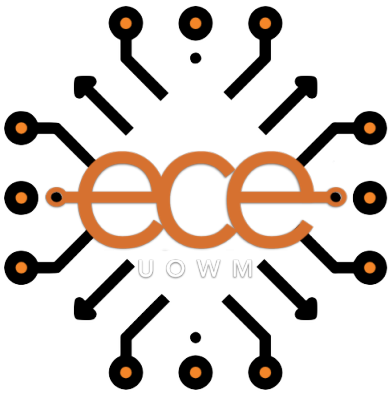


Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Υλοποίηση και Υπολογιστική Σύγκριση Αλγορίθμων για Προβλήματα Βέλτιστης Κοπής

Κωνσταντίνος Λάμπρου (ΑΜ: 706)

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

Κοζάνη

14 Μαρτίου 2022

Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η υλοποίηση και σύγκριση αλγορίθμων αναφορικά με το πρόβλημα της βέλτιστης κοπής υλικών και του συνακόλουθου αποθέματος που εμφανίζεται σε γραμμές παραγωγής. Το ως άνω πρόβλημα θα μελετηθεί εντός του πλαισίου κοπής υλικών σταθερού αρχικού μεγέθους σε μικρότερα κομμάτια.

Το πρόβλημα ανήκει στην κατηγορία του γραμμικού προγραμματισμού, ενώ για την επίλυσή του χρησιμοποιήθηκαν διάφοροι ευρετικοί αλγόριθμοι. Τα αποτελέσματα που προκύπτουν σχολιάζονται ως προς το απόθεμα, την ποσότητα του υλικού που χρησιμοποιήθηκε, την πολυπλοκότητα των αλγορίθμων αλλά και ως προς την ταχύτητα εύρεσης λύσης του κάθε αλγόριθμου. Τα δεδομένα αντλήθηκαν από πραγματικές ανάγκες γραμμών παραγωγής (βάσει έρευνας που πραγματοποιήθηκε το 2021 στην περιοχή του Δήμου Λαγκαδά Νομού Θεσσαλονίκης) και από τυχαία προβλήματα που παράχθηκαν από μία γεννήτρια τυχαίων αριθμών.

Λέξεις κλειδιά: Πρόβλημα βέλτιστης κοπής, Γραμμικός προγραμματισμός, Ευρετικοί αλγόριθμοι, Υπολογιστική μελέτη

Abstract

The purpose of this thesis is to implement and compare algorithms regarding the cutting stock problem and the consequent stock that appears in production lines. The above problem will be studied within the framework of cutting materials of constant initial size into smaller pieces.

The problem belongs to the category of linear programming problems, while to solve it various heuristic algorithms were used. The results are compared in terms of stock, the amount of material used, the complexity of the algorithms but also in terms of the speed of finding a solution. Data was drawn from real needs of production lines (research carried out in 2021 in the area of the Municipality of Lagkada, Perfecture of Thessaloniki) and from randomly generated problems.

Keywords: Cutting stock problem, Linear programming, Heuristic algorithms, Computational study

Δήλωση Πνευματικών Δικαιωμάτων

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

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

Υπογραφή Φοιτητή

Περιεχόμενα

Κατάλογος αλγορίθμων

Κατάλογος πινάκων

Κατάλογος απεικονίσεων

Κεφάλαιο 1

Εισαγωγή

Το πρόβλημα της βέλτιστης κοπής (cutting stock problem) είναι ένα πρόβλημα βελτιστοποίησης και ανήκει στην κατηγορία των προβλημάτων γραμμικού προγραμματισμού (linear programming) με τις μεταβλητές απόφασης να περιορίζονται περαιτέρω να λαμβάνουν ακέραιες τιμές. Επί του παρόντος τεχνικές με βελτιστοποίηση έχουν γίνει απαραίτητο εργαλείο για βιομηχανικές εφαρμογές συμπεριλαμβανομένου κατανομής των πόρων και αποθέματος και λήψης αποφάσεων. Οι τεχνικές βελτιστοποίησης έχουν διάφορους κλάδους και ένας τέτοιος κλάδος είναι ο γραμμικός προγραμματισμός ο οποίος είναι μία μαθηματική τεχνική που βοηθά στον σχεδιασμό και τη λήψη αποφάσεων σχετικά με τη σωστή κατανομή των πόρων [13]. Σε όλα τα προβλήματα γραμμικού προγραμματισμού υπάρχουν εναλλακτικές λύσεις, εκ των οποίων επιλέγεται η βέλτιστη, η οποία αποβλέπει συνήθως στη μεγιστοποίηση του κέρδους ή στην ελαχιστοποίηση του κόστους, αντίστοιχα. Ο γραμμικός προγραμματισμός είναι μία ευρέως χρησιμοποιούμενη τεχνική μαθηματικής μοντελοποίησης για τον καθορισμό της βέλτιστης κατανομής των πόρων ανάμεσα σε ανταγωνιστικές απαιτήσεις. Οι πόροι μπορεί να περιλαμβάνουν πρώτες ύλες, ανθρώπινο δυναμικό, μηχανήματα, χρόνο, χρήματα, χώρο κ.λπ. Ο γραμμικός προγραμματισμός επίσης είναι ένα δυναμικό εργαλείο που θεωρείται εξαιρετικά χρήσιμο λόγω της εφαρμογής του σε πολλούς διαφορετικούς τύπους πραγματικών επαγγελματικών προβλημάτων σε τομείς, όπως ο χρηματοπιστωτικός, η παραγωγή, η διανομή και οι πωλήσεις, το προσωπικό, το μάρκετινγκ και πολλοί ακόμα τομείς της διοίκησης.

Το πρόβλημα της βέλτιστης κοπής είναι ένα πρόβλημα βελτιστοποίησης, στο οποίο αντικείμενα διαφορετικών μεγεθών πρέπει να συσκευάζονται σε έναν αριθμό κάδων, καθένας με σταθερή δεδομένη χωρητικότητα, με τρόπο που ελαχιστοποιεί

τον αριθμό των κάδων που χρησιμοποιούνται. Δεδομένου ενός συνόλου απαιτήσεων παραγγελίας, κάθε απαίτηση που αποτελείται από μια ζήτηση και ένα πλάτος, και έναν αριθμό πανομοιότυπων ρολών, το πρόβλημα της βέλτιστης κοπής υλικού συνίσταται στον προσδιορισμό του μικρότερου αριθμού ρολών που πρέπει να κοπούν για να ικανοποιηθούν όλες οι απαιτήσεις. Σε αυτήν την εργασία, εστιάζουμε σε μονοδιάστατα προβλήματα.

1.1 Ορισμός του προβλήματος

Το πρόβλημα της βέλτιστης κοπής αφορά τη συμπλήρωση μιας παραγγελίας από διάφορα μήκη υλικού διαφόρων διαστάσεων σε έναν ή και περισσότερους χώρους, με τέτοιον τρόπο ώστε να έχουμε το ελάχιστο κόστος. Τα διάφορα μήκη υλικού δεν πρέπει να αλληλεπικαλύπτονται, οι άκρες τους δεν πρέπει να προεξέχουν και συνήθως τα αντικείμενα έχουν διαφορετικό μέγεθος.

Τις τελευταίες δεκαετίες το συγκεκριμένο πρόβλημα έχει προκαλέσει μεγάλο ενδιαφέρον σε διάφορους επιστημονικούς τομείς. Αυτό έχει ως αποτέλεσμα να έχουμε πολυάριθμες επιστημονικές δημοσιεύσεις και έρευνες για διάφορα θέματα που αφορούν τα προβλήματα αυτά. Είναι αξιοσημείωτο ότι τα θέματα της μελέτης δεν αφορούν μόνο την επιστήμη των υπολογιστών και τον τομέα της παραγωγής αλλά τελείως διαφορετικούς τομείς της επιστημονικής κοινότητας, όπως για παράδειγμα τις Επιστήμες Μηχανικής, Διοίκησης, Πληροφορικής, Μαθηματικών καθώς και την Επιχειρησιακή Έρευνα [23].

Καθώς το πεδίο εφαρμογής του προβλήματος πλησιάζει τα όρια της διαθέσιμης τεχνολογίας από άποψη της μνήμης του υπολογιστή και του αριθμού των απαιτούμενων λειτουργιών, τόσο η ανάγκη όσο και η χρήση προηγμένων τεχνικών της Επιστήμης Υπολογιστών γίνονται πιο εμφανής. Το πρόβλημα βέλτιστης κοπής διατυπώθηκε για πρώτη φορά το 1939. Λίγα χρόνια αργότερα το 1951, και πριν από τους πρώτους υπολογιστές, ο Kantorovich μαζί με τον Zalgaller πρότειναν την επίλυση του προβλήματος της οικονομικής χρήσης ενός υλικού, κατά το στάδιο της κοπής με τη βοήθεια του γραμμικού προγραμματισμού. Η επίλυση ενός τέτοιου προβλήματος βελτιστοποίησης μπορεί να είναι οικονομικά σημαντική για μια σύγχρονη παραγωγή ράβδων αλουμινίου και μπορεί να αποφέρει κέρδος της τάξης των εκατομμυρίων δολαρίων.

Στην περίπτωση του προβλήματος των αποθεμάτων κοπής στη βιομηχανία χαρτιού και λαμβάνοντας ρολά χαρτιού σταθερού πλάτους και μια σειρά παραγγελιών για ρολά μικρότερου πλάτους, ο στόχος του προβλήματος βέλτιστης κοπής είναι να καθοριστεί ο τρόπος κοπής των ρολών σε μικρότερα πλάτη για την εκπλήρωση των παραγγελιών με τέτοιο τρόπο ώστε να ελαχιστοποιείται το ποσό των απορριμμάτων. Η μονοδιάστατη κοπή συμβαίνει όταν κόβονται για παράδειγμα σωλήνες, καλώδια και χαλύβδινες ράβδους, ενώ αντιμετωπίζονται δισδιάστατα προβλήματα στα έπιπλα, τα ρούχα και στην παραγωγή γυαλιού. Τέλος, δεν είναι γνωστές πολλές τρισδιάστατες εφαρμογές κοπής.

1.2 Κίνητρα και στόχοι

Πολλά προβλήματα του πραγματικού κόσμου λύνονται με μοντελοποίηση και αυτοματοποιημένα συστήματα με αλγόριθμους οι οποίοι εξελίσσονται συνεχώς. Μεγάλος αριθμός αυτών μπορεί να προσεγγιστεί από γραμμικά μοντέλα. Υπάρχουν πασίγνωστες επιτυχείς εφαρμογές των άνω σε Βιομηχανία, Μάρκετινγκ, Χρηματοοικονομικά κ.τ.λ. Ένα παράδειγμα του πόσο σημαντική είναι η σωστή διαχείριση των υλικών και αναδεικνύει την ανάγκη της μελέτης τέτοιων αλγόριθμων είναι το εξής: Το 2020 στην Κίνα κατασκευάστηκαν 322 εκατομμύρια ξύλινα έπιπλα, τα οποία έχουν μέση κατανάλωση σανίδας ανά έπιπλο $0.6\ m^3$. Αν καταφέρουμε να αυξήσουμε το ποσοστό της χρήσης αυτών των σανίδων κατά 1%, αυτό θα έχει σαν αποτέλεσμα να εξοικονομηθούν 193 εκατομμύρια m^3 σανίδες. Μια μικρή αλλαγή στο ποσοστό μπορεί να έχει μεγάλη αλλαγή στη συνολική κατανάλωση ξύλου και γενικότερα πρώτων υλών. Σε βιομηχανίες, όπως η κατασκευή επίπλων, η βέλτιστη χρησιμοποίηση πρώτων υλών είναι το πρόβλημα που αντιμετωπίζουν και κάθε αλλαγή στο ποσοστό φέρνει τεράστια οφέλη στα οικονομικά μιας επιχείρησης [9] [24] [22].

1.3 Διάρθρωση κειμένου

Μετά την εισαγωγή του προβλήματος της βέλτιστης κοπής, τον ορισμό, τα κίνητρα, τους στόχους και τα πραγματικά παραδείγματα που δόθηκαν για την καλύτερη κατανόηση του προβλήματος, η διάρθρωση του κειμένου συνεχίζει με τα ακόλουθα κεφάλαια. Στο κεφάλαιο 2 παρουσιάζεται η βιβλιογραφική μελέτη του προβλήμα-

τος της βέλτιστης κοπής, το μαθηματικό μοντέλο και σχετική έρευνα με χρονολογική σειρά με τα κριτήρια που χρησιμοποιήθηκαν για να λυθεί το πρόβλημα. Επίσης, παρουσιάζονται τα υπολογιστικά αποτελέσματα παρόμοιας με τη δική μας έρευνας.

Στο κεφάλαιο 3 παρουσιάζονται οι αλγόριθμοι επίλυσης, η πολυπλοκότητά τους, τα σχετικά διαγράμματα ροής, ο ψευδοκώδικας των αλγορίθμων και από ένα παράδειγμα με βήμα προς βήμα επίλυση για κάθε αλγόριθμο.

Στο κεφάλαιο 4 παρουσιάζονται αρχικά 40 παραδείγματα με πραγματικές τιμές από το εργαστάσιο επίπλων με την επωνυμία "Ν. Μάρκου" και από επιχείρηση με αντικείμενο την εμπορία οικοδομικών υλικών. Στη συνέχεια δίνονται 10 επιπλέον παραδείγματα από γεννήτρια τυχαίων αριθμών με μέγεθος δεκάδες έως μερικές εκατοντάδες χιλιάδες στοιχεία. Τέλος, υπάρχουν οι συνοπτικοί πίνακες με τους μέσους όρους των ως άνω προβλημάτων καθώς και σχολιασμός αυτών.

Στο κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα των συγκρίσεων βάσει διαφόρων κριτηρίων, παρατίθενται τα γενικά συμπεράσματα και τέλος σχολιάζονται οι μελλοντικές προεκτάσεις.

Επίσης, στο παράρτημα παρατίθεται συνοδευτικό και υποστηρικτικό υλικό.

Κεφάλαιο 2

Βιβλιογραφική ανασκόπηση

2.1 Το πρόβλημα της βέλτιστης κοπής

Το πρόβλημα βέλτιστης κοπής είναι ένα ευρέως γνωστό πρόβλημα στον τομέα της επιχειρησιακής έρευνας. Η πιο συνήθης εφαρμογή του είναι σε μια διάσταση, αλλά υπάρχουν εφαρμογές του και σε άλλες διαστάσεις. Το πρόβλημα σε δύο διαστάσεις είναι στην ουσία μια γενίκευση του προβλήματος σε μια διάσταση με ακριβώς τα ίδια δεδομένα και στόχους αλλά με διαφορετική προσέγγιση στη γεωμετρία των αντικειμένων, τα οποία πλέον έχουν πέρα από μήκος και πλάτος ή ύψος [7]. Το τρισδιάτατο πρόβλημα βέλτιστης κοπής έχει μελετηθεί σημαντικά λιγότερο και παρουσιάζει λιγότερες εφαρμογές.

Δεδομένου ότι δεν πρόκειται για ένα σύγχρονο πρόβλημα αλλά ένα πρόβλημα που υπάρχει εδώ και δεκαετίες, καταλαβαίνουμε ότι οι αρχικές λύσεις δόθηκαν με χειρωνακτική προσέγγιση της λύσης, μια διαδικασία που αργεί να μας δώσει λύση και απαιτεί χρόνο και εργασία αρκετών ατόμων. Αυτό άλλαξε με τη χρήση των ηλεκτρονικών υπολογιστών, κάτι το οποίο έκανε και να μηδενιστεί το ανθρώπινο λάθος στο κομμάτι των πράξεων και συγκρίσεων αλλά και το ενδεχόμενο να παραλειφθεί ένα αντικείμενο και να αλλάξει τελείως το αποτέλεσμα.

Το πρόβλημα της βέλτιστης κοπής είναι το πρόβλημα της πλήρωσης μιας παραγγελίας με το ελάχιστο κόστος για καθορισμένους αριθμούς μηκών υλικού που πρόκειται να κοπούν από δεδομένα μήκη αποθεμάτων δεδομένου κόστους. Όταν εκφράζεται ως πρόβλημα γραμμικού προγραμματισμού, ο μεγάλος αριθμός των μεταβλητών που εμπλέκονται καθιστά ανέφικτη την επίλυσή του [8]. Αρκετοί αλγόριθμοι χρησιμοποιούνται επί του παρόντος στη βιομηχανία, αλλά σχεδόν όλα αυτά

τα συστήματα θεωρούνται κλειστά και συγκεκριμένα για τις ανάγκες του κάθε προβλήματος. Το πρόβλημα των αποθεμάτων κοπής έχει προσαρμοστεί ποικιλοτρόπως σε εφαρμογές σε πολλές βιομηχανίες. Από την αρχική του μορφή έχει τροποποιηθεί για χρήση σε χαρτί, ξυλεία, υφάσματα, μέταλλο, σφράγιση δέρματος και άλλες βιομηχανίες. Σε κάθε περίπτωση, το πρόβλημα έχει αναδιατυπωθεί για να ταιριάζει στις ανάγκες του συγκεκριμένου κλάδου.

Παρακάτω παρουσιάζεται το μαθηματικό μοντέλο του προβλήματος της βέλτιστης κοπής. Κάθε κομμάτι προς κοπή κόβεται έτσι ώστε κάθε ράβδος να μην υπερβαίνει το μήκος της ράβδου c και ο αριθμός των ράβδων που χρησιμοποιούνται να είναι ο ελάχιστος. Έστω W_j το μήκος κομματιού j, οι οποίοι είναι θετικοί ακέραιοι αριθμοί, το σύνολο των κομματιών και B το σύνολο των κάδων. Οι μεταβλητές απόφασεις είναι δύο, y_i είναι η μεταβλητή για το κάδο i (1 αν χρησιμοποιείται, 0 αν δε χρησιμοποιείται) και x_{ij} είναι η μεταβλητή ανάθεσης στοιχείου j στον κάδο i (1 αν εκχωρήθηκε, 0 αν δεν εκχωρήθηκε).

$$\begin{aligned} \min_{z} \quad & z = \sum_{i=1}^{n} y_i \\ \text{subject to:} \quad & \sum_{j=1}^{n} x_{ij} w_j \leq c y_i, \quad i \in B \\ & \sum_{j=1}^{n} x_{ij} = 1, \qquad\qquad j \in N \end{aligned}$$

$$y_i = \begin{cases} 1, & \text{αν ο κάδος } i \text{ χρησιμοποιείται} \\ 0, & \text{αν ο κάδος } i \text{ δε χρησιμοποιείται} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{αν το κομμάτι } j \text{ εκχωρήθηκε στον κάδο } i \\ 0, & \text{αν το κομμάτι } j \text{ δεν εκχωρήθηκε στον κάδο } i \end{cases}$$

Ο πρώτος περιορισμός εγγυάται ότι το άθροισμα των μεγεθών των αντικειμένων που έχουν εκχωρηθεί στο κάδο i δεν υπερβαίνει τη χωρητικότητά του, c, ενώ ο δεύτερος περιορισμός βεβαιώνει ότι κάθε αντικείμενο έχει εκχωρηθεί σε έναν κάδο. Η αντικειμενική συνάρτηση ελαχιστοποιεί τη χρήση των διαθέσιμων κάδων.

2.2 Σχετική έρευνα

Ο Farley [4] ήταν ο πρώτος συγγραφέας που αντιμετώπισε το πρόβλημα βέλτιστης κοπής αποθεμάτων και παρτίδων. Εξέτασε τη βιομηχανία ένδυσης και πρότεινε ακέραια και τετραγωνικά μοντέλα προγραμματισμού έτσι ώστε να ελαχιστοποιείται η συνολική κοπή, το ράψιμο και τα έξοδα αποθήκευσης. Το ερώτημα που εξετάστηκε και απαντήθηκε αφορούσε τον τρόπο συνδυασμού τέτοιων μοτίβων ώστε να ικανοποιηθούνται οι διάφοροι στόχοι.

Οι Hendry et al. [14] πρότειναν μια προσέγγιση λύσης δύο σταδίων για το πρόβλημα βέλτιστης κοπής στη βιομηχανία χαλκού. Στο πρώτο στάδιο, το πρόβλημα των αποθεμάτων κοπής λύθηκε ευρετικά και στο δεύτερο στάδιο το μέγεθος της παρτίδας υπολογίστηκε χρησιμοποιώντας ένα μοντέλο ακέραιου προγραμματισμού. Αποδεικνύεται ότι η τρέχουσα απόδοση θα μπορούσε να βελτιωθεί χρησιμοποιώντας όλες τις ευρετικές μεθόδους που εξετάστηκαν στο πρώτο στάδιο.

Οι Nonås και Thorstenson [18] μελέτησαν το δισδιάστατο πρόβλημα κοπής που προέκυψε σε μια νορβηγική εταιρεία φορτηγών. Υπέθεσαν ακανόνιστα σχήματα και στοχαστική ζήτηση η οποία στόχευε στην ελαχιστοποίηση της συνολικής πρώτης ύλης και του κόστους εγκατάστασης. Παρουσίασαν μια διαδικασία για την επίλυση περιπτώσεων προβλημάτων μικρού μεγέθους, ενώ βελτίωσαν τη διαδικασία χρησιμοποιώντας την ευρετική μέθοδο που προτάθηκε από τον Haessler [12] για την επίλυση περιπτώσεων προβλημάτων μεγαλύτερου μεγέθους.

Οι Poltroniere et al. [19] μελέτησαν ένα μονοδιάστατο πρόβλημα κοπής που προκύπτει στη βιομηχανία χαρτιού. Ο προγραμματισμός που ελαχιστοποιεί τις εγκαταστάσεις και το κόστος παραγωγής μπορεί να παράγει ρολά που μπορεί να αυξήσουν τα απόβλητα στη διαδικασία κοπής. Από την άλλη πλευρά, ο καλύτερος αριθμός ρολών από την άποψη της ελαχιστοποίησης των απορριμμάτων μπορεί να οδηγήσει σε υψηλό κόστος εγκατάστασης. Στόχος τους ήταν να ελαχιστοποιήσουν το άθροισμα του κόστους αποθέματος, του κόστους εγκατάστασης, του κόστους απορριμμάτων υλικών και του κόστους απογραφής τελικού είδους. Ανέπτυξαν ένα μοντέλο προγραμματισμού και δύο ευρετικές διαδικασίες. Ο πρώτος ευρετικός αλγόριθμος χρησιμοποιεί μια Lagrangian ιδέα χαλάρωσης και αρχικά λύνει το πρόβλημα με το μέγεθος. Ο δεύτερος ευρετικός λύνει πρώτα το πρόβλημα του αποθέματος κοπής

και μετά χρησιμοποιεί τη λύση για να βρει τα μεγέθη της παρτίδας.

Οι Gramani και França [10] μελέτησαν ένα δισδιάστατο πρόβλημα κοπής και το πρόβλημα μεγέθους παρτίδας σε μια βιομηχανία ξύλου. Στόχος τους ήταν να ελαχιστοποιήσουν τα συνολικά κόστη, τον αριθμό των ράβδων και του κόστος μεταφοράς αποθεμάτων ειδών και πρότειναν μια προσέγγιση λύσης που βασίζεται στη ροή του δικτύου. Στο πλαίσιο αυτό παρουσίασαν ορισμένα υπολογιστικά αποτελέσματα που συγκρίνουν τις συνδυασμένες λύσεις προβλημάτων με εκείνες που λαμβάνονται με τη μέθοδο που χρησιμοποιείται γενικά στη βιομηχανία. Αυτά τα αποτελέσματα καταδεικνύουν ότι με τον συνδυασμό των προβλημάτων είναι δυνατό να υπάρξουν οφέλη έως και 28% κέρδους.

Οι Gramani et al. [11] επέκτεινε το μοντέλο των Gramani and Franca [10] ώστε να συμπεριληφθεί η παραγωγή και η κατοχή αποθεμάτων κόστους των τελικών προϊόντων. Πρότειναν μια ευρετική βασισμένη στη Lagrangian χαλάρωση προσέγγιση όπου τα υποπροβλήματα απαιτούν εκθετική προσπάθεια.

Οι Silva et al. [21] μελέτησαν ένα δισδιάστατο πρόβλημα κοπής και το μέγεθος της παρτίδας στη βιομηχανία επίπλων. Στόχος τους ήταν να ελαχιστοποιήσουν το συνολικό υλικό, τη φύρα και το κόστος αποθήκευσης. Εξέτασαν τα υπολείμματα για την πιθανή χρήση τους στις επόμενες περιόδους του σχεδιασμού που μπορεί να συμβάλουν στη μείωση της συνολικής σπατάλης. Πρότειναν δύο μοντέλα μαθηματικού προγραμματισμού, επεκτείνοντας τα προηγούμενα αναφερόμενα μοντέλα στη βιβλιογραφία. Έδειξαν ότι δεν υπάρχει κάποιο μοντέλο που κυριαρχεί, επομένως μπορούν να χρησιμοποιηθούν μαζί.

Οι Leao et al. [17] μελέτησαν το μονοδιάστατο πρόβλημα κοπής και το μέγεθος της παρτίδας στη βιομηχανία χαρτιού. Ανέπτυξαν τρία μαθηματικά μοντέλα: προσανατολισμένα σε πρότυπα, αποσύνθεση περιόδου και αποσύνθεση μηχανών. Για να λυθεί γραμμικά το πρόβλημα των μοντέλων, χρησιμοποιήθηκε μια τεχνική column generation. Για να λύσουν το μοντέλο αποσύνθεσης μηχανής, πρότειναν μια ευρετική μέθοδο που χρησιμοποιεί μια μέθοδο column generation μαζί με προσαρμοστική αναζήτηση γειτονιάς. Παρατήρησαν ότι το αποσυντεθειμένο μοντέλο προσανατολισμένο στο πρότυπο έχει ως αποτέλεσμα σχεδόν βέλτιστες λύσεις για το σύνολο δεδομένων που αναφέρονταν στη βιβλιογραφία και η ευρετική αποσύνθεση μηχανής δίνει καλύτερα αποτελέσματα για τα πραγματικά δεδομένα.

Οι Vanzela et al. [22] μελέτησαν το δισδιάστατο πρόβλημα κοπής και παρτίδας που προκύπτει στη βιομηχανία επίπλων της Βραζιλίας. Πρότειναν ένα μαθηματικό μοντέλο που ελαχιστοποιεί τη σπατάλη πρώτων υλών, την παραγωγή και το κόστος αποθέματος και περιλαμβάνει το επίπεδο αποθέματος ασφαλείας και τους περιορισμούς χωρητικότητας του πριονιού. Η λύση του μοντέλου συγκρίνεται με μια προσομοίωση της κοινής πρακτικής λήψης του μεγέθους της παρτίδας και των αποφάσεων κοπής αποθεμάτων χωριστά και διαδοχικά. Πρότειναν μια ευρετική προσέγγιση βασισμένη σε τεχνικές column generation για την επίλυση πραγματικών περιπτώσεων μεγάλου μεγέθους. Έδειξαν την ανωτερότητα των προσεγγίσεών τους όσον αφορά τη βελτίωση των λειτουργιών σχεδιασμού της εταιρείας έναντι των προσεγγίσεων που κάνουν τις αποφάσεις κοπής αποθεμάτων και μεγεθών παρτίδας χωριστά. Υποδεικνύουν επίσης ότι το μοντέλο μπορεί να υποστηρίξει τις κύριες αποφάσεις που λαμβάνονται και μπορεί να φέρει βελτιώσεις στον προγραμματισμό παραγωγής του εργοστασίου.

Οι Campello et al. [3] παρουσίασαν μια μελέτη πολυκριτήριας βελτιστοποίησης για το μονοδιάστατο πρόβλημα κοπής και τα προβλήματα σχετικά με το μέγεθος της παρτίδας που προκύπτουν στη βιομηχανία χαρτιού. Οι δύο στόχοι τους ήταν η ελαχιστοποίηση του συνολικού κόστους παραγωγής, του κόστους απογραφής των ρολών χαρτιού και του κόστους εγκατάστασης των μηχανών και η ελαχιστοποίηση των συνολικών απορριμμάτων υλικών και του κόστους απογραφής των ειδών. Πρότειναν δύο προσεγγίσεις λύσης: την προσέγγιση στάθμισης που ελαχιστοποιεί το σταθμισμένο άθροισμα των στόχων (πλήθος ράβδων) και μια μέθοδο περιορισμού ’ε’ όπου ο στόχος που σχετίζεται με το μέγεθος ελαχιστοποιείται και ο στόχος που σχετίζεται με το απόθεμα κοπής προστίθεται ως ένας περιορισμός. Για την επίλυση περιπτώσεων μεγαλύτερων μεγεθών, χρησιμοποιήθηκαν ευρετικές προσεγγίσεις θέσης των ακριβών λύσεων μοντέλου. Τα υπολογιστικά πειράματα αποκάλυψαν ότι οι στόχοι που σχετίζονται με το μέγεθος της παρτίδας και την κοπή αποθεμάτων είναι αντικρουόμενοι. Η κεντρική πτυχή αυτής της συνεισφοράς περιλαμβάνει τον ανασυνδυασμό αυτών των υπολειμματικών σχεδίων κοπής με διαφορετικούς τρόπους. Σε αυτή τη μελέτη, εξετάστηκε η προσέγγιση πολλαπλών κριτηρίων για το πρόβλημα κοπής αποθέματος και μεγέθους παρτίδας. Συγκρίθηκαν αλγόριθμοι με βάση το πλήθος των ράβδων που χρησιμοποιήθηκαν, τη φύρα, την υλοποίηση των

αλγορίθμων και το πλήθος συγκρίσεων για το τελικό αποτέλεσμα.

Παρόμοια με τη δική μας έρευνα έγινε από τον Rieck το 2021 [20], ο οποίος σύγκρινε παρόμοιους αλγόριθμους με αυτούς που παρουσιάζονται στο πλαίσιο αυτής της διπλωματικής. Τα αποτελέσματα των αλγορίθμων αυτών παρουσιάζονται στον Πίνακα 2.1. Ο αλγόριθμος επόμενος στη σειρά (NF) έχει την καλύτερη απόδοση στον χρόνο καθώς ο τρόπος λειτουργίας του είναι ο εξής: χωρίς να χρησιμοποιεί προηγούμενη φύρα, πηγαίνει στο επόμενο κάθε φορά κομμάτι, ελέγχει αν χωράει στον κάδο και σε αρνητική περίπτωση χρησιμοποιεί τον επόμενο κάδο. Για τον λόγο αυτό έχει τη χειρότερη απόδοση στους κάδους. Ο αλγόριθμος πρώτος στη σειρά (FF) είναι ο δεύτερος καλύτερος σε επόδοση στους κάδους απο τους μη ταξινομημένους και ο τρίτος στο σύνολο των αλγορίθμων του πίνακα. Έχει τον χειρότερο χρόνο καθώς πηγαίνει στο επόμενο κάθε φορά κομμάτι, ελέγχει αν χωράει στον κάδο και σε αρνητική περίπτωση συγκρίνει την προηγούμενη ράβδο αν χωράει η φύρα της τωρινής πριν ξεκινήσει επόμενο κάδο. Ο καλύτερος στη σειρά (BF) έχει την καλύτερη απόδοση από τους μη ταξινομημένους και είναι ο δεύτερος στο σύνολο των αλγορίθμων του πίνακα. Έχει χρόνο καλύτερο από τον NF και τον FF και αποτελεί τον τρίτο σε απόδοση στο σύνολο. Ελέγχει αν χωράει στον κάδο και σε αρνητική περίπτωση συγκρίνει από όλες τις υπόλοιπες φύρες αυτή που δίνει τη μικρότερη διαφορά προκειμένου να χωρέσει σε εκείνο τον κάδο. Ο αλγόριθμος επόμενος στη σειρά με ταξινόμηση (NFD) είναι ίδιος με τον αλγόριθμο επόμενος στη σειρά (NF), με τη διαφορά ότι έχει ταξινομημένα τα δεδομένα. Έχει τη δεύτερη χειρότερη επόδοση στον κάδο μετά τον NF και δεύτερο καλύτερο χρόνο μετά τον NF για τον ίδιο ακριβώς λόγο. Ο αλγόριθμος πρώτος στη σειρά με ταξινόμηση (FFD) είναι ίδιος με τον αλγόριθμο πρώτος στη σειρά (FF) με τη διαφορά ότι έχει ταξινομημένα τα δεδομένα. Έχει την καλύτερη απόδοση στους κάδους και τον δεύτερο χειρότερο χρόνο μετά τον FF.

	NF	FF	BF	NFD	FFD
Κάδοι	420985	319559	318739	408978	317997
Χρόνος	0.001349	0.8195	0.13565	0.026988	0.473721

Πίνακας 2.1: Πίνακας μέσων όρων χρόνου και ράβδων

Κεφάλαιο 3

Αλγόριθμοι επίλυσης των προβλημάτων

Στο παρόν κεφάλαιο θα αναφερθούν οι αλγόριθμοι επίλυσης που υλοποιήθηκαν, θα παρουσιαστούν τα σχετικά διαγράμματα ροής και ο ψευδοκώδικας των αλγορίθμων και στη συνέχεια θα μελετηθεί από ένα παράδειγμα με βήμα προς βήμα επίλυση. Οι παρακάτω αλγόριθμοι είναι ευρετικοί, δηλαδή αξιολογούν προσεγγιστικά τις ενδιάμεσες καταστάσεις ως προς την εκτιμώμενη απόσταση τους από μία τελική κατάσταση που θεωρείται λύση. Επεκτείνουν πρώτα αυτές που θεωρούν ως τοπικά βέλτιστες λύσεις (οι οποίες αναμένεται να οδηγήσουν συντομότερα σε λύση) ή/και κλαδεύουν τις υπόλοιπες. Όταν είναι δύσκολο να βρεθεί η βέλτιστη λύση ή μη πρακτικά εφικτό, μπορούν να χρησιμοποιηθούν ευρετικές μέθοδοι για να επιταχυνθεί η διαδικασία εύρεσης μιας ικανοποιητικής λύσης.

Όλοι οι αλγόριθμοι δέχονται ως είσοδο τα μήκη των κομματιών προς κοπή, το πλήθος τους και το σταθερό μέγεθος της ράβδου. Στα προβλήματα που θα παρουσιαστούν και αναλυθούν παρακάτω το μήκος της ράβδου είναι σταθερό και διαφορετικό για κάθε πρόβλημα για να ταιριάζει με τις πραγματικές συνθήκες κοπής υλικών. Οι ράβδοι έχουν ίσο ή μεγαλύτερο πλήθος από τα κομμάτια προς κοπή άρα δεν σχολιάζονται περιορισμοί προς αυτή την κατεύθυνση. Τα κομμάτια προς κοπή επίσης έχουν ίσο ή μικρότερο μήκος από τις ράβδους.

Οι αλγόριθμοι διακρίνονται βάσει της σειράς εισροής των δεδομένων σε δύο κατηγορίες: τους online και τους offline. Οι online αλγόριθμοι εφαρμόζονται όταν έρθει το πρώτο κομμάτι προς κοπή που τοποθετείται και δεν υπάρχει η δυνατότητα για επανατοποθέτηση. Αντίθετα, οι offline περιλαμβάνουν όλη την πληροφορία

για το μέγεθος των κομματιών προς κοπή και όλα τα δεδομένα έχουν εισρεύσει στο πρόγραμμα πριν ξεκινήσει ο αλγόριθμος και έτσι μπορούν να ταξινομηθούν. Στην παρούσα εργασία θα μελετηθούν αλγόριθμοι που λύνουν το ως άνω πρόβλημα με τέτοιο τρόπο ώστε τα αποτελέσματα να είναι συγκρίσιμα μεταξύ τους μέσω διαφόρων κριτηρίων. Επίσης, χωρίζονται και κατηγογοριοποιούνται οι αλγόριθμοι ανάλογα με τα κριτήρια που τίθενται. Με την πάροδο του χρόνου τα κριτήρια που εφαρμόζονταν για την επίλυση του προβλήματος της βέλτιστης κοπής εμπλουτίστηκαν μέχρι την κατάληξη στη χρήση πολλαπλών κριτηρίων.

3.1 Ονομαστική αναφορά αλγορίθμων

Στην παρακάτω λίστα γίνεται ονομαστική αναφορά των αλγόριθμων μαζί με τη συντομογραφία τους και στη συνέχεια αναλυτική επεξήγηση του καθενός ξεχωριστά στις επόμενες ενότητες.

- Επόμενος στη σειρά - Next Fit (NF)
- Επόμενος στη σειρά με ταξινόμηση - Next Fit Decreasing (NFD)
- Πρώτος στη σειρά - First Fit (FF)
- Πρώτος στη σειρά με ταξινόμηση - First Fit Decreasing (FFD)
- Καλύτερος στη σειρά - Best Fit (BF)
- Καλύτερος στη σειρά με ταξινόμηση - Best Fit Decreasing (BFD)
- Χειρότερος στη σειρά - Worst Fit (WF)
- Χειρότερος στη σειρά με ταξινόμηση - Worst Fit Decreasing (WFD)
- Full Bin Packing (FBP)

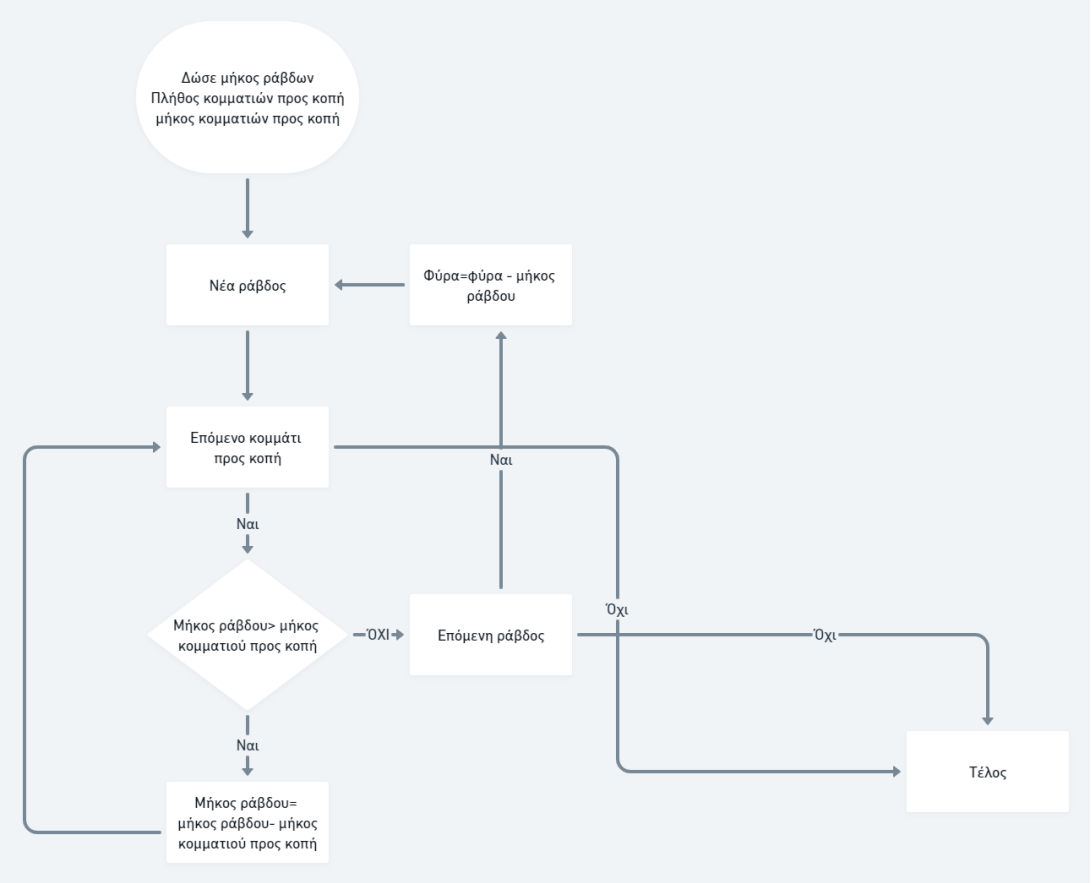
3.2 Ο αλγόριθμος Next Fit

Ο Next Fit είναι ο πιο απλός σε λειτουργία καθώς δεν αξιοποιεί προηγούμενη φύρα, δέχεται κάθε φορά το επόμενο μέγεθος κομματιού προς κοπή (έστω k_1), ελέγχει αν χωράει στο προς κοπή υλικό (έστω l) και σε θετική περίπτωση κάνει

την αφαίρεση $l - k_1 = l_1$ και πηγαίνει στο επόμενο μέγεθος προς κοπή (k_2). Σε διαφορετική περίπτωση, δηλαδή αν το μέγεθος του κομματιού προς κοπή είναι μεγαλύτερο από το μέγεθος του εναπομείναντος υλικού ($k_2 > l_1$), τότε εισάγει νέο υλικό (l). Ο Next Fit επαναλαμβάνει τα προηγούμενα βήματα μέχρι να μην υπάρχει νέο κομμάτι προς κοπή.

Η πολυπλοκότητα του Next Fit είναι $O(n)$ καθώς στην υλοποίηση του τρέχει όλα τα μεγέθη μια φορά. Η χρονική πολυπλοκότητα του NF επιτυγχάνεται χρησιμοποιώντας ένα δέντρο όπου τα φύλλα του αποθηκεύουν την πληροφορία για την υπολειπόμενη χωρητικότητα των χρησιμοποιημένων ράβδων. Στο δέντρο ισχύουν τα εξής: (1) κάθε κόμβος που δεν είναι φύλλο έχει 2 έως 3 παιδιά, (2) κάθε μονοπάτι από τη ρίζα στο φύλλο έχει το ίδιο μήκος, και (3) ετικέτες στους κόμβους επιτρέπουν την αναζήτηση συγκεκριμένου κόστους φύλλου, αναβαθμίζοντας το ή εισάγοντας νέο φύλλο. Η μελέτη για τη συγκεκριμένη δομή δεδομένων μπορεί να βρεθεί στο [1].

Στο Σχήμα 3.1 φαίνεται το διάγραμμα ροής του NF.



Σχήμα 3.1: Το διάγραμμα ροής του αλγόριθμου Next Fit

Στον Αλγόριθμο 1 δίνεται ο αλγόριθμος σε ψευδοκώδικα του Next Fit.

Result: Next Fit
Initialize $result$ (count of bins) and remaining $capacity$ in current bin.;
res = 0 ;
bin_rem = c;
Place items one by one ;
while *the Bin is empty* **do**
    i++;
    if *size[i]* $>$ *binRem* **then**
        // If this item can't fit in current bin;
        // then use a new bin;
        res++ ;
        bin-rem = c - size[i];
    $else$;
    binRem -= size[i];
end

Αλγόριθμος 1: Αλγόριθμος Next Fit.

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο Next Fit. Το πλήθος των κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 3 μέτρα είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 3$, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το δεύτερο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι προς κοπή που είναι 5 μέτρα είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 > 5$, άρα κόβεται από το 7 το 5 με υπόλοιπο 2.

Το τρίτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 4 είναι μικρότερο του 10. $10 > 4$, άρα κόβεται το 4 από το 10 με υπόλοιπο 6.

Το τέταρτο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το 7 είναι μικρότερο του 6 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $6 < 7$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεται το 7 από το 10 με υπόλοιπο 3.

Το πέμπτο κομμάτι προς κοπή είναι 1 μέτρο, γίνεται έλεγχος αν το 1 είναι μικρότερο του 3 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $1 < 3$ άρα κόβεται από το 3 το 1 με υπόλοιπο 2.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $2 < 3$, άρα παίρνουμε νέα ράβδο

Στη νέα ράβδο γίνεται έλεγχος αν το 3 είναι μικρότερο του 10. $10 > 3$, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το έβδομο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το 8 είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 < 8$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 8 είναι μικρότερο του 10. $10 > 8$, άρα κόβεται το 8 από το 10 με υπόλοιπο 2.

Ο αλγόριθμος χρησιμοποίησε συνολικά 5 ράβδους και η φύρα ήταν 19 μέτρα.

3.3 Ο αλγόριθμος Next Fit Decreasing

Είναι ο ίδιος αλγόριθμος με τον NF με τη διαφορά ότι τα κομμάτια που θα αφαιρεθούν (k) από το αρχικό υλικό (l), είναι ταξινομημένα. Η πολυπλοκότητα του NFD είναι $nlog(n)$ και η ανάλυσή του έγινε από τους Becker και Coffman [2].

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τον Next Fit Decreasing. Το πλήθος των κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Με ταξινόμηση η σειρά έχει ως εξής: $8, 7, 5, 4, 3, 3, 1$. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 8 μέτρα, είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 8$, άρα κόβεται το 8 από το 10 με υπόλοιπο 2.

Το δεύτερο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι προς κοπή, τα 7 μέτρα είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεται το 7 από το 10 με υπόλοιπο 3.

Το τρίτο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το 5 είναι μικρότερο του 3 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $5 > 3$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 5 είναι μικρότερο του 10. $10 > 5$, άρα κόβεταιτο 5 από το 10 με υπόλοιπο 5.

Το τέταρτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 5 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 < 5$, άρα κόβεται από το 4 το 5 με υπόλοιπο 1.

Το πέμπτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 1 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 1$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 3 είναι μικρότερο του 10. $10 > 3$, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 < 7$, άρα κόβεται από το 3 το 7 με υπόλοιπο 4.

Το έβδομο κομμάτι προς κοπή είναι 1 μέτρο, γίνεται έλεγχος αν το 1 είναι μικρότερο του 4 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $1 < 4$, άρα κόβεται από το 4 το 1 με υπόλοιπο 3.

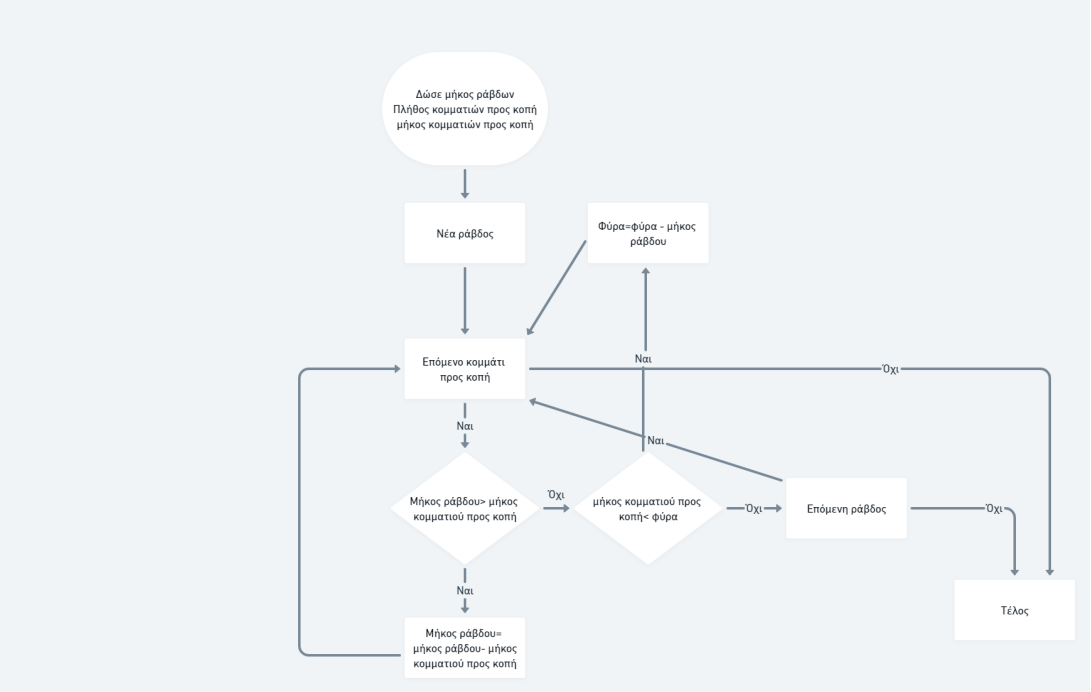
Ο αλγόριθμος χρησιμοποίησε συνολικά 4 ράβδους και είχε φύρα 8 μέτρα.

3.4 Ο αλγόριθμος First Fit

Ο First Fit δέχεται κάθε φορά το επόμενο μέγεθος κομματιού προς κοπή (έστω k_1), ελέγχει αν χωράει στο προς κοπή υλικό (έστω l) και σε θετική περίπτωση κάνει την αφαίρεση $l - k_1 = l_1$ και πηγαίνει στο επόμενο μέγεθος προς κοπή (k_2). Σε διαφορετική περίπτωση, δηλαδή αν το μέγεθος του κομματιού προς κοπή είναι μεγαλύτερο από το μέγεθος του εναπομείναντος υλικού ($k_2 > l_1$), ελέγχει αν χωράει κάποιο από τα προηγούμενα κομμάτια προς κοπή (k) και κάνει την αφαίρεση στο πρώτο που θα βρει. Αν δεν υπάρχει κάποιο κομμάτι που χωράει, εισάγει νέο υλικό. Ο First Fit επαναλαμβάνει τα προηγούμενα βήματα μέχρι να μην υπάρχει νέο κομμάτι προς κοπή.

Η πολυπλοκότητα του αλγόριθμου First-fit είναι $O(nlogn)$ και έχει αποδειχθεί από τους Johnson και Demers [16] και Ullman et al. [6].

Στο Σχήμα 3.2 φαίνεται το διάγραμμα ροής του FF.



Σχήμα 3.2: Το διάγραμμα ροής του αλγόριθμου First Fit

Στον Αλγόριθμο 2 δίνεται ο αλγόριθμος σε ψευδοκώδικα του First Fit.

Result: First Fit
Initialize $result$ (Count of bins) and remaining $capacity$ in current bin;
Create an array to store remaining space in bins there can be at most n bins ;
res = 0 ;
$bin_rem[]$;
Place items one by one ;
while *the Bin is empty* **do**
  $i++$;
  Find the first bin that can accommodate $weight[i]$;
  while $j < res$ **do**
    $j++$;
    if $bin_rem[j] >$*size[j]* **then**
      $bin_rem[j]$ = $bin_rem[j]$ - $size[i]$*;*
      break ;
  end
  If no bin could accommodate $size[i]$;
  if j==res **then**
    $bin_rem[res]$ = c - $size[i]$;
    $res++$;
end

Αλγόριθμος 2: Αλγόριθμος First Fit.

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο First Fit. Το πλήθος των κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 3 μέτρα είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 3$, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το δεύτερο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι προς κοπή, τα 5 μέτρα είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 > 5$, άρα κόβεται το 5 από το 7 με υπόλοιπο 2.

Το τρίτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 4 είναι μικρότερο του 10. $10 > 4$, άρα κόβεται το 4 από το 10 με υπόλοιπο 6.

Το τέταρτο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το 7 είναι μικρότερο του 6 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $6 < 7$,

στη συνέχεια συγκρίνεται με το ακριβώς προηγούμενο υπόλοιπο το 2, με $2 < 7$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεται το 7 από το 10 με υπόλοιπο 3.

Το πέμπτο κομμάτι προς κοπή είναι 1 μέτρο, γίνεται έλεγχος αν το 1 είναι μικρότερο του 3 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 1$, άρα κόβεται το 1 από το 3 με υπόλοιπο 2.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 1 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 < 1$, στη συνέχεια το συγκρίνω με προηγούμενο υπόλοιπο που είναι το 3, $3 - 3 = 0$.

Το έβδομο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το 8 είναι μικρότερο του 1. $8 > 1$ άρα περνάει στον επόμενο έλεγχο. Γίνεται έλεγχος αν το 8 είναι μικρότερο του 3. $8 > 3$ άρα περνάει στον επόμενο έλεγχο. Γίνεται έλεγχος αν το 8 είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 < 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 8 είναι μικρότερο του 10. $10 > 8$, άρα κόβεται το 8 από το 10 με υπόλοιπο 2.

Ο αλγόριθμος χρησιμοποίησε συνολικά 4 ράβδους και είχε φύρα 9 μέτρα.

3.5 Ο αλγόριθμος First Fit Decreasing

Είναι ο ίδιος αλγόριθμος με τον ανωτέρω με τη διαφορά ότι τα κομμάτια που θα αφαιρεθούν (k) από το αρχικό υλικό (l), είναι ταξινομημένα. Η πολυπλοκότητα του FFD είναι $nlog(n)$ και αναλύθηκε από τους από τους Johnson et al. [16], βασισμένη σε προηγούμενα αποτελέσματα του Johnson [16].

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο First Fit Decreasing. Το πλήθοςτων κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Με ταξινόμηση η σειρά έχει ως εξής: $8, 7, 5, 4, 3, 3, 1$. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 8 μέτρα είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 8$, άρα κόβεται το 8 από το 10 με υπόλοιπο 2.

Το δεύτερο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι προς κοπή, τα 7 μέτρα είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεται το 7 από το 10 με υπόλοιπο 3.

Το τρίτο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το 5 είναι μικρότερο του 3 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $5 > 3$, άρα στη συνέχεια γίνεται έλεγχος με το προηγούμενο υπόλοιπο το 2. $5 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 5 είναι μικρότερο του 10. $10 > 5$, άρα κόβεται το 5 από το 10 με υπόλοιπο 5.

Το τέταρτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 5 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 < 5$, άρα κόβεται από το 4 το 5 με υπόλοιπο 1.

Το πέμπτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 1 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 1$, άρα στη συνέχεια ελέγχουμε ένα προς ένα τα προηγούμενα υπόλοιπα και αφαιρούμε από το πρώτο που χωράει, $3 - 3 = 0$.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 0 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 0$, άρα

στη συνέχεια ελέγχει τα προηγούμενα υπόλοιπα και εφόσον δε χωράει σε κάποιο ξεκινάει νέα ράβδο. Άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το έβδομο κομμάτι προς κοπή είναι 1 μέτρο, γίνεται έλεγχος αν το 1 είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $1 < 7$, άρα κόβεται από το 1 από το 7 με υπόλοιπο 6.

Ο αλγόριθμος χρησιμοποίησε 4 ράβδους και έχει φύρα 9 μέτρα.

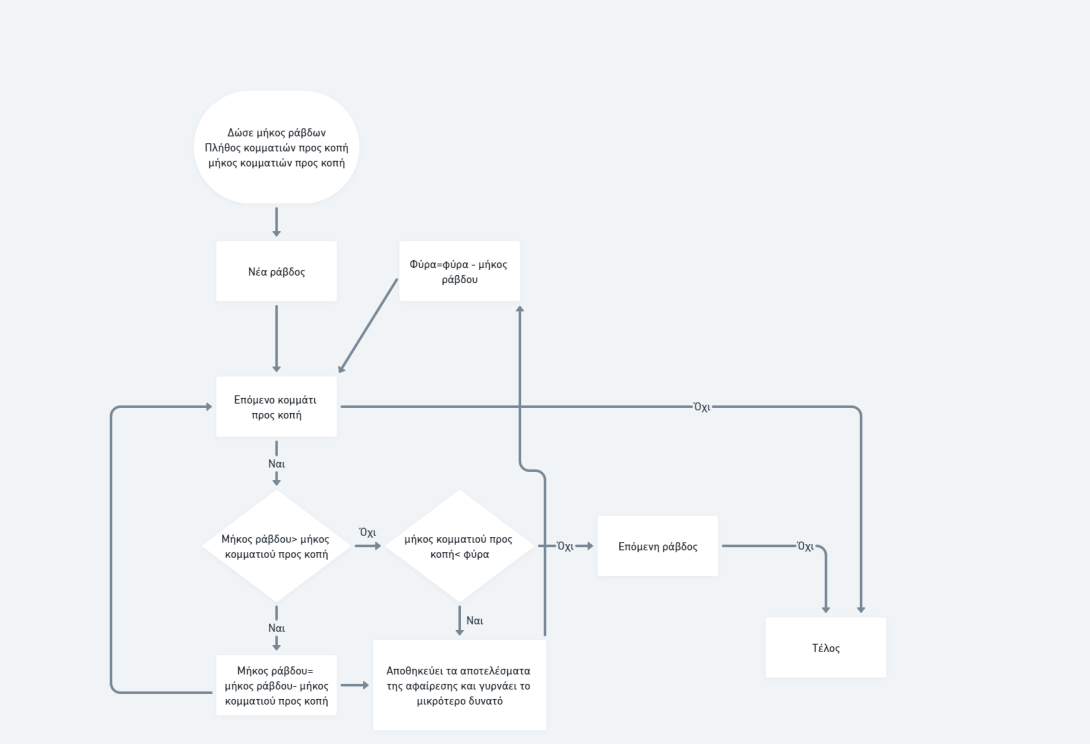
3.6 Ο αλγόριθμος Best Fit

Ο συγκεκριμένος αλγόριθμος επιλέγει κάθε φορά από όλα τα διαθέσιμα κομμάτια που μπορεί να γίνει κοπή αυτό που ταιριάζει περισσότερο (δηλαδή η αφαίρεση του $l - k$ αφήνει το μικρότερο υπόλοιπο), επιλέγει το επόμενο κομμάτι προς κοπή προς τα εμπρός και σε περίπτωση που δε χωράει στη ράβδο προβαίνει σε σύγκριση με όλα τα υπόλοιπα κομμάτια που έχουν περισσέψει σα φύρα, ώστε να βρει την καλύτερη λύση του προβλήματος.

Η πολυπλοκότητα του BF είναι $nlog(n)$ και επιτυγχάνεται χρησιμοποιώντας ένα δέντρο όπου τα φύλλα του αποθηκεύουν την πληροφορία για την υπολειπόμενη χωρητικότητα τον χρησιμοποιημένων ράβδων. Στο δέντρο ισχύουν τα εξής: (1) κάθε κόμβος που δεν είναι φύλλο έχει 2 έως 3 παιδιά, (2) κάθε μονοπάτι από τη ρίζα στο φύλλο έχει το ίδιο μήκος και (3) ετικέτες στους κόμβους επιτρέπουν την αναζήτηση συγκεκριμένου κόστους φύλλου, αναβαθμίζοντας το ή εισάγοντας νέο φύλλο. Η μελέτη για τη συγκεκριμένη δομή δεδομένων μπορεί να βρεθεί στο [1].

Στο Σχήμα 3.3 φαίνεται το διάγραμμα ροής του BF.

Στον Αλγόριθμο 3 δίνεται ο αλγόριθμος σε ψευδοκώδικα του Best Fit.



Σχήμα 3.3: Το διάγραμμα ροής του αλγόριθμου Best Fit

Result: Best Fit
Initialize $result$ (Count of bins) and remaining $capacity$ in current bin;
Create an array to store remaining space in bins there can be at most n bins ;
res = 0 ;
$bin_rem[]$;
Place items one by one ;
while *the Bin is empty* **do**
    $i++$;
    Find the best bin that can accommodate $weight[i]$;
    Initialize minimum space left and index of best bin;
    $min = c+1$, $bi = 0$;
    while $j < res$ **do**
        $j++$;
        if *$bin_rem[j]$ >size[j]$bin_rem[j]$ - $size[i] < min$)* **then**
            $bi = j$;
            $min = bin_rem[j]$ - $size[i]$
    end
    If no bin could accommodate $size[i]$;
    if min==$c+1$ **then**
        $bin_rem[res] = c$ - $size[i]$;
        $res++$;
        $else$ sign the item to best bin ;
        $bin_rem[bi]$ -= $size[i]$;
end

Αλγόριθμος 3: Αλγόριθμος Best Fit.

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο Best Fit. Το πλήθος των κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 3 μέτρα είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 3$, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το δεύτερο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι προς κοπή, τα 5 μέτρα είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 > 5$, άρα κόβεται το 5 από το 7 με υπόλοιπο 2.

Το τρίτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 4 είναι μικρότερο του 10. $10 > 4$, άρα κόβεται το 4 από το 10 με υπόλοιπο 6.

Το τέταρτο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το 7 είναι μικρότερο του 6 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $6 < 7$, άρα στη συνέχεια γίνεται έλεγχος σε όλα τα προηγούμενα στοιχεία και αφαιρείται αυτό που έχει τη μικρότερη διαφορά. Δεν υπάρχει κάποιο καθώς $7 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεται το 7 από το 10 με υπόλοιπο 3.

Το πέμπτο κομμάτι προς κοπή είναι 1 μέτρο, γίνεται έλεγχος αν το 1 είναι μικρότερο του 3. $3 > 1$ με αποτέλεσμα $3 - 1 = 2$.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα γίνεται έλεγχος αν το 3 είναι μικρότερο του 2. $3 > 2$, άρα στη συνέχεια γίνεται έλεγχος σε όλα τα προηγούμενα στοιχεία και αφαιρείται αυτό που έχει τη μικρότερη διαφορά. Γίνεται έλεγχος αν το 3 είναι

μικρότερο ή ίσο του 3, με υπόλοιπο 0 επιλέγεται αυτό, άρα κόβεται το 3 από το 3 με υπόλοιπο 0.

Το έβδομο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το 8 είναι μικρότερο του 6. $8 > 6$, άρα γίνεται έλεγχος αν το 8 είναι μικρότερο του 0. $0 < 8$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 8 είναι μικρότερο του 10. $10 > 8$, άρα κόβεται το 8 από το 10 με υπόλοιπο 2.

Ο αλγόριθμος χρησιμοποίησε 4 ράβδους και έχει συνολική φύρα 9 μέτρα.

3.7 Ο αλγόριθμος Best Fit Decreasing

Είναι ο ίδιος αλγόριθμος με τον ανωτέρω με τη διαφορά ότι τα κομμάτια που θα αφαιρεθούν από το αρχικό υλικό είναι ταξινομημένα. Η πολυπλοκότητα του BFD είναι $nlog(n)$ και αναλύθηκε από τους Johnson et al. [16], βασισμένη σε προηγούμενα αποτελέσματα του Johnson [16].

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο Best Fit Decreasing. Το πλήθος των κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Με ταξινόμηση η σειρά έχει ως εξής: $8, 7, 5, 4, 3, 3, 1$. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 8 μέτρα είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 8$, άρα κόβεται το 8 από το 10 με υπόλοιπο 2.

Το δεύτερο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι προς κοπή, τα 7 μέτρα είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 > 2$ άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεται το 7 από το 10 με υπόλοιπο 3.

Το τρίτο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το 5 είναι μικρότερο του 3 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $5 > 3$, άρα στη συνέχεια γίνεται έλεγχος με το προηγούμενο υπόλοιπο το 2. $5 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 5 είναι μικρότερο του 10. $10 > 5$, άρα κόβεται το 5 από το 10 με υπόλοιπο 5.

Το τέταρτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 5 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 < 5$, άρα κόβεται από το 4 το 5 με υπόλοιπο 1.

Το πέμπτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 1 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 1$, άρα στη συνέχεια ελέγχουμε ένα προς ένα τα προηγούμενα υπόλοιπα και αφαιρούμε το καλύτερο που χωράει, $3 - 3 = 0$.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 0 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 0$, άρα δε χωράει, στη συνέχεια ελέγχει τα προηγούμενα υπόλοιπα και εφόσον δε χωράει σε κάποιο ξεκινάει νέα ράβδο, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το έβδομο κομμάτι προς κοπή είναι 1 μέτρο, γίνεται έλεγχος αν το 1 είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $1 < 7$, άρα κόβεται από το 7 το 1 με υπόλοιπο 6.

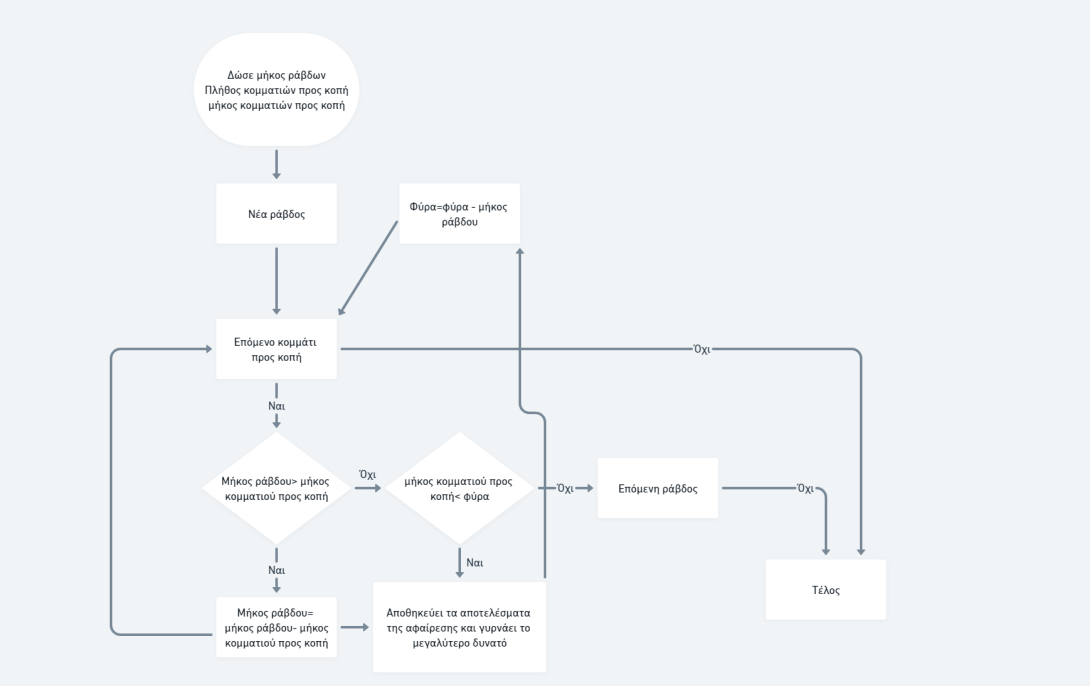
Ο αλγόριθμος χρησιμοποίησε συνολικά 4 ράβδους και είχε φύρα 9 μέτρα.

3.8 Ο αλγόριθμος Worst Fit

Ο συγκεκριμένος αλγόριθμος επιλέγει κάθε φορά από όλα τα διαθέσιμα κομμάτια που μπορεί να γίνει κοπή αυτό που ταιριάζει λιγότερο (δηλαδή η αφαίρεση του $l - k$ να αφήνει το μεγαλύτερο υπόλοιπο), επιλέγει το επόμενο κομμάτι προς κοπή προς τα εμπρός και κάθε φορά κάνει τη σύγκριση με όλα τα διαθέσιμα υπόλοιπα κομμάτια (φύρα) για να βρει τη χειρότερη λύση του προβλήματος. Ουσιαστικά έχει παρόμοια λογική με τον Best Fit, αλλά ακριβώς αντίθετη σύγκριση.

Η πολυπλοκότητα του WF είναι $nlog(n)$ και επιτυγχάνεται χρησιμοποιώντας ένα δέντρο όπου τα φύλλα του αποθηκεύουν την πληροφρία για την υπολειπόμενη χωρητικότητα τον χρησιμοποιημένων ράβδων. Στο δέντρο ισχύουν τα εξής: (1) κάθε κόμβος που δεν είναι φύλλο έχει 2 έως 3 παιδιά, (2) κάθε μονοπάτι από τη ρίζα στο φύλλο έχει το ίδιο μήκος και (3) ετικέτες στους κόμβους επιτρέπουν την αναζήτηση συγκεκριμένου κόστους φύλλου, αναβαθμίζοντας το ή εισάγοντας νέο φύλλο. Η μελέτη για τη συγκεκριμένη δομή δεδομένων μπορεί να βρεθεί στο [1].

Στο Σχήμα 3.4 φαίνεται το διάγραμμα ροής του WF.



Σχήμα 3.4: Το διάγραμμα ροής του αλγόριθμου Worst Fit

Στον Αλγόριθμο 4 δίνεται ο αλγόριθμος σε ψευδοκώδικα του Worst Fit.

Result: Worst Fit
Initialize $result$ (Count of bins) and remaining $capacity$ in current bin;
Create an array to store remaining space in bins there can be at most n bins ;
res = 0 ;
$bin_rem[]$;
Place items one by one ;
while *the Bin is empty* **do**
    $i++$;
    Find the worst bin that can accommodate $weight[i]$;
    Initialize minimum space left and index of worst bin;
    mx = -1, wi = 0 ;
    while $j < res$ **do**
        $j++$;
        if $bin_rem[j]$ >*size[j]*$bin_rem[j]$ - $size[i] > mx$*)* **then**
            wi = j ;
            mx = $bin_rem[j]$ - $size[i]$
    end
    If no bin could accommodate $size[i]$;
    if mx==-1 **then**
        $bin_rem[res]$ = c - $size[i]$;
        $res++$;
        $else$ ssign the item to worst bin ;
        bin$_rem[wi]-=size[i]$
        end

Αλγόριθμος 4: Αλγόριθμος Worst Fit.

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο Best Fit. Το πλήθος των κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 3 μέτρα είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 3$, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

Το δεύτερο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι που είναι 5 μέτρα είναι μικρότερο του 7 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $7 > 5$, άρα κόβεται το 5 από το 7 με υπόλοιπο 2.

Το τρίτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 2 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 > 2$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 4 είναι μικρότερο του 10. $10 > 4$, άρα κόβεται το 4 από το 10 με υπόλοιπο 6.

Το τέταρτο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το 7 είναι μικρότερο του 6 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $6 < 7$ άρα δε χωράει, στη συνέχεια αναζητούμε στα προηγούμενα υπόλοιπα αν χωράει και επιλέγουμε αυτό που αφήνει τη μεγαλύτερη διαφορά. $7 > 2$, άρα δεν υπάρχει κάποιο καθώς στο μοναδικό προηγούμενο υπόλοιπο δε χωράει. Άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεται το 7 από το 10 με υπόλοιπο 3.

Το πέμπτο κομμάτι προς κοπή είναι 1 μέτρο γίνεται έλεγχος αν το 1 είναι μικρότερο του 3. $3 > 1$ με υπόλοιπο $3 - 1 = 2$.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα γίνεται έλεγχος αν το 3 είναι μικρότερο του 2. $3 > 2$, άρα γίνεται έλεγχος αν το 3 είναι είναι μικρότερο ή ίσο του 5. Ταιριάζει, άρα στη συνέχεια συγρίνεται και μετα προηγούμενα υπόλοιπα που δε χωράει, άρα κόβεται το 3 από το 6 με υπόλοιπο 3.

Το έβδομο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το 8 είναι μικρότερο του 0. Μετά γίνεται έλεγχος αν το 8 είναι μικρότερο του 2. Μετά γίνεται έλεγχος αν το 8 είναι μικρότερο του 2 από άλλη ράβδο και τέλος με το 3. $8 > 3$, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 8 είναι μικρότερο του 10. $10 > 8$, άρα κόβεταιτο 8 από το 10 με υπόλοιπο 2.

Ο αλγόριθμος χρησιμοποίησε συνολικά 4 ράβδους και είχε φύρα 9 μέτρα.

3.9 Ο αλγόριθμος Worst Fit Decreasing

Είναι ο ίδιος αλγόριθμος με τον ανωτέρω με τη διαφορά ότι τα κομμάτια που θα αφαιρεθούν από το αρχικό υλικό είναι ταξινομημένα. Η πολυπλοκότητα του WFD είναι $nlog(n)$.

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο Worst Fit Decreasing. Το πλήθος των κομματιών προς κοπή είναι 7 με τιμές $3, 5, 4, 7, 1, 3, 8$ μέτρα. Με ταξινόμηση η σειρά έχει ως εξής: $8, 7, 5, 4, 3, 3, 1$. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Το πρώτο κομμάτι προς κοπή είναι 8 μέτρα, γίνεται έλεγχος αν το πρώτο κομμάτι που είναι 8 μέτρα είναι μικρότερο των 10 μέτρων που είναι το μήκος της ράβδου. $10 > 8$, άρα κόβεται το 8 από το 10 με υπόλοιπο 2.

Το δεύτερο κομμάτι προς κοπή είναι 7 μέτρα, γίνεται έλεγχος αν το δεύτερο κομμάτι προς κοπή. Τα 7 μέτρα είναι μεγαλύτερα των 2 που είχαν περισσέψει από την προηγούμενη κοπή της ράβδου, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 7 είναι μικρότερο του 10. $10 > 7$, άρα κόβεταιτο 7 από το 10 με υπόλοιπο 3.

Το τρίτο κομμάτι προς κοπή είναι 5 μέτρα, γίνεται έλεγχος αν το 5 είναι μικρότερο του 3 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $5 > 3$ και $5 > 2$ από την πρώτη ράβδο δε χωράνε, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 5 είναι μικρότερο του 10. $10 > 5$, άρα κόβεται το 5 από το 10 με υπόλοιπο 5.

Το τέταρτο κομμάτι προς κοπή είναι 4 μέτρα, γίνεται έλεγχος αν το 4 είναι μικρότερο του 5 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $4 < 5$, άρα κόβεται από το 4 το 5 με υπόλοιπο 1.

Το πέμπτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 1 ή του 2 αρχικά που είχαν περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 1$ και $3 > 2$, ενώ στη σύκριση 3 με το 3 ταιριάζει, άρα κόβεται με υπόλοιπο 0.

Το έκτο κομμάτι προς κοπή είναι 3 μέτρα, γίνεται έλεγχος αν το 3 είναι μικρότερο του 0 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $3 > 0$, στη συνέχεια συγκρίνεται με 1 και το 2, όπου δε χωράει, άρα παίρνουμε νέα ράβδο.

Στη νέα ράβδο γίνεται έλεγχος αν το 3 είναι μικρότερο του 10. $10 > 3$, άρα κόβεται το 3 από το 10 με υπόλοιπο 7.

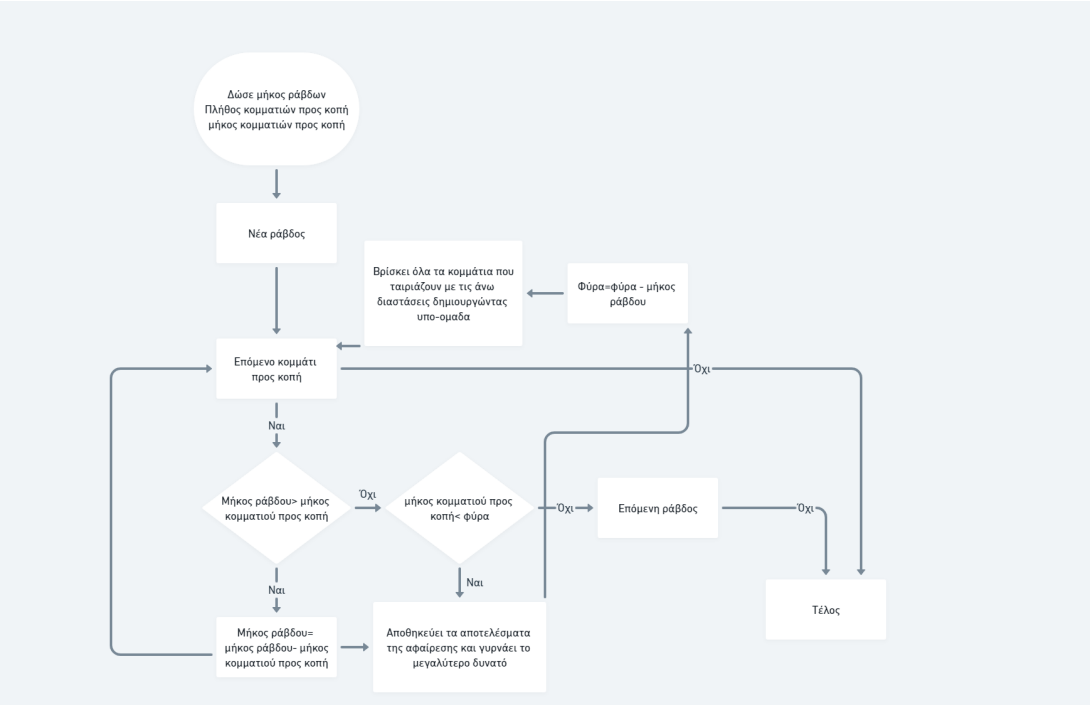
Το έβδομο κομμάτι προς κοπή είναι 1 μέτρο, γίνεται έλεγχος αν το 1 είναι μικρότερο του 3 που είχε περισσέψει από την προηγούμενη κοπή της ράβδου. $1 < 3$, άρα κόβεται από το 3 το 1 με υπόλοιπο 2.

Ο αλγόριθμος χρησιμοποίησε συνολικά 4 ράβδους και είχε φύρα 9 μέτρα.

3.10 Ο αλγόριθμος Full Bin Packing

Ο αλγόριθμος δέχεται σα δεδομένα το σύνολο των αντικειμένων προς κοπή, τα μήκη τους και το μέγεθος του κουτιού ή ράβδου στο οποίο πρέπει να αποθηκεύσει ή κόψει. Στη συνέχεια, βάσει των περιορισμών, δηλαδή του πλήθους των αντικειμένων και του μεγέθους του κουτιού ή ράβδου, ομαδοποιεί τα αντικείμενα ώστε να έχει τη βελτιστη απόδοση προκειμένου να χρησιμοποιεί τον μέγιστο χώρο άρα να έχει την ελάχιστη φύρα. Πρόκειται για offline αλγόριθμο, δηλαδή είναι γνωστά από την αρχή όλα τα κομμάτια προς κοπή. Ο αλγόριθμος full bin Packing είναι αυτός στον οποίο συνδυασμοί αντικειμένων που γεμίζουν μια ράβδο ομαδοποιούνται για να γεμίσουν όσο το δυνατόν λιγότερες ράβδους. Τα υπόλοιπα αντικείμενα τοποθετούνται στη συνέχεια σε άλλες ράβδους.

Στο Σχήμα 3.5 φαίνεται το διάγραμμα ροής του WF.



Σχήμα 3.5: Το διάγραμμα ροής του αλγόριθμου Full Bin Packing

Στον Αλγόριθμο 5 δίνεται ο αλγόριθμος σε ψευδοκώδικα του Full Bin Packing.

Result: Full Bin Packing
$theBin = create\_empty\_bin()$;
$theQueue.enqueue(root\_node)$;
$theupoloipa.enqueue(root\_upoloipa)$;
while *theBin is empty* **do**
  if *Object i fits in group j* **then**
    Pack object i in group j. Break the loop and pack the best object.
    $theBin.enqueue(root.Bin)$;
  if *group j fits in bin* **then**
    Pack group j in group . $theBin.enqueue(root.Bin)$;
  if *Object i did not fit in any available group* **then**
    Create new group and pack object i. $theQueue.enqueue(root.Node)$;
    $theupoloipa.enqueue(root.upoloipa)$
  theFit.push(root)
end

Αλγόριθμος 5: Αλγόριθμος Full Bin Packing.

Στα παρακάτω σχήματα αναλύεται η διαδικασία κοπής με τη μέθοδο Full Bin Packing. Το αρχικό μέγεθος της ράβδου είναι 10 μέτρα.

Έχουμε 7 αντικείμενα μεγέθους $3, 5, 4, 2, 6, 5, 1$ μέτρων. Η ομαδοποίηση έχει ως εξής. Αρχικά ξεκινάει με το 3, στη συνέχεια πρσθέτει το επόμενο στοιχείο το 5, αγνοεί το 4 και προσθέτει το 2 που είναι το επόμενο κομμάτι και χωράει ακριβώς. Ουσιαστικά προσπαθεί να δημιουργήσει ομάδα στοιχείων με το άθροισμα 10. Άρα, η πρώτη υποομάδα είναι τα αντικείμενα με μήκος $3, 5, 2$.

Στη συνέχεια έχουμε το 4 και προσθέτουμε το επόμενο στοιχείο που είναι το 6 και έχουμε και τη δεύτερη υποομάδα με τα στοιχεία $4, 6$.

Τέλος, προσθέτουμε το 5 και το 1 με φύρα 4.

Ο αλγόριθμος χρησιμοποίησε συνολικά 3 ράβδους και είχε φύρα 4 μέτρα.

3.11 Ταξινόμηση

Στους αλγόριθμους που απαιτούν πρώτα την ταξινόμηση των κομματιών προς κοπή, π.χ. BFD, έχει χρησιμοποιηθεί ο αλγόριθμος QuickSort. Ο QuickSort είναι ένας διαίρει και βασίλευε αλγόριθμος, ο οποίος αναπτύχθηκε από τον Tony Hoare [25]. Επιλέγει ένα στοιχείο ως άξονα και χωρίζει τον πίνακα γύρω από τον επιλεγμένο άξονα. Πιο αναλυτικά τα βήματα με αναδρομή είναι τα εξής: 1) Επιλέγουμε ένα στοιχείο p (το οποίο ονομάζουμε pivot - άξονα) και το αφαιρούμε από τον/την

πίνακα/λίστα εισόδου. 2) Χωρίζουμε τον πίνακα/λίστα σε 2 μέρη: S_1 και S_2, όπου το S_1 θα περιέχει όλα τα στοιχεία που είναι μικρότερα από το p και το S_2 όπου περιέχονται όλα τα υπόλοιπα στοιχεία, τα οποία είναι μεγαλύτερα ή ίσα με p. 3) Καλούμε τον αλγόριθμο αναδρομικά στο S_1, παίρνουμε απάντηση στο T_1 και στο S_2 και παίρνουμε απάντηση στο T_2. 4) Επιστρέφουμε τον πίνακα $[T_1, p, T_2]$.

Η μέση πολυπλοκότητα είναι $nlog(n)$, με τη χειρότερη περίπτωση να είναι n^2 [5].

Στον Αλγόριθμο 6 δίνεται ο αλγόριθμος σε ψευδοκώδικα του Quick sort.

Result: Sort
/* low –> Starting index, high –> Ending index */
if *low* < *high* **then**
  pi = partition(arr, low, high); quickSort(arr, low, pi - 1); // Before pi
  quickSort(arr, pi + 1, high); // After pi
theSort.push(root)

Αλγόριθμος 6: Αλγόριθμος Quick sort.

Κεφάλαιο 4

Υπολογιστική μελέτη

4.1 Προβλήματα βέλτιστης κοπής

Το πρόβλημα της βέλτιστης κοπής, όπως προαναφέρθηκε, είναι ένα πραγματικό πρόβλημα που αντιμετωπίζουν κυρίως οι γραμμές παραγωγής με σημαντικά οικονομικά οφέλη στη σωστή χρήση και αξιοποίηση του χώρου των αποθεμάτων και της φύρας. Στο παρόν κεφάλαιο παρατίθενται 40 παραδείγματα. **Οι τιμές που χρησιμοποιήθηκαν είναι πραγματικές τιμές που δόθηκαν από το εργαστάσιο επίπλων με την επωνυμία "Ν. Μάρκου" και οι υπόλοιπες από επιχείρηση με αντικείμενο την εμπορία οικοδομικών υλικών, το οποίο συνεργάζεται με μηχανουργείο στην περιοχή των Λαγυνών Θεσσαλονίκης.** Τα προβλήματα θα παρουσιαστούν συνοπτικά σε δύο πίνακες, στους οποίους κάθε γραμμή θα αποτελεί και ένα διαφορετικό πρόβλημα, με διαφορετικές τιμές τόσο στο μήκος της ράβδου όσο και στο πλήθος και το μήκος των κομματιών. Οι στήλες θα είναι οι 9 αλγόριθμοι, ενώ η τιμή μέσα στον Πίνακα 4.1 αντιπροσωπεύει το πλήθος των ράβδων που χρησιμοποιήθηκαν. Στον Πίνακα 4.2 παρουσιάζονται η φύρα των κομματιών, δηλαδή το άθροισμα από τα υπόλοιπα των ράβδων που δεν έχουν χρησιμοποιηθεί. Στη συνέχεια παρατίθενται στο Παράρτημα τα ως άνω παραδείγματα αναλυτικότερα αναφορικά με τις τιμές των κομματιών προς κοπή, το πλήθος τους και το σταθερό μέγεθος της ράβδου που γίνεται η κοπή.

Τέλος, παρατίθενται επιπλέον 10 προβλήματα από γεννήτρια τυχαίων αριθμών με πλήθος τιμών των κομματιών προς κοπή απο 1000 μέχρι 200 χιλ. Για την εκτέλεση των πειραμάτων έχει χρησιμοποιηθεί ένας προσωπικός υπολογιστής με Windows 10, ο επεξεργαστής $AMD\ RYZEN\ 5\ 3600\ 3.6\ GHz$ και ο compiler $gcc\ version\ 6.3.0$.

4.2 Συνοπτικοί πίνακες αποτελεσμάτων

Ο Πίνακας 4.1 δείχνει τα αποτελέσματα των ράβδων για τους εννιά αλγόριθμους στα 40 προβλήματα με τις πραγματικές τιμές από το εργοστάσιο επίπλων που αναφέρθηκε ανωτέρω, ενώ ο Πίνακας 4.2 δείχνει τα αποτελέσματα της φύρας για τους εννιά αλγόριθμους. Τέλος, παρατίθενται οι πίνακες των μέσων όρων για τις ως άνω τιμές, δηλαδή ο Πίνακας 4.3 με τους μέσους όρους ράβδων και ο Πίνακας 4.4 με τους μέσους όρους φύρας για τα 40 προβλήματα με τις πραγματικές τιμές. Το άθροισμα του πλήθους των ράβδων για τον κάθε αλγόριθμο έχει ως εξής: NF 258, FF 232, BF 232, WF 233, FBP 219, NFD 253, FFD 223, BFD 223, WFD 223. Κατόπιν σύγκρισης και μελέτης, μπορούν να εξαχθούν τα εξής συμπεράσματα. Ο NF έχει χρησιμοποιήσει τις περισσότερες ράβδους με μ.ο. 25.8 ράβδους ανά πρόβλημα. Στη συνέχεια, τη χειρότερη επίδοση έχει ο ίδιος αλγόριθμος με ταξινομημένα δεδομένα με μ.ο. 25.3 ράβδους. Οι WF, FF και BF είναι οι επόμενοι στη σειρά με μ.ο. ο πρώτος 23.3 ράβδους και 22.3 ράβδους οι επόμενοι δύο. Την καλύτερη απόδοση έχει ο FBP με μ.ο. 21.9 ράβδους, ενώ συνεχίζουν τις καλές αποδόσεις οι FFD, WFD και BFD με μ.ο. 22.3 ράβδους. Συνεπώς, οι offline αλγόριθμοι που γνωρίζουν από την αρχή τα κομμάτια προς κοπή και μπορούν να ταξινομηθούν φέρνουν καλύτερα αποτελέσματα από τους αντίστοιχους μη ταξινομημένους. Όσον αφορά στο κριτήριο του αριθμού των ράβδων, είναι εύκολα αντιληπτό πως όσο μικρότερος είναι ο αριθμός των ράβδων που χρησιμοποιήθηκαν, τόσο καλύτερος και πιο αποδοτικός είναι ο αλγόριθμος δεδομένου ότι αξιοποιεί καλύτερα τα δεδομένα του.

Στη συνέχεια θα σχολιαστεί το κριτήριο της φύρας. Οι χρόνοι εκτέλεσης θεωρούνται αμελητέοι καθώς τα προβλήματα είναι μικρά σε μέγεθος και οι πραγματικοί χρόνοι εκτέλεσης είναι κάτω του ενός δευτερολέπτου ανά αλγόριθμο. Η φύρα και οι ράβδοι είναι ανάλογες τιμές καθώς όσες περισσότερες ράβδοι χρησιμοποιηθούν για ένα σταθερό αριθμό κομματιών προς κοπή, με σταθερές αμετάβλητες τιμές για το κάθε κομμάτι, τόσο αυξάνεται η φύρα, συνεπώς αναμένεται η ίδια σειρά απόδοσης με διαφορετικές τιμές. Ο NF έχει την περοσσότερη φύρα, όπως συνέβη και με το κριτήριο του αριθμού των ράβδων, με μ.ο. 110.6 φύρα ανά πρόβλημα. Ο ίδιος αλγόριθμος αλλά με ταξινομημένα τα δεδομένα έχει τη δεύτερη χειρότερη απόδοση με μ.ο. 103.5 φύρα. Οι WF, FF και BF είναι οι επόμενοι στη σειρά με μ.ο. ο πρώτος

58.6 φύρα και 58.5 φύρα οι επόμενοι δύο. Την καλύτερη απόδοση έχει ο FBP με μ.ο. 31.6 φύρα, ενώ καλές αποδόσεις έχουν οι FFD, WFD και BFD με μ.ο. 27.1 φύρα. Συνεπώς, οι offline αλγόριθμοι που γνωρίζουν από την αρχή τα κομμάτια προς κοπή και μπορούν να ταξινομηθούν, φέρνουν καλύτερα αποτελέσματα από τους αντίστοιχους μη ταξινομημένους, όπως συνέβη και με το κριτήριο του αριθμού των ράβδων. Όση λιγότερη φύρα έχει χρησιμοποιηθεί, τόσο καλύτερος και πιο αποδοτικός είναι ο αλγόριθμος καθώς αξιοποιεί καλύτερα τα δεδομένα του. Ο αλγόριθμος που έφερε τις περισσότερες φορές το καλύτερο αποτέλεσμα, όπως φαίνεται από τον Πίνακα 4.5 είναι και τις 40 φορές ο FBP και ο αλγόριθμος με τα χειρότερα είναι ο NF.

A/A	NF	FF	BF	WF	FBP	NFD	FFD	BFD	WFD
1	3	2	2	2	2	3	2	2	2
2	4	3	3	3	3	4	3	3	3
3	4	3	3	3	3	4	3	3	3
4	4	4	4	4	3	4	3	3	3
5	4	3	3	3	3	3	3	3	3
6	4	4	4	4	4	4	4	4	4
7	5	4	4	4	4	4	4	4	4
8	4	3	3	3	3	4	3	3	3
9	6	5	5	5	5	6	5	5	5
10	5	4	4	4	4	5	4	4	4
11	5	4	4	4	4	4	4	4	4
12	5	4	4	4	4	5	4	4	4
13	5	4	4	4	4	4	4	4	4
14	6	5	5	5	5	5	5	5	5
15	3	3	3	3	3	3	3	3	3
16	4	3	3	3	3	4	3	3	3
17	4	3	3	3	3	4	3	3	3
18	5	5	5	5	4	5	4	4	4
19	5	4	4	4	4	5	4	4	4
20	4	4	4	4	4	4	4	4	4
21	6	5	5	5	4	5	5	5	5
22	5	4	4	4	4	5	5	5	5
23	6	5	5	5	5	6	5	5	5
24	7	6	6	6	5	6	5	5	5
25	9	8	8	8	7	9	7	7	7
26	7	6	6	6	6	7	6	6	6
27	9	7	7	7	7	9	7	7	7
28	8	7	7	7	7	9	7	7	7
29	9	8	8	8	8	8	8	8	8
30	11	8	8	9	8	10	8	8	8
31	10	9	9	9	8	10	8	8	8
32	9	8	8	8	8	9	8	8	8
33	10	10	10	10	8	8	8	8	8
34	8	7	7	7	7	8	7	7	7
35	7	7	7	7	6	8	7	7	7
36	5	5	5	5	5	6	5	5	5
37	10	10	10	10	8	11	9	9	9
38	11	10	10	10	9	11	9	9	9
39	11	9	9	9	9	11	9	9	9
40	11	9	9	9	8	11	8	8	8

Πίνακας 4.1: Πίνακας ράβδων στα πραγματικά προβλήματα

A/A	NF	FF	BF	WF	FBP	NFD	FFD	BFD	WFD
1	11	1	1	1	1	11	1	1	1
2	17	7	7	7	7	7	7	7	7
3	15	5	5	5	5	5	5	5	5
4	13	13	13	13	3	13	3	3	3
5	14	4	4	4	4	4	4	4	4
6	9	9	9	9	9	9	9	9	9
7	17	7	7	7	7	7	7	7	7
8	14	4	4	4	4	4	4	4	4
9	14	6	6	6	6	14	6	6	6
10	11	3	3	3	3	11	3	3	3
11	12	4	4	4	4	4	4	4	4
12	10	2	2	2	2	10	2	2	2
13	12	4	4	4	4	4	4	4	4
14	16	8	8	8	8	8	8	8	8
15	110	110	110	110	110	110	110	110	110
16	270	70	70	70	70	270	70	70	70
17	240	30	30	30	30	230	30	30	30
18	260	260	260	260	260	260	60	60	60
19	250	50	50	50	50	250	50	50	50
20	130	130	130	130	130	130	130	130	130
21	440	240	240	240	40	240	240	240	240
22	240	40	40	40	40	240	240	240	240
23	240	40	40	40	40	240	40	40	40
24	420	220	220	220	20	220	20	20	20
25	960	560	560	560	160	960	160	160	160
26	73	33	33	33	33	73	33	33	33
27	11	1	1	1	1	11	1	1	1
28	8	3	3	3	3	13	3	3	3
29	18	8	8	8	8	18	8	8	8
30	35	5	5	15	5	25	5	5	5
31	27	17	17	17	7	27	7	7	7
32	19	9	9	9	9	19	9	9	9
33	7	7	7	7	1	1	1	1	1
34	4.5	1.5	1.5	1.5	1.5	4.5	1.5	1.5	1.5
35	260	260	260	260	90	460	260	260	260
36	120	120	120	120	120	120	120	120	120
37	18	18	18	18	0	28	8	8	8
38	24	14	14	14	4	24	4	4	4
39	28	8	8	8	8	28	8	8	8
40	30	10	10	10	0	30	0	0	0

Πίνακας 4.2: Πίνακας φύρας στα πραγματικά προβλήματα

NF	FF	BF	WF	FBP	NFD	FFD	BFD	WFD
25.8	23.2	23.2	23.3	21.9	25.3	22.3	22.3	22.3

Πίνακας 4.3: Πίνακας μέσων όρων ράβδων στα πραγματικά προβλήματα

NF	FF	BF	WF	FBP	NFD	FFD	BFD	WFD
110.6	58.5	58.5	58.6	31.6	103.5	27.1	27.1	27.1

Πίνακας 4.4: Πίνακας μέσων όρων φύρας στα πραγματικά προβλήματα

NF	NFD	FF	FFD	BF	BFD	FBP	WF	WFD
5	11	29	33	29	34	40	28	33

Πίνακας 4.5: Πίνακας πλήθους πρώτης θέσης στα πραγματικά προβλήματα

Οι τιμές που δημιουργήθηκαν από Γεννήτριες Τυχαίων Αριθμών ακολουθούν περιορισμούς στο εύρος τιμών καθώς δεν είναι δυνατό να είναι το κομμάτι προς κοπή μεγαλύτερο από τη ράβδο, ούτε να είναι αρνητικός αριθμός κάποιος από τους δύο. Στον Πίνακα 4.6 παρατίθενται τα αποτελέσματα των 10 προβλημάτων με κάθε στήλη να είναι ένας αλγόριθμος και η διάταξη των γραμμών ανά τριάδα ένα ξεχωριστό πρόβλημα. Μέσα στην τριάδα είναι οι ράβδοι, η φύρα και ο χρόνος εκτέλεσης του κάθε αλγόριθμου. Τα αποτελέσματα της φύρας είναι ανάλογα με τις ράβδους, όπως και προηγουμένως, καθώς τα ποσά είναι ανάλογα μεταξύ τους. Περισσότεροι ράβδοι σημαίνει περισσότερη φύρα. Τέλος, παρατίθενται οι πίνακες των μέσων όρων για τις ως άνω τιμές. Ο Πίνακας 4.7 με τους μέσους όρους ράβδων, ο Πίνακας 4.8 με τους μέσους όρους φύρας και ο Πίνακας 4.11 με τους συνολικούς χρόνους εκτέλεσης για τα 10 προβλήματα. Από τον Πίνακα 4.7 προκύπτουν τα εξής συμπεράσματα. Ο NF και ο NFD έχουν τα χειρότερα αποτελέσματα στην χρησιμοποίηση ράβδων με μ.ο. 51169 και 51570. Στη συνέχεια, τη χειρότερη απόδοση και μεγαλύτερη χρησιμοποίηση ράβδων έχουν οι WF με μ.ο. 43144 ανά πρόβλημα και ο ταξινομημένος WFD με μ.ο. 42477, ενώ ακολουθούν οι ταξινομημένοι FFD και BFD με μ.ο. 41549 και 40994 αντίστοιχα. Τέλος, τις καλύτερες αποδόσεις στις ράβδους έχουν οι FF και BF με μ.ο. 40975 και 40431 αντίστοιχα και ο καλύτερος των 9 ο FBP με μ.ο. 39907. Η φύρα και οι ράβδοι είναι ανάλογες τιμές καθώς όσες περισσότερες ράβδοι χρησιμοποιηθούν για ένα σταθερό αριθμό κομματιών προς κοπή, με σταθερές αμετάβλητες τιμές για το κάθε κομμάτι, τόσο αυξάνεται η φύρα,

συνεπώς αναμένεται η ίδια σειρά απόδοσης με διαφορετικές τιμές, όπως φαίνεται και από τον Πίνακα 4.8 με τους μέσους όρους φύρας. Ο NF και ο NFD έχουν τα χειρότερα αποτελέσματα στην φύρα με μ.ο. 1874135 και 2456453, αντίστοιχα. Στη συνέχεια τη χειρότερη απόδοση και μεγαλύτερη χρησιμοποίηση ράβδων έχουν οι WF με μ.ο. 838722 ανά πρόβλημα και ο ταξινομημένος WFD με μ.ο. 854359. Ακολουθούν οι ταξινομημένοι FFD και BFD με μ.ο. 701607 και 800727, αντίστοιχα. Τις καλύτερες αποδόσεις στη φύρα έχουν οι FF και BF με μ.ο. 555070 και 701067, αντίστοιχα, και ο καλύτερος των 9 ο FBP με μ.ο. 325115.

Τέλος, στο κριτήριο του χρόνου εκτέλεσης, όπως φαίνεται από τον Πίνακα 4.11, ο πιο γρήγορος είναι ο NF με 28 δευτερόλεπτα και ο NFD με 27 δευτερόλεπτα για το σύνολο των δέκα προβλημάτων, το οποίο είναι λογικό καθώς δεν κάνουν καθόλου συγκρίσεις με την προηγούμενη φύρα. Τρίτος είναι ο FBP με 166 δευτερόλεπτα. Στη συνέχεια, οι πιο γρήγοροι είναι οι FFD με 132 δευτερόλεπτα, ο FBD με 166 δευτερόλεπτα και ο FF με 184 δευτερόλεπτα. Με σημαντική διαφορά ο επόμενος είναι ο WFD με 496 δευτερόλεπτα και οι δύο πιο αργοί είναι ο BF με 1280 δευτερόλεπτα και ο BFD με 2842 δευτερόλεπτα για το σύνολο των 10 προβλημάτων. Ο αλγόριθμος που έφερε τις περισσότερες φορές το καλύτερο αποτέλεσμα, όπως φαίνεται από τον Πίνακα 4.9, με εξαίρεση μία φορά, είναι ο FBP και ο αλγόριθμος με τα χειρότερα αποτελέσματα είναι ο NF.

Μετρήσεις	NF	NFD	FF	FFD	BF	BFD	FBP	WF	WFD
Ράβδοι	709	685	563	685	568	568	544	668	654
Φύρα	17464	15065	2863	12097	2021	12084	960	12063	12009
Χρόνος	0	0	0	0	0	0	0	0	0
Ράβδοι	6986	6718	5420	5529	5562	5529	5320	5701	5536
Φύρα	168733	141934	12132	8964	10309	8964	2172	12892	9037
Χρόνος	0	0	0	0	1	1	0	0	0
Ράβδοι	33396	32173	25453	27855	26439	27832	29153	28947	28754
Φύρα	827428	705129	33127	45319	41302	45319	3329	49648	49648
Χρόνος	4	3	3	3	3	4	3	3	3
Ράβδοι	36622	37047	30155	30847	30164	30484	30111	31826	31047
Φύρα	33155	35281	819	802	796	802	596	963	952
Χρόνος	1	2	6	5	2	2	4	4	5
Ράβδοι	63076	59850	52020	50788	50912	50419	49949	51671	50850
Φύρα	1812185	1489586	506584	496584	536611	526394	499526	546911	526324
Χρόνος	2	3	18	12	104	153	16	40	57
Ράβδοι	42081	47936	36862	40865	32922	40652	39275	40635	40752
Φύρα	9606643	15461644	4387642	5535162	2389791	6535162	5800891	6574162	6527462
Χρόνος	4	3	13	12	654	1545	12	23	48
Ράβδοι	93617	104094	79746	70945	71534	70567	70128	77915	71945
Φύρα	1865645	2913346	478544	306739	314756	291334	306839	388521	336385
Χρόνος	5	6	53	41	294	751	31	76	132
Ράβδοι	66710	64645	51052	54720	53224	5424945	50729	58403	54975
Φύρα	1618379	1411898	52596	481897	92018	471897	20434	641844	471897
Χρόνος	4	3	31	14	71	101	41	62	75
Ράβδοι	108320	104454	82599	87495	86057	85064	82352	87147	93164
Φύρα	1315374	1122075	29323	76241	69932	69294	16970	72294	842493
Χρόνος	6	5	45	33	95	191	42	29	128
Ράβδοι	60177	58099	45884	45768	45937	45884	45517	48536	47099
Φύρα	1476380	1268581	47079	46873	47079	46026	10536	87928	67388
Χρόνος	2	2	15	12	56	94	17	33	48

Πίνακας 4.6: Αποτελέσματα για τα τυχαία προβλήματα

NF	NFD	FF	FFD	BF	BFD	FBP	WF	WFD
51169	51570	40975	41549	40431	40994	39907	43144	42477

Πίνακας 4.7: Πίνακας μέσων όρων ράβδων για τα τυχαία προβλήματα

NF	NFD	FF	FFD	BF	BFD	FBP	WF	WFD
1874135	2456453	555070	701067	325461	800727	325115	838722	854359

Πίνακας 4.8: Πίνακας μέσων όρων φύρας για τα τυχαία προβλήματα

NF	NFD	FF	FFD	BF	BFD	FBP	WF	WFD
0	0	1	0	0	0	9	0	0

Πίνακας 4.9: Πίνακας πλήθους πρώτης θέσης για τα τυχαία προβλήματα

NF	NFD	FF	FFD	BF	BFD	FBP	WF	WFD
7	3	0	0	0	0	0	0	0

Πίνακας 4.10: Πίνακας πλήθους τελευταίας θέσης για τα τυχαία προβλήματα

NF	NFD	FF	FFD	BF	BFD	FBP	WF	WFD
28	27	184	132	1280	2842	166	270	496

Πίνακας 4.11: Πίνακας συνολικών χρόνων των αποτελεσμάτων σε δευτερόλεπτα για τα τυχαίο

4.3 Σύνοψη αποτελεσμάτων

Τα αποτελέσματα ακολουθούν την ίδια περίπου σειρά με τα προβλήματα μικρού μεγέθους. Πιο συγκεκριμένα, ο FBP είναι ο καλύτερος ως προς τις ράβδους και τη φύρα και οι NF και NFD είναι οι χειρότεροι.

Κεφάλαιο 5

Συμπεράσματα και μελλοντικές προεκτάσεις

Σε αυτό το κεφάλαιο γίνεται η μελέτη και σύγκριση των αποτελεσμάτων των αλγόριθμων ως προς τα κριτήρια της φύρας, των ράβδων, της πολυπλοκότητας, του χρόνου εκτέλεσης, της ευκολίας υλοποίησης και ευκολίας κατανόησης του τρόπου λειτουργίας του κάθε αλγόριθμου. Στη συνέχεια, παρατίθενται τα γενικά συμπεράσματα των συγκρίσεων και σχολιάζονται οι μελλοντικές προεκτάσεις.

5.1 Σύγκριση αλγορίθμων

Από την εφαρμογή των αλγορίθμων στα παραδείγματα και τη μελέτη αυτών, προκύπτουν ορισμένα συμπεράσματα. Τα τελευταία μπορούν να κατηγοριοποιηθούν και να αναλυθούν καλύτερα βάσει των παρακάτω κριτηρίων:

1) Με κριτήριο τη συνολική φύρα

Ο αλγόριθμος FBP είναι πιο αποδοτικός στη φύρα σε σχέση με τους υπόλοιπους αλγορίθμους σε όλες τις κλίμακες προβλημάτων με μικρή διαφορά μεταξύ των 3 πρώτων και η οποία ανάλογα μεγαλώνει με το μέγεθος του προβλήματος. Στα μικρά προβλήματα οι offline αλγόριθμοι BFD, FFD και WFD, έρχονται στη δεύτερη θέση με μικρές διαφορές, καθώς αξιοποιούν την πληροφορία των κομματιών προς κοπή και ταξινομούν τα δεδομένα. Εν συνεχεία οι BF, FF και WF έχουν την καλύτερη απόδοση, ενώ ο NFD έχει τη δεύτερη χειρότερη απόδοση, με τον NF να έχει με διαφορά τη χειρότερη απόδοση σε κάθε πρόβλημα. Στα προβλήματα με μεγαλύτερο μέγεθος έχουμε μικρή διαφορά στη σειρά, με τον πρώτο να συνεχίζει να είναι ο FBP, δεύτερος ο BF ο οποίος έχει μικρή διαφορά από τον πρώτο και συνεχίζει στη σειρά

ο ίδιος με ταξινόμηση. Επόμενοι στην απόδοση είναι οι FF με τον ίδιο σε offline αλγόριθμο να ακολουθεί. Ο WFD έχει οριακά χειρότερη απόδοση από τους FFD και WF. Οι NF και NFD να βρίσκονται, όπως και στα μικρά προβλήματα, στην τελευταία θέση στην απόδοση, με σημαντικές διαφορές στην αξιοποίηση της φύρας μεταξύ τους. Συμπερασματικά, η αξιοποίηση προηγούμενων κομματιών προς κοπή και η χρησιμοποίηση της φύρας έχει θετικά αποτελέσματα, τα οποία αυξάνονται με το μέγεθος του προβλήματος.

2) Με κριτήριο το σύνολο των χρησιμοποιηθέντων ράβδων

Η φύρα και οι ράβδοι είναι ανάλογες τιμές καθώς όσες περισσότερες ράβδοι χρησιμοποιηθούν για ένα σταθερό αριθμό κομματιών προς κοπή, με σταθερές τιμές για το κάθε κομμάτι, τόσο αυξάνεται η φύρα, συνεπώς η σειρά των αλγόριθμων όσον αφορά στην χρησιμοποίηση ράβδων είναι η ίδια με αυτή της χρησιμοποίησης φύρας. Ο αλγόριθμος FBP είναι ο πιο αποδοτικός στις ράβδους σε σχέση με τους υπόλοιπους αλγορίθμους σε όλες τις κλίμακες προβλημάτων, με μικρή διαφορά μεταξύ των 3 πρώτων και διαφορά που μεγαλώνει με το μέγεθος του προβλήματος. Στα μικρά προβλήματα οι offline BFD, FFD και WFD με μικρές διαφορές έρχονται στη δεύτερη θέση, καθώς αξιοποιούν την πληροφορία των κομματιών προς κοπή και ταξινομούν τα δεδομένα. Στη συνέχεια οι BF, FF και WF κατά σειρά, έχουν την καλύτερη απόδοση. Ο NFD έχει τη δεύτερη χειρότερη απόδοση, με τον NF να έχει με διαφορά τη χειρότερη σε κάθε πρόβλημα. Οι αλγόριθμοι που αξιοποιούν προηγούμενη φύρα χρησιμοποιούν λιγότερες ράβδους. Στα προβλήματα με μεγαλύτερο μέγεθος έχουμε μικρή διαφορά στη σειρά, με τον πρώτο να συνεχίζει να είναι ο FBP, δεύτερος με μικρή διαφορά ο BF και στη συνέχεια ο ίδιος με ταξινόμηση. Επόμενοι στην αξιοποίηση είναι οι FF με τον ίδιο σε offline αλγόριθμο να ακολουθεί. Ο WFD έχει οριακά χειρότερη απόδοση από τον FFD όπως και τον WF, με τους NF και NFD να βρίσκονται, όπως και στα μικρά προβλήματα, στην τελευταία θέση της απόδοσης στην αξιοποίηση της φύρας. Συμπερασματικά, η αξιοποίηση προηγούμενων κομματιών προς κοπή και η χρησιμοποίηση της φύρας έχει θετικά αποτελέσματα στην αξιοποίηση λιγότερων ράβδων.

3) Με κριτήριο τη πολυπλοκότητα

Ο Πίνακας 5.1 δείχνει την πολυπλοκότητα των αλγορίθμων.

Σε αυτό το σημείο να αναφερθεί ότι η πολυπλοκότητα των αλγορίθμων που

NF	FF	BF	WF	FBP	NFD	FFD	BFD	WFD
$O(n)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$	$O(nlogn)$

Πίνακας 5.1: Συνοπτικός πίνακας πολυπλοκότητας αλγορίθμων

αναλύθηκε είναι η καλύτερη δυνατή που μπορεί να υλοποιηθεί και δεν αφορά την υλοποίηση που επιτεύχθηκε στα πλαίσια της παρούσας διπλωματικής.

4) Με κριτήριο τον χρόνο εκτέλεσης

Στο κριτήριο του χρόνου εκτέλεσης, ο πιο γρήγορος είναι ο NF με 28 δευτερόλεπτα και ο NFD με 27 δευτερόλεπτα για το σύνολο των δέκα προβλημάτων, το οποίο είναι λογικό καθώς δεν κάνουν καθόλου συγκρίσεις με τις προηγούμενες ράβδους για να αξιοποιήσουν την φύρα. Τρίτος κατά σειρά με αύξοντα χρόνο εκτέλεσης είναι ο FFD με 132 δευτερόλεπτα, ενώ στη συνέχεια οι πιο γρήγοροι είναι οι FBP με 166 δευτερόλεπτα, ο FF με 184 δευτερόλεπτα και ο WF με 270 δευτερόλεπτα. Με σημαντική διαφορά επόμενος είναι ο WFD με 496 δευτερόλεπτα και οι δύο πιο αργοί αλγόριθμοι εκτέλεσης είναι ο BF με 1280 δευτερόλεπτα και ο BFD με 2842 δευτερόλεπτα για το σύνολο των 10 προβλημάτων. Στα πρώτα 40 προβλήματα με τις πραγματικές τιμές από το εργαστάσιο επίπλων και το μηχανουργείο που αναφέρθηκαν ανωτέρω οι χρόνοι εκτέλεσης είναι μικρότεροι του ενός δευτερολέπτου και δεν παρουσιάζονται καθώς θεωρούνται αμελητέοι.

5) Με κριτήριο την ευκολία υλοποίησης και τον βαθμό δυσκολίας στην κατανόηση του τρόπου λειτουργίας του

Ο πιο εύκολος τόσο στην υλοποίηση όσο και στην κατανόηση είναι ο NF καθώς δεν αξιοποιεί προηγούμενες ράβδους με φύρα. Για τον ίδιο λόγο ο NFD έρχεται δεύτερος και ακολουθούν κατά σειρά οι FF, BF και ο WF, οι οποίοι αξιοποιούν τη φύρα με απλό τρόπο. Ξεκινώντας από τον FF ο οποίος βρίσκει και αξιοποιεί την πρώτη ράβδο που του λύνει το πρόβλημα και θεωρεί αυτό ως βελτιστη λύση. Στη συνέχεια οι BF και WF με παρόμοια υλοποίηση και τρόπο λειτουργίας οι οποίοι με συγκρίσεις όλων των ράβδων με φύρα αναζητούν αυτόν που αφήνει μικρότερο (BF) ή μεγαλύτερο (WF) υπόλοιπο, με τον πιο δύσκολο να είναι ο FBP καθώς χωρίζει σε υπο-ομάδες που γεμίζουν όσο καλύτερα γίνεται τις ράβδους και στη συνέχεια κάνει αναζήτηση για παρόμοια κομμάτια και συνεχίζει μέχρι να τελειώσουν οι υποομάδες για να φτάσει στον στόχο που θεωρεί ως βέλτιστη λύση. Η ταξινόμηση ποικίλλει ανάλογα με τον τρόπο που υλοποιείται ο αλγόριθμος. Εν προκειμένω,

χρησιμοποιήθηκε ο quicksort [25] αλλά σε γενικότερο πλαίσιο δεν αλλάζει τον αλγόριθμο ως προς την ευκολία υλοποίησης ή τη δυσκολία κατανόησης του τρόπου λειτουργίας του, καθώς χρησιμοποιείται κατευθείαν στα δεδομένα.

5.2 Γενικά συμπεράσματα

Ανάλογα με το μέγεθος του πίνακα παρουσιάστηκαν μικρές αποκλίσεις σε όλους τους αλγόριθμους και στη διάταξη μεταξύ τους. Η ταξινόμηση είχε μαγάλη σημασία σε μικρούς πίνακες αλλά χανόταν σε μεγάλους. Άρα η γνώση των κομματιών και του μεγέθους τους από την αρχή δε βεβαιώνει καλύτερα αποτελέσματα πάντα, αλλά πιο σημαντικός είναι ο τρόπος κοπής που μεταφράζεται σε καλύτερη απόδοση, το οποίο παρουσιάζεται στον Πίνακα 4.3 και στους Πίνακες 4.4 και 4.6. Αν το ζητούμενο του προβλήματος είναι ο χρόνος εκτέλεσης, τότε επιλέγουμε κάποιον από τους πρώτους που αναφέρονται στο κριτήριο χρόνοι εκτέλεσης. Ενώ αν αναζητούμε κάποιον που αξιοποιεί καλύτερα τη φύρα ψάχνουμε κάποιον με κριτήριο τη φύρα ή τις ράβδους. Αντίστοιχα αν το ζητούμενο είναι η ευκολία ακολουθούμε ανάλογο τρόπο σκέψης για την εύρεση της λύσης στο πρόβλημα. Τέλος, αν το ζητούμενο είναι η πολυπλοκότητα τότε πρέπει να απαντηθούν τα ερωτήματα αν ψάχνουμε για μέση πολυπλοκότητα, καλύτερη ή χειρότερη περίπτωση.

5.3 Μελλοντικές εξελίξεις

Σε πρόσφατες μελέτες έγινε με επιτυχία χρήση τεχνικών που προέρχονται από το χώρο της Τεχνητής Νοημοσύνης. Στο άμεσο μέλλον, αναμένονται εξελίξεις αναφορικά με την υλοποίηση των αλγορίθμων με τις ως άνω τεχνικές, ενώ συγχρόνως έχει αρχίσει να αλλάζει η δομή της σχεδίασης αλγορίθμων. Πιο συγκεκριμένα, οι αλγόριθμοι δεν είναι πλέον στατικοί αλλά εξελίσσονται, υπό την έννοια ότι θα βελτιώνονται αυτόματα και θα αυτο-διορθώνεται η συμπεριφορά τους. Σε αυτό το σημείο να αναφερθεί η αυξανόμενη τάση του αριθμού των δημοσιεύσεων εντός των τελευταίων 20 ετών, όπως προκύπτει από τις κύριες βάσεις δεδομένων του Σχήματος 5.1. Ειδικότερα, ο συνολικός αριθμός των δημοσιεύσεων σχετικά με το πρόβλημα βέλτιστης κοπής τα τελευταία 20 χρόνια ήταν 1410, 774 και 575 σύμφωνα με τις βάσεις δεδομένων Scholar, Scopus and WoS, αντίστοιχα [15].

Απεικόνιση 5.1: Τάση του αριθμού των δημοσιεύσεων

Σχήμα 5.1: Συνολικός αριθμός δημοσιεύσεων σύμφωνα με τις βάσεις δεδομένων Scholar, Scopus and WoS

Κεφάλαιο 6

Παράρτημα

Οι τιμές των κομματιών προς κοπή και η σταθερή τιμή της ράβδου από την οποία γίνεται η κοπή για τα πραγματικά παραδείγματα τα οποία δεδομένα αντλήθηκαν από το εργαστάσιο επίπλων με την επωνυμία "Ν. Μάρκου" και οι υπόλοιπες από επιχείρηση με αντικείμενο την εμπορία οικοδομικών υλικών το οποίο συνεργάζεται με μηχανουργείο στην περιοχή των Λαγυνών Θεσσαλονίκης παρουσιάζονται παρακάτω.

1ο πρόβλημα: 6 τιμές $(2, 6, 3, 2, 5, 1)$ με τιμή ράβδου 10.

2ο πρόβλημα: 6 τιμές $(3, 5, 4, 8, 1, 2)$ με τιμή ράβδου 10.

3ο πρόβλημα έχουμε 6 τιμές $(3, 1, 8, 7, 4, 2)$ με τιμή ράβδου 10.

4ο πρόβλημα έχουμε 6 τιμές $(7, 4, 2, 6, 3, 5)$ με τιμή ράβδου 10.

5ο πρόβλημα έχουμε 7 τιμές $(2, 3, 5, 3, 2, 8, 3)$ με τιμή ράβδου 10.

6ο πρόβλημα έχουμε 6 τιμές $(7, 5, 6, 3, 3, 1, 4, 2)$ με τιμή ράβδου 10.

7ο πρόβλημα έχουμε 7 τιμές $(4, 7, 9, 2, 3, 4, 4)$ με τιμή ράβδου 10.

8ο πρόβλημα έχουμε 7 τιμές $(3, 4, 8, 3, 2, 1, 5)$ με τιμή ράβδου 10.

9ο πρόβλημα έχουμε 7 τιμές $(4, 5, 7, 3, 7, 2, 6)$ με τιμή ράβδου 8.

10ο πρόβλημα έχουμε 7 τιμές $(6, 3, 4, 7, 2, 6, 1)$ με τιμή ράβδου 8.

11ο πρόβλημα έχουμε 7 τιμές $(6, 3, 7, 3, 5, 2, 2)$ με τιμή ράβδου 8.

12ο πρόβλημα έχουμε 7 τιμές $(6, 3, 5, 7, 2, 6, 1)$ με τιμή ράβδου 8.

13ο πρόβλημα έχουμε 8 τιμές $(6, 3, 2, 7, 1, 4, 3, 2)$ με τιμή ράβδου 8.

14ο πρόβλημα έχουμε 8 τιμές $(7, 2, 4, 3, 6, 1, 7, 2)$ με τιμή ράβδου 8.

15ο πρόβλημα έχουμε 8 τιμές $(60, 70, 60, 70, 50, 60, 40, 80)$ με τιμή ράβδου 200.

16ο πρόβλημα έχουμε 8 τιμές $(120, 90, 40, 50, 40, 30, 100, 60)$ με τιμή ράβδου 200.

17ο πρόβλημα έχουμε 8 τιμές $(100, 60, 80, 100, 40, 140, 30, 20)$ με τιμή ράβδου 200.

18ο πρόβλημα έχουμε 8 τιμές $(180, 60, 60, 120, 100, 50, 150, 20)$ με τιμή ράβδου 200.

19ο πρόβλημα έχουμε 10 τιμές $(160, 50, 40, 90, 30, 40, 140, 20, 130, 50)$ με τιμή ράβδου 200.

20ο πρόβλημα έχουμε 12 τιμές $(130, 40, 50, 60, 80, 30, 10, 70, 80, 60, 20, 40)$ με τιμή ράβδου 200.

21ο πρόβλημα έχουμε 12 τιμές $(60, 30, 70, 80, 20, 120, 40, 80, 130, 40, 70, 20)$ με τιμή ράβδου 200.

22ο πρόβλημα έχουμε 12 τιμές $(120, 60, 30, 60, 70, 150, 20, 40, 30, 60, 100, 20)$ με τιμή ράβδου 200.

23ο πρόβλημα έχουμε 14 τιμές $(130, 40, 70, 80, 60, 40, 20, 180, 10, 40, 60, 90, 110, 30)$ με τιμή ράβδου 200.

24ο πρόβλημα έχουμε 14 τιμές $(40, 80, 120, 60, 50, 40, 170, 50, 80, 20, 40, 20, 80, 130)$ με τιμή ράβδου 200.

25ο πρόβλημα έχουμε 14 τιμές $(150, 130, 180, 310, 140, 180, 250, 330, 260, 50, 100, 140, 250, 60)$ με τιμή ράβδου 400.

26ο πρόβλημα έχουμε 14 τιμές $(16, 18, 22, 6, 19, 32, 9, 16, 25, 6, 7, 8, 12, 11)$ με τιμή ράβδου 40.

27ο πρόβλημα έχουμε 14 τιμές $(1, 2, 3, 3, 2, 4, 2, 3, 2, 3, 2, 2, 3, 1)$ με τιμή ράβδου 5.

28ο πρόβλημα έχουμε 14 τιμές $(3, 3, 2, 1, 4, 2, 3, 2, 1, 3, 2, 4, 1, 1)$ με τιμή ράβδου 5.

29ο πρόβλημα έχουμε 16 τιμές $(3, 6, 2, 5, 7, 4, 6, 3, 2, 1, 7, 2, 9, 8, 3, 4)$ με τιμή ράβδου 10.

30ο πρόβλημα έχουμε 16 τιμές$(6, 5, 3, 8, 3, 2, 6, 4, 5, 7, 4, 2, 8, 3, 5, 4)$με τιμή ράβδου 10.

31ο πρόβλημα έχουμε 16 τιμές $(8, 3, 4, 6, 7, 2, 8, 7, 6, 1, 5, 3, 1, 2, 6, 4)$ με τιμή ράβδου 10.

32ο πρόβλημα έχουμε 16 τιμές$(6, 5, 3, 8, 4, 2, 6, 5, 4, 3, 2, 4, 3, 7, 1, 8)$με τιμή ράβδου 10.

33ο πρόβλημα έχουμε 16 τιμές $(1, 1, 5, 2, 2, 5, 2, 5, 1, 0, 5, 1, 5, 0, 5, 0, 5, 2, 2, 5, 1, 5, 1, 5)$ με τιμή ράβδου 3.

34ο πρόβλημα έχουμε 16 τιμές $(0, 5, 2, 1, 0, 5, 1, 5, 1, 5, 1, 5, 0, 5, 2, 1, 5, 1, 5, 1, 0, 5, 2, 1, 1)$ με τιμή ράβδου 3.

35ο πρόβλημα έχουμε 18 τιμές $(180, 180, 60, 70, 80, 90, 180, 170, 80, 80, 15, 15, 60, 50,$

$80, 70, 15, 15$) με τιμή ράβδου 250.

36ο πρόβλημα έχουμε 18 τιμές $(180, 50, 30, 30, 80, 20, 20, 10, 80, 90, 40, 70, 180, 60, 70, 180, 60, 70, 15, 15, 90)$ με τιμή ράβδου 250.

37ο πρόβλημα έχουμε 20 τιμές $(5, 4, 3, 6, 7, 2, 8, 3, 4, 5, 3, 2, 5, 6, 2, 1, 7, 2, 2, 5)$ με τιμή ράβδου 10.

38ο πρόβλημα έχουμε 20 τιμές $(8, 3, 5, 6, 4, 2, 3, 4, 3, 2, 6, 5, 9, 2, 8, 3, 1, 5, 4, 3)$ με τιμή ράβδου 10.

39ο πρόβλημα έχουμε 20 τιμές $(3, 4, 8, 2, 5, 6, 7, 4, 3, 2, 3, 4, 2, 6, 7, 4, 5, 2, 3, 2)$ με τιμή ράβδου 10.

40ο πρόβλημα: 20 τιμές $(5, 6, 7, 4, 3, 2, 5, 1, 6, 2, 1, 3, 1, 6, 7, 8, 6, 2, 3, 2)$ με τιμή ράβδου 10.

Βιβλιογραφία

[1] V Aho Alfred, E Hopcroft John, D Ullman Jeffrey, V Aho Alfred, H Bracht Glenn, D Hopkin Kenneth, C Stanley Julian, Brachu Jean-Pierre, Brown A Samler, Brown A Peter, et al. *Data structures and algorithms*. USA: Addison-Wesley, 1983.

[2] Brenda S Baker and Edward G Coffman, Jr. A tight asymptotic bound for next-fit-decreasing bin-packing. *SIAM Journal on Algebraic Discrete Methods*, 2(2):147–152, 1981.

[3] BSC Campello, CTLS Ghidini, AOC Ayres, and WA Oliveira. A residual recombination heuristic for one-dimensional cutting stock problems. *Top*, pages 1–27, 2021.

[4] Alan A Farley. Mathematical programming models for cutting-stock problems in the clothing industry. *Journal of the Operational Research Society*, 39(1):41–53, 1988.

[5] James Allen Fill and Svante Janson. Quicksort asymptotics. *Journal of Algorithms*, 44(1):4–28, 2002.

[6] Michael R Garey, Ronald L Graham, David S Johnson, and Donald Ervin Knuth. Complexity results for bandwidth minimization. *SIAM Journal on Applied Mathematics*, 34(3):477–495, 1978.

[7] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[8] Paul C Gilmore and Ralph E Gomory. A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859, 1961.

[9] Dušan Gordić, Milun Babić, Dubravka Jelić, Davor Končalović, and Vladimir Vukašinović. Integrating energy and environmental management in wood furniture industry. *The Scientific World Journal*, 2014, 2014.

[10] Maria Cristina N Gramani and Paulo M França. The combined cutting stock and lot-sizing problem in industrial processes. *European Journal of Operational Research*, 174(1):509–521, 2006.

[11] MCN Gramani, PM França, and MN Arenales. A lagrangian relaxation approach to a coupled lot-sizing and cutting stock problem. *International Journal of Production Economics*, 119(2):219–227, 2009.

[12] Robert W Haessler. A heuristic programming solution to a nonlinear cutting stock problem. *Management Science*, 17(12):B–793, 1971.

[13] Robert W Haessler and Paul E Sweeney. Cutting stock problems and solution procedures. *European Journal of Operational Research*, 54(2):141–150, 1991.

[14] LC Hendry, KK Fok, and KW Shek. A cutting stock and scheduling problem in the copper industry. *Journal of the Operational Research Society*, 47(1):38–47, 1996.

[15] Manuel Iori, Vinicius Loti de Lima, Silvano Martello, Flavio Miyazawa, and Michele Monaci. Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, 289, 07 2020.

[16] David S. Johnson, Alan Demers, Jeffrey D. Ullman, Michael R Garey, and Ronald L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on computing*, 3(4):299–325, 1974.

[17] Aline AS Leao, Marcos M Furlan, and Franklina MB Toledo. Decomposition methods for the lot-sizing and cutting-stock problems in paper industries. *Applied Mathematical Modelling*, 48:250–268, 2017.

[18] Sigrid Lise Nonås and Anders Thorstenson. A combined cutting-stock and lot-sizing problem. *European Journal of Operational Research*, 120(2):327–342, 2000.

[19] Sônia Cristina Poltroniere, Silvio Alexandre Araujo, and Kelly Cristina Poldi. Optimization of an integrated lot sizing and cutting stock problem in the paper industry. *TEMA (São Carlos)*, 17:305–320, 2016.

[20] Bastian Rieck. Basic analysis of bin-packing heuristics. *arXiv preprint arXiv:2104.12235*, 2021.

[21] Elsa Silva, F Alvelos, and JM Valério de Carvalho. Integrating two-dimensional cutting stock and lot-sizing problems. *Journal of the Operational Research Society*, 65(1):108–123, 2014.

[22] Matheus Vanzela, Gislaine Mara Melega, Socorro Rangel, and Silvio Alexandre de Araujo. The integrated lot sizing and cutting stock problem with saw cycle constraints applied to furniture production. *Computers & Operations Research*, 79:148–160, 2017.

[23] Gerhard Wäscher and Thomas Gau. Heuristics for the integer one-dimensional cutting stock problem: A computational study. *Operations-Research-Spektrum*, 18(3):131–144, 1996.

[24] Xian-qing Xiong, Wei-juan Guo, Lu Fang, Min Zhang, Zhi-hui Wu, Rong Lu, and Tetsuo Miyakoshi. Current state and development trend of chinese furniture industry. *Journal of Wood Science*, 63(5):433–444, 2017.

[25] Neelam Yadav and Sangeeta Kumari. Sorting algorithms. *Int. Res. J. Eng. Technol*, 3:425–446, 2016.