ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΧΗΜΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΑΝΑΛΥΣΗΣ, ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΔΙΕΡΓΑΣΙΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

ΕΥΡΕΤΙΚΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΤΟΥ ΜΟΝΟΔΙΑΣΤΑΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΠΑΚΕΤΟΠΟΙΗΣΗΣ

Διπλωματική Εργασία
Κωνσταντίνος Παππάς Ζουντουρίδης

Επιβλέπων Καθηγητής:
Χρήστος Κυρανούδης

Αθήνα 2014

Περίληψη

Στην παρούσα διπλωματική εργασία μελετάται το πρόβλημα της πακετοποίησης σε κάδους (bin packing problem) σε μία διάσταση. Αποτελεί ένα πρόβλημα με NP – hard βαθμό πολυπλοκότητας, του οποίου οι εφαρμογές παρατηρούνται σε πολλές καταστάσεις της καθημερινής ζωής.

Αρχικά αναφέρονται τα διαφορετικά είδη αλγορίθμων που αναπτύσσονται ακόμα και σήμερα για την επίλυση του προβλήματος. Υπάρχουν πολλές περιπτώσεις πακετοποίησης με διαφοροποιήσεις στις συνθήκες και τους περιορισμούς της καθεμίας, άρα είναι σχεδόν αδύνατη η δημιουργία ενός γενικού αλγορίθμου για την επίλυση όλων των περιπτώσεων. Είναι ανάγκη να εξεταστούν οι παράμετροι της κάθε περίστασης και να επιλέγεται το κατάλληλο είδος αλγόριθμου για την αντιμετώπιση της.

Για το μονοδιάστατο πρόβλημα πακετοποίησης επιλέγονται οι ευρετικές μεθόδοι, με τους κώδικες τους διαμορφωμένους σε MATLAB. Αναπτύσσεται η βασική λογική της καθεμίας, με τη βοήθεια των γραφικών αναπαραστάσεων που παρέχει η MATLAB. Στη συνέχεια επιλέγονται κάποιες περιπτώσεις προβλημάτων και παρουσιάζονται τα αποτελέσματα της κάθε μεθόδου για τη σύγκριση αυτών.

Τέλος, γίνεται αναφορά στα συμπεράσματα που προκύπτουν από τη θεωρητική ανασκόπηση αλλά και τα πρακτικά αποτελέσματα που παρέχουν οι ευρετικές μέθοδοι για την επίλυση του μονοδιάστατου προβλήματος πακετοποίησης.

Abstract

In the present thesis the one dimensional bin packing problem is considered. It is a problem with NP – hard class complexity with various applications into everyday life.

To start with, the variety of algorithms that are developed up to today to deal with this problem is presented. Due to the many different cases and constraints the bin packing problem has, the composition of a general algorithm that will solve all the different circumstances of the problem is almost impossible. All the parameters of each circumstance must be examined in order to choose the appropriate type of algorithm to solve the problem.

The heuristic methods developed in MATLAB are chosen for the solution of the one dimensional bin packing problem. The main concept of each method is presented, with the graphical illustrations that the MATLAB environment has greatly helping for the understanding of each concept. Subsequently, some problem cases are brought up in order to display and compare the results that each method concludes.

To sum up, the conclusions that are excluded from the theoretical view are delivered, with annotations to the results that are produced from the heuristic methods during the resolve of the one dimensional bin packing problem.

Περιεχόμενα

Περιεχόμενα Πινάκων

Περιεχόμενα Εικόνων

Κεφάλαιο 1

Το πρόβλημα πακετοποίησης

1.1 Εισαγωγή

Η διπλωματική αυτή εργασία έχει ως βασικό αντικείμενο μελέτης την ανάπτυξη ενός αλγορίθμου για την ακριβή διατύπωση του προβλήματος πακετοποίησης (bin packing problem) και την περιγραφή των μεθόδων επίλυσής του. Μια σύντομη περιγραφή του προβλήματος πακετοποίησης είναι η εξής: Θεωρείται ως δεδομένο τυχαίος αριθμός από διαθέσιμα αντικείμενα διαφορετικού μεγέθους και ο σκοπός είναι η αποθήκευση αυτών σε έναν πεπερασμένο αριθμό δοχείων ή κάδων (bins) με συγκεκριμένες διαστάσεις, έτσι ώστε να ελαχιστοποιηθεί ο αριθμός των κάδων αυτών. Το πρόβλημα πακετοποίησης σε κάδους αποτελεί παρακλάδι μιας μεγαλύτερης κατηγορίας προβλημάτων κοπής και πακετοποίησης (cutting and packing problems) (Dyckoff, 1990).

Σε εργασιακό περιβάλλον, λόγω της εντατικοποίησης των ρυθμών της σημερινής εποχής, είναι εύκολο να αναλογιστεί κανείς την ανάγκη ανάπτυξης μεθόδων που έχουν ως βασικό σκοπό την εξοικονόμηση χρόνου και το οικονομικό όφελος. Επομένως, το πρόβλημα πακετοποίησης ανήκει σε μια βασική κατηγορία προβλημάτων στον τομέα της επιχειρησιακής έρευνας και της εφοδιαστικής διαχείρισης. Καθώς το πρόβλημα αυτό απασχολεί τους ερευνητές τις τελευταίες δεκαετίες, η αρχική του προσέγγιση γινόταν χειρωνακτικά, μια επίπονη εργασία που συχνά απαιτούσε την συμβολή αρκετών ατόμων για την ολοκλήρωσή της και πολλές φορές αρκετά μεγάλα χρονικά διαστήματα. Ακόμη, η λύση που προκύπτει με αυτόν τον τρόπο είναι αμφισβητούμενης ποιότητας και δεν εγγυάται ότι θα καλύπτει όλες τις παραμέτρους του προβλήματος καθώς δεν υπάρχει μια συστηματική διαδικασία για τον έλεγχο της χειρωνακτικής λύσης.

Είναι πλέον κατανοητό ότι υπήρχε ανάγκη για διαφορετική προσέγγιση του προβλήματος και αυτό έγινε στις αρχές της δεκαετίας του ’70. Τότε οι ερευνητές στράφηκαν στην κατασκευή αλγορίθμων, που λαμβάνουν υπόψη τις χαρακτηριστικές παραμέτρους για κάθε πρόβλημα και προσφέρουν αν όχι τη βέλτιστη, μια αρκετά ικανοποιητική λύση σε πολύ μικρό χρονικό διάστημα.

Το πρόβλημα πακετοποίησης μοντελοποιείται ως πρόβλημα ακέραιου γραμμικού προγραμματισμού (integer linear programming, ILP). Σε αυτήν την κατηγορία ανήκει οποιοδήποτε πρόβλημα αφορά μαθηματική βελτιστοποίηση με ακέραιους παράγοντες και λύνεται με γραμμικές συναρτήσεις και συνθήκες (Schrijver, 1998). Το αρχικό πρόβλημα εμφανίζεται σε μία διάσταση (π.χ. μέγεθος), αλλα πλέον υπάρχουν πολλές παραλλαγές, σε δύο, τρεις ή και περισσότερες διαστάσεις. Επίσης, η βέλτιστη λύση έχει αναζητηθεί με διάφορους τύπους αλγορίθμων (ευρετικοί, εξελικτικοί κ.τ.λ). Στην εργασία αυτή θα μελετηθεί το μονοδιάστατο πρόβλημα πακετοποίησης συνθέτοντας έναν ευρετικό αλγόριθμο για την επίλυσή του.

1.2 Εφαρμογές του προβλήματος

Στο μονοδιάστατο πρόβλημα πακετοποίησης τα αντικείμενα έχουν μία διάσταση. Αυτή μπορεί να είναι το βάρος, το μέγεθος, το κόστος ή ο χρόνος. Ο όρος “κάδος” είναι στην ουσία ένας γενικός όρος όπου μπορεί να αντιπροσωπεύει την κυριολεκτική του σημασία, δηλαδή έναν αποθηκευτικό χώρο, έναν χώρο στον χρόνο όταν το πρόβλημα αφορά χρονικό προγραμματισμό ή μια επιφάνεια όταν χρησιμοποιείται στην βιομηχανία κοπής γυαλιού. Το πρόβλημα έχει πολλές πρακτικές εφαρμογές: Στην περίπτωση των μεταφορικών πλοίων (container), ο στόχος είναι να χρησιμοποιηθούν όσα λιγότερα πλοία γίνεται για να μεταφερθεί το εμπόρευμα, δίχως να υπερβαίνεται η χωρητικότητα αυτών. Ένα άλλο παράδειγμα είναι η εγκατάσταση σωλήνων ή καλωδίων σε χώρους δεδομένου μήκους. Εδώ το ζητούμενο είναι η χρήση όσο λιγότερων σωλήνων ή καλωδίων γίνεται για την ομαλή λειτουργία και την εξοικονόμηση υλικού. Επίσης, στον προγραμματισμό παραγωγής μιας βιομηχανίας, όπου εργασίες διαφορετικής διάρκειας πρεπεί να διανεμηθούν έτσι ώστε να χρησιμοποιηθούν οι λιγότερες μηχανές.

Υπάρχουν πολλές παραλλαγές και επεκτάσεις του προβλήματος πακετοποίησης. Η πιο προφανής είναι το δυσδιάστατο πρόβλημα πακετοποίησης (two dimensional bin packing problem), όπου, για παράδειγμα, πέρα από το βάρος πρέπει να ληφθεί υπ’όψη και ο όγκος των αντικειμένων. Επίσης, αυξάνοντας την διάσταση όπως είναι λογικό αυξάνονται και οι περιορισμοί. Υπάρχουν και άλλες επεκτάσεις του προβλήματος, όπως το τρισδιάστατο πρόβλημα και προβλήματα μεγίστου κόστους (maximum value problems).

1.3 Το μαθηματικό μοντέλο

Όπως αναφέρεται παραπάνω, η μορφή του προβλήματος ακολουθεί το μοντέλο του ακέραιου γραμμικού προγραμματισμού. Υποθέτουμε ότι έχουμε n αντικείμενα μήκους w_j, $j \in \{1,...,n\}$ και στη διάθεση μας κάδους χωρητικότητας W. Χρειάζονται δύο μεταβλητές απόφασης: $x_{ij} = 1$ αν το αντικείμενο j έχει τοποθετηθεί στον κάδο i, όπου οι δείκτες $i,j \in \{1,\dots,n\}$ (το n είναι το μέγιστο όριο για τον αριθμό των κάδων όπως είναι εύκολα κατανοητό), και $u_i = 1$ αν τουλάχιστον ένα αντικείμενο έχει τοποθετηθεί στον κάδο i. Οι περιορισμοί του μοντέλου είναι ότι το άθροισμα των μηκών των αντικειμένων που έχουν τοποθετηθεί σε έναν κάδο δεν ξεπερνά την χωρητικότητα αυτού και ότι κάθε αντικείμενο έχει τοποθετηθεί μόνο σε ένα κάδο (Mohamadi, 2010). Το μοντέλο που προκύπτει είναι το εξής:

$$min \sum_{i=1}^{n} u_i$$

$$\sum_{j=1}^{n} w_j x_{ij} \leq W\, u_i \qquad i \in \{1, \dots, n\}$$

$$\sum_{i=1}^{n} x_{ij} \geq 1 \qquad j \in \{1, \dots, n\}$$

$$x_{ij} \in \{0,1\} \qquad i, j \in \{1, \dots, n\}$$

$$u_i \in \{0,1\} \qquad i \in \{1, \dots, n\}$$

Το παραπάνω μοντέλο μπορεί να επιλυθεί χρησιμοποιώντας τις συνηθισμένες τεχνικές επίλυσης ακέραιου γραμμικού προγραμματισμού. Έχει όμως ιδιαίτερη σημασία το γεγονός ότι για μεγάλες τιμές του n, ο αριθμός των περιπτώσεων που πρέπει να εξεταστούν αυξάνεται κατακόρυφα για να μελετηθούν όλοι οι συνδυασμοί. Ο αλγόριθμος έχει βαθμό πολυπλοκότητας NP-hard και έτσι επιλέγεται συχνά η προσέγγιση του με ευρετικές διαδικασίες.

1.4 Αλγοριθμική πολυπλοκότητα

Η πολυπλοκότητα ενός αλγορίθμου αντιπροσωπεύεται από τις χρονικές και χωρικές απαιτήσεις αυτού κατά την επίλυσή του.

Ως χρονική πολυπλοκότητα εννοείται η μέτρηση του χρόνου που απαιτείται για να εκτελεστεί ένας αλγόριθμος με συγκεκριμένο αριθμό δεδομένων εισόδου. Για να πραγματοποιηθεί η συγκεκριμένη μέτρηση αξιολογείται ο βαθμός αύξησης του χρόνου σε σύγκριση με τον απαιτούμενο χρόνο εκτέλεσης των καθιερωμένων συναρτήσεων. Οι εξεταζόμενες συναρτήσεις διακρίνονται σε σταθερές, λογαριθμικές, πολυωνυμικές και εκθετικές.

Η χωρική πολυπλοκότητα αναφέρεται στην αποθηκευτική μνήμη που δεσμεύει ο αλγόριθμος κατα την επίλυσή του. Αυξάνεται με τον αριθμό των μεταβλητών, τις επαναλήψεις και τις συνθήκες που μπορεί να περιλαμβάνει.

Το ενδιαφέρον των προγραμματιστών επικεντρώνεται κυρίως στην ελάττωση της χρονικής πολυπλοκότητας, καθώς τα τεχνολογικά άλματα των τελευταίων ετών επιτρέπουν την παραγωγή υπολογιστικής μνήμης με μεγάλο χώρο αποθήκευσης και την οικονομική διάθεση της.

Για να μετρηθεί η δυσκολία επίλυσης ενός υπολογιστικού προβλήματος, ως λογική μέθοδος μπορεί να υποτεθεί η μέτρηση του χρόνου που κάνει ο αλγόριθμος για να επιλύσει το πρόβλημα. Αυτό όμως δεν είναι απαραίτητα σωστό καθώς ο χρόνος εκτέλεσης εξαρτάται από την περίπτωση. Για παράδειγμα, στο μοντέλο της πακετοποίησης που εξετάστηκε πριν, για μεγαλύτερες τιμές του αριθμού αντικειμένων n, θα απαιτείται αναλογικά μεγαλύτερος χρόνος εκτέλεσης. Άρα, ο χρόνος που απαιτείται για την εκτέλεση ενος προβλήματος (ή ο χώρος και κάθε άλλο μέγεθος πολυπλοκότητας) υπολογίζεται συναρτήσει του αριθμού δεδομένων εισόδου της κάθε περίστασης. Αν ο αριθμός δεδομένων εισόδου εκφραστεί ως n, ο χρόνος εκτέλεσης του αλγορίθμου μπορεί να διατυπωθεί ως μια συνάρτηση του n (Papadimitriou, 1994). Διακρίνονται τρεις περιπτώσεις πολυπλοκότητας χρόνου: Η χειρότερη περίπτωση (worst-case time complexity) που καταλαμβάνει τον περισσότερο χρόνο για τον συγκεκριμένο αριθμό δεδομένων εισόδου n από κάθε άλλον αριθμό. Η καλύτερη περίπτωση (best-case time complexity), που καταλαμβάνει τον ελάχιστο χρόνο εκτέλεσης σε σύγκριση με κάθε άλλον αριθμό εισόδου και η μέση περίπτωση (average-case time complexity) που αναφέρει τον χρόνο εκτέλεσης που χρειάζεται ο μέσος όρος των αριθμών δεδομένων εισόδου. Για να οριστεί αυτή η περίπτωση είναι αναγκαίο να γίνουν αρκετές εκτελέσεις. Αν ο χρόνος εκτέλεσης *T(n)* είναι πολυώνυμο του *n*, τότε ο αλγόριθμος αναφέρεται ως αλγόριθμος πολυωνυμικού χρόνου (polynomial time algorithm).

Επίσης, οι αλγόριθμοι διακρίνονται σε ντετερμινιστικούς και μη – ντετερμινιστικούς (deterministic, non-deterministic). Μη – ντετερμινιστικός είναι ο αλγόριθμος που παρουσιάζει διαφορετική συμπεριφορά σε κάθε εκτέλεση, σε αντίθεση με τον ντετερμινιστικό που παρουσιάζει την ίδια. Ένας μη - ντετερμινιστικός αλγόριθμος αποτελείται από δύο στάδια. Στο πρώτο στάδιο υποθέτει μια δομή του δεδομένου προβλήματος, η οποία εισάγεται στο δεύτερο στάδιο και ελέγχεται αν η συγκεκριμένη δομή αποτελεί λύση του προβλήματος. Στο δεύτερο στάδιο χρησιμοποιείται ένας ντετερμινιστικός αλγόριθμος που λειτουργεί σε πολυωνυμικό χρόνο. Με αυτό το δεδομένο, γινεται κατανοητό ότι το NP (Non – deterministic polynomial – time) σύνολο έχει ως υποσύνολο το P, δηλαδή P ⊆ NP (Arora, Barak, 2009).

Το μονοδιάστατο πρόβλημα πακετοποίησης απαιτεί στην χειρότερη περίπτωση χρόνο εκτέλεσης $O(n^2)$, ενώ στη μέση και καλύτερη περίπτωση απαιτεί χρόνο *O(nlogn)*. Άρα ο αλγόριθμος επίλυσης θεωρείται πολυωνυμικού χρόνου. Ακόμη, διακρίνεται σε μη – ντετερμινιστικός καθώς ο αριθμός δεδομένων εισόδου (αριθμός αντικειμένων *n*) αλλάζει ανά εκτέλεση. Λόγω της δυσκολίας και των σύνθετων λειτουργιών που πρέπει να αντιμετωπιστούν για την εύρεση μιας λύσης, ακόμη και όχι της βέλτιστης, το πρόβλημα θεωρείται στην γενική του περίπτωση από τα δυσκολότερα προβλήματα των παραπάνω κατηγοριών. Έτσι εξηγείται η διάκριση του σε NP-hard (Non - deterministic polynomial - time hard), δηλαδή το σύνολο των προβλημάτων που είναι τουλάχιστον τόσο δύσκολα όσο τα δυσκολότερα προβλήματα στο σύνολο NP – πλήρη (NP – complete). Για τα προβλήματα αυτά δεν μπορεί να αποδειχθεί ότι ανήκουν στην κατηγορία των NP καθώς δεν υπάρχει προφανής πολυωνυμικού χρόνου διαδικασία που να πιστοποιεί τη λύση. Στην κατηγορία αυτή, οι αλγόριθμοι που παρουσιάζουν βέλτιστες λύσεις είναι σχεδόν αδύνατο να εγγυηθούν γι' αυτές και για να εκτελέσουν τις διεργασίες τους προϋποθέτουν πολύ ισχυρό εξοπλισμό, αυξάνοντας το υπολογιστικό κόστος. Για τον λόγο αυτό, η μελέτη των προβλημάτων αυτών γίνεται κυρίως με αλγορίθμους που καταλήγουν σε προσεγγιστικές λύσεις, κοντά στις βέλτιστες.

1.5 Κατηγορίες αλγορίθμων

Λόγω της συνθετότητας του προβλήματος πακετοποίησης σε κάδους, έχουν διατυπωθεί πολλές διαφορετικές προσεγγίσεις, μεθοδολογίες και τεχνικές. Στην παρούσα φάση επιχειρείται η διατύπωση της λογικής της κάθε μεθόδου επίλυσης για την κατανόηση όλων των μεθόδων που αντιμετωπίζουν το πρόβλημα πακετοποίησης.

1.5.1 Ακριβείς αλγόριθμοι (Exact algorithms)

Η βασική λογική των αλγορίθμων αυτής της κατηγορίας είναι ότι για να βρεθεί η καλύτερη δυνατή λύση (ο παγκόσμιος άριστος), είναι απαραίτητο να βρεθούν όλες οι δυνατές λύσεις του προβλήματος, ώστε να εντοπιστεί η βέλτιστη που θα αριστοποιεί την αντικειμενική συνάρτηση. Η τεχνική αυτή είναι πολύ αποτελεσματική σε απλά προβλήματα, όμως όταν πρόκειται για προβλήματα αυξημένης πολυπλοκότητας δεν ανταποκρίνονται πάντα με επιτυχία. Αυτό γιατί αυξάνεται κατά πολύ το υπολογιστικό κόστος κατά τη λειτουργία της, όταν αποθηκεύονται πολλά δεδομένα και ακολουθούν συνήθως μη πρακτικές μεθόδους. Δηλαδή ο χρόνος που απαιτείται για να βρεθεί η βέλτιστη λύση είναι ασύμφορος και δεσμεύεται πολύ μεγάλο ποσοστό της αποθηκευτικής μνήμης.

1.5.1.1 Γραμμικός προγραμματισμός (Linear Programming)

Ο γραμμικός προγραμματισμός χρησιμοποιεί την λογική των αλγορίθμων ακρίβειας. Αποτελεί το αρχικό εργαλείο επίλυσης προβλημάτων, καθώς αναπτύχθηκε στην δεκαετία του 1940 για την αντιμετώπιση προβλημάτων προγραμματισμού στη πολεμική περίοδο. Στη συνέχεια προσαρμόστηκε στις ανάγκες των βιομηχανιών, γνωρίζοντας ραγδαία εξέλιξη. Στα προβλήματα πακετοποίησης χρησιμοποιείται ορίζοντας τις εξεταζόμενες μεταβλητές και εκφράζοντας τις σχέσεις και τις συνθήκες που απαιτούνται για τον προσδιορισμό της επίλυσης, υπακούοντας στους εκάστοτε περιορισμούς. Ορίζοντας την αντικειμένικη συνάρτηση εκφράζονται τα χαρακτηριστικά των λύσεων με απώτερο σκοπό την βελτιστοποίηση της.

Η μέθοδος αυτή ακολουθεί τα χαρακτηριστικά των αλγορίθμων ακρίβειας, δηλαδή είναι αποτελεσματική μόνο για μικρές τιμές δεδομένων εισόδου. Σε μεγαλύτερες τιμές, η μέθοδος κρίνεται χαμηλής αποδοτικότητας και αυτό εξηγείται από το γεγονός ότι για να χρησιμοποιηθεί πρέπει οι μαθηματικές συναρτήσεις να είναι γραμμικές, συμπεριλαμβανομένων και των συναρτήσεων των περιορισμών και της αντικειμενικής συνάρτησης. Ακόμη, όπως γίνεται κατανοητό και από το μοντέλο της μεθόδου, η επίλυση θα έχει πολύ μεγάλο βαθμό συμμετριών.

Για παράδειγμα, αν στη βέλτιστη λύση που θα βρεθεί μετατεθούν οι θέσεις των κάδων, η λύση που θα προκύψει θα είναι πάλι η βέλτιστη. Αυτές οι συμμετρίες δεν είναι δυνατόν να αναπαρασταθούν με τη μέθοδο του γραμμικού προγραμματισμού.

Για το πρόβλημα της πακετοποίησης, πιο διαδεδομένες είναι δύο άλλες προσεγγίσεις στην κατηγορία των αλγορίθμων ακρίβειας: η μέθοδος ορίου διακλάδωσης (branch-and-bound) και ο δυναμικός προγραμματισμός (dynamic programming).

1.5.1.2 Μέθοδος ορίου διακλάδωσης (Branch-and-bound)

Η στρατηγική της μεθόδου branch-and-bound επικεντρώνεται στην δημιουργία μιας οριακής συνθήκης (upper bound) και την αναδρομική διαίρεση του βασικού προβλήματος σε μικρότερα υπο-προβλήματα που ελέγχονται από την οριακή συνθήκη. Η επίλυση των μικρών υπο-προβλημάτων συνεπάγεται και την επίλυση του αρχικού προβλήματος. Η αρχική προσέγγιση με την συγκεκριμένη μέθοδο έγινε με τον αλγόριθμο των Martello και Toth για ένα παρεμφερές πρόβλημα με αυτό της πακετοποίησης, το πρόβλημα του σακιδίου (knapsack problem) (Martello, Pisinger, Toth, 2000). Αναφορικά, στο πρόβλημα αυτό η λογική είναι να χωρέσουν όσα περισσότερα αντικείμενα γίνεται σε έναν δεδομένο χώρο (σακίδιο), παράλληλη δηλαδή με αυτήν της πακετοποίησης.

1.5.1.3 Δυναμικός προγραμματισμός (Dynamic programming)

Ο δυναμικός προγραμματισμός βρήκε αρχική εφαρμογή στα διαδοχικά προβλήματα απόφασης γνωρίζοντας ραγδαία εξέλιξη και πλέον χρησιμοποιείται σε πολλά συνδυαστικά προβλήματα. Τα χαρακτηριστικά της μεθόδου αποτελούν την βάση της τεχνικής branch-and-bound: Το πρόβλημα αντιμετωπίζεται ως πολλά μικρότερα σε μέγεθος επικαλυπτόμενα προβλήματα, για τα οποία υπολογίζεται η λύση. Η λύση αυτή αποθηκεύεται σε έναν πίνακα που μεγαλώνει σε στοιχεία όσο εκτελείται ο αλγόριθμος και υπάρχει εξάρτηση μεταξύ των λύσεων, καθώς οι νέες λύσεις συνθέτονται απο αυτές που έχουν ήδη υπολογιστεί. Η μέθοδος παύει να εκτελείται όταν υπολογιστεί η λύση του αρχικού προβλήματος με τον έλεγχο της ανάλογης συνθήκης. Σε αυτήν την περίπτωση, το βασικό αρνητικό έγκειται στο γεγονός ότι η μέθοδος αυτή είναι ιδιαίτερα απαιτητική σε αποθηκευτικό χώρο, αφού οι λύσεις που αποθηκεύονται στον πίνακα καταλαμβάνουν μεγάλο ποσοστό μνήμης σε σύνθετα προβλήματα. Το χαρακτηριστικό αυτό καθιστά τον δυναμικό προγραμματισμό καταλληλότερο για μικρών ή μεσαίων δεδομένων προβλήματα, που θα υπάρχει μικρότερη δέσμευση μνήμης και λιγότερος υπολογιστικός χρόνος κατά την εκτέλεση του αλγορίθμου.

1.5.2 Ευρετικοί αλγόριθμοι (Heuristic algorithms)

Τα αρνητικά χαρακτηριστικά των αλγορίθμων ακρίβειας έρχεται να αντιμετωπίσει ένα άλλο είδος αλγορίθμων, οι ευρετικοί αλγόριθμοι. Οι αλγόριθμοι αυτής της κατηγορίας έχουν την δυνατότητα να καταλήξουν σε μια αρκετά καλή λύση με προσεγγιστικές τεχνικές που απαιτούν πολύ μικρό υπολογιστικό χρόνο και χαμηλό χώρο αποθήκευσης. Οι λύσεις που παρουσιάζουν πολλές φορές είναι και οι βέλτιστες που υπάρχουν, αν και αυτό στις περισσότερες περιπτώσεις δεν μπορεί ή είναι πολύ δύσκολο να αποδειχθεί. Οι ευρετικοί αλγόριθμοι συνήθως δεν έχουν ακριβή διατύπωση και δεν ακολουθούν συγκεκριμένο μοντέλο, όπως για παράδειγμα το μοντέλο γραμμικού προγραμματισμού στο πρόβλημα της πακετοποίησης. Βασίζονται κυρίως σε εμπειρικές παρατηρήσεις και εμπνεύσεις του προγραμματιστή που δεν είναι απευθείας κατανοητές από τον έξω παρατηρητή. Συνήθως αποτελούνται από δύο μέρη, το κατασκευαστικό μέρος και το μέρος βελτίωσης των λύσεων: Αρχικά δημιουργείται μια αρχική λύση και στη συνέχεια τροποποιείται με διάφορες μεθόδους καταλήγοντας σε νέες βελτιωμένες λύσεις.

Τα κριτήρια επιλογής μιας ευρετικής μεθόδου για την επίλυση ενός δεδομένου προβλήματος συνοψίζονται στα παρακάτω (Russell, Norvig, 2010):

- *Βέλτιστη λύση:* Όταν για ένα πρόβλημα υπάρχουν πολλαπλές λύσεις, ο ευρετικός αλγόριθμος δεν εγγυάται ότι η λύση που παρουσιάζει είναι η βέλτιστη. Πρέπει να καθοριστεί αν υπάρχει ανάγκη προσδιορισμού της καλύτερης και όχι απλά μιας λύσης.
- *Πληρότητα:* Όπως αναφέρθηκε παραπάνω, είναι δυνατό για ένα πρόβλημα να υπάρχουν πολλές λύσεις. Το ερώτημα είναι αν χρειάζονται όλες καθώς ο ευρετικός αλγόριθμος θα παρουσιάσει μόνο μια.
- *Ακρίβεια:* Σε ορισμένες περιπτώσεις, μπορεί να είναι δύσκολο να αποφασιστεί αν η προτεινόμενη λύση είναι αρκετά ικανοποιητική, καθώς οι διαδικασίες για την εύρεσή της δεν περιγράφονται με ιδιαίτερα λεπτομερή τρόπο.
- *Χρόνος εκτέλεσης:* Κάποιοι ευρετικοί αλγόριθμοι για το ίδιο πρόβλημα λειτουργούν γρηγορότερα από κάποιους άλλους. Αν ο αλγόριθμος δεν έχει αρκετά πρακτική διατύπωση είναι μόνο οριακά πιο ταχύς από τις κλασικές μεθόδους.

Στην κατηγορία των NP – hard προβλημάτων, η ευρετική μέθοδος είναι πολλές φορές η μόνη δυνατή επιλογή για μια ποικιλία σύνθετων προβλημάτων βελτιστοποίησης. Ιδιαίτερα στο πρόβλημα της πακετοποίησης σε κάδους, προτιμούνται συνήθως λόγω των πρακτικών αναγκών του προβλήματος και της πολυπλοκότητας του αποτελέσματος. Κατά την δημιουργία του αλγορίθμου, πρέπει η προσοχή να εστιαστεί στην παραγωγή λύσης που να μην απέχει πολύ από την βέλτιστη σε σύντομο χρονικό διάστημα, διαθέτοντας την απαραίτητη ευελιξία για την προσαρμογή διαφορετικών συνθηκών στο πρόβλημα. Οι πιο διαδεδομένες τεχνικές επίλυσης του προβλήματος πακετοποίησης θα μελετηθούν παρακάτω σε αυτή την εργασία.

1.5.3 Μεταευρετικοί αλγόριθμοι (Metaheuristic algorithms)

Στη μαθηματική βελτιστοποίηση, οι μεταευρετικοί αλγόριθμοι αποτελούνται από ευρετικές μεθόδους που επεξεργάζονται περαιτέρω για την εύρεση ή παραγωγή ακόμα καλύτερης λύσης σε ένα πρόβλημα βελτιστοποίησης, λειτουργώντας με μικρή υπολογιστική χωρητικότητα. Όπως και οι ευρετικές μέθοδοι, δεν μπορούν να εγγυηθούν την εύρεση του παγκόσμιου άριστου στα προβλήματα που χρησιμοποιούνται. Οι μεταευρετικοί αλγόριθμοι αποτελούνται κυρίως από στρατηγικές που κατευθύνουν την ερευνητική διαδικασία, με σκοπό την αποτελεσματική εξερεύνηση του διαστήματος που περιέχει τις λύσεις για την εύρεση λύσεων κοντά στη βέλτιστη. Περιλαμβάνουν πολλές διαφορετικές τεχνικές που διακυμαίνονται από απλές διαδικασίες τοπικής αναζήτησης έως πιο σύνθετες διεργασίες. Υπερτερούν των ευρετικών καθώς εκφράζουν πιο ικανοποιητικά αποτελέσματα και έχουν γενική εφαρμογή σε πολλά προβλήματα βελτιστοποίησης: Σε αντίθεση με τους ευρετικούς, δεν κατασκευάζονται με βάση τους περιορισμούς κάθε προβλήματος. Για το πρόβλημα της πακετοποίησης, έχουν αναπτυχθεί αλγόριθμοι από τις παρακάτω βασικές κατηγορίες μεταευρετικών:

- Αλγόριθμοι τοπικής αναζήτησης
- Εξελικτικοί αλγόριθμοι
- Αλγόριθμοι πλήθους

1.5.3.1 Αλγόριθμοι τοπικής αναζήτησης (Local search algorithms)

Η κατηγορία αλγορίθμων τοπικής αναζήτησης αναπτύχθηκε για την επίλυση δύσκολων υπολογιστικών προβλημάτων βελτιστοποίησης. Ανταποκρίνονται με επιτυχία σε προβλήματα που μπορούν να διατυπωθούν με την εύρεση λύσης που αριστοποιεί κάποιο κριτήριο μεταξύ ενός συνόλου υποψήφιων λύσεων. Οι νέες λύσεις προκύπτουν από τις προηγούμενες με την εφαρμογή κάποιου βήματος και η διαδικασία επαναλαμβάνεται μέχρι η λύση να θεωρηθεί ικανοποιητική ή μέχρι να συμπληρωθεί η διάρκεια κάποιου χρονικού ορίου (Battiti, Brunato, Mascia, 2008).

Στην κατηγορία των αλγορίθμων τοπικής αναζήτησης ανήκει η μέθοδος ανάβασης λόφου (hill climbing), η προσομοιωμένη ανόπτηση (simulated annealing) και η μέθοδος έρευνας ταμπού (tabu search).

Η μέθοδος ανάβασης λόφου βασίζεται σε έναν σχετικά απλό στην κατανόησή του αλγόριθμο, ο οποίος στην αρχή της εκτέλεσής του υποθέτει μια τυχαία λύση της αντικειμενικής συνάρτησης. Στη συνέχεια, αλλάζοντας ένα στοιχείο της λύσης αυτής προσπαθεί να καταλήξει σε μια νέα λύση, η οποία εφαρμόζεται στην αντικειμενική συνάρτηση και γίνεται δεκτή μόνο αν έχει ίδιο ή καλύτερο αποτέλεσμα με την προηγούμενη. Ο αλγόριθμος χαρακτηρίζεται από σταθερή συμπεριφορά, υψηλή ταχύτητα εκτέλεσης, αλλά δυσχρηστία σε ορισμένες περιπτώσεις καθώς υπάρχει κίνδυνος παγίδευσής του σε τοπικό ακρότατο της αντικειμενικής συνάρτησης, που μπορεί να απέχει πολύ από το ολικό. Για την αποφυγή αυτού του σφάλματος έχουν αναπτυχθεί ορισμένες παραλλαγές της μεθόδου που επανεκκινούν την διεργασία, επιλέγοντας νέα αρχική λύση.

Μια άλλη μέθοδος βελτιστοποίησης για τα προβλήματα πακετοποίησης είναι η προσωμοιωμένη ανόπτηση. Η κεντρική ιδέα είναι παρόμοια με αυτή της παραπάνω μεθόδου, επιλέγεται τυχαία αρχική λύση και συγκρίνεται με τις υποψήφιες γειτονικές. Σε αυτήν την περίπτωση όμως αποφεύγεται περίτεχνα η παγίδευση σε τοπικά ακρότατα, με την εισαγωγή των κατάλληλων συνθηκών που εξασφαλίζουν ότι η λύση που παρέχεται είναι η βέλτιστη (Kämpke, 1988). Στατιστικά, η μέθοδος αυτή έχει πολύ ικανοποιητικά αποτελέσματα επιτυχίας.

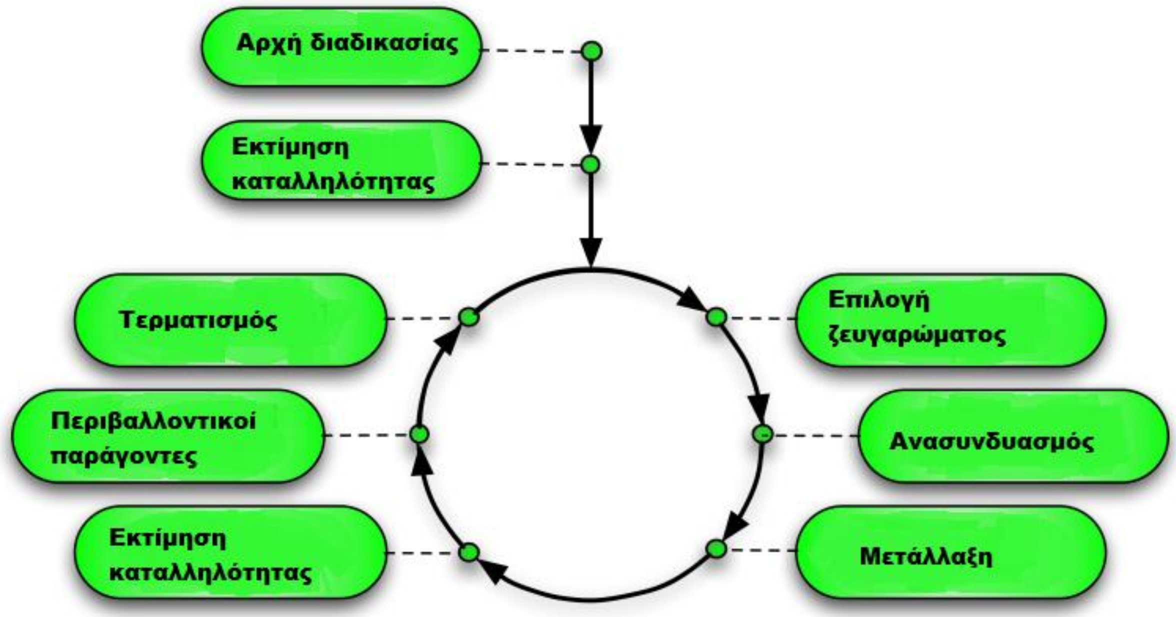
Η μέθοδος έρευνας ταμπού αποτελεί μια ακόμα παραλλαγή αλγορίθμου τοπικής αναζήτησης. Όμοια με τις προηγούμενες, ξεκινά από τυχαία αρχική τιμή για τον υπολογισμό της λύσης που αριστοποιεί την αντικειμενική συνάρτηση, με την διαφορά ότι αποθηκεύει κάθε λύση που υπολογίζει για να αποφευχθεί η ανακύκλωση των λύσεων και η παγίδευση σε τοπικό ακρότατο. Παρόλα αυτά, το σχέδιο αυτό δεν είναι πάντα επιτυχημένο καθώς σε ορισμένες περιπτώσεις περιορίζεται κατά πολύ η αναζήτηση λύσεων και προσπερνώνται ακόμα και πολύ ικανοποιητικές λύσεις.

Οι παραπάνω τεχνικές είναι ιδιαίτερα δημοφιλείς και χρησιμοποιούνται ευρέως. Τα τελευταία χρόνια έχουν αναπτυχθεί πολλές παραλλαγές αλγορίθμων που ακολουθούν τις παραπάνω τεχνικές για επίλυση προβλημάτων πακετοποίησης δύο ή περισσότερων διαστάσεων.

1.5.3.2 Εξελικτικοί αλγόριθμοι (Evolutionary algorithms)

Οι εξελικτικοί αλγόριθμοι είναι ένα υποσύνολο των μεταευρετικών αλγορίθμων βελτιστοποίησης. Χρησιμοποιούν μηχανισμούς εμπνευσμένους απο τη βιολογική εξέλιξη, όπως η αναπαραγωγή (reproduction), η μεταλλαγή (mutation) και ο ανασυνδυασμός (recombination). Στην γενική τους ονομασία αναφέρονται ως αλγόριθμοι βελτιστοποίησης βάσει πληθυσμού (population-based), όπου οι υποψήφιες λύσεις του προβλήματος έχουν τον ρόλο του ατόμου σε έναν πληθυσμό και εξετάζεται η ποιότητα της επίλυσης μέσω ενός μέτρου καταλληλότητας (fitness function). Στη συνέχεια, ο πληθυσμός εξελίσσεται μετα την επαναλαμβανόμενη εφαρμογή των παραπάνω διεργασιών και τις συνθήκες του προβλήματος που αντιπροσωπεύονται από τους περιβαλλοντικούς παράγοντες (Ashlock, 2006). Το μέτρο καταλληλότητας διατυπώνεται από την αντικειμενική συνάρτηση του προβλήματος, η οποία αριστοποιείται από τις υποψήφιες λύσεις που προκύπτουν. Αυτές συγκρίνονται και επιλέγονται οι πιο αποτελεσματικές, οι οποίες υποβάλλονται σε ανασυνδυασμό ή μετάλλαξη για την δημιουργία της επόμενης γενιάς λύσεων. Η διαδικασία αυτή επαναλαμβάνεται έως ότου βρεθεί μία λύση με ικανοποιητική ποιότητα ή το νέο σύνολο λύσεων ταυτιστεί με ένα παλιό, δηλαδή δεν μπορεί να βελτιωθεί παραπάνω. Για τη λειτουργία του αλγορίθμου είναι απαραίτητος ο καθορισμός της αντικειμενικής συνάρτησης και μίας κατάλληλης συνθήκης τερματισμού των επαναληπτικών διεργασιών τροποποίησης των λύσεων.

Στα πλαίσια του προβλήματος πακετοποίησης σε κάδους έχουν αναπτυχθεί διάφοροι εξελικτικοί αλγόριθμοι, με την μεγαλύτερη κατηγορία να προέρχεται από το υποσύνολο των γενετικών αλγορίθμων. Οι γενετικοί αλγόριθμοι, βασιζόμενοι στην αρχή των εξελικτικών, χρησιμοποιούν ευρετικές και μεταευρετικές μεθόδους για την παραγωγή λύσεων, εφαρμόζοντας στην πράξη τεχνικές εμπνευσμένες από την φυσική εξέλιξη.



Εικόνα 1 - Διάγραμμα ροής διεργασιών του γενικού μοντέλου των εξελικτικών αλγορίθμων.

Σε έναν γενετικό αλγόριθμο, ο πληθυσμός των υποψήφιων λύσεων του προβλήματος εξελίσσεται για την παραγωγή νέων καλύτερων λύσεων. Με βάση την προσομοίωση της φυσικής εξέλιξης, οι ιδιότητες των λύσεων αντιπροσωπεύονται από τα χρωμοσώματα. Στην αρχή του αλγορίθμου, ο πληθυσμός αποτελείται από τυχαία άτομα που παράγονται κατά μία επαναληπτική διαδικασία, με τον πληθυσμό κάθε επανάληψης να αποκαλείται γενιά. Σε κάθε γενιά εκτιμάται η καταλληλότητα κάθε ατόμου, δηλαδή η τιμή της αντικειμενικής συνάρτησης με χρησιμοποιούμενη μεταβλητή την κάθε λύση. Τα πιο κατάλληλα άτομα επιλέγονται από τον κάθε πληθυσμό και τα χρωμοσώματά τους τροποποιούνται για την δημιουργία μιας νέας γενιάς. Η νέα αυτή γενιά χρησιμοποιείται στην επόμενη επανάληψη του αλγόριθμου και η διαδικασία τερματίζεται όταν παραχθεί ένας μέγιστος αριθμός γενεών, ή ο βαθμός καταλληλότητας του πληθυσμού κριθεί ικανοποιητικός (Holland, 1975, 1992).

Τα βασικά βήματα που ακολουθεί ένας γενετικός αλγόριθμος μπορούν να αναπαρασταθούν σε μορφή ψευδοκώδικα ως εξής:

Παραγωγή του αρχικού πληθυσμού
Μέχρι τη συνάντηση κάποιου κριτηρίου, κάνε τα παρακάτω

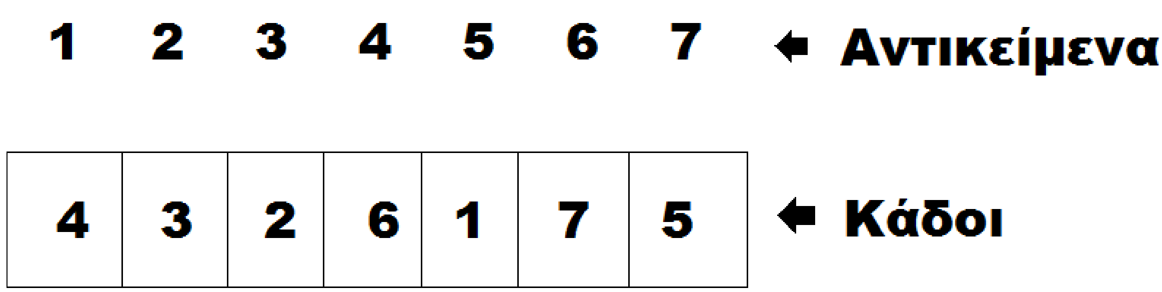
    Διάλεξε ζεύγη για αναπαραγωγή

Εκτέλεσε διασταυρώσεις για την δημιουργία απογόνων
Εκτίμησε την καταλληλότητα των απογόνων
Παραγωγή νέου πληθυσμού

Τέλος διεργασίας

Η προσαρμογή των γενετικών αλγορίθμων στο πρόβλημα πακετοποίησης γίνεται με πολλές διαφορετικές προσεγγίσεις.Μερικές από αυτές αποτελούν οι παρακάτω: Αναπαράσταση βασισμένη στους κάδους (bin-based representation), αναπαράσταση βασισμένη στα αντικείμενα (object-based representation) και αναπαράσταση βασισμένη σε ομάδες (group-based representation) (Mohamadi, 2010).

Στην αναπαράσταση βασισμένη στους κάδους, το χρωμόσωμα έχει μήκος ίσο με τον αριθμό των αντικειμένων. Κάθε κάδος αντιπροσωπεύεται από ένα γονίδιο. Η θέση του κάθε γονιδίου σε ένα χρωμόσωμα υποδεικνύει το αντικείμενο που τοποθετείται στον κάδο στον οποίο αντιστοιχεί το γονίδιο. Για παράδειγμα, το χρωμόσωμα 4 3 2 6 1 7 5 υποδεικνύει ότι το πρώτο αντικέιμενο τοποθετείται στον κάδο 4, το δεύτερο αντικείμενο τοποθετείται στον κάδο 3 κτλ.



Εικόνα 2 - Αναπαράσταση βασισμένη σε κάδους ενός χρωμοσώματος.

Τα βασικό πλεονέκτημα αυτής της αναπαράστασης είναι ότι το κάθε χρωμόσωμα έχει καθορισμένο μήκος, το οποίο είναι κατάλληλο για εφαρμογή των γενετικών τεχνικών. Ωστόσο, σε μεγάλους αριθμούς αντικειμένων θα δημιουργηθούν πολλές περιττές λύσεις που θα αυξάνουν την διάρκεια εκτέλεσης και θα μειώνουν την αποτελεσματικότητα του αλγορίθμου.

Στην αναπαράσταση με βάση τα αντικείμενα, το χρωμόσωμα διαχωρίζεται για να παρουσιάσει τη τοποθέτηση των αντικειμένων. Το χρωμόσωμα 1 2| 3 4 5| 6 7 υποδεικνύει ότι στον πρώτο κάδο ανήκουν τα αντικείμενα 1,2, στον δεύτερο τα αντικείμενα 3,4,5 και στον τρίτο τα αντικείμενα 6,7.

Πιο αποτελεσματική στο πρόβλημα της πακετοποίησης δείχνει να είναι η αναπαράσταση με βάση τις ομάδες. Σε αυτήν την περίπτωση, οι κάδοι έχουν την μορφή γραμμάτων και η σειρά γραφής τους στο χρωμόσωμα αναδεικνύει ποιο αντικείμενο τοποθετείται στον αντίστοιχο κάδο. Δηλαδή, το χρωμόσωμα DABEBCA αναπαριστά ότι το αντικείμενο 1 έχει τοποθετηθεί στον κάδο D, το αντικείμενο 2 μαζί με το αντικείμενο 7 έχουν τοποθετηθεί στον κάδο A κτλ.

1.5.3.3 Αλγόριθμοι πλήθους (Swarm algorithms)

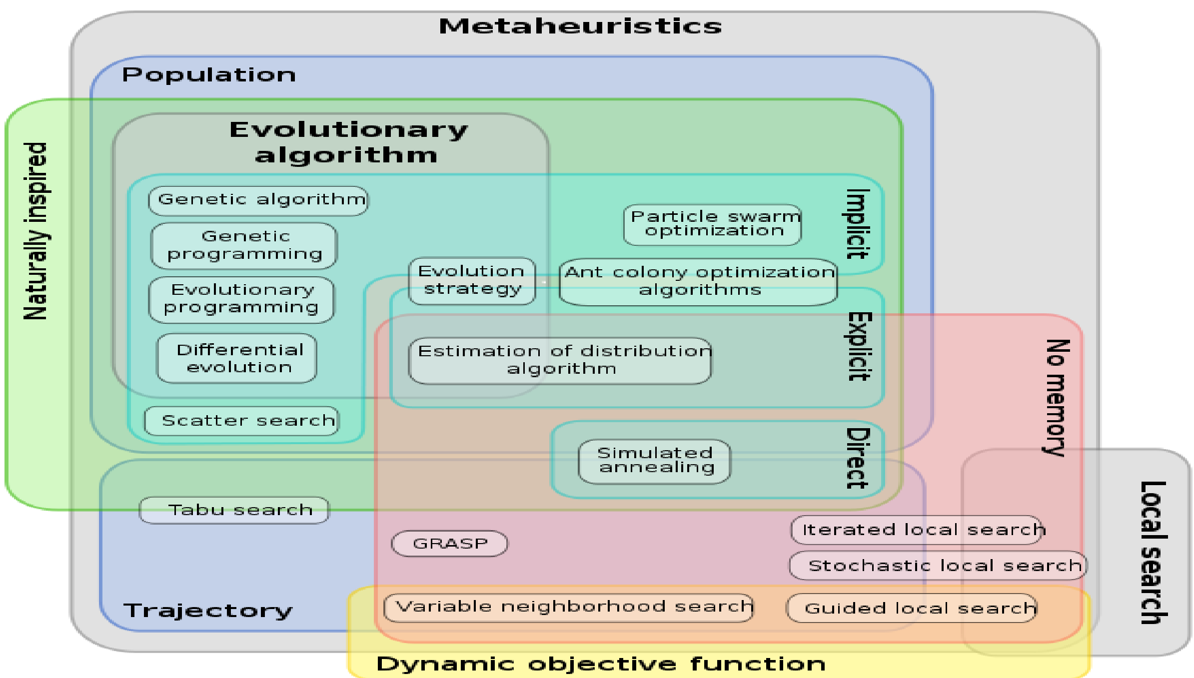
Τα συστήματα αλγορίθμων πλήθους αποτελούνται από ένα πληθυσμό που αλληλεπιδρά με τον γειτονικό και με το περιβάλλον για την εύρεση λύσης. Ακολουθώντας το πρότυπο των εξελικτικών αλγορίθμων, είναι εμπνευσμένοι από τη φύση και τα βιολογικά συστήματα. Η πιο διάσημη κατηγορία αλγορίθμων πλήθους αποτελείται από τους αλγορίθμους αποικίας μυρμηγκιών (ant colony optimization algorithms). Η αρχική μορφή αυτού του αλγορίθμου προτάθηκε απο τον Marco Dorigo το 1992, με σκοπό την εύρεση ενός βέλτιστου σε διάγραμμα βασιζόμενος στην συμπεριφορά των μυρμηγκιών όταν αναζητούν το δρόμο ανάμεσα στην αποικία τους και το φαγητό.

Το πρόβλημα πακετοποίησης σε κάδους προσεγγίζεται αποτελεσματικά από τους αλγορίθμους αποικίας μυρμηγκιών. Χρησιμοποιούν μια αποικία τεχνητών μυρμηγκιών που δημιουργούν νέες λύσεις συνδυάζοντας ευρετικές πληροφορίες και ένα τεχνητό ίχνος φερομόνης. Κατά την αναζήτηση του φαγητού, τα μυρμήγκια αφήνουν στο πέρασμά τους φερομόνη. Η ποσότητα της φερομόνης που υπάρχει σε κάθε διαδρομή εξαρτάται από το μήκος της διαδρομής και από την ποιότητα της τροφής. Η φερομόνη που υπάρχει στις διαδρομές δίνει πληροφορίες στα υπόλοιπα μυρμήγκια της αποικίας καθώς τα προσελκύει. Αναλογικά, τα μονοπάτια που οδηγούν σε τροφές μεγάλης ποσότητας και κοντά στη φωλιά γίνονται πιο πολυσύχναστα και σημαδεύονται με μεγαλύτερη ποσότητα φερομόνης. Η ποσότητα της φερομόνης αυξάνεται σύμφωνα με την ποιότητα της επίλυσης που δημιουργείται από τα μυρμήγκια. Το αρνητικό αυτής της μεθόδου είναι ότι μπορεί να απαιτεί αρκετά υψηλούς χρόνους εκτέλεσης και ο κώδικας που χρειάζεται να διατυπωθεί είναι ιδιαίτερα σύνθετος.

1.6 Σύνοψη 1^{ου} κεφαλαίου

Όπως διαπιστώθηκε, έχουν διατυπωθεί πολλές διαφορετικές τεχνικές προσέγγισης του προβλήματος πακετοποίησης σε κάδους. Οι αλγόριθμοι ακρίβειας διατυπώνουν με σαφήνεια το μοντέλο, όμως στην πράξη μπορούν να εφαρμοστούν για μικρές περιπτώσεις προβλημάτων. Για τα προβλήματα που αντιστοιχούν σε εφαρμογές της καθημερινότητας συνιστάται η χρήση ευρετικών μεθόδων.

Τα τελευταία χρόνια έχει δοθεί ιδιαίτερη έμφαση στις μεταευρετικές τεχνικές λόγω της ικανότητάς τους να αντιμετωπίζουν σύνθετα προβλήματα καταλήγοντας σε ποιοτικές λύσεις. Η συνεχής έρευνα που γίνεται για ακόμα καλύτερες τεχνικές επίλυσης των πολυδιάστατων προβλημάτων πακετοποίησης έχει επικεντρώσει το ενδιαφέρον των ερευνητών στις μεταευρετικές τεχνικές, αναπτύσσοντας συνεχώς το σύνολο των μεθόδων που ανήκουν σε αυτές. Μια προσπάθεια κατηγοριοποίησης των μεθόδων της συγκεκριμένης κατηγορίας γίνεται με το παρακάτω σχήμα:



Εικόνα 3 - Κατηγοριοποίηση των μεταευρετικών τεχνικών με βάση τα χαρακτηριστικά τους.

Για την αντιμετώπιση ενός προβλήματος πακετοποίησης, πρέπει να διατυπωθούν με ακρίβεια οι συνθήκες και οι περιορισμοί του προβλήματος για την επιλογή της καταλληλότερης μεθόδου επίλυσης. Εάν πρόκειται για πρόβλημα δύο ή περισσότερων διαστάσεων είναι προτιμότερες οι μεταευρετικές τεχνικές, όμως στην περίπτωση της παρούσας εργασίας που ερευνάται το μονοδιάστατο πρόβλημα οι ευρετικές τεχνικές είναι αρκετά αποτελεσματικές.

Κεφάλαιο 2

Ο ευρετικός αλγόριθμος

2.1 Περιγραφή της ευρετικής προσέγγισης

Οι ευρετικοί αλγόριθμοι αποσκοπούν στην εύρεση λύσης σε μικρό χρονικό διάστημα, χωρίς να μπορούν να εγγυηθούν αν η λύση αυτή είναι η βέλτιστη. Προσεγγίζοντας το μονοδιάστατο πρόβλημα πακετοποίησης σε κάδους, αναπτύσσονται τέσσερις διαφορετικές τεχνικές επίλυσης με γραφική αναπαράσταση των αποτελεσμάτων τους, χρησιμοποιώντας τη γλώσσα προγραμματισμού MATLAB.

Ανακεφαλαιώνοντας, στο μονοδιάστατο πρόβλημα της πακετοποίησης σε κάδους, υπάρχουν στην διάθεσή μας απεριόριστοι κάδοι συγκεκριμένης χωρητικότητας. Ένα πλήθος αντικειμένων με διάσταση μεγέθους μικρότερη ή ίση με αυτή των κάδων πρέπει να τοποθετηθεί σε αυτούς. Ο σκοπός είναι τα αντικείμενα να τοποθετηθούν με τέτοιο τρόπο ώστε να ελαχιστοποιείται ο αριθμός των κάδων που χρησιμοποιούνται και να μην ξεπερνούν την χωρητικότητα αυτών. Μέσω της γραφικής αναπαράστασης, δίνονται ακριβείς πληροφορίες για το πόσοι κάδοι χρησιμοποιήθηκαν και για την τοποθεσία του κάθε αντικειμένου μετά την εκτέλεση των ευρετικών αλγορίθμων.

2.2 Ανάλυση των μεθόδων

Η ευρετική προσέγγιση του μονοδιάστατου προβλήματος πακετοποίησης βασίζεται σε τέσσερις διαφορετικές μεθόδους επίλυσης που ονομάζονται ως εξής:

- First Fit
- Best Fit
- First Fit Decreasing
- Best Fit Decreasing

Κάθε μία από αυτές τις μεθόδους επιλύει το πρόβλημα πακετοποίησης βασιζόμενη σε μία κεντρική λογική. Επειδή ανήκουν στις ευρετικές μεθόδους επίλυσης, οι λύσεις που προσφέρουν δεν μπορεί να αποδειχθεί ότι είναι οι βέλτιστες, αλλά αναλύοντας κάθε βήμα του αλγόριθμου μπορεί να διεξαχθεί ένα ασφαλές συμπέρασμα για την ποιότητα της κάθε λύσης. Στην αρχή του αλγορίθμου θεωρείται ένα ορισμένο πλήθος αντικειμένων τυχαίων διαστάσεων που παράγεται με κάθε εκτέλεση. Στη συνέχεια, τα αντικείμενα αυτά τοποθετούνται σε κάδους και παρουσιάζεται ο αριθμός των κάδων που χρησιμοποιήθηκαν.

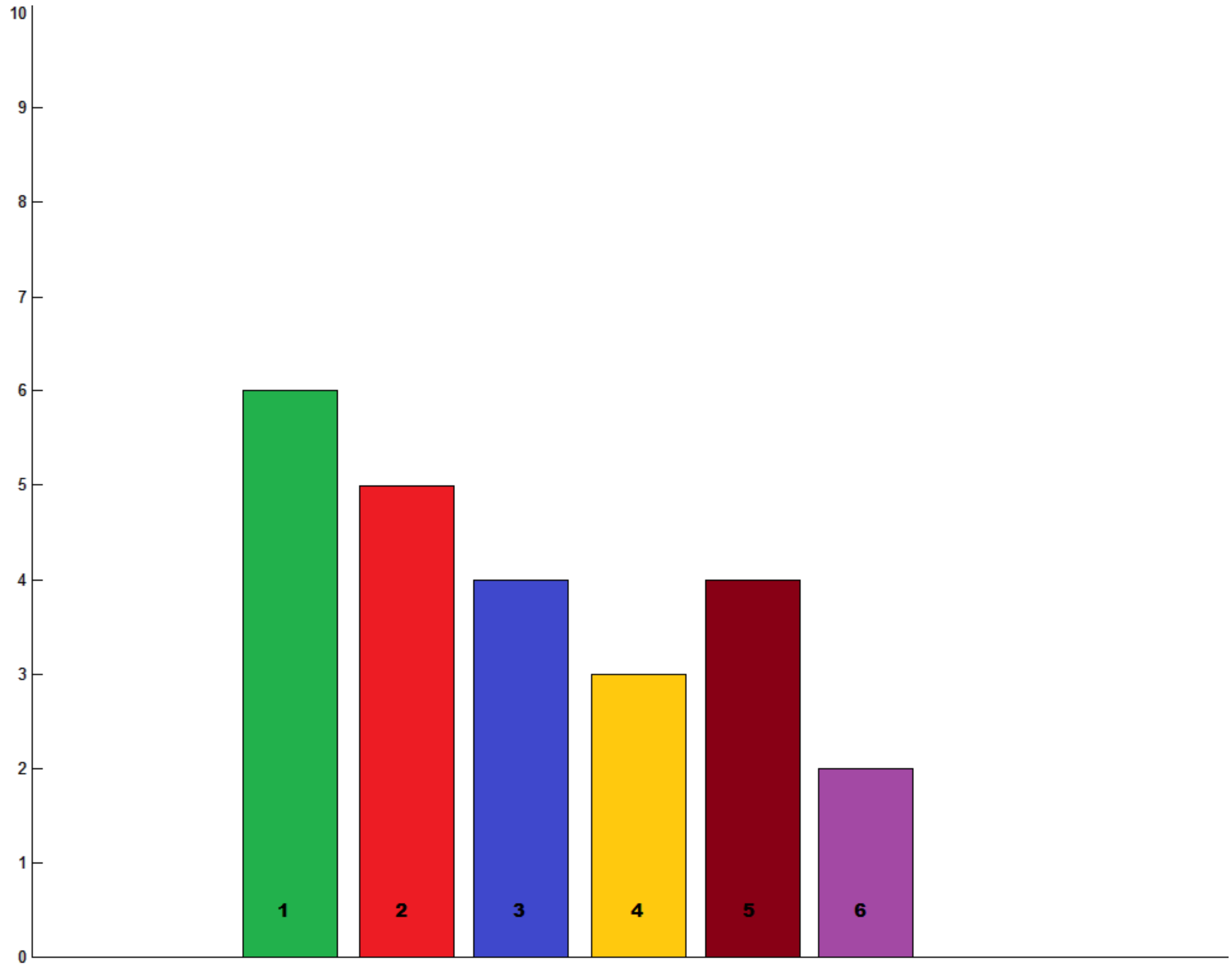
2.2.1 First Fit (FF)

Ο ευρετικός αλγόριθμος First Fit αντιπροσωπεύει την πιο απλή προσέγγιση του προβλήματος. Τα αντικείμενα που παράγονται στην αρχή του αλγορίθμου εξετάζονται διαδοχικά και κάθε ένα από αυτά τοποθετείται στον πρώτο κάδο που έχει τη δυνατότητα να το αποθηκεύσει. Ο κάθε κάδος έχει αρχική χωρητικότητα $r = C$ και όταν τοποθετηθεί σε αυτόν το αντικείμενο i με διάσταση u, η υπολειπόμενη χωρητικότητα του κάδου μειώνεται κατά $r = r - u_i$. Αν κανένας από τους είδη υπάρχοντες κάδους δεν έχει αρκετό χώρο για να τοποθετηθεί σε αυτόν το αντικείμενο, γίνεται χρήση νέου κάδου. Ο ψευδοκώδικας της συγκεκριμένης μεθόδου έχει την παρακάτω μορφή:

First Fit heuristic

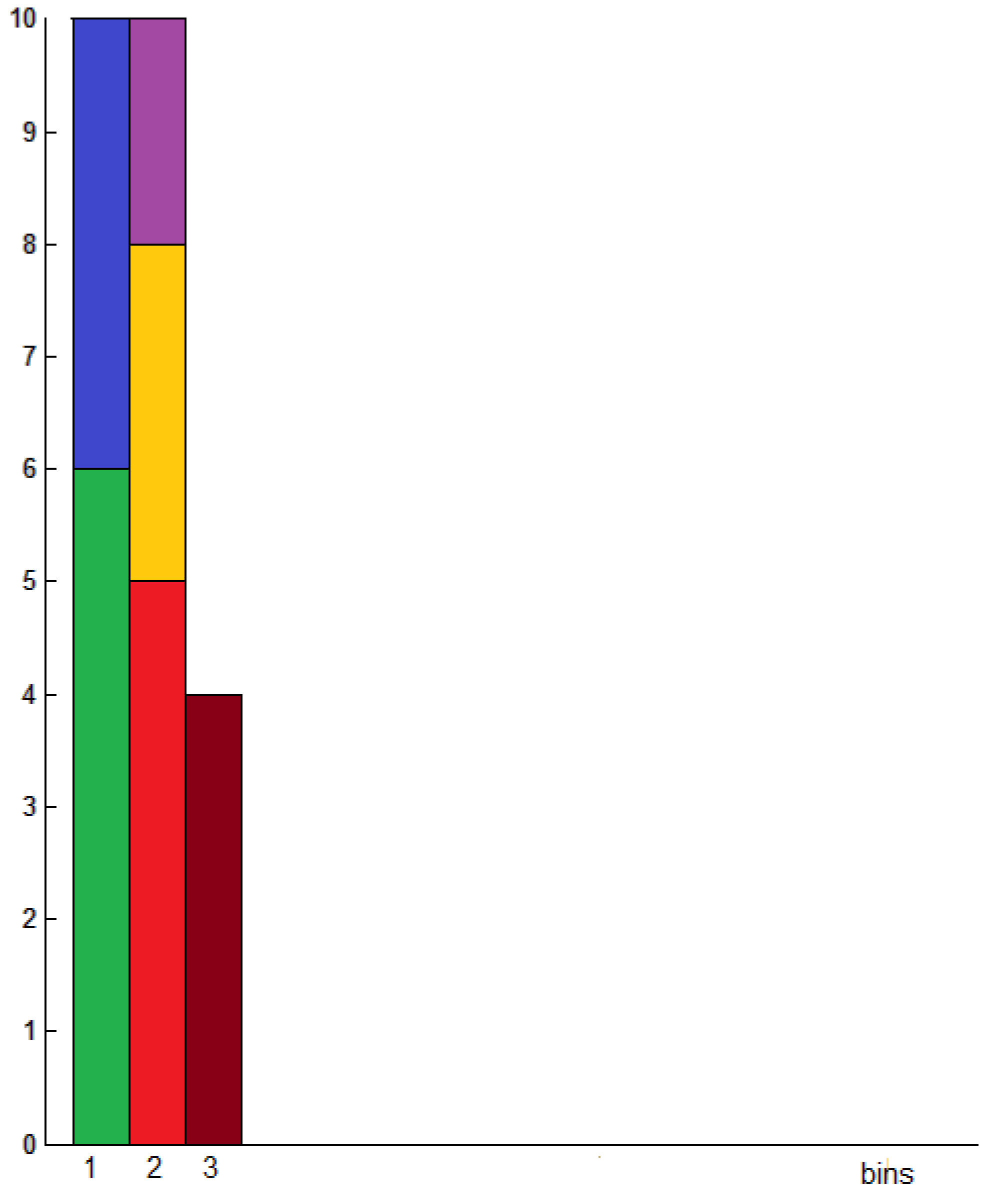
```
Παραγωγή πλήθους n αντικειμένων με τυχαία μήκη
Αρχή επανάληψης
   Εξέταση του πρώτου αντικειμένου από τη λίστα
   Έλεγχος διαστάσεων του αντικειμένου
   Εύρεση του πρώτου κάδου με διαθέσιμο χώρο
   Αν υπάρχει κάδος με διαθέσιμο χώρο τότε
      Τοποθέτηση του αντικειμένου
      Αναπροσαρμογή του διαθέσιμου χώρου του κάδου που τοποθετήθηκε
   Αλλιώς
      Τοποθέτηση του αντικειμένου σε νέο κάδο με αναπροσαρμογή του χώρου
   Τέλος αν
   Ανανέωση της λίστας αντικειμένων
   Απομάκρυνση του πρώτου αντικείμενου από τη λίστα
Τέλος επανάληψης
```

Για τη πλήρη κατανόηση της λειτουργίας του αλγορίθμου, παρουσιάζεται η εκτέλεσή του για ένα παράδειγμα του μονοδιάστατου προβλήματος πακετοποίησης. Σε αυτήν την περίπτωση, τα αντικείμενα έχουν μόνο μια διάσταση διαφοροποιημένη, το μήκος, που είναι ακέραιος αριθμός και επιλέγεται τυχαία για το κάθε αντικείμενο από το σύνολο [0 , 10]. Οι κάδοι έχουν μέγιστη χωρητικότητα $C=10$. Στα παρακάτω σχήματα δίνονται μερικά στιγμιότυπα από την εκτέλεση του First Fit αλγορίθμου για τις παραπάνω συνθήκες.



Εικόνα 4 - Τα παραγόμενα αντικείμενα κατά την εκτέλεση του First Fit αλγορίθμου με τα μήκη τους.

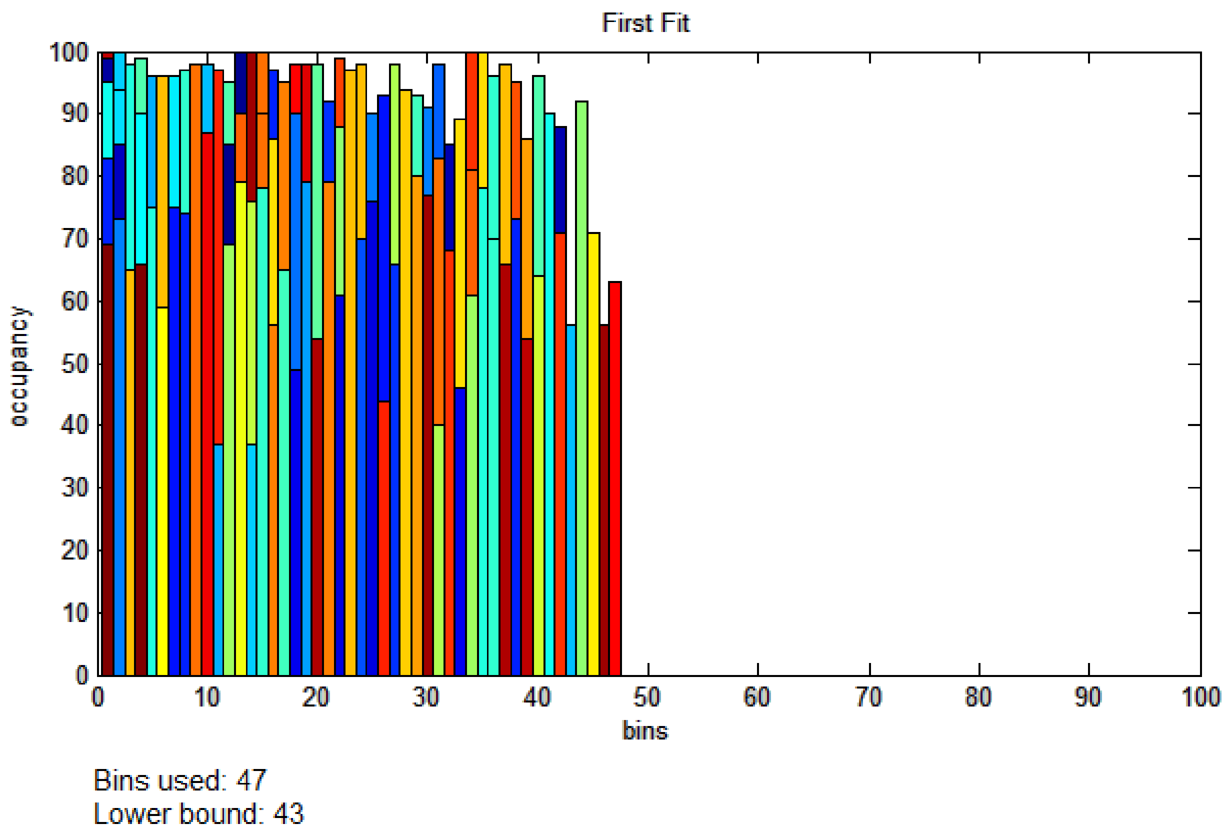
Στην παραπάνω εικόνα δίνονται τα μήκη των 6 αρχικών αντικειμένων που δημιουργήθηκαν. Το πρώτο αντικείμενο έχει μήκος 6, το δεύτερο 5, το τρίτο και το πέμπτο 4, το τέταρτο 3 και το έκτο 2. Στην First Fit μέθοδο επιλέγονται τα αντικείμενα με τη σειρά χωρίς κάποια αρχική επεξεργασία ξεκινώντας από το πρώτο. Το επόμενο στιγμιότυπο αναπαριστά το αποτέλεσμα πακετοποίησης των αντικειμένων αυτών με τη First Fit μέθοδο.



Εικόνα 5 - Παρουσίαση του αποτελέσματος πακετοποίησης των παραγόμενων αντικειμένων με τη μέθοδο First Fit.

Στην παραπάνω εικόνα παρουσιάζεται το αποτέλεσμα της πακετοποίησης με τη μέθοδο First Fit. Πιο αναλυτικά, το πρώτο αντικείμενο με μήκος 6 τοποθετείται στον πρώτο κάδο. Το δεύτερο αντικείμενο με μήκος 5 ξεπερνά τη χωρητικότητα του πρώτου κάδου άρα δημιουργείται νέος κάδος για την τοποθέτησή του. Το τρίτο αντικείμενο με μήκος 4 μπορεί να τοποθετηθεί στον πρώτο κάδο, ο οποίος μετα από αυτή την αποθήκευση δεν έχει άλλο διαθέσιμο χώρο. Επομένως, το τέταρτο αντικείμενο με μήκος 3 τοποθετείται απευθείας στον δεύτερο κάδο. Για το πέμπτο με μήκος 4 διαπιστώνεται ότι δεν υπάρχει χώρος στον δεύτερο κάδο άρα αποθηκεύεται σε τρίτο. Τέλος, για το έκτο αντικείμενο με μήκος 2 εξετάζονται με τη σειρά οι κάδοι και διαπιστώνεται ότι χωράει οριακά στον δεύτερο, όπου και τοποθετείται.

Τώρα θα θεωρηθεί μια περίπτωση που αντιπροσωπεύει καλύτερα τα προβλήματα πακετοποίησης που αντιμετωπίζονται. Σε αυτή την περίπτωση, το μήκος των αντικειμένων κυμαίνεται από 0 έως 100, και επεξεργάζονται 100 αντικείμενα συνολικά. Τα αντικείμενα που χρησιμοποιούνται είναι τα ίδια για όλες τις μεθόδους για τη συγκεκριμένη περίσταση. Η πακετοποίηση με τη First Fit μέθοδο δίνει τα παρακάτω αποτελέσματα.



Εικόνα 6 - Αποτέλεσμα First Fit μεθόδου για 100 αντικείμενα με μήκος (0 , 100].

Η First Fit μέθοδος έχει τη δυνατότητα να αποθηκεύσει τα αντικείμενα σε 47 κάδους. Το κάτω όριο (lower bound) υπολογίζεται αθροίζοντας τα μήκη όλων των αντικειμένων και διαιρώντας με την χωρητικότητα των κάδων. Ο αριθμός που προκύπτει είναι το ουτοπικό βέλτιστο και εξετάζεται κατά πόσο απέχουν από αυτό οι μέθοδοι που χρησιμοποιούνται. Ο αλγόριθμος υπολογίζει αυτόματα το κάτω όριο της κάθε περίπτωσης και για τη συγκεκριμένη προκύπτει ότι είναι 43.

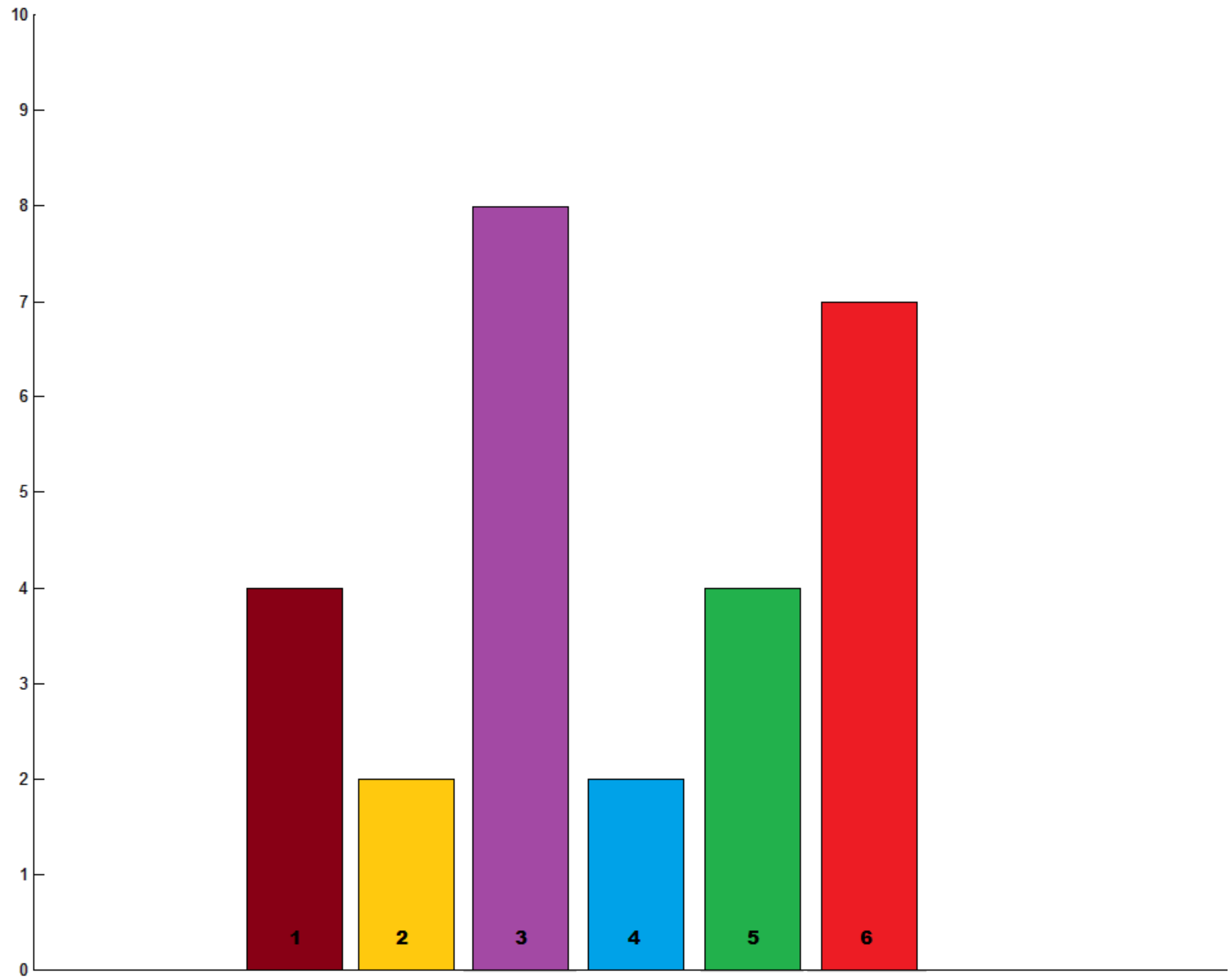
2.2.2 Best Fit (BF)

Η επόμενη προσέγγιση του προβλήματος πακετοποίησης που θα εξεταστεί αποτελείται από τη Best Fit μέθοδο. Και σε αυτή τη μέθοδο ο κάθε κάδος έχει αρχική χωρητικότητα $r = C$ και όταν τοποθετηθεί σε αυτόν το αντικείμενο *i* με διάσταση *u*, η υπολειπόμενη χωρητικότητα του κάδου μειώνεται κατά $r = r - u_i$. Όμως, η διαδικασία που ακολουθείται για την τοποθέτηση των αντικειμένων στους κάδους διαφοροποιείται από αυτή της First Fit μεθόδου, καθώς τα αντικείμενα δεν τοποθετούνται στον πρώτο κάδο που έχει διαθέσιμο χώρο. Στη μέθοδο αυτή, πριν τοποθετηθεί το αντικείμενο εξετάζονται όλοι οι κάδοι ως προς τη χωρητικότητά τους και τοποθετείται σε αυτόν που θα έχει τη μικρότερη υπολειπόμενη χωρητικότητα. Η μέθοδος μπορεί να περιγραφεί από τον παρακάτω ψευδοκώδικα:

Best Fit heuristic

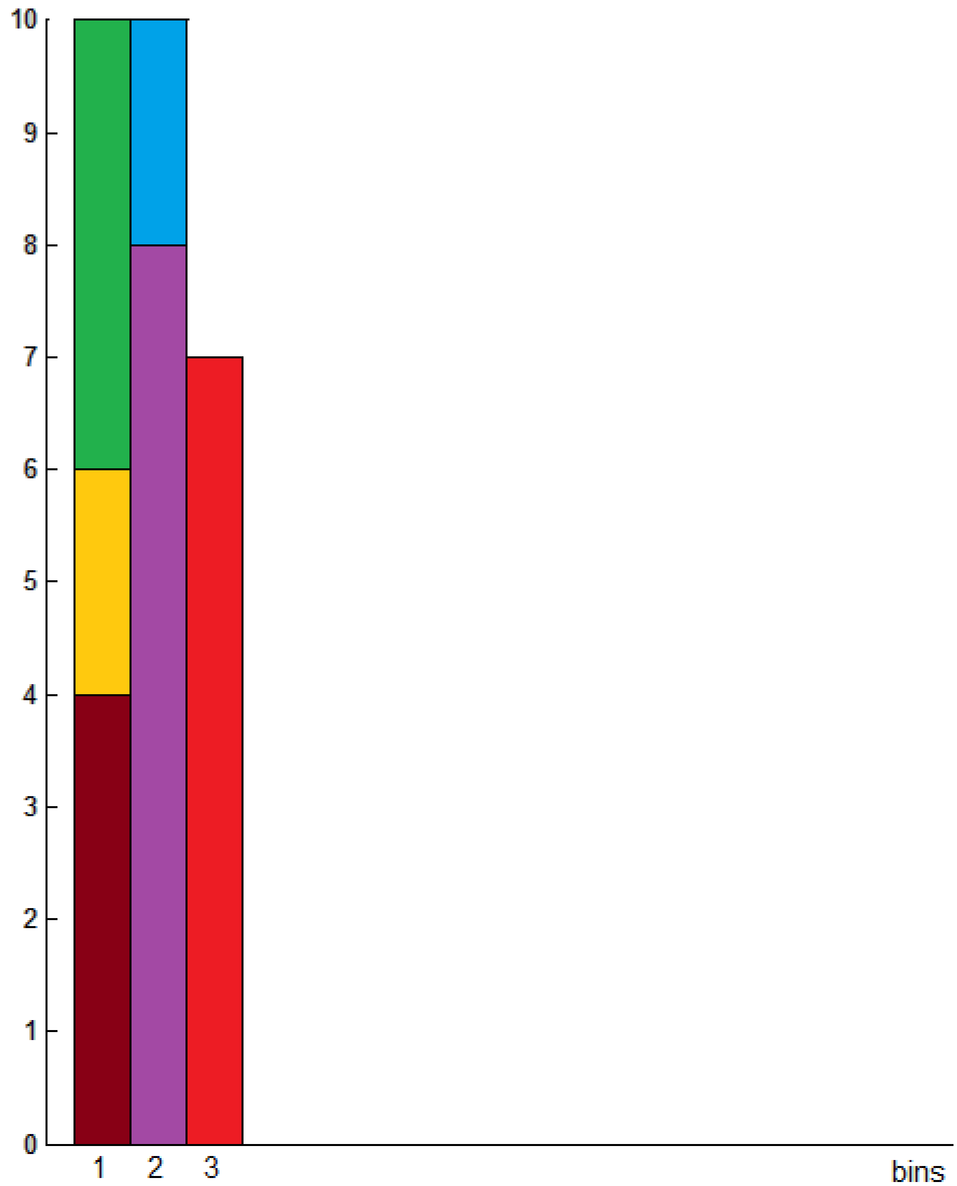
```
Παραγωγή πλήθους n αντικειμένων με τυχαία μήκη
Αρχή επανάληψης
   Εξέταση του πρώτου αντικειμένου από τη λίστα
   Έλεγχος διαστάσεων του αντικειμένου
   Εύρεση των κάδων με διαθέσιμο χώρο
   Αν υπάρχει κάδος με διαθέσιμο χώρο τότε
      Επιλογή του κάδου με τη μικρότερη υπολειπόμενη χωρητικότητα
      Τοποθέτηση του αντικειμένου
      Αναπροσαρμογή του διαθέσιμου χώρου του κάδου που τοποθετήθηκε
   Αλλιώς
      Τοποθέτηση του αντικειμένου σε νέο κάδο με αναπροσαρμογή του χώρου
   Τέλος αν
   Ανανέωση της λίστας αντικειμένων
   Απομάκρυνση του πρώτου αντικείμενου από τη λίστα
Τέλος επανάληψης
```

Όπως και στη προηγούμενη περίπτωση, θα παρουσιαστεί και εδώ ένα παράδειγμα της μεθόδου για ένα μικρού μεγέθους πρόβλημα για την καλύτερη κατανόησή της. Στο πρόβλημα αυτό προσδίδονται στα αντικείμενα τυχαία ακέραια μήκη με μέγιστο το 10, ενώ οι κάδοι έχουν χωρητικότητα 10. Παρουσιάζονται παρακάτω τα στιγμιότυπα του Best Fit αλγορίθμου για τις δεδομένες συνθήκες.



Εικόνα 7 - Τα παραγόμενα αντικείμενα κατά την εκτέλεση του Best Fit αλγορίθμου με τα μήκη τους.

Δίνονται τα μήκη των αντικειμένων που παράγονται κατά την εκτέλεση του Best Fit αλγορίθμου. Το πρώτο με το πέμπτο έχουν μήκος 4, το δεύτερο με το τέταρτο έχουν μήκος 2, το τρίτο έχει μήκος 8 και το έκτο έχει μήκος 7. Αντίστοιχα με τη First Fit μέθοδο, δεν γίνεται κάποια αρχική επεξεργασία των αντικειμένων αλλά επιλέγεται το καθέ ένα διαδοχικά ξεκινώντας από το πρώτο. Η πακετοποίηση των αντικειμένων αυτών χρησιμοποιώντας τη Best Fit μέθοδο καταλήγει στο αποτέλεσμα που παρουσιάζεται παρακάτω.

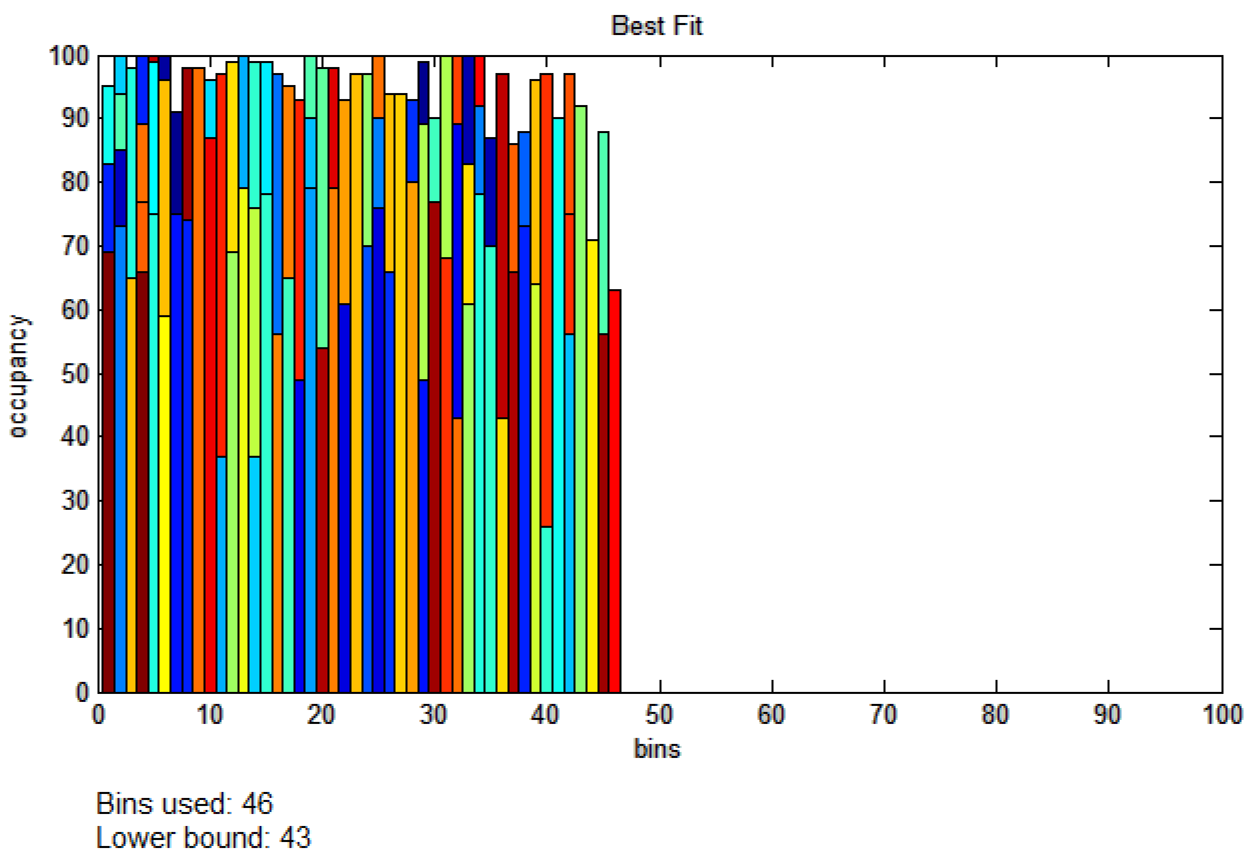


Εικόνα 8 - Παρουσίαση του αποτελέσματος πακετοπίησης των παραγόμενων αντικειμένων με τη μέθοδο Best Fit.

Όπως φαίνεται στην παραπάνω εικόνα, το πρώτο με το δεύτερο αντικείμενο, με διαστάσεις 4 και 6 αντίστοιχα, τοποθετούνται στον πρώτο κάδο. Το τρίτο αντικείμενο με μήκος 8 ξεπερνά τη χωρητικότητα του πρώτου άρα τοποθετείται σε νέο κάδο. Για το τέταρτο αντικείμενο με μήκος 2 παρατηρείται ότι υπάρχει διαθέσιμος χώρος στον πρώτο και στον δεύτερο κάδο, όμως επιλέγεται να τοποθετηθεί στον δεύτερο για να συμπληρωθεί η χωρητικότητα αυτού. Παρατηρείται ότι στη First Fit μέθοδο θα είχε τοποθετηθεί απευθείας στον πρώτο κάδο χωρίς περαιτέρω έλεγχο. Το πέμπτο αντικείμενο με μήκος 4 τοποθετείται στον πρώτο κάδο, ενώ δημιουργείται τρίτος κάδος για να τοποθετηθεί το έκτο αντικείμενο με μήκος 7. Διαπιστώνεται ότι χρησιμοποιούνται συνολικά τρεις κάδοι για την πακετοποίηση.

Στο σημείο διαφοροποίησης από τη First Fit μέθοδο, δηλαδή μετά την τοποθέτηση του τέταρτου αντικειμένου οι κάδοι έχουν υπολειπόμενη χωρητικότητα 4, 0 κατά σειρά. Έτσι, για το πέμπτο αντικείμενο μήκους 4 υπάρχει χώρος να τοποθετηθεί στον πρώτο κάδο, τοποθετώντας το έκτο σε νέο. Αν είχε χρησιμοποιηθεί η First Fit μέθοδος, η υπολειπόμενη χωρητικότητα θα ήταν 2, 2 αφού το τέταρτο αντικείμενο θα είχε τοποθετηθεί στον πρώτο κάδο. Έτσι, για το πέμπτο αντικείμενο με μήκος 4 θα χρειαζόταν να χρησιμοποιηθεί τρίτος κάδος και στη συνέχεια τέταρτος για το έκτο αντικείμενο με μήκος 7, καθώς δεν θα χώραγε στον τρίτο.

Το επόμενο παράδειγμα αφορά την πιο σύνθετη περίπτωση πακετοποίησης 100 αντικειμένων με μήκη από 0 έως 100. Για τη συγκεκριμένη περίσταση, τα αντικείμενα που χρησιμοποιούνται είναι τα ίδια με αυτά της First Fit μεθόδου. Η Best Fit μέθοδος δίνει τα παρακάτω αποτελέσματα.

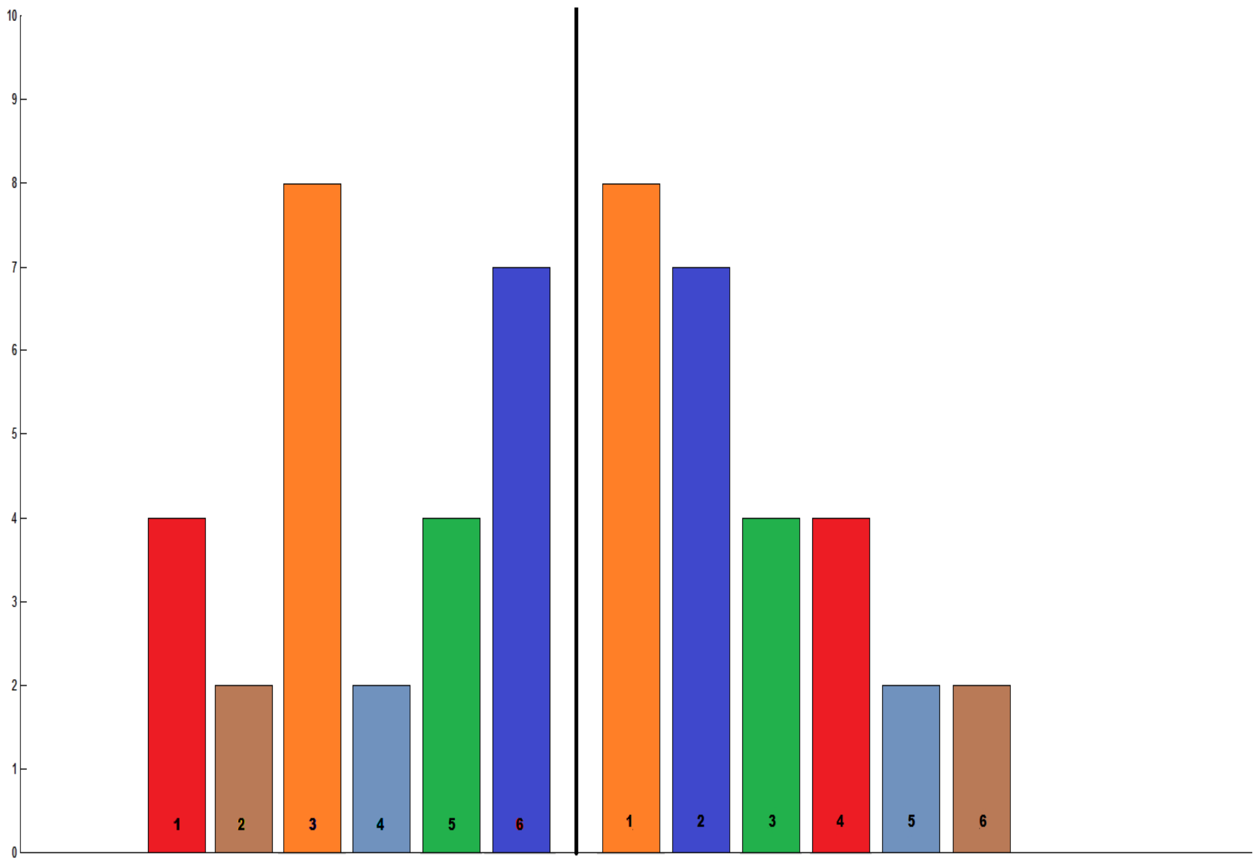


Εικόνα 9 - Αποτελέσματα Best Fit μεθόδου για 100 αντικείμενα με μήκος (0 , 100].

Με τη Best Fit μέθοδο, τα αντικείμενα αποθηκεύονται με τη χρήση 46 κάδων. Εφόσον χρησιμοποιήθηκαν τα ίδια αντικείμενα, το κάτω όριο παραμένει σταθερό.

2.2.3 First Fit Decreasing (FFD)

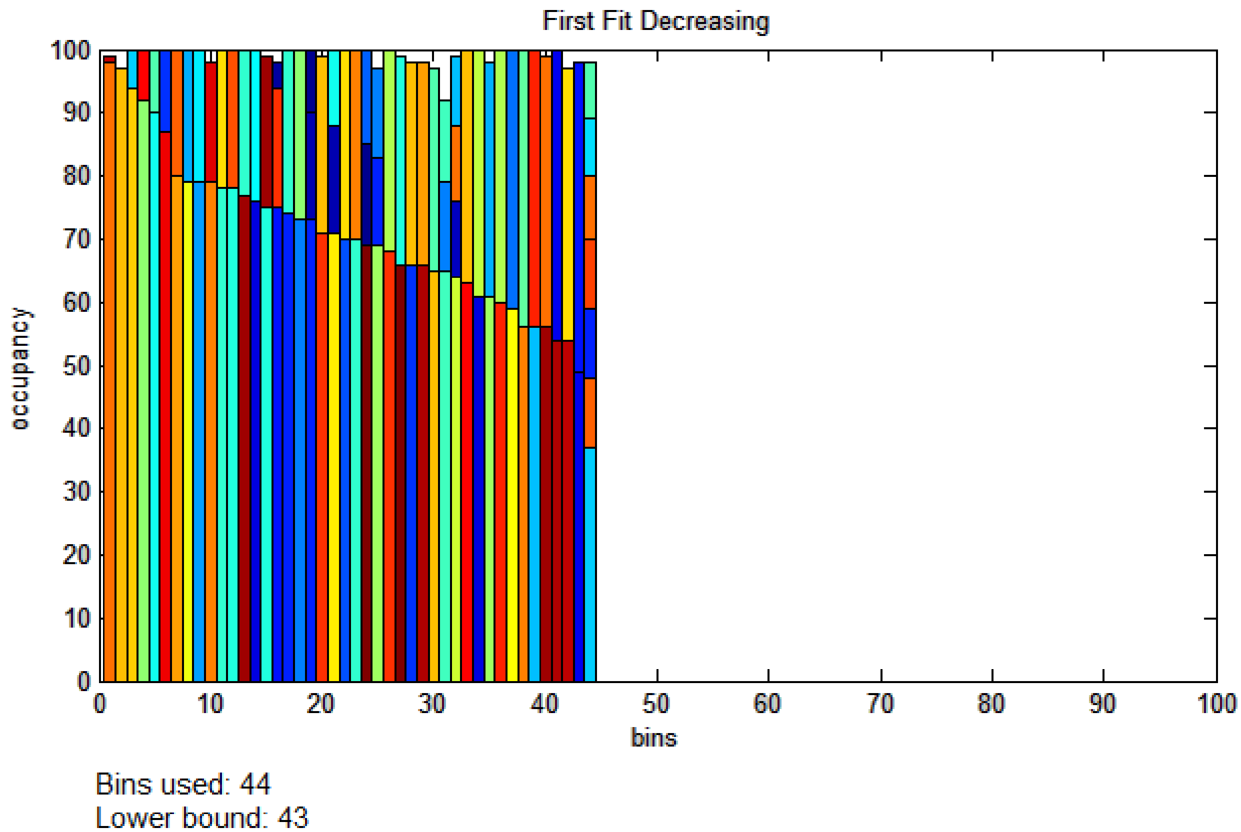
Η First Fit Decreasing μέθοδος είναι σχεδόν ίδια με τη First Fit με μία βασική διαφοροποίηση: Τα παραγόμενα αντικείμενα στην αρχή του αλγορίθμου, πριν αρχίσουν να επιλέγονται για τοποθέτηση στους κάδους, ταξινομούνται κατά φθίνουσα σειρά μήκους. Αυτό όπως αποδεικνύεται βελτιώνει κατά πολύ τα αποτελέσματα της μεθόδου.



Εικόνα 10 - Ταξινόμηση των αντικειμένων κατά φθίνουσα σειρά ως προς το μήκος.

Μετά την ταξινόμηση εφαρμόζεται ο ίδιος αλγόριθμος με αυτόν της First Fit μεθόδου. Επιλέγεται κανονικά το πρώτο αντικείμενο από τη λίστα (αυτό με το μεγαλύτερο μήκος) και τοποθετείται στον πρώτο διαθέσιμο κάδο. Συνεχίζεται η διαδικασία για το αμέσως επόμενο αντικείμενο μέχρι την τοποθέτηση όλων των αντικειμένων.

Για το σύνθετο παράδειγμα των 100 αντικειμένων που εφαρμόζεται σε όλες τις μεθόδους, η First Fit Decreasing μέθοδος παρουσιάζει τα παρακάτω αποτελέσματα.



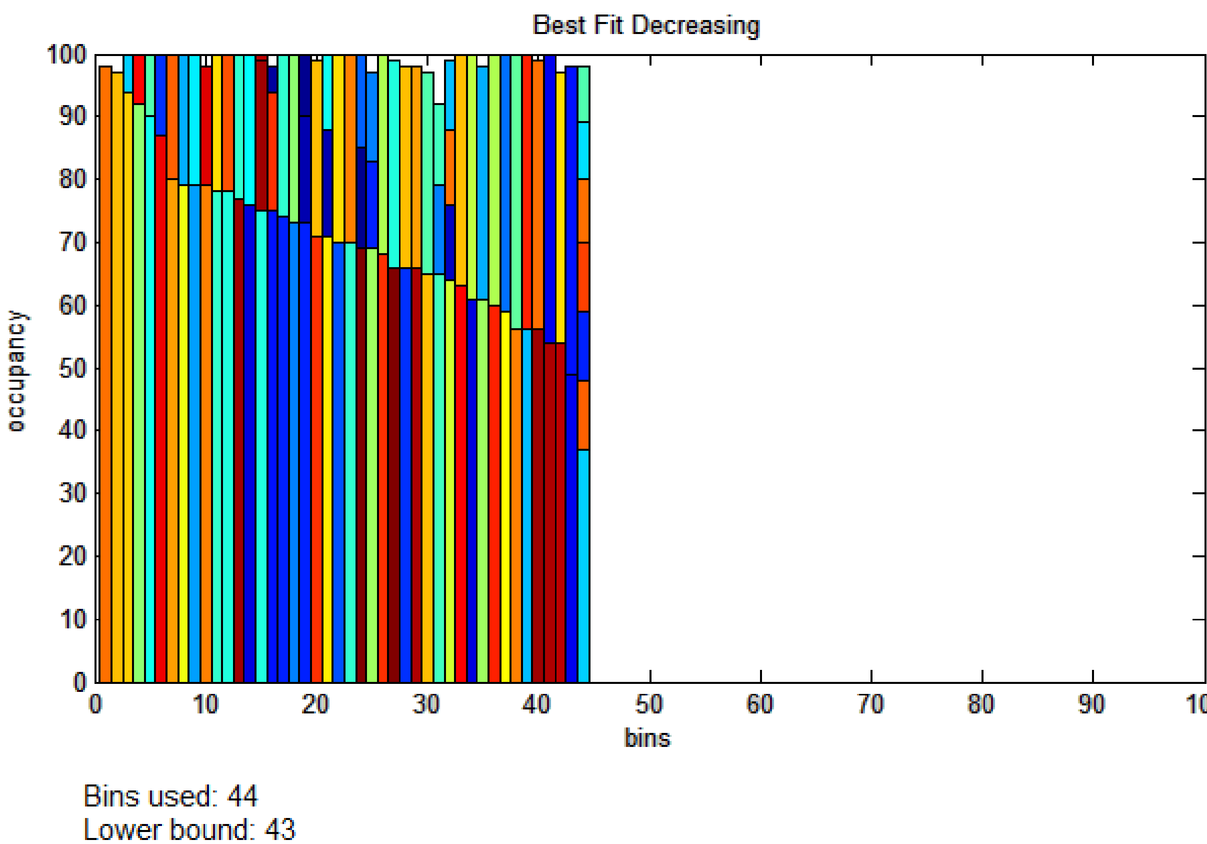
Εικόνα 11 - Αποτελέσματα First Fit Decreasing μεθόδου για 100 αντικείμενα με μήκος (0 , 100]

Επιλύοντας το πρόβλημα πακετοποίησης με τη First Fit Decreasing μέθοδο, είναι δυνατή η αποθήκευση των αντικειμένων χρησιμοποιώντας 44 κάδους, μόλις έναν παραπάνω από το κάτω όριο. Το αποτέλεσμα αυτό ξεπερνά σε ποιότητα τα αποτελέσματα των προηγούμενων μεθόδων (First Fit 47 bins, Best Fit 46 bins). Παρατηρείται λοιπόν η σημασία της ταξινόμησης στην εκκίνηση του αλγορίθμου: Τοποθετώντας αρχικά τα μεγαλύτερα αντικείμενα συμβάλει στην χρησιμοποίηση λιγότερων κάδων, άρα και καλύτερων αποτελεσμάτων. Υπάρχει όμως ένα βασικό μειονέκτημα αυτής της μεθόδου σε σχέση με τις προηγούμενες, το γεγονός ότι για να χρησιμοποιηθεί όλα τα αντικείμενα πρέπει να είναι προκαθορισμένα. Δεν είναι δυνατή δηλαδή η πρόσθεση κάποιου αντικειμένου κατά τη λειτουργία του αλγορίθμου καθώς πριν ξεκινήσει ο αλγόριθμος πακετοποίησης πρέπει να γίνει ταξινόμηση των αντικειμένων. Οσες μέθοδοι έχουν αυτόν τον περιορισμό χαρακτηρίζονται offline μέθοδοι ενώ όσες δεν τον έχουν χαρακτηρίζονται online.

2.2.4 Best Fit Decreasing (BFD)

Αντίστοιχα με τη First Fit Decreasing, η Best Fit Decreasing μέθοδος είναι ίδια με τη Best Fit ενώ έχει προηγηθεί η ταξινόμηση των αντικειμένων. Τα αντικείμενα στην αρχή του αλγορίθμου ταξινομούνται κατα φθίνουσα σειρά ως προς το μήκος και στη συνέχεια επιλέγεται το κάθε ένα διαδοχικά για τοποθέτηση, ξεκινώντας από αυτό με τη μεγαλύτερη διάσταση. Εξετάζονται όλοι οι κάδοι για να τοποθετηθεί σε αυτόν με τη μικρότερη υπολειπόμενη χωρητικότητα, ενώ στην περίπτωση που δεν υπάρχει διαθέσιμος χώρος σε κάποιον από τους διαθέσιμους κάδους, το αντικείμενο τοποθετείται σε νέο κάδο.

Για το πρόβλημα των 100 αντικειμένων με τυχαία μήκη από 0 έως 100, η Best Fit Decreasing μέθοδος παρουσιάζει τα παρακάτω αποτελέσματα.



Εικόνα 12 - Αποτελέσματα Best Fit Decreasing μεθόδου για 100 αντικείμενα με μήκος (0 , 100].

Παρατηρείται ότι με τη χρήση της Best Fit Decreasing μεθόδου χρησιμοποιούνται 44 κάδοι, όσοι δηλαδή χρησιμοποιούνται και με τη First Fit Decreasing. Προκύπτει δηλαδή το συμπέρασμα ότι με την ταξινόμηση των αντικειμένων στην αρχή γίνεται η καλύτερη δυνατή πακετοποίηση, με την επιλογή ανάμεσα στις μεθόδους First Fit, Best Fit να έχει δευτερεύουσα σημασία.

2.3 Σύνοψη 2ου κεφαλαίου

Στο κεφάλαιο αυτό έγινε ανάλυση των ευρετικών τεχνικών που χρησιμοποιήθηκαν για την αντιμετώπιση του μονοδιάστατου προβλήματος πακετοποίησης σε κάδους. Οι τεχνικές αυτές χρησιμοποιήθηκαν σε ένα δείγμα 100 αντικειμένων για μια αρχική αναπαράσταση των αποτελεσμάτων και μια πρώτη σύγκριση της αποδοτικότητας.

Η First Fit αποτελεί την πιο απλή μέθοδο, όπου το κάθε αντικείμενο τοποθετείται στον πρώτο κάδο με διαθέσιμο χώρο. Η Best Fit περιέχει μια παραπάνω επεξεργασία, καθως τοποθετεί το κάθε αντικείμενο στον κάδο με τη μικρότερη υπολειπόμενη χωρητικότητα μετά την τοποθέτηση του, βελτιώνοντας συνήθως το αποτέλεσμα. Οι τεχνικές First Fit Decreasing, Best Fit Decreasing αντιστοιχούν στις First Fit, Best Fit με τη διαφορά ότι στην αρχή της λειτουργίας τους ταξινομούν τα αντικείμενα κατα φθίνουσα σειρά ως προς το μήκος. Αυτό έχει ιδιαίτερη σημασία για την ποιότητα της λύσης που προτείνεται από τον αλγόριθμο, όπως θα διαπιστωθεί στο επόμενο κεφάλαιο.

Κεφάλαιο 3

Αποτελέσματα σε πίνακες

3.1 Περιγραφή των τεχνικών σύγκρισης

Στο σημείο αυτό επιχειρείται η σύγκριση των ευρετικών τεχνικών που αναλύθηκαν στο προηγούμενο κεφάλαιο. Όλοι οι κώδικες των ευρετικών αλγορίθμων έχουν διατυπωθεί σε γλώσσα MATLAB, η οποία παρέχει τη δυνατότητα γραφικής αναπαράστασης των αποτελεσμάτων. Η γραφική αναπαράσταση είναι εξαιρετικά χρήσιμη, όχι μόνο γιατί δημιουργεί όμορφη και σαφή εικόνα για τα αποτελέσματα, αλλά και επειδή είναι δυνατός ο σωστός έλεγχος του κώδικα σε κάθε χρονική στιγμή. Επίσης, με τη βοήθεια των στιγμιοτύπων έγινε κατανοητός ο τρόπος λειτουργίας της κάθε τεχνικής.

Παρατηρήθηκε η επίδοση των ευρετικών μεθόδων για μία περίπτωση προβλήματος, αυτή των 100 αντικειμένων με τυχαία μήκη από 0 έως 100. Σε αυτήν την περίπτωση, η καλύτερη λύση δόθηκε από τις First Fit Decreasing, Best Fit Decreasing μεθόδους χρησιμοποιώντας 44 κάδους. Παρόλα αυτά, για να βγουν ασφαλή συμπεράσματα είναι αναγκαία η εξέταση περισσότερων περιπτώσεων του μονοδιάστατου προβλήματος πακετοποίησης με διαφορετικές συνθήκες. Για τον λόγο αυτό, έγιναν μερικές διαφοροποιήσεις στο αρχικό πρόβλημα και καταγράφηκαν τα αποτελέσματα, τα οποία υποβλήθηκαν σε περαιτέρω ανάλυση. Διαπιστώθηκε η αποτελεσματικότητα των μεθόδων για διάφορα πλήθη αντικειμένων των οποίων τα μήκη επιλέγονται από διαφορετικά σύνολα, και οι επιδόσεις καταγράφηκαν σε πίνακες υπολογίζοντας την απόσταση του αποτελέσματος της κάθε μεθόδου από το κάτω όριο για την κάθε περίσταση. Με τα δεδομένα αυτά, παρουσιάζεται με σαφήνεια ποια μέθοδος ανταποκρίνεται καλύτερα για τους εξεταζόμενους τύπους προβλημάτων και έγινε ασφαλέστερη η διεξαγωγή συμπερασμάτων.

3.2 Ανάλυση των αποτελεσμάτων

Στους επόμενους πίνακες έχουν συλλεχθεί τα αποτελέσματα για κάποιες διαφορετικές περιπτώσεις προβλημάτων. Στην πρώτη περίπτωση, τα μήκη των αντικειμένων έχουν επιλεγεί τυχαία μέσα από το διάστημα [20 , 80]. Τα αντικείμενα αυτά ανήκουν στην κατηγορία A η οποία εκπροσωπεί συχνές πρακτικές καταστάσεις όπου τα αντικείμενα δεν μπορεί να είναι πολύ μικρά ή πολύ μεγάλα σε σχέση με τη χωρητικότητα του κάδου. Για τον λόγο αυτό εξαιρούνται τα διαστήματα [0 , 20] και [80 , 100], ενώ η χωρητικότητα των κάδων παραμένει 100. Για τα αντικείμενα της κατηγορίας B επιλέγονται μήκη από τα διαστήματα [20 , 40] και [60 , 80], με σκοπό να αποκλειστούν τα μεσαίου μεγέθους και να διαπιστωθεί η λειτουργία των μεθόδων σε μικρά ή μεγάλα αντικείμενα. Οι εκτελέσεις των αλγορίθμων έχουν γίνει για 5 διαφορετικά πλήθη αντικειμένων, πιο συγκεκριμένα για 20, 50, 100, 200 και 500 αντικείμενα. Αρχικά παρουσιάζονται τα αποτελέσματα που προκύπτουν για 20 εκτελέσεις του αλγορίθμου. Επίσης, υπολογίζεται στους παρακάτω πίνακες το κάτω όριο (Lower bound, LB) αθροίζοντας τα μήκη των αντικειμένων και διαιρώντας με την χωρητικότητα των κάδων.

3.2.1 Αποτελέσματα μεθόδων για αντικείμενα κατηγορίας Α

Σε αυτήν την κατηγορία ανήκουν τα αντικείμενα με μήκη που επιλέγονται τυχαία από το διάστημα [20 , 80]. Γίνονται 20 εκτελέσεις του συνολικού ευρετικού αλγορίθμου που περιλαμβάνει όλες τις μεθόδους πακετοποίησης για κάθε έναν από τους διαφορετικούς αριθμούς αντικειμένων, δηλαδή για 20, για 50, για 100, για 200 και για 500 αντικείμενα υπολογίζοντας σε κάθε εκτέλεση το κάτω όριο.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	13	13	12	12	10.25
2	12	12	11	11	10.1
3	14	14	14	14	10.56
4	13	13	13	13	10.44
5	14	14	14	14	10.6
6	11	11	11	11	9.16
7	11	11	10	10	9.27
8	14	14	13	13	10.58
9	13	13	13	13	10.91
10	14	14	14	14	11.21
11	11	11	10	10	9.48
12	13	13	12	12	10.57
13	13	12	12	12	10.16
14	14	13	13	13	10.75
15	12	12	11	11	9.96
16	12	12	12	12	10.28
17	11	11	11	11	9.76
18	13	13	13	13	10.67
19	13	13	12	12	10.24
20	12	12	11	11	9.36

Πίνακας 1. Αποτελέσματα των μεθόδων για n=20 αντικείμενα κατηγορίας Α.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	28	28	27	27	25.21
2	28	28	26	26	24.72
3	30	29	26	26	24.93
4	28	28	26	26	24.69
5	27	27	26	26	24.2
6	30	30	28	28	25.38
7	31	30	27	27	25.25
8	27	26	25	25	23.23
9	29	28	26	26	24.81
10	28	28	27	27	24.83
11	32	32	29	29	26.32
12	29	28	27	27	25.16
13	27	26	25	25	23.52
14	29	28	27	27	25.33
15	30	30	29	29	26.44
16	31	30	30	30	25.93
17	29	28	26	26	24.88
18	32	31	27	27	24.67
19	31	30	29	29	26
20	33	33	31	31	27

Πίνακας 2. Αποτελέσματα των μεθόδων για n=50 αντικείμενα κατηγορίας A.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	56	55	53	53	50.6
2	54	53	53	53	49.52
3	55	54	53	53	49.93
4	56	55	52	52	50.39
5	60	60	55	55	51.77
6	56	54	52	52	50.25
7	54	53	50	50	48.78
8	56	56	52	52	48.51
9	65	65	63	63	53.2
10	55	55	52	52	49.18
11	56	55	53	53	50.41
12	56	56	51	51	49.03
13	51	51	48	48	46.27
14	60	59	57	57	52.29
15	67	66	64	64	53.04
16	59	59	54	54	50.94
17	59	59	56	56	51.06
18	55	55	51	51	49.18
19	57	57	50	50	48.44
20	61	60	57	57	52.57

Πίνακας 3. Αποτελέσματα των μεθόδων για n=100 αντικείμενα κατηγορίας A.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	115	114	110	110	102.63
2	107	105	100	100	98.05
3	110	108	103	103	99.91
4	112	108	106	106	100.71
5	102	100	96	96	94.56
6	115	113	106	106	102.21
7	122	121	120	120	105.44
8	110	109	104	104	99.79
9	111	110	107	107	100.16
10	111	108	102	102	99.6
11	109	108	101	101	98.9
12	116	115	107	107	101.82
13	116	112	112	112	102.57
14	115	113	108	108	102.61
15	115	112	109	109	102.01
16	109	108	102	102	99.62
17	115	112	109	109	102.82
18	109	106	103	103	99.75
19	115	113	108	108	101.88
20	116	113	108	108	102.5

Πίνακας 4. Αποτελέσματα των μεθόδων για n=200 αντικείμενα κατηγορίας Α.

Αριθμος Εκτέλεσης	FF	BF	FFD	BFD	LB
1	256	252	246	246	242.64
2	271	265	261	261	251.58
3	271	266	256	256	248.58
4	266	261	251	251	247.59
5	271	266	254	254	250.01
6	275	268	257	257	250.46
7	262	258	248	248	244.24
8	271	269	259	259	251.1
9	274	270	263	263	253.57
10	275	272	259	259	251.44
11	274	271	254	254	249.61
12	268	264	254	254	249.51
13	277	272	263	263	253.73
14	262	258	250	250	246.14
15	260	256	248	248	245.11
16	281	279	274	274	256.12
17	267	263	251	251	247.99
18	268	263	258	258	250.18
19	280	276	256	256	250.54
20	276	271	255	255	250.7

Πίνακας 5. Αποτελέσματα των μεθόδων για n=500 αντικείμενα κατηγορίας Α.

3.2.2 Αποτελέσματα μεθόδων για αντικείμενα κατηγορίας Β

Στην κατηγορία Β ανήκουν τα αντικείμενα των οποίων τα μήκη επιλέγονται τυχαία από το διάστημα [20 , 40]∪[60 , 80]. Δηλαδή, η κατηγορία αυτή αποτελείται από μικρά ή μεγάλα αντικείμενα και εξετάζεται η αποτελεσματικότητα των μεθόδων πακετοποίησης τέτοιων περιπτώσεων. Στους παρακάτω πίνακες παρουσιάζονται τα αποτελέσματα 20 εκτελέσεων του συνολικού ευρετικού αλγορίθμου που περιλαμβάνει όλες τις μεθόδους για 20, 50 ,100 ,200 και 500 αντικείμενα κατηγορίας Β.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	15	15	15	15	12
2	12	12	11	11	9.72
3	11	11	10	10	9.26
4	8	8	8	8	7.17
5	14	13	13	13	10.6
6	11	11	9	9	8.82
7	13	13	12	12	10.81
8	12	12	12	12	10.44
9	13	13	13	13	10.18
10	11	11	11	11	9.67
11	12	12	11	11	9.9
12	16	16	16	16	12.51
13	12	12	11	11	10.15
14	11	11	10	10	9.43
15	14	14	14	14	11.47
16	12	12	12	12	10.1
17	13	13	13	13	11.01
18	13	13	13	13	10.44
19	12	11	11	11	9.52
20	14	14	13	13	10.88

Πίνακας 6. Αποτελέσματα των μεθόδων για n=20 αντικείμενα κατηγορίας Β.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	33	32	30	30	26.32
2	28	28	25	25	24.25
3	26	26	25	25	24.08
4	28	28	27	27	24.86
5	29	29	27	27	24.9
6	31	31	28	28	26.23
7	30	30	29	29	25.74
8	30	30	29	29	25.76
9	29	29	27	27	24.85
10	29	29	28	28	25.64
11	26	25	25	25	23.4
12	30	30	28	28	25.54
13	27	27	26	26	24.37
14	26	26	24	24	23.11
15	28	28	26	26	24.65
16	32	32	32	32	26.9
17	31	30	28	28	25.74
18	27	27	25	25	23.75
19	30	28	27	27	25.65
20	29	28	26	26	24.32

Πίνακας 7. Αποτελέσματα των μεθόδων για n=50 αντικείμενα κατηγορίας Β.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	57	55	54	54	51.37
2	56	56	55	55	50.78
3	53	52	50	50	48.16
4	50	49	47	47	46.22
5	57	57	52	52	50.26
6	53	52	50	50	48.46
7	52	51	49	49	48.62
8	56	55	51	51	49.53
9	51	51	49	49	47.97
10	62	61	58	58	53.56
11	57	55	54	54	50.89
12	55	54	50	50	48.79
13	56	55	52	52	50.15
14	57	56	52	52	49.91
15	59	59	57	57	52.42
16	52	51	50	50	48.33
17	57	57	52	52	49.49
18	58	58	54	54	51.69
19	55	54	51	51	49.3
20	55	55	54	54	49.18

Πίνακας 8. Αποτελέσματα των μεθόδων για n=100 αντικείμενα κατηγορίας Β.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	108	106	104	104	99.51
2	112	110	106	106	102.27
3	113	110	110	110	102.57
4	120	118	114	114	105.78
5	104	103	98	98	96.57
6	109	106	101	101	98.89
7	111	110	103	103	100.7
8	107	105	101	101	98.83
9	105	104	100	100	98.16
10	108	106	102	102	99.36
11	113	111	109	109	102.87
12	113	111	108	108	103.1
13	109	107	102	102	99.15
14	107	106	102	102	98.53
15	105	103	99	99	96.92
16	110	109	104	104	100.78
17	114	112	108	108	101.72
18	99	97	95	95	93.66
19	107	106	102	102	99.14
20	107	103	100	100	97.88

Πίνακας 9. Αποτελέσματα των μεθόδων για n=200 αντικείμενα κατηγορίας Β.

Αριθμός Εκτέλεσης	FF	BF	FFD	BFD	LB
1	264	260	250	250	247.55
2	273	270	259	259	250.73
3	261	258	247	247	244.8
4	276	274	267	267	255.96
5	278	272	265	265	254.35
6	264	260	253	253	247.73
7	270	267	255	255	250.24
8	274	269	260	260	252.58
9	259	253	249	249	246.33
10	267	260	256	256	250.88
11	266	263	250	250	247.43
12	255	252	243	243	241.41
13	269	265	256	256	251.65
14	262	256	251	251	246.28
15	266	260	258	258	251.74
16	251	247	243	243	240.75
17	259	253	248	248	245.95
18	282	276	269	269	258.16
19	274	270	260	260	251.52
20	258	253	242	242	241.16

Πίνακας 10. Αποτελέσματα των μεθόδων για n=500 αντικείμενα κατηγορίας Β.

3.2.3 Σύγκριση αποτελεσμάτων

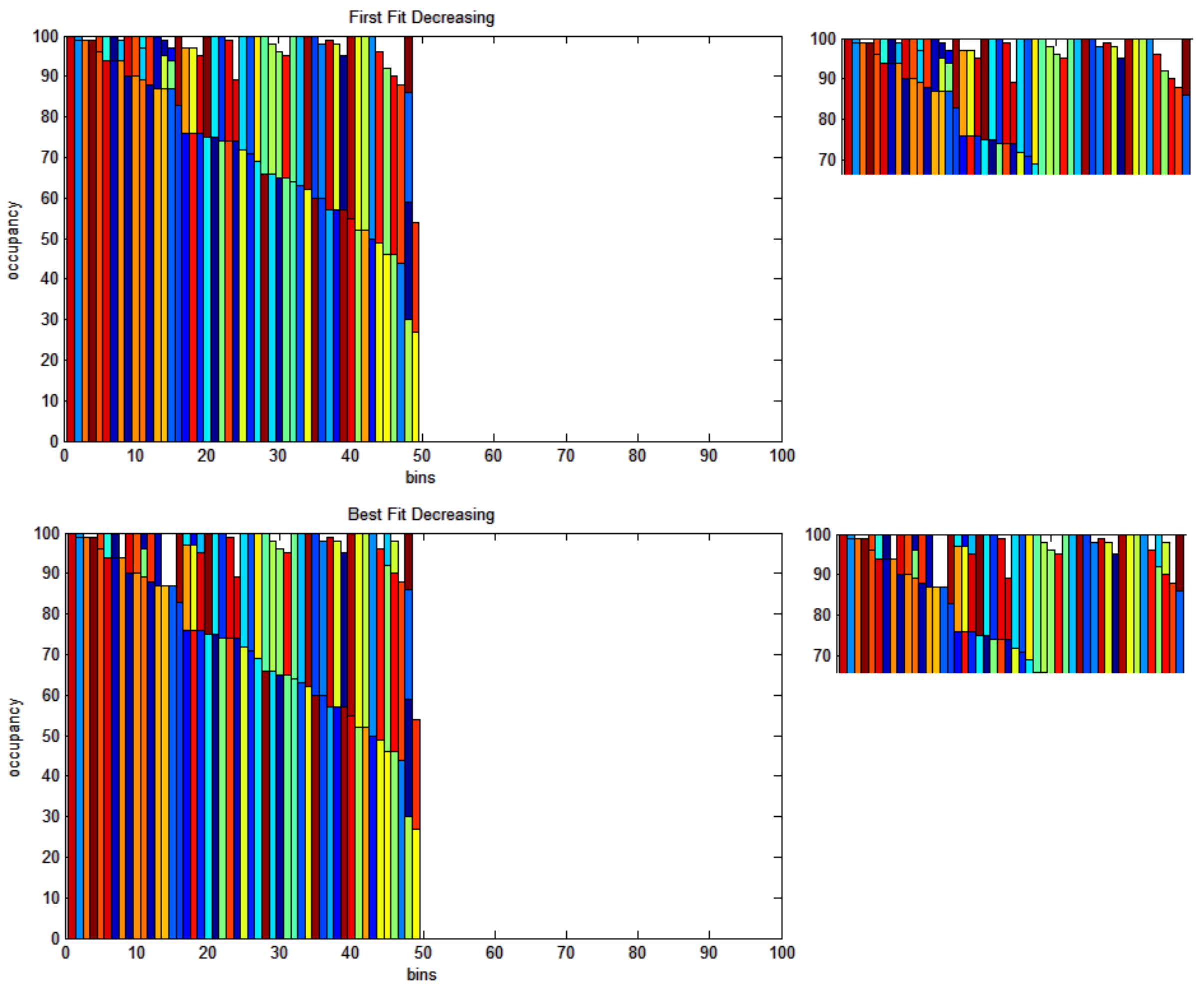
Για μια πιο συγκεντρωτική απεικόνιση των αποτελεσμάτων, στον παρακάτω πίνακα παρουσιάζονται τα αποτελέσματα και για τις δύο κατηγορίες αντικειμένων. Υπολογίζεται ο μέσος όρος των αποτελεσμάτων της κάθε μεθόδου από τις εκτελέσεις του αλγορίθμου που έγιναν για τον κάθε αριθμό αντικειμένων και για τα κάτω όρια που προκύπτουν. Στο δεύτερο μέρος του πίνακα υπολογίζεται η απόσταση της κάθε μεθόδου από το κάτω όριο της κάθε περίστασης που προκύπτει από τους μέσους όρους των εκτελέσεων κάθε κατηγορίας αντικειμένων.

Πλήθος αντικειμένων n	Κατηγορία	FF	BF	FFD	BFD	LB
20	A	12.65	12.55	12.1	12.1	10.21
50	A	29.45	28.9	27.2	27.2	25.12
100	A	57.4	56.85	53.8	53.8	50.26
200	A	112.5	110.4	106.05	106.05	100.87
500	A	270.25	266	255.85	255.85	249.54
20	B	12.45	12.35	11.9	11.9	10.2
50	B	28.95	28.65	27.1	27.1	25
100	B	55.4	54.65	52.05	52.05	49.75
200	B	109.05	107.15	103.4	103.4	99.81
500	B	266.4	261.9	254.05	254.05	248.86
Απόσταση από το κάτω όριο		FF	BF	FFD	BFD	LB
Αντικείμενα κατηγορίας A		9.25	7.74	3.8	3.8	-
Αντικείμενα κατηγορίας B		7.73	6.22	2.98	2.98	-
Όλα τα αντικείμενα		8.48	6.97	3.38	3.38	-

Πίνακας 11. Λύσεις ευρετικών μεθόδων με τη μορφή μέσων όρων και αποστάσεις από τα κάτω όρια.

Παρατηρώντας τα αποτελέσματα που διαγράφονται στους παραπάνω πίνακες προκύπτουν τα εξής συμπεράσματα:

- Οι μέθοδοι First Fit Decreasing, Best Fit Decreasing έχουν ακριβώς τα ίδια αποτελέσματα. Χρειάζονται τον ίδιο αριθμό κάδων σε όλες τις περιπτώσεις, αλλά ορισμένες φορές διαφοροποιούνται ως προς τον τρόπο αξιοποίησης των κάδων αυτών, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 13 - Γραφική σύγκριση First Fit Decreasing - Best Fit Decreasing.

Όπως παρατηρείται στην παραπάνω εικόνα από μια τυχαία περίπτωση προβλήματος πακετοποίησης, οι ευρετικές μέθοδοι First Fit Decreasing, Best Fit Decreasing χρησιμοποιούν τον ίδιο αριθμό κάδων, 49. Όμως, το αποτέλεσμα της γραφικής αναπαράστασης διαφέρει για την κάθε μία κυρίως στα μικρά αντικείμενα. Στην First Fit Decreasing τα αντικείμενα τοποθετούνται στον πρώτο ελεύθερο κάδο, ενώ στην Best Fit Decreasing σκοπός είναι η πλήρης αξιοποίηση των κάδων, δηλαδή τα αντικείμενα τοποθετούνται σε αυτόν με τη μικρότερη υπολειπόμενη χωρητικότητα.

- Τα καλύτερα αποτελέσματα σε όλες τις περιπτώσεις δίνονται από τις First Fit Decreasing, Best Fit Decreasing μεθόδους. Σε πολλές μάλιστα περιπτώσεις η απόσταση από το κάτω όριο είναι πολύ μικρή, ειδικά για τα αντικείμενα της δεύτερης κατηγορίας. Με αυτά τα αποτελέσματα γίνεται αισθητή η σημασία της ταξινόμησης των αντικειμένων πριν την εκκίνηση της πακετοποίησης. Παρόλα αυτά, οι συγκεκριμένες μέθοδοι δεν μπορούν να χρησιμοποιηθούν για online περίπτωση προβλήματος καθώς πρέπει να έχουν προκαθοριστεί όλα τα αντικείμενα για να γίνει η ταξινόμηση, ένα σημείο που υστερούν έναντι των First Fit, Best Fit μεθόδων που δεν έχουν τέτοιους περιορισμούς.

- Όλες οι μέθοδοι δίνουν αρκετά ικανοποιητικά αποτελέσματα, ειδικά για τα αντικείμενα της κατηγορίας Β. Δηλαδή οι ευρετικές τεχνικές είναι πιο αποδοτικές όταν η πακετοποίηση αφορά αντικείμενα μικρού ή μεγάλου μεγέθους σε σχέση με τη χωρητικότητα των κάδων, όταν εξαιρούνται δηλαδή τα αντικείμενα μεσαίου μεγέθους.
- Όπως είναι λογικό, όσο αυξάνεται το πλήθος των αντικειμένων, τόσο πιο αισθητές γίνονται οι διαφορές στα αποτελέσματα μεταξύ των ευρετικών μεθόδων. Για 20 αντικείμενα τα αποτελέσματα διαφέρουν ελάχιστα σε σχέση με αυτά για 500 αντικείμενα.

3.3 Σύνοψη 3^{ου} κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκαν οι πίνακες των αποτελεσμάτων που προέκυψαν από τις εκτελέσεις του αλγορίθμου για τις κατηγορίες προβλημάτων που εξετάστηκαν. Καταγράφηκαν σε αναλυτική μορφή τα αποτελέσματα για κάθε πλήθος και κατηγορία αντικειμένων από όλες τις ευρετικές τεχνικές. Στη συνέχεια, υπολογίστηκαν οι μέσοι όροι των περιπτώσεων και έγινε απευθείας σύγκριση των μεθόδων και της απόστασης αυτών από τα κάτω όρια των περιπτώσεων.

Όπως προέκυψε, στα καλύτερα αποτελέσματα κατέληξαν οι μέθοδοι First Fit Decreasing, Best Fit Decreasing που ταξινομούν τα αντικείμενα κατα φθίνουσα σειρά μήκους πριν αρχίσουν τη διαδικασία πακετοποίησης. Όμως, οι μέθοδοι First Fit, Best Fit είναι εξίσου διαδεδομένες, αφού έχουν την δυνατότητα να επιλύουν online περιπτώσεις προβλημάτων όπου τα αντικείμενα για πακετοποίηση προστίθενται κατά τη λειτουργία του αλγορίθμου, ενώ οι μέθοδοι που απαιτούν ταξινόμηση στην αρχή δεν έχουν αυτή τη δυνατότητα.

Κεφάλαιο 4

Συμπεράσματα

Σε αυτή τη διπλωματική εργασία έγινε προσπάθεια διατύπωσης του μονοδιάστατου προβλήματος πακετοποίησης σε κάδους και σύνταξης ενός αλγορίθμου βασιζόμενου σε ευρετικές τεχνικές για την αντιμετώπισή του.

Αρχικά παρουσιάστηκαν οι διαστάσεις και οι εφαρμογές του προβλήματος με προσεγγίσεις από την καθημερινή ζωή. Έγινε κατανοητή η σπουδαιότητα ύπαρξης ενός προγράμματος επίλυσης που θα παράγει ποιοτικές λύσεις. Διαπιστώθηκε ότι το πρόβλημα έχει αντιμετωπιστεί από πολλά διαφορετικά είδη αλγορίθμων, καθώς δεν μπορεί να βρεθεί εύκολα αλγόριθμος που θα δίνει βέλτιστη λύση λόγω του βαθμού πολυπλοκότητάς του. Η βιβλιογραφική ανασκόπηση έδειξε πολλές μεθόδους που βασίζονται σε διαφορετικά επιστημονικά πεδία να αναπτύσσονται για την αντιμετώπιση του προβλήματος. Οι μεταευρετικές τεχνικές κυριαρχούν τα τελευταία χρόνια στην αντιμετώπιση του προβλήματος πακετοποίησης 2 ή περισσοτέρων διαστάσεων. Για το μονοδιάστατο πρόβλημα, παρατηρήθηκε ότι οι ευρετικές τεχνικές καταλήγουν σε πολύ ικανοποιητικά αποτελέσματα.

Για την επιλογή μεθόδου επίλυσης ενός προβλήματος πακετοποίησης που αντιστοιχεί σε ένα πρόβλημα της καθημερινότητας, είναι θεμελιώδης σημασίας η ακριβής διατύπωση όλων των συνθηκών και περιορισμών του προβλήματος. Το πρόβλημα πρέπει να μοντελοποιηθεί μαθηματικά και στη συνέχεια προσαρμόζονται σε αυτό οι κεντρικές ιδέες λειτουργίας της κάθε κατηγορίας αλγορίθμων. Οι ευρετικές τεχνικές ξεχωρίζουν καθώς παρέχουν μια ιδιαίτερα αποτελεσματική και εύχρηστη μέθοδο προσέγγισης του προβλήματος, που απαιτεί ελάχιστο υπολογιστικό χρόνο για την εφαρμογή της.

Στη συγκεκριμένη εργασία επιλέχθηκαν οι ευρετικές τεχνικές αντιμετώπισης του προβλήματος πακετοποίησης, διατυπωμένες σε MATLAB. Οι ευρετικές τεχνικές δεν μπορούν να δεσμευτούν για την καλή ποιότητα της επίλυσης που παρουσιάζουν. Έτσι, επιλέχθηκε η MATLAB ως γλώσσα προγραμματισμού, η οποία έχει τη δυνατότητα της γραφικής αναπαράστασης των αποτελεσμάτων για την άμεση διαπίστωση της καλής λειτουργίας του αλγορίθμου. Οι ευρετικές τεχνικές αποτελούνται από τις First Fit, Best Fit, First Fit Decreasing, Best Fit Decreasing μεθόδους, οι οποίες αναλύθηκαν ως προς τον τρόπο λειτουργίας τους.

Τα αποτελέσματα που προέκυψαν από την εφαρμογή της κάθε μεθόδου σε συγκεκριμένες συνθήκες προβλημάτων ανέδειξαν τις μεθόδους που χρησιμοποιούν ταξινόμηση των αντικειμένων ως πιο αποδοτικές. Παρόλα αυτά, όλες οι ευρετικές μέθοδοι κατέληξαν σε αρκετά ικανοποιητικά αποτελέσματα, απέχοντας μικρές αποστάσεις από το κάτω όριο της κάθε περίπτωσης προβλήματος.

Κεφάλαιο 5

Βιβλιογραφία

- Arora Sanjeev, Barak Boaz, *Computational Complexity: A Modern Approach.* Cambidge Publications, 2009
- Aschlock, D, *Evolutionary Computation for Modeling and Optimization.* Springer, 2006
- Bansal Nikhil, Lodi Andrea, Sviridenko Maxim, *A Tale of Two Dimensional Bin Packing.* Foundations of Computer Science, 2005
- Battiti Roberto, Brunato Mauro, Mascia Franco, *Reactive Search and Intelligent Optimization*. Springer Verlag, 2008
- Cormen Thomas H, *Introduction to Algorithms (3rd edition).*MIT Press, 2009
- Djang Phillip A., Finch Paul R., *Solving One Dimensional Bin Packing Problems.* Journal of Heuristics, 1998
- Dósa Gyorgy, Sgall Jiri, *First Fit bin packing: A tight analysis.* Leibniz International Proceedings in Informatics Schloss Dagstuhl, 1998
- Dyckhoff Harald, *A typology of cutting and packing problems.* European Journal of Operational Research, 1990
- Falkenauer E, Delchambre A, *A Genetic Algorithm for Bin Packing and Line Balancing.* Research Centre for Belgian Metalworking Industry, 1992
- Falkenauer Emanuel, *A Hybrid Grouping Genetic Algorithm for Bin Packing.* Research Centre for Belgian Metalworking Industry, 1996
- Fleszar K, Hindi K.S., *New heuristics for one-dimensional bin packing.* Computers and Operations Research, 2002
- Garey Michael R., Johnson David S., *Computers and Intractability: a Guide to the Theory of NP-Completeness.* Freeman, 1979
- Gent I.P., *Heuristic Solutions of Open Bin Packing Problems*. Journal of Heuristics, 1998
- Gomez-Vilouta Giglia, Hamiez Jean-Philippe, Hao Jin-Kao, *Tabu Search with Consisten Neighbourhood for Strip Packing.* Springer Verlag, 2010
- Gupta J.N.D., Ho J.C., *A new heuristic algorithm for the one-dimensional bin-packing problem*. Production Planning and Control, 1999
- Hamiez Jean-Philippe, Robet Juliet, Hao Jin-Kao, *A Tabu Search Algorithm with Direct Representation for Strip Packing, Springer Verlag, 2009*

- Holland J, *Adaptation in natural and artificial systems.* University of Michigan press, 1975, 1992
- Johnson D.S., *Near-optimal bin-packing algorithms*. PhD thesis, MIT Department of Mathematics, Cambridge, 1973
- Johnson D.S., Demers A., Ullman J.D., Garey M.R., Graham R.L., *Worst-case Performance Bounds for Simple One-Dimensional Packing Algorithms.* SIAM J. Comput., 1974
- Jϋnger Michael, Reinelt Gerhard, Rinaldi Giovanni, *Combinatorial Optimization – Eureka, You Shrink!*. Springer, 2001
- Kämpke Thomas, *Stimulated Annealing: Use of a new tool in bin packing.* Annals of Operations Research, 1988
- Karmarkar Narendra, *An efficient approximation scheme for the one-dimensional bin-packing problem.* Foundations of Computer Science, 1982
- Kumar Satyendra, Rao V. Venkata, Tirupati Devanath, *A heuristic procedure for one dimensional bin packing problem with additional constraints.*Indian Institute of Management, 2003
- Levine John, Ducatelle Frederick, *Ant colony optimization and local search for bin packing and cutting stock problems.* Journal of the operational research society, 2004
- Lodi Andrea, *Algorithms for Two-Dimensional Bin Packing and Assignment Problems.* Universita Degli Studi Di Bologna, 1999
- Mancapa Vusisizwe, *A General Genetic Algorithm for One and Two Dimensional Cutting and Packing Problems.* Nelson Mandela Metropolitan University, 2008
- Martello Silvano, Pisinger David, Toth Paolo, *New trends in exact algorithms for the 0-1 knapsack problem.* European Journal of Operational Research, 2000
- Martello Silvano, Toth Paolo, *Knapsack Problems. Algorithms and Computer Implementations.* John Wiley and Sons, 1990
- Mohamadi N, *Application of Genetic Algorithm for the bin packing problem with a new representation scheme.* Department of Mathematics Islamic Azad University, 2010
- Ortmann F.G., *Heuristics for Offline Rectangular Packing Problems.* Ph.D. Thesis, 2010
- Papadimitriou Christos, *Computational Complexity.* Addison Wesley, 1994

- Puschinger Jakob, Günther R. Raidl, *An Evolutionary Algorithm for Column Generation in Integer Programming: an Effective Approach for 2D Bin Packing*. Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2004
- Reeves C., *Hybrid genetic algorithms for bin-packing and related problems.* Annals of Operations Research, 1996
- Register Andy H., A Guide to MATLAB: Object-Oriented Programming, Chapman & Hall/CRC, 2007
- Ross Peter, Schulenburg Sonia, Marin-Blazquez Javier G., Hart Emma, *Hyper-heuristics: learning to combine simple heuristics in bin-packing problems.* Morgan Kaufmann Publishers Inc, 2002
- Russel Stuart, Norvig Peter, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010
- Schrijver Alexander, *Theory of linear and integer programming.* John Wiley and Sons, 1998
- Shaw Paul, *A Constraint for Bin Packing.* Springer Verlag, 2004
- Stawowy Adam, *Evolutionary algorithm for bin packing problem.* Faculty of Management, Universitu of Mining and Metallurgy, 2006
- Trivella Alessio, Pisinger David, *Heuristic and Exact Approaches for the 1D and 2D Bin-packing Problem*. Complementary Topics in Integer Programming-Departments of Mathematics-Management, Technical University of Denmark, 2012
- Valerio di Carvalho J.M., *Exact solution of bin packing problems using column generation and branch-and-bound.* Kluwer Academic Publishers, 1999
- Vanderbeck F., *Computational study of a column generation algorithm for bin packing and cutting stock problems.*Computational Optimization and Applications, 1994
- Wäscher Gerhard, Haußner Heike, Schumann Holger, *An improved typology of cutting and packing problems.* European Journal of Operational Research, 2007
- White Robert E., *Computational Mathematics: Models, Methods and Analysis with MATLAB and MPI.* Chapman & Hall/CRC, 2004

Κεφάλαιο 6

Παράρτημα

Σε αυτό το κεφάλαιο παρατίθεται ο κώδικας που αναπτύχθηκε για την εκπόνηση της διπλωματικής αυτής εργασίας χρησιμοποιώντας τη γλώσσα προγραμματισμού MATLAB. Το βασικό script περιέχει και τις τέσσερις ευρετικές μεθόδους που χρησιμοποιήθηκαν, ενώ τα δύο functions αναφέρονται στην παραγωγή αντικειμένων για την κάθε κατηγορία που εξετάστηκε και στην εύρεση του κάτω ορίου (Lower Bound).

```
% Heuristics for 1D bin packing
% data structure:
% u(1:n): size of n objects,
% C: capacity of each bin
% w(n, n): assignment matrix. w(i,j)= 1 if u(i) is assigned to b(j)
% r(1:n): residue capacity, initially, r = C*ones(1,n);
clear all;clc;
[data]=DataGen(100,0);
u = data;
n = length(u);
C = 100;
% First-Fit
r = C*ones(1,n);
w = zeros(n);
color=zeros(1,n);
for i=1:n,
   color(1,i)=rand();
   %disp('r = '); disp(r);
   %disp(['u(' int2str(i) ') = ' int2str(u(i))]);
   %pause
   cr1=find(u(i)*ones(1,n)<=r, 1 );%first fit criterion
   %disp(['u(' int2str(i) ') is assigned to b(' int2str(idx) ');']);
   subplot(2,2,1)
   w(i,cr1)=1;
   if r(cr1)==100
       r(cr1)=r(cr1)-u(i);
       fill([cr1-0.5  cr1-0.5  cr1+0.5  cr1+0.5],[0  100-r(cr1)  100-r(cr1)
0],color(1,i)),xlabel('bins'),ylabel('occupancy'),title('First Fit')
   else
       fill([cr1-0.5  cr1-0.5  cr1+0.5  cr1+0.5],[100-r(cr1)  100-r(cr1)+u(i)
100-r(cr1)+u(i)                                                        100-
r(cr1)],color(1,i)),xlabel('bins'),ylabel('occupancy'),title('First Fit')
       r(cr1)=r(cr1)-u(i);
   end
   axis([0 100 0 100]);
   axis on
   hold on
   %disp('Press any key to continue'); pause
end
hold off
oc=u*w;
nbin=sum(oc > 0);
disp('First Fit method')
disp('The Lower Bound is')
DataAnalysis(n,u,nbin)
disp(['Total # of bins used = ' int2str(nbin)]);
% Best-Fit
```

```
r = C*ones(1,n);
w = zeros(n);
for i=1:n,
   %disp('r = '); disp(r);
   %disp(['u(' int2str(i) ') = ' int2str(u(i))]);
   %pause
   cr1=find(u(i)*ones(1,n) <= r); %best fit
   [tmp,cr2]=min(r(cr1)-u(i));   %criterion
   %disp(['u(' int2str(i) ') is assigned to b(' int2str(idx1(idx2)) ');']);
   subplot(2,2,2)
   w(i,cr1(cr2))=1;
   if r(cr1(cr2))==100
       r(cr1(cr2))=tmp;
       fill([cr1(cr2)-0.5  cr1(cr2)-0.5  cr1(cr2)+0.5  cr1(cr2)+0.5],[0  100-
r(cr1(cr2))                                                      100-r(cr1(cr2))
0],color(1,i)),xlabel('bins'),ylabel('occupancy'),title('Best Fit')
   else
       fill([cr1(cr2)-0.5   cr1(cr2)-0.5   cr1(cr2)+0.5   cr1(cr2)+0.5],[100-
r(cr1(cr2))       100-r(cr1(cr2))+u(i)       100-r(cr1(cr2))+u(i)        100-
r(cr1(cr2))],color(1,i)),xlabel('bins'),ylabel('occupancy'),title('Best Fit')
       r(cr1(cr2))=tmp;
   end
   axis([0 100 0 100]);
   axis on
   hold on
   %disp('Press any key to continue'); pause
end
hold off
oc=u*w;
nbin=sum(oc > 0);
disp('---------------------------')
disp('Best fit Method')
disp('The Lower Bound is')
DataAnalysis(n,u,nbin)
disp(['Total # of bins used = ' int2str(nbin)]);
% First-Fit Decreasing
r = C*ones(1,n);
w = zeros(n);
% Sorting items in decreasing height for ffd,bfd
[ud1,idec]=sort(-u);
ud=-ud1;
[tmp,irec]=sort(idec);
% ud=u(idec); u = ud(irec);
for i=1:n,
   %disp('r = '); disp(r);
   %disp(['ud(' int2str(i) ') = ' int2str(ud(i))]);
   cr1=find(ud(i)*ones(1,n)<=r, 1 );
   %disp(['u(' int2str(i) ') is assigned to b(' int2str(idx) ');']);
   subplot(2,2,3)
   w(i,cr1)=1;
   if r(cr1)==100
       r(cr1)=r(cr1)-ud(i);
       fill([cr1-0.5  cr1-0.5  cr1+0.5  cr1+0.5],[0  100-r(cr1)  100-r(cr1)
0],color(1,idec(i))),xlabel('bins'),ylabel('occupancy'),title('First      Fit
Decreasing')
   else
```

```
        fill([cr1-0.5  cr1-0.5  cr1+0.5  cr1+0.5],[100-r(cr1)  100-r(cr1)+ud(i)
100-r(cr1)+ud(i)                                                          100-
r(cr1)],color(1,idec(i))),xlabel('bins'),ylabel('occupancy'),title('First  Fit
Decreasing')
        r(cr1)=r(cr1)-ud(i);
   end
   axis([0 100 0 100]);
   axis on
   hold on
   %disp('Press any key to continue'); pause
end
hold off
w=w(irec,:);
oc=u*w;
nbin=sum(oc > 0);
disp('----------------------------')
disp('First Fit Decreasing method')
disp('The Lower Bound is')
DataAnalysis(n,u,nbin)
disp(['Total # of bins used = ' int2str(nbin)]);
% Best-Fit Decreasing
r = C*ones(1,n);
w = zeros(n);
for i=1:n,
   %disp('r = '); disp(r);
   %disp(['u(' int2str(i) ') = ' int2str(ud(i))]);
   cr1=find(ud(i)*ones(1,n) <= r);
   [tmp,cr2]=min(r(cr1)-ud(i));
   %disp(['u(' int2str(i) ') is assigned to b(' int2str(idx1(idx2)) ');']);
   subplot(2,2,4)
   w(i,cr1(cr2))=1;
   if r(cr1(cr2))==100
       r(cr1(cr2))=tmp;
       fill([cr1(cr2)-0.5  cr1(cr2)-0.5  cr1(cr2)+0.5  cr1(cr2)+0.5],[0  100-
r(cr1(cr2))                                                    100-r(cr1(cr2))
0],color(1,idec(i))),xlabel('bins'),ylabel('occupancy'),title('Best        Fit
Decreasing')
   else
       fill([cr1(cr2)-0.5   cr1(cr2)-0.5   cr1(cr2)+0.5   cr1(cr2)+0.5],[100-
r(cr1(cr2))       100-r(cr1(cr2))+ud(i)       100-r(cr1(cr2))+ud(i)       100-
r(cr1(cr2))],color(1,idec(i))),xlabel('bins'),ylabel('occupancy'),title('Best
Fit Decreasing')
       r(cr1(cr2))=tmp;
   end
   axis([0 100 0 100]);
   axis on
   hold on
   %disp('Press any key to continue'); pause
end
hold off
oc=u*w;
nbin=sum(oc > 0);
disp('--------------------------')
disp('Best Fit Decreasing method')
disp('The Lower Bound is')
DataAnalysis(n,u,nbin)
disp(['Total # of bins used = ' int2str(nbin)]);
```

```
function [data] = DataGen(n,type)
%Type can be either set to 0(A) or 1(B)
%Type A items length:[20,80]
%Type B items length;[20,40]U[60,80]
if type==0
    for i=1:n
    a(i)=ceil(rand*100);
    while a(i)<20 | a(i)>80
        a(i)=ceil(rand*100);
    end
    end
else
    for i=1:n
        a(i)=ceil(rand*100);
        while ((a(i)<20)|(a(i)>40))&&((a(i)<60)|(a(i)>80))
             a(i)=ceil(rand*100);
        end
    end
end
data=[a];
return

function [LB,per] = DataAnalysis(n,u,nbin)
%LB: Lower Bound of the specified n
s1=0;
for i=1:n
    s1=s1+u(i);
end
LB=(s1/100);
return
```