Χρησιμοποίησα το bscp\_generate.exe , ώστε να γίνει η τυχαία παραγωγή προβλημάτων . Δημιούργησα 16 ειδών διαφορετικά προβλήματα με 10 προβλήματα το καθένα , άρα 160 συνολικός αριθμός προβλημάτων προς επίλυση . Οι μετρικές των προβλημάτων είναι N = 8 , K = 4 και Μ ανάλογα την 10αδα (10-140 με μια εξαίρεση στις μεταβάσεις από 130->133 και 133-137) .

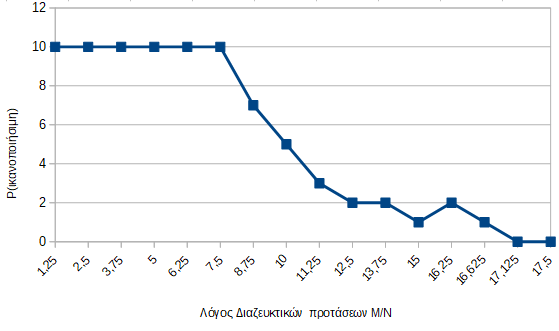
Μετά την παραγωγή των προβλημάτων , δημιούργησα ένα αρχείο τύπου batch όπου έγινε μαζική εκτέλεση της κάθε δεκάδας προβλημάτων . Στην αρχή , έγινε εκτέλεση με τον αλγόριθμο Depth ώστε να διαπιστωθούν ποια προβλήματα είναι επιλύσιμα και ποια όχι (φαίνεται και στον πίνακα παρακάτω).

|  |  |  |
| --- | --- | --- |
| **Depth** | | |
| **M** | **Problems Solved** | **Problems Unsolved** |
| 10 | 10 | 0 |
| 20 | 10 | 0 |
| 30 | 10 | 0 |
| 40 | 10 | 0 |
| 50 | 10 | 0 |
| 60 | 10 | 0 |
| 70 | 7 | 3 |
| 80 | 5 | 5 |
| 90 | 3 | 7 |
| 100 | 2 | 8 |
| 110 | 2 | 8 |
| 120 | 1 | 9 |
| 130 | 2 | 8 |
| 133 | 1 | 9 |
| 137 | 0 | 10 |
| 140 | 0 | 10 |

Καθώς διαπίστωσα ποιες δεκάδες προβλημάτων είχαν μη επιλύσιμα προβλήματα , έκανα χρήση ξανά του batch file(run.bat) αλλά αυτή τη φορά για τον αλγόριθμο Hill όπου έτρεξα κάθε δεκάδα προβλημάτων 5 φόρες με τη διαφορά ότι ήξερα τα προβλήματα στα οποία θα χρειαστεί να εξαντλήσει το όριο του χρόνου εκτέλεσης δίχως να τα επιλύσει .

Πιο κάτω παραθέτω τον πίνακα και το διάγραμμα με την πιθανότητα επιλύσιμων προβλημάτων για κάθε δυνατή τιμή M ως προς το λόγο M/N :

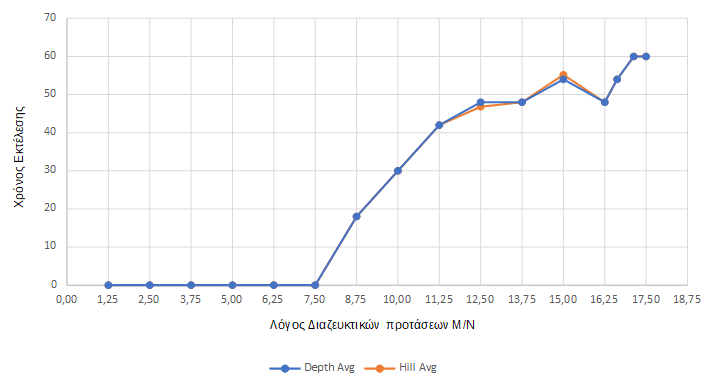
|  |  |
| --- | --- |
| **M** | **M/N** |
| 10 | 1,25 |
| 20 | 2,5 |
| 30 | 3,75 |
| 40 | 5 |
| 50 | 6,25 |
| 60 | 7,5 |
| 70 | 8,75 |
| 80 | 10 |
| 90 | 11,25 |
| 100 | 12,5 |
| 110 | 13,75 |
| 120 | 15 |
| 130 | 16,25 |
| 133 | 16,625 |
| 137 | 17,125 |
| 140 | 17,5 |



Με βάση το παραπάνω διάγραμμα , προκύπτει ότι μέχρι Μ = 60 όλα τα προβλήματα είναι επιλύσιμα . Από Μ = 70 και μετά ξεκινάνε τα μη επιλύσιμα προβλήματα και με κάθε αύξηση του Μ αυξάνεται και ο αριθμός των μη επιλύσιμων προβλημάτων .

Για το πιο σημαντικό κομμάτι της εργασίας δηλαδή τους μέσους χρόνους εκτέλεσης κάθε αλγορίθμου , έτρεξα το batch file(run.bat) που ανέφερα και στην αρχή και παρακάτω είναι ο πίνακας αποτελεσμάτων για κάθε αλγόριθμο(5 φορές για τον αλγόριθμο Hill , 1 φορά για τον αλγόριθμο Depth , η αποθήκευση των αποτελεσμάτων έγινε στα αντίστοιχα Depth\_Results.txt,Hill\_Results.txt). Παρατηρούμε ότι όσο αυξάνεται ο λόγος M/N αυξάνεται και ο μέσος χρόνος επίλυσης των προβλημάτων καθώς έχουμε προβλήματα τα οποία είναι αδύνατο να επιλυθούν εντός του ορίου των 60 δευτερολέπτων και αυτό αυξάνει αντίστοιχα και το μέσο χρόνο επίλυσης .

|  |  |  |
| --- | --- | --- |
| **M/N** | **Depth Average Time** | **Hill Average Time** |
| 1,25 | 0 | 0 |
| 2,5 | 0,0001 | 0 |
| 3,75 | 0 | 0 |
| 5 | 0 | 0 |
| 6,25 | 0,0002 | 0,006 |
| 7,5 | 0,001 | 0,0012 |
| 8,75 | 18,0001 | 18 |
| 10 | 30 | 30 |
| 11,25 | 42 | 42 |
| 12,5 | 48 | 46,802 |
| 13,75 | 48 | 48 |
| 15 | 54 | 55,203 |
| 16,25 | 48 | 48 |
| 16,625 | 54 | 54 |
| 17,125 | 60 | 60 |
| 17,5 | 60 | 60 |



Αυτό που προκύπτει από τα διαγράμματα και τους σχετικούς πίνακες τους οποίους είδαμε μέχρι στιγμής είναι ότι μέχρι Μ = 133 , υπάρχει τουλάχιστον ένα πρόβλημα το οποίο είναι επιλύσιμο . Μετά από αυτό , για Μ =137 δεν υπάρχουν πλέον επιλύσιμα προβλήματα και όσο αυξάνεται το Μ χειροτερεύει και η κατάσταση πράγμα το οποίο φαίνεται και για Μ = 140 . Με τη χρήση των δικών μου υπολογιστικών πόρων , φαίνεται πως η διάρκεια εκτέλεσης στα επιλύσιμα προβλήματα δεν έχει διαφορά ανεξαρτήτως του Μ που θέτουμε . Για τα προβλήματα γενικότερα , δηλαδή επιλύσιμα και μη , ο χρόνος φαίνεται ότι αυξάνεται εκθετικά για Μ = 70 και πάνω , καθώς εκεί ξεκινάνε και τα μη επιλύσιμα προβλήματα(διότι έχουμε και το όριο των 60 δευτερολέπτων) .

Συμπερασματικά , με τα αποτελέσματα που προκύπτουν από τα πειράματα τα οποία διεξήχθησαν , καταλήγουμε ότι ο αλγόριθμος Depth είναι πιο γρήγορος από τον αλγόριθμο Hill . Ωστόσο και χωρίς τα πειράματα γνωρίζουμε ότι ο αλγόριθμος Depth έχει περισσότερες πιθανότητες να δώσει γρηγορότερα λύση , καθώς επεκτείνει κόμβους οι οποίοι φαίνεται πως βρίσκονται πιο κοντά στη λύση .