ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΥΡΕΤΙΚΕΣ ΜΕΘΟΔΟΙ ΓΙΑ ΤΗΝ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΧΩΡΟΘΕΤΗΣΗΣ ΕΓΚΑΤΑΣΤΑΣΕΩΝ

Διπλωματική Εργασία

Δέσποινα Παναγιωτοπούλου

Θεσσαλονίκη , 11/2018

ΕΥΡΕΤΙΚΕΣ ΜΕΘΟΔΟΙ ΓΙΑ ΤΗΝ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΧΩΡΟΘΕΤΗΣΗΣ ΕΓΚΑΤΑΣΤΑΣΕΩΝ

Δέσποινα Παναγιωτοπούλου

Πτυχίο Μαθηματικών, Πανεπιστήμιο Ιωαννίνων, 2013

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής

Άγγελος Σιφαλέρας

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 06/11/2018

Ονοματεπώνυμο 1	Ονοματεπώνυμο 2	Ονοματεπώνυμο 3
....	....	....

Δέσποινα Παναγιωτοπούλου

....

Ευχαριστίες

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στα πλαίσια του Μεταπτυχιακού Προγράμματος «Υπολογιστικές Μέθοδοι και Εφαρμογές» του τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας, κατά τα έτη 2015-2018 και αποτελεί κομβικό σημείο στην μεταπτυχιακή μου πορεία, καθώς μου δόθηκε η ευκαιρία να γνωρίσω, μέσα από προσωπική μελέτη, την επιστήμη της Πληροφορικής και να ασχοληθώ με την ανάπτυξη αλγορίθμων. Για την επίτευξη αυτών των στόχων, καθοριστικός ήταν ο ρόλος κάποιων ανθρώπων.

Την εποπτεία της εργασίας αυτής είχε ο Επίκουρος Καθηγητής του τμήματος Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας κ. Αγγελος Σιφαλέρας, τον οποίο και θέλω να ευχαριστήσω θερμά τόσο για την επιστημονική βοήθεια που μου παρείχε όσο και για την άψογη συνεργασία μας και την εμπιστοσύνη που έδειξε στο πρόσωπό μου από το μεταπτυχιακό μάθημα έως και την τελική διεξαγωγή της μεταπτυχιακής εργασίας. Η αγάπη και η εμψύχωση της οικογένειας μου και η θερμή υποστήριξη των φίλων μου ήταν σημαντικοί παράμετροι επιτυχίας αυτής της προσπάθειας.

Στη Θένα

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται τη διερεύνηση του προβλήματος διαμέσων (p-median problem) και την παρουσίαση ευρετικών μεθόδων για την επίλυσή του. Ιδιαίτερη προσοχή δίνεται στο p-median πρόβλημα στο πεδίο της χωροθέτησης μονάδων, με την επίλυση αυτού του NP-hard προβλήματος να παραμένει μια πολύ ενδιαφέρουσα πρόκληση στον τομέα της επιχειρησιακής έρευνας, ειδικά όταν αυξάνεται η διάσταση του προβλήματος. Αρχικά, πραγματοποιήθηκε μια βιβλιογραφική ανασκόπηση του p-median προβλήματος, όπως επίσης και των μεθοδολογιών επίλυσης αυτού του προβλήματος. Ακόμη, ακολούθησε η μαθηματική μοντελοποίηση του προβλήματος σε Python με χρήση του λύτη βελτιστοποίησης Gurobi ώστε να κατανοηθεί τόσο το πρόβλημα όσο και οι περιορισμοί του.

Η διπλωματική εργασία στοχεύει στην ανάπτυξη ενός αποδοτικού αλγορίθμου στη γλώσσα προγραμματισμού Python, ο οποίος να μπορεί να επιλύσει πραγματικά προβλήματα χωροθέτησης τα οποία προκύπτουν από δίκτυα εφοδιαστικής αλυσίδας. Η χωροθέτηση των εγκαταστάσεων είναι σημαντική παράμετρος στην εξοικονόμηση χρόνου παραγωγής και κόστους εγκαταστάσεων. Στην εργασία το πρόβλημα της χωροθέτησης διατυπώνεται ως πρόβλημα κατανομής των εγκαταστάσεων. Συνεπώς σχεδιάστηκαν και υλοποιήθηκαν ένας άπληστος αλγόριθμος αρχικοποίησης και μία μεθευρετική μεθοδολογία Αναζήτηση Μεταβλητής Γειτνίασης - Variable Neighborhood Search, μια σχετικά πρόσφατα αναπτυγμένη μεθευρετική μέθοδος που έχει αποδειχθεί αποτελεσματική για πολλά προβλήματα βελτιστοποίησης.

Τέλος, πραγματοποιήθηκε μία συγκριτική υπολογιστική μελέτη που περιλαμβάνει την εφαρμογή του αλγορίθμου πάνω σε διαθέσιμα μετροπροβλήματα (benchmark problems) και αξιολογήθηκε η αποτελεσματικότητά του σε σύγκριση με τις ακριβής βέλτιστες τιμές των αντικειμενικών συναρτήσεων αυτών των προβλημάτων.

Λέξεις Κλειδιά: Ευρετικές Μέθοδοι, Αναζήτηση Μεταβλητής Γειτνίασης, Επιχειρησιακή Έρευνα, Πρόβλημα Διαμέσων, Βελτιστοποίηση, Πρόβλημα Εγκαταστάσεων

Abstract

This thesis consists of a detailed explanation about the p-median problem and presents a solution for it using a heuristic method. Special consideration is given to the version of the p-median problem that arises in the field of location. The problem of facility location is NP-hard problem and a quite challenging one in the field of Operational Research. A bibliographical research about the problem and methodologies to solve it has been conducted. Additionally, we explicitly state the constraints needed for the p-median problem and then we program the mathematical model in Python utilizing the Gurobi optimization solver.

The thesis purpose is to develop an efficient algorithm for solving the p-median problem in Python programming language. The algorithm can solve real instances of the facility location problem that emerge from supply chain networks. The location of the facility is a crucial parameter both in saving production time and facility costs. The location problem is being formulated as a problem of allocation the facilities. A greedy construction method and a metaheuristic approach of Variable Neighborhood Search(VNS) has been designed and programmed. VNS is the state of art in metaheuristic methods and so far has been used to solve efficiently a lot of optimization problems.

To sum up, a comparative computational research has been carried out, which includes the use of the algorithm on available benchmark problems. Finally the effectiveness of the algorithm has been assessed by comparing the exact optimal values of those problems' objective function with the optimal value of the algorithm.

Keywords: Heuristic methods, Variable Neighborhood Search, Operations Research, Median Problem, Optimization, Facility Location

ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος Πινάκων

Κατάλογος Σχημάτων

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Εισαγωγή στη βελτιστοποίηση

Για να αντιμετωπίσουμε προβλήματα, αναπτύσσουμε μαθηματικά μοντέλα, τα οποία αναπαριστούν τα πραγματικά αυτά προβλήματα που θέλουμε να επιλύσουμε. Στα μοντέλα ορίζουμε μεταβλητές απόφασης (decision variables) που αντιστοιχούν σε συγκεκριμένες αποφάσεις του εκάστοτε εξεταζόμενου προβλήματος. Μέσω των μεταβλητών αυτών είμαστε σε θέση να εκφράσουμε τους πραγματικούς λογικούς περιορισμούς του προβλήματος μετατρέποντας τους σε αντίστοιχους μαθηματικούς περιορισμούς (constraints). Παράλληλα, διαμορφώνουμε και την αντικειμενική συνάρτηση (objective function) που εκφράζει το μέγεθος απόδοσης το οποίο θέλουμε να βελτιστοποιήσουμε (π.χ. κόστος, κέρδος, κτλ.). Το σύνολο των περιορισμών του προβλήματος ορίζει ένα σύνολο λύσεων, το οποίο ονομάζεται εφικτό (feasible). Με αυτό τον τρόπο, προκύπτει ένα σύστημα ανισοτήτων/εξισώσεων με αγνώστους τις μεταβλητές απόφασης. Η λύση ονομάζεται εφικτή όταν ικανοποιεί όλους τους περιορισμούς του προβλήματος και μη εφικτή όταν παραβιάζεται έστω και ένας από αυτούς. Μεταξύ όλων των εφικτών λύσεων θα πρέπει να επιλεγεί εκείνη η λύση που βελτιστοποιεί την τιμή της αντικειμενικής συνάρτησης. Μια τέτοια λύση ονομάζεται βέλτιστη (optimal), αλλά δεν είναι απαραίτητα και μοναδική. Η διαδικασία κατά την οποία αναζητούμε την καλύτερη λύση (με βάση κάποια προδηλωμένη μέθοδο απόδοσης) μεταξύ ενός συνόλου εφικτών λύσεων, ονομάζεται Βελτιστοποίηση.

Η Βελτιστοποίηση μπορεί να εκληφθεί και ως μία διαδικασία εύρεσης και λήψης βέλτιστων αποφάσεων (decision making). Στα μαθηματικά, διατυπώνεται ένα πρόβλημα βελτιστοποίησης σαν πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης μιας ή περισσοτέρων γραμμικών / μη γραμμικών συναρτήσεων (κριτήρια βελτιστοποίησης) αγνώστων πραγματικών μεταβλητών των οποίων το πεδίο τιμών οριοθετείται έμμεσα από γραμμικούς / μη γραμμικούς περιορισμούς των μεταβλητών αυτών. Οι άγνωστες μεταβλητές μοντελοποιούν το αντικείμενο απόφασης του προβλήματος και ονομάζονται μεταβλητές απόφασης. Στην ελαχιστοποίηση / μεγιστοποίηση συναρτήσεων μπορούν να χρησιμοποιηθούν αναλυτικές και αλγεβρικές μέθοδοι για τον ακριβή ορισμό ελάχιστων (ή μέγιστων) σημείων, ενώ στη μελέτη συναρτήσεων πολλών μεταβλητών χρησιμοποιούνται κυρίως αριθμητικές μέθοδοι για έναν προσεγγιστικό ορισμό ελάχιστων / μέγιστων σημείων. Το πρόβλημα χωροθέτησης μονάδων, γνωστό ως ανάλυση τοποθεσίας ή πρόβλημα διαμέσων, είναι ένα κομμάτι της επιχειρησιακής έρευνας που ασχολείται με την βέλτιστη τοποθέτηση των εγκαταστάσεων για την ελαχιστοποίηση του κόστους μεταφοράς υπό περιορισμούς. Στα εφαρμοσμένα μαθηματικά η βελτιστοποίηση είναι μέρος της επιχειρησιακής έρευνας και αναφέρεται στην αναζήτηση βέλτιστων παραμέτρων ενός περίπλοκου συστήματος. Ένα πρόβλημα βελτιστοποίησης αντιμετωπίζεται σαν πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης μιας αντικειμενικής συνάρτησης μίας μεταβλητής ή πολλών μεταβλητών.

Ας θεωρήσουμε ένα πρόβλημα βελτιστοποίησης.

$$\min_{\forall x \in X} f(x)$$

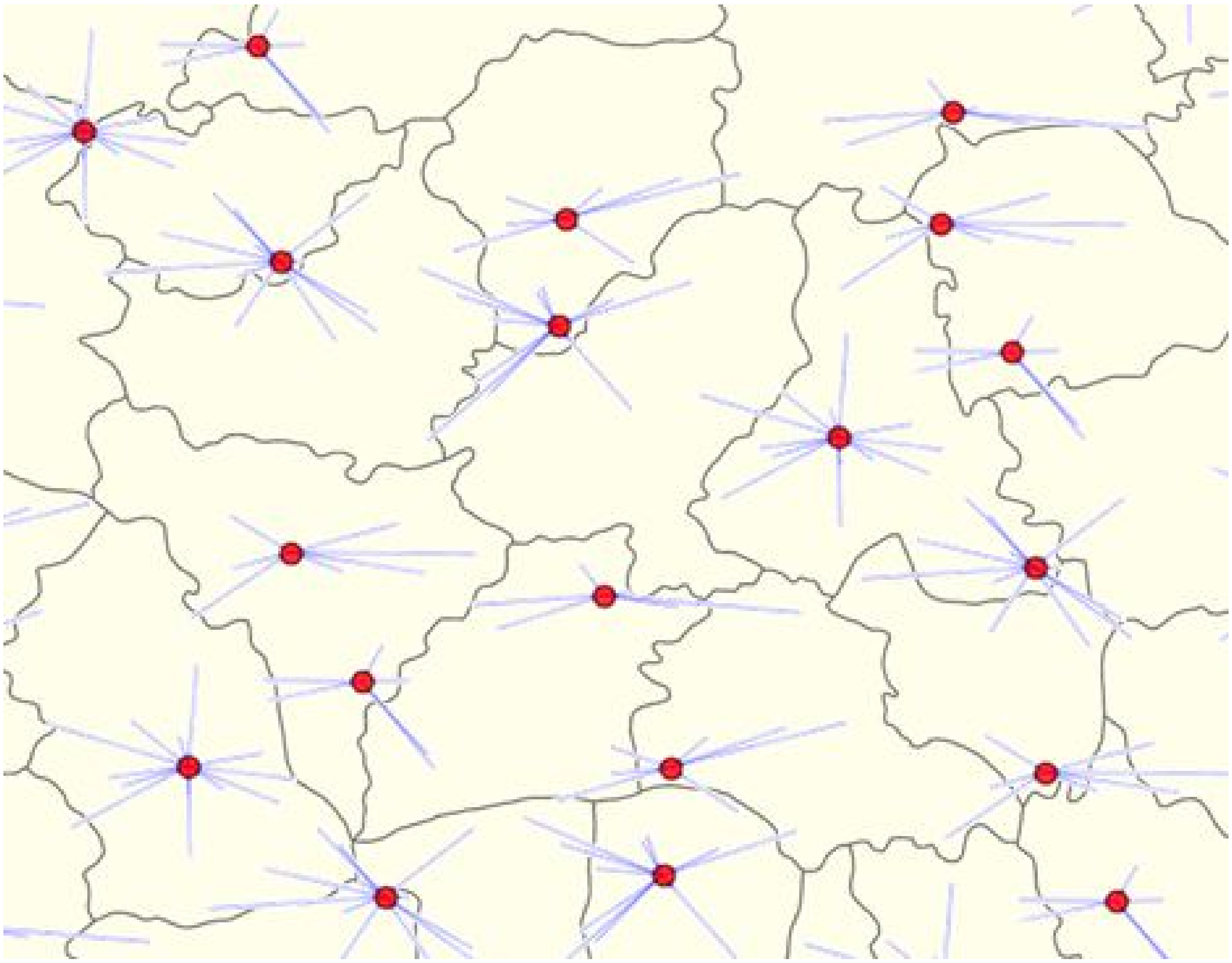
όπου $f(x)$ είναι η αντικειμενική συνάρτηση που πρέπει να ελαχιστοποιηθεί και x το σύνολο των εφικτών λύσεων. Xείναι η βέλτιστη εάν $f(x^*) \leq f(x)$,$\forall x \in X$

Ένας ακριβής αλγόριθμος για το παραπάνω πρόβλημα , θα βρει τη βέλτιστη λύση x^* ή θα καταλήξει ότι δεν υπάρχει εφικτή λύση ($Q = \emptyset$), ή ότι το πρόβλημα είναι απεριόριστο, (Mladenović, Brimberg, Hansen, and Moreno-Pérez 2007). Σε περίπτωση που δεν βρεθεί ακριβής αλγόριθμος, πληθώρα ευρετικών, μεθευρετικών και υβριδικών μεθόδων έχουν αναπτυχθεί έτσι ώστε χαλαρώνοντας τα κριτήρια της βελτιστοποίησης να έρθουν αρκετά κοντά σε ικανοποιητικές λύσεις και σε εύλογο χρόνο, τέτοιες μέθοδοι αναφέρονται στο Κεφάλαιο 2.

1.2 Εισαγωγή στον ακέραιο προγραμματισμό

Ο ακέραιος προγραμματισμός (integer programming) περιλαμβάνει όλα τα προβλήματα στα οποία οι μεταβλητές απόφασης μπορούν να πάρουν μόνο ακέραιες τιμές. Ένα πρόβλημα ακέραιου προγραμματισμού μπορεί να είναι γραμμικό ή μη γραμμικό. Σε περίπτωση που οι μεταβλητές απόφασης του προβλήματος περιορίζονται μόνο στις τιμές 0 ή 1, τότε βρισκόμαστε σε μια ειδική κατηγορία προβλημάτων ακέραιου προγραμματισμού, τον δυαδικό ακέραιο προγραμματισμό (binary integer programming).

1.3 Εισαγωγή στο πρόβλημα της χωροθέτησης εγκαταστάσεων (facility location)



Σχήμα **1.3.1**: Χωροθέτηση εγκαταστάσεων

Έστω ένα σύνολο n σημείων όπου μπορούν να δημιουργηθούν εγκαταστάσεις εξυπηρέτησης πελατών και ένα σύνολο m πελατών. Η δημιουργία εγκαταστάσεων στο σημείο j κοστίζει c_j , j = 1,...,n. Το κόστος εξυπηρέτησης του πελάτη i από την εγκατάσταση j είναι h_{ij}. Θέλουμε να βρούμε σε ποιά σημεία θα πρέπει να δημιουργηθούν εγκαταστάσεις και πώς θα πρέπει να γίνει η ανάθεση των πελατών στις εγκαταστάσεις έτσι ώστε να εξυπηρετηθούν όλοι οι πελάτες στο ελάχιστο δυνατό κόστος.

Ορίζουμε δυαδικές μεταβλητές απόφασης $y_j = 1$ αν δημιουργηθεί εγκατάσταση στο σημείο j και 0 αν δεν δημιουργηθεί εγκατάσταση.

Επίσης, ορίζουμε δυαδικές μεταβλητές απόφασης $x_{ij} = 1$ αν ο πελάτης στο σημείο i εξυπηρετηθεί από την εγκατάσταση στο σημείο j και 0 αν όχι. Το πρόβλημα μορφοποιείται ως εξής:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij} * x_{ij} + \sum_{j=1}^{n} c_j * y_j \tag{1}$$

υπό τους περιορισμούς

- $\sum_{j=1}^{n} x_{ij} = 1$, γιαi = 1,...,m
- $x_{ij} < y_j, \forall i = 1, ..., m, \forall j = 1, ..., n$
- $x_{ij}, y_j, \in \{0,1\} \forall i = 1, ..., m, \forall j = 1, ..., n$

Η πρώτη ομάδα περιορισμών εξασφαλίζει ότι κάθε πελάτης θα εξυπηρετηθεί από ένα σημείο εξυπηρέτησης και η δεύτερη ομάδα ότι για να εξυπηρετηθεί κάποιος πελάτης από κάποιο σημείο θα πρέπει στο αντίστοιχο σημείο να δημιουργηθούν εγκαταστάσεις.

1.4 Κατηγοριοποίηση του προβλήματος χωροθέτησης εγκαταστάσεων

Το πρόβλημα τοποθεσιών είναι ένα πεδίο της επιχειρησιακής έρευνας που περιλαμβάνει πολλά μαθηματικά μοντέλα. Ένα πρόβλημα ανήκει στα προβλήματα τοποθεσιών, αν πρέπει να παρθεί κάποια απόφαση σχετικά με τη θέση νέων εγκαταστάσεων, λαμβάνοντας υπόψη την απόσταση μεταξύ των εγκαταστάσεων αλλά και κάποιες επιπρόσθετες παραπέτρους. Οι (Mladenović, Brimberg, Hansen, and Moreno-Pérez 2007) χωρίζουν τα μοντέλα τοποθεσίας σε τρεις βασικές ομάδες:

- σε συνεχή μοντέλα ($Q \subseteq R^q$),
- σε διακριτά μοντέλα (X είναι πεπερασμένο)
- σε μοντέλα δικτύου (το X αποτελεί ένωση πεπερασμένου με γραμμικά και συνεχή σύνολα)

Μια άλλη κατηγοριοποίηση των προβλημάτων τοποθεσιών είναι η κατηγοριοποίηση ως προς τη διάμεση τιμή (minisum) ή ως προς το κέντρο (minimax). Τέλος, τα μοντέλα τοποθεσιών μπορούν να είναι ντετερμινιστικά ή στοχαστικά (Owen and Daskin 1998), γραμμικά ή μη γραμμικά, μονοκριτηριακά ή πολυκριτηριακά, καθώς και πολλά ακόμη μοντέλα όπως αυτό το 1988 των ερευνητών Love, Morris και Wesolowsky έως το 2002 των Drezner και Hamacher που αναφέρουν οι (Mladenović, Brimberg, Hansen, and Moreno-Pérez 2007).

1.5 Σκοπός εργασίας

Η παρούσα διπλωματική εργασία πραγματεύεται τη διερεύνηση του προβλήματος διαμέσων (p-median problem) και την παρουσίαση ευρετικών μεθόδων για την επίλυσή του. Η παρούσα θέση εστιάζει ιδιαίτερα στην εκδοχή του p-median προβλήματος στο πεδίο της χωροθέτησης μονάδων, με την επίλυση αυτού του NP-hard προβλήματος να παραμένει μια πολύ ενδιαφέρουσα πρόκληση στον τομέα της επιχειρησιακής έρευνας, ειδικά όταν αυξάνεται η διάσταση του προβλήματος.

Η εργασία στοχεύει στην κατανόηση του p-median προβλήματος μέσα από μια βιβλιογραφική ανασκόπηση αλλά και την μαθηματική μοντελοποίηση του προβλήματος και επίλυση του με χρήση του λύτη βελτιστοποίησης Gurobi. Σε δεύτερη φάση παρουσιάζεται μια μεθευρετική Αναζήτηση Μεταβλητής Γειτνίασης (Variable Neighborhood Search) μεθοδολογία.

Στόχος της διπλωματικής εργασίας είναι ο αλγόριθμος που θα αναπτυχθεί να είναι αποδοτικός σε πραγματικά προβλήματα χωροθέτησης τα οποία προκύπτουν από δίκτυα εφοδιαστικής αλυσίδας, στη γλώσσα προγραμματισμού Python. Τέλος, θα πραγματοποιηθεί μία συγκριτική υπολογιστική μελέτη που θα περιλαμβάνει την εφαρμογή του αλγορίθμου πάνω σε διαθέσιμα μετροπροβλήματα (benchmark problems) και θα αξιολογεί την αποτελεσματικότητά του ως προς την ποιότητα του σφάλματος αλλά και ως προς τον χρόνο εύρεσης των λύσεων.

1.6 Συνεισφορά

Η συνεισφορά της παρούσας εργασίας δύναται να συνοψισθεί ως ακολούθως:

- Θεωρητικό υπόβαθρο του προβλήματος διαμέσων.
- Παρουσίαση των μεθόδων επίλυσης του προβλήματος διαμέσων.
- Παρουσίαση του άπληστου αλγορίθμου αρχικοποίησης.
- Παρουσίαση του μεθευρετικού πλαισίου της Αναζήτησης Μεταβλητής Γειτνίασης.
- Ανάλυση του υλοποιηθέντος αλγορίθμου στα συστατικά του μέρη, με τη χρήση μικρού μεγέθους μετροπροβλημάτων.
- Αξιολόγηση του υλοποιηθέντος αλγορίθμου μέσω της υπολογιστικής μελέτης στα μετροπροβλήματα της OR library.
- Σύγκριση αποτελεσμάτων του μεθευρετικού αλγορίθμου με την βέλτιστη λύση που απορρέει από την δημιουργία και υλοποιήση του μοντέλου με χρήση του Gurobi λύτη βελτιστοποίησης.

1.7 Διάρθρωση της Μελέτης

Στην παρούσα υποενότητα σκιαγραφείται η δομή, η οποία ακολουθείται για την ανάπτυξη της θεματολογίας της παρούσας διπλωματικής εργασίας. Αρχικά, παρατίθενται ορισμένα βασικά ενημερωτικά για τον αναγνώστη στοιχεία και οι κύριοι λόγοι ανάληψης της παρούσας διπλωματικής. Στο επόμενο κεφάλαιο παρουσιάζεται το απαραίτητο θεωρητικό υλικό, με σκοπό την ενημέρωση του αναγνώστη, ώστε να επιτευχθεί η καλύτερη δυνατή κατανόηση του προβλήματος διαμέσων μέσα από την παρουσίαση του μαθηματικού μοντέλου. Έτσι, παρατίθεται ιστορικό υλικό σχετικά με το πρόβλημα διαμέσων και απαραίτητες πληροφορίες σχετικά με ευρετικές και μεθευρετικές μεθόδους που ανέπτυξαν διάφοροι ερευνητές, οι σύγχρονες τάσεις στις μεθόδους επίλυσης καθώς και μία συνοπτική περιγραφή της μαθηματικής μοντελοποίησης του προβλήματος διαμέσων με χρήση του λύτη βελτιστοποίησης Gurobi. Ιδιαίτερη προσοχή δίνεται στα βασικά δομικά μέρη της άπληστης ευρετικής μεθόδου αλλά και την περιγραφή της μεθευρετικής διαδικασίας Variable Neighborhood Search. Έπειτα περιγράφονται τα συστατικά μέρη του προς υλοποίηση μεθευρετικού αλγορίθμου, ενώ στο τελευταίο κεφάλαιο ελέγχεται ο αλγόριθμος μέσω μίας υπολογιστικής μελέτης για την ακρίβεια και την απόδοσή του σε 40 μετροπροβλήματα της βιβλιοθήκης OR library. Για την ακριβέστερη υπολογιστική μελέτη προηγήθηκε η ανάπτυξη του μαθηματικού μοντέλου του προβλήματος διαμέσου στο Gurobi λύτη βελτιστοποίησης και η εκτέλεσή του με τα προαναφερθέντα μετροπροβλήματα της OR library. Έπειτα ακολούθησε η σύγκριση των αποτελεσμάτων του μεθευρετικού αλγορίθμου με την βέλτιστη λύση που απορρέει από το μαθηματικό μοντέλο που εκτελέστηκε με χρήση του Gurobi λύτη βελτιστοποίησης.

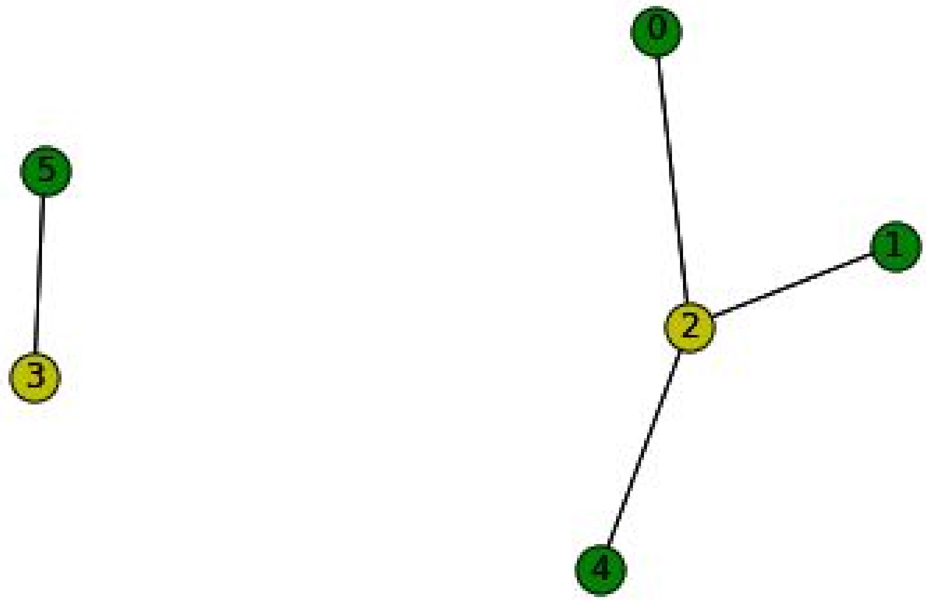
ΚΕΦΑΛΑΙΟ 2

Βιβλιογραφική Επισκόπηση

Η αναζήτηση και η επιλογή της βιβλιογραφίας που περιλαμβάνει η εργασία πραγματοποιήθηκε κυρίως απο έγκυρες ηλεκτρονικές πηγές. Καθώς το πρόβλημα διαμέσων χρονολογείται από τον 17ο αιώνα και χρησιμοποιείται ευρέως ακόμη και σήμερα, δεν θα επιθυμούσα να στηρίξω τη βιβλιογραφική έρευνα αποκλειστικά στις περιορισμένες δυνατότητες μερικών βιβλίων. Με θέμα αναζήτησης λοιπόν τις λέξεις κλειδιά που αναφέρονται παραπάνω, εντοπίστηκαν και αξιολογήθηκαν αρκετά σχετικά επιστημονικά άρθρα και βιβλία μέσα από έγκυρες ιστοσελίδες όπως (Springer), (Scholar), (Sciencedirect), κτλ.

2.1 Το πρόβλημα Διαμέσων

2.1.1 Ιστορικά Στοιχεία



Σχήμα **2.1.1**: P-median πρόβλημα, $p = 2$.

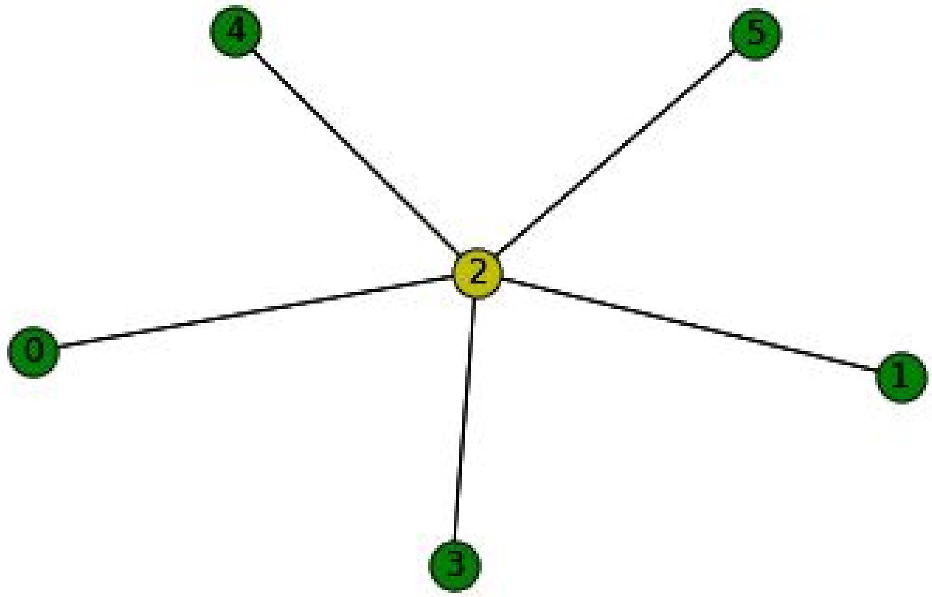
Η ιστορική αναδρομή για το πρόβλημα διαμέσων ξεκινάει γύρω στο 17ο αιώνα. Το κύριο κίνητρο για την παρούσα μελέτη είναι ένα διάσημο πρόβλημα, που φέρεται ότι διατυπώθηκε για πρώτη φορά το 1640 από τον Pierre de Fermat, και το λεγόμενο Συμπληρωματικό Πρόβλημα (CP). Μια πιο γενική διατύπωση του απλού αυτού προβλήματος εγκαταστάσεων αποδίδεται στους (Weber and Pick 1909), (Drezner and Hamacher 2001). Με μοναδικό κριτήριο βελτιστοποίησης να είναι η ελαχιστοποίηση του γεωμετρικού μέσου τριών μόνο σημείων, μια εκδοχή του προβλήματος χωροθέτησης, αναζητάται το κόστος μεταφοράς ανά απόσταση να είναι το ίδιο για όλους τους προορισμούς. Στη γεωμετρία, το πρόβλημα Weber, όπως ονομάζεται, είναι ένα από τα πιο διάσημα προβλήματα της θεωρίας εγκαταστάσεων. Απαιτεί την εύρεση ενός σημείου στο επίπεδο που ελαχιστοποιεί το άθροισμα του κόστους μεταφοράς των αποστάσεων από το σημείο αυτό έως τα σημεία προορισμού, όπου τα διαφορετικά σημεία προορισμού σχετίζονται με διαφορετικό κόστος ανά μονάδα απόστασης. Το πρόβλημα Weber γενικεύει τη γεωμετρική μέση τιμή, η οποία αναλαμβάνει το κόστος μεταφοράς ανά μονάδα απόστασης να είναι το ίδιο για όλα τα σημεία προορισμού. Στο πρόβλημα υπολογισμού του σημείου Fermat, υπολογίζεται η γεωμετρική μέση τιμή των τριών σημείων. Για το λόγο αυτό, καλείται μερικές φορές πρόβλημα Fermat-

Weber. Ο ιταλός φυσικός Evangelista Torricelli, περίπου το 1645, πρόσθεσε στο πρόβλημα Fermat μια ακόμη διάσταση έχοντας ως δεδομένα τρία σημεία σε ένα επίπεδο, και ψάχνοντας ένα τέταρτο σημείο, τέτοιο ώστε το άθροισμα των αποστάσεων του από τα άλλα τρία σημεία να είναι όσο το δυνατόν μικρότερο, το οποίο ο (Dörrie 2013) το επίλυσε γεωμετρικά. Αντίθετα, το πρόβλημα Weber διατυπώθηκε και επιλύθηκε γεωμετρικά από τον Thomas Simpson το 1750. Αριθμητικά απέδειξε το 1972 το πρόβλημα Fermat- Weber με τριγωνομετρία ο Luc-Normand Tellier, ενώ τελική και πιο γενικευμένη λύση έδωσαν το 1962 οι Kuhn and Kuenne, οι οποίοι μετέφεραν το πρόβλημα στην περίπτωση των πολυγώνων που έχουν περισσότερες από τρεις πλευρές, πράγμα που δεν συνέβει στις προηγούμενες αποδείξεις.

Όπως σε όλα τα ιστορικά ντοκουμέντα, έτσι και στο πρόβλημα διαμέσων η καταγωγή και η εξέλιξη του προβλήματος δεν είναι τελείως ξεκάθαρη. Σε αυτήν τη θέση θα ήταν χρήσιμο να αναφερθεί ότι οι Cavalieri (1647), Viviani (1659) και Martelli (1998) συνέβαλαν καθοριστικά στην ορθότητα της απόδειξης αυτού του απλού προβλήματος. Επιπλέον, πολλοί είναι αυτοί που πιστεύουν ότι η απόδειξη του θεωρήματος του Πτολεμαίου (περίπου μ.Χ. 150) και μια παρατήρηση από τον Heinen (1834) πρόσθεσαν ιδιαίτερη αξία στην τελική απόδειξη του προβλήματος (Jalal and Krarup 2003). Το 1909 ο Alfred Weber χρησιμοποίησε ένα παρόμοιο μοντέλο ελαχιστοποίησης κόστους μεταφοράς τριών σημείων που απεικόνιζε βιομηχανικές περιοχές προς τους πελάτες τους (Weber and Pick 1909).

Η παραπάνω διατύπωση είναι ένα από τα πιο απλά συνεχή μοντέλα χωροθέτησης. Υπάρχουν πλέον πιο σύνθετα προβλήματα που εξετάζουν την τοποθέτηση πολλαπλών εγκαταστάσεων, με ποικίλους περιορισμούς σχετικά με τις τοποθεσίες των εγκαταστάσεων καθώς και πιο σύνθετα κριτήρια βελτιστοποίησης. Το πρόβλημα χωροθέτησης επικεντρώνεται στην επιλογή της τοποθέτησης των εγκαταστάσεων ώστε να ανταποκρίνονται καλύτερα στους περιορισμούς. Το πρόβλημα συχνά συνίσταται στην επιλογή μιας τοποθεσίας - εργοστάσιο που ελαχιστοποιεί τις σταθμισμένες αποστάσεις προμηθευτών-πελατών, όπου τα βάρη είναι αντιπροσωπευτικά της δυσκολίας μεταφοράς των υλικών. Η λύση στο πρόβλημα αυτό δίνει την βέλτιστη επιλογή κέρδους που εξυπηρετεί πιο αποτελεσματικά τις ανάγκες όλων των πελατών-καταναλωτών (Drezner and Hamacher 2001).

2.2 Ορισμός και περιγραφή του απλού p=1 προβλήματος διαμέσου



Σχήμα **2.2.2**: P-median πρόβλημα, $p = 1$.

Το πρόβλημα p διαμέσων έχει ιδιαίτερο νόημα όταν αναφερόμαστε στη θεωρία και στα διακριτά μοντέλα εγκαταστάσεων. Ως ένα NP-hard πρόβλημα είναι υπολογιστικά δύσκολο, για αυτό μια ποικιλία από ευρετικές και μεθευρετικές μεθόδους έχουν αναπτυχθεί ώστε να προσεγγίσουν την λύση του σε εύλογο χρονικό διάστημα (Kariv and Hakimi 1979), ωστόσο για ένα γενικό διάγραμμα η λυση πραγματοποιείται σε πολυωνυμικό χρόνο σε ένα δέντρο. Ο Goldman πρότεινε ένα αλγόριθμο, ο οποίος εκτελείται σε γραμμικό χρόνο με p=1, (Daskin and Maass 2015), και αποτέλεσε τον λόγο ονομασίας του προβλήματος διαμέσων, (Goldman 1971). Θεώρησε ότι αν κάθε κόμβος ενός δέντρου έχει τη μισή ή παραπάνω της συνολικής ζήτησης όλων των κόμβων στο δέντρο, τότε το βέλτιστο εντοπίζεται σε αυτόν τον κόμβο. Η απομάκρυνση από αυτόν τον κόμβο θα μετακινήσει την εγκατάσταση από τα σημεία ζήτησης, αυξάνοντας έτσι την τιμή της αντικειμενικής συνάρτησης.

Έστω λοιπόν τα σύνολα:

$I = 1, ..., i, ..., m$ σύνολο σημείων πιθαθών τοποθεσιών - εγκαταστάσεων

και

$J = 1, ..., j, ..., n$ σύνολο σημείων ζήτησης, με τη ζήτηση του κόμβου j να εκφράζεται ως d_j

και c_{ij} το κόστος ικανοποίησης της ζήτησης των πελατών j από την εγκατάστασης i.

Αν υποθέσουμε ότι κανένας κόμβος δεν ικανοποιεί τη μισή ή περισσότερο της συνολικής ζήτησης

$$D = \sum_{j \in J} d_j = 1 \text{ για } i = 1, ..., m,$$

δηλαδή δεν υπάρχει κατάλληλη εγκατάσταση, τότε καλείται ένας κόμβος που βρίσκεται στα φύλλα (leaf node), έστω με ζήτηση $(d_j)'$.

Ο ψευδοκώδικας εκτελεί τα ακόλουθα βήματα:

- Βήμα 1. Έστω $(d_j)' = d_j$ για όλα τα σημεία ζήτησης $j \in J$.
- Βήμα 2. Αν ένας κόμβος A από τα φύλλα συνδέεται με κάποιον άλλον B, πρόσθεσε την ζήτηση του A σε αυτήν του B μέσα από την ακμή (A,B). Αν η συνολική ζήτηση B είναι μεγαγαλύτερη ή ίση από την μισή της συνολικής $D/2$, τότε σταμάτα. Ο κόμβος B είναι ο πρώτος διάμεσος. Αλλιώς, επανέλαβε το Βήμα 2. (Daskin and Maass 2015).

Εφόσον σε κάθε κόμβο εξετάζεται το πολύ μία φορά το δεύτερο Βήμα, συμπεραίνουμε ότι η πολυπλοκότητα χειρότερης περίπτωσης του πρώτου διαμέσου αλγορίθμου είναι $O(n)$. Θα ήταν χρήσιμο να αναφερθεί ότι για τον υπολογισμό της αντικειμενικής συνάρτησης θα πρέπει να λάβουμε υπόψιν τις αποστάσεις των κόμβων μεταξύ τους, κάτι που δεν υπολογίζεται στον παραπάνω αλγόριθμο. Ο αλγόριθμος των Kariv and Hakimi το 1979 είχε πολυπλοκότητα $On^2 * p^2$ για το πρόβλημα διαμέσων σε δέντρο (Kariv and Hakimi 1979), ενώ τέλος ο Tamir βελτίωσε το υπολογιστικό κόστος σε $Op * n^2$, (Tamir 1996).

2.3 Μέθοδοι επίλυσης

Στη βιβλιογραφία το πρόβλημα της χωροθέτησης εγκαταστάσεων απασχόλησε αρκετούς ερευνητές. Εφικτή ακριβής βέλτιστη λύση υπάρχει, αλλά ένα θεώρημα που οφείλεται στον Hakimi δείχνει ότι τουλάχιστον μία βέλτιστη λύση θα βρίσκεται εξ ολοκλήρου στους κόμβους του δικτύου, το οποίο απλοποιεί το πρόβλημα σε πολλές περιπτώσεις. Το πρόβλημα διαμέσων ανήκει στα NP-hard προβλήματα

(Papadimitriou and Steiglitz 1982), η χρήση μεθόδων βελτιστοποίησης περιορίζεται μόνο σε μικρών διαστάσεων προβλήματα, ενώ όσο ανεβαίνει η διάσταση χρησιμοποιούνται ευρετικοί, μεθευρετικοί ή υβριδικοί αλγόριθμοι, δηλαδή αλγόριθμοι που συνδυάζουν ευρετικές και ακριβείς μεθόδους, με τους οποίους επιδιώκεται η εύρεση ικανοποιητικών λύσεων.

Μια τεχνική επίλυσης προβλημάτων θεωρείται ευρετική όταν αναζητά λύσεις που μπορούν να επιτευχθούν διαθέτοντας λογική υπολογιστική ισχύ, δεν υπάρχει όμως εγγύηση ότι οι λύσεις που παράγονται είναι βέλτιστες. Οι ευρετικές τεχνικές στηρίζονται στην υπόθεση ότι η αποδοτική λύση ορισμένων συνδυαστικών προβλημάτων απαιτεί περισσότερη ευελιξία από ότι μπορεί να επιτευχθεί με τεχνικές που αποδεδειγμένα παρουσιάζουν ιδιότητες σύγκλισης. Σε κάποιο βαθμό μπορούν να θεωρηθούν ως αλγόριθμοι αναζήτησης που εφαρμόζουν επαναληπτικά ένα σκεπτικό επιλογής για την συνέχεια της αναζήτησης. Αυτό μπορεί να συμβαίνει μέσω κάποιας απλής λογικής αλληλουχίας ενεργειών καθώς και με τεχνικές πειραματισμού και διόρθωσης (trial and error).

Ορισμένες ευρετικές τεχνικές που χρησιμοποιούνται για την εύρεση καλών λύσεων σε συγκεκριμένα προβλήματα είναι ανεξάρτητες από τα ίδια τα προβλήματα που αντιμετωπίζουν. Οι τεχνικές αυτές ονομάζονται μεθευρετικές και πολλές από αυτές έχουν εμπνευστεί από μηχανισμούς που παρατηρούνται στην φύση. Οι μεθευρετικές τεχνικές μπορούν να ομαδοποιηθούν με βάση τον τρόπο με τον οποίο εξετάζουν τις νέες πιθανές λύσεις. Οι ευρετικές λύσεις μπορούν να παρέχουν καλές ή ακόμη και βέλτιστες λύσεις σε εύλογο χρονικό διάστημα, αλλά η αξιολόγηση της ποιότητας ορισμένων από αυτές τις προσεγγίσεις δεν είναι απλή. Στις ευρετικές προσεγγίσεις η αξιοποίηση των χαρακτηριστικών του προβλήματος (τα κριτήρια και η σειρά με την οποία επιλέγονται οι υποψήφιες λύσεις) αλλά και η επαλήθευση των περιορισμών παίζουν κύριο ρόλο. Βέβαια ενδέχεται για ορισμένες τιμές των παραμέτρων κάποιος ευρετικός αλγόριθμος να παρέχει καλύτερες λύσεις από κάποιον άλλον. Ως εκ τούτου συχνά οι ερευνητές εξετάζουν τα αποτελέσματα των εκάστοτε αλγορίθμων μέσα από υπολογιστική ανάλυση και συγκριτική μελέτη. Στη σημερινή εποχή έχουν εισαχθεί νέες υπολογιστικές τεχνικές, όπως μεθευρετικοί αλγόριθμοι (κατασκευάζουν αρχικά μια εφικτή λύση και έπειτα την βελτιώνουν με τοπική αναζήτηση γειτονιάς), και γενετικοί ή εξελικτικοί αλγόριθμοι, οι οποίοι βελτιώνουν την λύση μέσα από την εξέλιξη ενός πληθυσμού λύσεων (με ή χωρίς γεωμετρικούς περιορισμούς), (Zouein, Harmanani, and Hajar 2002). Γενικά, διάφορες ευρετικές μέθοδοι έχουν διατυπωθεί σε εφαρμογές της επιχειρησιακής έρευνας και της τεχνητής νοημοσύνης για τον προγραμματισμό του χώρου (Zouein and Tommelein 1999). Μια σειρά από τις μεθόδους αυτές παραθέτονται ομαδοποιημένες σε πίνακες στις επόμενες υποενότητες.

2.3.1 Ευρετικοί αλγόριθμοι

Ο αλγόριθμος του Hakimi (1964), ο οποίος προσπάθησε να λύσει τη χωροθέτηση κέντρων σε ένα τηλεπικοινωνιακό σύστημα αποτελεί τη θεωρητική και ιστορική απαρχή της επίλυσης προβλημάτων χωροθέτησης στα πλαίσια της θεωρίας γραφημάτων. Όπως παραθέτεται και στον πίνακα παρακάτω, από το 1963 οι ερευνητές άρχισαν να αναπτύσουν ευρετικές μεθόδους για να προσεγγίσουν την βέλτιστη λύση σε αποδεκτό χρόνο. Οι ευρετικές μέθοδοι αρχικοποίησης αποτελούν ένα είδος ευρετικών μεθόδων, οι οποίες ξεκινούν με μια κενή λύση και μέσα από επαναλήψεις, επεκτείνουν την τρέχουσα λύση μέχρι να επιτευχθεί μια ολοκληρωμένη λύση. Μια τέτοια μέθοδος είναι η λεγόμενη άπληστη τεχνική, Greedy (Myopic), η οποία ξεκινάει με ένα κενό σύνολο λύσεων μέχρι που επιλύει το πρόβλημα διαμέσων για $p = 1$, το οποίο το προσθέτει στο σύνολο λύσεων. Με επαναλήψεις αυτή η διαδικασία συνεχίζεται έως το σύνολο λύσεων να αποκτήσει p τοποθεσίες σύμφωνα με τους (Kuehn and Hamburger 1963) και (Whitaker 1983).

Οι (Feldman, Lehrer, and Ray 1966), (Moreno, Rodríguez, and Jiménez 1991),(Salhi and Atkinson 1995) ακολούθησαν μια διαφορετική προσέγγιση, την Stingy ή Greedy-Drop τεχνική. Η τεχνική αυτή έρχεται σε αντίθεση με την απλή άπληστη ευρετική μέθοδο καθώς ξεκινάει με ένα σύνολο λύσεων γεμάτο k-τοποθεσίες και αφαιρετικά σε κάθε επανάληψη προσπαθεί να φθάσει πλήθος συνόλου λύσεων ίσο με p διαμέσους.

Η μέθοδος δυαδικής ανόδου απασχόλησε τον Erlenkotter το 1978, (Erlenkotter 1978), τον οποίο αργότερα ακολούθησαν οι ερευνητές (Galvão 1980), (Galvão 1993), (Captivo 1991). Η μέθοδος αυτή επιλύει το δυϊκό πρόβλημα αργά, με μικρές αυξήσεις της δυϊκής αντικειμενικής συνάρτησης και αρκετές αυστηρές παραδοχές.

Τέλος, αρκετές παραλλαγές των παραπάνω μεθόδων και συγκρίσεις μεθόδων για το πρόβλημα διαμέσων ανέπτυξαν οι ((Captivo 1991)), συνδυασμός μεθόδων Stingy με Greedy και Multistart Alternate), ((Captivo 1991), GreedyGr αλγόριθμο), ((Pizzolato 1994), έναν συνδυασμό Alternate - Interchange), και τέλος ο ((Salhi 1997), συνδυασμός μεθόδων Stingy και Greedy), (Mladenović, Brimberg, Hansen, and Moreno-Pérez 2007).

Πίνακας **2.1**: Ευρετικές μέθοδοι αρχικοποίησης

Μέθοδος	Ερευνητές
Greedy (Myopic)	Kuehn and Hamburger (1963), Whitaker (1983), Xiao (2016)
Stingy	Feldman et al. (1966), Moreno-Pérez et al. (1991), Salhi and Atkinson (1995)
Dual Ascent	Galvão (1980, 1993), Erlenkotter (1978), Captivo (1991)
Composite	Moreno-Pérez et al. (1991), Captivo (1991), Pizzolato (1994), Salhi (1997)

2.3.2 Ευρετικές μέθοδοι τοπικής αναζήτησης

Η διαφορά των μεθόδων τοπικής αναζήτησης από αυτές της αρχικοποίησης είναι ότι αρχίζουν με μια ολοκληρωμένη εφικτή λύση και στη συνέχεια προσπαθούν να βελτιώσουν την τρέχουσα λύση μέσα από επαναλήψεις τοπικής αναζήτησης. Μια τέτοια μέθοδος είναι και η Alternate, η οποία σύμφωνα με τον Maranzana ξεκινάει με p σημεία που δηλώνουν τοποθεσίες εκλεγμένες από σύνολα χρηστών κοντά σε αυτές. Αρχικά, λύνεται το πρόβλημα διαμέσων για $p = 1$ και για κάθε σύνολο χρηστών των τοποθεσιών. Αργότερα, οδηγείται σε μια εξαντλητική μέθοδο όπου όμως τελικά οι εναλλακτικές τοποθεσίες θα αντιστοιχίζονται σε ικανοποιημένους χρήστες. Μια ακόμη κλασσική ευρετική μέθοδος τοπικής αναζήτησης είναι η Interchange. Με p τοποθεσίες αρχικά και μέσα από μια επαναληπτική δομή δοκιμάζει μία προς μία τις υπόλοιπες τοποθεσίες ώστε να ελαχιστοποιηθεί το συνολικό κόστος. Σημαντικός ήταν ο ρόλος των Teitz and Bart το 1968, καθώς και των (Whitaker 1983), (Densham and Rushton 1992),(Hansen and Mladenović 1997). Αλλά και στην σύγχρονη εποχή ακόμη οι (Resende and Werneck 2003) και (Kochetov, Levanova, Alekseeva, and Loresh 2005) βοήθησαν αρκετά στην εξέλιξη αυτής της μεθόδου.

Πίνακας **2.2**: Ευρετικές μέθοδοι τοπικής αναζήτησης

Μέθοδος	Ερευνητές
Alternate	Maranzana (1964)
Interchange	Teitz and Bart (1968), Whitaker (1983), Densham and Rushton (1992),Hansen and Mladenović (1997), Narula, Ogbu, and Samuelsson (1977),Resende and Werneck (2003), Kochetov et al. (2005),Xiao (2016)

2.3.3 Αλγόριθμοι Μαθηματικού Προγραμματισμού

Ο Δυναμικός Προγραμματισμός ή ΔΠ (Dynamic Programming, DP), είναι μια μαθηματική τεχνική που αναπτύχθηκε από τον (Bellman 1954) για τα προβλήματα βελτιστοποίησης με διαδοχικες και αλληλοσυνδεόμενες αποφάσεις. Συνοπτικά, η χρήση του ΔΠ επικεντρώνεται σε μια συστηματική διαδικασία για τον προσδιορισμό του συνδυασμού των αποφάσεων, που μεγιστοποιεί τη συνολική αποτελεσματικότητα. Σε αντίθεση με τη μέθοδο διαίρει και βασίλευε, ο ΔΠ δεν διαιρεί το κύριο πρόβλημα σε ανεξάρτητα υποπροβλήματα, αλλά επιλύει το πρόβλημα συνδυάζοντας λύσεις υποπροβλημάτων. Η κατάστρωση του προβλήματος γίνεται σε φάσεις ή σταδια (stages), γι' αυτό και ο ΔΠ οναμάζεται και μεθοδολογία διαδοχικών καταμερισμών (sequential allocation process) ή διαδικασία λήψης αποφάσεων πολλών σταδίων (multistage decision-making procedure). Αν και ΔΠ έχει κατανοητή μεθοδολογία, σε αντίθεση με τον γραμμικό προγραμματισμό, δεν χαρακτηρίζεται από κάποιο γενικό πρότυπο ή μια τυποποιημένη μαθηματική διατύπωση. Τα προβλήματα δυναμικού προγραμματισμού αντιμετωπίζονται διαφορετικά κάθε φορά και προσαρμόζονται ανάλογα με το πρόβλημα σε κάθε συγκεκριμένη κατάσταση, ώστε να προσεγγιστεί το πότε ένα πρόβλημα μπορεί να λυθεί με διαδικασίες δυναμικού προγραμματισμού και πως αυτό υλοποιείται. Η πρώτη φάση στη σχεδίαση ενός αλγορίθμου ΔΠ είναι η εύρεση της δομής της βέλτιστης λύσης, έπειτα ορίζεται αναδρομικά η τιμή της βέλτιστης λύσης και τέλος στην τελευταία φάση υπολογίζεται η βέλτιστη λύση και η τιμή της από τα υποπροβλήματα προς το κυρίως προβλημα.

Οι (Hribar and S. 1997) πρότειναν μία ευρετική μέθοδο που έκανε χρήση ΔΠ, η οποία είχε κύριο σκοπό να εντοπίσει επαναληπτικά και να αποθηκεύσει τις q καλύτερες λύσεις, σταματώντας όταν επιτευχθούν p εγκαταστάσεις. Λίγα χρόνια νωρίτερα είχε προταθεί από τους (Cornuejols, Fisher, and Nemhauser 1977) η μοντελοποίησή του προβλήματος διαμέσων με Lagrangian χαλάρωση ή διαδικασία LR, Lagrangian Relaxation, με τη βελτιστοποίηση δηλαδή μιας προσαρμοστικής, προβλεπόμενης μεθόδου subgradient, μια τεχνική που χρησιμοποιείται για να ενημερώσει την τιμή των πολλαπλασιαστών Lagrange. Αρκετοί ήταν οι ερευνητές που ασχολήθηκαν με την Lagrangian χαλάρωση, (βλ. Πίνακα: Μέθοδοι μαθηματικού προγραμματισμού), αλλά και με παραλλαγές αυτής της μεθόδου όπως η πρόταση Semi Lagrangian Relaxation των (Beltran, Tadonki, and Vial 2006) το 2006. Για να μειωθεί ο χρόνος υπολογισμού των λύσεων, συχνή είναι η μέθοδος της Συνάθροισης Aggregation, δηλαδή η προσεκτική μείωση των σημείων ζήτησης συγκεντρώνοντας τα πιο αντιπροσωπευτικά σημεία ώστε να αποδοθεί μια καλή προσσέγγιση της λύσης πιο γρήγορα. Περιορίζοντας όμως τα σημεία ζήτησης, μειώνεται η πηγαία πληροφορία και στο πρόβλημα εισέρχονται σφάλματα κόστους, βελτιστότητας και τοποθεσίας, (Erkut and Bozkaya 1999). Μια πιο ολοκληρωμένη θεωρητική βάση όσον αφορά τα σφάλματα της συνάθροισης διατυπώθηκε από τους (Francis, Lowe, and Tamir 2002).

Πίνακας **2.3**: Μέθοδοι μαθηματικού προγραμματισμού

Μέθοδος	Ερευνητές
Dynamic Programming	Hribar and Daskin (1997)
Relaxation Induced Neighborhood Search	Danna et al. (2005), Yaghini et al. (2012)
Lagrangian Relaxation	Cornuejols et al. (1977), Mulvey and Crowder (1979), Galvão (1980), Beasley (1993), Daskin (1995), Daskin (2013), Senne and Lorena (2000), Barahona and Anbil (2000), Beltran et al. (2006)
Aggregation	Hillsman and Rhoda (1978), Goodchild (1979), Erkut and Bozkaya (1999), Casillas (1987), Current and Schilling (1987), Hodgson and Neuman (1993), Hodgson and Salhi (1998), Bowerman et al. (1999), Francis et al. (2003)

2.3.4 Μεθευρετικοί αλγόριθμοι

Η τεχνική αναζήτησης Tabu είναι μια μέθοδος τοπικής αναζήτησης, η οποία προτάθηκε και πρωτοεμφανίστηκε από τον (Glover 1989). Η μέθοδος αυτή χρησιμοποιεί τον ορισμό της γειτονιάς και βήμα προς βήμα εντοπίζει την ελάχιστη τιμή μιας συνάρτησης, εντοπίζοντας γειτονικές λύσεις. Μια ταμπού λίστα T των απαγορευμένων κινήσεων ενημερώνεται κατά τη διάρκεια των επαναλήψεων για να αποφευχθεί ο κύκλος ή αλλιώς ο εγκλωβισμός σε τοπικά ελάχιστα. Η μέθοδος Αντίστροφης Εξάλειψης (REM), που χρησιμοποιήθηκε από τον (Voss 1996) το 1996, είναι μια δυναμική στρατηγική για τη διαχείριση της λίστα ταμπού και παρέχει μια αναγκαία και ικανή συνθήκη για να αποτρέψει τον εγκλωβισμό σε τοπικά βέλτιστες λυσεις. Οι (Rolland, Schilling, and Current 1997) και (Salhi 2002) πρότειναν μια παραλλαγή της μεθόδου που ονομάζεται στρατηγική ταλάντωσης (strategic oscillation), μια δομή που χρησιμοποιείται σε προσεγγιστικές μεθόδους κατασκευής ή βελτίωσης. Τέλος, μια πιθανολογική μέθοδος ταμπού προτάθηκε από τον (Kochetov 2001) το 2001 σε συνδυασμό με αλυσίδες Markov ώστε να αναπτυχθούν ασυμπτωτικές θεωρητικές ιδιότητες και αναλύθηκαν από τους (Goncharov and Kochetov 2002), (Mladenović, Brimberg, Hansen, and Moreno-Pérez 2007). Η επόμενη μεθευρετική μέθοδος που παρουσιάζεται στον παρακάτω πίνακα είναι η αναζήτηση μεταβλητής γειτνίασης Variable Neighborhod Search, VNS. Η αναζήτηση μεταβλητής γειτνίασης διερευνά επαναληπτικά από μικρές έως και μεγαλύτερες απομακρυσμένες γειτονιές για ένα τοπικό βέλτιστο μέχρι να βρεθεί μια βελτιωμένη λύση της υπάρχουσας μετά από κάθε επανάληψη. Αρκετοί είναι οι ερευνητές που ασχολήθηκαν με τη μέθοδο αναζήτησης μεταβλητής γειτνίασης για το πρόβλημα διαμέσων, αν και η μέθοδος είχει προταθεί αρχικά το 1995 από τον Mladenović και συνεχίζεται έως και σήμερα η διερεύνησή της.

Οι γενετικοί αλγόριθμοι αποτελούν μια κατηγορία μεθευρετικών μεθόδων των οποίων ο βασικός μηχανισμός είναι εμπνευσμένος από τη Δαρβινική εξελικτική θεωρία της φύσης. Οι γενετικοί αλγόριθμοι διερευνούν το χώρο των υποψήφιων λύσεων με στόχο τον εντοπισμό αποδεκτών, σύμφωνα με κάποιο κριτήριο λύσεων. Για το πρόβλημα διαμέσων η αρχή έγινε από τους (Hosage and Goodchild 1986) που κωδικοποίησαν μια λύση ως μια σειρά από m δυαδικά ψηφία-γονίδια. Στη δημοσίευση των Dibbie and Densham το 1993, κάθε άτομο αποτελείται από ακριβώς p γονίδια και κάθε γονίδιο αναπαραστά ένα δείκτη εγκατάσταση. Μια τεχνική πολύ καλύτερη από την πρώτη καθώς συνδυάζει μαζί μεθόδους μετάλλαξης και cross-over. Με χρήση πολλαπλού πληθυσμού ομάδων, οι οποίες ανταλλάσσουν μεταξύ τους υποψήφιες λύσεις το 1994 οι (Perez, Garcia, and Moreno 1994) προσέγγισαν το πρόβλημα διαμέσων. Στον παρακάτω πίνακα ακολουθούν και πιο σύγχρονες δημοσιεύσεις για την επίλυση του προβλήματος διαμέσων με γενετικούς αλγορίθμους, (Mladenović, Brimberg, Hansen, and Moreno-Pérez 2007). Η προσομοιωμένη ανόπτηση είναι μία στοχαστική μέθοδος βελτιστοποίησης που μπορεί να ανταποκριθεί καλά σε μεγάλης κλίμακας προβλήματα διακριτής ή συνδυαστικής βελτιστοποίησης. Η μέθοδος αυτή ορίζει εκ νέου την τοπική αναζήτηση εισάγοντας την έννοια της θερμοκρασίας T. Μια βασική μεθευρετική μέθοδος προτάθηκε από τους (Murray and Church 1996) και λίγα χρόνια αργότερα οι Chiyoshi και Galvão. Τα υπολογιστικά αποτελέσματα της μεθόδου βελτιώνονται αρκετά με το χρόνο και πολλές φορές αποδίδει και ακριβείς λύσεις αναφορικά με το πρόβλημα διαμέσων.

Η ευρετική συγκέντρωση (HC) με βάση τους (Rosing and Revelle 1997), το 1997, έχει δύο φάσεις. Αρχικά, ένα σύνολο λύσεων λαμβάνεται με επανάληψη q φορές τη Drop / Add ευρετική μέθοδο, και στη συνέχεια διατηρούνται οι m καλύτερες λύσεις - εγκατάστασεις. Στο δεύτερο στάδιο περιορίζεται το σύνολο των πιθανών εγκαταστάσεων στο πλήθος διαμέσων. Η ευρετική μέθοδος Gamma (Rosing, Revelle, and Schilling 1999) περιλαμβάνει και ένα επιπλέον τρίτο στάδιο. Οι θεμελιώδεις έννοιες και αρχές της μεθόδου αναζήτησης Scatter προτάθηκαν για πρώτη φορά στη δεκαετία του 80. Η αναζήτηση Scatter (SS) είναι μια εξελικτική στρατηγική που χρησιμοποιεί στρατηγικές διαφοροποίησης αναζήτησης και εντατικοποίησης με έμφαση στον συνδυασμό κανόνων λήψης αποφάσεων και επίλυσης περιορισμών. Ο παρακάτω πίνακας 4 εμπεριέχει τους ερευνητές που ασχολήθηκαν με την αναζήτηση Scatter (SS) από το 2000 έως και το 2016. Η βελτιστοποίηση αποικίας μυρμηγκιών (ACO) προτάθηκε αρχικά από τους (Dorigo, Maniezzo, and Colorni 1991). Η ιδέα για τη μέθοδο αυτή προέρχεται από τη φύση και πιο συγκεκριμμένα αποσκοπεί στην αναζήτηση μιας βέλτιστης διαδρομής σε ένα γράφο επιδεικνύοντας την συλλογική συμπεριφορά των μυρμηγκιών. Κάθε μέλος της αποικίας μπορεί να εκτελέσει απλές ενέργειες, υπακούοντας σε απλούς κανόνες. Τα μυρμήγκια στο πέρασμά τους αφήνουν μια ορμόνη τη φερομόνη, έτσι ψάχνοντας τροφή τα μυρμήγκια που ακολουθούν από τη φωλιά προς την πηγή τροφής, επιλέγουν το μονοπάτι πιθανοτικά με την ισχυρότερη συγκέντρωση φερομόνης. Αυτή η διαδικασία επαναλαμβάνεται και τελικά όλα τα μυρμήγκια ακολουθούν τη συντομότερη διαδρομή, δεδομένου ότι λαμβάνει το μεγαλύτερο ποσό της φερομόνης.

Μια ακόμη προσεγγιστική μέθοδος εμπνευσμένη από τη βιολογία, προσέγγιση της λειτουργίας του εγκεφάλου, είναι τα νευρωνικά δίκτυα, η υπολογιστική προσομοίωση της λειτουργίας των βιολογικών νευρωνικών δικτύων με βάση κάποιο μαθηματικό μοντέλο. Τα νευρωνικά δίκτυα είναι δίκτυα από απλούς υπολογιστικούς κόμβους (νευρώνες), διασυνδεδεμένους μεταξύ τους και μπορούν να θεωρηθούν ως μεγάλης κλίμακας παράλληλοι επεξεργαστές που αποτελούνται από απλές επεξεργαστικές μονάδες. Έχει την δυνατότητα να μαθαίνει από πειράματα και η γνώση που αποκτά αποτυπώνεται στην βαρύτητα κάθε σύνδεσης μεταξύ των επιμέρους μονάδων. Οι (Merino and Perez 2002) το 2002 επίλυσαν το πρόβλημα διαμέσων με νευρωνικά δίκτυα δύο επιπέδων. Ένα χρόνο αργότερα, ακολούθησαν τρεις αλγόριθμοι ανταγωνιστικού δικτύου ενός επιπέδου 2NP νευρώνων. Ένα ανταγωνιστικό νευρωνικό δίκτυο αποτελείται από δύο επίπεδα πλήρως συνδεδεμένου δικτύου του οποίου οι νευρώνες ανταγωνίζονται να ενεργοποιηθούν υπό ορισμένες προϋποθέσεις. Από την άλλη πλευρά, υπάρχει και η στρατηγική της διάσπασης μεγάλης κλίμακας προβλημάτων σε υποπροβλήματα. Οι μέθοδοι διάσπασης ή αποσύνθεσης λειτουργούν με την ομαδοποίηση των μεταβλητών σε υποσύνολα, και την επίλυση ενός υποπροβλήματος για κάθε ένα από αυτά. Η αρχή για την επίλυση του προβλήματος διαμέσων έγινε από τους Dai και Cheung το 1997 με δύο ευρετικές μεθόδους. Ακολούθησε η δημοσίευση του (Taillard 2003), που πρότεινε τρεις ευρετικές μεθόδους που μπορούν να εφαρμοστούν γενικά σε προβλήματα τέτοιας φύσεως και κλίμακας, π.χ. p-διαμέσων, multisource Weber, ελάχιστη τετραγωνική ομαδοποίηση clustering. Αξίζει να σημειωθεί ότι μία από τις ευρετικές μεθόδους που προτείνει ο Taillard είναι η στρατηγική υποψήφιας λίστας CLS, μια μέθοδος αρκετά κοντά στη λογική της μεθόδου VNS.

Τέλος, παραθέτονται οι υβριδικοί ευρετικοί αλγόριθμοι που συνδυάζουν στοιχεία γνήσιων μεθευρετικών μεθόδων, όπως ο GRASP (Άπληστη Διαδικασία Τυχαιοποιημένης Προσαρμοστικής Αναζήτηση), (Feo and Resende 1995), όπου κάθε επανάληψη αποτελείται από την κατασκευή των αρχικών σημείων από μια τυχαιοποιημένη άπληστη τεχνική, και έπεται από μια τεχνική τοπικής αναζήτησης. Όπως και στις TS και SS, η μέθοδός τους δανείζεται την ιδέα της επανασύνδεσης διαδρομής ,(Laguna and Marti 1999). Εντοπίζεται ένα μονοπάτι μεταξύ δύο λύσεων από ένα σύνολο καλών λύσεων και εκτελείται τοπική αναζήτηση ξεκινώντας από κάθε λύση σε αυτό το μονοπάτι. Επιπλέον, αυξάνεται το μονοπάτι-επανασύνδεσης με την έννοια των επιπλέον γενιών, ένα βασικό χαρακτηριστικό των γενετικών αλγορίθμων. Σε σύγκριση με άλλες μεθόδους, οι υβριδικές διαδικασίες παρέχουν συνήθως καλύτερα αποτελέσματα ως προς την ποιότητα και τον χρόνο.

Πίνακας **2.4**: Μεθευρετικοί αλγόριθμοι

Μέθοδος	Ερευνητές
Tabu Search	Mladenović et al. (1996), Glover (1989, 1990), Hansen and Jaumard (1990), Voss (1996), Rolland et al. (1996), Salhi (2002), Kochetov (2001), Goncharov and Kochetov (2002)
Variable Neighborhood Search	Mladenović (1995), Hansen and Mladenović (1997), Hansen et al. (2001), Crainic et al. (2004), Chaves and Lorena (2002, 2010), Fleszar and Hindi(2008)
Genetic Search	Correa et al. (2004), Houck et al. (1996), Brimberg (2000), Hosage and Goodchild (1986), Dibbie and Densham (1993), Moreno-Pérez et al. (1994), Estivill-Castro (1999), Alp et al. (2003), Chaudhry et al. (2003)
Simulated Annealing	Metropolis (1953), Chardaire and Lutton (1993), Murray and Church (1996), Chiyoshi and Galvão (2000), Levanova and Loresh (2004), Salhi and Gamal (2003), Xiao (2016)
Heuristic Concentration	Rosing et al. (1998), Rosing and ReVelle (1997), Rosing et al. (1999)
Scatter search	Fernandez (2006), Schreuerer and Wendolsky (2006), Garcia-Lopez et al. (2003)
Ant colony	Levanova and Loresh (2004), De Franca (2005), Anagnwst'opoulos (2005)
Neural networks	Dominguez Merino and Muñoz Perez (2002, 2003)
Decomposition	Dai and Cheung (1997), Taillard (2003)
Hybrids	Resende and Werneck (2004), Captivo (1991), Pizzolato (1994), Moreno et al. (1991)

2.4 Μαθηματική μοντελοποίηση του προβλήματος

Ας εξετάσσουμε το πρόβλημα διαμέσων ως ένα πρόβλημα βελτιστοποίησης με στόχο τη δημιουργία πλήθους p εγκαταστάσεων που πρέπει να τοποθετηθούν σε ιδίου πλήθους τοποθεσίες. Κάθε εγκατάσταση θέλουμε να τοποθετηθεί σε μία τοποθεσία.

Αν $I = 1, ..., i, ..., n$ είναι το σύνολο σημείων ζήτησης , $J = 1, ..., j, ..., m$ είναι το σύνολο σημείων πιθανόν τοποθεσιών - εγκαταστάσεων, δημιουργείται ο πίνακας c_{ij} με $n * m$ διάσταση που απεικονίζει το κόστος ικανοποίησης της ζήτησης των πελατών i από την εγκατάσταση j.

Ορίζουμε με $x_{ij} = 1$ εάν ο πελάτης i εξυπηρετείται από την εγκατάσταση j και προσδιορίζουμε μια διακριτή μεταβλητή απόφασης τέτοια ώστε:

$$y_j = \begin{cases} 1, & \text{εάν η εγκατάσταση j είναι ανοιχτή} \\ 0, & \text{αλλιώς} \end{cases} \tag{1}$$

Έτσι η μαθηματική μοντελοποίηση του προβλήματος χωροθέτησης στοχεύει στην ελαχιστοποίηση της ακόλουθης αντικειμενικής συνάρτησης:

$$\min \textstyle\sum_{i \in I} \sum_{j \in J} c_{ij} * x_{ij}$$

υπό τους περιορισμούς

- $\sum_{j \in J} x_{ij} = 1, \forall i \in I$
  όλοι οι πελάτες πρέπει να εξυπηρετηθούν
- $\sum_{j \in J} y_j = p$
  να υπάρχουν ακριβώς p το πλήθος ανοικτές εγκαταστάσεις
- $x_{ij} \leq y_j \, \forall i \in I, \forall j \in J$
  ο πελάτης i μπορεί να εξυπηρετηθεί από την εγκατάσταση j μόνο εάν η j είναι ανοιχτή
- $x_{ij} \in \{0, 1\}, \forall i \in I$
  διακριτή μεταβλητή που προσδιορίζει εάν ένας πελάτης i εξυπηρετείται από την εγκατάσταση j
- $y_j \in \{0, 1\}, \forall j \in J$
  διακριτή μεταβλητή που προσδιορίζει αν η j-εγκατάσταση είναι ανοιχτή

Η αντικειμενική συνάρτηση ελαχιστοποιεί το ισοσταθμισμένο συνολικό κόστος ζήτησης. Ο πρώτος περιορισμός σημαίνει ότι για κάθε πελάτη που εκφράζει ζήτηση υπάρχει μια εγκατάσταση. Ο δεύτερος περιορισμός επαληθεύει ότι ακριβώς p εγκαταστάσεις έχουν δημιουργηθεί. Ο τρίτος περιορισμός συνδέει τις μεταβλητές εγκαταστάσεων με τις μεταβλητές κατανομής. Τέλος, ο τέταρτος και πέμπτος περιορισμός διασφαλίζουν ότι οι μεταβλητές εγκαταστάσεων και κατανομής είναι δυαδικές. Ο στόχος είναι να βρεθεί μια κατανομή των εγκαταστάσεων στις τοποθεσίες, έτσι ώστε να ελαχιστοποιηθεί το συνολικό κόστος μεταφοράς.

Αν υπάρχει μία βέλτιστη λύση στο πρόβλημα με p εγκαταστάσεις, στη συνέχεια με την προσθήκη ενός επιπλέον σημείου p+1 σε οποιουσδήποτε από τους υποψήφιους κόμβους θα μειώσει την ζήτηση στο σταθμισμένο συνολικό κόστος ή την απόσταση. Ως εκ τούτου, θα μειωθεί η τιμή της αντικειμενικής συνάρτησης, (Daskin and Maass 2015).

2.5 Παραλλαγές του προβλήματος

2.5.1 Lagrangian Χαλάρωση

Η χαλάρωση Lagrange (Lagrangian Relaxation) αναπτύχθηκε από τους (Held and Karp 1970) το 1970. Για ένα πρόβλημα ελαχιστοποίησης ακέραιου προγραμματισμού η μέθοδος αυτή μπορεί να χρησιμοποιηθεί για την δημιουργία κάτω ορίων σε σχέση με την βέλτιστη λύση. Αλλά και σαν μέθοδος γραμμικής χαλάρωσης (Linear Programming relaxation) στην οποία αφαιρούνται οι περιορισμοί ακεραιότητας από τις μεταβλητές του προβλήματος και ακολουθεί επίλυση με τον αλγόριθμο Simplex ή τον αλγόριθμο εσωτερικού σημείου ή με τη μέθοδο Dual Ascent. Η λύση που προκύπτει αποτελεί κάτω όριο εφόσον το πρόβλημα είναι ελαχιστοποίησης. Ωστόσο πολλά ειναι τα προβλήματα που η λύση της γραμμικής χαλάρωσης ενός προβλήματος δεν είναι εφικτή υπολογιστικά λόγω του μεγάλου αριθμού μεταβλητών και περιορισμών. Η LR διαφέρει από την μέθοδο γραμμικής χαλάρωσης, καθώς αντιστοιχεί μη αρνητικούς πολλαπλασιαστές Lagrange σε περιορισμούς τους οποίους στην συνέχεια προσαρτά στην συνάρτηση κόστους του προβλήματος, επιλύοντας το πρόβλημα ακέραιου προγραμματισμού που προκύπτει ευκολότερα από το αρχικό. Στον παρακάτω πίνακα, ακολουθεί ο μαθηματικός μετασχηματισμός Lagrange του αρχικού προβλήματος.

Πίνακας **2.5**: Lagrangian Relaxation

Αρχικό πρόβλημα	Μετασχηματισμένο πρόβλημα κατά Lagrange
$\min cx$ υ.π. $ax >= b, bx >= d, x \in 0,1$	$\min cx + l(b - ax)$ υ.π. $bx >= d, x \in 0,1, l >= 0$

Στην μέθοδο Lagrangian Relaxation:

Όσο το $l >= 0$ και το $(b - ax) <= 0$ ο παράγοντας που προστίθεται $l(b - ax)$ στην συνάρτηση κόστους είναι μη θετικός πιέζοντας την συνάρτηση κόστους προς χαμηλότερες τιμές. Άρα η απόρριψη των περιορισμών $ax >= b$ μπορεί να γίνει από το μετασχηματισμένο πρόβλημα καθώς η τήρηση των περιορισμών θα επιτευχθεί εμμέσως από την προσπάθεια ελαχιστοποίησης της συνάρτησης κόστους. Η λύση στο μετασχηματισμένο πρόβλημα αποτελεί απαραίτητα εφικτή λύση του αρχικού προβλήματος (duality gap). Ένας ευρετικός Lagrangian αλγόριθμος θα πρέπει να μετασχηματίζει την λύση του προβλήματος σε εφικτή. Ο αλγόριθμος αυτός μπορεί να είναι απλός στην σύλληψή, δίνοντας ικανοποιητικά αποτελέσματα όπως φαίνεται στο παράδειγμα του Beasley, (Beasley 1990). Επίσης η επίλυση του αρχικού προβλήματος μπορεί να επιτευχθεί αναγνωρίζοντας μεταβλητές που δεν μπορούν να συμμετέχουν στην βέλτιστη λύση καθώς και μεταβλητές που πρέπει να συμμετέχουν στην βέλτιστη λύση, επιλύοντας το αρχικό πρόβλημα υπολογιστικά ευκολότερα και μειώνοντας τις διαστάσεις του. Ο εντοπισμός των πολλαπλασιαστών Lagrange που δίνουν το μεγαλύτερο κάτω όριο για το Lagrangian δυϊκό πρόβλημα είναι ο στόχος, για τον καθορισμό των οποίων υπάρχουν δύο μέθοδοι, η τεχνική subgradient optimization, που αποτελεί μια επαναληπτική διαδικασία που ξεκινώντας από ένα αρχικό σύνολο πολλαπλασιαστών Lagrange, δημιουργεί νέες τιμές για τους πολλαπλασιαστές με ένα συστηματικό τρόπο, και η ρύθμιση των πολλαπλασιαστών (multiplier adjustment), η οποία είναι ένας απλός ευρετικός τρόπος προσδιορισμού των πολλαπλασιαστών Lagrange με πλεονέκτημα τον μικρό χρόνο εκτέλεσης. Η τεχνική subgradient optimization δεν εξασφαλίζει ότι η τιμή του ορίου θα αυξάνεται σε κάθε βήμα και γι' αυτό τον λόγο πρέπει να διατηρείται η καλύτερη τιμή που έχει σημειωθεί από όλες τις επαναλήψεις. Η συμπεριφορά που συνήθως παρατηρείται είναι ότι η μεγαλύτερη τιμή του κάτω ορίου αυξάνεται γρήγορα αρχικά και ο ρυθμός αύξησης μειώνεται καθώς προχωρά η διαδικασία. Στην τεχνική της ρύθμιση των πολλαπλασιαστών συνήθως αλλάζει ένας μόνο πολλαπλασιαστής σε κάθε επανάληψη και παράγει λιγότερο καλές τιμές ορίων, σε αντίθεση με την προηγούμενη τεχνική στην οποία σε μια επανάληψη θεωρητικά μπορούν να αλλάξουν όλοι οι πολλαπλασιαστές.

Το πρόβλημα διάμεσων λόγω της ιδιαιτερότητας του σε μεγάλης κλίμακας προβλήματα, πολλές φορές προτείνεται στη βιβλιογραφία η μοντελοποίησή του με Lagrangian χαλάρωση ή διαδικασία LR, Lagrangian Relaxation, με τη βελτιστοποίηση δηλαδή μιας προσαρμοστικής, προβλεπόμενης μεθόδου subgradient, μια τεχνική που χρησιμοποιείται για να ενημερώσει την τιμή των πολλαπλασιαστών Lagrange. Αυτή η διαδικασία είναι μια μορφή "Branch and bound" αλγορίθμου, (Land and Doig 1960), σύμφωνα με την οποία ο αλγόριθμος εξαρτάται από την αποτελεσματική εκτίμηση των κάτω και άνω ορίων ή bounds της περιοχής branch του χώρου αναζήτησης και των εξαντλητικών προσεγγίσεων απαρίθμησης, όμως δεν θα επεκταθούμε περαιτέρω σε αυτήν την εργασία. Οι εξελίξεις αυτού του είδους της προσέγγισης περιγράφονται από τους Christofides και Beasley.

Το πλεονέκτημα της Lagrangian χαλάρωσης σε οποιαδήποτε ευρετική προσέγγιση είναι διπλό. Αρχικά, σε κάθε επανάληψη παίρνουμε κάτω και άνω φράγματα της αντικειμενικής αξίας λειτουργίας καθώς η ενσωμάτωση της μεθόδου σε έναν αλγόριθμο διακλάδωσης και οριοθέτησης για την βέλτιστη λύση καθίσταται πολύ εύκολος.

$$\max\min\sum_{i\in I}\sum_{j\in J} d_j * c_{ij} * x_{ij} + \sum_{j\in J} l_j * (1-\sum_{i\in I} x_{ij}) = \max\min\sum_{i\in I}\sum_{j\in J}(d_j * c_{ij} - l_j) * x_{ij} + \sum_{j\in J} l_j \tag{2}$$

Για σταθερές τιμές των πολλαπλασιαστών Lagrange lj, υπολογίζουμε την τιμή της περίπτωσης να προσθεθεί μια εγκατάσταση στον κόμβο $i \in I$. Η τιμή αυτή υπολογίζεται από την παρακάτω ισότητα:

$$V_i = sum_{i\in I}\min 0,_{d_j * c_{ij} - l_j} \tag{3}$$

Στη συνέχεια, επιλέγονται οι p πιο αρνητικές τιμές της παραπάνω ισότητας, καθορίζοντας έτσι τις τιμές των μεταβλητών τοποθεσιών y_i. Εάν $y_i = 1$ τότε $x_{ij} = 1$ αλλιώς εάν $d_j * c_{ij} - l_j < 0$ τότε $x_{ij} = 0$

Έτσι οι τιμές που προκύπτουν μπορούν να χρησιμοποιηθούν για να αξιολογήσουν τον μετασχηματισμό, δίνοντας ένα κατώτερο όριο για την αντικειμενικής συνάρτηση. Για να προσδιοριστεί ένα άνω όριο απλά να εκχωρείται σε κάθε κόμβο ζήτησης ο πλησιέστερος υποψήφιος κόμβος εγκατάστασης, πρεπει να είναι τέτοιος ώστε $y_i = 1$.

Είναι πιθανόν κάποιοι περιορισμοί να παραβιαστούν από την λύση του μετασχηματισμένου προβλήματος. Ειδικότερα, ορισμένοι κόμβοι ζήτησης δεν θα μπορέσουν να ανατεθούν σε μία εγκατάσταση και άλλοι μπορεί να ανατεθούν σε πολλαπλές εγκαταστάσεις, κυρίως όταν οι πολλαπλασιαστές Lagrange δεν κατέχουν τις βέλτιστες τιμές τους. Η τεχνική subgradient optimization χρησιμοποιείται για να βελτιώσει τις τιμές αυτές, (Daskin 2011).

Έτσι για καλύτερο κάτω (ΚΦ) και άνω φράγμα (ΑΦ) και χρησιμοποιώντας τους πολλαπλασιαστές Lagrange που ταξινομούν τις τιμές V_i, πέρνουμε την πιο αρνητική τιμή V_1 και γνωρίζοντας την τελευταία, έστω V_r, υποψήφια εγκατάσταση, γνωρίζουμε ότι η τιμή V_{p+1} είναι η επόμενη μεγαλύτερη τιμή. Με άλλα λόγια, εάν το $AF < KF - V_i + V_{p+1}$, για τοποθεσία $i \in I$ αποτελεί υποψήφια λύση και θέτουμε $y_i = 1$ σε όλες τις επόμενες Lagrangian επαναλήψεις και σε οποιαδήποτε διακλάδωση κάτω από τον κόμβο κατά την οποία αυτός ο έλεγχος επαληθεύεται. Ομοίως, εάν η τοποθεσία $i \in I$ δεν είναι μέρος των καλών λύσεων και $AF < KF + V_i - V_p$, τότε σαν τοποθεσία θέτουμε $y_i = 0$, η τοποθεσία δεν συμπεριλαμβάνεται στις λύσεις σε καμία από τις επόμενες επαναλήψεις.

2.6 Σύγχρονες τάσεις και μέθοδοι επίλυσης

Καθώς το πρόβλημα διαμέσων χρονολογείται κοντά στο 17ο αιώνα, αυτό έχει ως άμεσο επακόλουθο την ανάπτυξη μιας πληθώρα αλγορίθμων και μεθόδων exact, heurestic, metaheurestic ή και hybrid με το πέρασμα του χρόνου. Σε αυτήν την ενότητα θα ήταν καλό να αναφέρουμε τις τάσεις επίλυσης αυτού του προβλήματος που ακολουθούν οι ερευνητές κυρίως τα τελευταία 10 χρόνια. Οι μεθευρετικοί αλγόριθμοι αποτελούν μια σημαντική κατηγορία τεχνικών επίλυσης δύσκολων υπολογιστικά προβλημάτων, καθώς οι σύγχρονοι ερευνητές σε θέματα της συνδυαστικής βελτιστοποίησης γνωρίζουν ότι για την επίλυση NP-hard προβλημάτων, η χρήση exact μεθόδων είναι χρονικά αρκετά ακριβή. Πρόσφατες έρευνες έχουν δείξει ότι ο υβριδισμός μεθευρετικών μεθόδων αποτελεί ισχυρό μηχανισμό ανάπτυξης ακόμη πιο ισχυρών και αποτελεσματικών μεθόδων για την επίλυση προβλημάτων βελτιστοποίησης.

Λαμβάνωντας πρότυπα από μια τεχνική εξόρυξης δεδομένων, από ένα σύνολο υπο-βέλτιστων λύσεων ενός συνδυαστικού προβλήματος βελτιστοποίησης, μπορούν να χρησιμοποιηθούν για να οδηγήσουν τις μεθευρετικές μεθόδους στην αναζήτηση καλύτερων λύσεων, (Plastino, Fuchshuber, Martins, Freitas, and Salhi 2011). Οι (Plastino, Fuchshuber, Martins, Freitas, and Salhi 2011) πρότειναν έναν υβριδικό συνδυασμό εξόρυξης δεδομένων για την επίλυση του προβλήματος διαμέσων με GRASP τεχνική. Στην πρότασή τους, μετά την εκτέλεση ενός πλήθους επαναλήψεων, αναλαμβάνει η διαδικασία εξόρυξης δεδομένων να εξάγει τα πρότυπα από ένα σύνολο σημαντικών υπο-βέλτιστων λύσεων για το πρόβλημα p-μέσου. Αυτά τα πρότυπα παρουσιάζουν χαρακτηριστικά σχεδόν βέλτιστων λύσεων και μπορούν να χρησιμοποιηθούν για να πατρονάρουν τις επόμενες επαναλήψεις της GRASP ώστε να φτάσουν σε καλύτερο συνδιαστικά χώρο αναζήτησης. Η υπολογιστική μελέτη αποδεικύει ότι ο υβριδικός GRASP βρίσκει καλύτερα αποτελέσματα με την χρήση τεχνικών data mining αλλά και επιταχύνει την διαδικασία εξεύρεσης προσεγγιστικών ή και βέλτιστων λύσεων. Και οι (Martins, Vianna, Rosseti, Martins, and Plastino 2014) ανέπτυξαν μία υβριδική μεθοδολογία εξόρυξης δεδομένων ενός state-of-the-art ευρετικού αλγορίθμου για το κλασσικό πρόβλημα διαμέσων. Τα υπολογιστικά αποτελέσματα έδειξαν ότι η νέα έκδοση του ευρετικού αλγορίθμου ήταν σε θέση να βρει κατά μέσο όρο 27 με 32% ταχύτερα από την αρχική στρατηγική βέλτιστες ή σχεδόν βέλτιστες λύσεις. Οι (Santos, Martins, and Plastino 2008), παρουσίασαν τον ευρετικό αλγόριθμο DM-GRASP ως μια ακόμη υβριδική μέθοδο του GRASP που ενσωματώνει μια διαδικασία εξόρυξης δεδομένων και εξετάζει τον τρόπο υλοποίησης και εφαρμογής του.

Δύο νέους γενετικούς αλγόριθμους προτείναν οι (Neema, Maniruzzaman, and Ohgai 2011) με διαφορετικές διαδικασίες αντικατάστασης για την αντιμετώπιση του προβλήματος διαμέσων σε συνεχή χώρο και διαπιστώθηκε ότι οι μέθοδοι είναι αποτελεσματικοί στον εντοπισμό των βέλτιστων εγκαταστάσεων. Με παρόμοιο θέμα ασχολήθηκαν και οι ερευνητές (Krömer and Platoš 2014), οι οποίοι οδηγήθηκαν στην δημιουργία ενός νέου γενετικού αλγορίθμου για το πρόβλημα των διαμέσων και αξιολόγησαν την αποδοτικότητά του. Η έρευνά των (Erdoğan, Laporte, and Chía 2016) παρουσιάζει έναν ακριβή αλγόριθμο, έναν ευρετικό αλγόριθμο αρχικοποίησης δυναμικού προγραμματισμού και ένα μεθευρετικό αλγόριθμο τοπικής αναζήτησης με 2-exchange, 1-opt operators για το Hamiltonian p-Median πρόβλημα. Τα υπολογιστικά αποτελέσματα δείχνουν ότι ο αλγόριθμος διακλάδωσης - κοπής branch-and-cut ξεπερνά τις υπάρχουσες ακριβείς μεθόδους λύσης. Οι (Rebreyend, Lemarchand, and Euler 2015) τόσο στα μεσαίας κλίμακας προβλήματα που δημοσιεύθηκαν από τον Beasley όσο και σε μεγάλης κλίματας σουηδικά πραγματικά δεδομένα, τα πειράματα έδειξαν ότι η ImpGA μέθοδος παράγει σχετικά καλά αποτελέσματα όσον αφορά την ποιότητα και το χρόνο εκτέλεσης.

Τέλος, ο (Daskin 2011) ασχολήθηκαν με την ανάπτυξη, ανάλυση και παρουσίαση μιάς προσέγγισης Lagrangian χαλάρωσης για τα 40 κλασσικά μετροπροβλήματα της OR library ένός προβλήματος 500 κόμβων. Ένας νέος αλγόριθμος για την επίλυση του κλασικού προβλήματος διαμέσων δημιουργήθηκε από τους (Hale, Zhou, and Peng 2017), οι οποίοι προσπάθησαν με επιτυχία να προσδιορίσουν τους βέλτιστους διευρυμένους πολλαπλασιαστές που αντιστοιχούν στην βέλτιστη λύση. Τα αριθμητικά αποτελέσματα έδειξαν ότι ο προτεινόμενος αλγόριθμος μπορεί να βρει τη βέλτιστη λύση ή μία λύση πολύ κοντά της.

2.7 Εφαρμογές του προβλήματος

Ένα παράδειγμα στο οποίο ζητείται να βρεθεί η διάμεσος του γραφήματος, είναι αυτό της εξυπηρέτησης ενός αριθμού πελατών από μια αποθήκη. Οι πελάτες μπορούν να ομαδοποιηθούν κατά γειτονιές, έτσι το φορτηγό φορτώνει από την αποθήκη, διανέμει τα εμπορεύματα σε μια ομάδα πελατών και γυρίζει πάλι πίσω σε αυτήν. Οι ομάδες πελατών τοποθετούνται στις κορυφές του γραφήματος και το οδικό δίκτυο αποτελεί τις ακμές του γραφήματος.

Επειδή οι τάσεις της αγοράς εξελίσσονται και πολλοί περιβαλλοντικοί παράγοντες αλλάζουν, η ανάγκη για επανατοποθετήσεις (relocations) εξαπλώνεται και οι προσαρμοσμένες υπηρεσίες εξασφαλίζουν την εξέλιξη των νέων απαιτήσεων του σχεδιασμού. Οι σχεδιαστές στρατηγικών καλούνται συχνά να πάρουν τις ανάλογες χωρικές αποφάσεις κατανομής. Η δημιουργία και ανάπτυξη ενός κέντρου παροχής υπηρεσιών είναι μια δαπανηρή έρευνα και χρονικά απαιτητική διαδικασία. Πριν αγοραστεί ή κατασκευαστεί ένα κέντρο παροχής υπηρεσιών πρέπει να οριστούν θέσεις, κατάλληλες προδιαγραφές των δυνατοτήτων των κέντρων παροχής υπηρεσιών και επιπλέον πρέπει να κατανεμηθούν μεγάλα ποσά κεφαλαίου. Τα υψηλά κόστη που συνδέονται με τη διαδικασία της Χωροθέτησης Κέντρων Παροχής Υπηρεσιών (Facility Location) πρέπει να μετατρέπουν την οποιαδήποτε έρευνα Χωροθέτησης (location project) σε μια διαδικασία για εκτεταμένο χρόνο. Ο καθορισμός των καλύτερων θέσεων για κάθε νέο κέντρο παροχής υπηρεσιών αποτελεί μια σημαντική στρατηγική πρόκληση. Τα προβλήματα της Χωροθέτησης Κέντρων Παροχής Υπηρεσιών περιλαμβάνουν την τοποθέτηση ενός ή περισσότερων κέντρων παροχής υπηρεσιών σε μια περιοχή όπου υπάρχει ζήτηση μεσα από τη βελτιστοποίηση της αντικειμενικής συνάρτησης. Μια απλοποίηση αυτού του μετασχηματισμού είναι η αντικατάσταση της συνεχούς ζήτησης στην περιοχή με ένα διακεκριμένο σύνολο από σημεία ζήτησης (demand points) τα οποία το καθένα αναπαριστάνει μια υπογειτονιά, έτσι είναι ευκολότερη η κατασκευή της αντικειμενικής συνάρτησης μέσα από ένα πεπερασμένο σύνολο σημείων ζήτησης παρά σε μια συνεχή συναρτησιακή ζήτηση (continuous functional demand). Συχνά, η αντικειμενική συνάρτηση συνίσταται σαν ένα σύνολο από όρους, ένα για κάθε σημείο ζήτησης.

Οι εφαρμογές στα μοντέλα της Χωροθέτησης Κέντρων Παροχής Υπηρεσιών ποικίλουν. Στον δημόσιο τομέα ενδεικτικά μπορεί να αναφερθεί η χωροθέτηση πυροσβεστικών σταθμών, αστυνομικών τμημάτων, νοσοκομείων, και ασθενοφόρων. Και στις τρεις περιπτώσεις η χωροθέτηση είναι δυνατόν να μεταφραστεί σε αύξηση των πιθανοτήτων καταστροφής περιουσίας ή και απώλειας ζωής. Στον ιδιωτικό τομέα, οι βιοτεχνίες και οι βιομηχανίες θα πρέπει να χωροθετήσουν γραφεία, χώρους παραγωγής, η τοποθέτηση αποθηκών, κέντρα διανομής και λιανικής αγοράς αγαθών (retail outlet). Σε αυτήν την περίπτωση εσφαλμένες χωροθετικές αποφάσεις θα οδηγήσουν σε αύξηση του επενδυτικού κόστους και μείωση της ανταγωνιστικότητας της επιχείρησης. Επομένως, η επιτυχία ή η αποτυχία λειτουργιών του δημόσιου και του ιδιωτικού τομέα εξαρτάται σε σημαντικό βαθμό από τις τοποθεσίες που θα επιλεγούν για τις συγκεκριμένες λειτουργίες.

Υπάρχουν διάφορες μέθοδοι και τεχνικές που χρησιμοποιούνται για την ανάλυση και την ανεύρεση των βέλτιστων χωροθετικών προτύπων και της χωρικής αλληλεπίδρασης. Επιπρόσθετες εφαρμογές της Χωροθέτησης Κέντρων Παροχής Υπηρεσιών συναντώνται στην τοποθέτηση ηλεκτρονικών συνιστωσών (electronic components), σειρήνων συναγερμού, συστημάτων πυρόσβεσης, κεραιών ραντάρ, εξερευνητικών πετρελαιοπηγών κλπ. Αυτά ονομάζονται "facilities" (εγκαταστάσεις, κέντρα παροχής υπηρεσιών, παροχές, υπηρεσίες). Τα προβλήματα της Χωροθέτησης Κέντρων Παροχής Υπηρεσιών αποτελούν την αφορμή για τη λύση διαφόρων γεωμετρικών και συνδυαστικών προβλημάτων. Η έρευνα των προβλημάτων της Χωροθέτησης Κέντρων Παροχής Υπηρεσιών συνδέει πολλά ερευνητικά πεδία όπως τις εφαρμογές σε Ερευνητικές και Διοικητικές Επιστήμες, τη Μηχανολογία Μηχανικών σε βιομηχανίες, τη Γεωγραφία, τα Οικονομικά, την Επιστήμη των Υπολογιστών, τα Μαθηματικά, το marketing, την Ηλεκτρολογία Μηχανικών, τον Μη Γραμμικό Προγραμματισμό και άλλα σχετικά πεδία. Επιπλέον, η Χωροθέτηση Κέντρων Παροχής Υπηρεσιών είναι ένα σημαντικό στοιχείο για το σχεδιασμό στρατηγικών για ένα γενικό φάσμα των δημόσιων και ιδιωτικών προϊόντων. Όταν μια επιχείρηση θέλει να βγάλει στην αγορά κάποια νέα προϊόντα, ο κατασκευαστής θα πρέπει να διαλέξει τον τόπο που θα τοποθετηθεί μια αποθήκη.

ΚΕΦΑΛΑΙΟ 3

Περιγραφή μετροπροβλημάτων της OR library

Όπως είδαμε, υπάρχουν πολλοί αποτελεσματικοί αλγόριθμοι και προσεγγίσεις για την επίλυση του p-median προβλήματος. Βασικές μέθοδοι κατασκευής και βελτίωσης των αλγορίθμων αποτελούν διάφορα metaheuristics συμπεριλαμβανομένων την αναζήτηση Tabu, ευρετική συγκέντρωση, γενετικοί αλγόριθμοι, προσομοιωμένη ανόπτηση ή και κάποια Lagrangian προσέγγιση χαλάρωσης. Η OR library είναι μια συλλογή από σύνολα δεδομένων (benchmarks) διεθνούς βιβλιογραφίας που μπορούν να εξυπηρετήσουν μια ποικιλία προβλημάτων επιχειρησιακής έρευνας (OR: Operations/ Operational Research).

Σε αυτά τα σύνολα δεδομένων η πρόσβαση είναι εφικτή μέσω κάποιου browser, χρησιμοποιώντας τον ακόλουθο σύνδεσμο (Beasley a), όπου είναι συγκεντρωμένα διάφορα προβλήματα επιχειρησιακής έρευνας και είναι διαθέσιμα σε ερευνητές που εξετάζουν τόσο το p-median όσο και άλλα προβλήματα χωροθέτησης όπως:

- p-median - uncapacitated
- p-median - capacitated
- p-hub
- capacitated warehouse location
- uncapacitated warehouse location

Για την παρούσα διπλωματική εργασία, επιλέχτηκε η πρώτη κατηγορία p-median - uncapacitated από τις παραπάνω. Τα προβλήματα παρουσιάστηκαν στη δημοσίευση του ερευνητή J.E.Beasley - "Μια σημείωση για την επίλυση μεγάλων p-προβλημάτων διαμέσων", (J.E.Beasley - "A note on solving large p-median problems", European Journal of Operational Research 21 (1985) 270-273.).

3.1 Μορφή των αρχείων της Βιβλιοθήκης

Η κατηγορία προβλημάτων χωροθέτησης p-median - uncapacitated περιέχει 40 μετροπροβλήματα (benchmarks) με αρχεία pmed1, pmed2, ..., pmed40, στα οποία ο αριθμός των κόμβων ποικίλλει από 100 έως 900, ενώ η τιμή του p έχει εύρος από 5 έως 200. Κάθε μία από τις 40 περιπτώσεις είναι ένας γράφος με αντίστοιχο πλήθος για p διαμέσους ή εγκατάστασεις. Κάθε κόμβος είναι ένας πελάτης και ουσιαστικα μιά δυνητική εγκατάσταση. Τέλος το κόστος της ανάθεσης ενός πελάτη σε μια εγκατάσταση είναι το μήκος της μικρότερης διαδρομής μεταξύ αυτών των κόμβων.

people.brunel.ac.uk/~mastjjb/jeb/orlib/files/

pmed1.txt	07-Sep-2004 12:46	2.1K
pmed2.txt	07-Sep-2004 12:46	2.1K
pmed3.txt	07-Sep-2004 12:46	2.1K
pmed4.txt	07-Sep-2004 12:46	2.1K
pmed5.txt	07-Sep-2004 12:46	2.1K
pmed6.txt	07-Sep-2004 12:46	9.3K
pmed7.txt	07-Sep-2004 12:46	9.3K
pmed8.txt	07-Sep-2004 12:46	9.3K
pmed9.txt	07-Sep-2004 12:46	9.2K
pmed10.txt	07-Sep-2004 12:46	9.2K
pmed11.txt	07-Sep-2004 12:46	21K
pmed12.txt	07-Sep-2004 12:46	21K
pmed13.txt	07-Sep-2004 12:46	21K
pmed14.txt	07-Sep-2004 12:46	21K
pmed15.txt	07-Sep-2004 12:46	21K
pmed16.txt	07-Sep-2004 12:46	39K
pmed17.txt	07-Sep-2004 12:46	39K
pmed18.txt	07-Sep-2004 12:46	39K
pmed19.txt	07-Sep-2004 12:46	39K
pmed20.txt	07-Sep-2004 12:46	39K
pmed21.txt	07-Sep-2004 12:46	61K
pmed22.txt	07-Sep-2004 12:46	61K
pmed23.txt	07-Sep-2004 12:46	61K
pmed24.txt	07-Sep-2004 12:46	61K
pmed25.txt	07-Sep-2004 12:46	61K
pmed26.txt	07-Sep-2004 12:46	88K
pmed27.txt	07-Sep-2004 12:46	88K
pmed28.txt	07-Sep-2004 12:46	88K
pmed29.txt	07-Sep-2004 12:46	88K
pmed30.txt	07-Sep-2004 12:46	88K
pmed31.txt	07-Sep-2004 12:46	121K
pmed32.txt	07-Sep-2004 12:46	121K
pmed33.txt	07-Sep-2004 12:46	121K
pmed34.txt	07-Sep-2004 12:46	121K
pmed35.txt	07-Sep-2004 12:46	158K
pmed36.txt	07-Sep-2004 12:46	158K
pmed37.txt	07-Sep-2004 12:46	158K
pmed38.txt	07-Sep-2004 12:46	201K
pmed39.txt	07-Sep-2004 12:46	201K
pmed40.txt	07-Sep-2004 12:46	201K

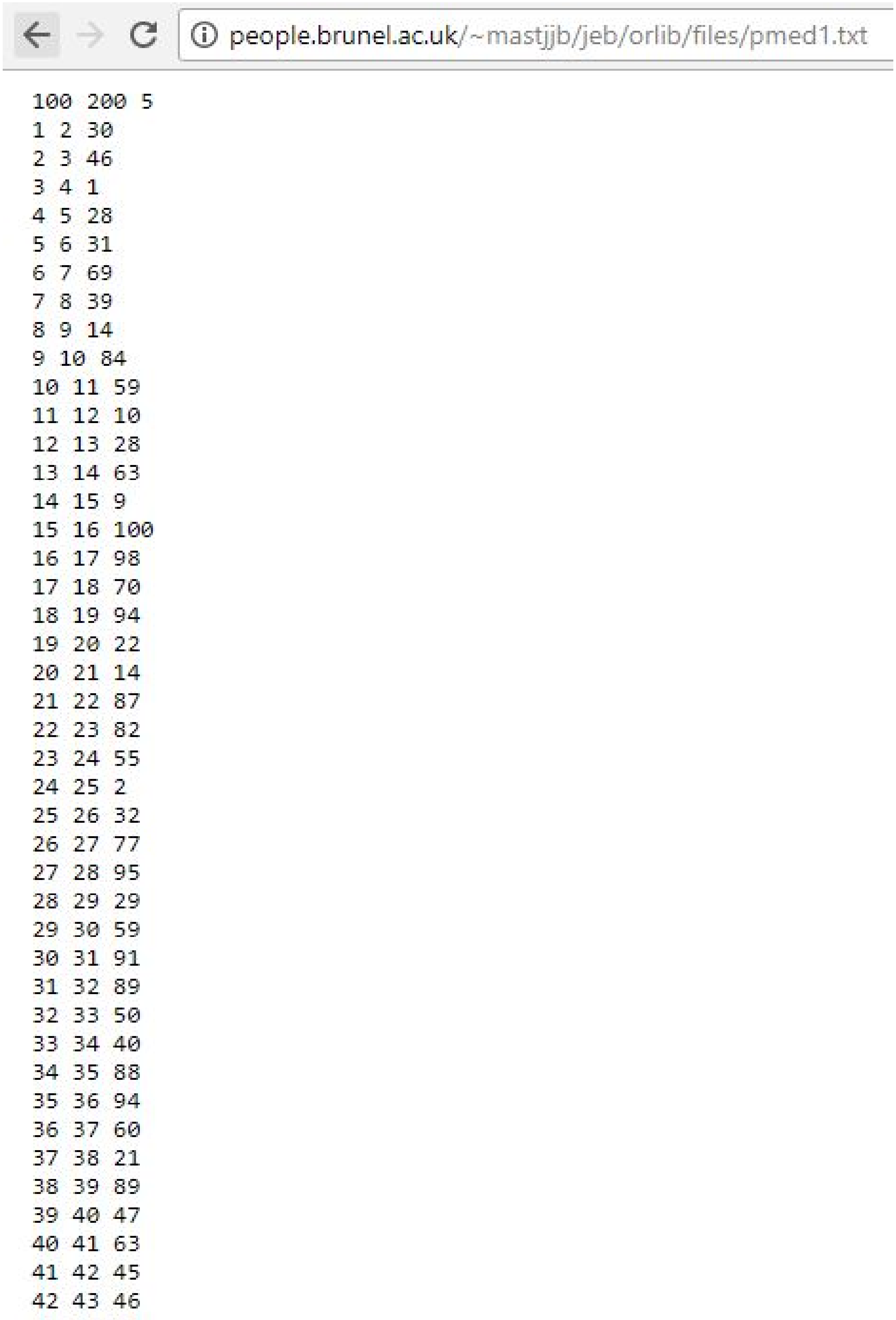
Σχήμα **3.1.1**: 40 p median benchmarks, OR Library

Τα 40 μετροπροβλήματα έχουν μορφή κειμένου .txt και περιλαμβάνουν ανά αρχείο τρεις στήλες. Πιο αναλυτικά, κάθε αρχείο της OR library διακρίνεται σε δύο τμήματα. Το πρώτο τμήμα ορίζει τις βασικές μεταβλητές του κάθε προβλήματος που περιλαμβάνονται στην πρώτη γραμμή κάθε αρχείου ως εξής:

- το πλήθος των κορυφών-πελατών n, βρίσκεται στη θέση: πρώτη γραμμή-πρώτη στήλη
- τον αριθμός ακμών m, βρίσκεται στη θέση: πρώτη γραμμή-δεύτερη στήλη
- το πλήθος των διάμεσων σημείων p που θα εκλεγούν ως εγκαταστάσεις, βρίσκεται στη θέση: πρώτη γραμμή-τρίτη στήλη

Το δεύτερο τμήμα περιμαβάνει ρητά δεδομένα. Συνεπώς, στις υπόλοιπες γραμμές πλήν της πρώτης του κάθε αρχείου σκιαγραφείται για κάθε ακμή η αρχή, το τέλος και το κόστος της απόστασης από τον έναν κόμβο στον άλλον.

Ας δούμε παρακάτω το περιεχόμενο του πρώτου αρχείου pmed1.txt.




```
100 200 5
1 2 30
2 3 46
3 4 1
4 5 28
5 6 31
6 7 69
7 8 39
8 9 14
9 10 84
10 11 59
11 12 10
12 13 28
13 14 63
14 15 9
15 16 100
16 17 98
17 18 70
18 19 94
19 20 22
20 21 14
21 22 87
22 23 82
23 24 55
24 25 2
25 26 32
26 27 77
27 28 95
28 29 29
29 30 59
30 31 91
31 32 89
32 33 50
33 34 40
34 35 88
35 36 94
36 37 60
37 38 21
38 39 89
39 40 47
40 41 63
41 42 45
42 43 46
```

Σχήμα **3.1.2**: First p-median benchmark, OR Library

Για το οποίο από την πρώτη γραμμή συμπεραίνουμε ότι:

- το πλήθος των κόμβών του προβλήματος είναι $n = 100$
- ο αριθμός των ακμών είναι $m = 200$
- το πλήθος των διάμεσων σημείων είναι $p = 5$

Αντίστοιχα ως προς το δεύτερο τμήμα του αρχείου, δηλαδή από την δεύτερη γραμμή και κάτω, παρατηρούμε το κόστος της απόστασης από τον έναν κόμβο στον άλλον.

Αξίζει να σημειωθεί ότι η OR library προτείνει την χρήση του αλγορίθμου αποστάσεων του Floyd Warshall, ώστε να εφαρμοστεί στον πίνακα κόστους που δίνεται από το αρχείο δεδομένων προκειμένου να ληφθεί ο πλήρες πίνακας κατανομής κόστους. Ο αλγόριθμος Floyd Warshall είναι ένας αλγόριθμος ικανός να βρει την κοντινότερη διαδρομή μεταξύ κόμβων σε ένα σταθμισμένο γράφημα. Μια απλή εκτέλεση του αλγορίθμου θα βρει τα μήκη (αθροιστικά βάρη) των κοντινότερων διαδρομών μεταξύ όλων των κόμβων. Για τον υπολογισμό του κόστους αποστάσεων χρησιμοποιήθηκε ο αλγόριθμος Floyd Warshall (Scipy), που υπήρχε διαθέσιμος από το οικοσύστημα ανοιχτού λογισμικού SciPy, βασισμένο στην γλώσσα προγραμματισμού Python.

Τέλος σε ένα επιπρόσθετο αρχείο η OR library εμπεριέχει την βέλτιστη τιμή της αντικειμενικής συνάρτησης για κάθε ένα από αυτά τα αρχεία δεδομένων (διαθέσιμο αρχείο: pmedopt, στον ιστότοπο (Beasley b).

people.brunel.ac.uk/~mastjjb/jeb/orlib/files/pmedopt.txt

Data file	Optimal solution value
pmed1	5819
pmed2	4093
pmed3	4250
pmed4	3034
pmed5	1355
pmed6	7824
pmed7	5631
pmed8	4445
pmed9	2734
pmed10	1255
pmed11	7696
pmed12	6634
pmed13	4374
pmed14	2968
pmed15	1729
pmed16	8162
pmed17	6999
pmed18	4809
pmed19	2845
pmed20	1789
pmed21	9138
pmed22	8579
pmed23	4619
pmed24	2961
pmed25	1828
pmed26	9917
pmed27	8307
pmed28	4498
pmed29	3033
pmed30	1989
pmed31	10086
pmed32	9297
pmed33	4700
pmed34	3013
pmed35	10400
pmed36	9934
pmed37	5057
pmed38	11060
pmed39	9423
pmed40	5128

Σχήμα **3.1.3**: Βέλτιστες τιμές 40 p median benchmarks, OR Library

ΚΕΦΑΛΑΙΟ 4

Ανάπτυξη μοντέλου p-median με τον λύτη βελτιστοποίησης Gurobi

4.1 Gurobi Optimizer

Ο Gurobi Optimizer είναι ένα εμπορικό πακέτο βελτιστοποίησης που αναπτύχθηκε από την εταιρεία (Gurobi). Το όνομα Gurobi προέρχεται από τα αρχικά των ονομάτων των δημιουργών του:

- Zonghao Gu
- Edward Rothberg
- Robert Bixby

Το πρόγραμμα Gurobi γενικά υποστηρίζει τις ακόλουθες γλώσσες προγραμματισμού:

- C++
- C
- Java
- .NET
- MATLAB
- R
- Python.

Το πακέτο αυτό βελτιστοποίησης μπορεί να αντιμετωπίσει προβλήμματα τόσο γραμμικού όσο και μικτού ακέραιου προγραμματισμού, τετραγωνικού με ή χωρίς περιορισμούς. Στα πλαίσια της παρούσας διπλωματικής εργασίας, το μοντέλο p-median αναπτύχθηκε σε προγραμματιστικό περιβάλλον Python και επιλύθηκε με τη χρήση του λογισμικού βελτιστοποίησης Gurobi Optimizer 6.5.1 ((Gurobi), 2014).

4.2 Εκτέλεση του μοντέλου p-median σε Python με χρήση του λύτη βελτιστοποίησης Gurobi

Η βιβλιοθήκη OR library προσφέρει πληθώρα μετροπροβλημάτων, πιο συγκεκριμένα για το πρόβλημα διαμέσων υπάρχουν 40 διαθέσιμα προβλήματα που εμπεριέχουν και την πληροφορία της βέλτιστης τιμής της αντικειμενικής συνάρτησης για κάθε ένα από αυτά τα αρχεία δεδομένων. Αν και υπάρχει λοιπόν ήδη αρκετή πληροφορία σχετικά με τις διαστάσεις των προβλημάτων και τις βέλτιστες τιμές τους, θεώρησα ενδιαφέρουσα πρόκληση να μοντελοποιήσω το p-median πρόβλημα από την αρχή, ώστε να επιλύει αυτά τα 40 μετροπροβλήματα. Η μοντελοποίηση του προβλήματος διαμέσων ήταν μια καλή ευκαιρία ώστε να κατανοήσω σε βάθος το πρόβλημα αλλά και τους περιορισμούς του, καθώς και να αποκτήσω πρόσβαση σε περισσότερα δεδομένα που απορέουν από την επίλυση και ανάλυση του κάθε αρχείου. Και αυτό γιατί η γνώση μόνο της τιμής της αντικειμενικής συνάρτησης δεν είναι αρκετή για τον υπολογισμό και την μετέπειτα σύγκριση των αποτελεσμάτων βέλτιστων με προσεγγιστικών μεθόδων επίλυσης.

Το Gurobi Python interface μπορεί να χρησιμοποιηθεί με διάφορους τρόπους. Είναι η βάση του Gurobi Interactive Shell, όπου συνήθως χρησιμοποιείται για να λειτουργεί με τα υπάρχοντα μοντέλα. Μπορεί επίσης να χρησιμοποιηθεί για δημιουργία αυτόνομου προγράμματος, με τον ίδιο τρόπο που θα χρησιμοποιούσαμε μια άλλη γλώσσα προγραμματισμού. Για περισσότερες δυνατότητες, εγκαταστάθηκε και χρησιμοποιήθηκε και το Anaconda Python, το οποίο περιλαμβάνει το γραφικό περιβάλλον ανάπτυξης (Spyder). Χρησιμοποιήθηκε λοιπόν το (Spyder) ώστε να γραφθεί ένα αυτόνομο p-median μοντέλο που επιλύει ακριβώς τα 40 μετροπροβλήματα που προτείνει η OR library.

Αξίζει να σημειωθεί ότι μετά από την εκτέλεση και των 40 προβλημάτων της βιβλιοθήκη OR library, θα πρέπει να επιβεβαιώσουμε ότι το μαθηματικό μοντέλο p-median που δημιουργήθηκε βρίσκει ακριβώς τις τιμές της αντικειμενικής συνάρτησης καθώς επαληθεύτηκε από τη σύγκριση των αποτελεσμάτων του Gurobi λύτη με αυτών που προτείνει η OR library. Ο αναλυτικός πίνακας των αποτελεσμάτων και του χρόνου εντοπισμού των βέλτιστων λύσεων από το μοντέλο ακολουθεί στο επόμενο κεφάλαιο.

ΚΕΦΑΛΑΙΟ 5

Περιγραφή μεθευρετικών μεθοδολογιών

5.1 Περιγραφή άπληστης μεθόδου αρχικοποίησης Greedy

Για να λύσουμε οποιοδήποτε πρόβλημα βελτιστοποίησης μπορούμε αντί να παράξουμε μια αρχική τυχαία λύση, να δημιουργήσουμε μία αρχική εφικτή λύση από την αρχή, (Xiao 2015). Για το πρόβλημα p-median, μέσα από διάφορα κριτήρια εφικτότητας είναι δυνατόν αν ξεκινήσουμε με μία μερικώς εφικτή λύση όπου διαθέτει έναν κόμβο εγκαταστάσεων και στη συνέχεια προσθέτοντας επιμέρους εφικτούς νέους κόμβους μπορεί να οδηγηθούμε σε μία ολοκληρωμένη εφικτή λύση. Κάθε φορά ένας νέος κόμβος θα πρέπει:

- να προστεθεί στη λύση αν οδηγεί στην μεγαλύτερη μείωση της τιμής της αντικειμενικής συνάρτησης
- να ικανοποιούνται τα κριτήρια εφικτότητας προβλήματος
- τερματίζει όταν έχουν προσπελασθεί όλοι οι κόμβοι

Οι ευρετικές μέθοδοι κατασκευής και βελτίωσης αποτελούν συνήθως άπληστες τεχνικές. Ο απλούστερος αλγόριθμος είναι ο μυωπικός ή άπληστος αλγόριθμος προσθήκης. Η μέθοδος αυτή είναι γνωστή στην βιβλιογραφία ως άπληστη (Greedy), καθώς είναι μια γρήγορη τεχνική κατασκευής εφικτής λύσης ξεκινώντας από μία μερική λύση, λαμβάνοντας υπόψη την καλύτερη επιλογή ανά επανάληψη και ικανοποιώντας κάποιους περιορισμούς προτού προστεθούν νέοι κόμβοι και μέχρι να δημιουργηθεί μια αρχική, εφικτή και ολοκληρωμένη λύση. Η διαδικασία συνεχίζεται έως ότου η λύση περιλαμβάνει p εγκαταστάσεις. Θα πρέπει να τονίσουμε ότι η λύση που τελικά προτείνει η Greedy μέθοδος δεν είναι βέλτιστη, ωστόσο είναι μια αρχική, εφικτή και ολοκληρωμένη λύση που χρήζει περεταίρω βελτίωσης.

Παρακάτω ακολουθεί ο ψευδοκώδικας για τον Greedy αλγόριθμο για το p-median πρόβλημα χρησιμοποιώντας την αρχή της προσθήκης addition principle, όπως περιγράφει στο βιβλίο του ο ερευνητής

(Xiao 2015).

Input: distances, n, p

Output: Initial Constructed Solution provided by Greedy Algorithm:cost, p

initialization;

$median \longleftarrow []$;

$candidates \longleftarrow [0, 1, 2, ..., n]$;

for j *in range* (p) **do**

  $dmin \longleftarrow INF$;

  $imin \longleftarrow -1$;

  for i *in candidates* **do**

    $d \longleftarrow Evaluate(distances, median + [i], n)$;

    if $d < dmin$ **then**

      $dmin \longleftarrow d$;

      $imin \longleftarrow i$;

    end

  end

  remove *imin* from *candidates*;

  append *imin* to *median*;

end

cost=dmin;

p=median;

Algorithm 1: Greedy

Ακολουθεί η συνάρτηση αξιολόγησης, η αποία καλείται από τον παραπάνω άπληστο αλγόριθμο.

Input: distances, median+[i], n
Output: Returns the sum of distances from each node to its nearest facility: sumdist
Function *Evaluate(distances, median+[i], n)*

  $sumdist \longleftarrow 0$;

  $p = len(median)$;

  for i *in range* (n) **do**

    $dist0 = INF$;

    for j *in range* (p) **do**

      if $dist[i][median[j]] < dist0$ **then**

        $dist0 = dist[i][median[j]]$;

      end

    end

    $sumdist+ = dist0$;

  end

  return sumdist;

Algorithm 2: Evaluate Function

Ας δούμε ένα απλό πρόβλημα p-median με $p = 2$ και αριθμό κόμβων $n = 6$. Για κάθε ακμή δίνεται παρακάτω η αρχή, το τέλος και το κόστος της απόστασης από τον έναν κόμβο στον άλλον.

Πίνακας **5.1**: Πίνακας Κόστους Α

Αρχικός κόμβος	Τελικός κόμβος	Κόστος
0	1	6
0	5	6
1	3	3
1	4	7
2	3	6
2	5	4
3	2	3
3	0	5
4	0	2
4	3	3
5	1	8
5	3	6

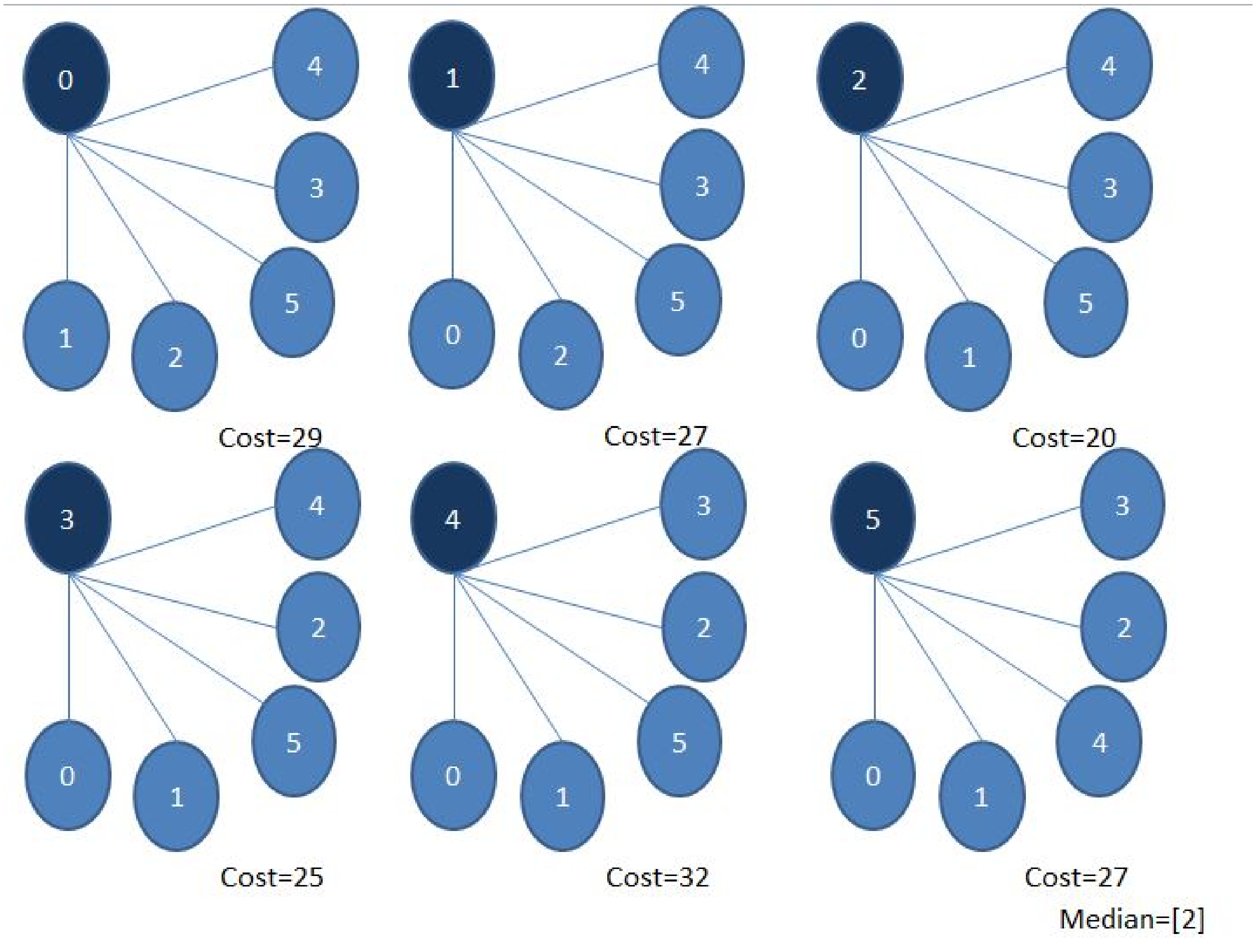
Μέσα από τον αλγόριθμο αποστάσεων του Floyd υπολογίζουμε τον πίνακα κόστους που δίνεται από το αρχείο δεδομένων προκειμένου να ληφθεί ο τελικός πίνακας κατανομής κόστους, ο οποίος δίνεται παρακάτω:

$$\begin{matrix} 0 & 6 & 3 & 6 & 8 & 6 \\ 6 & 0 & 3 & 6 & 4 & 8 \\ 3 & 3 & 0 & 3 & 6 & 5 \\ 6 & 6 & 3 & 0 & 8 & 2 \\ 8 & 4 & 6 & 8 & 0 & 6 \\ 6 & 8 & 5 & 2 & 6 & 0 \end{matrix}$$

Αν εκτελέσουμε την Greedy μέθοδο για το παραπάνω πρόβλημα ο αλγόριθμος μας επιστρέφει τα εξής αποτελέσματα:

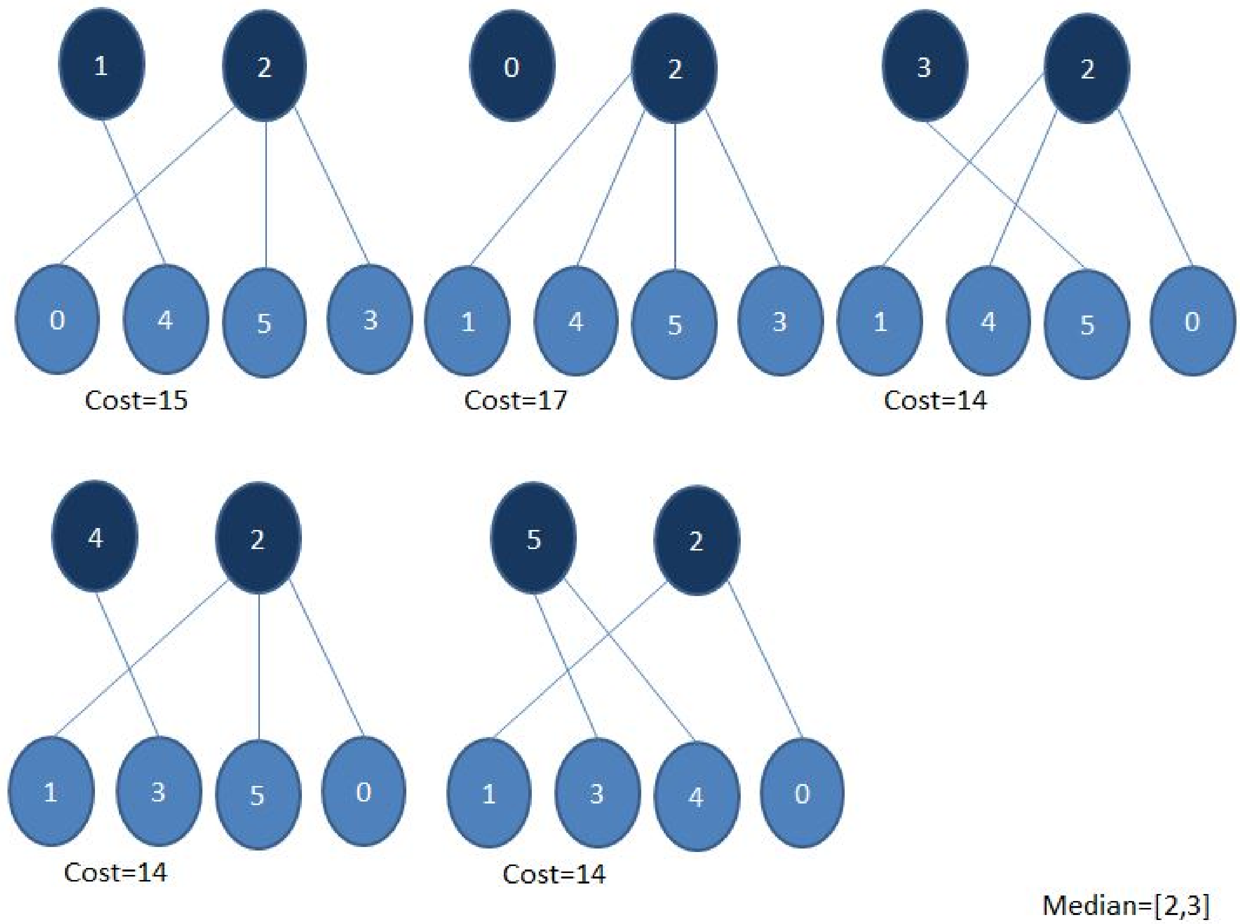
- Πρώτη τιμή διαμέσων επιλέχθηκε η $p = [2]$, με αντικειμενική τιμή $Cost = 20.0$
- Δεύτερη τιμή διαμέσων επιλέχθηκαν οι $p = [2, 3]$, με αντικειμενική τιμή $Cost = 25.0$
- $Time = 0.001$, ο χρόνος που απαιτήθηκε για να βρεθεί η λύση

Για την εύρεση της πρώτης τιμής διαμέσων ακολουθήθηκαν τα παρακάτω βήματα:



Σχήμα **5.1.1**: Επίλυση p=2 median με Greedy-Εύρεση πρώτης τιμής διαμέσων

Παρακάτω απεικονίζονται τα βήματα που ακολουθήθηκαν για την εύρεση και των δύο τιμών διαμέσων.



Σχήμα **5.1.2**: Επίλυση p=2 median με Greedy-Εύρεση δεύτερης τιμής διαμέσων

5.2 Περιγραφή μεθευρετικού αλγορίθμου-BVNS

Η αναζήτηση μεταβλητής γειτνίασης (Variable neighborhood search, VNS) συνδυάζει την τοπική αναζήτηση με δυναμικές δομές γειτονιάς οι οποίες μεταβάλλονται κατά τη διάρκεια της αναζήτησης. Η εξερεύνηση αυτών των γειτονιών μπορεί να γίνει με δύο τρόπους. Από τις μικρότερες, δηλαδή αυτές που βρίσκονται πλησιέστερα στην τρέχουσα λύση, διερευνούνται συστηματικά μέχρις ότου βρεθεί μια λύση καλύτερη από την τρέχουσα ή από τις μεγαλύτερες, δηλαδή εκείνες που απέχουν πολύ περισσότερο από την τρέχουσα λύση, μπορούν να διερευνηθούν εν μέρει από μία τυχαία λύση και ξεκινώντας μια τοπική αναζήτηση από αυτήν, (Mladenović, Brimberg, Hansen, and Moreno-Pérez 2007). Οι τελεστές τοπικής αναζήτησης και η αναπαράσταση της λύσης, ορίζουν ποιες λύσεις είναι γειτονικές. Με χρήση διαφορετικών δομών γειτονιάς, είναι εφικτός ο απεγκλωβισμός από τα τοπικά βέλτιστα και εξερεύνηση μεγαλύτερων περιοχών του χώρου αναζήτησης.

Η BVNS μεθευρετική μέθοδος παραμένει στην ίδια λύση μέχρις ότου βρεθεί μια άλλη καλύτερη λύση. Οι γειτονιές συνήθως κατατάσσονται με τέτοιο τρόπο ώστε να αναζητώνται λύσεις όλο και πιο μακριά από την τρέχουσα. Με αυτόν τον τρόπο, η εντατικοποίηση της αναζήτησης γύρω από την τρέχουσα λύση ακολουθείται από τη διαφοροποίηση. Στο επίπεδο της εντατικοποίησης συνδράμει η τοπική αναζήτηση ενώ η διατάραξη και η εναλλαγή γειτονιών συνδράμουν στη διαφοροποίηση. Στη φάση της διατάραξης (shaking) επιλέγεται μια τυχαία γειτονική λύση που προκύπτει με τη χρήση κάποιας γειτονιάς. Άρα, αποφεύγεται η κύκλωση και διερευνούνται νέες περιοχές του χώρου αναζήτησης. Ενώ η φάση της τοπικής αναζήτησης (local search) επιτρέπει την τοπική αναζήτηση μέχρι να εντοπιστεί ένα τοπικό βέλτιστο. Οταν βρεθεί, τότε γίνεται μετακίνηση στην επόμενη γειτονιά και ξεκινά η αναζήτηση από το καλύτερο τοπικό ελάχιστο που εντοπίστηκε.

Θα πρέπει να σημειωθεί ότι:

- Ένα τοπικό ελάχιστο μιας γειτονιάς δεν είναι τοπικό ελάχιστο για μια άλλη γειτονιά.
- Ένα ολικό ελάχιστο είναι ταυτόχρονα ολικά ελάχιστο για όλες τις δυνατές γειτονιές. Οι δομές γειτονιών αλλάζουν τον τρόπο με τον οποίο οι λύσεις είναι γειτονικές, αλλά όχι τη συνάρτηση αξιολόγησης(fitness).
- Οι τελεστές τοπικής αναζήτησης επηρεάζουν μόνο τα τοπικά βέλτιστα.

Παρακάτω ακολουθεί ο ψευδοκώδικας για τον BVNS αλγόριθμο για το p-median πρόβλημα μαζί με τις συναρτήσεις που καλεί κατά την εκτέλεσή του. Εφόσον αρχικοποιηθεί μια πρώτη εφικτή λύση με χρήση της άπληστης μεθόδου που παρουσιάστηκε στην προηγούμενη υποενότητα ακολουθεί η τεχνική BVNS, η οποία προσπαθεί να βελτιστοποιησεί την αρχική λύση περαιτέρω και σε εύλογο χρονικό διάστημα.

Input: x,kmax,tmax
Output: x
Function *BasicVNS(x,kmax,tmax)*

$t \longleftarrow 0$;
$x_(best) \longleftarrow x$;
while $t < tmax$ **do**
  $k \longleftarrow 1$;
  repeat
    $x' \longleftarrow Shaking(x, k)$;
    $x'' \longleftarrow LocalSearch(x')$;
    $x, k \longleftarrow NeighborhoodChange(x, x'', k)$;
    if x'' *is better than* x **then**
      $x_(best) \longleftarrow x''$;
      $x \longleftarrow x_(best)$;
    end
  until k=$kmax$;
  $t \longleftarrow CpuTime()$;
end
return x

Algorithm 3: BasicVNS

Input: x,k
Output: Shaking phase - generates a point w randomly from the k-th neighborhood of x denoted by $N_k(x)$, in other words chooses a randomly neighboring solution resulted from the use of the current neighborhood, in order to avoid cycling and thus exploring new areas: x'
Function *Shaking(x,k)*

$w \longleftarrow [1 + Random(0, 1) * \mid \ N_k(x) \mid]$;
$x' \longleftarrow x^w$;
return x'

Algorithm 4: Shaking

Input: x'

Output: Local search improves the current solution until reaching a local optimum. This is fulfilled by using a swap-based neighborhood search: x''

Function *LocalSearch(x')*

```
x ⟵ x';
n ⟵| N(x) | ;
for i ← 1 to n − 1 do
    for j ← i + 1 to n do
        if f(x^j) < f(x^i) then
            temp ⟵ f(x^i);
            f(x^i) ⟵ f(x^j);
            f(x^j) ⟵ temp;
        end
    end
    x'' ⟵ argmin(f(x), f(x^i)), x^i ∈ N(x);
end
return x''
```

Algorithm 5: Local Search

Θα πρέπει να σημειωθεί ότι η παραπάνω μέθοδος Basic VNS (BVNS) εχει δύο κριτήρια τερματισμού. Το πρώτο κριτήριο τερματισμού βασίζεται στον προκαθορισμένο χρόνο CPU(s), π.χ. 30, 60, 120, 240, 300, 600 δευτερόλεπτα. Το δεύτερο κριτήριο τερματισμού σχετίζεται με την βέλτιστη λύση. Συνεπώς αν κατά την διάρκεια εκτέλεσης του αλγορίθμου ξεπεραστεί ο προκαθορισμένος χρόνος εύρεσης λύσης ή εντοπιστεί η βέλτιστη λύση τότε ο αλγόριθμος τερματίζει.

Η μέθοδος Basic VNS (BVNS) συνδυάζει ντετερμινιστικές με στοχαστικές αλλαγές της δομής γειτνίασης.

- Το αιτιοκρατικό της τμήμα αντιπροσωπεύεται από την ευρετική τοπική αναζήτηση.
- Το στοχαστικό της τμήμα αντιπροσωπεύεται από την τυχαία επιλογή ενός σημείου από την κ-οστή γειτονιά, (το σημείο $x^{'}$ λαμβάνεται με τυχαίο τρόπο ώστε να αποφευχθεί η κύκλωση, η οποία θα μπορούσε να συμβεί με έναν αιτιοκρατικό κανόνα).

Έστω το ίδιο πρόβλημα p-median με $p = 2$ και αριθμό κόμβων $n = 6$ που περιγράφτηκε στο προηγούμενο κεφάλαιο. Δηλαδή θεωρούμε τις ίδιες ακμές:

Πίνακας **5.2**: Πίνακας Κόστους Β

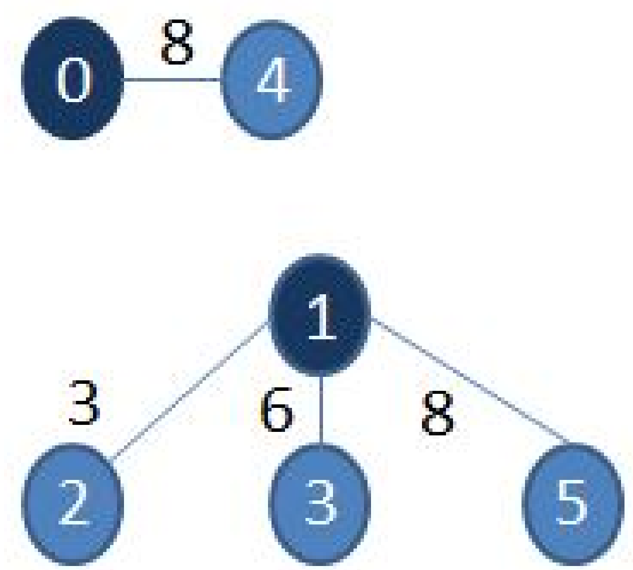
Αρχικός κόμβος	Τελικός κόμβος	Κόστος
0	1	6
0	5	6
1	3	3
1	4	7
2	3	6
2	5	4
3	2	3
3	0	5
4	0	2
4	3	3
5	1	8
5	3	6

και τον ίδιο τελικό πίνακα κατανομής κόστου:

$$\begin{matrix} 0 & 6 & 3 & 6 & 8 & 6 \\ 6 & 0 & 3 & 6 & 4 & 8 \\ 3 & 3 & 0 & 3 & 6 & 5 \\ 6 & 6 & 3 & 0 & 8 & 2 \\ 8 & 4 & 6 & 8 & 0 & 6 \\ 6 & 8 & 5 & 2 & 6 & 0 \end{matrix}$$

Έστω ότι ο αλγόριθμος ξεκινάει με μία εφικτή λύση όχι όμως βέλτιστη. Έστω λοιπόν η λύση:

- $p = [0, 1]$, οι διάμεσοι που επιλέχθηκαν
- $Cost = 25.0$, η αντικειμενική τιμή
- $Time = 0.000$, ο χρόνος που απαιτήθηκε για να βρεθεί η λύση

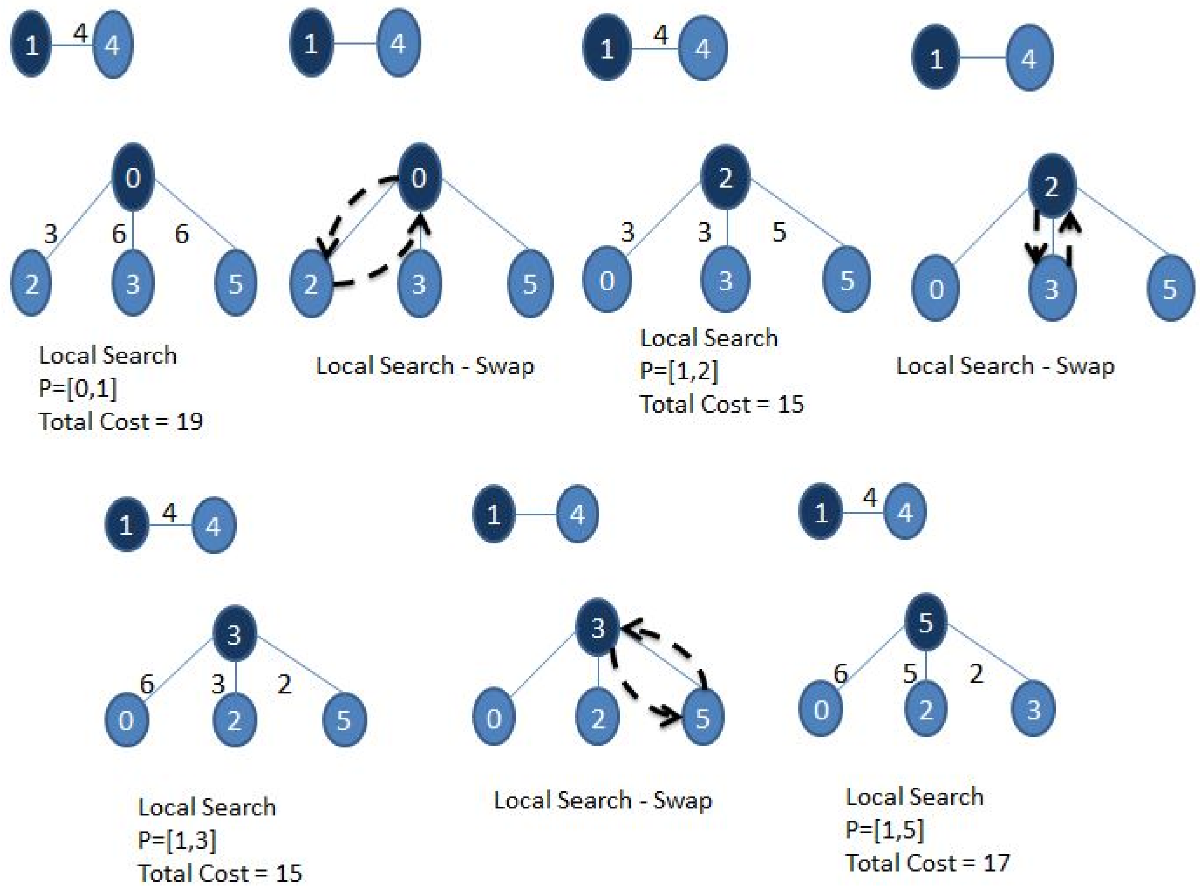


Σχήμα **5.2.3**: p=2 median λύση Greedy

Αυτή η λύση τροφοδοτεί τον αλγόριθμο VNS, ο οποίος μέσα από διατάραξη και τοπική αναζήτηση αποθηκεύει ως υπάρχουσα λύση αυτήν που θα έχει μικρότερη τιμή αντικειμενικής συνάρτησης από την προηγούμενη καλύτερη που είχε βρεθεί. Για το συγκεκριμένο παράδειγμα ο αλγόριθμος δέχεται ως αρχική εφικτή λύση την λύση που προκύπτει από την άπληστη τεχνική Greedy και έπειτα από δύο ενδιάμεσα καλές λύσεις κατόρθωσε να εντοπίσει μια λύση αρκετά κοντά στην βέλτιστη. Η BVNS μέθοδος για το παραπάνω πρόβλημα μας επιστρέφει διαδοχικά τα εξής αποτελέσματα:

1. Greedy: $Cost = 25.0$, $Time = 0.000$
2. $Cost = 19.0$, $Time = 0.000$
3. $Cost = 15.0$, $Time = 0.001$
4. $Cost = 15.0$, $Time = 0.001$
5. $p = [1, 2]$, $pmedianedges = 1 : [1, 4], 2 : [0, 2, 3, 5]$, $Cost = 15.0$, $Time = 0.001$

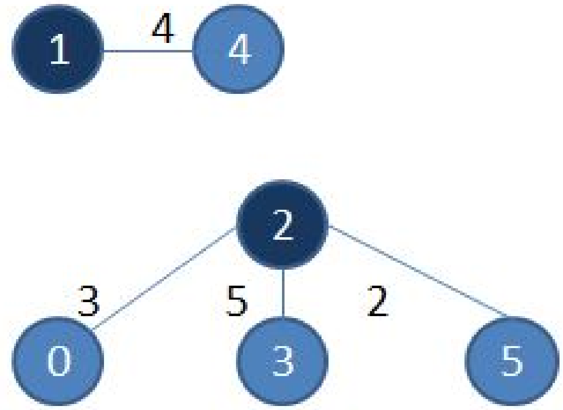
Κατά την διάρκεια εκτέλεσης του αλγορίθμου εκτελείται τοπική αναζήτηση μέχρι να εντοπίστεί ένα τοπικό βέλτιστο. Ένας αλγόριθμος Swap τοπικής αναζήτησης χρησιμοποιείται και παρακάτω απεικονίζεται βήμα-βήμα ο τρόπος που ανταλλάσσει τους κόμβους ώστε στο τέλος να κρατήσει την καλύτερη λύση τοπικά, δηλαδή την λύση που θα έχει την μικρότερη τιμή κόστους.



Σχήμα **5.2.4**: p=2 median Swap based local search

Θα πρέπει να αναφερθεί ότι η βέλτιστη λύση του παραπάνω προβλήματος είναι η ακόλουθη: $p = [2, 3]$, p median edges $= 2 : [0, 1, 2, 4], 3 : [3, 5]$, $Cost = 14.0$

Αν παρατηρήσουμε το κόστος της προσεγγιστικής λύσης, οδηγούμαστε στο συμπέρασμα ότι η προσεγγιστική λύση είναι αρκετά ικανοποιητική και βρέθηκε μετά από σχεδόν μηδενικό χρόνο εκτέλεσης του προσεγγιστικού αλγορίθμου.



Σχήμα **5.2.5**: p=2 median λύση BVNS

ΚΕΦΑΛΑΙΟ 6

Υπολογιστική μελέτη των αποτελεσμάτων μεθευρετικού αλγορίθμου

Στο παρόν κεφάλαιο αναλύονται τα υπολογιστικά στιγμιότυπα της βιβλιοθήκης OR library, με σκοπό να γίνει έλεγχος τόσο της μεθόδου αρχικοποίησης όσο και του μεθευρετικού αλγορίθμου που επιλέγχηκε. Τα πειράματα που ακολουθούν εκτελέστηκαν σε laptop Dell με τα εξής χαρακτηριστικά:

1. Επεξεργαστής: Intel(R) Core (TM)2 Duo i5-5200U CPU (2.20GHz 2.20GHz)
2. Εγκατεστημένη μνήμη: 4,00 GB
3. Λειτουργικό σύστημα: 64-bit operating system, Windows 10 Home Edition

Στην πρώτη υποενότητα αυτού του κεφαλαίου παρουσιάζονται τα αποτελέσματα εκτέλεσης του μαθηματικού μοντέλου p-median με το λύτη βελτιστοποίσης Gurobi. Θα ήταν χρήσιμο να τονιστεί σε αυτό το σημείο ότι το μαθηματικό μοντέλο του προβήματος p-median που επιλύθηκε στον λύτη Gurobi βρίσκει τις exact-βέλτιστες τιμές της αντικειμενικής συνάρτησης και παρακάτω παρατίθεται και ο χρόνος CPU(s) που χρειάστηκε το μοντέλου μέχρις ότου εντοπίσει την βέλτιστη λύση για κάθε ένα από τα 40 προβλήματα. Στην δεύτερη υποενότητα δίνονται τα αποτελέσματα της Greedy μεθόδου αρχικοποίησης, ενώ στη τρίτη αποδίδονται τα αποτελέσματα της Basic VNS υλοποίησης, η οποία δέχεται σαν είσοδο την λύση της Greedy μεθόδου αρχικοποίησης και την βελτιώνει περεταίρω αν βρεθεί κάποια λύση με μικρότερη τιμή της αντικειμενικής συνάρτησης.

6.1 Αποτελέσματα εκτέλεσης του μοντέλου p-median με το λύτη βελτιστοποίησης Gurobi

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, με σκοπό την καλύτερη κατανόηση του προβλήματος διαμέσων, μοντελοποιήθηκε το πρόβλημα σε γλώσσα προγραμματισμού Python και καταγράφτηκαν τα αποτελέσματα επίλυσης του με την βοήθεια του λύτη βελτιστοποίησης Gurobi. Το μαθηματικό μοντέλο εκτελέστηκε με χρήση του λύτη βελτιστοποίησης Gurobi για διαφορετικούς χρόνους τερματισμού. Με κριτήριο τερματισμού τους χρόνους 30, 60, 120, 240, 300, 600 CPU(s), όπου CPU(s) αντιπροσωπεύει τιμές δευτερολέπτων, μετρήθηκαν από τα 40 προβλήματα που έτρεξαν σειριακά για κάθε έναν από τους παραπάνω χρόνος, το πλήθος των λύσεων που ταυτίζονται με την βέλτιστη λύση της αντικειμενικής συνάρτησης και το πλήθος των περιπτώσεων που το μοντέλο κατάφερε να εξάγει προσεγγιστική ευρετική λύση. Θα πρέπει να σημειωθεί ότι για τους χρόνους τερματισμού που ισοδυναμούν με 30, 60 και 120 δευτερόλεπτα, το μοντέλο δεν κατάφερε να εντοπίσει βέλτιστη λύση και για τα 40 προβλήματα. Έτσι, σε χρόνο τερματισμού 30 δευτερολέπτων το μοντέλο δεν εντόπισε λύση για 4 προβλήματα από τα 40 της βιβλιοθήκης OR, ενώ σε χρόνο τερματισμού 60 και 120 δευτερολέπτων το πλήθος των άλυτων προβλημάτων μειώθηκε από 4 σε ένα πρόβλημα από τα 40 διαθέσιμα. Αυτό οδηγεί στο συμπέρασμα ότι ακόμη και ένας εμπορικός λύτης βελτιστοποίησης όπως είναι ο Gurobi, σε σχετικά μικρό χρονικό διάστημα όχι μόνο δεν επιστρέφει ευρετική λύση αλλά σε κάποια προβλήματα αδυνατεί να επιστρέψει γενικότερα λύση. Αυτός είναι και ο λόγος πού πολλές φορές για επίλυση προβλημάτων χωροθέτησης facility location , επιλέγεται μια ευρετική τεχνική επίλυσης, η οποία ακόμη και αν δεν εντοπίσει την βέλτιστη λύση, έχει σκοπό σε εύλογο χρονικό διάστημα να προσεγγίσει μία ικαποποιητικά συμφέρουσα λύση.

Στον παρακάτω πίνακα παρουσιάζεται για κάθε ένα από τα 40 αρχεία της OR library που επιλέχθηκαν να εξετασθούν στα πλαίσια αυτής της εργασίας κάποια χαρακτηριστικά του κάθε προβλήματος μαζί με το χρόνο που χρειάστηκε το μοντέλο μέχρι να εντοπίσει την βέλτιστη λύση και τελικά να τερματίσει. Θα πρέπει να σημειωθεί ότι για την δημιουργία του ακόλουθου πίνακα δεν χρησιμοποιήθηκε κάποιο κριτήριο τερματισμού βασισμένο στο χρόνο, καθώς σκοπός μας σε αυτήν την φάση ήταν να εξορύξουμε βασικές πληροφορίες των 40 προβλημάτων (π.χ. διαστάσεις, αριθμός κόμβων, αριθμός διαμέσων, κτλπ.) αλλά και του χρόνου που απαιτείται για τη εύρεση της βέλτιστης λύσης του κάθε προβλήματος. Ο πίνακας περιέχει τις εξής στήλες:

- το όνομα του αρχείου της OR library
- η βέλτιστη τιμή της αντικειμενικής συνάρτησης που είναι δημοσιευμένη στην OR library και ταυτίζεται με την αντικειμενική τιμή του μαθηματικού μοντέλου που αναπτύχθηκε στον λύτη βελτιστοποίησης Gurobi
- το πλήθος των κόμβων n του προβλήματος
- το πλήθος των διαμέσων p του προβλήματος
- ο χρόνος CPU σε δευτερόλεπτα που απαιτήθηκε ώστε ο λύτης Gurobi να επιλύσει βέλτιστα το πρόβλημα

Πίνακας **6.1**: Αποτελέσματα εκτέλεσης του μοντέλου p-median με το λύτη βελτιστοποίησης Gurobi

Αρχείο	Βέλτιστη τιμή	n	p	p-median CPU(s)
pmed1	5819	100	5	1,15
pmed2	4093	100	10	1,40
pmed3	4250	100	10	1,34
pmed4	3034	100	20	1,02
pmed5	1355	100	33	1,04
pmed6	7824	200	5	11,45
pmed7	5631	200	10	4,68
pmed8	4445	200	20	4,49
pmed9	2734	200	40	4,56
pmed10	1255	200	67	4,68
pmed11	7696	300	5	16,80
pmed12	6634	300	10	16,10
pmed13	4374	300	30	10,08
pmed14	2968	300	60	10,51
pmed15	1729	300	100	10,04
pmed16	8162	400	6	90,73
pmed17	6999	400	10	58,14
pmed18	4809	400	40	20,71
pmed19	2845	400	80	18,20
pmed20	1789	400	133	18,07
pmed21	9138	500	5	44,27
pmed22	8579	500	10	566,87
pmed23	4619	500	60	32,01
pmed24	2961	500	100	30,15
pmed25	1828	500	167	29,91
pmed26	9917	600	5	494,17
pmed27	8307	600	10	126,80
pmed28	4498	600	60	46,29
pmed29	3033	600	120	43,75
pmed30	1989	600	200	43,91
pmed31	10086	700	5	399,10
pmed32	9297	700	10	159,91
pmed33	4700	700	70	66,28
pmed34	3013	700	140	63,62
pmed35	10400	800	5	1015,98
pmed36	9934	800	10	12951,74
pmed37	5057	800	80	206,76
pmed38	11060	900	5	15389,33
pmed39	9423	900	10	2617,00
pmed40	5128	900	90	236,52

6.2 Αποτελέσματα άπληστης μεθόδου αρχικοποίησης Greedy

Η Greedy μέθοδος υλοποιήθηκε και για τα 40 αρχεία μετροπροβλημάτων της OR library. Τα αποτελέσματα δείχνουν ότι μπορεί να βρεθεί σχετικά γρήγορα μια εφικτή λύση, η οποία όμως δεν είναι βέλτιστη. Ο Greedy ευρετικός αλγόριθμος είναι μια γρήγορη και απλή τεχνική η οποία αν και δεν εγγυάται την εύρεση της βέλτιστης λύσης ενός προβλήματος, μπορεί σε εύλογο χρονικό διάστημα να βρει μια αρχική-εφικτή λύση.

Ο πρώτος πίνακας που δίνεται στην παρούσα ενότητα παρουσιάζει τα αποτελέσματα των 40 μετροπροβλημάτων της OR library με αρχική λύση τη Greedy μέθοδος. Επίσης, για κάθε αρχείο της OR library, γίνεται σύγκριση της αντικειμενικής τιμής που βρίσκει η άπληστη μεθόδος ως προς την βέλτιστη. Για την σύγκριση χρισιμοποιείται η μετρική του ποσοστιαίου σφάλματος, δηλαδή το σφάλμα ως ποσοστό της ακριβούς τιμής:

$$PercentageError = 100 * \frac{Greedy - Optimal}{Optimal}$$

Προφανώς, όσο μικρότερο είναι το σφάλμα τόσο ακριβέστερη είναι η τιμή της αντικειμενικής συνάρτησης της άπληστης τεχνικής για κάθε αρχείο της OR library.

Στη συνέχεια, ακολουθούν οι πίνακες με τα αποτελέσματα των πειραμάτων όπου περιλαμβάνουν:

- το όνομα του αρχείου της OR library
- η βέλτιστη τιμή της αντικειμενικής συνάρτησης που προσφέρει η OR library
- το πλήθος των κόμβων n
- το πλήθος των διαμέσων p
- η τιμή της αντικειμενικής συνάρτησης της Greedy μεθόδου
- ο χρόνος CPU(s) που χρειάστηκε η Greedy μέθοδος ώστε να βρει την λύση
- το ποσοστιαίο σφάλμα της διαφοράς της τελευταίας από την βέλτιστη λύση

Πίνακας **6.2**: Αποτελέσματα Greedy μέθοδου και σύγκριση με βέλτιστη τιμή

OR library				Greedy		
Αρχείο	Βέλτιστη τιμή	n	p	Αντικειμενική τιμή	CPU(s)	Error(%)
pmed1	5819	100	5	5891	0,02	1,24
pmed2	4093	100	10	4118	0,04	0,61
pmed3	4250	100	10	4399	0,02	3,51
pmed4	3034	100	20	3088	0,08	1,78
pmed5	1355	100	33	1378	0,07	1,70
pmed6	7824	200	5	8027	0,02	2,59
pmed7	5631	200	10	5646	0,05	0,27
pmed8	4445	200	20	4472	0,12	0,61
pmed9	2734	200	40	2841	0,30	3,91
pmed10	1255	200	67	1295	0,55	3,19
pmed11	7696	300	5	7721	0,03	0,32
pmed12	6634	300	10	6651	0,09	0,26
pmed13	4374	300	30	4467	0,39	2,13
pmed14	2968	300	60	3013	1,01	1,52
pmed15	1729	300	100	1761	2,08	1,85
pmed16	8162	400	6	8232	0,05	0,86
pmed17	6999	400	10	7019	0,13	0,29
pmed18	4809	400	40	4873	0,94	1,33
pmed19	2845	400	80	2899	2,81	1,90
pmed20	1789	400	133	1866	6,16	4,30
pmed21	9138	500	5	9138	0,07	0,00
pmed22	8579	500	10	8670	0,18	1,06
pmed23	4619	500	60	4694	2,11	1,62
pmed24	2961	500	100	3009	6,36	1,62
pmed25	1828	500	167	1896	17,69	3,72
pmed26	9917	600	5	10093	0,10	1,77
pmed27	8307	600	10	8364	0,27	0,69
pmed28	4498	600	60	4579	3,85	1,80
pmed29	3033	600	120	3104	15,66	2,34
pmed30	1989	600	200	2037	44,08	2,41
pmed31	10086	700	5	10086	0,11	0,00
pmed32	9297	700	10	9331	0,30	0,37
pmed33	4700	700	70	4798	8,56	2,09
pmed34	3013	700	140	3097	31,65	2,79
pmed35	10400	800	5	10406	0,12	0,06
pmed36	9934	800	10	9954	0,39	0,20
pmed37	5057	800	80	5118	14,09	1,21
pmed38	11060	900	5	11153	0,17	0,84
pmed39	9423	900	10	9451	0,46	0,30
pmed40	5128	900	90	5190	27,56	1,21

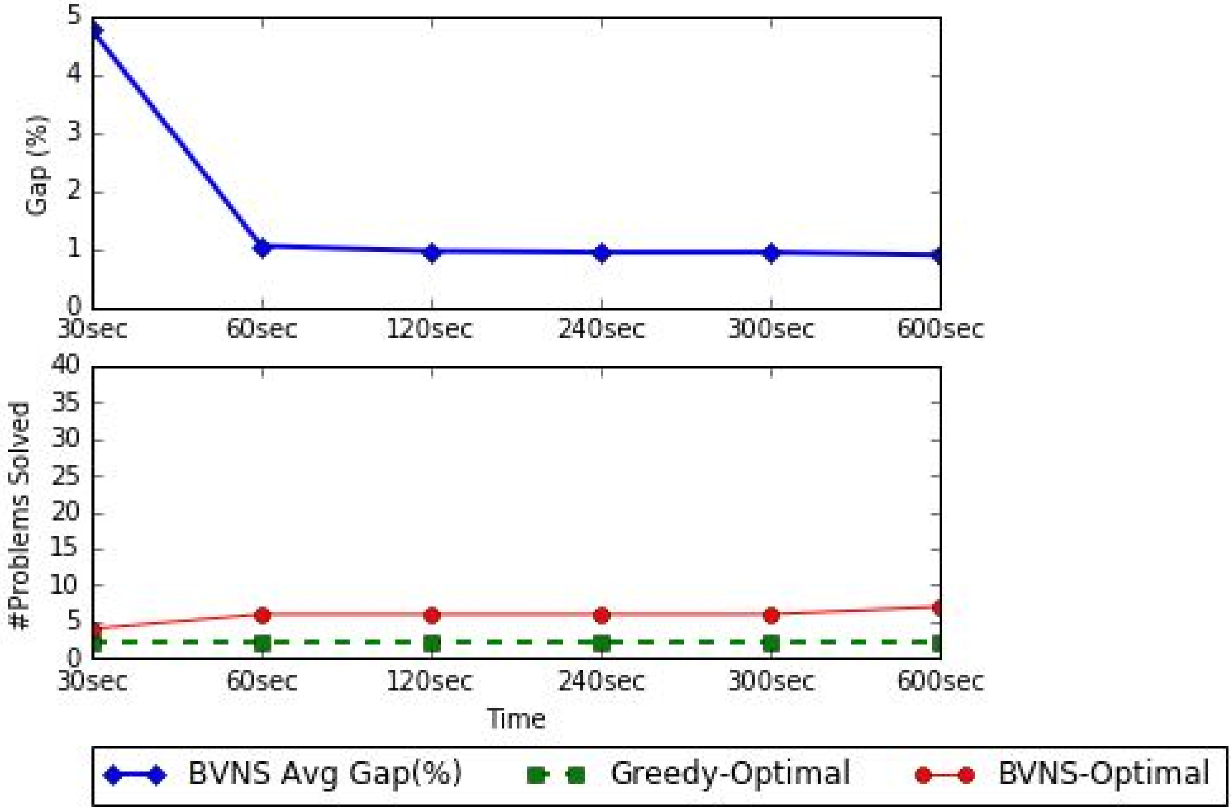
6.3 Περιγραφή μεθευρετικού αλγορίθμου BVNS

Οι επόμενοι πίνακες παρουσιάζουν τα αποτελέσματα των 40 μετροπροβλημάτων της OR library με τη λύση τη BVNS υλοποίσης ξεκινώντας από την αρχική λύση της Greedy μεθόδου. Επίσης, για κάθε αρχείο της OR library, γίνεται σύγκριση της αντικειμενικής τιμής που βρίσκει η BVNS μεθόδος ως προς την βέλτιστη για διαφορετικούς χρόνους τερματισμού του αλγορίθμου, που είναι: 30, 60, 120, 240, 300, 600 CPU(s). Για την σύγκριση χρισιμοποιείται η μετρική του ποσοστιαίου σφάλματος, δηλαδή το σφάλμα ως ποσοστό της ακριβούς τιμής:

$$PercentageError = 100 * \frac{BVNS-Optimal}{Optimal}$$

Όσο μικρότερο είναι το σφάλμα τόσο ακριβέστερη είναι η τιμή της αντικειμενικής συνάρτησης της BVNS υλοποίοησης για κάθε αρχείο της OR library.

Παρακάτω ακολουθεί ένα διπλό γράφημα γραμμής (line chart) των λύσεων της μεθόδου BVNS. Στο πάνω μέρος του γραφήματος, απεικονίζεται ο μέσος όρος του σφάλματος των λύσεων των 40 προβλημάτων για διαφορετικούς χρόνους τερματισμού 30, 60, 120, 240, 300, 600 CPU(s) που κατάφερε να εντοπίσει η μέθοδος BVNS. Η τεχνική BVNS για τους προαναφερθέντες χρόνους τερματισμού εντόπισε είτε ευρετικές είτε βέλτιστες λύσεις για αυτά τα 40 προβλήματα. Θα πρέπει να σημειωθεί ότι στον υπολογισμό του μέσου όρου σφάλματος που απεικονίζεται στο πάνω μέρος του ακόλουθου γραφήματος συμπεριλαμβάνονται και οι βέλτιστες και οι προσεγγιστικές λύσεις. Συνεπώς, η γραμμή που αντιπροσωπεύει το μέσο σφάλμα, (Average Gap (%), βλέπε Σχήμα 6.3.1, γραμμή Avg Gap (%)) εμπεριέχει και τα μηδενικά σφάλματα των προβλημάτων όπου εντοπίστηκαν βέλτιστες λύσεις. Στο κάτω τμήμα του γραφήματος, για τους ίδιους ακριβώς χρόνους τερματισμού, παρατηρούμε πόσα είναι σε πλήθος τα προβλήματα που ο αλγόριθμος εντόπισε βέλτιστες λύσεις με την άπληστη μέθοδο (βλέπε Σχήμα 6.3.1, γραμμή Greedy-Optimal) και πόσα βρήκε βέλτιστες λύσεις με την μέθοδο BVNS (βλέπε Σχήμα 6.3.1, γραμμή BVNS-Optimal). Όπως βλέπουμε στο γράφημα, η άπληστη μέθοδος για όλους τους προκαθορισμένους χρόνους τερματισμού κατάφερε να επιλύσει βέλτιστα μόνο 2 από τα 40 μετροπροβλήματα και θα ήταν χρήσιμο να αναφερθεί ότι για διαφορετικούς χρόνους τερματισμού 30, 60, 120, 240, 300, 600 το μέσο σφάλμα των λύσεων της άπληστης μεθόδου καταγράφηκε να είναι 4.3, 1.5, 1.5, 1.5, 1.5, 1.5 αντίστοιχα, συμπεριλαμβανομένων βέλτιστων και οι προσεγγιστικών λύσεων. Σε αντίθεση με το μέσο σφάλμα των λύσεων του BVNS που ήταν 4.8, 1.1, 1.0, 0.9, 0.9, 0.9. Η προηγούμενη σύγκριση μας οδηγεί στο συμπέρασμα ότι η μεθευρετική μέθοδολογία BVNS χρησιμοποιεί τα πλεονεκτήματα της άπληστης μεθόδου (καθώς αρχικοποιείται από αυτήν) και αποδεικνύεται πιο συμφέρουσα από την άπληστη και ως προς το πλήθος εύρεσης των βέλτιστων λύσεων, και ως προς το χρόνο εύρεσης τους αλλά και ως προς το μέσο σφάλμα των λύσεων της.



Σχήμα **6.3.1**: Γράφημα Γραμμής λύσεων του Greedy+BVNS

Αν αξιολογήσουμε το παραπάνω γράφημα γραμμής, το θετικό σημείο που πρέπει να τονιστεί πρώτα από όλα είναι ότι για όλους του χρόνους τερματισμού αποδίδει βέλτιστη ή ευρετική λύση, σε αντίθεση με τα αποτελέσματα του λύτη βελτιστοποίησης Gurobi όπου για τους χρόνους τερματισμού 30, 60 και 120 δευτερολέπτων δεν ήταν εφικτό να επιλυθούν 3, 1 και 1 προβλήματα αντίστοιχα από το σύνολο των 40 διαθέσιμων προβλημάτων. Πράγμα που σημαίνει ότι η προτεινόμενη μεθοδολογία Greedy+BVNS, ακόμη και αν δεν βρίσκει για όλα τα προβλήματα βέλτιστη λύση τουλάχιστον ως προς την αξιολόγηση λύσεων καταφέρνει να εντοπίσει συμφέρουσες χρονικά λύσεις για όλα τα μετροπροβλήματα που εξετάστηκαν. Περαιτέρω αξιολόγηση όμως θα πρέπει να γίνει ως προς την ποιότητα των λύσεων. Καθώς μια λύση για να είναι ικανοποιητική και αποδεκτή δεν αρκεί να είναι μόνο γρήγορη. Αλλά θα πρέπει να είναι όσο το δυνατόν πιό κοντά στην βέλτιστη. Το παραπάνω γράφημα γραμμής αποδεικνύει ότι ο αλγόριθμος BVNS σε χρόνο τερματισμού 30 δευτερολέπτων εντοπίζει λύσεις (βελτιστες και ευρετικές) και για τα 40 μετροπροβλήματα με μέσο σφάλμα 4.8% το οποίο σταδιακά μειώνεται στο ποσοστό 0.9% που αντιπροσωπεύει το μέσο σφάλμα των 40 λύσεων για χρόνο τερματισμού 600 δευτερόλεπτα. Τέλος, θα ήταν καλό να αναφερθεί ότι όσο αυξάνεται ο χρόνος τερματισμού αυξάνονται σε πλήθος και οι βέλτιστες λύσεις που εντοπίζει ο αλγόριθμος. Έτσι, με κριτήριο τερματισμού 30 δευτερολέπτων προσεγγίζονται 35 προβλήματα και επιλύονται 5 προβλήματα βέλτιστα, ενώ στα 600 δευτερόλεπτα προσεγγίζονται 33 προβλήματα και επιλύονται 7 προβλήματα βέλτιστα.

Στη συνέχεια, ακολουθούν οι πίνακες με τα αποτελέσματα των πειραμάτων τα οποία περιλαμβάνουν:

- το όνομα του αρχείου της OR library
- η βέλτιστη τιμή της αντικειμενικής συνάρτησης που προσφέρει η OR library
- το πλήθος των κόμβων n
- το πλήθος των διαμέσων p
- το ποσοστιαίο σφάλμα της διαφοράς της αντικειμενικής συνάρτησης της BVNS μεθόδου από την βέλτιστη λύση
- ο χρόνος CPU(s) που χρειάστηκε η BVNS μέθοδος ώστε να βρει λύση (βέλτιστη ή ευρετική)

Πίνακας **6.3**: Υπολογιστικά Αποτελέσματα της BVNS μέθοδου για CPU(s): 30, 60, 120

OR library				30sec		60sec		120sec	
Αρχείο	Βέλτιστη τιμή	n	p	Error(%)	CPU(s)	Error(%)	CPU(s)	Error(%)	CPU(s)
pmed1	5819	100	5	0,00	2,34	0,00	2,22	0,00	2,17
pmed2	4093	100	10	0,29	3,71	0,29	0,45	0,29	12,91
pmed3	4250	100	10	0,28	1,40	0,28	1,45	0,28	1,42
pmed4	3034	100	20	0,76	9,41	0,76	7,51	0,76	8,05
pmed5	1355	100	33	0,15	2,64	0,00	19,50	0,00	19,01
pmed6	7824	200	5	1,51	17,66	1,51	17,44	1,51	47,94
pmed7	5631	200	10	0,25	26,26	0,25	24,65	0,25	23,90
pmed8	4445	200	20	0,31	3,20	0,31	2,89	0,27	44,91
pmed9	2734	200	40	3,07	27,39	3,07	1,23	2,01	75,23
pmed10	1255	200	67	2,71	26,65	2,71	32,07	2,71	26,16
pmed11	7696	300	5	0,08	11,65	0,08	16,03	0,08	11,47
pmed12	6634	300	10	0,09	5,55	0,09	6,45	0,09	5,45
pmed13	4374	300	30	1,71	22,60	1,71	46,12	1,17	67,50
pmed14	2968	300	60	1,21	23,97	1,21	47,67	1,18	83,40
pmed15	1729	300	100	1,68	21,91	1,68	25,78	1,62	119,71
pmed16	8162	400	6	0,39	24,36	0,39	25,86	0,39	24,26
pmed17	6999	400	10	0,00	3,16	0,00	10,10	0,00	9,76
pmed18	4809	400	40	1,33	0,98	1,16	33,38	0,83	32,15
pmed19	2845	400	80	1,90	2,69	1,58	34,62	1,51	32,04
pmed20	1789	400	133	2,85	28,44	2,63	12,79	2,63	32,99
pmed21	9138	500	5	0,00	0,07	0,00	0,07	0,00	0,07
pmed22	8579	500	10	1,06	0,16	1,06	0,18	1,06	0,20
pmed23	4619	500	60	1,62	2,27	1,52	33,26	1,52	37,12
pmed24	2961	500	100	1,35	25,24	1,38	22,49	1,32	74,77
pmed25	1828	500	167	2,57	25,37	2,41	52,55	2,41	32,49
pmed26	9917	600	5	0,81	14,94	0,81	15,57	0,78	70,21
pmed27	8307	600	10	0,69	0,22	0,69	0,24	0,69	0,23
pmed28	4498	600	60	1,80	3,84	1,76	35,47	1,96	34,68
pmed29	3033	600	120	2,34	14,10	2,01	34,77	1,98	57,04
pmed30	1989	600	200	57,21	29,55	2,41	45,61	1,81	109,27
pmed31	10086	700	5	0,00	0,11	0,00	0,13	0,00	0,11
pmed32	9297	700	10	0,37	0,31	0,33	36,21	0,23	118,25
pmed33	4700	700	70	1,85	8,54	1,85	9,14	1,85	8,99
pmed34	3013	700	140	2,79	29,77	2,79	32,67	2,46	93,35
pmed35	10400	800	5	0,06	0,15	0,00	57,45	0,00	53,62
pmed36	9934	800	10	0,20	0,39	0,20	0,38	0,13	97,23
pmed37	5057	800	80	1,21	14,10	1,21	14,47	1,21	14,78
pmed38	11060	900	5	0,84	0,17	0,84	0,17	0,71	114,52
pmed39	9423	900	10	0,28	13,31	0,28	15,23	0,28	13,62
pmed40	5128	900	90	93,37	29,77	1,21	38,42	1,21	29,60

Πίνακας **6.4**: Υπολογιστικά Αποτελέσματα της BVNS μέθοδου για CPU(s): 240, 300, 600

OR library				240sec		300sec		600sec	
Αρχείο	Βέλτιστη τιμή	n	p	Error(%)	CPU(s)	Error(%)	CPU(s)	Error(%)	CPU(s)
pmed1	5819	100	5	0,00	0,24	0,00	2,17	0,00	2,20
pmed2	4093	100	10	0,29	0,45	0,29	0,42	0,29	3,69
pmed3	4250	100	10	0,28	1,43	0,28	13,23	0,28	1,70
pmed4	3034	100	20	0,76	7,35	0,76	7,18	0,76	7,60
pmed5	1355	100	33	0,00	19,64	0,00	18,92	0,00	20,13
pmed6	7824	200	5	1,51	34,66	1,51	47,70	1,51	64,91
pmed7	5631	200	10	0,25	24,51	0,25	24,13	0,25	25,38
pmed8	4445	200	20	0,27	112,92	0,27	95,35	0,27	2,73
pmed9	2734	200	40	2,01	83,55	1,98	81,27	1,98	84,17
pmed10	1255	200	67	2,71	27,78	2,71	34,01	2,31	328,89
pmed11	7696	300	5	0,08	11,78	0,08	11,30	0,08	11,69
pmed12	6634	300	10	0,09	5,58	0,09	5,38	0,09	5,72
pmed13	4374	300	30	1,17	76,96	1,14	75,64	1,14	69,77
pmed14	2968	300	60	1,18	159,21	1,21	296,78	1,01	339,89
pmed15	1729	300	100	1,62	56,59	1,62	21,97	1,50	79,54
pmed16	8162	400	6	0,39	24,18	0,39	23,43	0,39	24,72
pmed17	6999	400	10	0,00	9,78	0,00	9,46	0,00	9,99
pmed18	4809	400	40	0,83	206,25	0,83	199,58	0,83	218,15
pmed19	2845	400	80	1,51	67,98	1,51	166,31	1,12	364,29
pmed20	1789	400	133	2,63	124,29	2,63	10,43	2,63	74,69
pmed21	9138	500	5	0,00	0,07	0,00	0,07	0,00	0,07
pmed22	8579	500	10	1,05	210,73	1,05	211,49	1,05	219,02
pmed23	4619	500	60	1,47	77,94	1,47	30,95	1,47	510,66
pmed24	2961	500	100	1,32	163,39	1,32	8,22	1,32	78,28
pmed25	1828	500	167	2,41	26,13	2,41	24,29	2,35	357,36
pmed26	9917	600	5	0,78	71,37	0,78	68,16	0,78	69,74
pmed27	8307	600	10	0,28	149,82	0,28	145,04	0,28	548,32
pmed28	4498	600	60	1,33	200,28	1,33	201,38	1,33	208,94
pmed29	3033	600	120	1,98	33,21	1,98	32,90	1,95	395,52
pmed30	1989	600	200	1,81	73,99	1,91	100,44	1,71	541,18
pmed31	10086	700	5	0,00	0,11	0,00	0,11	0,00	0,12
pmed32	9297	700	10	0,23	112,64	0,23	113,63	0,23	114,93
pmed33	4700	700	70	1,85	9,18	1,85	8,66	1,79	245,89
pmed34	3013	700	140	2,46	95,18	2,46	92,89	2,42	221,57
pmed35	10400	800	5	0,00	64,74	0,00	54,00	0,00	55,61
pmed36	9934	800	10	0,13	99,60	0,08	278,60	0,08	292,83
pmed37	5057	800	80	1,09	138,50	1,09	137,69	1,07	156,68
pmed38	11060	900	5	0,71	109,40	0,71	108,82	0,71	118,12
pmed39	9423	900	10	0,28	12,93	0,28	13,01	0,00	554,82
pmed40	5128	900	90	1,21	30,29	1,21	26,05	1,17	448,57

ΚΕΦΑΛΑΙΟ 7

Επίλογος

Η διπλωματική αυτή εργασία πραγματοποιήθηκε στο πλαίσιο των μεταπτυχιακών σπουδών μου στο Τμήμα Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας και πραγματεύεται μεθόδους αρχικοποίησης με τη χρήση της αναζήτησης μεταβαλλόμενης γειτονιάς (VNS) για το πρόβλημα διαμέσων.

7.1 Συμπεράσματα

Με χρήση της μεθευρετικής μεθόδου VNS, ακριβέστερα της BVNS, οδηγηθήκαμε στην επίλυση του NP-hard προβλήματος διαμέσων. Κατ' επέκταση, στην ενότητα αυτή, μπορούμε να διατυπώσουμε κάποια συμπεράσματα, τόσο για τα γενικότερα τμήματα του θεωρητικού υπόβαθρου της εργασίας (p-median πρόβλημα, ανάπτυξη μαθηματικού μοντέλου στο Gurobi Optimizer, μεθευρετικές μέθοδοι, κτλπ.), όσο και για τα πιο εξειδικευμένα αποτελέσματα της εργασίας (επίλυση προβλήματος με Gurobi και με BVNS).

Στα πλαίσια της εργασίας αυτής, για την καλύτερη κατανόηση των περιορισμών αλλά και διαστάσεών του προβλήματος διαμέσων, πραγματοποιήθηκε η θεωρητική μελέτη και η μαθηματική μοντελοποιήθηκε του πρόβλημα σε γλώσσα προγραμματισμού Python. Για την καταγραφή των αποτελεσμάτων έγινε χρήση του λύτη βελτιστοποίησης Gurobi. Τα αποτελέσματα εκτέλεσης του μαθηματικού μοντέλου με τον λύτη βελτιστοποίησης Gurobi έδειξαν πως ο λύτης βελτιστοποίησης Gurobi είναι αρκετά αποτελεσματικός στην ποιοτητα των λύσεων που επιστρέφει, αδυνατεί ωστόσο σε μικρούς χρόνους να εντοπίσει λύσεις (βέλτιστες ή ευρετικές) για όλο το πλήθος των διαθέσιμων προβληματων της OR library που επιλέχτηκαν. Με άλλα λόγια, ο λύτης βελτιστοποίησης Gurobi σε σχετικά μικρό χρονικό διάστημα όχι μόνο δεν επιστρέφει προσεγγιστική λύση αλλά σε κάποια προβλήματα αδυνατεί να εντοπίσει γενικά λύση. Δεν συμβαίνει το ίδιο βέβαια για τον αλγόριθμο BVNS, ο οποίος ακόμη και σε μικρό χρονικό διάστημα είναι σε θέση να αποδόσει βέλτιστη ή ευρετική λύση. Ο BVNS κατάφερε να εντοπίσει συμφέρουσες χρονικά και ποιοτικά λύσεις για όλα τα μετροπροβλήματα που εξετάστηκαν, με το μέσο ποσοστιαίο σφάλμα να μειώνεται δραματικά όσο αυξάνεται το χρονικό κριτήριο τερματισμού.

Στο δυναμικό περιβάλλον της επιχειρησιασής έρευνας και της χωροθέτησης εγκαταστάσεων, όπου ο χρόνος και το κόστος αποτελούν βασικά συστατικά μέρη προς βελτιστοποίηση, αλγόριθμοι γρήγοροι και ποιοτικοί μπορούν να αποβούν αρκετά πλεονεκτικοί. Αυτός είναι και ο λόγος που στην εργασία αυτή αναπτύχθηκε και το μαθηματικό μοντέλο που εντοπίζει βέλτιστες λύσεις και μια μεθευρετική υλοποίηση, η οποία επιστρέφει βέλτιστες ή σχεδόν βέλτιστες λύσεις. Και οι δύο περιπτώσεις παρουσιάζουν πληθώρα θετικών και αρνητικών σημείων. Όταν όμως αυτά τα σημεία συνδέονται με σημαντικές επιχειρησιακές αποφάσεις, θα πρέπει η απόφαση της επιλογής μεθόδου να εξυπηρετεί οικονομικές, χρονικές και ποιοτικές ανάγκες.

7.2 Μελλοντική έρευνα

Στο πλαίσιο αυτής της διπλωματικής εργασίας, αναπτύχθηκε σε γλώσσα προγραμματισμού Python μια υλοποίηση της BVNS μεθευρετικής μεθόδου για την επίλυση του προβλήματος διαμέσων. Αυτή η υλοποίηση αφορά την αρχικοποίηση μιας γρήγορης εφικτής λύσης και τη περαιτέρω βελτιστοποίηση της λύσης αυτής με τεχνικές τοπικής αναζήτησης. Κατ' επέκταση, η βελτιστοποίηση της τοπικής αναζήτησης ακολουθώντας τις νέες τάσεις της έρευνας θα μπορούσε να αποτελέσει ένα πεδίο μελλοντικής έρευνας, μελέτης και ανάπτυξης. Ακόμη, αρκετά ενδιαφέρον παρουσιάζει και η ενδεχόμενη υλοποίηση παραλλαγών της VNS μεθόδου πιθανότατα σε συνδυασμό με μια τεχνική εξόρυξης δεδομένων data mining, ώστε να διερευνηθεί αν και κατά πόσο ένα σύνολο υπο-βέλτιστων λύσεων μπορεί να χρησιμοποιηθεί για να οδηγήσει μεθευρετικές μεθόδους όπως η BVNS, στην αναζήτηση ακόμη καλύτερων λύσεων σε απόδοση χρόνου και ποιότητας λύσεων.

ΑΝΑΦΟΡΕΣ

Beasley, J. E. Or-library data. `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`. [Last Accessed: 10/04/2018].

Beasley, J. E. Or-library optimal values. `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/pmedopt.txt`. [Last Accessed: 10/04/2018].

Beasley, J. E. (1990). Or-library: distributing test problems by electronic mail. *Journal of the operational research society 41*(11), 1069–1072.

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society 60*(6), 503–515.

Beltran, C., C. Tadonki, and J. P. Vial (2006). Solving the p-median problem with a semi-lagrangian relaxation. *Computational Optimization and Applications 35*(2), 239–260.

Captivo, M. E. (1991). Fast primal and dual heuristics for the p-median location problem. *European Journal of Operational Research 52*(1), 65–74.

Cornuejols, G., M. L. Fisher, and G. L. Nemhauser (1977). Exceptional paper - location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management science 23*(8), 789–810.

Daskin, M. S. (2011). *Network and discrete location: models, algorithms, and applications.* John Wiley & Sons.

Daskin, M. S. and K. L. Maass (2015). The p-median problem. In *Location science*, pp. 21–45. Springer.

Densham, P. J. and G. Rushton (1992). A more efficient heuristic for solving largep-median problems. *Papers in Regional Science 71*(3), 307–329.

Dorigo, M., V. Maniezzo, and A. Colorni (1991). The ant system: An autocatalytic optimizing process.

Dörrie, H. (2013). *100 great problems of elementary mathematics.* Courier Corporation.

Drezner, Z. and H. W. Hamacher (2001). *Facility location: applications and theory.* Springer Science & Business Media.

Erdoğan, G., G. Laporte, and A. M. R. Chía (2016). Exact and heuristic algorithms for the hamiltonian p-median problem. *European Journal of Operational Research 253*(2), 280–289.

Erkut, E. and B. Bozkaya (1999). Analysis of aggregation errors for the p-median problem. *Computers & Operations Research 26*(10-11), 1075–1096.

Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research 26*(10), 992–1009.

Feldman, E., F. Lehrer, and T. Ray (1966). Warehouse location under continuous economies of scale. *Management Science 12*(9), 670–684.

Feo, T. A. and M. G. Resende (1995). Greedy randomized adaptive search procedures. *Journal of global optimization 6*(2), 109–133.

Francis, R. L., T. J. Lowe, and A. Tamir (2002). Worst-case incremental analysis for a class of p-facility location problems. *Networks 39*(3), 139–143.

Galvão, R. D. (1980). A dual-bounded algorithm for the p-median problem. *Operations Research 28*(5), 1112–1121.

Galvão, R. D. (1993). The use of lagrangean relaxation in the solution of uncapacitated facility location problems. *Computers & Operations Research*.

Glover, F. (1989). Tabu search - part i. *ORSA Journal on computing 1*(3), 190–206.

Goldman, A. (1971). Optimal center location in simple networks. *Transportation science 5*(2), 212–221.

Goncharov, E. and Y. Kochetov (2002). Probabilistic tabu search for the unconstrained discrete optimization problems. *Discrete Analysis and Operations Research 9*(2), 13–30.

Gurobi. Gurobi optimization inc. management team. `http://www.gurobi.com/company/management-team`. Last Accessed: 10/04/2018.

Hale, J. Q., E. Zhou, and J. Peng (2017). A lagrangian search method for the p-median problem. *Journal of Global Optimization 69*(1), 137–156.

Hansen, P. and N. Mladenović (1997). Variable neighborhood search for the p-median. *Location Science 5*(4), 207–226.

Held, M. and R. M. Karp (1970). The traveling-salesman problem and minimum spanning trees. *Operations Research 18*(6), 1138–1162.

Hosage, C. and M. Goodchild (1986). Discrete space location-allocation solutions from genetic algorithms. *Annals of Operations Research 6*(2), 35–46.

Hribar, M. and D. M. S. (1997). A dynamic programming heuristic for the p-median problem. *European Journal of Operational Research 101*(3), 499–508.

Jalal, G. and J. Krarup (2003). Geometrical solution to the fermat problem with arbitrary weights. *Annals of Operations Research 123*(1-4), 67–104.

Kariv, O. and S. L. Hakimi (1979). An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics 37*(3), 539–560.

Kochetov, Y. (2001). Probabilistic local search algorithms for the discrete optimization problems. *Discrete Mathematics and Applications, Moscow, MSU 84117*.

Kochetov, Y., T. Levanova, E. Alekseeva, and M. Loresh (2005). Large neighborhood local search for the p-median problem. *Yugoslav Journal of Operations Research 15*(1), 53–63.

Krömer, P. and J. Platoš (2014). New genetic algorithm for the p-median problem. In *Intelligent Data analysis and its Applications, Volume II*, pp. 35–44. Springer.

Kuehn, A. A. and M. J. Hamburger (1963). A heuristic program for locating warehouses. *Management science 9*(4), 643–666.

Laguna, M. and R. Marti (1999). Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing 11*(1), 44–52.

Land, A. H. and A. G. Doig (1960). An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, 497–520.

Martins, D., G. M. Vianna, I. Rosseti, S. L. Martins, and A. Plastino (2014). Making a state-of-the-art heuristic faster with data mining. *Annals of Operations Research*, 1–22.

Merino, E. D. and J. M. Perez (2002). An efficient neural network algorithm for the p-median problem. In *Ibero-American Conference on Artificial Intelligence*, pp. 460–469. Springer.

Mladenović, N., J. Brimberg, P. Hansen, and J. A. Moreno-Pérez (2007). The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research 179*(3), 927–939.

Moreno, J., C. Rodríguez, and N. Jiménez (1991). Heuristic cluster algorithm for multiple facility location-allocation problem. *RAIRO-Operations Research 25*(1), 97–107.

Murray, A. T. and R. L. Church (1996). Applying simulated annealing to location-planning models. *Journal of Heuristics 2*(1), 31–53.

Neema, M., K. Maniruzzaman, and A. Ohgai (2011). New genetic algorithms based approaches to continuous p-median problem. *Networks and Spatial Economics 11*(1), 83–99.

Owen, S. H. and M. S. Daskin (1998). Strategic facility location: A review. *European journal of operational research 111*(3), 423–447.

Papadimitriou, C. and K. Steiglitz (1982). Combinatorial optimization: Algorithms and complexity. prentice-hall: Englewood cliffs nj. *Cited Figure 2*.

Perez, J. M., J. R. Garcia, and M. Moreno (1994). A parallel genetic algorithm for the discrete p-median problem. *Studies in Locational Analysis 7*, 131–141.

Pizzolato, N. D. (1994). A heuristic for large-sizep-median location problems with application to school location. *Annals of operations research 50*(1), 473–485.

Plastino, A., R. Fuchshuber, S. d. L. Martins, A. A. Freitas, and S. Salhi (2011). A hybrid data mining metaheuristic for the p-median problem. *Statistical Analysis and Data Mining: The ASA Data Science Journal 4*(3), 313–335.

Rebreyend, P., L. Lemarchand, and R. Euler (2015). A computational comparison of different algorithms for very large p -median problems. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pp. 13–24. Springer.

Resende, M. G. and R. F. Werneck (2003). On the implementation of a swap-based local search procedure for the p-median problem. In *Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX 03)*, pp. 119–127.

Rolland, E., D. A. Schilling, and J. R. Current (1997). An efficient tabu search procedure for the p-median problem. *European Journal of Operational Research 96*(2), 329–342.

Rosing, K. E. and C. S. Revelle (1997). Heuristic concentration: Two stage solution construction. *European Journal of Operational Research 97*(1), 75–86.

Rosing, K. E., C. S. Revelle, and D. A. Schilling (1999). A gamma heuristic for the p-median problem. *European Journal of Operational Research 117*(3), 522–532.

Salhi, S. (1997). A perturbation heuristic for a class of location problems. *Journal of the Operational Research Society 48*(12), 1233–1240.

Salhi, S. (2002). Defining tabu list size and aspiration criterion within tabu search methods. *Computers & Operations Research 29*(1), 67–86.

Salhi, S. and R. Atkinson (1995). Subdrop: A modified drop heuristic for location problems. *Location Science 3*(4), 267–273.

Santos, L. F., S. L. Martins, and A. Plastino (2008). Applications of the dm-grasp heuristic: a survey. *International Transactions in Operational Research 15*(4), 387–416.

Scholar. Scholar google. `http://scholar.google.gr`. [Last Accessed: 10/04/2018].

Sciencedirect. Science direct. `http://sciencedirect.com`. [Last Accessed: 10/04/2018].

Scipy. Scipy sparse csgraph floyd warshall. `https://docs.scipy.org/doc/scipy/reference/generated`. [Last Accessed: 10/04/2018].

Springer. Springer. `http://link.springer.com`. [Last Accessed: 10/04/2018].

Taillard, É. D. (2003). Heuristic methods for large centroid clustering problems. *Journal of Heuristics 9*(1), 51–73.

Tamir, A. (1996). An o (pn2) algorithm for the p-median and related problems on tree graphs. *Operations Research Letters 19*(2), 59–64.

Voss, S. (1996). A reverse elimination approach for the p-median problem. *Studies in Locational Analysis 8*, 49–58.

Weber, A. and G. Pick (1909). *Über den Standort der Industrien: Reine Theorie des Standorts*. JCB Mohr.

Whitaker, R. (1983). 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.

Xiao, N. (2015). *GIS algorithms*. Sage.

Zouein, P., H. Harmanani, and A. Hajar (2002). Genetic algorithm for solving site layout problem with unequal-size and constrained facilities. *Journal of computing in civil engineering 16*(2), 143–151.

Zouein, P. and I. Tommelein (1999). Dynamic layout planning using a hybrid incremental solution method. *Journal of construction engineering and management 125*(6), 400–408.