V30_HE_2  | 135031988 | 144318072 | 6,88  | 1,78 |
| DS_FUN_C70_V30_HE_3  | 162759203 | 165559410 | 1,72  | 0,99 |
| DS_FUN_C70_V30_HE_4  | 155855123 | 164141399 | 5,32  | 1,72 |
| DS_FUN_C70_V30_HE_5  | 156557723 | 162390589 | 3,73  | 1,06 |
| DS_FUN_C90_V40_HE_1  | 190627186 | 196733405 | 3,20  | 1,33 |
| DS_FUN_C90_V40_HE_2  | 189770977 | 208708080 | 9,98  | 1,48 |
| DS_FUN_C90_V40_HE_3  | 211038412 | 225678120 | 6,94  | 1,21 |
| DS_FUN_C90_V40_HE_4  | 210449287 | 215546400 | 2,42  | 1,70 |
| DS_FUN_C90_V40_HE_5  | 197804917 | 211374061 | 6,86  | 1,95 |
| DS_FUN_C100_V50_HE_1 | 205826535 | 216102072 | 4,99  | 2,03 |
| DS_FUN_C100_V50_HE_2 | 207809147 | 222518309 | 7,08  | 1,74 |
| DS_FUN_C100_V50_HE_3 | 217000928 | 239358598 | 10,30 | 3,88 |
| DS_FUN_C100_V50_HE_4 | 220879632 | 234357596 | 6,10  | 1,72 |
| DS_FUN_C100_V50_HE_5 | 223265017 | 232262522 | 4,03  | 1,33 |
| Average              |           |           | 4,62  | 1,62 |

Πίνακας 6.4: Αποτελέσματα υποκατηγορίας "Μεγάλα"

τα σύγκλισης του αλγορίθμου. Η μέγιστη ηλικία σωματιδίου  $It_{mpa}$  ορίστηκε ως 10, ώστε να του παραχωρείται αρκετός χρόνος για τη βελτίωσης της λύσης του. Η μέγιστη ηλικία του σμήνους  $It_{msa}$  ορίστηκε ως 100, ώστε να υπάρχει αρκετό περιθώριο ανανέωσης του πληθυσμού. Η ηλικία του κυρίαρχου σωματιδίου  $It_{mka}$  υπολογίζεται δυναμικά ως  $20 \times n$  (αριθμός εμπορευμάτων), προσδίδοντας του την απαραίτητη ευελιξία προσαρμογής στο μέγεθος των δεδομένων. Οι τοπικές αναζητήσεις, αποδείχθηκε ότι λειτουργούν καλύτερα όταν, ο μέγιστος αριθμός αποτυχημένων επαναλήψεων  $\lambda$  είναι 5. Τέλος, όπως αναφέρεται αναλυτικότερα παραπάνω, η τιμή του παράγοντα περιορισμού της ταχύτητας  $\chi$  είναι 0.642, και ο κοινωνικός όρος  $\alpha$  ισούται με 2.2.

Αναφορικά με την δομή του [ACPSO](#), ο πυρήνας του αλγορίθμου, όπως αποτυπώνεται στην υποενότητα [5.3.1](#), παρουσίασε προοπτικές επιτυχούς εφαρμογής για τη βελτιστοποίηση του [ITSRSP](#), αλλά δεν ήταν αρκετός για να ικανοποιήσει τον πρωταρχικό μας στόχο.

Σχόλια

| ΑΡΧΕΙΟ               | OPTIMAL   | ACPSO     | ER    | T     |
|----------------------|-----------|-----------|-------|-------|
| DS_FUN_C110_V52_HE_1 | 240011111 | 244774226 | 1,98  | 2,16  |
| DS_FUN_C120_V53_HE_1 | 248614953 | 278211308 | 11,90 | 2,81  |
| DS_FUN_C130_V58_HE_1 | 288771846 | 303818562 | 5,21  | 2,19  |
| DS_FUN_C140_V62_HE_1 | 303231470 | 325820213 | 7,45  | 2,37  |
| DS_FUN_C150_V67_HE_1 | 323442822 | 334507575 | 3,42  | 4,25  |
| DS_FUN_C160_V71_HE_1 | 370429540 | 391681142 | 5,74  | 7,77  |
| DS_FUN_C170_V76_HE_1 | 395641818 | 411235863 | 3,94  | 7,54  |
| DS_FUN_C200_V89_HE_1 | 430970779 | 449013129 | 4,19  | 11,65 |
| Average              |           |           | 5,48  | 5,09  |

Πίνακας 6.5: Αποτελέσματα υποκατηγορίας "Πολύ μεγάλα"

Κατά την πειραματική διαδικασία, διαπιστώθηκε πως η εξίσωση ταχύτητας του κλασσικού PSO έκανε τα σωματίδια να συγκλίνουν ταχύτατα σε κάποιες λύσεις, και ο μηχανισμός που σε συνεχή προβλήματα τους επιτρέπει να δραπετεύσουν από τοπικά ελάχιστα, δε δούλευε ικανοποιητικά για αυτό το διακριτό πρόβλημα. Οι πιο χαρακτηριστικές αστοχίες της μεθόδου ήταν ότι πρώτον, οι βέλτιστες λύσεις του PSO από εκτέλεση σε εκτέλεση είχαν σημαντικές αποκλίσεις, και δεύτερον η επανεμφάνιση της ολικά βέλτιστης λύσης ήταν σχεδόν απίθανη. Οι επιπλέον μεθοδολογίες που αναπτύχθηκαν, αποσκοπούσαν στην γεφύρωση αυτού του χάσματος απόδοσης.

Εξίσου κρίσιμη για την αποτελεσματικότητα του αλγορίθμου αποδείχτηκε η διατήρηση της ισορροπίας μεταξύ εντατικοποίησης και διάχυσης. Ένας από τους παράγοντες που επηρεάζουν την ισορροπία, είναι η επαναληπτική διαδικασία και οι συνθήκες τερματισμού που θα επιλεγούν για τις τοπικές αναζητήσεις.

Σαν επαναληπτική διαδικασία, πρώτα εφαρμόζεται η **LS** του 1-0 Relocate που, όπως περιγράφηκε προηγουμένως, επιχειρεί να δημιουργήσει χώρο στις διαδρομές των πλοίων, λειτουργώντας ταυτόχρονα και ως προκαταρκτικό στάδιο της επόμενης **LS**. Έπειτα η 1-Y Interexchange, προσπαθεί να εισαγάγει ναυλωμένα εμπορεύματα και τελευταία η 1-1 Interexchange βελτιώνει τις εσωτερικές διαδρομές.

Αύξηση της διάχυσης επέρχεται από αύξηση του χάους, και για να επιτευχθεί αυτό, σαν συνθήκη τερματισμού των **LS** επιλέχθηκε ένας αριθμός αποτυχημένων επαναλήψεων  $\lambda$ . Δηλαδή, όσο μια **LS** βρίσκει βελτιωμένες λύσεις, συνεχίζει την αναζήτηση στη γειτονιά της. Όσο το  $\lambda$  ελαττώνεται, η διάχυση αυξάνεται. Αντιθέτως, όσο το  $\lambda$  αυξάνεται, αυξάνεται και η εντατικοποίηση. Τιμές που διαπιστώθηκε ότι προτιμούνται, ανήκουν στο διάστημα  $[5, 10]$ . Με τη μέθοδο αυτή, οι τροχιές που ακολουθούν τα σωματίδια στον



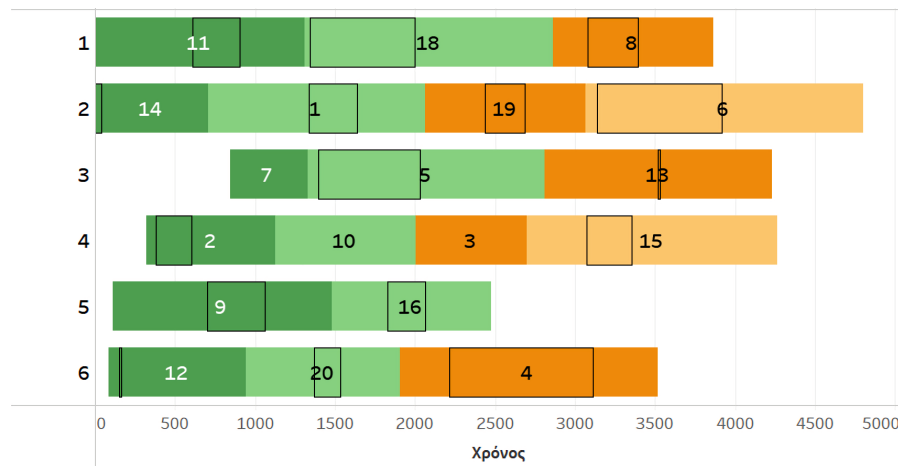
χώρο λύσεων είναι περισσότερο χαοτικές, και συνεπώς είναι πιο δύσκολο να παγιδευτούν σε τοπικά ελάχιστα.

Ο αλγόριθμος του **ACPSO** υλοποιήθηκε στο προγραμματιστικό περιβάλλον της Matlab, σε υπολογιστή με επεξεργαστή Intel Core i5-3570 @3.40GHz και 8GB RAM. Προκειμένου να εξεταστεί η αξιοπιστία του **ACPSO** ως προς την ικανότητά του να παράγει το βέλτιστο δυνατό αποτέλεσμα με ελάχιστες εκτελέσεις, πραγματοποιήθηκε η ακόλουθη μελέτη. Σε κάθε παράδειγμα, εφαρμόστηκε μία επανάληψη του **ACPSO** με δέκα σμήνη. Τα αποτελέσματα  $x_i$  κάθε σμήνους συλλέχθηκαν, και στον **Πίνακα Α'1** παρουσιάζονται η ελάχιστη  $\min(x_i)$  και η μέγιστη  $\max(x_i)$  απόκλιση από το ολικό βέλτιστο, καθώς και η μέση απόδοση  $\bar{x}$ , ανά παράδειγμα. Επίσης, συμπεριλαμβάνεται η απόλυτη απόκλιση  $D$  της ελάχιστης τιμής από τη μέση απόδοση  $|\min(x_i) - \bar{x}|$  ανά παράδειγμα, για να μελετηθεί η χρησιμότητα των πολλαπλών σμηνών. Όπως είναι φανερό, οι αποκλίσεις σε ορισμένα παραδείγματα είναι σημαντικές. Συμπερασματικά, η χρήση πολλαπλών σμηνών κρίνεται απαραίτητη. Αναφορικά με την αξιοπιστία του, σε 61 από τα 68 παραδείγματα ο αλγόριθμος καταφέρνει να ξαναβρεί τη βέλτιστη λύση του.

Αποτελεσματικό-  
τητα

## 6.1 ΠΑΡΑΔΕΙΓΜΑ

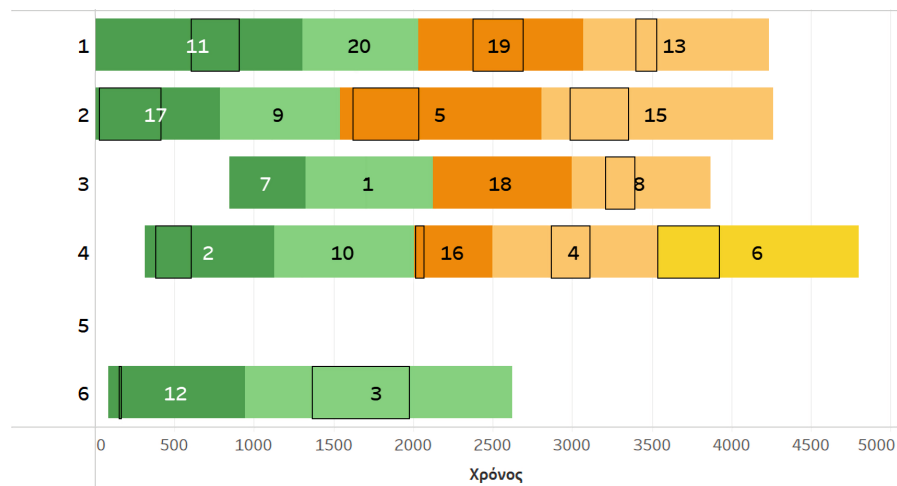
Η φύση του προβλήματος θα γίνει πιο κατανοητή στον αναγνώστη με το ακόλουθο απλό παράδειγμα, που εμπλέκει 20 εμπορεύματα προς εξυπηρέτηση, από 6 πλοία (DS\_FUN\_C20\_V6\_HE2). Στον κάθετο άξονα παραθέτονται τα πλοία και στον οριζόντιο ο χρόνος εργασίας σε ώρες. Η συνολική διάρκεια λειτουργίας κάθε πλοίου, καταμερίζεται στα εμπορεύματα που εξυπηρετεί. Ένα διακριτικό κενό ορθογώνιο, πλαισιώνει τις ώρες που το πλοίο παραμένει αδρανές.



Εικόνα 6.2: Κατανομή των εμπορευμάτων στα πλοία βάση της αρχικής λύσης

Στην πρώτη εικόνα 6.2, παρουσιάζεται η καλύτερη αρχική λύση που κατάφερε να κατασκευάσει ο **WEGRASP**. Αρχικά, αξίζει να

σημειωθεί η ανάθεση των εμπορευμάτων στα πλοία και ο χρόνος αδράνειάς τους. Επίσης, οφείλει να υπογραμμιστεί πως δεν είναι διαθέσιμα για αξιοποίηση όλα τα πλοία τη χρονική στιγμή μηδέν. Συνεπώς, γίνεται άμεσα αντιληπτό ότι οι δύο παραπάνω περιορισμοί του προβλήματος αυξάνουν την πολυπλοκότητά του. Δεκαεννέα από τα είκοσι εμπορεύματα εξυπηρετούνται από τον στόλο, και ένα ανατίθεται προς ναύλωση. Η εν λόγω λύση κατάφερε να επιτύχει απόκλιση 8,2% από το ολικά βέλτιστο.



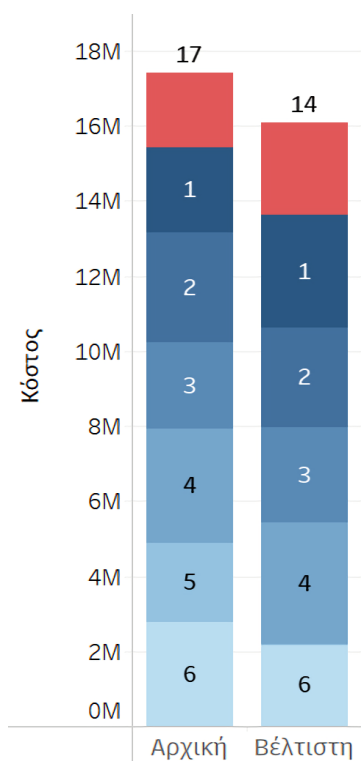
Εικόνα 6.3: Κατανομή των εμπορευμάτων στα πλοία βάση της ολικά βέλτιστης λύσης

Στη δεύτερη εικόνα 6.3, παρουσιάζεται η ολικά βέλτιστη λύση<sup>4</sup> που κατάφερε να βρει ο ACP SO. Οι βελτιώσεις που παρατηρούνται είναι οι εξής:

1. Αποδεικνύεται η σχέση δράσης - αντίδρασης που επιφέρει ο περιορισμός των χρονικών παραθύρων. Δηλαδή, η αλλαγή ενός εμπορεύματος μέσα σε μια διαδρομή, επηρεάζει όλες τις ακόλουθες επιλογές.
2. Αποδεικνύεται η χρησιμότητα του “Μηχανισμού Διχοτόμησης” στις τοπικές αναζητήσεις, από τη μετακίνηση του ζεύγους 18 - 8.
3. Ευχάριστη εντύπωση προκαλεί η απουσία του 5ου πλοίου, καθώς τα εμπορεύματά του μοιράστηκαν με καλύτερο τρόπο στα υπόλοιπα πλοία.
4. Η συνολική αδράνεια του στόλου, μειώθηκε από 5467 ώρες σε μόλις 3642.
5. Τα εμπορεύματα που εξυπηρετήθηκαν από τον στόλο παραμένουν δεκαεννέα αλλά ανταλλάχθηκε το εμπόρευμα προς ναύλωση 17 με το 14.

<sup>4</sup> 0% απόκλιση από την έως σήμερα γνωστή βέλτιστη λύση.





Στην τρίτη εικόνα 6.4, απεικονίζεται το συνολικό κόστος ανά λύση. Στα αριστερά βρίσκεται η αρχική λύση με κόστος €17 401 711.00 και στα δεξιά η βέλτιστη λύση με €16 079 401.00 επει-  
 τυγχάνοντας εξοικονόμηση δαπανών ύψους €1 322 310.00. Με αποχρώσεις του μπλε επιμερίζεται το συνολικό κό-  
 στος λειτουργίας σε κάθε πλοίο, και με κόκκινο το κόστος ναύλωσης του ε-  
 μπορεύματος που δεν εξυπηρετήθηκε από τον στόλο. Όπως είναι εμφανές, ένα μέρος του κόστους λειτουργίας α-  
 πορροφήθηκε από το 1ο, 3ο και 4ο πλοίο. Η απουσία, όμως, του 5ου και η εξοικονόμηση από το 2ο και 6ο, ή-  
 ταν ικανές να υπερνικήσουν την αύξη-  
 ση του κόστους λειτουργίας και ναύ-  
 λωσης.

Εικόνα 6.4: Σύγκριση κό-  
 στους ανά λύση



Στην παρούσα διπλωματική εργασία προτάθηκε μια νέα παραλλαγή του PSO, ο Adaptive Constriction PSO (**ACPSO**), για την επίλυση του Industrial and Tramp Shipping Routing and Scheduling Problem. Ο **ACPSO** οφείλει την αποτελεσματικότητά του σε τρία κομβικής σημασίας στοιχεία, στο *Weight Enhanced GRASP* (**WEGRASP**) για την αρχικοποίηση θέσεων των σωματιδίων, στη δομή των *τοπικών αναζητήσεων*, και στην *προσαρμοστικότητα* του.

Ο **WEGRASP** είναι σημαντικός, γιατί μπορεί να κατασκευάζει πολύ ποιοτικές λύσεις, με ικανοποιητική διασπορά, ανεξαρτήτως μεγέθους. Αναφορικά με την ποιότητα των λύσεων, δύο απλούστερες μέθοδοι παρασκευής αρχικών λύσεων δοκιμάστηκαν στο παρελθόν και απέβησαν αναποτελεσματικές. Αρχικά, μία εντελώς τυχαία προσέγγιση στην ανάθεση των εμπορευμάτων στα πλοία, παρουσίασε διαδρομές με ελάχιστα εξυπηρετημένα εμπορεύματα και μεγάλο πλήθος ναυλωμένων. Η δεύτερη μέθοδος που επιχειρήθηκε, αξιοποιούσε έναν μηχανισμό ομαδοποίησης των εμπορευμάτων (clustering) βάση των χρονικών παραθύρων τους. Ένας ευρετικός αλγόριθμος ήταν υπεύθυνος για τον σχηματισμό εφικτών διαδρομών για κάθε πλοίο. Η μέθοδος έδειξε προοπτικές ικανοποιητικής εφαρμογής, αλλά σε σχέση με τον **WEGRASP** υστερούσε στην ποιότητα των παραχθέντων λύσεων, καθώς και στη διασπορά τους στον χώρο.

Οι τοπικές αναζητήσεις αποτελούν τον κρισιμότερο μηχανισμό εντατικοποίησης του αλγορίθμου. Η ικανότητά τους να αναδιατάσσουν και να εισάγουν επιπλέον εμπορεύματα στα πλοία, θα μπορούσε να χαρακτηριστεί ως το κλειδί για την επίλυση του πολυσύνθετου προβλήματος του **ITSRSP**, διότι διαθέτουν τη δύναμη να μετατρέψουν μια πολύ καλή λύση σε μία βέλτιστη. Επίσης, ο “Μηχανισμός Διχοτόμησης” σε συνδυασμό με την κατάλληλη επιλογή της συνθήκης τερματισμού, δίνουν τη δυνατότητα στα σωματίδια να ξεφεύγουν από τοπικά ελάχιστα που παγιδεύονται. Τέλος, μια συνετή εστίαση στη γειτονία αναζήτησης, επιταχύνει σε βάθος χρόνου την εκτέλεση του **ACPSO** σημαντικά.

Φυσικά, η χρησιμότητα του αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων, δεν μπορεί να τονιστεί αρκετά. Ο PSO αποτελεί τη βάση πάνω στην οποία συνδυάζονται όλοι οι μηχανισμοί. Αυτός είναι που προσδίδει την ευφυΐα στο σύστημα. Σε αυτόν οφείλεται η επίτευξη ποιοτικών λύσεων σε τέτοιους συναρπαστικούς χρόνους. Η προσαρμοστική του φύση, αν και του κοστίζει σε υπολογιστικό χρόνο, είναι αυτή που του επιτρέπει να βρει το βέλτιστο αποτέλεσμα που είναι ικανός να παράξει, σε ελάχιστες εκτελέσεις, ανεξαρτήτως πολυπλοκότητας και μεγέθους των δεδομένων. Συμπε-

ρασματικά, θα μπορούσε να ειπωθεί πως ο πρωταρχικός στόχος της έρευνας, κατακτήθηκε.

Μελλοντική έρευνα, θα μπορούσε να ακολουθήσει τις εξής οδούς. Αρχικά, μπορούν να γίνουν ενέργειες για την μείωση των αποκλίσεων. Αυτό θα μπορούσε να επιτευχθεί με μία πιο προηγμένη και εξεζητημένη δομή τοπικών αναζητήσεων. Ένα άλλο σημείο που χρήζει περισσότερης διερεύνησης, είναι ο τρόπος προσαρμογής του PSO σε διακριτά δεδομένα. Η προσέγγιση που ακολουθήθηκε στην παρούσα εργασία, ήταν η μετάφραση και επισκευή των διανυσμάτων θέσης. Άλλοι επιστήμονες στη βιβλιογραφία έχουν σχεδιάσει εκπληκτικές μεθόδους, που είτε δεν απαιτούν τη διαδικασία μετάφρασης, είτε δεν εμπλέκουν καθόλου τη θέση των σωματιδίων στη λύση [13, 14]. Όσον αφορά τα παραδείγματα μεγαλύτερων διαστάσεων, μια πιο στοχευμένη διαδικασία υπολογισμού των ταχυτήτων των σωματιδίων θα μπορούσε να αναπτυχθεί, πιθανά με κάποια “Unified Crossover” μέθοδο, ώστε να πραγματοποιούνται λεπτεπίλεπτες μεταβολές στα διανύσματα θέσης, για να επιταχυνθεί περισσότερο η εκτέλεση του αλγορίθμου.

Περαιτέρω, θα μπορούσε να διερευνηθεί ο ορίζοντας του προβλήματος, συμπεριλαμβάνοντας ταξίδια ρηχής θάλασσας (short sea), διαδρομές μικτού φορτίου (mixed load), είτε διάφοροι συνδυασμοί. Το κάθε πρόβλημα έχει να αντιμετωπίσει τις δικές του προκλήσεις. Για παράδειγμα, τα short sea έχουν πιο στενά χρονικά παράθυρα από τα deep sea, και είναι πιθανό η παραδοχή που έγινε για την επιλογή κατάλληλης γειτονιάς, να μην μπορεί να εφαρμοστεί επιτυχώς εκεί. Από την άλλη, τα μικτού φορτίου κατά την επίσκεψή τους σε ένα λιμάνι, έχουν τη δυνατότητα να φορτώσουν και να ξεφορτώσουν πάνω από ένα εμπόρευμα τη φορά. Συνεπώς, η μοντελοποίηση τους πρέπει να γίνει σε επίπεδο λιμανιού, αντί σε επίπεδο εμπορεύματος, όπως γίνεται με τα full load.

Μέρος III

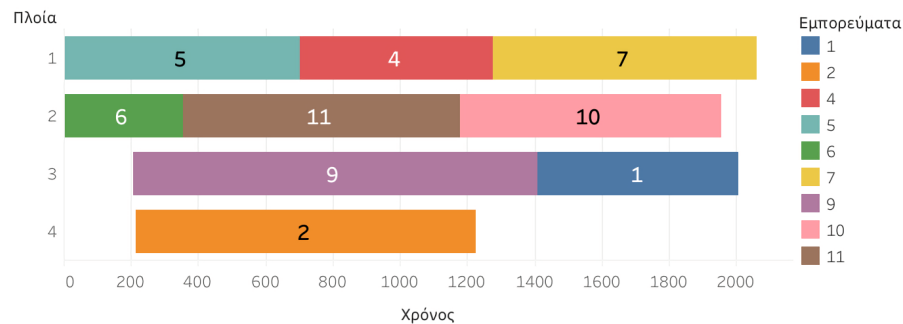
APPENDIX



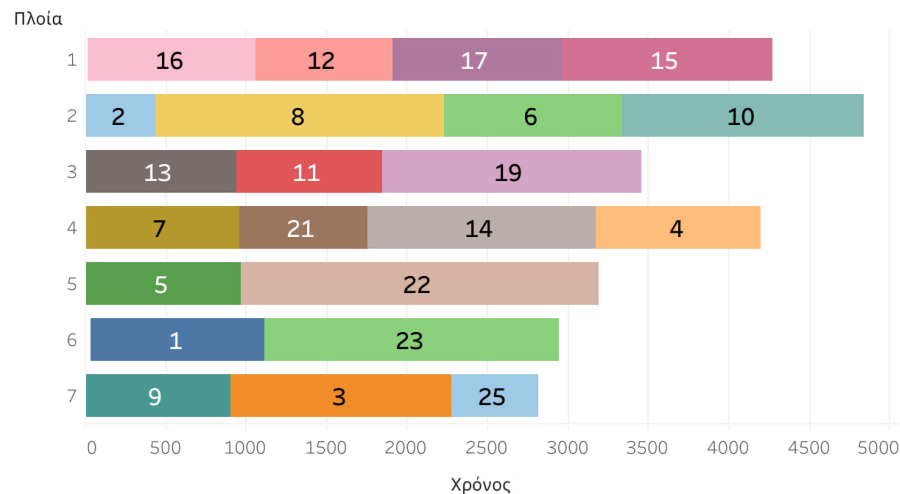
## APPENDIX

Υπενθυμίζεται πως στον Πίνακα Α'1 παρουσιάζονται τα αποτελέσματα της έρευνας που έγινε για να εξεταστεί η ικανότητα του ACPSO να ανακαλύπτει το καλύτερο αποτέλεσμα του. Με  $x$  συμβολίζεται η ελάχιστη απόκλιση από το ολικά βέλτιστο, με  $\bar{x}$  η μέση, με  $X$  η μέγιστη και με  $D = |x - \bar{x}|$  η απόλυτη απόκλιση της καλύτερης από την μέση. Οι έγχρωμες τιμές αντιστοιχούν στα αποτελέσματα που δεν κατάφεραν να ανακαλύψουν την καλύτερη λύση του ACPSO.

Στην συνέχεια εμφανίζεται ενδεικτικά μία λύση από κάθε κατηγορία.



Εικόνα Α'1: Παράδειγμα<sup>1</sup>για "Μικρά"



Εικόνα Α'2: Παράδειγμα<sup>2</sup>για "Μεσαία"

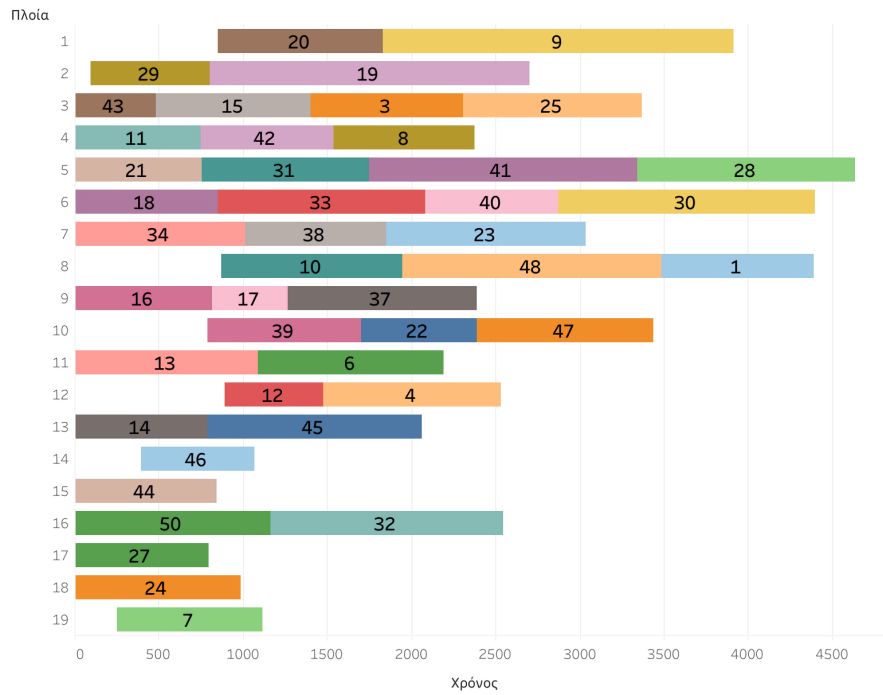
1 DS\_FUN\_C11\_V4\_HE\_1

2 DS\_FUN\_C25\_V7\_HE\_4

| ΑΡΧΕΙΟ               | $x$   | $\bar{x}$ | $X$   | $D$  |
|----------------------|-------|-----------|-------|------|
| DS_FUN_C8_V3_HE_1    | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C8_V3_HE_2    | 7,86  | 7,86      | 7,86  | 0,00 |
| DS_FUN_C8_V3_HE_3    | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C8_V3_HE_4    | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C8_V3_HE_5    | 21,23 | 21,23     | 21,23 | 0,00 |
| DS_FUN_C11_V4_HE_1   | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C11_V4_HE_2   | 1,20  | 1,20      | 1,20  | 0,00 |
| DS_FUN_C11_V4_HE_3   | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C11_V4_HE_4   | 0,13  | 0,13      | 0,13  | 0,00 |
| DS_FUN_C11_V4_HE_5   | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C13_V5_HE_1   | 0,00  | 0,12      | 0,40  | 0,12 |
| DS_FUN_C13_V5_HE_2   | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C13_V5_HE_3   | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C13_V5_HE_4   | 2,46  | 2,46      | 2,46  | 0,00 |
| DS_FUN_C13_V5_HE_5   | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C16_V6_HE_1   | 0,00  | 0,06      | 0,13  | 0,06 |
| DS_FUN_C16_V6_HE_2   | 0,00  | 0,01      | 0,01  | 0,01 |
| DS_FUN_C16_V6_HE_3   | 0,00  | 0,00      | 0,00  | 0,00 |
| DS_FUN_C16_V6_HE_4   | 0,88  | 0,91      | 0,95  | 0,03 |
| DS_FUN_C16_V6_HE_5   | 0,00  | 1,58      | 5,03  | 1,58 |
| DS_FUN_C17_V13_HE_1  | 0,00  | 0,51      | 1,53  | 0,51 |
| DS_FUN_C17_V13_HE_2  | 0,26  | 1,00      | 1,66  | 0,74 |
| DS_FUN_C17_V13_HE_3  | 0,31  | 1,12      | 2,17  | 0,81 |
| DS_FUN_C17_V13_HE_4  | 0,18  | 0,22      | 0,36  | 0,04 |
| DS_FUN_C17_V13_HE_5  | 0,00  | 0,17      | 0,87  | 0,17 |
| DS_FUN_C20_V6_HE_1   | 0,00  | 0,28      | 1,37  | 0,28 |
| DS_FUN_C20_V6_HE_2   | 2,00  | 2,64      | 4,24  | 0,64 |
| DS_FUN_C20_V6_HE_3   | 0,03  | 0,31      | 1,06  | 0,28 |
| DS_FUN_C20_V6_HE_4   | 0,23  | 0,88      | 1,29  | 0,65 |
| DS_FUN_C20_V6_HE_5   | 1,14  | 1,44      | 2,16  | 0,30 |
| DS_FUN_C25_V7_HE_1   | 0,12  | 0,55      | 1,83  | 0,44 |
| DS_FUN_C25_V7_HE_2   | 0,86  | 3,68      | 6,00  | 2,81 |
| DS_FUN_C25_V7_HE_3   | 0,00  | 1,68      | 3,66  | 1,68 |
| DS_FUN_C25_V7_HE_4   | 0,00  | 0,39      | 1,33  | 0,39 |
| DS_FUN_C25_V7_HE_5   | 3,84  | 4,13      | 4,81  | 0,29 |
| DS_FUN_C35_V13_HE_1  | 3,54  | 4,64      | 6,61  | 1,09 |
| DS_FUN_C35_V13_HE_2  | 2,97  | 6,66      | 12,96 | 3,69 |
| DS_FUN_C35_V13_HE_3  | 5,11  | 6,97      | 8,88  | 1,86 |
| DS_FUN_C35_V13_HE_4  | 5,36  | 6,18      | 7,63  | 0,82 |
| DS_FUN_C35_V13_HE_5  | 6,88  | 8,73      | 10,70 | 1,85 |
| DS_FUN_C50_V20_HE_1  | 1,64  | 2,37      | 4,80  | 0,73 |
| DS_FUN_C50_V20_HE_2  | 1,39  | 4,10      | 6,61  | 2,70 |
| DS_FUN_C50_V20_HE_3  | 1,33  | 2,06      | 3,25  | 0,73 |
| DS_FUN_C50_V20_HE_4  | 0,67  | 2,67      | 4,11  | 2,00 |
| DS_FUN_C50_V20_HE_5  | 1,20  | 2,76      | 5,08  | 1,56 |
| DS_FUN_C70_V30_HE_1  | 6,71  | 7,17      | 7,56  | 0,46 |
| DS_FUN_C70_V30_HE_2  | 7,00  | 8,16      | 10,20 | 1,16 |
| DS_FUN_C70_V30_HE_3  | 1,72  | 3,97      | 6,28  | 2,25 |
| DS_FUN_C70_V30_HE_4  | 5,32  | 6,25      | 7,23  | 0,94 |
| DS_FUN_C70_V30_HE_5  | 3,73  | 4,32      | 4,91  | 0,59 |
| DS_FUN_C90_V40_HE_1  | 3,20  | 4,16      | 5,48  | 0,96 |
| DS_FUN_C90_V40_HE_2  | 9,98  | 11,06     | 12,17 | 1,08 |
| DS_FUN_C90_V40_HE_3  | 6,94  | 8,22      | 9,27  | 1,28 |
| DS_FUN_C90_V40_HE_4  | 2,42  | 3,60      | 5,06  | 1,18 |
| DS_FUN_C90_V40_HE_5  | 6,86  | 7,64      | 8,02  | 0,78 |
| DS_FUN_C100_V50_HE_1 | 4,99  | 5,70      | 6,32  | 0,71 |
| DS_FUN_C100_V50_HE_2 | 7,08  | 7,96      | 8,43  | 0,88 |
| DS_FUN_C100_V50_HE_3 | 10,30 | 11,47     | 12,50 | 1,17 |
| DS_FUN_C100_V50_HE_4 | 6,10  | 6,79      | 8,60  | 0,68 |
| DS_FUN_C100_V50_HE_5 | 4,03  | 5,37      | 6,59  | 1,34 |
| DS_FUN_C110_V52_HE_1 | 1,98  | 2,95      | 4,47  | 0,97 |
| DS_FUN_C120_V53_HE_1 | 11,90 | 12,36     | 12,92 | 0,46 |
| DS_FUN_C130_V58_HE_1 | 5,21  | 6,01      | 6,84  | 0,80 |
| DS_FUN_C140_V62_HE_1 | 7,56  | 7,83      | 8,07  | 0,28 |
| DS_FUN_C150_V67_HE_1 | 3,42  | 4,18      | 4,70  | 0,76 |
| DS_FUN_C160_V71_HE_1 | 5,74  | 6,22      | 6,85  | 0,49 |
| DS_FUN_C170_V76_HE_1 | 4,55  | 4,98      | 5,50  | 0,43 |
| DS_FUN_C200_V89_HE_1 | 5,03  | 5,57      | 6,38  | 0,53 |

Πίνακας Α'1: Αποτελέσματα έρευνας για την αξιοπιστία του ACP SO



Εικόνα Α.3: Παράδειγμα<sup>3</sup>για "Μεγάλα"<sup>3</sup> DS\_FUN\_C50\_V20\_HE\_4<sup>4</sup> DS\_FUN\_C200\_V89\_HE\_1



## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

- [1] Ιωάννης Μαρινάκης and Αθανάσιος Μυγδαλάς. *Σχεδιασμός και βελτιστοποίηση της εφοδιαστικής αλυσίδας*. Σοφία Α.Ε., 2008.
- [2] Ιωάννης Μαρινάκης, Αθανάσιος Μυγδαλάς, and Μαγδαληνή Μαρινάκη. *Προβλήματα Δρομολόγησης Οχημάτων στη Διαχείριση της Εφοδιαστικής Αλυσίδας*. Εκδόσεις Νέων Τεχνολογιών, 2019.
- [3] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. 1st ed. Santa Fe Institute Studies on the Sciences of Complexity. Oxford University Press, USA, 1999.
- [4] Marielle Christiansen, Kjetil Fagerholt, Bjørn Nygreen, and David Ronen. “Chapter 4 Maritime Transportation”. In: ed. by Cynthia Barnhart and Gilbert Laporte. *Handbooks in Operations Research and Management Science*. Elsevier, 2007. doi: [https://doi.org/10.1016/S0927-0507\(06\)14004-9](https://doi.org/10.1016/S0927-0507(06)14004-9).
- [5] Marielle Christiansen, Kjetil Fagerholt, Bjørn Nygreen, and David Ronen. “Ship routing and scheduling in the new millennium”. In: *European Journal of Operational Research* (2013). doi: <https://doi.org/10.1016/j.ejor.2012.12.002>.
- [6] IBM Cloud Education. *What is artificial intelligence*. URL: <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>.
- [7] Kjetil Fagerholt. “A computer-based decision support system for vessel fleet scheduling—experience and future research”. In: *Decision Support Systems* 37.1 (2004), pp. 35–47.
- [8] P. Festa. “A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems”. In: *2014 16th International Conference on Transparent Optical Networks (ICTON)*. 2014, pp. 1–20. doi: [10.1109/ICTON.2014.6876285](https://doi.org/10.1109/ICTON.2014.6876285).
- [9] Birger Funke, Tore Grünert, and Stefan Irnich. “Local Search for Vehicle Routing and Scheduling Problems: Review and Conceptual Integration”. In: *Journal of Heuristics* (2005). doi: [10.1007/s10732-005-1997-2](https://doi.org/10.1007/s10732-005-1997-2).
- [10] Ahmad Hemmati, Lars Magnus Hvattum, Kjetil Fagerholt, and Inge Norstad. “Benchmark Suite for Industrial and Tramp Ship Routing and Scheduling Problems”. In: *INFOR: Information Systems and Operational Research* 52.1 (2014), pp. 28–38. doi: [10.3138/infor.52.1.28](https://doi.org/10.3138/infor.52.1.28).

- [11] Gabriel Homsí, Rafael Martinelli, Thibaut Vidal, and Kjetil Fagerholt. “Industrial and tramp ship routing problems: Closing the gap for real-scale instances”. In: *European Journal of Operational Research* 283.3 (2020), pp. 972–990. URL: <https://doi.org/10.1016/j.ejor.2019.11.068>.
- [12] Yannis Marinakis, Georgia-Roumbini Iordanidou, and Magdalene Marinaki. “Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands”. In: *Applied Soft Computing* (2013). doi: <https://doi.org/10.1016/j.asoc.2013.01.007>.
- [13] Yannis Marinakis and Magdalene Marinaki. “Combinatorial expanding neighborhood topology particle swarm optimization for the vehicle routing problem with stochastic demands”. In: *Proceedings of the 15th annual Conference on Genetic and Evolutionary Computation*. 2013, pp. 49–56.
- [14] Yannis Marinakis, Magdalene Marinaki, and Athanasios Migdalas. “A multi-adaptive particle swarm optimization for the vehicle routing problem with time windows”. In: *Information Sciences* (2019). doi: <https://doi.org/10.1016/j.ins.2018.12.086>.
- [15] Rafael Martí, Panos M. Pardalos, and Mauricio G. C. Resende. *Handbook of Heuristics*. 1st ed. Springer International Publishing, 2018.
- [16] John McCarthy. “What is Artificial Intelligence?” In: (2004).
- [17] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. 2nd ed. Vol. 1. Mathematical optimization. Prentice Hall, 1998.
- [18] George Polya. *Mathematical Discovery, On understanding, learning, and teaching problem solving*. 2nd ed. New York: John Wiley & Sons, 1962.
- [19] David L. Poole and Alan K. Mackworth. *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, 2010.
- [20] David Poole, Alan Mackworth, and Randy Goebel. *Computational Intelligence: A Logical Approach*. Oxford University Press, 1998.
- [21] Stefan Schroedl Stefan Edelkamp. *Heuristic Search: Theory and Applications*. 1st ed. Elsevier, 2011.
- [22] El-Ghazali Talbi. *Metaheuristics: from design to implementation*. New. Wiley Series on Parallel and Distributed Computing. Wiley, 2009.
- [23] UNCTAD. *Review of Maritime Transport*. 2013. URL: [https://unctad.org/en/PublicationsLibrary/rmt2013\\_en.pdf](https://unctad.org/en/PublicationsLibrary/rmt2013_en.pdf).
- [24] UNCTAD. *Review of Maritime Transport*. 2020. URL: [https://unctad.org/system/files/official-document/rmt2020\\_en.pdf](https://unctad.org/system/files/official-document/rmt2020_en.pdf).

- [25] Xin-She Yang. *Nature-Inspired Optimization Algorithms*. 1st ed. Elsevier Insights. Elsevier, 2014.
- [26] Stelios H. Zanakis and James R. Evans. “Heuristic “Optimization”: Why, When, and How to Use It”. In: *INFORMS Journal on Applied Analytics* 11.5 (1981), pp. 84–91.