Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Πληροφορικής

# Υλοποίηση και υπολογιστική σύγκριση αλγορίθμων για προβλήματα χωροθέτησης μονάδων

Δημήτριος Τσακμάκης

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

5 Φεβρουαρίου 2021

# Περίληψη

Η παρούσα διπλωματική εργασία έχει ως αντικείμενο τη μελέτη αλγορίθμων για το πρόβλημα της χωροθέτησης μονάδων. Πιο συγκεκριμένα, εξετάζεται το διακριτό μοντέλο του p-μέσου στο οποίο η χωροθέτηση των μονάδων στους κόμβους ενός δοσμένου γραφήματος γίνεται με τρόπο τέτοιο ώστε να ελαχιστοποιείται η συνολική απόσταση των κόμβων ζήτησης από τις μονάδες που χωροθετούνται. Σε αυτό το πλαίσιο υλοποιήθηκε ένας αλγόριθμος ωμής βίας, ο οποίος εγγυάται βέλτιστη λύση σε μη πρακτικό χρόνο εκτέλεσης. Επιπλέον, επιλέχθηκε και υλοποιήθηκε ένα πλήθος ευρετικών μεθόδων (κατασκευαστικοί και βελτιωτικοί αλγόριθμοι) που λύνουν το πρόβλημα προσεγγιστικά. Τα αποτελέσματα των κατασκευαστικών και βελτιωτικών αλγορίθμων χρησιμοποιήθηκαν για τη σύγκρισή τους και για να αναδείξουν πόσο αποδοτικοί είναι ως προς το χρόνο που χρειάζονται για να λύσουν το πρόβλημα καθώς και ως προς την ακρίβεια της λύσης που βρίσκουν.

**Λέξεις κλειδιά:** χωροθέτηση μονάδων, πρόβλημα p-μέσου, ωμή βία, ευρετικές μέθοδοι, κατασκευαστικοί αλγόριθμοι, βελτιωτικοί αλγόριθμοι

# Abstract

The goal of this dissertation is to implement various algorithms for the facility location problem. More specifically, the discrete model of the p-medium is examined in which the units are located at the nodes of a given graph in such a way as to minimize the total distance of the demand nodes from the units being located. In this context, a brute force algorithm was implemented, which guarantees an optimal solution in impractical execution time. In addition, a number of heuristic methods (construction and improvement algorithms) were selected and implemented that approximately solve the problem. The results of the construction and improvement algorithms were used to compare them and to show how efficient they are in terms of the time they need to solve the problem as well as in terms of the accuracy of the solution they find.

**Keywords**: facility location, p-median problem, brute force, heuristic methods, constructive algorithms, improvement algorithms

# Δήλωση Πνευματικών Δικαιωμάτων



Υπογραφή Φοιτητή

# Περιεχόμενα

# Κατάλογος σχημάτων

# Κατάλογος αλγορίθμων

# Κατάλογος πινάκων

# Κεφάλαιο 1

# Εισαγωγή

## 1.1 Ορισμός του προβλήματος

Το πρόβλημα του $p$-μέσου ορίζεται ως η εύρεση $p$ σημείων εφοδιασμού έτσι ώστε να ελαχιστοποιείται η συνολική απόσταση μεταξύ των σημείων αυτών και ενός αριθμού πελατών. Το πρόβλημα αυτό είναι συνηθισμένο στην καθημερινή ζωή χωρίς να γίνεται αντιληπτό. Ο προσδιορισμός της θέσης μίας βιομηχανικής μονάδας (1-median problem) και η χωροθέτηση των νοσοκομείων σε μία πόλη αποτελούν στιγμιότυπα του ίδιου αυτού προβλήματος. Είτε πρέπει να μετακινηθούμε εμείς προς τους πελάτες είτε αυτοί προς εμάς το πρόβλημα παραμένει το ίδιο, αφού οι πελάτες θα προτιμήσουν την πλησιέστερη μονάδα σε ένα περιβάλλον ανταγωνισμού.

## 1.2 Κίνητρο για τη συγγραφή της εργασίας

Κίνητρο της παρούσας διπλωματικής εργασίας αποτελεί η ενασχόληση με το πρόβλημα της χωροθέτησης μονάδων και η υλοποίηση διάφορων αλγόριθμων για την επίλυσή του. Το πρόβλημα χωροθέτησης μονάδων παραγωγής είναι ένα πολύ σημαντικό πρόβλημα και οι αλγόριθμοι που έχουν υλοποιηθεί αρκετοί. Ωστόσο, μια σύγκριση μεταξύ αυτών των αλγορίθμων λείπει από τη βιβλιογραφία, οπότε αυτό το κενό αποσκοπεί να καλύψει αυτή η διπλωματική εργασία.

## 1.3 Σκοπός και στόχοι εργασίας

Σκοπός της εργασίας είναι η υλοποίηση διάφορων αλγορίθμων επίλυσης του προβλήματος του $p$-μέσου και, τελικά, η σύγκρισή τους ως προς την απόκλιση που αποδίδουν οι λύσεις τους από τις βέλτιστες λύσεις και ως προς τους χρόνους εκτέλεσης. Οι στόχοι της εργασίας είναι η εξοικείωση με το πρόβλημα της χωροθέτησης μονάδων, η εισαγωγή στον σύνθετο αλγοριθμικό τρόπο σκέψης για την επίλυση του προβλήματος και η κατανόηση των διαφορών μεταξύ διαφορετικών τρόπων επίλυσης.

## 1.4 Δομή της εργασίας

Στο δεύτερο κεφάλαιο παρουσιάζεται η μοντελοποίηση του προβλήματος του $p$-μέσου και αναφέρονται μέθοδοι που έχουν χρησιμοποιηθεί στο παρελθόν από άλλους ερευνητές που έχουν ασχοληθεί με το πρόβλημα. Στο τρίτο κεφάλαιο αναλύονται περιγραφικά και με τη χρήση ενός παραδείγματος οι αλγόριθμοι που υλοποιήθηκαν στο πλαίσιο της παρούσας διπλωματικής. Στο τέταρτο κεφάλαιο παρουσιάζονται τα αποτελέσματα του πειραματικού μέρους με χρήση πινάκων και γραφημάτων και συγκρίνονται οι διαφορετικές μέθοδοι επίλυσης του προβλήματος. Τέλος, στο πέμπτο κεφάλαιο δίνονται τα συμπεράσματα της εργασίας και προτείνονται μελλοντικές βελτιώσεις και υλοποιήσεις.

# Κεφάλαιο 2

# Βιβλιογραφική ανασκόπηση

## 2.1 Διατύπωση του προβλήματος του p-μέσου

Το πρόβλημα της χωροθέτησης μονάδων περιλαμβάνει τον προσδιορισμό της φυσικής θέσης βιομηχανικών μονάδων με στόχο να ελαχιστοποιηθεί το κόστος μεταφοράς ενός προϊόντος ή μιας υπηρεσίας σε σημεία ζήτησης. Υποκατηγορία αυτού αποτελεί το πρόβλημα του $p$-μέσου (PMP) το οποίο εξετάζεται στην παρούσα εργασία και ορίζεται ως εξής: δοθέντος $m$ πιθανών κόμβων εξυπηρέτησης και $n$ κόμβων ζήτησης σε ένα συνεκτικό γράφημα, να επιλεχθούν $p$ από τους $m$ κόμβους για τοποθέτηση βιομηχανικών μονάδων έτσι ώστε να ελαχιστοποιηθεί η συνολική βεβαρημένη με τη ζήτηση απόσταση μεταξύ των κόμβων εξυπηρέτησης και των κόμβων ζήτησης. Το κόστος για την εξυπηρέτηση της ζήτησης στον κόμβο $i$ προκύπτει ως το γινόμενο της ζήτησης του κόμβου αυτού επί την απόσταση του κόμβου από την πλησιέστερη σε αυτόν μονάδα.

Παραδείγματα του προβλήματος αποτελούν η χωροθέτηση νοσοκομείων, αστυνομικών τμημάτων, πυροσβεστικών σταθμών, σχολείων, αποθηκών μιας επιχείρησης, καταστημάτων λιανικής πώλησης, πυρηνικών εργοστασίων. Οι Kariv και Hakimi [?] έδειξαν ότι το πρόβλημα ανήκει στην κατηγορία NP-hard σε ένα γράφημα, όμως σε ένα δέντρο μπορούμε να πάρουμε λύση σε πολυωνυμικό χρόνο. Επιπλέον, ο Hakimi [?] έδειξε ότι τουλάχιστον μία βέλτιστη λύση υπάρχει εάν χωροθετηθούν μονάδες σε υπάρχοντες κόμβους του γραφήματος.

## 2.2 Μοντελοποίηση

Παρακάτω διατυπώνεται η μαθηματική μοντελοποίηση για το πρόβλημα του p-μέσου [?]:

$A = \{1, 2, ..., m\}$: το σύνολο των υποψήφιων κόμβων εξυπηρέτησης

$B = \{1, 2, ..., n\}$: το σύνολο των κόμβων ζήτησης

$d_j$: η ζήτηση του κόμβου $j$

$c_{ij}$: κόστος εξυπηρέτησης του κόμβου $j$ από τον κόμβο $i$

$p$: ο αριθμός των μονάδων προς χωροθέτηση

$$y_i = \begin{cases} 1 & \text{αν τοποθετηθεί μονάδα στον κόμβο i} \\ 0 & \text{αν δεν τοποθετηθεί μονάδα στον κόμβο i} \end{cases}$$

$x_{ij}$: το μέρος της ζήτησης του κόμβου $j$ που εξυπηρετείται από τον κόμβο $i$

Αντικειμενική συνάρτηση:

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

Περιορισμοί:

$$\sum_{i=1}^{m} x_{ij} = 1, \forall j \in B \tag{2.1}$$

$$\sum_{i=1}^{m} y_i = p \tag{2.2}$$

$$x_{ij} - y_i \leq 0, \forall i \in A, \forall j \in B \tag{2.3}$$

$$y_i \in \{0, 1\}, \forall i \in A \tag{2.4}$$

$$x_{ij} \geq 0, \forall i \in A, \forall j \in B \tag{2.5}$$

Ο περιορισμός (2.1) εξασφαλίζει ότι όλη η ζήτηση στον κόμβο $j$ θα εξυπηρετηθεί. Ο περιορισμός (2.2) απαιτεί τη χωροθέτηση ακριβώς $p$ μονάδων. Ο περιορισμός (2.3) σημαίνει ότι οι κόμβοι ζήτησης μπορούν να ανατεθούν μόνο σε κόμβους που έχουν επιλεχθεί ως εξυπηρετητές. Εάν δεν υπήρχε ο περιορισμός (2.3) τότε το πρόβλημα ανήκει στην κατηγορία NP-Complete και είναι πολύ δύσκολο να λυθεί. Ο περιορισμός (2.4) ορίζει τη μεταβλητή $y_i$ ως δυαδική μεταβλητή και, τέλος, ο περιορισμός (2.5) απαγορεύει αρνητικές τιμές στις μεταβλητές $x_{ij}$. Το πλήθος των πιθανών λύσεων προκύπτει ότι είναι $\binom{m}{p}$.

## 2.3 Μέθοδοι επίλυσης

Ως πρόβλημα βελτιστοποίησης υπάρχουν δύο κατηγορίες μεθόδων για την επίλυση του p-μέσου, οι ακριβείς μέθοδοι και οι ευρετικές μέθοδοι. Οι ακριβείς μέθοδοι βρίσκουν τη βέλτιστη λύση όμως δεν είναι πρακτικές για μεγάλα προβλήματα, ενώ οι ευρετικές βρίσκουν γρήγορα μία προσεγγιστική λύση.

Ο Hakimi [?] χρησιμοποίησε την απαρίθμηση, μια ακριβή μέθοδο, για να λύσει ένα πρόβλημα 10 κόμβων με τρεις μονάδες προς χωροθέτηση. Η μέθοδος της απαρίθμησης εγγυάται τη βέλτιστη λύση όμως σε μεγάλα δίκτυα ο υπολογισμός όλων των συνδυασμών θα χρειαζόταν χρόνια. Οι Kariv και Hakimi [?] παρουσίασαν έναν $O(p^2n^2)$ αλγόριθμο για το πρόβλημα του p-μέσου σε ένα δέντρο. Ο Tamir [?] βελτίωσε τον υπολογιστικό χρόνο και παρουσίασε έναν $O(pn^2)$ αλγόριθμο για το πρόβλημα σε ένα δέντρο. Οι ReVelle και Swain [?], χρησιμοποιώντας μια άλλη ακριβή μέθοδο, διατύπωσαν το πρόβλημα του p-μέσου ως ένα πρόβλημα μικτού ακέραιου προγραμματισμού. Οι Rosing, ReVelle, και Rosing-Vogelaar [?] απλοποίησαν ένα μοντέλο του προβλήματος απαλείφοντας τις γραμμές και στήλες όπου δεν υπήρχε ανάθεση ζήτησης. Ο Church [?] εξάλειψε τις αναθέσεις για ζήτηση όπου η μονάδα ήταν πολύ μακριά για να εξυπηρετήσει τη ζήτηση.

Οι ευρετικές μέθοδοι παράγουν προσεγγιστικές λύσεις σε μικρό χρόνο και για αυτό το λόγο προτιμώνται στην επίλυση του προβλήματος. Υπάρχουν πολλές ευρετικές μέθοδοι που έχουν προταθεί για την επίλυση του προβλήματος καθώς και συνδιασμοί τους.

Η άπληστη μέθοδος, greedy heuristic (Kuehn και Hamburger) [?] ξεκινάει θεωρώντας κενό το σύνολο των κόμβων προς χωροθέτηση. Στη συνέχεια λύνει το 1-median πρόβλημα και προσθέτει τον πρώτο κόμβο στο σύνολο των κόμβων λύσης. Σε κάθε βήμα προστίθεται στη λύση (δηλαδή χωροθετείται μια νέα μονάδα) κάθε φορά ο κόμβος που μειώνει περισσότερο το συνολικό κόστος της αντικειμενικής συνάρτησης. Η διαδικασία επαναλαμβάνεται p φορές. Μια αποδοτική υλοποίηση παρουσίασε ο Robert Whitetaker [?].

Η μέθοδος greedy-drop [?], εναλλακτικά stingy heuristic, θεωρεί αρχικά ότι όλοι οι κόμβοι ανήκουν στη λύση. Αφαιρείται ένας κόμβος κάθε φορά μέχρι να μείνουν p κόμβοι στο σύνολο κόμβων λύσης. Ο κόμβος που αφαιρείται σε κάθε επανάληψη είναι αυτός που αυξάνει λιγότερο το συνολικό κόστος. Οι Salhi και Atkinson [?] υλοποίησαν μια τροποποιημένη έκδοση του αλγορίθμου όπου το σύνολο λύσης περιλαμβάνει, αρχικά, ένα υποσύνολο των κόμβων του γραφήματος. Μια ακόμη ευρετική μέθοδος, γνωστή ως dual ascent heuristic, προτάθηκε από τους Captivo [?] και Galvao [?]. Οι τρεις παραπάνω μέθοδοι ανήκουν στις κατασκευαστικές μεθόδους (constructive heuristics), δηλαδή παράγουν μια προσεγγιστική εφικτή λύση χωρίς να χρειάζονται μία αρχική λύση ως είσοδο.

Η εναλλακτική μέθοδος, alternating heuristic [?] θεωρεί $p$ χωροθετημένες μονάδες και αναθέτει τους κόμβους στην πλησιέστερη μονάδα. Σε κάθε γειτονιά λύνεται το 1-median πρόβλημα και εάν βρεθεί καλύτερος κόμβος λύσης στις γειτονιές τότε ανανεώνεται η αρχική λύση. Η διαδικασία επαναλαμβάνεται έως ότου να μην υπάρξει αλλαγή σε όλες τις γειτονιές.

Η μέθοδος αντικατάστασης κόμβου, interchange heuristic [?], γνωστή και ως Teitz and Bart, ξεκινά με μία υπάρχουσα λύση και σε κάθε βήμα αντικαθιστά έναν από τους p κόμβους της λύσης με τον κόμβο του γραφήματος που μειώνει περισσότερο το συνολικό κόστος, εάν υπάρχει τέτοιος κόμβος. Η μέθοδος αντικατάστασης καθώς και η εναλλακτική μέθοδος ανήκουν στην κατηγορία τοπικής αναζήτησης (local search).

Οι Wangshu Mu και Daoqin Tong [?] υλοποίησαν μία παραλλαγή του αλγορίθμου Teitz and Bart η οποία δίνει προτεραιότητα στην αναζήτηση κόμβων για αντικατάσταση βασισμένη στη χωρική πυκνότητα ζήτησης του δικτύου.

Η χαλάρωση Lagrange [?] [?] [?] είναι μία μεθευρετική μέθοδος που έχει χρησιμοποιηθεί για την επίλυση του προβλήματος. Η αναζήτηση Tabu [?] (Tabu search) και παραλλαγές της είναι άλλη μια μέθοδος που έχει προταθεί. Άλλες μεθευρετικές μέθοδοι που επιχειρούν να επιλύσουν το πρόβλημα του p-μέσου είναι η αναζήτηση μεταβλητής γειτονιάς [?] (Variable Neighborhood Search),

Ένας άλλος υβριδικός ευρετικός αλγόριθμος με την ονομασία GRASP with path relinking υλοποιήθηκε από τους Resende και Werneck [?] στον οποίο μία τυχαία κατασκευασμένη λύση χρησιμοποιεί στη συνέχεια τοπική αναζήτηση και σε συνδυασμό με μία διαδικασία επανασύνδεσης μονοπατιού παράγει μία τελική λύση. Άλλες μέθοδοι που χρησιμοποιήθηκαν είναι μεθευρετικές μέθοδοι Ant Colony [?], Simulated Annealing [?] [?] [?], Scatter search [?], Genetic programming [?] [?] [?] .

# Κεφάλαιο 3

# Παρουσίαση αλγορίθμων

Στο κεφάλαιο αυτό περιγράφονται οι αλγόριθμοι που υλοποιήθηκαν για το πρόβλημα της χωροθέτησης μονάδων. Ο αλγόριθμος ωμής βίας ανήκει στην κατηγορία των ακριβών μεθόδων, ενώ οι υπόλοιποι αλγόριθμοι ανήκουν στην κατηγορία των ευρετικών μεθόδων. Επιπλέον, για κάθε αλγόριθμο παρατίθεται ψευδοκώδικας και ένα αναλυτικό παράδειγμα. Το γράφημα είναι το ίδιο σε όλους τους αλγορίθμους ώστε να είναι κατανοητή και ευκολότερη η σύγκρισή τους.

## 3.1 Αλγόριθμος brute force

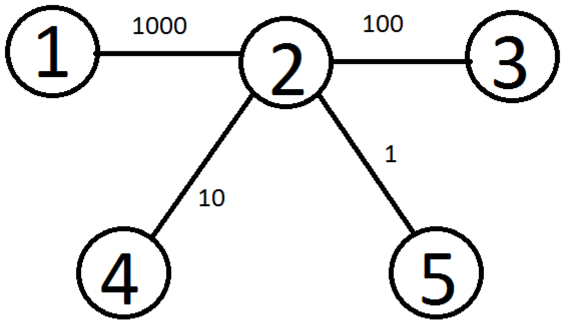
Ο αλγόριθμος ωμής βίας (brute force algorithm) είναι ο μόνος αλγόριθμος (από αυτούς που υλοποιήθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας) που εγγυάται την εύρεση της βέλτιστης λύσης, όμως με σημαντικά αυξημένο κόστος χρόνου εκτέλεσης για ρεαλιστικά προβλήματα. Έστω $F$ το σύνολο των πιθανών κόμβων προς χωροθέτηση ($m$ συμβολίζεται το πλήθος των στοιχείων του $F$), $S$ το σύνολο που περιέχει τους κόμβους λύσης και $p$ ο απαιτούμενος αριθμός μονάδων προς χωροθέτηση. Ο αλγόριθμος υπολογίζει όλους τους συνδυασμούς κόμβων $m$ ανά $p$ καθώς και τα αντίστοιχα κόστη των συνδυασμών αυτών. Ως λύση επιλέγεται ο συνδυασμός που αποδίδει το μικρότερο κόστος ($C$). Η πολυπλοκότητα του αλγορίθμου που υλοποιήθηκε είναι $m!/p!(m-p)!$

### 3.1.1 Παράδειγμα αλγορίθμου brute force

Όπως ήδη αναφέρθηκε, όλοι οι αλγόριθμοι θα παρουσιαστούν μέσω του παραδείγματος που βρίσκεται στην Εικόνα 3.1.

Τα σύνολα $F$ και $S$ είναι:

Σχήμα 3.1: Γράφημα παραδειγμάτων για το πρόβλημα του $p$-μέσου




$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Ο αλγόριθμος υπολογίζει τους παρακάτω συνδυασμούς και τα αντίστοιχα κόστη τους. Ο συνδυασμός με το ελάχιστο κόστος αποτελεί το σύνολο λύσης $S$ ($C_{i,j}$ συμβολίζεται το κόστος εξυπηρέτησης του κόμβου $i$ από τον κόμβο $j$):

$\{1, 2\} \rightarrow C = C_{1,1} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2} = 0 + 0 + 100 + 10 + 1 = 111$

$\{1, 3\} \rightarrow C = C_{1,1} + C_{2,3} + C_{3,3} + C_{4,3} + C_{5,3} = 0 + 100 + 0 + 110 + 101 = 311$

$\{1, 4\} \rightarrow C = C_{1,1} + C_{2,4} + C_{3,4} + C_{4,4} + C_{5,4} = 0 + 10 + 110 + 0 + 11 = 131$

$\{1, 5\} \rightarrow C = C_{1,1} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5} = 0 + 1 + 101 + 11 + 0 = 113$

$\{2, 3\} \rightarrow C = C_{1,2} + C_{2,2} + C_{3,3} + C_{4,2} + C_{5,2} = 1000 + 0 + 0 + 10 + 1 = 1011$

$\{2, 4\} \rightarrow C = C_{1,2} + C_{2,2} + C_{3,2} + C_{4,4} + C_{5,2} = 1000 + 0 + 100 + 0 + 1 = 1101$

$\{2, 5\} \rightarrow C = C_{1,2} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,5} = 1000 + 0 + 100 + 10 + 0 = 1110$

$\{3, 4\} \rightarrow C = C_{1,4} + C_{2,4} + C_{3,3} + C_{4,4} + C_{5,4} = 1010 + 10 + 0 + 0 + 11 = 1031$

$\{3, 5\} \rightarrow C = C_{1,5} + C_{2,5} + C_{3,3} + C_{4,5} + C_{5,5} = 1001 + 1 + 0 + 11 + 0 = 1013$

$\{4, 5\} \rightarrow C = C_{1,5} + C_{2,5} + C_{3,5} + C_{4,4} + C_{5,5} = 1001 + 1 + 101 + 0 + 0 = 1103$

Άρα τελικά θα εγκατασταθούν μονάδες στους κόμβους 1 και 2 με κόστος 111 το οποίο αποτελεί τη βέλτιστη λύση.

Λύση: $S = \{1, 2\}, C = 111$

### 3.1.2 Ψευδοκώδικας αλγορίθμου brute force

Στον πίνακα 1 δίνεται ο ψευδοκώδικας του αλγορίθμου brute force.

**Data**: Γράφημα $G = (V, E)$ (m σε πλήθος κορυφές) και $p$ το πλήθος των μονάδων προς χωροθέτηση
**Result**: Σύνολο $S$, $p$ στοιχείων-κορυφών (από τα m), και το αντίστοιχο κόστος

```
F = [m];
S = [p];
cost = inf;
i = 0;
tempSet = [p];
N = factorial(m)/factorial(p) * factorial(m − p);
while i < N do
    tempSet = calculateCombination(i);
    tempCost = calculateCost(S);
    if tempCost < cost then
        S = tempSet;
        cost = tempCost;
    i = i + 1;
end
```

**Αλγόριθμος 1**: Brute force algorithm

## 3.2 Αλγόριθμος myopic (greedy-add algorithm)

Ο μυωπικός αλγόριθμος, ο οποίος είναι ένας άπληστος αλγόριθμος, είναι ο απλούστερος (ευρετικός) αλγόριθμος για το πρόβλημα χωροθέτησης μονάδων. Ο αλγόριθμος λειτουργεί ως εξής: έστω $F$ το σύνολο των πιθανών κόμβων προς χωροθέτηση, $S$ το σύνολο που περιέχει τους κόμβους λύσης και $p$ ο αριθμός των μονάδων προς εγκατάσταση. Αρχικά, το σύνολο $S$ είναι κενό και θεωρούμε άπειρο (ή πολύ μεγάλο) κόστος. Σε κάθε βήμα του αλγορίθμου υπολογίζεται ποιος κόμβος από το σύνολο $F$, ο οποίος δεν έχει προστεθεί ήδη στο σύνολο $S$, ελαττώνει περισσότερο το συνολικό κόστος εάν τοποθετήσουμε σε αυτόν μονάδα. Ο κόμβος αυτός προστίθεται στους κόμβους λύσης (σύνολο $S$). Η παραπάνω διαδικασία επαναλαμβάνεται μέχρις ότου το σύνολο $S$ να έχει $p$ στοιχεία, που σημαίνει ότι θα έχουν βρεθεί $p$ κόμβοι για χωροθέτηση. Το μειονέκτημα του αλγορίθμου είναι ότι από τη στιγμή που ένας κόμβος προστεθεί στους κόμβους λύσης δε μπορεί να αφαιρεθεί (όπως συμβαίνει σε άλλους αλγορίθμους π.χ. Teitz and Bart). Συνεπώς, ο αλγόριθμος είναι βέλτιστος για $p = 1$ ενώ για τιμές του $p$ μεγαλύτερες της μονάδας η λύση του μπορεί να απέχει πολύ από την βέλτιστη. Η πολυπλοκότητα του αλγορίθμου που υλοποιήθηκε είναι $O(p^2n^2)$.

### 3.2.1 Παράδειγμα αλγορίθμου myopic

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Αρχικά ο κόμβος 2 προστίθεται στο σύνολο $S$ επειδή είναι αυτός που δίνει το μικρότερο κόστος το οποίο είναι: $C = C_{1,2} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2} = 1000 + 0 + 100 + 10 + 1 = 1111$. Όλοι οι κόμβοι εξυπηρετούνται από τον 2 σε αυτό το σημείο και $S = \{2\}$.

Για παράδειγμα αν επιλέγαμε τον κόμβο 5 το κόστος θα ήταν: $C = C_{1,5} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5} = 1001 + 1 + 101 + 11 + 0 = 1113$ $(> 1111)$. Με παρόμοιο τρόπο μπορούμε να δούμε ότι και οι υπόλοιποι κόμβοι δίνουν κόστος μεγαλύτερο από 1111.

Στη συνέχεια ο αλγόριθμος προσθέτει τον κόμβο 1 στο σύνολο $S$, επειδή αυτός μειώνει περισσότερο το κόστος με δεδομένο ότι ο κόμβος 2 είναι ήδη μέρος της λύσης, και το νέο κόστος είναι: $C = C_{1,1} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2} = 0 + 0 + 100 + 10 + 1 = 111$. Ο κόμβος 1 εξυπηρετείται από τον 1, και οι υπόλοιποι από τον 2 (ο 2 από τον εαυτό του). $S = \{2, 1\}$.

Άρα για $p = 2$ ο αλγόριθμος επιλέγει τους κόμβους 1 και 2 με συνολικό κόστος 111.

Λύση: $S = \{2, 1\}, C = 111$ (στο συγκεκριμένο παράδειγμα ο αλγόριθμος έτυχε να βρει τη βέλτιστη λύση για το πρόβλημα)

### 3.2.2 Ψευδοκώδικας αλγορίθμου myopic

Στον πίνακα 2 δίνεται ο ψευδοκώδικας του αλγορίθμου myopic/greedy-add.

**Data**: Γράφημα $G = (V, E)$ (m σε πλήθος κορυφές) και $p$ το πλήθος των μονάδων προς χωροθέτηση
**Result**: Σύνολο $S$, $p$ στοιχείων-κορυφών (από τα m), και το αντίστοιχο κόστος

```
F = [m];
S = [p];
for i ← 0 to p − 1 do
    cost = inf;
    for j ← 0 to m − 1 do
        if tempCost(F[j]) < cost then
            cost = tempCost(F[j]);
            S[i] = F[j];
    end
end
calculateCost(S);
```

**Αλγόριθμος 2**: Myopic/Greedy-add algorithm

## 3.3 Αλγόριθμος Teitz and Bart (exchange algorithm)

Ο αλγόριθμος αυτός αξιοποιεί μία εφικτή λύση που έχει παραχθεί από κάποιον άλλο ευρετικό αλγόριθμο και τη βελτιώνει. Έστω $F$ το σύνολο των πιθανών κόμβων προς χωροθέτηση, $S$ το σύνολο που περιέχει έναν πιθανό συνδυασμό κόμβων λύσης και $p$ ο αριθμός των μονάδων προς χωροθέτηση. Ο αλγόριθμος υπολογίζει το κόστος της λύσης εάν αντικαθιστούσαμε τον πρώτο κόμβο του συνόλου $S$ με έναν κόμβο του γραφήματος που δεν είναι μέρος της λύσης που τροφοδότησε τον αλγόριθμο (που δεν ανήκει στο $S$ αλλά ανήκει στο $F$), αυτό επαναλαμβάνεται για όλα τα στοιχεία του συνόλου $F$. Εάν βρεθεί κόστος μικρότερο από αυτό που έδινε η αρχική λύση $S$, τότε αντικαθιστούμε τον πρώτο κόμβο του $S$ με αυτόν που παρέχει το μικρότερο κόστος. Η διαδικασία επαναλαμβάνεται $p$ φορές, δηλαδή μία για κάθε κόμβο της αρχικής λύσης $S$. Ο αλγόριθμος δε βελτιώνει πάντα την αρχική λύση που παίρνει ως είσοδο. Η πολυπλοκότητα του αλγορίθμου που υλοποιήθηκε είναι $O(p^2n^2)$.

### 3.3.1 Παράδειγμα αλγορίθμου Teitz and Bart

Έστω ότι η αρχική λύση που εισάγουμε στον αλγόριθμο είναι οι κόμβοι 4 και 5 με κόστος: $C = C_{1,5} + C_{2,5} + C_{3,5} + C_{4,4} + C_{5,5} = 1001 + 1 + 101 + 0 + 0 = 1103$

Άρα τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{4, 5\}$

Στο πρώτο βήμα του αλγορίθμου εξετάζεται εάν έχουμε μείωση του κόστους με την αντικατάσταση του πρώτου κόμβου του $S$ (δηλαδή του 4) με κάθε έναν από τους υπόλοιπους κόμβους του $F$. Η μεγαλύτερη μείωση παρατηρείται εάν αντικαταστήσουμε τον 4 με τον 1 άρα το $S$ γίνεται $S = \{1, 5\}$ και πλέον το κόστος είναι: $C = C_{1,1} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5} = 0 + 1 + 101 + 11 + 0 = 113$.

Στο δεύτερο βήμα εξετάζουμε τον δεύτερο κόμβο του $S$ δηλαδή τον 5 με παρόμοιο τρόπο. Με αντικατάστασή του από τον κόμβο 2 έχουμε το λιγότερο κόστος $C = C_{1,1} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2} = 0 + 0 + 100 + 10 + 1 = 111$ και ο αλγόριθμος σταματάει επειδή εξετάσαμε όλα τα στοιχεία του $S$. Τελικά, καταλήγουμε σε μια καλύτερη λύση με κόμβους προς χωροθέτηση τους 1 και 2 και κόστος 111.
Λύση: $S = \{1, 2\}, C = 111$ (καταλήξαμε πάλι σε βέλτιστη το οποίο οφείλεται στη φύση του παραδείγματος, πιθανότατα δε θα συνέβαινε σε πραγματικό πρόβλημα μεγάλων διαστάσεων)

### 3.3.2 Ψευδοκώδικας αλγορίθμου Teitz and Bart

Στον πίνακα 3 δίνεται ο ψευδοκώδικας του αλγορίθμου Teitz and Bart.

```
Data: Γράφημα G = (V, E) (m σε πλήθος κορυφές) και p το πλήθος των
      μονάδων προς χωροθέτηση και μία εφικτή λύση έστω S' κόστους C'
Result: Σύνολο S, p στοιχείων-κορυφών (από τα m), και το αντίστοιχο
        κόστος
F = [m];
S = [p];
S = S';
cost = C';
for i ← 0 to p − 1 do
    for j ← 0 to m − 1 do
        swap(S[i], F[j]);
        if calculateCost(S) < cost then
            cost = calculate_cost(S);
            S[i] = F[j];
    end
end
calculateCost(S);
```

**Αλγόριθμος 3**: Teitz and Bart algorithm

## 3.4 Αλγόριθμος neighborhood search

Είναι ένας ακόμη βελτιωτικός αλγόριθμος που απαιτεί ως είσοδο μία εφικτή λύση. Έστω $S$ το σύνολο που περιέχει ένα πιθανό συνδυασμό κόμβων λύσης. Ο αλγόριθμος αυτός δημιουργεί $p$ γειτονιές, όπου $p$ ο απαιτούμενος αριθμός μονάδων προς χωροθέτηση. Μια γειτονιά αποτελείται από έναν κόμβο λύσης $s_i$ ("κεντρικός" κόμβος της γειτονιάς) και όλους τους κόμβους του γραφήματος οι οποίοι είναι πλησιέστεροι στον $s_i$ από ότι σε οποιονδήποτε άλλο κόμβο λύσης. Στη συνέχεια ο αλγόριθμος λύνει το 1-median πρόβλημα (με χρήση του μυωπικού αλγορίθμου) σε κάθε γειτονιά και εάν κάποιος κόμβος της γειτονιάς μειώνει το κόστος τότε αυτός γίνεται ο νέος "κεντρικός" κόμβος της γειτονιάς. Στην επόμενη επανάληψη δημιουργούμε ξανά τις (νέες πλέον) γειτονιές και λύνουμε πάλι το 1-median πρόβλημα ανά γειτονιά. Η διαδικασία επαναλαμβάνεται μέχρι να μην υπάρξει αντικατάσταση του "κεντρικού" κόμβου σε οποιαδήποτε γειτονιά.

### 3.4.1 Παράδειγμα αλγορίθμου neighborhood search

Έστω ότι η αρχική λύση που εισάγουμε στον αλγόριθμο είναι οι κόμβοι $4$ και $5$ με κόστος: $C = C_{1,5} + C_{2,5} + C_{3,5} + C_{4,4} + C_{5,5} = 1001 + 1 + 101 + 0 + 0 = 1103$

Άρα τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{4, 5\}$

Αρχικά δημιουργούνται $2$ γειτονιές, μία με κεντρικό κόμβο τον $4$ και μία με τον κόμβο $5$, έστω $A$ και $B$ αντίστοιχα: $A = \{4\}$ και $B = \{1, 2, 3, 5\}$

Στην γειτονιά $A$ δεν υπάρχει αλλαγή όμως στην $B$ η καλύτερη επιλογή είναι ο κόμβος $2$ (δηλαδή ο κόμβος αυτός είναι η λύση του 1-median προβλήματος για το σύνολο B), άρα: $S = \{2, 4\}$ και κόστος $C = C_{1,2} + C_{2,2} + C_{3,2} + C_{4,4} + C_{5,2} = 1000 + 0 + 100 + 1 + 0 = 1101$ (μικρότερο του 1103).

Δημιουργούνται ξανά οι γειτονιές με κεντρικούς κόμβους τους $4$ και $2$: $A = \{4\}$ και $B = \{1, 2, 3, 5\}$. Αυτή τη φορά δεν υπάρχει αλλαγή η οποία ελαττώνει το κόστος, άρα το $S$ δε μεταβάλλεται και ο αλγόριθμος σταματάει. Δηλαδή τοποθετούμε μονάδες στους κόμβους $2$ και $4$ με κόστος 1103.

Λύση: $S = \{2, 4\}, C = 1101$ (βελτίωση ως προς το αρχικό κόστος 1103)

### 3.4.2 Ψευδοκώδικας αλγορίθμου neighborhood search

Στον πίνακα 4 δίνεται ο ψευδοκώδικας του αλγορίθμου neighborhood search.

```
Data: Γράφημα G = (V, E) (m σε πλήθος κορυφές) και p το πλήθος των
      μονάδων προς χωροθέτηση και μία εφικτή λύση έστω S' κόστους C'
Result: Σύνολο S, p στοιχείων-κορυφών (από τα m), και το αντίστοιχο
        κόστος
F = [m];
S = [p];
S = S';
cost = C';
tempSet = S;
flag = true;
while i < N do
    for i ← 0 to p − 1 do
        createNeighborhood(i);
        solveOneMedian(i);
        if calculateCost(S) < cost then
            // solution(i) is the solution vertex of the 1-median
            problem
            tempSet[i] = solution(i);
    end
    if tempSet == S then
        flag = false;
        S = tempSet;
end
calculateCost(S);
```

**Αλγόριθμος 4**: Neighborhood search algorithm

## 3.5 Αλγόριθμος drop approach

Έστω $F$ το σύνολο των πιθανών κόμβων προς χωροθέτηση, $S$ το σύνολο που περιέχει τους κόμβους λύσης και $p$ ο απαιτούμενος αριθμός μονάδων προς εγκατάσταση. Ο αλγόριθμος αυτός θεωρεί, αρχικά, ότι το $S$ περιέχει όλα τα στοιχεία του $F$ δηλαδή ότι έχουμε εγκαταστήσει μονάδες σε όλους τους υποψήφιους κόμβους άρα το αρχικό κόστος είναι μηδέν. Στη συνέχεια, υπολογίζει πόσο θα αυξανόταν το κόστος για κάθε έναν από τους κόμβους που υπάρχουν στο $S$ εάν καταργούσαμε τη μονάδα σε αυτόν. Ο κόμβος που επιφέρει τη μικρότερη αύξηση του κόστους αφαιρείται από το $S$. Η διαδικασία συνεχίζεται $m-p$ φορές ώσπου, δηλαδή, να μείνουν $p$ κόμβοι λύσης στο $S$. Η πολυπλοκότητα του αλγορίθμου που υλοποιήθηκε είναι $O((m-p)m^2)$.

### 3.5.1 Παράδειγμα αλγορίθμου drop approach

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{1, 2, 3, 4, 5\}$

Αρχικά, το κόστος είναι μηδέν επειδή θεωρούμε χωροθετημένες μονάδες σε κάθε έναν από τους πέντε κόμβους. Στην πρώτη επανάληψη δοκιμάζουμε να αφαιρέσουμε κάθε κόμβο έναν κάθε φορά και υπολογίζουμε το κόστος. Για παράδειγμα, εάν αφαιρέσουμε τον κόμβο 1, τότε το κόστος γίνεται $C = C_{1,2}+C_{2,2}+C_{3,3}+C_{4,4}+C_{5,5} = 1000+0+0+0+0 = 1000$. Ακολουθούν τα κόστη που υπολογίζονται στο πρώτο βήμα του αλγορίθμου αφαιρώντας κάθε φορά τον κόμβο που βρίσκεται στις παρενθέσεις.

$(1) : C_{1,2} + C_{2,2} + C_{3,3} + C_{4,4} + C_{5,5} = 1000 + 0 + 0 + 0 + 0 = 1000$

$(2) : C_{1,1} + C_{2,5} + C_{3,3} + C_{4,4} + C_{5,5} = 0 + 1 + 0 + 0 + 0 = 1$

$(3) : C_{1,1} + C_{2,2} + C_{3,2} + C_{4,4} + C_{5,5} = 0 + 0 + 100 + 0 + 0 = 100$

$(4) : C_{1,1} + C_{2,2} + C_{3,3} + C_{4,2} + C_{5,5} = 0 + 0 + 0 + 10 + 0 = 10$

$(5) : C_{1,1} + C_{2,2} + C_{3,3} + C_{4,4} + C_{5,2} = 0 + 0 + 0 + 0 + 1 = 1$

Η αφαίρεση του κόμβου 2 ή του κόμβου 5 επιφέρει τη μικρότερη αύξηση κόστους. Επιλέγουμε αυθαίρετα τον κόμβο 2, άρα $S = \{1, 3, 4, 5\}$ στο τέλος της πρώτης επανάληψης του αλγορίθμου. Στη δεύτερη επανάληψη κάνουμε το ίδιο με τους κόμβους

που απομένουν στο $S$. Τα κόστη είναι:
$(2,1): C_{1,5} + C_{2,5} + C_{3,3} + C_{4,4} + C_{5,5} = 1001 + 1 + 0 + 0 + 0 = 1002$
$(2,3): C_{1,1} + C_{2,5} + C_{3,5} + C_{4,4} + C_{5,5} = 0 + 1 + 101 + 0 + 0 = 102$
$(2,4): C_{1,1} + C_{2,5} + C_{3,3} + C_{4,5} + C_{5,5} = 0 + 1 + 0 + 11 + 0 = 12$
$(2,5): C_{1,1} + C_{2,4} + C_{3,3} + C_{4,4} + C_{5,4} = 0 + 10 + 0 + 0 + 11 = 21$

Οπότε, αφαιρούμε τον κόμβο 4 από κόμβο λύσης και έχουμε $S = \{1,3,5\}$ με την ολοκλήρωση της δεύτερης επανάληψης. Επαναλαμβάνουμε τη διαδικασία για τρίτη και τελευταία φορά και τα κόστη είναι:
$(2,4,1): C_{1,5} + C_{2,5} + C_{3,3} + C_{4,5} + C_{5,5} = 1001 + 1 + 0 + 11 + 0 = 1012$
$(2,4,3): C_{1,1} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5} = 0 + 1 + 101 + 11 + 0 = 113$
$(2,4,5): C_{1,1} + C_{2,3} + C_{3,3} + C_{4,3} + C_{5,3} = 0 + 100 + 0 + 110 + 101 = 311$

Άρα ο κόμβος 3 δεν ανήκει στη λύση και αφαιρείται από το $S$. Δηλαδή οι κόμβοι στους οποίους θα εγκαταστήσουμε μονάδες είναι οι κόμβοι 1 και 5 και το κόστος είναι $C = C_{1,1} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5} = 0 + 1 + 101 + 11 + 0 = 113$.
Λύση: $S = \{1,5\}, C = 113$

### 3.5.2 Ψευδοκώδικας αλγορίθμου drop approach

Στον πίνακα 5 δίνεται ο ψευδοκώδικας του αλγορίθμου drop approach.

**Data**: Γράφημα $G = (V, E)$ (m σε πλήθος κορυφές) και $p$ το πλήθος των μονάδων προς χωροθέτηση
**Result**: Σύνολο $S$, $p$ στοιχείων-κορυφών (από τα m), και το αντίστοιχο κόστος

```
F = [m];
S = [m];
cost = inf;
for i ← 0 to m − p do
    for j ← 0 to m − i do
        if tempCost(F[j]) < cost then
            cost = tempCost(F[j]);
            index = j;
    end
    remove(S, index);
    cost = tempCost(F[j]);
end
calculateCost(S);
```

**Αλγόριθμος 5**: Drop approach algorithm

## 3.6 Αλγόριθμος random

Ο αλγόριθμος αυτός δίνει μία γρήγορη και τυχαία λύση (συνήθως πολύ χειρότερη της βέλτιστης). Έστω $F$ το σύνολο των πιθανών κόμβων, $S$ το σύνολο που περιέχει τους κόμβους λύσης και $p$ ο αριθμός των μονάδων που θα εγκαταστήσουμε. Με τη χρήση μίας γεννήτριας τυχαίων αριθμών επιλέγουμε $p$ σε πλήθος στοιχεία (κόμβους) από το σύνολο $F$, αυτά αποτελούν το σύνολο $S$, δηλαδή τους κόμβους λύσης μας. Τέλος, υπολογίζουμε το κόστος ($C$) που παράγουν οι παραπάνω κόμβοι. Η πολυπλοκότητα του αλγορίθμου αυτού είναι $O(m + pn)$.

### 3.6.1 Παράδειγμα αλγορίθμου random

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Επιλέγουμε με τυχαίο τρόπο δύο στοιχεία του $F$, έστω τα 1 και 3. Άρα θα τοποθετήσουμε μονάδες στους κόμβους 1 και 3 με κόστος $C = 311$.

Λύση: $S = \{1, 3\}, C = 311$

Στον πίνακα 6 δίνεται ο ψευδοκώδικας του αλγορίθμου random.

**Data**: Γράφημα $G = (V, E)$ (m σε πλήθος κορυφές) και $p$ το πλήθος των μονάδων προς χωροθέτηση
**Result**: Σύνολο $S$, $p$ στοιχείων-κορυφών (από τα m), και το αντίστοιχο κόστος

```
F = [m];
S = [p];
for i ← 0 to p − 1 do
    x = random();
    while exist(x, S) do
        x = random();
    end
    S[i] = x;
end
calculateCost(S);
```

**Αλγόριθμος 6**: Random algorithm

## 3.7 Αλγόριθμος random plus greedy

Όπως υποδηλώνει και το όνομα του είναι συνδυασμός των αλγορίθμων random και myopic. Αρχικά επιλέγονται με τυχαίο τρόπο $\alpha$ (παράμετρος που δίνεται ως είσοδος) σε πλήθος κόμβοι από το σύνολο $F$ των πιθανών προς χωροθέτηση κόμβων και προστίθενται στο σύνολο $S$ (κόμβοι λύσης). Οι υπόλοιποι $(p - \alpha)$ κόμβοι επιλέγονται με τη χρήση του μυωπικού αλγορίθμου δηλαδή σε κάθε βήμα επιλέγουμε τον κόμβο που ελαχιστοποιεί το συνολικό κόστος $(C)$. Εάν $\alpha = 0$ τότε ο αλγόριθμος ταυτίζεται με τον μυωπικό ενώ εάν $\alpha = p$ τότε ταυτίζεται με τον αλγόριθμο τυχαίας επιλογής που παρουσιάστηκε στην προηγούμενη ενότητα του κεφαλαίου (random algorithm). Η πολυπλοκότητα του αλγόριθμου αυτού είναι $O((m + \alpha pn) + (1 - \alpha)(pmn))$.

### 3.7.1 Παράδειγμα αλγορίθμου random plus greedy

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Επιλέγουμε με τυχαίο τρόπο ένα στοιχείο του $F$, έστω τον κόμβο 5. Άρα $S = \{5\}$ και $C = C_{1,5} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5} = 1001 + 1 + 101 + 11 + 0 = 1113$

Στη συνέχεια επιλέγουμε από τους υπόλοιπους κόμβους του $F$ αυτόν που ελαχιστοποιεί το κόστος δηλαδή τον κόμβο 1. Άρα χωροθετούνται μονάδες στους κόμβους

1 και 5 με κόστος 113.

Λύση: $S = \{1, 5\}, C = 113$

### 3.7.2 Ψευδοκώδικας αλγορίθμου random plus greedy

Στον πίνακα 7 δίνεται ο ψευδοκώδικας του αλγορίθμου random plus greedy.

```
Data: Γράφημα G = (V, E) (m σε πλήθος κορυφές) και p το πλήθος των
      μονάδων προς χωροθέτηση
Result: Σύνολο S, p στοιχείων-κορυφών (από τα m), και το αντίστοιχο
        κόστος
F = [m];
S = [p];
for i ← 0 to a − 1 do
    x = random();
    while exist(x, S) do
        x = random();
    end
    S[i] = x;
end
for i ← a to p − a do
    for j ← 0 to m − 1 do
        if tempCost(F[j]) < cost then
            cost = tempCost(F[j]);
            S[i] = F[j];
    end
end
calculateCost(S);
```

**Αλγόριθμος 7**: Random plus greedy algorithm

## 3.8 Αλγόριθμος randomized greedy

Η λειτουργία του αλγορίθμου randomized greedy βρίσκεται μεταξύ αυτής του μυωπικού αλγορίθμου και του αλγορίθμου τυχαίας επιλογής. Έστω $F$ το σύνολο των πιθανών κόμβων προς χωροθέτηση, $S$ το σύνολο που περιέχει τους κόμβους λύσης και $p$ ο αριθμός των μονάδων που θα εγκαταστήσουμε. Σε κάθε επανάληψη ο αλγόριθμος randomized greedy αντί να επιλέξει μεταξύ των $m-i+1$ πιθανών κόμβων, επιλέγει έναν από τους $\lceil \alpha(m-i+1) \rceil$ καλύτερους κόμβους (ως προς τη μείωση του κόστους) τυχαία. Το $\alpha$ είναι μια παράμετρος στο διάστημα $(0,1]$. Παρατηρούμε ότι εάν το $\alpha$ τείνει στο μηδέν τότε ο αλγόριθμος συμπίπτει με τον μυωπικό ενώ εάν το $\alpha$ τείνει στη μονάδα τότε συμπίπτει με αυτόν την τυχαίας επιλογής. Η παραπάνω διαδικασία επαναλαμβάνεται $p$ φορές. Η πολυπλοκότητα του αλγόριθμου αυτού είναι $O(pmn)$.

### 3.8.1 Παράδειγμα αλγορίθμου randomized greedy

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Έστω $\alpha = 0.5$. Στην πρώτη επανάληψη ο αλγόριθμος επιλέγει τυχαία έναν κόμβο μεταξύ των τριών καλύτερων κόμβων (επειδή $\lceil 0.5 * (5–1+1) \rceil = 3$). Δηλαδή μεταξύ των κόμβων 2, 4 και 5, έστω επιλέγεται ο κόμβος 2 και προστίθεται στο σύνολο $S$. Άρα $S = \{2\}$ στο τέλος της πρώτης επανάληψης.

Το κόστος εάν τοποθετήσουμε μονάδα στον κόμβο με αριθμό αυτόν στις παρενθέσεις είναι:

$(1) : C_{1,1} + C_{2,1} + C_{3,1} + C_{4,1} + C_{5,1} = 0 + 1000 + 1100 + 1010 + 1001 = 4111$

$(2) : C_{1,2} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2} = 1000 + 0 + 100 + 10 + 1 = 1111$

$(3) : C_{1,3} + C_{2,3} + C_{3,3} + C_{4,3} + C_{5,3} = 1100 + 100 + 0 + 110 + 101 = 1411$

$(4) : C_{1,4} + C_{2,4} + C_{3,4} + C_{4,4} + C_{5,4} = 1010 + 10 + 110 + 0 + 11 = 1141$

$(5) : C_{1,5} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5} = 1001 + 1 + 101 + 11 + 0 = 1113$

Στη δεύτερη και τελευταία επανάληψη (αφού $p = 2$) επιλέγουμε τυχαία μεταξύ των δύο καλύτερων κόμβων (επειδή $\lceil 0.5*(5–2+1) \rceil = 2$). Δηλαδή μεταξύ των κόμβων

1 και 3, έστω ότι επιλέγεται ο κόμβος 1. Άρα $S = \{1, 2\}$. στο τέλος του αλγορίθμου.

Το κόστος εάν τοποθετήσουμε μονάδα στον κόμβο με αριθμό αυτόν στις παρενθέσεις δεδομένου ότι υπάρχει μονάδα στον κόμβο 2 είναι:

$(1, 2) : C_{1,1} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2} = 0 + 0 + 100 + 10 + 1 = 111$

$(3, 2) : C_{1,2} + C_{2,2} + C_{3,3} + C_{4,2} + C_{5,2} = 1000 + 0 + 0 + 10 + 1 = 1011$

$(4, 2) : C_{1,2} + C_{2,2} + C_{3,2} + C_{4,4} + C_{5,2} = 1000 + 0 + 100 + 0 + 1 = 1101$

$(5, 2) : C_{1,2} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,5} = 1000 + 0 + 100 + 10 + 0 = 1110$

Τελικά, θα χωροθετηθούν μονάδες στους κόμβους 1 και 2 με συνολικό κόστος 111.

Λύση: $S = \{2, 1\}, C = 111$ (τυχαίνει να είναι η βέλτιστη λύση)

### 3.8.2 Ψευδοκώδικας αλγορίθμου randomized greedy

Στον πίνακα 8 δίνεται ο ψευδοκώδικας του αλγορίθμου randomized greedy.

```
Data: Γράφημα G = (V, E) (m σε πλήθος κορυφές) και p το πλήθος των
      μονάδων προς χωροθέτηση
Result: Σύνολο S, p στοιχείων-κορυφών (από τα m), και το αντίστοιχο
        κόστος
F = [m];
S = [p];
for i ← 0 to p − 1 do
    cost = inf;
    size = ⌈a(m − i + 1)⌉;
    tArray = [size];
    for j ← 0 to m − 1 do
        if tempCost(F[j]) < tempCost(tArray[0]) then
            tArray[0] = tempCost(F[j]);
        sort(tArray);
    end
    for j ← 0 to m − 1 do
        if tempCost(F[j]) < cost then
            cost = tempCost(F[j]);
            S[i] = F[j];
    end
end
calculateCost(S);
```

**Αλγόριθμος 8**: Randomized greedy algorithm

## 3.9 Αλγόριθμος proportional greedy

Έστω $F$ το σύνολο των πιθανών κόμβων, $S$ το σύνολο που περιέχει τους κόμβους λύσης και $p$ ο απαιτούμενος αριθμός μονάδων προς χωροθέτηση. Ο αλγόριθμος proportional greedy λειτουργεί με τον παρακάτω τρόπο. Για κάθε πιθανό κόμβο-μονάδα $f_i$ προς εγκατάσταση υπολογίζεται η μείωση του κόστους που θα υπήρχε εάν τοποθετούσαμε μια μονάδα σε αυτόν (έστω $s(f_i)$ αυτή η μείωση). Έπειτα επιλέγεται ένας κόμβος με πιθανότητα ανάλογη της διαφοράς $s(f_i)–s(f_j)min$ όπου $s(f_j)min$ είναι η μικρότερη τιμή μεταξύ των $s(f_i)$. Η παραπάνω διαδικασία επαναλαμβάνεται $p$ φορές. Η πολυπλοκότητα του αλγόριθμου αυτού είναι $O(pmn)$.

### 3.9.1 Παράδειγμα αλγορίθμου proportional greedy

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Θεωρούμε ένα πολύ μεγάλο κόστος ως αρχικό, έστω 10000. Στην πρώτη επανάληψη υπολογίζεται το κόστος που θα αποθηκευόταν για κάθε έναν από τους κόμβους του $F$ εάν τοποθετήσουμε μονάδα σε αυτόν, όπως φαίνεται παρακάτω:

$s(f_1)$ : 10000–$(C_{1,1} + C_{2,1} + C_{3,1} + C_{4,1} + C_{5,1})$ = 10000–4111 = 5889 = $s(f_j)min$

$s(f_2)$ : 10000–$(C_{1,2} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2})$ = 10000–1111 = 8889

$s(f_3)$ : 10000–$(C_{1,3} + C_{2,3} + C_{3,3} + C_{4,3} + C_{5,3})$ = 10000–1411 = 8589

$s(f_4)$ : 10000–$(C_{1,4} + C_{2,4} + C_{3,4} + C_{4,4} + C_{5,4})$ = 10000–1141 = 8859

$s(f_5)$ : 10000–$(C_{1,5} + C_{2,5} + C_{3,5} + C_{4,5} + C_{5,5})$ = 10000–1114 = 8886

$s(f_1)–s(f_j)min$ : 5889–5889 = 0

$s(f_2)–s(f_j)min$ : 8889–5889 = 3000

$s(f_3)–s(f_j)min$ : 8589–5889 = 2700

$s(f_4)–s(f_j)min$ : 8859–5889 = 2970

$s(f_5)–s(f_j)min$ : 8886–5889 = 2997

Επιλέγουμε τυχαία, αλλά ανάλογα με τις τιμές του δεύτερου πίνακα, έναν κόμβο έστω τον κόμβο 2. Άρα $S = \{2\}$ και κόστος 1111 στο τέλος της πρώτης επανάληψης. Σημειώνεται ότι ο κόμβος 1 είχε μηδενική πιθανότητα να επιλεγεί.

Στη δεύτερη επανάληψη το κόστος είναι 1111 επειδή έχουμε ήδη επιλέξει τον κόμβο 2. Υπολογίζουμε με τον ίδιο τρόπο τις τιμές $s(f_i)$ και $s(f_i)$–$s(f_j)min$:

$s(f_1)$ : 1111–$(C_{1,1} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2}) = 1111$–$111 = 1000$

$s(f_3)$ : 1111–$(C_{1,2} + C_{2,2} + C_{3,3} + C_{4,2} + C_{5,2}) = 1111$–$1011 = 100$

$s(f_4)$ : 1111–$(C_{1,2} + C_{2,2} + C_{3,2} + C_{4,4} + C_{5,2}) = 1111$–$1101 = 10$

$s(f_5)$ : 1111–$(C_{1,2} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,5}) = 1111$–$1110 = 1 = s(f_j)min$

$s(f_1)$–$s(f_j)min$ : 1000–1 = 999

$s(f_3)$–$s(f_j)min$ : 100–1 = 99

$s(f_4)$–$s(f_j)min$ : 10–1 = 9

$s(f_5)$–$s(f_j)min$ : 1–1 = 0

Η ελάχιστη τιμή είναι $s(f_5) = 1$. Ανάλογα με τις τιμές του δεύτερου πίνακα επιλέγουμε τυχαία έναν κόμβο, έστω τον κόμβο 1 (ο κόμβος 1 έχει δεκαπλάσια πιθανότητα από τον κόμβο 3 να επιλεγεί και ο 3 με τη σειρά του δεκαπλάσια πιθανότητα από τον 4, ο 5 έχει μηδενική πιθανότητα). Άρα η λύση μας αποτελείται από τους κόμβους 1 και 2 με κόστος 111.
Λύση: $S = \{1, 2\}, C = 111$

### 3.9.2 Ψευδοκώδικας αλγορίθμου proportional greedy

Στον πίνακα 9 δίνεται ο ψευδοκώδικας του αλγορίθμου proportional greedy.

```
Data: Γράφημα G = (V, E) (m σε πλήθος κορυφές) και p το πλήθος των
      μονάδων προς χωροθέτηση
Result: Σύνολο S, p στοιχείων-κορυφών (από τα m), και το αντίστοιχο
        κόστος
F = [m];
S = [p];
d = [m];
p = [m];
for v ← 0 to p − 1 do
    sum = 0;
    min = inf;
    for i ← 0 to m − 1 do
        d[i] = calculateS(F[i]);
        sum = sum + d[i];
        if d[i] < min then
            min = d[i];
    end
    for i ← 0 to m − 1 do
        d[i] = d[i] = min;
    end
    for i ← 0 to m − 1 do
        p[i] = d[i]/sum;
        // now p[i] contains the possibility to pick vertex i
    end
    // pick one out of m proportional to values in p matrix
    S[v] = proportionalRandom(m, p);
end
calculateCost(S);
```

**Αλγόριθμος 9**: Proportional greedy algorithm

## 3.10 Αλγόριθμος proportional worst

Ο αλγόριθμος αυτός επιλέγει, αρχικά, τον πρώτο κόμβο τυχαία με ομοιόμορφη κατανομή. Στη συνέχεια υπολογίζεται για κάθε κόμβο η διαφορά μεταξύ της τρέχουσας και της βέλτιστης ανάθεσης των κόμβων (έστω $d$ η διαφορά αυτή) και επιλέγεται ως κόμβος λύσης ένας από τους κόμβους του $F$ τυχαία με πιθανότητα ανάλογη της διαφοράς $d$ που έχει υπολογιστεί. Με τον τρόπο αυτό οι κόμβοι οι οποίοι συνεισφέρουν υψηλό κόστος, με βάση την τρέχουσα λύση, έχουν υψηλότερη πιθανότητα να επιλεγούν ως κόμβοι λύσεις. Η πολυπλοκότητα του αλγόριθμου αυτού είναι $O(mn)$.

### 3.10.1 Παράδειγμα αλγορίθμου proportional worst

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Αρχικά, διαλέγουμε τον πρώτο κόμβο τυχαία, έστω τον κόμβο 5 (άρα $C = 1114$). Στη συνέχεια υπολογίζουμε τη διαφορά του τρέχοντος κόστους από το βέλτιστο κόστος για κάθε κόμβο:

$d_1 = 1001$

$d_2 = 1$

$d_3 = 101$

$d_4 = 11$

Επιλέγουμε τυχαία έναν κόμβο με πιθανότητα ανάλογη των παραπάνω τιμών. Έστω ότι επιλέγεται ο κόμβος 3. Άρα, τελικά, θα έχουμε μονάδες στους κόμβους 3 και 5 και κόστος 1013.

Λύση: $S = \{3, 5\}, C = 1013$

Όπως φαίνεται παρακάτω ο κόμβος 1 έχει περίπου δεκαπλάσια πιθανότητα να επιλεχθεί σε σύγκριση με τον κόμβο 3 (0.89 (ή 89%) έναντι 0.09 (ή 9%)). Ακολουθούν οι πιθανότητες επιλογής κάθε κόμβου (με ακρίβεια τεσσάρων δεκαδικών).

$P_1 = 1001/1114 = 0.8986$

$P_2 = 1/1114 = 0.0009$

$P_3 = 101/1114 = 0.0907$

$P_4 = 11/1114 = 0.0099$

### 3.10.2 Ψευδοκώδικας αλγορίθμου proportional worst

Στον πίνακα 10 δίνεται ο ψευδοκώδικας του αλγορίθμου proportional worst.

```
Data: Γράφημα G = (V, E) (m σε πλήθος κορυφές) και p το πλήθος των
      μονάδων προς χωροθέτηση
Result: Σύνολο S, p στοιχείων-κορυφών (από τα m), και το αντίστοιχο
        κόστος
F = [m];
S = [p];
d = [m];
p = [m];
S[0] = random();
for v ← 0 to p − 1 do
    sum = 0;
    for i ← 0 to m − 1 do
        d[i] = calculateCost(F[i]) − calculateOptimalCost(F[i]);
        sum = sum + d[i];
    end
    for i ← 0 to m − 1 do
        p[i] = d[i]/sum;
        // now p[i] contains the possibility to pick vertex i
    end
    // pick one out of m proportional to values in p matrix
    S[v] = proportionalRandom(m, p);
end
calculateCost(S);
```

**Αλγόριθμος 10**: Proportional worst algorithm

## 3.11 Αλγόριθμος sample greedy

Ο αλγόριθμος sample greedy είναι παρόμοιος με τον μυωπικό. Έστω $F$ το σύνολο των πιθανών κόμβων προς χωροθέτηση, $S$ το σύνολο που περιέχει τους κόμβους λύσης και $p$ ο απαιτούμενος αριθμός μονάδων προς εγκατάσταση. Σε κάθε επανάληψη θεωρούμε $q < m$ κόμβους ως πιθανές θέσεις εγκατάστασης οι οποίοι ανήκουν στο $F$ και επιλέγονται ομοιόμορφα τυχαία. Από τους $q$ κόμβους επιλέγεται αυτός που επιφέρει τη μεγαλύτερη μείωση του κόστους και προστίθεται στο $S$ ως κόμβος λύσης. Η παραπάνω διαδικασία επαναλαμβάνεται $p$ φορές. Επιδιώκουμε το $q$ να είναι αρκετά μικρό ώστε ο χρόνος εκτέλεσης να είναι καλύτερος σε σύγκριση με τον μυωπικό αλγόριθμο. Η πολυπλοκότητα του αλγόριθμου αυτού είναι $O(m + pqn)$.

### 3.11.1 Παράδειγμα αλγορίθμου sample greedy

Τα σύνολα $F$ και $S$ είναι:

$\rightarrow F = \{1, 2, 3, 4, 5\}$

$\rightarrow S = \{\}$ (αρχικά κενό)

Δεδομένου ότι $q = 3$ επιλέγουμε τυχαία τρεις κόμβους, έστω τους 1, 2 και 3. Επιλέγουμε από αυτούς τους τρεις κόμβους τον κόμβο που έχει το μικρότερο κόστος (όπως ακριβώς θα κάναμε και στον μυωπικό αλγόριθμο). Άρα επιλέγεται ο κόμβος 2 και $S = \{2\}$ επειδή το κόστος αυτού ($C_{1,2} + C_{2,2} + C_{3,2} + C_{4,2} + C_{5,2} = 1111$) είναι το μικρότερο. Για παράδειγμα, εάν επιλέγαμε τον κόμβο 1 το κόστος θα ήταν 4111 ($C_{1,1} + C_{2,1} + C_{3,1} + C_{4,1} + C_{5,1} = 0 + 1000 + 1100 + 1010 + 1001$).

Στο επόμενο βήμα επιλέγουμε ξανά τρείς τυχαίους κόμβους, έστω τους 3, 4 και 5. Από αυτούς τους τρεις επιλέγεται ο πιο κερδοφόρος δηλαδή ο 3. Συνεπώς, $S = \{2, 3\}$. Οπότε τοποθετούνται μονάδες στους κόμβους 2 και 3 και το κόστος είναι 1011 ($C_{1,2} + C_{2,2} + C_{3,3} + C_{4,2} + C_{5,2} = 1000 + 0 + 0 + 10 + 1$).
Λύση: $S = \{2, 3\}, C = 1011$

### 3.11.2 Ψευδοκώδικας αλγορίθμου sample greedy

Στον πίνακα 11 δίνεται ο ψευδοκώδικας του αλγορίθμου sample greedy.

**Data**: Γράφημα $G = (V, E)$ (m σε πλήθος κορυφές) και $p$ το πλήθος των μονάδων προς χωροθέτηση
**Result**: Σύνολο $S$, $p$ στοιχείων-κορυφών (από τα m), και το αντίστοιχο κόστος

```
F = [m];
S = [p];
A = [q];
cost = inf;
for v ← 0 to p − 1 do
    for i ← 0 to q − 1 do
        A[i] = F[random() mod m];
    end
    for i ← 0 to q − 1 do
        if tempCost(A[i]) < cost then
            cost = tempCost(A[i]);
            S[v] = A[i];
    end
end
calculateCost(S);
```

**Αλγόριθμος 11**: Sample greedy algorithm

# Κεφάλαιο 4

# Υπολογιστικά Αποτελέσματα

## 4.1 Περιγραφή του πειράματος

Για την πειραματική διαδικασία χρησιμοποιήθηκε η βιβλιοθήκη ORLIB (OR-Library) η οποία παρουσιάστηκε από τον Beasley [?] και αποτελεί μια συλλογή από δεδομένα για διάφορα προβλήματα της επιχειρησιακής έρευνας. Από το σύνολο των αρχείων της βιβλιοθήκης χρησιμοποιήθηκαν τα 40 αρχεία που αφορούν το πρόβλημα του $p$ μέσου μη περιορισμένης δυναμικότητας των μονάδων με ονομασία pmed1, pmed2 έως pmed40. Κάθε αρχείο αναπαριστά ένα γράφημα. Η μορφή κάθε αρχείου είναι η εξής: η πρώτη γραμμή περιέχει τον αριθμό των κόμβων (πελατών), το σύνολο των ακμών και τον απαιτούμενο αριθμό μονάδων $p$ προς χωροθέτηση και, στις υπόλοιπες γραμμές, τους τερματικούς κόμβους κάθε ακμής καθώς και το κόστος της ακμής. Το πλήθος των κόμβων ποικίλλει από 100 μέχρι 900, ενώ ο αριθμός $p$ κυμαίνεται μεταξύ 5 και 200. Επίσης, στη βιβλιοθήκη υπάρχει το αρχείο pmedopt το οποίο περιέχει 40 τιμές, τις βέλτιστες λύσεις για κάθε ένα από τα 40 προβλήματα. Στον πίνακα 4.1 απεικονίζεται η βέλτιστη λύση καθώς και οι αριθμοί κόμβων και το $p$ για κάθε ένα από τα αρχεία (pmed1, ..., pmed40). Η ποικιλία των παραμέτρων κάθε προβλήματος, ο τρόπος υλοποίησης των αλγορίθμων καθώς και οι δυνατότητες κάθε μηχανήματος καθιστούν αυτονόητο ότι οι λύσεις και οι χρόνοι εκτέλεσης του πειράματος μπορούν να διαφέρουν πολύ από μηχάνημα σε μηχάνημα. Ωστόσο, η στατιστική μέθοδος που χρησιμοποιείται παρέχει μια καλή εικόνα για την αξιολόγηση και σύγκριση των αλγορίθμων. Ο αλγόριθμος Brute force, ο οποίος παρουσιάστηκε στο προηγούμενο κεφάλαιο, απαιτεί υπερβολικό χρόνο εκτέλεσης και συνεπώς εξαιρέθηκε από τη διαδικασία σύγκρισης των αλγορίθμων (από το δεύτερο κιόλας πρόβλημα ο χρόνος επίλυσης του προβλήματος ξεπερνούσε τη μία ώρα). Στην παρούσα διπλωματική οι αλγόριθμοι υλοποιήθηκαν στη γλώσσα προγραμματισμού C στο περιβάλλον Dev-C++ (Version 5.11). Ο υπολογιστής που χρησιμοποιήθηκε ήταν εξοπλισμένος με επεξεργαστή Intel(R) Core(T) i7-8700 CPU @ 3.20GHz 12 πυρήνων, μνήμη RAM 4.00GB και λειτουργικό σύστημα Windows 10 Pro 64-bit.

## 4.2 Αποτελέσματα των αλγορίθμων

Ο πίνακας 4.1 παρουσιάζει:

- τα ονόματα των προβλημάτων όπως δίνονται από τη βιβλιοθήκη ORLIB
- το πλήθος των κόμβων του προβλήματος
- το πλήθος των μονάδων προς χωροθέτηση
- τη βέλτιστη λύση του κάθε προβλήματος

Στους πίνακες 4.2 έως 4.9 παρουσιάζονται τα αποτελέσματα των κατασκευαστικών αλγορίθμων που υλοποιήθηκαν, οι οποίοι δε χρειάζονται ως είσοδο μια αρχική λύση. Πιο συγκεκριμένα οι πίνακες αυτοί περιλαμβάνουν:

- το όνομα του προβλήματος
- τη λύση που έδωσε ο αλγόριθμος
- την απόκλιση (gap) της λύσης που βρήκε ο αλγόριθμος σε ποσοστό %
- τον χρόνο εκτέλεσης σε δευτερόλεπτα
- τον γεωμετρικό μέσο των αποκλίσεων και των χρόνων εκτέλεσης στην τελευταία γραμμή κάθε πίνακα

Η απόκλιση υπολογίζεται ως:

$$Gap = \frac{solution - OptimalSolution}{OptimalSolution} * 100(\%)$$

Ο γεωμετρικός μέσος υπολογίζεται ως η ν-οστή ρίζα του γινομένου ν αριθμών και χρησιμοποιήθηκε για τη σύγκριση των αλγορίθμων.

Στους πίνακες 4.10, 4.11 και 4.12 παρουσιάζονται οι αποκλίσεις από τις βέλτιστες λύσεις και οι αποκλίσεις από τις αρχικές λύσεις των κατασκευαστικών αλγορίθμων (Myopic, Random, κ.ο.κ) καθώς και οι χρόνοι εκτέλεσης, αντίστοιχα, του αλγορίθμου Teitz and Bart. Ομοίως, έχουμε τους πίνακες 4.13, 4.14 και 4.15 για τον βελτιωτικό αλγόριθμο Neighborhood search.

Στα γραφήματα 4.1 και 4.2 παρουσιάζονται συγκεντρωτικά οι αποκλίσεις από τη βέλτιστη λύση και οι χρόνοι εκτέλεσης για όλους τους αλγορίθμους[1]. Στους πίνακες 4.16 και 4.17 φαίνονται αναλυτικά οι αποκλίσεις και χρόνοι εκτέλεσης των αλγορίθμων, αντίστοιχα.

Σχήμα 4.1: Αποκλίσεις από βέλτιστη λύση


## 4.3 Σύγκριση αλγορίθμων

Μεταξύ των κατασκευαστικών αλγορίθμων παρατηρούμε από τους γεωμετρικούς μέσους των πινάκων 4.2 έως 4.9 ότι ο αλγόριθμος Myopic, έχοντας τον μικρότερο γεωμετρικό μέσο απόκλισης, βρίσκει λύση πιο κοντά στη βέλτιστη σε σύγκριση

[1]Στα γραφήματα για τους αλγορίθμους Teitz and Bart και Neighborhood search επιλέχθηκε η περίπτωση που δέχονται λύση από τον αλγόριθμο myopic επειδή είχε τη μικρότερη απόκλιση

Σχήμα 4.2: Χρόνοι εκτέλεσης


με τους υπόλοιπους αλγορίθμους. Ταυτόχρονα παρατηρούμε ότι ο ταχύτερος αλγόριθμος είναι ο Random, καθώς έχει το μικρότερο γεωμετρικό μέσο χρόνο εκτέλεσης, με αισθητά μεγαλύτερη απόκλιση από τον Myopic. Από τους παραπάνω πίνακες φαίνεται αναλυτικότερα:

- Ο αλγόριθμος Myopic έχει γεωμετρικό μέσο απόκλισης 0.00556 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.07232. Είναι πρώτος σε απόδοση ως προς τη λύση που βρίσκει και έκτος ως προς τον χρόνο εκτέλεσης. Βρήκε τη βέλτιστη λύση σε 2 από τα 40 προβλήματα (στα pmed21 και pmed31), ενώ βρήκε καλύτερη λύση σε όλα τα προβλήματα σε σύγκριση με τους υπόλοιπους κατασκευαστικούς αλγορίθμους. Ο αλγόριθμος Myopic είχε μέση απόκλιση 1.51% η οποία έπεσε στο 0.33% με τον αλγόριθμο Teitz and Bart και στο 0.97% με τον αλγόριθμο Neighborhood search.

- Ο αλγόριθμος Random έχει γεωμετρικό μέσο απόκλισης 0.44208 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.00031. Είναι πέμπτος σε απόδοση ως προς τη λύση που βρίσκει και πρώτος ως προς τον χρόνο εκτέλεσης. Δε βρήκε τη βέλτιστη λύση σε κανένα από τα προβλήματα. Η μέση απόκλιση του αλγορίθμου ήταν 46.17% που βελτιώθηκε σε 0.89% με τη χρήση του Teitz and Bart και σε 23.80% με χρήση του αλγορίθμου Neighborhood search.

- Ο αλγόριθμος Random-plus-greedy έχει γεωμετρικό μέσο απόκλισης 0.11386 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.04416. Είναι δεύτερος σε απόδοση ως προς τη λύση που βρίσκει και τέταρτος ως προς τον χρόνο εκτέλεσης. Δε βρήκε τη βέλτιστη λύση σε κανένα από τα προβλήματα. Ο αλγόριθμος Teitz and Bart έριξε την απόκλιση σε 0.52% από την αρχική 15.86%, ενώ ο Neighborhood search την έριξε στο 10.78%.

- Ο αλγόριθμος Randomized-greedy έχει γεωμετρικό μέσο απόκλισης 0.12690 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.11772. Είναι τρίτος σε απόδοση ως προς τη λύση που βρίσκει και όγδοος ως προς τον χρόνο εκτέλεσης. Δε βρήκε τη βέλτιστη λύση σε κανένα από τα προβλήματα. Η μέση απόκλιση του Randomized-greedy ήταν 15.17%. Οι αλγόριθμοι Teitz and Bart και Neighborhood search βελτίωσαν το ποσοστό σε 0.74% και 7.97%, αντίστοιχα.

- Ο αλγόριθμος Proportional-greedy έχει γεωμετρικό μέσο απόκλισης 0.76209 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.07952. Είναι όγδοος σε απόδοση ως προς τη λύση που βρίσκει και έβδομος ως προς τον χρόνο εκτέλεσης. Δε βρήκε τη βέλτιστη λύση σε κανένα από τα προβλήματα. Η μέση απόκλιση του αλγορίθμου ήταν 83.07% και έπεσε σε 0.95% από τον αλγόριθμο Teitz and Bart και σε 19.28% από τον αλγόριθμο Neighborhood search.

- Ο αλγόριθμος Proportional-worst έχει γεωμετρικό μέσο απόκλισης 0.57531 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.00113. Είναι έκτος σε απόδοση ως προς τη λύση που βρίσκει και τρίτος ως προς τον χρόνο εκτέλεσης. Δε βρήκε τη βέλτιστη λύση σε κανένα από τα προβλήματα. Η τροφοδότηση του αλγορίθμου Teitz and Bart απέδωσε μέση απόκλιση 1.17% από την αρχική 59.09% του Proportional-worst. Το αντίστοιχο ποσοστό του Neighborhood search ήταν 22.70%.

- Ο αλγόριθμος Sample-greedy έχει γεωμετρικό μέσο απόκλισης 0.24826 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.0006. Είναι τέταρτος σε απόδοση ως προς τη λύση που βρίσκει και δεύτερος ως προς τον χρόνο εκτέλεσης. Δε βρήκε τη βέλτιστη λύση σε κανένα από τα προβλήματα. Ο αλγόριθμος Sample-greedy είχε 27.73% μέση απόκλιση στα προβλήματα που έλυσε. Το ποσοστό αυτό έπεσε σε 1.07% και 15.42% με τη χρήση των Teitz and Bart και Neighborhood search, αντίστοιχα.

- Ο αλγόριθμος Drop-approach έχει γεωμετρικό μέσο απόκλισης 0.76042 και γεωμετρικό μέσο χρόνου εκτέλεσης 0.0442. Είναι έβδομος σε απόδοση ως προς τη λύση που βρίσκει και πέμπτος ως προς τον χρόνο εκτέλεσης. Δε βρήκε τη βέλτιστη λύση σε κανένα από τα προβλήματα. Η μέση απόκλιση έφτασε στο 78.22%. Ο αλγόριθμος Teitz and Bart τη βελτίωσε σε 1.35%, ενώ ο αλγόριθμος Neighborhood search σε 25.41%.

Όσον αφορά τους βελτιωτικούς αλγορίθμους που εξετάστηκαν (Teitz and Bart και Neighborhood search), αυτούς δηλαδή που δέχονται μία εφικτή λύση ως είσοδο και τη βελτιώνουν (όταν αυτό είναι δυνατό), από τους πίνακες 4.10, 4.11, 4.12 και 4.13, 4.14, 4.15 παρατηρείται ότι ο αλγόριθμος Teitz and Bart έχει μικρότερο γεωμετρικό μέσο απόκλισης και μεγαλύτερο γεωμετρικό μέσο χρόνου εκτέλεσης από τον αλγόριθμο Neighborhood search σε κάθε περίπτωση συνδυασμού με κατασκευαστικό αλγόριθμο. Αυτό σημαίνει πως ο πρώτος είναι πιο αργός όμως δίνει λύση εγγύτερα στη βέλτιστη. Παρακάτω φαίνεται (τα στοιχεία εξάγονται από τους πίνακες 4.10 και 4.13) σε πόσα προβλήματα βρήκε τη βέλτιστη λύση ο κάθε ένας από τους δύο βελτιωτικούς αλγορίθμους χρησιμοποιώντας λύση που έλαβε ως είσοδο από τους αντίστοιχους κατασκευαστικούς αλγορίθμους.

Teitz and Bart

- Myopic, Random plus greedy: 10/40
- Random: 5/40
- Proportional greedy: 3/40
- Randomized greedy, Proportional worst, Sample greedy, Drop approach: 2/40

Neighborhood search

- Myopic: 2/40
- Random, Random plus greedy, Randomized greedy, Proportional greedy, Proportional worst, Sample greedy, Drop approach: 0/40

Ο αλγόριθμος Teitz and Bart έχει τη μικρότερη απόκλιση όταν χρησιμοποιεί ως είσοδο τη λύση του αλγορίθμου Myopic. Το ίδιο συμβαίνει και στη περίπτωση του αλγορίθμου Neighborhood search. Αυτό δείχνει ότι η λύση του μυωπικού δεν επιδέχεται μεγάλη βελτίωση επειδή είναι ήδη πολύ κοντά στη βέλτιστη. Από την άλλη πλευρά, η απόκλιση των δύο αυτών βελτιωτικών αλγορίθμων είναι η μεγαλύτερη (και για τους δύο) στην περίπτωση του αλγορίθμου Drop approach, το οποίο σημαίνει ότι ο τελευταίος βρίσκει λύση ”μακριά” από τη βέλτιστη. Ακολουθεί η ταξινόμηση των συνδυασμών για κάθε ζέυγος βελτιωτικού-κατασκευαστικού αλγορίθμου σύμφωνα με την απόκλιση της λύσης του συνδυασμού τους από τη λύση που έδωσε αρχικά ο κατασκευαστικός (σύμφωνα με τους πίνακες 4.11 και 4.14). Ο κατασκευαστικός αλγόριθμος με τη μικρότερη απόκλιση είναι πρώτος και ακολουθεί αύξουσα ταξινόμηση.

- Teitz and Bart
    1. Myopic (γεωμετρικός μέσος: 0.00364)
    2. Random plus greedy (γεωμετρικός μέσος: 0.09431)
    3. Randomized greedy (γεωμετρικός μέσος: 0.12118)
    4. Sample greedy (γεωμετρικός μέσος: 0.18744)
    5. Random (γεωμετρικός μέσος: 0.29806)
    6. Proportional worst (γεωμετρικός μέσος: 0.35541)
    7. Proportional greedy (γεωμετρικός μέσος: 0.41678)
    8. Drop approach (γεωμετρικός μέσος: 0.42065)
- Neighborhood search
    1. Myopic (γεωμετρικός μέσος: 0.00013)
    2. Random plus greedy (γεωμετρικός μέσος: 0.01842)
    3. Randomized greedy (γεωμετρικός μέσος: 0.05844)
    4. Sample greedy (γεωμετρικός μέσος: 0.08135)
    5. Random (γεωμετρικός μέσος: 0.14262)
    6. Proportional worst (γεωμετρικός μέσος: 0.21483)
    7. Proportional greedy (γεωμετρικός μέσος: 0.27083)
    8. Drop approach (γεωμετρικός μέσος: 0.27434)

Πίνακας 4.1: Δεδομένα εισόδου αλγορίθμων

| όνομα | κόμβοι | p | λύση |
|---|---|---|---|
| pmed1 | 100 | 5 | 5819 |
| pmed2 | 100 | 10 | 4093 |
| pmed3 | 100 | 10 | 4250 |
| pmed4 | 100 | 20 | 3034 |
| pmed5 | 100 | 33 | 1355 |
| pmed6 | 200 | 5 | 7824 |
| pmed7 | 200 | 10 | 5631 |
| pmed8 | 200 | 20 | 4445 |
| pmed9 | 200 | 40 | 2734 |
| pmed10 | 200 | 67 | 1255 |
| pmed11 | 300 | 5 | 7696 |
| pmed12 | 300 | 10 | 6634 |
| pmed13 | 300 | 30 | 4374 |
| pmed14 | 300 | 60 | 2968 |
| pmed15 | 300 | 100 | 1729 |
| pmed16 | 400 | 5 | 8162 |
| pmed17 | 400 | 10 | 6999 |
| pmed18 | 400 | 40 | 4809 |
| pmed19 | 400 | 80 | 2845 |
| pmed20 | 400 | 133 | 1789 |
| pmed21 | 500 | 5 | 9138 |
| pmed22 | 500 | 10 | 8579 |
| pmed23 | 500 | 50 | 4619 |
| pmed24 | 500 | 100 | 2961 |
| pmed25 | 500 | 167 | 1828 |
| pmed26 | 600 | 5 | 9917 |
| pmed27 | 600 | 10 | 8307 |
| pmed28 | 600 | 60 | 4498 |
| pmed29 | 600 | 120 | 3033 |
| pmed30 | 600 | 200 | 1989 |
| pmed31 | 700 | 5 | 10086 |
| pmed32 | 700 | 10 | 9297 |
| pmed33 | 700 | 70 | 4700 |
| pmed34 | 700 | 140 | 3013 |
| pmed35 | 800 | 5 | 10400 |
| pmed36 | 800 | 10 | 9934 |
| pmed37 | 800 | 80 | 5057 |
| pmed38 | 900 | 5 | 11060 |
| pmed39 | 900 | 10 | 9423 |
| pmed40 | 900 | 90 | 5128 |

Πίνακας 4.2: Αποτελέσματα αλγορίθμου myopic

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5891 | 1.24% | 0.0001 |
| pmed2 | 4118 | 0.61% | 0.0001 |
| pmed3 | 4399 | 3.51% | 0.0001 |
| pmed4 | 3088 | 1.78% | 0.0001 |
| pmed5 | 1378 | 1.70% | 0.0131 |
| pmed6 | 8027 | 2.59% | 0.0001 |
| pmed7 | 5646 | 0.27% | 0.0001 |
| pmed8 | 4472 | 0.61% | 0.0271 |
| pmed9 | 2841 | 3.91% | 0.0941 |
| pmed10 | 1295 | 3.19% | 0.2191 |
| pmed11 | 7721 | 0.32% | 0.0161 |
| pmed12 | 6651 | 0.26% | 0.0151 |
| pmed13 | 4467 | 2.13% | 0.1191 |
| pmed14 | 3013 | 1.52% | 0.4101 |
| pmed15 | 1761 | 1.85% | 1.0301 |
| pmed16 | 8232 | 0.86% | 0.0001 |
| pmed17 | 7019 | 0.29% | 0.0271 |
| pmed18 | 4873 | 1.33% | 0.3441 |
| pmed19 | 2899 | 1.90% | 1.2651 |
| pmed20 | 1866 | 4.30% | 3.2291 |
| pmed21 | 9138 | 0.00% | 0.0141 |
| pmed22 | 8670 | 1.06% | 0.0461 |
| pmed23 | 4694 | 1.62% | 0.8271 |
| pmed24 | 3009 | 1.62% | 2.9791 |
| pmed25 | 1896 | 3.72% | 7.6101 |
| pmed26 | 10093 | 1.77% | 0.0311 |
| pmed27 | 8364 | 0.69% | 0.0621 |
| pmed28 | 4579 | 1.80% | 1.6871 |
| pmed29 | 3104 | 2.34% | 6.1251 |
| pmed30 | 2037 | 2.41% | 16.0401 |
| pmed31 | 10086 | 0.00% | 0.0301 |
| pmed32 | 9331 | 0.37% | 0.0781 |
| pmed33 | 4798 | 2.09% | 3.1091 |
| pmed34 | 3097 | 2.79% | 11.1771 |
| pmed35 | 10406 | 0.06% | 0.0461 |
| pmed36 | 9954 | 0.20% | 0.1261 |
| pmed37 | 5118 | 1.21% | 5.2661 |
| pmed38 | 11153 | 0.84% | 0.0541 |
| pmed39 | 9451 | 0.30% | 0.1571 |
| pmed40 | 5190 | 1.21% | 8.3161 |
| **geometric mean** | - | **0.00556** | **0.07232** |

Πίνακας 4.3: Αποτελέσματα αλγορίθμου random

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 7718 | 32.63% | 0.0001 |
| pmed2 | 6686 | 63.35% | 0.0001 |
| pmed3 | 5648 | 32.89% | 0.0001 |
| pmed4 | 4670 | 53.92% | 0.0001 |
| pmed5 | 2503 | 84.72% | 0.0001 |
| pmed6 | 10556 | 34.92% | 0.0001 |
| pmed7 | 8833 | 56.86% | 0.0001 |
| pmed8 | 6656 | 49.74% | 0.0011 |
| pmed9 | 3989 | 45.90% | 0.0001 |
| pmed10 | 2155 | 71.71% | 0.0001 |
| pmed11 | 10342 | 34.38% | 0.0001 |
| pmed12 | 8936 | 34.70% | 0.0001 |
| pmed13 | 6439 | 47.21% | 0.0001 |
| pmed14 | 4543 | 53.07% | 0.0001 |
| pmed15 | 2965 | 71.49% | 0.0001 |
| pmed16 | 11315 | 38.63% | 0.0001 |
| pmed17 | 9228 | 31.85% | 0.0001 |
| pmed18 | 6850 | 42.44% | 0.0001 |
| pmed19 | 4506 | 58.38% | 0.0001 |
| pmed20 | 3210 | 79.43% | 0.0091 |
| pmed21 | 12070 | 32.09% | 0.0001 |
| pmed22 | 11497 | 34.01% | 0.0001 |
| pmed23 | 6329 | 37.02% | 0.0001 |
| pmed24 | 4526 | 52.85% | 0.0001 |
| pmed25 | 3007 | 64.50% | 0.0311 |
| pmed26 | 13650 | 37.64% | 0.0001 |
| pmed27 | 10493 | 26.32% | 0.0001 |
| pmed28 | 6142 | 36.55% | 0.0191 |
| pmed29 | 4587 | 51.24% | 0.0161 |
| pmed30 | 3270 | 64.40% | 0.0621 |
| pmed31 | 14251 | 41.29% | 0.0001 |
| pmed32 | 13003 | 39.86% | 0.0001 |
| pmed33 | 6814 | 44.98% | 0.0151 |
| pmed34 | 4380 | 45.37% | 0.0311 |
| pmed35 | 14631 | 40.68% | 0.0001 |
| pmed36 | 12766 | 28.51% | 0.0001 |
| pmed37 | 7056 | 39.53% | 0.0001 |
| pmed38 | 15244 | 37.83% | 0.0001 |
| pmed39 | 12230 | 29.79% | 0.0001 |
| pmed40 | 7398 | 44.27% | 0.0201 |
| **geometric mean** | - | **0.44208** | **0.00031** |

Πίνακας 4.4: Αποτελέσματα αλγορίθμου random plus greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6544 | 12.46% | 0.0001 |
| pmed2 | 5484 | 33.98% | 0.0001 |
| pmed3 | 5204 | 22.45% | 0.0151 |
| pmed4 | 3133 | 3.26% | 0.0201 |
| pmed5 | 1420 | 4.80% | 0.0161 |
| pmed6 | 8416 | 7.57% | 0.0001 |
| pmed7 | 7440 | 32.13% | 0.0001 |
| pmed8 | 4847 | 9.04% | 0.0011 |
| pmed9 | 3140 | 14.85% | 0.0471 |
| pmed10 | 1393 | 11.00% | 0.1931 |
| pmed11 | 8058 | 4.70% | 0.0001 |
| pmed12 | 7143 | 7.67% | 0.0001 |
| pmed13 | 5554 | 26.98% | 0.0401 |
| pmed14 | 3215 | 8.32% | 0.3881 |
| pmed15 | 1952 | 12.90% | 0.8141 |
| pmed16 | 9057 | 10.97% | 0.0151 |
| pmed17 | 7107 | 1.54% | 0.0321 |
| pmed18 | 5913 | 22.96% | 0.1401 |
| pmed19 | 3923 | 37.89% | 0.2551 |
| pmed20 | 3114 | 74.06% | 0.1391 |
| pmed21 | 11098 | 21.45% | 0.0001 |
| pmed22 | 8836 | 3.00% | 0.0471 |
| pmed23 | 5854 | 26.74% | 0.1731 |
| pmed24 | 3920 | 32.39% | 0.6721 |
| pmed25 | 2347 | 28.39% | 3.2571 |
| pmed26 | 11383 | 14.78% | 0.0271 |
| pmed27 | 8995 | 8.28% | 0.0321 |
| pmed28 | 5186 | 15.30% | 0.9371 |
| pmed29 | 3689 | 21.63% | 3.3121 |
| pmed30 | 2069 | 4.02% | 16.2881 |
| pmed31 | 11617 | 15.18% | 0.0201 |
| pmed32 | 9629 | 3.57% | 0.0931 |
| pmed33 | 5948 | 26.55% | 1.0541 |
| pmed34 | 3404 | 12.98% | 9.4671 |
| pmed35 | 11413 | 9.74% | 0.0161 |
| pmed36 | 10329 | 3.98% | 0.1231 |
| pmed37 | 5275 | 4.31% | 5.2371 |
| pmed38 | 11549 | 4.42% | 0.0371 |
| pmed39 | 10337 | 9.70% | 0.1051 |
| pmed40 | 5568 | 8.58% | 7.1631 |
| **geometric mean** | - | **0.11386** | **0.04416** |

Πίνακας 4.5: Αποτελέσματα αλγορίθμου randomized greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6181 | 6.22% | 0.0001 |
| pmed2 | 4590 | 12.14% | 0.0001 |
| pmed3 | 4918 | 15.72% | 0.0001 |
| pmed4 | 3374 | 11.21% | 0.0001 |
| pmed5 | 1700 | 25.46% | 0.0171 |
| pmed6 | 8545 | 9.22% | 0.0161 |
| pmed7 | 6465 | 14.81% | 0.0171 |
| pmed8 | 5222 | 17.48% | 0.0181 |
| pmed9 | 3228 | 18.07% | 0.0901 |
| pmed10 | 1531 | 21.99% | 0.2501 |
| pmed11 | 8316 | 8.06% | 0.0001 |
| pmed12 | 7242 | 9.16% | 0.0321 |
| pmed13 | 5006 | 14.45% | 0.1251 |
| pmed14 | 3485 | 17.42% | 0.4531 |
| pmed15 | 2005 | 15.96% | 1.0931 |
| pmed16 | 9346 | 14.51% | 0.0161 |
| pmed17 | 8158 | 16.56% | 0.0341 |
| pmed18 | 5377 | 11.81% | 0.4211 |
| pmed19 | 3319 | 16.66% | 1.4021 |
| pmed20 | 2173 | 21.46% | 3.4311 |
| pmed21 | 10527 | 15.20% | 0.0321 |
| pmed22 | 9595 | 11.84% | 0.0631 |
| pmed23 | 5276 | 14.22% | 1.0001 |
| pmed24 | 3411 | 15.20% | 3.3341 |
| pmed25 | 2162 | 18.27% | 8.2121 |
| pmed26 | 11533 | 16.30% | 0.0321 |
| pmed27 | 9476 | 14.07% | 0.1241 |
| pmed28 | 5138 | 14.23% | 2.0311 |
| pmed29 | 3556 | 17.24% | 6.7731 |
| pmed30 | 2356 | 18.45% | 17.3191 |
| pmed31 | 11959 | 18.57% | 0.0631 |
| pmed32 | 10921 | 17.47% | 0.1881 |
| pmed33 | 5329 | 13.38% | 3.7361 |
| pmed34 | 3537 | 17.39% | 12.1261 |
| pmed35 | 12031 | 15.68% | 0.0941 |
| pmed36 | 11306 | 13.81% | 0.2191 |
| pmed37 | 5748 | 13.66% | 6.1701 |
| pmed38 | 12856 | 16.24% | 0.1281 |
| pmed39 | 10721 | 13.77% | 0.3501 |
| pmed40 | 5817 | 13.44% | 9.9371 |
| **geometric mean** | - | **0.14690** | **0.11772** |

Πίνακας 4.6: Αποτελέσματα αλγορίθμου proportional greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 11715 | 101.32% | 0.0001 |
| pmed2 | 7269 | 77.60% | 0.0001 |
| pmed3 | 7054 | 65.98% | 0.0001 |
| pmed4 | 4813 | 58.64% | 0.0001 |
| pmed5 | 2012 | 48.49% | 0.0001 |
| pmed6 | 14526 | 85.66% | 0.0001 |
| pmed7 | 10618 | 88.56% | 0.0151 |
| pmed8 | 7611 | 71.23% | 0.0311 |
| pmed9 | 3824 | 39.87% | 0.0931 |
| pmed10 | 1935 | 54.18% | 0.2251 |
| pmed11 | 16384 | 112.89% | 0.0001 |
| pmed12 | 14184 | 113.81% | 0.0311 |
| pmed13 | 7295 | 66.78% | 0.1381 |
| pmed14 | 4325 | 45.72% | 0.4851 |
| pmed15 | 2438 | 41.01% | 1.1551 |
| pmed16 | 18840 | 130.83% | 0.0161 |
| pmed17 | 15119 | 116.02% | 0.0461 |
| pmed18 | 7628 | 58.62% | 0.4601 |
| pmed19 | 4038 | 41.93% | 1.4491 |
| pmed20 | 2644 | 47.79% | 3.5611 |
| pmed21 | 19787 | 116.54% | 0.0171 |
| pmed22 | 18665 | 117.57% | 0.0641 |
| pmed23 | 7701 | 66.72% | 1.0311 |
| pmed24 | 4171 | 40.86% | 3.4841 |
| pmed25 | 2701 | 47.76% | 8.8561 |
| pmed26 | 23160 | 133.54% | 0.0031 |
| pmed27 | 19025 | 129.02% | 0.0931 |
| pmed28 | 7785 | 73.08% | 2.1551 |
| pmed29 | 4328 | 42.70% | 7.1251 |
| pmed30 | 2852 | 43.39% | 18.0411 |
| pmed31 | 22085 | 118.97% | 0.0301 |
| pmed32 | 20762 | 123.32% | 0.1411 |
| pmed33 | 7874 | 67.53% | 3.7431 |
| pmed34 | 4562 | 51.41% | 13.7861 |
| pmed35 | 25954 | 149.56% | 0.0471 |
| pmed36 | 21984 | 121.30% | 0.1841 |
| pmed37 | 9259 | 83.09% | 6.4681 |
| pmed38 | 25421 | 129.85% | 0.0311 |
| pmed39 | 21266 | 125.68% | 0.2331 |
| pmed40 | 8926 | 74.06% | 10.0941 |
| **geometric mean** | - | **0.76209** | **0.07952** |

Πίνακας 4.7: Αποτελέσματα αλγορίθμου proportional worst

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 7997 | 37.43% | 0.0001 |
| pmed2 | 7026 | 71.66% | 0.0001 |
| pmed3 | 7530 | 77.18% | 0.0001 |
| pmed4 | 5114 | 68.56% | 0.0001 |
| pmed5 | 2196 | 62.07% | 0.0001 |
| pmed6 | 11815 | 51.01% | 0.0001 |
| pmed7 | 7661 | 36.05% | 0.0001 |
| pmed8 | 6507 | 46.39% | 0.0001 |
| pmed9 | 4247 | 55.34% | 0.0001 |
| pmed10 | 2097 | 67.09% | 0.0041 |
| pmed11 | 13660 | 77.49% | 0.0001 |
| pmed12 | 12193 | 83.80% | 0.0001 |
| pmed13 | 7453 | 70.39% | 0.0001 |
| pmed14 | 4911 | 65.46% | 0.0141 |
| pmed15 | 2747 | 58.88% | 0.0311 |
| pmed16 | 13031 | 59.65% | 0.0001 |
| pmed17 | 11860 | 69.45% | 0.0001 |
| pmed18 | 8423 | 75.15% | 0.0001 |
| pmed19 | 4884 | 71.67% | 0.0151 |
| pmed20 | 2725 | 52.32% | 0.0311 |
| pmed21 | 14167 | 55.03% | 0.0001 |
| pmed22 | 12843 | 49.70% | 0.0001 |
| pmed23 | 7491 | 62.18% | 0.0171 |
| pmed24 | 4598 | 55.29% | 0.0471 |
| pmed25 | 2823 | 54.43% | 0.0761 |
| pmed26 | 13123 | 32.33% | 0.0001 |
| pmed27 | 12822 | 54.35% | 0.0001 |
| pmed28 | 7644 | 69.94% | 0.0141 |
| pmed29 | 4924 | 62.35% | 0.0631 |
| pmed30 | 2799 | 40.72% | 0.1251 |
| pmed31 | 16372 | 62.32% | 0.0001 |
| pmed32 | 14096 | 51.62% | 0.0001 |
| pmed33 | 7893 | 67.94% | 0.0311 |
| pmed34 | 4823 | 60.07% | 0.0991 |
| pmed35 | 13958 | 34.21% | 0.0001 |
| pmed36 | 13569 | 36.59% | 0.0041 |
| pmed37 | 9000 | 77.97% | 0.0621 |
| pmed38 | 17947 | 62.27% | 0.0311 |
| pmed39 | 15422 | 63.66% | 0.0001 |
| pmed40 | 7869 | 53.45% | 0.0941 |
| **geometric mean** | - | **0.57531** | **0.00113** |

Πίνακας 4.8: Αποτελέσματα αλγορίθμου sample greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 7549 | 29.73% | 0.0001 |
| pmed2 | 5390 | 31.69% | 0.0001 |
| pmed3 | 5278 | 24.19% | 0.0001 |
| pmed4 | 3847 | 26.80% | 0.0001 |
| pmed5 | 1879 | 38.67% | 0.0001 |
| pmed6 | 8849 | 13.10% | 0.0001 |
| pmed7 | 7014 | 24.56% | 0.0001 |
| pmed8 | 5635 | 26.77% | 0.0001 |
| pmed9 | 3818 | 39.65% | 0.0161 |
| pmed10 | 2168 | 72.75% | 0.0001 |
| pmed11 | 9484 | 23.23% | 0.0001 |
| pmed12 | 7854 | 18.39% | 0.0001 |
| pmed13 | 5416 | 23.82% | 0.0001 |
| pmed14 | 3893 | 31.17% | 0.0001 |
| pmed15 | 2535 | 46.62% | 0.0001 |
| pmed16 | 10309 | 26.30% | 0.0001 |
| pmed17 | 8480 | 21.16% | 0.0001 |
| pmed18 | 5824 | 21.11% | 0.0001 |
| pmed19 | 3845 | 35.15% | 0.0151 |
| pmed20 | 2622 | 46.56% | 0.0311 |
| pmed21 | 11391 | 24.66% | 0.0001 |
| pmed22 | 9692 | 12.97% | 0.0001 |
| pmed23 | 5725 | 23.94% | 0.0141 |
| pmed24 | 3925 | 32.56% | 0.0161 |
| pmed25 | 3077 | 68.33% | 0.0341 |
| pmed26 | 11415 | 15.11% | 0.0001 |
| pmed27 | 9560 | 15.08% | 0.0001 |
| pmed28 | 5447 | 21.10% | 0.0191 |
| pmed29 | 4109 | 35.48% | 0.0311 |
| pmed30 | 2829 | 42.23% | 0.0781 |
| pmed31 | 11097 | 10.02% | 0.0001 |
| pmed32 | 11014 | 18.47% | 0.0001 |
| pmed33 | 5738 | 22.09% | 0.0161 |
| pmed34 | 4240 | 40.72% | 0.0321 |
| pmed35 | 11179 | 7.49% | 0.0001 |
| pmed36 | 11475 | 15.51% | 0.0001 |
| pmed37 | 6238 | 23.35% | 0.0311 |
| pmed38 | 13427 | 21.40% | 0.0001 |
| pmed39 | 10665 | 13.18% | 0.0001 |
| pmed40 | 6358 | 23.99% | 0.0341 |
| **geometric mean** | - | **0.24826** | **0.00060** |

Πίνακας 4.9: Αποτελέσματα αλγορίθμου drop approach

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 10249 | 76.13% | 0.0201 |
| pmed2 | 7222 | 76.45% | 0.0001 |
| pmed3 | 6825 | 60.59% | 0.0001 |
| pmed4 | 4806 | 58.40% | 0.0091 |
| pmed5 | 2340 | 72.69% | 0.0001 |
| pmed6 | 15833 | 102.36% | 0.0151 |
| pmed7 | 11160 | 98.19% | 0.0001 |
| pmed8 | 8406 | 89.11% | 0.0161 |
| pmed9 | 4524 | 65.47% | 0.0001 |
| pmed10 | 2146 | 71.00% | 0.0001 |
| pmed11 | 12721 | 65.29% | 0.0461 |
| pmed12 | 12426 | 87.31% | 0.0321 |
| pmed13 | 8193 | 87.31% | 0.0311 |
| pmed14 | 5324 | 79.38% | 0.0321 |
| pmed15 | 2512 | 45.29% | 0.0311 |
| pmed16 | 17951 | 119.93% | 0.0621 |
| pmed17 | 13117 | 87.41% | 0.0791 |
| pmed18 | 8915 | 85.38% | 0.0811 |
| pmed19 | 4818 | 69.35% | 0.0631 |
| pmed20 | 2742 | 53.27% | 0.0631 |
| pmed21 | 16349 | 78.91% | 0.1431 |
| pmed22 | 15931 | 85.70% | 0.1411 |
| pmed23 | 8905 | 92.79% | 0.1411 |
| pmed24 | 4872 | 64.54% | 0.1401 |
| pmed25 | 2718 | 48.69% | 0.1411 |
| pmed26 | 16813 | 69.54% | 0.2721 |
| pmed27 | 15147 | 82.34% | 0.2611 |
| pmed28 | 8233 | 83.04% | 0.2471 |
| pmed29 | 4945 | 63.04% | 0.2511 |
| pmed30 | 2857 | 43.64% | 0.2361 |
| pmed31 | 16922 | 67.78% | 0.4081 |
| pmed32 | 20738 | 123.06% | 0.4071 |
| pmed33 | 8463 | 80.06% | 0.4071 |
| pmed34 | 5221 | 73.28% | 0.3771 |
| pmed35 | 16245 | 56.20% | 0.6101 |
| pmed36 | 19559 | 96.89% | 0.5931 |
| pmed37 | 10152 | 100.75% | 0.5781 |
| pmed38 | 18837 | 70.32% | 0.8611 |
| pmed39 | 19496 | 106.90% | 0.8731 |
| pmed40 | 9793 | 90.97% | 0.8571 |
| **geometric mean** | - | **0.76042** | **0.04420** |

Πίνακας 4.10: Αποκλίσεις αλγορίθμου Teitz and Bart από τις βέλτιστες λύσεις

| αρχείο | myopic | random | rpg | Rgreedy | Pgreedy | Pworst | Sgreedy | Dapproach |
|---|---|---|---|---|---|---|---|---|
| pmed1 | 0.00% | 1.24% | 0.00% | 0.03% | 1.24% | 1.24% | 0.03% | 1.24% |
| pmed2 | 0.29% | 0.00% | 0.00% | 0.22% | 0.00% | 1.05% | 0.59% | 2.17% |
| pmed3 | 0.09% | 0.73% | 0.42% | 0.07% | 0.26% | 2.45% | 0.94% | 0.99% |
| pmed4 | 0.40% | 0.63% | 0.49% | 0.63% | 0.79% | 0.13% | 1.58% | 0.73% |
| pmed5 | 0.00% | 0.44% | 0.52% | 1.18% | 0.37% | 0.52% | 1.77% | 1.92% |
| pmed6 | 0.77% | 0.36% | 0.60% | 1.98% | 1.80% | 2.59% | 1.61% | 2.59% |
| pmed7 | 0.25% | 0.36% | 0.00% | 0.34% | 0.67% | 0.27% | 0.34% | 0.30% |
| pmed8 | 0.27% | 0.65% | 0.34% | 0.79% | 1.46% | 0.31% | 0.45% | 1.55% |
| pmed9 | 0.69% | 1.46% | 0.26% | 0.62% | 1.43% | 1.24% | 2.85% | 0.88% |
| pmed10 | 1.35% | 1.51% | 0.56% | 1.43% | 1.75% | 1.67% | 1.83% | 1.91% |
| pmed11 | 0.00% | 0.00% | 0.00% | 0.36% | 0.32% | 0.32% | 0.00% | 0.32% |
| pmed12 | 0.00% | 1.04% | 0.00% | 0.65% | 0.26% | 0.45% | 0.23% | 0.26% |
| pmed13 | 0.14% | 0.57% | 0.30% | 0.91% | 1.23% | 1.03% | 2.42% | 2.33% |
| pmed14 | 0.47% | 1.65% | 0.94% | 0.57% | 1.68% | 2.02% | 1.92% | 3.13% |
| pmed15 | 0.52% | 1.62% | 1.39% | 1.04% | 1.10% | 1.45% | 1.27% | 1.50% |
| pmed16 | 0.04% | 1.07% | 0.04% | 0.06% | 0.86% | 0.06% | 0.06% | 0.86% |
| pmed17 | 0.00% | 0.16% | 0.16% | 0.49% | 0.54% | 0.29% | 0.21% | 0.33% |
| pmed18 | 0.06% | 1.46% | 0.52% | 0.87% | 2.43% | 1.35% | 1.68% | 0.48% |
| pmed19 | 0.56% | 2.11% | 1.69% | 1.09% | 1.44% | 2.74% | 1.30% | 2.46% |
| pmed20 | 0.95% | 1.96% | 1.45% | 1.17% | 1.01% | 1.40% | 1.40% | 1.96% |
| pmed21 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 2.95% | 0.00% |
| pmed22 | 1.05% | 1.70% | 0.00% | 0.40% | 1.13% | 1.03% | 1.07% | 1.17% |
| pmed23 | 0.11% | 1.65% | 0.80% | 0.39% | 0.52% | 0.56% | 0.58% | 1.21% |
| pmed24 | 0.14% | 1.22% | 1.35% | 0.98% | 1.11% | 2.16% | 1.35% | 2.80% |
| pmed25 | 0.93% | 1.81% | 2.35% | 2.46% | 1.75% | 2.57% | 2.79% | 2.13% |
| pmed26 | 0.00% | 0.07% | 0.12% | 0.78% | 1.77% | 0.44% | 0.07% | 1.77% |
| pmed27 | 0.14% | 0.11% | 0.11% | 0.63% | 0.69% | 0.69% | 0.57% | 0.69% |
| pmed28 | 0.38% | 0.93% | 0.80% | 0.53% | 0.93% | 1.58% | 0.98% | 1.76% |
| pmed29 | 0.43% | 1.02% | 1.35% | 1.32% | 1.06% | 2.57% | 1.68% | 1.95% |
| pmed30 | 1.01% | 2.61% | 0.90% | 1.66% | 1.86% | 3.17% | 1.71% | 3.02% |
| pmed31 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| pmed32 | 0.04% | 0.31% | 0.10% | 0.68% | 0.37% | 0.33% | 0.67% | 0.37% |
| pmed33 | 0.13% | 1.13% | 0.79% | 0.60% | 1.30% | 2.28% | 0.79% | 2.57% |
| pmed34 | 0.70% | 0.63% | 0.86% | 0.60% | 1.00% | 1.86% | 1.00% | 2.26% |
| pmed35 | 0.00% | 0.06% | 0.00% | 0.01% | 0.06% | 0.05% | 0.81% | 0.06% |
| pmed36 | 0.00% | 0.71% | 0.34% | 1.36% | 0.20% | 0.95% | 0.08% | 0.42% |
| pmed37 | 0.16% | 0.61% | 0.24% | 0.57% | 0.97% | 1.21% | 0.95% | 1.23% |
| pmed38 | 0.68% | 0.24% | 0.00% | 0.42% | 0.84% | 0.84% | 0.88% | 0.84% |
| pmed39 | 0.14% | 0.00% | 0.66% | 0.47% | 0.30% | 0.53% | 0.06% | 0.53% |
| pmed40 | 0.49% | 1.64% | 0.51% | 1.21% | 1.33% | 1.35% | 1.21% | 1.42% |
| **geometric mean** | **0.00013** | **0.00135** | **0.00019** | **0.00280** | **0.00295** | **0.00431** | **0.00371** | **0.00540** |

Πίνακας 4.11: Αποκλίσεις αλγορίθμου Teitz and Bart από τις αρχικές λύσεις

| αρχείο | myopic | random | rpg | Rgreedy | Pgreedy | Pworst | Sgreedy | Dapproach |
|---|---|---|---|---|---|---|---|---|
| pmed1 | 1.22% | 23.67% | 11.08% | 5.82% | 49.71% | 26.33% | 22.89% | 42.52% |
| pmed2 | 0.32% | 38.78% | 25.36% | 10.63% | 43.69% | 41.13% | 23.62% | 42.09% |
| pmed3 | 3.30% | 24.20% | 17.99% | 13.52% | 39.59% | 42.18% | 18.72% | 37.11% |
| pmed4 | 1.36% | 34.63% | 2.68% | 9.51% | 36.46% | 40.59% | 19.89% | 36.41% |
| pmed5 | 1.67% | 45.63% | 4.08% | 19.35% | 32.41% | 37.98% | 26.61% | 40.98% |
| pmed6 | 1.78% | 25.62% | 6.48% | 6.62% | 45.17% | 32.06% | 10.16% | 49.30% |
| pmed7 | 0.02% | 36.02% | 24.31% | 12.61% | 46.61% | 26.30% | 19.45% | 49.39% |
| pmed8 | 0.34% | 32.78% | 7.98% | 14.21% | 40.74% | 31.47% | 20.76% | 46.30% |
| pmed9 | 3.10% | 30.46% | 12.71% | 14.78% | 27.48% | 34.82% | 26.35% | 39.04% |
| pmed10 | 1.78% | 40.88% | 9.40% | 16.85% | 34.01% | 39.15% | 41.05% | 40.40% |
| pmed11 | 0.32% | 25.58% | 4.49% | 7.12% | 52.87% | 43.48% | 18.85% | 39.31% |
| pmed12 | 0.26% | 24.99% | 7.13% | 7.80% | 53.11% | 45.35% | 15.34% | 46.48% |
| pmed13 | 1.95% | 31.68% | 21.01% | 11.83% | 39.30% | 40.71% | 17.28% | 45.37% |
| pmed14 | 1.03% | 33.59% | 6.81% | 14.35% | 30.22% | 38.34% | 22.30% | 42.51% |
| pmed15 | 1.31% | 40.74% | 10.19% | 12.87% | 28.30% | 36.15% | 30.93% | 30.14% |
| pmed16 | 0.81% | 27.10% | 9.85% | 12.62% | 56.31% | 37.33% | 20.78% | 54.14% |
| pmed17 | 0.28% | 24.04% | 1.36% | 13.79% | 53.46% | 40.82% | 17.29% | 46.47% |
| pmed18 | 1.25% | 28.77% | 18.25% | 9.78% | 35.42% | 42.13% | 16.04% | 45.80% |
| pmed19 | 1.31% | 35.53% | 26.26% | 13.35% | 28.53% | 40.15% | 25.05% | 39.50% |
| pmed20 | 3.22% | 43.18% | 41.71% | 16.71% | 31.66% | 33.43% | 30.82% | 33.48% |
| pmed21 | 0.00% | 24.29% | 17.66% | 13.19% | 53.82% | 35.50% | 17.41% | 44.11% |
| pmed22 | 0.01% | 24.11% | 2.91% | 10.23% | 53.52% | 32.52% | 10.53% | 45.52% |
| pmed23 | 1.49% | 25.82% | 20.46% | 12.11% | 39.71% | 37.99% | 18.85% | 47.50% |
| pmed24 | 1.46% | 33.78% | 23.44% | 12.34% | 28.22% | 34.21% | 23.54% | 37.52% |
| pmed25 | 2.69% | 38.11% | 20.28% | 13.37% | 31.14% | 33.58% | 38.93% | 31.31% |
| pmed26 | 1.74% | 27.30% | 12.77% | 13.34% | 56.42% | 24.10% | 13.06% | 39.97% |
| pmed27 | 0.54% | 20.75% | 7.55% | 11.79% | 56.04% | 34.77% | 12.62% | 44.78% |
| pmed28 | 1.40% | 26.08% | 12.57% | 11.99% | 41.68% | 40.23% | 16.61% | 44.41% |
| pmed29 | 1.87% | 33.20% | 16.67% | 13.58% | 29.18% | 36.82% | 24.95% | 37.47% |
| pmed30 | 1.37% | 37.58% | 3.00% | 14.18% | 28.96% | 26.69% | 28.49% | 28.28% |
| pmed31 | 0.00% | 29.23% | 13.18% | 15.66% | 54.33% | 38.39% | 9.11% | 40.40% |
| pmed32 | 0.32% | 28.28% | 3.35% | 14.29% | 55.06% | 33.83% | 15.03% | 55.01% |
| pmed33 | 1.92% | 30.25% | 20.36% | 11.28% | 39.54% | 39.10% | 17.45% | 43.03% |
| pmed34 | 2.03% | 30.78% | 10.72% | 14.31% | 33.30% | 36.37% | 28.23% | 40.99% |
| pmed35 | 0.06% | 28.88% | 8.88% | 13.55% | 59.91% | 25.45% | 6.22% | 35.94% |
| pmed36 | 0.20% | 21.63% | 3.50% | 10.94% | 54.72% | 26.10% | 13.36% | 49.00% |
| pmed37 | 1.04% | 27.89% | 3.91% | 11.52% | 44.85% | 43.13% | 18.16% | 49.58% |
| pmed38 | 0.16% | 27.28% | 4.23% | 13.60% | 56.13% | 37.86% | 16.91% | 40.79% |
| pmed39 | 0.16% | 22.95% | 8.24% | 11.70% | 55.56% | 38.57% | 11.59% | 51.41% |
| pmed40 | 0.71% | 29.55% | 7.44% | 10.78% | 41.79% | 33.96% | 18.37% | 46.89% |
| **geometric mean** | **0.00364** | **0.29806** | **0.09431** | **0.12118** | **0.41678** | **0.35541** | **0.18744** | **0.42065** |

Πίνακας 4.12: Χρόνοι εκτέλεσης αλγορίθμου Teitz and Bart (seconds)

| αρχείο | myopic | random | rpg | Rgreedy | Pgreedy | Pworst | Sgreedy | Dapproach |
|---|---|---|---|---|---|---|---|---|
| pmed1 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0202 |
| pmed2 | 0.0002 | 0.0002 | 0.0162 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 |
| pmed3 | 0.0002 | 0.0002 | 0.0152 | 0.0002 | 0.0162 | 0.0002 | 0.0002 | 0.0002 |
| pmed4 | 0.0022 | 0.0152 | 0.0202 | 0.0222 | 0.0112 | 0.0002 | 0.0172 | 0.0092 |
| pmed5 | 0.0442 | 0.0352 | 0.0282 | 0.0482 | 0.0312 | 0.0322 | 0.0002 | 0.0312 |
| pmed6 | 0.0002 | 0.0002 | 0.0002 | 0.0312 | 0.0002 | 0.0002 | 0.0002 | 0.0152 |
| pmed7 | 0.0242 | 0.0062 | 0.0192 | 0.0172 | 0.0272 | 0.0182 | 0.0002 | 0.0302 |
| pmed8 | 0.0742 | 0.0322 | 0.0482 | 0.0642 | 0.0632 | 0.0472 | 0.0472 | 0.0472 |
| pmed9 | 0.2682 | 0.1572 | 0.2012 | 0.2412 | 0.2572 | 0.1372 | 0.1912 | 0.1382 |
| pmed10 | 0.6222 | 0.4012 | 0.5982 | 0.6392 | 0.6322 | 0.3892 | 0.4132 | 0.3912 |
| pmed11 | 0.0162 | 0.0002 | 0.0162 | 0.0002 | 0.0152 | 0.0002 | 0.0172 | 0.0462 |
| pmed12 | 0.0302 | 0.0312 | 0.0322 | 0.0632 | 0.0472 | 0.0312 | 0.0162 | 0.0632 |
| pmed13 | 0.3382 | 0.1872 | 0.2432 | 0.3462 | 0.3232 | 0.2192 | 0.2032 | 0.2192 |
| pmed14 | 1.1602 | 0.7502 | 1.1432 | 1.1892 | 1.2602 | 0.7632 | 0.7812 | 0.7672 |
| pmed15 | 2.8942 | 1.8902 | 2.7862 | 2.9772 | 3.0942 | 1.9072 | 1.9662 | 1.9072 |
| pmed16 | 0.0162 | 0.0322 | 0.0302 | 0.0312 | 0.0162 | 0.0162 | 0.0162 | 0.0772 |
| pmed17 | 0.0742 | 0.0472 | 0.0792 | 0.0812 | 0.0772 | 0.0472 | 0.0472 | 0.1252 |
| pmed18 | 0.9772 | 0.6092 | 0.8162 | 1.0972 | 1.0762 | 0.6242 | 0.6252 | 0.6952 |
| pmed19 | 3.4842 | 2.2972 | 2.5202 | 3.7472 | 3.7452 | 2.2502 | 2.3062 | 2.2732 |
| pmed20 | 9.0892 | 6.1032 | 6.1692 | 9.4542 | 9.4532 | 5.8352 | 6.1252 | 5.8282 |
| pmed21 | 0.0482 | 0.0162 | 0.0322 | 0.0452 | 0.0472 | 0.0212 | 0.0122 | 0.1612 |
| pmed22 | 0.1232 | 0.0782 | 0.1252 | 0.1582 | 0.1252 | 0.0632 | 0.0992 | 0.2072 |
| pmed23 | 2.3112 | 1.4382 | 1.6892 | 2.5152 | 2.4842 | 1.4902 | 1.5422 | 1.5792 |
| pmed24 | 8.3532 | 5.3752 | 6.1112 | 8.7552 | 9.0162 | 5.4582 | 5.4952 | 5.4842 |
| pmed25 | 22.4722 | 15.2112 | 18.5702 | 23.0872 | 24.4772 | 14.8622 | 15.4712 | 14.8942 |
| pmed26 | 0.0632 | 0.0342 | 0.0482 | 0.0642 | 0.0352 | 0.0312 | 0.0512 | 0.2992 |
| pmed27 | 0.1612 | 0.1092 | 0.1572 | 0.2342 | 0.1862 | 0.0942 | 0.1262 | 0.3492 |
| pmed28 | 5.1512 | 3.4302 | 4.6122 | 5.5752 | 5.5922 | 3.4392 | 3.5872 | 3.6022 |
| pmed29 | 17.3942 | 11.9492 | 15.0842 | 18.4552 | 18.5492 | 11.2802 | 11.4362 | 11.4702 |
| pmed30 | 46.8552 | 31.4652 | 47.4302 | 48.3942 | 49.6182 | 31.1272 | 32.2182 | 31.2992 |
| pmed31 | 0.0712 | 0.0352 | 0.0812 | 0.1112 | 0.0732 | 0.0322 | 0.0472 | 0.4552 |
| pmed32 | 0.2352 | 0.1552 | 0.2652 | 0.3462 | 0.2802 | 0.1402 | 0.1722 | 0.5322 |
| pmed33 | 8.5452 | 5.5012 | 6.6932 | 9.3652 | 9.1172 | 5.4352 | 5.4532 | 5.7222 |
| pmed34 | 32.4412 | 22.2042 | 31.9972 | 33.7202 | 35.4272 | 21.3802 | 22.0322 | 21.7202 |
| pmed35 | 0.1082 | 0.0632 | 0.0802 | 0.1592 | 0.0902 | 0.0642 | 0.0612 | 0.6602 |
| pmed36 | 0.3462 | 0.2182 | 0.3572 | 0.4302 | 0.3542 | 0.2062 | 0.2452 | 0.7502 |
| pmed37 | 14.6022 | 9.6162 | 14.9092 | 15.7482 | 15.6402 | 9.1782 | 9.7592 | 9.6812 |
| pmed38 | 0.1322 | 0.0772 | 0.1152 | 0.2082 | 0.0942 | 0.1122 | 0.0592 | 0.9252 |
| pmed39 | 0.4392 | 0.2562 | 0.4272 | 0.6472 | 0.4522 | 0.2332 | 0.2812 | 1.0932 |
| pmed40 | 23.7682 | 15.6812 | 23.2072 | 25.7172 | 25.3432 | 15.3452 | 15.5352 | 16.0652 |
| **geometric mean** | **0.24506** | **0.15541** | **0.29045** | **0.29740** | **0.28722** | **0.14075** | **0.14192** | **0.38660** |

Πίνακας 4.13: Αποκλίσεις αλγορίθμου Neighborhood search από τις βέλτιστες λύσεις

| αρχείο | myopic | random | rpg | Rgreedy | Pgreedy | Pworst | Sgreedy | Dapproach |
|---|---|---|---|---|---|---|---|---|
| pmed1 | 1.24% | 16.57% | 9.71% | 1.41% | 15.50% | 3.21% | 11.70% | 12.06% |
| pmed2 | 0.29% | 27.12% | 26.22% | 2.71% | 13.76% | 16.98% | 16.37% | 41.68% |
| pmed3 | 2.02% | 22.54% | 9.95% | 7.84% | 12.94% | 33.04% | 5.55% | 36.21% |
| pmed4 | 1.78% | 29.33% | 1.78% | 6.72% | 17.90% | 22.38% | 11.93% | 27.82% |
| pmed5 | 0.15% | 55.20% | 1.70% | 12.47% | 41.48% | 29.74% | 24.28% | 19.04% |
| pmed6 | 2.59% | 23.27% | 6.42% | 5.70% | 7.96% | 35.29% | 8.79% | 19.57% |
| pmed7 | 0.25% | 12.75% | 20.81% | 12.72% | 19.36% | 22.11% | 8.95% | 17.39% |
| pmed8 | 0.27% | 27.24% | 4.72% | 10.37% | 16.74% | 32.69% | 15.95% | 23.31% |
| pmed9 | 2.34% | 25.93% | 10.94% | 14.16% | 18.95% | 20.45% | 21.18% | 34.13% |
| pmed10 | 2.63% | 37.45% | 9.32% | 13.86% | 33.15% | 12.67% | 45.42% | 17.93% |
| pmed11 | 0.23% | 14.48% | 4.55% | 2.82% | 7.47% | 12.46% | 4.44% | 18.06% |
| pmed12 | 0.26% | 22.69% | 7.27% | 4.37% | 24.04% | 18.30% | 6.71% | 34.31% |
| pmed13 | 1.58% | 27.18% | 20.05% | 8.76% | 19.52% | 31.98% | 12.71% | 43.58% |
| pmed14 | 0.98% | 28.00% | 6.20% | 10.44% | 25.37% | 26.04% | 21.63% | 21.66% |
| pmed15 | 1.10% | 41.70% | 9.43% | 11.86% | 23.42% | 19.32% | 31.93% | 24.18% |
| pmed16 | 0.86% | 8.81% | 8.94% | 3.42% | 11.20% | 17.52% | 13.60% | 4.03% |
| pmed17 | 0.16% | 13.89% | 1.19% | 11.69% | 12.39% | 35.56% | 6.54% | 27.65% |
| pmed18 | 0.31% | 21.15% | 17.43% | 6.18% | 23.91% | 27.72% | 11.04% | 31.03% |
| pmed19 | 0.77% | 30.05% | 21.83% | 7.87% | 21.97% | 30.97% | 17.05% | 31.63% |
| pmed20 | 2.63% | 54.05% | 48.80% | 9.61% | 34.54% | 16.66% | 32.42% | 27.72% |
| pmed21 | 0.00% | 13.16% | 11.88% | 6.30% | 2.45% | 11.39% | 13.22% | 12.40% |
| pmed22 | 1.06% | 15.82% | 2.31% | 6.25% | 13.03% | 23.03% | 5.28% | 25.67% |
| pmed23 | 0.87% | 22.71% | 14.40% | 7.38% | 30.57% | 25.46% | 13.18% | 34.34% |
| pmed24 | 0.61% | 23.57% | 19.93% | 9.12% | 22.86% | 22.09% | 17.56% | 22.56% |
| pmed25 | 1.86% | 42.29% | 19.91% | 11.71% | 29.87% | 17.78% | 43.00% | 19.75% |
| pmed26 | 1.77% | 12.13% | 11.47% | 8.51% | 6.51% | 16.63% | 14.23% | 19.42% |
| pmed27 | 0.64% | 16.46% | 5.51% | 6.74% | 8.70% | 12.17% | 12.07% | 26.06% |
| pmed28 | 1.18% | 22.23% | 11.47% | 9.16% | 26.86% | 36.26% | 13.81% | 30.12% |
| pmed29 | 1.02% | 26.44% | 14.47% | 9.43% | 22.29% | 28.55% | 18.76% | 32.41% |
| pmed30 | 1.46% | 41.13% | 2.46% | 12.47% | 26.29% | 18.10% | 30.57% | 19.56% |
| pmed31 | 0.00% | 6.10% | 10.34% | 5.95% | 3.20% | 24.67% | 8.10% | 16.68% |
| pmed32 | 0.23% | 14.32% | 2.97% | 8.28% | 9.72% | 18.18% | 9.35% | 19.97% |
| pmed33 | 1.21% | 23.09% | 17.49% | 5.74% | 30.55% | 29.72% | 9.66% | 27.62% |
| pmed34 | 1.59% | 27.95% | 9.53% | 8.53% | 25.66% | 25.59% | 24.00% | 32.82% |
| pmed35 | 0.06% | 7.72% | 9.74% | 7.94% | 13.38% | 8.46% | 4.86% | 30.27% |
| pmed36 | 0.13% | 16.51% | 2.28% | 7.35% | 8.36% | 23.03% | 7.58% | 23.05% |
| pmed37 | 0.65% | 28.46% | 3.10% | 7.02% | 42.69% | 44.95% | 11.33% | 30.51% |
| pmed38 | 0.71% | 3.90% | 3.85% | 2.88% | 5.45% | 16.39% | 7.61% | 32.68% |
| pmed39 | 0.30% | 21.82% | 4.74% | 4.33% | 12.93% | 19.00% | 9.63% | 15.91% |
| pmed40 | 1.03% | 26.68% | 6.06% | 8.66% | 28.28% | 21.61% | 14.63% | 31.69% |
| **geometric mean** | **0.00359** | **0.20862** | **0.07840** | **0.07193** | **0.16200** | **0.20790** | **0.13026** | **0.23713** |

Πίνακας 4.14: Αποκλίσεις αλγορίθμου Neighborhood search από τις αρχικές λύσεις

| αρχείο | myopic | random | rpg | Rgreedy | Pgreedy | Pworst | Sgreedy | Dapproach |
|---|---|---|---|---|---|---|---|---|
| pmed1 | 0.00% | 12.11% | 2.44% | 4.53% | 42.63% | 24.90% | 13.90% | 36.37% |
| pmed2 | 0.32% | 22.18% | 5.80% | 8.41% | 35.95% | 31.85% | 11.63% | 19.70% |
| pmed3 | 1.43% | 7.79% | 10.20% | 6.81% | 31.95% | 24.91% | 15.01% | 15.18% |
| pmed4 | 0.00% | 15.97% | 1.44% | 4.03% | 25.68% | 27.40% | 11.72% | 19.31% |
| pmed5 | 1.52% | 15.98% | 2.96% | 10.35% | 4.72% | 19.95% | 10.38% | 31.07% |
| pmed6 | 0.00% | 8.63% | 1.07% | 3.22% | 41.85% | 10.41% | 3.81% | 40.91% |
| pmed7 | 0.02% | 28.12% | 8.56% | 1.83% | 36.70% | 10.25% | 12.53% | 40.77% |
| pmed8 | 0.34% | 15.02% | 3.96% | 6.05% | 31.82% | 9.36% | 8.54% | 34.80% |
| pmed9 | 1.51% | 13.69% | 3.41% | 3.31% | 14.96% | 22.46% | 13.23% | 18.94% |
| pmed10 | 0.54% | 19.95% | 1.51% | 6.66% | 13.64% | 32.57% | 15.82% | 31.03% |
| pmed11 | 0.09% | 14.81% | 0.15% | 4.85% | 49.52% | 36.64% | 15.25% | 28.57% |
| pmed12 | 0.00% | 8.92% | 0.38% | 4.39% | 41.98% | 35.64% | 9.87% | 28.30% |
| pmed13 | 0.54% | 13.60% | 5.46% | 4.97% | 28.33% | 22.54% | 8.97% | 23.35% |
| pmed14 | 0.53% | 16.38% | 1.96% | 5.94% | 13.97% | 23.82% | 7.27% | 32.18% |
| pmed15 | 0.74% | 17.37% | 3.07% | 3.54% | 12.47% | 24.90% | 10.02% | 14.53% |
| pmed16 | 0.00% | 21.51% | 1.82% | 9.68% | 51.83% | 26.39% | 10.06% | 52.70% |
| pmed17 | 0.13% | 13.62% | 0.35% | 4.18% | 47.97% | 20.00% | 12.06% | 31.89% |
| pmed18 | 1.01% | 14.95% | 4.50% | 5.04% | 21.88% | 27.08% | 8.31% | 29.32% |
| pmed19 | 1.10% | 17.89% | 11.65% | 7.53% | 14.07% | 23.71% | 13.39% | 22.27% |
| pmed20 | 1.61% | 14.14% | 14.52% | 9.76% | 8.96% | 23.41% | 9.65% | 16.67% |
| pmed21 | 0.00% | 14.32% | 7.88% | 7.72% | 52.69% | 28.15% | 9.17% | 37.18% |
| pmed22 | 0.00% | 13.58% | 0.67% | 5.00% | 48.05% | 17.82% | 6.81% | 32.33% |
| pmed23 | 0.75% | 10.44% | 9.74% | 5.99% | 21.69% | 22.64% | 8.68% | 30.32% |
| pmed24 | 1.00% | 19.16% | 9.41% | 5.28% | 12.78% | 21.38% | 11.31% | 25.51% |
| pmed25 | 1.79% | 13.50% | 6.60% | 5.55% | 12.11% | 23.73% | 15.05% | 19.46% |
| pmed26 | 0.00% | 18.53% | 2.89% | 6.69% | 54.39% | 11.86% | 0.76% | 29.56% |
| pmed27 | 0.05% | 7.81% | 2.56% | 6.43% | 52.54% | 27.33% | 2.62% | 30.86% |
| pmed28 | 0.61% | 10.49% | 3.32% | 4.44% | 26.71% | 19.82% | 6.02% | 28.91% |
| pmed29 | 1.29% | 16.39% | 5.88% | 6.66% | 14.30% | 20.82% | 12.34% | 18.79% |
| pmed30 | 0.93% | 14.16% | 1.50% | 5.05% | 11.92% | 16.08% | 8.20% | 16.77% |
| pmed31 | 0.00% | 24.91% | 4.20% | 10.64% | 52.87% | 23.20% | 1.75% | 30.46% |
| pmed32 | 0.14% | 18.27% | 0.58% | 7.82% | 50.87% | 22.06% | 7.70% | 46.21% |
| pmed33 | 0.85% | 15.10% | 7.16% | 6.74% | 22.07% | 22.75% | 10.18% | 29.13% |
| pmed34 | 1.16% | 11.99% | 3.06% | 7.55% | 17.01% | 21.54% | 11.89% | 23.35% |
| pmed35 | 0.00% | 23.43% | 0.00% | 6.69% | 54.57% | 19.19% | 2.45% | 16.60% |
| pmed36 | 0.07% | 9.34% | 1.64% | 5.68% | 51.04% | 9.93% | 6.87% | 37.50% |
| pmed37 | 0.55% | 7.94% | 1.16% | 5.85% | 22.07% | 18.56% | 9.75% | 34.99% |
| pmed38 | 0.13% | 24.62% | 0.55% | 11.50% | 54.12% | 28.27% | 11.36% | 22.10% |
| pmed39 | 0.00% | 6.14% | 4.52% | 8.30% | 49.96% | 27.29% | 3.14% | 43.98% |
| pmed40 | 0.17% | 12.19% | 2.32% | 4.21% | 26.31% | 20.75% | 7.55% | 31.04% |
| **geometric mean** | **0.00013** | **0.14262** | **0.01842** | **0.05844** | **0.27083** | **0.21483** | **0.08135** | **0.27434** |

Πίνακας 4.15: Χρόνοι εκτέλεσης αλγορίθμου Neighborhood search (seconds)

| αρχείο | myopic | random | rpg | Rgreedy | Pgreedy | Pworst | Sgreedy | Dapproach |
|---|---|---|---|---|---|---|---|---|
| pmed1 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0202 |
| pmed2 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 |
| pmed3 | 0.0002 | 0.0002 | 0.0152 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 |
| pmed4 | 0.0002 | 0.0002 | 0.0202 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0092 |
| pmed5 | 0.0132 | 0.0002 | 0.0162 | 0.0172 | 0.0002 | 0.0002 | 0.0002 | 0.0002 |
| pmed6 | 0.0002 | 0.0002 | 0.0002 | 0.0162 | 0.0002 | 0.0002 | 0.0002 | 0.0312 |
| pmed7 | 0.0002 | 0.0002 | 0.0002 | 0.0172 | 0.0152 | 0.0002 | 0.0002 | 0.0002 |
| pmed8 | 0.0272 | 0.0012 | 0.0012 | 0.0182 | 0.0312 | 0.0002 | 0.0002 | 0.0162 |
| pmed9 | 0.0942 | 0.0002 | 0.0472 | 0.0902 | 0.0932 | 0.0002 | 0.0162 | 0.0002 |
| pmed10 | 0.2192 | 0.0002 | 0.1932 | 0.2502 | 0.2252 | 0.0042 | 0.0002 | 0.0002 |
| pmed11 | 0.0162 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0002 | 0.0462 |
| pmed12 | 0.0152 | 0.0002 | 0.0002 | 0.0322 | 0.0312 | 0.0002 | 0.0002 | 0.0322 |
| pmed13 | 0.1192 | 0.0002 | 0.0402 | 0.1252 | 0.1382 | 0.0002 | 0.0002 | 0.0312 |
| pmed14 | 0.4102 | 0.0002 | 0.3882 | 0.4532 | 0.4852 | 0.0142 | 0.0002 | 0.0322 |
| pmed15 | 1.0302 | 0.0002 | 0.8142 | 1.0932 | 1.1552 | 0.0312 | 0.0002 | 0.0312 |
| pmed16 | 0.0002 | 0.0002 | 0.0152 | 0.0162 | 0.0162 | 0.0002 | 0.0002 | 0.0622 |
| pmed17 | 0.0272 | 0.0002 | 0.0322 | 0.0342 | 0.0462 | 0.0002 | 0.0002 | 0.0792 |
| pmed18 | 0.3442 | 0.0002 | 0.1402 | 0.4212 | 0.4602 | 0.0002 | 0.0002 | 0.0812 |
| pmed19 | 1.2652 | 0.0002 | 0.2552 | 1.4022 | 1.4492 | 0.0152 | 0.0422 | 0.0632 |
| pmed20 | 3.2292 | 0.0092 | 0.1392 | 3.4312 | 3.5612 | 0.0312 | 0.0312 | 0.0632 |
| pmed21 | 0.0142 | 0.0002 | 0.0132 | 0.0322 | 0.0172 | 0.0002 | 0.0002 | 0.1432 |
| pmed22 | 0.0462 | 0.0002 | 0.0472 | 0.0702 | 0.0642 | 0.0002 | 0.0002 | 0.1412 |
| pmed23 | 0.8272 | 0.0002 | 0.1732 | 1.0002 | 1.0312 | 0.0172 | 0.0142 | 0.1412 |
| pmed24 | 2.9792 | 0.0002 | 0.6722 | 3.3342 | 3.4842 | 0.0472 | 0.0162 | 0.1402 |
| pmed25 | 7.6102 | 0.0312 | 3.2572 | 8.2122 | 8.8562 | 0.0762 | 0.0342 | 0.1412 |
| pmed26 | 0.0312 | 0.0002 | 0.0272 | 0.0322 | 0.0032 | 0.0002 | 0.0002 | 0.2872 |
| pmed27 | 0.0622 | 0.0002 | 0.0322 | 0.1242 | 0.0932 | 0.0002 | 0.0092 | 0.2612 |
| pmed28 | 1.6872 | 0.0192 | 0.9372 | 2.0312 | 2.1552 | 0.0142 | 0.0192 | 0.2472 |
| pmed29 | 6.1252 | 0.0162 | 3.3122 | 6.7732 | 7.1252 | 0.0632 | 0.0312 | 0.2512 |
| pmed30 | 16.0402 | 0.0622 | 16.2882 | 17.3192 | 18.0412 | 0.1252 | 0.0782 | 0.2362 |
| pmed31 | 0.0302 | 0.0002 | 0.0202 | 0.0632 | 0.0302 | 0.0002 | 0.0002 | 0.4232 |
| pmed32 | 0.0782 | 0.0002 | 0.0932 | 0.1882 | 0.1412 | 0.0002 | 0.0002 | 0.4232 |
| pmed33 | 3.1092 | 0.0152 | 1.0542 | 3.7362 | 3.7432 | 0.0312 | 0.0162 | 0.4072 |
| pmed34 | 11.1932 | 0.0312 | 9.4672 | 12.1262 | 13.7862 | 0.0992 | 0.0322 | 0.3772 |
| pmed35 | 0.0462 | 0.0002 | 0.0162 | 0.0942 | 0.0632 | 0.0002 | 0.0002 | 0.6102 |
| pmed36 | 0.1412 | 0.0002 | 0.1232 | 0.2192 | 0.1842 | 0.0042 | 0.0002 | 0.5932 |
| pmed37 | 5.2662 | 0.0092 | 5.2372 | 6.1702 | 6.4682 | 0.0622 | 0.0312 | 0.5782 |
| pmed38 | 0.0542 | 0.0002 | 0.0372 | 0.1482 | 0.0312 | 0.0412 | 0.0002 | 0.8612 |
| pmed39 | 0.1572 | 0.0002 | 0.1052 | 0.3502 | 0.2332 | 0.0152 | 0.0002 | 0.8732 |
| pmed40 | 8.3312 | 0.0202 | 7.1632 | 9.9372 | 10.0942 | 0.0942 | 0.0342 | 0.8572 |
| **geometric mean** | **0.08200** | **0.00059** | **0.05557** | **0.12934** | **0.09061** | **0.00189** | **0.00109** | **0.05016** |

Πίνακας 4.16: Αποκλίσεις από βέλτιστη λύση

| όνομα | γεωμ.μέσος (απόκλισης) |
|---|---|
| myopic | 0.00556 |
| random | 0.44208 |
| random plus greedy | 0.11386 |
| randomized greedy | 0.14690 |
| proportional greedy | 0.76209 |
| proportional worst | 0.57531 |
| sample greedy | 0.24826 |
| drop approach | 0.76042 |
| Teitz and Bart | 0.00013 |
| neighborhood search | 0.00359 |

Πίνακας 4.17: Χρόνοι εκτέλεσης

| όνομα | γεωμ.μέσος (χρόνου) |
|---|---|
| myopic | 0.07232 |
| random | 0.00031 |
| random plus greedy | 0.04416 |
| randomized greedy | 0.11772 |
| proportional greedy | 0.07952 |
| proportional worst | 0.00113 |
| sample greedy | 0.00060 |
| drop approach | 0.04420 |
| Teitz and Bart | 0.13935 |
| neighborhood search | 0.00015 |

# Κεφάλαιο 5

# Συμπεράσματα

Συνοψίζοντας, στην εργασία αυτή υλοποιήθηκαν οι παρακάτω αλγόριθμοι για την επίλυση του προβλήματος χωροθέτησης μονάδων και συγκεκριμένα του προβλήματος του $p$-μέσου:

- Ακριβής
  1. Brute force
- Κατασκευαστικοί
  1. Myopic
  2. Random
  3. Random plus greedy
  4. Randomized greedy
  5. Proportional greedy
  6. Proportional worst
  7. Sample greedy
  8. Drop approach
- Βελτιωτικοί
  1. Teitz and Bart
  2. Neighborhood search

Στο δεύτερο κεφάλαιο έγινε μία ιστορική αναδρομή σε εργασίες άλλων ερευνητών που ασχολήθηκαν με το πρόβλημα, ενώ στο τρίτο κεφάλαιο αναλύθηκε κάθε αλγόριθμος ξεχωριστά και δόθηκε ένα σύντομο παράδειγμα και ψευδοκώδικας για τον καθένα με στόχο την καλύτερη κατανόησή του. Στο τέταρτο κεφάλαιο παρουσιάστηκαν τα αποτελέσματα της εκτέλεσης όλων των αλγορίθμων (εκτός του Brute force) στα 40 αρχεία της βιβλιοθήκης ORLIB [?] (pmed1.txt, pmed2.txt, ..., pmed40.txt). Τα αποτελέσματα περιλαμβάνουν τους χρόνους εκτέλεσης, τις λύσεις και τις αποκλίσεις από τις βέλτιστες λύσεις για κάθε αλγόριθμο για κάθε αρχείο. Ο αλγόριθμος Brute force δεν εξετάστηκε καθώς οι χρόνοι εκτέλεσης είναι πολλοί μεγάλοι για τα συγκεκριμένα αρχεία (για τα 38 από τα 40 αρχεία απαιτείται χρόνος μεγαλύτερος της μίας ώρας).

Όσον αφορά τους κατασκευαστικούς αλγορίθμους, αυτούς δηλαδή που δε χρειάζεται να τροφοδοτηθούν με μία αρχική εφικτή λύση, ο καλύτερος ως προς την απόκλιση είναι ο Myopic ενώ ο χειρότερος είναι ο Drop approach. Από την πλευρά του χρόνου εκτέλεσης ταχύτερος φαίνεται να είναι ο αλγόριθμος Random ενώ ο πιο αργός είναι ο Randomized greedy. Στις λίστες που ακολουθουν φαίνεται η κατάταξη των κατασκευαστικών αλγορίθμων σύμφωνα με την απόκλιση και το χρόνο εκτέλεσης.

- Απόκλιση από βέλτιστη λύση
  1. Myopic
  2. Random plus greedy
  3. Randomized greedy
  4. Sample greedy
  5. Random
  6. Proportional worst
  7. Drop approach
  8. Proportional greedy
- Χρόνος εκέλεσης
  1. Random
  2. Sample greedy
  3. Proportional worst

4. Random plus greedy
5. Drop approach
6. Myopic
7. Proportional greedy
8. Randomized greedy

Τέλος, από τους βελτιωτικούς αλγορίθμους ο Teitz and Bart σε κάθε περίπτωση δίνει λύση πιο κοντά στη βέλτιστη (έχει μικρότερη απόκλιση) αλλά ταυτόχρονα απαιτεί περισσότερο χρόνο εκτέλεσης από τον αλγόριθμο Neighborhood search.

# Βιβλιογραφία


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

[2] John E Beasley. Lagrangean heuristics for location problems. *European Journal of Operational Research*, 65(3):383–399, 1993.

[3] Fernando Chiyoshi and Roberto D. Galvão. A statistical analysis of simulated annealing applied to the p-median problem. *Annals of Operations Research*, 96(1):61–74, Nov 2000.

[4] Richard L Church. Cobra: a new formulation of the classic p-median location problem. *Annals of Operations Research*, 122(1-4):103–120, 2003.

[5] Mark S Daskin and Kayse Lee Maass. The p-median problem. In *Location science*, pages 21–45. Springer, 2015.

[6] C. Dibble and Pj Densham. Generating interesting alternatives in gis and sdss using genetic algorithms. 1993.

[7] MA Efroymson and TL Ray. A branch-bound algorithm for plant location. *Operations Research*, 14(3):361–368, 1966.

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

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

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

[11] Félix Garcıá-López, Belén Melián-Batista, José A Moreno-Pérez, and J Marcos Moreno-Vega. Parallelization of the scatter search for the p-median problem. *Parallel Computing*, 29(5):575 – 589, 2003. Parallel computing in logistics.

[12] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, USA, 1997.

[13] S Louis Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475, 1965.

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

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

[16] O. Kariv and S. Hakimi. An algorithmic approach to network location problems. i: The p-centers. 1979.

[17] Oded Kariv and S Louis Hakimi. An algorithmic approach to network location problems. i: The p-centers. *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

[18] Alfred A Kuehn and Michael J Hamburger. A heuristic program for locating warehouses. *Management science*, 9(4):643–666, 1963.

[19] T. V. Levanova and M. A. Loresh. Algorithms of ant system and simulated annealing for the p-median problem. *Automation and Remote Control*, 65(3):431–438, Mar 2004.

[20] FE Maranzana. On the location of supply points to minimize transport costs. *Journal of the Operational Research Society*, 15(3):261–270, 1964.

[21] Wangshu Mu and Daoqin Tong. A spatial-knowledge-enhanced heuristic for solving the p-median problem. *Transactions in GIS*, 22(2):477–493, 2018.

[22] Alan T. Murray and Richard L. Church. Applying simulated annealing to location-planning models. *Journal of Heuristics*, 2(1):31–53, Jun 1996.

[23] JA Moreno Pérez, JL Roda Garcia, and M Moreno. A parallel genetic algorithm for the discrete p-median problem. *Studies in Locational Analysis*, 7:131–141, 1994.

[24] Mauricio GC Resende and Renato F Werneck. A hybrid heuristic for the p-median problem. *Journal of heuristics*, 10(1):59–88, 2004.

[25] Charles S ReVelle and Ralph W Swain. Central facilities location. *Geographical analysis*, 2(1):30–42, 1970.

[26] Kenneth E Rosing, CS ReVelle, and H Rosing-Vogelaar. The p-median and its linear programming relaxation: An approach to large problems. *Journal of the Operational Research Society*, 30(9):815–823, 1979.

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

[28] Edson LF Senne and Luiz AN Lorena. Lagrangean/surrogate heuristics for p-median problems. In *Computing tools for modeling, optimization and simulation*, pages 115–130. Springer, 2000.

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

[30] Michael B Teitz and Polly Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations research*, 16(5):955–961, 1968.

[31] R. Whitaker. A fast algorithm for the greedy interchange for large-scale clustering and median location problems. *Infor*, 21:95–108, 1983.

# Παραρτήματα

# Παράρτημα Α΄

# Αναλυτικά αποτελέσματα αλγορίθμων

Πίνακας Α΄.1: Myopic

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5891 | 1.24% | 0.0001 |
| pmed2 | 4118 | 0.61% | 0.0001 |
| pmed3 | 4399 | 3.51% | 0.0001 |
| pmed4 | 3088 | 1.78% | 0.0001 |
| pmed5 | 1378 | 1.70% | 0.0131 |
| pmed6 | 8027 | 2.59% | 0.0001 |
| pmed7 | 5646 | 0.27% | 0.0001 |
| pmed8 | 4472 | 0.61% | 0.0271 |
| pmed9 | 2841 | 3.91% | 0.0941 |
| pmed10 | 1295 | 3.19% | 0.2191 |
| pmed11 | 7721 | 0.32% | 0.0161 |
| pmed12 | 6651 | 0.26% | 0.0151 |
| pmed13 | 4467 | 2.13% | 0.1191 |
| pmed14 | 3013 | 1.52% | 0.4101 |
| pmed15 | 1761 | 1.85% | 1.0301 |
| pmed16 | 8232 | 0.86% | 0.0001 |
| pmed17 | 7019 | 0.29% | 0.0271 |
| pmed18 | 4873 | 1.33% | 0.3441 |
| pmed19 | 2899 | 1.90% | 1.2651 |
| pmed20 | 1866 | 4.30% | 3.2291 |
| pmed21 | 9138 | 0.00% | 0.0141 |
| pmed22 | 8670 | 1.06% | 0.0461 |
| pmed23 | 4694 | 1.62% | 0.8271 |
| pmed24 | 3009 | 1.62% | 2.9791 |
| pmed25 | 1896 | 3.72% | 7.6101 |
| pmed26 | 10093 | 1.77% | 0.0311 |
| pmed27 | 8364 | 0.69% | 0.0621 |
| pmed28 | 4579 | 1.80% | 1.6871 |
| pmed29 | 3104 | 2.34% | 6.1251 |
| pmed30 | 2037 | 2.41% | 16.0401 |
| pmed31 | 10086 | 0.00% | 0.0301 |
| pmed32 | 9331 | 0.37% | 0.0781 |
| pmed33 | 4798 | 2.09% | 3.1091 |
| pmed34 | 3097 | 2.79% | 11.1771 |
| pmed35 | 10406 | 0.06% | 0.0461 |
| pmed36 | 9954 | 0.20% | 0.1261 |
| pmed37 | 5118 | 1.21% | 5.2661 |
| pmed38 | 11153 | 0.84% | 0.0541 |
| pmed39 | 9451 | 0.30% | 0.1571 |
| pmed40 | 5190 | 1.21% | 8.3161 |

Πίνακας Α΄.2: Random

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 7718 | 32.63% | 0.0001 |
| pmed2 | 6686 | 63.35% | 0.0001 |
| pmed3 | 5648 | 32.89% | 0.0001 |
| pmed4 | 4670 | 53.92% | 0.0001 |
| pmed5 | 2503 | 84.72% | 0.0001 |
| pmed6 | 10556 | 34.92% | 0.0001 |
| pmed7 | 8833 | 56.86% | 0.0001 |
| pmed8 | 6656 | 49.74% | 0.0011 |
| pmed9 | 3989 | 45.90% | 0.0001 |
| pmed10 | 2155 | 71.71% | 0.0001 |
| pmed11 | 10342 | 34.38% | 0.0001 |
| pmed12 | 8936 | 34.70% | 0.0001 |
| pmed13 | 6439 | 47.21% | 0.0001 |
| pmed14 | 4543 | 53.07% | 0.0001 |
| pmed15 | 2965 | 71.49% | 0.0001 |
| pmed16 | 11315 | 38.63% | 0.0001 |
| pmed17 | 9228 | 31.85% | 0.0001 |
| pmed18 | 6850 | 42.44% | 0.0001 |
| pmed19 | 4506 | 58.38% | 0.0001 |
| pmed20 | 3210 | 79.43% | 0.0091 |
| pmed21 | 12070 | 32.09% | 0.0001 |
| pmed22 | 11497 | 34.01% | 0.0001 |
| pmed23 | 6329 | 37.02% | 0.0001 |
| pmed24 | 4526 | 52.85% | 0.0001 |
| pmed25 | 3007 | 64.50% | 0.0311 |
| pmed26 | 13650 | 37.64% | 0.0001 |
| pmed27 | 10493 | 26.32% | 0.0001 |
| pmed28 | 6142 | 36.55% | 0.0191 |
| pmed29 | 4587 | 51.24% | 0.0161 |
| pmed30 | 3270 | 64.40% | 0.0621 |
| pmed31 | 14251 | 41.29% | 0.0001 |
| pmed32 | 13003 | 39.86% | 0.0001 |
| pmed33 | 6814 | 44.98% | 0.0151 |
| pmed34 | 4380 | 45.37% | 0.0311 |
| pmed35 | 14631 | 40.68% | 0.0001 |
| pmed36 | 12766 | 28.51% | 0.0001 |
| pmed37 | 7056 | 39.53% | 0.0001 |
| pmed38 | 15244 | 37.83% | 0.0001 |
| pmed39 | 12230 | 29.79% | 0.0001 |
| pmed40 | 7398 | 44.27% | 0.0201 |

Πίνακας Α΄.3: Rrandom plus greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6544 | 12.46% | 0.0001 |
| pmed2 | 5484 | 33.98% | 0.0001 |
| pmed3 | 5204 | 22.45% | 0.0151 |
| pmed4 | 3133 | 3.26% | 0.0201 |
| pmed5 | 1420 | 4.80% | 0.0161 |
| pmed6 | 8416 | 7.57% | 0.0001 |
| pmed7 | 7440 | 32.13% | 0.0001 |
| pmed8 | 4847 | 9.04% | 0.0011 |
| pmed9 | 3140 | 14.85% | 0.0471 |
| pmed10 | 1393 | 11.00% | 0.1931 |
| pmed11 | 8058 | 4.70% | 0.0001 |
| pmed12 | 7143 | 7.67% | 0.0001 |
| pmed13 | 5554 | 26.98% | 0.0401 |
| pmed14 | 3215 | 8.32% | 0.3881 |
| pmed15 | 1952 | 12.90% | 0.8141 |
| pmed16 | 9057 | 10.97% | 0.0151 |
| pmed17 | 7107 | 1.54% | 0.0321 |
| pmed18 | 5913 | 22.96% | 0.1401 |
| pmed19 | 3923 | 37.89% | 0.2551 |
| pmed20 | 3114 | 74.06% | 0.1391 |
| pmed21 | 11098 | 21.45% | 0.0001 |
| pmed22 | 8836 | 3.00% | 0.0471 |
| pmed23 | 5854 | 26.74% | 0.1731 |
| pmed24 | 3920 | 32.39% | 0.6721 |
| pmed25 | 2347 | 28.39% | 3.2571 |
| pmed26 | 11383 | 14.78% | 0.0271 |
| pmed27 | 8995 | 8.28% | 0.0321 |
| pmed28 | 5186 | 15.30% | 0.9371 |
| pmed29 | 3689 | 21.63% | 3.3121 |
| pmed30 | 2069 | 4.02% | 16.2881 |
| pmed31 | 11617 | 15.18% | 0.0201 |
| pmed32 | 9629 | 3.57% | 0.0931 |
| pmed33 | 5948 | 26.55% | 1.0541 |
| pmed34 | 3404 | 12.98% | 9.4671 |
| pmed35 | 11413 | 9.74% | 0.0161 |
| pmed36 | 10329 | 3.98% | 0.1231 |
| pmed37 | 5275 | 4.31% | 5.2371 |
| pmed38 | 11549 | 4.42% | 0.0371 |
| pmed39 | 10337 | 9.70% | 0.1051 |
| pmed40 | 5568 | 8.58% | 7.1631 |

Πίνακας Α΄.4: Randomized greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6181 | 6.22% | 0.0001 |
| pmed2 | 4590 | 12.14% | 0.0001 |
| pmed3 | 4918 | 15.72% | 0.0001 |
| pmed4 | 3374 | 11.21% | 0.0001 |
| pmed5 | 1700 | 25.46% | 0.0171 |
| pmed6 | 8545 | 9.22% | 0.0161 |
| pmed7 | 6465 | 14.81% | 0.0171 |
| pmed8 | 5222 | 17.48% | 0.0181 |
| pmed9 | 3228 | 18.07% | 0.0901 |
| pmed10 | 1531 | 21.99% | 0.2501 |
| pmed11 | 8316 | 8.06% | 0.0001 |
| pmed12 | 7242 | 9.16% | 0.0321 |
| pmed13 | 5006 | 14.45% | 0.1251 |
| pmed14 | 3485 | 17.42% | 0.4531 |
| pmed15 | 2005 | 15.96% | 1.0931 |
| pmed16 | 9346 | 14.51% | 0.0161 |
| pmed17 | 8158 | 16.56% | 0.0341 |
| pmed18 | 5377 | 11.81% | 0.4211 |
| pmed19 | 3319 | 16.66% | 1.4021 |
| pmed20 | 2173 | 21.46% | 3.4311 |
| pmed21 | 10527 | 15.20% | 0.0321 |
| pmed22 | 9595 | 11.84% | 0.0631 |
| pmed23 | 5276 | 14.22% | 1.0001 |
| pmed24 | 3411 | 15.20% | 3.3341 |
| pmed25 | 2162 | 18.27% | 8.2121 |
| pmed26 | 11533 | 16.30% | 0.0321 |
| pmed27 | 9476 | 14.07% | 0.1241 |
| pmed28 | 5138 | 14.23% | 2.0311 |
| pmed29 | 3556 | 17.24% | 6.7731 |
| pmed30 | 2356 | 18.45% | 17.3191 |
| pmed31 | 11959 | 18.57% | 0.0631 |
| pmed32 | 10921 | 17.47% | 0.1881 |
| pmed33 | 5329 | 13.38% | 3.7361 |
| pmed34 | 3537 | 17.39% | 12.1261 |
| pmed35 | 12031 | 15.68% | 0.0941 |
| pmed36 | 11306 | 13.81% | 0.2191 |
| pmed37 | 5748 | 13.66% | 6.1701 |
| pmed38 | 12856 | 16.24% | 0.1281 |
| pmed39 | 10721 | 13.77% | 0.3501 |
| pmed40 | 5817 | 13.44% | 9.9371 |

Πίνακας Α΄.5: Proportional greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 11715 | 101.32% | 0.0001 |
| pmed2 | 7269 | 77.60% | 0.0001 |
| pmed3 | 7054 | 65.98% | 0.0001 |
| pmed4 | 4813 | 58.64% | 0.0001 |
| pmed5 | 2012 | 48.49% | 0.0001 |
| pmed6 | 14526 | 85.66% | 0.0001 |
| pmed7 | 10618 | 88.56% | 0.0151 |
| pmed8 | 7611 | 71.23% | 0.0311 |
| pmed9 | 3824 | 39.87% | 0.0931 |
| pmed10 | 1935 | 54.18% | 0.2251 |
| pmed11 | 16384 | 112.89% | 0.0001 |
| pmed12 | 14184 | 113.81% | 0.0311 |
| pmed13 | 7295 | 66.78% | 0.1381 |
| pmed14 | 4325 | 45.72% | 0.4851 |
| pmed15 | 2438 | 41.01% | 1.1551 |
| pmed16 | 18840 | 130.83% | 0.0161 |
| pmed17 | 15119 | 116.02% | 0.0461 |
| pmed18 | 7628 | 58.62% | 0.4601 |
| pmed19 | 4038 | 41.93% | 1.4491 |
| pmed20 | 2644 | 47.79% | 3.5611 |
| pmed21 | 19787 | 116.54% | 0.0171 |
| pmed22 | 18665 | 117.57% | 0.0641 |
| pmed23 | 7701 | 66.72% | 1.0311 |
| pmed24 | 4171 | 40.86% | 3.4841 |
| pmed25 | 2701 | 47.76% | 8.8561 |
| pmed26 | 23160 | 133.54% | 0.0031 |
| pmed27 | 19025 | 129.02% | 0.0931 |
| pmed28 | 7785 | 73.08% | 2.1551 |
| pmed29 | 4328 | 42.70% | 7.1251 |
| pmed30 | 2852 | 43.39% | 18.0411 |
| pmed31 | 22085 | 118.97% | 0.0301 |
| pmed32 | 20762 | 123.32% | 0.1411 |
| pmed33 | 7874 | 67.53% | 3.7431 |
| pmed34 | 4562 | 51.41% | 13.7861 |
| pmed35 | 25954 | 149.56% | 0.0471 |
| pmed36 | 21984 | 121.30% | 0.1841 |
| pmed37 | 9259 | 83.09% | 6.4681 |
| pmed38 | 25421 | 129.85% | 0.0311 |
| pmed39 | 21266 | 125.68% | 0.2331 |
| pmed40 | 8926 | 74.06% | 10.0941 |

Πίνακας Α΄.6: Proportional worst

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 7997 | 37.43% | 0.0001 |
| pmed2 | 7026 | 71.66% | 0.0001 |
| pmed3 | 7530 | 77.18% | 0.0001 |
| pmed4 | 5114 | 68.56% | 0.0001 |
| pmed5 | 2196 | 62.07% | 0.0001 |
| pmed6 | 11815 | 51.01% | 0.0001 |
| pmed7 | 7661 | 36.05% | 0.0001 |
| pmed8 | 6507 | 46.39% | 0.0001 |
| pmed9 | 4247 | 55.34% | 0.0001 |
| pmed10 | 2097 | 67.09% | 0.0041 |
| pmed11 | 13660 | 77.49% | 0.0001 |
| pmed12 | 12193 | 83.80% | 0.0001 |
| pmed13 | 7453 | 70.39% | 0.0001 |
| pmed14 | 4911 | 65.46% | 0.0141 |
| pmed15 | 2747 | 58.88% | 0.0311 |
| pmed16 | 13031 | 59.65% | 0.0001 |
| pmed17 | 11860 | 69.45% | 0.0001 |
| pmed18 | 8423 | 75.15% | 0.0001 |
| pmed19 | 4884 | 71.67% | 0.0151 |
| pmed20 | 2725 | 52.32% | 0.0311 |
| pmed21 | 14167 | 55.03% | 0.0001 |
| pmed22 | 12843 | 49.70% | 0.0001 |
| pmed23 | 7491 | 62.18% | 0.0171 |
| pmed24 | 4598 | 55.29% | 0.0471 |
| pmed25 | 2823 | 54.43% | 0.0761 |
| pmed26 | 13123 | 32.33% | 0.0001 |
| pmed27 | 12822 | 54.35% | 0.0001 |
| pmed28 | 7644 | 69.94% | 0.0141 |
| pmed29 | 4924 | 62.35% | 0.0631 |
| pmed30 | 2799 | 40.72% | 0.1251 |
| pmed31 | 16372 | 62.32% | 0.0001 |
| pmed32 | 14096 | 51.62% | 0.0001 |
| pmed33 | 7893 | 67.94% | 0.0311 |
| pmed34 | 4823 | 60.07% | 0.0991 |
| pmed35 | 13958 | 34.21% | 0.0001 |
| pmed36 | 13569 | 36.59% | 0.0041 |
| pmed37 | 9000 | 77.97% | 0.0621 |
| pmed38 | 17947 | 62.27% | 0.0311 |
| pmed39 | 15422 | 63.66% | 0.0001 |
| pmed40 | 7869 | 53.45% | 0.0941 |

Πίνακας Α΄.7: Sample greedy

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 7549 | 29.73% | 0.0001 |
| pmed2 | 5390 | 31.69% | 0.0001 |
| pmed3 | 5278 | 24.19% | 0.0001 |
| pmed4 | 3847 | 26.80% | 0.0001 |
| pmed5 | 1879 | 38.67% | 0.0001 |
| pmed6 | 8849 | 13.10% | 0.0001 |
| pmed7 | 7014 | 24.56% | 0.0001 |
| pmed8 | 5635 | 26.77% | 0.0001 |
| pmed9 | 3818 | 39.65% | 0.0161 |
| pmed10 | 2168 | 72.75% | 0.0001 |
| pmed11 | 9484 | 23.23% | 0.0001 |
| pmed12 | 7854 | 18.39% | 0.0001 |
| pmed13 | 5416 | 23.82% | 0.0001 |
| pmed14 | 3893 | 31.17% | 0.0001 |
| pmed15 | 2535 | 46.62% | 0.0001 |
| pmed16 | 10309 | 26.30% | 0.0001 |
| pmed17 | 8480 | 21.16% | 0.0001 |
| pmed18 | 5824 | 21.11% | 0.0001 |
| pmed19 | 3845 | 35.15% | 0.0151 |
| pmed20 | 2622 | 46.56% | 0.0311 |
| pmed21 | 11391 | 24.66% | 0.0001 |
| pmed22 | 9692 | 12.97% | 0.0001 |
| pmed23 | 5725 | 23.94% | 0.0141 |
| pmed24 | 3925 | 32.56% | 0.0161 |
| pmed25 | 3077 | 68.33% | 0.0341 |
| pmed26 | 11415 | 15.11% | 0.0001 |
| pmed27 | 9560 | 15.08% | 0.0001 |
| pmed28 | 5447 | 21.10% | 0.0191 |
| pmed29 | 4109 | 35.48% | 0.0311 |
| pmed30 | 2829 | 42.23% | 0.0781 |
| pmed31 | 11097 | 10.02% | 0.0001 |
| pmed32 | 11014 | 18.47% | 0.0001 |
| pmed33 | 5738 | 22.09% | 0.0161 |
| pmed34 | 4240 | 40.72% | 0.0321 |
| pmed35 | 11179 | 7.49% | 0.0001 |
| pmed36 | 11475 | 15.51% | 0.0001 |
| pmed37 | 6238 | 23.35% | 0.0311 |
| pmed38 | 13427 | 21.40% | 0.0001 |
| pmed39 | 10665 | 13.18% | 0.0001 |
| pmed40 | 6358 | 23.99% | 0.0341 |

Πίνακας Α΄.8: Drop approach

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 10249 | 76.13% | 0.0201 |
| pmed2 | 7222 | 76.45% | 0.0001 |
| pmed3 | 6825 | 60.59% | 0.0001 |
| pmed4 | 4806 | 58.40% | 0.0091 |
| pmed5 | 2340 | 72.69% | 0.0001 |
| pmed6 | 15833 | 102.36% | 0.0151 |
| pmed7 | 11160 | 98.19% | 0.0001 |
| pmed8 | 8406 | 89.11% | 0.0161 |
| pmed9 | 4524 | 65.47% | 0.0001 |
| pmed10 | 2146 | 71.00% | 0.0001 |
| pmed11 | 12721 | 65.29% | 0.0461 |
| pmed12 | 12426 | 87.31% | 0.0321 |
| pmed13 | 8193 | 87.31% | 0.0311 |
| pmed14 | 5324 | 79.38% | 0.0321 |
| pmed15 | 2512 | 45.29% | 0.0311 |
| pmed16 | 17951 | 119.93% | 0.0621 |
| pmed17 | 13117 | 87.41% | 0.0791 |
| pmed18 | 8915 | 85.38% | 0.0811 |
| pmed19 | 4818 | 69.35% | 0.0631 |
| pmed20 | 2742 | 53.27% | 0.0631 |
| pmed21 | 16349 | 78.91% | 0.1431 |
| pmed22 | 15931 | 85.70% | 0.1411 |
| pmed23 | 8905 | 92.79% | 0.1411 |
| pmed24 | 4872 | 64.54% | 0.1401 |
| pmed25 | 2718 | 48.69% | 0.1411 |
| pmed26 | 16813 | 69.54% | 0.2721 |
| pmed27 | 15147 | 82.34% | 0.2611 |
| pmed28 | 8233 | 83.04% | 0.2471 |
| pmed29 | 4945 | 63.04% | 0.2511 |
| pmed30 | 2857 | 43.64% | 0.2361 |
| pmed31 | 16922 | 67.78% | 0.4081 |
| pmed32 | 20738 | 123.06% | 0.4071 |
| pmed33 | 8463 | 80.06% | 0.4071 |
| pmed34 | 5221 | 73.28% | 0.3771 |
| pmed35 | 16245 | 56.20% | 0.6101 |
| pmed36 | 19559 | 96.89% | 0.5931 |
| pmed37 | 10152 | 100.75% | 0.5781 |
| pmed38 | 18837 | 70.32% | 0.8611 |
| pmed39 | 19496 | 106.90% | 0.8731 |
| pmed40 | 9793 | 90.97% | 0.8571 |

Πίνακας Α΄.9: Teitz and Bart (Myopic input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5819 | 0,00% | 0,0001 |
| pmed2 | 4105 | 0,29% | 0,0001 |
| pmed3 | 4254 | 0,09% | 0,0001 |
| pmed4 | 3046 | 0,40% | 0,0021 |
| pmed5 | 1355 | 0,00% | 0,0311 |
| pmed6 | 7884 | 0,77% | 0,0001 |
| pmed7 | 5645 | 0,25% | 0,0241 |
| pmed8 | 4457 | 0,27% | 0,0471 |
| pmed9 | 2753 | 0,69% | 0,1741 |
| pmed10 | 1272 | 1,35% | 0,4031 |
| pmed11 | 7696 | 0,00% | 0,0001 |
| pmed12 | 6634 | 0,00% | 0,0151 |
| pmed13 | 4380 | 0,14% | 0,2191 |
| pmed14 | 2982 | 0,47% | 0,7501 |
| pmed15 | 1738 | 0,52% | 1,8641 |
| pmed16 | 8165 | 0,04% | 0,0161 |
| pmed17 | 6999 | 0,00% | 0,0471 |
| pmed18 | 4812 | 0,06% | 0,6331 |
| pmed19 | 2861 | 0,56% | 2,2191 |
| pmed20 | 1806 | 0,95% | 5,8601 |
| pmed21 | 9138 | 0,00% | 0,0341 |
| pmed22 | 8669 | 1,05% | 0,0771 |
| pmed23 | 4624 | 0,11% | 1,4841 |
| pmed24 | 2965 | 0,14% | 5,3741 |
| pmed25 | 1845 | 0,93% | 14,8621 |
| pmed26 | 9917 | 0,00% | 0,0321 |
| pmed27 | 8319 | 0,14% | 0,0991 |
| pmed28 | 4515 | 0,38% | 3,4641 |
| pmed29 | 3046 | 0,43% | 11,2691 |
| pmed30 | 2009 | 1,01% | 30,8151 |
| pmed31 | 10086 | 0,00% | 0,0411 |
| pmed32 | 9301 | 0,04% | 0,1571 |
| pmed33 | 4706 | 0,13% | 5,4361 |
| pmed34 | 3034 | 0,70% | 21,2641 |
| pmed35 | 10400 | 0,00% | 0,0621 |
| pmed36 | 9934 | 0,00% | 0,2201 |
| pmed37 | 5065 | 0,16% | 9,3361 |
| pmed38 | 11135 | 0,68% | 0,0781 |
| pmed39 | 9436 | 0,14% | 0,2821 |
| pmed40 | 5153 | 0,49% | 15,4521 |

Πίνακας Α΄.10: Teitz and Bart (Random input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5891 | 1,24% | 0,0001 |
| pmed2 | 4093 | 0,00% | 0,0001 |
| pmed3 | 4281 | 0,73% | 0,0001 |
| pmed4 | 3053 | 0,63% | 0,0151 |
| pmed5 | 1361 | 0,44% | 0,0351 |
| pmed6 | 7852 | 0,36% | 0,0001 |
| pmed7 | 5651 | 0,36% | 0,0061 |
| pmed8 | 4474 | 0,65% | 0,0311 |
| pmed9 | 2774 | 1,46% | 0,1571 |
| pmed10 | 1274 | 1,51% | 0,4011 |
| pmed11 | 7696 | 0,00% | 0,0001 |
| pmed12 | 6703 | 1,04% | 0,0311 |
| pmed13 | 4399 | 0,57% | 0,1871 |
| pmed14 | 3017 | 1,65% | 0,7501 |
| pmed15 | 1757 | 1,62% | 1,8901 |
| pmed16 | 8249 | 1,07% | 0,0321 |
| pmed17 | 7010 | 0,16% | 0,0471 |
| pmed18 | 4879 | 1,46% | 0,6091 |
| pmed19 | 2905 | 2,11% | 2,2971 |
| pmed20 | 1824 | 1,96% | 6,0941 |
| pmed21 | 9138 | 0,00% | 0,0161 |
| pmed22 | 8725 | 1,70% | 0,0781 |
| pmed23 | 4695 | 1,65% | 1,4381 |
| pmed24 | 2997 | 1,22% | 5,3751 |
| pmed25 | 1861 | 1,81% | 15,1801 |
| pmed26 | 9924 | 0,07% | 0,0341 |
| pmed27 | 8316 | 0,11% | 0,1091 |
| pmed28 | 4540 | 0,93% | 3,4111 |
| pmed29 | 3064 | 1,02% | 11,9331 |
| pmed30 | 2041 | 2,61% | 31,4031 |
| pmed31 | 10086 | 0,00% | 0,0351 |
| pmed32 | 9326 | 0,31% | 0,1551 |
| pmed33 | 4753 | 1,13% | 5,4861 |
| pmed34 | 3032 | 0,63% | 22,1731 |
| pmed35 | 10406 | 0,06% | 0,0631 |
| pmed36 | 10005 | 0,71% | 0,2181 |
| pmed37 | 5088 | 0,61% | 9,6161 |
| pmed38 | 11086 | 0,24% | 0,0771 |
| pmed39 | 9423 | 0,00% | 0,2561 |
| pmed40 | 5212 | 1,64% | 15,6611 |

Πίνακας Α΄.11: Teitz and Bart (Random plus greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5819 | 0,00% | 0,0001 |
| pmed2 | 4093 | 0,00% | 0,0161 |
| pmed3 | 4268 | 0,42% | 0,0001 |
| pmed4 | 3049 | 0,49% | 0,0001 |
| pmed5 | 1362 | 0,52% | 0,0121 |
| pmed6 | 7871 | 0,60% | 0,0001 |
| pmed7 | 5631 | 0,00% | 0,0191 |
| pmed8 | 4460 | 0,34% | 0,0471 |
| pmed9 | 2741 | 0,26% | 0,1541 |
| pmed10 | 1262 | 0,56% | 0,4051 |
| pmed11 | 7696 | 0,00% | 0,0161 |
| pmed12 | 6634 | 0,00% | 0,0321 |
| pmed13 | 4387 | 0,30% | 0,2031 |
| pmed14 | 2996 | 0,94% | 0,7551 |
| pmed15 | 1753 | 1,39% | 1,9721 |
| pmed16 | 8165 | 0,04% | 0,0151 |
| pmed17 | 7010 | 0,16% | 0,0471 |
| pmed18 | 4834 | 0,52% | 0,6761 |
| pmed19 | 2893 | 1,69% | 2,2651 |
| pmed20 | 1815 | 1,45% | 6,0301 |
| pmed21 | 9138 | 0,00% | 0,0321 |
| pmed22 | 8579 | 0,00% | 0,0781 |
| pmed23 | 4656 | 0,80% | 1,5161 |
| pmed24 | 3001 | 1,35% | 5,4391 |
| pmed25 | 1871 | 2,35% | 15,3131 |
| pmed26 | 9929 | 0,12% | 0,0211 |
| pmed27 | 8316 | 0,11% | 0,1251 |
| pmed28 | 4534 | 0,80% | 3,6751 |
| pmed29 | 3074 | 1,35% | 11,7721 |
| pmed30 | 2007 | 0,90% | 31,1421 |
| pmed31 | 10086 | 0,00% | 0,0611 |
| pmed32 | 9306 | 0,10% | 0,1721 |
| pmed33 | 4737 | 0,79% | 5,6391 |
| pmed34 | 3039 | 0,86% | 22,5301 |
| pmed35 | 10400 | 0,00% | 0,0641 |
| pmed36 | 9968 | 0,34% | 0,2341 |
| pmed37 | 5069 | 0,24% | 9,6721 |
| pmed38 | 11060 | 0,00% | 0,0781 |
| pmed39 | 9485 | 0,66% | 0,3221 |
| pmed40 | 5154 | 0,51% | 16,0441 |

Πίνακας Α΄.12: Teitz and Bart (Randomized greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5821 | 0,03% | 0,0001 |
| pmed2 | 4102 | 0,22% | 0,0001 |
| pmed3 | 4253 | 0,07% | 0,0001 |
| pmed4 | 3053 | 0,63% | 0,0221 |
| pmed5 | 1371 | 1,18% | 0,0311 |
| pmed6 | 7979 | 1,98% | 0,0151 |
| pmed7 | 5650 | 0,34% | 0,0001 |
| pmed8 | 4480 | 0,79% | 0,0461 |
| pmed9 | 2751 | 0,62% | 0,1511 |
| pmed10 | 1273 | 1,43% | 0,3891 |
| pmed11 | 7724 | 0,36% | 0,0001 |
| pmed12 | 6677 | 0,65% | 0,0311 |
| pmed13 | 4414 | 0,91% | 0,2211 |
| pmed14 | 2985 | 0,57% | 0,7361 |
| pmed15 | 1747 | 1,04% | 1,8841 |
| pmed16 | 8167 | 0,06% | 0,0151 |
| pmed17 | 7033 | 0,49% | 0,0471 |
| pmed18 | 4851 | 0,87% | 0,6761 |
| pmed19 | 2876 | 1,09% | 2,3451 |
| pmed20 | 1810 | 1,17% | 6,0231 |
| pmed21 | 9138 | 0,00% | 0,0131 |
| pmed22 | 8613 | 0,40% | 0,0951 |
| pmed23 | 4637 | 0,39% | 1,5151 |
| pmed24 | 2990 | 0,98% | 5,4211 |
| pmed25 | 1873 | 2,46% | 14,8751 |
| pmed26 | 9994 | 0,78% | 0,0321 |
| pmed27 | 8359 | 0,63% | 0,1101 |
| pmed28 | 4522 | 0,53% | 3,5441 |
| pmed29 | 3073 | 1,32% | 11,6821 |
| pmed30 | 2022 | 1,66% | 31,0751 |
| pmed31 | 10086 | 0,00% | 0,0481 |
| pmed32 | 9360 | 0,68% | 0,1581 |
| pmed33 | 4728 | 0,60% | 5,6291 |
| pmed34 | 3031 | 0,60% | 21,5941 |
| pmed35 | 10401 | 0,01% | 0,0651 |
| pmed36 | 10069 | 1,36% | 0,2111 |
| pmed37 | 5086 | 0,57% | 9,5781 |
| pmed38 | 11107 | 0,42% | 0,0801 |
| pmed39 | 9467 | 0,47% | 0,2971 |
| pmed40 | 5190 | 1,21% | 15,7801 |

Πίνακας Α΄.13: Teitz and Bart (Proportional greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5891 | 1,24% | 0,0001 |
| pmed2 | 4093 | 0,00% | 0,0001 |
| pmed3 | 4261 | 0,26% | 0,0161 |
| pmed4 | 3058 | 0,79% | 0,0111 |
| pmed5 | 1360 | 0,37% | 0,0311 |
| pmed6 | 7965 | 1,80% | 0,0001 |
| pmed7 | 5669 | 0,67% | 0,0121 |
| pmed8 | 4510 | 1,46% | 0,0321 |
| pmed9 | 2773 | 1,43% | 0,1641 |
| pmed10 | 1277 | 1,75% | 0,4071 |
| pmed11 | 7721 | 0,32% | 0,0151 |
| pmed12 | 6651 | 0,26% | 0,0161 |
| pmed13 | 4428 | 1,23% | 0,1851 |
| pmed14 | 3018 | 1,68% | 0,7751 |
| pmed15 | 1748 | 1,10% | 1,9391 |
| pmed16 | 8232 | 0,86% | 0,0001 |
| pmed17 | 7037 | 0,54% | 0,0311 |
| pmed18 | 4926 | 2,43% | 0,6161 |
| pmed19 | 2886 | 1,44% | 2,2961 |
| pmed20 | 1807 | 1,01% | 5,8921 |
| pmed21 | 9138 | 0,00% | 0,0301 |
| pmed22 | 8676 | 1,13% | 0,0611 |
| pmed23 | 4643 | 0,52% | 1,4531 |
| pmed24 | 2994 | 1,11% | 5,5321 |
| pmed25 | 1860 | 1,75% | 15,6211 |
| pmed26 | 10093 | 1,77% | 0,0321 |
| pmed27 | 8364 | 0,69% | 0,0931 |
| pmed28 | 4540 | 0,93% | 3,4371 |
| pmed29 | 3065 | 1,06% | 11,4241 |
| pmed30 | 2026 | 1,86% | 31,5771 |
| pmed31 | 10086 | 0,00% | 0,0431 |
| pmed32 | 9331 | 0,37% | 0,1391 |
| pmed33 | 4761 | 1,30% | 5,3741 |
| pmed34 | 3043 | 1,00% | 21,6411 |
| pmed35 | 10406 | 0,06% | 0,0431 |
| pmed36 | 9954 | 0,20% | 0,1701 |
| pmed37 | 5106 | 0,97% | 9,1721 |
| pmed38 | 11153 | 0,84% | 0,0631 |
| pmed39 | 9451 | 0,30% | 0,2191 |
| pmed40 | 5196 | 1,33% | 15,2491 |

Πίνακας Α΄.14: Teitz and Bart (Proportional worst input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5891 | 1,24% | 0,0001 |
| pmed2 | 4136 | 1,05% | 0,0001 |
| pmed3 | 4354 | 2,45% | 0,0001 |
| pmed4 | 3038 | 0,13% | 0,0001 |
| pmed5 | 1362 | 0,52% | 0,0321 |
| pmed6 | 8027 | 2,59% | 0,0001 |
| pmed7 | 5646 | 0,27% | 0,0181 |
| pmed8 | 4459 | 0,31% | 0,0471 |
| pmed9 | 2768 | 1,24% | 0,1371 |
| pmed10 | 1276 | 1,67% | 0,3851 |
| pmed11 | 7721 | 0,32% | 0,0001 |
| pmed12 | 6664 | 0,45% | 0,0311 |
| pmed13 | 4419 | 1,03% | 0,2191 |
| pmed14 | 3028 | 2,02% | 0,7491 |
| pmed15 | 1754 | 1,45% | 1,8761 |
| pmed16 | 8167 | 0,06% | 0,0161 |
| pmed17 | 7019 | 0,29% | 0,0471 |
| pmed18 | 4874 | 1,35% | 0,6241 |
| pmed19 | 2923 | 2,74% | 2,2351 |
| pmed20 | 1814 | 1,40% | 5,8041 |
| pmed21 | 9138 | 0,00% | 0,0211 |
| pmed22 | 8667 | 1,03% | 0,0631 |
| pmed23 | 4645 | 0,56% | 1,4731 |
| pmed24 | 3025 | 2,16% | 5,4111 |
| pmed25 | 1875 | 2,57% | 14,7861 |
| pmed26 | 9961 | 0,44% | 0,0311 |
| pmed27 | 8364 | 0,69% | 0,0941 |
| pmed28 | 4569 | 1,58% | 3,4251 |
| pmed29 | 3111 | 2,57% | 11,2171 |
| pmed30 | 2052 | 3,17% | 31,0021 |
| pmed31 | 10086 | 0,00% | 0,0321 |
| pmed32 | 9328 | 0,33% | 0,1401 |
| pmed33 | 4807 | 2,28% | 5,4041 |
| pmed34 | 3069 | 1,86% | 21,2811 |
| pmed35 | 10405 | 0,05% | 0,0641 |
| pmed36 | 10028 | 0,95% | 0,2021 |
| pmed37 | 5118 | 1,21% | 9,1161 |
| pmed38 | 11153 | 0,84% | 0,0811 |
| pmed39 | 9473 | 0,53% | 0,2331 |
| pmed40 | 5197 | 1,35% | 15,2511 |

Πίνακας Α′.15: Teitz and Bart (Sample greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5821 | 0,03% | 0,0001 |
| pmed2 | 4117 | 0,59% | 0,0001 |
| pmed3 | 4290 | 0,94% | 0,0001 |
| pmed4 | 3082 | 1,58% | 0,0171 |
| pmed5 | 1379 | 1,77% | 0,0001 |
| pmed6 | 7950 | 1,61% | 0,0001 |
| pmed7 | 5650 | 0,34% | 0,0001 |
| pmed8 | 4465 | 0,45% | 0,0471 |
| pmed9 | 2812 | 2,85% | 0,1751 |
| pmed10 | 1278 | 1,83% | 0,4131 |
| pmed11 | 7696 | 0,00% | 0,0171 |
| pmed12 | 6649 | 0,23% | 0,0161 |
| pmed13 | 4480 | 2,42% | 0,2031 |
| pmed14 | 3025 | 1,92% | 0,7811 |
| pmed15 | 1751 | 1,27% | 1,9661 |
| pmed16 | 8167 | 0,06% | 0,0161 |
| pmed17 | 7014 | 0,21% | 0,0471 |
| pmed18 | 4890 | 1,68% | 0,6251 |
| pmed19 | 2882 | 1,30% | 2,2911 |
| pmed20 | 1814 | 1,40% | 6,0941 |
| pmed21 | 9408 | 2,95% | 0,0121 |
| pmed22 | 8671 | 1,07% | 0,0991 |
| pmed23 | 4646 | 0,58% | 1,5281 |
| pmed24 | 3001 | 1,35% | 5,4791 |
| pmed25 | 1879 | 2,79% | 15,4371 |
| pmed26 | 9924 | 0,07% | 0,0511 |
| pmed27 | 8354 | 0,57% | 0,1261 |
| pmed28 | 4542 | 0,98% | 3,5681 |
| pmed29 | 3084 | 1,68% | 11,4051 |
| pmed30 | 2023 | 1,71% | 32,1401 |
| pmed31 | 10086 | 0,00% | 0,0471 |
| pmed32 | 9359 | 0,67% | 0,1721 |
| pmed33 | 4737 | 0,79% | 5,4371 |
| pmed34 | 3043 | 1,00% | 22,0001 |
| pmed35 | 10484 | 0,81% | 0,0611 |
| pmed36 | 9942 | 0,08% | 0,2451 |
| pmed37 | 5105 | 0,95% | 9,7281 |
| pmed38 | 11157 | 0,88% | 0,0591 |
| pmed39 | 9429 | 0,06% | 0,2811 |
| pmed40 | 5190 | 1,21% | 15,5011 |

Πίνακας Α΄.16: Teitz and Bart (Drop approach input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5891 | 1,24% | 0,0001 |
| pmed2 | 4182 | 2,17% | 0,0001 |
| pmed3 | 4292 | 0,99% | 0,0001 |
| pmed4 | 3056 | 0,73% | 0,0001 |
| pmed5 | 1381 | 1,92% | 0,0311 |
| pmed6 | 8027 | 2,59% | 0,0001 |
| pmed7 | 5648 | 0,30% | 0,0301 |
| pmed8 | 4514 | 1,55% | 0,0311 |
| pmed9 | 2758 | 0,88% | 0,1381 |
| pmed10 | 1279 | 1,91% | 0,3911 |
| pmed11 | 7721 | 0,32% | 0,0001 |
| pmed12 | 6651 | 0,26% | 0,0311 |
| pmed13 | 4476 | 2,33% | 0,1881 |
| pmed14 | 3061 | 3,13% | 0,7351 |
| pmed15 | 1755 | 1,50% | 1,8761 |
| pmed16 | 8232 | 0,86% | 0,0151 |
| pmed17 | 7022 | 0,33% | 0,0461 |
| pmed18 | 4832 | 0,48% | 0,6141 |
| pmed19 | 2915 | 2,46% | 2,2101 |
| pmed20 | 1824 | 1,96% | 5,7651 |
| pmed21 | 9138 | 0,00% | 0,0181 |
| pmed22 | 8679 | 1,17% | 0,0661 |
| pmed23 | 4675 | 1,21% | 1,4381 |
| pmed24 | 3044 | 2,80% | 5,3441 |
| pmed25 | 1867 | 2,13% | 14,7531 |
| pmed26 | 10093 | 1,77% | 0,0271 |
| pmed27 | 8364 | 0,69% | 0,0881 |
| pmed28 | 4577 | 1,76% | 3,3551 |
| pmed29 | 3092 | 1,95% | 11,2191 |
| pmed30 | 2049 | 3,02% | 31,0631 |
| pmed31 | 10086 | 0,00% | 0,0471 |
| pmed32 | 9331 | 0,37% | 0,1251 |
| pmed33 | 4821 | 2,57% | 5,3151 |
| pmed34 | 3081 | 2,26% | 21,3431 |
| pmed35 | 10406 | 0,06% | 0,0501 |
| pmed36 | 9976 | 0,42% | 0,1571 |
| pmed37 | 5119 | 1,23% | 9,1031 |
| pmed38 | 11153 | 0,84% | 0,0641 |
| pmed39 | 9473 | 0,53% | 0,2201 |
| pmed40 | 5201 | 1,42% | 15,2081 |

Πίνακας Α′.17: Neighborhood Search (Myopic input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5891 | 1,24% | 0,0001 |
| pmed2 | 4105 | 0,29% | 0,0001 |
| pmed3 | 4336 | 2,02% | 0,0001 |
| pmed4 | 3088 | 1,78% | 0,0001 |
| pmed5 | 1357 | 0,15% | 0,0001 |
| pmed6 | 8027 | 2,59% | 0,0001 |
| pmed7 | 5645 | 0,25% | 0,0001 |
| pmed8 | 4457 | 0,27% | 0,0001 |
| pmed9 | 2798 | 2,34% | 0,0001 |
| pmed10 | 1288 | 2,63% | 0,0001 |
| pmed11 | 7714 | 0,23% | 0,0001 |
| pmed12 | 6651 | 0,26% | 0,0001 |
| pmed13 | 4443 | 1,58% | 0,0001 |
| pmed14 | 2997 | 0,98% | 0,0001 |
| pmed15 | 1748 | 1,10% | 0,0001 |
| pmed16 | 8232 | 0,86% | 0,0001 |
| pmed17 | 7010 | 0,16% | 0,0001 |
| pmed18 | 4824 | 0,31% | 0,0001 |
| pmed19 | 2867 | 0,77% | 0,0001 |
| pmed20 | 1836 | 2,63% | 0,0001 |
| pmed21 | 9138 | 0,00% | 0,0001 |
| pmed22 | 8670 | 1,06% | 0,0001 |
| pmed23 | 4659 | 0,87% | 0,0001 |
| pmed24 | 2979 | 0,61% | 0,0001 |
| pmed25 | 1862 | 1,86% | 0,0001 |
| pmed26 | 10093 | 1,77% | 0,0001 |
| pmed27 | 8360 | 0,64% | 0,0001 |
| pmed28 | 4551 | 1,18% | 0,0001 |
| pmed29 | 3064 | 1,02% | 0,0001 |
| pmed30 | 2018 | 1,46% | 0,0001 |
| pmed31 | 10086 | 0,00% | 0,0001 |
| pmed32 | 9318 | 0,23% | 0,0001 |
| pmed33 | 4757 | 1,21% | 0,0001 |
| pmed34 | 3061 | 1,59% | 0,0161 |
| pmed35 | 10406 | 0,06% | 0,0001 |
| pmed36 | 9947 | 0,13% | 0,0151 |
| pmed37 | 5090 | 0,65% | 0,0001 |
| pmed38 | 11139 | 0,71% | 0,0001 |
| pmed39 | 9451 | 0,30% | 0,0001 |
| pmed40 | 5181 | 1,03% | 0,0151 |

Πίνακας Α΄.18: Neighborhood Search (Random input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6783 | 16,57% | 0,0001 |
| pmed2 | 5203 | 27,12% | 0,0001 |
| pmed3 | 5208 | 22,54% | 0,0001 |
| pmed4 | 3924 | 29,33% | 0,0001 |
| pmed5 | 2103 | 55,20% | 0,0001 |
| pmed6 | 9645 | 23,27% | 0,0001 |
| pmed7 | 6349 | 12,75% | 0,0001 |
| pmed8 | 5656 | 27,24% | 0,0001 |
| pmed9 | 3443 | 25,93% | 0,0001 |
| pmed10 | 1725 | 37,45% | 0,0001 |
| pmed11 | 8810 | 14,48% | 0,0001 |
| pmed12 | 8139 | 22,69% | 0,0001 |
| pmed13 | 5563 | 27,18% | 0,0001 |
| pmed14 | 3799 | 28,00% | 0,0001 |
| pmed15 | 2450 | 41,70% | 0,0001 |
| pmed16 | 8881 | 8,81% | 0,0001 |
| pmed17 | 7971 | 13,89% | 0,0001 |
| pmed18 | 5826 | 21,15% | 0,0001 |
| pmed19 | 3700 | 30,05% | 0,0001 |
| pmed20 | 2756 | 54,05% | 0,0001 |
| pmed21 | 10341 | 13,16% | 0,0001 |
| pmed22 | 9936 | 15,82% | 0,0001 |
| pmed23 | 5668 | 22,71% | 0,0001 |
| pmed24 | 3659 | 23,57% | 0,0001 |
| pmed25 | 2601 | 42,29% | 0,0001 |
| pmed26 | 11120 | 12,13% | 0,0001 |
| pmed27 | 9674 | 16,46% | 0,0001 |
| pmed28 | 5498 | 22,23% | 0,0001 |
| pmed29 | 3835 | 26,44% | 0,0001 |
| pmed30 | 2807 | 41,13% | 0,0001 |
| pmed31 | 10701 | 6,10% | 0,0001 |
| pmed32 | 10628 | 14,32% | 0,0001 |
| pmed33 | 5785 | 23,09% | 0,0001 |
| pmed34 | 3855 | 27,95% | 0,0001 |
| pmed35 | 11203 | 7,72% | 0,0001 |
| pmed36 | 11574 | 16,51% | 0,0001 |
| pmed37 | 6496 | 28,46% | 0,0091 |
| pmed38 | 11491 | 3,90% | 0,0001 |
| pmed39 | 11479 | 21,82% | 0,0001 |
| pmed40 | 6496 | 26,68% | 0,0001 |

Πίνακας Α΄.19: Neighborhood Search (Random plus greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6384 | 9,71% | 0,0001 |
| pmed2 | 5166 | 26,22% | 0,0001 |
| pmed3 | 4673 | 9,95% | 0,0001 |
| pmed4 | 3088 | 1,78% | 0,0001 |
| pmed5 | 1378 | 1,70% | 0,0001 |
| pmed6 | 8326 | 6,42% | 0,0001 |
| pmed7 | 6803 | 20,81% | 0,0001 |
| pmed8 | 4655 | 4,72% | 0,0001 |
| pmed9 | 3033 | 10,94% | 0,0001 |
| pmed10 | 1372 | 9,32% | 0,0001 |
| pmed11 | 8046 | 4,55% | 0,0001 |
| pmed12 | 7116 | 7,27% | 0,0001 |
| pmed13 | 5251 | 20,05% | 0,0001 |
| pmed14 | 3152 | 6,20% | 0,0001 |
| pmed15 | 1892 | 9,43% | 0,0001 |
| pmed16 | 8892 | 8,94% | 0,0001 |
| pmed17 | 7082 | 1,19% | 0,0001 |
| pmed18 | 5647 | 17,43% | 0,0001 |
| pmed19 | 3466 | 21,83% | 0,0001 |
| pmed20 | 2662 | 48,80% | 0,0001 |
| pmed21 | 10224 | 11,88% | 0,0131 |
| pmed22 | 8777 | 2,31% | 0,0001 |
| pmed23 | 5284 | 14,40% | 0,0001 |
| pmed24 | 3551 | 19,93% | 0,0001 |
| pmed25 | 2192 | 19,91% | 0,0001 |
| pmed26 | 11054 | 11,47% | 0,0001 |
| pmed27 | 8765 | 5,51% | 0,0001 |
| pmed28 | 5014 | 11,47% | 0,0001 |
| pmed29 | 3472 | 14,47% | 0,0001 |
| pmed30 | 2038 | 2,46% | 0,0001 |
| pmed31 | 11129 | 10,34% | 0,0001 |
| pmed32 | 9573 | 2,97% | 0,0001 |
| pmed33 | 5522 | 17,49% | 0,0001 |
| pmed34 | 3300 | 9,53% | 0,0001 |
| pmed35 | 11413 | 9,74% | 0,0001 |
| pmed36 | 10160 | 2,28% | 0,0001 |
| pmed37 | 5214 | 3,10% | 0,0001 |
| pmed38 | 11486 | 3,85% | 0,0001 |
| pmed39 | 9870 | 4,74% | 0,0001 |
| pmed40 | 5439 | 6,06% | 0,0001 |

Πίνακας Α΄.20: Neighborhood Search (Randomized greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 5901 | 1,41% | 0,0001 |
| pmed2 | 4204 | 2,71% | 0,0001 |
| pmed3 | 4583 | 7,84% | 0,0001 |
| pmed4 | 3238 | 6,72% | 0,0001 |
| pmed5 | 1524 | 12,47% | 0,0001 |
| pmed6 | 8270 | 5,70% | 0,0001 |
| pmed7 | 6347 | 12,72% | 0,0001 |
| pmed8 | 4906 | 10,37% | 0,0001 |
| pmed9 | 3121 | 14,16% | 0,0001 |
| pmed10 | 1429 | 13,86% | 0,0001 |
| pmed11 | 7913 | 2,82% | 0,0001 |
| pmed12 | 6924 | 4,37% | 0,0001 |
| pmed13 | 4757 | 8,76% | 0,0001 |
| pmed14 | 3278 | 10,44% | 0,0001 |
| pmed15 | 1934 | 11,86% | 0,0001 |
| pmed16 | 8441 | 3,42% | 0,0001 |
| pmed17 | 7817 | 11,69% | 0,0001 |
| pmed18 | 5106 | 6,18% | 0,0001 |
| pmed19 | 3069 | 7,87% | 0,0001 |
| pmed20 | 1961 | 9,61% | 0,0001 |
| pmed21 | 9714 | 6,30% | 0,0001 |
| pmed22 | 9115 | 6,25% | 0,0071 |
| pmed23 | 4960 | 7,38% | 0,0001 |
| pmed24 | 3231 | 9,12% | 0,0001 |
| pmed25 | 2042 | 11,71% | 0,0001 |
| pmed26 | 10761 | 8,51% | 0,0001 |
| pmed27 | 8867 | 6,74% | 0,0001 |
| pmed28 | 4910 | 9,16% | 0,0001 |
| pmed29 | 3319 | 9,43% | 0,0001 |
| pmed30 | 2237 | 12,47% | 0,0001 |
| pmed31 | 10686 | 5,95% | 0,0001 |
| pmed32 | 10067 | 8,28% | 0,0001 |
| pmed33 | 4970 | 5,74% | 0,0001 |
| pmed34 | 3270 | 8,53% | 0,0001 |
| pmed35 | 11226 | 7,94% | 0,0001 |
| pmed36 | 10664 | 7,35% | 0,0001 |
| pmed37 | 5412 | 7,02% | 0,0001 |
| pmed38 | 11378 | 2,88% | 0,0201 |
| pmed39 | 9831 | 4,33% | 0,0001 |
| pmed40 | 5572 | 8,66% | 0,0001 |

Πίνακας Α΄.21: Neighborhood Search (Proportional greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6721 | 15,50% | 0,0001 |
| pmed2 | 4656 | 13,76% | 0,0001 |
| pmed3 | 4800 | 12,94% | 0,0001 |
| pmed4 | 3577 | 17,90% | 0,0001 |
| pmed5 | 1917 | 41,48% | 0,0001 |
| pmed6 | 8447 | 7,96% | 0,0001 |
| pmed7 | 6721 | 19,36% | 0,0001 |
| pmed8 | 5189 | 16,74% | 0,0001 |
| pmed9 | 3252 | 18,95% | 0,0001 |
| pmed10 | 1671 | 33,15% | 0,0001 |
| pmed11 | 8271 | 7,47% | 0,0001 |
| pmed12 | 8229 | 24,04% | 0,0001 |
| pmed13 | 5228 | 19,52% | 0,0001 |
| pmed14 | 3721 | 25,37% | 0,0001 |
| pmed15 | 2134 | 23,42% | 0,0001 |
| pmed16 | 9076 | 11,20% | 0,0001 |
| pmed17 | 7866 | 12,39% | 0,0001 |
| pmed18 | 5959 | 23,91% | 0,0001 |
| pmed19 | 3470 | 21,97% | 0,0001 |
| pmed20 | 2407 | 34,54% | 0,0001 |
| pmed21 | 9362 | 2,45% | 0,0001 |
| pmed22 | 9697 | 13,03% | 0,0001 |
| pmed23 | 6031 | 30,57% | 0,0001 |
| pmed24 | 3638 | 22,86% | 0,0001 |
| pmed25 | 2374 | 29,87% | 0,0001 |
| pmed26 | 10563 | 6,51% | 0,0001 |
| pmed27 | 9030 | 8,70% | 0,0001 |
| pmed28 | 5706 | 26,86% | 0,0001 |
| pmed29 | 3709 | 22,29% | 0,0001 |
| pmed30 | 2512 | 26,29% | 0,0001 |
| pmed31 | 10409 | 3,20% | 0,0001 |
| pmed32 | 10201 | 9,72% | 0,0001 |
| pmed33 | 6136 | 30,55% | 0,0001 |
| pmed34 | 3786 | 25,66% | 0,0001 |
| pmed35 | 11792 | 13,38% | 0,0161 |
| pmed36 | 10764 | 8,36% | 0,0001 |
| pmed37 | 7216 | 42,69% | 0,0001 |
| pmed38 | 11663 | 5,45% | 0,0001 |
| pmed39 | 10641 | 12,93% | 0,0001 |
| pmed40 | 6578 | 28,28% | 0,0001 |

Πίνακας Α΄.22: Neighborhood Search (Proportional worst input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6006 | 3,21% | 0,0001 |
| pmed2 | 4788 | 16,98% | 0,0001 |
| pmed3 | 5654 | 33,04% | 0,0001 |
| pmed4 | 3713 | 22,38% | 0,0001 |
| pmed5 | 1758 | 29,74% | 0,0001 |
| pmed6 | 10585 | 35,29% | 0,0001 |
| pmed7 | 6876 | 22,11% | 0,0001 |
| pmed8 | 5898 | 32,69% | 0,0001 |
| pmed9 | 3293 | 20,45% | 0,0001 |
| pmed10 | 1414 | 12,67% | 0,0001 |
| pmed11 | 8655 | 12,46% | 0,0001 |
| pmed12 | 7848 | 18,30% | 0,0001 |
| pmed13 | 5773 | 31,98% | 0,0001 |
| pmed14 | 3741 | 26,04% | 0,0001 |
| pmed15 | 2063 | 19,32% | 0,0001 |
| pmed16 | 9592 | 17,52% | 0,0001 |
| pmed17 | 9488 | 35,56% | 0,0001 |
| pmed18 | 6142 | 27,72% | 0,0001 |
| pmed19 | 3726 | 30,97% | 0,0001 |
| pmed20 | 2087 | 16,66% | 0,0001 |
| pmed21 | 10179 | 11,39% | 0,0001 |
| pmed22 | 10555 | 23,03% | 0,0001 |
| pmed23 | 5795 | 25,46% | 0,0001 |
| pmed24 | 3615 | 22,09% | 0,0001 |
| pmed25 | 2153 | 17,78% | 0,0001 |
| pmed26 | 11566 | 16,63% | 0,0001 |
| pmed27 | 9318 | 12,17% | 0,0001 |
| pmed28 | 6129 | 36,26% | 0,0001 |
| pmed29 | 3899 | 28,55% | 0,0001 |
| pmed30 | 2349 | 18,10% | 0,0001 |
| pmed31 | 12574 | 24,67% | 0,0001 |
| pmed32 | 10987 | 18,18% | 0,0001 |
| pmed33 | 6097 | 29,72% | 0,0001 |
| pmed34 | 3784 | 25,59% | 0,0001 |
| pmed35 | 11280 | 8,46% | 0,0001 |
| pmed36 | 12222 | 23,03% | 0,0001 |
| pmed37 | 7330 | 44,95% | 0,0001 |
| pmed38 | 12873 | 16,39% | 0,0101 |
| pmed39 | 11213 | 19,00% | 0,0151 |
| pmed40 | 6236 | 21,61% | 0,0001 |

Πίνακας Α΄.23: Neighborhood Search (Sample greedy input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6500 | 11,70% | 0,0001 |
| pmed2 | 4763 | 16,37% | 0,0001 |
| pmed3 | 4486 | 5,55% | 0,0001 |
| pmed4 | 3396 | 11,93% | 0,0001 |
| pmed5 | 1684 | 24,28% | 0,0001 |
| pmed6 | 8512 | 8,79% | 0,0001 |
| pmed7 | 6135 | 8,95% | 0,0001 |
| pmed8 | 5154 | 15,95% | 0,0001 |
| pmed9 | 3313 | 21,18% | 0,0001 |
| pmed10 | 1825 | 45,42% | 0,0001 |
| pmed11 | 8038 | 4,44% | 0,0001 |
| pmed12 | 7079 | 6,71% | 0,0001 |
| pmed13 | 4930 | 12,71% | 0,0001 |
| pmed14 | 3610 | 21,63% | 0,0001 |
| pmed15 | 2281 | 31,93% | 0,0001 |
| pmed16 | 9272 | 13,60% | 0,0001 |
| pmed17 | 7457 | 6,54% | 0,0001 |
| pmed18 | 5340 | 11,04% | 0,0001 |
| pmed19 | 3330 | 17,05% | 0,0271 |
| pmed20 | 2369 | 32,42% | 0,0001 |
| pmed21 | 10346 | 13,22% | 0,0001 |
| pmed22 | 9032 | 5,28% | 0,0001 |
| pmed23 | 5228 | 13,18% | 0,0001 |
| pmed24 | 3481 | 17,56% | 0,0001 |
| pmed25 | 2614 | 43,00% | 0,0001 |
| pmed26 | 11328 | 14,23% | 0,0001 |
| pmed27 | 9310 | 12,07% | 0,0091 |
| pmed28 | 5119 | 13,81% | 0,0001 |
| pmed29 | 3602 | 18,76% | 0,0001 |
| pmed30 | 2597 | 30,57% | 0,0001 |
| pmed31 | 10903 | 8,10% | 0,0001 |
| pmed32 | 10166 | 9,35% | 0,0001 |
| pmed33 | 5154 | 9,66% | 0,0001 |
| pmed34 | 3736 | 24,00% | 0,0001 |
| pmed35 | 10905 | 4,86% | 0,0001 |
| pmed36 | 10687 | 7,58% | 0,0001 |
| pmed37 | 5630 | 11,33% | 0,0001 |
| pmed38 | 11902 | 7,61% | 0,0001 |
| pmed39 | 10330 | 9,63% | 0,0001 |
| pmed40 | 5878 | 14,63% | 0,0001 |

Πίνακας Α΄.24: Neighborhood Search (Drop approach input)

| όνομα αρχείου | λύση | απόκλιση | χρόνος εκτέλεσης (s) |
|---|---|---|---|
| pmed1 | 6521 | 12,06% | 0,0001 |
| pmed2 | 5799 | 41,68% | 0,0001 |
| pmed3 | 5789 | 36,21% | 0,0001 |
| pmed4 | 3878 | 27,82% | 0,0001 |
| pmed5 | 1613 | 19,04% | 0,0001 |
| pmed6 | 9355 | 19,57% | 0,0161 |
| pmed7 | 6610 | 17,39% | 0,0001 |
| pmed8 | 5481 | 23,31% | 0,0001 |
| pmed9 | 3667 | 34,13% | 0,0001 |
| pmed10 | 1480 | 17,93% | 0,0001 |
| pmed11 | 9086 | 18,06% | 0,0001 |
| pmed12 | 8910 | 34,31% | 0,0001 |
| pmed13 | 6280 | 43,58% | 0,0001 |
| pmed14 | 3611 | 21,66% | 0,0001 |
| pmed15 | 2147 | 24,18% | 0,0001 |
| pmed16 | 8491 | 4,03% | 0,0001 |
| pmed17 | 8934 | 27,65% | 0,0001 |
| pmed18 | 6301 | 31,03% | 0,0001 |
| pmed19 | 3745 | 31,63% | 0,0001 |
| pmed20 | 2285 | 27,72% | 0,0001 |
| pmed21 | 10271 | 12,40% | 0,0001 |
| pmed22 | 10781 | 25,67% | 0,0001 |
| pmed23 | 6205 | 34,34% | 0,0001 |
| pmed24 | 3629 | 22,56% | 0,0001 |
| pmed25 | 2189 | 19,75% | 0,0001 |
| pmed26 | 11843 | 19,42% | 0,0151 |
| pmed27 | 10472 | 26,06% | 0,0001 |
| pmed28 | 5853 | 30,12% | 0,0001 |
| pmed29 | 4016 | 32,41% | 0,0001 |
| pmed30 | 2378 | 19,56% | 0,0001 |
| pmed31 | 11768 | 16,68% | 0,0151 |
| pmed32 | 11154 | 19,97% | 0,0161 |
| pmed33 | 5998 | 27,62% | 0,0001 |
| pmed34 | 4002 | 32,82% | 0,0001 |
| pmed35 | 13548 | 30,27% | 0,0001 |
| pmed36 | 12224 | 23,05% | 0,0001 |
| pmed37 | 6600 | 30,51% | 0,0001 |
| pmed38 | 14674 | 32,68% | 0,0001 |
| pmed39 | 10922 | 15,91% | 0,0001 |
| pmed40 | 6753 | 31,69% | 0,0001 |