



ΣΤΡΑΤΙΩΤΙΚΗ ΣΧΟΛΗ  
ΕΥΕΛΠΙΔΩΝ  
Τμήμα Στρατιωτικών Επιστημών

ΔΙΔΡΥΜΑΤΙΚΟ  
ΔΙΑΤΜΗΜΑΤΙΚΟ  
ΠΡΟΓΡΑΜΜΑ

ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
Σχολή Μηχανικών Παραγωγής &  
Διοίκησης

**«Εφαρμοσμένη Επιχειρησιακή Έρευνα και Ανάλυση»**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ**

**«Δρομολόγηση Οχημάτων Συλλογής Αστικών Αποβλήτων»**

**Κωνσταντίνος Γ. Μητρακόπουλος (ΑΜ:2015018045)**

Επιβλέπων Καθηγητής: **Μαρινάκης Ιωάννης**

**Βόλος, Ιούνιος 2019**



**ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ**

Μαρινάκης Ιωάννης (Επιβλέπων)  
Αναπληρωτής Καθηγητής

A handwritten signature in blue ink, appearing to read 'I. Marinakis', with a long horizontal stroke extending to the right.

Σαχαρίδης Γεώργιος  
Επίκουρος Καθηγητής

A handwritten signature in blue ink, appearing to read 'G. Sakaridis', with a stylized, looped structure.

Καραματσούκης Κωνσταντίνος  
Επίκουρος Καθηγητής

A handwritten signature in blue ink, appearing to read 'K. Karamatsoyakis', with a long horizontal stroke extending to the right.

## ΕΥΧΑΡΙΣΤΙΕΣ

Ολοκληρώνοντας την παρούσα μεταπτυχιακή διατριβή, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Ιωάννη Μαρινάκη για την πολύτιμη βοήθειά του, την καθοδήγησή του και τη συνεχή υποστήριξή του για την εκπόνηση αυτής της διατριβής.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου και τα κοντινά μου πρόσωπα, που με την αμέριστη υποστήριξη και εμπιστοσύνη που μου έδειξαν καθ' όλη τη διάρκεια της προσπάθειάς μου, συνέβαλλαν αποφασιστικά στην υλοποίηση και ολοκλήρωση αυτής της Διατριβής.

## ΠΕΡΙΛΗΨΗ

Η δρομολόγηση οχημάτων συλλογής αστικών αποβλήτων σε ένα αστικό δίκτυο διαχείρισης απορριμμάτων είναι ένα πρόβλημα που έχει απασχολήσει τα τελευταία χρόνια την επιστημονική κοινότητα, τόσο για τους πολίτες μεμονωμένα όσο και για την πολιτεία συνολικά.

Στην παρούσα διατριβή, προτείνουμε μια μεθοδολογία βελτιστοποίησης της συλλογής αστικών αποβλήτων σε μία αστική περιοχή που στόχος της είναι η μείωση του κόστους αποκομιδής για την πολιτεία με την καλύτερη δρομολόγηση των απορριμματοφόρων. Τα μοντέλα που προτείνουμε βασίζονται στον μικτό ακέραιο γραμμικό προγραμματισμό. Η εφαρμογή τους αναμένουμε να δώσει αρκετά καλά αποτελέσματα, τα οποία μπορούν να εφαρμοστούν στην πράξη δίνοντας μεγάλη ακρίβεια και βελτιώνοντας πολύ την αρχική κατάσταση.

## Περιεχόμενα

<b>ΚΕΦΑΛΑΙΟ 1: Διαχείριση Απορριμμάτων</b>	8
1.1 Εισαγωγή	8
1.2 “Πληρώνω όσο πετάω” (ΠΟΠ) – “Pay as you throw” (PAYT)	9
1.2.1 Περιγραφή μοντέλων PAYT	10
1.2.2 Εφαρμογή συστημάτων PAYT	11
<b>ΚΕΦΑΛΑΙΟ 2: Ανασκόπηση Μελετών</b>	13
2.1 Μεθοδολογίες για βελτιστοποίηση διαχείρισης αποβλήτων και εφαρμογές	13
2.1.1 Μεθοδολογία που βασίζεται σε ακριβείς μεθόδους επίλυσης των Badran M.F. και El-Hagaar S.M. (2006)	13
2.1.2 Μεθοδολογία που βασίζεται σε ευρετική μέθοδο των Ghiani G., Manni A., Manni E. και Toraldo M. (2014)	14
2.1.3 Μεθοδολογία με πρόταση μιας ευρετικής λύσης των Das και Bhattacharyya (2015)	14
2.1.4 Μεθοδολογία σε συνθήκες αβεβαιότητας των Li, Y., Huang G., Nie S. (2012)	15
2.2 Μεθοδολογίες εκτίμησης της ημερήσιας παραγωγής απορριμμάτων	16
2.2.1 Ανάλυση χρονικών σειρών των ποσοστών παραγωγής αστικών στερεών αποβλήτων των Katsamaki A., Willems S. και Diamadopoulos E. (1998)	16
2.2.2 Πρόβλεψη δημιουργίας στερεών αποβλήτων βασισμένο σε grey fuzzy δυναμικό μοντέλο των Chen H. και Chang Ni-Bin (2000)	17
<b>ΚΕΦΑΛΑΙΟ 3: Πρόβλημα του Πλανόδιου Πωλητή &amp; Πρόβλημα της Δρομολόγησης Οχημάτων</b>	18
3.1.1 Ορισμός του προβλήματος	18
3.1.2 Ιστορική αναδρομή του προβλήματος	18
3.1.3 Πολυπλοκότητα	19
3.1.4 Παραλλαγές του TSP	19
3.1.5 Εφαρμογές στην καθημερινότητα	20
3.2 Το Πρόβλημα της Δρομολόγησης Οχημάτων	23
3.2.1 Ορισμός του προβλήματος	23
3.2.2 Ιστορική αναδρομή του προβλήματος	24
3.2.3 Πολυπλοκότητα	25
3.2.4 Παραλλαγές του VRP	25
3.3 Ερευνητικές μέθοδοι επίλυσης	30
3.3.1 Τεχνικές ομαδοποίησης (Clustering Approach)	30
3.3.2 Αλγόριθμος k-means	31

3.3.3 Ευρετική μέθοδος επίλυσης προβλήματος πλανόδιου πωλητή .....	33
<b>ΚΕΦΑΛΑΙΟ 4: Μαθηματικό μοντέλο προβλήματος δρομολόγησης οχημάτων για την επίλυση προβλήματος αποκομιδής απορριμμάτων .....</b>	<b>39</b>
4.1 Προσέγγιση προβλήματος .....	39
4.2 Ονοματολογία .....	43
4.3 Μαθηματικό μοντέλο .....	45
<b>ΚΕΦΑΛΑΙΟ 5: Εφαρμογή και αποτελέσματα .....</b>	<b>48</b>
5.1 Δεδομένα προβλήματος .....	48
5.2 Αποτελέσματα βελτιστοποίησης .....	50
5.3 Σύγκριση λύσεων .....	52
<b>Βιβλιογραφία .....</b>	<b>53</b>
<b>Παράρτημα .....</b>	<b>58</b>

---

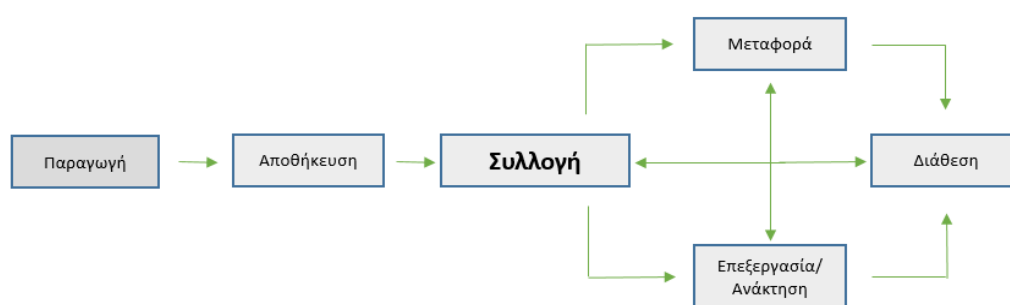
## ΚΕΦΑΛΑΙΟ 1: Διαχείριση Απορριμμάτων

### 1.1 Εισαγωγή

Με τον όρο απορρίμματα εννοούμε όλα τα στερεά απόβλητα, τα οποία δεν μπορούν φαινομενικά να αξιοποιηθούν. Σε αυτά περιλαμβάνονται μόνο τα στερεά απόβλητα διαφόρων προελεύσεων όπως, οικιακά, βιομηχανικά, σκουπίδια δρόμων κ.ά. Οι βασικές πηγές απορριμμάτων είναι η περίσσεια απόκτησης πρώτων υλών, ο μετασχηματισμός πρώτων υλών σε προϊόντα και τα υπολείμματα προϊόντων μετά από τη χρήση τους. Οι μέθοδοι διάθεσης απορριμμάτων είναι η υγειονομική ταφή, η θερμική επεξεργασία με καύση ή πυρόλυση, η αναερόβια χώνευση και ο διαχωρισμός των απορριμμάτων με σκοπό την ανάκτηση χρήσιμων υλικών.

Η διαχείριση απορριμμάτων είναι ένα βασικό στοιχείο της πολιτικής προστασίας του περιβάλλοντος και έχει σκοπό την προστασία της δημόσιας υγείας, συνεισφέροντας στην εξοικονόμηση ενέργειας και πρώτων υλών. Σκοπός της διαχείρισης των απορριμμάτων είναι:

- η συλλογή, εναπόθεση, επεξεργασία ή καταστροφή με τον ευνοϊκότερο τρόπο,
- η μείωση της παραγωγής τους,
- η ανάκτηση, επαναφορά και επαναχρησιμοποίηση διάφορων υλικών (Τερζής, 2009).



Εικόνα 1: Διαδικασία διαχείρισης απορριμμάτων

Η παραγωγή αποβλήτων αποτελεί ένα ζήτημα που απασχολεί ιδιαίτερες τις κοινωνίες, όχι μόνο για την ποσοτική αύξηση των παραγόμενων αποβλήτων, αλλά και λόγω της αυξανόμενης πολυπλοκότητας ορισμένων προϊόντων (Han & Ponce-Cueto, 2015).



Στην παρούσα εργασία θα επικεντρωθούμε στο στάδιο της συλλογής απορριμμάτων η οποία σύμφωνα με το *Glossary of enviromental statistics* των Ηνωμένων Εθνών (1997), περιλαμβάνει τη συγκέντρωση και τη μεταφορά αποβλήτων στον τόπο επεξεργασίας από δημοτικές υπηρεσίες ή παρόμοιους οργανισμούς ή εξειδικευμένες επιχειρήσεις. Η συλλογή αστικών αποβλήτων μπορεί να είναι επιλεκτική, δηλαδή να πραγματοποιείται για συγκεκριμένο τύπο προϊόντος, ή αδιαφοροποίητη, δηλαδή να καλύπτει ταυτόχρονα όλα τα είδη αποβλήτων. Η συλλογή αποβλήτων είναι το μέρος της διαδικασίας που περιλαμβάνει τη συγκέντρωση απορριμμάτων (στερεά απόβλητα και ανακυκλώσιμα υλικά) από τα σημεία συλλογής έως τη μονάδα ανακύκλωσης, ανάκτησης ενέργειας, ή προς το χώρο υγειονομικής ταφής. Τυπικά, το σύστημα συλλογής απορριμμάτων (συλλογή και μεταφορά) περιλαμβάνει ένα υψηλό λειτουργικό κόστος (Han & Ponce-Cueto, 2015).

Η διαχείριση στερεών αποβλήτων είναι μείζον ζήτημα στον αστικό χώρο και πολλές χώρες έχουν ήδη ξεκινήσει να κάνουν αξιόλογες προσπάθειες για να βελτιώσουν την κατάσταση σχετικά με την παραγωγή και διαχείριση αποβλήτων. Παράλληλα, η αυξανόμενη περιβαλλοντική πίεση καθώς και η ανάγκη για συμμόρφωση με το Ευρωπαϊκό νομικό πλαίσιο προώθησε την ανάγκη για βιωσιμότητα στον τομέα διαχείρισης αστικών αποβλήτων.

## 1.2 “Πληρώνω όσο πετάω” (ΠΟΠ) – “Pay as you throw” (PAYT)

Το “Pay as you throw” είναι ένα σύστημα διαχείρισης αποβλήτων, με το οποίο οι κάτοικοι χρεώνονται για τις υπηρεσίες διαχείρισης αποβλήτων με βάση τον παραγόμενο όγκο αποβλήτων τους. Σύμφωνα με αυτό, όσο λιγότερα απορρίμματα παράγονται από ένα νοικοκυριό, τόσο μικρότερη είναι η χρέωση, παρέχοντας ένα άμεσο κίνητρο για μείωση της παραγωγής. Οι στόχοι ενός συστήματος περιλαμβάνουν: α) Αύξηση των εσόδων για την υποστήριξη του προγράμματος, β) Ενθάρρυνση της μείωσης των αστικών στερεών αποβλήτων λόγω κόστους, γ) Βελτίωση της κατανόησης του κόστους διαχείρισης των στερεών αποβλήτων στους πολίτες και ευαισθητοποίησή τους σχετικά με θέματα περιβάλλοντος, δ) Ενθάρρυνση για ανακύκλωση, ε) Διατήρηση ενός προγράμματος εύκολου στη χρήση και εκτέλεση. (Tobias, 2010).

Το σύστημα PAYT βασίζεται στην αρχή ο “ρυπαίνων πληρώνει” και ακολουθεί μια μεθοδολογία παρόμοια με αυτή που ισχύει για την κοστολόγηση του νερού και του

ρεύματος. Δηλαδή, τα τέλη που πληρώνει κάθε νοικοκυριό συνδέονται με την ποσότητα των αποβλήτων που παράγει αυτό. Αυτή η λειτουργία επιτυγχάνεται τεχνικά, είτε με τη ζύγιση του κάδου, είτε με την αγορά από τους δημότες ειδικών σακούλων με προπληρωμένα τέλη, είτε με ατομικό κάδο σε κάθε νοικοκυριό και χρέωση με τον όγκο και τη συχνότητα αποκομιδής, είτε τέλος με προπληρωμένες κάρτες, με τις οποίες μπορεί κάποιος να τοποθετήσει τα απορρίμματά του σε press containers και να αφαιρούνται μονάδες. Αναγκαίο για την αποτελεσματική εφαρμογή του PAYT είναι η ύπαρξη υποδομών που υποστηρίζουν τη μείωση, επαναχρησιμοποίηση, ανακύκλωση και κομποστοποίηση των αποβλήτων, έτσι ώστε ο δημότης να ελαχιστοποιεί τις ποσότητες αποβλήτων προς τελική διάθεση και συνεπώς να μειώνει το κόστος διαχείρισης που τον επιβαρύνει. (Project LIFE07/ENV/GR/000271 HEC-PAYT)

Η εφαρμογή των συστημάτων PAYT στηρίζεται σε τρεις βασικούς πυλώνες:

1. Στην αναγνώριση του παραγωγού των απορριμμάτων. Ο καθένας είναι υπεύθυνος για τα δικά του απορρίμματα
2. Στη μέτρηση της παραγόμενης ποσότητας απορριμμάτων
3. Στη μοναδιαία χρέωση σύμφωνα με την παρεχόμενη υπηρεσία

Τα παραπάνω αποτελούν αναγκαίες προϋποθέσεις για την εφαρμογή ενός συστήματος PAYT, ωστόσο δεν επαρκούν για να εφαρμοστεί πλήρως ένα τέτοιο πρόγραμμα. Η ενσωμάτωσή τους σε ένα πλαίσιο πολιτικής, στο οποίο έχουν προτεραιότητα η ευθύνη των παραγωγών αποβλήτων, η περιβαλλοντική συνείδηση και η ανακύκλωση υλικών θα μπορούσε να επιτύχει μια αποτελεσματική εφαρμογή ενός τέτοιου συστήματος.

### 1.2.1 Περιγραφή μοντέλων PAYT

- Συστήματα με μαγνητική κάρτα

- Χρέωση με βάση τον όγκο. Ο κάδος διαθέτει μια συσκευή, η οποία επιτρέπει τη ρίψη ενός μέγιστου όγκου αποβλήτων σε κάθε χρήση και ο χρήστης αναγνωρίζεται χρησιμοποιώντας μαγνητική κάρτα.
- Πληρωμή κατά βάρος με ενσωματωμένο σύστημα ζύγισης στο δοχείο, το οποίο καταγράφει το βάρος των αποβλήτων που απορρίπτονται από κάθε χρήστη. Όπως και στο προηγούμενο μοντέλο, ο χρήστης αναγνωρίζεται από την μαγνητική κάρτα που διαθέτει.

#### - Συστήματα με αναγνώριση κάδου

- Χρέωση ανά κάδο με ατομική συμφωνία. Σε κάθε χρήστη ή ομάδα χρηστών δίνεται ένας κάδος γνωστού όγκου (ο όγκος του οποίου επιλέγεται από τους χρήστες) και ο οποίος αναγνωρίζεται με ένα τσιπ ή με μία ετικέτα. Τα απορριμματοφόρα αναγνωρίζουν τους κάδους και καταγράφουν το φόρο που αναλογεί σε κάθε χρήστη.
- Χρέωση ανά κάδο με προκαθορισμένη συχνότητα. Στην περίπτωση αυτή, οι διανεμημένοι κάδοι συλλέγονται με προκαθορισμένο πρόγραμμα. Οι χρήστες επιλέγουν τον όγκο των κάδων και τη συχνότητα συλλογής με βάση τις επιλογές που τους παρέχει ο δήμος. Αυτό το σύστημα μπορεί να εφαρμοστεί σε κάθε αστική περιοχή, αλλά είναι ευκολότερο να εφαρμοστεί σε αραιοκατοικημένες περιοχές ή σε περιοχές με μονοκατοικίες. Η χρέωση γίνεται με βάση το ογκομετρικό σχήμα του κάδου και τη συχνότητα αποκομιδής.
- Αναγνώριση και ζύγιση κάδου. Ο φόρος προσδιορίζεται με βάση το βάρος των αποβλήτων στο παραδοθέν δοχείο, το οποίο διαθέτει τσιπ ή ετικέτα. Οι κάδοι ζυγίζονται με μηχανισμούς που διαθέτουν τα απορριμματοφόρα.
- Χρέωση ανά σακούλα. Σε αυτήν την περίπτωση, ο χρήστης πληρώνει προκαταβολικά, αγοράζοντας τυποποιημένες σακούλες για διάθεση αποβλήτων. Κάθε νοικοκυριό είναι υποχρεωμένο να αγοράζει τις ειδικές σακούλες (συνήθως 75 έως 115 lt) και να τοποθετεί εκεί τα απορρίμματά του. Η τιμή της σακούλας περιλαμβάνει το κόστος παραγωγής της, καθώς και τα δημοτικά τέλη, συνεπώς όσο λιγότερες σακούλες αγοράζει ένα νοικοκυριό, τόσο λιγότερα δημοτικά τέλη πληρώνει. Η υπηρεσία συλλογής δέχεται μόνο το συγκεκριμένο είδος σακούλας, οι οποίες διανέμονται από το Δημοτικό Συμβούλιο ή συνεργάτες και διαφοροποιούνται ανάλογα με το είδος των αποβλήτων (Ventosa *et al.*, 2010, Κυρκίτσος 2011).

#### 1.2.2 Εφαρμογή συστημάτων PAYT

Τα πρώτα συστήματα “Πληρώνω όσο πετάω” τέθηκαν σε λειτουργία στις ΗΠΑ. Πιο διαδεδομένα έγιναν από τη δεκαετία του 70 και μετά, ιδιαίτερα στην Καλιφόρνια, στο Μίσιγκαν, στη Νέα Υόρκη και στην Ουάσιγκτον. Μέχρι το 2010 ήταν ήδη εγκατεστημένα συστήματα PAYT σε 7.000 πόλεις των ΗΠΑ. Η λειτουργία αυτών των συστημάτων είναι ιδιαίτερα αξιοσημείωτη και στο Βανκούβερ. Στη συνέχεια, τα συστήματα αυτά άρχισαν να εξαπλώνονται και σε Ευρωπαϊκές χώρες με ιδιαίτερη

εφαρμογή στην Ελβετία και στη βορειοανατολική Γερμανία. Η Αυστρία ήταν η πρώτη ευρωπαϊκή χώρα που εφάρμοσε ατομική χρέωση τελών για απόβλητα το 1945, ενώ η Δρέσδη της Γερμανίας ήταν η πρώτη ευρωπαϊκή πόλη που εφάρμοσε ένα ηλεκτρονικό σύστημα ταυτοποίησης και τιμολόγησης για τέλη αποβλήτων.

Στην Ελλάδα, εφαρμόστηκε για πρώτη φορά πιλοτικά το σύστημα “Πληρώνω όσο πετάω” στην Ελευσίνα, στα πλαίσια του προγράμματος LIFE “HEC PAYT”. Στόχος του προγράμματος ήταν η προώθηση συστημάτων PAYT στην Ελλάδα, στην Εσθονία και στην Κύπρο μέσω της εξέτασης και της επανεκτίμησης του κόστους και των μηχανισμών τιμολόγησης για τον προσδιορισμό ενός μοντέλου βέλτιστης πρακτικής. Αυτό πραγματοποιήθηκε μέσω μιας εμπεριστατωμένης ανάλυσης των υπαρχόντων συστημάτων και αξιολογώντας ένα σύστημα επίδειξης PAYT σε πιλοτική κλίμακα σε 1.500 νοικοκυριά και 69 επιχειρήσεις του δήμου Ελευσίνας, ο οποίος περικλείει 5.500 κατοίκους. Αγοράστηκε και εγκαταστάθηκε εξοπλισμός ζύγισης σε φορτηγά συλλογής απορριμμάτων και πραγματοποιήθηκε ανάλυση ευαισθησίας για τη σύγκριση των επιπτώσεων των διαφορετικών συστημάτων τιμολόγησης. Βάσει μιας εξαντλητικής απογραφής της παραγωγής αποβλήτων, η εφαρμογή αξιολογήθηκε, έγιναν οι κατάλληλες προσαρμογές και επιτεύχθηκαν τα εξής:

- Μείωση του 25,8% των αποβλήτων από το χώρο υγειονομικής ταφής
- Ανακύκλωση του 56% των απορριμμάτων συσκευασίας
- Ανακύκλωση 4,6 kg από απορρίμματα ηλεκτρικού και ηλεκτρονικού εξοπλισμού ανά συμμετέχον άτομο
- Κομποστοποίηση 17,1% οργανικών αποβλήτων (HEC PAYT The Development of Pay as You Throw Systems in Hellas, Estonia and Cyprus)

## ΚΕΦΑΛΑΙΟ 2: Ανασκόπηση Μελετών

---

### 2.1 Μεθοδολογίες για βελτιστοποίηση διαχείρισης αποβλήτων και εφαρμογές

#### 2.1.1 Μεθοδολογία που βασίζεται σε ακριβείς μεθόδους επίλυσης των Badran M.F. και El-Hagaar S.M. (2006)

Οι Bardan και Haggag στη μελέτη τους παρουσίασαν ένα μοντέλο διαχείρισης αστικών στερεών αποβλήτων για την Port Said της Αιγύπτου, το οποίο περιλαμβάνει την ιδέα των σταθμών συλλογής. Το προτεινόμενο μοντέλο βελτιστοποίησης ελαχιστοποιεί το κόστος διαχείρισης του συστήματος με τη βοήθεια μικτού ακέραιου προγραμματισμού. Η βέλτιστη τοποθέτηση των σταθμών συλλογής επιλέχθηκε ανάμεσα από τις πιθανές θέσεις, έτσι ώστε το κόστος του συστήματος να ελαχιστοποιείται.

Στο συγκεκριμένο μοντέλο, τα αστικά απόβλητα περιλαμβάνουν τα επικίνδυνα, τα κατασκευαστικά και τα απόβλητα από κατεδαφίσεις. Το προτεινόμενο μοντέλο διατυπώνεται λαμβάνοντας υπόψη ορισμένα βασικά σημεία ροής των αποβλήτων στην Αιγυπτιακή πόλη, στην οποία παρατηρείται βιομηχανική δραστηριότητα. Τα απόβλητα δε διαχωρίζονται στην πηγή (απόβλητα βιομηχανικά, κατεδάφισης και κατασκευής) και μεταφέρονται στον πλησιέστερο σταθμό συλλογής με ευθύνη του παραγωγού τους. Τα οχήματα συλλογής είναι φορτηγά μεταφοράς χωρητικότητας ενός τόνου με δυνατότητα ανύψωσης κάδων και μεταφέρουν τα απόβλητα από τους κάδους συλλογής στις χωματερές.

Στόχος του μοντέλου είναι να γίνεται επιλογή ανάμεσα στις εναλλακτικές τοποθεσίες των σταθμών συλλογής, έτσι ώστε να ελαχιστοποιείται το συνολικό κόστος του συστήματος διαχείρισης. Το συνολικό κόστος απαρτίζεται από δύο μέρη. Το ένα μέρος του κόστους περιλαμβάνει τα έξοδα μεταφοράς, δηλαδή το σταθερό και το μεταβλητό κόστος των εγκαταστάσεων και το δεύτερο μέρος περιλαμβάνει το κόστος του λειτουργικού συστήματος, το οποίο αποτελείται από το κόστος των τυποποιημένων κάδων, τον εξοπλισμό του προσωπικού συλλογής και τα έξοδα διαχείρισης. Από το προτεινόμενο μοντέλο μικρού ακέραιου προγραμματισμού ελαχιστοποιείται το κόστος που εξαρτάται από τις εγκαταστάσεις, ενώ το κόστος της λειτουργίας προστίθεται στο προηγούμενο ελαχιστοποιημένο κόστος και τελικά προκύπτει το συνολικό ελαχιστοποιημένο κόστος του συστήματος. (Bardan & El-Hagaar, 2006)

### 2.1.2 Μεθοδολογία που βασίζεται σε ευρετική μέθοδο των Ghiani G., Manni A., Manni E. και Toraldo M. (2014)

Οι Ghiani *et al.* μελέτησαν δύο προβλήματα λήψης αποφάσεων που προκύπτουν κατά το σχεδιασμό της συλλογής στερεών αποβλήτων, δηλαδή την τοποθεσία των τόπων συλλογής και τη χωροθέτηση της υπηρεσίας. Προτείνουν μια ακριβή και μια ευρετική προσέγγιση για την τοποθέτηση των κάδων σε μια κατοικημένη περιοχή, καθώς και την απόφαση για τα χαρακτηριστικά των κάδων που θα τοποθετηθούν στα σημεία συλλογής. Στη μεθοδολογία αυτή, συνεκτιμάται η συμβατότητα των διαφορετικών τύπων κάδων και προτείνεται μια γρήγορη και αποτελεσματική ευρετική προσέγγιση για τον εντοπισμό ομοιογενών ζωνών που μπορούν να εξυπηρετηθούν από μόνο ένα όχημα συλλογής. Τα υπολογιστικά αποτελέσματα που βασίστηκαν σε δεδομένα πραγματικής ζωής τα οποία σχετίζονται με το πρόβλημα, έδειξαν την αποτελεσματικότητα των προτεινόμενων διαδικασιών, με αποτέλεσμα να απαιτείται ένα απορριμματοφόρο λιγότερο για την συλλογή και η συνολική διανυθείσα απόσταση να μειώνεται κατά 25% κατά μέσο όρο. Οι μειώσεις αυτές συντελούν τόσο σε νομισματική εξοικονόμηση όσο και σε μείωση περιβαλλοντικού αντικτύπου. (Ghiani *et al.*, 2014)

### 2.1.3 Μεθοδολογία με πρόταση μιας ευρετικής λύσης των Das και Bhattacharyya (2015)

Οι Swapan Das και Bidyut Kr. Bhattacharyya, στη μελέτη τους διαμόρφωσαν το πρόβλημα συλλογής και μεταφοράς στερεών αποβλήτων σε ένα μικτό ακέραιο προγραμματισμό και πρότειναν μια ευρετική λύση για το πρόβλημα αυτό, η οποία θα μπορούσε να αποτελέσει το βέλτιστο τρόπο για τη συλλογή των αποβλήτων και τη μεταφορά. Το προτεινόμενο μοντέλο χωρίζει το ολοκληρωμένο σύστημα διαχείρισης σε τέσσερα τμήματα:

- i. Συλλογή των διαχωριζόμενων στερεών αποβλήτων από διάφορες πηγές (νοικοκυριά, αγορές, γραφεία) και μεταφορά των αποβλήτων στο πλησιέστερο κέντρο διαλογής.
- ii. Σωρευτική μεταφορά στερεών αποβλήτων από το κέντρο συλλογής στον παρακείμενο σταθμό μεταφοράς.
- iii. Μεταφορά των διαχωρισμένων στερεών αποβλήτων από το σταθμό μεταφοράς προς τα κοντινά κατηγοριοποιούμενα εργοστάσια επεξεργασίας αποβλήτων.

- iv. Μεταφορά των παραγόμενων αποβλήτων από το εργοστάσιο μεταποίησης στον πλησιέστερο χώρο υγειονομικής ταφής.

Κάθε τμήμα περιέχει μεγάλο κόστος συλλογής και μεταφοράς αποβλήτων και το προτεινόμενο σύστημα υπολογίζει τη βέλτιστη συλλογή και μεταφορά αποβλήτων σε κάθε τμήμα του συστήματος. Το προτεινόμενο σύστημα εφαρμόστηκε στην Kolkata, μια μητροπολιτική πόλη της Ινδίας και τα υπολογιστικά αποτελέσματα έδειξαν ότι το προτεινόμενο σύστημα είναι σε θέση να βελτιώσει κατά 30% το συνολικό κόστος. (Swapan & Bhattacharyya, 2015)

#### 2.1.4 Μεθοδολογία σε συνθήκες αβεβαιότητας των Li, Y., Huang G., Nie S. (2012)

Σε αυτή τη μελέτη, αναπτύσσεται ένα μοντέλο ασαφούς - στοχαστικού τετραγωνικού προγραμματισμού, βασισμένο σε ένα σενάριο για τον εντοπισμό της βέλτιστης πολιτικής διαχείρισης των αστικών κέντρων ώστε να επιτρέπεται οι διπλές αβεβαιότητες που παρουσιάζονται ως κατανομές πιθανοτήτων και τα ασαφή σύνολα να ενσωματώνονται μέσα στην διαδικασία της βελτιστοποίησης. Μπορεί επίσης να χρησιμοποιηθεί για την ανάλυση διαφόρων σεναρίων που συνδέονται με διαφορετικά επίπεδα οικονομικών κυρώσεων όταν παραβιάζονται οι στόχοι πολιτικής που έχουν οριστεί εκ των προτέρων. Αυτή η εξελιγμένη μέθοδος εφαρμόστηκε σε μια μελέτη περιπτώσεων μακροχρόνιας διαχείρισης αστικών αποβλήτων και προγραμματισμού.

Στη μελέτη αυτή, οι αβεβαιότητες μπορούν να παρουσιάζονται είτε ως membership function είτε ως κατανομές πιθανοτήτων. Για παράδειγμα, ο ρυθμός παραγωγής αποβλήτων μπορεί να είναι υψηλής αβεβαιότητας, δεδομένου ότι επηρεάζεται από έναν αριθμό παραγόντων, όπως η οικονομική ανάπτυξη, η ανάπτυξη πληθυσμού και ανθρώπινων δραστηριοτήτων. Οι αβεβαιότητες στο ρυθμό παραγωγής αποβλήτων μπορεί να περιέχουν όχι μόνο τυχαίο χαρακτήρα με κατανομές πιθανοτήτων, αλλά και ασάφειες σε μεμονωμένα γεγονότα με ποικίλα επίπεδα πιθανοτήτων. Η μέθοδος αυτή έχει τα πλεονεκτήματα που αντικατοπτρίζουν την πολυπλοκότητα των αβεβαιοτήτων και τα αποτελέσματα που ελήφθησαν μπορούν να βοηθήσουν στον εντοπισμό των επιθυμητών σχεδίων κατανομής ροών αποβλήτων για να επιτευχθεί συμβιβασμός μεταξύ του κόστους, του βαθμού ικανοποίησης και του κινδύνου παράβασης. Τα αποτελέσματα από τη μελέτη αυτή, υποδηλώνουν ότι η εξελιγμένη αυτή μέθοδος μπορεί να εφαρμοστεί και σε άλλα προβλήματα σχεδιασμού περιβάλλοντος και πόρων που σχετίζονται με δυναμικές πολυπλοκότητες πολλών



σταδίων καθώς και αβεβαιότητες σε πολλαπλές μορφές, όπως η διαχείριση ποιότητας του αέρα και ο σχεδιασμός υδάτινων πόρων. (Li *et al.*, 2012)

## 2.2 Μεθοδολογίες εκτίμησης της ημερήσιας παραγωγής απορριμμάτων

Η ανάπτυξη ενός σχεδίου διαχείρισης αστικών στερεών αποβλήτων αποτελεί σύνθετη διαδικασία και η ποσοτικοποίηση και η πρόβλεψη της παραγωγής στερεών αποβλήτων είναι πολύ σημαντικές. Ο επιτυχημένος σχεδιασμός ενός συστήματος διαχείρισης στερεών αποβλήτων εξαρτάται σημαντικά από την ακρίβεια πρόβλεψης της παραγωγής στερεών αποβλήτων και για το λόγο αυτό, πολλοί ερευνητές έχουν ασχοληθεί με την πρόβλεψη της παραγωγής αποβλήτων χρησιμοποιώντας διάφορες μεθόδους μοντελοποίησης. Τα ήδη δημοσιευμένα μοντέλα είναι ποικίλης φύσεως και μπορούν να εφαρμοστούν είτε σε ολόκληρη τη χώρα, είτε στα επιμέρους νοικοκυριά. Στη συνέχεια, παρουσιάζονται δυο μελέτες για την πρόβλεψη παραγωγής στερεών αποβλήτων, η πρώτη εκ των οποίων χρησιμοποιεί μια σειρά γνωστών δεδομένων για την παραγωγή αποβλήτων και η δεύτερη εφαρμόζεται σε περιπτώσεις περιορισμένων δεδομένων.

### 2.2.1 Ανάλυση χρονικών σειρών των ποσοστών παραγωγής αστικών στερεών αποβλήτων των Katsamaki A., Willems S. και Diamadopoulos E. (1998)

Στη μελέτη των Katsamaki *et al.* παρέχεται μια ανάλυση δεδομένων και γίνεται στοχαστική μοντελοποίηση των ημερήσιων ποσοστών παραγωγής αστικών στερεών αποβλήτων. Τα σύνολα δεδομένων που εξετάστηκαν είναι οι ημερήσιες ποσότητες αστικών στερεών αποβλήτων για διαδοχικές ημέρες και για κάθε ημέρα χωριστά. Κάθε ακολουθία παρατηρήσεων μοντελοποιήθηκε από τα στοχαστικά μοντέλα Box-Jenkins ως συνάρτηση αυτοπαλινδρόμησης, κινούμενου μέσου όρου και των εποχιακών τάσεων. Για τις συνολικές χρονολογικές σειρές βρέθηκε ένα επαρκές μοντέλο ARMA και για τις ξεχωριστές ημερήσιες σειρές, τα απλά μοντέλα αυτοπαλινδρόμησης ήταν επαρκή χωρίς να συμπεριληφθούν οι εποχικές τάσεις.

Σε γενικές γραμμές, τα μοντέλα αυτά έδειξαν πως είναι στατιστικά εφαρμόσιμα και η μοντελοποίηση των τάσεων ήταν ικανοποιητική. Η ικανότητα πρόβλεψης των μοντέλων Box-Jenkins συγκρίθηκε με απλούστερα στατιστικά στοιχεία, όπως η μέση τιμή και οι κινητές μέσες τιμές. Ανάλογα με τη συγκεκριμένη ημέρα, τα διαφορετικά μοντέλα έδωσαν τα βέλτιστα αποτελέσματα πρόβλεψης. (Katsamaki *et al.*, 1998)



### 2.2.2 Πρόβλεψη δημιουργίας στερεών αποβλήτων βασισμένο σε grey fuzzy δυναμικό μοντέλο των Chen H. και Chang Ni-Bin (2000)

Οι Chen H. και Chang Ni-Bin στη μελέτη τους παρουσιάζουν ένα δυναμικό μοντέλο γκρι ασταθών συστημάτων (grey fuzzy dynamic modeling) για την πρόβλεψη παραγωγής στερεών αποβλήτων σε μια αστική περιοχή, βασιζόμενοι σε ένα σύνολο περιορισμένων δειγμάτων. Η θεωρία των γκρι συστημάτων παρουσιάστηκε πρώτη φορά από τον Deng JU (1982), ο οποίος περιγράφει τυχαίες μεταβλητές ως ένα διάστημα μεταβλητών αριθμών που ποικίλει ανάλογα με τους παράγοντες χρόνου και χρησιμοποιεί “χρώμα” το οποίο αντιπροσωπεύει το βαθμό αβεβαιότητας σε ένα δυναμικό σύστημα.

Στη θεωρία των γκρι συστημάτων, όλα τα συστήματα χωρίζονται σε τρεις κατηγορίες: λευκό, γκρι και μαύρο σύστημα. Το λευκό σύστημα έχει συγκεκριμένα και σαφή μηνύματα (όπως αρχεία παρακολούθησης σε χώρους υγειονομικής ταφής ή κατά τη συλλογή), ενώ τα μαύρα έχουν άγνωστα μηνύματα. Ένα γκριζο σύστημα έχει εν μέρει γνωστά και εν μέρει άγνωστα μηνύματα. Η κεντρική ιδέα είναι να αξιοποιηθεί η τεχνική παλινδρόμησης των ασαφών στόχων για βελτίωση του συμβατικού γκρι δυναμικού μοντέλου, ώστε να ελαχιστοποιείται η διαφορά μεταξύ των προβλεπόμενων τιμών και των παρατηρούμενων τιμών.

Στη μελέτη αυτή, επιλέχθηκαν διαφορετικές διαστάσεις για ένα γκρι ασταθές μοντέλο και πραγματοποιήθηκε ανάλυση ευαισθησίας, η οποία προβλέπει ένα σετ καμπυλών ως ένα επιτρεπόμενο εύρος για την πρόβλεψη της παραγωγής αποβλήτων. Με την ανάλυση ευαισθησίας, πραγματοποιείται παράλληλα και σύγκριση μεταξύ του συμβατικού γκρι μοντέλου και του νέου γκρι ασταθούς μοντέλου. Το μοντέλο αυτό εφαρμόστηκε πρακτικά στην πόλη Ταϊνάν στην Ταϊβάν και τα γενικά αποτελέσματα έδειξαν ότι ο αριθμός των διαφορετικών διαστάσεων σε μια γκρι διαφορική εξίσωση αποτελεί σημαντικό παράγοντα, ειδικά για μοντέλα μεγάλων διαστάσεων. (Cheng & Chang, 2000)

## ΚΕΦΑΛΑΙΟ 3: Πρόβλημα του Πλανόδιου Πωλητή & Πρόβλημα της Δρομολόγησης Οχημάτων

---

### 3.1 Το Πρόβλημα του Πλανόδιου Πωλητή

#### 3.1.1 Ορισμός του προβλήματος

Το Πρόβλημα του Πλανόδιου Πωλητή (Travelling Salesman Problem, στο εξής αναφερόμενο ως TSP), αποτελεί αντικείμενο ευρείας μελέτης στις επιστήμες της Πληροφορικής και του Γραμμικού Προγραμματισμού.

Ορίζεται από ένα πλήρες γράφημα ( $G$ ), ένα σύνολο κορυφών ( $V$ ), ένα σύνολο ακμών ( $E$ ) και ένα κόστος  $c_{ij}$ , το οποίο σχετίζεται με κάθε ακμή. Η τιμή που λαμβάνει το  $c_{ij}$  είναι το κόστος που δημιουργείται κατά τη διαδρομή από την κορυφή  $i \in V$  στην κορυφή  $j \in V$ . Με δεδομένη αυτή την πληροφορία, η λύση του TSP θα πρέπει να καταλήγει στη Χαμιλτονιανή λύση (Hamiltonian cycle) του γραφήματος  $G$  με το ελάχιστο κόστος [Η Χαμιλτονιανή λύση είναι μία συνάρτηση, βάσει της οποίας σε ένα γράφημα, κάθε κόμβος επισκέπτεται ακριβώς μία φορά (Skiena, 1990). Σε όρους TSP αυτό αναφέρεται ως «γύρος»].

Το πρόβλημα του πλανόδιου πωλητή συναντάται στην καθημερινότητα έχοντας πολλές πρακτικές εφαρμογές. Για παράδειγμα, ο διανομέας της αλληλογραφίας που προσπαθεί να βρει τη βέλτιστη διαδρομή ούτως ώστε να καλύψει όλες τις απαιτούμενες στάσεις ή ο υπεύθυνος αρχιτεκτονικής δικτύου, ο οποίος προσπαθεί να σχεδιάσει το αποτελεσματικότερο και βέλτιστο τρόπο για να συνδέσει εκατοντάδες ηλεκτρονικούς υπολογιστές. Σε όλες τις περιπτώσεις, η απόσταση ή το κόστος μεταξύ κάθε τοποθεσίας, είτε είναι μία πόλη, ένα κτίριο ή ένα δίκτυο, είναι γνωστά. Δεδομένης αυτής της πληροφορίας, ο βασικός στόχος και το ζητούμενο είναι η εύρεση της βέλτιστης λύσης, προκειμένου να προσδιοριστεί η σειρά με την οποία κάθε τοποθεσία θα πρέπει να επισκεφθεί ακριβώς μία φορά και η συνολική απόσταση που θα διανυθεί, ή το κόστος που θα προκύψει, να είναι τα ελάχιστα. Στη γενική μορφή του TSP, δεν υπάρχουν περιορισμοί στις τιμές της απόστασης ή του κόστους (Zambito, 2006).

#### 3.1.2 Ιστορική αναδρομή του προβλήματος

Στις αρχές του 1800, ο Ιρλανδός μαθηματικός Sir William Rowan Hamilton και ο Βρετανός μαθηματικός Thomas Penyngton Kirkman ασχολήθηκαν με τα πρώτα

μαθηματικά προβλήματα που σχετίζονταν με το TSP. Ωστόσο, το TSP στη γενική του μορφή μελετήθηκε το 1920, όταν ο μαθηματικός Karl Menger το παρουσίασε στους συναδέλφους του στη Βιέννη· το 1930 η μαθηματική κοινότητα του Princeton ασχολήθηκε με το πρόβλημα και το 1940 ο μαθηματικός Merrill Meeks Flood δημοσιοποίησε το όνομα “TSP” στη μαθηματική κοινότητα (Huixian et al., 2001). Πιο συγκεκριμένα, το 1948 ο Merrill Meeks Flood παρουσίασε το Πρόβλημα του Πλανόδιου Πωλητή στην εταιρεία RAND (η οποία είναι μη κερδοσκοπικός οργανισμός και αποτελεί το κέντρο της έρευνας και ανάπτυξης στις Ηνωμένες Πολιτείες), οδηγώντας έτσι στη δημοσιοποίησή του. Έτσι, σύντομα, το TSP έγινε πολύ δημοφιλές λόγω της σχέσης του με τα αναδυόμενα συνδυαστικά προβλήματα του Γραμμικού Προγραμματισμού και της εφαρμογής του σε πολλά προβλήματα της καθημερινής ζωής.

### 3.1.3 Πολυπλοκότητα

Στα μέσα του 1960, έγινε εμφανές ότι σε γενικές περιπτώσεις, το TSP δεν θα μπορούσε να επιλυθεί σε πολυωνυμικό χρόνο με τεχνικές Γραμμικού Προγραμματισμού. Στην πραγματικότητα, αποδείχθηκε ότι το TSP είχε τόσο μεγάλη υπολογιστική πολυπλοκότητα που όσες προσπάθειες και αν γίνονταν προκειμένου να λυθεί, θα δημιουργούσαν ακόμη μεγαλύτερα προβλήματα, ανάλογα με το μέγεθος του προβλήματος.

Έτσι, λόγω της πολυπλοκότητάς του, αναπτύχθηκαν αλγόριθμοι αποτελεσματικής προσέγγισης, οι οποίοι μπορούν να φανούν πολύ χρήσιμοι στην πράξη [για παράδειγμα ο αλγόριθμος “διακλάδωσης και οριοθέτησης” – “branch and bound” (Mataija *et al.*, 2016), η “Χαλάρωση Lagrange” – “Lagrange Relaxation”, καθώς και ποικίλες άλλες αριθμητικές μέθοδοι]. Από τον 19ο αιώνα και έπειτα, προκειμένου να χρησιμοποιηθεί η επίλυση του TSP σε προβλήματα της καθημερινότητας, αναπτύχθηκαν ακόμη περισσότερες χρήσιμες παραλλαγές, στις οποίες κατά περίπτωση, επιβάλλονται συγκεκριμένοι περιορισμοί.

### 3.1.4 Παραλλαγές του TSP

Οι διάφοροι περιορισμοί που τίθενται στη γενική μορφή του TSP (είτε για να διευκολυνθεί η λύση του, είτε για να αντικατοπτριστούν καλύτερα οι καθημερινές πρακτικές εφαρμογές του), οδηγούν σε παραλλαγές του (Monnot J, Toulouse S., 2014), οι δημοφιλέστερες εκ των οποίων είναι οι εξής (Gutin G, Punnen A.P., 2007):

- Το Συμμετρικό TSP (Symmetric TSP): Σε αυτή τη μορφή, όλες οι ακμές έχουν συμμετρικά κόστη. Αυτό σημαίνει ότι, για όλους τους κόμβους στο γράφημα, το κόστος της διαδρομής από τον κόμβο  $a$  στον κόμβο  $b$  ισούται με το κόστος της αντίθετης διαδρομής, δηλαδή από τον κόμβο  $b$  στον κόμβο  $a$ . Σε αντίθεση με το συμμετρικό TSP, το ασύμμετρο TSP δεν θέτει τέτοιους περιορισμούς και για αυτό το λόγο μια εισροή στο ασύμμετρο TSP θα ήταν ένα κατευθυνόμενο γράφημα.
- Το Μετρικό TSP (Metric TSP): Σε αυτή τη μορφή, όλα τα κόστη των ακμών είναι συμμετρικά και ικανοποιούν την τριγωνική ανισότητα. Η τριγωνική ανισότητα σημαίνει ότι για οποιονδήποτε από τρεις κόμβους  $a$ ,  $b$  και  $c$ , το κόστος της απευθείας διαδρομής από τον κόμβο  $a$  στον κόμβο  $c$  είναι πάντα μικρότερο από το κόστος της διαδρομής από τον κόμβο  $a$  στο κόμβο  $b$  και από τον κόμβο  $b$  στον κόμβο  $c$ . Επιπλέον, οι κόμβοι θεωρούνται σημεία σε κάποιον χώρο και το κόστος των ακμών προσδιορίζεται υπολογίζοντας τη μετρική απόσταση μεταξύ τους.
- Το Ευκλείδειο TSP (Euclidean TSP): Σε αυτή τη μορφή, όλοι οι κόμβοι βρίσκονται μέσα σε ένα σχήδιο, που σημαίνει ότι το TSP είναι συμμετρικό και ταυτόχρονα υπάρχει τριγωνική ανισότητα. Το κόστος κάθε ακμής  $e$  που συνδέει τους κόμβους  $a$  και  $b$ , ορίζεται από την Ευκλείδεια απόσταση μεταξύ τους. Σε γενικές γραμμές, το σχήδιο μπορεί να είναι  $d$ -διάστατο, όπου  $d > 1$  (Gutin G., Yeo A., Zverovich A., 2002)

### 3.1.5 Εφαρμογές στην καθημερινότητα

Η χρήση του TSP στην καθημερινότητα ποικίλει και κυρίως αφορά σε θέματα δρομολόγησης μεταφορών.

Η πιο ευρέως διαδεδομένη εφαρμογή αφορά στην ελάχιστη διαδρομή που θα πρέπει να ακολουθήσει ο πλανόδιος πωλητής, προκειμένου να επισκεφθεί κάθε γεωγραφικό σημείο μιας δεδομένης λίστας. Χρησιμοποιώντας ως παράδειγμα έναν πλανόδιο πωλητή που επισκέπτεται τα διάφορα σημεία από πόρτα σε πόρτα σε ένα υποσύνολο κατοικιών, θα ήταν πολύ χρήσιμο να καταρτιστεί μία λίστα με όλα τα σπίτια του υποσυνόλου που θα πρέπει να επισκεφθεί, χρησιμοποιώντας τη βέλτιστη διαδρομή. Επιπλέον, λαμβάνοντας υπόψη ότι ένας πλανόδιος πωλητής πρέπει να επισκεφθεί εκατοντάδες πόλεις που βρίσκονται διασκορπισμένες σε μία χώρα, γνωρίζοντας τη βέλτιστη διαδρομή, θα μπορούσε να εξοικονομήσει πολύτιμο χρόνο

(ημέρες, ακόμη και εβδομάδες) μετακινήσεων. Στην περίπτωση ενός διανομέα αλληλογραφίας, ο οποίος πηγαίνει στη δουλειά του το πρωί και έχει καθήκον να παραδώσει μεγάλο αριθμό δεμάτων, φακέλων κτλ, θα ήταν πολύ χρήσιμο να γνωρίζει με ποια σειρά θα πρέπει να κάνει τις παραδόσεις του, προκειμένου να διανύσει τη μικρότερη δυνατή απόσταση. Σε όλες αυτές τις περιπτώσεις, οι κόμβοι στα γραφήματα θα αντιστοιχούν στις γεωγραφικές τοποθεσίες και οι αποστάσεις μεταξύ τους θα είναι οι μετρικές αξίες που θα βασίζονται στο μήκος των δρόμων ή οδών που συνδέουν τις τοποθεσίες μεταξύ τους.

Οι παραπάνω εφαρμογές του TSP είναι ενδεικτικές, καθώς υπάρχουν και άλλες περιπτώσεις πρακτικής χρήσης του. Για παράδειγμα, στην περίπτωση μηχανημάτων (τρυπανιών) που βρίσκονται σε μία γραμμή παραγωγής και συναρμολόγησης με σκοπό τη δημιουργία οπών σε ένα συγκεκριμένο κομμάτι υλικού (π.χ. μία ηλεκτρονική πλακέτα, το πλαίσιο ενός οχήματος, ή ακόμα και ένα κομμάτι ξύλου το οποίο θα χρησιμοποιηθεί για την κατασκευή ενός ραφιού). Το τρυπάνι επανατοποθετείται με τη χρήση κινητήρων, οι οποίοι ολισθαίνουν κατά μήκος των διαδρομών ούτως ώστε να μπορεί να μετακινηθεί σε οποιαδήποτε θέση εντός συγκεκριμένης περιοχής. Θα χρειαζόταν αρκετός χρόνος για την επανατοποθέτηση του τρυπανιού, ανάλογα με την απόσταση την οποία θα πρέπει να διανύσει το τρυπάνι από το ένα σημείο στο άλλο. Μία λύση του TSP θα μπορούσε να βοηθήσει στην εύρεση της βέλτιστης σειράς με την οποία θα γίνουν οι οπές, έτσι ώστε να υπάρξει μεγαλύτερη παραγωγή μέσω της εξοικονόμησης χρόνου που θα επιτευχθεί στη γραμμή παραγωγής και συναρμολόγησης. Προκειμένου να επιλυθεί αυτό το πρόβλημα χρησιμοποιώντας το TSP, οι κορυφές θα είναι οι τοποθεσίες στις οποίες θα πρέπει να γίνουν οι οπές και οι ακμές θα είναι η απόσταση μεταξύ τους.

Μία άλλη εφαρμογή του TSP αφορά σε ηλεκτρονική ή μηχανική τοποθέτηση σύνδεσης. Για παράδειγμα, ο σχεδιασμός μίας ηλεκτρονικής πλακέτας, ή ενός κυκλώματος που αφορά είτε το δίκτυο ηλεκτρικών καλωδίων σε ένα κτίριο, είτε το δίκτυο υδροδότησης και αποχέτευσης, απαιτεί όλες οι συνδέσεις να γίνονται με τέτοιο τρόπο ούτως ώστε όλα τα στοιχεία του (καλώδια, σωλήνες κτλ) να συνδέονται σε έναν κύκλο. Στην περίπτωση της ηλεκτρονικής πλακέτας, τα καλώδια είναι οι συνδέσεις και τα τρανζίστορ και οι αντιστάσεις είναι τα στοιχεία. Στην περίπτωση του δικτύου ηλεκτρικών καλωδίων ενός κτιρίου, τα καλώδια είναι οι συνδέσεις και οι διακόπτες, τα

βύσματα, τα φωτιστικά κτλ είναι τα στοιχεία. Τέλος, στην περίπτωση της υδραυλικής διάταξης ενός κτιρίου, οι σωλήνες είναι οι συνδέσεις και οι βρύσες είναι τα στοιχεία.

Σε όλες τις παραπάνω περιπτώσεις, σκοπός είναι να βρεθεί η μικρότερη Χαμιλτονιανή διαδρομή προκειμένου να γίνει εξοικονόμηση υλικών και βελτιστοποίηση της ροής μέσω της μείωσης του μήκους του κύκλου. Κατά το σχεδιασμό της σύνδεσης κυκλωμάτων και της καλωδίωσης ηλεκτρονικών εξαρτημάτων θα πρέπει ο προγραμματισμός να είναι τέτοιος, ούτως ώστε να μεγιστοποιηθεί η αποτελεσματικότητα και η συνολική απόδοση μέσω της μικρότερης δυνατής διαδρομής που πρέπει να διανυθεί.

Η τελευταία εφαρμογή η οποία θα αναλυθεί είναι το πολλαπλό TSP (m-TSP), σκοπός του οποίου είναι η “επίσκεψη” ενός συνόλου πόλεων, όπου κάθε πόλη θα πρέπει να επισκεφθεί ακριβώς μία φορά από οποιονδήποτε από τους  $m$  πλανόδιους πωλητές. Σε κάθε  $j$  πλανόδιο πωλητή αντιστοιχεί ένα προκαθορισμένο κόστος  $d_j$ , που αντιπροσωπεύει την αμοιβή του. Κάθε  $m$  πλανόδιος πωλητής θα πρέπει να ολοκληρώσει μία υπο-διαδρομή ενώ από το συνδυασμό όλων των υπο-διαδρομών που θα ολοκληρωθεί, θα πρέπει η κάθε πόλη να επισκεφθεί ακριβώς μία φορά. Το σημείο αφετηρίας και άφιξης για τους πλανόδιους πωλητές θα πρέπει να είναι το ίδιο, ενώ στόχος είναι να προσδιοριστεί ο αριθμός των πλανόδιων πωλητών που θα πρέπει να απασχοληθούν/προσληφθούν και οι υπο-διαδρομές που θα πρέπει να ακολουθηθούν προκειμένου να μειωθεί τόσο η απόσταση που θα διανυθεί, όσο και το κόστος των απασχολούμενων πωλητών (δηλαδή ο αριθμός τους).

Η μοντελοποίηση αυτού του προβλήματος γίνεται εάν στην αρχική μορφή του TSP προστεθούν  $m-1$  επιπλέον κορυφές που υποδηλώνονται  $d-1, \dots, - (m-1)$  στις εισροές του γραφήματος. Οι  $m$  πλανόδιοι πωλητές οι οποίοι θα είναι από 0 έως  $m-1$ , αντιπροσωπεύονται από τις υπάρχουσες κορυφές και τις νέες  $m-1$  κορυφές. Στη συνέχεια, προστίθενται οι ακμές προκειμένου να συνδεθούν οι  $m-1$  νέες κορυφές στο αρχικό γράφημα. Το κόστος των νέων κορυφών καθορίζεται προσθέτοντας τα  $d_j$ , όπου  $0 \leq j \leq (m-1)$  στο κόστος των υπαρχουσών κορυφών στο γράφημα (Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., Shmoys D. B., 1985).

## 3.2 Το Πρόβλημα της Δρομολόγησης Οχημάτων

### 3.2.1 Ορισμός του προβλήματος

Το Πρόβλημα της Δρομολόγησης Οχημάτων Vehicle Routing Problem, (στο εξής αναφερόμενο ως VRP), ουσιαστικά αποτελεί μία γενίκευση του Προβλήματος του Πλανόδιου Πωλητή και αποτελεί ένα από τα πιο διαδεδομένα προβλήματα στη Συνδυαστική Βελτιστοποίηση (Caric T., Gold H., 2008). Ορίζεται από ένα πλήρες, μη κατευθυνόμενο γράφημα  $G = (V, E)$ , όπου το  $V = \{0, \dots, n\}$  είναι ένα σύνολο κορυφών και το  $E = \{(i, j): i, j \in V, i < j\}$  είναι ένα σύνολο ακμών. Κάθε κορυφή  $i \in V \setminus \{0\}$  αντιπροσωπεύει έναν πελάτη με θετική ζήτηση  $q_i$ , ενώ η κορυφή 0 αντιστοιχεί στο σταθμό. Κάθε κορυφή  $e \in E$  συσχετίζεται με ένα κόστος μετακίνησης  $c_{ij}$ . Ένας σταθερός στόλος πανομοιότυπων οχημάτων  $m$ , καθένα από τα οποία έχει ικανότητα  $Q$ , είναι διαθέσιμος στο σταθμό. Ο σκοπός του προβλήματος είναι ο καθορισμός ενός συνόλου πολλαπλών διαδρομών  $m$ , των οποίων το συνολικό κόστος μετακινήσεων να ελαχιστοποιείται έτσι ώστε:

- κάθε πελάτης να επισκέπτεται ακριβώς μία φορά μέσω μίας διαδρομής
- κάθε διαδρομή να έχει ως αφετηρία και ως τερματισμό το σταθμό
- η συνολική ζήτηση των πελατών που εξυπηρετούνται από μία διαδρομή να μην υπερβαίνει την  $Q$  ικανότητα του οχήματος και
- το μήκος της κάθε διαδρομής να μην υπερβαίνει ένα προκαθορισμένο όριο  $L$

Το VRP εμφανίζεται με διάφορες παραλλαγές λόγω της ποικιλότητας των περιορισμών που εμφανίζονται στην πράξη. Η βασική παραλλαγή του VRP είναι το Πρόβλημα Δρομολόγησης Οχημάτων με Περιορισμένη Χωρητικότητα (Capacitated VRP – CVRP) και περιγράφεται ως εξής (Min W., 2010):

*“Ένα πλήθος οχημάτων όμοιας χωρητικότητας βρίσκονται σε έναν κεντρικό σταθμό και είναι διαθέσιμα να εξυπηρετήσουν ένα σύνολο απαιτήσεων πελατών (παραδόσεις, ή εναλλακτικά, παραλαβές). Η κάθε απαίτηση αναφέρεται σε μία συγκεκριμένη τοποθεσία και έχει ένα συγκεκριμένο μέγεθος. Τα κόστη μετακινήσεων μεταξύ όλων των τοποθεσιών είναι δεδομένα. Ο στόχος είναι να σχεδιαστεί ένα σύνολο διαδρομών για τα οχήματα με ελάχιστο κόστος, έτσι ώστε όλοι οι πελάτες να επισκέπτονται μία φορά και να τηρείται η ικανότητα (χωρητικότητα) των οχημάτων.”*

### 3.2.2 Ιστορική αναδρομή του προβλήματος

Το Πρόβλημα Δρομολόγησης Οχημάτων με Περιορισμένη Χωρητικότητα παρουσιάστηκε επίσημα από τους Dantzig G.B. και Ramser J.H. (1959), οι οποίοι πρότειναν μία απλή αντιστοίχιση μέσω μιας ευρετικής λύσης. Τα επόμενα χρόνια, πολλές ευρετικές λύσεις παρουσιάστηκαν, οι οποίες ήταν βασισμένες σε ποικίλες αρχές (ανάμεσα στις οποίες ήταν η αποταμίευση, η γεωγραφική εγγύτητα και οι αντιστοιχίες πελατών) και έγιναν βήματα βελτίωσης τόσο εντός των διαδρομών όσο και μεταξύ αυτών. Ίσως η πιο γνωστή λύση είναι αυτή των Clarke G. και Wright J.R. (1964) που αφορά στην ευρετική εξοικονόμηση και η οποία χρησιμοποιείται ευρέως ακόμη και σήμερα, πιθανώς λόγω της ταχύτητας, της απλότητας και της καλής ακρίβειας που παρέχει.

Η δημοσίευση των Cristofides N., Mingozzi A. και Toth P. το 1981, συνέβαλε ουσιαστικά στην ανάπτυξη αλγορίθμων ακρίβειας για το VRP, καθώς προτεινόταν ένας αλγόριθμος βασισμένος στο δυναμικό προγραμματισμό με απλοποίηση (χαλάρωση) της χωρικής κατάστασης και γινόταν πρόταση για δύο μαθηματικές διατυπώσεις, χρησιμοποιώντας  $q$  μονοπάτια και  $k$  ελάχιστα μήκη συνδέσεων. Λίγα χρόνια αργότερα, οι Laporte G., Nobert Y. και Desrochers M. (1985) πρότειναν μία πρώτη προσέγγιση τομής για ένα VRP, βασισμένη σε μία γραμμική απλοποίηση (χαλάρωση) ενός ακέραιου μοντέλου. Αυτές οι σημαντικές έννοιες έχουν πλέον εισαχθεί σε ορισμένους από τους πιο πρόσφατα αναπτυγμένους αλγόριθμους.

Έκτοτε, έχουν προταθεί ποικίλοι αλγόριθμοι ακριβείας βασισμένοι στο μαθηματικό προγραμματισμό. Ορισμένες διατυπώσεις περιέχουν τη ροή των οχημάτων ή μεταβλητές ροής εμπορευμάτων και για την επίλυσή τους προτείνεται η χρήση της μεθόδου branch – and – cut, ενώ το VRP μπορεί επίσης να αντιμετωπιστεί σαν ένα σύνολο επιμέρους προβλημάτων, στο οποίο προστίθενται κάποιες έγκυρες ανισότητες.

Η ανάπτυξη των σύγχρονων ευρετικών λύσεων για το VRP στην πραγματικότητα ξεκίνησε το 1990 με τη χρήση των metaheuristics. Θα μπορούσε κανείς να ισχυριστεί ότι η μελέτη του VRP έχει συμβάλει σημαντικά στην ανάπτυξη και στην κατανόηση αρκετών metaheuristic εννοιών, οι οποίες είναι πλέον ευρέως γνωστές. Η έρευνα σε αυτόν τον τομέα έως τότε, ήταν αρκετά κατακερματισμένη και αρκετά μεροληπτική όσον αφορά τις προσεγγίσεις που βασιζόνταν στην αναζήτηση· ενώ από τη μία πλευρά ορισμένοι αλγόριθμοι ήταν υπερεκτιμημένοι σε επίπεδο μηχανικής, από την άλλη,



άρχισε να επικρατεί η τάση των ορθολογιστικών προσεγγίσεων. Οι καλύτερες metaheuristics είναι αυτές που εκτελούν ταυτόχρονα μία εις βάθος έρευνα για την εύρεση της βέλτιστης λύσης και μπορούν να επιλύσουν αρκετές παραλλαγές του προβλήματος. Σε γενικές γραμμές, η εφαρμογή τους γίνεται μέσω πολλαπλών χρηστών, όπως για παράδειγμα στην περίπτωση έρευνας μεγάλης προσαρμοστικότητας σε επίπεδο μιας γειτονιάς ((Pisinger D., Ropke S., 2007) ή μέσω γενετικής και τοπικής αναζήτησης, όπως για παράδειγμα στην περίπτωση του υβριδικού γενετικού αλγόριθμου (Vidal T. *et al.*, 2012).

### 3.2.3 Πολυπλοκότητα

Η κλασική μορφή του VRP ουσιαστικά αποτελεί μία γενικευμένη προσέγγιση του TSP (και πιο συγκεκριμένα του πολλαπλού TSP, του m-TSP). Από τη στιγμή που αποδείχθηκε ότι το TSP είναι δύσκολο να επιλυθεί σε μορφή NP (δηλαδή σε μη ντετερμινιστικό πολυωνυμικό χρόνο - non-deterministic polynomial-time) (Arora S., 1998), το VRP είναι επίσης δύσκολο να επιλυθεί σε μορφή NP. Ως εκ τούτου, η υπολογιστική προσπάθεια που απαιτείται προκειμένου να επιλυθεί το πρόβλημα, αυξάνεται εκθετικά σε σχέση με την αύξηση του μεγέθους του προβλήματος.

### 3.2.4 Παραλλαγές του VRP

Η βασική παραλλαγή του VRP είναι το Πρόβλημα Δρομολόγησης Οχημάτων με Περιορισμένη Χωρητικότητα (Capacitated VRP, CVRP). Αυτή η βασική μορφή όμως, δεν μπορεί να αντιμετωπίσει τα προβλήματα δρομολόγησης που αντιμετωπίζονται στην πράξη καθώς είναι πιο πολύπλοκα λόγω των παρακάτω παραγόντων:

- Ορίζοντας προγραμματισμού: Στην πραγματικότητα, οι διαδρομές σχεδιάζονται στη βάση ενός δεδομένου ορίζοντα προγραμματισμού, ο οποίος όμως μπορεί να αποτελείται από πολλές περιόδους.
- Σταθμός: Μπορούν να υπάρχουν περισσότεροι του ενός σταθμοί σε ένα μεγάλο δίκτυο διανομών και οι οποίοι μπορεί να εξυπηρετούν διαφορετικούς σκοπούς, όπως αποθήκευση εμπορευμάτων ή απευθείας παράδοσή τους, προκειμένου να ελαχιστοποιήσουν το συνολικό κόστος που δημιουργείται στην εφοδιαστική αλυσίδα. Στο πρόβλημα των πολλαπλών σταθμών, κάθε όχημα μπορεί να συνδέεται με ένα μόνο σταθμό (βάση) και αυτός ο σταθμός θα αποτελεί και την αφετηρία και το τέλος της διαδρομής του.

- Πελάτες: Στη βασική μορφή του VRP, ο κάθε πελάτης αντιστοιχεί σε μία ζήτηση. Σε πιο περίπλοκα προβλήματα, οι πελάτες μπορεί να έχουν και άλλες απαιτήσεις, όπως για παράδειγμα στην ώρα παροχής της υπηρεσίας ή/και στον τύπο του οχήματος που θα χρησιμοποιηθεί. Θα μπορούσαν επίσης να υπάρχουν και διαφορετικού είδους υπηρεσίες, όπως για παράδειγμα υπηρεσία παραλαβής, υπηρεσία παράδοσης ή και τα δύο μαζί. Επιπλέον, σε ορισμένες περιπτώσεις, οι επισκέψεις στους πελάτες δύναται να γίνονται από διαφορετικά οχήματα, αντί της πρότερης κατάστασης όπου ένα όχημα επισκεπτόταν έναν πελάτη μόνο μία φορά. Εξάλλου, σε ορισμένες εφαρμογές όπου ο προγραμματισμός περιλαμβάνει πολλές ημέρες, υπάρχουν πελάτες που έχουν απαιτήσεις κάθε μέρα, ενώ ταυτόχρονα μπορούν να αποθηκεύουν τα προϊόντα για τις επόμενες ημέρες εάν έχουν παραλάβει περισσότερα από ό,τι μπορούν να καταναλώσουν. Σε αυτή την περίπτωση, ο διανομέας θα πρέπει να σχεδιάσει ένα πλάνο διαδρομών σύμφωνα με τη ζήτηση και την καταγραφή των προϊόντων των πελατών.
- Οδηγοί: Στα περισσότερα προβλήματα που αντιμετωπίζονται στην πράξη και προκειμένου να βρεθεί η βέλτιστη λύση, οι διανομείς θα πρέπει να λαμβάνουν υπόψιν τους εργασιακούς κανονισμούς, όπως για παράδειγμα τις εργατικές βάρδιες και τα διαλείμματα.
- Οχήματα: Τα οχήματα που χρησιμοποιούνται για τις διανομές μπορεί να ποικίλουν ανάλογα με τις δυνατότητες/χωρητικότητες και τα μεγέθη τους. Συνήθως, στην πραγματικότητα υπάρχει περιορισμένος αριθμός διαθέσιμων οχημάτων και η χρήση τους δεν περιορίζεται μόνο σε μία διαδρομή, αλλά σε πολλές.
- Αντικειμενικότητα: Η αντικειμενική συνάρτηση μπορεί να είναι αρκετά περίπλοκη στην πράξη, καθώς μπορεί να συμπεριλαμβάνει την ελαχιστοποίηση του συνολικού κόστους μεταφοράς, την ελαχιστοποίηση της διαφοράς μεταξύ της μεγαλύτερης και της μικρότερης διαδρομής προκειμένου να επέλθει ισορροπία του φόρτου εργασίας μεταξύ των οδηγών, την ελαχιστοποίηση του αριθμού των οχημάτων προκειμένου να μειωθούν τα γενικά έξοδα ή/και τη μεγιστοποίηση του αριθμού των εξυπηρετούμενων πελατών προκειμένου να βελτιωθεί το επίπεδο των παρεχόμενων υπηρεσιών.

- Ανθεκτικές συσκευασίες: Σε ορισμένες εφαρμογές, η πελατειακή ζήτηση συνοδεύεται από ένα σύνολο δύο ή τριών σταθμισμένων στοιχείων, καθώς μία εφικτή δρομολόγηση συνεπάγεται μία εφικτή συσκευασία, όσον αφορά τη γεωμετρική της διάταξη.
- Αβεβαιότητα: Κατά τη σχεδίαση του πλάνου των διαδρομών μπορεί να υπάρχουν αβεβαιότητες. Για παράδειγμα, οι τοποθεσίες ή/και η ζήτηση των πελατών δεν είναι γνωστές εξ' αρχής, αλλά αποκαλύπτονται με την πάροδο του χρόνου και ενώ τα οχήματα έχουν ήδη ξεκινήσει για τη διεκπεραίωση των εργασιών τους. Σε ορισμένες περιπτώσεις, η κατανομή πιθανοτήτων αυτών των αβεβαιοτήτων είναι διαθέσιμη, ενώ σε άλλες όχι.

Οι παραπάνω παράγοντες πολυπλοκότητας και ακόμη περισσότεροι που εμφανίζονται στην πραγματικότητα, οδηγούν σε διαφορετικές παραλλαγές του CVRP, οι οποίες είναι οι εξής (Kumar S.N., Panneerselvam R., 2012):

- VRP πολλαπλών σταθμών (multi-Αμαξοστάσιο VRP, MDVRP): Το πρόβλημα MDVRP αναφέρεται στην περίπτωση που υπάρχουν περισσότεροι τους ενός σταθμοί για την εξυπηρέτηση των πελατών (Contardo C., Martinelli R., 2014). Απαραίτητη προϋπόθεση για τη λειτουργία του MDVRP είναι η σύνδεση των πελατών σε σταθμούς. Ένα σύνολο οχημάτων βρίσκεται σε κάθε σταθμό και κάθε όχημα ξεκινάει από ένα σταθμό, εξυπηρετεί τους πελάτες που έχουν συνδεθεί με αυτό τον σταθμό και επιστρέφει στον ίδιο σταθμό. Στόχος του προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους μετακινήσεων και του αριθμού των οχημάτων που χρησιμοποιούνται για την εξυπηρέτηση όλων των πελατών.
- VRP με ανομοιογενή στόλο οχημάτων (VRP with Heterogeneous Fleet, HFVRP): Όταν ο διαθέσιμος στόλος των οχημάτων που χρησιμοποιείται για τις διανομές δεν είναι ομοιογενής όσον αφορά τις δυνατότητες, το είδος του φορτίου ή/και το κόστος, το πρόβλημα χαρακτηρίζεται ως HFVRP (Karagul K., 2014). Στόχος του είναι η ελαχιστοποίηση του συνολικού κόστους μετακινήσεων και του συνολικού κόστους των οχημάτων που θα χρησιμοποιηθούν.
- VRP με χρονικά περιθώρια (VRP with Time Windows, VRPTW): Στο VRPTW κάθε πελάτης χαρακτηρίζεται από ένα συγκεκριμένο χρονικό περιθώριο  $[a_i, b_i]$ , μέσα στο οποίο θα πρέπει να εξυπηρετηθεί (El-Sherbeny N.A., 2010). Το όχημα

επιτρέπεται να φτάσει στον πελάτη πριν το σημείο  $a_i$  και να περιμένει μέχρι ο πελάτης να είναι διαθέσιμος. Όμως, η άφιξη του οχήματος μετά το σημείο  $b_i$  απαγορεύεται. Στόχος του προβλήματος είναι να ελαχιστοποιηθεί αρχικά ο αριθμός των οχημάτων που χρησιμοποιούνται και έπειτα η απόσταση που θα διανυθεί.

- VRP Παραλαβής – Παράδοσης (VRP with Pick-up and Delivery, VRPPD): Στο VRPPD λαμβάνεται υπόψη η επιστροφή των προϊόντων στο όχημα που τα παρέδωσε και ως εκ τούτου η χωρητικότητα του οχήματος θα πρέπει να είναι τέτοια ώστε τα εμπορεύματα να χωρούν σε αυτό. Στόχος του προβλήματος είναι η ελαχιστοποίηση του στόλου των οχημάτων και της συνολικής απόστασης που θα διανυθεί, με τον περιορισμό ότι το όχημα θα πρέπει να έχει αρκετή χωρητικότητα για τα εμπορεύματα που είναι προς παράδοση και παραλαβή (Ning T., Guo C., Chen R., Hua J., 2016).
- VRP με διάσπαρτη παράδοση (VRP with Split Delivery, SDVRP): Στο SDVRP κάθε πελάτης μπορεί να εξυπηρετηθεί από παραπάνω του ενός οχήματα (Dror M., Laporte G., Trudeau P., 1994). Αυτή η απλοποίηση του VRP διευκολύνει την παροχή υπηρεσιών προς τους πελάτες, κυρίως όταν το μέγεθος των παραγγελιών είναι τόσο μεγάλο όσο και η χωρητικότητα των οχημάτων. Στόχος του προβλήματος είναι η ελαχιστοποίηση του στόλου των οχημάτων και της συνολικής απόστασης που θα διανυθεί.
- Δυναμικό VRP (Dynamic VRP, DVRP): Το DVRP αποτελεί προέκταση του κλασικού VRP, στο οποίο λαμβάνονται υπόψη οι αβεβαιότητες που προκύπτουν από την πραγματική ζωή. Στόχος του προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους μεταφοράς, το οποίο όμως υπόκειται σε περιορισμούς που προέρχονται από τους αστάθμητους παράγοντες της πραγματικής ζωής (Cheng S., Cheng R., Gao J., 2017).
- Περιοδικό VRP (Periodic VRP, PVRP): Στο PVRP, υπάρχει ένας ορίζοντας  $M$  ημερών και μία συχνότητα που αντιστοιχεί σε κάθε πελάτη, η οποία δηλώνει πόσες φορές μέσα στην περίοδο των  $M$  ημερών θα πρέπει ο πελάτης να επισκέπτεται από τα οχήματα. Στόχος του προβλήματος είναι η ελαχιστοποίηση του συνολικού κόστους όλων των διαδρομών μέσα στον προκαθορισμένο χρονικό ορίζοντα (Carotenuto P., Giordani S., Massari S., Vagaggini F., 2015).

- VRP με Στοχαστικούς Πελάτες (VRP with Stochastic Σημείο συλλογής, VRPSC): Στο VRPSC, κάθε πελάτης είναι παρών βάσει μίας πιθανότητας  $p$  και απών βάσει της πιθανότητας  $(1-p)$ . Η εύρεση της βέλτιστης λύσης γίνεται σε δύο στάδια. Αρχικά, ορίζεται μία λύση χωρίς να είναι γνωστός ο ακριβής αριθμός των πελατών. Στη συνέχεια, όταν ο αριθμός των πελατών είναι γνωστός, γίνεται μία διορθωτική κίνηση προκειμένου να βρεθεί η βέλτιστη λύση. Στόχος του είναι η ελαχιστοποίηση των χρησιμοποιούμενων οχημάτων και του συνολικού κόστους μεταφοράς (Gendreau M., Laporte G., Séguin R., 1995).
- VRP με Προγραμματισμό του Πληρώματος (VRP with Crew Scheduling, VRPCS): Το VRPCS είναι ο συνδυασμός των προβλημάτων δρομολόγησης οχημάτων και προγραμματισμού του πληρώματος. Παρόλο που είναι πιο εύκολο να μελετηθούν αυτά τα δύο προβλήματα ξεχωριστά λόγω της διαφορετικότητας που έχουν από τη φύση τους, η συνδυασμένη μελέτη τους, μπορεί να αποδώσει καλύτερο προγραμματισμό όσον αφορά στην απαιτούμενη ανθρωποπροσπάθεια και ταυτόχρονα να οδηγήσει σε αισθητή μείωση του συνολικού κόστους μεταφοράς (Mingozi A., Boschetti M.A., Bianco L., Ricciardelli S., 1999).
- VRP με Στοχαστική Ζήτηση (VRP with Stochastic Demand, VRPSD): Όταν η ζήτηση του κάθε πελάτη είναι μία τυχαία μεταβλητή, το πρόβλημα ορίζεται ως VRPSD. Όπως και στο VRPSC, η βέλτιστη λύση προσδιορίζεται σε δύο στάδια και στόχος είναι η ελαχιστοποίηση του αριθμού των οχημάτων και του συνολικού κόστους μεταφοράς (Chepuri K., Homem-de-Mello T., 2005).
- VRP με Δύο/Τρεις Διαστάσεις Φόρτωσης (VRP with Two/Three Dimensional Loading, 2L-VRP/3L-VRP): Τα 2L-VRP/3L-VRP είναι ο συνδυασμός του προβλήματος δρομολόγησης οχημάτων και του προβλήματος φόρτωσης οχημάτων. Στόχος του είναι να διαχωριστούν οι πελάτες σε υποσύνολα, τα οποία δε θα υπερβαίνουν το μέγιστο αριθμό διαθέσιμων οχημάτων ενώ για κάθε υποσύνολο θα ορίζεται μία διαδρομή που θα ξεκινάει και θα τελειώνει στο σταθμό, έτσι ώστε να ελαχιστοποιείται το συνολικό κόστος μετακίνησης (Bortfeldt A., 2012).

### 3.3 Ερευνητικές μέθοδοι επίλυσης

#### 3.3.1 Τεχνικές ομαδοποίησης (Clustering Approach)

Προκειμένου να αντιμετωπιστεί αποτελεσματικά το πρόβλημα δρομολόγησης απορριμματοφόρων οχημάτων, είναι απαραίτητη η διαίρεση/διαχωρισμός της συνολικής περιοχής που εξυπηρετείται σε μικρότερες περιοχές/συστάδες (clusters) και η αντίστοιχη μέθοδος clustering.

Ο όρος clustering χρησιμοποιείται όταν θα πρέπει να επιλυθεί ένα πρόβλημα το οποίο έχει μεγάλο πλήθος σημείων και αναφέρεται στη δημιουργία διαφορετικών ομάδων, οι οποίες χρησιμοποιούνται για την κατηγοριοποίηση αντικειμένων σε αυτές (Xu D., Tian Y., 2015). Η χρήση του είναι ευρέως γνωστή, κυρίως στον κλάδο των βιολογικών επιστημών, των επιστημών διαχείρισης δεδομένων, γνώσης μηχανών (machine learning), αλλά και σε πολλές άλλες. Προκειμένου να γίνει ο σωστός διαχωρισμός των δεδομένων και των αντικειμένων στις κατάλληλες ομάδες, χρησιμοποιείται η τεχνική clustering analysis, η οποία λαμβάνει υπόψιν της την εφαρμογή από την οποία προκύπτουν τα δεδομένα ούτως ώστε κάθε ομάδα δεδομένων να είναι στο σύνολό της διαφορετική από τα δεδομένα των άλλων ομάδων.

Οι τεχνικές clustering μπορούν να κατηγοριοποιηθούν σε δύο ομάδες, στην κλασική (ντετερμινιστική) ανάλυση και στην πιθανοτική ανάλυση (Laporte G., Gendreau M., Potvin J-Y., Semet F., 2000)

Η κλασική ανάλυση αφορά τις τεχνικές clustering, όπου η ομαδοποίηση των στοιχείων σε ομάδες έχει ντετερμινιστικό αποτέλεσμα, που σημαίνει ότι κάθε φορά που θα τρέχει ο αλγόριθμος, το αποτέλεσμα θα είναι πάντα το ίδιο.

Η πιθανοτική ανάλυση (probability cluster analysis) αφορά σε τεχνικές όπου η ανάθεση των δεδομένων σε συστάδες βασίζεται σε πιθανότητες. Ακριβώς αυτό το χαρακτηριστικό τις διαχωρίζει από τις τεχνικές της ντετερμινιστικής ανάλυσης, καθώς κάθε δεδομένο έχει πιθανότητα να ανήκει σε οποιαδήποτε ομάδα (Ling R.F., 1973).

Η τεχνική clustering analysis που χρησιμοποιείται στην παρούσα εργασία είναι της ντετερμινιστικής κλασικής ανάλυσης, καθώς τα χαρακτηριστικά των στοιχείων των συστάδων (δηλαδή των κόμβων) είναι σταθερά και γνωστά εκ των προτέρων (ζήτηση κάθε κόμβου και αποστάσεις μεταξύ των κόμβων).

Στα προβλήματα δρομολόγησης οχημάτων, υπάρχουν δύο διαφορετικές προσεγγίσεις όσον αφορά τη σειρά των απαραίτητων βημάτων για την επίλυσή τους. Η πρώτη προσέγγιση είναι η περίπτωση “route first, cluster second”, όπου πρώτα γίνεται ο σχεδιασμός των διαδρομών και μετά η δημιουργία των clusters, ενώ η δεύτερη προσέγγιση είναι η περίπτωση “cluster first, route second”, όπου πρώτα γίνεται η δημιουργία των clusters και μετά ο σχεδιασμός των διαδρομών.

Στην περίπτωση προσέγγισης “route first, cluster second”, πρώτα σχεδιάζεται μία μεγάλη διαδρομή, η οποία στη συνέχεια διασπάται σε μικρότερες υπο-διαδρομές, λόγω των περιορισμών που αφορούν στη χωρητικότητα των οχημάτων (Beasley JE, 1984).

Η παρούσα εργασία χρησιμοποιεί τη δεύτερη προσέγγιση “cluster first, route second”, όπου, όπως προαναφέρθηκε, το πρώτο βήμα αφορά στη δημιουργία συστάδων κόμβων (clusters), ενώ στη συνέχεια η δρομολόγηση γίνεται με τη χρήση ενός προβλήματος πλανόδιου πωλητή ανά cluster (Hiquebran D. T., Alfa A. S., Shapiro J. A., Gittoes D. H., 1993).

Προκειμένου να επιλυθεί το πρόβλημα δρομολόγησης απορριμματοφόρων οχημάτων, όπως παρουσιάζεται εδώ, γίνεται χρήση του αλγόριθμου k means για το στάδιο της ομαδοποίησης κόμβων.

Μετά τη δημιουργία των κόμβων, το επόμενο βήμα είναι η λύση του προβλήματος του πλανόδιου πωλητή (TSP) με τη χρήση του μαθηματικού προγραμματισμού ή με ευρετικούς αλγόριθμους επίλυσης, προκειμένου να γίνει η ανάθεση των κόμβων που δημιουργήθηκαν σε ένα απορριμματοφόρο όχημα. Είναι ιδιαίτερα σημαντικό να επιλεγεί ο κατάλληλος αλγόριθμος για την ομαδοποίηση των κόμβων, καθώς είναι άμεσα συνδεδεμένος με το χρόνο επίλυσης του προβλήματος και την αποτελεσματικότητα της τελικής λύσης.

### 3.3.2 Αλγόριθμος k-means

Όπως προαναφέρθηκε, ο αλγόριθμος k-means στην παρούσα εργασία χρησιμοποιείται για το στάδιο της ομαδοποίησης κόμβων.

Ο αλγόριθμος k-means χρησιμοποιείται ευρέως στις κλασικές αναλύσεις clustering. Σκοπός του αλγορίθμου είναι η διάσπαση των κόμβων  $n$  σε  $k$  clusters, ( $k < n$ ), στα οποία κάθε κόμβος ανήκει στο cluster που βρίσκεται στο πιο κοντινό κέντρο. Έτσι, οι κόμβοι που ανήκουν σε ένα cluster από τη μία είναι όσο το δυνατόν πιο κοντά μεταξύ

τους και από την άλλη όσο το δυνατόν πιο μακριά από τους κόμβους που ανήκουν σε άλλα clusters (Kanungo T. *et al.*, 2002).

Από τα παραπάνω, προκύπτει ότι είναι πολύ σημαντική η επιλογή των σωστών clusters, αφού σε περίπτωση λάθους, θα δημιουργηθούν διαφορετικά clusters και κατά συνέπεια θα επηρεαστούν και η ποιότητα και η αποτελεσματικότητα της λύσης.

Ο αλγόριθμος k-means βασίζεται στη χρήση συγκεκριμένων βημάτων τα οποία είναι τα εξής:

- Στο **βήμα 0** καθορίζεται από πριν ο αριθμός των k clusters, βάσει του αριθμού των σημείων που πρέπει να εξυπηρετηθούν.
- Στο **βήμα 1** γίνεται τυχαία επιλογή των k αρχικών κέντρων των clusters.
- Στο **βήμα 2**, αφού υπολογιστεί η μικρότερη ευκλείδεια απόσταση μεταξύ ενός cluster και των κέντρων που επιλέχθηκαν στο βήμα 1, γίνεται ανάθεση σε κάθε ένα στοιχείο από τα δεδομένα (κόμβοι) σε ένα συγκεκριμένο cluster.
- Στο **βήμα 3**, χρησιμοποιώντας τα αποτελέσματα του βήματος 2, βρίσκονται τα νέα κέντρα των clusters, με τη χρήση των μέσων όρων των αποστάσεων των κόμβων, η ανάθεση των οποίων έχει γίνει σε clusters.
- Στο **βήμα 4** γίνεται επανάληψη του βήματος 2 και του βήματος 3 με στόχο την επίτευξη ενός κριτηρίου σύγκλισης (το κριτήριο σύγκλισης θα είναι ο μέγιστος αριθμός των επαναλήψεων να είναι 100).
- Στο **βήμα 5** και δεδομένου ότι έχουν δημιουργηθεί τα clusters, γίνεται έλεγχος ούτως ώστε να μην υπερβαίνει ο αριθμός των κόμβων τον περιορισμό των 70 που είχε τεθεί αρχικά. Εάν γίνεται τέτοιου είδους υπέρβαση, ένας από τους επιπλέον κόμβους μετακινείται σε εκείνο το cluster, το κέντρο του οποίου έχει τη μικρότερη απόσταση από τον κόμβο αυτό και ξαναγίνεται ο έλεγχος αυτού του βήματος.
- Στο **βήμα 6** τερματίζεται ο αλγόριθμος εάν ο έλεγχος του βήματος 5 καταλήξει στο αποτέλεσμα πως όλα τα clusters περιέχουν μέχρι 70 κόμβους.

Τα πλεονεκτήματα αυτής της μεθόδου είναι αφενός η δημιουργία οπτικά συνεπυγμένων clusters και αφετέρου ο διαχωρισμός μεταξύ των κόμβων, με αποτέλεσμα την εξισορρόπηση του εργασιακού φόρτου μεταξύ των απορριμματοφόρων οχημάτων. Όμως, η εξισορρόπηση του εργασιακού φόρτου που επιτυγχάνεται αποτελεί ταυτόχρονα και μειονέκτημα της μεθόδου, καθώς τα



απορριμματοφόρα οχήματα που φτάνουν στο χώρο εναπόθεσης απορριμμάτων δεν είναι πλήρη και συνεπώς δεν εξοικονομείται χωρητικότητα των οχημάτων στις περισσότερες περιπτώσεις.

### 3.3.3 Ευρετική μέθοδος επίλυσης προβλήματος πλανόδιου πωλητή

Οι αναλυτικές μέθοδοι επίλυσης του TSP που παρουσιάστηκαν παραπάνω αφορούν σε μαθηματικά μοντέλα που χρησιμοποιούν μία αντικειμενική συνάρτηση και έναν ή περισσότερους περιορισμούς.

Όταν όμως υπάρχει ένα πρόβλημα συνδυαστικής βελτιστοποίησης με τη χρήση μαθηματικού προγραμματισμού, η λύση του δυσκολεύει εκθετικά με την αύξηση του προβλήματος και η χρήση διαφόρων τεχνικών δεν οδηγεί σε ολικά βέλτιστες λύσεις, αλλά σε ικανοποιητικές.

Ένας ευρετικός αλγόριθμος θεωρείται αποδεκτός εάν μέσω της χρήσης του ικανοποιούνται κάποια κριτήρια. Για παράδειγμα, εάν η απόκλιση της λύσης από τη βέλτιστη είναι ικανοποιητική, εάν μία λογική λύση είναι εύκολο να αποκομιστεί, εάν υπάρχει λογική σειρά πάνω στην οποία βασίστηκαν οι κανόνες του ευρετικού αλγόριθμου κτλ. Σε κάθε πρόβλημα βελτιστοποίησης δεν αντιστοιχεί μόνο ένας ευρετικός αλγόριθμος, αλλά μπορούν να χρησιμοποιηθούν διαφορετικοί αλγόριθμοι, οι οποίοι σωστά συνδυασμένοι μπορούν να οδηγήσουν σε ολοένα καλύτερες λύσεις.

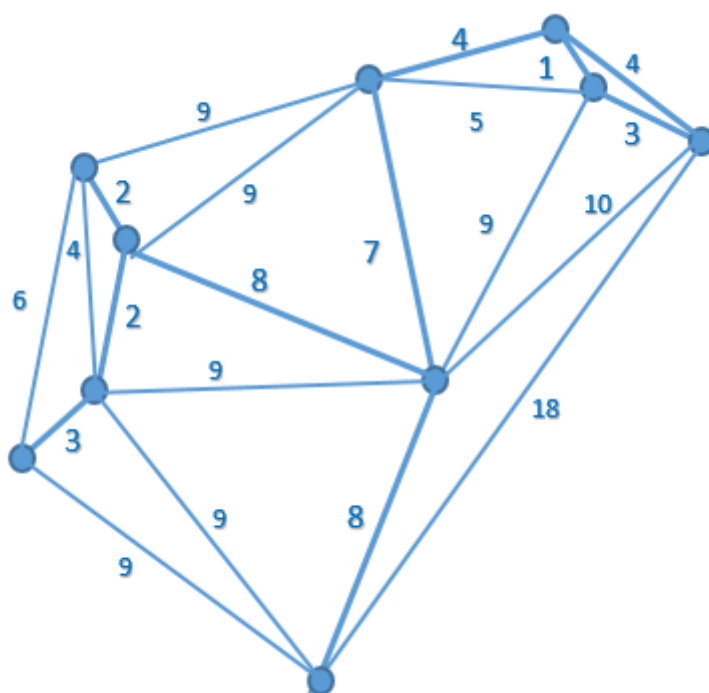
Ο αλγόριθμος “Twice Around the Minimum Spanning Tree”, θα χρησιμοποιηθεί ως ευρετική μέθοδος επίλυσης, καθώς μπορεί να αποδώσει γρήγορα αποτελέσματα / διαδρομές στις συστάδες που προέκυψαν (Burkard R.E., Deĭneko V.G., Woeginger G.J., 1999) και ο οποίος χρησιμοποιεί τους παρακάτω ορισμούς.

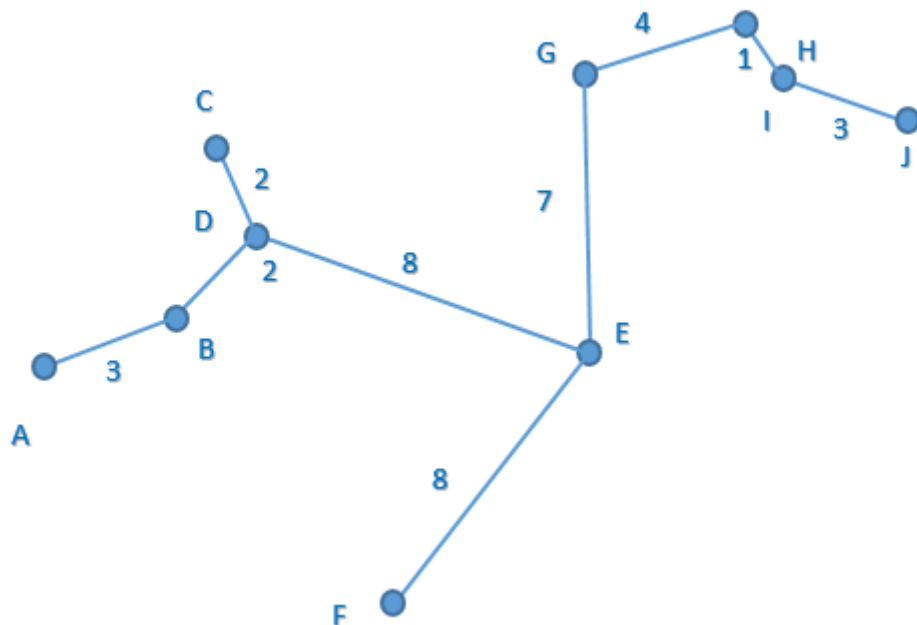
Το **γράφημα** είναι ένα σύνολο από κόμβους (κορυφές), τα οποία ενώνονται μεταξύ τους με τόξα (ακμές). Ο ορισμός του γραφήματος προκύπτει από τον τρόπο που γίνεται η σύνδεση μεταξύ των κορυφών. Το γράφημα ονομάζεται “μη κατευθυνόμενο” όταν τα τόξα δεν προσανατολίζονται, ενώ “σταθμισμένο” όταν τα τόξα του συνδέονται με κάποια αξία.

Το **δέντρο** είναι ένα κατευθυνόμενο γράφημα (δηλαδή τα τόξα του είναι προσανατολισμένα) και στο οποίο δύο οποιοδήποτε κόμβοι συνδέονται μεταξύ τους με μία μόνο απλή διαδρομή (μονοπάτι).

Το **δέντρο κάλυψης** ενός γραφήματος είναι ένα δέντρο, στοιχεία του οποίου είναι όλοι οι κόμβοι και μέρος (ή σύνολο) των τόξων ενός δικτύου. Αποτελείται ουσιαστικά από επιλογή τόξων που ανήκουν στο συνολικό δίκτυο ούτως ώστε όλοι οι κόμβοι του υποδικτύου να συνδέονται μεταξύ τους.

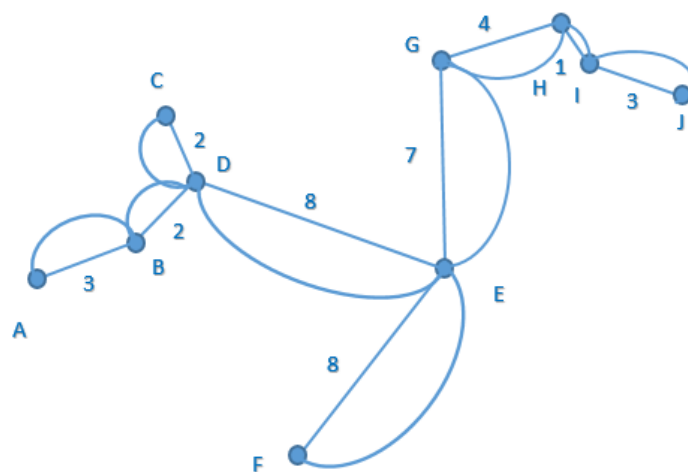
Σκοπός του προβλήματος Minimum Spanning Tree (MST), είναι να συνδεθούν όλοι οι κόμβοι που βρίσκονται στο δέντρο κάλυψης με το ελάχιστο δυνατό συνολικό μήκος κλάδων – τόξων και οι βασικότεροι αλγόριθμοι που χρησιμοποιούνται είναι ο αλγόριθμος Prim (Abhilasha R., 2013) και ο αλγόριθμός Kruskal (Li H., Xia Q., Wang Y., 2017).





Εικόνα 3: Ελάχιστο δέντρο κάλυψης

- Στο **βήμα 2**, γίνεται διπλασιασμός των κλάδων που του Ελάχιστου Δέντρου Κάλυψης, καθώς σε κάθε σημείο που υπήρχε κλάδος ένωσης δύο κόμβων, προστίθεται ένας όμοιος κλάδος, οδηγώντας στη δημιουργία ενός κλειστού δικτύου.



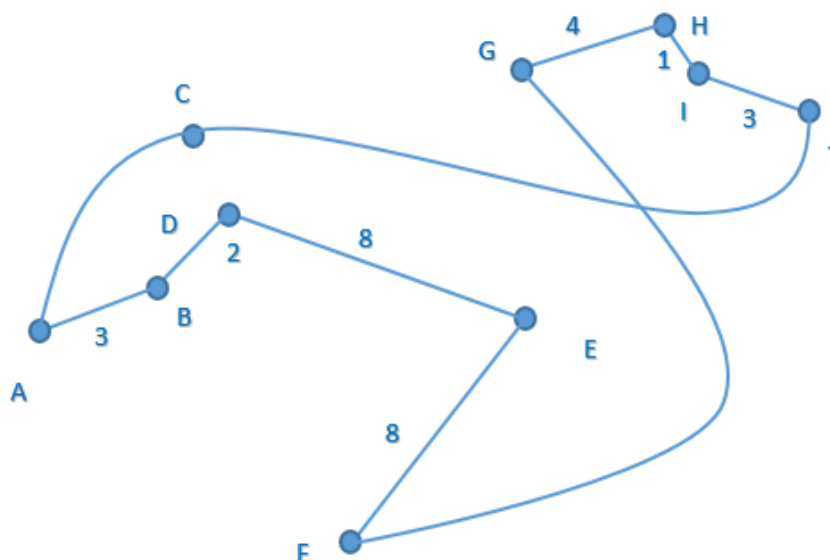
Εικόνα 4: Διπλασιασμός κλάδων ελάχιστου δέντρου κάλυψης

- Στο **βήμα 3**, από τους παραπάνω κλάδους, δημιουργείται ένα κυκλικό Οϊλεριανό (Eulerian) μονοπάτι, το οποίο είναι αυτό στο οποίο περιέχεται κάθε τόξο του δικτύου ακριβώς μία φορά (Pevzner P.A., Tang H., Waterman M.S., 2001). Το Οϊλεριανό μονοπάτι των παραπάνω εικόνων αποτελείται από τους κόμβους:

A – B – D – E – F – E – G – H – I – J – I – H – G – E – D – C – D – B – A

- Στο **βήμα 4**, με τη σειριακή διαλογή κόμβων από το Οϊλεριανό μονοπάτι, ο αλγόριθμος δημιουργεί ένα Χαμιλτονιανό κύκλο. Εάν ένας κόμβος επιλεγεί δύο φορές, στη δεύτερη φορά που θα τον βρει στο Οϊλεριανό μονοπάτι, τον προσπερνά και ελέγχει εάν δεν έχει πάρει τον επόμενο του. Αυτοί οι έλεγχοι συνεχίζονται μέχρι να εντοπιστεί ο κόμβος που δεν έχει ακόμα συμπεριληφθεί στο Χαμιλτονιανό κύκλο. Στις παραπάνω εικόνες, ξεκινώντας από τον κόμβο A, η πρώτη Χαμιλτονιανή διαδρομή θα είναι:

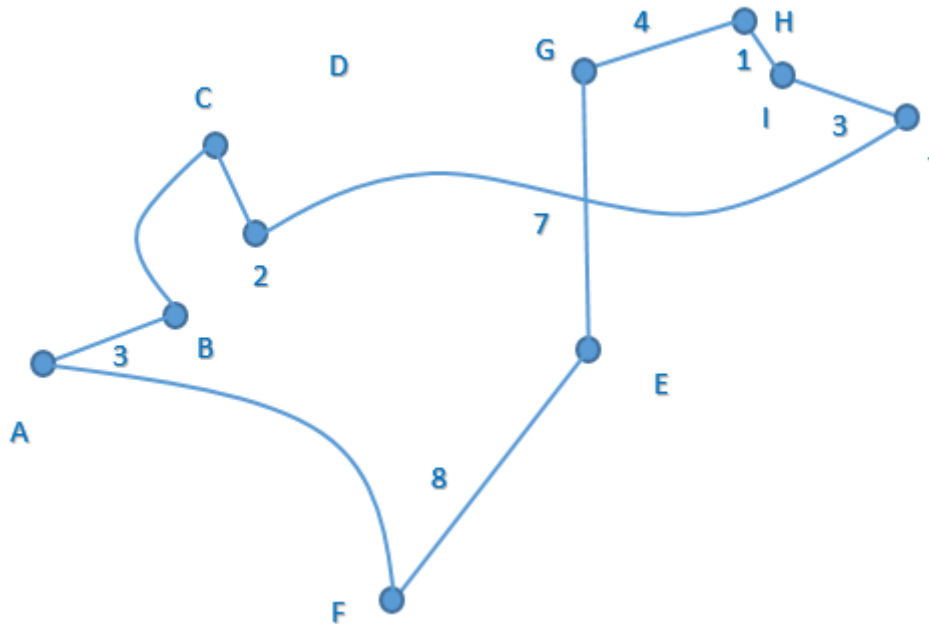
A – B – D – E – F – G – H – I – J – C – A



Εικόνα 5: Πρώτη διαδρομή του Προβλήματος Πλανόδιου Πωλητή

- Στο **βήμα 5**, με διαφορετικό κόμβο εκκίνησης κάθε φορά, δημιουργούνται ποικίλες διαδρομές TSP. Έτσι, εάν για παράδειγμα η διαδρομή έχει ως αφετηρία τον κόμβο F:

F – E – G – H – I – J – D – C – B – A – F



Εικόνα 6: Εναλλακτική διαδρομή TSP

- Στο **βήμα 6**, επαναλαμβάνεται η προηγούμενη διαδικασία, μέχρι να δημιουργηθεί ένας αποδεκτός αριθμός διαδρομών και επιλέγεται αυτή με το μικρότερο κόστος.

Στην παρούσα εργασία, για τη λύση του Προβλήματος Δρομολόγησης Απορριμματοφόρων Οχημάτων, χρησιμοποιείται ο αλγόριθμος Prim, με τον οποίο θα κατασκευαστεί το Ελάχιστο Δέντρο Κάλυψης και τα βήματα αυτής της διαδικασίας παρουσιάζονται παρακάτω.

Αρχικά, το δέντρο ξεκινά με μόνο έναν κόμβο και προστίθεται ένας επιπλέον κόμβος σε κάθε βήμα του αλγορίθμου, μέχρι να καλυφθούν όλοι οι κόμβοι που βρίσκονται στο δίκτυο.

- Το **βήμα 0** αφορά στη δημιουργία δύο συνόλων. Το ένα σύνολο θα είναι το σύνολο  $V$ , το οποίο θα περιέχει όλους τους κόμβους του δικτύου και το δεύτερο σύνολο θα είναι το κενό σύνολο  $VT$ , στο οποίο θα υπάρχουν οι κόμβοι του Ελάχιστου Δέντρου Κάλυψης. Εκτός των συνόλων  $V$  και  $VT$ , δημιουργούνται άλλα δύο σύνολα, το ένα εκ των οποίων θα είναι το  $E$  στο οποίο θα περιέχονται τα τόξα του αρχικού δικτύου και το δεύτερο το  $ET$  που στην αρχή θα είναι κενό σύνολο και θα περιέχονται μόνο τα τόξα του Ελάχιστου Δέντρου Κάλυψης.
- Στο **βήμα 1**, τοποθετείται στο  $VT$  ένας τυχαίος κόμβος του δικτύου (έστω ότι ορίζεται ως κόμβος  $u$ ), και ο οποίος αφαιρείται από το σύνολο  $V$ .
- Στο **βήμα 2**, γίνεται έλεγχος του πιο κοντινού κόμβου σε σχέση με τον αρχικό συγκρίνοντας τα μήκη των τόξων που συνδέουν τον αρχικό κόμβο με τους υπόλοιπους κόμβους (έστω ότι ορίζεται ως κόμβος  $v$ ).
- Στο **βήμα 3**, ο προαναφερθείς κόμβος  $v$  μετακινείται από το σύνολο  $V$  στο σύνολο  $VT$ , ενώ μετακινείται και το τόξο ( $uv$ ) που συνδέει τους δύο κόμβους από το σύνολο  $E$  στο σύνολο  $ET$ .
- Στο **βήμα 4**, ο κόμβος που απέχει την ελάχιστη απόσταση από οποιονδήποτε από τους κόμβους που βρίσκονται ήδη στο  $VT$  μέχρι εκείνη τη στιγμή, επιλέγεται και μετακινείται από το σύνολο  $V$  στο σύνολο  $VT$  (έστω ότι ορίζεται ως κόμβος  $p$  από το τόξο της ελάχιστης απόστασης  $pu$ ).
- Στο **βήμα 5**, γίνεται επανάληψη των διαδικασιών του βήματος 3 και ο επιλεγθείς κόμβος ( $p$ ) μετακινείται από το σύνολο  $V$  στο σύνολο  $VT$ , ενώ το τόξο ελάχιστης απόστασης  $pu$  μετακινείται από το σύνολο  $E$  στο σύνολο  $ET$ .
- Τέλος, στο **βήμα 6**, η διαδικασία αυτή συνεχίζεται έως ότου το σύνολο  $V$  μείνει κενό, ενώ το σύνολο  $ET$  θα περιέχει τα τόξα, τα οποία και θα αποτελούν το Ελάχιστο Δέντρο Κάλυψης.

## ΚΕΦΑΛΑΙΟ 4: Μαθηματικό μοντέλο προβλήματος δρομολόγησης οχημάτων για την επίλυση προβλήματος αποκομιδής απορριμμάτων

### 4.1 Προσέγγιση προβλήματος

Τα γενικά δεδομένα που απαιτούνται για την επίλυση του Προβλήματος Δρομολόγησης Απορριματοφόρων Οχημάτων (*Waste Collection Vehicle Routing Problem*) είναι τα εξής (Kim B., Kim S., Sahoo S., 2005):

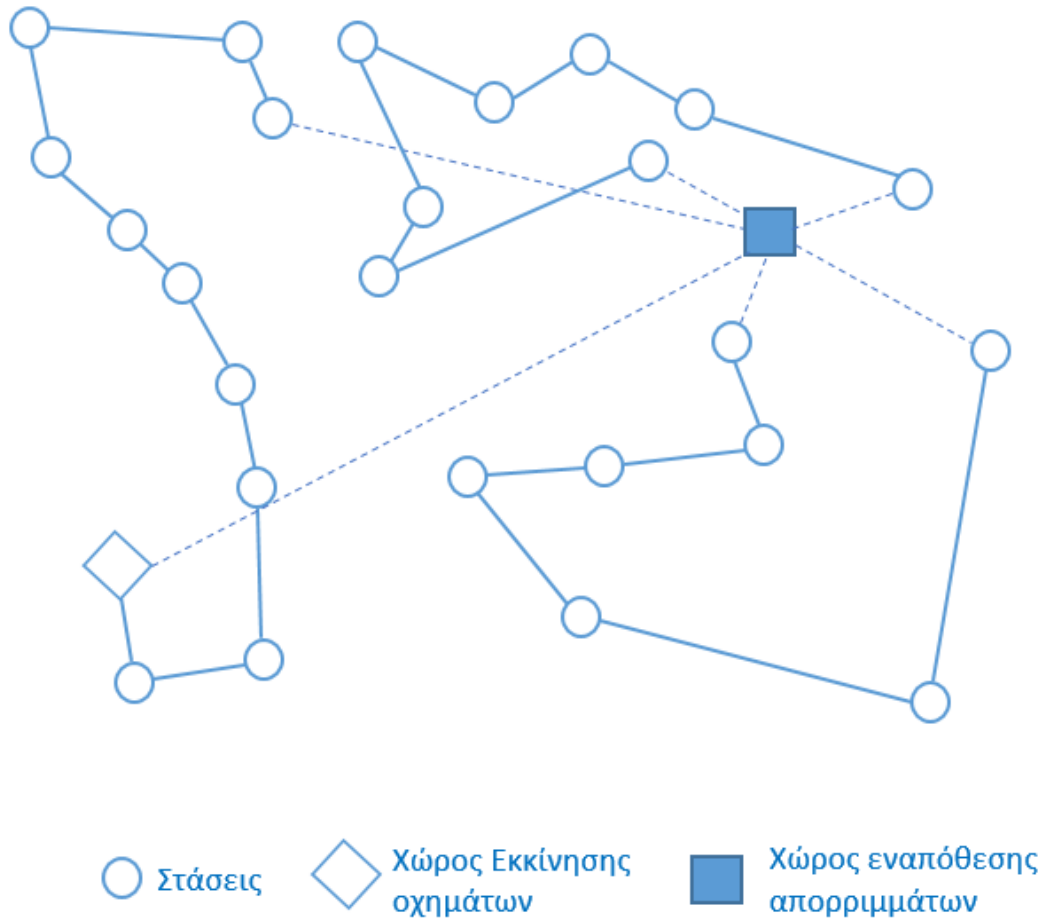
- Οι συντεταγμένες των κανονικών κόμβων (*Stops*)
- Ο αριθμός των απορριματοφόρων (*Fleet*),
- Η χωρητικότητα των απορριματοφόρων (*Capacity*)
- Η χωρητικότητα των κάδων
- Η τοποθεσία εκκίνησης απορριματοφόρων οχημάτων (*Αμαξοστάσιο-Garage*)
- Η τοποθεσία του χώρου εναπόθεσης απορριμμάτων (*Disposal Facility*)

Η ιδιαίτερη αντιμετώπιση που χρήζει το πρόβλημα δρομολόγησης απορριματοφόρων οχημάτων έναντι του προβλήματος δρομολόγησης οχημάτων έγκειται στο γεγονός ότι υπάρχουν κάποιοι εξειδικευμένοι περιορισμοί. Πιο συγκεκριμένα, τα απορριματοφόρα οχήματα, προτού επιστρέψουν στο σημείο εκκίνησής τους (αφετηρία), θα πρέπει να έχουν εναποθέσει το φορτίο που μεταφέρουν στον αντίστοιχο χώρο εναπόθεσης απορριμμάτων. Στη διαδρομή λοιπόν που ακολουθεί το όχημα, εκτός από την αφετηρία (σημείο εκκίνησης – ιδιαίτερος χώρος parking – γκαράζ) και τους κόμβους (κάδοι), θα πρέπει να υπάρχει και ένα σημείο που θα αντιπροσωπεύει το χώρο εναπόθεσης των οχημάτων.

Η δρομολόγηση των απορριματοφόρων οχημάτων ανήκει στην κατηγορία των Προβλημάτων Δρομολόγησης Οχημάτων με Περιορισμένη Χωρητικότητα (*CVRP - Capacitated VRP*).

Η διαδρομή που ακολουθούν τα απορριματοφόρα οχήματα είναι η εξής: Ξεκινούν από την αφετηρία· στη συνέχεια, με ενδιάμεσες στάσεις, περνούν από τους κόμβους (κάδοι), και αφού συλλέξουν την κατάλληλη ποσότητα απορριμμάτων (η οποία δεν μπορεί να είναι μεγαλύτερη από τη χωρητικότητά τους), καταλήγουν στο σημείο εναπόθεσης των απορριμμάτων, όπου και ξεφορτώνουν το φορτίο τους. Η

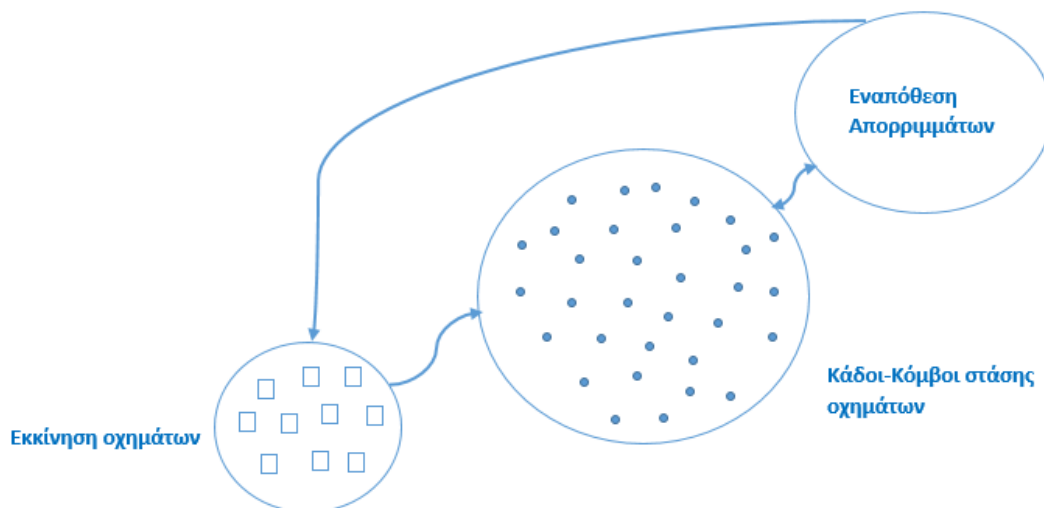
διαδρομή των απορριμματοφόρων λήγει μόλις αυτά επιστρέψουν από το σημείο εναπόθεσης των απορριμμάτων στην αφετηρία τους.



Εικόνα 7: Ενδεικτική εικόνα με τα βασικά στοιχεία του προβλήματος συλλογής απορριμμάτων

Το μαθηματικό μοντέλο που παρουσιάζεται στην παρούσα εργασία, επιτρέπει σε κάθε όχημα (απορριμματοφόρο) να κάνει περισσότερες της μίας διαδρομής από και προς το χώρο εναπόθεσης απορριμμάτων κατά τη διάρκεια μίας βάρδιας, έτσι ώστε η αντικειμενική συνάρτηση που θα αναπτυχθεί να ελαχιστοποιεί τη συνολική απόσταση που θα διανυθεί.





Εικόνα 8: Επιτρεπόμενες διαδρομές μεταξύ των συνόλων των κόμβων, της αφετηρίας και του σημείου εναπόθεσης

Στην παρούσα μελέτη, αναπτύχθηκε ένα πρόβλημα δρομολόγησης οχημάτων περισυλλογής απορριμμάτων. Το πρόβλημα αποτελείται από ένα σύνολο απορριμματοφόρων, τα οποία πρέπει να συλλέξουν τα απορρίμματα από ένα σύνολο κάδων που βρίσκονται τοποθετημένοι σε ένα δεδομένο δίκτυο. Τα απορριμματοφόρα έχουν συγκεκριμένη χωρητικότητα, ξεκινούν από ένα συγκεκριμένο αμαξοστάσιο για να επισκεφθούν τα σημεία και πρέπει να επιστρέψουν σε αυτό, εντός ενός συγκεκριμένου χρονικού διαστήματος που εκφράζει την βάρδια του προσωπικού του απορριμματοφόρου. Οι κάδοι απορριμμάτων βρίσκονται σε συγκεκριμένα σημεία και ο όγκος απορριμμάτων που περιλαμβάνουν είναι δεδομένος.

Το παραπάνω πρόβλημα αποτελεί ένα κλασικό πρόβλημα δρομολόγησης οχημάτων περιορισμένης χωρητικότητας (*Capacitated Vehicle Routing Problem*). Επιπρόσθετα, έχει εισαχθεί στο πρόβλημα ο περιορισμός στη διάρκεια του δρομολογίου του κάθε φορτηγού, με την μορφή χρονικού παραθύρου του χρόνου άφιξης στο αμαξοστάσιο, δηλαδή πρόκειται για ένα πρόβλημα δρομολόγησης οχημάτων με περιορισμό στη διάρκεια (*Time Constraint Vehicle Routing Problem*).

Τα οχήματα του στόλου δεν έχουν την ίδια χωρητικότητα, επομένως πρόκειται για ένα πρόβλημα δρομολόγησης ετερογενούς στόλου (*Heterogeneous Fleet Vehicle Routing Problem*). Ο πίνακας αποστάσεων (*distance matrix*) αφορά πραγματικές αποστάσεις, συνεπώς δεν είναι συμμετρικός (*Asymmetric Vehicle Routing Problem*).

Παρόλο που θα ληφθούν υπόψη πραγματικές αποστάσεις, ωστόσο δεν έχουν ληφθεί υπόψη οι κατευθύνσεις του δικτύου, δηλαδή το αν υπάρχουν μονόδρομοι ή όχι (*Undirected Vehicle Routing Problem*). Υπάρχει μόνο ένα αμαξοστάσιο από το οποίο ξεκινούν όλα τα οχήματα (*Single Αμαξοστάσιο Vehicle Routing Problem*), ενώ μετά την δρομολόγηση όλα τα οχήματα πρέπει να επιστρέψουν στο αμαξοστάσιο. Δηλαδή δεν πρόκειται για ανοιχτό πρόβλημα δρομολόγησης (*Open Vehicle Routing Problem*), καθώς η δρομολόγηση δεν τερματίζει στο τελευταίο σημείο του δρομολογίου.

Παρόλο που το πρόβλημα δρομολόγησης απορριμματοφόρων οχημάτων αποτελεί κατά βάση ένα περιοδικό πρόβλημα δρομολόγησης (*Periodic Vehicle Routing Problem*), θεωρήσαμε στο συγκεκριμένο πρόβλημα πως δεν υπάρχει προκαθορισμένη συχνότητα επίσκεψης, δηλαδή σε χρονικό ορίζοντα μίας εβδομάδας η συχνότητα επίσκεψης είναι κάθε ημέρα. Κατά συνέπεια, το πρόβλημα δεν αντιμετωπίστηκε ως ένα περιοδικό πρόβλημα δρομολόγησης, αλλά ως ένα πρόβλημα μίας περιόδου (*Single Period Vehicle Routing Problem*) όσον αφορά τον χρονικό ορίζοντα.

Επιπλέον, παρόλο που υπάρχει χρονικό παράθυρο στο αμαξοστάσιο, ώστε να διασφαλισθεί πως θα τηρούνται οι βάρδιες των εργαζομένων, τα σημεία από τα οποία πρέπει να γίνει η περισυλλογή δεν έχουν χρονικά παράθυρα. Για αυτό το λόγο, το πρόβλημα δεν αντιμετωπίστηκε ως πρόβλημα δρομολόγησης με χρονικά παράθυρα (*Vehicle Routing Problem with Time Windows*).

Στο υπό εξέταση πρόβλημα δεν υπάρχουν ενδιάμεσα αμαξοστάσια, όπου δυνητικά το κάθε απορριμματοφόρο θα μπορούσε να ξεφορτώσει το ήδη συλλεχθέν φορτίο και να συνεχίσει τις διαδρομές του· δηλαδή δεν πρόκειται για πρόβλημα δρομολόγησης οχημάτων με ενδιάμεσες εγκαταστάσεις (*Vehicle Routing Problem with Intermediate Facilities*). Επίσης, δεν επιτρέπεται η επιστροφή οχήματος στο αμαξοστάσιο ώστε να ξεφορτώσει το ήδη συλλεχθέν φορτίο και η επανέναρξη της δρομολόγησης, δηλαδή τα πολλαπλά δρομολόγια (*Vehicle Routing Problem with Multiple Trips*).

Τέλος, παρόλο που υπάρχει περιορισμός στη διάρκεια των δρομολογίων, κανένας περιορισμός δεν τέθηκε για την απόσταση που θα διανύσει το κάθε όχημα, συνεπώς το πρόβλημα δεν αντιμετωπίστηκε ως ένα πρόβλημα δρομολόγησης οχημάτων με περιορισμό στην διανυθείσα απόσταση (*Distance Constraint Vehicle Routing Problem*).

Το κριτήριο βελτιστοποίησης είναι η διανυθείσα απόσταση, την οποία και θέλουμε να ελαχιστοποιήσουμε. Δεν λαμβάνονται υπόψη ως κριτήρια βελτιστοποίησης η ελαχιστοποίηση της συνολικής διάρκειας των δρομολογίων, η ελαχιστοποίηση του αριθμού των οχημάτων που θα χρησιμοποιηθούν και λοιπά λειτουργικά κόστη, παρά το γεγονός πως για τη μεν διάρκεια υπάρχει περιορισμός ώστε να καλυφθεί το θέμα των βαρδιών, ενώ μέσω της βελτιστοποίησης της απόστασης ενδεχομένως ταυτόχρονα να καλύπτονται και κριτήρια ελαχιστοποίησης οχημάτων και λοιπών λειτουργικών δαπανών.

## 4.2 Ονοματολογία

Τα βασικά σύνολα του προβλήματος, οι δείκτες, τα δεδομένα και οι μεταβλητές απόφασης που θα χρησιμοποιηθούν για την επίλυση του προβλήματος που παρουσιάστηκε στην Ενότητα 4.1, παρουσιάζονται στον ακόλουθο πίνακα:

**Πίνακας 1: Μεταβλητές απόφασης, δεδομένα και δείκτες για την μοντελοποίηση CVRP**

<u>Σύνολα</u>	
$A$	Σύνολο που περιλαμβάνει όλα τα σημεία του δικτύου όπου είναι τοποθετημένοι κάδοι απορριμμάτων
$V$	Σύνολο που περιλαμβάνει όλες τις εφικτές διαδρομές μεταξύ των σημείων του δικτύου.
$K$	Σύνολο που περιλαμβάνει όλα τα διαθέσιμα οχήματα της εταιρείας συλλογής απορριμμάτων
<u>Δείκτες</u>	
$i, j \in A$	Δείκτης κόμβων του δικτύου όπου είναι τοποθετημένοι οι κάδοι απορριμμάτων
$k \in K$	Δείκτης οχημάτων

<b><u>Λεδομένα</u></b>	
$p_i$	Ημερήσια πλήρωση του κάδου $i$ .
$c_{ijk}$	Κόστος μετακίνησης του οχήματος $k$ από τον κάδο $i$ στον κάδο $j$ του δικτύου. Στο υπό εξέταση πρόβλημα εκφράζει την απόσταση.
$C_k$	Χωρητικότητα του οχήματος $k$ .
$s_i$	Χρόνος εξυπηρέτησης του κόμβου $i$ .
$t_{ijk}$	Χρόνος μετάβασης του οχήματος $k$ από τον κάδο $i$ στον κάδο $j$ του υπό εξέταση δικτύου. Προκύπτει θεωρώντας μία μέση ταχύτητα κίνησης σε αστική περιοχή μέσω της απόστασης μεταξύ των σημείων.
$D_k$	Μέγιστη επιτρεπόμενη διάρκεια διαδρομής του οχήματος $k$ .
$M$	Πολύ μεγάλος αριθμός
<b><u>Μεταβλητές Απόφασης</u></b>	
$x_{ijk}$	Δυαδική μεταβλητή που υποδεικνύει αν το όχημα $k$ θα διανύσει την διαδρομή μεταξύ $i$ και $j$ . Παίρνει την τιμή 1 όταν το όχημα $k$ διανύει το τόξο $(i, j)$ .
$P_{ijk}$	Συνεχής μεταβλητή που δείχνει το φορτίο που έχει συλλεχθεί έως το σημείο $i$ όταν ξεκινάει η εξυπηρέτηση του σημείου $j$ από το όχημα $k$ .
$T_{ik}$	Συνεχής μεταβλητή που υποδεικνύει τον χρόνο έναρξης εξυπηρέτησης του σημείου $i$ από το όχημα $k$ . Πρόκειται

	για τον χρόνο έναρξης εξυπηρέτησης του σημείου $i$ από το όχημα $k$ , που ισούται με τον χρόνο έναρξης εξυπηρέτησης του σημείου $j$ στο οποίο προηγουμένως βρισκόταν το όχημα $k$ , συν τον χρόνο μετάβασης από το σημείο $j$ στο σημείο $i$ , συν τον χρόνο εξυπηρέτησης του σημείου $j$ .
$U_{ik}$	Βοηθητική μεταβλητή για τον περιορισμό των υποδιαδρομών.

### 4.3 Μαθηματικό μοντέλο

Το μαθηματικό μοντέλο που εκφράζει το υπό εξέταση πρόβλημα είναι το ακόλουθο:

$$\min \sum_{k \in K} \sum_{(i,j) \in V} c_{ijk} x_{ijk} \quad (1)$$

$$\sum_{k \in K} \sum_{i \in A} x_{ijk} = 1 \quad \forall j \in A/\{0\} \quad (2)$$

$$\sum_{i \in A} x_{ijk} = \sum_{i \in A} x_{jik} \quad \forall j \in A, k \in K \quad (3)$$

$$\sum_{j \in A/\{0\}} x_{0jk} \leq 1 \quad \forall k \in K \quad (4)$$

$$\sum_{j \in A/\{0\}} x_{0jk} = \sum_{i \in A/\{0\}} x_{i0k} \quad \forall k \in K \quad (5)$$

$$U_{ik} = U_{jk} + |K| \cdot x_{ijk} - |K| + 1 \quad \forall i \in A/\{0\}, \forall j \in A/\{0\}, k \in K \quad (6)$$

$$\sum_{i \in A} \sum_{k \in K} p_{jik} - \sum_{j \in A} \sum_{k \in K} p_{ijk} = p_j \quad j \in A/\{0\} \quad (7)$$

$$P_{ijk} \leq \sum_{k \in K} C_k \cdot x_{ijk} \quad (i, j) \in V, k \in K \quad (8)$$

$$T_{jk} = T_{ik} + t_{ijk} + s_i + M(X_{ijk} - 1) \quad (i, j) \in V/\{0\}, \quad k \in K \quad (9)$$

$$T_{0k} + t_{i0k} + s_i + M(X_{i0k} - 1) \leq D_k \quad i \in A/\{0\}, k \in K \quad (10)$$

$$x_{ijk} \geq 0 \quad \forall k \in K, (i, j) \in V, \quad t \in T \quad (11)$$

$$x_{ijk} \text{ binary} \quad \forall k \in K, (i, j) \in V, \quad t \in T \quad (12)$$

$$P_{ijk} \geq 0 \quad (i, j) \in V, \quad k \in K \quad (13)$$

$$T_{ik} \geq 0 \quad i \in A, \quad k \in K \quad (14)$$

Η αντικειμενική συνάρτηση **(1)** ελαχιστοποιεί το συνολικό κόστος εξυπηρέτησης του δικτύου (ελαχιστοποίηση της συνολικής διανυθείσας απόστασης).

Ο περιορισμός **(2)** διασφαλίζει πως σε κάθε κάδο πηγαίνει ένα και μόνο όχημα. Ο συγκεκριμένος περιορισμός διασφαλίζει πως δεν θα υπάρχουν διαχωρισμένες διαδρομές στο πρόβλημα (*split deliveries*).

Ο περιορισμός **(3)** διασφαλίζει πως αν ένα όχημα  $k$  έχει επισκεφθεί έναν κάδο του δικτύου  $j$  ερχόμενο από έναν κάδο  $i$  τότε θα πρέπει να αναχωρήσει προς έναν άλλο κάδο του δικτύου  $i$ . Σε συνδυασμό με τον περιορισμό (2) διασφαλίζεται πως ο κάδος  $i$  για τον οποίο θα αναχωρήσει το όχημα  $k$  θα είναι διαφορετικός από αυτόν στον οποίο βρισκόταν πριν επισκεφθεί το σημείο  $j$ .

Ο περιορισμός **(4)** διασφαλίζει πως κάθε όχημα θα κάνει ένα και μόνο δρομολόγιο, δηλαδή πως δεν θα υπάρχουν πολλαπλά δρομολόγια (*multiple trips*). Αυτό προκύπτει,

περιορίζοντας την αναχώρηση του οχήματος  $k$  από το αμαξοστάσιο (σημείο 0) σε μία ή καμία φορά, δηλαδή να ξεκινήσει το πολύ για ένα δρομολόγιο, ή να μην ξεκινήσει καθόλου.

Ο περιορισμός (5) διασφαλίζει πως ένα όχημα το οποίο θα ξεκινήσει από το αμαξοστάσιο θα επιστρέψει σε αυτό. Δηλαδή, αν υπάρχει ένα σημείο  $j$  εκτός του σημείου 0 για το οποίο  $X_{0jk} = 1$  (που σημαίνει πως το όχημα θα δρομολογηθεί), τότε θα υπάρχει ένα σημείο  $i$  εκτός του σημείου 0 για το οποίο  $X_{i0k} = 1$  (δηλαδή το όχημα θα επιστρέψει αναγκαστικά στο σημείο 0, που είναι το αμαξοστάσιο).

Ο περιορισμός (6) αποτελεί τον περιορισμό των υποδιαδρομών, που εμποδίζει ένα όχημα να κάνει κυκλική διαδρομή μεταξύ των σημείων του δικτύου εκτός του αμαξοστασίου.

Ο περιορισμός (7) διασφαλίζει πως οι απαιτήσεις περισυλλογής θα ικανοποιούνται για κάθε κάδο του δικτύου. Αν ένα όχημα επισκεφθεί το σημείο  $j$  του δικτύου εκτός του αμαξοστασίου, ερχόμενο από οποιοδήποτε σημείο  $i$  του δικτύου, τότε η ποσότητα που θα έχει αναχωρώντας από το σημείο  $j$  προς οποιοδήποτε σημείο  $i$  του δικτύου, θα είναι η ποσότητα περισυλλογής  $p_j$  που αντιστοιχεί στο σημείο  $j$ .

Ο περιορισμός (8) διασφαλίζει πως οι ποσότητες που θα συλλεχθούν κατά τη δρομολόγηση του οχήματος  $k$   $P_{ijk}$  δεν θα ξεπερνούν την χωρητικότητα του  $C_k$ .

Ο περιορισμός (9) δείχνει πως ο χρόνος άφιξης του οχήματος  $k$  στο σημείο  $j$  του δικτύου  $T_{jk}$ , ισούται με το χρόνο άφιξης του οχήματος  $k$  στο σημείο  $i$  του δικτύου  $T_{ik}$ , συν τον χρόνο εξυπηρέτησης του σημείου  $i$   $s_i$ , συν τον χρόνο μετάβασης του οχήματος  $k$  από το σημείο  $i$  στο σημείο  $j$   $t_{ijk}$ , εφόσον το όχημα εκτελεί δρομολόγιο.

Ο περιορισμός (10) διασφαλίζει πως ο χρόνος άφιξης του κάθε οχήματος στο αμαξοστάσιο  $T_{0k}$ , εφόσον αυτό έχει δρομολογηθεί, δεν θα ξεπερνάει το μέγιστο επιτρεπόμενο χρόνο διάρκειας δρομολογίων του οχήματος  $k$ ,  $D_k$ .

Οι περιορισμοί (11) έως (14) αποτελούν περιορισμούς των μεταβλητών απόφασης.

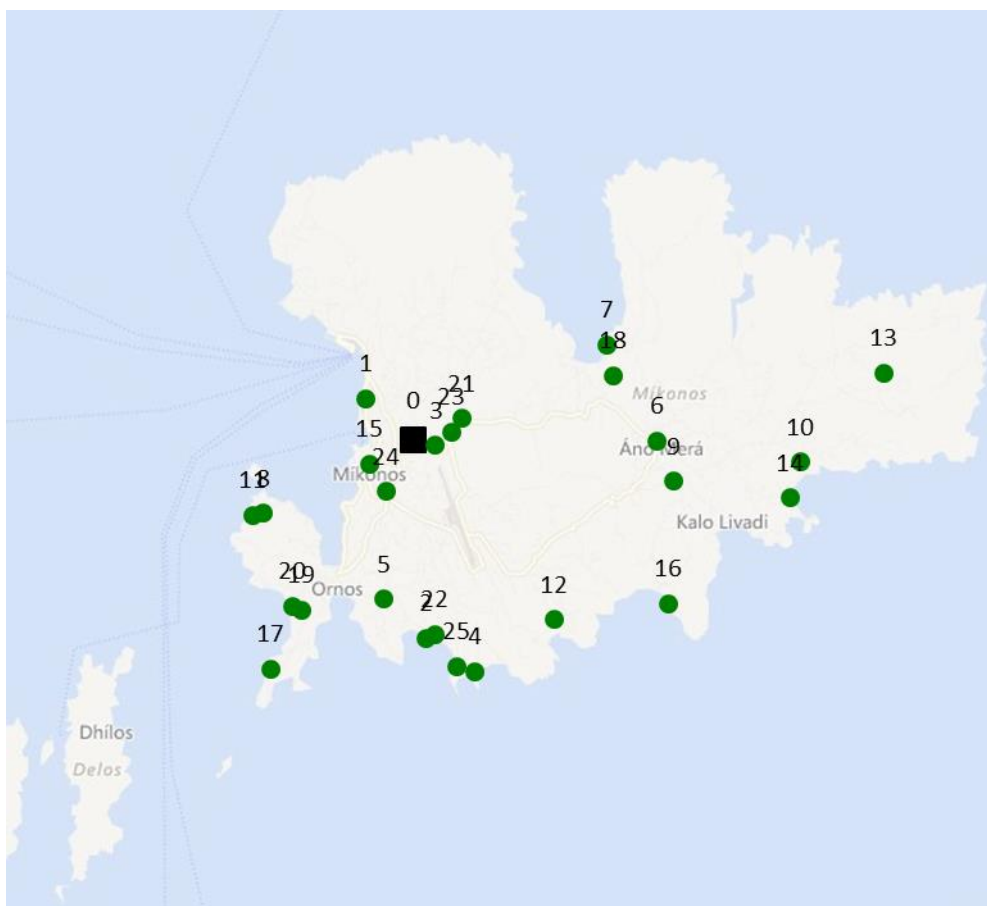
Για την επίλυση του παραπάνω μαθηματικού μοντέλου αναπτύχθηκε κώδικας με τη βοήθεια του πακέτου βελτιστοποίησης IBM ILOG CPLEX Optimization Studio, ο οποίος παρουσιάζεται αναλυτικά στο παράρτημα.

## ΚΕΦΑΛΑΙΟ 5: Εφαρμογή και αποτελέσματα

### 5.1 Δεδομένα προβλήματος

Το μοντέλο που αναπτύχθηκε στην προηγούμενη ενότητα εφαρμόστηκε σε ένα πρόβλημα συλλογής χρησιμοποιημένων λαδιών από σημεία στην περιοχή της Μυκόνου. Πιο συγκεκριμένα, υπάρχουν 25 σημεία στα οποία οι πελάτες αφήνουν χρησιμοποιημένα λάδια. Η εταιρεία που έχει αναλάβει την περισυλλογή τους, έχει 5 οχήματα περισυλλογής με διαφορετική χωρητικότητα το καθένα. Το πρόβλημα είναι η εύρεση της συντομότερης διαδρομής ώστε να πραγματοποιηθεί η περισυλλογή.

Τα σημεία περισυλλογής παρουσιάζονται στην ακόλουθη εικόνα:



Εικόνα 9: Δίκτυο σημείων περισυλλογής

Το σημείο 0 αποτελεί το σημείο στο οποίο βρίσκεται το αμαξοστάσιο, ενώ τα υπόλοιπα σημεία από τα οποία θα γίνει η περισυλλογή είναι τα σημεία 1 έως 25.



Τα οχήματα με τα οποία θα γίνει η περισυλλογή έχουν τις ακόλουθες χωρητικότητες (σε λίτρα):

**Πίνακας 2: Χωρητικότητες διαθέσιμων οχημάτων του στόλου**

Όχημα	Χωρητικότητα
K1	1.000
K2	2.000
K3	800
K4	500
K5	150

Για κάθε όχημα θεωρήθηκε μία μέση ταχύτητα 20 χλμ/ώρα, που αποτελεί μία ικανοποιητική προσέγγιση της ταχύτητας κίνησης σε αστικό περιβάλλον.

Η μέγιστη διάρκεια του κάθε δρομολογίου ορίστηκε σε 8 ώρες (ή 480 λεπτά).

Τα δρομολόγια των οχημάτων σχεδιάζονται με εμπειρικό τρόπο. Για το συγκεκριμένο παράδειγμα, η δρομολόγηση ήταν:

**Πίνακας 3: Δρομολόγηση με εμπειρικό τρόπο**

Όχημα	Διαδρομή	Ζήτηση που καλύφθηκε και ποσοστό φόρτωσης οχήματος	Διανυθείσα απόσταση
K1	Αμαξοστάσιο → Σημείο συλλογής 1 → Σημείο συλλογής 15 → Σημείο συλλογής 24 → Σημείο συλλογής 8 → Σημείο συλλογής 11 → Σημείο συλλογής 19 → Σημείο συλλογής 5 → Σημείο συλλογής 2 → Αμαξοστάσιο	997 (99,7%)	24,65 χλμ
K2	Αμαξοστάσιο → Σημείο συλλογής 20 → Σημείο συλλογής 17 → Σημείο συλλογής 22 → Σημείο συλλογής 25 → Σημείο συλλογής 4 → Σημείο συλλογής 12 → Σημείο συλλογής 16 → Σημείο συλλογής 21 → Σημείο συλλογής 3 → Αμαξοστάσιο	1.843 (92,15%)	44,78 χλμ
K3	Αμαξοστάσιο → Σημείο συλλογής 23 → Σημείο συλλογής 7 → Σημείο συλλογής 18 → Σημείο συλλογής 6 → Αμαξοστάσιο	789 (98,63%)	15,76 χλμ
K4	Αμαξοστάσιο → Σημείο συλλογής 9 → Σημείο συλλογής 13 → Σημείο συλλογής 10 → Αμαξοστάσιο	362 (72,4%)	26,30 χλμ
K5	Αμαξοστάσιο → Σημείο συλλογής 14 → Αμαξοστάσιο	144 (96%)	19,03 χλμ
<b>Συνολική διανυθείσα απόσταση</b>			<b>130,52 χλμ</b>

Η εμπειρική λύση σχηματικά αποδίδεται στην ακόλουθη εικόνα:



Εικόνα 10: Εμπειρική λύση του προβλήματος δρομολόγησης

## 5.2 Αποτελέσματα βελτιστοποίησης

Εφαρμόζοντας το μαθηματικό μοντέλο που αναπτύχθηκε στην προηγούμενη ενότητα, η λύση που προκύπτει είναι η ακόλουθη:

**Πίνακας 4: Λύση με μοντέλο βελτιστοποίησης**

Όχημα	Διαδρομή	Ζήτηση που μεταφέρθηκε και ποσοστό φόρτωσης οχήματος	Διανυθείσα απόσταση
K1	Αμαξοστάσιο → Σημείο συλλογής 7 → Σημείο συλλογής 18 → Σημείο συλλογής 6 → Σημείο συλλογής 9 → Σημείο συλλογής 16 → Σημείο συλλογής 14 → Σημείο συλλογής 13 → Σημείο συλλογής 10 → Αμαξοστάσιο	939 (93.9%)	40,41 χλμ
K2	Αμαξοστάσιο → Σημείο συλλογής 3 → Σημείο συλλογής 21 → Σημείο	1.975 (98.75%)	24,51 χλμ

	συλλογής 24 → Σημείο συλλογής 20 → Σημείο συλλογής 19 → Σημείο συλλογής 17 → Σημείο συλλογής 11 → Σημείο συλλογής 8 → Σημείο συλλογής 15 → Σημείο συλλογής 1 → Αμαξοστάσιο		
K3	Αμαξοστάσιο → Σημείο συλλογής 2 → Σημείο συλλογής 22 → Σημείο συλλογής 5 → Σημείο συλλογής 25 → Σημείο συλλογής 4 → Σημείο συλλογής 12 → Αμαξοστάσιο	721 (90.13%)	25,89 χλμ
K4	Αμαξοστάσιο → Σημείο συλλογής 23 → Αμαξοστάσιο	500 (100%)	2,35 χλμ
K5	-	-	-
<b>Συνολική διανυθείσα απόσταση</b>			<b>93,16 χλμ</b>

Η σχηματική απεικόνιση της βελτιστοποιημένης λύσης παρουσιάζεται στην ακόλουθη εικόνα:



Εικόνα 11: Λύση με μοντέλο βελτιστοποίησης

### 5.3 Σύγκριση λύσεων

Συγκρίνοντας τις παραπάνω λύσεις συμπεραίνουμε πως το μοντέλο βελτιστοποίησης μείωσε την συνολική διανυθείσα απόσταση κατά 28.62%. Παράλληλα, μειώθηκε και ο αριθμός των οχημάτων που δρομολογήθηκαν, αφού έχουν δρομολογηθεί 4 από τα 5 διαθέσιμα οχήματα, ενώ στην εμπειρική λύση είχαν δρομολογηθεί όλα τα διαθέσιμα οχήματα του στόλου.

Τέλος, αξίζει να σημειωθεί πως με την βελτιστοποιημένη λύση υπάρχει μεγαλύτερη πληρότητα στα οχήματα, συμβάλλοντας με αυτόν τον τρόπο στην μείωση των διαδρομών χωρίς φορτίο (*empty runs*).

## Βιβλιογραφία

- Abhilasha R., “*Minimum Cost Spanning Tree Using Prim’s Algorithm*”, International Journal of Advance Research in Computer Science and Management Studies, 2013, **1**(1): p. 15-20
- Arora S., “*Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems*”, Journal of the ACM, 1998, **45**(5): p. 753–782
- Bardan, M.F. and El-Hagar, S.M., “*Optimization of municipal solid waste management in Port Said-Egypt*”, Waste Management, 2006, **26**: p. 534-545
- Beasley JE, “*Route first—Cluster second methods for vehicle routing*”, Omega, 1984, **11**(4): p. 403-408
- Bortfeldt A., “*A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints*”, Computers and Operations Research, 2012, **39**(9): p. 2248-2257
- Burkard R.E., Deĭneko V.G., Woeginger G.J., 1999, “*Erratum: The Travelling Salesman and the PQ-Tree*”, Mathematics of Operations Research, 1999, **24**(1): p. 262-272
- Caric T., Gold H., “*Vehicle Routing Problem*”, In-Teh, 2008
- Carotenuto P., Giordani S., Massari S., Vagaggini F., “*Periodic capacitated vehicle routing for retail distribution of fuel oils*”, Transportation Research Procedia, 2015, **10**: p. 735-744
- Cheng, H., Chang, Ni, “*Prediction analysis of solid waste generation based on grey fuzzy dynamic modeling*”, Resources, Conservation and Recycling, 2000, **29**: p. 1-18
- Cheng S., Cheng R., Gao J., “*A Monarch Butterfly Optimization for the Dynamic Vehicle Routing Problem*”, Algorithms, 2017
- Chepuri K., Homem-de-Mello T., “*Solving the Vehicle Routing Problem with Stochastic Demands using the Cross-Entropy Method*”, Annals of Operations Research, 2005, **134**: p. 153-181
- Christofides N., Mingozzi A., Toth P., “*Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations*”, Mathematical Programming, 1981, **20**(1): p. 255-282

- Clarke, G., Wright, J.R., “*Scheduling of Vehicle Routing Problem from a Central Αμαξοστάσιο to a Number of Delivery Points*”, Operations Research, 1964, **12**: p. 568-581
- Contardo C., Martinelli R., “*A new exact algorithm for the multi-Αμαξοστάσιο vehicle routing problem under capacity and route length constraints*”, Discrete Optimization, 2014, **12**: p. 129-146
- Dantzig G. B., Ramser J. H., “*The Truck Dispatching Problem*”, Management Science, 1959, **6**(1): p. 80-91
- Deng J.L., “*Control problems of grey systems*”, Systems & Control Letters, 1982, **1**: p. 5
- Dror M., Laporte G., Trudeau P., “*Vehicle routing with split deliveries*”, Discrete Applied Mathematics, 1994, **50**: p. 239-254
- El-Sherbeny N.A., “*Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods*”, Journal of King Saud University – Science, 2010, **22**(3): p. 123-131
- Gendreau M., Laporte G., Séguin R., “*An Exact Algorithm for the Vehicle Routing Problem with Stochastic Demands and Σημείο συλλογής*”, Transportation Science, 1995, **29**(2): p. 107-197
- Ghiani, G., Manni, A., Manni, E., Toraldo, M., “*The impact of an efficient collection sites location on zoning phase in municipal solid waste management*”, Waste Management, 2014. **34**: p. 1949-1956
- Gutin G, Punnen A.P., “*The Traveling Salesman Problem and Its Variations*”, Springer, 2007
- Gutin G., Yeo A., Zverovich A., “*Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP*”, Discrete Applied Mathematics, 2002, **117**: p. 81-86
- Han, H. and Ponce-Cueto, E., “*Waste collection vehicle routing problem: literature review*”, Traffic Engineering Review, 2015. **27**(4): p. 345-358
- Hiquebran D. T., Alfa A. S., Shapiro J. A., Gittoes D. H., “*A revised simulated annealing and cluster-first route-second algorithm applied to the vehicle routing problem*”, Engineering Optimization, 1993, **22**: p. 77
- Huixian Xu et al. “*Merrill M. Flood: 2nd President of TIMS (1955) and 10th President of ORSA, 1961–62*”, 2001

- Kanungo T., Mount D.M., Netanyahu N.S., Piatko C.D., Silverman R., Wu A.Y., *“An Efficient k-Means Clustering Algorithm: Analysis and Implementation”*, IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002, **24**(7): p. 881-892
- Karagul K., *“A New Heuristic Routing Algorithm for Fleet Size and Mix Vehicle Routing Problem”*, Gazi University Journal of Science, 2014, **27**(3): p. 979-986
- Katsamaki, A., Willems S., Diamadopoulos E., *“Time series analysis of municipal solid waste generation rates”*, Journal of Environmental Engineering, 1998, **124**(2): p. 178-183
- Kim B., Kim S., Sahoo S., *“Waste collection vehicle routing problem with time windows”*, Computers & Operations Research, 2005, **33**: p. 3624-3642
- Kulkarni R. V., Bhawe P. R., *“Integer programming formulations of vehicle routing problems”*, European Journal of Operational Research, 1985, **20**(1): p. 58-67
- Kumar S.N., Panneerselvam R., *“A Survey on the Vehicle Routing Problem and Its Variants”*, Intelligent Information Management, 2012, **4**: p. 66-74
- Laporte G., Gendreau M., Potvin J-Y., Semet F., *“Classical and modern heuristics for the vehicle routing problem”*, International Transactions in Operational Research, **7**: p. 285-300
- Laporte G., Nobert Y., Desrochers M., *“Optimal Routing under Capacity and Distance Restrictions”*, Operations Research, 1985, **33**(5): p. 1050-1073
- Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G., Shmoys D. B., *“The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization”*, John Wiley & Sons, 1985
- Layman’s Report, *“Ανάπτυξη των συστημάτων Πληρώνω όσο Πετάω σε Ελλάδα, Εσθονία και Κύπρο”*, Life + Environment Policy and Governance LIFE 07ENV/GR/000271
- Li, Y., Huang, G., Nie, S., *“A Mathematical model for identifying an optimal waste management policy under uncertainty”*, Applied Mathematical Modelling, 2012, **36**: p. 2658–2673
- Li H., Xia Q., Wang Y., *“Research and Improvement of Kruskal Algorithm”*, Journal of Computer and Communications, 2017, **5**(12): p. 63-69
- Ling R.F., *“A Probability Theory of Cluster Analysis”*, Journal of the American Statistical Association, 1978, **68**(341): p. 159-164

- Mataija M., Rakamarić Š., Jozić F., “*Solving the travelling salesman problem using the Branch and Bound method*”, Zbornik Veleučilišta u Rijeci, 2016, **4**(1): p. 259-270
- Min W. “*Rich Vehicle Routing Problems and Applications*” PhD Thesis, DTU Management Engineering, 2010
- Mingozi A., Boschetti M.A., Bianco L., Ricciardelli S., “*A Set Partitioning Approach to the Crew Scheduling Problem*”, Operations Research, 1996, **47**(6): p. 883-888
- Monnot J., Toulouse S., “*Paradigms of Combinatorial Optimization*”, ISTE Ltd and John Wiley & Sons, Inc., 2014, Edited by Paschos V.
- Ning T., Guo C., Chen R., Hua J., “*A Novel Hybrid Method on VRP with Pickup and Delivery*”, The Open Cybernetics & Systemics Journal, 2016, **10**: p. 56-60
- Pevzner P.A., Tang H., Waterman M.S., “*An Eulerian path approach to DNA fragment assembly*”, Applied Mathematics, Genetics, 2001, **98**(17): p. 9748–9753
- Pisinger D., Ropke S. “*A general heuristic for vehicle routing problems*”, Computers & Operations Research, 2007, **34**(8): p. 2403-2435
- Sahoo S., Kim B., Kim S., Kraas B., Popov Jr. A., “*Routing Optimization for Waste Management*”, Interfaces, 2005, **35**(1): p. 24-36
- Skiena, S. “*Hamiltonian Cycles*” Addison-Wesley, 1990, p. 196-198.
- Swapan, D. and Bhattacharyya B., “*Optimization of municipal solid waste collection and transportation routes*”, Waste Management, 2015, **43**: p. 9-18
- Tobias, K., “*SWANA Technical Assistance Project*”, 2010
- UNSD, “*Glossary of environment Statistics*”, United Nations Publication, Studies in Methods, 1997. Series F, No 7: p. 1-83
- Ventosa, I., Forn, M., Montserrat, M., “*Guide for the implementation of Pay as you throw systems for municipal waste*”, Agència de Residus de Catalunya (ARC), 2010
- Vidal T., Crainic T.G., Gendreau M., Lahrichi N., Rei W., “*A Hybrid Genetic Algorithm for MultiΑμαξοστάσιο and Periodic Vehicle Routing Problems*”, Operations Research, 2012, **60**(3): p. iii-768
- Xu D., Tian Y., “*A Comprehensive Survey of Clustering Algorithms*”, Annals of Data Science, 2015, **2**(2): p. 165-193
- Zambito, L. “*The Traveling Salesman Problem: A Comprehensive Survey*”, 2006. s.l. : Submitted as a project for CSE 4080.



- Κυρκίτσος, Φ., Παπαθεοχάρη, Σ., Χωματίδης, Δ., Χρυσόγελος, Ν., Δαλαμάγκα, Α., Γκιοκά, Π., Μακρυνίκα Κ., Δήμου, Α. “Μελέτη στο πλαίσιο του προγράμματος LIFE07ENV/GR/000271, Δράση 2.2.2.1. Η Εφαρμογή των συστημάτων πληρώνω όσο πετάω στην Ελλάδα”, Αθήνα, 2011
- Τερζής, Ε., “Διαχείριση Απορριμμάτων”, Οδηγός για το περιβάλλον, WWF Ελλάς, 2009

#### **Ιστοσελίδες**

- European Commission, “*HEC PAYT The Development of Pay as You Throw Systems in Hellas, Estonia and Cyprus*”, διαθέσιμο σε: [http://ec.europa.eu/environment/life/project/Projects/index.cfm?fuseaction=search.dspPage&n\\_proj\\_id=3258](http://ec.europa.eu/environment/life/project/Projects/index.cfm?fuseaction=search.dspPage&n_proj_id=3258) (3/5/2017)
- Wikipedia, the free encyclopedia - Merrill M. Flood. [Online] August 11, 2018, διαθέσιμο σε: [https://en.wikipedia.org/wiki/Merrill\\_M.\\_Flood](https://en.wikipedia.org/wiki/Merrill_M._Flood)
- Wikipedia, the free encyclopedia - RAND. [Online] August 23, 2018, διαθέσιμο σε: <http://en.wikipedia.org/wiki/RAND>

## Παράρτημα

```
#include <ilcplex/ilocplex.h>
#include <stdio.h>
#include <ilconcert/iloexpression.h>
#include <string>
#include <vector>
using std::vector;

ILOSTLBEGIN

int i,j,k,f,count;
const int imax=26; //node index
const int jmax=26; //node index
const int kmax=5; //vehicle index
const int fmax = 3; //number of data files
int Dmax = 480;
float M=10000; //very large number
int lastl;

int main (int argc, char **argv)
{
    double c[imax][jmax][kmax]; //cost of vehicle k to travel from node i to
j
    double c1[imax][jmax]; //cost of vehicle k to travel from node i to
j
    float C[kmax]; //capacity of vehicle k
    double p[imax]; //pick-up demand of node i
    double s[imax]; //service time of node i
    int time[imax][jmax][kmax]; //time of vehicle k to travel from node i to
j
    float speed[kmax]; //average speed of vehicle k
    double x[imax]; //x coordinate (longitude) of
node i
    double y[imax]; //y coordinate (latitude) of
node i
    char* filename[fmax]; //name of data file to be
imported

    for (i=0;i<imax;i++){
        p[i]=0;
        s[i]=0;
        for (j=0;j<jmax;j++){
            for (k=0;k<kmax;k++){
                c[i][j][k]=0;
                c1[i][j]=0;
                time[i][j][k]=0;
            }
        }
    }

    for (k=0;k<kmax;k++){
        C[k]=0;
        speed[k]=0;
    }
}
```

```

// Data import

filename[0]="[...path...]/1 Nodes data.txt";
filename[1]=" [...path...]/2 Vehicles Data.txt";
filename[2]=" [...path...]/3 Distance Data.txt";

for (f=0;f<fmax;f++){
    ifstream file(filename[f]);
    if (f == 0){
        for (i = 0; i < imax; i++){
            file >> i;
            file >> x[i];
            file >> y[i];
            file >> p[i];
            file >> s[i];
        }
    }

    else if (f==1){
        for (k = 0; k<kmax; k++){
            file >> k;
            file >> C[k];
            file >> speed[k];
            cout << "k=" << k << " " << "C[k]=" << C[k] << " "
<< " speed[k]=" << speed[k] << endl;
        }
    }

    else if (f==2){
        for (i = 0; i<imax; i++){
            for (j = 0; j<jmax; j++){
                file >> i;
                file >> j;
                file >> c1[i][j];
            }
        }

        file.close();
    }

    for (i=0;i<imax;i++){
        for (j=0;j<jmax;j++){
            for (k=0;k<kmax;k++){
                if (i!=j){
                    c[i][j][k]=c1[i][j];
                    time[i][j][k]=c[i][j][k]/speed[k];
                }
                else if (i==j){
                    c[i][j][k]=M;
                    time[i][j][k]=M;
                }
            }
        }
    }

    //-----
    //----- Model-----
    IloEnv env;

    try {

```

```

IloModel model (env);

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

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

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

IloCplex cplex(env);

//----- Model-----
//-----

//-----Decision Variables-----

//-----Decision Variable X-----
//binary variable indicating whether vehicle k visits arc (i,j)

IloNumVarMatrix3x3 Xijk(env,0);
for (i=0;i<imax;i++){
    IloNumVarMatrix2x2 Xjk(env,0);
    for (j=0;j<jmax;j++){
        IloNumVarArray Xk(env,0);
        for (k=0;k<kmax;k++){
            char Variable_X[70];

            sprintf(Variable_X,"Xijk(%d,%d,%d)",i,j,k);
            IloNumVar
X(env,0,1,ILOINT,Variable_X);
            Xk.add(X);
        }
        Xjk.add(Xk);
    }
    Xijk.add(Xjk);
}

//-----Decision Variable T-----
//time variable indicating the time at which vehicle k starts servicing node i

IloNumVarMatrix2x2 Tik(env,0);
for (i=0;i<imax;i++){
    IloNumVarArray Tk(env,0);
    for (k=0;k<kmax;k++){
        char Variable_T[70];

        sprintf(Variable_T,"Tik(%d,%d)",i,k);
        IloNumVar
T(env,0,IloInfinity,ILOINT,Variable_T);
        Tk.add(T);
    }
    Tik.add(Tk);
}

//-----Decision Variable U-----
//auxiliary variable for subtour elimination

```

```

        IloNumVarMatrix2x2 Uik(env,0);
        for (i=0;i<imax;i++){
            IloNumVarArray Uk(env,0);
            for (k=0;k<kmax;k++){
                char Variable_U[70];

                sprintf(Variable_U,"Uik(%d,%d)",i,k);
                IloNumVar
U(env,0,IloInfinity,ILOFLOAT,Variable_U);
                Uk.add(U);
            }
            Uik.add(Uk);
        }

//-----Decision Variable P-----
//continuous variable indicating the load picked-up in nodes routed up to node
i (including node i) and transported in arc (i,j) on day t

        IloNumVarMatrix3x3 Pijk(env,0);
        for (i=0;i<imax;i++){
            IloNumVarMatrix2x2 Pjk(env,0);
            for (j=0;j<jmax;j++){
                IloNumVarArray Pk(env,0);
                for (k=0;k<kmax;k++){
                    char Variable_P[70];

                    sprintf(Variable_P,"Pijk(%d,%d,%d)",i,j,k);
                    IloNumVar
P(env,0,IloInfinity,ILOINT,Variable_P);
                    Pk.add(P);
                }
                Pjk.add(Pk);
            }
            Pijk.add(Pjk);
        }

//-----Decision Variables-----
//-----

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

//Capacitated VRP-CVRP
//-----Constraint 2-----
//each client is visited by exactly one vehicle

        IloRangeArray Constraint_2j(env,0);
        for (j=1;j<jmax;j++){
            IloExpr expr(env,0);
            IloExpr expr1(env,0);
            IloExpr expr2(env,0);
            for (i=0;i<imax;i++){
                for (k = 0; k < kmax; k++){
                    expr1 += Xijk[i][j][k];
                }
            }
            expr=expr1-1;
            char Con_2[60];
            sprintf(Con_2,"Constraint_2j(%d)",j);
            double LB=0,UB=0;

```

```

        IloRange Constraint_2(env, LB, expr, UB, Con_2);
        expr1.end();
        expr.end();
        model.add(Constraint_2);
        Constraint_2j.add(Constraint_2);
    }

//-----Constraint 3-----
//the same vehicle arrives and departs from each client it serves

    IloRangeMatrix2x2 Constraint_3jk(env, 0);
    for (j=0; j<jmax; j++){
        IloRangeArray Constraint_3k(env, 0);
        for (k=0; k<kmax; k++){
            IloExpr expr(env, 0);
            IloExpr expr1(env, 0);
            IloExpr expr2(env, 0);
            for (i=0; i<imax; i++){
                expr1+=Xijk[i][j][k];
                expr2+=Xijk[j][i][k];
            }
            expr=expr1-expr2;
            char Con_3[60];
            sprintf(Con_3, "Constraint_3jk(j%d), (k%d)", j, k);
            double LB=0, UB=0;
            IloRange Constraint_3(env, LB, expr, UB, Con_3);
            expr1.end();
            expr2.end();
            expr.end();
            model.add(Constraint_3);
            Constraint_3k.add(Constraint_3);
        }
        Constraint_3jk.add(Constraint_3k);
    }

//-----Constraint 4-----
//each vehicle is used at most once a day

    IloRangeArray Constraint_4k(env, 0);
    for (k=0; k<kmax; k++){
        IloExpr expr(env, 0);
        IloExpr expr1(env, 0);
        for (i = 0; i < 1; i++){
            for (j=1; j<jmax; j++){
                expr1+=Xijk[i][j][k];
            }
        }
        expr=expr1-1;
        char Con_4[60];
        sprintf(Con_4, "Constraint_4k(k%d)", k);
        double LB=-IloInfinity, UB=0;
        IloRange Constraint_4(env, LB, expr, UB, Con_4);
        expr1.end();
        expr.end();
        model.add(Constraint_4);
        Constraint_4k.add(Constraint_4);
    }

```

```

//-----Constraint 5-----
//all vehicles utilized will return to the destination Αμαξοστάσιο

IloRangeArray Constraint_5k(env, 0);
for (k=0;k<kmax;k++){
    IloExpr expr(env,0);
    IloExpr expr1(env,0);
    IloExpr expr2(env,0);
    for (i = 0; i < 1;i++){
        for (j=1;j<jmax;j++){
            expr1+=Xijk[i][j][k];
        }
    }
    for (i=1;i<imax;i++){
        for (j=0;j<1;j++){
            expr2+=Xijk[i][j][k];
        }
    }
    expr=expr2-expr1;
    char Con_5[60];
    sprintf(Con_5, "Constraint_5k(k%d)", k);
    float LB=0,UB=0;
    IloRange Constraint_5(env, LB, expr, UB, Con_5);
    expr1.end();
    expr2.end();
    expr.end();
    model.add(Constraint_5);
    Constraint_5k.add(Constraint_5);
}

//-----Constraint 6-----
//subtour elimination constraint

IloRangeMatrix3x3 Constraint_6ijk(env,0);
for (i=1;i<imax;i++){
    IloRangeMatrix2x2 Constraint_6jk(env,0);
    for (j=1;j<jmax;j++){
        IloRangeArray Constraint_6k(env, 0);
        for (k=0;k<kmax;k++){
            IloExpr expr(env,0);
            if (i!=j){
                expr=Uik[i][k]-
Uik[j][k]+(jmax)*Xijk[i][j][k]-jmax+1;
            }
            char Con_6[60];

            sprintf(Con_6, "Constraint_6ijk(i%d),(j%d),(k%d)", i,j,k);
            double LB=-IloInfinity,UB=0;
            IloRange

Constraint_6(env, LB, expr, UB, Con_6);

            expr.end();
            model.add(Constraint_6);

            Constraint_6k.add(Constraint_6);
        }
        Constraint_6jk.add(Constraint_6k);
    }
    Constraint_6ijk.add(Constraint_6jk);
}

//-----Constraint 7-----

```

```
//flow equations for pick-up demands. They guarantee that pick up demands are
satisfied for each node.
```

```
IloRangeArray Constraint_7j(env, 0);
for (j=1;j<jmax;j++){
    IloExpr expr(env,0);
    IloExpr expr1(env,0);
    IloExpr expr2(env,0);
    for (i=0;i<imax;i++){
        for (k=0;k<kmax;k++){
            expr1+=Pijk[j][i][k]-Pijk[i][j][k];
        }
        expr2+=p[j];
        expr=expr1-expr2;
        char Con_7[60];
        sprintf(Con_7,"Constraint_7j(i%d)",j);
        float LB=0,UB=0;
        IloRange Constraint_7(env,LB,expr,UB,Con_7);
        expr1.end();
        expr2.end();
        expr.end();
        model.add(Constraint_7);
        Constraint_7j.add(Constraint_7);
    }
}
```

```
//-----Constraint 8-----
//pickup demands will only be transported using arcs included in the solution
```

```
IloRangeMatrix3x3 Constraint_8ijk(env,0);
for (i=0;i<imax;i++){
    IloRangeMatrix2x2 Constraint_8jk(env,0);
    for (j=0;j<jmax;j++){
        IloRangeArray Constraint_8k(env, 0);
        for (k=0;k<kmax;k++){
            IloExpr expr(env,0);
            IloExpr expr1(env,0);
            IloExpr expr2(env,0);
            expr1=C[k]*Xijk[i][j][k];
            expr2=Pijk[i][j][k];
            expr=expr2-expr1;
            char Con_8[60];

            sprintf(Con_8,"Constraint_8ijk(i%d),(j%d),(k%d)",i,j,k);
            double LB=-IloInfinity,UB=0;
            IloRange Constraint_8(env,LB,expr,UB,Con_8);
            expr1.end();
            expr2.end();
            expr.end();
            model.add(Constraint_8);
            Constraint_8k.add(Constraint_8);
        }
        Constraint_8jk.add(Constraint_8k);
    }
    Constraint_8ijk.add(Constraint_8jk);
}
}
```

```
//-----Constraint 9-----
//the starting time of service of node j will be equal to the starting time of
service of node i, plus the service time of node i and the time need by vehicle
```



k to travel from node i to node j if arc (i,j) is on the route of vehicle k on day t

```

IloRangeMatrix3x3 Constraint_9ijk(env,0);
for (i=1;i<imax;i++){
    IloRangeMatrix2x2 Constraint_9jk(env,0);
    for (j=1;j<jmax;j++){
        IloRangeArray Constraint_9k(env,0);
        for (k=0;k<kmax;k++){
            IloExpr expr(env,0);
            if (i!=j){
                expr=Tik[i][k]+time[i][j][k]+s[i]-
Tik[j][k]-M+M*Xijk[i][j][k];
            }
            char Con_9[60];

            sprintf(Con_9,"Constraint_9ijk(%d),(j%d),(k%d)",i,j,k);
            double LB=-IloInfinity,UB=0;
            IloRange Constraint_9(env,LB,expr,UB,Con_9);
            expr.end();
            model.add(Constraint_9);
            Constraint_9k.add(Constraint_9);
        }
        Constraint_9jk.add(Constraint_9k);
    }
    Constraint_9ijk.add(Constraint_9jk);
}

//-----Constraint 10-----
-
//Restriction to route lenght, all vehicles should return to the Αμαξοστάσιο
before time Dmax (latest possible arrival at the Αμαξοστάσιο).

IloRangeMatrix2x2 Constraint_10ik(env,0);
for (i = 1; i < imax; i++){
    IloRangeArray Constraint_10k(env, 0);
    for (k = 0; k < kmax; k++){
        IloExpr expr(env, 0);
        for (j = 0; j < 1; j++){
            if (i != j){
                expr = Tik[i][k] + s[i] +
time[i][j][k] - M + M*Xijk[i][j][k] - Dmax;
            }
        }
        char Con_10[60];
        sprintf(Con_10, "Constraint_10ik(%d),(k%d)", i,
k);

        double LB = -IloInfinity, UB = 0;
        IloRange Constraint_10(env, LB, expr, UB, Con_10);
        expr.end();
        model.add(Constraint_10);
        Constraint_10k.add(Constraint_10);
    }
    Constraint_10ik.add(Constraint_10k);
}

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

```

```

//-----Objective Function 32-----
//Minimizing total cost taking into account vehicle utilization costs along
with travel costs

IloExpr expr1(env);
IloExpr expr2(env);
IloExpr expr3(env);

for (i = 0; i < imax; i++){
    for (j = 0; j < jmax; j++){
        for (k = 0; k < kmax; k++){
            expr1 += c[i][j][k] * Xijk[i][j][k];
        }
    }
}
model.add(IloMinimize(env, expr1));
expr1.end();

//-----Objective Function-----
//-----

cplex.extract(model);
cplex.exportModel("[...path...]/model.lp");

cplex.solve();

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

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

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

for (k = 0; k < kmax; k++){
    for (i = 1; i < imax; i++){
        float g = cplex.getValue(Tik[i][k]);
        if (g != 0) cout << "Tik" << "(" << i << ","
<< k << ")" << "=" << g << endl;
    }
}

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

```

```

                                float g =
cplex.getValue(Pijk[i][j][k]);
                                if (g != 0) cout << "Pijk" << "(" << i
<< "," << j << "," << k << ")" << "=" << g << endl;
                                }
                                }

}
catch ( IloException& e){
    cerr << "concert exception caught:"<<e<<endl;
}
catch (...){
    cerr<<"Unknown exception caught" <<endl;
}
env.end();
return 0;
}

```