ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΕΝΑ ΜΟΝΤΕΛΟ ΚΑΙ ΕΝΑΣ ΑΠΛΗΣΤΟΣ ΑΛΓΟΡΙΘΜΟΣ ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΖΗΤΟΥΜΕΝΟ ΤΗΝ ΕΞΙΣΟΡΡΟΠΗΣΗ ΕΡΓΑΣΙΑΚΟΥ ΦΟΡΤΟΥ

υπό

ΚΑΛΤΗ ΤΙΜΟΛΕΟΝΤΑ

Διπλωματούχου Μαθηματικού Πανεπιστημίου Ιωαννίνων 2018

Μεταπτυχιακή Εργασία

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης

Βόλος, 2021

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων) Δρ. Αθανάσιος Ζηλιασκόπουλος
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής Δρ. Γεώργιος Σαχαρίδης
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Τρίτος Εξεταστής Δρ. Αθανάσιος Λόης
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

ΕΝΑ ΜΟΝΤΕΛΟ ΚΑΙ ΕΝΑΣ ΑΠΛΗΣΤΟΣ ΑΛΓΟΡΙΘΜΟΣ ΓΙΑ ΠΡΟΒΛΗΜΑΤΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ ΜΕ ΖΗΤΟΥΜΕΝΟ ΤΗΝ ΕΞΙΣΟΡΡΟΠΗΣΗ ΕΡΓΑΣΙΑΚΟΥ ΦΟΡΤΟΥ

ΚΑΛΤΗΣ ΤΙΜΟΛΕΩΝ

Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας, 2021

Επιβλέπων Καθηγητής: Δρ. Αθανάσιος Ζηλιασκόπουλος,
Καθηγητής Βελτιστοποίησης Συστημάτων Παραγωγής/Μεταφορών

Περίληψη

Το πρόβλημα δρομολόγησης οχημάτων είναι ένα από τα ευρέως γνωστά προβλήματα προς βελτιστοποίηση σε συστήματα μεταφοράς. Μια παραλλαγή του προβλήματος δρομολόγησης οχημάτων, που όλο και περισσότερο απασχολεί την επιστημονική κοινότητα τελευταία, είναι αυτή της δίκαιης κατανομής του φόρτου εργασίας, με την έννοια της εύρεσης ενός σημείου ισορροπίας για την χρησιμότητα των πόρων του συστήματος. Στην εργασία αυτή, θα εισάγουμε ένα μαθηματικό μοντέλο για την εύρεση λύσεων στα προβλήματα αυτά, υπό την προϋπόθεση ότι η μετρική, η οποία αποτελεί το αντικείμενο της ισορροπίας, πληρεί ορισμένες συνθήκες. Θα θεωρήσουμε δύο κλασσικές εφαρμογές που συναντάμε σε σχετική βιβλιογραφία και θα προσπαθήσουμε να προσεγγίσουμε λύσεις για το μοντέλο μας, μέσω ενός άπληστου ευρετικού αλγορίθμου που αναπτύξαμε. Ο αλγόριθμος μας, σε πρώτη φάση, προσπαθεί να φέρει την επιθυμητή, για τον σχεδιαστή, ισορροπία στο σύστημα και στην συνέχεια να βρει την καλύτερη εκδοχή αυτής. Για την αξιολόγηση των επιδόσεων και της αποτελεσματικότητας του αλγορίθμου, δημιουργήσαμε τρία πειράματα και συγκρίναμε τις μεθόδους του αλγορίθμου μας με μια άλλη μέθοδο. Μέσα από την ανάλυση των αποτελεσμάτων, μοιραζόμαστε τις παρατηρήσεις μας και δίνουμε κατευθύνσεις προς μελλοντική έρευνα.

A MODEL AND A GREEDY ALGORITHM FOR THE WORKLOAD EQUITY IN VEHICLE ROUTING PROBLEMS

KALTIS TIMOLEON

Department of Mechanical Engineering, University of Thessaly, 2021

Supervisor: Dr Athanios Ziliaskopoulos,
Professor in Optimising Production/Transportation Systems

Abstract

The Vehicle Routing Problem(VRP) is one of the most widely known problems in transportation systems. A variant of the VRP, which is lately growing in popularity amongst the scientific community, is that of the workload equity, in the sense of balancing the utilization of the system resources. In this master thesis, we will introduce a mathematical model for solving such problems, providing the equity metric fulfills certain conditions. We will consider two classic applications found in relevant articles and try to find good approximate solutions for our model using a greedy heuristic we developed. Our algorithm firstly focuses on achieving the balance, set by the designer, and then finding its optimal version. In order to evaluate the efficiency and performance of the algorithm, we constructed three experiments and compared the methods used by our algorithm and these of another. After analyzing the results, we share our thoughts and provide directions for future research.

Περιεχόμενα

Κεφάλαιο 1

ΕΙΣΑΓΩΓΗ

Το πρόβλημα δρομολόγησης οχημάτων(Vehicle Routing Problem)είναι ένα από τα πιο κρίσιμα προβλήματα στον τομέα διαχείρισης εφοδιαστικών αλυσίδων και Logistics,όπως φαίνεται και από την τεράστια βιβλιογραφία γύρω από αυτό. Συστήματα μεταφορών και διανομής ασχολούνται κατά κόρον με το ΠΔΟ(VRP). Το πρόβλημα εστιάζει στον σχεδιασμό ενός συνόλου δρομολογίων ελαχίστου κόστους για ένα στόλο οχημάτων, με την δυνατότητα κάλυψης πλήρους ζήτησης.

Το ΠΔΟ είναι ένα NP-hard πρόβλημα, το οποίο σημαίνει ότι για προβλήματα μεγάλων διαστάσεων ή αληθινά παραδείγματα, η εύρεση της βέλτιστης λύσης του προβλήματος είναι πρακτικά αδύνατη. Ακόμα και για σχετικά μικρά σε διαστάσεις προβλήματα, ο σχεδιασμός των δρομολογίων μπορεί να δυσκολεύει εξαιτίας επιπλέον περιορισμών που έχουν τεθεί για το πρόβλημα, όπως η χωρητικότητα των οχημάτων(Capacitated VRP) ή η εξυπηρέτηση πελατών σε συγκεκριμένα χρονικά διαστήματα (VRP with Time Windows)[1]. Για το λόγο αυτό, μέθοδοι όπως η διαίρεση του προβλήματος σε συστάδες(clusters), ευρετικοί(heuristics)και μετα-ευρετικοί αλγόριθμοι(metaheuristics), αλλά και συνδυασμός όλων των παραπάνω, υβριδικοί(hybrids), έχουν αναπτυχθεί με σκοπό την εύρεση καλών λύσεων σε ικανοποιητικό υπολογιστικό χρόνο [1][2].

Μια παραλλαγή του ΠΔΟ που απασχολεί ιδιαίτερα την επιστημονική κοινότητα τα τελευταία χρόνια, σύμφωνα με το άρθρο [3] είναι η δίκαιη κατανομή εργασιακού φόρτου(workload equity)σε ΠΔΟ. Δηλαδή παράγοντες όπως, η ικανοποίηση και η ποιότητα εξυπηρέτησης των πελατών, η αποδοχή και το ανεβασμένο ηθικό των εργατών, φέρουν κόστος για το σύστημα που εξετάζουμε και επομένως αποτελούν αντικείμενο προς βελτιστοποίηση.

Σε τέτοια προβλήματα συναντάμε συνήθως αντικρουόμενα συμφέροντα, δηλαδή η βελτίωση μίας συνθήκης συνοδεύεται από την υποτίμηση μίας άλλης. Η βελτιστοποίηση τέτοιων προβλημάτων εκφέρεται στην βιβλιογραφία ως Παρέτο Βελτιστοποίηση(Pareto Optimization/Efficiency). Οι συγγραφείς μέσα από μία βιβλιογραφική ανασκόπηση, παρατήρησαν ότι, ενώ υπάρχει μια πληθώρα μοντέλων για διάφορες πρακτικές εφαρμογές, λίγη συζήτηση έχει γίνει σχετικά με τους πιθανούς τρόπους αξιολόγησης των μοντέλων που χρησιμοποιούνται και των αντίστοιχων αποτελεσμάτων τους, αλλά και για έναν εν γένη τρόπο αντιμετώπισης του προβλήματος ανάλογο του «φόρτου»(equity metric).

Ορμώμενοι από αυτήν την παρατήρηση, οι συγγραφείς συλλέγουν τα έξι πιο συχνά εμφανιζόμενα μέτρα ισορροπίας(equity measures/functions)από σχετική βιβλιογραφία και στη συνέχεια συγκρίνουν τις ιδιότητες αυτών με τις ιδιότητες ενός ιδεατού μέτρου ισορροπίας, βάση ενός συνόλου αξιωμάτων, με σκοπό την αξιολόγηση της αποτελεσματικότητας τους σε κάθε περίπτωση.

Συγκεκριμένα, το πρόβλημα στο οποίο εστιάζουν είναι αυτό της εξισορρόπησης των δρομολογίων(VRP Route Balancing) σε σχέση με το μήκος ή το χρόνο των δρομολογίων. Τα μέτρα ισορροπίας προς εξέταση είναι *min-max*:Βελτιστοποίηση του μέγιστου δρομολογίου, *lexicographic min-max*: Βελτιστοποίηση του μέγιστου δρομολογίου, βελτιστοποίηση του δεύτερου κατά σειρά μέγιστου δρομολογίου σε σχέση με το πρώτο κτλ. *range*:Βελτιστοποίηση της διαφοράς μεταξύ του μέγιστου και του ελάχιστου δρομολογίου, *mean absolute deviation*:Βελτιστοποίηση της αθροιστικής απόλυτης απόκλισης των δρομολογίων από το μέσο δρομολόγιο, *standard deviation*:Βελτιστοποίηση της διακύμανσης των δρομολογίων και *Gini-coefficient*.

Τα υπολογιστικά αποτελέσματα επιβεβαιώνουν τις παρατηρήσεις των συγγραφέων όσο αναφορά τη σχέση μεταξύ των ιδιοτήτων των αξιωμάτων που πληρούν τα ισορροπητικά μέτρα και την ποιότητα των Παρέτο Συνόλων(Pareto Sets)που προσφέρουν για το το πρόβλημα εξισορρόπησης δρομολογίων. Συγκεκριμένα, τα απόλυτα μέτρα(absolute measures),*min-max* καί *lex min-max*, τα οποία υπακούουν στο αξίωμα της μονοτονίας πρόσφεραν ταυτόχροναTSP –Βέλτιστες(τα δρομολόγια είναι βέλτιστα σε κόστος) και σε ισορροπία διατηρητέες(Inequality Consistent) λύσεις, σε αντίθεση με τα σχετικά μέτρα(relative measures), όλα τα υπόλοιπα. Το βασικό μειονέκτημα αυτών *(min-max/lex min-max)* είναι η δυσκολία που παρουσιάζουν στην ερμηνεία των αποτελεσμάτων τους από τον υπεύθυνο αποφάσεων(decision maker).

Τέλος, οι συγγραφείς, μεταξύ άλλων, συμπεραίνουν ότι για ΠΔΟ με μετρικές ισορροπίας(equity metrics)να είναι μερικώς ή πλήρως αθροιστικά μεταβλητές(variable sum), η κατάλληλη προσέγγιση είναι αυτή ενός μέτρου ισορροπίας, το οποίο πληρεί το αξίωμα της μονοτονίας. Σε αντίθετη περίπτωση, δηλαδή όταν η μετρική ισορροπίας είναι αθροιστικά σταθερή(constant sum), η πληρότητα της αρχής των Pigou-Dalton αρκεί, ακόμα και στην ασθενή της μορφή. Μάλιστα, ενθαρρύνεται η χρήση σχετικών μέτρων λόγω της μεγαλύτερης ποικιλίας των Παρέτο Συνόλων που μπορούν να προσφέρουν.

Σε αυτή την εργασία, θα προσπαθήσουμε να δημιουργήσουμε ένα καθολικό μοντέλο για την εξισορρόπηση «φόρτου» σε ΠΔΟ, αλλά και να προσεγγίσουμε καλές λύσεις για αυτό μέσω ενός άπληστου ευρετικού αλγορίθμου που δημιουργήσαμε.

Έμπνευση μας αποτέλεσε το άρθρο [4]. Η κάλυψη της ζήτησης μιας περιοχής αποτελεί ένα βασικό πρόβλημα για τα συστήματα διανομής, ιδικά όταν οι διαθέσιμοι πόροι του συστήματος είναι περιορισμένοι. Οι συγγραφείς αντιμετωπίζουν το πρόβλημα αυτό διαιρώντας την περιοχή αυτή σε υποπεριοχές, δίνοντας σε αυτές συγκεκριμένες ιδιότητες με στόχο την αποτελεσματικότερη κάλυψη της όλης περιοχής.

Αρχικά, διασφαλίζεται η οριοθέτηση των υποπεριοχών με τη χρήση πολυγώνων(Thiessen Polygons/Voronoi diagrams) δεδομένων των γεωγραφικών συντεταγμένων της περιοχής, το οποίο μεταξύ άλλων μας προσφέρει μια καλύτερη εικόνα της περιοχής. Οι υποπεριοχές δημιουργούνται με τρόπο τέτοιο έτσι ώστε να είναι όσο το δυνατόν πιο συμπαγείς, εξασφαλίζοντας έτσι μικρότερες αποστάσεις μεταξύ των πελατών μιας υποπεριοχής και ελαχιστοποίηση σε χρόνους εξυπηρέτησης. Τέλος, οι υποπεριοχές είναι ισορροπημένες όσο αναφορά τη ζήτηση κάθε υποπεριοχής, έτσι ώστε κάθε υποπεριοχή να μπορεί να εξυπηρετηθεί από το διαθέσιμο στόλο.

Για να επιτύχουν το ζητούμενο, οι συγγραφείς δημιούργησαν έναν αλγόριθμο, ο οποίος σε πρώτη φάση κατασκευάζει μια αρχική λύση για τις υποπεριοχές με τις παραπάνω ιδιότητες. Στη συνέχεια, μέσα από μια σειρά μεταφορών και ανταλλαγών πελατών μεταξύ υποπεριοχών, νέες λύσεις αξιολογούνται μέσα από μια αντικειμενική συνάρτηση, ελέγχοντας πάντα την συνδεσιμότητα των υποπεριοχών. Για να αποφύγουν μεγάλους υπολογιστικούς χρόνους, οι συγγραφείς, θέτουν ένα κατώφλι *(threshold)*, το οποίο αν ξεπεραστεί από μια υποψήφια λύση τότε αυτή θα γίνεται και αυτομάτως δεκτή. Ο αλγόριθμος τερματίζει όταν πλέον δεν μπορούν να βρεθούν καλύτερες λύσεις. Τα αποτελέσματα που παρουσιάζονται φαίνονται να πετυχαίνουν σε μεγάλο βαθμό ισορροπίες σε ζήτηση σε ικανοποιητικό χρόνο.

Αυτό για το οποίο δεν μπορούμε να είμαστε απόλυτα σίγουροι είναι η ποιότητα των δρομολογίων που μας προσφέρει ο εν λόγω αλγόριθμος, δηλαδή αν μια δεδομένη ισορροπία για το σύστημα μπορεί να επιτευχθεί και με μικρότερα σε κόστος δρομολόγια. Αν και μια τέτοια παρατήρηση είναι ίσως προφανής, από τη στιγμή που μιλάμε για λύσεις που παράγονται από έναν ευρετικό αλγόριθμο, η προσοχή μας στέφεται στον τρόπο με τον οποίο βελτιώνονται αυτές οι λύσεις και συγκεκριμένα στην έναν προς έναν ανταλλαγή κόμβων. Είναι εύκολο να δει κανείς ότι αν η ζήτηση κάθε πελάτη δεν είναι ίση με τη μονάδα τότε η μέθοδος αυτή είναι κάπως περιοριστική, ακόμα και για μικρή διακύμανση της ζήτησης των πελατών. Φυσικά, μια παραλλαγή της μεθόδου αυτής θα μπορούσε να ήταν η ανταλλαγή ενός πελάτη μιας περιοχής με έναν αριθμό πελατών μιας άλλης, μια τέτοια διαδικασία όμως θα αύξανε δραματικά τον υπολογιστικό χρόνο του αλγορίθμου.

Η μέθοδος που προτείνουμε εδώ, αντί των ανταλλαγών, είναι ένας άπληστος αλγόριθμος, βασισμένος στην εύρεση της μέγιστης τομής(Max-Cut) ενός γραφήματος, μέσω συστολών των ακμών του, που παρουσιάζεται στο άρθρο [5]. Πριν περάσουμε σε λεπτομέρειες, δεδομένου ενός γραφήματος με μη αρνητικά βάρη ακμών, ορίζουμε ως μέγιστη σε κόστος τομή, το πρόβλημα διαχωρισμού του συνόλου των κόμβων του γραφήματος σε δύο ξένα υποσύνολα έτσι ώστε το αθροιστικό βάρος των ακμών, των οποίων τα τελικά σημεία να βρίσκονται σε διαφορετικά υποσύνολα, να μεγιστοποιείται.

Η εύρεση της βέλτιστης σε κόστος μέγιστης τομής ενός γραφήματος είναι NP-hard. Σύγχρονες μέθοδοι αντιμετώπισης του προβλήματος περιλαμβάνουν προσεγγιστικά σχήματα(approximation schemes) για ημι-ορισμένο προγραμματισμό(semi-definite programming) και μετα-ευρετικούς αλγορίθμους όπως, GRASP(greedy randomised adaptive search procedure), VNS(variable neighborhood search) και VND(variable neighborhood descent). Για περισσότερες πληροφορίες, προτείνουμε το άρθρο [6] στον αναγνώστη. Οι συγγραφείς του [5] αποδεικνύουν ότι αν W^* είναι η βέλτιστη τιμή του κόστους της μέγιστης τομής και W_c η τιμή του αλγορίθμου συστολής, τότε $W_c > \frac{1}{3}W^*$. Επίσης, θεωρούνται διάφοροι συνδυασμοί του αλγορίθμου με τον αλγόριθμο των Sahni-Gonzalez[8]. Τέλος, οι συγγραφείς συγκρίνουν τα αποτελέσματα της κάθε εκδοχής του αλγορίθμου, με την τελευταία εκδοχή(Sahni-Gonzalez 3) να είναι καλύτερη από τον αλγόριθμο συστολής.

Παρακάτω, στο κεφάλαιο 2, θα περιγράψουμε το πρόβλημα και αφού παρουσιάσουμε τους απαραίτητους ορισμούς θα περάσουμε στην διατύπωση του μοντέλου μας. Στο κεφάλαιο 3, θα παρουσιάσουμε τον αλγόριθμο που αναπτύξαμε και επιμέρους φάσεις του. Στην συνέχεια θα προχωρήσουμε στα υπολογιστικά αποτελέσματα του αλγορίθμου μας, στο κεφάλαιο 4 και τελειώνουμε με τα συμπεράσματα στο κεφάλαιο 5.

Κεφάλαιο 2

ΠΡΟΒΛΗΜΑ

2.1 Περιγραφή

Όπως είπαμε και στην εισαγωγή, η βελτιστοποίηση προβλημάτων πολλαπλών κριτηρίων, τα οποία είναι αντιφατικά μεταξύ τους, εκφέρεται βιβλιογραφικά ως βελτιστοποίηση Παρέτο(Pareto Optimization). Προβλήματα που παρουσιάζουν τέτοια φαινόμενα εκφράζονται από ένα μοντέλο πολλαπλών αντικειμενικών συναρτήσεων(multi-objective model) στο οποίο η βελτιστοποίηση μιας συνάρτησης συμβαίνει ταυτόχρονα με τις άλλες. Στόχος των μοντέλων αυτών είναι η εύρεση λύσεων για τις οποίες καμία μεμονωμένη συνάρτηση να μπορεί να βελτιωθεί χωρίς να επιφέρει την εκφύλιση τουλάχιστον μίας άλλης, αυτές οι λύσεις ονομάζονται Παρέτο βέλτιστες(Pareto Optimal). Να σημειώσουμε εδώ ότι όλες οι Παρέτο βέλτιστες λύσεις θεωρούνται καλές λύσεις, το οποίο μας παροτρύνει στην εύρεση, εύκολων προς ερμηνεία, λύσεων προκειμένου να μπορούν να αναλυθούν από τον σχεδιαστή ή τον υπεύθυνο λήψης αποφάσεων.

Θεωρώντας λοιπόν το πρόβλημα της ισορροπίας του εργασιακού φόρτου (workload equity)σε προβλήματα δρομολόγησης οχημάτων, βλέπουμε ότι το δυικό μοντέλο που παρουσιάζεται στο άρθρο [3] έχει νόημα. Ειδικότερα, αν υποθέσουμε ότι έχουμε έναν στόλο k οχημάτων και κάθε όχημα πρέπει να εξυπηρετεί ένα και μόνο ένα δρομολόγιο, ξεκινώντας και τελειώνοντας την διαδρομή του σε ένα συγκεκριμένο σημείο, τότε ένα μοντέλο για το πρόβλημα μας είναι το εξής:

$$\begin{aligned} \min \quad & (I(\mathbf{x}), C(\mathbf{x})) \\ \textbf{s.t.} \quad & \mathbf{x} \in \mathbf{X} \end{aligned} \tag{2.1}$$

Όπου, $\mathbf{x} = (x_1, ..., x_k)$ είναι το διάνυσμα των δρομολογίων κάθε οχήματος, C η συνάρτηση κόστους, I η συνάρτηση ισορροπίας/ανισορροπίας και $\mathbf{X}$ το σύνολο των εφικτών λύσεων.

Για την επίλυση του μοντέλου 2.1, προτείνουμε την μέθοδο των σταθμισμένων αθροισμάτων(weighted sum):

$$\begin{aligned} \min \quad & a \times I(\mathbf{x}) + b \times C(\mathbf{x}) \\ \textbf{s.t.} \quad & \mathbf{x} \in \mathbf{X} \end{aligned} \tag{2.2}$$

Όπου, $a, b \in \mathbb{R}$ τα βάρη των συναρτήσεων τέτοια ώστε: $0 < a < 1, 0 < b < 1$ και $a + b = 1$. Για ρεαλιστικά αποτελέσματα, θα πρέπει να ομαλύνουμε(normalise) την αντικειμενική συνάρτηση του μοντέλου.

2.2 Ορισμοί

Πριν διατυπώσουμε το μοντέλο για το πρόβλημα μας θα εισάγουμε τους απαραίτητους ορισμούς. Θεωρούμε ένα μη κατευθυνόμενο(undirected)γράφημα $G(V(G), E(G))$, όπου για χάριν ευκολίας θα συμβολίζουμε με $V = V(G)$ το σύνολο των κόμβων του και με $E = E(G)$ το σύνολο των ακμών. Επίσης, θεωρούμε ότι το γράφημα G είναι ολοκληρωμένο(complete), δηλαδή $\forall p, q \in V \exists! e = (p, q) \in E$. Τέλος, θεωρούμε το γράφημα G στον Ευκλείδειο μετρικό χώρο (G, c) με Ευκλείδεια μετρική την συνάρτηση $c : E \to \mathbb{R}_{\geq 0}$ και στην οποία θα αναφερόμαστε και ως συνάρτηση κόστους των ακμών του G. Μάλιστα, για τυχόν γράφημα $H \subseteq G$, $H(V(H), E(H), c)$, το κόστος του H είναι $c(H) = \sum\limits_{e \in E(H)} c(e)$.

Μια συλλογή υπογραφημάτων $(C_1, C_2, \ldots, C_k), k \in \mathbb{N}$, του G, αποτελεί k-ομαδοποίηση(clustering) για το γράφημα G, αν:

- $\bigcup\limits_{i=1}^{k} V(C_i) = V$
- $C_i \cap C_j = \emptyset, \forall i, j$ με $i \neq j$

Κάθε στοιχείο της ομαδοποίησης, C_i, $i = 1, ..., k$, ονομάζεται ομάδα ή συστάδα(cluster).

Ορίζουμε τη συνάρτηση βάρους, $w : H \to \mathbb{R}_{\geq 0}$, ενός γραφήματος $H \subseteq G$ έτσι ώστε για $A = (H_1, \ldots, H_k)$, τυχαία k-ομαδοποίηση του H, να ισχύει $w(H) = w(H_1) + \cdots + w(H_k)$. Ορίζουμε $S = \{(C_1, C_2, ..., C_k) |\ k \in \mathbb{N}\}$ το σύνολο όλων των πιθανών ομαδοποιήσεων του G.

Μία ομαδοποίηση $A \in S_\varepsilon$ θα λέμε ότι είναι ισορροπημένη ως προς ε αν $\exists \varepsilon \geqslant 0$, τέτοιο ώστε:

$$|w(C_i) - w(C_j)| \leqslant \varepsilon \ \forall C_i, C_j \in A \text{ για } i = 1, \ldots, k, j = 1, \ldots, k \text{ με } i \neq j$$

2.3 Διατύπωση

Έχοντας αναπτύξει την αναγκαία ορολογία θα προχωρήσουμε στην διατύπωση και μοντελοποίηση του προβλήματος μας. Έστω $G(V, E, c)$ γράφημα στον Ευκλείδειο μετρικό χώρο όπως ορίστηκε παραπάνω. Τότε βλέπουμε ότι μπορούμε να ερμηνεύσουμε με φυσικό τρόπο τη δομή του δικτύου του προβλήματος μας μέσω του γραφήματος G. Συγκεκριμένα, οι πελάτες του δικτύου δίνονται από τους κόμβους $p \in V$ του G και οι αποστάσεις μεταξύ αυτών από την συνάρτηση κόστους, c. Αν k είναι ο αριθμός των οχημάτων και $p_0 \in V$ το σημείο εκκίνησης και τερματισμού όλων των οχημάτων μας(depot), τότε υποθέτοντας ότι κάθε όχημα κάνει ένα και μόνο ένα δρομολόγιο για όλη τη χρονική περίοδο του συστήματος και κάθε πελάτης μπορεί να εξυπηρετηθεί από ένα όχημα και λαμβάνοντας υπόψιν το μοντέλο 2.2 μπορούμε να διατυπώσουμε το πρόβλημα μας ως εξής:

Ψάχνουμε μια k-ομαδοποίηση A του $G \setminus p_0$ ισορροπημένη ως προς κάποιο ε τέτοια ώστε:

$$\begin{aligned} \min \quad & a \sum_{i=1}^{k} \sum_{j=1}^{k} |w(C_i) - w(C_j)| + b \sum_{i=1}^{k} c(HC_i) \\ \textbf{s.t.} \quad & A \in S_\varepsilon \end{aligned} \tag{2.3}$$

Όπου $a, b \in \mathbb{R}$ με $0 < a < 1, 0 < b < 1$ και $a + b = 1$ τα βάρη των συναρτήσεων ανισορροπίας και κόστους, $w : H \to \mathbb{R}_{\geq 0}$ η συνάρτηση βάρους που ορίστηκε παραπάνω και HC_i ο κύκλος του Χάμιλτον(Hamiltonian Cycle) του γραφήματος $C_i \cup \{p_0\}$. Σημειώνουμε εδώ ότι έχει νόημα να μιλάμε για κύκλους Χάμιλτον από την στιγμή που το γράφημα G είναι ολοκληρωμένο και επομένως και τα $C_i \cup \{p_0\}$ είναι ολοκληρωμένα, ως υπογραφήματα ενός ολοκληρωμένου γραφήματος.

Ένας τρόπος που θα μας βοηθήσει στο να απλοποιήσουμε το μοντέλο 2.3 είναι ο ορισμός ενός συγκεκριμένου βάρους για τις συστάδες μας. Θα αναφερόμαστε σε αυτό ως στόχος βάρους(target weightή TW). Θέτοντας λοιπόν εκ των προτέρων ένα στόχο βάρους για το πρόβλημα μας, μπορούμε να επαναδιατυπώσουμε το μοντέλο 2.3 ως εξής:

$$\begin{aligned} \min \quad & a\sum_{i=1}^{k}|w(C_i) - TW| + b\sum_{i=1}^{k}c(HC_i) \\ \textbf{s.t.} \quad & A \in S \end{aligned} \tag{2.4}$$

Το πρόβλημα φυσικά είναι και παραμένει NP-hard. Για το λόγο αυτό παρακάτω θα προσπαθήσουμε να αναπτύξουμε έναν ευρετικό αλγόριθμο για το μοντέλο 2.4 με σκοπό να την προσέγγιση καλών λύσεων σε ικανοποιητικό υπολογιστικό χρόνο. Πριν περάσουμε όμως στον αλγόριθμο, θα μεταποιήσουμε για ακόμη μια φορά το μοντέλο μας. Αυτή τη φορά κινητήριος δύναμη αποτελεί η αμέσως παρακάτω πρόταση.

Πρόταση: Έστω $G(V, E, c)$ ένα ολοκληρωμένο και μη κατευθυνόμενο γράφημα στον Ευκλείδειο μετρικό χώρο, HC^*, ο ελάχιστος σε κόστος κύκλος του Χάμιλτον του G και T το ελάχιστο γεννητικό δέντρο(Minimum Spanning Tree-MST)του G. Τότε $c(T) \leqslant c(HC^*) \leqslant 2c(T)$.

Η παραπάνω πρόταση μας λέει ουσιαστικά ότι το MST αποτελεί ένα άνω και κάτω όριο για το HC^*_G. Επίσης, γνωρίζουμε ότι η εύρεση του ΕΓΔ ενός γραφήματος, όπως το έχουμε περιγράψει, μπορεί να γίνει σε χρόνο $O(E \log V)$, όπου V το σύνολο των κόμβων και E το σύνολο των ακμών του γραφήματος. Μια περιγραφική απόδειξη παρουσιάζεται στο Παράρτημα.

Έτσι, επαναδιατυπώνουμε το μοντέλο 2.4 ως εξής:

$$\begin{aligned} \min \quad & a\sum_{i=1}^{k}|w(C_i) - TW| + b\sum_{i=1}^{k} c(MST_i) \\ \textbf{s.t.} \quad & A \in S \end{aligned} \tag{2.5}$$

Όπου, MST_i είναι το ΕΓΔ του γραφήματος $C_i \cup \{p_0\}$.

Πριν προχωρήσουμε στην περιγραφή του αλγορίθμου και των επιμέρους φάσεων του, θα πρέπει πρώτα να σταθούμε στον τρόπο με τον οποίο ορίζεται το βάρος μίας συστάδας. Όπως είδαμε παραπάνω, η συνάρτηση βάρους, w, προσδίδει σε κάθε συστάδα μια τιμή βάρους και γνωρίζοντας το στόχο βάρους, μπορούμε να υπολογίσουμε την ισορροπία/ανισορροπία μίας λύσης του προβλήματος. Ισχυριζόμαστε ότι για προβλήματα δρομολόγησης οχημάτων με επιπλέον ζήτημα την ισορροπία ενός συγκεκριμένου «φόρτου», ορισμένο από τον σχεδιαστή, το παραπάνω μοντέλο δίνει λύσεις αρκεί $w(A) = M, \forall A \in S$, όπου $M > 0$ πραγματική σταθερά. Ο ορισμός βάρους για τις συστάδες μας θα γίνεται μέσω του ορισμού βάρους για τους κόμβους του G, δηλαδή $w(H) = \sum_{p \in V(H)} w(p)$ για $H \subseteq G$.

Επισημαίνουμε ξανά ότι τα βάρη των κόμβων θα πρέπει να οριστούν με τέτοιο τρόπο έτσι ώστε να είναι αθροιστικά σταθερά, $\sum_{p \in V} w(p) = \sum_{C \in A} w(C), \forall A \in S$.

Δύο εφαρμογές τέτοιων προβλημάτων που θα εξετάσουμε είναι η ισόρροπη κατανομή της ζήτησης μιας περιοχής και τα ισοδύναμα σε κόστη δρομολόγια. Για την πρώτη εφαρμογή, ο ορισμός βάρους για τον κόμβο είναι εντελώς φυσικός. Για την δεύτερη, πρέπει να προσέξουμε το βάρος, που θα δώσουμε στους κόμβους του G, να τηρεί την παραπάνω ιδιότητα. Ο ορισμός βάρους για $p \in V$, που προτείνουμε, είναι $w(p) = c(p_0, p)$.

Κεφάλαιο 3

ΑΛΓΟΡΙΘΜΟΣ

Ο αλγόριθμος που αναπτύσσουμε παρακάτω είναι εμπνευσμένος από τον αλγόριθμο που παρουσιάζεται στο [4]. Συγκεκριμένα, ξεκινάμε με την δημιουργία μιας αρχικής λύσης για το σύστημα μας. Στην συνέχεια, μέσα από μια σειρά μεταφορών κόμβων, από μια συστάδα σε άλλη, προσπαθούμε να εντοπίσουμε την καλύτερη δυνατή ισορροπία για το σύστημα μας. Τέλος, διατηρώντας την ισορροπία που φέραμε στο σύστημα μας προσπαθούμε να αναιρέσουμε όποια «βλάβη» και συνάμα να βελτιώσουμε το κόστος των δρομολογίων μας. Ο χρόνος πολυπλοκότητας του αλγορίθμου κυριαρχείται από την δεύτερη φάση βελτιστοποίησης και είναι $O((\Lambda+1)k^2|V|^4)$, όπου V το σύνολο των κόμβων του γραφήματος, k ο αριθμός των συστάδων και Λ ο αριθμός των λύσεων που βρίσκει ο αλγόριθμος πριν τερματίσει.

3.1 Κατασκευαστική Φάση

Στόχος της κατασκευαστικής φάσης είναι η δημιουργία μίας ικανοποιητικής αρχικής λύσης για το πρόβλημα μας,τόσο ως προς την ισορροπία των βαρών των συστάδων, όσο και ως προς την ελαχιστοποίηση των αντίστοιχων ελάχιστων γεννητικών τους δέντρων. Η ποιότητα της αρχικής λύσης είναι εξαιρετικά σημαντική, διότι από αυτήν θα εξαρτηθεί σε μεγάλο βαθμό το μέγεθος των υπολογισμών του αλγορίθμου, όπως και η ποιότητα του τοπικού ελαχίστου που θα βρούμε για τη αντικειμενική συνάρτηση του μοντέλου μας.

Ορίζουμε ακόμα ως $\Omega = \{p \in V \setminus V(S)\}$ το σύνολο των κόμβων του V που δεν έχουν ανατεθεί ακόμα σε κάποια συστάδα, $C_i, i = 1, \dots, k$. Ακόμα, ορίζουμε ως $c(p, C) = \min_{q \in C} c(p, q)$ την απόσταση ενός κόμβου $p \in \Omega$ από κάποια συστάδα.

Δεδομένου k-tuple$(p_1, \ldots, p_k)$(initial seeding), δημιουργούμε k στον αριθμό συστάδες με μια ένα προς ένα αντιστοιχία. Στην συνέχεια, για δεδομένο κατώφλι(threshold), κατασκευάζουμε με τυχαία σειρά τις συστάδες, με αυτό τον τρόπο προσδίδουμε στις συστάδες μια μορφή. Τέλος, μέσα από μια μέθοδο αναγκαιότητας, «χτίζουμε» τις συστάδες. Οι τρόποι με τους οποίους παίρνουμε τα initial seeds είναι με τις μέγιστες αποστάσεις μεταξύ τους, με τον αλγόριθμο K-means ή προκαθορισμένα. Συνήθης επιλογή για τον στόχο βάρους είναι $TW = \frac{1}{k} \sum\limits_{p \in V} w(p)$.

Σε περίπτωση που $w(p) = 1, \forall p \in V$, η εκτέλεση της πρώτης φάσης βελτιστοποίησης δεν είναι απαραίτητη. Ένας ψευδοκώδικας του αλγορίθμου παρουσιάζεται στον Αλγόριθμο 1.

3.2 Πρώτη Φάση Βελτιστοποίησης

Στη φάση αυτή του αλγορίθμου προσπαθούμε να προσεγγίσουμε την καλύτερη δυνατή ισορροπία μεταξύ των τιμών βάρους των συστάδων μας. Δεδομένης αρχικής λύσης, πραγματοποιείται μια σειρά μεταφορών κόμβων από μια συστάδα σε μια άλλη με τέτοιο τρόπο έτσι ώστε η ανισορροπία του συστήματος να μειώνεται. Συγκεκριμένα, κόμβοι από συστάδες για τις οποίες, $w(C) - TW > 0$, μεταφέρονται σε συστάδες με $w(D) - TW < 0$. Η διαδικασία επαναλαμβάνεται μέχρι ο αλγόριθμος να μην βρίσκει καλύτερες λύσεις. Είναι φανερό πως ο αλγόριθμος είναι ένας άπληστος αλγόριθμος(greedy algorithm) και γι΄ αυτό το λόγο παρακάτω προτείνουμε κάποιες πιο φιλοσοφημένες μεθόδους προς ενίσχυση της διαδικασίας.

Προτεραιότητα σε $C \in P$: Επιλέγουμε πρώτα cluster του συνόλου $\{C \mid \max\limits_{C \in P} w(C) - TW\}$.

Προτεραιότητα σε κόμβους των $C \in P$: Δημιουργούμε μια λίστα με τους κόμβους του $C \in P$ και την ταξινομούμε κατά φθίνουσα σειρά ως προς την τιμή $|w(C) - w(p) - TW|$.

Προτεραιότητα σε $C \in N$: Αν p ο κόμβος που εξετάζουμε προς μεταφορά, μπορούμε να ταξινομήσουμε τα $C \in N$ κατά φθίνουσα σειρά ως προς την τιμή $|w(C) + w(p) - TW|$ ή κατά φθίνουσα σειρά ως προς την τιμή $c(p, C)$ ή και τα δύο.

Algorithm 1: ConstructionPhase

input : $G = (V, E, c)$:complete, $w : V \rightarrow \mathbb{R}_{\geq 0}$, p_0, k, TW, $threshold$, $p_1, \ldots, p_k$

output: $C \in S$

for $i = 1$ **to** k **do**
    $C_i \leftarrow \{p_i\}$
    $w(C_i) \leftarrow w(p_i)$

$\Omega \leftarrow \Omega \setminus \{p_0\}$

for $i = 1$ **to** k **do**
    while $w(C_i) < threshold$ **do**
        $p^* \leftarrow$`FindBestNode(`C_i`)`
        $C_i \leftarrow C_i \cup \{p^*\}$
        $w(C_i) \leftarrow w(C_i) + w(p^*)$
        $\Omega \leftarrow \Omega \setminus \{p^*\}$

while $|\Omega| \neq 0$ **do**
    $L \leftarrow \{C_i \mid \min_{i \in K} w(C_i) - TW\}$
    if $|L| = 1$ **then**
        $C^* \leftarrow L$
    else
        $C^* \leftarrow \{C_i \in L \mid \max_{i \in K} c(q, C_i) - c(p, C_i), p \in \Omega, q \in \Omega \setminus \{p\} \}$
        `σε περίπτωση που είναι παραπάνω από ένα, δίαλεξε τυχαία`
    $p^* \leftarrow$`FindBestNode(`C^*`)`
    $C_i \leftarrow C_i \cup \{p^*\}$
    $w(C_i) \leftarrow w(C_i) + w(p^*)$
    $\Omega \leftarrow \Omega \setminus \{p^*\}$

Algorithm 2: FindBestNode

input : C
output: p

$L \leftarrow \{p \in \Omega \mid c(p, C) = \min_{q \in \Omega} c(q, C)\}$
if $|L| = 1$ **then**
  return p
else
  return $p : \min_{p \in L} \sum_{q \in V(C)} c(p, q)$

Προτεραιότητα σε «κοντινά» $C \in N$: Εξετάζουμε μεταφορές κόμβων μόνο για $C \in N$ για τα οποία ισχύει $c(centroid_P, centroid_N) \leqslant \xi$, όπου $\xi \in \mathbb{R}_{\geq 0}$ και $centroid_P, centroid_N$ να είναι τα κέντρα των $D \in P$,$C \in P$ αντίστοιχα. Τρόποι με τους οποίους μπορούμε να προσεγγίσουμε τα κέντρα (centroids)των clusters είναι μέσω των συντεταγμένων των σημείων του cluster (αν γνωρίζουμε τις γεωγραφικές πληροφορίες του δικτύου) ή με τον αλγόριθμο του 1-median (χωρίς να λαμβάνουμε υπόψιν μας τα βάρη των κόμβων).
Ένας ψευδοκώδικας του αλγορίθμου παρουσιάζεται στον αλγόριθμο 3.

3.3 Δεύτερη Φάση Βελτιστοποίησης

Σε αυτή τη φάση θα προσπαθήσουμε να βελτιστοποιήσουμε τα κόστη των ελάχιστων γεννητικών δέντρων (minimum spanning trees) των συστάδων μας, χωρίς να βλάψουμε την ισορροπία που έχουμε καταφέρει από την προηγούμενη φάση για το σύστημα μας. Για να έχουμε μια καλύτερη εικόνα του προβλήματος μας, παρατηρούμε την απλή εκδοχή, δηλαδή την περίπτωση δύο συστάδων. Έστω $G(V, E, c)$ ένα γράφημα όπως έχει περιγραφεί παραπάνω και $w : V \rightarrow \mathbb{R}_{\geq 0}$. Τότε ψάχνουμε $H \subseteq G \setminus \{p_0\}$, τέτοιο ώστε:

$$\begin{aligned} &\min && c(MST_H) + c(MST_{G \setminus H}) \\ &\text{subject to} && w(H) \leqslant M \\ &&& w(H) \geqslant m \\ &&& H \subseteq G \setminus \{p_0\} \end{aligned} \tag{3.1}$$

Όπου, $M \geqslant m$ πραγματικές σταθερές, με $M + m = w(G \setminus \{p_0\})$ και MST_H και $MST_{G \setminus H}$ είναι τα ΕΓΔ των συνόλων $H \cup \{p_0\}$ και $(G \setminus H) \cup \{p_0\}$ αντίστοιχα.

Algorithm 3: OptimisePhaseA

```
input  : C ∈ S, a, b
output: C ∈ S

for i = 1 to k do
    if w(C_i) − TW > 0 then
        P ← P ∪ C_i
    else
        N ← N ∪ C_i
while P ≠ ∅ do
    for C_i ∈ P do
        for C_j ∈ N do
            for p ∈ C_i do
                C'_i ← C_i \ {p}
                w(C'_i) ← w(C_i) − w(p)
                C'_j ← C_j ∪ {p}
                w(C'_j) ← w(C_j) + w(p)
                Calculate(newsolution) #υπολογίζει τη λύση
                if newsolution < solution then
                    OptimisePhaseA(newsolution)
    P ← P \ C_i
```

Παρατηρούμε λοιπόν, πως το παραπάνω πρόβλημα στον πυρήνα του είναι ένα πρόβλημα διαχωρισμού(partitioning problem). Επομένως, έχει νόημα η προσέγγιση του προβλήματος ως ένα πρόβλημα εύρεσης του μέγιστου κόστους της τομής(Max-Cut) του G, του οποίου και η λύση θα μας έδινε σύνολα των οποίων οι κόμβοι θα ήταν κοντά μεταξύ τους.

Συγκεκριμένα, θα χρησιμοποιήσουμε τον αλγόριθμο συστολής ακμών που παρουσιάστηκε στο άρθρο [5] και θα ελέγχουμε τις συνθήκες ισορροπίας με τη βοήθεια της μεθόδου της μέγιστης διαφοράς(largest differencing method)[7]. Για κάθε ζεύγος συστάδων θα εφαρμόζουμε επαναληπτικά τον αλγόριθμο συστολής μέχρις ότου βρεθεί αποδεκτή λύση ή πληρείτε η συνθήκη τερματισμού. Η φάση θα τερματίζει όταν ο αλγόριθμος δεν μπορεί να βρει καλύτερες λύσεις.Ένας ψευδοκώδικας του αλγορίθμου παρουσιάζεται στον αλγόριθμο 4.

Algorithm 4: OptimisePhaseB

```
input  : C ∈ S
output: C ∈ S

L ← {C_1, ..., C_k}
while |L| > 1 do
    for C_i ∈ L do
        for C_j ∈ L do
            H ← C_i ∪ C_j
            if w(C_i) < w(C_j) then
                upperbound ← w(C_j)
                lowerbound ← w(C_i)
            else
                upperbound ← w(C_i)
                lowerbound ← w(C_j)
            H1, H2 ← Contraction(H)
            S ← {H1, H2}
            if w(H1) < w(H2) then
                limboset ← H2
            else
                limboset ← H1
            while Partition(S) = False do
                if |limboset| = 1 then
                    Break
                H1, H2 ← Contraction(limboset)
                S ← S ∪ {H1, H2}
                S ← S \ limboset
                limboset ← min_{H∈S} min_{p∈V(H)} Σ_{W∈S\H} c(p, W)
            if Partition(S) = True then
                newsolution ← Partition(S)
                Calculate(newsolution) #υπολογίζει τη λύση
                if newsolution < solution then
                    OptimisePhaseB(newsolution)
        L ← L \ C_i
```

Algorithm 5: Contraction

input : $G(V, E, c)$ complete
output: $H \cup W = V, H \cap W = \emptyset$

for $i = 1$**to** $|V|$ **do**
    $contractionlist_i \leftarrow \{i\}$

for $i = 1$**to** $|V| - 2$ **do**
    $c(x, y) \leftarrow \min_{p,q \in V} c(p, q)$
    $V \leftarrow V \cup \{v\} \setminus \{x, y\}$ # συστολή της ακμής (x, y)
    $E \leftarrow E \setminus (x, y)$
    for $i \in V \setminus \{v\}$ **do**
        $c(v, i) \leftarrow c(i, x) + c(i, y)$
    $contractionlist_v \leftarrow contractionlist_x \cup contractionlist_y$

$V(H) \leftarrow contractionlist_h$ # υποθέτουμε ότι $V = \{h, w\}$
$V(W) \leftarrow contractionlist_w$
return H, W

Algorithm 6: Partition

input : S
output: $(True,\ H1, H2)$ ή $(False,\ None)$

for $H \in S$ **do**
    $L \leftarrow L \cup w(H)$

Sort(L) # ταξινομεί την λίστα L κατά φθίνουσα σειρά
while $|L| \neq 1$ **do**
    # αν A το πρώτο στοιχείο της λίστας και B το δεύτερο
    $difference \leftarrow A - B$
    $M \leftarrow M \cup difference$
    $L \leftarrow L \cup difference \setminus \{A, B\}$
    Sort(L)

if $L \leqslant upperbound - lowerbound$ **then**
    # υπάρχειpartition του S και το βρίσκουμε με backtracking στο M
    return $True$,Backtracking(M)

else
    return $False, None$

Κεφάλαιο 4

ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα αποτελέσματα που μας έδωσε ο αλγόριθμος μέσα από τρία πειράματα, σχεδιασμένα έτσι ώστε να ελέγξουν τις επιδόσεις και την αποτελεσματικότητα του αλγορίθμου μας. Η κωδικοποίηση των αλγορίθμων έγινε με τη γλώσσα προγραμματισμού Python(έκδοση 3.7.4) και με επεξεργαστή κειμένου IDLE. Ο υπολογισμός των δρομολογίων έγινε με τον αλγόριθμο OR-Tools της Google. Όλα τα προγράμματα τα «τρέξαμε» σε ένα απλό λάπτοπ της αγοράς. Επίσης, κατασκευάσαμε έναν αλγόριθμο όμοιο με το δικό μας και αντικαταστήσαμε την δεύτερη φάση βελτιστοποίησης που αναπτύξαμε στη ενότητα 3.3 με μια μέθοδο ανταλλαγής κόμβων, θέλοντας να συγκρίνουμε τις δύο διαδικασίες.

Για το πρώτο πείραμα, κατασκευάσαμε τυχαία ολοκληρωμένα(complete)γραφήματα πλήθους κόμβων, $n = 12, 14, 16$, με μικρό αλλά και μεγάλο εύρος για τα βάρη των ακμών τους. Θέσαμε σαν βάρος κόμβου τη μονάδα σε κάθε ένα από αυτά και ορίσαμε ως $threshold$ των αλγορίθμων το $TW/2$ και $k = 2$. Σκοπός του πειράματος είναι να διαπιστώσουμε κατά πόσο ανώτερη είναι η μέθοδος της ανταλλαγής κόμβων σε τέτοιες περιπτώσεις, αλλά και την ποιότητα των λύσεων των δύο, λύνοντας κάθε περίπτωση στο βέλτιστο με πλήρης απαρίθμηση των λύσεων(complete enumeration). «Τρέξαμε» τους αλγορίθμους από πέντε φορές για κάθε περίπτωση, (n, εύρος), και υπολογίσαμε την μέση τιμή των λύσεων τους. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 4.1. Στα κελιά έχουμε βάλει πρώτα την προσέγγιση της βέλτιστης λύσης για τα γραφήματα που κατασκευάστηκαν με μικρό εύρος. Βλέπουμε πως η ανταλλαγή κόμβων(Αλγόριθμος2) βρίσκει καλύτερες λύσεις, όπως και περιμέναμε, άλλα σίγουρα όχι σε τέτοιο βαθμό που να μας αποθαρρύνει από το να χρησιμοποιήσουμε την μέθοδο που αναπτύξαμε στην εργασία αυτή.

Πίνακας 4.1: Σύγκριση Μεθόδων σε Τυχαία Γραφήματα

n	Αλγόριθμος1	Αλγόριθμος2
12	0,988 - 0,984	0,996 - 0,996
14	0,996 - 0,984	0,996 - 0,992
16	0,994 - 0,993	0,999 - 0,997

Σαν δεύτερο πείραμα, θεωρήσαμε την εφαρμογή της ισορροπίας της ζήτησης και συγκρίναμε τις Παρέτο βέλτιστες[1] λύσεις που μας δίνουνε οι δύο αλγόριθμοι για μία συλλογή τεστ(Christofedes and Eilon, Set E - 1969) του προβλήματος δρομολόγησης οχημάτων με περιορισμένη χωρητικότητα (CVRP). Τα γραφήματα των τεστ είναι ολοκληρωμένα γραφήματα στον Ευκλείδειο χώρο. Τα αποτελέσματα του πειράματος δύο παρουσιάζονται στον Πίνακα 4.2. Στα κελιά των αλγορίθμων, έχουμε δώσει τις Παρέτο βέλτιστες λύσεις με τις ελάχιστες τιμές για την συνάρτηση ισορροπίας-αθροιστική απόλυτη διαφορά από TW(κάτω τιμή) και την συνάρτηση κόστους-άθροισμα ΕΓΔ(επάνω τιμή), ενώ σε παρένθεση έχουμε βάλει το συνολικό κόστος των δρομολογίων της λύσης αυτής.

Βλέπουμε πως αρκετές Παρέτο βέλτιστες λύσεις της μεθόδου ανταλλαγής κόμβων(Αλγόριθμος2) κυριαρχούνται(Pareto Dominated) από τις Παρέτο βέλτιστες λύσεις της μεθόδου μας(Αλγόριθμος1), ενώ το αντίθετο δεν συμβαίνει τόσο συχνά. Εδώ θέλουμε να σημειώσουμε ότι, κατά τη διάρκεια του πειράματος διαπιστώσαμε ότι αυξομειώσεις στα βάρη των συναρτήσεων δεν είχαν μεγάλο αποτέλεσμα στην αύξηση του εύρους των λύσεων. Γι αυτό το λόγο όλα τα αποτελέσματα που παρουσιάζονται έχουν όλα παραχθεί για $a = b = 0.5$. Ως $threshold$ των αλγορίθμων ορίσαμε το $TW/2$.

Τέλος θεωρήσαμε, για την ίδια συλλογή τεστ, την εφαρμογή των ισοδύναμων, ως προς τα κόστη, δρομολογίων. Για το πείραμα αυτό κατασκευάσαμε την εκδοχή του αλγορίθμου μας που ικανοποιεί τον περιορισμό της χωρητικότητας. Σκοπός του πειράματος αυτού, είναι πρώτα απ όλα, να εκτιμήσουμε το κατά πόσο ο ορισμός που δώσαμε για το βάρος κόμβου πετυχαίνει ή όχι το στόχο του, δηλαδή αν καταφέρνει να προσεγγίζει τα κόστη ενός δρομολογίου μίας συστάδας και δεύτερον, να κάνουμε μια ανάλυση ευαισθησίας για τον προγραμματισμό δρομολογίων σε τέτοια προβλήματα. Ως $threshold$ των αλγορίθμων

[1] δεν έγινε πλήρης απαρίθμηση των λύσεων, αλλά παρήγαμε ένα σεβαστό αριθμό λύσεων($>$ 20) για διαφορετικά initial seeds μέσω του αλγορίθμου K-means για να έχουμε όσο το δύνατον γίνεται πιο αντιπροσωπευτικά αποτελέσματα

Πίνακας 4.2: Πρώτη Εφαρμογή και Σύγκριση

instance	OPT	Απόκλιση TW	Avg $w(p)$	Αλγόριθμος1		Αλγόριθμος2	
En22k4	375	550	1071	271(415) 350	283(428) 150	272(441) 550	282(412) 150
En51k5	521	12.4	15.54	471(855) 2.4	419(604) 3.6	463(668) 2.4	431(612) 3.6
En76k7	682	229.71	18.18	595(863) 5.14	562(768) 8.85	569(807) 7.14	610(870) 1.7
En101k8	873	183	14.58	658(917) 3	640(908) 4.5	725(1042) 3	711(1012) 8.5

ορίσαμε το $TW/4$, τα αποτελέσματα του πειράματος τρία παρουσιάζονται στον Πίνακα 4.3. Στα κελιά των αλγορίθμων δίνονται οι Παρέτο βέλτιστες λύσεις με τις ελάχιστες τιμές για την συνάρτηση ισορροπίας-αθροιστική απόλυτη διαφορά από TW(κάτω τιμή) και την συνάρτηση κόστους-άθροισμα ΕΓΔ, το συνολικό κόστος των δρομολογίων(επάνω τιμή) και σε παρένθεση η διαφορά του μέγιστου με το ελάχιστο σε κόστος δρομολόγιο της λύσης(range). Από τα αποτελέσματα του πειράματος μας βλέπουμε πως δεν υπάρχει κάποια ουσιαστική διαφορά μεταξύ της περιορισμένης(Αλγόριθμος2) και μη περιορισμένης(Αλγόριθμος1) εκδοχής του αλγορίθμου μας. Επίσης, δεν φαίνεται να υπάρχει κάποια σχέση μεταξύ της απόλυτης αθροιστικής απόκλισης των βαρών των συστάδων και την ισορροπία των δρομολογίων ως προς το κόστος και όποια ισορροπία επιτυγχάνεται, μάλλον, οφείλετε στον σχεδιασμό των συγκεκριμένων δικτύων.

Πίνακας 4.3: Δεύτερη Εφαρμογή και Σύγκριση

instance	Range	Απόκλιση TW	Avg $w(p)$	Αλγόριθμος1		Αλγόριθμος2	
En22k4	36	191	27.7	426(9) 17	461(49) 7	393(40) 191	468(69) 7
En51k5	21	180	23.96	576(20) 14.4	664(60) 5.59	690(109) 145	560(52) 146.4
En76k7	97	531	24.14	710(35) 12.28	806(43) 23.71	834(47) 257.42	– –
En101k8	98	698	24.86	1011(59) 8	993(56) 10	990(67) 44.5	– –

Κεφάλαιο 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ

Στην εργασία αυτή, προσπαθήσαμε να εισαγάγουμε ένα γενικό μοντέλο για την αντιμετώπιση του προβλήματος εξισορρόπησης του εργασιακού φόρτου(workload equity)σε προβλήματα δρομολόγησης οχημάτων. Έμπνευση μας αποτέλεσε η δουλεία των συγγραφέων του άρθρου [3].Μέσα από τις παρατηρήσεις και τις υποδείξεις, των συγγραφέων, σχηματίσαμε ένα μοντέλο του οποίου η ουσία βρίσκεται στην ευκαμψία του και στην ικανότητα του να δώσει κατανοητές λύσεις σε σχετικά προβλήματα, αρκεί ο σχεδιαστής να καταφέρει να προσεγγίσει σωστά την, προς ισορροπία, μετρική(equity metric).Τονίζουμε την αναγκαιότητα για ευκολία στην ερμηνεία των λύσεων, εξαιτίας της φύσης των προβλημάτων αυτών.

Στην συνέχεια, περιγράφουμε έναν άπληστο ευρετικό αλγόριθμο για την επίλυση του μοντέλου μας βασισμένο στον αλγόριθμο του άρθρου [4]. Η βασική διαφορά του δικού μας αλγορίθμου από αυτόν του άρθρου [4] βρίσκεται στον τρόπο εύρεσης λύσεων, συγκεκριμένα, ο δικό μας αλγόριθμος προσπαθεί να ισορροπήσει, όσο του δίνεται η δυνατότητα περισσότερο την λύση του και στη συνέχεια διατηρώντας αυτή την ισορροπία να βελτιώσει τη λύση, σε αντίθεση, με τον αλγόριθμο[4] που η διαδικασία αυτή γίνεται ταυτόχρονα. Επίσης, αναγνωρίζοντας πως το πρόβλημα που αντιμετωπίζουμε, είναι ένα πρόβλημα διαχωρισμού στον πυρήνα του, δημιουργούμε μία μέθοδο με κύριο μηχανισμό με την εύρεση τομών μέγιστου κόστους.

Τέλος, μέσα από τα υπολογιστικά αποτελέσματα σημειώνουμε τα πλεονεκτήματα και τις αδυναμίες του αλγορίθμου. Ειδικότερα, ο αλγόριθμος φαίνεται να φέρνει καλά αποτελέσματα(ειδικά για την ισορροπία ζήτησης) και να είναι ικανός να προσεγγίσει τη δομή των προβλημάτων, όμως η αδυναμία του να παράγει ένα ευρύ σύνολο λύσεων σε συνδυασμό με τη χρήση προσεγγιστικών μεθόδων, αφήνει αρκετό περιθώριο για βελτίωση. Γι αυτόν το λόγο μελλοντικές έρευνες θα πρέπει να εστιάσουν στην αντιμετώπιση αυτών των μειονεκτημάτων και συγκεκριμένα, στην αναζήτηση διαφορετικών τρόπων μοντελοποίησης(π.χ. ε-φραγμένες συναρτήσεις) και πιο φιλοσοφημένων μεθόδων διαχωρισμού.

Κεφάλαιο 6

ΠΑΡΑΡΤΗΜΑ

Έστω $G(V, E, c)$ ένα ολοκληρωμένο(complete) γράφημα με $c : E \to \mathbb{R}$ και έστω HC_G^* το ελάχιστο σε κόστος Hamiltonian Cycle του G. Έστω $e \in E(HC_G^*)$ μια τυχαία ακμή του HC_G^*, τότε το γράφημα $T(V(HC_G^*), E(HC_G^*) \setminus \{e\}, c)$ αποτελεί ένα δέντρο για το γράφημα G, οπότε $c(T) \leqslant c(HC_G^*)$. Όμως, $c(MST) \leqslant c(T), \forall T$ του G, άρα $c(MST) < c(HC_G^*)$.
Για να αποδείξουμε τη δεύτερη ανίσωση, $c(HC^*) \leqslant 2c(T)$, χρησιμοποιούμε τον αλγόριθμοDFSστο MST του G και εφαρμόζουμε την τριγωνική ανισότητα.

Βιβλιογραφία

[1] Marius M. Solomon, “Algorithms for the Vehicle Routing and Scheduling Problems with Time Windows," *Northeastern University, Boston, Massachuset*,1985

[2] Suresh Nanda Kumar, Ramasamy Panneerselvam, “A Survey on the Vehicle Routing Problem and its Variants," *Education Department, CII Institute of Logistics, Chennai, India* ,2012

[3] P. Matl, R.F. Hartl, “Workload Equity in Vehicle Routing Problems: A Survey and Analysis," *University of Vienna, Austria*,2018

[4] Buyang Cao, Fred Glover, “Creating Balanced and Connected Clusters to Imrove Service Delivery Routes in Logistics Planning ," *Systems Engineering Society of China & Springer-Verlag Berlin Heidelberg*, 2010

[5] Sera Kahruman, Elif Kolotoglu, Sergiy Butenko, Illya V. Hicks, “On Greedy Construction Heuristics for the MAX-CUT problem," *Department of Industrial and Systems Engineering Texas A& M University*, 2007

[6] P. Festa , P.M. Pardalos , M.G.C. Resende, C.C. Ribeiro “Randomized heuristics for the Max-Cut problem," 2002

[7] Narenda Karmakar, Richard M. Karp, “The Differencing Method of Set Partitioning," *University of California, Berkeley*, 1982

[8] Sartaj Sahni, Teofilo Gonzalez, “P-Complete Approximation Problems," *University of Minnesota, Minneapolis, Minnesota* , 1976