

ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΥΝΔΥΑΣΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΜΗ
ΕΠΟΠΤΕΥΟΜΕΝΗΣ ΜΑΘΗΣΗΣ-ΣΥΣΤΑΔΟΠΟΙΗΣΗΣ
ΚΑΙ ΑΝΑΖΗΤΗΣΗΣ ΜΕΤΑΒΛΗΤΗΣ ΓΕΙΤΝΙΑΣΗΣ ΓΙΑ
ΤΗΝ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ
ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΗ
ΧΩΡΗΤΙΚΟΤΗΤΑ

Διπλωματική Εργασία

του

Χρήστου Τρανίδη

Θεσσαλονίκη , 25/06/2020

ΣΥΝΔΥΑΣΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΜΗ ΕΠΟΠΤΕΥΟΜΕΝΗΣ
ΜΑΘΗΣΗΣ-ΣΥΣΤΑΔΟΠΟΙΗΣΗΣ ΚΑΙ ΑΝΑΖΗΤΗΣΗΣ ΜΕΤΑΒΛΗΤΗΣ
ΓΕΙΤΝΙΑΣΗΣ ΓΙΑ ΤΗΝ ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ
ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΠΕΡΙΟΡΙΣΜΕΝΗ
ΧΩΡΗΤΙΚΟΤΗΤΑ

Χρήστος Τρανίδης

Πτυχίο Γεωλογίας, ΑΠΘ, 2016

Μεταπτυχιακό Γεωφυσικής, ΑΠΘ, 2019

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής

Άγγελος Σιφαλέρας

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25/06/2020

Άγγελος Σιφαλέρας

Νικόλαος Σαμαράς

Δημήτριος

Χρήστου-Βαρκασέλης

.....

Τρανίδης Χρήστος

.....

Περίληψη

Προβλήματα βελτιστοποίησης προκύπτουν σε πολλές πρακτικές εφαρμογές. Ένα από τα πιο μελετημένα προβλήματα συνδυαστικής βελτιστοποίησης, είναι εκείνο της δρομολόγησης οχημάτων VRP, το οποίο ανήκει στην κατηγορία των NP-hard προβλημάτων. Το θέμα της παρούσας διατριβής, είναι το πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (CVRP).

Η επίλυση του προβλήματος επιτεύχθηκε σε δύο στάδια. Αρχικά, χρησιμοποιήθηκε ένας αλγόριθμος συσταδοποίησης δεδομένων, ο οποίος ανήκει στην κατηγορία των centroid based. Εφόσον σχηματίστηκαν οι συστάδες, κάθε μία από αυτές αντιμετωπίστηκε ως ένα ξεχωριστό πρόβλημα TSP, βελτιστοποιώντας την διαδρομή της κάθε μίας αντίστοιχα. Έπειτα, υπολογίστηκε το άθροισμα του κόστους της κάθε συστάδας. Τέλος, εφαρμόστηκε ένας αλγόριθμος μεταβλητής γειτονιάς αναζήτησης VNS μεταξύ των διαφορετικών συστάδων, έτσι ώστε να προκύψει μία ικανοποιητική τελική λύση, ελαχιστοποιώντας το τελικό συνολικό κόστος διαδρομών της κάθε συτάδας.

Τα αποτελέσματα της παραπάνω διαδικασίας, συγκρίνονται στο τέλος με τις καλύτερες λύσεις οι οποίες έχουν επιτευχθεί επίσημα για την κάθε περίπτωση δεδομένων CVRP.

Λέξεις Κλειδιά: βελτιστοποίηση, δρομολόγηση οχημάτων, συσταδοποίηση, κόστος διαδρομής, αλγόριθμος μεταβλητής γειτονιάς αναζήτησης

Abstract

Optimization problems arise in many practical applications. One of the most studied practical combinatorial optimization problem is that of vehicle routing (VRP), which belongs to the category of NP-hard. The subject of this dissertation is the vehicle routing with limited capacity, known as CVRP.

The solution to the problem was achieved in two stages. Initially, a centroid based clustering algorithm was applied to the data, to form clusters (routes). Once the clusters were formed, each one of them was treated as a separate TSP problem, optimizing the path of each one respectively. Then, the sum of the cost of each cluster was calculated. Finally, a variable neighborhood search metaheuristic was applied between the different clusters, in order to obtain a satisfactory final solution, minimizing the final total cost of routes of each group.

The results of the above process, are compared at the end with the best solutions that have been officially achieved for each case of CVRP data.

Keywords: optimisation, vehicle routing, clustering, route's cost, variable neighborhood search algorithm

Περιεχόμενα

1	Supply chain και logistics	2
1.1	Ορισμός του supply chain	2
1.2	Ορισμός των Logistics	4
1.2.1	Η διαδικασία μεταφοράς των προϊόντων	6
1.3	Διαχείριση της εφοδιαστικής αλυσίδας (Supply chain management)	7
2	Το πρόβλημα δρομολόγησης οχημάτων (VRP)	9
2.1	Ορισμός και ανάλυση του προβλήματος CVRP	12
2.1.1	Μαθηματική διατύπωση του προβλήματος CVRP	13
2.2	Μέθοδοι λύσης του προβλήματος CVRP	15
2.2.1	Construction heuristics	15
2.2.2	Improvement heuristics	19
2.2.3	Metaheuristics	25
2.2.4	Exact αλγόριθμοι	29
3	Συσταδοποίηση (Clustering)	31
3.1	Ορισμός και αλγόριθμοι συσταδοποίησης	31
3.1.1	Ο αλγόριθμος συσταδοποίησης (K-Means)	33
3.1.2	Ο αλγόριθμος συσταδοποίησης (DBSCAN)	35
3.2	Παραλλαγές αλγορίθμων συσταδοποίησης για την επίλυση του (C)VRP	36
3.3	Ο αλγόριθμος συσταδοποίησης της παρούσας διατριβής	37
3.3.1	Δημιουργία συστάδων	38
3.3.2	Βελτιστοποίηση συστάδων	39
4	Μεθοδολογία παρούσας διατριβής για την επίλυση του CVRP	40
5	Αποτελέσματα και ερμηνεία	42
5.1	Αρχικοποίηση παραμέτρων	42
5.2	Κατηγοριοποίηση περιπτώσεων και πειραματική αρχικοποίηση παραμέτρων VNS	43
5.2.1	B-n52-k7 & B-n57-k9	44
5.2.2	A-n60-k9 & B-n78-k10	45
5.2.3	A-n80-k10 & E-n101-k14	47
5.3	Πίνακες αποτελεσμάτων	50

5.3.1	Σετ δεδομένων A	50
5.3.2	Σετ δεδομένων B	51
5.3.3	Σετ δεδομένων E	52
5.4	Σχήματα λύσεων	53
5.4.1	A-n62-k8	53
5.4.2	A-n80-k10	55
5.4.3	B-n50-k8	57
5.4.4	B-n68-k9	59
5.4.5	A-n78-k10	61
5.4.6	E-n101-k8	63
5.4.7	E-n101-k14	65
6	Συμπεράσματα και προτάσεις	67
	Αναφορές	68

Κατάλογος Σχημάτων

1	Στάδια του supply chain, [8].	3
2	Κύκλος δραστηριοτήτων στον τομέα των Logistics, [11].	5
3	Παράδειγμα βελτιστοποίησης της διαδικασίας μεταφοράς προϊόντων, [16].	7
4	Σύνολο διεργασιών στην διαχείριση μίας εφοδιαστικής αλυσίδας, [9].	8
5	Βασικές αποφάσεις στην διαχείριση της εφοδιαστικής αλυσίδας, [18].	8
6	Ένα απλό παράδειγμα του προβλήματος VRP, [23].	10
7	Σχέσεις μεταξύ των διαφορετικών υποπροβλημάτων της κατηγορίας VRP, εικόνα από [22].	10
8	Γράφος και παράμετροι του προβλήματος CVRP, εικόνα από [28].	13
9	Βήματα εφαρμογής αλγορίθμου Sequential Insertion algorithm (SI), εικόνα από [33].	16
10	Ψευδοκώδικας αλγορίθμου nearest neighbour Σχήμα από [35].	18
11	Αναπαράσταση ανταλλαγής σημείων (exchange heuristic), [36].	20
12	Διαδικασία ελέγχου για μετακίνηση ενός πελάτη σε άλλη διαδρομή, [36].	21
13	Αναπαράσταση του relocate heuristic, [36].	22
14	Παράδειγμα εφαρμογής του αλγορίθμου 2 - Opt, εικόνα από [37].	23
15	Βήματα εφαρμογής του αλγορίθμου 2 - Opt, εικόνα από [38].	23
16	Διάγραμμα το οποίο τονίζει την παρουσία τοπικών ελαχίστων και του ολικού ελαχίστου σε μία συνάρτηση βελτιστοποίησης, [36].	25
17	Βήματα εφαρμογής του αλγορίθμου simulated annealing, εικόνα από [36].	27
18	Παράδειγμα συσταδοποίησης μίας ομάδας δεδομένων, εικόνα από [55].	31
19	Παράδειγμα hard-soft clustering, [55].	32
20	Παράδειγμα συσταδοποίησης με την χρήση του K-Means, [55].	34
21	Παράδειγμα συσταδοποίησης με την χρήση του DBSCAN, εικόνα από [57].	36
22	Παράδειγμα απεικόνισης των κεντρών (GC) των συστάδων, [61].	38
23	Ψευδοκώδικας δημιουργίας συστάδων, [61].	39
24	Σημείο συστάδας το οποίο πρέπει να μετακινηθεί σε άλλη, [61].	39
25	Ψευδοκώδικας βελτιστοποίησης συστάδων, [61].	40
26	Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση B-n52-k7, για τιμές $k = 5 \dots 20$	44

27	Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση B-n57-k9 , για τιμές $k = 5...20$	45
28	Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση A-n60-k9 , για τιμές $k = 5...20$	46
29	Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση B-n78-k10 , για τιμές $k = 5...20$	47
30	Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση A-n80-k10 , για τιμές $k = 5...20$	48
31	Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση E-n101-k14 , για τιμές $k = 5...20$	49
32	<i>InstanceA</i> – n62 – k8, $k_{max} = 5$, $dev = 3.72\%$	53
33	<i>InstanceA</i> – n62 – k8, $k_{max} = 19$, $dev = 2.69\%$	54
34	<i>InstanceA</i> – n80 – k10, $k_{max} = 5$, $dev = 3.13\%$	55
35	<i>InstanceA</i> – n80 – k10, $k_{max} = 19$, $dev = 4.78\%$	56
36	<i>InstanceB</i> – n50 – k8, $k_{max} = 5$, $dev = 1.35\%$	57
37	<i>InstanceB</i> – n50 – k8, $k_{max} = 19$, $dev = 1.41\%$	58
38	<i>InstanceB</i> – n68 – k9, $k_{max} = 5$, $dev = 3.26\%$	59
39	<i>InstanceB</i> – n68 – k9, $k_{max} = 19$, $dev = 2.56\%$	60
40	<i>InstanceB</i> – n78 – k10, $k_{max} = 5$, $dev = 4.3\%$	61
41	<i>InstanceB</i> – n78 – k10, $k_{max} = 19$, $dev = 3.46\%$	62
42	<i>InstanceE</i> – n101 – k8, $k_{max} = 5$, $dev = 6.29\%$	63
43	<i>InstanceE</i> – n101 – k8, $k_{max} = 19$, $dev = 7.21\%$	64
44	<i>InstanceE</i> – n101 – k14, $k_{max} = 5$, $dev = 7.78\%$	65
45	<i>InstanceE</i> – n101 – k14, $k_{max} = 19$, $dev = 8.7\%$	66

Κατάλογος Πινάκων

1	Σετ δεδομένων A	50
2	Σετ δεδομένων B	51
3	Σετ δεδομένων E	52

Εισαγωγή

Η βελτιστοποίηση της διαδικασίας μεταφοράς προϊόντων, είναι υψίστης σημασίας καθώς συνδέεται άμεσα με την διαχείριση της εφοδιαστικής αλυσίδας, αποτελώντας ίσως τον πιο σημαντικό τομέα αυτής. Αυτό γίνεται εύκολα αντιληπτό, αν αναλογιστεί κανείς πως η ελαχιστοποίηση του συνολικού κόστους μεταφοράς, αποφέρει στην επιχείρηση πολύ μεγαλύτερα κέρδη, καθώς ο τομέας των μεταφορών αντιπροσωπεύει περίπου το ήμισυ των συνολικών δαπανών μίας εφοδιαστικής αλυσίδας. Την λύση στο συγκεκριμένο πρόβλημα, επιχειρούν να δώσουν διάφοροι αλγόριθμοι δρομολόγησης οχημάτων. Η παρούσα εργασία, αφορά το πρόβλημα δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (CVRP).

Στο συγκεκριμένο πρόβλημα, ένας αριθμός πελατών εξυπηρετείται από έναν αριθμό οχημάτων που ξεκινούν και ολοκληρώνουν την διαδρομή τους από μία αφετηρία. Ο στόχος είναι να βρεθούν διαδρομές με το χαμηλότερο δυνατό συνολικό κόστος. Σε κάθε πελάτη ανατίθεται μόνο ένα όχημα, ενώ το μέγιστο όριο χωρητικότητας του κάθε οχήματος δεν πρέπει να ξεπεραστεί. Το πρόβλημα CVRP ανήκει στην κατηγορία των NP-hard προβλημάτων, επειδή όσο μεγαλύτερο γίνεται το σύνολο των δεδομένων, τόσο πιο δύσκολη γίνεται η ανεύρεση της βέλτιστης λύσης σε πολυωνυμικό χρόνο.

Διάφοροι αλγόριθμοι επίλυσης του παραπάνω προβλήματος έχουν προταθεί κατά καιρούς. Στο άρθρο [1], οι συγγραφείς αναφέρονται στην ανάπτυξη exact αλγορίθμων για το πρόβλημα CVRP, ενώ στα άρθρα [2] και [3], οι συγγραφείς αναφέρονται στην επίλυση του προβλήματος χρησιμοποιώντας μεθευρετικές μεθόδους, (ant colony) μέθοδο και χρήση γενετικού αλγορίθμου αντίστοιχα.

Ο τρόπος με τον οποίον επιλύθηκε το πρόβλημα στην παρούσα εργασία χωρίζεται σε δύο φάσεις. Κατά την διάρκεια της πρώτης φάσης εφαρμόστηκε ένας αλγόριθμος συσταδοποίησης, έτσι ώστε να κατασκευαστούν οι αρχικές διαδρομές για την κάθε συστάδα, ενώ για να επιτευχθεί η τελική λύση, εφαρμόστηκε ο μεθευρετικός αλγόριθμος μεταβλητής γειτονιάς αναζήτησης (VNS) [4]. Ο συγκεκριμένος αλγόριθμος έχει χρησιμοποιηθεί ευρέως για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης. Η βασική ιδέα είναι η συστηματική αλλαγή στη γειτονιά της τοπικής αναζήτησης, έτσι ώστε ο αλγόριθμος να μην παγιδεύεται σε τοπικά ελάχιστα της αντικειμενικής συνάρτησης.

1 Supply chain και logistics

1.1 Ορισμός του supply chain

Στον χώρο των επιχειρήσεων και της οικονομίας, η αλυσίδα εφοδιασμού (supply chain) ορίζεται ως ένα σύστημα οργανισμών, ανθρώπων, δραστηριοτήτων, πληροφοριών και πόρων που σχετίζονται με τη μετακίνηση ενός προϊόντος ή μίας υπηρεσίας από τον προμηθευτή στον πελάτη. Οι δραστηριότητες της περιλαμβάνουν την μετατροπή των φυσικών πόρων, των πρώτων υλών και των συστατικών τους σε προϊόν το οποίο παραδίδεται στον τελικό πελάτη [5].

Ωστόσο, μία αλυσίδα εφοδιασμού περιλαμβάνει όχι μόνο τον κατασκευαστή και τους προμηθευτές του, αλλά επίσης τους μεταφορείς, τις αποθήκες, τους λιανοπωλητές και τους ίδιους τους καταναλωτές. Περιλαμβάνει την διαδικασία δημιουργίας νέων προϊόντων, την διαφήμιση αυτών, την χρηματοδότηση και την τελική εξυπηρέτηση των πελατών. Το Σχήμα 1 απεικονίζει μια γενική αλυσίδα εφοδιασμού στο πλαίσιο του συνολικού δικτύου αλυσίδας εφοδιασμού. Κάθε επιχείρηση ανήκει σε τουλάχιστον μία αλυσίδα εφοδιασμού: δηλαδή συνήθως έχει πολλαπλούς προμηθευτές και πελάτες [6].

Μια αλυσίδα εφοδιασμού μπορεί να περιλαμβάνει διάφορα στάδια, όπως:

- πελάτες
- λιανοπωλητές
- διανομείς
- κατασκευαστές
- προμηθευτές υλικών και πρώτων υλών

Κάθε στάδιο στο Σχήμα 1 δεν είναι απαραίτητο ότι θα υπάρχει σε μια αλυσίδα εφοδιασμού. Ο κατάλληλος σχεδιασμός αυτής εξαρτάται τόσο από τις ανάγκες του πελάτη όσο και από τους ρόλους που διαδραματίζουν τα σχετικά στάδια [7].

Το σημαντικότερο πρόβλημα που απασχολεί τα τελευταία χρόνια τους υπεύθυνους της εφοδιαστικής αλυσίδας, είναι η μείωση του κόστους και η εξυπηρέτηση των πελατών όσο καλύτερα και πιο γρήγορα γίνεται. Οι κίνδυνοι για τους οργανισμούς δεν προέρχονται μόνο από τους εσωτερικούς ή εξωτερικούς προμηθευτές τους αλλά από αποτυχίες σε ολόκληρο το προμηθευτικό περιβάλλον, συμπεριλαμβανομένης της μελλοντικής διανομής ή της προμήθειας. Είναι ύψιστης



Σχήμα 1: Στάδια του supply chain, [8].

σημασίας για τον λόγο αυτόν, εκείνοι οι οποίοι ασχολούνται και ελέγχουν αυτήν, όχι μόνο να έχουν πλήρη κατανόηση του τρόπου λειτουργίας της, αλλά και του ρυθμιστικού περιβάλλοντος στο οποίο λειτουργεί η οργάνωσή τους.

Ο κόσμος αλλάζει με απίστευτη ταχύτητα και η ανάγκη για τους οργανισμούς και τις αλυσίδες εφοδιασμού τους να εξελιχθούν με μία νέα νοοτροπία είναι ιδιαίτερα σημαντική και πρέπει να γίνει άμεσα. Αυτό οφείλεται στο γεγονός πως τα παλιότερα χρόνια, οι αλυσίδες εφοδιασμού είχαν κυρίως τοπικό χαρακτήρα, ενώ την σημερινή εποχή πολλές από αυτές καλύπτουν ανάγκες πελατών παγκοσμίως, αυξάνοντας έτσι εκθετικά τις ευθύνες τους, αλλά και τις ανάγκες των πελατών τις οποίες πρέπει να καλύψουν με τον καλύτερο δυνατό τρόπο.

Χαρακτηριστικό παράδειγμα αποτελούν ορισμένες επιχειρήσεις οι οποίες υπέστησαν τις συνέπειες της κακής διαχείρισης της αλυσίδας εφοδιασμού, ενώ άλλες οι οποίες λειτουργούν με διαφορετική νοοτροπία, όπως οι Amazon, Wal-Mart, Zara έχουν ξεπεράσει την ανταγωνιστικότητα κατά πολύ [9].

Ένας από τους παράγοντες που χαρακτηρίζουν μια σωστά διαχειρίσιμη εφοδιαστική αλυσίδα, είναι ο τομέας μεταφορών των προϊόντων. Ένα αναποτελεσματικό σύστημα μεταφοράς διαταράσσει την αλυσίδα εφοδιασμού και επιβάλλει σοβαρές ζημιές σε αυτήν. Ο τομέας των μεταφορών ανήκει σε μία ειδική κατηγορία διαχείρισης της εφοδιαστικής αλυσίδας, η οποία ονομάζεται logistics.

1.2 Ορισμός των Logistics

Θεωρητικά, δεν υπάρχει κάποιος ορισμός για τα Logistics ο οποίος να είναι επίσημα αποδεκτός και μοναδικός, όσον αφορά την πλήρη έννοια αυτού. Γενικά, πολλοί είναι οι συγγραφείς εκείνοι, οι οποίοι τονίζουν τις διάφορες πτυχές της έννοιας του τομέα στον χώρο των επιχειρήσεων και της οικονομίας. Παράλληλα, διαφορετικοί φαίνεται να είναι και οι ορισμοί της λέξης στην κάθε ήπειρο [10].

Πιο συγκεκριμένα, ο πιο διαδεδομένος όρος των Logistics στις Ηνωμένες Πολιτείες Αμερικής, σύμφωνα με το Αμερικάνικο Συμβούλιο διαχείρισης των Logistics είναι ο εξής:

Ορίζεται ως η διαδικασία σχεδιασμού, υλοποίησης και ελέγχου της αποτελεσματικής ροής και αποθήκευσης αγαθών, υπηρεσιών και συναφών πληροφοριών από το σημείο προέλευσης έως το σημείο κατανάλωσης με σκοπό την συμμόρφωση προς τις απαιτήσεις των πελατών.

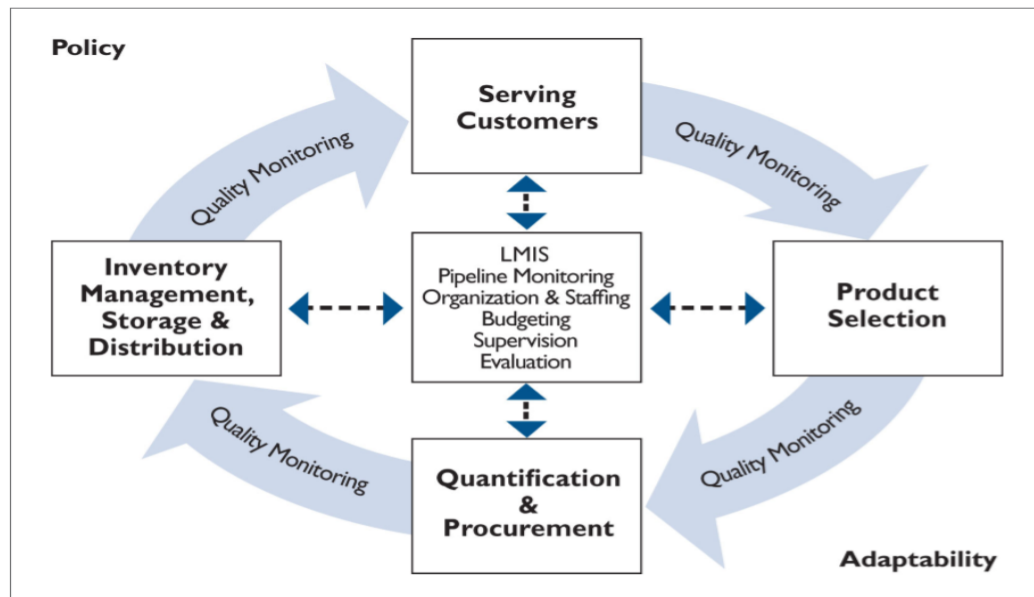
Αντίθετα ο επίσημος ορισμός της Ευρωπαϊκής ένωσης για τα Logistics είναι ο εξής:

Αφορά την οργάνωση, τον προγραμματισμό, τον έλεγχο και την εκτέλεση της ροής αγαθών από τον τόπο παραγωγής (αγορά) έως την διαδικασία διανομής στον τελικό καταναλωτή, ενώ έχει ως στόχο να ικανοποιήσει τις απαιτήσεις της αγοράς με την ελάχιστη δυνατή δέσμευση και το ελάχιστο δυνατό κόστος [10].

Αναλύοντας λοιπόν κανείς τους διάφορους ορισμούς οι οποίοι ισχύουν για τον τομέα των Logistics, συμπεραίνεται ότι οι στόχοι αυτού συνοψίζονται με τον ακόλουθο τρόπο:

- είναι μια διαδικασία φυσικής ροής αγαθών / υπηρεσιών και πληροφοριών.
- είναι μια έννοια ολοκληρωμένης διαχείρισης αγαθών / υπηρεσιών και ροής πληροφοριών.
- είναι ένας διεπιστημονικός τομέας, όπου το αντικείμενο της έρευνας είναι οι κανονικότητες και τα φαινόμενα που συμβαίνουν κατά τη διάρκεια της ροής αγαθών / αλυσίδα.

Όσον αφορά τώρα την διαχείριση του συγκεκριμένου τομέα, του τομέα των Logistics, αποτελείται από τις δραστηριότητες οι οποίες παρατηρούνται στο Σχήμα 2 [11]. Παρατηρείται ότι, κάθε δραστηριότητα όπως εξυπηρέτηση πελατών, επιλογή προϊόντων, ποσοτικοποίηση και προμήθεια και διαχείριση αποθεμάτων - εξαρτάται και επηρεάζεται από τις άλλες δραστηριότητες. Τέλος, οι δραστηριότητες στο κέντρο του κύκλου αντιπροσωπεύουν τις λειτουργίες υποστήριξης διαχείρισης που ενημερώνουν και επηρεάζουν τα άλλα στοιχεία γύρω από τον κύκλο δραστηριοτήτων [11].



Σχήμα 2: Κύκλος δραστηριοτήτων στον τομέα των Logistics, [11].

Η αποτελεσματική και αποδοτική υλοποίηση των στόχων του τομέα των Logistics είναι δυνατή χάρη στις ακόλουθες δραστηριότητες [10]:

- κυκλοφορία και μεταφορά
- αποθήκευση
- βιομηχανική συσκευοποίηση
- χειρισμός υλικών
- έλεγχος αποθεμάτων
- καθορισμός παραγγελιών
- πρόβλεψη ζήτησης
- προγραμματισμός παραγωγής
- αγορά
- εξυπηρέτηση πελατών σε κατάλληλο επαγγελματικό επίπεδο
- αποθήκες και τοποθεσία εγκατάστασης αυτών
- συλλογή και διάθεση αποβλήτων

1.2.1 Η διαδικασία μεταφοράς των προϊόντων

Οι μεταφορές έχουν τεράστιες οικονομικές, κοινωνικές και περιβαλλοντικές επιπτώσεις. Πιο συγκεκριμένα στο άρθρο [12] ο συγγραφέας αναφέρει πως το έτος 2009 το 7 % του ακαθάριστου εθνικού προϊόντος της Ευρώπης προήλθε από τις μεταφορές, ένας τομέας ο οποίος παράλληλα ευθυνόταν για το 5 % των θέσεων εργασίας στην Ευρώπη το ίδιο έτος. Επιπλέον, το 0.6 % του ΑΕΠ το 2009 προήλθε από τους φόρους στις μεταφορές ενώ μεγάλο μέρος των φόρων στην ενέργεια προήλθε από τους φόρους στην κατανάλωση καυσίμων, όπου το 60 % της παγκόσμιας κατανάλωσης αυτών, γίνεται εξαιτίας των μεταφορών [13].

Η σχέση μεταξύ όλων των στοιχείων που συνθέτουν μία εφοδιαστική αλυσίδα είναι ιδιαίτερα σημαντική, καθώς αυτά αλληλεπιδρούν μεταξύ τους για έναν κοινό τελικό σκοπό, ο οποίος είναι η εξυπηρέτηση των πελατών με τον καλύτερο δυνατό τρόπο. Σε διαφορετική περίπτωση, παρατηρούνται επιπλοκές στον τρόπο λειτουργίας της αλυσίδας, οι οποίες έχουν ως συνέπεια την χαμηλή παραγωγικότητα [14]. Η μεταφορά των προϊόντων είναι ένας από τους σημαντικότερους παράγοντες της εφοδιαστικής αλυσίδας ο οποίος όπως αναφέρθηκε παραπάνω ανήκει στην κατηγορία των Logistics.

Τα έξοδα μεταφοράς περιλαμβάνουν ένα κύριο μέρος των Logistics και συνεπώς του τελικού κόστους των προϊόντων. Η ανάλυση των μεταφορών για τις επιχειρήσεις έδειξε ότι αποτελούν σημαντικό μέρος των δαπανών για αυτές. Πιο συγκεκριμένα, ο τομέας των μεταφορών αντιπροσωπεύει περίπου το ήμισυ των συνολικών δαπανών μίας εφοδιαστικής αλυσίδας. Επομένως είναι ζωτικής σημασίας να επιδιώκεται ο βέλτιστος προγραμματισμός των διαδικασιών οι οποίες αφορούν την μεταφορά των προϊόντων (πχ Σχήμα 3), όπως είναι για παράδειγμα η δρομολόγηση των οχημάτων τα οποία διανέμουν τα τελικά προϊόντα. Τέτοιες διαδικασίες εφαρμόζονται ευρέως στο εξωτερικό σε πολλούς τομείς της διανομής πχ βέλτιστος προγραμματισμός δρομολόγησης για διανομή εφημερίδων, για παράδοση γάλατος, δρομολόγησης απορριμματοφόρων κ.α [15].



Σχήμα 3: Παράδειγμα βελτιστοποίησης της διαδικασίας μεταφοράς προϊόντων, [16].

1.3 Διαχείριση της εφοδιαστικής αλυσίδας (Supply chain management)

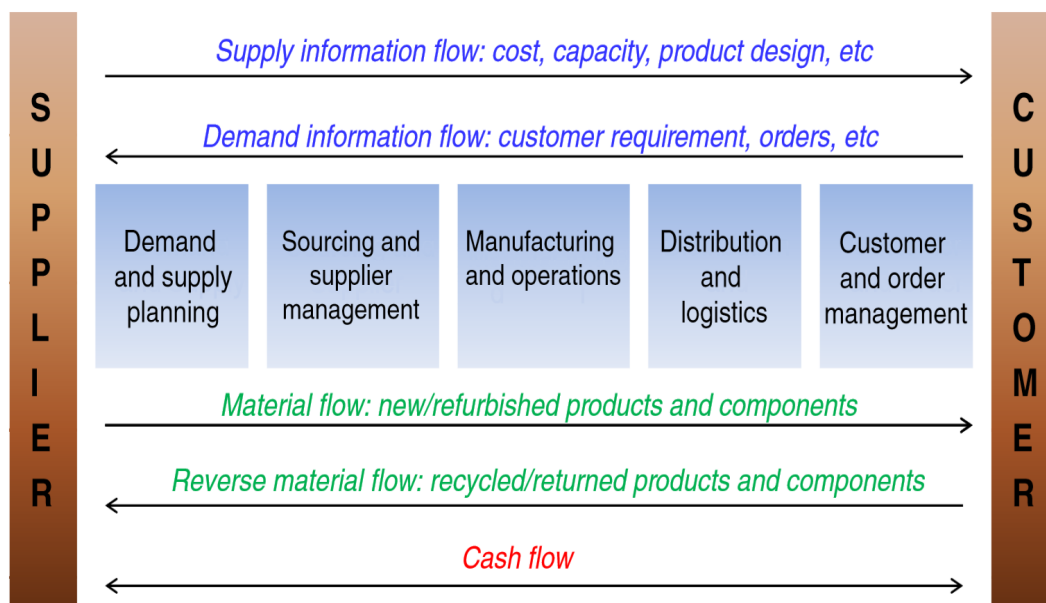
Σύμφωνα με τον συγγραφέα του άρθρου [9], η έννοια διαχείρισης της εφοδιαστικής αλυσίδας πρωτοεμφανίστηκε το 1950. Εκείνη την περίοδο, ορισμένοι ερευνητές προσπαθούσαν να βρουν τον βέλτιστο τρόπο με τον οποίον θα μπορούσε να γίνει η καταγραφή των εμπορευμάτων στις διάφορες επιχειρήσεις. Η διαχείριση της αλυσίδας εφοδιασμού είναι μία από τις πιο σημαντικές πτυχές της επιχειρηματικής δραστηριότητας.

Ο επίσημος ορισμός αυτής, ο οποίος έχει διατυπωθεί από το Συμβούλιο επαγγελματιών διαχείρισης της εφοδιαστικής αλυσίδας (Council of Supply Chain Management Professionals (CSCMP)) είναι ο εξής:

η διαχείριση της εφοδιαστικής αλυσίδας, περιλαμβάνει τον προγραμματισμό και τη διαχείριση όλων των δραστηριοτήτων που εμπλέκονται στην προμήθεια και σε όλες τις δραστηριότητες διαχείρισης των Logistics. Περιλαμβάνει επίσης το συντονισμό και τη συνεργασία μεταξύ των επικοινωνιακών διαύλων, οι οποίοι μπορούν να είναι προμηθευτές, μεσάζοντες και πελάτες. Στην ουσία, η διαχείριση της εφοδιαστικής αλυσίδας ενσωματώνει τη διαχείριση της προσφοράς και της ζήτησης εντός και μεταξύ των εταιρειών.

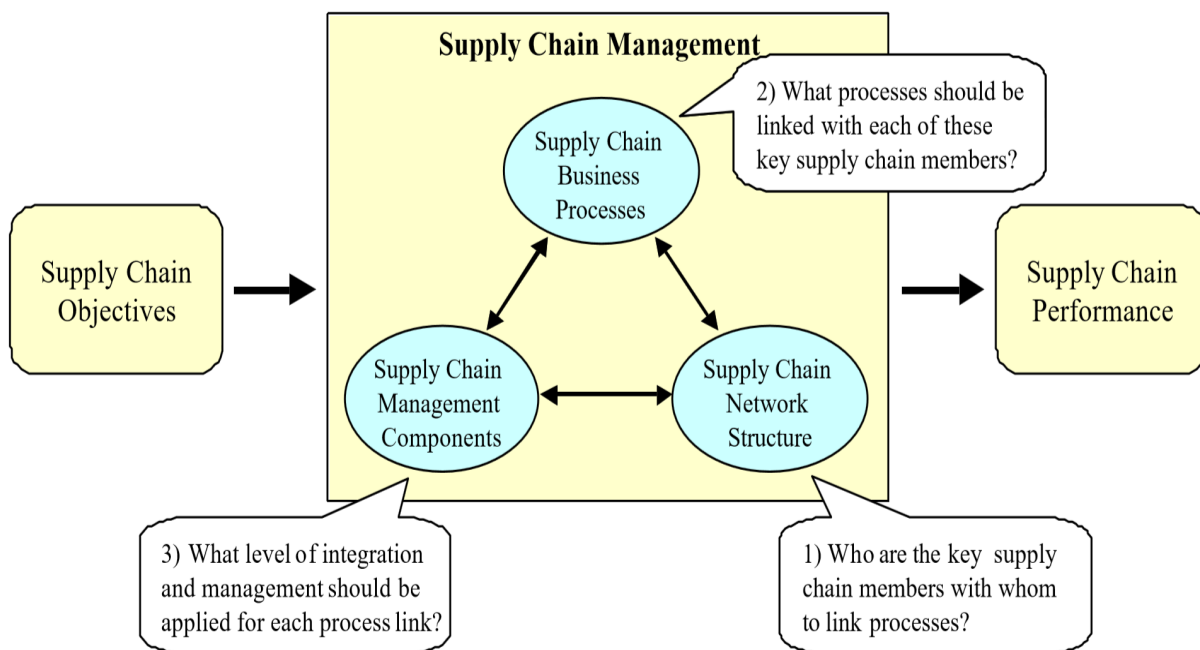
Αυτή η ολοκληρωμένη διαδικασία λοιπόν περιλαμβάνει τον σχεδιασμό, την προμήθεια, τον προγραμματισμό και την πρόβλεψη προϊόντων, την παραγωγή, τη διανομή, τη συμπλήρωση και

την υποστήριξη μετά την πώληση (Σχήμα 4).



Σχήμα 4: Σύνολο διεργασιών στην διαχείριση μίας εφοδιαστικής αλυσίδας, [9].

Παράλληλα, οι συγγραφείς στο άρθρο [17] αναφέρουν τρία σημαντικά κλειδιά στην διαχείριση της εφοδιαστικής αλυσίδας, που συνοψίζονται στο Σχήμα 5. Τονίζουν επίσης την ανάγκη να σχεδιαστεί και να διαχειριστεί η αλυσίδα εφοδιασμού με διάφορα βήματα.



Σχήμα 5: Βασικές αποφάσεις στην διαχείριση της εφοδιαστικής αλυσίδας, [18].

2 Το πρόβλημα δρομολόγησης οχημάτων (VRP)

Το πρόβλημα δρομολόγησης οχημάτων VRP αποτελεί ένα από τα πιο πολυσυζητημένα θέματα βελτιστοποίησης μεταξύ των ερευνητών λόγω της σύνδεσής του με σημαντικά προβλήματα στον πραγματικό κόσμο [19] και ανήκει στην κατηγορία των NP-hard προβλημάτων [20]. Πιο συγκεκριμένα, ορισμένα προβλήματα ανήκουν στην κλάση P, όπου η λύση αυτών μπορεί να επιτευχθεί σε πολυωνυμικό χρόνο, αποφεύγοντας την εξαντλητική αναζήτηση. Εκτός από αυτά όμως, υπάρχουν και τα προβλήματα βελτιστοποίησης τα οποία ανήκουν στην κλάση NP, όπου όλες οι προσπάθειες να αποφευχθεί η εξαντλητική αναζήτηση έχει μέχρι στιγμής αποτύχει [21]. Επομένως δεν υπάρχει κάποιος αλγόριθμος ο οποίος να επιλύει τα συγκεκριμένα προβλήματα σε πολυωνυμικό χρόνο. Η λύση των προβλημάτων τα οποία ανήκουν στην τελευταία κατηγορία ονομάζεται NP-hard.

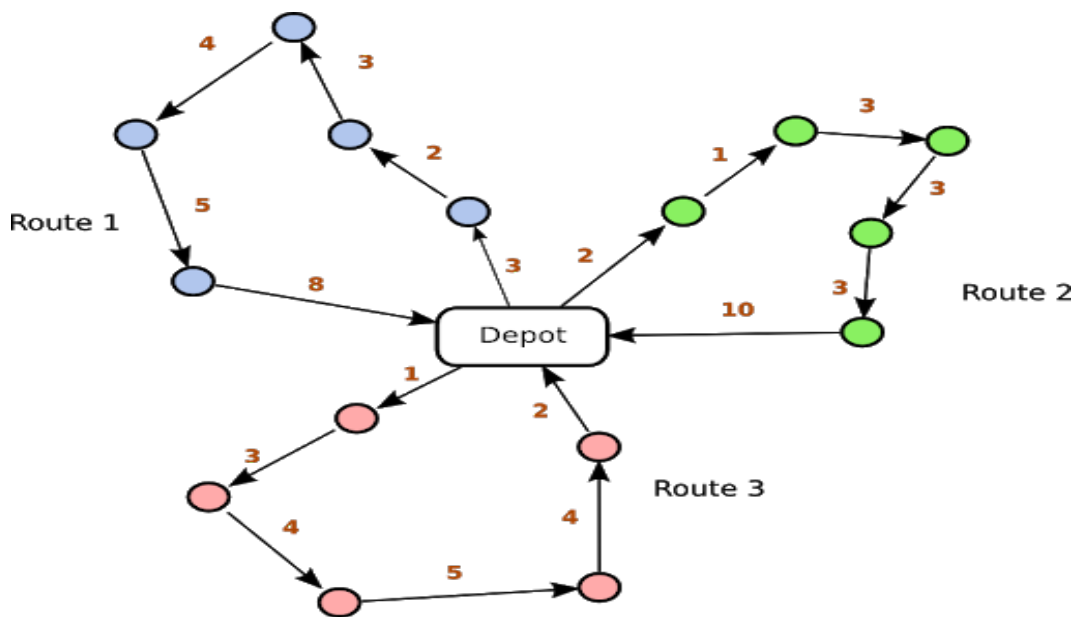
Σημειώνεται επίσης ότι:

- εντατικές προσπάθειες για ταχεία επίλυση NP προβλημάτων έχουν μέχρι στιγμής αποτύχει.
- αυτό οδηγεί στην υποψία ότι οποιοσδήποτε αλγόριθμος επίλυσης των προβλημάτων αυτών απαιτεί χρόνο της τάξης $O(2^n)$.
- ωστόσο, αυτό δεν έχει ακόμα αποδειχθεί.

Όσον αφορά τώρα την περιγραφή του προβλήματος, διατυπώνεται ως εξής [19]:

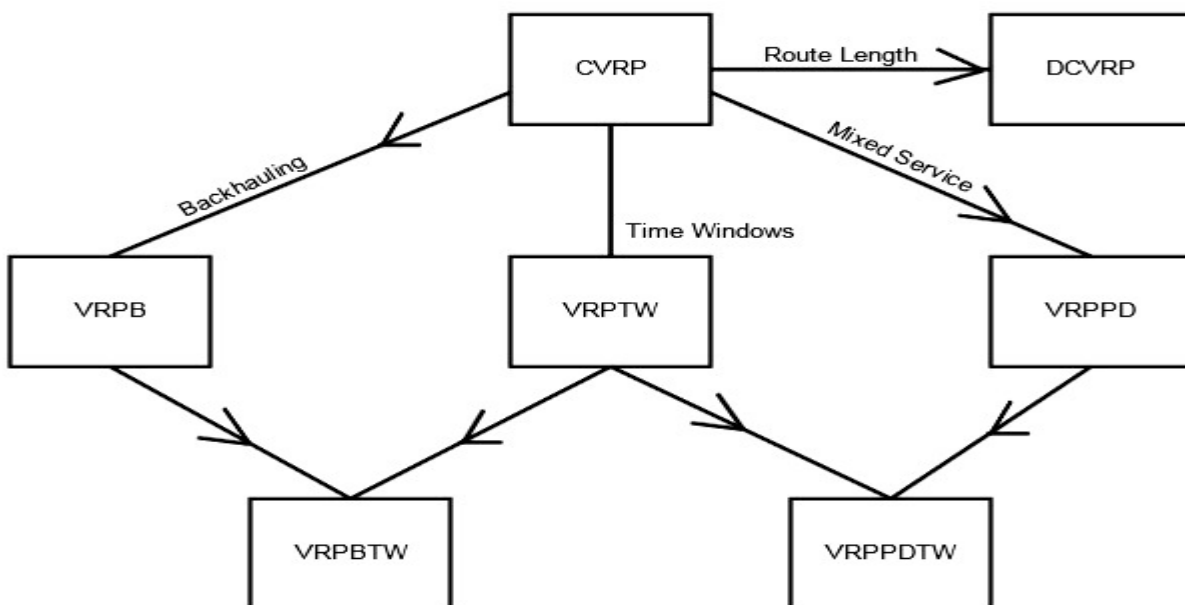
Το πρόβλημα δρομολόγησης οχημάτων VRP αναφέρεται σε μια σειρά προβλημάτων στα οποία πρέπει να προσδιοριστεί ένα σύνολο διαδρομών για ένα σύνολο οχημάτων, τα οποία βρίσκονται σε ένα (ή περισσότερα) σημεία αφετηρίας και πρέπει να εξυπηρετήσουν έναν αριθμό γεωγραφικά διασκορπισμένων πελατών. Στόχος είναι η παράδοση αγαθών στους πελάτες, έχοντας διανύσει την μικρότερη δυνατή απόσταση στον λιγότερο χρόνο, ξεκινώντας σε κάθε περίπτωση από την αφετηρία και επιστρέφοντας πάλι σε αυτήν (Σχήμα 6).

Για να γνωρίζουμε το συνολικό κόστος κάθε διαδρομής, πρέπει να γνωρίζουμε το κόστος και τον χρόνο διαδρομής μεταξύ κάθε πελάτη και αφετηρίας. Για να γίνει αυτό, το αρχικό γράφημα μετατρέπεται σε ένα, όπου το κάθε σημείο αντιπροσωπεύει έναν πελάτη, με ξεχωριστό σύμβολο σημειώνεται η αφετηρία, ενώ τα τόξα είναι οι δρόμοι μεταξύ τους. Το κόστος σε κάθε τόξο είναι το χαμηλότερο κόστος μεταξύ των δύο σημείων του αρχικού οδικού δικτύου [22].



Σχήμα 6: Ένα απλό παράδειγμα του προβλήματος VRP, [23].

Τέλος, αναφέρεται ότι υπάρχουν διάφορες παραλλαγές του προβλήματος δρομολόγησης οχημάτων (Σχήμα 7), κάθε μία από τις οποίες λειτουργεί με διαφορετικό τρόπο και έχει διαφορετικό τελικό στόχο. Μερικές από τις πιο γνωστές παραλλαγές, είναι οι παρακάτω [22]:



Σχήμα 7: Σχέσεις μεταξύ των διαφορετικών υποπροβλημάτων της κατηγορίας VRP, εικόνα από [22].

- Vehicle Routing Problem with Profits (VRPP)
- Vehicle Routing Problem with Pickup and Delivery (VRPPD)
- Vehicle Routing Problem with Time Windows (VRPTW)
- Capacitated Vehicle Routing Problem: CVRP or CVRPTW
- Vehicle Routing Problem with Multiple Trips (VRPMT)
- Open Vehicle Routing Problem (OVRP)
- Inventory Routing Problem (IRP)

Όπως προαναφέρθηκε, κάθε κατηγορία προβλήματος VRP, έχει και διαφορετικό τελικό στόχο. Μερικοί από τους πιο συνηθισμένους στόχους επίλυσης ενός προβλήματος δρομολόγησης οχημάτων είναι:

1. ελαχιστοποίηση του συνολικού κόστους μεταφοράς, με βάση την συνολική απόσταση που διανύθηκε αλλά και τα κόστη που αφορούν τα οχήματα και τους οδηγούς.
2. ελαχιστοποίηση του αριθμού των οχημάτων τα οποία εξυπηρετούν το σύνολο των πελατών.
3. ελαχιστοποίηση των μεταβολών στον χρόνο ταξιδιού καθώς και στο φορτίο των οχημάτων.
4. ελαχιστοποίηση των ποινών για εξυπηρέτηση χαμηλής ποιότητας.

Οι περιορισμοί οι οποίοι θα πρέπει να τηρούνται τώρα σε κάθε λύση του προβλήματος VRP, με βάση τον γράφο της Εικόνας 6, είναι οι ακόλουθοι:

1. κάθε σημείο (πελάτης) θα πρέπει να συνδέεται αποκλειστικά και μόνο με δύο σημεία (πελάτες).
2. ο αριθμός των οχημάτων τα οποία ξεκινούν από την αφετηρία, θα πρέπει να ισούται με τον αριθμό των οχημάτων που επιστρέφουν σε αυτήν.
3. κάθε διαδρομή είναι κυκλική, δηλαδή καταλήγει στο σημείο από όπου ξεκίνησε.
4. οι απαιτήσεις των πελατών της κάθε διαδρομής δεν πρέπει να υπερβαίνουν την χωρητική ικανότητα του εκάστοτε οχήματος.

2.1 Ορισμός και ανάλυση του προβλήματος CVRP

Το CVRP θεωρείται μία από τις κλασικές εκδοχές της γενικής κατηγορίας προβλημάτων δρομολόγησης οχημάτων VRP. Στην συγκεκριμένη κατηγορία, εισάγεται ο παράγοντας της χωρητικότητας των οχημάτων. Αυτό πρακτικά σημαίνει, ότι ο κάθε πελάτης έχει συγκεκριμένες απαιτήσεις οι οποίες για τα οχήματα μεταφράζονται σε χωρητικότητα. Το τελικό πρόγραμμα δρομολόγησης οχημάτων λοιπόν, θα πρέπει να αποτελείται από διαδρομές οι οποίες εξυπηρετούν όλους τους πελάτες με τον καλύτερο δυνατό τρόπο, διανύοντας την μικρότερη δυνατή απόσταση χωρίς όμως η χωρητικότητα του κάθε οχήματος στην εκάστοτε διαδρομή να μπορεί να υπερβεί την μέγιστη που έχει οριστεί.

Πιο συγκεκριμένα, σχηματίζεται ένας γράφος $G = (V, E)$, ακριβώς όπως στην Εικόνα 6, όπου $V = \{0, \dots, n\}$ ένα σύνολο σημείων (customers) και $E = (i, j) : i, j \in V, i < j$ ένα σύνολο ακμών, οι οποίες συνδέουν τα σημεία μεταξύ τους. Κάθε σημείο $i \in V \setminus \{0\}$ αντιπροσωπεύει έναν πελάτη με μη αρνητική απαίτηση q_i , ενώ το σημείο 0 αντιπροσωπεύει την αφετηρία (depot) των οχημάτων αλλά και το σημείο στο οποίο καταλήγουν και πάλι όλα τα οχήματα, αφού ολοκληρώσουν την διαδρομή τους. Σε κάθε ακμή τώρα αντιστοιχεί ένα κόστος c_{ij} , το οποίο αντιστοιχεί στο κόστος διαδρομής από το σημείο i στο σημείο j [24].

Τελικός στόχος λοιπόν ενός προβλήματος CVRP είναι το συνολικό κόστος διαδρομής όλων των οχημάτων να είναι το μικρότερο δυνατό, ενώ παράλληλα τα οχήματα θα πρέπει να εξυπηρετήσουν όλους τους πελάτες και να επιστρέψουν στην αφετηρία. Οι απαιτήσεις των πελατών στην κάθε διαδρομή δεν θα πρέπει να υπερβαίνει την χωρητικότητα των οχημάτων.

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, το πρόβλημα CVRP, ανήκει στην κατηγορία των NP-hard προβλημάτων. Έτσι, η χρήση των exact αλγορίθμων είναι χρήσιμη μόνο για λίγα σχετικά δεδομένα (λιγότερα από 60) [25]. Σε διαφορετική περίπτωση το κόστος επίλυσης του προβλήματος είναι πολύ μεγάλο. Για τον λόγο αυτόν, για την επίλυση των NP-hard προβλημάτων χρησιμοποιούνται συνήθως metaheuristics, τα οποία πολλές φορές είναι συνδυασμός construction και improvement heuristics [26]. Τα metaheuristics, τις περισσότερες φορές δίνουν λύσεις κατώτερης ποιότητας από εκείνη των exact αλγορίθμων. Ωστόσο, ο λόγος για τον οποίον χρησιμοποιούνται ευρέως, είναι ότι εξάγουν μια ικανοποιητική λύση σε ικανοποιητικό χρόνο. Παρατηρείται λοιπόν, ότι δύο παράγοντες επηρεάζουν κυρίως την μέθοδο που θα χρησιμοποιηθεί, η ποιότητα της τελικής λύσης και ο χρόνος επίλυσης του προβλήματος.

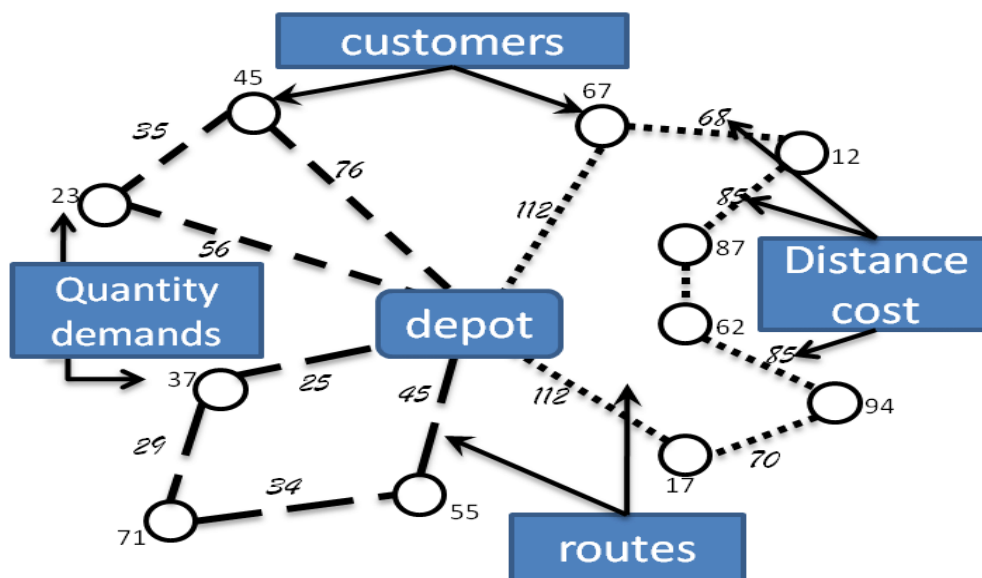
2.1.1 Μαθηματική διατύπωση του προβλήματος CVRP

Οι Fisher και Jaikumar πρότειναν την εξής μαθηματική εξήγηση για το πρόβλημα CVRP [27]: Αρχικά, ορίζονται οι παράμετροι του προβλήματος (Σχήμα 8):

- n : ορίζεται ο συνολικός αριθμός των πελατών οι οποίοι πρέπει να εξυπηρετηθούν. Με 0 σημειώνεται μόνο η αφετηρία της κάθε διαδρομής.
- m : ορίζεται ο συνολικός αριθμός των διαθέσιμων οχημάτων.
- Q : η χωρητικότητα του κάθε οχήματος.
- q_i : η απαίτηση του πελάτη i .
- d_{ij} : η απόσταση μεταξύ του πελάτη i και j .

Οι παράμετροι απόφασης:

- y_i^k : παίρνει την τιμή 1 εάν ο πελάτης i έχει ήδη εξυπηρετηθεί, 0 διαφορετικά.
- x_{ij}^k : παίρνει την τιμή 1 σε περίπτωση που το όχημα k έχει επισκεφτεί τον πελάτη i ακριβώς πριν από τον πελάτη j .



Σχήμα 8: Γράφος και παράμετροι του προβλήματος CVRP, εικόνα από [28].

Μαθηματική διατύπωση

$$\min \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}^k \quad (1)$$

$$\sum_{i=1}^n y_i^k \leq Q \quad \forall k = 1, 2, \dots, m \quad (2)$$

$$\sum_{k=1}^m y_i^k = 1 \quad \forall i = 1, 2, \dots, n \quad (3)$$

$$\sum_{k=1}^m y_i^k = m \quad \text{for } i = 0 \quad (4)$$

$$\sum_{i=0}^n x_{ij}^k = y_i^k \quad \forall j = 0, 1, \dots, n; \quad \forall k = 1, 2, \dots, m \quad (5)$$

$$\sum_{j=0}^n x_{ij}^k = y_i^k \quad \forall i = 0, 1, \dots, n; \quad \forall k = 1, 2, \dots, m \quad (6)$$

$$\sum_{i,j \in S} x_{ij}^k \leq |S| - 1 \quad S \subset \{1, 2, \dots, n\}; \quad |S| \geq 2; \quad \forall k = 1, 2, \dots, m \quad (7)$$

$$x_{ij}^k \in [0, 1] \quad \forall i = 0, 1, \dots, n; \quad \forall j = 0, 1, \dots, n; \quad \forall k = 1, 2, \dots, m \quad (8)$$

$$y_i^k \in \{0, 1\} \quad \forall i = 0, 1, \dots, n; \quad \forall k = 1, 2, \dots, m \quad (9)$$

Όπως γίνεται αντιληπτό, η συνάρτηση (1) αποτελεί την αντικειμενική συνάρτηση του συνολικού κόστους διαδρομής και τελικός στόχος του προβλήματος είναι η ελαχιστοποίηση αυτής. Όσον αφορά τις υπόλοιπες:

- η συνάρτηση (2), αποτελεί περιοριστικό παράγοντα και δεν επιτρέπει στα οχήματα να ξεπεράσουν την μέγιστη χωρητικότητα, η οποία είναι σταθερή και ίση για κάθε ένα από αυτά.
- η συνάρτηση (3) είναι αυτή που ελέγχει ότι ο κάθε πελάτης του γράφου εξυπηρετείται από κάποιο όχημα.
- η συνάρτηση (4) ορίζει ότι κάθε όχημα θα πρέπει να ξεκινάει και να τερματίζει στο ίδιο σημείο.
- οι συναρτήσεις (5-7) αφορούν το πρόβλημα του πλανόδιου εμπόρου (TSP).
- οι συναρτήσεις (8) και (9) αφορούν τις δυαδικές μεταβλητές που αναφέρθηκαν παραπάνω.

2.2 Μέθοδοι λύσης του προβλήματος CVRP

Τις τελευταίες δεκαετίες έχουν προταθεί διάφορες ευρετικές μέθοδοι (heuristics) για την επίλυση του προβλήματος CVRP. Αυτές μπρούν να κατηγοριοποιηθούν ως εξής [29]:

- construction heuristics
- improvement heuristics
- metaheuristics

2.2.1 Construction heuristics

Τα construction heuristics δημιουργούν σταδιακά μία αρχική λύση. Πιο συγκεκριμένα, σε κάθε βήμα εισάγουν σημεία customers σε μία διαδρομή ή συνδυάζουν υποδιαδρομές, εξετάζοντας κάθε φορά ορισμένα κριτήρια τα οποία σχετίζονται με την χωρητικότητα των οχημάτων και το συνολικό κόστος διαδρομής [30]. Ορισμένα από τα πιο διαδεδομένα construction heuristics για την επίλυση του προβλήματος CVRP, είναι τα παρακάτω [31]:

- Sequential Insertion algorithm (SI)
- Improved Parallel Insertion algorithm
- Nearest Neighbor heuristic (NN)
- Clarke and Wright Savings heuristic (CWS)

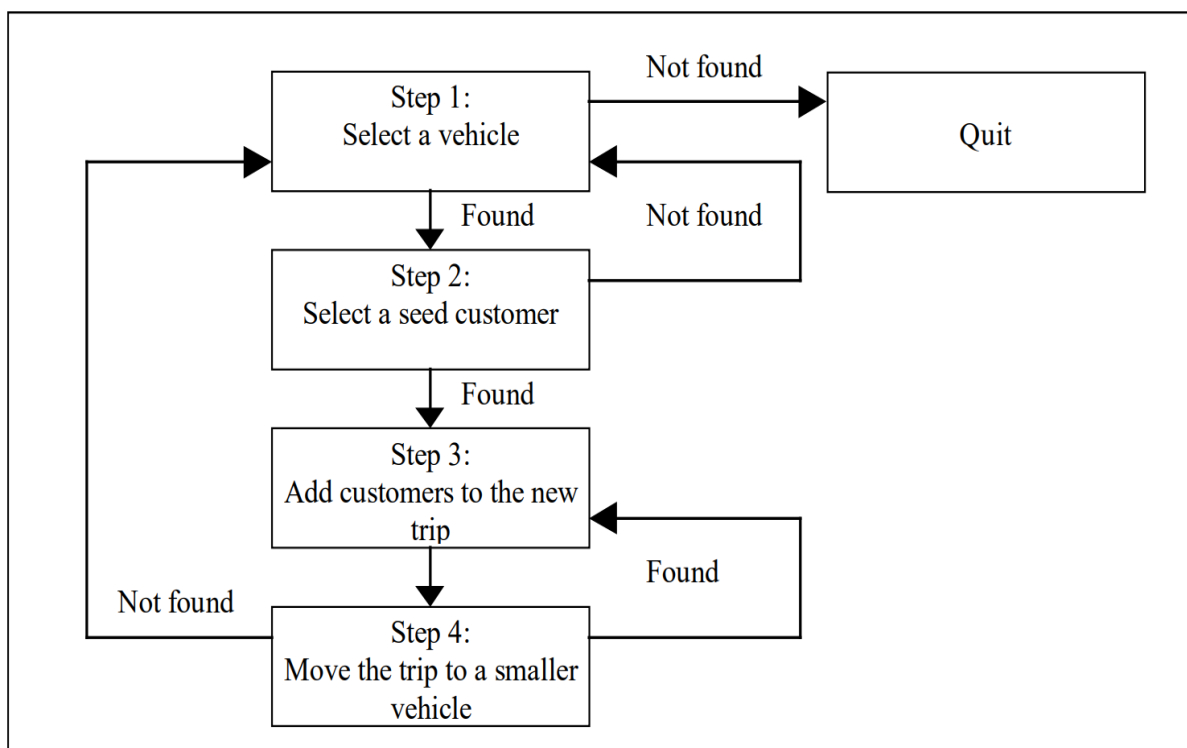
Sequential Insertion algorithm (SI) Το συγκεκριμένο construction heuristic, δημιουργεί σταδιακά διαδρομές (routes) την μία μετά την άλλη [32]. Η διαδικασία που ακολουθείται συνοψίζεται σε δύο βήματα.

Στο πρώτο βήμα, δημιουργείται μία νέα διαδρομή k , όπου $k \leq K$, εισάγοντας σε αυτήν ένα τυχαίο σημείο v_i , όπου $i \neq 0$ και επίσης αυτό δεν είναι το σημείο της αφετηρίας v_0 . Έτσι, δημιουργείται μία διαδρομή (v_0, v_i, v_0) .

Στο δεύτερο βήμα επιλέγεται ένα σημείο v_j , όπου $j \neq 0$ και εισάγεται στην αρχική διαδρομή του βήματος 1 αφού τηρεί τις εξής προϋποθέσεις: Η προσθήκη του σημείου στην αρχική διαδρομή επιφέρει την μικρότερη διαφορά κόστους που θα μπορούσε να προστεθεί, ενώ παράλληλα η συνολική χωρητικότητα της διαδρομής μαζί με το νεοεισαχθέν σημείο, δεν ξεπερνάει την μέγιστη χωρητικότητα που έχει οριστεί για το κάθε όχημα. Επομένως, θα πρέπει να τηρούνται αυστηρώς οι παρακάτω περιορισμοί:

- $\operatorname{argmin} c(v_a, v_j) + c(v_j, v_{a+1}) - c(v_a, v_{a+1})$, όπου $v_a \in$ διαδρομή k , $v_{a+1} \in$ διαδρομή k και $v_j \notin$ διαδρομή k .
- D διαδρομής $k + d_j \leq C$, όπου D διαδρομής k , η συνολική χωρητικότητα της διαδρομής k .

Σε περίπτωση που τηρούνται, τότε το σημείο v_j , όπου $j \neq 0$, εισάγεται στην διαδρομή k ανάμεσα από τα σημεία v_a και v_{a+1} . Το συγκεκριμένο βήμα επαναλαμβάνεται έως ότου να μην είναι η δυνατή η προσθήκη άλλων σημείων στην διαδρομή k λόγω έλλειψης χωρητικότητας. Τότε δημιουργείται εκ νέου μία διαδρομή k και η διαδικασία ξεκινάει και πάλι από το βήμα 1.



Σχήμα 9: Βήματα εφαρμογής αλγορίθμου Sequential Insertion algorithm (SI), εικόνα από [33].

Improved Parallel Insertion algorithm Το συγκεκριμένο construction heuristic, αποτελεί μία παραλλαγή του προηγούμενου και δημιουργεί διαδρομές ταυτόχρονα [34]. Στην συγκεκριμένη περίπτωση ο αριθμός K_{min} των επιχτών διαδρομών ορίζεται από τον τύπο

$$K_{min} =$$

$$\sum_{i \in |V|} d_i / C$$

Κάθε μία διαδρομή k αρχικοποιείται με K_{min} διαφορετικό, εισάγοντας κάθε φορά το πλησιέστερο προς την αφετηρία σημείο v_i , όπου $i \neq 0$. Με αυτόν τον τρόπο, προκύπτουν διαδρομές της μορφής (v_0, v_i, v_0) .

Στο δεύτερο βήμα του αλγορίθμου, εισάγεται ένα τυχαίο σημείο v_j , όπου $j \neq 0$ σε μία από τις διαδρομές που κατασκευάστηκαν στο πρώτο βήμα. Οι ίδιοι περιορισμοί όπως και στον προηγούμενο αλγόριθμο ισχύουν και εδώ. Πιο συγκεκριμένα, η εισαγωγή κάθε νέου σημείου v_j σε μία από τις διαδρομές k , δεν θα πρέπει να ξεπερνά την χωρητικότητα αυτής, ενώ παράλληλα θα πρέπει να οδηγεί στην μικρότερη συνολική αύξηση κόστους όσον αφορά την διαδρομή.

- $\operatorname{argmin} c(v_a, v_j) + c(v_j, v_{a+1}) - c(v_a, v_{a+1})$, όπου $v_a \in$ διαδρομή k , $v_{a+1} \in$ διαδρομή k και $v_j \notin$ διαδρομή k .
- D διαδρομής $_k + d_j \leq C$, όπου D διαδρομής $_k$, η συνολική χωρητικότητα της διαδρομής $_k$.

Τα βήματα 1 και 2, εκτελούνται στην συνέχεια με τον ίδιο τρόπο που εκτελούνται και στον αλγόριθμο Sequential Insertion(SI).

Nearest Neighbor heuristic (NN) Ο συγκεκριμένος αλγόριθμος κατασκευής αρχικών λύσεων ανήκει στην κατηγορία των άπληστων αλγορίθμων, δημιουργεί αρχικές διαδρομές σταδιακά, και λειτουργεί με τον παρακάτω τρόπο:

Στο πρώτο βήμα, επιλέγεται ένα σημείο v_i , όπου $i \neq 0$, όχι τυχαίο, αλλά εκείνο το οποίο βρίσκεται πλησιέστερα στην αφετηρία v_0 και προστίθεται στην αρχική διαδρομή. Έτσι λοιπόν, κατασκευάζεται μια διαδρομή της μορφής (v_0, v_i) .

Στο δεύτερο βήμα, επιλεγεται ένα σημείο v_j , όπου $j \neq 0$, το οποίο είναι το πλησιέστερο σημείο στο τελευταίο που προστέθηκε στην διαδρομή του βήματος 1.

Στον αλγόριθμο αυτόν οι περιορισμοί οι οποίοι δεν πρέπει να παραβιάζονται είναι οι εξής:

- $\operatorname{argmin} c(v_i, v_j)$, όπου $v_i \in$ διαδρομή k , $v_j \notin$ διαδρομή k .
- D διαδρομής $_k + d_j \leq C$, όπου D διαδρομής $_k$, η συνολική χωρητικότητα της διαδρομής $_k$.

Εφόσον λοιπόν, τηρούνται οι παραπάνω συνθήκες, κάθε φορά ένα σημείο v_j προστίθεται ακριβώς μετά από το τελευταίο σημείο της διαδρομής που αρχικά είναι το v_i το οποίο είναι πλησιέστερο προς την αφετηρία v_0 . Το δεύτερο βήμα του αλγορίθμου σταματάει και δημιουργείται μία νέα διαδρομή, όταν δεν είναι δυνατό να προστεθούν επιπλέον σημεία στην προηγούμενη διαδρομή.

Procedure Nearest Neighbor Heuristic

```

10 Select an arbitrary node  $j$ 
20 Set  $l = j$  and  $W = \{1, 2, \dots, n\} \setminus \{j\}$ 
30 While  $W \neq \emptyset$  do
40   Let  $j \in W$  such that  $c_{lj} = \min\{c_{li} \mid i \in W\}$ 
50   Connect  $l$  to  $j$  and set  $W = W \setminus \{j\}$  and  $l = j$ .
60 Connect  $l$  to the node selected in 10 to form a Hamiltonian cycle.

```

Σχήμα 10: Ψευδοκώδικας αλγορίθμου nearest neighbour Σχήμα από [35].

Clarke and Wright Savings heuristic (CWS) Στον αλγόριθμο αυτόν, όλα τα σημεία $v_i \in V$, $i \neq 0$ πρέπει να δημιουργήσουν $|V - 1|$ διαδρομές. Επομένως, θα προκύψουν $|V - 1|$ διαδρομές της μορφής (v_0, v_i, v_0) [31].

Στο δεύτερο βήμα του αλγορίθμου, $\forall v_i \in V, \forall v_j \in V$, $i \neq 0, j \neq 0, i \neq j$ ο παράγοντας s υπολογίζεται από τον τύπο $s(v_i, v_j) = c(v_i, v_0) + c(v_0, v_j) - c(v_i, v_j)$

Όλοι αυτοί οι υπολογισμοί, αποθηκεύονται σε μία λίστα \vec{S} σε μη αύξουσα σειρά.

Στο τρίτο βήμα του αλγορίθμου χρησιμοποιείται ο πρώτος από τους υπολογισμούς που βρίσκονται αποθηκευμένοι στην λίστα \vec{S} . Γίνεται επίσης ο έλεγχος για δύο συνθήκες, σε περίπτωση που ορίζονται δύο διαδρομές x και y .

- υπάρχει μία ακμή (v_i, v_0) στην διαδρομή x και μία ακμή (v_0, v_j) στην διαδρομή y
- $D_x + D_y \leq C$

Σε περίπτωση που υπάρχουν οι δύο παραπάνω διαδρομές, τότε ενώνονται στο σημείο (v_i, v_j) . Έπειτα από αυτό, ελέγχεται ο επόμενος υπολογισμός ο οποίος είναι αποθηκευμένος στην λίστα \vec{S} .

2.2.2 Improvement heuristics

Αφού εφαρμοστεί κάποιο construction heuristic και κατασκευαστεί μια αρχική εφικτή λύση, το επόμενο στάδιο για την επίλυση του προβλήματος CVRP είναι η εφαρμογή των improvement heuristics. Στόχος αυτών, είναι η βελτίωση της υπάρχουσας λύσης. Αυτό επιτυγχάνεται επαναληπτικά, αναζητώντας στον γειτονικό χώρο της υπάρχουσας λύσης για όλο και καλύτερες, έως ότου δεν παρατηρείται πλέον κάποια βελτίωση. Αρκετοί ερευνητές έχουν δημοσιεύσει εργασίες που αφορούν διάφορες τεχνικές τοπικής αναζήτησης [36].

Οι αλγόριθμοι improvement heuristics χωρίζονται σε δύο κατηγορίες:

1. inter-route: στην περίπτωση αυτή, επιδιώκεται η βελτίωση της υπάρχουσας λύσης, ανταλλάσσοντας σημεία (customers) μεταξύ πολλών διαδρομών (routes) και όχι μόνο μέσα σε κάθε μία χωριστά.
2. intra-route: στην περίπτωση αυτή, η βελτίωση της υπάρχουσας λύσης επιτυγχάνεται ανταλλάσσοντας σημεία (customers) μόνο μέσα σε κάθε μία διαδρομή χωριστά, χωρίς να υπάρχει καμία επικοινωνία με τις υπόλοιπες.

Τρία από τα πιο συνηθισμένα improvement heuristics τα οποία χρησιμοποιούνται στην επίλυση του CVRP είναι:

- relocate
- exchange
- 2-Opt

Σημειώνεται ότι μία μέθοδος τοπικής αναζήτησης πχ (relocate), μπορεί να χρησιμοποιηθεί ως inter-route αλλά και ως intra-route improvement heuristic. Όπως αναφέρθηκε παραπάνω, στόχος των μεθόδων improvement heuristics είναι η ελαχιστοποίηση της αντικειμενικής συνάρτησης του προβλήματος. Στην περίπτωση του CVRP, αυτή είναι το συνολικό κόστος διαδρομής.

Exchange

Ο αλγόριθμος εφαρμόζεται για όλους τους πελάτες - σημεία έως ότου δεν παρατηρείται κάποια βελτίωση στην λύση.

Ο ψευδοκώδικας ο οποίος παρουσιάζει αναλυτικά τα βήματα του συγκεκριμένου improvement heuristic είναι ο παρακάτω [36]:

1. εντόπισε κάποιον πελάτη, ο οποίος μπορεί να απομακρυνθεί από την παρούσα διαδρομή σε μία άλλη (λαμβάνοντας υπόψιν ότι η μετακίνηση αυτού στην νέα διαδρομή δεν θα ξεπεράσει το όριο χωρητικότητας της).
2. βελτίωσε την λύση της διαδρομής από την οποίαν απομακρύνθηκε ο πελάτης i , εφαρμόζοντας intra-route heuristics.
3. για κάθε πιθανή διαδρομή k_2 :

για κάθε πιθανό πελάτη j της κάθε διαδρομής του παραπάνω βήματος:

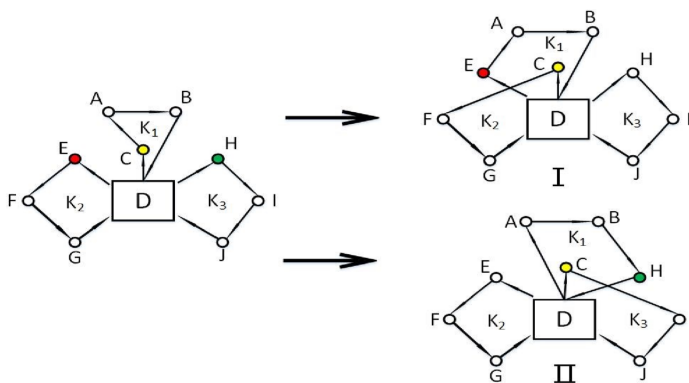
απομάκρυνε τον πελάτη j από την συγκεκριμένη διαδρομή.

εάν ο πελάτης i μπορεί να εισέλθει στην νέα διαδρομή του k_2 και ο πελάτης j μπορεί να εισέλθει στην νέα διαδρομή του k_1 , τότε:

αντάλλαξε τους δύο πελάτες μεταξύ τους και βελτίωσε την λύση της κάθε διαδρομής, εφαρμόζοντας intra-route heuristics σε κάθε μία από αυτές.

υπολόγισε τις νέες αποστάσεις

4. Σε περίπτωση που κάποια αλλαγή προκαλεί βελτίωση της τελικής λύσης, γίνεται αποδεκτή, (Σχήμα 11).



Σχήμα 11: Αναπαράσταση ανταλλαγής σημείων (exchange heuristic), [36].

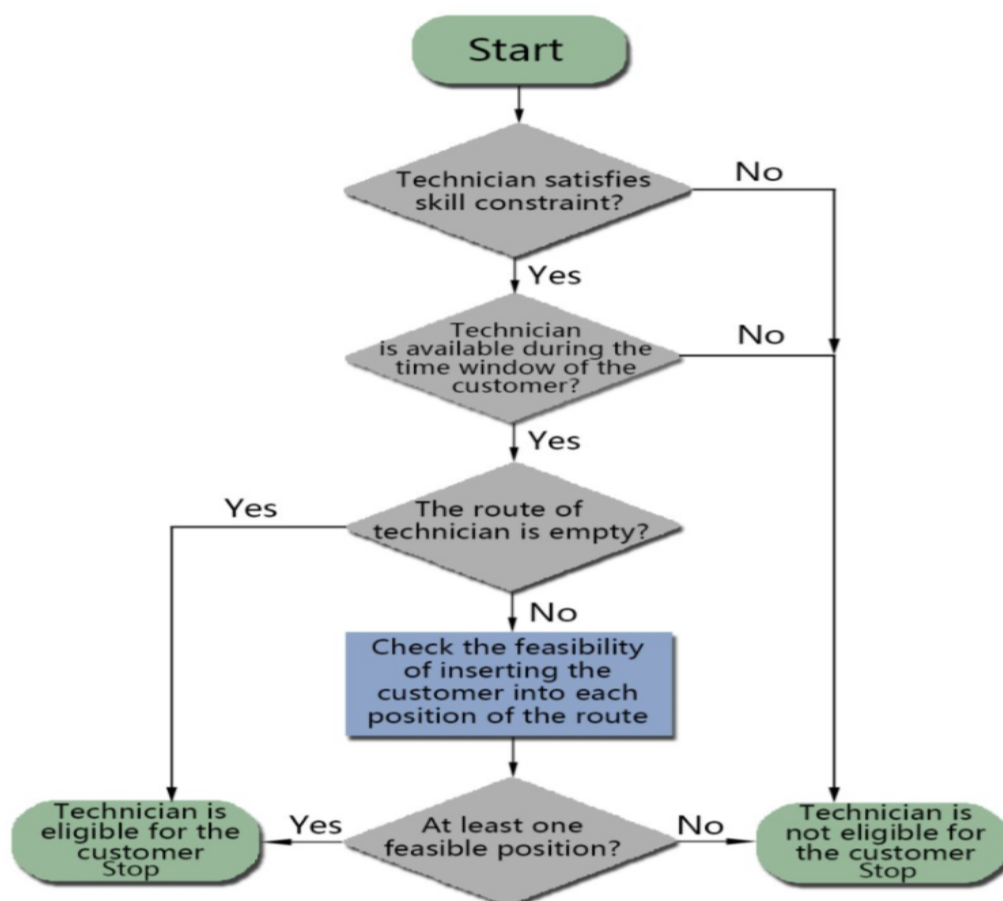
Relocate

Ο αλγόριθμος relocate, ακριβώς όπως και ο προηγούμενος, επαναλαμβάνεται έως ότου δεν παρατηρείται καμία βελτίωση στην τελική λύση.

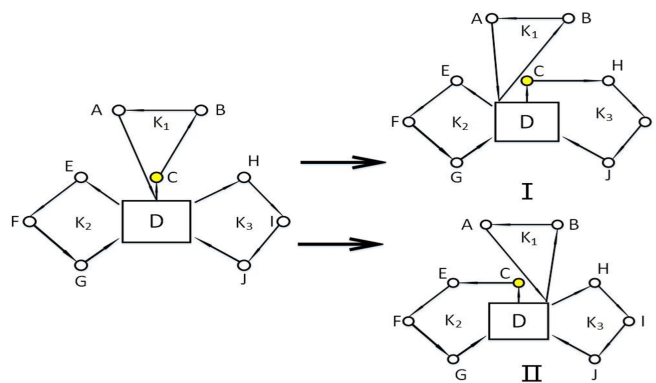
Αντίστοιχα, ο ψευδοκώδικας ο οποίος περιγράφει τον συγκεκριμένο αλγόριθμο είναι ο εξής:

1. βρες πιθανές διαδρομές routes στα οποία μπορεί να μετακινηθεί ο πελάτης i μέσα από την διαδικασία του Σχήματος 12.
2. αφού απομακρυνθεί ο πελάτης i από την αρχική διαδρομή, βελτίωσε την λύση αυτής εφαρμόζοντας intra-route heuristics.
3. μετακίνησε τον πελάτη i σε μία από τις υπόλοιπες διαδρομές.

Αν κάποια μετακίνηση του πελάτη i βελτιώσει την τελική λύση, τότε γίνεται αποδεκτή (Σχήμα 13).



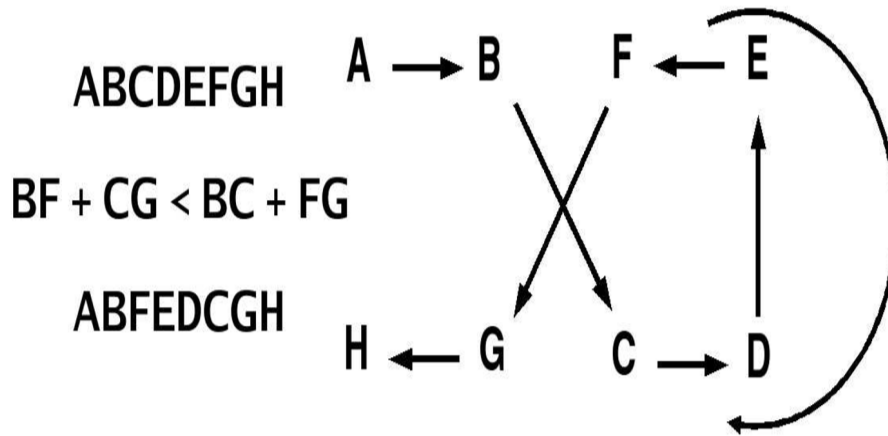
Σχήμα 12: Διαδικασία ελέγχου για μετακίνηση ενός πελάτη σε άλλη διαδρομή, [36].



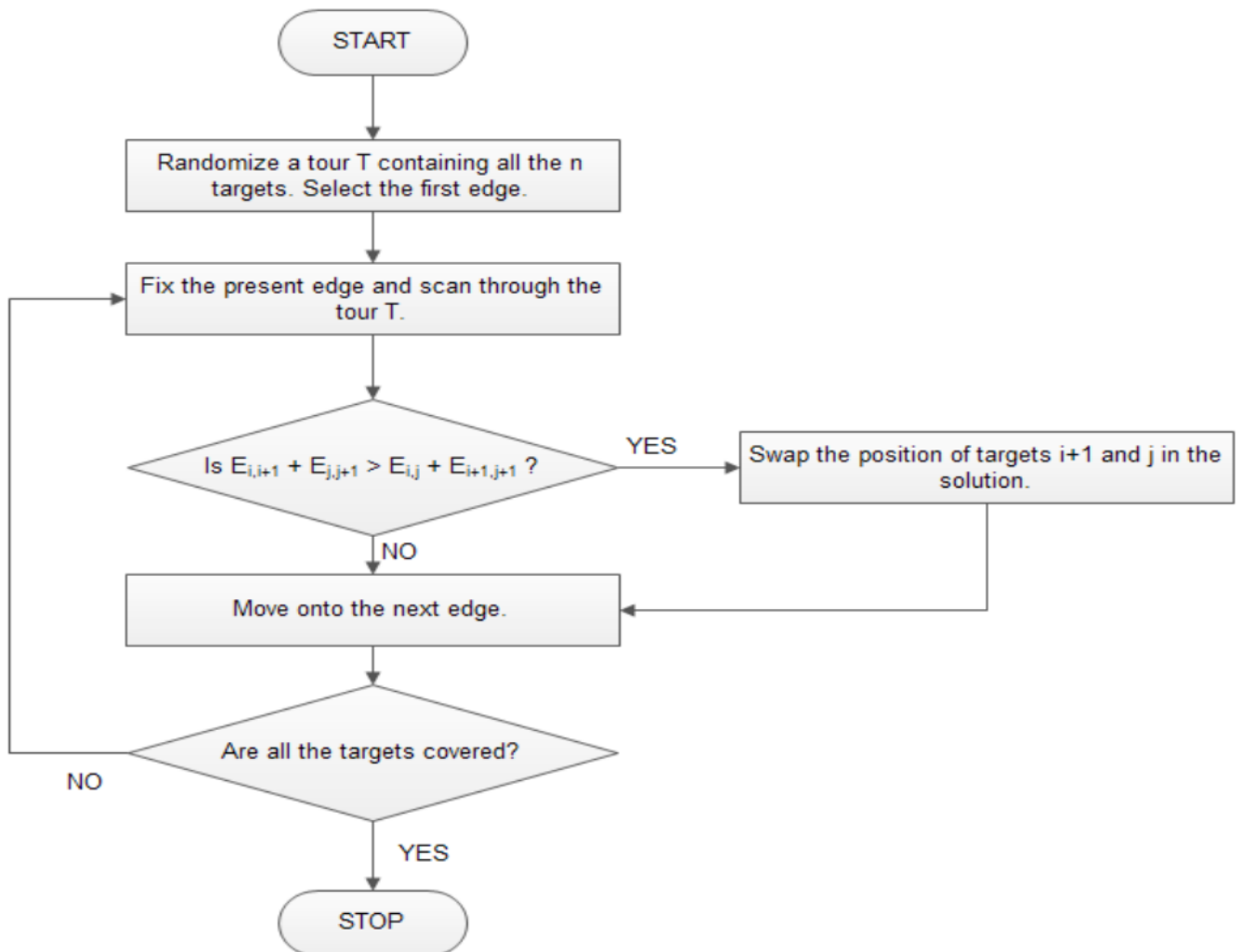
Σχήμα 13: Αναπαράσταση του relocate heuristic, [36].

2 - Opt

Το συγκεκριμένο improvement heuristic, χρησιμοποιείται πολύ συχνά για την βελτίωση της αρχικής λύσης του CVRP, αλλά και στο πρόβλημα του πλανόδιου εμπόρου TSP [37]. Η λογική που ακολουθείται κατά την εκτέλεση του αλγορίθμου είναι απλή. Κάθε φορά, ξεκινώντας από μία αρχική λύση, δύο ακμές συγκρίνονται μεταξύ τους και αν η ανταλλαγή μεταξύ των δύο οδηγεί σε μία καλύτερη λύση, τότε οι δύο αυτές ακμές ανταλλάσσονται. Παρατηρώντας το Σχήμα 14, γίνεται αντιληπτό ότι η απόφαση στο σημείο αυτό, βασίζεται στο αν το άθροισμα των ακμών $BF + CG$ είναι μικρότερο από εκείνο του αθροίσματος των ακμών $BC + FG$ το οποίο εντοπίζεται στην αρχική λύση. Έτσι λοιπόν, στην περίπτωση που ισχύει η σχέση $BF + CG \leq BC + FG$, τότε τα διανύσματα C και F αναλάσσονται για να δημιουργήσουν δύο νέες ακμές, τις BF και CG . Τονίζεται ότι για να διατηρηθούν οι υπόλοιπες ακμές, η σειρά των διανυσμάτων μεταξύ των σημείων C και F αντιστρέφεται. Αυτοί οι επαναληπτικοί έλεγχοι μεταξύ διαφορετικών ακμών εκτελούνται, έως ότου ο αλγόριθμος καταλήξει στην καλύτερη δυνατή λύση (Σχήμα 15). Όπως παρατηρείται και στο Σχήμα 14, οπτικά αυτό που επιτυγχάνεται με την χρήση του συγκεκριμένου αλγορίθμου, είναι να μην διασταυρώνονται τα μονομάτια μεταξύ διαδοχικών σημείων στην συνολική διαδρομή, βελτιώνοντας έτσι σε σημαντικό βαθμό την ποιότητα της λύσης.



Σχήμα 14: Παράδειγμα εφαρμογής του αλγορίθμου 2 - Opt, εικόνα από [37].



Σχήμα 15: Βήματα εφαρμογής του αλγορίθμου 2 - Opt, εικόνα από [38].

Στρατηγική αποδοχής λύσης των **improvement heuristics**

Στο σημείο αυτό, τονίζεται ότι σε έναν αλγόριθμο επίλυσης του προβλήματος CVRP, όταν εφαρμόζεται κάποιο **improvement heuristic**, υπάρχουν οι ακόλουθες δύο επιλογές όσον αφορά την αποδοχή μίας νέας καλύτερης λύσης [39]:

- first improvement
- best improvement

Στο **first improvement**, επιδιώκεται η αποφυγή της πολυπλοκότητας του χρόνου αξιολόγησης όλων των πιθανών γειτονικών λύσεων. Έτσι, γίνεται αποδεκτή η πρώτη πιθανή γειτονική λύση που παρουσίασε βελτίωση.

Αντίθετα, στο **best improvement**, σε κάθε βήμα αναζήτησης επιλέγεται εκείνη η γειτονική λύση, η οποία επιτυγχάνει την μέγιστη μείωση κόστους της συνολικής διαδρομής.

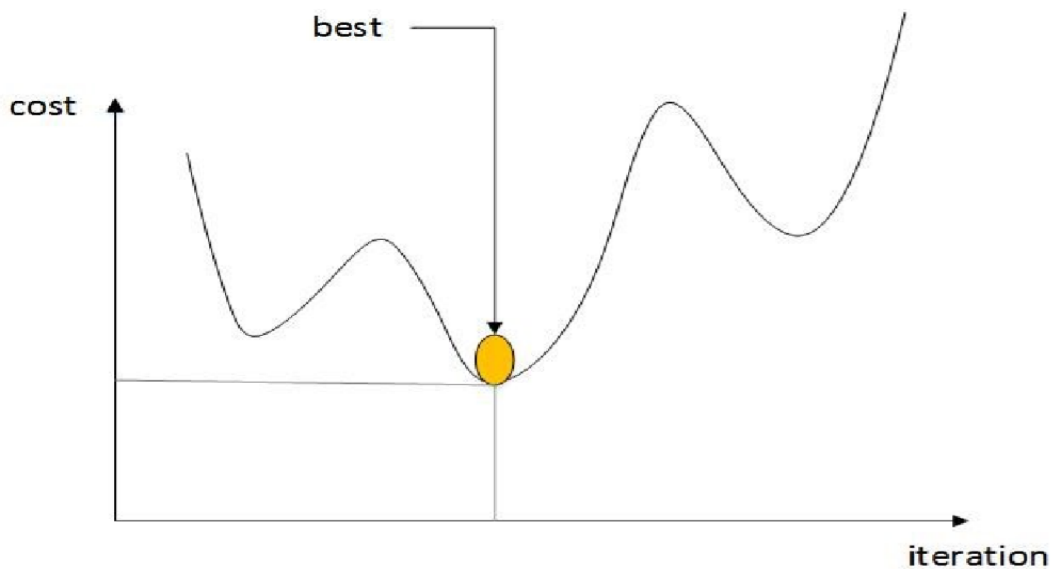
Ωστόσο, σύμφωνα με ορισμένες έρευνες, η εφαρμογή του **best improvement** έναντι του **first improvement**, δεν εγγυάται πάντοτε καλύτερα αποτελέσματα και πιο γρήγορη εκτέλεση του αλγορίθμου. Πολλοί φαίνεται να είναι οι παράγοντες οι οποίοι παίζουν σημαντικό ρόλο σε αυτό. Πιο συγκεκριμένα, με βάση την έρευνα στον άρθρο [40], ο συγγραφέας φαίνεται να καταλήγει στο εξής συμπέρασμα:

- Εάν η αρχική λύση επιλεγεί τυχαία, το **first improvement** οδηγεί σε καλύτερα αποτελέσματα και ταχύτερα από το **best improvement**.
- Ωστόσο, αν η αρχική λύση βρεθεί με κάποιο **construction heuristic** πχ **nearest neighbour**, δηλαδή εάν η αρχική λύση δεν είναι πολύ κακή, τότε το **best improvement** οδηγεί σε καλύτερα αποτελέσματα και πολλές φορές ταχύτερα κατά μέσο όρο από το **first improvement**.

Αν και τα παραπάνω **improvement heuristics** τα οποία αναλύθηκαν βελτιώνουν τις περισσότερες φορές την λύση, ωστόσο περιγράφονται ως άπληστες μέθοδοι και για τον λόγο αυτόν παρουσιάζουν ένα σημαντικό ελάττωμα. Στην προσπάθειά τους να βελτιώνουν σε κάθε επανανάληψη την υπάρχουσα λύση, παγιδεύονται πολλές φορές σε τοπικά ελάχιστα της αντικειμενικής συνάρτησης, χωρίς να μπορούν να ξεφύγουν. Αυτό έχει ως αποτέλεσμα, να μην μπορούν να αναζητήσουν καλύτερες λύσεις σε νέες γειτονίες αναζήτησης, επιτυγχάνοντας μια λύση η οποία αν δεν αποτελεί το ολικό ελάχιστο της συνάρτησης, τουλάχιστον βρίσκεται πολύ κοντά σε αυτό. Την λύση στο πρόβλημα αυτό, δίνει η χρήση των **metaheuristics** [41].

2.2.3 Metaheuristics

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, μία αδυναμία που παρουσιάζουν οι αλγόριθμοι improvement heuristics είναι ότι μπορεί να παγιδευτούν πολύ εύκολα σε τοπικά ελάχιστα. Σε αντίθεση με τα improvement heuristics, τα οποία αναζητούν ολοένα και καλύτερη λύση, οι αλγόριθμοι metaheuristics πολλές φορές στις επαναλήψεις δέχονται χειρότερες λύσεις από την υπάρχουσα. Η τακτική αυτή εφαρμόζεται για να αντιμετωπιστεί το πρόβλημα των τοπικών ελαχίστων, καθώς ο αλγόριθμος ψάχνει συνεχώς σε μεγαλύτερες γειτονίες για μία καλύτερη λύση. Στο σημείο αυτό βέβαια αναφέρεται, ότι ακόμα και αν ο αλγόριθμος εντοπίσει το ολικό ελάχιστο της συνάρτησης, δεν το αντιλαμβάνεται και συνεχίζει την αναζήτηση (Σχήμα 16).



Σχήμα 16: Διάγραμμα το οποίο τονίζει την παρουσία τοπικών ελαχίστων και του ολικού ελαχίστου σε μία συνάρτηση βελτιστοποίησης, [36].

Οι μεθευρετικές μέθοδοι αναζήτησης (metaheuristics), κατατάσσονται σε τρεις κύριες κατηγορίες, όσον αφορά την επίλυση του προβλήματος πλανόδιου εμπόρου TSP, ωστόσο αυτοί εφαρμόζονται και στην επίλυση του CVRP:

- Local search: στην κατηγορία αυτήν, σε κάθε επανάληψη σημειώνεται βελτίωση της λύσης μόνο σε μία διαδρομή. Μερικοί από τους πιο γνωστούς αλγορίθμους της κατηγορίας είναι: Tabu search, Simulated annealing, ILS, VNS...
- Population-based: εδώ η αναζήτηση και η βελτίωση της λύσης γίνεται σε έναν πληθυσμό διαδρομών. Μερικοί από τους πιο γνωστούς αλγορίθμους της κατηγορίας είναι: Hybrid

GA, evolutionary algorithms, ACO, path relinking...

- Learning mechanisms: Ορισμένοι διαδεδομένοι αλγόριθμοι της κατηγορίας είναι: neural networks, ant colony optimization
- Hybrid approaches: Στην κατηγορία αυτήν, ανήκουν εκείνοι οι αλγόριθμοι οι οποίοι συνδυάζουν περισσότερες από μία μεθόδους μεθευρετικής αναζήτησης.

Σημειώνεται ότι, πολλοί μεθευρετικοί αλγόριθμοι είναι εμπνευσμένοι από τα φαινόμενα της φύσης, όπως για παράδειγμα ο αλγόριθμος simulated annealing. Στον αλγόριθμο αυτόν, η διαδικασία που ακολουθείται είναι παρόμοια με εκείνη της τήξης και σταθεροποίησης των μετάλλων [36]. Πιο συγκεκριμένα, όταν η θερμοκρασία είναι πολύ υψηλή η κατάσταση του μετάλλου είναι ασταθής, καθώς βρίσκεται ακόμα σε υγρή μορφή, ενώ όσο η θερμοκρασία μειώνεται, αυτό ψύχεται και στο τέλος συμπαγοποιείται αποκτώντας μια σταθερή κατάσταση. Παρόμοια, ο αλγόριθμος simulated annealing ξεκινάει με μία αρχική λύση και έναν παράγοντα T , ο οποίος όταν είναι πολύ μεγάλος, ο αλγόριθμος κινείται τυχαία σε όλο και μεγαλύτερες γειτονιές αναζήτησης και δέχεται σχετικά εύκολα ακόμα και λύσεις χειρότερης ποιότητας (Σχήμα 17). Ωστόσο, όταν ο παράγοντας T μειωθεί αρκετά, λύσεις χειρότερης ποιότητας γίνονται πολύ δύσκολα ή καθόλου αποδεκτές. Εφόσον λοιπόν ο αλγόριθμος έχει εξερευνήσει μεγάλο μέρος του χώρου αναζήτησης, στο τελικό στάδιο συγκλίνει σε μία λύση πολύ καλής ποιότητας και μερικές φορές στην βέλτιστη (ολικό ελάχιστο συνάρτησης).

Εκφράζοντας με μαθηματικούς όρους το παραπάνω πρόβλημα:

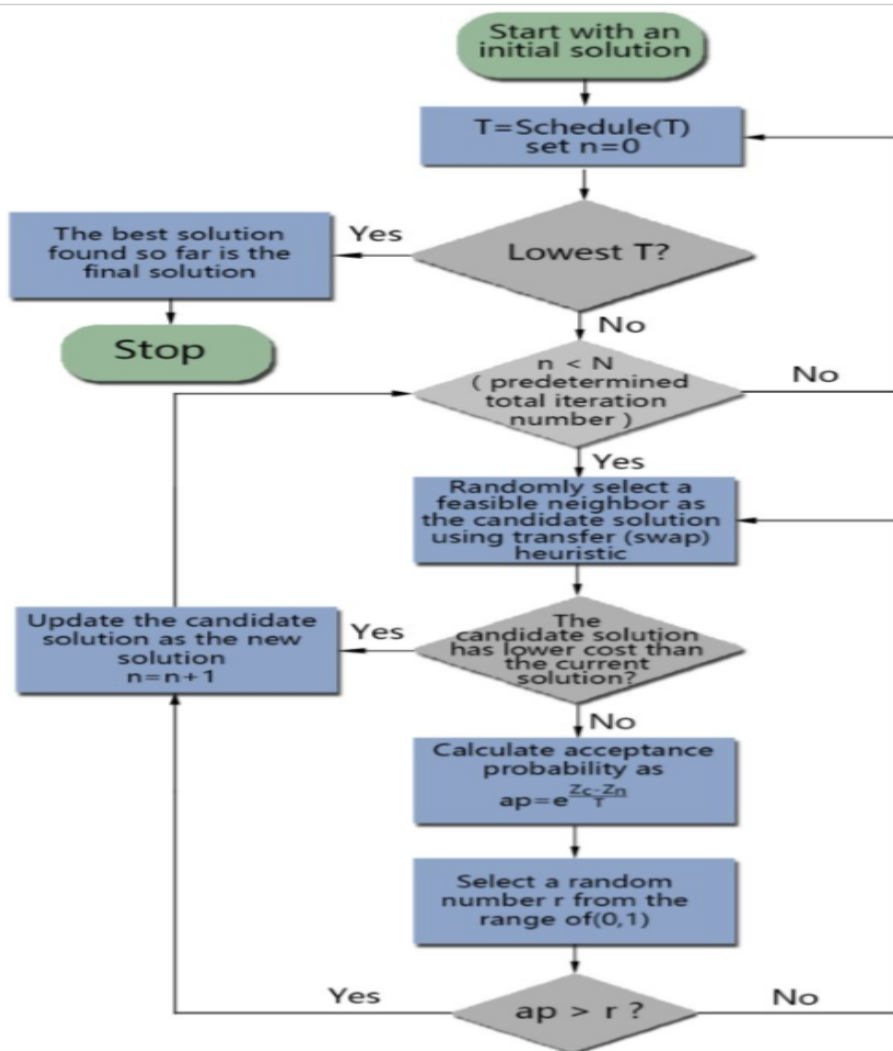
Έστω ότι:

- Z_c η αντικειμενική συνάρτηση της παρούσας λύσης
- Z_n η αντικειμενική συνάρτηση της επόμενης λύσης
- T ο παράγοντας που επηρεάζει την πιθανότητα αποδοχής μίας λύσης

Τότε:

- αν $Z_n \leq Z_c$ τότε η επόμενη διαδρομή γίνεται πάντα δεκτή ως η νέα λύση
- αν τώρα $Z_n \geq Z_c$ τότε η λύση n γίνεται αποδεκτή με την εξής πιθανότητα: acceptance

$$= e^{-\frac{Z_c - Z_n}{T}}$$



Σχήμα 17: Βήματα εφαρμογής του αλγορίθμου simulated annealing, εικόνα από [36].

Εμπνευσμένοι από την φύση, φαίνεται να είναι και οι γενετικοί αλγόριθμοι(GA) επίλυσης του προβλήματος. Οι γενετικοί αλγόριθμοι ανήκουν στην κατηγορία των population based, το οποίο πρακτικά σημαίνει ότι σε κάθε επανάληψη (γενιά) αξιολογούν πολλαπλές λύσεις (πληθυσμούς). Ο αλγόριθμος ξεκινά με μία αρχική λύση και από όλους του πληθυσμούς επιλέγει τους δύο καλύτερους και τους μεταφέρει ακριβώς όπως είναι στην επόμενη γενιά. Για να δημιουργηθούν επιπλέον λύσεις, μεταλλάσσει τις δυό καλύτερες λύσεις εκτελώντας ορισμένες διεργασίες μετάλλαξης όπως είναι η διασταύρωση και τις μεταφέρει και αυτές στην επόμενη γενιά. Η έννοια του ανταγωνισμού είναι αυτή που κυριαρχεί στους γενετικούς αλγορίθμους, εφόσον σε κάθε γενιά επιβιώνουν μόνο οι καλύτερες λύσεις. Φτάνοντας στην τελευταία επανάληψη του αλγορίθμου, προκύπτει μία ικανοποιητική λύση πολύ κοντά στην βέλτιστη [42] [43].

Variable Neighbourhood Search (VNS)

Ο Variable Neighbourhood Search (VNS) είναι ο αλγόριθμος μεθευρετικής αναζήτησης (metaheuristic) ο οποίος χρησιμοποιήθηκε για την επίλυση του προβλήματος στην παρούσα διατριβή και για τον λόγο αυτόν εξετάζεται αναλυτικότερα. Αποτελεί μία σχετικά νέα μέθοδο αναζήτησης [44][45][46], η οποία ψάχνει σε διαφορετικές γειτονιές λύσεων για μία καλύτερη, τόσο για να φτάσει στο ελάχιστο της συνάρτησης αλλά και να δραπετεύσει από το τοπικό ελάχιστο αυτής. Η μέθοδος (VNS) βασίζεται στις εξής παρατηρήσεις για την αναζήτηση στον χώρο:

- το τοπικό ελάχιστο μίας γειτονιάς, δεν αποτελεί τοπικό ελάχιστο και για μία άλλη.
- το ολικό ελάχιστο αποτελεί τοπικό ελάχιστο συναρτήσει όλων των γειτονικών λύσεων.
- σε πολλά προβλήματα, τα τοπικά ελάχιστα γειτονικών λύσεων βρίσκονται κοντά μεταξύ τους.

Η βασική δομή λειτουργίας VNS είναι η εξής [47]:

Αρχικοποίηση

- διάλεξε το σύνολο των γειτονικών δομών N_k , $k = 1, \dots, k_{max}$, οι οποίες θα χρησιμοποιηθούν στην αναζήτηση
- βρες μία αρχική λύση x
- όρισε το κριτήριο τερματισμού εκτέλεσης του αλγορίθμου

επανάλαβε το παρακάτω σύνολο εντολών όσο δεν συναντάς το κριτήριο τερματισμού:

- (1) όρισε $k=1$
- (2) επανέλαβε τα παρακάτω βήματα έως ότου $k = k_{max}$
 - (a) shaking: δημιούργησε ένα σημείο x' τυχαία από την k^{th} γειτονιά του x ($x' \in N_k(x)$).
 - (b) local search: εφάρμοσε μεθόδους τοπικής αναζήτησης με αρχική λύση την x' , και σημείωσε με x'' το τοπικό βέλτιστο.
 - (c) κριτήριο αποδοχής λύσης: αν η λύση είναι καλύτερη από του x , κράτησέ την και συνέχισε την αναζήτηση με N_1 ($k = 1$), αλλιώς $k = k + 1$.

Αρχικά λοιπόν, ορίζεται ένα σύνολο γειτονικών δομών γύρω από ένα σημείο $x \in X$. Στην συνέχεια, εφαρμόζεται κάποιο local search προκειμένου να βρεθεί το βέλτιστο x . Έπειτα, επιλέγεται τυχαία ένα σημείο x' από την πρώτη γειτονική δομή $N_1(x)$ του x και ένα ελάχιστο βρίσκεται με εφαρμογή local search στην δομή του x' . Έτσι λοιπόν, γίνεται η εύρεση ενός νέου ελαχίστου x'' . Στο σημείο αυτό, προκύπτουν οι εξής περιπτώσεις:

- $x'' = x$. Στην περίπτωση αυτήν, η διαδικασία επαναλαμβάνεται χρησιμοποιώντας την επόμενη δομή γειτονιάς $N_k(x)$, $k \geq 2$
- $x'' \neq x$, με $f(x'') \geq f(x)$. Και στην συγκεκριμένη περίπτωση η διαδικασία συνεχίζεται με την επόμενη δομή γειτονιάς.
- $x'' \neq x$, με $f(x'') \leq f(x)$. Αυτό σημαίνει ότι έχει βρεθεί κάποιο ελάχιστο, το οποίο οδηγεί σε καλύτερη λύση σε σχέση με το προηγούμενο. Η αναζήτηση εστιάζεται τότε γύρω από το x'' ξεκινώντας και πάλι από την πρώτη δομή γειτονιάς.

Η διαδικασία συνεχίζεται μέχρις ότου φτάσουμε σε ένα από τα κριτήρια τερματισμού του αλγορίθμου (μέγιστος χρόνος εκτέλεσης αλγορίθμου ή μέγιστος αριθμός επαναλήψεων ή μέγιστος αριθμός επαναλήψεων μετά τον τελευταίο εντοπισμό καλύτερης λύσης).

Παραλλαγές της VNS έχουν χρησιμοποιηθεί επίσης αρκετά για την επίλυση του προβλήματος CVRP. Ορισμένες από αυτές είναι:

- Variable Neighborhood descent (VND)
- Reduced VNS (RVNS)
- General VNS (GVNS)

2.2.4 Exact αλγόριθμοι

Οι αλγόριθμοι ευρετικής και μεθευρετικής αναζήτησης, οι οποίοι αναλύθηκαν παραπάνω, βρίσκουν ικανοποιητικές λύσεις, σε ικανοποιητικό χρόνο, αλλά δεν εγγυώνται ότι θα βρουν το ολικό ελάχιστο της συνάρτησης. Σε αντίθεση με αυτούς, οι exact αλγόριθμοι, φτάνουν πάντοτε στο ολικό ελάχιστο της αντικειμενικής συνάρτησης του προβλήματος, παρόλα αυτά ο υπολογιστικός χρόνος που απαιτείται συνήθως είναι πολύ μεγάλος, ειδικά όσον αφορά μεγάλου όγκου δεδομένα. Όπως αναφέρθηκε παραπάνω, οι exact αλγόριθμοι μπορούν να βρουν το ολικό ελάχιστο μίας συνάρτησης σε λογικό χρόνο, συνήθως για δεδομένα που δεν ξεπερνούν τα 60 σημεία [25].

Ωστόσο, τις τελευταίες δεκαετίες έχουν γίνει πολλές έρευνες, όσον αφορά τους συγκεκριμένους αλγόριθμους στην επίλυση του προβλήματος VRP.

Αυτοί κατατάσσονται σε τρεις κατηγορίες [48]:

- άμεσες μέθοδοι αναζήτησης δέντρου (direct tree search methods)
- μέθοδοι δυναμικού προγραμματισμού (dynamic programming)
- μέθοδοι ακέραιου γραμμικού προγραμματισμού (integer linear programming)()

Η άμεση μέθοδος αναζήτησης δέντρων λειτουργεί με τη διαδοχική κατασκευή διαδρομών οχημάτων, χρησιμοποιώντας το μέσο ενός κλάδου και ενός συνδεδεμένου δέντρου (branch and bound tree). Αυτός ο αλγόριθμος χρησιμοποιήθηκε το 1981 [49] για την επίλυση του προβλήματος TSP και του VRP αργότερα, χρησιμοποιώντας δεδομένα 25 σημείων. Ωστόσο, το 1986 [50], ο ίδιος αλγόριθμος με ορισμένες αλλαγές στον πίνακα αποστάσεων, χρησιμοποιήθηκε για να λυθεί το πρόβλημα του ασύμμετρου CVRP με 260 δεδομένα. Τέλος, το 2015 [51] χρησιμοποιήθηκε και υπολόγισε το ολικό ελάχιστο συνάρτησης, η οποία αφορούσε την ελαχιστοποίηση του συνολικού κόστους διαδρομής για 65 δρομολόγια σχολικών λεωφορείων.

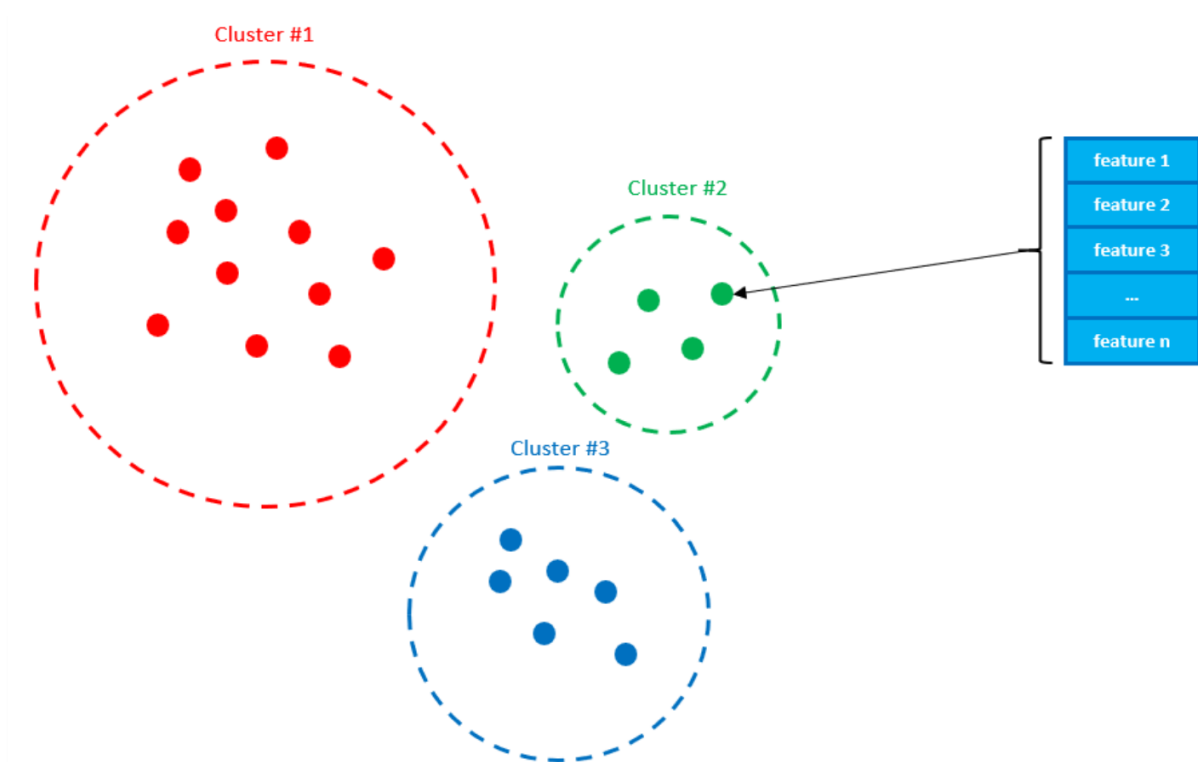
Η επίλυση του προβλήματος VRP με την χρήση μεθόδων δυναμικού προγραμματισμού, προτάθηκε το 1974 [52]. Ωστόσο, στην συγκεκριμένη κατηγορία απαιτείται μεγάλος αριθμός υπολογισμών και ένα σύνολο περιορισμών, οι οποίοι θα πρέπει να τηρούνται σε κάθε περίπτωση. Χρήση του αλγορίθμου για την επίλυση του προβλήματος CVRP με δεδομένα 50 σημείων, έγινε το 1981 [49].

Τέλος, στην κατηγορία του ακέραιου γραμμικού προγραμματισμού, ο πιο γνωστός αλγόριθμος ο οποίος έχει προταθεί είναι αυτός του διαχωρισμού (set partitioning formulation)[53]. Η μέθοδος αυτή, χρησιμοποιεί έναν δυαδικό πίνακα m γραμμών και n στηλών. Κάθε στήλη αντιπροσωπεύει και μία πιθανή διαδρομή, ενώ η κάθε γραμμή αντιπροσωπεύει έναν πελάτη. Όλες οι σειρές θα πρέπει να καλύπτονται με ελάχιστο κόστος από υποσύνολο των στηλών. Ωστόσο, επειδή περιλαμβάνεται ένας εκθετικός αριθμός μεταβλητών σε προβλήματα με πολλές εφικτές λύσεις, συνήθως δεν μπορεί να χρησιμοποιηθεί άμεσα για την επίλυση του προβλήματος VRP.

3 Συσταδοποίηση (Clustering)

3.1 Ορισμός και αλγόριθμοι συσταδοποίησης

Συσταδοποίηση (clustering), ονομάζεται η ομαδοποίηση ενός συνόλου αντικειμένων, με τέτοιο τρόπο ώστε τα αντικείμενα της ίδιας ομάδας (που ονομάζεται συστάδα) να είναι πιο παρόμοια μεταξύ τους (σε κάποια έννοια) με αυτά που ανήκουν σε άλλες ομάδες (συστάδες), Σχήμα 18. Η συσταδοποίηση, χρησιμοποιείται ευρέως σε τομείς που αφορούν εξόρυξη γνώσης από δεδομένα καθώς και στατιστική ανάλυση αυτών [54].



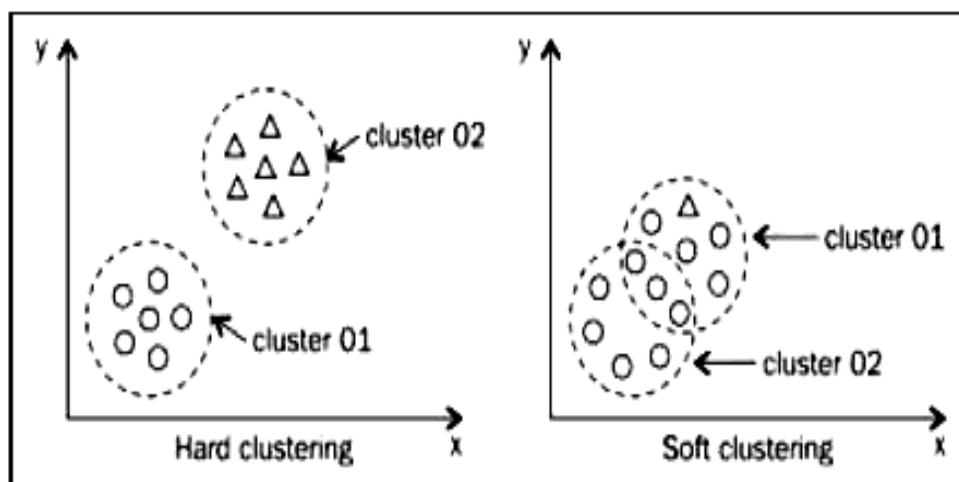
Σχήμα 18: Παράδειγμα συσταδοποίησης μίας ομάδας δεδομένων, εικόνα από [55].

Η συσταδοποίηση μπορεί να χωριστεί σε δύο υποκατηγορίες:

- Hard clustering
- Soft clustering

Η πρώτη κατηγορία, αφορά την ομαδοποίηση των δεδομένων έτσι ώστε το κάθε ένα από αυτά να ανήκει μόνο σε μία συστάδα.

Το αντίθετο συμβαίνει στην δεύτερη κατηγορία συσταδοποίησης, όπου κάποιο από τα στοιχεία των δεδομένων μπορεί να ανήκει σε παραπάνω από μία συστάδες.



Σχήμα 19: Παράδειγμα hard-soft clustering, [55].

Όσον αφορά τον τρόπο λειτουργίας και την μεθοδολογία με την οποία πραγματοποιείται η συσταδοποίηση, διακρίνονται 4 κατηγορίες:

1. Centroid based clustering
2. Density based clustering
3. Hierarchical based clustering
4. Grid based clustering

Η συσταδοποίηση (1), βασίζεται σε έναν επαναληπτικό αλγόριθμο, στον οποίο οι συστάδες σχηματίζονται με βάση την εγγύτητα των σημείων δεδομένων στο κεντροειδές των συστάδων. Εδώ, το κέντρο της κάθε συστάδας, δηλαδή το κεντροειδές, είναι κατασκευασμένο έτσι ώστε η απόσταση των σημείων δεδομένων να είναι ελάχιστη από αυτό.

Η συσταδοποίηση (2), πραγματοποιείται με βάση την πυκνότητα των σημείων στον χώρο των δεδομένων. Περιοχές με διαφορετική πυκνότητα σημείων από τις άλλες, ομαδοποιούνται σε διαφορετική συστάδα.

Η συσταδοποίηση (3), ομαδοποιεί τα δεδομένα ως δομή τύπου δέντρου βασισμένη στην ιεραρχία. Στην κατηγορία αυτήν, διακρίνονται δύο επιπλέον υποκατηγορίες:

- Agglomerative μέθοδος
- Divisive μέθοδος

Στην συσταδοποίηση (4), τα δεδομένα ομαδοποιούνται σε δομή τύπου πλέγματος. Οι λειτουργίες συσταδοποίησης στην κατηγορία αυτήν είναι ιδιαίτερα γρήγορες.

Ορισμένοι από τους πιο διαδεδομένους αλγόριθμους, οι οποίοι χρησιμοποιούνται στην ανάλυση των δεδομένων και στην επιστήμη της μηχανικής μάθησης είναι οι εξής:

- K-Means
- Density-Based Spatial Clustering of Applications with Noise (DBSCAN)
- Mean-Shift
- Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)

3.1.1 Ο αλγόριθμος συσταδοποίησης (K-Means)

Ο συγκεκριμένος αλγόριθμος, ανήκει στην κατηγορία των Centroid based μεθόδων συσταδοποίησης και είναι ίσως ο πιο διαδεδομένος.

Η μαθηματική περιγραφή του τρόπου λειτουργίας του αλγορίθμου, σύμφωνα με την επίσημη βιβλιογραφία [56] είναι η εξής:

Σε ένα σύνολο δεδομένων (x_1, x_2, \dots, x_n) , ο αλγόριθμος συσταδοποίησης (K-Means) προσπαθεί να τοποθετήσει τα δεδομένα (n) σε $k (\leq n)$ συστάδες $S = \{S_1, S_2, \dots, S_k\}$ έτσι ώστε να ελαχιστοποιήσει σε κάθε συστάδα των άθροισμα των τετραγώνων των σημείων από το κέντρο της συστάδας.

Η αντικειμενική συνάρτηση του αλγορίθμου περιγράφεται ως:

$$\operatorname{argmin}_S \sum_{i=1}^k \sum_{x \in S_i} |x - m_i|^2 = \operatorname{argmin}_S \sum_{i=1}^k |S_i| \operatorname{Var} S_i$$
, όπου m_i ο μέσος των σημείων στο κάθε S_i .

Η παραπάνω σχέση είναι ίση με την σχέση:

$$\operatorname{argmin}_S \sum_{i=1}^k \frac{1}{2|S_i|} \sum_{x,y \in S_i} |x - y|^2$$

η οποία μπορεί να γραφτεί και ως:

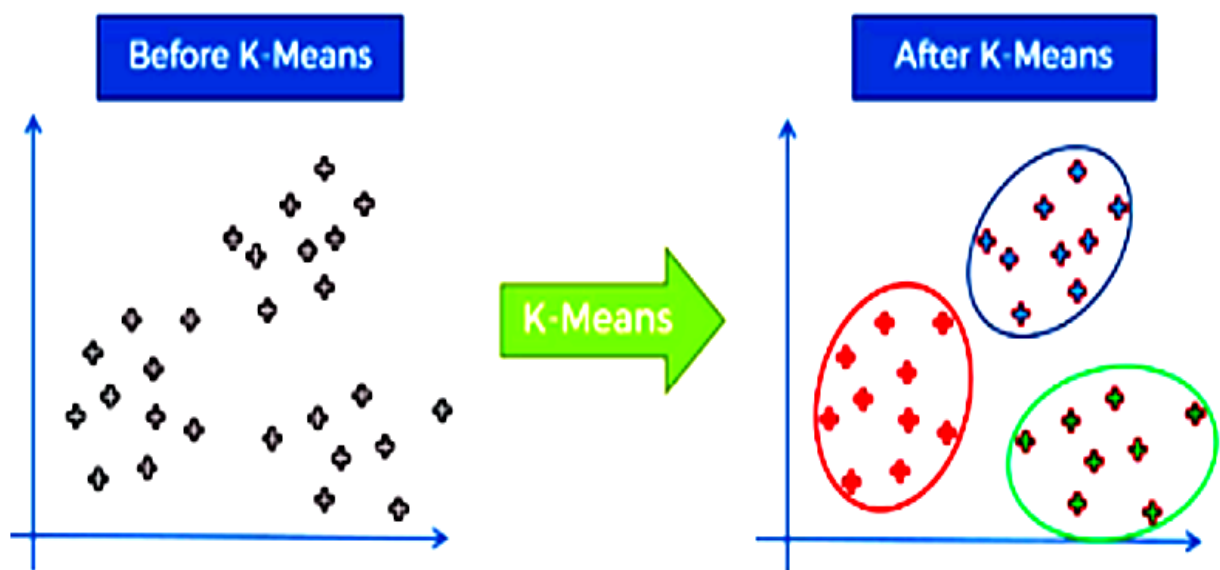
$$\sum_{x \in S_i} |x - m_i|^2 = \sum_{x \neq y \in S_i} (x - m_i)(m_i - y)$$

Επειδή η συνολική διακύμανση είναι σταθερή, αυτό ισοδυναμεί με την μεγιστοποίηση του αθροίσματος των τετραγωνικών αποκλίσεων μεταξύ σημείων στις διαφορετικές συστάδες.

Τα βήματα που ακολουθεί ο αλγόριθμος για να συσταδοποιήσει ένα σύνολο δεδομένων, είναι τα παρακάτω:

1. αρχικά, επιλέγεται μια σειρά κλάσεων / ομάδων για να χρησιμοποιηθεί και να οριστούν τυχαία τα κεντρώα αυτών. Ο αριθμός των συστάδων επιλέγεται στην αρχή με βάση την κατανομή των σημείων στον χώρο.
2. για κάθε σημείο, υπολογίζεται η απόσταση αυτού από τα κεντρώα που ορίστηκαν στο πρώτο βήμα και εισάγεται στην συστάδα του κεντρώου που βρίσκεται πλησιέστερα.
3. έπειτα υπολογίζεται η νέα θέση των κεντρώων, με βάση την μέση τιμή των δεδομένων που ανήκουν στην συστάδα αυτών.
4. επανέλαβε τα παραπάνω βήματα μέχρι να μην παρατηρείται μετατόπιση των κεντρώων.

Ο συγκεκριμένος αλγόριθμος, έχει το πλεονέκτημα ότι είναι γρήγορος όσον αφορά την συσταδοποίηση. Ωστόσο, ένα από τα μειονεκτήματα που παρουσιάζει είναι ότι πρέπει να οριστεί αρχικά ο αριθμός των συστάδων, το οποίο πολλές φορές είναι δύσκολο να προσδιοριστεί. Επιπλέον, εξάγει ικανοποιητικά αποτελέσματα σε περίπτωση που τα δεδομένα παρουσιάζουν ευδιάκριτη ομαδοποίηση στον χώρο (Σχήμα 20).



K-Means Clustering

Σχήμα 20: Παράδειγμα συσταδοποίησης με την χρήση του K-Means, [55].

3.1.2 Ο αλγόριθμος συσταδοποίησης (DBSCAN)

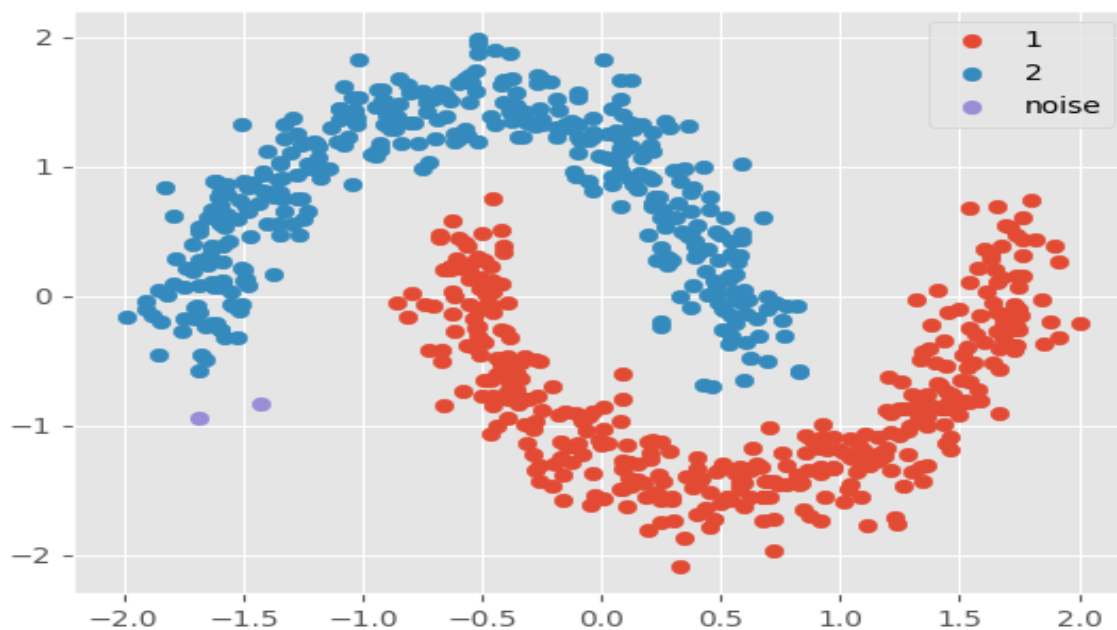
Ο αλγόριθμος συσταδοποίησης (DBSCAN), ανήκει στην κατηγορία των (density based) αλγορίθμων και είναι ιδανικός στην περίπτωση που στα δεδομένα παρατηρείται θόρυβος.

Η διαδικασία που ακολουθείται για την ομαδοποίηση των δεδομένων είναι η παρακάτω:

1. ο DBSCAN αρχίζει με ένα αυθαίρετο σημείο εκκίνησης που δεν έχει επισκεφθεί. Η γειτονιά αυτού του σημείου εξάγεται χρησιμοποιώντας μια απόσταση ϵ (όλα τα σημεία που βρίσκονται εντός της απόστασης ϵ θεωρούνται σημεία κοινής γειτονιάς).
2. εάν υπάρχει επαρκής αριθμός σημείων (σύμφωνα με έναν αριθμό minPoints) εντός αυτής της γειτονιάς τότε ξεκινά η διαδικασία ομαδοποίησης και το πρώτο σημείο δεδομένων γίνεται το πρώτο σημείο της συστάδας. Διαφορετικά, το σημείο θα χαρακτηριστεί ως θόρυβος (αργότερα αυτό το θορυβώδες σημείο μπορεί να γίνει σημείο μίας συστάδας). Και στις δύο περιπτώσεις το σημείο αυτό επισημαίνεται ως "επισκέφθηκε".
3. Για αυτό το πρώτο σημείο της νέας συστάδας, τα σημεία εντός της απόστασης ϵ εισάγονται επίσης στην ίδια συστάδα. Αυτή η διαδικασία ελέγχου πραγματοποιείται για όλα τα σημεία εντός της απόστασης ϵ .
4. Όταν ολοκληρωθεί η δημιουργία της πρώτης συστάδας, επιλέγεται και πάλι ένα σημείο και ξεκινάει η διαδικασία από το πρώτο βήμα, έως ότου ελεγχθούν όλα τα σημεία και συσταδοποιηθούν αντίστοιχα.

Ο αλγόριθμος συσταδοποίησης (DBSCAN) είναι ιδανικός για την συσταδοποίηση δεδομένων τα οποία σχηματίζουν διάφορα μονοπάτια στον χώρο (Εικόνα 21) και δεν ομαδοποιούνται σωστά με τους υπόλοιπους αλγορίθμους.

Ωστόσο, επειδή βασίζεται στην πυκνότητα των δεδομένων στον χώρο, αποτυγχάνει να τα συσταδοποιήσει σωστά, όταν εντοπίζονται πολλές ομάδες σχετικά κοντά μεταξύ τους, οι οποίες παρουσιάζουν διαφορετική πυκνότητα.



Σχήμα 21: Παράδειγμα συσταδοποίησης με την χρήση του DBSCAN, εικόνα από [57]

3.2 Παραλλαγές αλγορίθμων συσταδοποίησης για την επίλυση του (C)VRP

Στο προηγούμενο υποκεφάλαιο, αναφέρθηκαν οι κατηγορίες των αλγορίθμων συσταδοποίησης, ενώ αναλύθηκαν εκτενέστερα και δύο από αυτούς, ο K-Means και ο DBSCAN. Ωστόσο, όπως έγινε αντιληπτό, η γενική μορφή αυτών, αφορά συσταδοποίηση η οποία βασίζεται στις αποστάσεις μεταξύ των σημείων ή με βάση την πυκνότητα. Για παράδειγμα, δεν μπορεί να χρησιμοποιήσει κανείς κάποιον από τους παραπάνω αλγορίθμους, για να συσταδοποιήσει τους πελάτες του CVRP προβλήματος με βάση τις αποστάσεις μεταξύ τους αλλά επιπλέον να θέσει ως περιορισμό ότι η κάθε συστάδα δεν θα ξεπερνά την μέγιστη χωρητικότητα του κάθε οχήματος. Για τον λόγο αυτόν, επιπλέον μεταβλητές και περιορισμοί είναι ανάγκη να θεσπίζονται κάθε φορά, όσον αφορά τον τελικό στόχο του προβλήματος.

Πολλές δημοσιεύσεις έχουν γίνει στον τομέα του προβλήματος (C)VRP, όπου ερευνητές προσπαθούν να προσαρμόσουν τους παραπάνω αλγόριθμους συσταδοποίησης, στις ανάγκες του δικού τους προβλήματος. Στο άρθρο [58], γίνεται αναφορά στον Balanced K-Means αλγόριθμο, ο οποίος σχεδιάστηκε ειδικά για προβλήματα δρομολόγησης μεγάλης κλίμακας.

Τελικός στόχος φαίνεται να είναι ο σχεδιασμός μιας ομάδας γεωγραφικά κοντινών μεταξύ τους πελατών (συστάδες) αλλά με ισορροπημένο αριθμό πελατών. Παράλληλα στο άρθρο [59] γίνεται αναφορά σε μία ακόμα παραλλαγή του κλασικού K-Means αλγορίθμου, τον οποίον οι συγγραφείς έχουν ονομάσει Improved K-Means. Τελικός στόχος εδώ, είναι η συσταδοποίηση των πελατών με τον βέλτιστο τρόπο, ωστόσο εισάγεται και ο περιορισμός της συνολικής χωρητικότητας της κάθε συστάδας, η οποία δεν θα πρέπει να υπερβαίνει την μέγιστη χωρητικότητα του κάθε οχήματος που έχει οριστεί. Γίνεται φανερό ότι η περίπτωση αυτή, αφορά συσταδοποίηση δεδομένων στο πρόβλημα CVRP. Τέλος, στο άρθρο [60], οι συγγραφείς του άρθρου προτείνουν και αυτοί μία παραλλαγή του συγκεκριμένου αλγορίθμου, τον οποίον ονομάζουν Imbalanced K-Means. Όπως φανερώνει και το όνομα του αλγορίθμου, τελικός στόχος των ερευνητών ήταν η συσταδοποίηση μη ομοιόμορφα κατανεμημένων στον χώρο δεδομένων.

3.3 Ο αλγόριθμος συσταδοποίησης της παρούσας διατριβής

Ο αλγόριθμος συσταδοποίησης των δεδομένων του προβλήματος CVRP που χρησιμοποιήθηκε στην παρούσα διατριβή, αναφέρεται στο άρθρο [61] και ανήκει στην κατηγορία των centroid based.

Ο αλγόριθμος αποτελείται από δύο φάσεις:

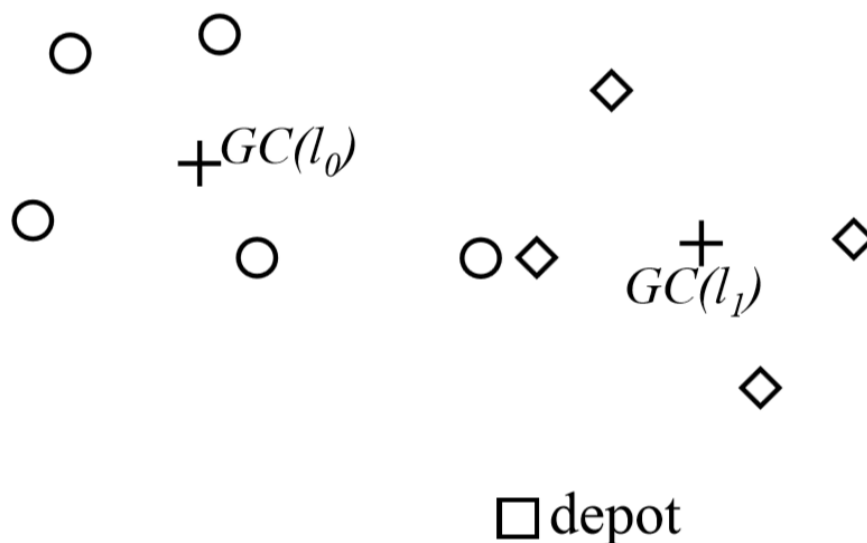
- δημιουργία των συστάδων
- βελτιστοποίηση των συστάδων

Κεντροειδές μίας συστάδας

Το κεντροειδές, μίας συστάδας ορίζεται ως εξής (Σχήμα 22):

Αν $I_i = u_0, u_1, \dots, u_k$, όπου i μία συστάδα και u_j ένα στοιχείο αυτής, τότε το κεντροειδές της συγκεκριμένης συστάδας ορίζεται ως:

$$GC(I_i) = \left(\sum_{j=0}^k \frac{u_j^x}{k}, \sum_{j=0}^k \frac{u_j^y}{k} \right), \text{ όπου } u_j^x \text{ και } u_j^y \text{ είναι οι συντεταγμένες } x \text{ και } y \text{ του } u_j.$$



Σχήμα 22: Παράδειγμα απεικόνισης των κεντρών (GC) των συστάδων, [61].

3.3.1 Δημιουργία συστάδων

Η πρώτη συστάδα I_0 δημιουργείται, εισάγοντας σε αυτήν το πιο απομακρυσμένο σημείο u_i από την αφετηρία. Τότε, το κεντρώο της πρώτης συστάδας $GC(I_0)$ υπολογίζεται από την σχέση που αναφέρθηκε παραπάνω. Στην συνέχεια βρίσκει το πλησιέστερο σημείο u_j ως προς το κεντρώο $GC(I_0)$ το οποίο δεν ανήκει σε κάποια συστάδα. Αν η χωρητικότητα του σημείου u_j , αθροιστεί με εκείνη της συστάδας I_0 και δεν υπερβαίνει την μέγιστη χωρητικότητα, τότε προστίθεται σε αυτήν. Στην περίπτωση αυτή, η συνολική χωρητικότητα της συστάδας I_0 αυξάνεται. Η εισαγωγή των σημείων τα οποία δεν ανήκουν σε κάποια συστάδα, συνεχίζεται στην συγκεκριμένη συστάδα, μέχρι η χωρητικότητά της να ξεπεράσει την μέγιστη. Τότε δημιουργείται μία νέα συστάδα και η διαδικασία ξεκινάει και πάλι από την αρχή. Ο αλγόριθμος τερματίζει όταν δεν υπάρχουν πλέον σημεία τα οποία να μην ανήκουν σε κάποια συστάδα.

```

Set  $i = 0$  and  $Q = \text{truck\_capacity}$ 
While (unvisited node exist)
   $v_j =$  the farthest node among un-clustered nodes from the depot
  Generate cluster  $l_i$  with  $v_j$ 
  Set capacity of  $l_i = Q$ 
  While (demand of  $v_j$  does not exceed available capacity of  $l_i$ )
    Add  $v_j$  to  $l_i$ 
    Reduce available capacity of  $l_i$  by demand of  $v_j$ 
    Calculate  $GC(l_i)$ 
     $v_j =$  the closest node among un-clustered nodes from  $GC(l_i)$ 
  End while
   $i = i+1$ 
End while

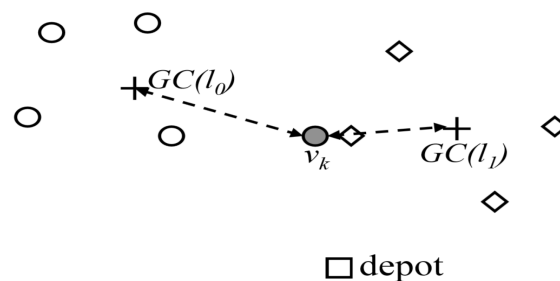
```

Σχήμα 23: Ψευδοκώδικας δημιουργίας συστάδων, [61].

3.3.2 Βελτιστοποίηση συστάδων

Στο σημείο αυτό, αν κάποιο σημείο u_k το οποίο ανήκει σε μία συστάδα I_i βρίσκεται πλησιέστερα στο κεντρώ $GC(I_j)$ της συστάδας I_j , τότε αν η χωρητικότητα του σημείου προστεθεί στην συστάδα I_j και δεν ξεπεράσει η χωρητικότητα της συστάδας την μέγιστη, τότε μετακινείται στην συστάδα I_j (Σχήμα 24). Ωστόσο, τότε πρέπει να γίνουν δύο διεργασίες:

- να ενημερωθούν οι χωρητικότητες των συστάδων I_i και I_j αντίστοιχα
- να ενημερωθούν τα κεντρώ $GC(I_i)$ και $GC(I_j)$ των συστάδων I_i και I_j αντίστοιχα



Σχήμα 24: Σημείο συστάδας το οποίο πρέπει να μετακινηθεί σε άλλη, [61].

```

Set cluster_group = { $l_0, l_1, \dots, l_m$ }
For  $i = 0$  to  $m$  repeat
  For every node  $v_k$  in  $l_i$ 
    For every  $l_j$  in cluster_group
      If ( $i \neq j$  and  $v_k$  is closer to  $GC(l_j)$  than  $GC(l_i)$  and available
        capacity of  $l_j$  is equal to or bigger than demand of  $v_k$ )
        Move  $v_k$  from  $l_i$  to  $l_j$ 
        Recalculate  $GC(l_i)$  and  $GC(l_j)$ 
      End if
    End for
  End for
End for

```

Σχήμα 25: Ψευδοκώδικας βελτιστοποίησης συστάδων, [61].

4 Μεθοδολογία παρούσας διατριβής για την επίλυση του CVRP

Στην παρούσα διατριβή, η μεθοδολογία επίλυσης του προβλήματος CVRP είναι παρόμοια με αυτήν του άρθρου [61]. Εφαρμόζεται πρώτα ο αλγόριθμος συσταδοποίησης των δεδομένων και έπειτα λύνεται το πρόβλημα με την χρήση της VNS ως μεθευρετική μέθοδο αναζήτησης.

Πιο συγκεκριμένα, ακολουθούνται τα εξής βήματα:

1. εφαρμογή του αλγορίθμου συσταδοποίησης ο οποίος αναλύθηκε παραπάνω και σχηματισμός m συστάδων.
2. στο στάδιο αυτό το πρόβλημα αντιμετωπίζεται ως πολλαπλά TSP, εφόσον έχουν προκύψει m συστάδες.

Για την κάθε συστάδα:

εφαρμογή της μεθόδου nearest neighbour ως construction heuristic και σχηματισμός συνολικά m αρχικών διαδρομών.

για την κάθε συστάδα, βελτιώνεται η αρχική λύση αυτής (αντικειμενική συνάρτηση του προβλήματος TSP παρακάτω), εφαρμόζοντας ως improvement heuristic τον αλγόριθμο

- 2-Opt, ο οποίος λειτουργεί παράλληλα ως intra route heuristic για το συνολικό πρόβλημα CVRP.
3. αφού βελτιωθούν οι m αρχικές διαδρομές, υπολογίζεται το συνολικό κόστος διαδρομής αθροίζοντας τα κόστη όλων των συστάδων.
 4. στο σημείο αυτό, το πρόβλημα αντιμετωπίζεται και πάλι στο σύνολό του ως CVRP.
 5. εφαρμόζονται τα inter route heuristics της VNS, τα οποία είναι τα (exchange, relocate) και έτσι τώρα οι διαδρομές των συστάδων που έχουν προκύψει αλληλεπιδρούν μεταξύ τους. Έπειτα, ξεκινάει η διαδικασία πάλι από το βήμα 2, χωρίς να εφαρμοστεί αυτήν την φορά το construction heuristic.
 6. ως στρατηγική αποδοχής νέας λύσης χρησιμοποιείται το first improvement, δηλαδή στην περίπτωση που κάποιο από τα inter route heuristics της VNS οδηγεί σε καλύτερη λύση, αυτή γίνεται αποδεκτή και υπολογίζεται εκ νέου το συνολικό κόστος διαδρομής.
 7. ο αλγόριθμος τερματίζει με βάση το κριτήριο τερματισμού της VNS, δηλαδή μόλις επιτευχθεί ο μέγιστος αριθμός επαναλήψεων του αλγορίθμου.

Μαθηματική διατύπωση προβλήματος (TSP) για την κάθε συστάδα

Θεωρώντας για την κάθε συστάδα τα σημεία (πελάτες) αυτής με αριθμούς $1, \dots, n$ όρισε:

$$x_{ij} = \begin{cases} 1 & \text{αν η διαδρομή πηγαίνει από το } i \text{ στο } j \\ 0 & \text{διαφορετικά} \end{cases}$$

Για $i = 1, \dots, n$, θεώρησε μία μεταβλητή u_i και την απόσταση από την πόλη i ως την πόλη j ως c_{ij} .

Τότε, το πρόβλημα TSP για την κάθε συστάδα διατυπώνεται ως εξής [62]:

$$\min \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} x_{ij}:$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, \dots, n$$

$$u_i \in Z \quad i = 2, \dots, n$$

$$\sum_{i=1, i \neq j}^n x_{ij} = 1 \quad j = 1, \dots, n$$

$$\sum_{j=1, j \neq i}^n x_{ij} = 1 \quad i = 1, \dots, n$$

$$u_i - u_j + nx_{ij} \leq n - 1 \quad 2 \leq i \neq j \leq n$$

$$0 \leq u_i \leq n - 1 \quad 2 \leq i \leq n$$

Για την κάθε συστάδα λοιπόν, πρέπει να επισκεφτεί κανείς όλους τους πελάτες και να ξαναγυρίσει στην αφετηρία, έχοντας διανύσει την μικρότερη δυνατή απόσταση. Αυτή είναι η αντικειμενική συνάρτηση του προβλήματος TSP.

5 Αποτελέσματα και ερμηνεία

5.1 Αρχικοποίηση παραμέτρων

Η αρχικοποίηση των παραμέτρων του αλγορίθμου επίλυσης του προβλήματος, είναι ιδιαίτερα σημαντική και αποτελεί καθοριστικό βήμα όσον αφορά την ποιότητα της τελικής λύσης.

Οι παράμετροι του αλγορίθμου οι οποίοι έπρεπε να οριστούν στην παρούσα διατριβή, ήταν οι εξής:

- ο αριθμός των επαναλήψεων του local search πριν από την εκτέλεση της VNS. Πιο συγκεκριμένα, στο βήμα αυτό χρησιμοποιήθηκαν 80 επαναλήψεις, στις οποίες όπως προαναφέρθηκε σε προηγούμενο κεφάλαιο, εφαρμόστηκαν διαδοχικά οι τελεστές relocate και exchange. Κάθε φορά επιλέγονταν με τυχαίο τρόπο δύο διαδρομές για την εφαρμογή των προαναφερθέντων τελεστών. Ως στρατηγική αποδοχής καλύτερης λύσης χρησιμοποιήθηκε η first improvement.
- ο αριθμός των επαναλήψεων του local search κατά την εκτέλεση της VNS. Στο βήμα αυτό, χρησιμοποιήθηκαν 40 επαναλήψεις εφαρμόζοντας και πάλι παρόμοιους τελεστές αλλά και στρατηγική αποδοχής καλύτερης λύσης.
- στο shaking phase, το οποίο αποτελεί τμήμα της VNS για την αλλαγή γειτονιάς, χρησιμοποιήθηκε ο τελεστής relocate.
- Τέλος, ένας πολύ σημαντικός παράγοντας όσον αφορά την εκτέλεση της VNS, είναι η επιλογή του αριθμού k_{max} στην VNS. Η επιλογή της κατάλληλης τιμής του k_{max} γίνεται πειραματικά.
- ο χρόνος εκτέλεσης της VNS αποτελεί επίσης έναν παράγοντα ο οποίος ορίζεται αρχικά, αφού μαζί με την παράμετρο k_{max} , αποτελούν τις δύο συνθήκες τερματισμού του αλγορίθμου VNS.

5.2 Κατηγοριοποίηση περιπτώσεων και πειραματική αρχικοποίηση παραμέτρων VNS

Στην παρούσα διατριβή, εξετάστηκαν όλα τα instances τα οποία βρίσκονται στην βιβλιοθήκη του προβλήματος CVRP (<http://vrp.galagos.inf.puc-rio.br/index.php/en/>).

Οι περιπτώσεις (instances) ταξινομήθηκαν σε τρεις κατηγορίες (μικρής δυσκολίας, μεσαίας δυσκολίας και μεγάλης δυσκολίας), κυρίως ανάλογα με το μέγεθος κάθε περίπτωσης (αριθμός πελατών) και τον αριθμό των διαδρομών (routes). Ωστόσο, άλλοι σημαντικοί παράγοντες (η τοποθέτηση αποθήκης, ο υψηλός συντελεστής μεταβολής των απαιτήσεων και ο σχηματισμός συστάδων λόγω των θέσεων των πελατών) παίζουν επίσης ουσιαστικό ρόλο στην ταξινόμηση αυτή [63].

- **Μικρής δυσκολίας:** ($A - n36 - k5, A - n37 - k6, A - n38 - k5, A - n39 - k5, A - n39 - k6, A - n55 - k9, B - n31 - k5, Bn34 - k5, B - n45 - k6, B - n52 - k7, B - n57 - k9, E - n22 - k4, E - n33 - k4$)
- **Μεσαίας δυσκολίας:** ($A - n44 - k6, A - n45 - k6, A - n45 - k7, A - n46 - k7, A - n48 - k7, A - n53 - k7, A - n54 - k7, A - n60 - k9, A - n61 - k9, A - n62 - k8, A - n63 - k10, A - n63 - k9, A - n69 - k9, B - n38 - k6, B - n41 - k6, B - n43 - k6, B - n45 - k5, B - n50 - k8, B - n56 - k7, B - n64 - k9, B - n66 - k9, B - n78 - k10, E - n31 - k7, E - n51 - k5$)
- **Μεγάλης δυσκολίας:** ($A - n64 - k9, A - n65 - k9, A - n80 - k10, B - n63 - k10, B - n67 - k10, B - n68 - k9, E - n76 - k7, E - n76 - k8, E - n76 - k10, E - n76 - k14, E - n101 - k14, E - n101 - k8, M - n101 - k10, M - n200 - k16$)

Όσον αφορά την πειραματική επιλογή της παραμέτρου (k_{max}) στον αλγόριθμο VNS, τα βήματα τα οποία εκτελέστηκαν ήταν τα εξής:

1. από κάθε κατηγορία προβλημάτων που παρουσιάστηκε παραπάνω, επιλέχθηκαν τυχαία δύο περιπτώσεις.
2. για κάθε περίπτωση εξετάστηκαν τιμές $k_{max} = 5 \dots 20$.
3. για κάθε περίπτωση και για κάθε k εκτελέστηκε ο αλγόριθμος 5 φορές. Στο τέλος υπολογίστηκε για την κάθε περίπτωση, ο μέσος όρος απόκλισης των λύσεων της παρούσας διατριβής από τις καλύτερες που έχουν βρεθεί ως τώρα (benchmarks).

5.2.1 B-n52-k7 & B-n57-k9

Οι δύο συγκεκριμένες περιπτώσεις ανήκουν στην κατηγορία μικρής δυσκολίας προβλημάτων CVRP. Για την κάθε μία περίπτωση δίνονται οι ακόλουθες πληροφορίες:

B-n52-k7

Αριθμός οχημάτων: 7

Βέλτιστη λύση: 747

Αριθμός δεδομένων: 52

Μέγιστη χωρητικότητα οχημάτων: 100

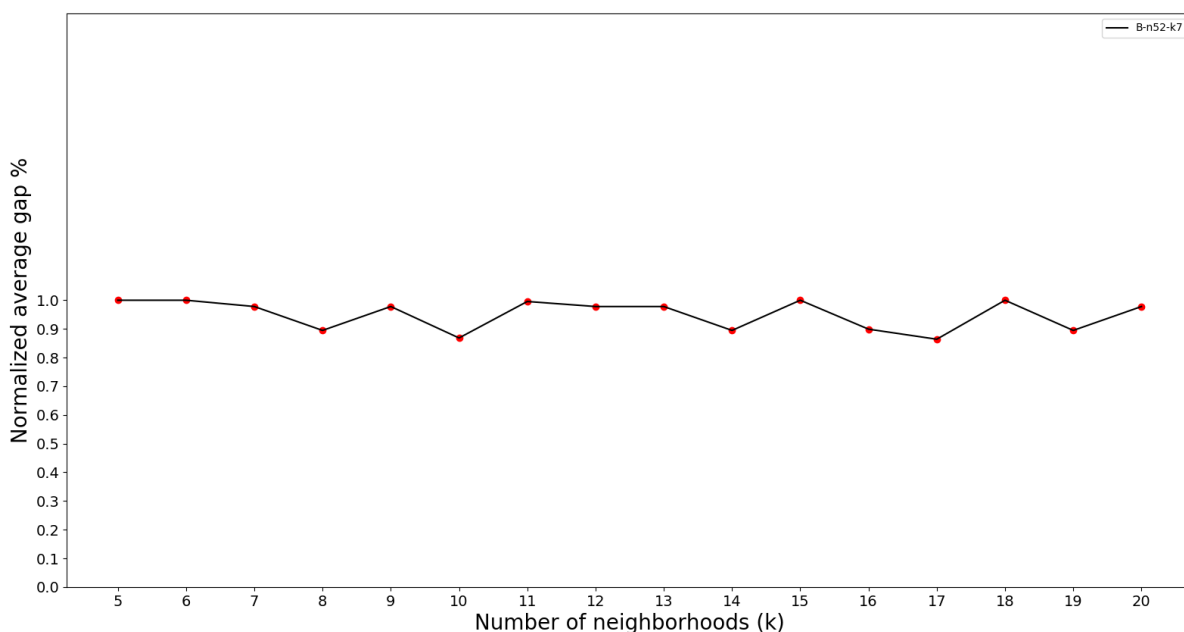
B-n57-k9

Αριθμός οχημάτων: 9

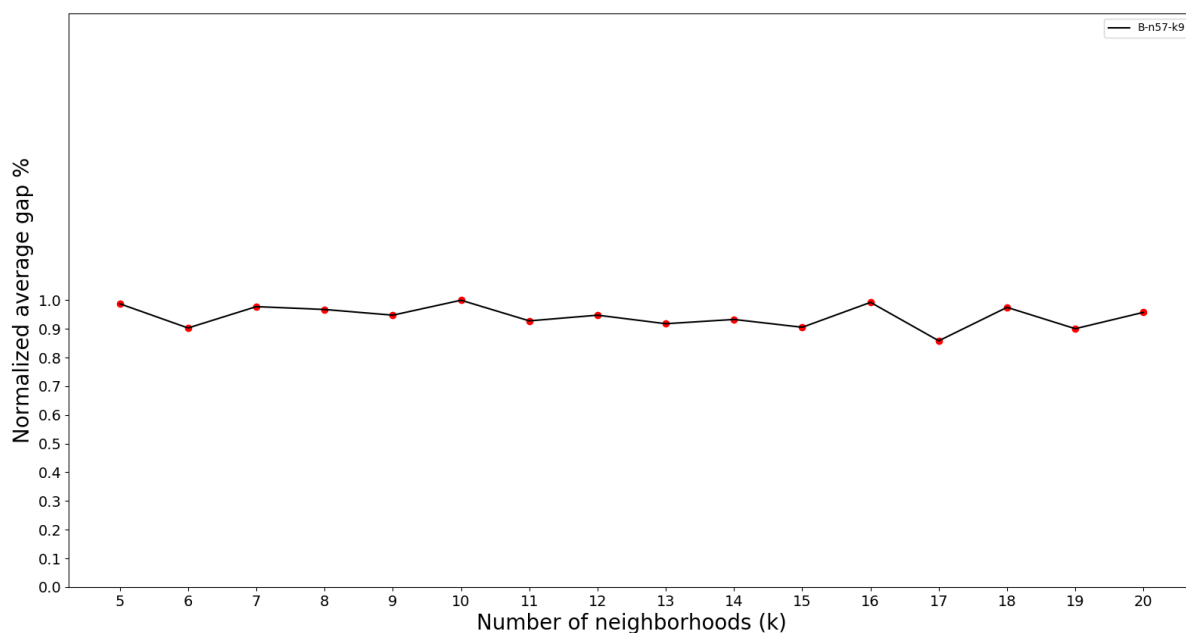
Βέλτιστη λύση: 1598

Αριθμός δεδομένων: 57

Μέγιστη χωρητικότητα οχημάτων: 100



Σχήμα 26: Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση B-n52-k7, για τιμές $k = 5 \dots 20$.



Σχήμα 27: Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση B-n57-k9 , για τιμές $k = 5 \dots 20$.

Από τα παραπάνω διαγράμματα (Σχήμα 26 και 27), δεν φαίνεται να προκύπτει κάποιο συγκεκριμένο συμπέρασμα, όσον αφορά την καλύτερη επιλογή του k_{max} κατά την εκτέλεση της VNS. Πιο συγκεκριμένα, στα δύο παραδείγματα που χρησιμοποιήθηκαν και ανήκουν στην κατηγορία των περιπτώσεων μικρής δυσκολίας, δεν αποδεικνύεται ότι μεγαλύτερος αριθμός k_{max} στην VNS συνεπάγεται και καλύτερα τελικά αποτελέσματα.

Ωστόσο, παρατηρήθηκε ότι και στους δύο περιπτώσεις ο αριθμός $k_{max} = 17$, οδήγησε στην καλύτερη τελική λύση.

5.2.2 A-n60-k9 & B-n78-k10

Οι δύο συγκεκριμένες περιπτώσεις ανήκουν στην κατηγορία μεσαίας δυσκολίας προβλημάτων CVRP. Για την κάθε μία περίπτωση δίνονται οι ακόλουθες πληροφορίες:

A-n60-k9

Αριθμός οχημάτων: 9

Βέλτιστη λύση: 1354

Αριθμός δεδομένων: 60

Μέγιστη χωρητικότητα οχημάτων: 100

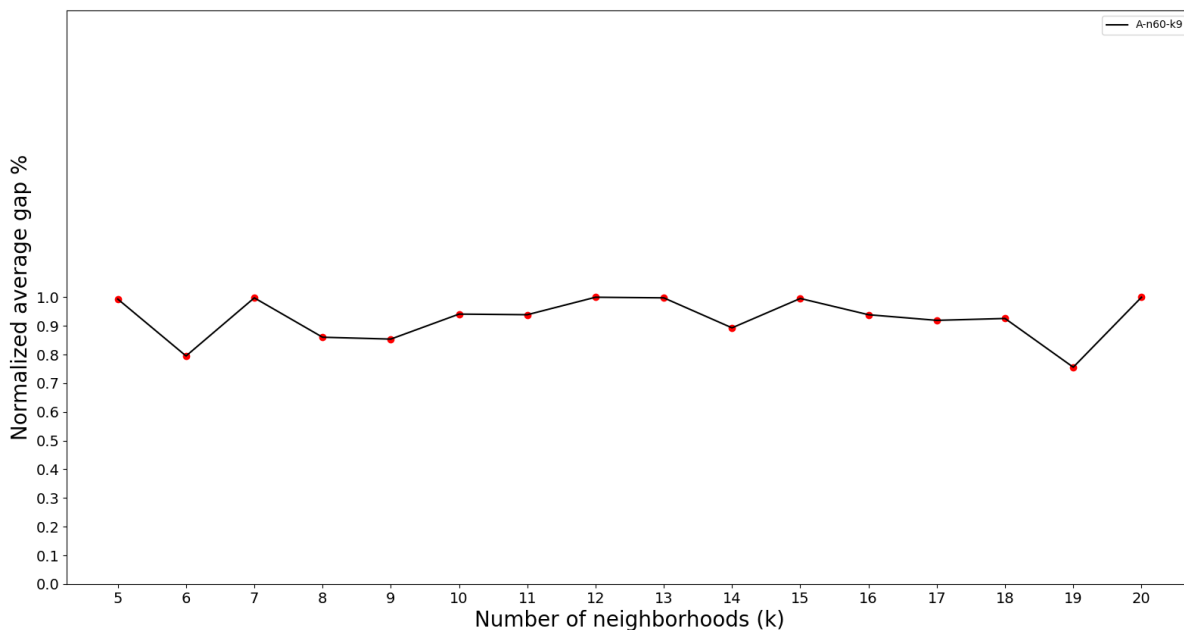
B-n78-k10

Αριθμός οχημάτων: 10

Βέλτιστη λύση: 1221

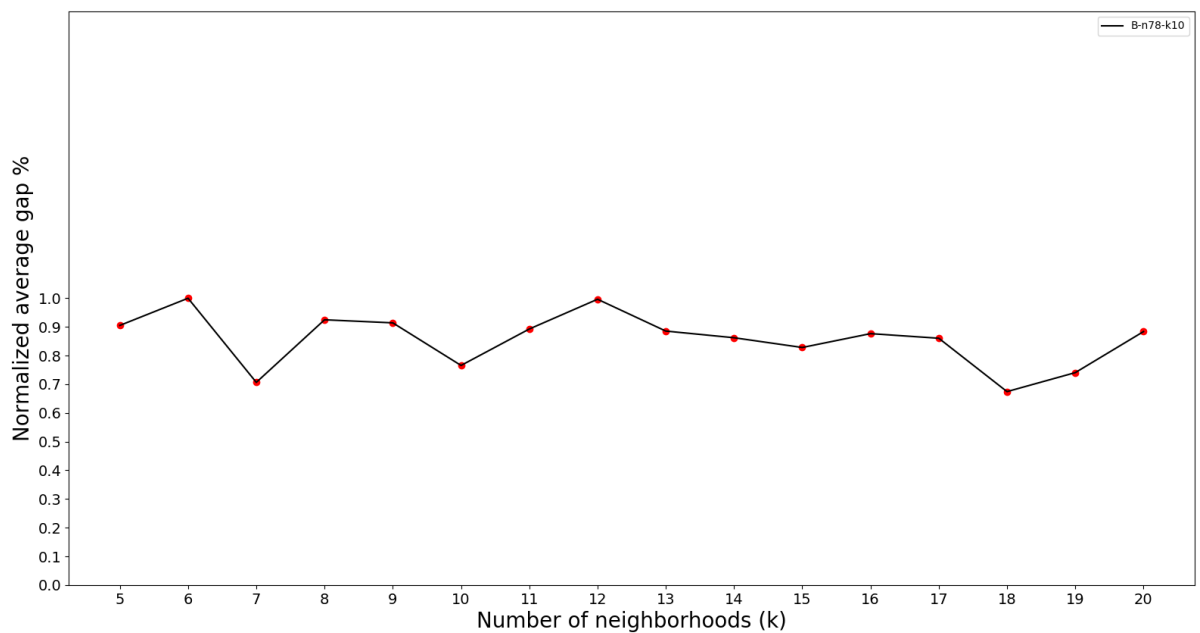
Αριθμός δεδομένων: 78

Μέγιστη χωρητικότητα οχημάτων: 100



Σχήμα 28: Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση A-n60-k9 , για τιμές $k = 5 \dots 20$.

Από τα παραπάνω διαγράμματα (Σχήμα 28 και 29), φαίνεται ότι και στην συγκεκριμένη περίπτωση, διαφορετικός αριθμός k_{max} οδηγεί και σε διαφορετική ποιότητα λύσης. Μικρή απόκλιση από την καλύτερη λύση, παρατηρείται ότι παρουσιάζει ο αριθμός k_{max} 6 και 7, για την πρώτη και την δεύτερη περίπτωση αντίστοιχα. Ωστόσο, οι καλύτερες λύσεις φαίνεται ότι παρατηρήθηκαν πάλι και για τις δύο περιπτώσεις για μεγάλο αριθμό k_{max} , πιο συγκεκριμένα $k_{max} = 19$ για την περίπτωση A-n60-k9 και $k_{max} = 18$ για την περίπτωση B-n78-k10.



Σχήμα 29: Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση B-n78-k10 , για τιμές $k = 5 \dots 20$.

5.2.3 A-n80-k10 & E-n101-k14

Οι δύο συγκεκριμένες περιπτώσεις ανήκουν στην κατηγορία μεγάλης δυσκολίας προβλημάτων CVRP. Για την κάθε μία περίπτωση δίνονται οι ακόλουθες πληροφορίες:

A-n80-k10

Αριθμός οχημάτων: 10

Βέλτιστη λύση: 1763

Αριθμός δεδομένων: 80

Μέγιστη χωρητικότητα οχημάτων: 100

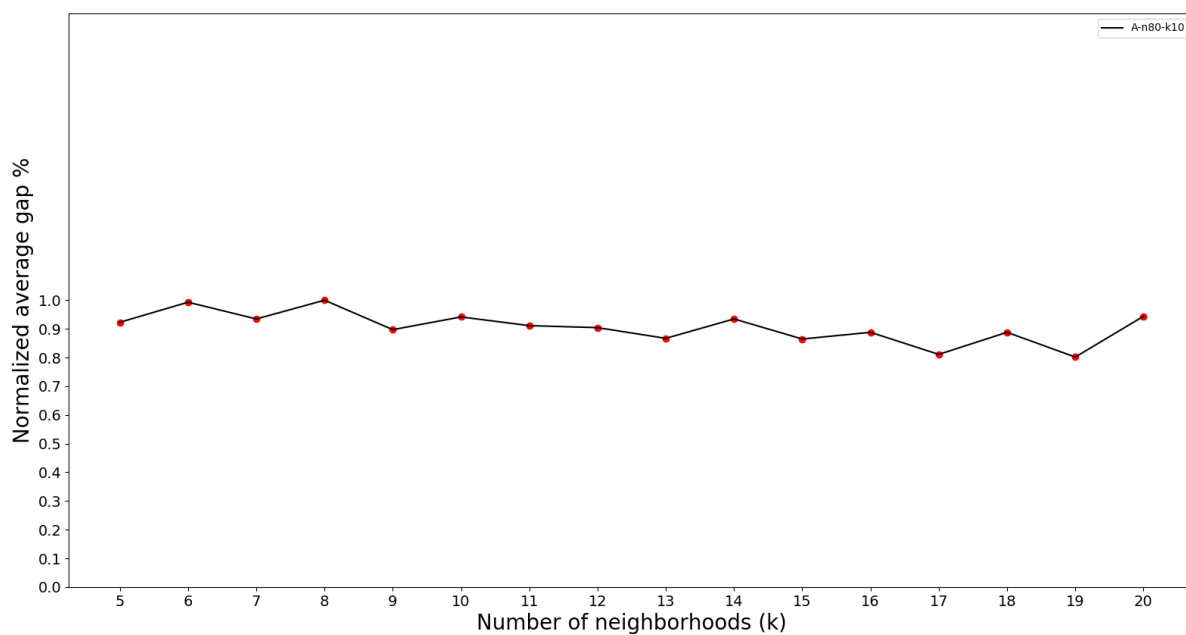
E-n101-k14

Αριθμός οχημάτων: 14

Βέλτιστη λύση: 1067

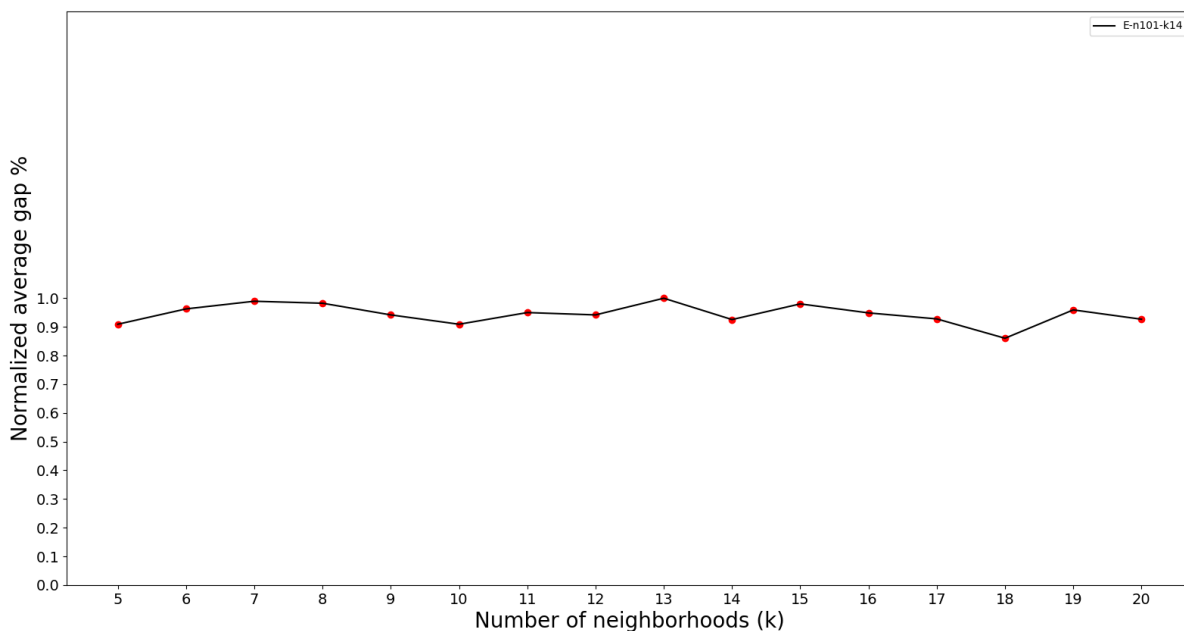
Αριθμός δεδομένων: 101

Μέγιστη χωρητικότητα οχημάτων: 112



Σχήμα 30: Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση A-n80-k10 , για τιμές $k = 5 \dots 20$.

Παρατηρώντας τα διαγράμματα (Σχήμα 30 και 31), φαίνεται και εδώ ότι οι καλύτερες λύσεις και στις δύο περιπτώσεις παρατηρήθηκαν για μεγάλες τιμές του k_{max} . Πιο συγκεκριμένα, για την περίπτωση A-n80-k10 η καλύτερη λύση φαίνεται ότι επιτυγχάνεται για $k_{max} = 19$, ενώ για την περίπτωση E-n101-k14 για $k_{max} = 18$



Σχήμα 31: Μέσος όρος απόκλισης από την καλύτερη γνωστή λύση για την περίπτωση E-n101-k14 , για τιμές $k = 5...20$.

Στο επόμενο υποκεφάλαιο, παρουσιάζονται σε πίνακες τα αποτελέσματα για κάθε σύνολο δεδομένων A, B και E αντίστοιχα. Τα στοιχεία που περιέχει ο κάθε πίνακας είναι τα εξής:

- στην πρώτη στήλη βρίσκεται το όνομα της κάθε περίπτωσης, το οποίο παράλληλα περιέχει πληροφορίες για τον αριθμό των δεδομένων και τον αριθμό των διαδρομών.
- στην δεύτερη στήλη βρίσκεται η καλύτερη λύση η οποία έχει βρεθεί ως τώρα για την κάθε περίπτωση (BKS).
- η τρίτη στήλη αφορά την καλύτερη λύση, η οποία προέκυψε από την εκτέλεση του αλγορίθμου της παρούσας διατριβής.
- στην τέταρτη στήλη, παρατηρείται ο χρόνος εκτέλεσης του αλγορίθμου μέχρι να φτάσει στην τελική λύση.
- τέλος, στην πέμπτη στήλη βρίσκεται η απόκλιση της λύσης του αλγορίθμου της διατριβής, σε σχέση με την λύση των (BKS).

Σημειώνεται ότι για την κάθε περίπτωση, ο αλγόριθμος εκτελέστηκε 5 φορές. Στο τέλος, επιλέχθηκε εκείνη η λύση η οποία παρουσίασε την μικρότερη απόκλιση.

5.3 Πίνακες αποτελεσμάτων

5.3.1 Σετ δεδομένων A

Πίνακας 1: Σετ δεδομένων A

Instance	-- BKS --	-- Best --	-- routes --	Cpu Time (s)	% Dev
A-n32-k5	784	828.73	5	74	5.71
A-n33-k5	661	669.31	5	151	1.26
A-n33-k6	742	744.24	6	67	0.3
A-n34-k5	778	790.61	5	59	1.62
A-n36-k5	799	806.78	5	51	0.97
A-n37-k5	669	686.37	5	171	2.6
A-n37-k6	949	970.84	6	38	2.3
A-n38-k5	730	737.05	6	204	0.69
A-n39-k5	822	838.99	5	77	2.07
A-n39-k6	831	842.72	6	52	1.41
A-n44-k6	937	981.09	6	217	4.71
A-n45-k6	944	962.2	7	76	1.93
A-n45-k7	1146	1170.44	7	103	2.13
A-n46-k7	914	918.13	7	103	0.45
A-n48-k7	1073	1088.28	7	125	1.42
A-n53-k7	1010	1049.28	7	147	3.89
A-n54-k7	1167	1185.99	8	157	1.63
A-n55-k9	1073	1082.54	9	84	0.89
A-n60-k9	1354	1380.86	9	167	1.98
A-n61-k9	1034	1045.01	10	171	1.06
A-n62-k8	1288	1323.65	8	153	2.77

Instance	-- BKS --	-- Best --	-- routes --	Cpu Time (s)	% Dev
A-n63-k9	1616	1642.99	10	103	1.67
A-n63-k10	1314	1332.16	10	125	1.38
A-n64-k9	1401	1481.33	10	86	5.73
A-n65-k9	1174	1191.27	9	176	1.47
A-n69-k9	1159	1205.58	9	354	4.02
A-n80-k10	1763	1793.72	10	212	1.74

5.3.2 Σετ δεδομένων B

Πίνακας 2: Σετ δεδομένων B

Instance	-- BKS --	-- Best --	-- routes --	Cpu Time (s)	% Dev
B-n31-k5	672	686.99	5	85	2.23
B-n34-k5	788	796.72	6	96	1.11
B-n35-k5	955	956.3	5	94	0.14
B-n38-k6	805	811.15	6	137	0.76
B-n39-k5	549	563.55	5	201	2.65
B-n41-k6	829	833.82	6	135	0.58
B-n43-k6	742	758.85	6	143	2.27
B-n44-k7	909	929.39	7	48	2.24
B-n45-k5	751	753.96	6	39	0.39
B-n45-k6	678	717.67	7	138	5.85
B-n50-k7	741	745.17	7	91	0.56
B-n50-k8	1312	1326.84	8	112	1.13
B-n51-k7	1032	1040.27	8	202	0.8

Instance	-- BKS --	-- Best --	-- routes --	Cpu Time (s)	% Dev
B-n52-k7	747	757.05	7	123	1.35
B-n56-k7	707	721.97	7	267	2.12
B-n57-k7	1153	1166.84	8	206	1.2
B-n57-k9	1598	1643.76	9	189	2.86
B-n63-k10	1496	1578.8	10	119	5.53
B-n64-k9	861	904.81	10	122	5.09
B-n66-k9	1316	1347.7	10	168	2.41
B-n67-k10	1032	1086.72	10	162	5.3
B-n68-k9	1272	1305.54	9	255	2.64
B-n78-k10	1221	1252.99	10	175	2.62

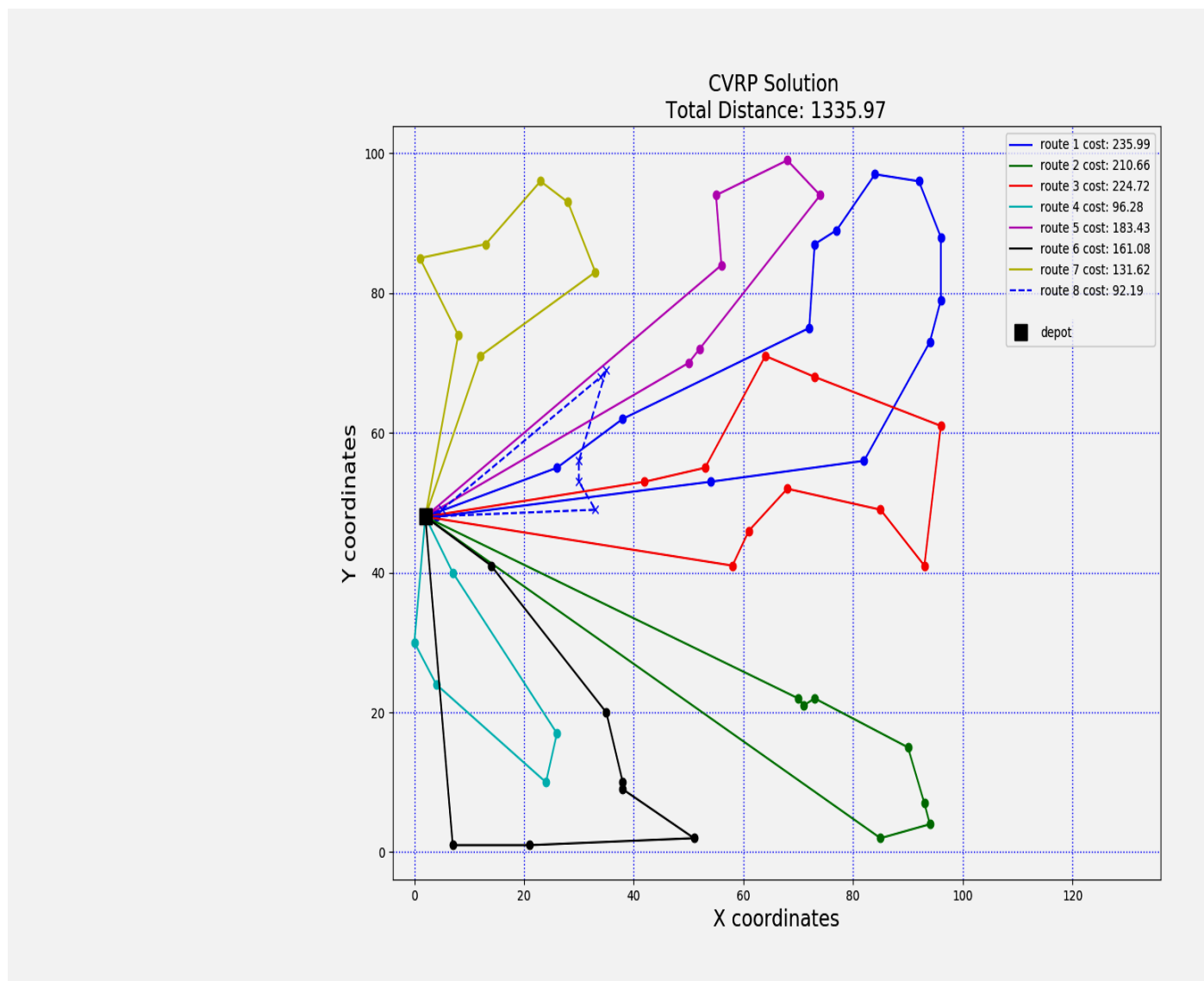
5.3.3 Σετ δεδομένων E

Πίνακας 3: Σετ δεδομένων E

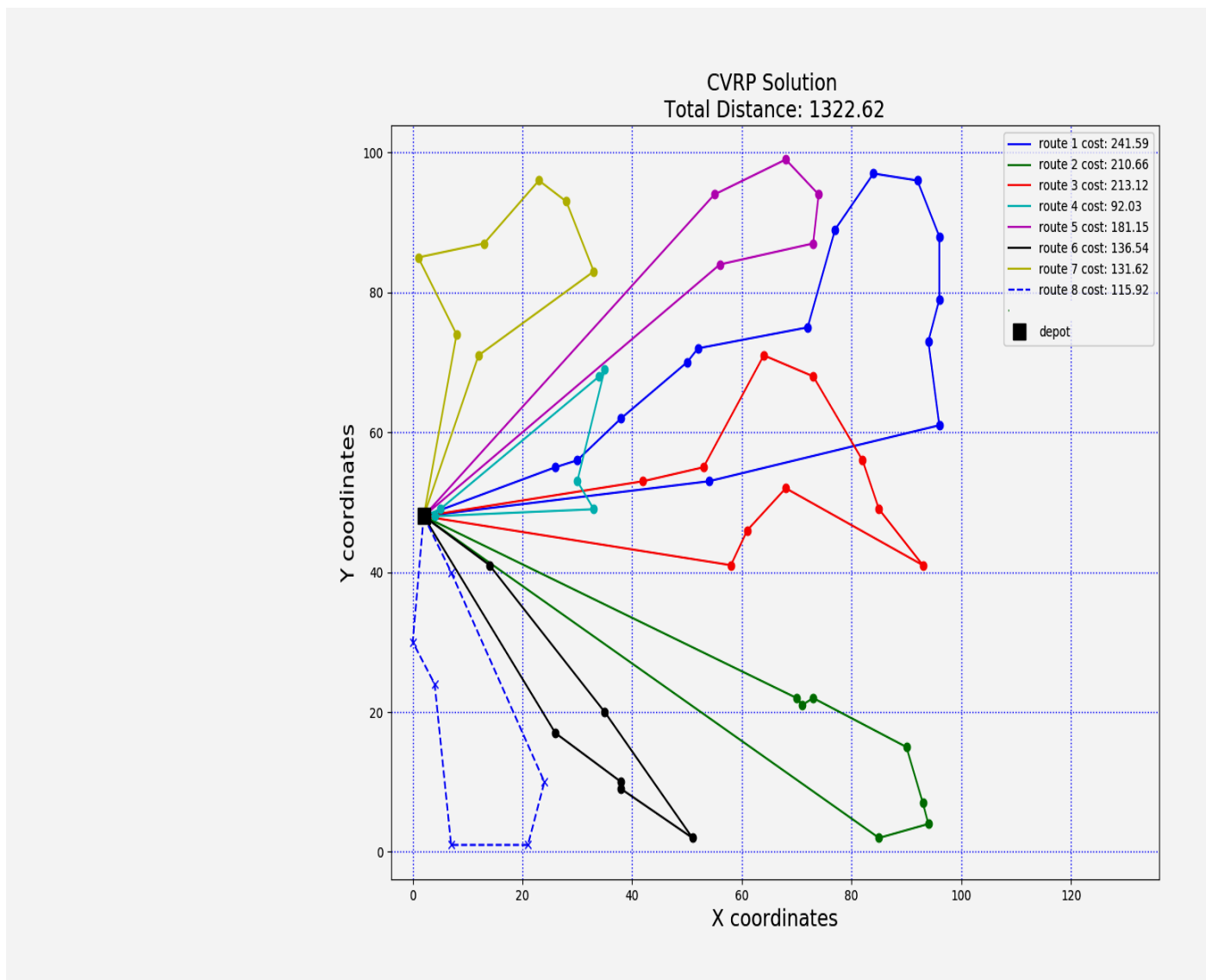
Instance	-- BKS --	-- Best --	-- routes --	Cpu Time (s)	% Dev
E-n22-k4	375	384.06	4	19	2.41
E-n23-k3	569	569.75	3	76	0.13
E-n33-k4	835	837.67	4	88	0.32
E-n51-k5	521	537.37	5	129	3.14
E-n76-k7	682	707.23	7	561	3.7
E-n76-k8	735	751.07	8	531	2.19
E-n76-k10	830	864.75	11	137	4.19
E-n76-k14	1021	1058.23	15	99	3.6
E-n101-k8	815	846.55	8	625	3.87
E-n101-k14	1067	1134.16	14	308	6.29

5.4 Σχήματα λύσεων

5.4.1 A-n62-k8

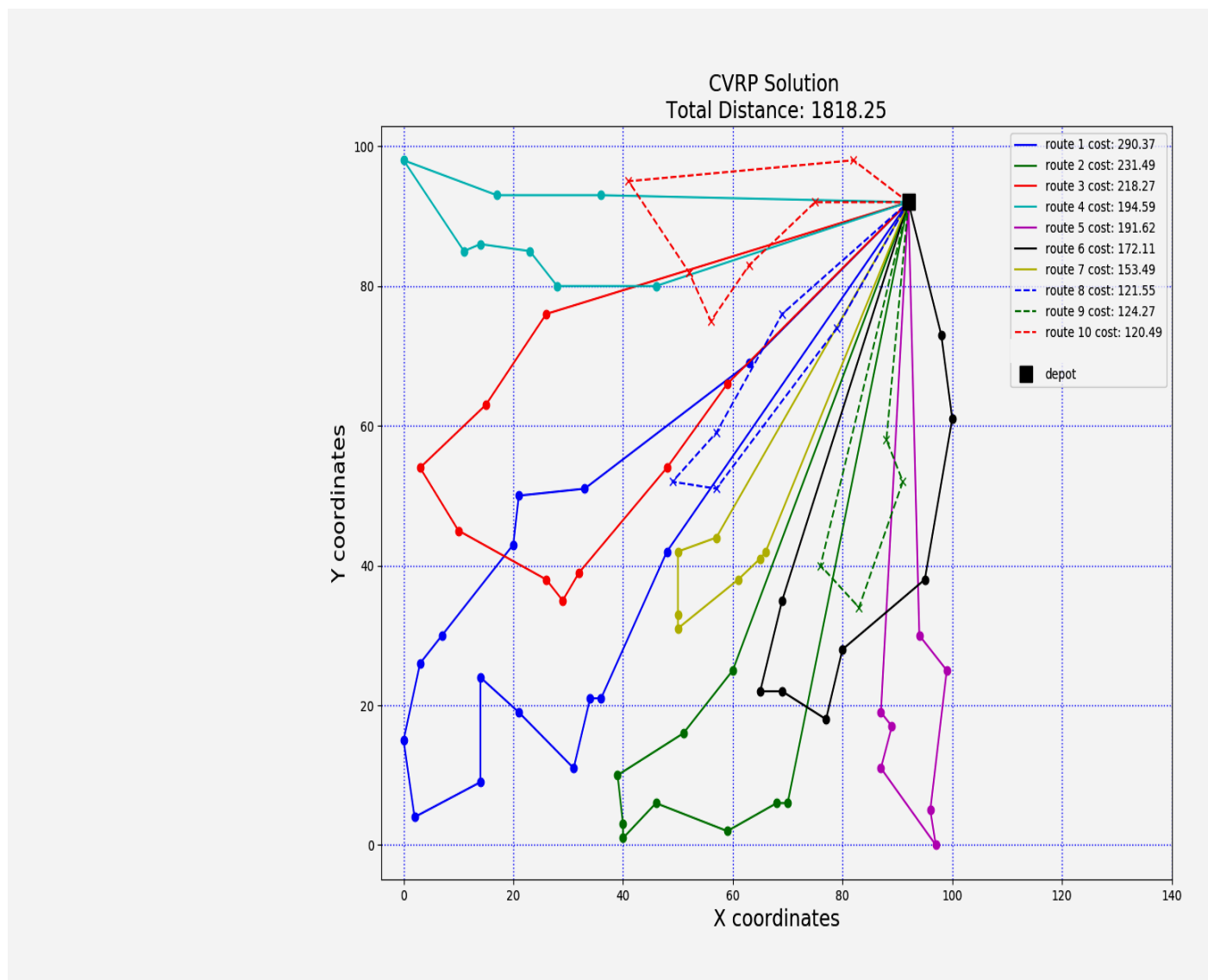


Σχήμα 32: Instance A - n62 - k8, $k_{max} = 5$, $dev = 3.72\%$

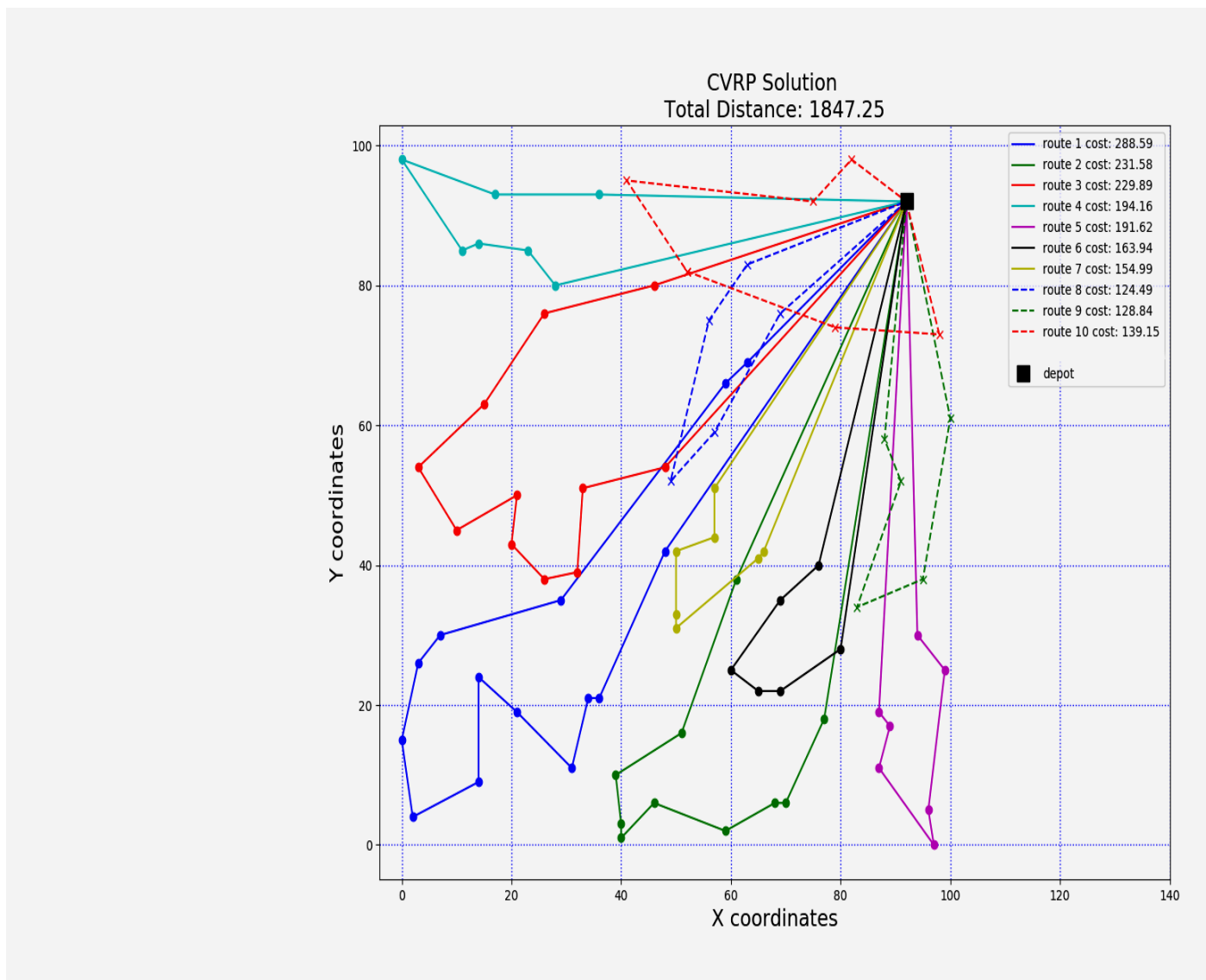


Σχήμα 33: *InstanceA* – $n62 - k8$, $k_{max} = 19$, $dev = 2.69\%$

5.4.2 A-n80-k10

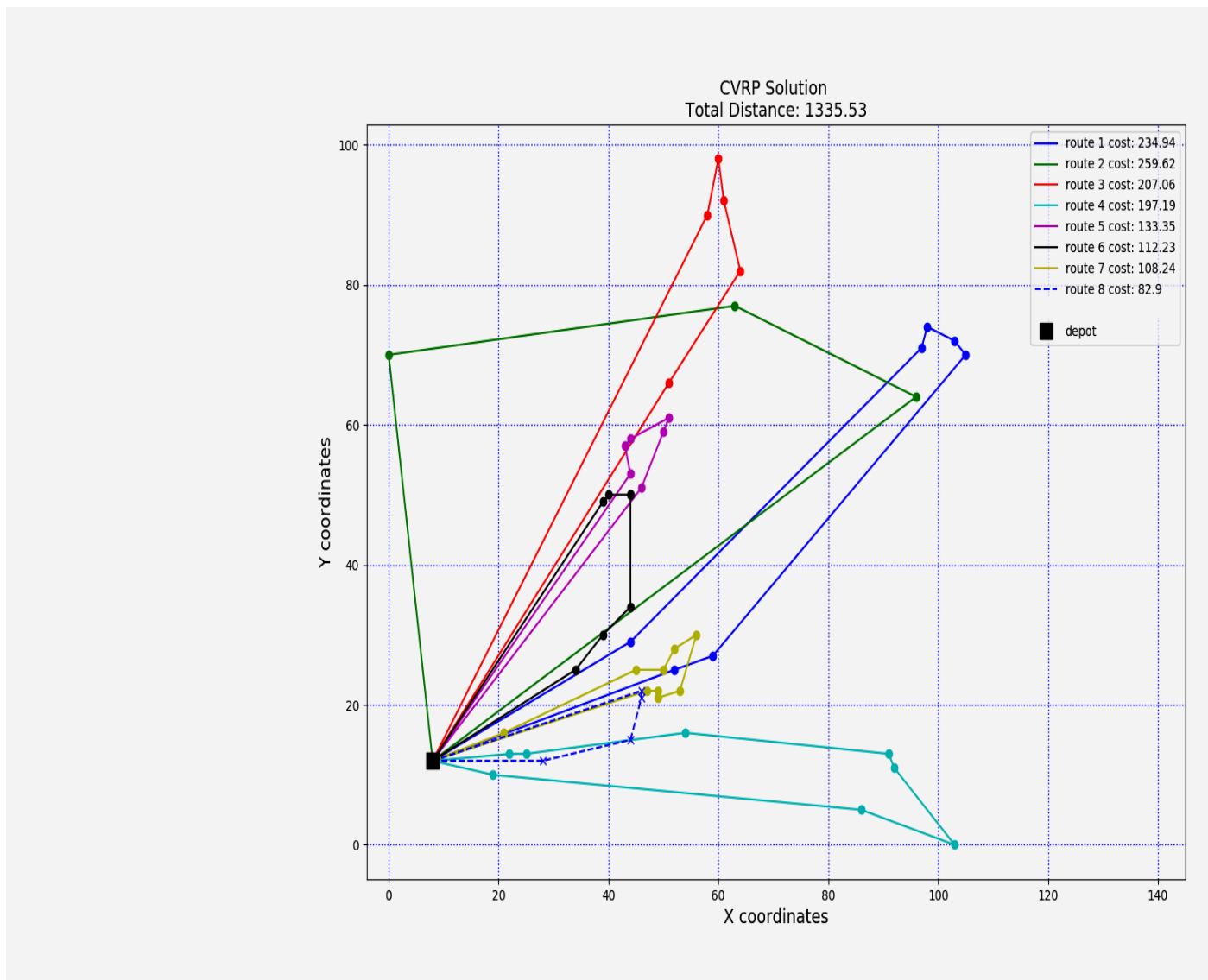


Σχήμα 34: Instance A – n80 – k10, $k_{max} = 5$, $dev = 3.13\%$

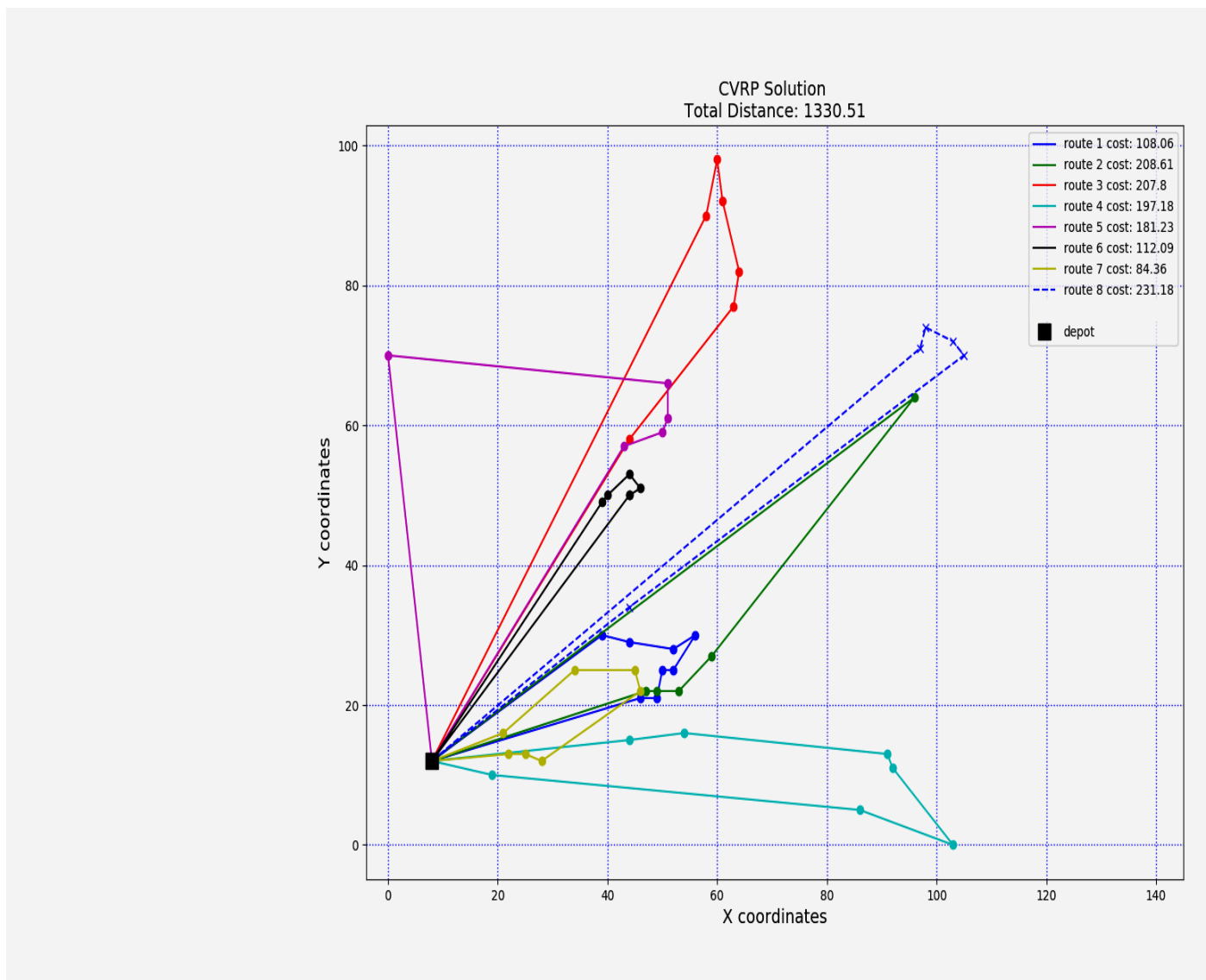


Σχήμα 35: *InstanceA - n80 - k10*, $k_{max} = 19$, $dev = 4.78\%$

5.4.3 B-n50-k8

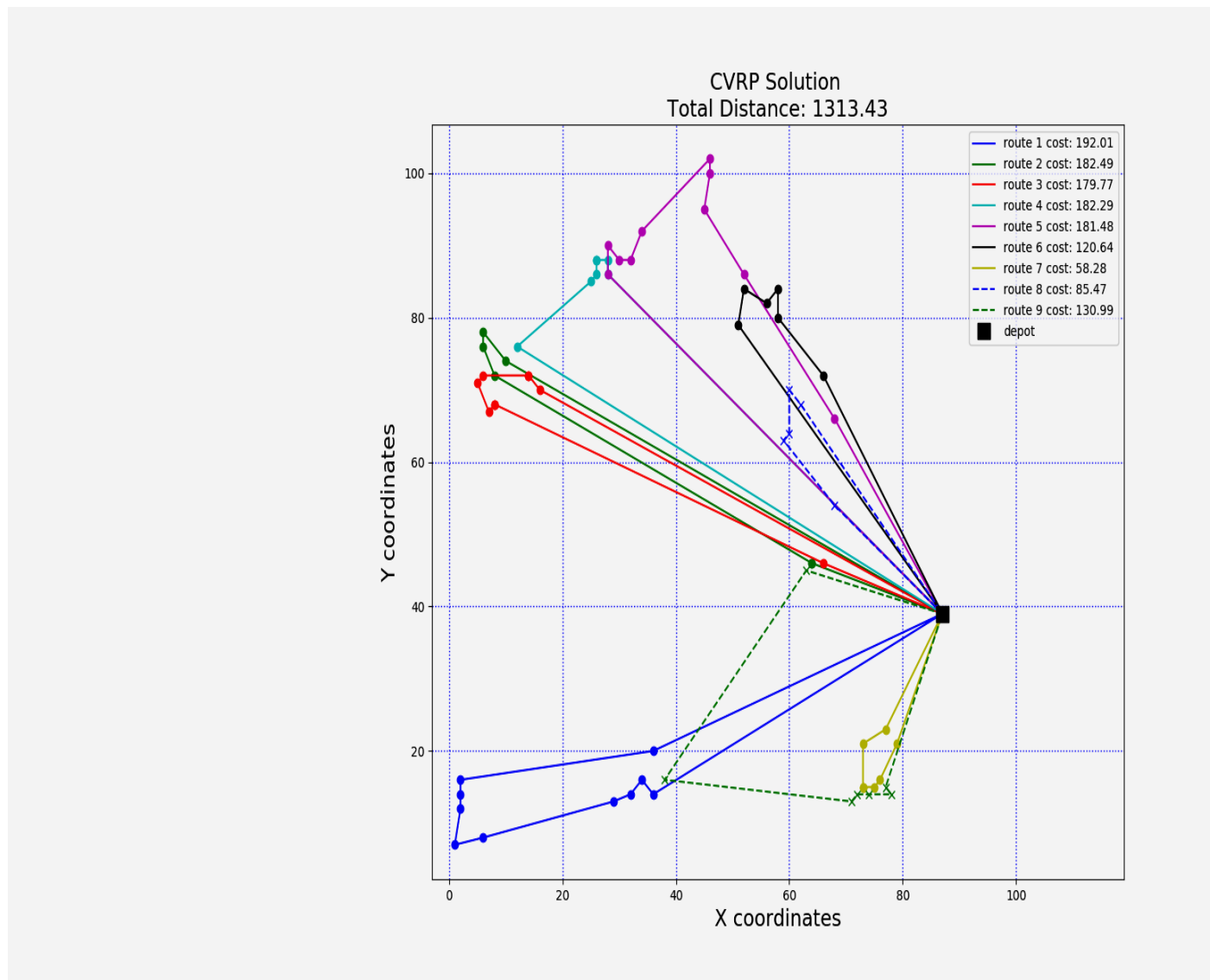


Σχήμα 36: Instance B - n50 - k8, $k_{max} = 5$, $dev = 1.35\%$

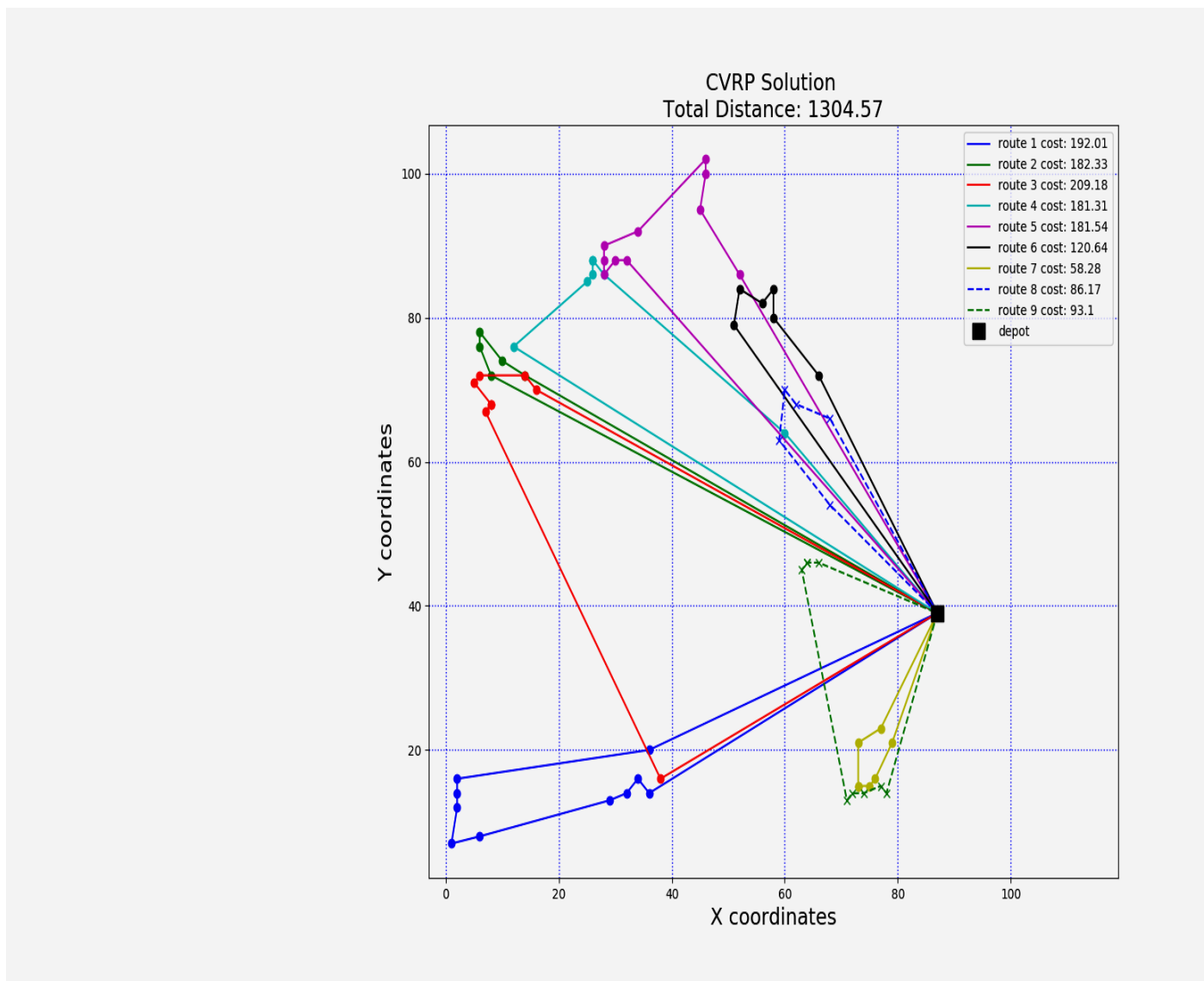


Σχήμα 37: InstanceB - n50 - k8, $k_{max} = 19$, $dev = 1.41\%$

5.4.4 B-n68-k9

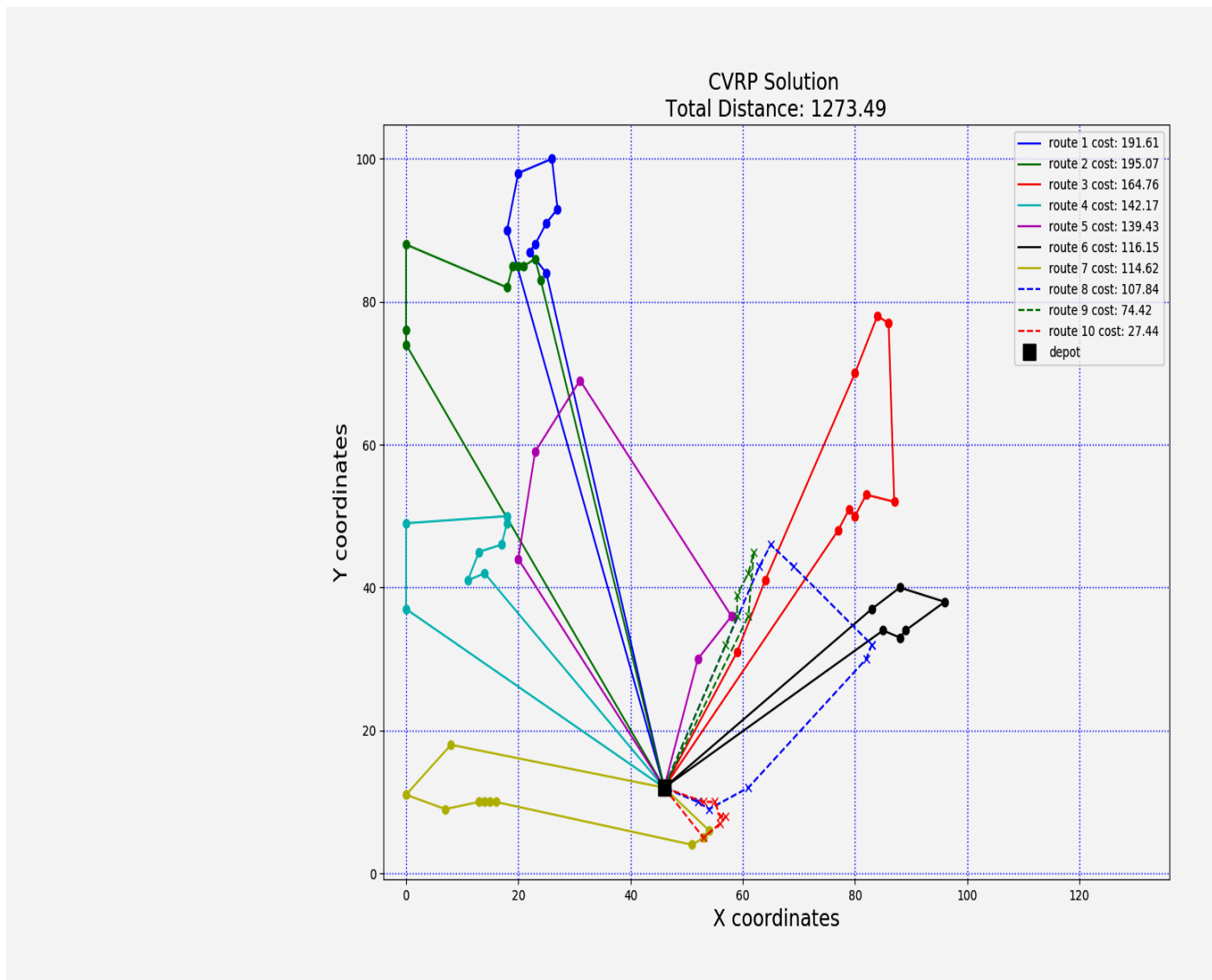


Σχήμα 38: Instance B - n68 - k9, $k_{max} = 5$, $dev = 3.26\%$

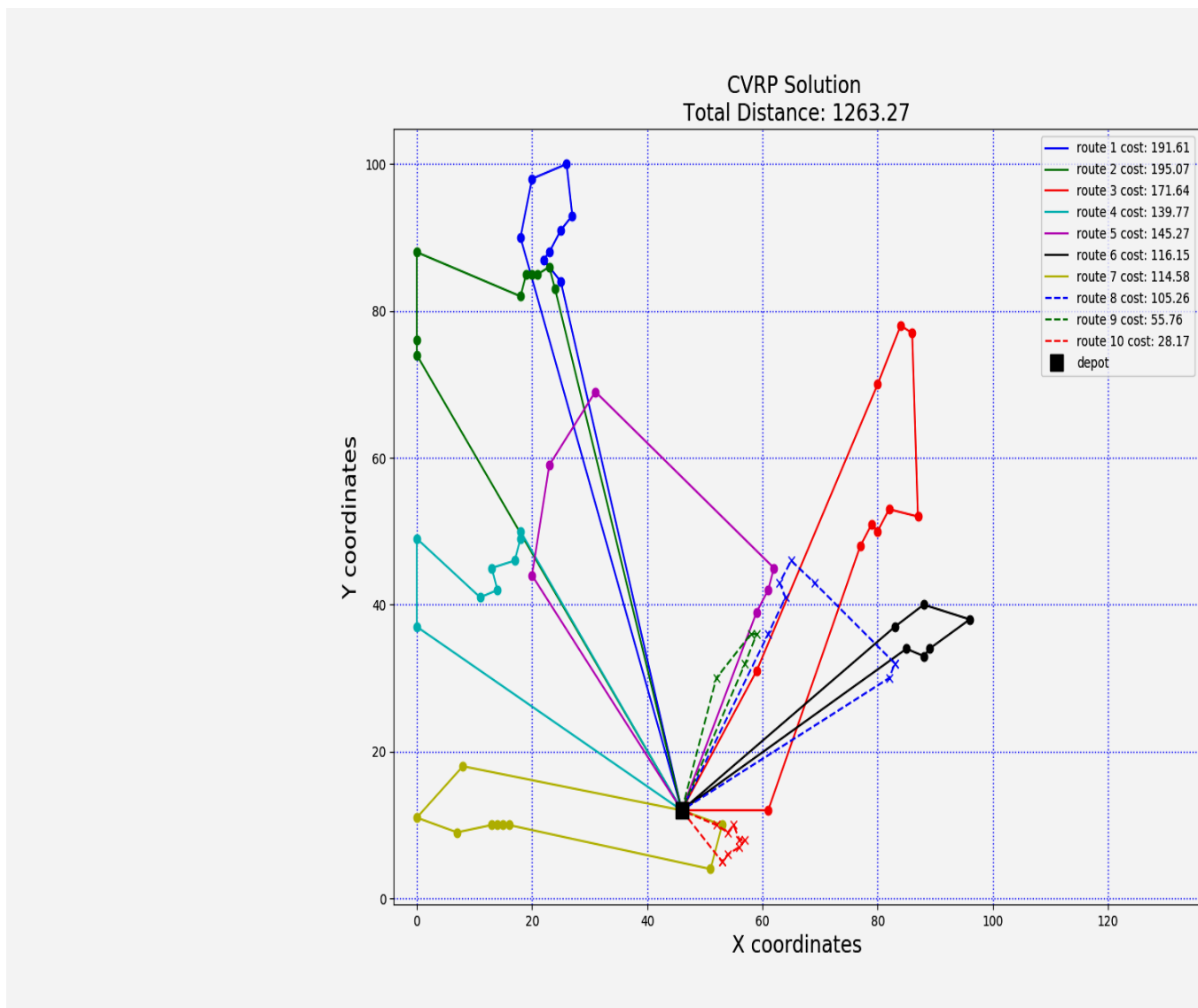


Σχήμα 39: *InstanceB* – $n68 - k9$, $k_{max} = 19$, $dev = 2.56\%$

5.4.5 A-n78-k10

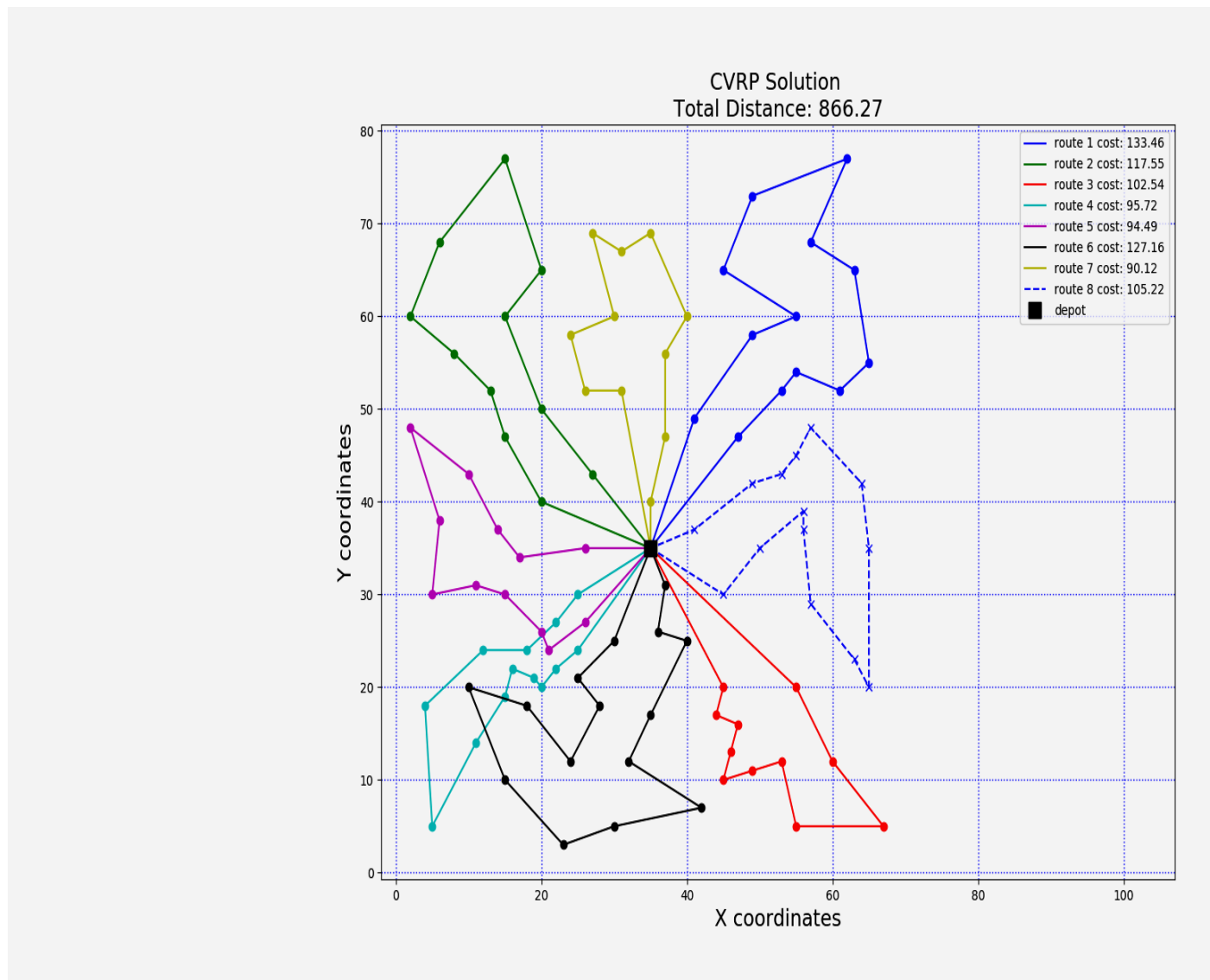


Σχήμα 40: Instance B - n78 - k10, $k_{max} = 5$, $dev = 4.3\%$

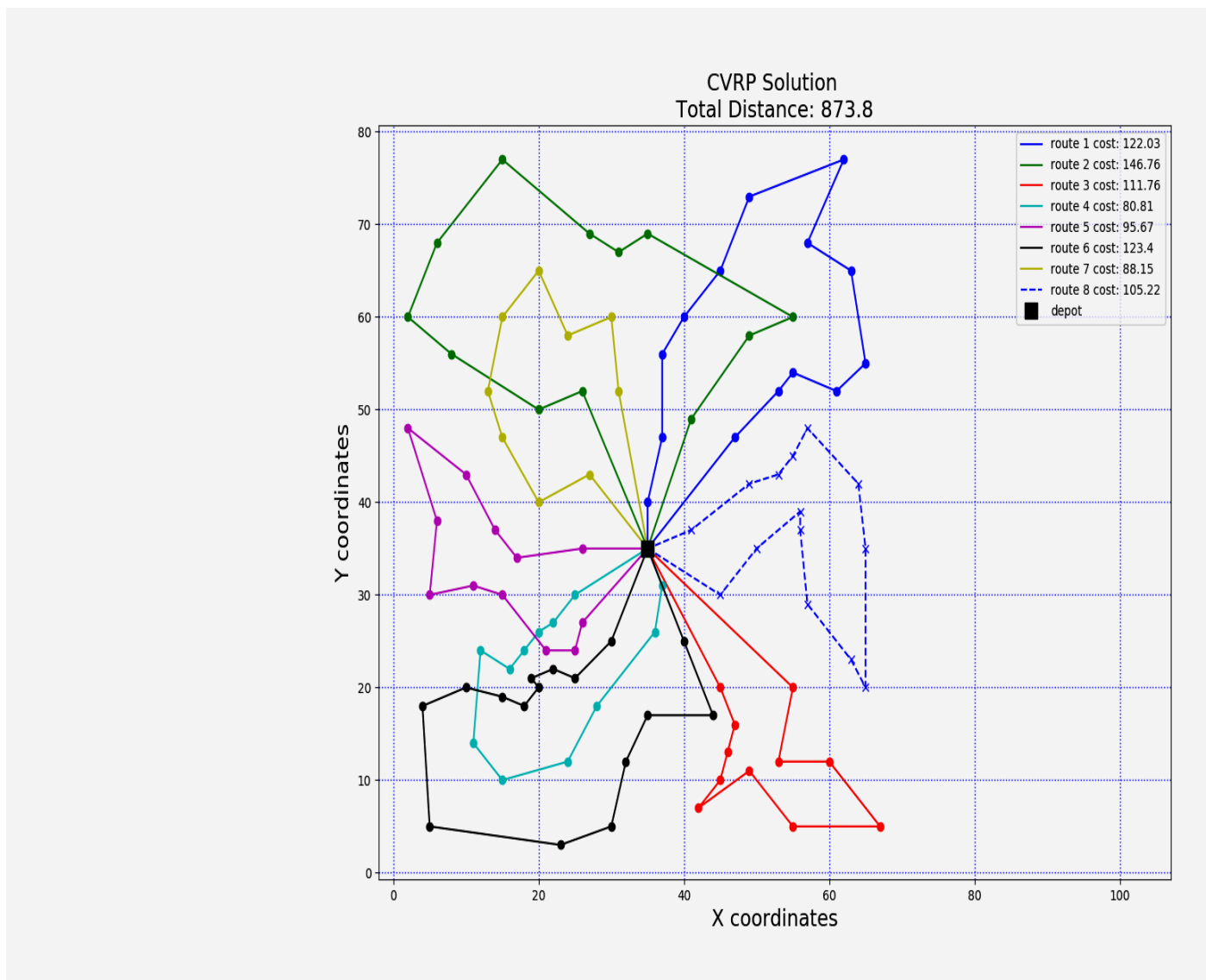


Σχήμα 41: InstanceB – n78 – k10, $k_{max} = 19$, $dev = 3.46\%$

5.4.6 E-n101-k8

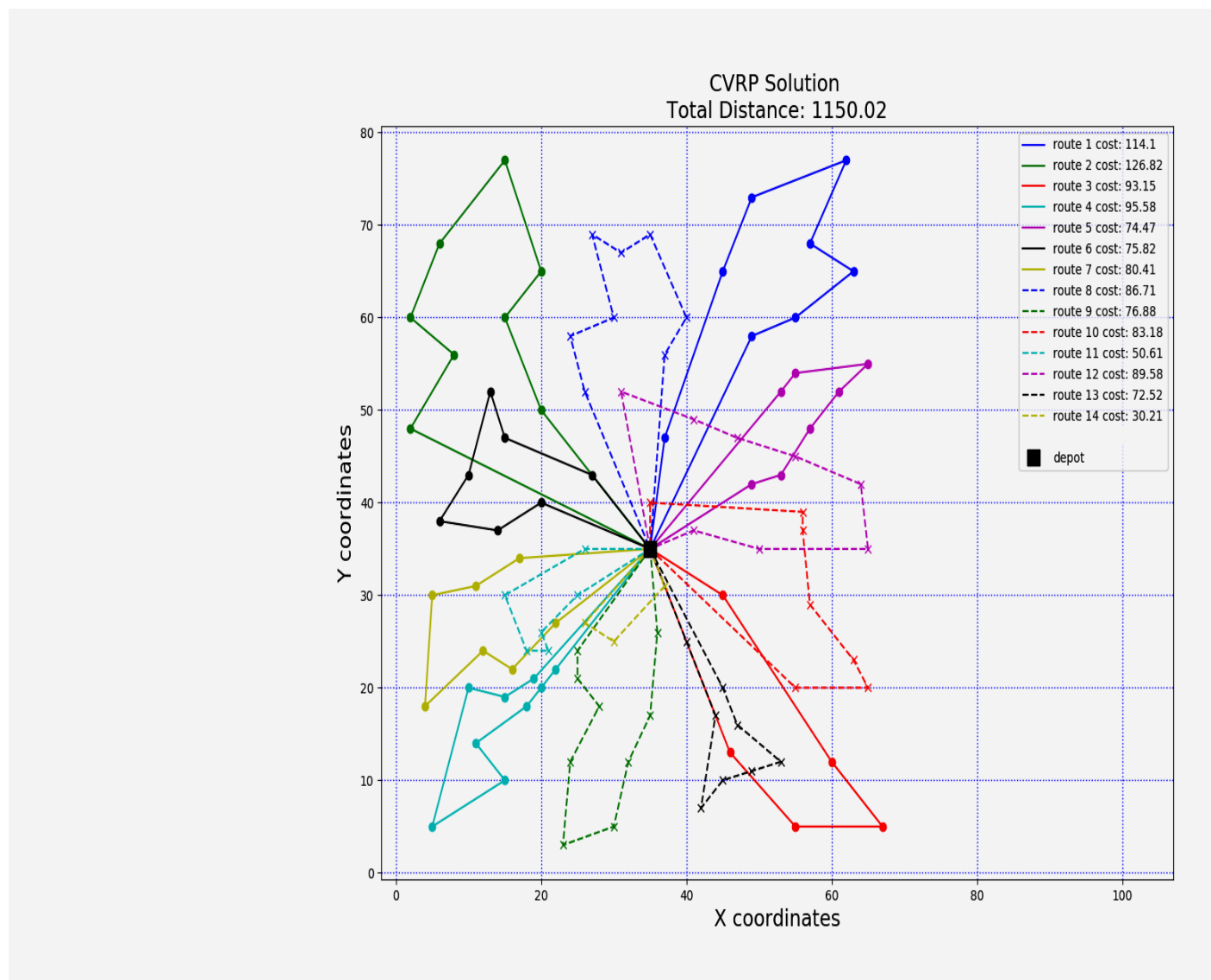


Σχήμα 42: Instance E - n101 - k8, $k_{max} = 5$, $dev = 6.29\%$

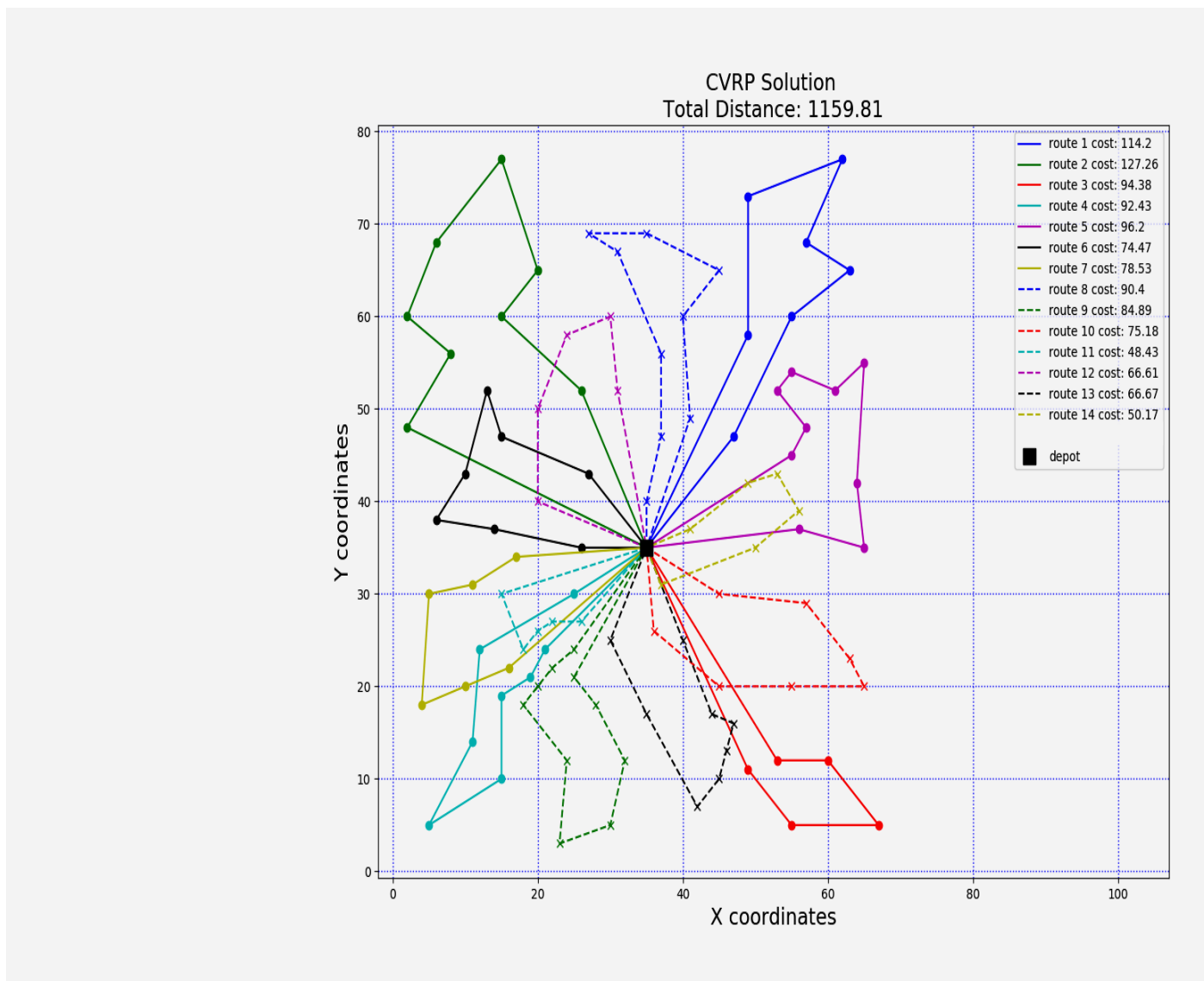


Σχήμα 43: Instance E - n101 - k8, $k_{max} = 19$, $dev = 7.21\%$

5.4.7 E-n101-k14



Σχήμα 44: Instance E - n101 - k14, $k_{max} = 5$, $dev = 7.78\%$



Σχήμα 45: Instance E - n101 - k14, $k_{max} = 19$, $dev = 8.7\%$

6 Συμπεράσματα και προτάσεις

Στην παρούσα διατριβή, χρησιμοποιήθηκε ένας μειθευρετικός αλγόριθμος μεταβλητής γειτονιάς αναζήτησης VNS για την επίλυση του προβλήματος CVRP. Πριν την εφαρμογή της VNS, ένας αλγόριθμος συσταδοποίησης της κατηγορίας centroid based εφαρμόστηκε στα δεδομένα, έτσι ώστε αυτά να ομαδοποιηθούν σε συστάδες και με την χρήση του nearest neighbour construction heuristic κατασκευάστηκαν οι αρχικές διαδρομές.

Τα αποτελέσματα τα οποία προέκυψαν από την χρήση του παραπάνω αλγορίθμου, φαίνεται να είναι ικανοποιητικά, καθώς σε λογικό χρόνο εξάγονται αρκετά καλές λύσεις. Πιο συγκεκριμένα, για το σετ δεδομένων A, ο μέσος όρος απόκλισης όλων των λύσεων υπολογίστηκε ότι είναι 2.14 %, για το σετ δεδομένων B 2.25 %, ενώ για το σετ δεδομένων E 2.99 %. Όσον αφορά την κατηγοριοποίηση των περιπτώσεων με βάση τον βαθμό δυσκολίας επίλυσης αυτών, ο μέσος όρος απόκλισης των λύσεων υπολογίστηκε ότι είναι 1.88 % για τις εύκολες περιπτώσεις, 2.16 % για τις περιπτώσεις μεσαίου βαθμού δυσκολίας και 3.86 % για τις περιπτώσεις μεγάλου βαθμού δυσκολίας.

Ορισμένες προτάσεις, οι οποίες θα μπορούσαν να οδηγήσουν σε περαιτέρω βελτίωση των αποτελεσμάτων του αλγορίθμου, θα ήταν ίσως η χρήση επιπλέον τελεστών τοπικής αναζήτησης (local search πχ 3 – opt ή επιπλέον τελεστών στο shaking phase της VNS, έτσι ώστε να προκύπτουν γειτονικές δομές αρκετά διαφορετικές σε σχέση με τις προηγούμενες. Τέλος, ενδιαφέρουσα θα ήταν και η εφαρμογή του παραπάνω αλγορίθμου με χρήση στρατηγικής αποδοχής καλύτερης λύσης το best improvement έναντι του first improvement, το οποίο χρησιμοποιήθηκε στην παρούσα διατριβή.

Αναφορές

- [1] Toth, P. and D. Vigo, Models, relaxations and exact approaches for the capacitated vehicle routing problem, *Discrete Applied Mathematics* 123 (2002), pp. 487–512.
- [2] Mazzeo, S. and I. Loiseau, An ant colony algorithm for the capacitated vehicle routing, *Electronic Notes in Discrete Mathematics* 18 (2004), pp. 181–186.
- [3] Sbai, I., Krichen, S., Limam, O., 2020. Two meta-heuristics for solving the capacitated vehicle routing problem: the case of the Tunisian Post Office. *Oper. Res.* <https://doi.org/10.1007/s12351-019-00543-8>
- [4] Mladenović, N. and P. Hansen, Variable neighborhood search, *Computers & Operations Research* 24 (1997), pp. 1097–1100.
- [5] Kozlenkova, Irina; et al. (2015). "The Role of Marketing Channels in Supply Chain Management". *Journal of Retailing*. 91 (4): 586–609. doi:10.1016/j.jretai.2015.03.003. Retrieved 28 September 2016.
- [6] Chopra, S. and P. Meindl (2001), *Supply Chain Management*, Prentice Hall
- [7] Chopra, S., Meindl, P., 2013. *Supply chain management: strategy, planning, and operation*, 5th ed. ed. Pearson, Boston.
- [8] <https://corporatefinanceinstitute.com/resources/knowledge/strategy/supply-chain/>
- [9] Lu, L.X., Swaminathan, J.M., 2015. *Supply Chain Management*, in: *International Encyclopedia of the Social & Behavioral Sciences*. Elsevier, pp. 709–713. <https://doi.org/10.1016/B978-0-08-097086-8.73032-7>
- [10] Szymonik, A., 2012. *Logistics and Supply Chain Management* 394.
- [11] U.S. Agency for International Development (USAID), 2011, *The Logistics Handbook A Practical Guide for the Supply Chain Management of Health Commodities*
- [12] Cattaruzza, D., Absi, N., Feillet, D., González-Feliu, J., 2017. Vehicle routing problems for city logistics. *EURO J. Transp. Logist.* 6, 51–79. <https://doi.org/10.1007/s13676-014-0074-0>

- [13] Commission of the European Communities (2009) A sustainable future for transport: Towards an integrated, technology-led and user friendly system. Technical Report COM, pp 279. <http://ec.europa.eu>
- [14] Hajghasem, M., shojaie, A.A., 2016. Optimal Routing in Supply Chain Aimed at Minimizing Vehicle Cost and Supply. *Procedia Econ. Finance* 36, 353–362. [https://doi.org/10.1016/S2212-5671\(16\)30047-8](https://doi.org/10.1016/S2212-5671(16)30047-8)
- [15] Zou, H., Zhang, T., 2018. Research on Vehicle Routing Algorithm for Supply Chain Logistics Distribution. *MATEC Web Conf.* 227, 02003. <https://doi.org/10.1051/mateconf/201822702003>
- [16] <https://www.warehouseoptimization.com/autoo2-transportation-optimization-software>
- [17] Lambert, D. and Martha C. Cooper (2000) Issues in Supply Chain Management, *Industrial Marketing Management* 29, 65–83
- [18] Jack Van der Vorst, 2004, Supply Chain Management: theory and practices https://www.researchgate.net/publication/40122004supplychainmanagement_theory_and_practices
- [19] A. Goel and V. Gruhn, “A general vehicle routing problem,” *European Journal of Operational Research*, vol. 191, no. 3, pp. 650 – 660, 2008.
- [20] Poulimenos, D., Bechtsis, D., Vlachos, D., Iakovou, E., 2018. A software tool for optimizing intra-logistic activities, in: *Computer Aided Chemical Engineering*. Elsevier, pp. 439–444. <https://doi.org/10.1016/B978-0-444-64235-6.50079-6>
- [21] <https://en.wikipedia.org/wiki/NP-hardness>
- [22] https://en.wikipedia.org/wiki/Vehicle_routing_problem
- [23] <http://neo.lcc.uma.es/dynamic/vrp.html>
- [24] Fosin, J., Carić, T., Ivanjko, E., 2014. Vehicle Routing Optimization Using Multiple Local Search Improvements. *Automatika* 55, 124–132. <https://doi.org/10.7305/automatika.2014.01.580>
- [25] J. Kytöjoki, T. Nuortio, O. Bräysy, and M. Gendreau, “An efficient variable neighbourhood search heuristic for very large scale vehicle routing problems,” *Computers and Operations Research*, vol. 34, no. 9, pp. 2743–2757, 2007.

- [26] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet, “A guide to vehicle routing heuristics,” *The Journal of the Operational Research Society*, vol. 53, no. 5, pp. 512–522, 2002.
- [27] Fisher, M. L. and R. Jaikumar, A generalized assignment heuristic for vehicle routing, *Networks* 11 (1981), pp. 109–124.
- [28] <https://www.researchgate.net/figure/The-Capacitated-Vehicle-Routing-Problem-CVRPfig1319754352>
- [29] Laporte, G., and Semet, F. (2002). “Classical heuristics for the capacitated VRP,” in *The Vehicle Routing Problem*, eds P. Toth and D. Vigo (Philadelphia, PA: Society for Industrial and Applied Mathematics), 109–128. doi: 10.1137/1.9780898718515.ch5
- [30] Feld, S., Roch, C., Gabor, T., Seidel, C., Neukart, F., Galter, I., Mauerer, W., Linnhoff-Popien, C., 2019. A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer. *Front. ICT* 6, 13. <https://doi.org/10.3389/fict.2019.00013>
- [31] Avdoshin, S.M., Beresneva, E.N., 2019. Constructive heuristics for Capacitated Vehicle Routing Problem: a comparative study. *Proc. Inst. Syst. Program. RAS* 31, 145–156. [https://doi.org/10.15514/ISPRAS-2019-31\(3\)-12](https://doi.org/10.15514/ISPRAS-2019-31(3)-12)
- [32] G. Laporte and F. Demet. Classical heuristics for the Capacitated VRP. In *The Vehicle Routing Problem*, SIAM, 2002, pp. 109-128.
- [33] Poot, A., 1999, A Savings Based Method for Real-Life Vehicle Routing Problems.
- [34] G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, vol. 33, no. 5, 1985, pp. 1050–1073.
- [35] Masehian, E., 2009. New Heuristic Algorithms for Solving Single-Vehicle and Multi-Vehicle Generalized Traveling Salesman Problems (GTSP). *J. Ind. Eng.* 11.
- [36] Han, L., 2016, Metaheuristic algorithms for the vehicle routing problem with time window and skill set constraints.
- [37] Sokolov, A., Sanyal, A., Whitley, D., Malaiya, Y., 2005. Dynamic Power Minimization During Combinational Circuit Testing as a Traveling Salesman Problem, in: 2005

- IEEE Congress on Evolutionary Computation. Presented at the 2005 IEEE Congress on Evolutionary Computation, IEEE, Edinburgh, Scotland, UK, pp. 1088–1095. <https://doi.org/10.1109/CEC.2005.1554812>
- [38] Sathyan, A., Boone, N., Cohen, K., 2015. Comparison of Approximate Approaches to Solving the Travelling Salesman Problem and its Application to UAV Swarming. *Int. J. Unmanned Syst. Eng.* 3, 1–16. <https://doi.org/10.14323/ijuseng.2015.1>
- [39] O.BräysyandM.Gendreau, “Vehicleroutingproblemwith time windows, part ii: Metaheuristics,” *Transportation Science*, vol. 39, no. 1, pp. 119–139, 2005.
- [40] Hansen, P., Mladenović, N., 2006. First vs. best improvement: An empirical study. *Discrete Appl. Math.* 154, 802–817. <https://doi.org/10.1016/j.dam.2005.05.020>
- [41] Hillier, F. S. (2012). *Introduction to operations research*. Tata McGraw-Hill Education.
- [42] Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem.. *Computers & operation research* 31, pp. 1985-2002.
- [43] Drezner, Z., 2003. new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing* 15, pp. 320-330.
- [44] N. Mladenović and P. Hansen (1997). Variable neighborhood search. *Computers Oper. Res.*24: 1097–1100.
- [45] P. Hansen and N. Mladenović (2001c). Developments of variable neighborhood search. In: C. RibeiroandP.Hansen(eds.), *Essays and Surveys in Metaheuristics*, pp. 415–440, KluwerAcademic Publishers, Boston/Dordrecht/London.
- [46] P. Hansen and N. Mladenović (1999). An introduction to variable neighborhood search. In: S. Voss et al. (eds), *Metaheuristics, Advances and Trends in Local Search Paradigms for Optimization*, pp. 433–458, Kluwer, Dordrecht.
- [47] Hansen, P., Mladenovic, N., 2003, A Tutorial on Variable Neighborhood Search.
- [48] Laporte, G., & Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. *North-Holland Mathematics Studies*, 132, 147-184.

- [49] Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1), 255-282.
- [50] Laporte, G., Mercure, H., & Nobert, Y. (1986). An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks*, 16(1), 33-46. [
- [51] Kumar, Y., & Jain, S. (2015, September). School bus routing based on branch and bound approach. In *Computer, Communication and Control (IC4), 2015 International Conference on* (pp. 1-4). IEEE.
- [52] Eilon, S., Watson-Gandy, C. D. T., Christofides, N., & de Neufville, R. (1974). *Distribution Management-Mathematical Modelling and Practical Analysis*. IEEE Transactions on Systems, Man, and Cybernetics, (6), 589-589.
- [53] Balinski, M. L., & Quandt, R. E. (1964). On an integer program for a delivery problem. *Operations Research*, 12(2), 300-304.
- [54] https://en.wikipedia.org/wiki/Cluster_analysis
- [55] <https://towardsdatascience.com/clustering-clearly-explained-5561642ec20c>
- [56] https://en.wikipedia.org/wiki/K-means_clustering
- [57] [https://lwileczek.github.io/Density-Based-Spatial-Clustering-of-Applications-with-Noise-\(DBSCAN\)](https://lwileczek.github.io/Density-Based-Spatial-Clustering-of-Applications-with-Noise-(DBSCAN))
- [58] He, R., Xu, W., Sun, J., Zu, B., 2009. Balanced K-Means Algorithm for Partitioning Areas in Large-Scale Vehicle Routing Problem, in: 2009 Third International Symposium on Intelligent Information Technology Application. Presented at the 2009 Third International Symposium on Intelligent Information Technology Application, IEEE, NanChang, China, pp. 87–90. <https://doi.org/10.1109/IITA.2009.307>
- [59] Geetha, S., Poonthalir, G., Vanathi, P.T., 2009, Improved K-Means Algorithm for Capacitated Clustering Problem.
- [60] Kumar, C.N.S., Rao, D.K.N., Govardhan, D.A., Reddy, D.S., 2014. Imbalanced K-Means: An algorithm to cluster imbalanced-distributed data 2, 9.
- [61] Kwangcheol Shin, Sangyong Han ,(2011), A centroid based heuristic algorithm for the capacitated vehicle routing problem

- [62] Miller, C.E., Tucker, A.W., Zemlin, R.A., 1960. Integer Programming Formulation of Traveling Salesman Problems. *J. ACM JACM* 7, 326–329. <https://doi.org/10.1145/321043.321046>
- [63] Kalatzantonakis, P., Sifaleras, A., Samaras, N., 2019. Cooperative versus non-cooperative parallel variable neighborhood search strategies: a case study on the capacitated vehicle routing problem. *J. Glob. Optim.* <https://doi.org/10.1007/s10898-019-00866-y>