

Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών
Υπολογιστών

Αλγόριθμοι συνδυαστικής
βελτιστοποίησης για προβλήματα
χωροθέτησης μονάδων παραγωγής

Δήμητρα Τσιικίτικου (ΑΜ: 1565)
Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

Εργαστήριο Ευφύων Συστημάτων & Βελτιστοποίησης
4 Δεκεμβρίου 2023

Περίληψη

Στην παρούσα διπλωματική εργασία εξετάζεται το θέμα του προβλήματος χωροθέτησης μονάδων παραγωγής. Συγκεκριμένα, επίκεντρο της εργασίας είναι το πρόβλημα του p μέσου. Αυτό το πρόβλημα χωροθέτησης μονάδων παραγωγής έχει ως στόχο να ελαχιστοποιήσει την απόσταση των κόμβων ζήτησης από τις εγκαταστάσεις που θα χωροθετηθούν. Αρχικά, πραγματοποιήθηκε η διατύπωση του προβλήματος και ο εντοπισμός αλγόριθμων επίλυσης του προβλήματος. Μετά την επιλογή των αλγόριθμων υλοποιήθηκε η μοντελοποίηση του προβλήματος ως μοντέλο γραμμικού προγραμματισμού. Έπειτα έγινε χρήση του λύτη Curobi για την εύρεση των βέλτιστων λύσεων αλλά και για τη χρήση του σε διάφορες τεχνικές για να γίνει σύγκριση επ' αυτού. Επίσης, υλοποιήθηκε η μέθοδος διακλάδωσης και περιορισμού (branch and bound) με τέσσερις τεχνικές. Οι πρώτες δύο αφορούν τον τρόπο εξερεύνησης κλαδιών και οι υπόλοιπες δύο ενσωματώνουν την προσθήκη μιας ευρετικής μεθόδου. Ακόμα, περιγράφονται όλοι οι υλοποιημένοι κώδικες (ευρετικοί και μεθευρετικοί) με τη συνοδεία του αντίστοιχου ψευδοκώδικα. Συνοπτικά παρατηρείται ότι η καλύτερη επιλογή αλγορίθμου, ως προς την απόκλιση αλλά και τον χρόνο εκτέλεσης, είναι η χρήση της μεθόδου διακλάδωσης και περιορισμού με την τεχνική κατά βάθος και περιορισμό απόκλισης 1%. Με τη χρήση αυτού του αλγορίθμου παρουσιάζεται απόκλιση 0.12% από τη βέλτιστη με χρόνο 3.97sec.

Λέξεις κλειδιά: χωροθέτηση μονάδων παραγωγής, πρόβλημα p μέσου, διακλάδωση και περιορισμός, ευρετικές μέθοδοι, μεθευρετικές μέθοδοι.

Abstract

In this thesis, the problem of locating facilities is examined. Specifically, the focus of the thesis is the p median problem. This facility location problem aims to minimize the distance of demand nodes from the facilities they are to be located at. Initially, the problem was described and algorithms for solving the problem were identified. After selecting the algorithms, the problem was modeled as a linear programming model. Then, the Gurobi solver was used to find the most optimal solutions as well as to be applied in various techniques for comparison purposes. Additionally, the branch and bound method was implemented with four techniques. The first two are about the way branches are explored, and the remaining two incorporate the addition of a heuristic method. Furthermore, all the implemented algorithms (heuristics and metaheuristics) are described together with their corresponding pseudocode. In summary, it is observed that the best algorithm choice, in terms of deviation as well as execution time, is the use of the branch-and-bound method with the depth first search and 1% deviation restriction technique. Using this algorithm, a deviation of 0.12% from the optimal solution is achieved with an execution time of 3.97sec.

Keywords: facility location, p median problem, branch and bound, heuristic methods, metaheuristic methods

Δήλωση Πνευματικών Δικαιωμάτων

Δήλωση Πνευματικών Δικαιωμάτων Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο "Αλγόριθμοι συνδυαστικής βελτιστοποίησης για προβλήματα χωροθέτησης μονάδων παραγωγής" καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Νικόλαου Πλόσκα αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Δήμητρα Τσιικίτικου & Νικόλαος Πλόσκας, 2023, Κοζάνη

Υπογραφή Φοιτητή
Δήμητρα Τσιικίτικου

Ευχαριστίες

Θα ήθελα να ευχαριστήσω, αρχικά, τον επιβλέποντα καθηγητή μου, κύριο Νικόλαο Πλόσκα, για την βοήθεια και την καθοδήγηση που μου έδωσε καθ' όλη την διάρκεια της ενασχόλησης μου με την παρούσα ερευνητική εργασία. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου, καθώς πάντα βρίσκεται δίπλα μου να με στηρίζει και να μου προσφέρει δύναμη.

Περιεχόμενα

1	Εισαγωγή	12
1.1	Ορισμός του προβλήματος	12
1.2	Κίνητρα και στόχοι υλοποίησης	13
1.3	Διάρθρωση κειμένου	13
2	Βιβλιογραφική ανασκόπηση	14
2.1	Διατύπωση του προβλήματος p μέσου	14
2.2	Μοντελοποίηση	15
2.3	Μέθοδοι επίλυσης	16
3	Υλοποίηση	20
3.1	Μοντέλο γραμμικού προγραμματισμού - περίπτωση του λύτη βελτιστοποίησης Gurobi	20
3.1.1	Γενικά	20
3.1.2	Τρόπος υλοποίησης	21
3.1.3	Ψευδοκώδικας της μοντελοποίησης του γραμμικού προβλήματος με χρήση λύτη Gurobi	22
3.2	Μέθοδος Branch and Bound	24
3.2.1	Γενικά	24
3.2.2	Τρόπος υλοποίησης	25
3.2.3	Ψευδοκώδικας του Branch and Bound	28
3.3	Μέθοδος Dijkstra	30
3.3.1	Γενικά	30
3.3.2	Τρόπος υλοποίησης	30
3.3.3	Ψευδοκώδικας του Dijkstra	30
3.4	Μυωπικός αλγόριθμος	30

3.4.1	Γενικά	30
3.4.2	Τρόπος υλοποίησης	31
3.4.3	Ψευδοκώδικας του μωπικού αλγορίθμου	32
3.5	Αλγόριθμος Neighborhood Search	32
3.5.1	Γενικά	32
3.5.2	Τρόπος υλοποίησης	33
3.5.3	Ψευδοκώδικας του neighborhood search αλγορίθμου	33
3.6	Αλγόριθμος Tabu Search	33
3.6.1	Γενικά	33
3.6.2	Τρόπος υλοποίησης	34
3.6.3	Ψευδοκώδικας του tabu search αλγορίθμου	35
3.7	Αλγόριθμος Drop Approach	35
3.7.1	Γενικά	35
3.7.2	Τρόπος υλοποίησης	35
3.7.3	Ψευδοκώδικας του Drop αλγορίθμου	37
3.8	Αλγόριθμος Subdrop Approach	38
3.8.1	Γενικά	38
3.8.2	Τρόπος υλοποίησης	38
3.8.3	Ψευδοκώδικας του subdrop αλγορίθμου	38
3.9	Αλγόριθμος Exchange Approach	39
3.9.1	Γενικά	39
3.9.2	Τρόπος υλοποίησης	39
3.9.3	Ψευδοκώδικας του exchange αλγορίθμου	40
4	Υπολογιστική μελέτη	41
4.1	Διαδικασία πειράματος	41
4.2	Σύγκριση αλγορίθμων	42
4.2.1	Ανάλυση αποτελεσμάτων αλγορίθμων	44
4.3	Αποτελέσματα μετρήσεων αλγορίθμων	46
5	Συμπεράσματα	63
	Παραρτήματα	73

Κατάλογος σχημάτων

1.1	Παράδειγμα p μέσου	13
3.1	Παράδειγμα προβλήματος pmed1 με λύτη Gurobi part1	22
3.2	Παράδειγμα προβλήματος pmed1 με λύτη Gurobi part2	24
3.3	Παράδειγμα προβλήματος pmed1 με λύτη Gurobi part3	24
3.4	Παράδειγμα branch and bound BFS - DFS τεχνική [44]	26
3.5	Παράδειγμα branch and bound BFS-myopic	27
4.1	Γεωμετρικός μέσος χρόνου εκτέλεσης με λύτη	43
4.2	Γεωμετρικός μέσος χρόνου εκτέλεσης χωρίς λύτη	43
4.3	Μέσος όρος απόκλισης από τη βέλτιστη λύση	44

Κατάλογος αλγορίθμων

1	Μοντελοποίηση προβλήματος με χρήση Gurobi	23
2	Αλγόριθμος Branch and Bound	29
3	Αλγόριθμος Dijkstra	31
4	Αλγόριθμος Myopic	32
5	Αλγόριθμος Neighborhood Search	34
6	Αλγόριθμος Tabu Search	36
7	Αλγόριθμος Drop Search	37
8	Αλγόριθμος Subdrop Approach	39
9	Αλγόριθμος Exchange Approach	40

Κατάλογος πινάκων

2.1	Μέθοδοι επίλυσης	17
4.1	Αποτελέσματα Gurobi	48
4.2	Αποτελέσματα Branch and Bound	49
4.3	Αποτελέσματα DFS με Myopic Gap 1%	50
4.4	Αποτελέσματα DFS με Myopic Gap 5%	51
4.5	Αποτελέσματα Myopic	52
4.6	Αποτελέσματα Neighborhood	53
4.7	Αποτελέσματα Neighborhood με Myopic	54
4.8	Αποτελέσματα Tabu	55
4.9	Αποτελέσματα Exchange	56
4.10	Αποτελέσματα Subdrop	57
4.11	Αποτελέσματα Drop	58
4.12	Χρόνοι εκτέλεσης με λύτη	59
4.13	Χρόνοι εκτέλεσης χωρίς λύτη	60
4.14	Αποκλίσεις αλγορίθμων	61
4.15	Συγκεντρωτικά αποτελέσματα αλγορίθμων	62
A.1	Αποτελέσματα Gurobi	75
A.2	Αποτελέσματα Branch and Bound	76
A.3	Αποτελέσματα DFS με Myopic Gap 1%	77
A.4	Αποτελέσματα DFS με Myopic Gap 5%	78
A.5	Αποτελέσματα Myopic	79
A.6	Αποτελέσματα Neighborhood	80
A.7	Αποτελέσματα Neighborhood με Myopic	81
A.8	Αποτελέσματα Tabu	82
A.9	Αποτελέσματα Exchange	83

A'.10 Αποτελέσματα Subdrop	84
A'.11 Αποτελέσματα Drop	85
A'.12 Χρόνοι εκτέλεσης με λύτη	86
A'.13 Χρόνοι εκτέλεσης χωρίς λύτη	87
A'.14 Αποκλίσεις αλγορίθμων	88
A'.15 Συγκεντρωτικά αποτελέσματα αλγορίθμων	89

Κεφάλαιο 1

Εισαγωγή

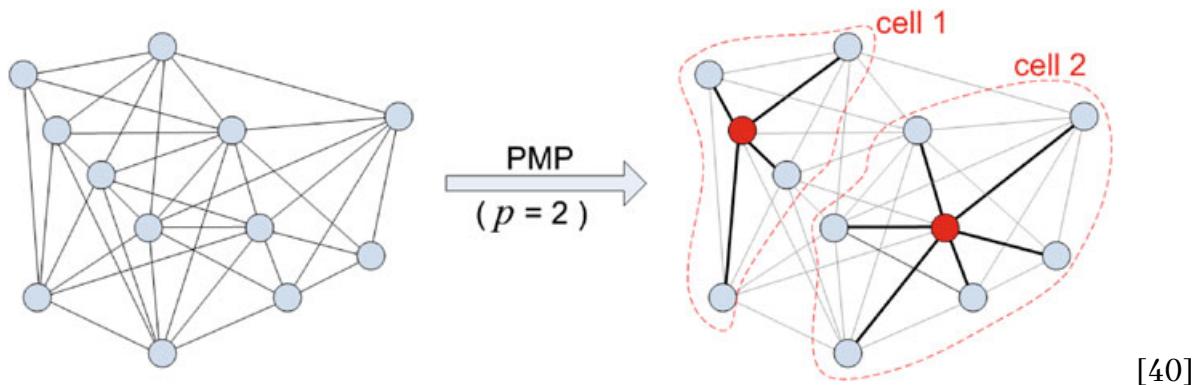
1.1 Ορισμός του προβλήματος

Το πρόβλημα της χωροθέτησης μονάδων παραγωγής αναφέρεται στην εύρεση p σημείων για την εγκατάσταση p επιχειρήσεων/μονάδων. Συγκεκριμένα, σκοπός αυτού του προβλήματος είναι η επιλογή p σημείων που είναι σε θέση να ελαχιστοποιήσουν το κόστος μέσα από έναν προεπιλεγμένο αριθμό σημείων. Το πρόβλημα αυτό διαθέτει τέσσερις παραλλαγές [18][20]. Επιγραμματικά οι παραλλαγές αυτές είναι οι εξής: p μέσου (p -median), p κέντρου (p -center), p διασποράς (p -dispersion) και το πρόβλημα κάλυψης (covering problems). Η διαφορά των παραπάνω μεθόδων αφορά κυρίως στην έννοια του κόστους. Το πρόβλημα χωροθέτησης προσπαθεί να ελαχιστοποιήσει το συνολικό κόστος για την εγκατάσταση και τη μεταφορά των πελατών στην εγκατάσταση. Το πρόβλημα p μέσου επικεντρώνεται στη μείωση του κόστους μεταφοράς των πελατών στις εγκαταστάσεις. Την ίδια στιγμή, το πρόβλημα του p κέντρου εστιάζει στην ελάττωση της μέγιστης απόστασης μεταξύ εγκατάστασης και πελατών. Τέλος, το πρόβλημα καλύμματος ασχολείται είτε με την εύρεση του ελάχιστου αριθμού εγκαταστάσεων για να εξυπηρετούνται όλοι οι πελάτες, είτε με την αντιστοίχιση των πελατών στις ανοικτές εγκαταστάσεις, σε περίπτωση που δίνεται ως δεδομένο ο απαιτούμενος αριθμός για τις μονάδες που πρέπει να ανοιχτούν. Στην παρούσα εργασία, ασχοληθήκαμε με το πρόβλημα p μέσου καθώς το κόστος για τη μεταφορά του πελάτη στις εγκαταστάσεις θεωρήθηκε πιο σημαντικός παράγοντας.

Στο Σχήμα 1.1 ακολουθεί παράδειγμα του προβλήματος. Ειδικότερα στο αριστερό κομμάτι απεικονίζεται το σύνολο από δώδεκα (12) κόμβους ενός γράφου,

όπου με μπλε εμφανίζονται οι πιθανοί κόμβοι ζήτησης αλλά και οι συνδέσεις μεταξύ τους. Ενώ στο δεξί κομμάτι απεικονίζεται η επιλογή των βέλτιστων σημείων για το συγκεκριμένο πρόβλημα p μέσου με $p = 2$. Με κόκκινο είναι ο κόμβος που θα ανοίξει και με μαύρες έντονες γραμμές απεικονίζεται η βέλτιστη σύνδεση μεταξύ των πελατών και των κόμβων που θα ανοίξουν.

Σχήμα 1.1: Παράδειγμα p μέσου



[40]

1.2 Κίνητρα και στόχοι υλοποίησης

Το πρόβλημα χωροθέτησης μονάδων παραγωγής είναι πολύ σημαντικό και αφορά και τον ιδιωτικό και τον δημόσιο τομέα, καθώς το άνοιγμα εγκαταστάσεων μπορεί να βρίσκει χώρα σε διάφορα είδη επιχειρήσεων π.χ. καφετεριών ή ακόμα για το άνοιγμα κρατικών μονάδων π.χ. νοσοκομείων. Η υλοποίηση κάποιων από τους υπάρχοντες αλγόριθμους στοχεύει στο να συγκριθούν τα αποτελέσματα και να βρεθούν ποιοι είναι οι κατάλληλοι αλγόριθμοι σε κάθε περίπτωση. Έτσι, ανάλογα με τον χρόνο, τον χώρο και την απόκλιση που θέλει κάποιος θα μπορεί να επιλέξει τον κατάλληλο αλγόριθμο.

1.3 Διάρθρωση κειμένου

Η παρούσα εργασία αποτελείται από πέντε κεφάλαια. Στο δεύτερο κεφάλαιο υπάρχει η εισαγωγή στο αντικείμενο της εργασίας. Στο επόμενο κεφάλαιο ακολουθεί η επεξήγηση του προβλήματος με τη μοντελοποίηση και τις μεθόδους επίλυσης. Στο τέταρτο κεφάλαιο υπάρχει η υπολογιστική μελέτη με την παρουσίαση των αποτελεσμάτων μέσω πινάκων και γραφικών παραστάσεων. Στο πέμπτο κεφάλαιο παρατίθενται τα συμπεράσματα της έρευνας.

Κεφάλαιο 2

Βιβλιογραφική ανασκόπηση

2.1 Διατύπωση του προβλήματος p μέσου

Το πρόβλημα χωροθέτησης μονάδων ασχολείται με την επιλογή των p σημείων εγκατάστασης μονάδων. Τα προβλήματα αυτά μπορούν να χωριστούν σε τρεις κατηγορίες. Η πρώτη κατηγορία βασίζεται με τον αν υπάρχει περιορισμός για τη χωρητικότητα των μονάδων, για παράδειγμα αν υπάρχει περιορισμός στον αριθμό πελατών που μπορεί να δεχτεί κάθε μονάδα. Η δεύτερη κατηγορία ασχολείται με το αν το πρόβλημα είναι συνεχές ή διακριτό. Η τελευταία κατηγορία ασχολείται με τον διαχωρισμό τους σε στοχαστικά και ντετερμινιστικά, δηλαδή σε περίπτωση που το πρόβλημα δε διαθέτει ξεκάθαρα τους περιορισμούς ή τα κόστη από το πρόβλημα ή ακόμα και τα δυο συνδυαστικά τότε το πρόβλημα θεωρείται στοχαστικό. Σε αντίθετη περίπτωση είναι ντετερμινιστικό [18].

Το πρόβλημα p μέσου της κατηγορίας προβλημάτων χωροθέτησης μονάδων παραγωγής αποτελεί ένα NP-hard πρόβλημα. Οι Kariv και Hakimi [55] αναφέρουν ότι με τη χρήση κάποιων αλγορίθμων και την περίπτωση που ο γράφος είναι δέντρο είναι εφικτό να λυθούν σε πολυωνυμικό χρόνο. Γενικά το πρόβλημα p μέσου έχει σκοπό την ελαχιστοποίηση του κόστους μεταφοράς από τον πελάτη στην εγκατάσταση. Συγκεκριμένα δέχεται σαν δεδομένα τρεις πληροφορίες, τον αριθμό Z κόμβων ζήτησης, τον αριθμό Y κόμβων πιθανών θέσεων για εγκατάσταση μονάδας και τον αριθμό p που αναφέρεται στο πόσες εγκαταστάσεις πρέπει να ανοίξουν. Η πρόθεση είναι να βρεθεί ο κατάλληλος συνδυασμός p από τους Y κόμβους που θα ελαχιστοποιούν το κόστος μεταφοράς.

2.2 Μοντελοποίηση

Παρακάτω παρουσιάζεται η μοντελοποίηση του προβλήματος p μέσου [20] ως ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού. Όπως προαναφέρθηκε, ο στόχος αυτών των προβλημάτων είναι να βρεθούν, από το σύνολο των υποψήφιων κόμβων εξυπηρέτησης, οι p μονάδες προς χωροθέτηση.

Γι' αυτό χρειάστηκε το $Y=\{1, 2, \dots, m\}$ σύνολο των υποψήφιων κόμβων εξυπηρέτησης, το $Z=\{1, 2, \dots, n\}$ σύνολο κόμβων ζήτησης και τον p αριθμό των μονάδων προς χωροθέτηση. Με τον συνδυασμό των δύο συνόλων δημιουργείται μια δυαδική μεταβλητή c_{ij} μεγέθους $m \times n$ που θα αποτελείται από τα κόστη εξυπηρέτησης για κάθε πελάτη i από την εγκατάσταση j . Επίσης, χρειάζεται και μια δυαδική μεταβλητή x_{ij} , που θα δείχνει αν ο πελάτης i εξυπηρετείται από το κόμβο j . Πιο αναλυτικά, όταν το $x_{ij} = 1$ τότε ο πελάτης i εξυπηρετείται από την εγκατάσταση j , σε αντίθετη περίπτωση όχι. Επίσης, χρειάζεται και μια μεταβλητή y_j , που θα αντιστοιχίζεται σε κάθε κόμβο εγκατάστασης και θα μας φανερώνει αν αυτή η εγκατάσταση θα είναι ανοικτή ($y_j = 1$) ή κλειστή ($y_j = 0$).

Η αντικειμενική συνάρτηση του προβλήματος βελτιστοποίησης είναι:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

με περιορισμούς:

$$\sum_{j=1}^n x_{ij} = 1, \forall i \in Y$$

$$\sum_{j=1}^n y_j = p$$

$$x_{ij} - y_j \leq 0, \forall i \in Y, \forall j \in Z$$

$$y_j \in \{0, 1\}, \forall j \in Z$$

$$x_{ij} \in \{0, 1\}, \forall i \in Y, \forall j \in Z$$

Ο πρώτος περιορισμός θέλει να εξασφαλίσει ότι όλοι οι πελάτες θα έχουν τουλάχιστον μία εγκατάσταση που τους εξυπηρετεί. Ο δεύτερος περιορισμός εξασφαλίζει τη χωροθέτηση p ακριβών μονάδων. Ο τρίτος περιορισμός επιτρέπει την ανάθεση κόμβων ζήτησης σε κόμβους που έχουν επιλεχτεί ως εξυπηρετητές. Ο τέταρτος και

ο πέμπτος περιορισμός δείχνει πως η μεταβλητή y και x είναι δυαδικές.

2.3 Μέθοδοι επίλυσης

Στον Πίνακα 2.1 αναφέρονται εργασίες που έχουν ασχοληθεί με τις συγκεκριμένες τεχνικές επίλυσης [67]. Οι τύποι των μεθόδων επίλυσης είναι οι εξής: Constructive heuristics (CH), Local search (LS), Mathematical programming (MP) και MetaHeuristics (MH).

Οι μέθοδοι επίλυσης των προβλημάτων έχουν αλλάξει σκοπό. Πιο παλιά ήθελαν την επίτευξη βέλτιστης λύσης ή ακριβών αποτελεσμάτων. Πλέον η έρευνα έχει επικεντρωθεί στην εύρεση εφικτής προσεγγιστικής λύσης [87]. Συγκεκριμένα, οι ευρετικές μέθοδοι ψάχνουν μια προσεγγιστική λύση χωρίς να χρειάζεται μια αυστηρή μαθηματική απόδειξη σύγκλισης. Ακόμα δεν εξερευνά όλες τις πιθανές λύσεις με αποτέλεσμα να κερδίζει χρόνο και χώρο στη μνήμη [3]. Η τοπική αναζήτηση βασίζεται στην επαναληπτική βελτίωση της λύσης. Πιο αναλυτικά, η μέθοδος αυτή παίρνει μια πιθανή λύση ως αρχική και προσπαθεί σε κάθε βήμα να τη βελτιώσει [57] [62]. Από την άλλη πλευρά, ο μαθηματικός προγραμματισμός ασχολείται με την ελαχιστοποίηση ή τη μεγιστοποίηση της αντικειμενικής συνάρτησης, καθώς προσπαθεί να βελτιώσει τον στόχο του προβλήματος [94]. Οι μεθευρετικές μέθοδοι επιδιώκουν να βρουν σχεδόν βέλτιστες λύσεις με την εφαρμογή ενός ή και περισσότερων ευρετικών μεθόδων [3].

Στην κατασκευαστική μέθοδο ανήκουν οι τέσσερις ακόλουθοι μέθοδοι. Η άπληστη μέθοδος αποτελεί μια από τις πιο εύκολες επιλογές για τη λύση του προβλήματος. Η διαδικασία αυτής της μεθόδου είναι οι συνεχόμενες επιλογές της τοπικής βέλτιστης επιλογής. Έτσι, κάθε φορά επιλέγεται μια βέλτιστη επιλογή και προστίθεται στη λύση του προβλήματος. Αυτό επαναλαμβάνεται p φορές, ώστε να βρεθούν όλες οι επιθυμητές μονάδες εγκατάστασης [24].

Η stingy ή αλλιώς drop approach μέθοδος είναι μια ανάποδη τεχνική. Ξεκινάει θεωρώντας όλες τις μονάδες ανοικτές. Στη συνέχεια κλείνει έναν έναν τους κόμβους, με στόχο να βρει ποιος κόμβος με το κλείσιμο του θα έχει τη μικρότερη αύξηση στο κόστος [29]. Μετά από μερικά χρόνια οι Salhi και Atkinson [84] έκαναν μια παραλλαγή της μεθόδου. Συγκεκριμένα η αλλαγή είναι η προσθήκη της τυχαίας επιλογής των κόμβων δοκιμής. Αντίθετα με την drop, η subdrop δεν εξετάζει όλους

Πίνακας 2.1: Μέθοδοι επίλυσης

Type	Heuristic	References
CH	Greedy	Kuehn and Hamburger (1963) [58], Whitaker (1983) [93]
	Stingy	Feldman et al. (1966) [29], Moreno-Perez et al. (1991) [70], Salhi and Atkinson (1995) [84]
	Dual ascent	Galvao (1980, 1993) [32] [33], Erlenkotter (1978) [27], Captivo (1991) [11]
	Composite	Moreno-Perez et al. (1991) [70], Captivo (1991) [11], Pizzolato (1994) [74], Salhi (1997) [82]
LS	Alternate	Maranzana (1964) [63]
	Interchange	Teitz and Bart (1968) [89], Whitaker (1983) [93], Densham and Rushton (1992) [21] Hansen and Mladenovic (1997) [47], Resende and Werneck (2003) [75], Kochetov et al. (2005) [56]
MP	Dynamic programming	Hribar and Daskin (1997) [54]
	Lagrangian relaxation	Cornuejols et al. (1977) [14], Mulvey and Crowder (1979), Galvao (1980), n Cornuejols et al. (1977), Mulvey and Crowder (1979) [71], Galvao (1980) [32], Beasley (1993) [7], Daskin (1995) [19], Senne and Lorena (2000) [59], Barahona and Anbil (2000) [4] Beltran et al. (2004) [9]
	Aggregation	Hillsman and Rhoda (1978) [49], Goodchild (1979) [42], Erkut and Bozkaya (1999) [26], Casillas (1987) Current and Schilling (1987) [16], Hodgson and Neuman (1993) [51], Hodgson and Salhi (1998) [52] Bowerman et al. (1999) [10], Francis et al. (2000, 2003) [31]
MH	Tabu search	Mladenovic et al. (1995, 1996) [68][69], Voss (1996) [90], Rolland et al. (1996) [76], Salhi (2002) [83] Kochetov (2001), Goncharov and Kochetov (2002) [41]
	Variable neighborhood search	Hansen and Mladenovic (1997) [47], Hansen et al. (2001) [48], Garcia-Lopez et al. (2002) [35] Crainic et al. (2004) [15]
	Genetic search	Hosage and Goodchild (1986) [53], Dibbie and Densham (1993) [22], Moreno-Perez et al. (1994) [70] Estivill-Castro (1999) [28], Alp et al. (2003) [1], Chaudhry et al. (2003) [12]
	Simulated annealing	Murray and Church (1996) [72], Chiyoshi and Galvao (2000) [13], Levanova and Loresh (2004) [61]
	Heuristic concentration	Rosing et al. (1998) [78], Rosing and ReVelle (1997) [80], Rosing et al. (1999) [79]
	Scatter search	Garcia-Lopez et al. (2003) [86]
	Ant colony	Levanova and Loresh (2004) [61]
	Neural networks	Dominguez Merino and Munoz Perez (2002) [65], Dominguez Merino et al. (2003) [64]
	Decomposition	Dai and Cheung (1997) [17], Taillard (2003) [88]
	Hybrids	Resende and Werneck (2004) [75]

τους κόμβους για να επιλέξει ποιον θα κλείσει αλλά εξετάζει μια ομάδα κόμβων που η επιλογή τους έγινε τυχαία.

Η dual ascent λύνει τη χαλάρωση του γραμμικού προγραμματισμού. Επίσης, η μέθοδος αυτή δημιουργεί αυστηρά κάτω όρια, τα οποία τα εισάγουν και στη μέθοδο διακλάδωσης και περιορισμού (branch and bound) [34]. Τέλος, η composite μέθοδος, δηλαδή μια μέθοδος που συνδυάζει πολλαπλές ευρετικές μεθόδους, όπως για παράδειγμα ο Moreno-Perez προτείνει μια παραλλαγή του greedy drop. Πρόκειται για συνδυασμό των Greedy, Alternate και Multistart [67]. Όλες αυτές οι μέθοδοι βρίσκουν μια προσεγγιστική εφικτή λύση χωρίς την ύπαρξη αρχικής λύσης.

Οι ακόλουθες μέθοδοι ανήκουν στη κατηγορία της τοπικής αναζήτησης. Η εναλλακτική μέθοδος Alternate ξεκινάει με μια αρχική λύση. Με βάση αυτή τη λύση χωρίζει τους κόμβους σε γειτονιές. Σε κάθε γειτονιά βρίσκεται η καλύτερη λύση με τη χρήση ενός ευρετικού αλγορίθμου. Αν η λύση που θα βρεθεί είναι διαφορετική από την υπάρχουσα τότε ενημερώνεται η αρχική λύση. Η διαδικασία αυτή σταματάει όταν οι λύσεις από τις γειτονιές είναι ίδιες με την υπάρχουσα [63]. Η μέθοδος αντικατάστασης Interchange, που είναι γνωστή από τους Teitz and Bart [89], ξεκινάει και αυτή με μια αρχική λύση. Στη συνέχεια, η διαδικασία συνεχίζει με το να ελέγξει όλους τους κόμβους που δεν αποτελούν λύση σε όλες τις πιθανές θέσεις για να δοκιμαστούν όλοι οι σχεδιασμοί, με σκοπό να βρεθεί λιγότερο κόστος από το τωρινό και να γίνει αλλαγή στη μεταβλητή που αποθηκεύει τους κόμβους που αποτελούν λύση.

Στον μαθηματικό προγραμματισμό ανήκουν ο δυναμικός προγραμματισμός και η χαλάρωση Lagrangian και η Aggregation. Ο δυναμικός προγραμματισμός έγινε γνωστός από τον Bellman [8] [54]. Η μέθοδος αυτή με αφορμή την κατάσταση που βρίσκεται αποφασίζει την επόμενη καλύτερη κίνηση. Η χαλάρωση Lagrangian αναπτύχθηκε το 1970 [30]. Η τεχνική αυτή ασχολήθηκε με την ενημέρωση των πολλαπλασιαστών Lagrange. Ακόμα, η τεχνική της χαλάρωσης Lagrangian χρησιμοποιείται και για την απόδοση ορίων στη μέθοδο της διακλάδωσης και του περιορισμού που χρησιμοποιείται στον γραμμικό προγραμματισμό. Η Aggregation που ασχολήθηκαν και οι Erkut and Bozkaya αφορά στη συνάθροιση και στα σφάλματα που επιφέρουν. Τα σφάλματα αυτά είναι αποτέλεσμα των αλλαγών που γίνεται στο πρόβλημα για να είναι πιο διαχειρίσιμο [26].

Στη συνέχεια ακολουθούν οι μεθευρετικές μέθοδοι. Η αναζήτηση Tabu ξεκινάει από μία αρχική λύση και κρατάει τις κινήσεις σε έναν πίνακα, ώστε να αποφευχθούν οι ίδιες κινήσεις [68] [69] [90] [76] [83]. Η αναζήτηση πολύτιμου γείτονα ξεκινάει με πρώτο στάδιο την εύρεση μιας αρχικής λύσης. Η υπάρχουσα αρχική λύση χωρίζεται σε κόμβους σε p γειτονιές. Κάθε γειτονιά έχει ως στόχο την εύρεση της βέλτιστης λύσης, η οποία θα επιφέρει μείωση στο κόστος απόστασης. Σε περίπτωση που η αρχική λύση είναι διαφορετική από την υπάρχουσα, τότε γίνεται ενημέρωση αυτής και ξανά χωρίζονται οι γειτονιές. Συνεχίζει η ίδια διαδικασία, μέχρις ότου η μέθοδος αυτή να σταματήσει όταν δεν υπάρχουν αλλαγές στη λύση [47]. Ακόμα, η γενετική αναζήτηση μια τεχνική δειγματοληψίας που παίρνει τα δεδομένα από τη γενετική πληθυσμού [53] [22] [70]. Επίσης, η Simulated annealing είναι μια μέθοδος που προσομοιώνεται ένα πρόγραμμα ανόπτησης που χρησιμοποιείται στην παραγωγή γυαλιού [72] [13] [61]. Η ευρετική συγκέντρωση ασχολείται με την εκτέλεση πολλαπλών φορών μιας ευρετικής ανταλλαγής, με αποτέλεσμα η λύση να είναι καλύτερη από την τοπικά καλύτερη βέλτιστη [78] [80]. Ακόμα, η Scatter search συνθέτει κανόνες απόφασης και περιορισμούς [86]. Η μέθοδος της αποικία μυρμηγκιών λαμβάνει τις επόμενες αποφάσεις αναπαριστώντας τις αποικίες των μυρμηγκιών και τον τρόπο εύρεσης της τροφής τους [61]. Τα νευρωνικά δίκτυα με τη χρήση μαθηματικού μοντέλου προσομοιώνουν τα βιολογικά νευρωνικά δίκτυα, προσπαθώντας να υπολογίσει το κόστος των νευρώνων - κόμβων [65] [64]. Η μέθοδος αυτή χρησιμοποιεί την εκμάθηση, καθώς δίνεται η δυνατότητα να μαθαίνει και να εξελίξει το βάρος της κάθε σύνθεσης. Τέλος δυο ακόμα μέθοδοι είναι η αποσύνθεση [17] [88] και οι υβριδικές μέθοδοι [75].

Κεφάλαιο 3

Υλοποίηση

3.1 Μοντέλο γραμμικού προγραμματισμού - περίπτωση του λύτη βελτιστοποίησης Gurobi

3.1.1 Γενικά

Η λύση ενός γραμμικού προβλήματος χρειάζεται τη χρήση ενός λύτη, καθώς απαιτείται μνήμη, αποδοτικότητα δικτύου και σταθερότητα. Η χρήση λύτων είναι πολύ αποτελεσματική σε τέτοιου είδους προβλήματα [66]. Κάποιοι γνωστοί λύτες είναι οι εξής: IBM ILOG CPLEX, SCIP, CLP και Gurobi. Για την παρούσα εργασία επιλέχθηκε η χρήση του λύτη Gurobi.

Ο λύτης βελτιστοποίησης Gurobi αποτελεί ένα από τα σημαντικότερα εργαλεία βελτιστοποίησης. Στην κατηγορία αυτή βρίσκονται διάφοροι λύτες που χωρίζονται με βάση την άδεια που διαθέτουν σε δυο κατηγορίες, τα εμπορικά συστήματα λογισμικού και τα συστήματα λογισμικού ανοικτού κώδικα. Ο Gurobi ανήκει στην κατηγορία του εμπορικού λογισμικού [2]. Οι ιδρυτές της Gurobi optimization το 2008 αποφάσισαν να δημιουργήσουν έναν λύτη με σκοπό την επίλυση μαθηματικών προβλημάτων όπως προβλήματα γραμμικού προγραμματισμού και μικτού ακέραιου προγραμματισμού, σε πολύ γρήγορο χρόνο. Το όνομα της εταιρείας σχηματίστηκε από την ένωση των δυο αρχικών γραμμάτων από τα επίθετα τους. Τα ονόματα των τριών ιδρυτών είναι Zonghao Gu, Ed Rothberg και Bob Bixby. Για τη χρήση αυτού του λύτη είναι δυνατή η χρήση των περισσότερων από τις γλώσσες προγραμματισμού. Συγκεκριμένα υποστηρίζει *C*, *C++*, *C#*, *Java*, *Microsoft.NET*, *Python*, *MATLAB* και *R*. Στην παρούσα εργασία χρησιμοποιήθηκε στη γλώσσα *C++* για

αναπτυχθεί το μοντέλο, με τη χρήση του λογισμικού Gurobi Optimizer version 9.0.3.

3.1.2 Τρόπος υλοποίησης

Για την υλοποίηση του μοντέλου γραμμικού προγραμματισμού με τη χρήση του λύτη Gurobi δημιουργήθηκε ένα μοντέλο γραμμικού προγραμματισμού που περιέχει δύο μεταβλητές. Οι δυο αυτές μεταβλητές περιέχουν δυαδικές τιμές. Η πρώτη με το όνομα *open* και μέγεθος όσες οι εγκαταστάσεις του προβλήματος, αναφέρεται στις εγκαταστάσεις. Αναλυτικότερα αν έχει τιμή 0 η εγκατάσταση αυτή είναι κλειστή ενώ αν είναι 1 τότε είναι ανοικτή. Η δεύτερη μεταβλητή αφορά τη μετακίνηση με όνομα *transport* και είναι δισδιάστατος πίνακας με μέγεθος όσες ο αριθμός των εγκαταστάσεων και στις γραμμές και στις στήλες. Στις γραμμές αναφέρονται οι πελάτες και στις στήλες οι εγκαταστάσεις. Η μεταβλητή αυτή παίρνει 1 όταν ο συγκεκριμένος πελάτης εξυπηρετείται από τη συγκεκριμένη εγκατάσταση. Ενώ όταν είναι 0 δεν εξυπηρετείται από αυτή την εγκατάσταση. Ακόμα η μεταβλητή αποτελεί μια σταθερά που περιέχει τα δεδομένα του κόστους που χρειάζεται ένας πελάτης για να πάει στη συγκεκριμένη εγκατάσταση. Μετά προστίθενται οι περιορισμοί που αναφέρθηκαν στη μοντελοποίηση στο κεφάλαιο 2.2. Στον κώδικα για λόγους ευκολίας χρησιμοποιήθηκαν άλλα ονόματα. Έτσι η μεταβλητή x αντιστοιχεί στην *transport* και η μεταβλητή y στην *open*.

Περιορισμός 1: Το άθροισμα των στοιχείων της *open* μεταβλητής να είναι ίσο με το ζητούμενο p του προβλήματος.

Περιορισμός 2: Το άθροισμα κάθε γραμμής της *transport* να είναι ίσο με 1, για να δείχνει ότι ο κάθε πελάτης εξυπηρετείται από μία εγκατάστασή.

Περιορισμός 3: Η διαφορά της *transport* με την *open* να είναι μικρότερη ίση με το 0. Με αυτόν τον περιορισμό αποτρέπεται ένας πελάτης να προτείνεται να εξυπηρετείται από μία εγκατάσταση που δεν είναι ανοικτή.

Έπειτα καλούνται οι εντολές για προσδιορίσουμε ότι το μοντέλο είναι για ελαχιστοποίηση αλλά και τη χρήση μεθόδου Barrier. Τέλος, καλείται η εντολή της βελτιστοποίησης για να μας υπολογίσει την καλύτερη λύση.

Στα Σχήματα 3.1 μέχρι 3.3 παρουσιάζεται η επίλυση του παραδείγματος του *rmed1* προβλήματος από τον λύτη Gurobi. Πιο αναλυτικά, στο Σχήμα 3.1 μπορεί να παρατηρηθεί ο αριθμός των γραμμών και των στηλών του μοντέλου. Στο παρά-

δειγμα αυτό ο αριθμός των εγκαταστάσεων είναι 100 όπως φαίνεται στην αρχή της εικόνας. Ο τύπος για τον υπολογισμό των γραμμών προέρχεται από τους περιορισμούς. Πιο αναλυτικά για τον πρώτο περιορισμό χρειάζεται μόνο μια γραμμή. Για την υλοποίηση του δεύτερου περιορισμού θα χρειαστεί περιορισμούς, ενώ για τον τρίτο περιορισμό $|Y| \times |Z|$. Έτσι ο τύπος είναι ο εξής $1 + |Z| + |Y| \times |Z|$. Οπότε σε αυτό το παράδειγμα είναι $1 + 100 + (100 \times 100) = 10,101$. Ενώ ο τύπος για τον αριθμό των στηλών προέρχεται από τις μεταβλητές του προβλήματος. Ο τύπος για τις μεταβλητές είναι $|Z| + |Y| \times |Z|$. Έτσι στο παράδειγμα οι μεταβλητές είναι $100 + (100 \times 100) = 10,100$. Στις υπόλοιπες εικόνες φαίνονται οι επαναλήψεις που χρειάστηκαν μέχρι να βρεθεί η βέλτιστη λύση, που σε αυτό το παράδειγμα χρειάστηκαν 19 επαναλήψεις. Έπειτα δείχνει το αποτέλεσμα από το Root Relaxation και στο τέλος εμφανίζει το τελικό αποτέλεσμα.

Σχήμα 3.1: Παράδειγμα προβλήματος pmed1 με λύτη Gurobi part1

```
The file is open
100 200 5
nodes=100
connections=200
p=5
Academic license - for non-commercial use only
Gurobi Optimizer version 9.0.3 build v9.0.3rc0 (linux64)
Optimize a model with 10101 rows, 10100 columns and 30100 nonzeros
Model fingerprint: 0x99c5751f
Variable types: 0 continuous, 10100 integer (10100 binary)
Coefficient statistics:
  Matrix range      [1e+00, 1e+00]
  Objective range   [1e+00, 3e+02]
  Bounds range      [1e+00, 1e+00]
  RHS range         [1e+00, 5e+00]
Found heuristic solution: objective 14895.000000
Presolve time: 0.05s
Presolved: 10101 rows, 10100 columns, 30100 nonzeros
Variable types: 0 continuous, 10100 integer (10100 binary)
Root barrier log...

Ordering time: 0.01s

Barrier statistics:
Dense cols : 100
AA' NZ     : 2.010e+04
Factor NZ   : 4.636e+04 (roughly 9 MBytes of memory)
Factor Ops  : 1.701e+06 (less than 1 second per iteration)
Threads    : 1
```

3.1.3 Ψευδοκώδικας της μοντελοποίησης του γραμμικού προβλήματος με χρήση λύτη Gurobu

Στον αλγόριθμο 1 δίνεται ο ψευδοκώδικας της μοντελοποίησης του προβλήματος με χρήση του λύτη.

```

Function gurobi form(int np, dijkstra * dijkst) ;
GRVar * open = 0;
GRVar **transport = 0;
for i = 0; i < nodes; i ++ do
    for j = 0; j < nodes; j ++ do
        | TransCost[i][j] = dijkst.getresulttable[i];
    end
end
open = model.addVars(GRB_BINARY);
for w = 0; w < nodes; w ++ do
    transport[w] = model.addVars;
    for p = 0; p < nodes; p ++ do
        | trnaspport[w][p].set(GRB_DoubleAttr_Obj, TransCost[w][p]);
    end
end
model.set(GRB_MINIMIZE);
for i = 0; i < nodes; i ++ do
    | ptop+ = open[i];
end
model.addConstr(ptop == np);
for i = 0; i < nodes; i ++ do
    for j = 0; j < nodes; j ++ do
        | dtot+ = transport[i][j];
    end
end
model.addConstr(dtot == 1);
for i = 0; i < nodes; i ++ do
    for j = 0; j < nodes; j ++ do
        | model.addConstr(transport[i][j] - open[j] <= 0);
    end
end
model.set(Method, Barrier);
model.optimize;

```

Αλγόριθμος 1: Μοντελοποίηση προβλήματος με χρήση Gurobi

Σχήμα 3.2: Παράδειγμα προβλήματος pmed1 με λύτη Gurobi part2

Iter	Objective		Residual		Compl	Time
	Primal	Dual	Primal	Dual		
0	2.40755056e+05	-5.95189971e+03	1.60e+01	0.00e+00	1.24e+01	0s
1	1.58086406e+04	-1.51615883e+04	3.02e-01	1.90e-08	1.06e+00	0s
2	1.14229447e+04	1.57986331e+03	7.66e-02	2.55e-07	3.12e-01	0s
3	8.42342507e+03	4.50445793e+03	1.48e-02	1.07e-07	1.26e-01	0s
4	7.19947517e+03	5.23136679e+03	6.43e-03	6.92e-08	6.36e-02	0s
5	6.92842797e+03	5.40441333e+03	4.82e-03	5.86e-08	4.93e-02	0s
6	6.80137076e+03	5.44868163e+03	4.14e-03	5.38e-08	4.38e-02	0s
7	6.62169777e+03	5.56821362e+03	3.27e-03	4.65e-08	3.41e-02	0s
8	6.49127405e+03	5.61857013e+03	2.67e-03	4.14e-08	2.83e-02	0s
9	6.30713840e+03	5.68299010e+03	1.81e-03	3.33e-08	2.02e-02	0s
10	6.20057076e+03	5.72375925e+03	1.33e-03	2.92e-08	1.55e-02	0s
11	6.12510077e+03	5.74713046e+03	1.02e-03	2.79e-08	1.23e-02	0s
12	6.03911124e+03	5.77886607e+03	6.97e-04	1.64e-08	8.45e-03	0s
13	5.95313106e+03	5.79803647e+03	3.99e-04	1.04e-08	5.04e-03	0s
14	5.93824021e+03	5.80313792e+03	3.50e-04	7.61e-09	4.39e-03	0s
15	5.87430106e+03	5.81696122e+03	1.47e-04	1.41e-09	1.86e-03	0s
16	5.85171250e+03	5.81746651e+03	8.60e-05	1.22e-09	1.11e-03	0s
17	5.81982730e+03	5.81868326e+03	4.03e-07	2.91e-10	3.78e-05	0s
18	5.81901691e+03	5.81899975e+03	5.00e-08	1.93e-11	5.66e-07	0s
19	5.81900000e+03	5.81900000e+03	1.02e-12	4.94e-11	6.93e-13	0s

Barrier solved model in 19 iterations and 0.13 seconds
Optimal objective 5.81900000e+03

Σχήμα 3.3: Παράδειγμα προβλήματος pmed1 με λύτη Gurobi part3

```

Root relaxation: objective 5.819000e+03, 1234 iterations, 0.08 seconds
Nodes | Current Node | Objective Bounds | Work
Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
* 0 0 0 5819.0000000 5819.00000 0.00% - 0s
Explored 0 nodes (1234 simplex iterations) in 0.15 seconds
Thread count was 12 (of 12 available processors)

Solution count 2: 5819 14895

Optimal solution found (tolerance 1.00e-04)
Best objective 5.819000000000e+03, best bound 5.819000000000e+03, gap 0.0000%

TOTAL COSTS: 5819
Time for gurobi is: 0.188436 seconds

```

3.2 Μέθοδος Branch and Bound

3.2.1 Γενικά

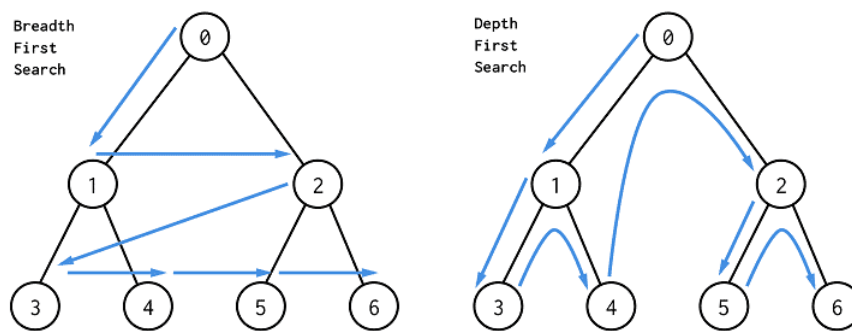
Η «branch and bound» είναι μια μέθοδος διαίρει και βασίλευε. Ειδικότερα, προσπαθεί να χωρίσει το πρόβλημα σε μικρότερα προβλήματα, ώστε να λυθούν πρώτα αυτά τα υποπροβλήματα [25]. Οι Lawler και Wood [60] αναφέρουν κάποια προβλήματα που χρησιμοποιούν τη μέθοδο «branch and bound». Επιγραμματικά αυτά τα προβλήματα είναι ο ακέραιος γραμμικός προγραμματισμός «integer linear programming», ο μη ακέραιος γραμμικός προγραμματισμός «nonlinear programming» και το πρόβλημα της τετραγωνικής ανάθεσης «the quadratic assignment problem».

3.2.2 Τρόπος υλοποίησης

Για την υλοποίηση αυτής της μεθόδου υλοποιήθηκαν τέσσερις τεχνικές. Οι τεχνικές αυτές διαχωρίζουν τον τρόπο ανάπτυξης και δημιουργίας των κλαδιών του δέντρου. Πιο αναλυτικά, η μέθοδος αυτή δημιουργεί συνέχεια κλαδιά, για να δημιουργηθούν όμως τα κλαδιά πρέπει κάθε φορά να επιλεγεί με συγκεκριμένο τρόπο το φύλλο που θα δημιουργηθούν τα επόμενα παιδιά - κλαδιά. Στο Σχήμα 3.4 παρουσιάζεται παράδειγμα των τεχνικών διαχωρισμού και περιορισμών. Η πρώτη τεχνική είναι η κατά πλάτος με τη χρήση τυχαίων αριθμών για τη δημιουργία των κλαδιών. Συγκεκριμένα σε αυτήν την τεχνική ξεκινάμε από τη ρίζα και εξετάζουμε κάθε κόμβο πριν πάμε στο επόμενο ύψος. Η διαδικασία ξεκινάει με την επίλυση του Root Relaxation και ορίζοντας τις τιμές του πάνω ορίου «upper bound» και του κάτω ορίου «lower bound». Το «upper bound» παίρνει την τιμή του μωπικού κόστους και το «lower bound» την τιμή του Root Relaxation. Στη συνέχεια ελέγχεται αν το αποτέλεσμα αποτελείται μόνο από ακέραιες τιμές. Στη περίπτωση που είναι ακέραιες το δέντρο σταματάει. Σε αντίθετη περίπτωση επιλέγεται τυχαία μια τιμή. Αυτή η τιμή αντιπροσωπεύει την εγκατάσταση που θα εξεταστεί. Μετά δημιουργούνται κάποιοι κόμβοι που έχουν σαν τύπο μια δομή δεδομένων που θα επεξηγηθεί παρακάτω. Ειδικότερα ο κάθε κόμβος δημιουργεί δυο παιδιά - κλαδιά. Το αριστερό κλαδί παρουσιάζει την περίπτωση για να ανοίξει αυτή η εγκατάσταση στα σίγουρα, ενώ το δεξί κλαδί την περίπτωση να είναι κλειστή αυτή η εγκατάσταση. Με τη δημιουργία κάθε παιδιού - κλαδιού γίνονται κάποιοι έλεγχοι για την επιτυχία της μεθόδου αυτής. Πιο αναλυτικά, ελέγχεται αν η λύση που βρίσκει είναι ακέραια ή όχι. Σε περίπτωση που είναι ακέραια η διαδικασία γι' αυτόν τον κόμβο σταματάει εκεί. Μετά, συγκρίνεται το κόστος αυτής της λύσης με το «upper bound». Αν είναι μικρότερο το κόστος αυτής της λύσης τότε το «upper bound» ανανεώνεται και παίρνει αυτή σαν νέα τιμή, αλλιώς συνεχίζει στον επόμενο κόμβο χωρίς καμία αλλαγή. Αν η λύση δεν είναι ακέραια τότε ελέγχεται το κόστος αν είναι μεγαλύτερο από το «upper bound». Αν είναι μεγαλύτερο τότε η διαδικασία γι' αυτόν τον κόμβο σταματάει εκεί, ειδάλλως συνεχίζει με την ίδια διαδικασία στον επόμενο κόμβο. Επίσης, αν το «upper bound» γίνει μικρότερο του «lower bound» τότε σταματάει τελείως το δέντρο. Η δεύτερη τεχνική είναι πάλι η κατά πλάτος με τη χρήση του μωπικού αλγορίθμου για την εύρεση της εγκατάστασης που θα

εξεταστεί σε κάθε ύψος. Η υπόλοιπη διαδικασία παραμένει ίδια. Η τρίτη τεχνική είναι η κατά βάθος με τη χρήση τυχειότητας για την επιλογή των εγκαταστάσεων που θα εξεταστούν κάθε φορά. Σε αυτή τη τεχνική οι έλεγχοι για τις τιμές «upper bound» και «lower bound» παραμένουν οι ίδιοι, η μόνη αλλαγή είναι ο τρόπος που εξετάζουμε τους κόμβους. Συγκεκριμένα εξετάζουμε πρώτα όλους τους αριστερούς κόμβους μέχρι να μην μπορεί να δημιουργήσει άλλα παιδιά και συνεχίζει από το σημείο που βρίσκεται να ελέγχει το δεξί αδερφό του και αν μπορεί να συνεχίσει περαιτέρω. Με την ίδια τακτική συνεχίζει μέχρι να ανέβει πάνω στη ρίζα και να ελέγξει με τον ίδιο τρόπο και το δεξί παιδί του. Η τέταρτη τεχνική είναι κατά βάθος με χρήση του μυωπικού αλγορίθμου για την επιλογή της επόμενης εγκατάστασης. Οι υπόλοιπες ενέργειες παραμένουν ίδιες.

Σχήμα 3.4: Παράδειγμα branch and bound BFS - DFS τεχνική [44]

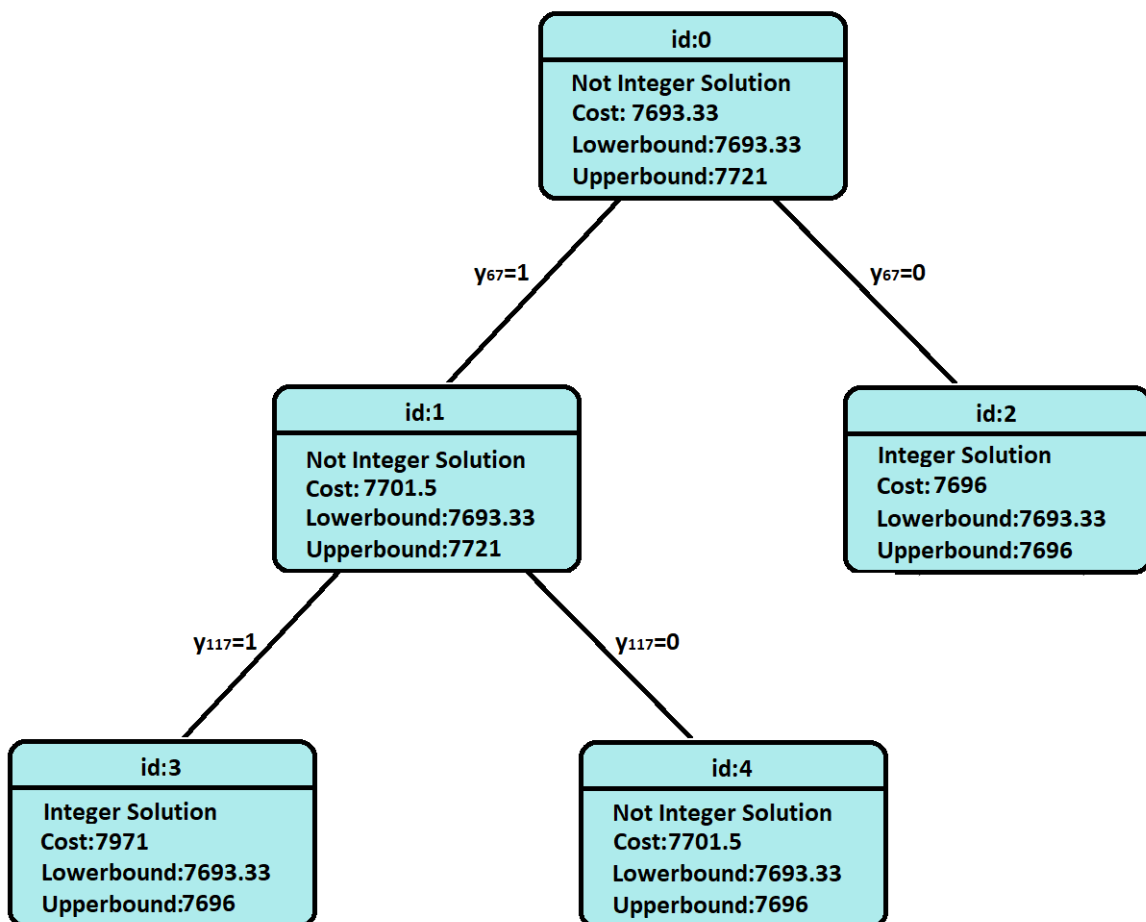


Για να μπορέσουν να επιτευχθούν αυτές οι τεχνικές δημιουργήθηκε μια δομή δεδομένων struct. Αυτή η δομή έχει ως δεδομένα τα βασικά χαρακτηριστικά του κόμβου. Συγκεκριμένα περιέχει έναν δυναμικό πίνακα που αποθηκεύει τις λύσεις για κάθε εγκατάσταση, έναν δυναμικό πίνακα που δείχνει αν κάποια εγκατάσταση πρέπει να έχει προκαθορισμένη τιμή καθώς είναι αυτή που επιλέχθηκε για τον διαχωρισμό των κλαδιών ή μπορεί να είχε εξεταστεί σε προηγούμενο ύψος. Ακόμη, έχει το κόστος της λύσης, τον πατέρα του και δυο δυναμικούς πίνακες με τύπο ίδιο με του struct για να μπορεί να συνδέεται με τα δεξιά και τα αριστερά παιδιά του αλλά ταυτόχρονα να διαθέτουν και αυτά όλα τα υπόλοιπα δεδομένα. Σε αυτή τη δομή υλοποιούνται οι συναρτήσεις για τη δημιουργία των αριστερών και δεξιών παιδιών - κλαδιών αλλά και η εύρεση του κόμβου που περιέχει την ακέραια λύση με το μικρότερο κόστος. Για τη δημιουργία των παιδιών γίνεται χρήση μιας τροποποιημένης μορφής του κώδικα gurobi. Πιο αναλυτικά, δεδομένου ότι έχει δημιουργηθεί ένα πε-

ριβάλλον και ένα μοντέλο στον αλγόριθμο του branch and bound, προστέθηκε στη συνάρτηση ένα ακόμα όρισμα για να παίρνει το υπάρχον μοντέλο και να τροποποιεί αυτό. Με αυτόν τον τρόπο επιτυγχάνεται να αποφευχθεί η δημιουργία πολλαπλών μοντέλων, καθώς η δημιουργία τους χρειάζεται κάποιον χρόνο. Επιπλέον, πριν κληθεί ο τροποποιημένος gurobi γράφεται πρώτα σε ένα αρχείο τα αποτελέσματα που είχε ο πατέρας του κόμβου που δημιουργείται. Έτσι, ο gurobi χρησιμοποιεί αυτή τη λύση ως αρχική λύση για να κερδίσει χρόνο από την εύρεση της λύσης αυτού του κόμβου.

Στο Σχήμα 3.5 ακολουθεί παράδειγμα του προβλήματος pmed11 με τη χρήση του Branch and Bound κατά πλάτος και με συνδυασμό της μυωπικής τεχνικής.

Σχήμα 3.5: Παράδειγμα branch and bound BFS-myopic



Στο $id : 0$ γίνεται το Root Relaxation όπου ορίζεται το lower bound και το upper bound. Συγκεκριμένα το lower bound παίρνει την τιμή του Root Relaxation, ενώ το upper bound παίρνει το κόστος από τον μυωπικό. Η λύση που βρέθηκε δεν

είναι ακέραια σε όλες τις εγκαταστάσεις. Έτσι, ο μωπικός προτείνει την καλύτερη εγκατάσταση για τη δημιουργία του δέντρου. Σε αυτό το παράδειγμα ο μωπικός πρότεινε την εγκατάσταση 67. Με αυτή τη πληροφορία το δέντρο δημιουργεί δυο κλαδιά. Το αριστερό να ανοίγει την εγκατάσταση ($y_{67} = 1$) και το δεξί να την κλείνει ($y_{67} = 0$). Στη δημιουργία του αριστερού κλαδιού $id : 1$ πάλι δε βρέθηκε ακέραια λύση και το κόστος του είναι μικρότερο από το upper bound, οπότε όταν είναι να εξεταστεί αυτό το κλαδί θα συνεχίσει με την ίδια διαδικασία. Στο δεξί κλαδί $id : 2$ βρέθηκε ακέραια λύση. Μετά, έγινε έλεγχος αν το κόστος του είναι μικρότερο από το upper bound. Επειδή, είναι μικρότερο άλλαξε το upper bound παίρνοντας αυτή τη τιμή. Επίσης, επειδή η λύση ήταν ακέραια η διαδικασία σε αυτό το κλαδί σταματάει εδώ. Στη συνέχεια ακολούθησε η δημιουργία νέων κλαδιών από το $id : 1$. Εδώ ο μωπικός αλγόριθμος πρότεινε την εγκατάσταση 117. Δημιουργήθηκε πρώτα το αριστερό κλαδί $id : 3$ που βρήκε ακέραια λύση. Το κόστος του όμως ήταν μεγαλύτερο από το upper bound γι' αυτό και δεν άλλαξε. Στο δεξί κλαδί $id : 4$ δε βρέθηκε ακέραια λύση, όμως το κόστος του είναι μεγαλύτερο από το upper bound και έτσι σταματάει εδώ το δέντρο. Συνεπώς την καλύτερη λύση την έχει το $id : 2$ με κόστος 7696.

3.2.3 Ψευδοκώδικας του Branch and Bound

Στον αλγόριθμο 2 δίνεται ο ψευδοκώδικας του αλγορίθμου Branch and Bound.

Η ίδια μεθοδολογία είναι και στις τέσσερις τεχνικές με κάποιες μικρές διαφορές. Συγκεκριμένα για τις Branch and Bound DFS τεχνικές η διαφορά είναι στο τρόπο εισαγωγής των Branch στη λίστα Branches. Πιο αναλυτικά πρώτα θα εισάγεται μπροστά το δεξί κλαδί και μετά το αριστερό κλαδί (`push.front(rightbranch)`, `push.front(leftbranch)`). Με αυτό το τρόπο πάντα παίρνουμε πρώτα το αριστερό κλαδί και όταν τελειώσουν τα αριστερά συνεχίζει με το δεξί. Ενώ στις τεχνικές με τη χρήση του μωπικού η αλλαγή είναι στον τρόπο εύρεσης του `selectnumber`. Ειδικότερα τρέχει τον μωπικό αλγόριθμο για ένα μόνο p .

```

Function branchandboundmethod(int **graph, int np, dijkstra *dijkst) ;
RootRelaxation //gurobi with continuous values;
listBranchbranches;
upperbound = myopiccost(p);
lowerbound = RootRelaxationcost;
if RootRelaxationanswer == 1&& upperbound! = lowerbound then
    selectnumber = (rand%(up - low + 1)) + low;
    branches.push(createleftbranch(branch, selectnumber, branch- >
        getid(), globalmax, env, open, &model));
    //check if the answer is integer. ;
    if currentbranchcost < upperbound then
        | //if yes upperbound = branch- > right- > getcost;
    if br- > right- > getcost >= upperbound then
        | //if no //stop the branch there
    branches.push(createrightbranch(branch, selectnumber, branch- >
        getid(), globalmax, env, open, &model));
    //check the same thing;
while != branches.empty()&& lowerbound < upperbound do
    if answergrombranch == 1 then
        | //if answergrombranch=1 it means it is integer;
        branches.pop_front() selectnumber = (rand%(up - low + 1)) + low;
        branches.push(createleftbranch(branch, selectnumber, branch- >
            getid(), globalmax, env, open, &model));
        //check the same constraint;
        branches.push(createrightbranch(branch, selectnumber, branch- >
            getid(), globalmax, env, open, &model));
        //check the same constraint;
    else
        | branches.pop_front()
    end
end
findminbranch()

```

Αλγόριθμος 2: Αλγόριθμος Branch and Bound

3.3 Μέθοδος Dijkstra

3.3.1 Γενικά

Ο αλγόριθμος Dijkstra [23] πήρε το όνομα του από τον Έντσκερ Ντάικστρα το 1956. Ο αλγόριθμός αυτός ασχολείται με την εύρεση συντομότερων μονοπατιών σε γράφους που μπορεί να είναι κατευθυνόμενοι ή μη. Συγκεκριμένα ξεκινάει από έναν κόμβο του γράφου και ελέγχει τους γείτονες του με σκοπό να συνεχίσει στον κόμβο που έχει το λιγότερο κόστος. Στη πορεία αν κάποιος από τους γείτονες ενώνεται και με άλλους κόμβους κρατιέται το ελάχιστο κόστος. Ακόμα, όταν κάποιος κόμβος έχει ανοίξει δεν ξανά ελέγχεται. Επίσης, μετά τον Edsger W. Dijkstra ασχολήθηκαν κι άλλοι με την εύρεση συντομότερων μονοπατιών όπως είναι ο Richard Bellman and Lester Ford [91] που επινόησαν τον Bellman-Ford το 1956 και 1958 και ο Floyd-Warshall το 1962 [91]

3.3.2 Τρόπος υλοποίησης

Ο Dijkstra είναι ένας αλγόριθμος που βοηθάει στην εύρεση του κοντινότερου μονοπατιού σε ένα πρόβλημα. Συγκεκριμένα ξεκινώντας από ένα συγκεκριμένο κόμβο ελέγχει το κόστος από τους γειτονικούς κόμβους, υπολογίζοντας το κόστος τους. Μετά, συνεχίζει στον κόμβο που έχει το μικρότερο κόστος και ελέγχει τους δικούς του γειτονικούς κόμβους, προσθέτοντας το κόστος που είχε αυτός. Σε περίπτωση που ένας κόμβος ανήκει στους γείτονες σε περισσότερους από έναν κόμβο τότε γίνεται έλεγχος μεταξύ των κοστών και κρατείται το ελάχιστο. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να ελεγχθούν όλοι οι κόμβοι.

3.3.3 Ψευδοκώδικας του Dijkstra

Στον αλγόριθμο 3 δίνεται ο ψευδοκώδικας του αλγορίθμου Dijkstra.

3.4 Μυωπικός αλγόριθμος

3.4.1 Γενικά

Όπως αναφέρει και ο Whitaker [93] αρκετοί ασχολήθηκαν με τους αλγόριθμους βελτιστοποίησης. Συμπεραίνοντας ότι στα πολύ μεγάλα προβλήματα δεν είναι επι-

```

Function dijkstraform(int start) ;
distance[nodes] ← MAX;
boolset[nodes] ← FALSE;
for j = 0; j < nodes - 1; j ++ do
    u = findMinDistance(distance, boolset, nodes);
    boolset[u] ← TRUE;
    for each n of nodes do
        if nodes = FALSE && cost between(u, n) ≠ 0 && distance[u] ≠
            MAX && distance[u] + graph[u][n] < distance[n] then
            | distance[n] = distance[u] + graph[u][n];
    end
end

```

Αλγόριθμος 3: Αλγόριθμος Dijkstra

κτή η εύρεση της βέλτιστης λύσης. Έτσι, άρχισαν να μελετούν τους ευρετικούς αλγορίθμους. Ο Kuehn and Hamburger το 1963 [58] αναφέρουν κάποια «heuristics» για τη χρήση τους σε έναν άπληστο αλγόριθμο. Αυτός ονομάζεται μυωπικός. Η χρήση των ευρετικών προσπαθεί να βελτιώσει την άπληστη λύση και έχουν χρησιμοποιηθεί σε παραδείγματα των Teitz and Bart [89], Rushton and Kohler [81], Hodgson [50], Rosing et al. [77], κ.ά. Ο Whitaker [93] και οι Daskin και Maass [20] εξηγούν πως η εκτέλεση αυτού του αλγορίθμου γίνεται μέσω αναδρομής παρουσιάζοντας οι Daskin και Maass έναν ψευδοκώδικα. Επεξηγηματικά, γίνονται πολλές επαναλήψεις μέχρι να βρεθούν όλες οι p εγκαταστάσεις. Σε κάθε επανάληψη προσπαθεί να βρει τον κόμβο που θα επιφέρει τη μεγαλύτερη μείωση κόστους στο πρόβλημα.

3.4.2 Τρόπος υλοποίησης

Ο "myopic"/μυωπικός αλγόριθμος είναι όπως προαναφέρθηκε ένας άπληστος ευρετικός αλγόριθμος. Ειδικότερα, ο αλγόριθμος έχει σαν δεδομένα τον πίνακα με τις ενώσεις των κόμβων και τα αντίστοιχα κόστη. Δημιουργεί ένα άδειο σύνολο (πίνακα) για να μπορέσει να αποθηκεύσει p αριθμό κόμβων που θα προστεθούν στη λύση του προβλήματος. Πιο αναλυτικά, έχουμε ένα σύνολο K από N κόμβους. Αυτοί οι κόμβοι είναι πιθανές θέσεις για να ανοίξει η επιχείρηση. Μετά, έχουμε ένα σύνολο M , που θα αποθηκεύσουμε τα αποτελέσματα του αλγορίθμου, που αποτελείται από p θέσεις καθώς είναι ο ζητούμενος αριθμός μονάδων για την εγκατάσταση. Η διαδικασία που ακολουθεί είναι η αναζήτηση σε αυτούς τους N κόμβους ποιος θα φέρει το λιγότερο κόστος. Έτσι, εφαρμόζοντας και τον αλγόριθμο Dijkstra βρίσκουμε

τον κόμβο με το λιγότερο κόστος. Μετά, έχοντας υπόψη ότι ο προηγούμενος κόμβος συμπεριλαμβάνεται στη λύση, προσπαθούμε να βρούμε τον επόμενο κόμβο που με συνδυασμό με τον υπάρχοντα θα μειώσει επιπλέον το κόστος. Αυτή η διαδικασία γίνεται για $p-1$ φορές για να μπορέσει το M σύνολο να γεμίσει όλες τις θέσεις του και έτσι να έχουν βρεθεί όλες οι ζητούμενες μονάδες.

3.4.3 Ψευδοκώδικας του μυωπικού αλγορίθμου

Στον αλγόριθμο 4 δίνεται ο ψευδοκώδικας του μυωπικού αλγορίθμου.

```

Function myopicform(int p) ;
myopicresult[p];
cost[nodes] ← 0;
fcost[nodes];
for j = 1; j < p; j ++ do
    for i = 0; i < nodes; i ++ do
        if dijkstracost[i] ≠ MAX then
            for n = 0; n < nodes; n ++ do
                if fcost[n] < dijkstracost[i] then
                    cost[i] = fcost[i] + cost[i];
                else
                    if fcost[n] ≥ dijkstracost[i] then
                        cost[i] = dijkstra[i] + cost[i];
                    end
                end
            end
        end
    end
    for n = 0; n < nodes; n ++ do
        if fcost[n] > dijkstracost[n] then
            fcost[n] = dijkstracost[n]
        end
    end
end

```

Αλγόριθμος 4: Αλγόριθμος Myopic

3.5 Αλγόριθμος Neighborhood Search

3.5.1 Γενικά

Πρόσφατα προτάθηκε η τεχνική επίλυσης Neighborhood Search για συνδυαστικά προβλήματα [68] [47]. Η ιδέα αυτής της μεθόδου είναι να χωρίζει το πρόβλημα σε πιο μικρά κομμάτι, τις γειτονιές. Σε αυτές τις γειτονιές εφαρμόζονται μέθοδοι

τοπικής αναζήτησης. Με σκοπό να βρεθούν οι βέλτιστες λύσεις, ώστε να μειωθεί το κόστος [73]. Για την εύρεση της καλύτερης λύσης μπορούν να χρησιμοποιηθούν διαφορετικές μέθοδοι, όπως branch and bound ή ευρετικοί αλγόριθμοι[85].

3.5.2 Τρόπος υλοποίησης

Η μέθοδος Neighborhood Search είναι μια μεθευρετική μέθοδος. Ειδικότερα, η Neighborhood μέθοδος χρησιμοποιεί τοπική αναζήτηση. Πιο αναλυτικά, χωρίζει το πρόβλημα σε p γειτονίες, έχοντας p κεντρικούς κόμβους. Στην αρχή οι p κεντρικοί κόμβοι επιλέγονται με τυχαίο τρόπο. Έπειτα, διαχωρίζει τους κόμβους σε ποιές γειτονίες ανήκουν. Αυτό επιτυγχάνεται ελέγχοντας το κόστος που έχουν από τους κεντρικούς κόμβους. Μετά, από αυτή τη διαδικασία, για κάθε γειτονία προσπαθεί να βρει ποια είναι η καλύτερη λύση. Η μέθοδος που χρησιμοποιήθηκε για την εύρεση της καλύτερης λύση είναι με τη χρήση του μυωπικού αλγορίθμου. Έτσι, ο ευρετικός αλγόριθμος επιστρέφει την καλύτερη λύση και συγκρίνεται με την υπάρχουσα. Σε περίπτωση που είναι διαφορετικοί κόμβοι τότε η διαδικασία πρέπει να ξανά γίνει από τη αρχή, έχοντας πλέον αυτές τις βέλτιστες λύσεις ως κεντρικούς κόμβους. Η διαδικασία σταματάει όταν δεν υπάρξουν περαιτέρω αλλαγές στις βέλτιστες λύσεις.

3.5.3 Ψευδοκώδικας του neighborhood search αλγορίθμου

Στον αλγόριθμο 5 δίνεται ο ψευδοκώδικας του Neighborhood Search

3.6 Αλγόριθμος Tabu Search

3.6.1 Γενικά

Ο Tabu Search είναι μία μεθευρετική μέθοδος που χρησιμοποιείται για την επίλυση βέλτιστων προβλημάτων [38] με την ικανότητα να ξεφεύγει από τα τοπικά βέλτιστα. Από τα τέλη του 1960 και στις αρχές της δεκαετίας του 1970 υπάρχουν αναφορές σε αυτή τη μέθοδο [39] στον τομέα της τεχνητής νοημοσύνης από τον Nilsson [36]. Τελικά όμως ο πρώτος που πρότεινε αυτή τη μέθοδο ήταν ο Glover [36] [37]. Ενώ και ο Hansen [46] ασχολήθηκε και αυτός με την περιγραφή της βασικής ιδέας της μεθόδου. Ο Glover εφάρμοσε αυτή τη μέθοδο ως μεθευρετική και όχι σαν ευρετική. Με αυτή την προσέγγιση ο Glover ήθελε να βοηθήσει στη διαφυγή

```

Function neighborhoodmethod(int **graph, dijkstra *dijkst) ;
s[p];
for i = 0; i < p; i ++ do
| s[i] = random
end
for k = 0; k < maxTries; k ++ do
| if haschanged == 1 then
| | haschanged = 0;
| | for i = 0; i < p; i ++ do
| | | findNeighborhood(i);
| | | dijkstra hdi[nodes];
| | | for n = 0; n < nodes; n ++ do
| | | | hdi[n].dijkstraform(n);
| | | end
| | | mp.myopicform(1);
| | | if isNeighbor && isDifferent then
| | | | s[i] = mp.getmyopicresult();
| | | | haschanged = 1;
| | | end
| end
end

```

Αλγόριθμος 5: Αλγόριθμος Neighborhood Search

από τα τοπικά βέλτιστα, όπως γίνεται στην hill climbing. Έτσι, για να αποφευχθούν όταν βρίσκεται σε τοπικό βέλτιστο δίνεται η δυνατότητα να γίνει μια κίνηση που δεν είναι τόσο ιδανική, δηλαδή δεν επιφέρει βελτίωση. Ακόμα, χρησιμοποιήθηκε μνήμη, με αποτέλεσμα να αποφεύγονται ίδιες πρόσφατες κινήσεις, για την αποφυγή κύκλων.

3.6.2 Τρόπος υλοποίησης

Ο Tabu Search είναι μια μεθευρετική μέθοδος. Για την υλοποίηση αυτής της μεθόδου δημιουργήσαμε μια λίστα, με μέγεθος που δίνεται σαν όρισμα της συνάρτησης, για να κρατάει τις κινήσεις που έχει κάνει. Έπειτα, χρειαζόμαστε τρεις πίνακες, με μέγεθος όσο ο αριθμός των απαιτούμενων ανοικτών εγκαταστάσεων. Ο πρώτος πίνακας κρατάει την αρχική λύση (s). Ο δεύτερος έχει την πρόσφατη βέλτιστη λύση ($current_best$) και ο τρίτος τη λύση που εξετάζει εκείνη τη στιγμή (new_s). Συγκεκριμένα για την αντιστοίχιση της πρώτης λύσης χρησιμοποιήθηκε ο μωωπικός αλγόριθμος. Έπειτα, δημιουργήθηκαν γειτονιές, που έχουν ως κέντρο τις λύσεις από τον μωωπικό. Επιλέγοντας μια γειτονία στην τύχη, βρίσκουμε ποιος είναι ο καλύτερος κόμβος εγκατάστασης, που θα έχει σαν αποτέλεσμα μείωση στο

κόστος. Εφαρμόζουμε αυτή την αλλαγή στον *new_s* και υπολογίζουμε το συνολικό κόστος αυτής της λύσης. Μετά, ακολουθεί έλεγχος για να δούμε αν η κίνηση υπάρχει μέσα στη λίστα. Πιο αναλυτικά, αν η κίνηση δεν υπάρχει μέσα στη λίστα ή αν υπάρχει αλλά το κόστος αυτής της λύσης είναι καλύτερο από το υπάρχον, τότε η κίνηση γίνεται. Αμέσως μετά γίνεται η ενημέρωση της λίστα και του χρόνου που έχουν οι υπάρχουσες ήδη κινήσεις μέσα σε αυτή. Έτσι, όταν μια κίνηση φτάσει στο προκαθορισμένο χρόνο μέσα στη λίστα αφαιρείται από αυτή. Η διαδικασία αυτή γίνεται μέχρι να τελειώσουν οι απαιτούμενες επαναλήψεις.

3.6.3 Ψευδοκώδικας του tabu search αλγορίθμου

Στον αλγόριθμο 6 δίνεται ο ψευδοκώδικας του Tabu Search

3.7 Αλγόριθμος Drop Approach

3.7.1 Γενικά

Η Drop approach ή αλλιώς Stingy είναι μια ευρετική μέθοδος. Την ανέπτυξαν οι Feldman et al. [29]. Αυτή η μέθοδος σχεδιάστηκε για την επίλυση προβλημάτων χωροθέτησης. Η διαδικασία εύρεσης λύσης για τέτοιου είδους προβλήματα είναι διαφορετική με ότι ασχοληθήκαμε μέχρι τώρα [92]. Πιο αναλυτικά, ξεκινάει θεωρώντας όλες τις πιθανές εγκαταστάσεις ανοικτές. Έτσι, το κόστος είναι μηδέν. Έπειτα, προσπαθεί κλείνοντας μία μία τις εγκαταστάσεις να βρει την εγκατάσταση που αν κλείσει θα έχει τη μικρότερη αύξηση κόστους. Μετά, επαναλαμβάνεται αυτή η διαδικασία μέχρι να μείνουν p εγκαταστάσεις στη λύση που θα αποτελούν τη λύση του προβλήματος.

3.7.2 Τρόπος υλοποίησης

Για την υλοποίηση αυτού του αλγορίθμου χρησιμοποιήσαμε κάποιους πίνακες. Συγκεκριμένα στον *tempolution* αποθηκεύσαμε όλους τους κόμβους, που μπορούν να είναι χώρος εγκατάστασης. Στην αρχή τους θεωρούμε όλους ανοικτούς, οπότε το κόστος θα είναι μηδέν. Έπειτα, η διαδικασία είναι να κλείνει μια εγκατάσταση και να υπολογίζει το κόστος που θα επιφέρει αυτή η αλλαγή. Ο υπολογισμός του κόστους γίνεται με τη χρήση μια συνάρτησης *calculatecost*. Σε αυτή τη συνάρτηση

```

Function tabumethod(int **graph, int maxIterations, int tabuLength, dijkstra *
  dijkst ;
s[p];
mp.myopicform(p);
s = mp.getmyopicspecificresult(i);
findNeighbors selectx dijkstra * hdi = newdijkstra[nodes];
if x = neighbors then
  | if min > hdi[i].getcost() then
  | | min = hdi[i].getcost();
  | | min = i;
if x is not in the list OR x is in the list && currentcost < best then
  | current_best[x] = mint;
  | if currentcost < bestsolution then
  | | s = current_best;
  | | bestsolution = currentcost;
  | if best > currentcost then
  | | best = currentcost;
  | if x is not in the list then
  | | if tabulist.size() < tabulenght then
  | | | tabulist.push_back(mint);
  | | else
  | | | tabulist.pop_front();
  | | | tabulist_back(mint);
  | | end
  | if tabulist.pop_front() == tabulimit then
  | | tabulist.pop_front();
  | | tabulistlimit ++;
else
  | tabulistlimit ++;
end
calculatecost();

```

Αλγόριθμος 6: Αλγόριθμος Tabu Search

χωρίζονται οι εγκαταστάσεις σε γειτονίες με βάση σε ποια ανοικτή εγκατάσταση είναι πιο κοντά και προστίθεται το κόστος από όλες τις γειτονίες για να βρεθεί το ολικό κόστος. Αφού, υπολογιστεί το ολικό κόστος επιλέγουμε την εγκατάσταση που με το κλείσιμο της είχε τη μικρότερη αύξηση στο κόστος. Αυτή την εγκατάσταση την αποθηκεύουμε στον δεύτερο πίνακα *dropsolutions* και την αφαιρούμε από τη λύση, ώστε την επόμενη φορά να είναι ήδη κλειστές. Αυτή η διαδικασία επαναλαμβάνεται για $nodes - p$ φορές μέχρι να γεμίσει ο πίνακας *dropsolutions*. Μετά, το πέρας των απαιτούμενων επαναλήψεων για την εύρεση της τελικής λύσης αφαιρούμε από το σύνολο όλων των κόμβων εγκατάστασης τους κόμβους που βρίσκονται στον *dropsolutions* πίνακα.

3.7.3 Ψευδοκώδικας του Drop αλγορίθμου

Στον αλγόριθμο 7 δίνεται ο ψευδοκώδικας του Drop Approach.

```

Function dropmethod(int graph[N][N], dijkstra dijkst[N]) ;
tempolution[nodes];
dropsolutions[nodes - p];
for i = 0; i < nodes - p; i ++ do
    for j = 0; j < nodes; j ++ do
        tempolution[j] = MAX //close the facility
        calculatecost(graph, tempolution, dijkst);
        tempolution[j] = j //open the facility
    end
    min = MAX;
    mint = 0;
    for m = 0; m < nodes; m ++ do
        if possiblecost[m] < min then
            min = possiblecost[m];
            mint = m;
        end
    dropsolution[i] = mint;
end
showdropresult();

```

Αλγόριθμος 7: Αλγόριθμος Drop Search

3.8 Αλγόριθμος Subdrop Approach

3.8.1 Γενικά

Η Subdrop μέθοδος αποτελεί μια εναλλαγή της drop approach. Συγκεκριμένα οι Salhi και Atkinson [84], έχοντας σαν κεντρική ιδέα την drop approach έκαναν μια διαφορετική προσέγγιση ως προς την επιλογή των κόμβων εγκατάστασης που εξετάζουν. Πιο αναλυτικά, προσθέτουν την τυχαιότητα μέσα στη μέθοδο, καθώς δεν επεξεργάζονται ταυτόχρονα όλες τους πιθανούς κόμβους. Σε αντίθεση με την drop διαλέγουν τυχαία m εγκαταστάσεις και εφαρμόζουν πάνω σε αυτούς την drop τεχνική παίρνοντας ένα αποτέλεσμα. Η διαδικασία αυτή γίνεται για k φορές. Στο τέλος βρίσκουμε ανάμεσα σε αυτά τα k αποτελέσματα πιο είναι το καλύτερο για αυτό το πρόβλημα.

3.8.2 Τρόπος υλοποίησης

Για την υλοποίηση αυτής της μεθόδου δημιουργήσαμε μια λίστα, που θα αποθηκεύει τις εγκαταστάσεις που επιλέγονται. Συγκεκριμένα, η συνάρτηση δέχεται σαν όρισμα τον αριθμό m για το πόσες εγκαταστάσεις θα ελέγχει κάθε φορά, και τον αριθμό k για το πόσες φορές να γίνει αυτή η διαδικασία. Διαλέγει τυχαία m εγκαταστάσεις και τα προσθέτει στη λίστα. Μετά, τρέχει την drop approach μέθοδο ελέγχοντας ότι οι κόμβοι που επιλέγονται για λύση βρίσκονται μέσα στη λίστα, άρα είναι ανοιχτοί παίρνοντας το αποτέλεσμα. Έπειτα, συγκρίνει την τωρινή λύση με τη βέλτιστη. Σε περίπτωση που η τωρινή είναι καλύτερη τότε αλλάζει η βέλτιστη λύση με την τωρινή. Σε αντίθεση περίπτωση απλά συνεχίζει στην επόμενη επανάληψη ή αν έχει έρθει στο τέλος των επαναλήψεων εμφανίζει το αποτέλεσμα.

3.8.3 Ψευδοκώδικας του subdrop αλγορίθμου

Στον αλγόριθμο 8 δίνεται ο ψευδοκώδικας του Subdrop Approach.

```

Function subdropmethod(int **graph, dijkstra *dijkst, int m, int k) ;
tempsolution[nodes];
tempsolutioncost = MAX;
listpossiblefacilities;
for ik = 0; ik < k; ik ++ do
    tempsolution = findmrandomfacilities();
    possiblefacilities.push(tempsolution);
    dropalgorithm;
    //with the list we check if the drop solution is in the m selected facilities;
    if tempsolutioncost < solutioncost then
        for i = 0; i < k; i ++ do
            | s[i] = tempsolution;
        end
    possiblefacilities.clear();
end
showdropresult();

```

Αλγόριθμος 8: Αλγόριθμος Subdrop Approach

3.9 Αλγόριθμος Exchange Approach

3.9.1 Γενικά

Ο Exchange είναι ένας ευρετικός αλγόριθμος που παρέχει πολύ καλά αποτελέσματα [45]. Οι πρώτοι που πρότειναν αυτών των αλγόριθμο είναι οι Teitz and Bar το 1968 [89]. Μετά, από αυτούς αρκετοί έκαναν διάφορες τροποποιήσεις στον αρχικό αλγόριθμο για να επιφέρουν πιο γρήγορα αποτελέσματα όπως οι Goodwin's and Noronha's [43] [93]. Σκοπός αυτού του αλγορίθμου είναι να αλλάζουν οι εγκαταστάσεις που βρίσκονται στη λύση με εγκαταστάσεις που δεν εμπεριέχονται στη λύση.

3.9.2 Τρόπος υλοποίησης

Η υλοποίηση αυτού του αλγορίθμου απαιτεί αρχικά μια πρώτη λύση, που πάνω σε αυτή θα γίνουν οι τροποποιήσεις. Πιο συγκεκριμένα, χωρίζουμε τις εγκαταστάσεις σε δύο πίνακες. Ο ένας θα περιέχει αυτά που αποτελούν λύση του προβλήματος και ο άλλος θα περιέχουν αυτά που περίσσεψαν. Έπειτα, η διαδικασία είναι να πάρει τη πρώτη εγκατάσταση από τον πίνακα που δεν αποτελούν λύση και να δοκιμάσουν να την αλλάξουν με κάθε εγκατάσταση που βρίσκεται στη λύση. Αποθκεύοντας τα νέα κόστη που υπολογίζονται από τη κάθε αλλαγή, στο τέλος αυτής

της διαδικασίας ελέγχεται σε ποία θέση έχει ως αποτέλεσμα το μικρότερο κόστος. Αν αυτό το νέο κόστος είναι μικρότερο από την αρχική λύση τότε ενημερώνουμε τον πίνακα που έχει τη λύση του προβλήματος. Η διαδικασία αυτή επαναλαμβάνεται για $nodes - p$ φορές. Όστε να δοκιμαστούν όλες οι μη επιλεγμένες λύσεις. Ο αλγόριθμος αυτός τελειώνει όταν πραγματοποιηθεί ένας κύκλος και δεν υπάρξει καμία αλλαγή στη λύση.

3.9.3 Ψευδοκώδικας του exchange αλγορίθμου

Στον αλγόριθμο 9 δίνεται ο ψευδοκώδικας του Exchange.

```

Function xchangemethod(int **graph, dijkstra *dijkst) ;
s[p];
sub[nodes - p];
mp.myopicform(p);
s = mp.getmyopicspecificresult(i);
sub = is not in the s;
bestsolutioncost = mp.getmyopiccost();
while change==1 do
    for i = 0; i < nodes - p; i ++ do
        for j = 0; j < p; j ++ do
            currentbest[j] = sub[i];
            currentcost[j] = calculatecost(graph, currentbest, dijkst);
            currentbest[j] = s[j];
        end
        for j = 0; j < p; j ++ do
            if currentcost[j] < min then
                min = currentcost[j];
                mint = j;
            end
            if min < bestsolution then
                help = sub[i];
                sub[i] = s[mint];
                s[mint] = help;
                change = 1;
                bestsolution = mint;
            end
        end
    end
    min = globalmax;
    calculatecost();

```

Αλγόριθμος 9: Αλγόριθμος Exchange Approach

Κεφάλαιο 4

Υπολογιστική μελέτη

4.1 Διαδικασία πειράματος

Για την υλοποίηση του πειράματος χρησιμοποιήθηκε η βιβλιοθήκη ORLIB (OR-Library). Η βιβλιοθήκη αυτή παρουσιάστηκε από τον Beasley[5][6] και περιέχει πολλά δεδομένα για διάφορα προβλήματα. Για το πείραμα αυτό χρησιμοποιήθηκαν 40 προβλήματα (pmed1, pmed2 έως pmed40) που αφορούν τα προβλήματα p μέσου μη περιορισμένης χωρητικότητας. Τα προβλήματα αυτά είναι όλα σε μορφή txt (text file) και αναπαριστούν έναν γράφο. Ειδικότερα, στη πρώτη σειρά του αρχείου αναγράφονται τρεις αριθμοί. Ο πρώτος δείχνει τον αριθμό των κόμβων του προβλήματος, ο δεύτερος παρουσιάζει τον αριθμό των ακμών που συνδέουν τους κόμβους μεταξύ τους και ο τρίτος είναι ο απαιτούμενος αριθμός p για χωροθέτηση. Στις ακόλουθες γραμμές του αρχείου παρουσιάζονται όλες οι ακμές του γράφου, αναγράφοντας πρώτα τους δύο κόμβους που αφορά η ακμή και στο τέλος το κόστος αυτής. Για την υλοποίηση του προβλήματος υποθέτουν πως όλοι οι κόμβοι είναι και πελάτες και εγκαταστάσεις. Τα προβλήματα παρουσιάζουν γράφους από 100 μέχρι 900 κόμβους. Συγκεκριμένα για να μπορέσουν να συγκρίνουν στον ίδιο αριθμό κόμβων διάφορους αριθμούς p , τα προβλήματα χωρίζονται σε πεντάδες. Δηλαδή ανά πέντε προβλήματα ο αριθμός των κόμβων αυξάνεται κατά 100 κόμβους. Στη παρούσα διπλωματική εργασία η γλώσσα που χρησιμοποιήθηκε για την υλοποίηση των αλγορίθμων είναι η C++ στο περιβάλλον Visual Studio Code έκδοση 1.78.2. Η διαδικασία εκτέλεσης των αλγορίθμων έγινε σε έναν υπολογιστή με Intel(R) Core(TM) i7-8700 CPU στα 3.20GHz και 16GB μνήμη σε λειτουργικό σύστημα CentOS.

Στη διαδικασία εκτέλεσης των αλγορίθμων για όλα τα προβλήματα, έτρεξαν όλα

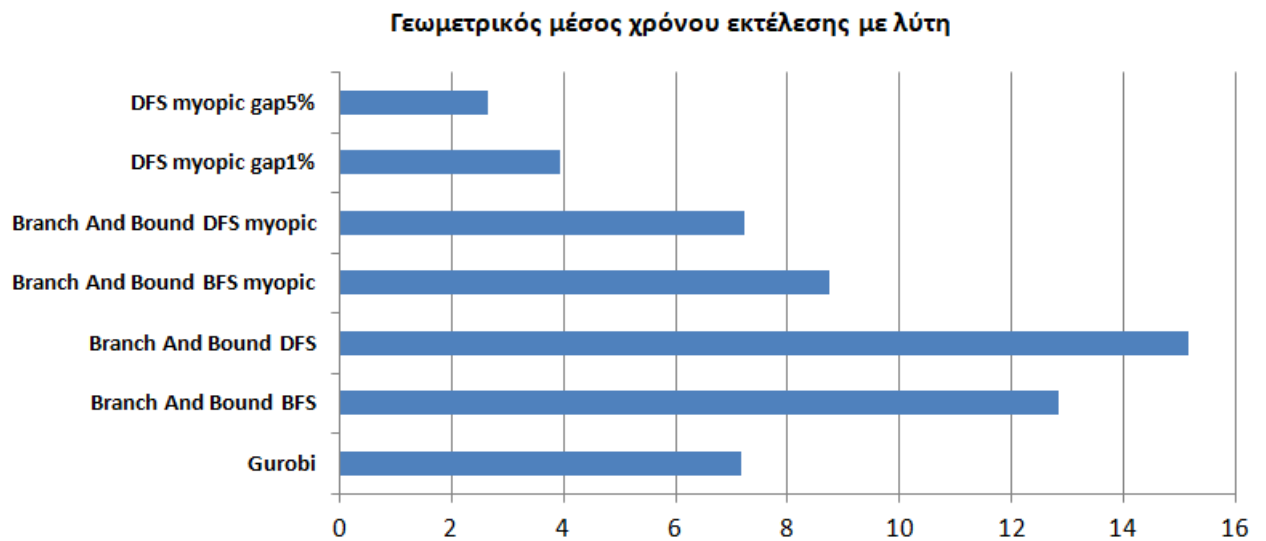
τα προβλήματα για έναν αλγόριθμο και μετά συνεχίστηκε η ίδια διαδικασία και για τους υπόλοιπους αλγόριθμους. Για την εκτέλεση κάποιων αλγόριθμων εφαρμόστηκε ένας περιορισμός χρόνου. Συγκεκριμένα, αν ο χρόνος εκτέλεσης ξεπερνούσε τη μια ώρα, η διαδικασία σταματάει και εμφανίζει τη λύση που είχε βρει ως τότε. Αυτή η προσθήκη του περιορισμού έγινε μετά το τρέξιμο των branch and bound τεχνικών. Εκεί, παρατηρήθηκε ότι χρειάστηκε σε κάποια προβλήματα από δυο με τρεις μέρες για να βρει λύση. Έτσι, αποφασίστηκε να σταματάει ο αλγόριθμος μετά το πέρας της μίας ώρας. Επίσης, στις τεχνικές του branch and bound και στον αλγόριθμο του neighborhood έτρεξαν πέντε φορές για το κάθε πρόβλημα, καθώς υπάρχει τυχαιότητα. Με αυτόν τον τρόπο υπολογίστηκε ο μέσος όρος για τους χρόνους εκτέλεσης του κάθε προβλήματος.

4.2 Σύγκριση αλγορίθμων

Στο Σχήμα 4.1 παρουσιάζεται η γραφική παράσταση του γεωμετρικού μέσου χρόνου από τους αλγόριθμους που έχουν τη χρήση του λύτη Gurobi. Πιο συγκεκριμένα, παρατηρούνται οι χρόνοι για όλες τις τεχνικές του Branch and Bound μεθόδου και του Gurobi. Οι Branch and Bound τεχνικές πλην τις τεχνικές με τον περιορισμό του gap έχουν σχεδόν τα ίδια αποτελέσματα. Αναλυτικότερα, ο Branch and Bound BFS έχει σε τέσσερα προβλήματα διαφορετική λύση (από +6 μέχρι +93), ο Branch and Bound DFS έχει σε πέντε προβλήματα (από +3 μέχρι +93). Ενώ ο Branch and Bound BFS και ο Branch and Bound DFS με μυωπική τεχνική έτρεξαν μέχρι το τέλος καθώς σε μόνο δυο προβλήματα (pmed36, pmed38) ξεπέρασαν τη μία ώρα και έκαναν μέχρι δύο. Έτσι, από αυτές τις τέσσερις τεχνικές είναι καλύτερα αυτές που συνδυάζονται και τη μυωπική τεχνική. Από αυτές τις δύο η DFS έχει ακόμα καλύτερο χρόνο (7.23) από την BFS (8.73). Επίσης, είναι σημαντικό να σημειωθεί ότι ο χρόνος εκτέλεσης του DFS είναι ελάχιστα χειρότερος από τον χρόνο του Gurobi (7.17) και προσφέροντάς και οι δύο τις βέλτιστες λύσεις δεν υπάρχει μεγάλη διαφορά στην επιλογή μια εκ των δύο αυτών. Τέλος, οι τεχνικές του Branch and Bound με τη χρήση του περιορισμού απόκλισης (gap 1% ή 5%) μπορούν να μειώσουν ακόμα περισσότερο τον χρόνο με την ύπαρξη μιας μικρής απόκλισης έπο τη βέλτιστη λύση όπως φαίνεται και παρακάτω στο Σχήμα 4.3.

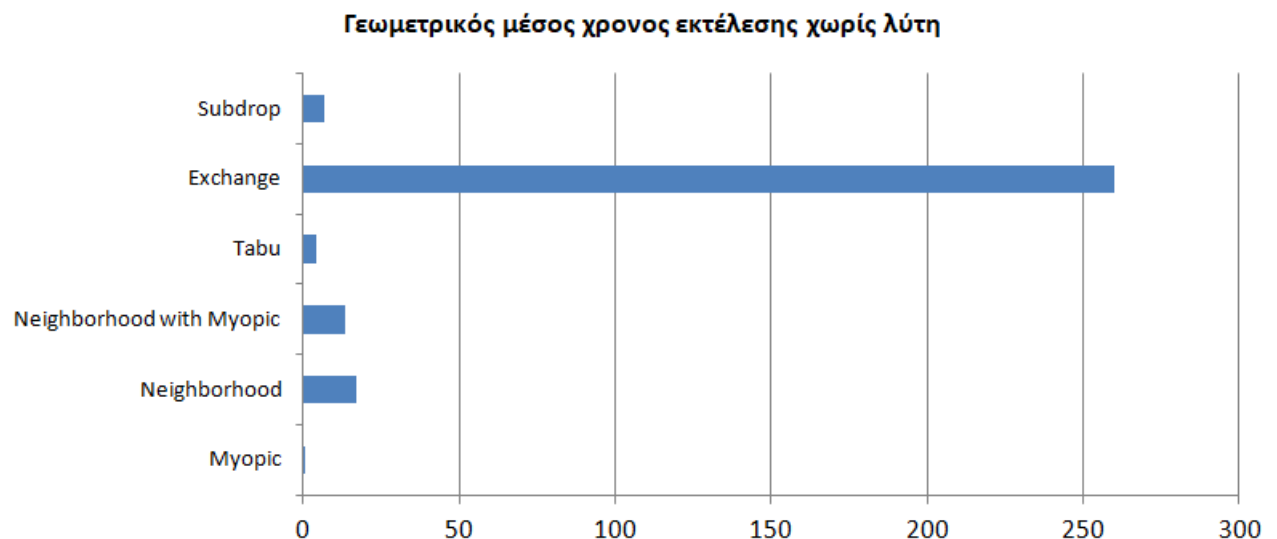
Στο Σχήμα 4.2 παρουσιάζεται η γραφική παράσταση του γεωμετρικού μέσου

Σχήμα 4.1: Γεωμετρικός μέσος χρόνου εκτέλεσης με λύτη



χρόνου από τους αλγόριθμους που δε χρησιμοποιούν τον λύτη Gurobi. Εδώ φαίνεται πως ο αλγόριθμος exchange είναι ο πιο χρονοβόρος σε σχέση με τις άλλες ευρετικές μεθόδους. Έπειτα, επικεντρώνοντας στις υπόλοιπες πέντε μεθόδους ο μυωπικός είναι ο πιο γρήγορος (0.02) και ακολουθεί ο μεθευρετικός tabu (3.92).

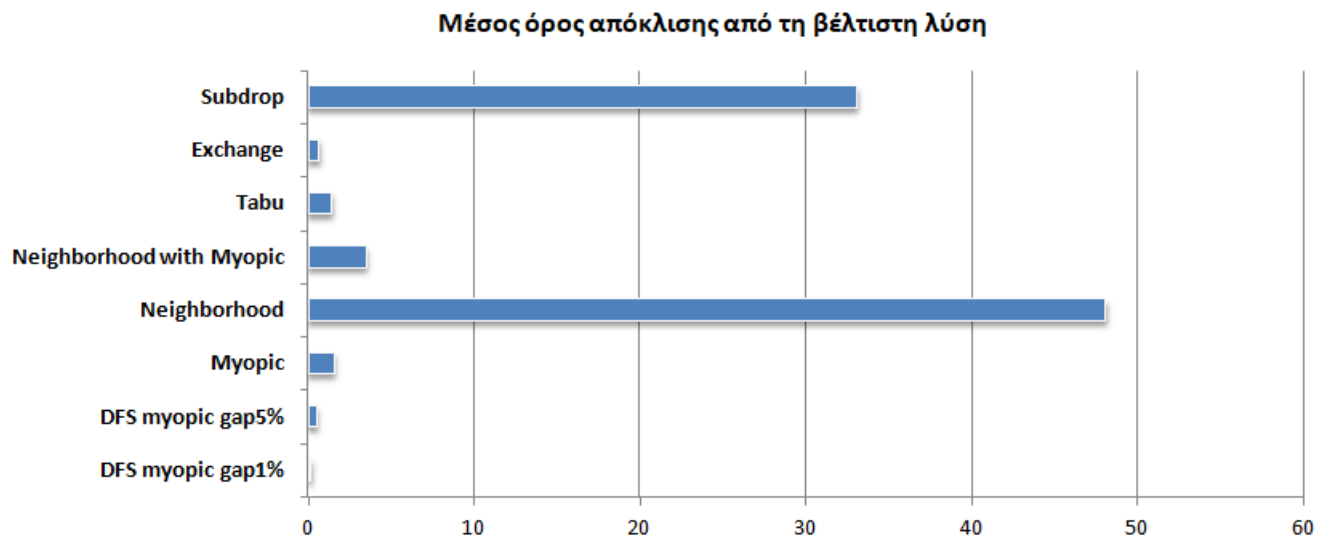
Σχήμα 4.2: Γεωμετρικός μέσος χρόνου εκτέλεσης χωρίς λύτη



Στο Σχήμα 4.3 παρουσιάζεται η γραφική παράσταση του μέσου όρου των αποκλίσεων των αλγορίθμων από τη βέλτιστη λύση. Το γράφημα αυτό αναφέρεται σε όλους τους κώδικες εκτός των Branch and Bound καθώς όπως προαναφέρθηκε αυτοί

έχουν σαν λύσεις σχεδόν όλες ίδιες με τη βέλτιστη. Η σύγκριση με βάση τους υπόλοιπους αλγόριθμους φανερώνει ότι ο Neighborhood (47.98) και ο Subdrop (32.97) έχουν τη μεγαλύτερη απόκλιση. Έπειτα, παρατηρείται ότι οι DFS με τον περιορισμό της απόκλισης (για 1% 0.177 και για 5% 0.47575) έχουν τις μικρότερες απόκλισης. Ακόμα, σε αυτό το σημείο μπορεί να παρατηρηθεί ότι ο αλγόριθμος Exchange έχει πολύ μικρή απόκλιση σε σχέση με τους άλλους αλγόριθμους που είχαν πολύ λιγότερο χρόνο εκτέλεσης.

Σχήμα 4.3: Μέσος όρος απόκλισης από τη βέλτιστη λύση



4.2.1 Ανάλυση αποτελεσμάτων αλγορίθμων

Ο Gurobi μας δίνει τις βέλτιστες λύσεις για κάθε πρόβλημα με χρόνο εκτέλεσης 7.17 sec.

Ο Branch and Bound BFS έχει χρόνο εκτέλεσης 12.85 sec. Το ποσοστό απόκλισης σχεδόν 0% καθώς σε 36 προβλήματα από τα 40 βρίσκει βέλτιστη λύση και στα άλλα 4 είναι σχεδόν βέλτιστη λύση. Σε αυτά τα 4 προβλήματα έχει αποτυχία καθώς ο χρόνος ξεπερνάει τη μία ώρα που υπάρχει σαν περιορισμός.

Ο Branch and Bound DFS έχει χρόνο εκτέλεσης 15.16 sec. Το ποσοστό απόκλισης είναι σχεδόν 0% αφού σε 35 προβλήματα βρίσκει βέλτιστη λύση και στα υπόλοιπα 5 βρίσκει σχεδόν βέλτιστη λύση. Τα 5 αυτά προβλήματα αποτελούν αποτυχία αφού χρειάζονται πάνω από το όριο χρόνου.

Ο αλγόριθμος Branch and Bound BFS με χρήση μυωπικής μεθόδου χρειάζεται 8.73 sec, πολύ λιγότερο χρόνο από τους απλούς Branch and Bound. Ακόμα, το ποσοστό της απόκλισης είναι 0% καθώς βρίσκει σε όλα τις βέλτιστες λύσεις. Πρακτικά όμως έχει 2 αποτυχημένα προβλήματα (pmed36, pmed38) καθώς ο χρόνος εκτέλεσης είναι πάνω από μια ώρα.

Ο αλγόριθμος Branch and Bound DFS με χρήση μυωπικής μεθόδου χρειάζεται 7.23 sec, πολύ λιγότερο χρόνο από τους απλούς Branch and Bound. Ακόμα, το ποσοστό της απόκλισης είναι 0% καθώς βρίσκει σε όλα τις βέλτιστες λύσεις. Έχει όμως και ένα πρόβλημα με αποτυχία αφού ξεπερνά τη μια ώρα εκτέλεσης.

Ο DFS με μυωπική τεχνική και απόκλιση 1% έχει χρόνο εκτέλεσης 3.97 sec. Ακόμα η απόκλιση του είναι η μικρότερη σε σχέση με όλους τους αλγόριθμους με ποσοστό 0.11%. Βρίσκει βέλτιστη λύση σε 25 προβλήματα και σχεδόν βέλτιστη λύση σε 15 προβλήματα. Σε αυτή τη μέθοδο δεν υπάρχει καμία αποτυχία εφόσον όλα τα προβλήματα χρειάζονται λιγότερο από μια ώρα.

Ο DFS με μυωπική τεχνική και απόκλιση 5% έχει τον λιγότερο χρόνο σε σχέση με τους αλγόριθμους που κάνουν χρήση του λύτη με χρόνο 2.65 sec. Η απόκλιση του είναι η δεύτερη μικρότερη με 0.44%. Ο αριθμός των βέλτιστων λύσεων που βρίσκει είναι σε 22 προβλήματα. Ενώ ο αριθμός των σχεδόν βέλτιστων είναι 11. Επίσης, δεν υπάρχει καμία αποτυχία στα προβλήματα καθώς ο χρόνος είναι πολύ λιγότερος της μίας ώρας.

Ο Μυωπικός αλγόριθμος είναι από τις καλύτερες επιλογές ειδικά ανάμεσα στους ευρετικούς. Είναι ο πιο γρήγορος σε εκτέλεση από όλους τους αλγορίθμους με χρόνο 0.02 και έχει τον πέμπτο καλύτερο μέσο όρο απόκλισης (1.50%). Βρήκε βέλτιστη λύση σε δυο προβλήματα (pmed21 και pmed 31) και σχεδόν βέλτιστη σε 13 προβλήματα. Επιπλέον, όλα τα προβλήματα εκτελούνται σε κλάσματα, οπότε δεν υπάρχει αποτυχία.

Ο Neighborhood χρειάζεται ως χρόνο εκτέλεσης έναν από τους μεγαλύτερους χρόνους εκτέλεσης 17 sec. Επιπλέον, ο μέσος όρος απόκλισης είναι ο μεγαλύτερος από όλους τους αλγόριθμους με ποσοστό 47.98%. Δεν βρίσκει καμία βέλτιστη ή σχεδόν βέλτιστη λύση, οπότε δεν αποτελεί καλή επιλογή.

Ο Neighborhood με χρήση μυωπικής μεθόδου έχει αρκετή μείωση στον χρόνο σε σχέση με τον απλό Neighborhood, καθώς ο χρόνος εκτέλεσης είναι 13.51 sec.

Ακόμα, πιο δραματική μείωση έχει η απόκλιση που έχει ποσοστό 3.42%. Βρίσκει μια βέλτιστη λύση (pmed31) και μια σχετικά βέλτιστη (pmed27). Ενώ δεν υπάρχει καμία αποτυχία ως τον περιορισμό του χρόνου.

Ο Tabu έχει από τις πιο γρήγορες εκτελέσεις με 3.92 sec. Επίσης, η απόκλιση του είναι η τέταρτη καλύτερη με ποσοστό 1.34%. Βρίσκει δύο βέλτιστες λύσεις (pmed21 και pmed 31) και 15 σχεδόν βέλτιστες. Επίσης, κανένα πρόβλημα δεν υπερβαίνει το όριο χρόνου, με αποτέλεσμα να μην υπάρχει αποτυχία.

Ο αλγόριθμος Exchange απαιτεί τον περισσότερο χρόνο εκτέλεσης 259.96 sec. Αντιθέτως, το ποσοστό της απόκλισης είναι το τρίτο καλύτερο 0.58%. Επιπρόσθετα, βρίσκει σε 18 προβλήματα τη βέλτιστη λύση και σε 12 προβλήματα σχεδόν βέλτιστη. Στην εκτέλεση αυτής της μεθόδου υπάρχουν 12 προβλήματα που αποτελούν αποτυχία ως προς τον χρόνο εκτέλεσης, καθώς απαιτούν πάνω από μία ώρα.

Ο αλγόριθμος Subdrop έχει έναν από τους καλούς χρόνους 6.94 sec. Αντίθετα η απόκλιση του είναι η δεύτερη χειρότερη με ποσοστό 32.97%. Κανένα πρόβλημα δεν ξεπερνάει τον περιορισμό του χρόνου. Ακόμα, δεν καταφέρνει να βρει ούτε βέλτιστη ούτε σχεδόν βέλτιστη λύση. Με αποτέλεσμα να μην αποτελεί καλή επιλογή.

Τέλος, ο αλγόριθμος Drop σε 29 από τα προβλήματα χρειάζεται πάνω από μία ώρα εκτέλεσης. Με αποτέλεσμα να μην προλαβαίνει ο αλγόριθμός να απορρίψει τον απαραίτητο αριθμό κόμβων, ώστε να περισσέψουν μόνο οι p κόμβοι. Έτσι, ο αλγόριθμος αυτός στα 10 πρώτα προβλήματα βρίσκει λύση. Στα υπόλοιπα έχει σαν αποτέλεσμα μια λύση που αφορά σε πολύ μεγαλύτερο αριθμό εγκαταστάσεων προς άνοιγμα.

Με τον συνδυασμό και των τριών γραφημάτων ο καλύτερος αλγόριθμος ως προς τον χρόνο αλλά και την απόκλιση είναι ο DFS με τη χρήση μυωπικής τεχνικής και εφαρμογή του περιορισμού της απόκλισης 1%. Ο DFS με περιορισμό 5% είναι εξίσου πολύ καλή επιλογή απλά ελάχιστα χειρότερος στην απόκλιση. Έπειτα, αν δεν υπάρχει επίπτωση με την ελάχιστη αύξηση του χρόνου η επόμενη επιλογή είναι η χρήση του Gurobi ή η χρήση του DFS με τη μυωπική μέθοδο.

4.3 Αποτελέσματα μετρήσεων αλγορίθμων

Στον Πίνακα 4.1 δίνονται οι βέλτιστες λύσεις για το κάθε πρόβλημα με τη χρήση του λύτη Gurobi. Ειδικότερα, στον πίνακα αυτόν αναφέρονται τα ονόματα από τα

προβλήματα της βιβλιοθήκης ORLIB, με την παρουσίαση των δυο βασικών δεδομένων του κάθε προβλήματος. Αυτά είναι το πλήθος των κόμβων και ο ζητούμενος αριθμός p . Έπειτα, αναγράφεται η βέλτιστη λύση και ο χρόνος εκτέλεσης.

Στον Πίνακα 4.2 δίνονται οι χρόνοι του Branch and Bound για όλες τις τεχνικές του. Οι λύσεις δεν περιλαμβάνονται στον πίνακα καθώς είναι ίδιες με τις βέλτιστες, εκτός κάποιων περιπτώσεων. Οι περιπτώσεις αυτές αφορούν στα προβλήματα που σταμάτησαν πριν βρουν τη βέλτιστη λύση, λόγω του περιορισμού χρόνου της μιας ώρας, έτσι αναφέρονται σε παρένθεση το κόστος που είχαν ως εκείνη τη στιγμή.

Από τον Πίνακα 4.3 μέχρι τον Πίνακα 4.10 περιγράφονται οι λύσεις, η απόκλιση και ο χρόνος εκτέλεσης για τον κάθε αλγόριθμο. Η απόκλιση (gap) είναι η απόλυτη διαφορά της βέλτιστης λύσης από τη λύση του κάθε αλγόριθμου διά της βέλτιστης λύσης επί 100%.

Στον Πίνακα 4.11 αναγράφονται τα αποτελέσματα για τον drop αλγόριθμο. Εδώ παρατηρήθηκε ότι ο περιορισμός για τον χρόνο εκτέλεσης δεν μπορεί να εμφανίσει το τελικό αποτέλεσμα, καθώς ο χρόνος είναι πολύ λίγος για να προλάβει να απορρίψει τους κόμβους. Έτσι, σε αυτούς τους Πίνακες αναφέρονται και ο αριθμός p που έχει φτάσει μέχρι εκείνη τη στιγμή. Επίσης, η απόκλιση είναι 100% καθώς δεν έχει επιτευχθεί το ζητούμενο για την εύρεση p ανοικτών εγκαταστάσεων.

Στους Πίνακες 4.12 μέχρι τον Πίνακα 4.15 φαίνονται τα αναλυτικά αποτελέσματα για τον κάθε αλγόριθμο.

Πίνακας 4.1: Αποτελέσματα Gurobi

Πρόβλημα	Πλήθος Κόμβων	p	Λύση	Χρόνος
pmed1	100	5	5,819	0.18
pmed2	100	10	4,093	0.36
pmed3	100	10	4,250	0.30
pmed4	100	20	3,034	0.18
pmed5	100	33	1,355	0.19
pmed6	200	5	7,824	5.11
pmed7	200	10	5,631	0.87
pmed8	200	20	4,445	0.83
pmed9	200	40	2,734	0.77
pmed10	200	67	1,255	0.84
pmed11	300	5	7,696	4.45
pmed12	300	10	6,634	3.86
pmed13	300	30	4,374	1.85
pmed14	300	60	2,968	1.80
pmed15	300	100	1,729	1.71
pmed16	400	5	8,162	14.69
pmed17	400	10	6,999	10.90
pmed18	400	40	4,809	4.80
pmed19	400	80	2,845	3.96
pmed20	400	133	1,789	3.60
pmed21	500	5	9,138	6.45
pmed22	500	10	8,579	119.53
pmed23	500	50	4,619	6.08
pmed24	500	100	2,961	5.46
pmed25	500	167	1,828	4.90
pmed26	600	5	9,917	120.51
pmed27	600	10	8,307	25.08
pmed28	600	60	4,498	12.40
pmed29	600	120	3,033	8.80
pmed30	600	200	1,989	7.20
pmed31	700	5	10,086	59.97
pmed32	700	10	9,297	30.78
pmed33	700	70	4,700	14.93
pmed34	700	140	3,013	16.03
pmed35	800	5	10,400	163.23
pmed36	800	10	9,934	2,113.73
pmed37	800	80	5,057	20.80
pmed38	900	5	11,060	603.23
pmed39	900	10	9,423	547.66
pmed40	900	90	5,128	31.35
Geomean				7.17

Πίνακας 4.2: Αποτελέσματα Branch and Bound

Προβλήματα	BFS	DFS	BFS with myopic	DFS with myopic
pmed1	0.11	0.11	0.12	0.11
pmed2	0.45	0.59	0.43	0.46
pmed3	0.44	0.54	0.44	0.39
pmed4	0.07	0.07	0.07	0.07
pmed5	0.06	0.06	0.07	0.07
pmed6	48.45	65.55	6.13	8.34
pmed7	0.43	0.51	0.49	0.45
pmed8	0.31	0.36	0.36	0.34
pmed9	0.27	0.32	0.31	0.28
pmed10	0.28	0.30	0.33	0.29
pmed11	16.15	19.60	3.44	2.89
pmed12	41.23	32.82	2.29	1.93
pmed13	0.86	1.11	1.08	0.89
pmed14	2.52	5.67	166.43	66.50
pmed15	0.74	0.93	0.88	0.74
pmed16	919.17	657.34	76.92	34.32
pmed17	90.43	222.16	53.58	35.13
pmed18	7.54	10.00	14.06	11.29
pmed19	1.76	2.52	2.10	1.74
pmed20	1.74	2.50	2.08	1.92
pmed21	5.11	7.83	6.44	5.25
pmed22	2717.56	1960.40	202.47	191.04
pmed23	3.42	4.19	5.41	4.24
pmed24	3.53	4.43	4.46	4.40
pmed25	3.36	4.36	4.24	3.78
pmed26	3,534.48 (9917)	3,603.43 (9917)	92.40	78.45
pmed27	615.88	3603.61 (8310)	122.28	76.51
pmed28	5.62	6.88	8.06	6.96
pmed29	5.25	6.29	6.51	6.50
pmed30	4.95	5.86	5.97	5.99
pmed31	3,595.20 (10086)	2,499.56	193.91	135.28
pmed32	579.05	699.60	64.43	122.85
pmed33	11.29	9.45	14.52	9.21
pmed34	9.81	8.40	12.44	9.53
pmed35	3,601.51 (10,406)	3,602.17 (10,406)	203.35	151.17
pmed36	3,601.60 (9954)	3,602.10 (9954)	9612.27	4561.91
pmed37	16.16	14.42	35.54	29.87
pmed38	3,601.74 (11,153)	3,601.81 (11,153)	4,769.68	3,035.33
pmed39	3,601.31 (9,451)	3,601.98 (9,451)	1,341.80	900.95
pmed40	22.30	17.79	24.07	21.47
Geomean	12.85	15.16	8.73	7.23

Πίνακας 4.3: Αποτελέσματα DFS με Myopic Gap 1%

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	0%	0.11
pmed2	4,118	0.61%	0.09
pmed3	4,253	0.07%	0.20
pmed4	3,034	0%	0.07
pmed5	1,355	0%	0.07
pmed6	7,824	0%	7.60
pmed7	5,631	0%	0.44
pmed8	4,445	0%	0.33
pmed9	2,734	0%	0.27
pmed10	1,255	0%	0.28
pmed11	7,721	0.32%	2.03
pmed12	6,651	0.26%	1.35
pmed13	4,374	0%	0.85
pmed14	2,996	0.94%	27.92
pmed15	1,729	0%	0.80
pmed16	8,165	0.04%	18.18
pmed17	7,019	0.29%	2.75
pmed18	4,812	0.06%	3.03
pmed19	2,845	0%	1.71
pmed20	1,789	0%	1.72
pmed21	9,138	0%	5.12
pmed22	8,586	0.08%	141.00
pmed23	4,619	0%	3.41
pmed24	2,961	0%	3.56
pmed25	1,828	0%	3.45
pmed26	9,942	0.25%	30.15
pmed27	8,364	0.69%	7.98
pmed28	4,498	0%	5.53
pmed29	3,033	0%	5.21
pmed30	1,989	0%	5.18
pmed31	10,086	0%	13.93
pmed32	9,331	0.37%	14.73
pmed33	4,700	0%	9.05
pmed34	3,013	0%	7.85
pmed35	10,405	0.05%	41.37
pmed36	9,934	0%	3,295.23
pmed37	5,072	0.3%	23.59
pmed38	11,060	0%	2,393.62
pmed39	9,451	0.3%	21.89
pmed40	5,128	0%	17.35
Geomean		0.11	3.97

Πίνακας 4.4: Αποτελέσματα DFS με Myopic Gap 5%

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	0%	0.11
pmed2	4,118	0.61%	0.10
pmed3	4,399	3.51%	0.10
pmed4	3,034	0%	0.07
pmed5	1,355	0%	0.07
pmed6	8,027	2.59%	0.89
pmed7	5,631	0%	0.55
pmed8	4,445	0%	0.37
pmed9	2,734	0%	0.31
pmed10	1,255	0%	0.31
pmed11	7,721	0.32%	2.45
pmed12	6,651	0.26%	1.76
pmed13	4,374	0%	1.09
pmed14	3,013	1.52%	0.94
pmed15	1,729	0%	0.89
pmed16	8,232	0.86%	6.36
pmed17	7,019	0.29%	3.58
pmed18	4,873	1.33%	2.50
pmed19	2,845	0%	2.10
pmed20	1,789	0%	2.17
pmed21	9,138	0%	6.26
pmed22	8,670	1.86%	8.70
pmed23	4,619	0%	4.31
pmed24	2,961	0%	4.46
pmed25	1,828	0%	4.15
pmed26	10,093	1.77%	12.76
pmed27	8,364	0.69%	10.02
pmed28	4,498	0%	7.03
pmed29	3,033	0%	6.57
pmed30	1,989	0%	5.93
pmed31	10,086	0%	16.97
pmed32	9,331	0.37%	18.49
pmed33	4,700	0%	11.55
pmed34	3,013	0%	9.93
pmed35	10,406	0.06%	23.55
pmed36	9,954	0.2%	21.26
pmed37	5,118	1.21%	14.59
pmed38	11,153	0.84%	36.34
pmed39	9,451	0.3%	27.29
pmed40	5,128	0%	24.12
Geomean		0.44	2.65

Πίνακας 4.5: Αποτελέσματα Myopic

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	1.24%	0.0003
pmed2	4,118	0.61%	0.0005
pmed3	4,399	3.51%	0.001
pmed4	3,088	1.78%	0.001
pmed5	1,378	1.7%	0.001
pmed6	8,027	2.59%	0.001
pmed7	5,646	0.27%	0.002
pmed8	4,472	0.61%	0.004
pmed9	2,841	3.91%	0.01
pmed10	1,295	3.19%	0.01
pmed11	7,721	0.32%	0.002
pmed12	6,651	0.26%	0.005
pmed13	4,467	2.13%	0.01
pmed14	3,013	1.52%	0.02
pmed15	1,761	1.85%	0.04
pmed16	8,232	0.86%	0.004
pmed17	7,019	0.29%	0.01
pmed18	4,873	1.33%	0.03
pmed19	2,899	1.9%	0.05
pmed20	1,866	4.3%	0.08
pmed21	9,138	0%	0.01
pmed22	8,670	1.06%	0.01
pmed23	4,694	1.62%	0.06
pmed24	3,009	1.62%	0.11
pmed25	1,896	3.72%	0.16
pmed26	10,093	1.77%	0.01
pmed27	8,364	0.69%	0.02
pmed28	4,579	1.8%	0.10
pmed29	3,104	2.34%	0.18
pmed30	2,037	2.41%	0.28
pmed31	10,086	0%	0.01
pmed32	9,331	0.37%	0.03
pmed33	4,798	2.09%	0.15
pmed34	3,097	2.79%	0.29
pmed35	10,406	0.06%	0.02
pmed36	9,954	0.2%	0.03
pmed37	5,118	1.21%	0.23
pmed38	11,153	0.84%	0.02
pmed39	9,451	0.3%	0.04
pmed40	5,190	1.21%	0.33
Geomean		1.51	0.02

Πίνακας 4.6: Αποτελέσματα Neighborhood

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	7,866	35.18%	0.07
pmed2	5,453	33.23%	0.13
pmed3	6,592	55.11%	0.13
pmed4	4,335	42.88%	0.25
pmed5	2,219	63.76%	0.42
pmed6	10,846	38.62%	0.50
pmed7	7,630	35.5%	0.95
pmed8	6,196	39.39%	1.89
pmed9	4,169	52.49%	3.77
pmed10	2,239	78.41%	12.62
pmed11	10,445	35.72%	1.65
pmed12	9,205	38.75%	3.20
pmed13	6,574	50.3%	9.50
pmed14	4,449	49.9%	18.96
pmed15	2,817	62.93%	48.35
pmed16	11,991	46.91%	3.88
pmed17	9,587	36.98%	7.48
pmed18	6,771	40.8%	29.58
pmed19	4,294	50.93%	118.71
pmed20	2,875	60.7%	99.12
pmed21	12,260	34.17%	7.36
pmed22	11,894	38.64%	14.35
pmed23	6,548	41.76%	106.63
pmed24	4,471	51%	71.49
pmed25	3,170	73.41%	120.57
pmed26	15,069	51.95%	13.00
pmed27	11,809	42.16%	24.63
pmed28	6,803	51.24%	74.23
pmed29	4,457	46.95%	445.34
pmed30	3,266	64.2%	486.92
pmed31	12,695	25.87%	21.02
pmed32	12,100	30.15%	39.45
pmed33	6,820	45.11%	271.97
pmed34	4,487	48.92%	1,358.07
pmed35	14,047	35.07%	29.89
pmed36	12,430	25.13%	64.21
pmed37	12,430	145.8%	722.23
pmed38	15,591	40.97%	82.96
pmed39	12,934	37.26%	298.49
pmed40	7,229	40.97%	1,375.33
Geomean		47.98	17.03

Πίνακας 4.7: Αποτελέσματα Neighborhood με Myopic

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,960	2.42%	0.06
pmed2	4,404	7.6%	0.13
pmed3	4,516	6.26%	0.13
pmed4	3,088	1.78%	0.13
pmed5	1,378	1.7%	0.21
pmed6	8,480	8.38%	0.51
pmed7	5,873	4.3%	0.95
pmed8	4,750	6.86%	1.88
pmed9	2,841	3.91%	1.89
pmed10	1,323	5.42 %	6.29
pmed11	8,289	7.71%	2.44
pmed12	6,794	2.41%	3.16
pmed13	4,513	3.18%	9.41
pmed14	3,013	1.52%	9.44
pmed15	1,760	1.79%	31.35
pmed16	8,650	5.98%	3.80
pmed17	7,103	1.49%	7.39
pmed18	4,901	1.91%	29.31
pmed19	2,955	3.87%	87.75
pmed20	1,878	4.97%	97.24
pmed21	9,514	4.11%	7.32
pmed22	8,798	2.55%	14.28
pmed23	4,927	6.67%	106.03
pmed24	3,038	2.6%	141.31
pmed25	1,906	4.27%	235.86
pmed26	10,093	1.77%	6.30
pmed27	8,364	0.69%	12.31
pmed28	4,580	1.82%	145.52
pmed29	3,118	2.8%	580.94
pmed30	2,043	2.71%	484.55
pmed31	10,086	0%	11.38
pmed32	9,642	3.71%	38.77
pmed33	4,793	1.98%	268.66
pmed34	3,097	2.79%	268.19
pmed35	10,705	2.93%	29.73
pmed36	10,061	1.28%	57.84
pmed37	5,129	1.42%	455.16
pmed38	11,708	5.86%	42.54
pmed39	9,662	2.54%	81.87
pmed40	5,190	1.21%	364.10
Geomean		3.43	13.51

Πίνακας 4.8: Αποτελέσματα Tabu

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	1.24%	0.04
pmed2	4,113	0.49%	0.08
pmed3	4,357	2.52%	0.08
pmed4	3,088	1.78%	0.09
pmed5	1,362	0.52%	0.10
pmed6	8,027	2.59%	0.51
pmed7	5,645	0.25%	0.52
pmed8	4,465	0.45%	0.57
pmed9	2,802	2.49%	0.65
pmed10	1,295	3.19%	0.78
pmed11	7,714	0.23%	1.73
pmed12	6,651	0.26%	1.70
pmed13	4,459	1.94%	1.88
pmed14	3,011	1.45%	2.19
pmed15	1,760	1.79%	2.59
pmed16	8,232	0.86%	3.93
pmed17	7,019	0.29%	3.96
pmed18	4,836	0.56%	4.36
pmed19	2,897	1.83%	5.11
pmed20	1,863	4.14%	6.08
pmed21	9,138	0%	7.64
pmed22	8,679	1.06%	7.61
pmed23	4,694	1.62%	8.58
pmed24	3,009	1.62%	9.96
pmed25	1,895	3.67%	11.89
pmed26	10,093	1.77%	13.21
pmed27	8,360	0.64%	13.03
pmed28	4,565	1.49%	14.76
pmed29	3,099	2.18%	17.05
pmed30	2,037	2.41%	20.47
pmed31	10,086	0%	20.86
pmed32	9,318	0.23%	20.62
pmed33	4,793	1.98%	23.10
pmed34	3,093	2.66%	26.96
pmed35	10,406	0.06%	31.19
pmed36	9,947	0.13%	30.57
pmed37	5,107	0.99%	35.05
pmed38	11,153	0.84%	44.35
pmed39	9,451	0.3%	43.56
pmed40	5,190	1.21%	49.66
Geomean		1.34	3.92

Πίνακας 4.9: Αποτελέσματα Exchange

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,819	0%	0.65
pmed2	4,105	0.29%	1.58
pmed3	4,250	0%	3.16
pmed4	3,046	0.4%	5.65
pmed5	1,355	0%	13.37
pmed6	7,824	0%	4.95
pmed7	5,645	0.25%	12.12
pmed8	4,457	0.27%	45.78
pmed9	2,753	0.69%	167.55
pmed10	1,263	0.64%	614.73
pmed11	7,696	0%	10.99
pmed12	6,634	0%	40.77
pmed13	4,374	0%	338.39
pmed14	2,971	0.1%	2,044.44
pmed15	1,738	0.52%	3,600.60
pmed16	8,162	0%	40.69
pmed17	6,999	0%	101.34
pmed18	4,811	0.04%	1,494.19
pmed19	2,858	0.46%	3,608.40
pmed20	1,836	2.63%	3,621.06
pmed21	9,138	0%	27.49
pmed22	8,669	1.05%	221.57
pmed23	4,619	0%	3,602.69
pmed24	2,987	0.88%	3,609.05
pmed25	1,891	3.45%	3,611.28
pmed26	9,924	0.07%	95.24
pmed27	8,307	0%	509.93
pmed28	4,522	0.53%	3,610.29
pmed29	3,102	2.27%	3,632.70
pmed30	2,034	2.26%	3,614.50
pmed31	10,086	0%	78.51
pmed32	9,297	0%	584.06
pmed33	4,766	1.4%	3,618.23
pmed34	3,093	2.66%	3,607.36
pmed35	10,400	0%	229.49
pmed36	9,934	0%	851.69
pmed37	5,118	1.21%	3,621.63
pmed38	11,060	0%	513.12
pmed39	9,423	0%	1,226.82
pmed40	5,186	1.13%	3,605.55
Geomean		0.58	259.96

Πίνακας 4.10: Αποτελέσματα Subdrop

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	7,039	20.97%	0.03
pmed2	4,819	17.74%	0.10
pmed3	5,280	24.24%	0.10
pmed4	4,166	37.31%	0.38
pmed5	1,951	43.99%	3.23
pmed6	9,111	16.45%	0.10
pmed7	6,790	20.58%	0.35
pmed8	5,827	31.09%	1.34
pmed9	3,775	38.08%	5.39
pmed10	2,116	68.61%	15.78
pmed11	8,490	10.32%	0.23
pmed12	7,898	19.05%	0.77
pmed13	6,040	38.09%	6.50
pmed14	4,140	39.49%	26.52
pmed15	2,791	61.42%	76.87
pmed16	9,192	12.62%	0.39
pmed17	8,653	23.63%	1.34
pmed18	6,271	30.4%	19.87
pmed19	4,079	43.37%	81.38
pmed20	2,889	61.49%	235.21
pmed21	11,167	22.2%	0.62
pmed22	10,853	26.51%	2.07
pmed23	6,344	37.35%	47.78
pmed24	4,228	42.79%	197.17
pmed25	2,999	64.06%	578.09
pmed26	12,033	21.34%	0.86
pmed27	10,720	29.05%	2.96
pmed28	6,096	35.53%	97.32
pmed29	4,342	43.16%	401.55
pmed30	3,180	59.88%	1163.19
pmed31	11,974	18.72%	1.17
pmed32	11,689	25.73%	4.01
pmed33	6,328	34.64%	178.48
pmed34	4,298	42.65%	737.97
pmed35	12,348	18.73%	1.53
pmed36	12,370	24.52%	5.21
pmed37	6,896	36.37%	302.92
pmed38	13,410	21.25%	1.94
pmed39	11,580	22.89%	6.57
pmed40	6,801	32.62%	483.04
Geomean		32.97	6.96

Πίνακας 4.11: Αποτελέσματα Drop

Προβλήματα	Λύση	p	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,827	5	100%	35.98
pmed2	4,142	10	100%	35.85
pmed3	4,413	10	100%	36.22
pmed4	3,173	20	100%	35.89
pmed5	1,394	33	100%	34.95
pmed6	7,999	5	100%	1,170.95
pmed7	5,778	10	100%	1,169.07
pmed8	4,602	20	100%	1,135.77
pmed9	2,922	40	100%	1,105.05
pmed10	1,319	67	100%	1,105.005
pmed11	81	252	100%	3,622.03
pmed12	111	252	100%	3,633.68
pmed13	114	251	100%	3645.60
pmed14	97	251	100%	3,624.41
pmed15	98	251	100%	3,634.59
pmed16	14	386	100%	3,770.45
pmed17	14	386	100%	3,742.12
pmed18	14	386	100%	3,748.46
pmed19	14	386	100%	3,748.20
pmed20	14	386	100%	3,743.11
pmed21	6	494	100%	4,041.78
pmed22	6	494	100%	4,050.36
pmed23	6	494	100%	4,037.83
pmed24	6	494	100%	4,048.51
pmed25	6	494	100%	4,068.25
pmed26	3	597	100%	4,130.69
pmed27	3	597	100%	4,130.08
pmed28	3	597	100%	4,132.48
pmed29	3	597	100%	4,132.91
pmed30	3	597	100%	4,136.11
pmed31	2	698	100%	5,107.74
pmed32	2	698	100%	5,097.30
pmed33	2	698	100%	5,103.002
pmed34	2	698	100%	5,117.75
pmed35	1	799	100%	4,351.58
pmed36	1	799	100%	4,362.61
pmed37	1	799	100%	4,358.98
pmed38	1	899	100%	6,962.71
pmed39	1	899	100%	7,001.68
pmed40	1	899	100%	6,989.27
Geomean			-	-

Πίνακας 4.12: Χρόνοι εκτέλεσης με λύτη

Πρόβλημα	Gurobi	BFS	DFS	BFSmyopic	DFSmyopic	DFSgap1%	DFSgap5%
pmed1	0.18	0.11	0.11	0.12	0.11	0.11	0.11
pmed2	0.36	0.45	0.59	0.43	0.46	0.09	0.10
pmed3	0.30	0.44	0.54	0.44	0.38	0.20	0.10
pmed4	0.18	0.07	0.07	0.07	0.07	0.07	0.07
pmed5	0.19	0.06	0.07	0.07	0.07	0.07	0.07
pmed6	5.11	48.45	65.55	6.13	8.34	7.60	0.89
pmed7	0.87	0.43	0.51	0.49	0.45	0.44	0.55
pmed8	0.83	0.31	0.36	0.36	0.34	0.33	0.37
pmed9	0.77	0.27	0.32	0.31	0.28	0.27	0.31
pmed10	0.84	0.28	0.30	0.33	0.29	0.28	0.31
pmed11	4.45	16.15	19.60	3.44	2.89	2.03	2.45
pmed12	3.86	41.23	32.82	2.29	1.93	1.35	1.76
pmed13	1.85	0.86	1.11	1.08	0.89	0.85	1.09
pmed14	1.80	2.52	5.67	166.43	66.50	27.92	0.94
pmed15	1.71	0.74	0.94	0.88	0.74	0.80	0.89
pmed16	14.69	919.17	657.34	76.92	34.32	18.18	6.36
pmed17	10.89	90.43	222.16	53.58	35.13	2.75	3.58
pmed18	4.80	7.54	10.00	14.06	11.29	3.03	2.50
pmed19	3.96	1.76	2.52	2.10	1.74	1.71	2.10
pmed20	3.60	1.74	2.50	2.08	1.92	1.72	2.17
pmed21	6.45	5.11	7.83	6.44	5.25	5.12	6.26
pmed22	119.53	2,717.56	1,960.40	202.47	191.04	141.00	8.70
pmed23	6.08	3.42	4.19	5.41	4.24	3.41	4.31
pmed24	5.46	3.53	4.43	4.46	4.40	3.56	4.46
pmed25	4.90	3.36	4.36	4.24	3.78	3.45	4.15
pmed26	120.51	3,534.48	3,603.43	92.40	78.45	30.15	12.76
pmed27	25.08	615.88	3,603.61	122.28	76.51	7.98	10.02
pmed28	12.40	5.62	6.88	8.06	6.96	5.53	7.03
pmed29	8.80	5.25	6.29	6.51	6.50	5.21	6.57
pmed30	7.20	4.95	5.86	5.97	5.99	5.18	5.93
pmed31	59.97	3,595.20	2,499.56	193.91	135.28	13.93	16.97
pmed32	30.78	579.05	699.60	64.43	122.85	14.73	18.49
pmed33	14.93	11.29	9.45	14.52	9.21	9.05	11.55
pmed34	16.03	9.81	8.40	12.44	9.53	7.85	9.93
pmed35	163.23	3,601.51	3,602.17	203.35	151.17	41.37	23.55
pmed36	2,113.73	3,601.60	3,602.10	9,612.27	4,561.91	3,295.23	21.26
pmed37	20.80	16.16	14.42	35.54	29.87	23.59	14.59
pmed38	603.23	3,601.74	3,601.81	4,769.68	3,035.33	2,393.62	36.34
pmed39	547.66	3,601.31	3,601.98	1,341.80	900.95	21.89	27.29
pmed40	31.35	22.30	17.79	24.07	21.47	17.35	24.12
Geomean	7.17	12.85	15.16	8.73	7.23	3.97	2.65

Πίνακας 4.13: Χρόνοι εκτέλεσης χωρίς λύτη

Πρόβλημα	Myopic	Neighborhood	Neighborhood Myopic	Tabu	Exchange	Subdrop
pmed1	0.0003	0.07	0.06	0.04	0.65	0.03
pmed2	0.0005	0.13	0.12	0.08	1.58	0.10
pmed3	0.001	0.13	0.13	0.08	3.16	0.10
pmed4	0.01	0.25	0.13	0.09	5.65	0.38
pmed5	0.001	0.42	0.21	0.10	13.37	3.23
pmed6	0.001	0.50	0.51	0.51	4.95	0.10
pmed7	0.002	0.95	0.95	0.52	12.12	0.35
pmed8	0.004	1.89	1.88	0.57	45.78	1.34
pmed9	0.01	3.77	1.89	0.65	167.55	5.39
pmed10	0.01	12.62	6.29	0.78	614.73	15.78
pmed11	0.002	1.65	2.44	1.73	10.99	0.23
pmed12	0.005	3.20	3.16	1.70	40.77	0.77
pmed13	0.01	9.50	9.41	1.88	338.39	6.50
pmed14	0.02	18.96	9.44	2.19	2,044.44	26.52
pmed15	0.04	48.35	31.35	2.59	3,600.60	76.87
pmed16	0.004	3.88	3.80	3.93	40.69	0.39
pmed17	0.01	7.48	7.39	3.96	101.34	1.34
pmed18	0.03	29.58	29.31	4.36	1,494.19	19.87
pmed19	0.05	118.71	87.75	5.11	3,608.40	81.37
pmed20	0.08	99.12	97.24	6.08	361.06	235.21
pmed21	0.01	7.36	7.32	7.64	27.49	0.61
pmed22	0.01	14.35	14.28	7.61	221.57	2.07
pmed23	0.06	106.63	106.03	8.58	3,602.69	47.78
pmed24	0.11	71.49	141.31	9.96	3,609.05	197.17
pmed25	0.16	120.57	235.86	11.89	3,611.28	578.09
pmed26	0.01	13.00	6.30	13.21	95.24	0.86
pmed27	0.02	24.63	12.31	13.03	509.93	2.96
pmed28	0.10	74.23	145.52	14.76	3,610.29	97.32
pmed29	0.18	445.34	580.95	17.05	3,632.70	401.55
pmed30	0.28	486.92	484.55	20.47	3,614.50	1,163.19
pmed31	0.01	21.02	11.38	20.86	78.51	1.17
pmed32	0.03	39.45	38.77	20.62	584.06	4.01
pmed33	0.15	271.97	268.66	23.10	3,618.23	178.48
pmed34	0.29	1,358.07	268.19	26.96	3,607.36	737.97
pmed35	0.02	29.89	29.73	31.19	229.49	1.53
pmed36	0.03	64.21	57.84	30.57	851.69	5.21
pmed37	0.23	722.23	455.16	35.05	3,621.63	302.92
pmed38	0.02	82.96	42.54	44.35	513.12	1.94
pmed39	0.04	298.49	81.87	43.56	1,226.82	6.57
pmed40	0.33	1,375.33	364.10	49.66	3,605.54	483.04
Geomean	0.02	17.03	13.51	3.92	259.96	6.96

Πίνακας 4.14: Αποκλίσεις αλγορίθμων

Πρόβλημα	DFS1%	DFS5%	Myopic	Neighbor hood	Neighbor Myopic	Tabu	Exchange	Subdrop
pmed1	0%	0%	1.24%	35.18%	2.42%	1.24%	0%	20.97%
pmed2	0.61%	0.61%	0.61%	33.23%	7.6%	0.49%	0.29%	17.74%
pmed3	0.07%	3.51%	3.51%	55.11%	6.26%	2.52%	0%	24.24%
pmed4	0%	0%	1.78%	42.88%	1.78%	1.78%	0.4%	37.31%
pmed5	0%	0%	1.7%	63.76%	1.7%	0.52%	0%	43.99%
pmed6	0%	2.59%	2.59%	38.62%	8.38%	2.59%	0%	16.45%
pmed7	0%	0%	0.27%	35.5%	4.3%	0.25%	0.25%	20.58%
pmed8	0%	0%	0.61%	39.39%	6.86%	0.45%	0.27%	31.09%
pmed9	0%	0%	3.91%	52.49%	3.91%	2.49%	0.69%	38.08%
pmed10	0%	0%	3.19%	78.41%	5.42%	3.19%	0.64%	68.61%
pmed11	0.32%	0.32%	0.32%	35.72%	7.71%	0.23%	0%	10.32%
pmed12	0.26%	0.26%	0.26%	38.75%	2.41%	0.26%	0%	19.05%
pmed13	0%	0%	2.13%	50.3%	3.18%	1.94%	0%	38.09%
pmed14	0.94%	1.52%	1.52%	49.9%	1.52%	1.45%	0.1%	39.49%
pmed15	0%	0%	1.85%	62.93%	1.79%	1.79%	0.52%	61.42%
pmed16	0.04%	0.86%	0.86%	49.91%	5.98%	0.86%	0%	12.62%
pmed17	0.29%	0.29%	0.26%	36.98%	1.49%	0.29%	0%	23.63%
pmed18	0.06%	1.33%	1.33%	40.8%	1.91%	0.56%	0.04%	30.4%
pmed19	0%	0%	1.9%	50.93%	3.87%	1.83%	0.46%	43.37%
pmed20	0%	0%	4.3%	60.7%	4.97%	4.14%	2.63%	61.49%
pmed21	0%	0%	0%	34.17%	4.11%	0%	0%	22.2%
pmed22	0.08%	1.86%	1.06%	38.64%	2.55%	1.06%	1.05%	26.51%
pmed23	0%	0%	1.62%	41.76%	6.67%	1.62%	0%	37.35%
pmed24	0%	0%	1.62%	51%	2.6%	1.62%	0.88%	42.79%
pmed25	0%	0%	3.72%	73.41%	4.27%	3.67%	3.45%	64.06%
pmed26	0.25%	1.77%	1.77%	51.95%	1.77%	1.77%	0.07%	21.34%
pmed27	0.69%	0.69%	0.69%	42.16%	0.69%	0.64%	0%	29.05%
pmed28	0%	0%	1.8%	51.24%	1.82%	1.49%	0.53%	35.53%
pmed29	0%	0%	2.34%	46.95%	2.8%	2.18%	2.27%	43.16%
pmed30	0%	0%	2.41%	64.2%	2.71%	2.41%	2.26%	59.88%
pmed31	0%	0%	0%	25.87%	0%	0%	0%	18.72%
pmed32	0.37%	0.37%	0.37%	30.15%	3.71%	0.23%	0%	25.73%
pmed33	0%	0%	2.09%	45.11%	1.98%	1.98%	1.4%	34.64%
pmed34	0%	0%	2.79%	48.92%	2.79%	2.66%	2.66%	42.65%
pmed35	0.05%	0.06%	0.06%	35.07%	2.93%	0.06%	0%	18.73%
pmed36	0%	0.2%	0.2%	25.13%	1.28%	0.13%	0%	24.52%
pmed37	0.3%	1.21%	1.21%	145.8%	1.42%	0.99%	1.21%	36.37%
pmed38	0%	0.84%	0.84%	40.9%	5.86%	0.84%	0%	21.25%
pmed39	0.3%	0.3%	0.3%	37.26%	2.54%	0.3%	0%	22.89%
pmed40	0%	0%	1.21%	40.97%	1.21%	1.21%	1.13%	32.62%
Geomean	0.11	0.44	1.51	47.98	3.43	1.34	0.58	32.97

Πίνακας 4.15: Συγκεντρωτικά αποτελέσματα αλγορίθμων

Αλγόριθμοι	Απόκλιση	Χρόνος Εκτέλεσης(sec)
Gurobi	0%	7.17
BFS	0.037%	12.85
DFS	0.042%	15.16
BFS myopic	0%	8.73
DFS myopic	0%	7.23
DFS myopic gap 1%	0.11%	3.97
DFS myopic gap 5%	0.44%	2.65
Myopic	1.51%	0.02
Neighborhood	47.98%	17.03
Neighborhood myopic	3.43%	13.51
Tabu	1.34%	3.92
Exchange	0.58%	259.96
Subdrop	32.97%	6.94

Κεφάλαιο 5

Συμπεράσματα

Συμπεραίνοντας στην εργασία αυτή υλοποιήθηκαν 14 αλγόριθμοι για την επίλυση του προβλήματος χωροθέτησης p median. Συγκεκριμένα υλοποιήθηκε αλγόριθμος για τη μοντελοποίηση του γραμμικού προβλήματος με τη χρήση του λύτη Gurobi για να βρεθεί η βέλτιστη λύση κάθε προβλήματος. Έπειτα, υλοποιήθηκαν 2 βασικοί μέθοδοι του Branch and Bound, BFS και DFS και πάνω σε αυτούς εφαρμόστηκαν κάποιες τεχνικές με χρήση ευρετικών αλγορίθμων, συγκεκριμένα τον μυωπικό, και επιπλέον περιορισμών ως προς την απόκλιση από τη βέλτιστη λύση. Στη συνέχεια υλοποιήθηκαν διάφοροι ευρετικοί και μεθευρετικοί αλγόριθμοι. Αναλυτικότερα οι αλγόριθμοι αυτοί είναι ο μυωπικός, ο Neighborhood με δυο εκτελέσεις, μία με τυχαιότητα και μία με την εφαρμογή του μυωπικού. Ακόμα, υλοποιήθηκε ο Tabu με αρχική λύση από τον μυωπικό, ο Subdrop και ο Drop approach.

Εκτελώντας όλους τους αλγόριθμους κατά μονάς, παρατηρήθηκαν τα εξής συμπεράσματα: Αρχικά, εκτελέστηκε η μοντελοποίησή του γραμμικού προβλήματος με τη χρήση του λύτη Gurobi. Με αποτέλεσμα να βρεθούν οι βέλτιστες λύσεις για κάθε πρόβλημα για να μπορέσει να επιτευχθεί και η σύγκριση μεταξύ των άλλων μεθόδων. Ο χρόνος εκτέλεσης είναι 7.17 sec.

Έπειτα, εκτελέστηκαν οι branch and bound μέθοδοι. Πρώτα έτρεξε ο branch and bound με τεχνική κατά πλάτος. Από τα 40 προβλήματα δε βρέθηκε η βέλτιστη λύση μόνο σε 6 περιπτώσεις, καθώς υπήρχε ο περιορισμός της μίας ώρας για την ολοκλήρωση εκτέλεσης του κάθε προβλήματος. Ο χρόνος εκτέλεσης είναι 12.85 sec, δηλαδή 5 sec επιπλέον από τον αλγόριθμο με χρήση του λύτη.

Μετά, εκτελέστηκε ο branch and bound με τεχνική κατά βάθος. Αντίθετα με την προηγούμενη τεχνική σε αυτήν δεν υπήρξε βέλτιστη λύση για 5 προβλήματα.

Ενώ ο χρόνος εκτέλεσης είναι 15.16 sec. Έτσι, συμπεραίνουμε ότι αυτή η τεχνική χρειάζεται περισσότερο χρόνο από την προηγούμενη.

Στη συνέχεια ακολούθησε ο branch and bound με τεχνική κατά πλάτος και με χρήση του μωπικού αλγόριθμου. Με την εφαρμογή αυτής της μεθόδου παρατηρήθηκε μία πτώση στον χρόνο κατά 4 sec. Σε αυτή τη μέθοδο, σε σχέση με τις δυο προηγούμενες μεθόδους, βρέθηκαν όλες οι βέλτιστες λύσεις.

Μετάπειτα, εκτελέστηκε ο branch and bound με τεχνική κατά βάθος και με χρήση του μωπικού αλγόριθμου. Η προαναφερθείσα έχει μεγαλύτερη πτώση χρόνου εκτέλεσης κατά 8 sec σε σύγκριση με τη μέθοδο branch and bound με τεχνική κατά βάθος. Τέλος, όπως προαναφέρθηκε, η περίπτωση της branch and bound με τεχνική κατά πλάτος με χρήση του μωπικού διαφέρει κατά 4 sec σε χρόνο εκτέλεσης από την branch and bound κατά πλάτος. Εν κατακλείδι, φαίνεται ότι η χρήση του μωπικού αλγόριθμου επέφερε μεγαλύτερη πτώση στον branch and bound με τεχνική κατά βάθος.

Αφού εκτελέστηκαν όλες οι τεχνικές της branch and bound συμπεραίνεται ότι ο branch and bound με τεχνική κατά βάθος και με χρήση του μωπικού αλγόριθμου είχε τα καλύτερα αποτελέσματα. Έτσι, χρησιμοποιήθηκε αυτή η μέθοδος με την προσθήκη ενός ακόμα περιορισμού. Ο περιορισμός αυτός αφορά στην απόκλιση. Συγκεκριμένα έτρεξε για δύο περιπτώσεις. Στην πρώτη περίπτωση, όταν η απόκλιση αγγίζει το 1% και λιγότερο από αυτό σταματάει η εκτέλεση του αλγόριθμου. Αντίστοιχα στη δεύτερη περίπτωση όταν ο φτάσει στο 5% και κάτω. Για την πρώτη περίπτωση τα αποτελέσματα ήταν να έχει 0.11% απόκλιση από τις βέλτιστες λύσεις και ο χρόνος εκτέλεσης να είναι 3.97 sec. Ενώ στη δεύτερη περίπτωση η απόκλιση είναι 0.44% και ο χρόνος 2.65 sec. Επειδή, όμως, η διαφορά των χρόνων εκτέλεσης μεταξύ αυτών των δύο μεθόδων είναι λιγότερη από 1 sec δίνεται παραπάνω προσοχή στην απόκλιση. Έτσι, καλύτερη επιλογή αλγόριθμου -μεταξύ αυτών- είναι ο branch and bound κατά βάθος με χρήση μωπικού και περιορισμό απόκλισης 1%.

Έπειτα, από αυτές τις μεθόδους ακολούθησαν οι ευρετικές και οι μεθευρετικές μέθοδοι. Πρώτα εκτελέστηκε ο μωπικός. Αυτός είχε αποτέλεσμα απόκλισης 1.51% και χρόνο εκτέλεσης 0.02 sec. Άρα σε ελάχιστο χρόνο πάρα πολύ καλά αποτελέσματα.

Ακολούθησε η εκτέλεση του αλγόριθμου των καλύτερων γειτόνων. Με το πέρας

της εκτέλεσης παρατηρήθηκε ότι η απόκλιση είναι χειρότερη σε σχέση με τις άλλες. Πιο αναλυτικά, η απόκλιση του είναι 47.98% και ο χρόνος εκτέλεσης 17.03 sec.

Όταν, όμως, εκτελέστηκε ο αλγόριθμός των καλύτερων γειτόνων με χρήση μωπικού αλγόριθμου φάνηκε μεγάλη διαφορά ως προς την απόκλιση. Η απόκλιση αυτής της μεθόδου είναι 3.43%. Ακόμα, και ο χρόνος εκτέλεσης είναι μικρότερος κατά 4 sec.

Μετά, έγινε χρήση του αλγόριθμου tabu. Σε σύγκριση με τις μεθόδους των καλύτερων γειτόνων αυτή η μέθοδος είναι καλύτερη και ως προς χρόνο αλλά και ως προς απόκλιση. Καθώς, η απόκλιση αυτού του αλγόριθμου είναι 1.34% και ο χρόνος εκτέλεσης είναι 3.92 sec.

Στη συνέχεια εκτελέστηκε ο exchange αλγόριθμος. Ο αλγόριθμος αυτός ενώ είχε πολύ καλά αποτελέσματα απόκλισης, 0.58%, ο χρόνος εκτέλεσης που χρειάστηκε ήταν ο μεγαλύτερος.

Έπειτα, ακολούθησε ο subdrop αλγόριθμος. Αυτός είχε τη δεύτερη σε σειρά χειρότερη τιμή απόκλισης των 32.97%. Ενώ και ο χρόνος δεν είχε τα καλύτερα αποτελέσματα, καθώς χρειάστηκε 6.69 sec.

Τέλος, εκτελέστηκε και ο drop αλγόριθμος. Για την εκτέλεση αυτού του αλγόριθμου χρειάστηκε από το 11 πρόβλημα και έπειτα, πάνω από μία ώρα για να ολοκληρωθεί η εκτέλεση του αλγόριθμου. Με αποτέλεσμα να μην μπορούν να βρεθούν λύσεις στα προβλήματα και η απόκλιση να είναι 100%.

Συμπεραίνεται λοιπόν ότι ο DFS με μωπική μέθοδο και τη χρήση του περιορισμού για την απόκλιση 1% αλλά και 5% έχουν τα καλύτερα αποτελέσματα σε σύγκριση με τον χρόνο και την απόκλιση. Ενώ σε περίπτωση που ο χρόνος δεν επηρεάζει την απόφαση επιλογής αλγόριθμου ο Gurobi και ο DFS με μωπική μέθοδο είναι οι καλύτερες επιλογές.

Βιβλιογραφία

- [1] Osman Alp, Erhan Erkut, and Zvi Drezner. An efficient genetic algorithm for the p-median problem. *Annals of Operations Research*, 122:21–42, 2003.
- [2] Rimmi Anand, Divya Aggarwal, and Vijay Kumar. A comparative analysis of optimization solvers. *Journal of Statistics and Management Systems*, 20(4):623–635, 2017.
- [3] Sunith Bandaru and Kalyanmoy Deb. Metaheuristic techniques. *Decision Sciences*, pages 693–750, 2016.
- [4] Francisco Barahona and Ranga Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87:385–399, 2000.
- [5] John E Beasley. A note on solving large p-median problems. *European Journal of Operational Research*, 21(2):270–273, 1985.
- [6] John E Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
- [7] John E Beasley. Lagrangean heuristics for location problems. *European Journal of Operational Research*, 65(3):383–399, 1993.
- [8] Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.
- [9] Cesar Beltran, Claude Tadonki, and J Ph Vial. Solving the p-median problem with a semi-lagrangian relaxation. *Computational Optimization and Applications*, 35(2):239–260, 2006.
- [10] Robert L Bowerman, Paul H Calamai, and G Brent Hall. The demand partitioning method for reducing aggregation errors in p-median problems. *Computers & Operations Research*, 26(10-11):1097–1111, 1999.
- [11] M Eugénia Captivo. Fast primal and dual heuristics for the p-median location problem. *European Journal of Operational Research*, 52(1):65–74, 1991.
- [12] Sohail S Chaudhry, Shiwei He, and Peggy E Chaudhry. Solving a class of facility location problems using genetic algorithms. *Expert Systems*, 20(2):86–91, 2003.
- [13] Fernando Chiyoshi and Roberto D Galvão. A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operations Research*, 96(1-4):61–74, 2000.

-
- [14] Gerard Cornuejols, Marshall L Fisher, and George L Nemhauser. Exceptional paper—location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.
- [15] Teodor Gabriel Crainic, Michel Gendreau, Pierre Hansen, and Nenad Mladenović. Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics*, 10:293–314, 2004.
- [16] John R Current and David A Schilling. Elimination of source a and b errors in p-median location problems. *Geographical Analysis*, 19(2):95–110, 1987.
- [17] Zuo Dai and TY Cheung. A new heuristic approach for the p-median problem. *Journal of the Operational Research Society*, 48(9):950–960, 1997.
- [18] Sittipong Dantrakul, Chulin Likasiri, and Radom Pongvuthithum. Applied p-median and p-center algorithms for facility location problems. *Expert Systems with Applications*, 41(8):3596–3604, 2014.
- [19] M Daskin. *Network and discrete location wiley*. New York, 1995.
- [20] Mark S Daskin and Kayse Lee Maass. The p-median problem. In *Location Science*, pages 21–45. Springer, 2015.
- [21] Paul J Densham and Gerard Rushton. A more efficient heuristic for solving large p-median problems. *Papers in Regional Science*, 71(3):307–329, 1992.
- [22] Catherine Dibble and P Densham. Generating interesting alternatives in gis and sdss using genetic algorithms. In *Gis Lis-International Conference-*, volume 1, pages 180–180. American Society for Photogrammetry and Remote Sensing, 1993.
- [23] Edsger Wybe Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [24] Michael Dzator and J Dzator. An efficient modified greedy algorithm for the p-median problem. 2015.
- [25] M_A Efroymsen and TL Ray. A branch-bound algorithm for plant location. *Operations Research*, 14(3):361–368, 1966.
- [26] Erhan Erkut and Burçin Bozkaya. Analysis of aggregation errors for the p-median problem. *Computers & Operations Research*, 26(10-11):1075–1096, 1999.
- [27] Donald Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009, 1978.
- [28] Vladimir Estivill-Castro and Rodolfo Torres-Velázquez. Hybrid genetic algorithm for solving the p-median problem. In *Asia-Pacific Conference on Simulated Evolution and Learning*, pages 18–25. Springer, 1998.
- [29] EFATL Feldman, FA Lehrer, and TL Ray. Warehouse location under continuous economies of scale. *Management Science*, 12(9):670–684, 1966.

-
- [30] Marshall L Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.
- [31] Richard L Francis, Timothy J Lowe, and Arie Tamir. Aggregation error bounds for a class of location models. *Operations Research*, 48(2):294–307, 2000.
- [32] Roberto D Galvão. A dual-bounded algorithm for the p-median problem. *Operations Research*, 28(5):1112–1124, 1980.
- [33] Roberto Dieguez Galvao. The use of lagrangean relaxation in the solution of uncapacitated facility location problems. *Computers & Operations Research*, 1993.
- [34] Roberto Diéguez Galvão. Uncapacitated facility location problems: contributions. *Pesquisa Operacional*, 24:7–38, 2004.
- [35] Félix García-López, Belén Melián-Batista, José A Moreno-Pérez, and J Marcos Moreno-Vega. The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics*, 8:375–388, 2002.
- [36] Michel Gendreau and Jean-Yves Potvin. Tabu search. *Search methodologies: introductory tutorials in optimization and decision support techniques*, pages 165–186, 2005.
- [37] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.
- [38] Fred Glover. Tabu search: A tutorial. *Interfaces*, 20(4):74–94, 1990.
- [39] Fred Glover, Eric Taillard, and Eric Taillard. A user’s guide to tabu search. *Annals of Operations Research*, 41(1):1–28, 1993.
- [40] Boris Goldengorin, Dmitry Krushinsky, and Panos M Pardalos. Cell formation in industrial engineering. *Springer Optimization and Its Applications*, 2013.
- [41] E Goncharov and Y Kochetov. Probabilistic tabu search for the unconstrained discrete optimization problems. *Discrete Analysis and Operations Research*, 9(2):13–30, 2002.
- [42] Michael F Goodchild. The aggregation problem in location-allocation. *Geographical Analysis*, 11(3):240–255, 1979.
- [43] Michael F Goodchild and Valerian T Noronha. *Location-allocation for small computers*, volume 8. Department of Geography, University of Iowa, 1983.
- [44] Kitty Gupta. What is the difference between bfs and dfs algorithms. <https://www.freelancinggig.com/blog/wp-content/uploads/2019/02/BFS-and-DFS-Algorithms.png>, 2019.
- [45] Harsha Gwalani, Chetan Tiwari, and Armin R Mikler. Evaluation of heuristics for the p-median problem: Scale and spatial demand distribution. *Computers, Environment and Urban Systems*, 88:101656, 2021.

-
- [46] Pierre Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. In *Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy*, pages 70–145, 1986.
- [47] Pierre Hansen and Nenad Mladenović. Variable neighborhood search for the p-median. *Location Science*, 5(4):207–226, 1997.
- [48] Pierre Hansen, Nenad Mladenović, and Dionisio Perez-Britos. Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335–350, 2001.
- [49] Edward L Hillsman and Richard Rhoda. Errors in measuring distances from populations to service centers. *The Annals of Regional Science*, 12:74–88, 1978.
- [50] M John Hodgson. Toward more realistic allocation in location—allocation models: An interaction approach. *Environment and Planning A*, 10(11):1273–1285, 1978.
- [51] Michael John Hodgson and Susan Neuman. A gis approach to eliminating source c aggregation error in p-median models. *Computers & Operations Research*, 1993.
- [52] MJ Hodgson and S Salhi. Using a quadtree structure to eliminate aggregation error in point to point allocation. *IFORS, Montreal*, 1998.
- [53] Catherine M Hosage and Michael F Goodchild. Discrete space location-allocation solutions from genetic algorithms. *Annals of Operations Research*, 6:35–46, 1986.
- [54] Michelle Hribar and Mark S Daskin. A dynamic programming heuristic for the p-median problem. *European Journal of Operational Research*, 101(3):499–508, 1997.
- [55] O Kariv and SL Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.
- [56] Yuri Kochetov, Tatyana Levanova, Ekaterina Alekseeva, and Maxim Loresh. Large neighborhood local search for the p-median problem. *Yugoslav Journal of Operations Research*, 15(1):53–63, 2005.
- [57] Madhukar R Korupolu, C Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of Algorithms*, 37(1):146–188, 2000.
- [58] Alfred A Kuehn and Michael J Hamburger. A heuristic program for locating warehouses. *Management Science*, 9(4):643–666, 1963.
- [59] Manuel Laguna and José Luis González-Velarde. *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, volume 12. Springer Science & Business Media, 1999.
- [60] Eugene L Lawler and David E Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, 1966.
- [61] Tatyana Valentinovna Levanova and MA Loresh. Algorithms of ant system and simulated annealing for the p-median problem. *Automation and Remote Control*, 65:431–438, 2004.

-
- [62] Helena R Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search. In *Handbook of Metaheuristics*, pages 320–353. Springer, 2003.
- [63] FE Maranzana. On the location of supply points to minimize transport costs. *Journal of the Operational Research Society*, 15(3):261–270, 1964.
- [64] Enrique Domínguez Merino, José Muñoz-Pérez, and José M Jerez-Aragonés. Neural network algorithms for the p-median problem. In ESANN, pages 385–392, 2003.
- [65] Enrique Dominguez Merino and J Muñoz Perez. An efficient neural network algorithm for the p-median problem. *Lecture Notes in Computer Science*, pages 460–469, 2002.
- [66] Claude Michel and Michel Rueher. Handling software upgradeability problems with milp solvers. *arXiv preprint arXiv:1007.1020*, 2010.
- [67] Nenad Mladenović, Jack Brimberg, Pierre Hansen, and José A Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007.
- [68] Nenad Mladenovic, José A Moreno, and J Marcos Moreno-Vega. *A Chain-Interchange Tabu Search Method (Tabu search in solving p-facility location-allocation problems)*. École des hautes études commerciales, Groupe d'études et de recherche en ..., 1995.
- [69] Nenad Mladenovic, José A Moreno, and J Marcos Moreno-Vega. Chain-interchange heuristic method. *Yugosl J Oper Res*, 6(1):41–54, 1996.
- [70] JA Moreno-Perez, JL Roda Garcia, and J Marcos Moreno-Vega. A parallel genetic algorithm for the discrete p-median problem. *Studies in Location Analysis*, 7:131–141, 1994.
- [71] John M Mulvey and Harlan P Crowder. Cluster analysis: An application of lagrangian relaxation. *Management Science*, 25(4):329–340, 1979.
- [72] Alan T Murray and Richard L Church. Applying simulated annealing to location-planning models. *Journal of Heuristics*, 2:31–53, 1996.
- [73] David Pisinger and Stefan Ropke. Large neighborhood search. *Handbook of Metaheuristics*, pages 99–127, 2019.
- [74] Nelio Domingues Pizzolato. A heuristic for large-size p-median location problems with application to school location. *Annals of Operations Research*, 50:473–485, 1994.
- [75] Mauricio GC Resende and Renato F Werneck. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10:59–88, 2004.
- [76] Erik Rolland, David A Schilling, and John R Current. An efficient tabu search procedure for the p-median problem. *European Journal of Operational Research*, 96(2):329–342, 1997.
- [77] KE Rosing, EL Hillsman, and Hester Rosing-Vogelaar. The robustness of two common heuristics for the p-median problem. *Environment and Planning A*, 11(4):373–380, 1979.

-
- [78] Kenneth E Rosing, Charles S ReVelle, E Rolland, David A Schilling, and JR Current. Heuristic concentration and tabu search: A head to head comparison. *European Journal of Operational Research*, 104(1):93–99, 1998.
- [79] Kenneth E Rosing, Charles S ReVelle, and David A Schilling. A gamma heuristic for the p-median problem. *European Journal of Operational Research*, 117(3):522–532, 1999.
- [80] Kenneth Earl Rosing and CS ReVelle. Heuristic concentration: Two stage solution construction. *European Journal of Operational Research*, 97(1):75–86, 1997.
- [81] G Rushton. Alloc-heuristic solutions to multi-facility location problems on a graph. *Computer Programs for Location-Allocation Problems*, pages 163–187, 1973.
- [82] Said Salhi. A perturbation heuristic for a class of location problems. *Journal of the Operational Research Society*, 48(12):1233–1240, 1997.
- [83] Said Salhi. Defining tabu list size and aspiration criterion within tabu search methods. *Computers & Operations Research*, 29(1):67–86, 2002.
- [84] Said Salhi and RA Atkinson. Subdrop: A modified drop heuristic for location problems. *Location Science*, 3(4):267–273, 1995.
- [85] Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming—CP98: 4th International Conference, CP98 Pisa, Italy, October 26–30, 1998 Proceedings 4*, pages 417–431. Springer, 1998.
- [86] Yoshiaki Shimizu, Tatsuhiko Sakaguchi, and Takatobu Miura. Parallel computing for huge scale logistics optimization through binary pso associated with topological comparison. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 8(1):JAMDSM0005–JAMDSM0005, 2014.
- [87] Edward A Silver, R Victor, V Vidal, and Dominique de Werra. A tutorial on heuristic methods. *European Journal of Operational Research*, 5(3):153–162, 1980.
- [88] Éric D Taillard. Heuristic methods for large centroid clustering problems. *Journal of Heuristics*, 9:51–73, 2003.
- [89] Michael B Teitz and Polly Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16(5):955–961, 1968.
- [90] S. Voss. A reverse elimination approach for the p-median problem. *Studies in Locational Analysis*, 8:49–58, 1996.
- [91] Xiao Zhu Wang. The comparison of three algorithms in shortest path issue. In *Journal of Physics: Conference Series*, volume 1087, page 022011. IOP Publishing, 2018.
- [92] RA Whitaker. A tight bound drop exchange algorithm for solving the p median problem. *Environment and Planning A*, 13(6):669–680, 1981.
- [93] RA Whitaker. A fast algorithm for the greedy interchange for large-scale clustering and median location problems. *INFOR: Information Systems and Operational Research*, 21(2):95–108, 1983.

[94] H Paul Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.

Παραρτήματα

Παράρτημα Α΄

Αποτελέσματα Αλγορίθμων

Πίνακας Α.1: Αποτελέσματα Gurobi

Πρόβλημα	Πλήθος Κόμβων	p	Λύση	Χρόνος
pmed1	100	5	5,819	0.18
pmed2	100	10	4,093	0.36
pmed3	100	10	4,250	0.30
pmed4	100	20	3,034	0.18
pmed5	100	33	1,355	0.19
pmed6	200	5	7,824	5.11
pmed7	200	10	5,631	0.87
pmed8	200	20	4,445	0.83
pmed9	200	40	2,734	0.77
pmed10	200	67	1,255	0.84
pmed11	300	5	7,696	4.45
pmed12	300	10	6,634	3.86
pmed13	300	30	4,374	1.85
pmed14	300	60	2,968	1.80
pmed15	300	100	1,729	1.71
pmed16	400	5	8,162	14.69
pmed17	400	10	6,999	10.90
pmed18	400	40	4,809	4.80
pmed19	400	80	2,845	3.96
pmed20	400	133	1,789	3.60
pmed21	500	5	9,138	6.45
pmed22	500	10	8,579	119.53
pmed23	500	50	4,619	6.08
pmed24	500	100	2,961	5.46
pmed25	500	167	1,828	4.90
pmed26	600	5	9,917	120.51
pmed27	600	10	8,307	25.08
pmed28	600	60	4,498	12.40
pmed29	600	120	3,033	8.80
pmed30	600	200	1,989	7.20
pmed31	700	5	10,086	59.97
pmed32	700	10	9,297	30.78
pmed33	700	70	4,700	14.93
pmed34	700	140	3,013	16.03
pmed35	800	5	10,400	163.23
pmed36	800	10	9,934	2,113.73
pmed37	800	80	5,057	20.80
pmed38	900	5	11,060	603.23
pmed39	900	10	9,423	547.66
pmed40	900	90	5,128	31.35
Geomean				7.17

Πίνακας Α.2: Αποτελέσματα Branch and Bound

Προβλήματα	BFS	DFS	BFS with myopic	DFS with myopic
pmed1	0.11	0.11	0.12	0.11
pmed2	0.45	0.59	0.43	0.46
pmed3	0.44	0.54	0.44	0.39
pmed4	0.07	0.07	0.07	0.07
pmed5	0.06	0.06	0.07	0.07
pmed6	48.45	65.55	6.13	8.34
pmed7	0.43	0.51	0.49	0.45
pmed8	0.31	0.36	0.36	0.34
pmed9	0.27	0.32	0.31	0.28
pmed10	0.28	0.30	0.33	0.29
pmed11	16.15	19.60	3.44	2.89
pmed12	41.23	32.82	2.29	1.93
pmed13	0.86	1.11	1.08	0.89
pmed14	2.52	5.67	166.43	66.50
pmed15	0.74	0.93	0.88	0.74
pmed16	919.17	657.34	76.92	34.32
pmed17	90.43	222.16	53.58	35.13
pmed18	7.54	10.00	14.06	11.29
pmed19	1.76	2.52	2.10	1.74
pmed20	1.74	2.50	2.08	1.92
pmed21	5.11	7.83	6.44	5.25
pmed22	2,717.56	1,960.40	202.47	191.04
pmed23	3.42	4.19	5.41	4.24
pmed24	3.53	4.43	4.46	4.40
pmed25	3.36	4.36	4.24	3.78
pmed26	3,534.48 (9,917)	3,603.43 (9,917)	92.40	78.45
pmed27	615.88	3,603.61 (8,310)	122.28	76.51
pmed28	5.62	6.88	8.06	6.96
pmed29	5.25	6.29	6.51	6.50
pmed30	4.95	5.86	5.97	5.99
pmed31	3,595.20 (10,086)	2,499.56	193.91	135.28
pmed32	579.05	699.60	64.43	122.85
pmed33	11.29	9.45	14.52	9.21
pmed34	9.81	8.40	12.44	9.53
pmed35	3,601.51 (10,406)	3,602.17 (10,406)	203.35	151.17
pmed36	3,601.60 (9,954)	3,602.10 (9,954)	9,612.27	4,561.91
pmed37	16.16	14.42	35.54	29.87
pmed38	3,601.74 (11,153)	3,601.81 (11,153)	4,769.68	3,035.33
pmed39	3,601.31 (9,451)	3,601.98 (9,451)	1,341.80	900.95
pmed40	22.30	17.79	24.07	21.47
Geomean	12.85	15.16	8.73	7.23

Πίνακας Α'.3: Αποτελέσματα DFS με Myopic Gap 1%

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	0%	0.11
pmed2	4,118	0.61%	0.09
pmed3	4,253	0.07%	0.20
pmed4	3,034	0%	0.07
pmed5	1,355	0%	0.07
pmed6	7,824	0%	7.60
pmed7	5,631	0%	0.44
pmed8	4,445	0%	0.33
pmed9	2,734	0%	0.27
pmed10	1,255	0%	0.28
pmed11	7,721	0.32%	2.03
pmed12	6,651	0.26%	1.35
pmed13	4,374	0%	0.85
pmed14	2,996	0.94%	27.92
pmed15	1,729	0%	0.80
pmed16	8,165	0.04%	18.18
pmed17	7,019	0.29%	2.75
pmed18	4,812	0.06%	3.03
pmed19	2,845	0%	1.71
pmed20	1,789	0%	1.72
pmed21	9,138	0%	5.12
pmed22	8,586	0.08%	141.00
pmed23	4,619	0%	3.41
pmed24	2,961	0%	3.56
pmed25	1,828	0%	3.45
pmed26	9,942	0.25%	30.15
pmed27	8,364	0.69%	7.98
pmed28	4,498	0%	5.53
pmed29	3,033	0%	5.21
pmed30	1,989	0%	5.18
pmed31	10,086	0%	13.93
pmed32	9,331	0.37%	14.73
pmed33	4,700	0%	9.05
pmed34	3,013	0%	7.85
pmed35	10,405	0.05%	41.37
pmed36	9,934	0%	3,295.23
pmed37	5,072	0.3%	23.59
pmed38	11,060	0%	2,393.62
pmed39	9,451	0.3%	21.89
pmed40	5,128	0%	17.35
Geomean		0.11	3.97

Πίνακας Α'.4: Αποτελέσματα DFS με Myopic Gap 5%

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	0%	0.11
pmed2	4,118	0.61%	0.10
pmed3	4,399	3.51%	0.10
pmed4	3,034	0%	0.07
pmed5	1,355	0%	0.07
pmed6	8,027	2.59%	0.89
pmed7	5,631	0%	0.55
pmed8	4,445	0%	0.37
pmed9	2,734	0%	0.31
pmed10	1,255	0%	0.31
pmed11	7,721	0.32%	2.45
pmed12	6,651	0.26%	1.76
pmed13	4,374	0%	1.09
pmed14	3,013	1.52%	0.94
pmed15	1,729	0%	0.89
pmed16	8,232	0.86%	6.36
pmed17	7,019	0.29%	3.58
pmed18	4,873	1.33%	2.50
pmed19	2,845	0%	2.10
pmed20	1,789	0%	2.17
pmed21	9,138	0%	6.26
pmed22	8,670	1.86%	8.70
pmed23	4,619	0%	4.31
pmed24	2,961	0%	4.46
pmed25	1,828	0%	4.15
pmed26	10,093	1.77%	12.76
pmed27	8,364	0.69%	10.02
pmed28	4,498	0%	7.03
pmed29	3,033	0%	6.57
pmed30	1,989	0%	5.93
pmed31	10,086	0%	16.97
pmed32	9,331	0.37%	18.49
pmed33	4,700	0%	11.55
pmed34	3,013	0%	9.93
pmed35	10,406	0.06%	23.55
pmed36	9,954	0.2%	21.26
pmed37	5,118	1.21%	14.59
pmed38	11,153	0.84%	36.34
pmed39	9,451	0.3%	27.29
pmed40	5,128	0%	24.12
Geomean		0.44	2.65

Πίνακας Α'5: Αποτελέσματα Myopic

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	1.24%	0.0003
pmed2	4,118	0.61%	0.0005
pmed3	4,399	3.51%	0.001
pmed4	3,088	1.78%	0.001
pmed5	1,378	1.7%	0.001
pmed6	8,027	2.59%	0.001
pmed7	5,646	0.27%	0.002
pmed8	4,472	0.61%	0.004
pmed9	2,841	3.91%	0.01
pmed10	1,295	3.19%	0.01
pmed11	7,721	0.32%	0.002
pmed12	6,651	0.26%	0.005
pmed13	4,467	2.13%	0.01
pmed14	3,013	1.52%	0.02
pmed15	1,761	1.85%	0.04
pmed16	8,232	0.86%	0.004
pmed17	7,019	0.29%	0.01
pmed18	4,873	1.33%	0.03
pmed19	2,899	1.9%	0.05
pmed20	1,866	4.3%	0.08
pmed21	9,138	0%	0.01
pmed22	8,670	1.06%	0.01
pmed23	4,694	1.62%	0.06
pmed24	3,009	1.62%	0.11
pmed25	1,896	3.72%	0.16
pmed26	10,093	1.77%	0.01
pmed27	8,364	0.69%	0.02
pmed28	4,579	1.8%	0.10
pmed29	3,104	2.34%	0.18
pmed30	2,037	2.41%	0.28
pmed31	10,086	0%	0.01
pmed32	9,331	0.37%	0.03
pmed33	4,798	2.09%	0.15
pmed34	3,097	2.79%	0.29
pmed35	10,406	0.06%	0.02
pmed36	9,954	0.2%	0.03
pmed37	5,118	1.21%	0.23
pmed38	11,153	0.84%	0.02
pmed39	9,451	0.3%	0.04
pmed40	5,190	1.21%	0.33
Geomean		1.51	0.02

Πίνακας Α'.6: Αποτελέσματα Neighborhood

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	7,866	35.18%	0.07
pmed2	5,453	33.23%	0.13
pmed3	6,592	55.11%	0.13
pmed4	4,335	42.88%	0.25
pmed5	2,219	63.76%	0.42
pmed6	10,846	38.62%	0.50
pmed7	7,630	35.5%	0.95
pmed8	6,196	39.39%	1.89
pmed9	4,169	52.49%	3.77
pmed10	2,239	78.41%	12.62
pmed11	10,445	35.72%	1.65
pmed12	9,205	38.75%	3.20
pmed13	6,574	50.3%	9.50
pmed14	4,449	49.9%	18.96
pmed15	2,817	62.93%	48.35
pmed16	11,991	46.91%	3.88
pmed17	9,587	36.98%	7.48
pmed18	6,771	40.8%	29.58
pmed19	4,294	50.93%	118.71
pmed20	2,875	60.7%	99.12
pmed21	12,260	34.17%	7.36
pmed22	11,894	38.64%	14.35
pmed23	6,548	41.76%	106.63
pmed24	4,471	51%	71.49
pmed25	3,170	73.41%	120.57
pmed26	15,069	51.95%	13.00
pmed27	11,809	42.16%	24.63
pmed28	6,803	51.24%	74.23
pmed29	4,457	46.95%	445.34
pmed30	3,266	64.2%	486.92
pmed31	12,695	25.87%	21.02
pmed32	12,100	30.15%	39.45
pmed33	6,820	45.11%	271.97
pmed34	4,487	48.92%	1,358.07
pmed35	14,047	35.07%	29.89
pmed36	12,430	25.13%	64.21
pmed37	12,430	145.8%	722.23
pmed38	15,591	40.97%	82.96
pmed39	12,934	37.26%	298.49
pmed40	7,229	40.97%	1,375.33
Geomean		47.98	17.03

Πίνακας Α'.7: Αποτελέσματα Neighborhood με Myopic

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,960	2.42%	0.06
pmed2	4,404	7.6%	0.13
pmed3	4,516	6.26%	0.13
pmed4	3,088	1.78%	0.13
pmed5	1,378	1.7%	0.21
pmed6	8,480	8.38%	0.51
pmed7	5,873	4.3%	0.95
pmed8	4,750	6.86%	1.88
pmed9	2,841	3.91%	1.89
pmed10	1,323	5.42 %	6.29
pmed11	8,289	7.71%	2.44
pmed12	6,794	2.41%	3.16
pmed13	4,513	3.18%	9.41
pmed14	3,013	1.52%	9.44
pmed15	1,760	1.79%	31.35
pmed16	8,650	5.98%	3.80
pmed17	7,103	1.49%	7.39
pmed18	4,901	1.91%	29.31
pmed19	2,955	3.87%	87.75
pmed20	1,878	4.97%	97.24
pmed21	9,514	4.11%	7.32
pmed22	8,798	2.55%	14.28
pmed23	4,927	6.67%	106.03
pmed24	3,038	2.6%	141.31
pmed25	1,906	4.27%	235.86
pmed26	10,093	1.77%	6.30
pmed27	8,364	0.69%	12.31
pmed28	4,580	1.82%	145.52
pmed29	3,118	2.8%	580.94
pmed30	2,043	2.71%	484.55
pmed31	10,086	0%	11.38
pmed32	9,642	3.71%	38.77
pmed33	4,793	1.98%	268.66
pmed34	3,097	2.79%	268.19
pmed35	10,705	2.93%	29.73
pmed36	10,061	1.28%	57.84
pmed37	5,129	1.42%	455.16
pmed38	11,708	5.86%	42.54
pmed39	9,662	2.54%	81.87
pmed40	5,190	1.21%	364.10
Geomean		3.43	13.51

Πίνακας Α'.8: Αποτελέσματα Tabu

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,891	1.24%	0.04
pmed2	4,113	0.49%	0.08
pmed3	4,357	2.52%	0.08
pmed4	3,088	1.78%	0.09
pmed5	1,362	0.52%	0.10
pmed6	8,027	2.59%	0.51
pmed7	5,645	0.25%	0.52
pmed8	4,465	0.45%	0.57
pmed9	2,802	2.49%	0.65
pmed10	1,295	3.19%	0.78
pmed11	7,714	0.23%	1.73
pmed12	6,651	0.26%	1.70
pmed13	4,459	1.94%	1.88
pmed14	3,011	1.45%	2.19
pmed15	1,760	1.79%	2.59
pmed16	8,232	0.86%	3.93
pmed17	7,019	0.29%	3.96
pmed18	4,836	0.56%	4.36
pmed19	2,897	1.83%	5.11
pmed20	1,863	4.14%	6.08
pmed21	9,138	0%	7.64
pmed22	8,679	1.06%	7.61
pmed23	4,694	1.62%	8.58
pmed24	3,009	1.62%	9.96
pmed25	1,895	3.67%	11.89
pmed26	10,093	1.77%	13.21
pmed27	8,360	0.64%	13.03
pmed28	4,565	1.49%	14.76
pmed29	3,099	2.18%	17.05
pmed30	2,037	2.41%	20.47
pmed31	10,086	0%	20.86
pmed32	9,318	0.23%	20.62
pmed33	4,793	1.98%	23.10
pmed34	3,093	2.66%	26.96
pmed35	10,406	0.06%	31.19
pmed36	9,947	0.13%	30.57
pmed37	5,107	0.99%	35.05
pmed38	11,153	0.84%	44.35
pmed39	9,451	0.3%	43.56
pmed40	5,190	1.21%	49.66
Geomean		1.34	3.92

Πίνακας Α'.9: Αποτελέσματα Exchange

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,819	0%	0.65
pmed2	4,105	0.29%	1.58
pmed3	4,250	0%	3.16
pmed4	3,046	0.4%	5.65
pmed5	1,355	0%	13.37
pmed6	7,824	0%	4.95
pmed7	5,645	0.25%	12.12
pmed8	4,457	0.27%	45.78
pmed9	2,753	0.69%	167.55
pmed10	1,263	0.64%	614.73
pmed11	7,696	0%	10.99
pmed12	6,634	0%	40.77
pmed13	4,374	0%	338.39
pmed14	2,971	0.1%	2,044.44
pmed15	1,738	0.52%	3,600.60
pmed16	8,162	0%	40.69
pmed17	6,999	0%	101.34
pmed18	4,811	0.04%	1,494.19
pmed19	2,858	0.46%	3,608.40
pmed20	1,836	2.63%	3,621.06
pmed21	9,138	0%	27.49
pmed22	8,669	1.05%	221.57
pmed23	4,619	0%	3,602.69
pmed24	2,987	0.88%	3,609.05
pmed25	1,891	3.45%	3,611.28
pmed26	9,924	0.07%	95.24
pmed27	8,307	0%	509.93
pmed28	4,522	0.53%	3,610.29
pmed29	3,102	2.27%	3,632.70
pmed30	2,034	2.26%	3,614.50
pmed31	10,086	0%	78.51
pmed32	9,297	0%	584.06
pmed33	4,766	1.4%	3,618.23
pmed34	3,093	2.66%	3,607.36
pmed35	10,400	0%	229.49
pmed36	9,934	0%	851.69
pmed37	5,118	1.21%	3,621.63
pmed38	11,060	0%	513.12
pmed39	9,423	0%	1,226.82
pmed40	5,186	1.13%	3,605.55
Geomean		0.58	259.96

Πίνακας Α'.10: Αποτελέσματα Subdrop

Προβλήματα	Λύση	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	7,039	20.97%	0.03
pmed2	4,819	17.74%	0.10
pmed3	5,280	24.24%	0.10
pmed4	4,166	37.31%	0.38
pmed5	1,951	43.99%	3.23
pmed6	9,111	16.45%	0.10
pmed7	6,790	20.58%	0.35
pmed8	5,827	31.09%	1.34
pmed9	3,775	38.08%	5.39
pmed10	2,116	68.61%	15.78
pmed11	8,490	10.32%	0.23
pmed12	7,898	19.05%	0.77
pmed13	6,040	38.09%	6.50
pmed14	4,140	39.49%	26.52
pmed15	2,791	61.42%	76.87
pmed16	9,192	12.62%	0.39
pmed17	8,653	23.63%	1.34
pmed18	6,271	30.4%	19.87
pmed19	4,079	43.37%	81.38
pmed20	2,889	61.49%	235.21
pmed21	11,167	22.2%	0.62
pmed22	10,853	26.51%	2.07
pmed23	6,344	37.35%	47.78
pmed24	4,228	42.79%	197.17
pmed25	2,999	64.06%	578.09
pmed26	12,033	21.34%	0.86
pmed27	10,720	29.05%	2.96
pmed28	6,096	35.53%	97.32
pmed29	4,342	43.16%	401.55
pmed30	3,180	59.88%	1163.19
pmed31	11,974	18.72%	1.17
pmed32	11,689	25.73%	4.01
pmed33	6,328	34.64%	178.48
pmed34	4,298	42.65%	737.97
pmed35	12,348	18.73%	1.53
pmed36	12,370	24.52%	5.21
pmed37	6,896	36.37%	302.92
pmed38	13,410	21.25%	1.94
pmed39	11,580	22.89%	6.57
pmed40	6,801	32.62%	483.04
Geomean		32.97	6.96

Πίνακας Α.11: Αποτελέσματα Drop

Προβλήματα	Λύση	p	Απόκλιση	Χρόνος Εκτέλεσης(sec)
pmed1	5,827	5	100%	35.98
pmed2	4,142	10	100%	35.85
pmed3	4,413	10	100%	36.22
pmed4	3,173	20	100%	35.89
pmed5	1,394	33	100%	34.95
pmed6	7,999	5	100%	1170.95
pmed7	5,778	10	100%	1169,07
pmed8	4,602	20	100%	1135.77
pmed9	2,922	40	100%	1,105.05
pmed10	1,319	67	100%	1,105.005
pmed11	81	252	100%	3,622.03
pmed12	111	252	100%	3,633.68
pmed13	114	251	100%	3645.60
pmed14	97	251	100%	3,624.41
pmed15	98	251	100%	3,634.59
pmed16	14	386	100%	3,770.45
pmed17	14	386	100%	3,742.12
pmed18	14	386	100%	3,748.46
pmed19	14	386	100%	3,748.20
pmed20	14	386	100%	3,743.11
pmed21	6	494	100%	4,041.78
pmed22	6	494	100%	4,050.36
pmed23	6	494	100%	4,037.83
pmed24	6	494	100%	4048.51
pmed25	6	494	100%	4,068.25
pmed26	3	597	100%	4,130.69
pmed27	3	597	100%	4,130.08
pmed28	3	597	100%	4,132.48
pmed29	3	597	100%	4,132.91
pmed30	3	597	100%	4,136.11
pmed31	2	698	100%	5,107.74
pmed32	2	698	100%	5,097.30
pmed33	2	698	100%	5,103.002
pmed34	2	698	100%	5,117.75
pmed35	1	799	100%	4,351.58
pmed36	1	799	100%	4,362.61
pmed37	1	799	100%	4,358.98
pmed38	1	899	100%	6,962.71
pmed39	1	899	100%	7,001.68
pmed40	1	899	100%	6,989.27
Geomean			-	-

Πίνακας Α'.12: Χρόνοι εκτέλεσης με λύτη

Πρόβλημα	Gurobi	BFS	DFS	BFSmyopic	DFSmyopic	DFSgap1%	DFSgap5%
pmed1	0.18	0.11	0.11	0.12	0.11	0.11	0.11
pmed2	0.36	0.45	0.59	0.43	0.46	0.09	0.10
pmed3	0.30	0.44	0.54	0.44	0.38	0.20	0.10
pmed4	0.18	0.07	0.07	0.07	0.07	0.07	0.07
pmed5	0.19	0.06	0.07	0.07	0.07	0.07	0.07
pmed6	5.11	48.45	65.55	6.13	8.34	7.60	0.89
pmed7	0.87	0.43	0.51	0.49	0.45	0.44	0.55
pmed8	0.83	0.31	0.36	0.36	0.34	0.33	0.37
pmed9	0.77	0.27	0.32	0.31	0.28	0.27	0.31
pmed10	0.84	0.28	0.30	0.33	0.29	0.28	0.31
pmed11	4.45	16.15	19.60	3.44	2.89	2.03	2.45
pmed12	3.86	41.23	32.82	2.29	1.93	1.35	1.76
pmed13	1.85	0.86	1.11	1.08	0.89	0.85	1.09
pmed14	1.80	2.52	5.67	166.43	66.50	27.92	0.94
pmed15	1.71	0.74	0.94	0.88	0.74	0.80	0.89
pmed16	14.69	919.17	657.34	76.92	34.32	18.18	6.36
pmed17	10.89	90.43	222.16	53.58	35.13	2.75	3.58
pmed18	4.80	7.54	10.00	14.06	11.29	3.03	2.50
pmed19	3.96	1.76	2.52	2.10	1.74	1.71	2.10
pmed20	3.60	1.74	2.50	2.08	1.92	1.72	2.17
pmed21	6.45	5.11	7.83	6.44	5.25	5.12	6.26
pmed22	119.53	2,717.56	1,960.40	202.47	191.04	141.00	8.70
pmed23	6.08	3.42	4.19	5.41	4.24	3.41	4.31
pmed24	5.46	3.53	4.43	4.46	4.40	3.56	4.46
pmed25	4.90	3.36	4.36	4.24	3.78	3.45	4.15
pmed26	120.51	3,534.48	3,603.43	92.40	78.45	30.15	12.76
pmed27	25.08	615.88	3,603.61	122.28	76.51	7.98	10.02
pmed28	12.40	5.62	6.88	8.06	6.96	5.53	7.03
pmed29	8.80	5.25	6.29	6.51	6.50	5.21	6.57
pmed30	7.20	4.95	5.86	5.97	5.99	5.18	5.93
pmed31	59.97	3,595.20	2,499.56	193.91	135.28	13.93	16.97
pmed32	30.78	579.05	699.60	64.43	122.85	14.73	18.49
pmed33	14.93	11.29	9.45	14.52	9.21	9.05	11.55
pmed34	16.03	9.81	8.40	12.44	9.53	7.85	9.93
pmed35	163.23	3,601.51	3,602.17	203.35	151.17	41.37	23.55
pmed36	2,113.73	3,601.60	3,602.10	9,612.27	4,561.91	3,295.23	21.26
pmed37	20.80	16.16	14.42	35.54	29.87	23.59	14.59
pmed38	603.23	3,601.74	3,601.81	4,769.68	3,035.33	2,393.62	36.34
pmed39	547.66	3,601.31	3,601.98	1,341.80	900.95	21.89	27.29
pmed40	31.35	22.30	17.79	24.07	21.47	17.35	24.12
Geomean	7.17	12.85	15.16	8.73	7.23	3.97	2.65

Πίνακας Α.13: Χρόνοι εκτέλεσης χωρίς λύτη

Πρόβλημα	Myopic	Neighborhood	Neighborhood Myopic	Tabu	Exchange	Subdrop
pmed1	0.0003	0.07	0.06	0.04	0.65	0.03
pmed2	0.0005	0.13	0.12	0.08	1.58	0.10
pmed3	0.001	0.13	0.13	0.08	3.16	0.10
pmed4	0.01	0.25	0.13	0.09	5.65	0.38
pmed5	0.001	0.42	0.21	0.10	13.37	3.23
pmed6	0.001	0.50	0.51	0.51	4.95	0.10
pmed7	0.002	0.95	0.95	0.52	12.12	0.35
pmed8	0.004	1.89	1.88	0.57	45.78	1.34
pmed9	0.01	3.77	1.89	0.65	167.55	5.39
pmed10	0.01	12.62	6.29	0.78	614.73	15.78
pmed11	0.002	1.65	2.44	1.73	10.99	0.23
pmed12	0.005	3.20	3.16	1.70	40.77	0.77
pmed13	0.01	9.50	9.41	1.88	338.39	6.50
pmed14	0.02	18.96	9.44	2.19	2,044.44	26.52
pmed15	0.04	48.35	31.35	2.59	3,600.60	76.87
pmed16	0.004	3.88	3.80	3.93	40.69	0.39
pmed17	0.01	7.48	7.39	3.96	101.34	1.34
pmed18	0.03	29.58	29.31	4.36	1,494.19	19.87
pmed19	0.05	118.71	87.75	5.11	3,608.40	81.37
pmed20	0.08	99.12	97.24	6.08	361.06	235.21
pmed21	0.01	7.36	7.32	7.64	27.49	0.61
pmed22	0.01	14.35	14.28	7.61	221.57	2.07
pmed23	0.06	106.63	106.03	8.58	3,602.69	47.78
pmed24	0.11	71.49	141.31	9.96	3,609.05	197.17
pmed25	0.16	120.57	235.86	11.89	3,611.28	578.09
pmed26	0.01	13.00	6.30	13.21	95.24	0.86
pmed27	0.02	24.63	12.31	13.03	509.93	2.96
pmed28	0.10	74.23	145.52	14.76	3,610.29	97.32
pmed29	0.18	445.34	580.95	17.05	3,632.70	401.55
pmed30	0.28	486.92	484.55	20.47	3,614.50	1,163.19
pmed31	0.01	21.02	11.38	20.86	78.51	1.17
pmed32	0.03	39.45	38.77	20.62	584.06	4.01
pmed33	0.15	271.97	268.66	23.10	3,618.23	178.48
pmed34	0.29	1,358.07	268.19	26.96	3,607.36	737.97
pmed35	0.02	29.89	29.73	31.19	229.49	1.53
pmed36	0.03	64.21	57.84	30.57	851.69	5.21
pmed37	0.23	722.23	455.16	35.05	3,621.63	302.92
pmed38	0.02	82.96	42.54	44.35	513.12	1.94
pmed39	0.04	298.49	81.87	43.56	1,226.82	6.57
pmed40	0.33	1,375.33	364.10	49.66	3,605.54	483.04
Geomean	0.02	17.03	13.51	3.92	259.96	6.96

Πίνακας Α'.14: Αποκλίσεις αλγορίθμων

Πρόβλημα	DFS1%	DFS5%	Myopic	Neighbor hood	Neighbor Myopic	Tabu	Exchange	Subdrop
pmed1	0%	0%	1.24%	35.18%	2.42%	1.24%	0%	20.97%
pmed2	0.61%	0.61%	0.61%	33.23%	7.6%	0.49%	0.29%	17.74%
pmed3	0.07%	3.51%	3.51%	55.11%	6.26%	2.52%	0%	24.24%
pmed4	0%	0%	1.78%	42.88%	1.78%	1.78%	0.4%	37.31%
pmed5	0%	0%	1.7%	63.76%	1.7%	0.52%	0%	43.99%
pmed6	0%	2.59%	2.59%	38.62%	8.38%	2.59%	0%	16.45%
pmed7	0%	0%	0.27%	35.5%	4.3%	0.25%	0.25%	20.58%
pmed8	0%	0%	0.61%	39.39%	6.86%	0.45%	0.27%	31.09%
pmed9	0%	0%	3.91%	52.49%	3.91%	2.49%	0.69%	38.08%
pmed10	0%	0%	3.19%	78.41%	5.42%	3.19%	0.64%	68.61%
pmed11	0.32%	0.32%	0.32%	35.72%	7.71%	0.23%	0%	10.32%
pmed12	0.26%	0.26%	0.26%	38.75%	2.41%	0.26%	0%	19.05%
pmed13	0%	0%	2.13%	50.3%	3.18%	1.94%	0%	38.09%
pmed14	0.94%	1.52%	1.52%	49.9%	1.52%	1.45%	0.1%	39.49%
pmed15	0%	0%	1.85%	62.93%	1.79%	1.79%	0.52%	61.42%
pmed16	0.04%	0.86%	0.86%	49.91%	5.98%	0.86%	0%	12.62%
pmed17	0.29%	0.29%	0.26%	36.98%	1.49%	0.29%	0%	23.63%
pmed18	0.06%	1.33%	1.33%	40.8%	1.91%	0.56%	0.04%	30.4%
pmed19	0%	0%	1.9%	50.93%	3.87%	1.83%	0.46%	43.37%
pmed20	0%	0%	4.3%	60.7%	4.97%	4.14%	2.63%	61.49%
pmed21	0%	0%	0%	34.17%	4.11%	0%	0%	22.2%
pmed22	0.08%	1.86%	1.06%	38.64%	2.55%	1.06%	1.05%	26.51%
pmed23	0%	0%	1.62%	41.76%	6.67%	1.62%	0%	37.35%
pmed24	0%	0%	1.62%	51%	2.6%	1.62%	0.88%	42.79%
pmed25	0%	0%	3.72%	73.41%	4.27%	3.67%	3.45%	64.06%
pmed26	0.25%	1.77%	1.77%	51.95%	1.77%	1.77%	0.07%	21.34%
pmed27	0.69%	0.69%	0.69%	42.16%	0.69%	0.64%	0%	29.05%
pmed28	0%	0%	1.8%	51.24%	1.82%	1.49%	0.53%	35.53%
pmed29	0%	0%	2.34%	46.95%	2.8%	2.18%	2.27%	43.16%
pmed30	0%	0%	2.41%	64.2%	2.71%	2.41%	2.26%	59.88%
pmed31	0%	0%	0%	25.87%	0%	0%	0%	18.72%
pmed32	0.37%	0.37%	0.37%	30.15%	3.71%	0.23%	0%	25.73%
pmed33	0%	0%	2.09%	45.11%	1.98%	1.98%	1.4%	34.64%
pmed34	0%	0%	2.79%	48.92%	2.79%	2.66%	2.66%	42.65%
pmed35	0.05%	0.06%	0.06%	35.07%	2.93%	0.06%	0%	18.73%
pmed36	0%	0.2%	0.2%	25.13%	1.28%	0.13%	0%	24.52%
pmed37	0.3%	1.21%	1.21%	145.8%	1.42%	0.99%	1.21%	36.37%
pmed38	0%	0.84%	0.84%	40.9%	5.86%	0.84%	0%	21.25%
pmed39	0.3%	0.3%	0.3%	37.26%	2.54%	0.3%	0%	22.89%
pmed40	0%	0%	1.21%	40.97%	1.21%	1.21%	1.13%	32.62%
Geomean	0.11	0.44	1.51	47.98	3.43	1.34	0.58	32.97

Πίνακας Α'.15: Συγκεντρωτικά αποτελέσματα αλγορίθμων

Αλγόριθμοι	Απόκλιση	Χρόνος Εκτέλεσης(sec)
Gurobi	0%	7.17
BFS	0.037%	12.85
DFS	0.042%	15.16
BFS myopic	0%	8.73
DFS myopic	0%	7.23
DFS myopic gap 1%	0.11%	3.97
DFS myopic gap 5%	0.44%	2.65
Myopic	1.51%	0.02
Neighborhood	47.98%	17.03
Neighborhood myopic	3.43%	13.51
Tabu	1.34%	3.92
Exchange	0.58%	259.96
Subdrop	32.97%	6.94