ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

Μελέτη της Permanent, #P-Complete Προβλημάτων και Προσεγγιστικοί Αλγόριθμοι Αυτών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΤΡΙΟΜΜΑΤΗ ΘΕΟΦΙΛΟΥ

Επιβλέπων : Αριστείδης Παγουρτζής
Αν. Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ

Αθήνα, Φεβρουάριος 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ

Μελέτη της Permanent, #P-Complete Προβλημάτων και Προσεγγιστικοί Αλγόριθμοι Αυτών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΤΡΙΟΜΜΑΤΗ ΘΕΟΦΙΛΟΥ

Επιβλέπων : Αριστείδης Παγουρτζής
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 21η Φεβρουαρίου 2019.

.....	.....	.....
Αριστείδης Παγουρτζής	Αντώνιος Συμβώνης	Ευστάθιος Ζάχος
Αν. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2019

.....

ΤΡΙΟΜΜΑΤΗΣ ΘΕΟΦΙΛΟΣ

Διπλωματούχος Σχολής Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών Ε.Μ.Π.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΛΟΓΙΚΗΣ ΚΑΙ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΜΩΝ

Περίληψη

Η Permanent είναι μία συνάρτηση πινάκων που εμφανίζεται σε διάφορα προβλήματα συνδυαστικής καθώς και σε προβλήματα στατιστικής φυσικής. Μοιάζει πολύ με την ορίζουσα πίνακα, αλλά η πρώτη συνάρτηση υπολογίζεται πολύ πιο δύσκολα σε σχέση με την δεύτερη. Στην παρούσα εργασία θα παρουσιάσουμε τις αλγεβρικές ιδιότητες που έχει η Permanent και τι έχουν αποδείξει οι μαθηματικοί που ενασχολήθηκαν μαζί της. Πόσο «δύσκολο» είναι το πρόβλημα υπολογισμού της Permanent, σύμφωνα με την κλασική πολυπλοκότητα, καθώς και με την αλγεβρική πολυπλοκότητα που όρισε ο L. G. Valiant, μελετώντας την συνάρτηση αυτή. Θα αναφερθούμε σε προσεγγιστικούς αλγόριθμους που υπάρχουν για την Permanent καθώς και για άλλα προβλήματα παρόμοιας πολυπλοκότητας. Μελετώντας κάποια από αυτά τα προβλήματα, στα πλαίσια αυτής της εργασίας κάναμε κάποιες προσπάθειες για την κατασκευή ενός ντετερμινιστικού προσεγγιστικού αλγόριθμου για την Permanent. Τέλος ορίσμαε τυπικά κάποια μετρητικά προβλήματα και με όσα μελετήσαμε στα προηγούμενα κεφάλαια, δημιουργήσαμε FPTAS αλγόριθμους που σχεδιάστηκαν για αυτά

Λέξεις Κλειδιά: Permanent, #P, Μετρητικά Προβλήματα, Πλήθος Perfect Matching, Πλήθος λύσεων Knapsack, προσεγγιστικά σχήματα για μετρητικά προβλήματα

Abstract

Permanent is a function very similar to determinant but despite their similarity, the former satisfies much fewer properties than the latter. In particular, there are efficient algorithms to compute determinant, however, there does not appear any way of computing permanent efficiently. Moreover permanent appears repeatedly in combinatorial problems as well as in statistical physics problems. In this thesis we shall present the algebraic properties that permanent satisfies and some basic theorems that mathematicians proved. Also we will see "how difficult" is to compute the permanent of a matrix, via classic complexity theory as well as algebraic complexity theory which was introduced by L. G. Valliant. Afterwards we will refer to some approximation algorithms for the permanent and we will study some other problems that seem to be equally difficult. By studying these problems we attempt to create an approximate deterministic algorithm for the permanent. And finally we will define formally some counting problems and we will construct a FPTAS to approximate their solutions.

Keywords: Permanent, #P, Counting Problems, Number of perfect matchings, Counting Knapsack Solutions, Approximate Schemes for counting problems

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας, καθ. Α. Παγουρτζή, για την πολύτιμη καθοδήγηση καθώς και για την εξαιρετική συνεργασία που είχαμε.

Επιπρόσθετα, ευχαριστώ ιδιαιτέρως τους καθ. Α. Συμβώνη και Ε. Ζάχο για τις ανεκτίμητες γνώσεις που μου έχουν μεταδώσει καθώς και για την αστείρευτη πυγή έμπνευσης που αποτελούν. Ευχαριστώ επίσης τους συμφοιτητές μου Φ. Φιοραβάντε, Ν. Μελισσινό, Α. Αντωνόπουλο και Α. Χαλκή για την συνεισφορά τους ώστε να ολοκληρωθεί με επιτυχία η παρούσα εργασία.

Αφήνοντας για το τέλος μεν αλλά εξίσου σημαντικούς δε, ευχαριστώ όλους τους φίλους που έχουν βρεθεί στο πλάι μου κατά τη διάρκεια της φοιτητικής μου ζωής καθώς και τους γονείς και την αδερφή μου, για την στήριξη που μου παρέχουν τόσα χρόνια.

Περιεχόμενα

1 ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο της Διπλωματικής

Στην παρούσα εργασία θα ασχοληθούμε με μία συνάρτηση η οποία έχει αποδειχθεί ότι δεν υπολογίζεται «εύκολα» (εκτός αν $\#P = P$), και αν δούμε τον ορισμό της μοιάζει πάρα πολύ με την ορίζουσα. Η συνάρτηση στην οποία αναφερόμαστε είναι η permanent, η οποία κεντρίζει το ενδιαφέρον, διότι παρόλο που ο ορισμός της είναι παρόμοιος με αυτόν της ορίζουσας, οι δύο συναρτήσεις έχουν πολύ διαφορετικές ιδιότητες. Ακόμη η ορίζουσα υπολογίζεται σε πολυωνυμικό χρόνο ενώ η permanent έχει αποδειχθεί ότι είναι #P-complete για πίνακες με στοιχεία από το $\{0,1\}$ με αναγωγές κατά Cook, ενώ ανήκει στην $TotP$ με αναγωγές κατά Karp.

Ο Valiant ήταν αυτός απέδειξε ότι η permanent για πίνακες με στοιχεία από το $\{0,1\}$, είναι #P-complete, ενώ αν τα στοιχεία των πινάκων προέρχονται από ένα σώμα K με χαρακτηριστική $char(K) \neq 2$, τότε η permanent είναι VNP-complete. Οι κλάσεις VP και VNP είναι κλάσεις πολυπλοκότητας που όρισε ο Valiant, μελετώντας την Permanent και είναι αντίστοιχες των κλάσεων P και NP. Όπως αναφέραμε ήδη η ορίζουσα υπολογίζεται σε πολυωνιμικό χρόνο και πιστεύουμε ότι είναι χαμηλά στην VP ενώ η permanent είναι VNP-complete. Ένα ενδιαφέρον αποτέλεσμα που προκύπτει από την πολυπλοκότητα του Valiant είναι ότι αν ισχύει η εικασία του Riemann και $VP \neq VNP$ τότε και $P \neq NP$. Εν κατακλείδι έχει αξία να μελετήσει κανένας αυτήν την συνάρτηση και τις ιδιότητες της καθώς φαίνεται να έχει πολλές «ιδιοτροπίες» οι οποίες μπορούν να δώσουν καλύτερη κατανόηση για την πολυπλοκότητα του υπολογισμού δύσκολων συναρτήσεων.

1.2 Επισκόπηση της Διπλωματικής Εργασίας

Αρχικά στο κεφάλαιο 2 θα μελετήσουμε τον ορισμό της ορίζουσας, τις ιδιότητες της και θα τα συγκρίνουμε με τον ορισμό και τις ιδιότητες της permanent με σκοπό να γίνει αντιληπτό γιατί δεν υπάρχει κάποια αλγεβρική ιδιότητα που να μας επιτρέπει να υπολογίζουμε εύκολα την permanent. Η permanent ορίζεται με κλειστό τύπο και ένα φυσικό ερώτημα που προκύπτει είναι το εξής

«ποιο είναι το υπολογιστικό κόστος του υπολογισμού της permanent;»

Αυτό το ερώτημα προσπαθούμε να εξετάσουμε στο κεφάλαιο 3 παρουσιάζοντας διάφορα αποτελέσματα που προέρχονται από την θεωρία υπολογισιμότητας και πολυπλοκότητας.

Στη συνέχεια αφού έχουμε ξεκαθαρίσει ότι το να υπολογίσουμε επ' ακριβώς την permanent είναι υπολογιστικά ένα δύσκολο πρόβλημα, αναρωτιόμαστε αν μπορούμε να θυσιάσουμε ακρίβεια στη λύση του προβλήματος ώστε να το λύσουμε γρήγορα. Με άλλα λόγια θέλουμε να προσεγγίσουμε την τιμή της permanent με κάποιο σφάλμα αφενός, αλλά σε πολυωνυμικό χρόνο αφετέρου. Έτσι λοιπόν στο κεφάλαιο 4 θα αναφέρουμε πιθανοτικούς προσεγγιστικούς αλγόριθμους που το επιτυγχάνουν αυτό. Επίσης κάνουμε μία προσπάθεια για την δημιουργία ενός ντετερμινιστικού προσεγγιστικού αλγόριθμου για την permanent.

Παρατηρήσαμε ότι το πρόβλημα του να βρεθεί ένα Perfect Matching είναι εύκολο, ενώ η permanent που εκφράζει τον υπολογισμό του πλήθους των perfect matching σε ένα διμερές

γράφημα είναι δύσκολο. Κοιτάξαμε και άλλα προβλήματα με δύσκολο πρόβλημα εύρεσης πλήθους λύσεων και εύκολο πρόβλημα εύρεσης μίας μόνο λύσης. Ένα από αυτά είναι το Knapsack. Στα επόμενα κεφάλαια μελετάμε έναν πιθανοτικό προσεγγιστικό αλγόριθμο του προβλήματος #Knapsack για το οποίο έχει αποδειχθεί ότι είναι ίδιας δυσκολίας με το να υπολογισθεί η τιμή της permanent. Στη συνέχεια μελετάμε έναν ντετερμινιστικό προσεγγιστικό αλγόριθμο για το #Knapsack με σκοπό να κατανοήσουμε τι ήταν ικανό να αντικαταστήσει την τυχαιότητα.

Στο τελευταίο κεφάλαιο, έχοντας μελετήσει αυτούς τους αλγορίθμους παρουσιάζουμε προσεγγιστικούς αλγόριθμους για κάποια άλλα προβλήματα που έχουν δύσκολο μετρητικό πρόβλημα και εύκολο πρόβλημα εύρεσης μιας λύσης.

1.3 Το χρονικό της Permanent

Στο βιβλίο Permanents οι Marcus και Minc αναφέρουν ότ,ι η permanent εμφανίστηκε πρώτη φορά σε εργασίες του Cauchy το 1812, ο οποίος είναι και ο υπαίτιος για το όνομα αυτής της συνάρτησης. Διάφοροι άλλοι μαθηματικοί ασχολήθηκαν με την permanent, μεταξύ αυτών ήταν ο Binet, ο Cayley και ο Schur,

Εκείνη την περίοδο μελετήθηκε εκτεταμένα η ορίζουσα, που είναι στενά συνδεδεμένη με την έννοια της γραμμικής ανεξαρτησίας, έχει ιδιότητες που μας επιτρέπουν να την υπολογίζουμε εύκολα και γεωμετρικά εκφράζει ένα εμβαδόν στον $\mathbb{R}^2$ ή έναν υπερόγκο σε έναν χώρο διάστασης n. Αλλά σε αντίθεση με την ορίζουσα, η permanent, υπολογιζόταν δύσκολα και δεν φάνηκε να έχει κάποια γεωμετρική ιδιότητα ή γενικότερα κάποια χρήσιμη ιδιότητα. Έτσι οι μαθηματικοί εκείνης της περιόδου δεν ασχολήθηκαν περαιτέρω με την permanent.

Αργότερα παρατηρήθηκε ότι η permanent εμφανίζεται συχνά σε διάφορα προβλήματα συνδυαστική και μάλιστα ο H. J. Ryser το 1963 στο βιβλίο The Carus Monograph Combinatorial Mathematics αναφέρει για την permanent "appears repeatedly in the literature of combinatorics in connection with certain enumeration and external problems".

Από διάφορους μαθηματικούς έγινε μία προσπάθεια να βρεθεί κάποια σχέση της ορίζουσας και της Permanent μιας και οι ορισμοί τους δεν διαφέρουν και πολύ. Αλλά ένα βασικό αποτέλεσμα το οποίο προέρχεται από μία εργασία των M. Marcus και H. Minc και θα διατυπώσουμε πιο αυστηρά στο κεφάλαιο 2 είναι ότι για έναν πίνακα A δεν υπάρχει μετασχηματισμός T ώστε $per\big(T(A)\big) = \det A$.

Αποδείχθηκε από τον Valiant ότι η permanent σχετίζεται με μετρητικά προβλήματα και ότι το κόστος ώστε να υπολογισθεί αλγοριθμικά είναι αρκετά μεγάλο. Έτσι έγινε εμφανές ότι η permanent δεν μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο. Έτσι αναπτύχθηκαν διάφοροι προσεγγιστικοί πιθανοτικοί αλγόριθμοι για την Permanent οι οποίοι χρειάζονται πολυωνυμικό χρόνο για την εκτέλεση τους.

1.4 Βασικές Έννοιες

Σε αυτήν την παράγραφο παραθέτουμε κάποιους βασικούς ορισμούς από την Θεωρία Πολυπλοκότητας που θα χρησιμοποιήσουμε κατά κόρων στην εργασία αυτή.

Ορισμός 1.1: Μια **μηχανή Turing** (**TM**) είναι ένας απλός ιδεατός υπολογιστής, δηλαδή ένα υπολογιστικό μοντέλο. Ας θεωρήσουμε μια πεπερασμένη συσκευή με ταινία που προεκτείνεται δυνητικά μέχρι το άπειρο και προς τις δύο κατευθύνσεις και υποδιαιρείται σε κύτταρα που το καθένα περιέχει 1 ή 0, δηλαδή το αλφάβητο της μηχανής είναι το $\Sigma = \{0,1\}$. Σε κάθε χρονική στιγμή η κεφαλή της TM βρίσκεται σε ένα κύτταρο, το οποίο ονομάζεται το τρέχον. Οι βασικές λειτουργίες μιας TM είναι

- Διάβασε το περιεχόμενο του τρέχοντος κυττάρου
- Γράψε 1 ή 0 στο τρέχον κύτταρο
- Κάνε τρέχον κύτταρο το αμέσως αριστερότερο ή το αμέσως δεξιότερο κύτταρο.

Η TM έχει ένα πεπερασμένο αριθμό εσωτερικών καταστάσεων (internal states)

$$Q = \{q_1, \dots, q_k\}$$

Ορισμός 1.2: Ένα **πρόγραμμα μιας μηχανής Turing** είναι ένα σύνολο από τετράδες της μορφής $(q_i, e, d, q_j) \subseteq (Q \times \Sigma \times (\Sigma \cup \{R, L\}) \times Q)$. Όπου το

q_i : εκφράζει την τρέχουσα κατάσταση της μηχανής Turing
e : είναι το στοιχείο του τρέχοντος κυττάρου
d : είναι η λειτουργία που θα εκτελέσει η μηχανή Turing, δηλαδή γράψε 1 ή 0 στο τρέχον κύτταρο ή κάνε τρέχον κύτταρο το αμέσως αριστερότερο (L) ή το αμέσως δεξιότερο κύτταρο (R)
q_j : εκφράζει την νέα κατάσταση της μηχανής Turing αφού εκτελέσει την άνωθεν λειτουργία

Αν παραθέσουμε τον εξής συναρτησιακό (ντετερμινιστικό) περιορισμό για κάθε $< q_i, e >$ υπάρχει το πολύ ένα $< d, q_j >$ έτσι ώστε η τετράδα (q_i, e, d, q_j) να ανήκει στο πρόγραμμα, δηλαδή πρόκειται για μια **συνάρτηση μετάβασης** $\delta : Q \times \Sigma \to A \times Q$, όπου $A = \Sigma \cup \{R, L\}$.

Η συνάρτηση μετάβασης καθορίζει με βάση την παρούσα κατάσταση και το περιεχόμενο του τρέχοντος κυττάρου ποια από τις βασικές λειτουργίες θα εκτελεστεί και ποια θα είναι η επόμενη κατάσταση.Κατά σύμβαση αν η συνάρτηση μετάβασης δεν έχει οριστεί για την κατάσταση $< q, e >$ και η μηχανή φτάσει σε αυτήν, τότε η μηχανή σταματά.

Ορισμός 1.3: Μια **ντετερμινιστική μηχανή Turing** (**DTM**) δέχεται ως συνάρτηση μετάβασης μία $\delta : Q \times \Sigma \to Q \times A$. Ενώ μία **μη ντετερμινιστική μηχανή Turing** (**NTM**) έχει συνάρτηση μετάβασης της μορφής $\delta : Q \times \Sigma \to \mathcal{P}(Q \times A)$, όπου $\mathcal{P}(Q \times A)$ είναι το δυναμοσύνολο του $Q \times A$.

Όποτε αναφερόμαστε σε μηχανές Turing χωρίς ιδιαίτερη διευκρίνιση, συνήθως εννοούμε τις μη ντετερμινιστικές.

Θα χρησιμοποιήσουμε το υπολογιστικό μοντέλο της μηχανής Turing ώστε να ορίσουμε τις παρακάτω κλάσεις πολυπλοκότητας. Αν ένα πρόβλημα είναι μηχανιστικά επιλύσιμο μας ενδιαφέρει το κόστος σε χρόνο ή σε χώρο στην ταινία, που χρειάζεται μια μηχανή Turing ώστε να το επιλύσει. Θεωρούμε ότι μία μετάβαση της μηχανής, μέσω της συνάρτησης δ γίνεται σε έναν χρόνο.

Ορισμός 1.4: **FP** = Το σύνολο όλων των συναρτήσεων που υπολογίζονται σε πολυωνυμικό χρόνο από μία ντετερμινιστική μηχανή Turing.

Ορισμός 1.5: **FL** = Το σύνολο όλων των συναρτήσεων που υπολογίζονται σε επιπλέον λογαριθμικό χώρο από μία ντετερμινιστική μηχανή Turing.

Στον παραπάνω ορισμό λέμε «επιπλέον» διότι στο χώρο της ταινίας συμπεριλαμβάνεται και η είσοδος την οποία δεν θέλουμε να λάβουμε υπόψιν.

Ορισμός 1.6: Θα συμβολίζουμε με $\boldsymbol{PNTM}$ το σύνολο των μηχανών Turing που έχουν πολυωνυμικά φραγμένο αριθμό υπολογιστικών μονοπατιών.

Ορισμός 1.7: Θα λέμε ότι **ένας αλγόριθμος εκτελείται σε χρόνο $\boldsymbol{f(n)}$**, όπου n είναι το μέγεθος της εισόδου, όταν εκτελεί $f(n)$ το πλήθος στοιχειώδεις πράξεις. Οι στοιχειώδεις πράξει είναι η πρόσθεση, πολλαπλασιασμός, διαίρεση, αφαίρεση δύο αριθμών, η ανάθεση και ο έλεγχος μιας συνθήκης.

Στην θεωρία πολυπλοκότητας μας ενδιαφέρει πως συμπεριφέρεται ασυμπτωτικά ένας αλγόριθμος ως προς το μέγεθος της εισόδου. Οπότε η ακρίβεια στον χρόνο εκτέλεσης του αλγορίθμου δεν είναι πρωτεύουσας σημασίας, αν και στην πράξη παίζει σημαντικό ρόλο. Γενικά χρησιμοποιούμε τους παρακάτω συμβολισμούς.

Ορισμός 1.8: Για κάθε συνάρτηση $f : \mathbb{N} \to \mathbb{N}$ έχουμε τους εξής συμβολισμούς

$\boldsymbol{O(f(n))}$: συμβολίζει το σύνολο όλων των συναρτήσεων $g : \mathbb{N} \to \mathbb{N}$ για τις οποίες υπάρχει σταθερά $c > 0$ και $n_0 \in \mathbb{N}$ ώστε $\forall\, n \geq n_0$ να ισχύει $g(n) \leq c\, f(n)$

$\boldsymbol{\Omega(f(n))}$: συμβολίζει το σύνολο όλων των συναρτήσεων $g : \mathbb{N} \to \mathbb{N}$ για τις οποίες υπάρχει σταθερά $c > 0$ και $n_0 \in \mathbb{N}$ ώστε $\forall\, n \geq n_0$ να ισχύει $g(n) \geq c\, f(n)$

$\boldsymbol{\Theta(f(n))}$: συμβολίζει το σύνολο όλων των συναρτήσεων $g : \mathbb{N} \to \mathbb{N}$ για τις οποίες ισχύει $g(n) \in O(f(n))$ και $g(n) \in \Omega(f(n))$

$\boldsymbol{o(f(n))}$: συμβολίζει το σύνολο όλων των συναρτήσεων $g : \mathbb{N} \to \mathbb{N}$ για τις οποίες ισχύει

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$$

$\boldsymbol{\omega(f(n))}$: συμβολίζει το σύνολο όλων των συναρτήσεων $g : \mathbb{N} \to \mathbb{N}$ για τις οποίες ισχύει

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Οι αναγωγές πολυωνυμικού χρόνου είναι ένα χρήσιμο εργαλείο στη θεωρία υπολογισιμότητας και πολυπλοκότητας. Με τις αναγωγές συσχετίζουμε προβλήματα ώστε να επιλύουμε το ένα μέσω του άλλου. Ακόμη τις χρησιμοποιούμε για να αποδείξουμε ότι ένα πρόβλημα είναι δύσκολο να επιλυθεί ανάγοντας το σε ένα που ήδη γνωρίζουμε ότι είναι δυσεπίλυτο. Για να γίνουμε πιο ακριβείς αν ένα πρόβλημα Α ανάγεται σε ένα πρόβλημα Β

τότε το Α είναι τόσης υπολογιστικής δυσκολίας όσο και το Β. Παραθέτουμε τους παρακάτω τύπους αναγωγών.

Συμβολισμός: Θα συμβολίζουμε το πρόβλημα Α με είσοδο x ως $A(x)$

Ορισμός 1.9: Θα λέμε ότι έχουμε **αναγωγή πολυωνυμικού χρόνου πολλά προς ένα**, από το πρόβλημα Α στο πρόβλημα Β (τα οποία συνήθως είναι προβλήματα απόφασης) αν

$$\exists\, f \in FP \;\; \tau\acute{\epsilon}\tau o\iota\alpha\; \acute{\omega}\sigma\tau\epsilon \;\; \forall\; \epsilon\acute{\iota}\sigma o\delta o\; \tau o\upsilon\; A\;\; x,\;\; \alpha\nu\; y\; \mu\acute{\iota}\alpha\; \lambda\acute{\upsilon}\sigma\eta\; \tau o\upsilon\; B\big(f(x)\big)$$

$$\Longrightarrow y\; \lambda\acute{\upsilon}\sigma\eta\; \kappa\alpha\iota\; \tau o\upsilon\; A(x)$$

Η αναγωγή πολυωνυμικού χρόνου πολλά προς ένα είναι γνωστή και ως **αναγωγή κατά Karp**.

Ορισμός 1.10: Θα λέμε ότι έχουμε **αναγωγή πολυωνυμικού χρόνου κατά Turing**, από το πρόβλημα Α στο πρόβλημα Β (τα οποία συνήθως είναι προβλήματα απόφασης) αν g μία συνάρτηση που επιλύει το πρόβλημα Β για οποιαδήποτε τιμή και

$$\exists\, f \in FP \;\; \tau\acute{\epsilon}\tau o\iota\alpha\; \acute{\omega}\sigma\tau\epsilon \;\; \alpha\nu\;\; y\; \mu\acute{\iota}\alpha\; \lambda\acute{\upsilon}\sigma\eta\; \tau o\upsilon\; f\big(g(B)\big)$$

$$\Longrightarrow y\; \lambda\acute{\upsilon}\sigma\eta\; \kappa\alpha\iota\; \tau o\upsilon\; A$$

Η αναγωγή πολυωνυμικού χρόνου κατά Turing είναι γνωστή και ως **αναγωγή κατά Karp**.

Αν έχουμε NP-δύσκολα προβλήματα τότε δεν μπορούμε να κατασκευάσουμε αλγόριθμο που να έχουμε και τις παρακάτω ιδιότητες ταυτόχρονα

1. Να επιστρέφει την ακριβή λύση
2. Να επιλύει όλα τα στιγμιότυπα
3. Να εκτελείται σε πολυωνυμικό χρόνο

Αν αγνοήσουμε την συνθήκη (1), τότε έχουμε προσεγγιστικούς αλγορίθμους.

Ορισμός 1.11: Ένα **πρόβλημα βελτιστοποίησης** είναι μια τετράδα $(I, S, v, goal)$, όπου

- I : είναι το σύνολο των στιγμιότυπων του προβλήματος
- S: είναι μια συνάρτηση που αντιστοιχεί σε κάθε στιγμιότυπο στο σύνολο των εφικτών λύσεων του.
- v: η αντικειμενική συνάρτηση που αντιστοιχεί σε κάθε εφικτή λύση έναν θετικό ακέραιο.
- $goal$: min ή max για πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης της αντικειμενικής συνάρτησης, αντίστοιχα.

Η τιμή της αντικειμενικής συνάρτησης για την βέλτιστη λύση για είσοδο x συμβολίζεται με $OPT(x)$ και είναι ίση με $goal\{v(y) : y \in S(x)\}$.

Ορισμός 1.12: Ένας πολυωνυμικός αλγόριθμος Μ είναι p-προσεγγιστικός για ένα πρόβλημα μεγιστοποίησης αν για κάθε $x \in I$ επιστρέφει μία λύση $M(x) \in S(x)$ τέτοια ώστε

$$\frac{v(M(x))}{OPT(x)} \leq p$$

Αντίστοιχα ορίζεται p-προσεγγιστικός αλγόριθμος για το πρόβλημα ελαχιστοποίησης.

Ορισμός 1.13: Η **κλάση πολυπλοκότητας FPTAS** περιέχει όλα τα προβλήματα για τα οποία υπάρχει πλήρως πολυωνυμικού χρόνου προσεγγιστικό σχήμα, δηλαδή $(1+\varepsilon)$-προσεγγιστικός αλγόριθμος για κάθε σταθερά $\varepsilon > 0$, που επιπλέον ο χρόνος που χρειάζεται είναι πολυωνυμικός και ως προς το $1/\varepsilon$.

Ορισμός 1.14: Η **κλάση πολυπλοκότητας FPRAS** περιέχει όλα τα προβλήματα για τα οποία υπάρχει πιθανοτικό πλήρως πολυωνυμικού χρόνου προσεγγιστικό σχήμα, δηλαδή αλγόριθμος που δίνει, για κάθε σταθερά $\varepsilon > 0$, $(1+\varepsilon)$-προσέγγιση με συντριπτική πιθανότητα, που επιπλέον ο χρόνος που χρειάζεται είναι πολυωνυμικός και ως προς το $1/\varepsilon$.

2 ΟΡΊΖΟΥΣΑ ΚΑΙ PERMANENT

Σε αυτό το κεφάλαιο θα μελετήσουμε και θα συγκρίνουμε την ορίζουσα με την permanent, ώστε να γίνουν εμφανείς οι διαφορές των δύο αυτών συναρτήσεων και να είναι ξεκάθαρο, από τις αλγεβρικές τους ιδιότητες, γιατί η permanent είναι υπολογιστικά πιο δύσκολη συνάρτηση.

Θα εργαστούμε στον χώρο των $(n \times n)$ τετραγωνικών πινάκων με συντελεστές από το σώμα K, $M_n(K)$. (Μερικά χαρακτηριστικά παραδείγματα σωμάτων είναι τα σύνολα: $\mathbb{R}$ των πραγματικών, $\mathbb{C}$ των μιγαδικών αριθμών με τις γνωστές πράξεις πρόσθεσης και πολλαπλασιασμού)

Συμβολισμός: Θα συμβολίζουμε έναν πίνακα $A \in M_n(K)$ συναρτήσει των στηλών του, με $A = [\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_n}]$, όπου $\widetilde{\alpha_i}$ είναι η i-οστή στήλη του Α.

2.1 Μερικά Στοιχεία για τις Μεταθέσεις

Θα παραθέσουμε κάποιους βασικούς ορισμούς για τις μεταθέσεις, ένα βιβλίο που προτείνουμε για αναλυτικότερη μελέτη στις μεταθέσεις είναι το [**1**]

Ορισμός 2.1: Έστω ένα σύνολο $A \neq \emptyset$, μια **μετάθεση** είναι μια απεικόνιση $f: A \to A$ ένα προς ένα και επί. Δηλαδή η f απεικονίζει ένα στοιχείο του Α σε ένα άλλο ή και στον εαυτό του. Η εικόνα της f είναι μία **διάταξη** του Α.

Το **σύνολο των μεταθέσεων** του συνόλου $A = \{a_1, \dots, a_n\}$, συμβολίζεται με S_n, και είναι

$S_n = \{ f : A \to A \mid f \text{ '1 − 1' } και \ επί \}$. Ο αριθμός των στοιχείων του S_n είναι $n!$.

Παράδειγμα 2.1:

Έστω $A = \{1,2,3\} \subset \mathbb{N}$ τότε

$$\sigma_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \sigma_2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \sigma_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \sigma_4 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

$$\sigma_5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \sigma_6 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

Με $|S_3| = 6$ και χρησιμοποιείται και ο σμυβολισμός $\sigma_2(2) = 3, \sigma_4(3) = 1$

Παρατηρούμε ότι

$$1 \overset{\sigma_6}{\to} 3 \overset{\sigma_6}{\to} 1 \ και \ 2 \overset{\sigma_6}{\to} 2$$

$$1 \overset{\sigma_5}{\to} 3 \overset{\sigma_5}{\to} 2 \overset{\sigma_5}{\to} 1 \overset{\sigma_5}{\to} 3$$

$$1 \overset{\sigma_1}{\to} 1 \ , \ 2 \overset{\sigma_1}{\to} 2 \ και \ 3 \overset{\sigma_1}{\to} 3$$

Αν ξεκινήσουμε με κάποιο στοιχείο α_i του A και εφαρμόσουμε την σύνθεση συναρτήσεων για την μετάθεση σ

$$\underbrace{(\sigma \circ \sigma \circ \cdots \circ \sigma)}_{k-\varphi o \rho \acute{\varepsilon} \varsigma}(\alpha_i) = \sigma^{(k)}(a_i)$$

Κάθε φορά που εφαρμόζεται η σ θα προκύπτει ένα στοιχείο του Α. Το σύνολο των στοιχείων του Α που θα προκύψει, όταν έχουμε ξεκινήσει από το α_i και εφαρμόζουμε την σ ονομάζεται σ-τροχιά ή τροχιά του α_i. Μπορούμε να ορίσουμε την σχέση $\sim_\sigma$ στο Α κατά την οποία $\alpha_i \sim_\sigma \alpha_j$, αν και μόνο αν τα α_i $και$ $το$ α_j ανήκουν στην ίδια τροχιά. Αποδεικνύεται ότι η σχέση $\sim_\sigma$ είναι σχέση ισοδυναμίας στο Α. Πιο συγκεκριμένα έχουμε τους εξής ορισμούς.

Ορισμοί 2.2

- Μία **τροχιά** του στοιχείου $\alpha_i \in A$ είναι η κλάση ισοδυναμίας του α_i ως προς την $\sim_\sigma$

$$[a_i]_\sigma = \{\alpha_j \in A : a_i \sim_\sigma a_j\} = \{\alpha_j \in A : \exists\, k \in \mathbb{Z} : \sigma^{(k)}(a_i) = a_j\}$$

$$\Leftrightarrow \quad [a_i]_\sigma = \{\sigma^{(k)}(a_i) \in A : k \in \mathbb{Z}\}$$

- Μία μετάθεση $\sigma \in S_n$ λέγεται **κύκλος** αν έχει το πολύ μία τροχιά, η οποία περιέχει περισσότερα από ένα στοιχείο.
- Το **μήκος ενός κύκλου** είναι το πλήθος των στοιχείων της μεγαλύτερης τροχιάς του.
- **Αντιμετάθεση** είναι ένας κύκλος μήκους 2. Δηλαδή μια αντιμετάθεση αφήνει όλα τα στοιχεία, ενός συνόλου, εκτός από δύο, στην ίδια θέση και απεικονίζει το καθένα από τα δύο στο άλλο.
- Μια μετάθεση ενός πεπερασμένου συνόλου λέγεται **άρτια** ή **περιττή**, ανάλογα με το αν γράφεται ως σύνθεση άρτιου πλήθους αντιμεταθέσεων ή ως σύνθεση περιττού πλήθους αντιμεταθέσεων αντίστοιχα.

Ένας άλλος τρόπος να δούμε αν μια μετάθεση είναι άρτια ή περιττή είναι μέσω των αντιστροφών (i, k).

$Αν$ $για$ $i < k$ $ισχύει$ $\sigma(i) > \sigma(k)$ τότε έχουμε μία αντιστροφή. Ελέγχουμε όλες τις αντιστροφές στην μετάθεση σ και αν το πλήθος των αντιστροφών είναι άρτιος ή περιττός τότε η σ είναι άρτια ή περιττή αντίστοιχα.

Ορισμός 2.3: Τέλος ορίζουμε το **πρόσημο** ή **δείκτρια** μιας μετάθεσης σ

$$\varepsilon(\sigma) = \begin{cases} 1, & αν\ η\ \sigma\ είναι\ άρτια \\ -1, & αν\ η\ \sigma\ είναι\ περιττή \end{cases}$$

Παράδειγμα 2.2: Έστω πάλι $A = \{1,2,3\} \subset \mathbb{N}$ τότε

$$\sigma_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \sigma_2 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \sigma_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \sigma_4 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$$

$$\sigma_5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \sigma_6 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

Θα βρούμε το $\varepsilon(\sigma_5)$.

1ος τρόπος: Είναι εύκολο να δούμε ότι οι σ_6 και η σ_2 είναι αντιμεταθέσεις και ότι προκύπτει

$$\sigma_5 = \sigma_6 \circ \sigma_2 \implies \varepsilon(\sigma_5) = 1$$

$2^{ος}$ τρόπος:

$$1 < 2 \ \mu\varepsilon \ \sigma_5(1) > \sigma_5(2) \implies \alpha\nu\tau\iota\sigma\tau\rho o\varphi\acute{\eta}, \delta\varepsilon\nu \ \acute{\varepsilon}\chi o\upsilon\mu\varepsilon \ \alpha\nu\tau\iota\sigma\tau\rho o\varphi\acute{\eta} \ \gamma\iota\alpha \ \tau o \ \zeta\varepsilon\acute{\upsilon}\gamma o\varsigma \ (2,3)$$

$$\kappa\alpha\iota \ \ 1 < 3 \ \mu\varepsilon \ \sigma_5(1) > \sigma_5(3) \implies 1 \ \alpha\nu\tau\iota\sigma\tau\rho o\varphi\acute{\eta}$$

Έχουμε συνολικά 2 αντιστροφές άρα η μετάθεση είναι άρτια.

■

2.2 Ορίζουσα Πίνακα

Σε οποιδήπποτε βιβλίο και να κοιτάξει κανείς μπορεί να βρει τον ορισμό της ορίζουσας. Εμείς προτείνουμε τα [**2**],[**1**], [**3**], [**4**]

Ορισμός 2.4: Η **ορίζουσα** για έναν τετραγωνικό πίνακα $A \in M_n(K)$ ορίζεται ως η συνάρτηση $\boldsymbol{det}: \ M_n(K) \to K, \ A = [\widetilde{\alpha_1}, \dots, \widetilde{\alpha_n}] \to detA,$ με τύπο:

$$detA = \sum_{\sigma \in S_n} \varepsilon(\sigma) \prod_{i=1}^{n} a_{\sigma(i)i}$$

με τις εξής ιδιότητες:

i. $\det(\widetilde{\alpha_1}, \dots, \lambda\widetilde{\alpha_i} + \mu\boldsymbol{v}, \dots, \widetilde{\alpha_n}) = \lambda \det(\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_n}) + \mu \det(\widetilde{\alpha_1}, \dots, \boldsymbol{v}, \dots, \widetilde{\alpha_n})$,
$\forall \ \lambda, \mu \in K, \ i = 1, \dots, n, \ \boldsymbol{v} \in K^n$

ii. Αν υπάρχουν $\widetilde{\alpha_i}, \widetilde{\alpha_j} \ \mu\varepsilon \ i \neq j \ \ \omega\sigma\tau\varepsilon \ \widetilde{\alpha_i} = \widetilde{\alpha_j}, \tau\acute{o}\tau\varepsilon \ detA = 0 \quad i, j = 1, \dots, n$

iii. $detI_\nu = 1 \ \ \acute{o}\pi o\upsilon \ I_\nu \ o \ \tau\alpha\upsilon\tau o\tau\iota\kappa\acute{o}\varsigma \ \pi\acute{\iota}\nu\alpha\kappa\alpha\varsigma \ \sigma\tau o\nu \ M_n(K)$

Η ιδιότητα (i) λέει ότι η ορίζουσα είναι μια ***πολυγραμμική*** *συνάρτηση. Ενώ η ιδιότητα (ii) δηλώνει ότι αν δύο στήλες του πίνακα Α ταυτίζονται τότε η ορίζουσα του Α θα είναι μηδέν.*

2.2.1 Ιδιότητες Ορίζουσας

Πρόταση 2.1: Έστω $A \in M_n(K), \mu\varepsilon \ A = \left[\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_j}, \dots, \widetilde{\alpha_n}\right]$ αν δημιουργήσουμε τον πίνακα A' έτσι ώστε $A' = \left[\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i} + \lambda\widetilde{\alpha_j}, \dots, \widetilde{\alpha_j}, \dots, \widetilde{\alpha_n}\right]$, με $\lambda \in K$ τότε

$$\det(A) = \det\left(A'\right)$$

Απόδειξη:

$$\det\left(A'\right) = \det\left(\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i} + \lambda\widetilde{\alpha_j}, \dots, \widetilde{\alpha_j}, \dots, \widetilde{\alpha_n}\right) =$$
$$= \det\left(\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_j}, \dots, \widetilde{\alpha_n}\right) + \lambda \det\left(\widetilde{\alpha_1}, \dots, \widetilde{\alpha_j}, \dots, \widetilde{\alpha_j}, \dots, \widetilde{\alpha_n}\right) = \det A$$

Η ιδιότητα προκύπτει άμεσα από την εφαρμογή της ιδιότητας (ii) του 2.4. ■

Πρόταση 2.2: Έστω $A \in M_n(K), \mu\varepsilon \ A = [\widetilde{\alpha_1}, \dots, \widetilde{\alpha_n}]$ αν εφαρμόσουμε μία μετάθεση σ στις στήλες του Α και έχουμε $A' = \left[\widetilde{\alpha_{\sigma(1)}}, \dots, \widetilde{\alpha_{\sigma(n)}}\right]$ τότε

$$\det\left(\widetilde{\alpha_{\sigma(1)}}, \dots, \widetilde{\alpha_{\sigma(n)}}\right) = \varepsilon(\sigma)\det(\widetilde{\alpha_1}, \dots, \widetilde{\alpha_n})$$

Απόδειξη: Αρκεί να αποδείξουμε ότι μία αντιμετάθεση δύο στηλών $\widetilde{\alpha_i}, \widetilde{\alpha_j}$ αλλάζει το πρόσημο της ορίζουσας. Άρα θεωρούμε ότι στον πίνακα Α όλες οι στήλες παραμένουν σταθερές και αλλάζουν οι $\widetilde{\alpha_i}, \widetilde{\alpha_j}$ και θεωρούμε $\det A = d\left(\widetilde{\alpha_i}, \widetilde{\alpha_j}\right)$ τότε από τις ιδιότητες (i) και (ii) του ορισμού **2.4** έχουμε

$$0 = d\left(\widetilde{\alpha_i} + \widetilde{\alpha_j}, \widetilde{\alpha_i} + \widetilde{\alpha_j}\right) \Leftrightarrow d\left(\widetilde{\alpha_i}, \widetilde{\alpha_j}\right) + d\left(\widetilde{\alpha_j}, \widetilde{\alpha_i}\right) = 0 \Leftrightarrow d\left(\widetilde{\alpha_i}, \widetilde{\alpha_j}\right) = -d\left(\widetilde{\alpha_j}, \widetilde{\alpha_i}\right)$$

■

Πρόταση 2.3: Για κάθε $A \in M_n(K)$, ισχύει:

$$\det(A) = \det(A^T)$$

Από αυτή την πρόταση συμπεραίνουμε πως ό,τι ισχύει για την ορίζουσα $det(A)$ ως προς τις στήλες του Α, θα ισχύει και ως προς τις γραμμές του.

Απόδειξη:

Έστω $A^T = \left(\beta_{ij}\right) = \left(a_{ji}\right)$ θα έχουμε

$$\det(A^T) = \sum_{(i_1 \dots i_n) \in S_n} \varepsilon(i_1 \dots i_n)\beta_{i_1 1} \dots \beta_{i_n n} = \sum_{(i_1 \dots i_n) \in S_n} \varepsilon(i_1 \dots i_n)\alpha_{1 i_1} \dots \alpha_{n i_n}$$

Αν μεταθέσουμε τους παράγοντες του τυχαίου όρου του παραπάνω αθροίσματος, έστω $\varepsilon(i_1 i_2 \dots i_n)a_{i_1 1}a_{i_2 2} \dots a_{i_n n}$, έτσι ώστε οι δείκτες να πάρουν τη φυσική τους διάταξη, τότε ο όρος θα γίνει $\varepsilon(j_1 \dots j_n)\alpha_{1 j_1} \dots \alpha_{n j_n}$, όπου $(j_1, \dots, j_n)$ είναι η αντίστροφη μετάθεση της $(i_1, \dots, i_n)$, οπότε

$$(j_1, \dots, j_n) \in S_n \ \ και \ \ \varepsilon(j_1, \dots, j_n) = \varepsilon(i_1, \dots, i_n)$$

Άρα ο τυχαίος όρος του αθροίσματος είναι και όρος του αθροίσματος που ορίζει την $\det(A^T)$ και αφού το πλήθος των όρων των δύο αθροισμάτων είναι το ίδιο, δηλαδή $n!$ και είναι φανερό ότι

$$\det(A) = \det(A^T)$$

■

Πρόταση 2.4 (Πολλαπλασιαστικότητα): Για κάθε $A, B \in M_n(K)$, ισχύει:

$$\det(A \cdot B) = \det(A)\det(B)$$

Απόδειξη:

Για την απόδειξη θα συμβολίζουμε με $\underline{\alpha_i}$ την i-οστή γραμμή του Α.

$$A\widetilde{b_j} = \begin{bmatrix} \underline{\alpha_1} \\ \underline{\alpha_2} \\ \vdots \\ \underline{\alpha_n} \end{bmatrix} \widetilde{b_j} = \begin{bmatrix} \underline{\alpha_1}\widetilde{b_j} \\ \underline{\alpha_2}\widetilde{b_j} \\ \vdots \\ \underline{\alpha_n}\widetilde{b_j} \end{bmatrix} = \begin{bmatrix} a_{11}b_{1j} & +\cdots+ & a_{1n}b_{nj} \\ \vdots & \vdots & \vdots \\ a_{n1}b_{nj} & +\cdots+ & a_{nn}b_{nj} \end{bmatrix} = \sum_{i=1}^{n} b_{ij}\,\widetilde{a_i}$$

Άρα έχουμε

$$\det(AB) = \det\left(\sum_{i_1=1}^{n} b_{i1}\widetilde{a_i}\,, \dots, \sum_{i_n=1}^{n} b_{in}\,\widetilde{a_i}\right) = \sum_{i_1 \dots i_n=1}^{n} b_{i_1 1} \dots b_{i_n n}\,\det(\widetilde{a_1}, \dots, \widetilde{a_n})$$

$$\det A = \sum_{(i_1 \dots i_n)\in S_n} b_{i_1 1} \dots b_{i_n n}\;\varepsilon(i_1 \dots i_n)\det(\widetilde{a_1}, \dots, \widetilde{a_n}) =$$

$$= \det(\widetilde{a_1}, \dots, \widetilde{a_n}) \sum_{(i_1 \dots i_n)\in S_n} \varepsilon(i_1 \dots i_n) b_{i_1 1} \dots b_{i_n n} \;=$$

$$= \det(A) \cdot \det(B)$$

■

<u>Πρόταση 2.5:</u> Αν ο πίνακας $A = [\alpha_{ij}] \in M_n(K)$ είναι άνω τριγωνικός είτε κάτω τριγωνικός είτε διαγώνιος τότε

$$\det A = a_{11}a_{22} \dots a_{nn}$$

<u>Απόδειξη:</u> Γνωρίζουμε ότι

$$\det A = \sum_{(i_1 \dots i_n)\in S_n} \varepsilon(i_1 \dots i_n) a_{i_1 1} \dots a_{i_n n}$$

Αν είναι κάτω τριγωνικός ο πίνακας Α τότε για $i > j$ τότε $\alpha_{ij} = 0$. Θα προσπαθήσουμε να προσδιορίσουμε τους μη μηδενικούς όρους.

Επειδή $\alpha_{i_1 1} = 0$ για $i_1 > 1$, θα πρέπει $\alpha_{i_1 1} = a_{11}$, δηλαδή $i_1 = 1$.

Όμοια για $\alpha_{i_2 2} = 0$ για $i_2 > 2$, τότε $\alpha_{i_2 2} = a_{12}$ ή $\alpha_{i_2 2} = a_{22}$, δηλαδή $i_2 = 1$ ή $i_2 = 2$.

Όμως για $\varepsilon(i_1 \dots i_n) = \varepsilon(1\,1\,i_3 \dots i_n) = 0$ άρα θα έχουμε μη μηδενικό όρο για $i_2 = 2$.

Αν εργαστούμε με ανάλογο τρόπο τότε η μοναδική περίπτωση στην οποία υπολογίζουμε μη μηδενικό όρο στο άθροισμα της ορίζουσας είναι για $i_3 = 3, i_4 = 4, \dots, i_n = n$. Οπότε

$$\det A = a_{11} \dots a_{nn}$$

Αν ο Α είναι άνω τριγωνικός, τότε ο A^T είναι κάτω τριγωνικός οπότε με βάση την παραπάνω περίπτωση υπολογίζουμε το

$$\det(A^T) = a_{11} \dots a_{nn}$$

Και από την πρόταση **2.3** προκύπτει $\det A = \det(A^T) = a_{11} \dots a_{nn}$

■

Ακόμα ισχύουν και οι παρακάτω ιδιότητες.

Πρόταση 2.6: Για κάθε $A \in M_n(K)$, $A = [\widetilde{\alpha_1}, \dots, \widetilde{\alpha_n}]$ *και* $\lambda \in K$, ισχύουν:

1. $\det(A^{-1}) = \frac{1}{\det(A)} = \det(A)^{-1}$
2. $\det(\widetilde{\alpha_1}, \dots, \lambda\widetilde{\alpha_i}, \dots, \widetilde{\alpha_n}) = \lambda \det(\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_n})$
3. $\det(\lambda A) = \lambda^n \det(A)$

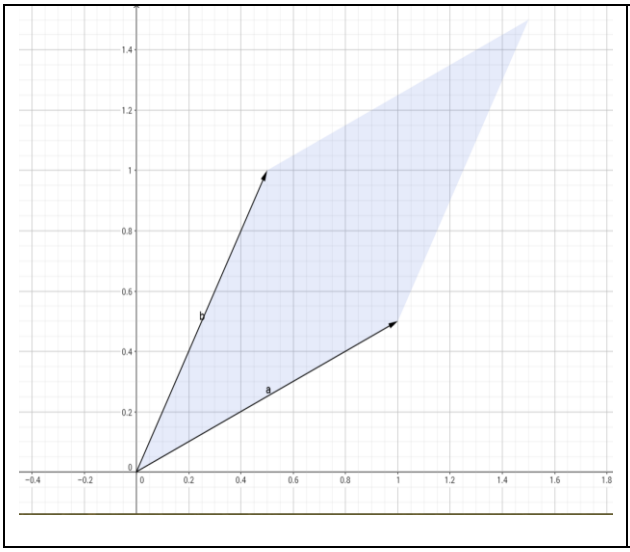
2.2.2 Γεωμετρική Ερμηνεία της Ορίζουσας

Για να μπορέσουμε να περιγράψουμε γεωμετρικά τι εκφράζει η ορίζουσα θα μελετήσουμε τον ευκλείδειο χώρο $\mathbb{R}^3$ και $\mathbb{R}^2$. Όπως αναφέρεται και στο [**3**]

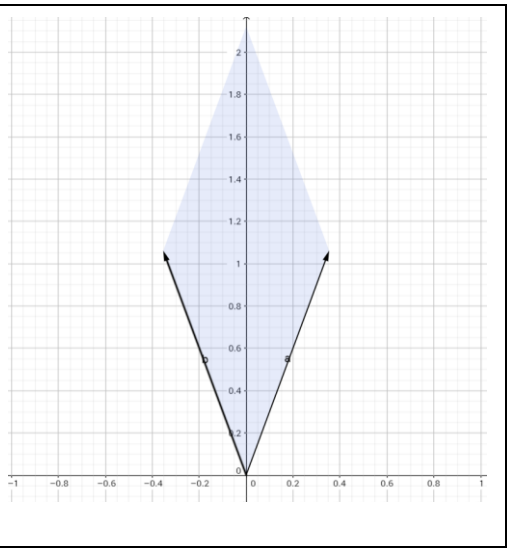
Η ορίζουσα στον $\mathbb{R}^3$ μετράει τον όγκο του παραλληλεπιπέδου που σχηματίζουν τα διανύσματα του πίνακα Α, ενώ στο $\mathbb{R}^2$ μετράει εμβαδόν του παραλληλογράμμου που σχηματίζουν τα διανύσματα του Α.

Πιο συγκεκριμένα αν έχουμε έναν πίνακα $A \in M_2(\mathbb{R})$, τέτοιος ώστε $A = [\widetilde{\alpha_1}, \widetilde{\alpha_2}]$ με $\widetilde{\alpha_1}, \widetilde{\alpha_2} \in \mathbb{R}^2$.

Αν ο βαθμός του πίνακα είναι $rank(A) = 2$, δηλαδή ο Α περιέχει δύο γραμμικώς ανεξάρτητα διανύσματα, τότε αυτά δημιουργούν ένα παραλληλόγραμμο όπως φαίνεται και στα σχήματα 2.1 και 2.2 αντίστοιχα. Η ορίζουσα υπολογίζει το εμβαδόν του εν λόγω παραλληλογράμμου.



Σχήμα 2.1



Σχήμα 2.2

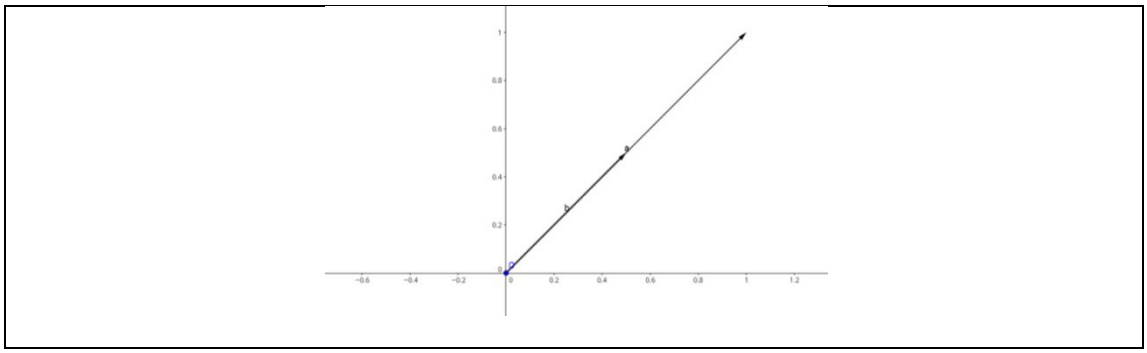
Στο σχήμα 2.1 έχουμε ορίσει $A = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}$, ενώ στο σχήμα 2.2 έχουμε στρίψει τον Α κατά $45°$ δεξιόστροφα, με χρήση του πίνακα στροφής U όπως φαίνεται παρακάτω

$$A = \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} \cos 45 & -\sin 45 \\ \sin 45 & \cos 45 \end{bmatrix} \; και \; AU = \begin{bmatrix} 0.35355 & -0.35355 \\ 1.06066 & 1.06066 \end{bmatrix}$$

Είναι σημαντικό να παρατηρήσουμε, ότι η ισχύει η πολλάπλασιαστική ιδιότητα της ορίζουσας και φαίνεται και από το εμβαδόν

$$\det AU = 0.75 = 0.75 \cdot 1 = \det A \cdot \det U$$

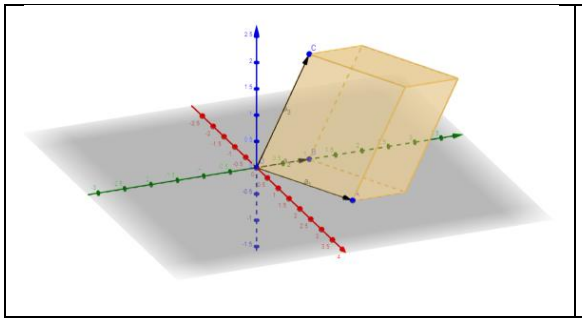
Βλέπουμε ότι αν $rank(A) < 2$ τότε τα δύο διανύσματα είναι συγγραμμικά και δεν σχηματίζουν παραλληλόγραμμο, αλλά ευθύγραμμα τμήματα. Ως γνωστόν ένα ευθύγραμμο τμήμα έχει μηδενικό εμβαδόν και αντίστοιχα είναι εύκολο να ελεγχθεί ότι η ορίζουσα σε αυτή την περίπτωση είναι μηδέν.



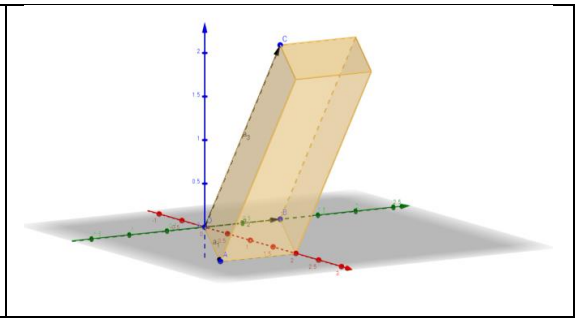
Σχήμα 2.3

Αντίστοιχα για τον $\mathbb{R}^3$, αν έχουμε έναν πίνακα $A \in M_3(\mathbb{R})$, τέτοιος ώστε $A = [\widetilde{\alpha_1}, \dots, \widetilde{\alpha_3}]$ με $\widetilde{\alpha_1}, \widetilde{\alpha_2}, \widetilde{\alpha_3} \in \mathbb{R}^3$.

Αν ο βαθμός του πίνακα Α είναι $rank(A) = 3$, τότε αυτό σημαίνει ότι τα διανύσματα $\widetilde{\alpha_1}, \widetilde{\alpha_2}, \widetilde{\alpha_3}$ είναι γραμμικά ανεξάρτητα και ορίζουν ένα παραλληλεπίπεδο στο χώρο. Η ορίζουσα μετράει τον όγκο του παραλληλεπιπέδου όπως φαίνεται και στα σχήματα 2.4 και 2.5 αντίστοιχα.



Σχήμα 2.4



Σχήμα 2.5

Αν τώρα ο βαθμός του πίνακα Α είναι $rank(A) \leq 2$, ας υποθέσουμε ότι $rank(A) = 2$, δηλαδή έχουμε δύο γραμμικώς ανεξάρτητα διανύσματα και το τρίτο είναι γραμμικά

εξαρτημένο από τα άλλα δύο. Ας πούμε ότι το $\widetilde{\alpha_3}$ είναι γραμμικά εξαρτημένο από τα $\widetilde{\alpha_1}, \widetilde{\alpha_2}$ δηλαδή υπάρχουν $\kappa, \lambda \in \mathbb{R}$ ώστε

$$\widetilde{\alpha_3} = \kappa\, \widetilde{\alpha_1} + \lambda \widetilde{\alpha_2}$$

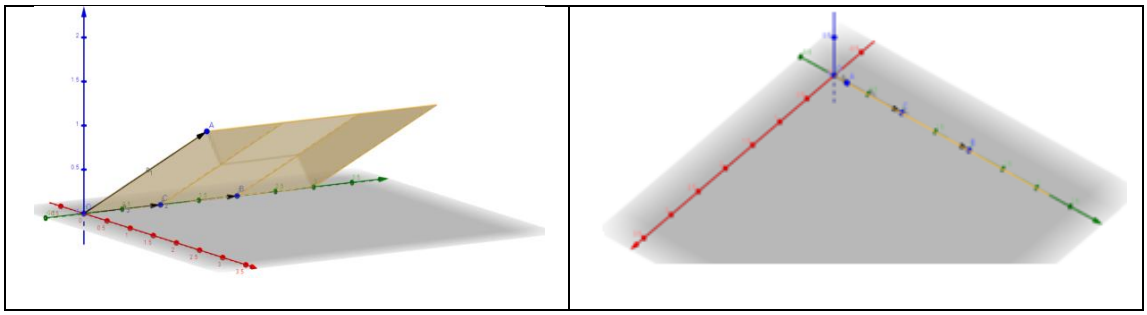
Τότε τα $\widetilde{\alpha_1}$ και $\widetilde{\alpha_2}$ ορίζουν ένα επίπεδο στον $\mathbb{R}^3$ και το $\widetilde{\alpha_3}$ ανήκει σε αυτό το επίπεδο. Σε αυτή την περίπτωση γνωρίζουμε ότι ένα επίπεδο έχει μηδενικό όγκο και ακόμη

$$\det A = \det(\widetilde{\alpha_1}, \widetilde{\alpha_2}, \widetilde{\alpha_3}) = \det(\widetilde{\alpha_1}, \widetilde{\alpha_2}, \kappa\widetilde{\alpha_1} + \lambda\widetilde{\alpha_2}) = \det(\widetilde{\alpha_1}, \widetilde{\alpha_2}, \kappa\widetilde{\alpha_1}) + \det(\widetilde{\alpha_1}, \widetilde{\alpha_2}, \lambda\widetilde{\alpha_2}) =$$

$$= \kappa \cdot \det(\widetilde{\alpha_1}, \widetilde{\alpha_2}, \widetilde{\alpha_1}) + \lambda \cdot \det(\widetilde{\alpha_1}, \widetilde{\alpha_2}, \widetilde{\alpha_2}) = 0$$

Η τελευταία ισότητα προκύπτει από την δεύτερη ιδιότητα του ορισμού **2.4**.

Αντίστοιχα αν είναι τα διανύσματα του Α συνευθειακά, δηλαδή $rank(A) = 1$, τότε η ευθεία έχει μηδενικό όγκο και $\det A = 0$.



Σχήμα 2.6

Με ανάλογο τρόπο μπορεί να γενικευθεί η γεωμετρική έκφραση της ορίζουσας και σε ένα χώρο n διαστάσεων. Έστω λοιπόν ο διανυσματικός χώρος V με διάσταση $\dim V = n$. Αν όλα τα διανύσματα του πίνακα $A_{n\times n}$ είναι γραμμικώς ανεξάρτητα τότε η ορίζουσα υπολογίζει το «υπερεμβαδόν» ενός πολύτοπου. Το πολύτοπο που σχηματίζουν τα n διανύσματα του Α είναι ένα γενικευμένο παραλληλόγραμμο του n-διάστατου χώρου, το οποίο θα έχει «υπερεμβαδόν» έναν θετικό αριθμό, δηλαδή $\det A > 0$, όταν και μόνο όταν το γενικευμένο παραλληλόγραμμο ως σύνολο σημείων, δεν ανήκει εξ' ολοκλήρου σε κανένα γνήσιο υπόχωρο του V. Τέλος με όσα έχουμε αναφέρει μπορούμε να δούμε και γεωμετρικά γιατί η ορίζουσα συνδέεται τόσο στενά με την γραμμική ανεξαρτησία.

2.2.3 Μέθοδοι Αποσύνθεσης Υπολογισμού της Ορίζουσας

Έστω ένας πίνακας Α, όπως μπορούμε να δούμε από τον ορισμό **2.4** της ορίζουσας το άθροισμα απευθύνεται σε όλες τις μεταθέσεις του συνόλου $\{1, \dots, n\}$ οι οποίες είναι $n!$ το πλήθος. Αν εκμεταλλευτούμε τις ιδιότητες των προτάσεων **2.4** και **2.5** μπορούμε να σπάσουμε τον Α σε γινόμενο πινάκων οι οποίοι είναι (άνω ή κάτω) τριγωνικοί ή διαγώνιοι ώστε να μπορέσουμε να υπολογίσουμε την ορίζουσα πιο εύκολα.

Τεχνικές κατά τις οποίες γράφουμε τον πίνακα Α σαν γινόμενο πινάκων, τις ονομάζουμε μεθόδους αποσύνθεσης και απλά θα τις αναφέρουμε και δεν θα τις αναλύσουμε. Για τον αναγνώστη που ενδιαφέρεται τον παραπέμπουμε στα εξής συγγράμματα [**4**] και[**5**].

Εμείς αναφέρουμε την μέθοδο LU η οποία σπάει έναν πίνακα A σε έναν άνω τριγωνικό U, έναν κάτω τριγωνικό πίνακα L και έναν πίνακα μεταθέσεων P

$$A = PLU$$

Σημειώνουμε ότι η LU για έναν πίνακα A $n \times n$, υπολογίζεται σε χρόνο $O(n^3)$. Άρα έχουμε ότι

$$\det A = \det PLU = \varepsilon(\sigma) \det LU = \varepsilon(\sigma) \det L \det U \qquad (2.a)$$

Όπου ε(σ) το πρόσημο της μετάθεσης που υποδεικνύει ο πίνακας μεταθέσεων P. Ένας τρόπος να το υπολογίσουμε είναι μετρώντας τις αντιμεταθέσεις του πίνακα P γενικά δεν είναι δύσκολο να υπολογίσουμε το $\varepsilon(\sigma)$.

Κατά την υλοποίηση της διαδικασίας LU μπορούμε να κατασκευάσουμε τον πίνακα L ώστε όλα τα στοιχεία που ανήκουν στην κύρια διαγώνιο του να είναι 1 και συνεπώς

$$\det L = 1$$

Έτσι η παραπάνω σχέση (2.a) γράφεται

$$\det A = \varepsilon(\sigma) \det U$$

Ο χρόνος που χρειαζόμαστε για να υπολογίσουμε την ορίζουσα ενός πίνακα A $n \times n$ είναι όσο για να υπολογίσουμε τους πίνακες LU, δηλαδή $O(n^3)$, συν το χρόνο που χρειάζεται να υπολογίσουμε την ορίζουσα του U, δηλαδή O(n), αφού από την πρόταση **2.5**

$$\det U = u_{11} u_{22} \cdot \ldots \cdot u_{nn}$$

Εκμεταλλευόμενοι λοιπόν, τις ιδιότητες της ορίζουσας, την υπολογίζουμε σε $O(n^3 + n) = O(n^3)$ χρόνο, κάτι που είναι αξιοθαύμαστο, διότι αν ακολουθούσαμε τυφλά τον ορισμό θα την υπολογίζαμε σε χρόνο $O(n!)$

2.3 Η Permanent ενός Πίνακα

2.3.1 Ορισμός

Ορισμός 2.5: Για έναν πίνακα $A \in M_n(K)$ ορίζουμε ως **permanent** την απεικόνιση $\boldsymbol{per}: M_n(K) \to K$, με τύπο

$$per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{\sigma(i)i}$$

Για να υπολογίσουμε την permanent προσθέτουμε τα γινόμενα των άρτιων μεταθέσεων και τα γινόμενα των περιττών μεταθέσεων. Ενώ αν δούμε τον ορισμό **2.4** της ορίζουσας, επειδή υπάρχει το πρόσημο μετάθεσης, προσθέτουμε τα γινόμενα άρτιων μεταθέσεων και

αφαιρούμε τα γινόμενα που προκύπτουν από περιττές μεταθέσεις. Παραθέτουμε παραδείγματα της permanent για n=2 και n=3 για να γίνει αντιληπτή η διαφορά.

Για n=2 έχουμε $per\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad + bc$

Ενώ για n=3, $per\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh + ceg + fha + bdi$

Μπορούμε να δούμε πως ισχύουν οι παρακάτω ιδιότητες:

i. $per(\widetilde{\alpha_1}, \dots, \lambda\widetilde{\alpha_i} + \mu\boldsymbol{v}, \dots, \widetilde{\alpha_n}) = \lambda\,\mathrm{per}(\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_n}) + \mu\,\mathrm{per}(\widetilde{\alpha_1}, \dots, \boldsymbol{v}, \dots, \widetilde{\alpha_n})$
$\forall\, \lambda, \mu \in K, \qquad i = 1, \dots, n\,, \qquad \forall\, \boldsymbol{v} \in K^n$

ii. $per(I_n) = 1,\ όπου\ I_n\ ο\ ταυτοτικός\ πίνακας\ στον\ M_n(K)$

Η permanent διατηρεί τις ιδιότητες (i) *και* (iii) *του ορισμού* **2.4** *της ορίζουσας. Η ειδοποιός διαφορά βρίσκεται στο ότι δεν ικανοποιείται η (ii) και κατά συνέπεια δεν έχει την ιδιότητα της πρότασης* **2.1**.

2.3.2 Ιδιότητες της Permanent

Πρόταση 2.7: Για κάθε $A \in M_n(K)$, ισχύει:

$$\mathrm{per}(A) = \mathrm{per}(A^T)$$

Πρόταση 2.8: Αν ο πίνακας $A = [\alpha_{ij}] \in M_n(K)$ είναι άνω τριγωνικός είτε κάτω τριγωνικός είτε διαγώνιος τότε

$$per(A) = a_{11}a_{22} \dots a_{nn}$$

Οι αποδείξεις των παραπάνω είναι ίδιες με τις αποδείξεις των προτάσεων **2.3** και **2.5** αντίστοιχα.

Αν θεωρήσουμε ε_1 να είναι μία δείκτρια συνάρτηση για την οποία ισχύει

$$\varepsilon_1(\sigma) = \begin{cases} 0, & αν\ η\ σ\ είναι\ άρτια \\ 1, & αν\ η\ σ\ είναι\ περιττή \end{cases}$$

Μια εναλλακτική ιδιότητα που ισχύει σε αντίστοιχη περίπτωση με την (ii) του ορισμού της ορίζουσας είναι η ακόλουθη:

Πρόταση 2.9: Έστω $A \in M_n(K), με\ A = [\widetilde{\alpha_1}, \dots, , \widetilde{\alpha_n}]$ και αν υπάρχουν $\widetilde{\alpha_i}, \widetilde{\alpha_j}\ με\ i \neq j$ τέτοια ώστε $\widetilde{\alpha_i} = \widetilde{\alpha_j}\,, \quad i, j = 1, \dots, n$ τότε ισχύει:

$$\sum_{\sigma \in S_n} \varepsilon_1(\sigma) \prod_{i=1}^{n} a_{\sigma(i)i} = \sum_{\sigma \in S_n} \big(1 - \varepsilon_1(\sigma)\big) \prod_{i=1}^{n} a_{\sigma(i)i}$$

Με άλλα λόγια η παραπάνω πρόταση λέει ότι αν δύο στήλες του πίνακα Α ταυτίζονται, τότε το άθροισμα όλων των άρτιων μεταθέσεων είναι ίσο με το άθροισμα όλων των περιττών μεταθέσεων στην permanent. (ισχύει και για την ορίζουσα με αποτέλεσμα η ορίζουσα να κάνει μηδέν)

<u>Απόδειξη:</u>

Έστω $A = [\widetilde{\alpha_1}, \dots, , \widetilde{\alpha_n}] \in M_n(K)$ και $\widetilde{\alpha_i}, \widetilde{\alpha_j}$ $με\ i \neq j$ τέτοια ώστε $\widetilde{\alpha_i} = \widetilde{\alpha_j}$ για κάποια $i, j = 1, ., n$

Άρα ο Α μπορεί να γραφεί $A = \left[\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_j}, \dots, \widetilde{\alpha_n}\right] = [\widetilde{\alpha_1}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_i}, \dots, \widetilde{\alpha_n}]$

Αρκεί να δείξουμε ότι για κάθε άρτια μετάθεση σ του συνόλου $\{1, \dots, n\}$ υπάρχει μία περιττή μετάθεση σ' για την οποία ισχύει

$$a_{\sigma(1),1}\ \alpha_{\sigma(2),2} \cdot \dots \cdot \alpha_{\sigma(n),n} = a_{\sigma'(1),1}\ \alpha_{\sigma'(2),2} \cdot \dots \cdot \alpha_{\sigma'(n),n}$$

Ας πάρουμε μια άρτια μετάθεση $\sigma_0 \in S_n$ $τότε$ επειδή ισχύει $\widetilde{\alpha_i} = \widetilde{\alpha_j}$

$$a_{\sigma_0(1),1} \cdot \dots \cdot \alpha_{\sigma_0(i),i} \cdot \dots \cdot\ \alpha_{\sigma_0(j),j} \cdot \dots \cdot \alpha_{\sigma_0(n),n} = \\ = a_{\sigma_0(1),1} \cdot \dots \cdot \alpha_{\sigma_0(i),j} \cdot \dots \cdot\ \alpha_{\sigma_0(j),i} \cdot \dots \cdot \alpha_{\sigma_0(n)n}$$

Όλα τα στοιχεία $\alpha_{i,j}$ ανήκουν στο σώμα K, γεγονός που εξασφαλίζει πως ισχύει η αντιμεταθετική ιδιότητα στον πολλαπλασιασμό άρα

$$a_{\sigma_0(1),1} \cdot \dots \cdot \alpha_{\sigma_0(i),j} \cdot \dots \cdot\ \alpha_{\sigma_0(j),i} \cdot \dots \cdot \alpha_{\sigma_0(n)n} = \\ = a_{\sigma_0(1),1} \cdot \dots \cdot \alpha_{\sigma_0(j),i} \cdot \dots \cdot\ \alpha_{\sigma_0(i),j} \cdot \dots \cdot \alpha_{\sigma_0(n)n}$$

Τώρα αν ορίσουμε την μετάθεση $\sigma_0' \in S_n$ με $\sigma_0' = \big(\sigma_0(i), \sigma_0(j)\big) \cdot \sigma_0$, όπου $\big(\sigma_0(i), \sigma_0(j)\big) \in S_n$ η αντινμετάθεση του $\sigma_0(i)$ με το $\sigma_0(j)$, δηλαδή η σ_0' αντιστοιχίζει όλα τα στοιχεία $\{1, \dots, \mathrm{n}\}$ όπως η σ_0, μόνο που αντιμεταθέτει το σ_0(i) με το σ_0(j). Τότε

$$a_{\sigma_0(1),1} \cdot \dots \cdot \alpha_{\sigma_0(j),i} \cdot \dots \cdot\ \alpha_{\sigma_0(i),j} \cdot \dots \cdot \alpha_{\sigma_0(n),n} = \\ = a_{\sigma_0'(1),1} \cdot \dots \cdot \alpha_{\sigma_0'(i),i} \cdot \dots \cdot\ \alpha_{\sigma_0'(j),j} \cdot \dots \cdot \alpha_{\sigma_0'(n)n}$$

Επειδή ισχύει ότι $\widetilde{\alpha_i} = \widetilde{\alpha_j}$ αν και μόνο αν $a_{k,i} = a_{k,j}$, για κάθε $k = 1, \dots, n$

Άρα δείξαμε πως για κάθε άρτια μετάθεση σ_0 υπάρχει μια σ_0' τέτοια ώστε

$$a_{\sigma_0(1),1} \cdot \dots \cdot \alpha_{\sigma_0(i),i} \cdot \dots \cdot\ \alpha_{\sigma_0(j),j} \cdot \dots \cdot \alpha_{\sigma_0(n)n} = \\ = a_{\sigma_0'(1),1} \cdot \dots \cdot \alpha_{\sigma_0'(i),i} \cdot \dots \cdot\ \alpha_{\sigma_0'(j),j} \cdot \dots \cdot \alpha_{\sigma_0'(n)n}$$

Μένει να αποδείξουμε πως η σ_0' είναι περιττή. Πράγματι είναι περιττή διότι έχουμε ορίσει ως άρτιες μεταθέσεις τις μεταθέσεις που γράφονται σαν γινόμενο άρτιου πλήθους αντιμεταθέσεων. Συνεπώς αν η σ_0 γράφεται σαν άρτιο πλήθος αντιμεταθέσεων, η $\sigma_0' = \big(\sigma_0(i), \sigma_0(j)\big) \circ \sigma_0$ εξ' ορισμού αποτελείται από περιττό πλήθος αντιμεταθέσεων.

■

Πόρισμα 2.10: Αν δύο γραμμές ή δύο στήλες του Α ταυτίζονται τότε ισχύει:

$$per(A) = 2\sum_{\sigma \in S_n} \varepsilon_1(\sigma) \prod_{i=1}^{n} a_{\sigma(i)i} = 2\sum_{\sigma \in S_n} \big(1 - \varepsilon_1(\sigma)\big) \prod_{i=1}^{n} a_{\sigma(i)i}$$

(Προκύπτει άμεσα από την παραπάνω πρόταση)

Σημαντική Παρατήρηση 2.3:

Μπορούμε να δούμε πως για την Permanent δεν ισχύει η πολλαπλασιαστικότητα (πρόταση **2.4**) και επίσης δεν παραμένει αναλλοίωτη στις γραμμοπράξεις (πρόταση **2.1**). Άρα συμπεραίνουμε ότι δεν μπορούμε να υπολογίσουμε την permanent χρησιμοποιώντας τις τεχνικές με τις οποίες υπολογίζουμε την ορίζουσα.

Από ότι φαίνεται ο μόνος τρόπος να υπολογίσουμε την permanent σε μία γενική περίπτωση ενός πίνακα $A_{n\times n}$ είναι μέσω του ορισμού, πράγμα που σημαίνει ότι χρειάζεται να υπολογισθούν $n!$ γινόμενα κάτι που θα χρειαστεί χρόνο υπολογισμού της τάξης $O(n!)$. Βλέπουμε ότι ήδη για έναν πίνακα 10×10 πρέπει να υπολογισθούν $10! \cong 3{,}63 \cdot 10^6$ γινόμενα.

2.3.3 Ερμηνεία της Permanent

Στην προηγούμενη παράγραφο αναλύσαμε τις αλγεβρικές και γεωμετρικές ιδιότητες της ορίζουσας, η οποία είναι πολλαπλασιαστική, παραμένει αναλλοίωτη στις γραμμοπράξεις ενός πίνακα και υποδεικνύει τη γραμμική εξάρτηση διανυσμάτων.

Η Permanent, σε αντίθεση με την ορίζουσα, δεν έχει τόσες καλές αλγεβρικές ιδιότητες και ούτε φαίνεται να έχει κάποια γεωμετρική ιδιότητα που να συνδέει τα διανύσματα του διανυσματικού χώρου K^n με την ποσότητα $perA$.

Θα περιοριστούμε σε δυαδικούς πίνακες, όπου εκεί έχουμε κάποια αποτελέσματα που συνδέουν την permanent με ιδιότητες γραφημάτων.

Ορισμός 2.6: Ένας **γράφος** $G = (V, E)$ αποτελείται από ένα σύνολο κορυφών V και ένα σύνολο ακμών Ε. Μία **ακμή** συνδέει δύο κορυφές μεταξύ τους και είναι μία δυάδα της μορφής

$$(i, j) \in V \times V$$

Αν σε έναν γράφο $G = (V, E)$ ισχύει ότι για κάθε $(i, j) \in E \Longrightarrow (j, i) \in E$ τότε ονομάζεται **μη κατευθυνόμενος γράφος** ή απλά γράφος.

Αν δεν ισχύει η παραπάνω ιδιότητα δηλαδή αν υπάρχουν ακμές $(i, j) \in E \ \ώστε\ (j, i) \notin E$, τότε ο γράφος ονομάζεται **κατευθυνόμενος**.

Ορισμός 2.7: Ένας **διμερής γράφος** είναι ένας γράφος του οποίου οι κόμβοι μπορούν να χωριστούν σε δύο ξένα σύνολα U και V τέτοια ώστε κάθε ακμή να συνδέει ένα κόμβο του U με ένα κόμβο του V. Τα σύνολα κόμβων U και V ονομάζονται και **μέρη του γράφου**.

Ορισμός 2.8: Ένας πίνακας $A = [a_{ij}]$ με $A \in M_n(\{0,1\})$, θα λέγεται **πίνακας γειτνίασης**, για το γράφημα $G = (V, E)$ αν το σύνολο των κόμβων του G ταυτίζεται με το $\{1, \dots, n\}$, $V \equiv \{1, \dots, n\}$, και το σύνολο ακμών

$$E \equiv \{(i,j) : a_{ij} = 1 \ \kappa\alpha\iota \ i,j \in \{1, \dots, n\}\}$$

Ο Valiant απέδειξε ότι αν ο A είναι πίνακας γειτνίασης ενός γράφου G τότε η permanent υπολογίζει τον αριθμό των Cyclic Covers του G, δηλαδή

$$perA = \#Cyclic\ Cover$$

Αντίστοιχα αν A είναι πίνακας γειτνίασης ενός διμερούς γράφου G τότε η permanent υπολογίζει πόσα Perfect Matching έχει ο G.

$$perA = \#Perfect\ Matching$$

Ορισμός 2.9: Σε ένα γράφημα $G = (V, E)$ το **Cyclic Cover** είναι το σύνολο κατευθυνόμενων κύκλων Hamilton πάνω σε όλα τα υπογραφήματα του G, ώστε οι κόμβοι που ανήκουν στους κύκλους Hamilton να αποτελούν μία διαμέριση του V.

Δηλαδή ένα Cyclic Cover C είναι ένα σύνολο $C = \{H_1, \dots, H_n\}$, όπου $H_1, \dots, H_n$ είναι κατευθηνόμενοι κύκλοι Hamilton πάνω σε υπογραφήματα του G ώστε

i. ένας κόμβος να ανήκει αποκλειστικά σε έναν κύκλο H_i, $i = 1, \dots, n$ και
ii. το σύνολο των κόμβων όλων των κύκλων Hamilton του C θα πρέπει να ταυτίζεται με το V.

Ορισμός 2.10: Σε ένα διμερή γράφο $G = (U, V, E)$ με $|U| = |V|$ τότε αν υπάρχει ένα σύνολο ακμών $E_0 \subseteq E$ ώστε κάθε $u \in U$ να γειτονεύει με ένα ακριβώς $v \in V$, τότε το E_0 είναι ένα **Perfect Matching** για τον γράφο G.

Παρατήρηση 2.4:

Με βάση το πόρισμα **2.10**, αν σε έναν πίνακα γειτνίασης δύο στήλες ή δύο γραμμές ταυτίζονται τότε ο αριθμός των perfect matching ή των cyclic cover είναι ζυγός.

Η permanent μετράει τον αριθμό των Cyclic Cover ή τον αριθμό των perfect matching αντίστοιχα. Παρόλο που είναι εύκολο υπολογιστικά, να βρούμε ένα Perfect matching σε ένα γράφημα, το να υπολογίσουμε το πλήθος τους είναι γνησίως δυσκολότερη διαδικασία. Οπότε το να γνωρίζουμε ότι σχετίζεται με αυτά τα προβλήματα δεν μας δίνει κάποιο πλεονέκτημα ώστε να την υπολογίσουμε εύκολα. Αντιθέτως μας υποδεικνύει ότι το να υπολογίσουμε permanent είναι γνησίως πιο δύσκολο από το να υπολογίσουμε την ορίζουσα. Στο επόμενο κεφάλαιο θα ασχοληθούμε αναλυτικότερα, χρησιμοποιώντας θεωρία πολυπλοκότητας, στο κατά πόσο δύσκολο είναι να υπολογίσουμε την permanent ενός πίνακα.

2.4 Αλγεβρικά Αποτελέσματα για την Permanent

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο η permanent μελετήθηκε από διάφορους μαθηματικούς οι οποίοι έκαναν μία προσπάθεια ώστε να μπορέσουν να την υπολογίσουν πιο εύκολα. Σε αυτή την παράγραφο θα παρουσιάσουμε κάποια από αυτά τα αποτελέσματα. Για τον αναγνώστη που ενδιαφέρεται και θέλει να μελετήσει και τις αποδείξεις, τον παραπέμπουμε στο [**6**].

Οι πρώτες ιδέες και το σκεπτικό τότε ήταν μήπως υπάρχει κάποια σχέση μεταξύ των $perA$ και $\det A$. Αναζητώντας μία τέτοια σχέση έφτασαν σε διάφορα αποτελέσματα ένα από αυτά είναι τα παρακάτω θεωρήματα.

Θεώρημα 2.1 (Cayley)
Έστω $A = [\alpha_{ij}]$ να είναι ένας 3×3 πίνακας με $\alpha_{ij} \neq 0$ για κάθε $i, j \in \{1,2,3\}\}$ και έστω ο 3×3 πίνακας $A^{(-1)} = \left[\frac{1}{a_{ij}}\right] = [a_{ij}^{-1}]$. Τότε

$$per(A)\det(A) = \det\left(A^{(2)}\right) + 2\left(\prod_{i,j} a_{ij}\right)\det\left(A^{(-1)}\right)$$

Απόδειξη:
Όπως είπαμε και στην προηγούμενη παράγραφο η permanent είναι το άθροισμα των άρτιων μεταθέσεων συν το άθροισμα των περιττών ενώ η ορίζουσα, λόγω του πρόσημου μετάθεσης, είναι η διαφορά των άρτιων μεταθέσεων από τις περιττές μεταθέσεις.

Οπότε αν πούμε Ε το σύνολο των άρτιων μεταθέσεων του S_n και F το σύνολο των περιττών μεταθέσεων αντίστοιχα έχουμε για έναν πίνακα Α $n \times n$

$$\det A = \sum_{\sigma \in E}\prod_{i=1}^{n} a_{\sigma(i)i} - \sum_{\sigma \in F}\prod_{i=1}^{n} a_{\sigma(i)i}$$

$$perA = \sum_{\sigma \in E}\prod_{i=1}^{n} a_{\sigma(i)i} + \sum_{\sigma \in F}\prod_{i=1}^{n} a_{\sigma(i)i}$$

$$per(A)\det(A) = \left(\sum_{\sigma \in E}\prod_{i=1}^{n} a_{\sigma(i)i}\right)^2 - \left(\sum_{\sigma \in F}\prod_{i=1}^{n} a_{\sigma(i)i}\right)^2 \qquad (2.\beta)$$

Μπορούμε να γράψουμε το δεύτερο μέλος της παραπάνω σχέσης

$$\sum_{\sigma \in E}\prod_{i=1}^{n} a^2{}_{\sigma(i)i} - \sum_{\sigma \in F}\prod_{i=1}^{n} a^2{}_{\sigma(i)i} + f(A) = \det\left(A^{(2)}\right) + f(A) \qquad (2.\gamma)$$

Όπου $\det\left(A^{(2)}\right) = \left[a_{ij}^2\right]$ και $f(A)$ εκφράζει τους όρους που υπολείπονται. Από τις σχέσεις (2.β), (2.γ) και για $n = 3$ προκύπτει το αποτέλεσμα του θεωρήματος.

■

Μία γενίκευση του παραπάνω έδωσε ο Sir Thomas Muir συνέβαλε σημαντικά στην μελέτη της Permanent αλλά και της ορίζουσας.

Θεώρημα 2.2 (Muir)

Έστω οι $n \times n$ πίνακες $A = [\alpha_{ij}]$ και $X = [x_{ij}]$. Τότε

$$per(A)\det(X) = \sum_{\sigma \in S_n} \varepsilon(\sigma)\det(A * X_\sigma)$$

Όπου X_σ είναι ο πίνακας X του οποίου οι γραμμές ακολουθούν την μετάθεση σ, δηλαδή στη θέση της i-οστής γραμμής βρίσκεται η $\sigma(i)$ γραμμή. Ακόμη ο πολλαπλασιασμός * είναι ο κατά Hadamard πολλαπλασιασμός πινάκων, δηλαδή $A * X = [\alpha_{ij} \cdot x_{ij}]$

Το θεώρημα του Muir δίνει το πιο γενικό αποτέλεσμα σχέση με άλλες προσπάθειες και φαίνεται πως για να δοκιμάσει κάποιος να υπολογίσει το γινόμενο $per(A)\,det(A)$ θα χρειαστεί να υπολογίσει n! το πλήθος ορίζουσες.

Παραθέτουμε ότι ο Schur με την ενασχόληση του με τις Permanents κατάφερε να φθάσει στο παρακάτω κομψό αποτέλεσμα

Θεώρημα 2.3 (Schur): Αν ο A είναι ένας θετικά ημιορισμένος ερμιτιανός πίνακας, τότε

$$\det(A) \leq per(A)$$

Η ισότητα ισχύει στην παραπάνω σχέση αν και μόνο αν ο A είναι διαγώνιος ή έχει μία μηδενική γραμμή.

Ένα εύλογο ερώτημα που έθεσε ο Pólya ήταν

Ερώτημα Pólya: Έστω S ένα σύνολο $n \times n$ πινάκων, τότε υπάρχει κάποιος γραμμικός μετασχηματισμός T στο S τέτοιος ώστε

$$per\big(T(A)\big) = \det(A) \quad ; \quad (2.\delta)$$

Παρατήρηση 2.5: Για $n = 2$ και

$$T\begin{bmatrix}\alpha_{11} & \alpha_{12}\\ \alpha_{21} & \alpha_{22}\end{bmatrix} = \begin{bmatrix}\alpha_{11} & -\alpha_{12}\\ \alpha_{21} & \alpha_{22}\end{bmatrix}$$

Παρατηρούμε ότι για κάθε 2×2 πίνακα A η σχέση (2.δ) στέκει. Για $n = 3$ και πίνακες της μορφής

$$\begin{bmatrix}\alpha_{11} & \alpha_{12} & 0\\ \alpha_{21} & \alpha_{22} & \alpha_{23}\\ \alpha_{31} & \alpha_{32} & \alpha_{33}\end{bmatrix} \quad και \quad T\begin{bmatrix}\alpha_{11} & \alpha_{12} & 0\\ \alpha_{21} & \alpha_{22} & \alpha_{23}\\ \alpha_{31} & \alpha_{32} & \alpha_{33}\end{bmatrix} = \begin{bmatrix}\alpha_{11} & \alpha_{12} & 0\\ -\alpha_{21} & \alpha_{22} & \alpha_{23}\\ \alpha_{31} & -\alpha_{32} & \alpha_{33}\end{bmatrix}$$

Τότε η (2.δ) πάλι στέκει.

Παρ' όλα αυτά ο Pólya έφτασε στο παρακάτω αποτέλεσμα όπως αναφέρεται και στο [**6**].

Θεώρημα 2.4 (Pólya)

Έστω S να είναι το σύνολο όλων των $n \times n$ πινάκων με $n \geq 3$, τότε δεν υπάρχει μετασχηματισμός T που να περιλαμβάνει αλλαγές πρόσημου στοιχείων του πίνακα A ώστε

$$per\big(T(A)\big) = \det(A)$$

Θεώρημα 2.5 (Marcus και Minc)

Έστω S να είναι το σύνολο όλων των $n \times n$ πινάκων με $n \geq 3$, τότε δεν υπάρχει γραμμικός μετασχηματισμός T στο S ώστε

$$per\big(T(A)\big) = \det(A)$$

Το αποτέλεσμα των Marcus και Minc είναι μία γενίκευση του θεωρήματος του Pólya και μάλιστα γίνεται εμφανές ότι η permanent δεν μπορεί να υπολογισθεί εύκολα μέσω της ορίζουσας.

3 Η ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΤΗΣ PERMANENT

Στη θεωρητική πληροφορική ταξινομούμε τα προβλήματα που μπορούν να επιλυθούν με μηχανηστικό τρόπο, ανάλογα με το υπολογιστικό κόστος που καταναλώνουμε ώστε να πάρουμε την λύση. Το υπολογιστικό κόστος συνήθως το μετράμε σε χώρο ή χρόνο που χρειάζεται μία μηχανή Turing για να φτάσει στην λύση. Τα προβλήματα τα ταξινομούμε σε κλάσεις και χρησιμοποιώντας εργαλεία όπως είναι η αναγωγή ενός προβλήματος σε ένα άλλο που ξέρουμε «τι δυσκολίας είναι», καταφέρνουμε να αποφανθούμε και για την «δυσκολία» του πρώτου.

Με βάση τα παραπάνω θα μελετήσουμε σε αυτό το κεφάλαιο σε ποια κλάση προβλημάτων ανήκει η permanent και η ορίζουσα ως προς την κλασική πολυπλοκότητα. Έπειτα θα δούμε την πολυπλοκότητα που ανέπτυξε ο Valiant βασιζόμενος στην permanent η οποία δίνει καλύτερα εργαλεία στην εύρεση υπολογιστικού κόστους για τον υπολογισμό πολυώνυμων πολλών μεταβλητών. Τέλος θα κοιτάξουμε κάποιες προσπάθειες που έγιναν ώστε να υπολογιστεί ένα κάτω φράγμα για το υπολογιστικό κόστος της Permanent. Αν βρεθεί ένα ικανοποιητικό κάτω φράγμα B, το οποίο εξαρτάται από το μέγεθος n της εισόδου B(n), θα είμαστε σε θέση να γνωρίζουμε ότι σε μία γενική περίπτωση, καλύτερα από B(n) δεν μπορούμε να την υπολογίσουμε.

3.1 Πολυώνυμα πολλών μεταβλητών

Ορισμοί 3.1:

Έστω n απροσδιόριστες μεταβλητές $x_1, \dots, x_n$ με συντελεστές από το σώμα K και $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ ένα διάνυσμα μη αρνητικών ακεραίων, δηλαδή $\alpha_i \geq 0 \; και \; \alpha_i \in \mathbb{N}$ για $i = 1, \dots, n$.
Τότε ορίζουμε ως **μονώνυμο των ν μεταβλητών** το γινόμενο

$$\boldsymbol{x^a} = \prod_{i=1}^{n} x_i^{a_i} = x_1^{a_1} \cdot \dots \cdot x_v^{\alpha_n}$$

Ορίζουμε ως **βαθμό μονωνύμου** το άθροισμα $\|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^{n} a_i$

Ένα **πολυώνυμο** είναι ένας πεπερασμένος γραμμικός συνδυασμός μονωνύμων με συντελεστές από το σώμα K

$$p = \sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i^{\boldsymbol{a}_i}, \qquad \lambda_i \in K, \;\; k \in \mathbb{N}, \;\; \boldsymbol{x}_i^{\boldsymbol{a}_i} \; \mu o \nu \acute{\omega} \nu \upsilon \mu o$$

Το πλήθος των συντελεστών των μη μηδενικών p_a είναι πεπερασμένο. Τέλος ορίζουμε σαν **βαθμό πολυωνύμου** τον μέγιστο βαθμό των μονωνύμων με μη μηδενικό συντελεστή.

Έστω X, Y, Z διανυσματικοί χώροι και $f: X \times Y \to Z$ μια συνάρτηση. Η f θα λέγεται **πολυγραμμική** αν ισχύουν τα παρακάτω

$$f(x_1 + x_2, y_1 + y_2) = f(x_1, y_1) + f(x_1, y_2) + f(x_2, y_1) + f(x_2, y_2) \;\; \forall \, x_1, x_2 \in X, \forall \, y_1, y_2 \in Y$$

$$f(\lambda_1 x, \lambda_2 y) = \lambda_1 f(x, \lambda_2 y) = \lambda_2 f(\lambda_1 x, y) = \lambda_1 \lambda_2 f(x, y) \;\; \forall \, \lambda_1, \lambda_2 \in \mathbb{R} \;\; \forall \, x \in X \, \forall \, y \in Y$$

3.2 Κλασική Πολυπλοκότητα

Στην θεωρία υπολογισιμότητας και πολυπλοκότητας η προσοχή μας είναι στραμμένη στα προβλήματα απόφαση, δηλαδή προβλήματα των οποίων η απάντηση είναι ένα «ναι» ή «όχι».

Ένα πρόβλημα απόφασης μπορεί να αντιπροσωπευθεί από μια γλώσσα L, μια σχέση R_L πάνω στις λέξεις του αλφαβήτου και ισχύε ότι για το στιγμιότυπο x του προβήματος, η απάντηση στο πρόβλημα είναι «ναι» αν και μόνο αν $x \in L$ αν και μόνο αν υπάρχει κάποιο y για το οποίο ισχύει $(x, y) \in R_L$.

3.2.1 Η Πολυπλόκότητα της Permanent

Ορισμός 3.2: Η **κλάση πολυωνυμικού χρόνου P** αποτελείται από όλες τις γλώσσες L για τις οποίες η επαλήθευση ότι η συμβολοσειρά x $(x \in \{0,1\}^n)$ ανήκει στην L μπορεί να γίνει από μία ντετερμινιστική μηχανή Turing, σε αριθμό βημάτων που είναι φραγμένος ως προς ένα πολυώνυμο του n (πολυωνυμικά φραγμένο ως προς n).

Η κλάση NP είναι η κλάση προβλημάτων που η λύση τους επαληθεύεται σε πολυωνυνικό αριθμό βημάτων. Πιο αυστηρά ορίζεται ως εξής

Ορισμός 3.3 (Κλάση NP):
Έστω $R \subseteq (\{0,1\}^* \times \{0,1\}^*)$ να είναι μια σχέση μεταξύ των συμβολοσειρών, η οποία είναι decidable σε πολυωνυμικό χρόνο και ισοροπημένη με τον εξής τρόπο

$$\exists\ πολυωνυμικά\ φραγμένη\ συνάρτηση\ t: \mathbb{N} \to \mathbb{N}\ \ ώστε\ \ |y| \leq t(|x|)\ \ \forall\ x, y \in R$$

$$Η\ γλώσσα\ L = \{x \in \{0,1\}^* : \exists\ y\ \ (x, y) \in R\} \in NP$$

και όλες οι γλώσσες στο NP είναι της παραπάνω μορφής.

Το y θα ονομάζεται **μάρτυρας** (**witness**) του γεγονότος ότι το x ανήκει στην L.

Ορισμός 3.4: Η γλώσσα A θα λέγεται **πολυωνυμικού χρόνου (πολλά προς ένα) αναγώγιμη** στη B αν-ν υπάρχει συνάρτηση μεταξύ των συμβολοσειρών $\rho : \{0,1\}^* \to \{0,1\}^*$, η οποία υπολογίζεται σε πολυωνυμικό χρόνο από μία μηχανή Turing, έτσι ώστε $A = \rho^{-1}(B)$

Ορισμός 3.5: Μια γλώσσα A θα λέγεται **δύσκολη ως προς C** (**C-hard**) αν

$$\forall\ B \in C, \qquad B \leq_m^p A$$

Ορισμός 3.6: Η γλώσσα B λέμε ότι είναι **πλήρης ως προς C** (**complete at C**) με την αναγωγή $\leq_m$ αν

$$B \in C\ \ και\ \ B\ is\ C - hard$$

Κάθε πρόβλημα στο NP εξ'ορισμού έχει μία ισορροπημένη πολυωνυμικού χρόνου αποκρίσιμη σχέση R μεταξύ των συμβολοσειρών.

Ως γνωστόν ένα πρόβλημα ένα πρόβλημα μπορεί να επιδέχεται παραπάνω από μία λύσεις. Επίσης όπως αναφέραμε και στο προηγούμενο κεφάλαιο, αν υπολογίσουμε την permanent ενός πίνακα A, τότε ο υπολογισμός αυτός ισοδυναμεί με το να μετρήσουμε τις λύσεις του προβλήματος

«Το διμερες γράφημα με πίνακα γειτνίασης τον A έχει Perfect Matching; »

Έχοντας αυτα στο νου μας μπορούμε για οποιοδήποτε πρόβλημα απόφασης, να ορίσουμε το αντίστοιχο μετρητικό πρόβλημα, αν ένα πρόβλημα απόφασης ονομάζεται «Problem» τότε το αντίστοιχο μετρητικό του το ονομάζουμε με «#Problem». Δοθέντος ενός προβλήματος απόφασης Π, το αντίστοιχο μετρητικό πρόβλημα θέτει ως ερώτημα, πόσες είναι οι δυνατές λύσεις του Π;

Εναλλακτικά σε ένα μετρητικό πρόβλημα μας ενδιαφέρει να μετρήσουμε από το υπολογιστικό δέντρο μιας μηχανής Turing που επιλύει το πρόβλημα, τα μονοπάτια, που απαντούν με «ναι»,

Το αντίστοιχο μετρητικό πρόβλημα σε ένα NP πρόβλημα απόφασης είναι το εξής

$$\text{Δοθέντος } x \text{ πόσα } y \text{ υπάρχουν ώστε } (x, y) \in R;$$

Με άλλα λόγια προσπαθούμε να υπολογίσουμε την συνάρτηση $\varphi : \{0,1\}^* \to \mathbb{N}$ ώστε

$$\varphi(x) = \#R_x := \#\{y : (x, y) \in R\} = |\{y : (x, y) \in R\}|$$

Ορισμός 3.7: Η **κλάση #P** είναι

$$\#P = \{\varphi : \{0,1\}^* \to \mathbb{N} \mid \varphi(x) = \#R_x\}$$

Μια αναγωγή από το μετρητικό πρόβλημα της σχέσης R στο μετρητικό πρόβλημα των συμβολοσειρών της σχέσης S γίνεται από δύο πολυωνυμικού χρόνου υπολογίσιμες συναρτήσεις $\rho : \{0,1\}^* \to \{0,1\}^*$ και $\sigma : \mathbb{N} \to \mathbb{N}$ (φυσικοί αριθμοί κωδικοποιημένοι σε δυαδικό σύστημα) τέτοιο ώστε $\#R_x = \sigma\left(\#S_{\rho(x)}\right)$ για όλα τα x.

Ορισμός 3.8: Μία αναγωγή θα λέγεται **parsimonious** αν-ν η σ είναι η ταυτοτική.

Ορισμός 3.9: Μια μετρητική συνάρτηση $f \in \#P$ λέγεται **#P-complete** αν-ν κάθε συνάρτηση στο #P ανάγεται στην f.

Ο Valiant έδειξε το 1979 στο [**7**] τα παρακάτω.

Θεώρημα 3.1: Το πρόβλημα υπολογισμού της permanent για δυαδικούς πίνακες είναι #P-complete.

Θεώρημα 3.2: Η πολυπλοκότητα υπολογισμού της permanent ενός πίνακα $A \in M_n(\{0,1\})$ είναι NP-hard και είναι τουλάχιστον τόσης δυσκολίας όσο και να μετρήσουμε τον αριθμό των καταστάσεων αποδοχής μιας μη ντετερμινιστικής πολυωνυμικής μηχανής Turing.

Είναι ανοικτό πρόβλημα το εξής

Το μετρητικό πρόβλημα ενός NP-complete προβλήματος είναι #P-complete;

Ο Valiant απέδειξε ότι το μετρητικό πρόβλημα των Perfect Matchings ενός διμερούς γράφου είναι #P-complete [**7**], όπως αναφέρουμε και στο παραπάνω θεώρημα. Αυτό είναι ένα ενδιαφέρον αποτέλεσμα γιατί το πρόβλημα απόφασης αν υπάρχει Perfect Matching ή γνωστό ως και το πρόβλημα του γάμου, επιλύεται σε πολυωνυμικό χρόνο. Συνεπώς το αντίστροφο του παραπάνω ερωτήματος δεν ισχύει, δηλαδή αν ένα πρόβλημα είναι #P-complete, τότε το αντίστοιχο πρόβλημα απόφασης δεν είναι κατ' ανάγκη NP-complete.

Μελετώντας αυτό το πρόβλημα οι Α. Κιαγιάς, Α. Παγουρτζής Κ. Sharna και Σ. Ζάχος όρισαν τις παρακάτω κλάσεις πολυπλοκότητας στα [**8**] και [**9**] αντίστοιχα

Ορισμός 3.10: Η $\boldsymbol{TotP}$ είναι η κλάση που περιέχει όλες τις συναρτήσεις οι οποίες μετράν τον αριθμό των υπολογιστικών μονοπατιών μιας PNTM.

Ορισμός 3.11: Η $\boldsymbol{\#PE}$ (#P "Easy") είναι η κλάση που περιέχει όλες τις συναρτήσεις f του #P τέτοιες ώστε για κάθε είσοδο x η ερώτηση αν "$f(x) > 0$ " είναι αποκρίσιμη σε πολυωνυμικό χρόνο.

Παρατήρηση 3.1: Εξ' ορισμού η $\#PE$ περιέχει το #Perfect Matching και το #DNF-SAT, αλλά δεν περιέχει το SAT εκτός και αν $P = NP$.

Αυτό που πρέπει να ξεκαθαρίσουμε σε αυτό το σημείο είναι ότι τα μετρητικά προβλήματα που έχουν «εύκολο» πρόβλημα απόφασης είναι #P-complete ως προς αναγωγές κατά Cook, αλλα δεν μπορούν να είναι #P-complete με αναγωγές κατά Karp, όπως αναφέρεται και στο [**9**]. Αυτό σημαίνει ότι οι αναγωγές κατά Karp έχουν καλύτερη ευκρίνεια μεταξύ των κλάσεων μετρητικών προβλημάτων σε σχέση με τις αναγωγές κατά Cook. Οι σχέσεις που αποδεικνύονται στο [**9**] για τις κλάσεις που ορίσαμε είναι οι εξείς

1. $FP \subseteq TotP \subseteq \#PE \subseteq \#P$, $\quad$ *οι εγκλεισμοί είναι γνήσιοι εκτός αν* $P = NP$
2. Ακόμη $TotP, \#PE$ *και* $\#P$ δεν είναι Karp ισοδύναμα εκτός αν $P = NP$.
3. $FP^{TotP}[1] = FP^{\#PE}[1] = FP^{\#P}[1]$, *δηλαδή* $TotP, \#PE$ *και* $\#P$ *είναι Cook ισοδύναμες*
4. Τα #Perfect Matching, το #DNF-SAT και το #NONCLIQUES ανήκουν στην TotP
5. Η $TotP$ είναι η κλειστότητα όλων των συναρτήσεων της #PE που ανάγονται στον εαυτό τους

Συμπέρασμα: *Με βάση τα παραπάνω έχουμε ότι και η Permanent ανήκει στην TotP.*

Παρά τη διατύπωση τωνπαραπάνω θεωρημάτων, η περίπτωση του υπολογισμού της permanent mod 2^k, $k \in \mathbb{N}$ υπολογίζεται πολύ πιο εύκολα όπως έχει αποδειχθεί στο [**7**].

Ορισμός 3.12: Μια μη **διφορούμενη μηχανή Turing** (**unambiguous TM**) είναι μια μη ντετερμινιστική μηχανή Turing τέτοια ώστε

$\forall\, input\ \ w = a_1 a_2 \dots a_n\ \ \exists$ *το πολύ μια ακολουθία* $configurations\ c_0 c_1 \dots c_m\ \ ώστε$

i. c_0 *είναι το αρχικό* $configuration$ *της εισόδου* w
ii. *το* c_i *προηγείται του* c_{i+1}
iii. *το* c_m *είναι το* $configuration$ *αποδοχής*

Αν το w γίνεται αποδεκτό από την TM τότε υπάρχει ακριβώς μια κατάσταση αποδοχής.

Ορισμός 3.13: Η **κλάση UP** (**Unambiguous Non-Deterministic Polynomial-time**) περιλαμβάνει όλα τα προβλήματα απόφασης τα οποία επιλύονται από μια μη διφορούμενη

*Σημειώνουμε ότι στην παράγραφο βασικές έννοιες του κεφαλαίου 1, υπάρχουν οι ορισμοί για αρκετές από τις έννοιες και τις κλάσεις που αναφέρουμε σε αυτή την παράγραφο.

μηχανή Turing (Unambiguous TM) σε πολυωνυμικό χρόνο, έχοντας το πολύ ένα μονοπάτι αποδοχής στο δέντρο υπολογισμού.

$$P \subseteq UP \subseteq NP$$

Η κλάση UP είναι κοντά στην P υπό την έννοια ότι αν προγραμματίζαμε μία ντετερμινιστική μηχανή Turing ώστε να ακολουθήσει το configuration οδηγεί στην κατάσταση αποδοχής, τότε το πρόβλημα θα λυνόταν σε πολυωνιμκό χρόνο.

Θεώρημα 3.3: Έστω $n \in \mathbb{N}$ και $K = 2^n$. Η πολυπλοκότητα υπολογισμού της permanent $mod\ K$ του $A \in M_n(\{0,1\})$ είναι UP-hard.

Θεώρημα 3.4: *Έστω* $A \in M_n(\mathbb{Z})$ *και* $\forall\ k \in \mathbb{Z}$ η $perm(A)\ mod\ 2^k$ μπορεί να υπολογιστεί σε

$$\begin{cases} O\left(n^{4k-3}\right) \text{ βήματα}, & \text{αν } k \geq 2 \\ O(n^{2.81}) \text{ βήματα}, & \text{αν } k = 1 \end{cases}$$

και αυτό ισχύει εφόσον $perm(A)\ mod\ 2 = \det(A)\ mod\ 2$.

3.2.2 Η Πολυπλοκότητα της Ορίζουσας

Αντίστοιχα με τους παραπάνω ορισμούς ορίζονται κλάσεις πολυπλοκότητας οι οποίες μετράνε το υπολογιστικό κόστος ως προς τον χώρο, δηλαδή το πλήθος των κελιών της ταινίας μιας μηχανής Turing στα οποία γράφει για να φτάσει στη λύση του προβλήματος.

Ορισμός 3.14: Η **κλάση λογαριθμικού χώρου L** αποτελείται από όλες τις γλώσσες G για τις οποίες η επαλήθευση ότι η συμβολοσειρά x ανήκει στην G μπορεί να γίνει από μία ντετερμινιστική μηχανή Turing, χρησιμοποιώντας επιπλέον αριθμό κελιών που είναι φραγμένος ως προς ένα λογάριθμο του n (λογαριθμικά φραγμένο ως προς n).

Ορισμός 3.15 (Κλάση NL):

Έστω $R \subseteq (\{0,1\}^* \times \{0,1\}^*)$ να είναι μια σχέση μεταξύ των συμβολοσειρών, η οποία είναι decidable σε λογαριθμικό χώρο και ισορροπημένη με τον εξής τρόπο

$$\exists\ \textit{λογαριθμικά φραγμένη συνάρτηση}\ t: \mathbb{N} \to \mathbb{N}\ \textit{ώστε}\ \ |y| \leq t(|x|)\ \ \forall\ x, y \in R$$

$$H\ \gamma\lambda\acute{\omega}\sigma\sigma\alpha\ G = \{x \in \{0,1\}^* : \exists\ y\ \ (x,y) \in R\} \in NL$$

και όλες οι γλώσσες στην NL είναι της παραπάνω μορφής.

Ορισμός 3.16: Η **κλάση #L** είναι

$$\#L = \left\{\varphi : \{0,1\}^* \to \mathbb{N} \mid G \in NL\ \ και\ \ \varphi(x) = \#R_{G_x}\ \right\}$$

Όπου με R_{G_x} συμβολίζουμε την σχέση της γλώσσας G για το στοιχείο x και όπως έχουμε ορίσει και παραπάνω

$$\#R_{G_x} := \#\{y : (x,y) \in R_G\} = |\{y : (x,y) \in R_G\}|$$

Τα παρακάτω αποτελέσματα ο αναγνώστης θα τα βρει με τη μορφή θεωρημάτων και προτάσεων στο [**10**]. Εμείς τα παραθέτουμε συμπτυγμένα στο εξής θεώρημα

Θεώρημα 3.5: Ισχύουν τα παρακάτω

1. $\#L \subseteq FP \subseteq \#P$
2. $\#L \subseteq NC^2$
3. $\#P = \#NL$
4. $NL = UP$

Ορισμός 3.17: **GapL** είναι η κλάση πολυπλοκότητας που αποτελείται από όλες τις συναρτήσεις f για τις οποίες υπάρχει μία μη ντετερμινιστική μηχανή Turing M που τις υπολογίζει σε επιπλέον λογαριθμικό χώρο και για κάθε είσοδο x, το $f(x)$ είναι η διαφορά των μονοπατιών για τα οποία η M αποδέχεται μείον τα μονοπάτια για τα οποία η M απορρίπτει την είσοδο. Η κλάση GapL είναι η κλειστότητα της #L ως προς την αφαίρεση.

Παρατηρούμε ότι $\#L \subseteq GapL$. Τέλος στο [**11**] αποδεικνύεται το παρακάτω βασικό αποτέλεσμα για την υπολογιστική δυσκολία της ορίζουσας.

Θεώρημα 3.6: Μία συνάρτηση $f \in GapL$ αν-ν είναι logspace πολλά προς ένα αναγώγιμη στην ορίζουσα ενός πίνακα που τα στοιχεία του είναι ακέραιοι.

Είναι άμεσο από το παραπάνω θεώρημα ότι η ορίζουσα ενός πίνακα ακεραίων είναι GapL-complete.

Πόρισμα 3.7: Η ορίζουσα είναι GapL-complete και υπολογίζεται παράλληλα από έναν αλγόριθμο πολυωνυμικού χρόνου. Άρα με βάση το θεώρημα **3.6** κάθε πρόβλημα της GapL μπορεί να υπολογισθεί σε πολυωνυμικό χρόνο. Έτσι φτάνουμε στο συμπέρασμα ότι $GapL \subseteq FP$.

Με βάση το πόρισμα **3.7** και θεώρημα **3.5** προκύπτει ότι $\#L \subseteq GapL \subseteq FP$

Ανάλογα αποτελέσματα ισχύουν για την permanent και την ορίζουσα ως προς κάποιες κλάσεις πολυπλοκότητας που έχει ορίσει ο Valiant, όπως θα δούμε παρακάτω.

3.3 Η Πολυπλοκότητα του Valiant

Σε αυτή την παράγραφο θα μελετήσουμε την πολυπλοκότητα που όρισε ο Valiant με βάση την permanent, η οποία προσφέρει καλύτερη μελέτη, όταν διερευνούμε συναρτήσεις που υπολογίζουν οικογένειες πολυωνύμων πολλών μεταβλητών. Παραθέτουμε κάποιους βασικούς ορισμούς και βασικά αποτελέσματα από το [**12**].

Συμβολισμός: Θα συμβολίζουμε με K ένα σώμα και $K[X_1, X_2, \dots]$ απροσδιόριστες μεταβλητές που πέρνουν τιμές στο K.

Ορισμοί 3.18:

1) Ένα **straight-line program** Γ με m αντικείμενα στην είσοδο θα λέμε μία ακολουθία $(\Gamma_1, \dots, \Gamma_r)$ με εντολές $\Gamma_\rho = (\omega_\rho; i_\rho, j_\rho)$ όπου $\omega_\rho \in \{+, -, *\}$ και διευθύνσεις i_ρ, j_ρ τα οποία ικανοποιούν την ανισότητα $-m < i_\rho, j_\rho < \rho$. Θα λέμε ότι r είναι το **μέγεθος** του Γ.
2) Ένα **straight-line program** Γ ορίζει ένα άκυκλο πολυγράφημα το οποίο έχει σύνολο κορυφών το $\{\rho \in \mathbb{Z} : -m < \rho \leq r\}$ και έχει ακμές (i_ρ, j_ρ). Θα λέμε ότι **μία ακμή είναι πολλαπλότητας 2** αν $i_\rho = j_\rho$ και ο αριθμός των ακμών του μεγαλύτερου κατευθηνώμενου μονοπατιού αποτελεί το **βάθος του Γ**.
3) Για κάθε ακολουθία στοιχείων εισόδου $\alpha_1, \dots, \alpha_m$ ένα straight line program Γ έχει μια μοναδική ακολουθία αποτελεσμάτων $(b_{-m+1}, \dots, b_r)$ με

$$\begin{cases} b_\rho = a_{m+\rho} \quad , & \rho \leq 0 \\ b_\rho = b_{i_\rho}\ \omega_\rho\ b_{j_\rho}\ , & \rho > 0 \end{cases}$$

Λέμε ότι **το Γ υπολογίζει ένα σύνολο πολυωνύμων F** ανν $F \subseteq \{b_{-m+1}, \dots, b_r\}$

Ένα αντίστοιχο εργαλείο των straight-line προγραμμάτων, που μας επιτρέπει να υπολογίσουμε την υπολογιστική δυσκολία πολυωνύμων είναι τα αριθμητικά κυκλώματα. Τα αριθμητικά κυκλώματα έχουν χρησιμοποιηθεί για την εύρεση κάτω φράγματος στο υπολογιστικό κόστος πολυωνυμικών οικογενειών αναφέρουμε το Manidra Agrawal et. al. [**13**]

<u>Ορισμοί 3.19:</u>

- Ένα **αριθμητικό κύκλωμα (arithmetic circuit)** πάνω σε ένα σώμα K είναι ένας κατευθυνόμενος άκυκλος γράφος με κορυφές που διαθέτουν μια επισήμανση.
- Οι κορυφές του γράφου με indegree μηδέν επισημαίνονται είτε με μία μεταβλητή $x_i \in K$ ή με μία σταθερά $c \in K$. Οι κορυφές που επισημαίνονται με μεταβλητές ονομάζονται **θύρες εισόδου (input gates)**.
- Οι υπόλοιπές κορυφές επισημαίνονται με $'+'$ ή $'*'$ και ονομάζονται **θύρες πρόσθεσης (άθροισης)** και **θύρες πολλαπλασιασμού** αντίστοιχα (**addition or multiplication gates**).
- Οι θύρες με outdegree μηδέν ονομάζονται **θύρες εξόδου**. Εμείς θα ασχοληθούμε κυρίως με κυκλώματα που έχουν ακριβώς μία θύρα εξόδου.

Ο αριθμός των ακμών που καταλήγουν σε μία θύρα είναι το fanin της θύρας. Θα θεωρούμε πως μία θύρα μπορεί να έχει άπειρο fanin, σε περίπτωση που δεν υπάρχει κάποιος περιορισμός που να το απαγορεύει.

- Το **μέγεθος (size) ενός κυκλώματος C** είναι ο αριθμός των θυρών που περιέχει.
- Το **βάθος (depth) ενός κυκλώματος C** είναι το μήκος του μεγαλύτερου μονοπατιού από μία θύρα εισόδου σε μία θύρα εξόδου.
- Ο **βαθμός (degree) ενός κυκλώματος C** ορίζεται αναδρομικά ως εξής:
    - Ο βαθμός μίας θύρας εισόδου είναι 1 ή 0 ανάλογα με το αν έχει την επισήμανση μεταβλητής ή σταθεράς αντίστοιχα.
    - Ο βαθμός μιας θύρας άθροισης είναι ο μέγιστος βαθμός των θυρών που οι ακμές εμπίπτουν στην εν λόγω θύρα.
    - Ο βαθμός μιας πολλαπλασιαστικής θύρας είναι το άθροισμα των βαθμών των θυρών, που οι ακμές τους εμπίπτουν στην εν λόγω θύρα.

- Ο βαθμός του C είναι ο βαθμός της θύρας εξόδου.

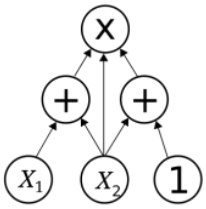
Ένα αριθμητικό κύκλωμα υπολογίζει πολυώνυμα με τον εξής τρόπο. Οι θύρες εισόδου επισημαίνονται με σταθερές και με τις μεταβλητές από τις οποίες αποτελούνται όλα τα μονώνυμα του πολυωνύμου . Για οποιαδήποτε άλλη θύρα g, θεωρούμε $g_1, \dots, g_k$ να είναι οι θύρες που έχουν μια ακμή που καταλήγει στην g. Ακόμη με p_{g_i} συμβολίζουμε το πολυώνυμο που έχει υπολογισθεί στην θύρα g_i $(i = 1, \dots, k)$. Τότε

$$p_g = \begin{cases} \sum_{i=1}^{k} p_{g_i}, & \text{αν } g \text{ αθροιστική θύρα} \\ \prod_{i=1}^{k} p_{g_i}, & \text{αν } g \text{ πολλαπλασιαστική θύρα} \end{cases}$$

Το πολυώνυμο που υπολογίζεται από το κύκλωμα είναι το πολυώνυμο που προκύπτει από την θύρα εξόδου.

Παράδειγμα 3.1:

Το παρακάτω αριθμητικό κύκλωμα είναι βαθμού 3 και υπολογίζει το $(x_1 + x_2)x_2(x_2 + 1)$



Έστω $\{p_n\}_{n>0}$ να είναι μια οικογένεια πολυωνύμων και το p_n να είναι ένα πολυώνυμο n μεταβλητών και βαθμού n. Μια **οικογένεια αριθμητικών κυκλωμάτων $\{C_n\}_{n>0}$ λέμε ότι υπολογίζει την $\{p_n\}_{n>0}$** αν για κάθε n, το πολυώνυμο που που υπολογίζει το C_n είναι ίσο με το p_n.

Συμβολισμός: *Χάριν απλότητας θα συμβολίζουμε την οικογένεια $\{p_n\}_{n>0}$ με $\{p_n\}$.*

Αριθμητικά προγράμματα διακλάδωσης είναι μορφές περιορισμένων αριθμητικών κυκλωμάτων στα οποία κάθε πολλαπλασιατική θύρα έχει ακριβώς δύο μεταβλητές εισόδου. Επιπλέον τουλάχιστον μία από τις δύο θύρες των οποίων οι ακμές φτάνουν στην πολλαπλασιατική θύρα, πρέπει να είναι θύρα εισόδου. Αυτά τα κυκλώματα είναι γνωστά και ως **skew circuits (λοξά κυκλώματα)**.

Συνήθως η ακολουθία των στοιχείων εισόδου είναι της μορφής $c_1, \dots, c_s, X_1, \dots, X_n$, με σταθερές $c_i \in K$ και X_j απροσδιόριστες μεταβλητές.

Ορισμός 3.20: Η **πολυπλοκότητα $L(F)$** ενός συνόλου πολυωνύμων $F \subseteq K[X_1, \dots, X_n]$ είναι το ελάχιστο μέγεθος ενός straight-line program που υπολογίζει το F για μεταβλητές X_i και σταθερές στο K.

Ορισμός 3.21: Θα ονομάζουμε μία συνάρτηση $f : \mathbb{N} \to \mathbb{N}$ **p-φραγμένη** (**p-bounded**) ανν

$$\exists\, c > 0 \ \text{τέτοιο ώστε} \ f(n) \leq n^c + c\,, \qquad \forall\, n \in \mathbb{N}$$

Επιπλέον αν

$$\exists\, c > 0 \;\; \tau\acute{\epsilon}\tau o\iota o \;\; \acute{\omega}\sigma\tau\epsilon \quad n^{1/c} - c \le f(n) \le n^c + c\,, \qquad \forall\, n \in \mathbb{N}$$

Θα λέμε ότι η f είναι **άνω και κάτω p-φραγμένη**.

Ορισμός 3.22: Μία ακολουθία $f = (f_n)$ πολυωνύμων πολλών μεταβλητών πάνω στο K θα λέγεται **p-family** πάνω στο K ανν ο αριθμός των μεταβλητών καθώς και ο βαθμός του πολυωνύμου f_n είναι p-φραγμένες συναρτήσεις του n.

Παρατηρούμε κάτι δεν είναι p-family αν ο βαθμούς του πολυωνύμου αυξάνεται εκθετικά ως προς το n για παράδειγμα.

Παράδειγμα 3.2: Ένα χαρακτηριστικό παράδειγμα μιας τέτοιας οικογένειας είναι η οικογένεια πολυωνύμων που υπολογίζει τη συνάρτηση permanent (ή και της ορίζουσας) η οποία έχει ως όρισμα έναν πίνακα A $n \times n$. Τις οικογένειες πολυωνύμων, με στοιχεία από το σώμα F, της permanent και της ορίζουσας αντίστοιχα, τις συμβολίζουμε με $per_F = \{per_{F,n}\}$ και $\det_F = \{\det_{F,n}\}$. Και οι δύο οικογένειες έχουν πολυώνυμα, βαθμού n, που αποτελούνται από n^2 το πλήθος μεταβλητές για κάθε n.

Ορισμοί 3.23:

Μια p-family είναι **p-υπολογίσιμη** (**p-computable**) αν-ν η πολυπλοκότητα $L(f_n)$ είναι p-φραγμένη συνάρτηση του n.

Η **κλάση πολυπλοκότητας VP ή VP_K** αποτελείται από όλες τις p-υπολογίσιμες οικογένειες πάνω στο K.

Η οικογένεια πολυωνύμων της ορίζουσας $(DET_n)_n$ είναι ένα χαρακτηριστικό παράδειγμα μιας οικογένειας που ανήκει στην VP. Όπου DET_n είναι η ορίζουσα ενός $n \times n$ πίνακα.

Ορισμός 3.24: Μια p-οικογένεια (f_n) θα λέγεται **p-προσδιορίσιμη** (p-definable) ανν ∃ p-υπολογίσιμη συνάρτηση $g = (g_n)$, $g_n \in K[X_1, \dots, X_{u(n)}]$, τέτοια ώστε $\forall\, n \in \mathbb{N}$

$$f_n(X_1, \dots, X_{v(n)}) = \sum_{e_{v(n)+1}=0}^{1} \sum_{e_{v(n)+2}=0}^{1} \dots \sum_{e_{u(n)}=0}^{1} g_n(X_1, \dots, X_{v(n)}, e_{v(n)+1}, \dots, e_{u(n)})$$

Ορισμός 3.25: Το σύνολο των p-προσδιορίσιμων οικογενειών αποτελεί την **κλάση πολυπλοκότητας $VNP = VNP_K$**.

Παρατήρηση 3.2: Η f_n υπολογίζεται με την άθροιση εκθετικά πολλών g_n, συνεπώς η πολυπλοκότητα της f_n μπορεί να είναι εκθετική ως προς n. Επίσης είναι άμεσο ότι η $VP \subseteq VNP$.

Ένα παράδειγμα p-προσδιορίσιμης οικογένειας είναι η permanent. Η permanent $per(A)$ ενός $n \times n$ πίνακα A ορίζεται ως

$$per(A) = \sum_{\pi \in S_n} \prod_{i=1}^{n} A_{i,\pi(i)}$$

Θα θεωρούμε ότι ένας πίνακας $n \times n$ αποτελείται από X_{n^2} απροσδιόριστες μεταβλητές και θα συμβολίζουμε με (PER_n) την οικογένεια πολυωνύμων της permanent για $n \times n$ πίνακες.

Λήμμα 3.1: Η οικογένεια πολυωνύμων $PER = (PER_n)$ είναι p-προσδιορίσιμη.

Απόδειξη:

Έστω $X = (x_{ij})$ και $Y = (y_{ij})$ να είναι δύο $n \times n$ πίνακες με στοιχεία απροσδιόριστες μεταβλητές..

Ορίζουμε το πολυώνυμο g_n ως εξής

$$g_n := \underbrace{\underbrace{\left(\prod_{i,j,l,m} (1 - y_{ij}\, y_{lm})\right)}_{a_n(Y)} \cdot \underbrace{\left(\prod_{i=1}^{n} \sum_{j=1}^{n} y_{ij}\right)}_{\beta_n(Y)}}_{\gamma_n(Y)} \cdot \underbrace{\left(\prod_{i=1}^{n} \sum_{j=1}^{n} x_{ij}\, y_{ij}\right)}_{\mu_n(X,Y)}$$

Όπου το γινόμενο στον όρο a_n είναι πάνω σε όλα τα $1 \le i, j, l, m \le n$ τέτοια ώστε $i = l$ αν-ν $j \neq m$.

Άρα η g_n είναι μία p-οικογένεια με $L(g_n) = O(n^3)$ συμπεραίνουμε ότι είναι p-υπολογίσιμη.

Ισχυριζόμαστε ότι για κάθε πίνακα $e \in M_n(\{0,1\})$, η ποσότητα $\gamma_n(e)$ είναι μη μηδενική αν-ν ο e είναι πίνακας μεταθέσεων.

Πράγματι, $a_n(e) \neq 0$ αν-ν κάθε γραμμή και κάθε στήλη του e περιέχουν το πολύ ένα «1». Ακόμη αν $\alpha_n(e) \neq 0$, τότε το $\beta_n(e) \neq 0$ αν-ν κάθε γραμμή του e περιέχει τουλάχιστον ένα «1». Συνεπώς το $\gamma_n(e) = a_n(e) \cdot \beta_n(e) \neq 0$ αν-ν ο e είναι πίνακας μεταθέσεων.

Στην περίπτωση λοιπόν που ο e είναι ο πίνακας της μετάθεσης σ έχουμε ότι

$$\mu_n(X, e) = \prod_{i=1}^{n} x_{i,\sigma(i)}$$

Συνεπώς η $PER_n = \sum_{e \in M_n(\{0,1\})} g_n(X, e)$, πράγμα που σημαίνει ότι η PER είναι p-προσδιορίσιμη.

■

Μια οικογένεια πολυωνύμων που σχετίζεται με την PER είναι η $HC = (HC_n)$, η οικογένεια πολυωνύμων κύκλων Hamilton που ορίζεται ως

$$HC_n = \sum_{\pi} \prod_{i=1}^{n} X_{i,\pi(i)}$$

Όπου το άθροισμα τρέχει σε όλους τους κύκλους μεγέθους n και άρα $\pi \in S_n$

Το αντίστοιχο της αναγωγής για την πολυπλοκότητα που όρισε ο Valiant είναι η προβολή πολυωνύμων η οποία ορίζεται παρακάτω.

Ορισμός 3.26:

Ένα πολυώνυμο f θα ονομάζεται **προβολή** του πολυωνύμου g, συμβολισμός $f \leq g$, ανν

$$f(X_1, \dots, X_n) = g(a_1, \dots, a_m), \qquad \gamma\iota\alpha\ \kappa\acute{\alpha}\pi o\iota\alpha\ \alpha_i \in K \cup \{X_1, \dots, X_n\}, (m \geq n)$$

Δηλαδή η f προκύπτει από την g αν κάποιες τις μετονομάσουμε ή τις αντικαταστήσουμε με κάποιες μεταβλητές.

Παρατήρηση 3.3: Η προβολή πολυωνύμων όπως την ορίσαμε παραπάνω αποτελεί σχέση μερικής διάταξης

Ορισμός 3.27:

Θα λέμε ότι η p-οικογένεια $f = (f_n)$ είναι **p-προβολή** (**p-projection**) της $g = (g_m)$, συμβολισμός $f \leq_p g$, αν-ν $\exists$ άνω και κάτω p-φραγμένη συνάρτηση $t : \mathbb{N} \to \mathbb{N}$ τέτοια ώστε

$$\exists\, n_0 \in \mathbb{N}\ \ \omega\sigma\tau\varepsilon\ \forall\, n \geq n_0\ : f_n \leq g_{t(n)}$$

Παρατήρηση 3.4: Ισχύει η μεταβατική ιδιότητα για την σχέση p-προβολή πάνω στο σύνολο των p-οικογενειών.

Ο ορισμός αυτός διαφέρει από τον αρχικό ορισμό που έδωσε ο Valiant. Επειδή θέλουμε η p-προβολή να σέβεται την μεταβατική ιδιότητα και επειδή ταυτοχρόνως μπορεί μια συνάρτηση $f_n \to \infty$ καθώς $n \to \infty$, χρησιμοποιούμε μία συνάρτηση t η οποία να είναι τουλάχιστον πολυωνυμική ως προς το n.

3.3.1 VNP-Πληρότητα

Ορισμός 3.28: Μια p-προσδιορίσιμη οικογένεια g θα λέγεται ότι είναι **VNP-πλήρης** (**VNP-complete**) ανν

$$\forall\ f \in VNP, \qquad f \leq_p g$$

Ο Ισχυρισμός του Valiant: Ο Valiant ισχυρίζεται ότι $VP \neq VNP$, το οποίο μένει να αποδειχθεί.

Είναι ενδιαφέρον όπως αποδεικνύεται και στο [**12**]*, ότι έχει αποδειχθεί πως αν ισχύει ότι* $P \neq NP$ *όπως έχει ορισθεί από τον Cook τότε και* $VP \neq VNP$.

Ορισμός 3.29: Η **χαρακτηριστική** ενός δακτυλίου R, συμβολισμός $char(R)$, είναι ο ελάχιστος αριθμός, συν ένα, που πρέπει να προσθέσουμε το μοναδιαίο στοιχείο του πολλαπλασιασμού στον εαυτό του ώστε το άθροισμα να γίνει ίσο με το μηδενικό στοιχείο της πρόσθεσης. Αν το παραπάνω δεν συμβαίνει ποτέ τότε θεωρούμε $char(R) = 0$.

Θεώρημα 3.8 (Valiant): Η οικογένεια HC είναι VNP-complete, όπως και η PER_F στην περίπτωση που $char(F) \neq 2$.

Παρατήρηση 3.5: Η υπόθεση του Valiant είναι αληθής <u>ανν</u> η Permanent δεν είναι p-υπολογίσιμη. Το ίδιο ισχύει και για οποιαδήποτε άλλη VNP-πλήρη οικογένεια.

Ορισμός 3.30: Θα ονομάζουμε μία p-οικογένεια **μονότονη** (**monotone**) ανν

$$f_n \quad \textit{είναι p-προβολή του} \quad f_{n+1}, \quad \forall\, n \in \mathbb{N}$$

Λήμμα 3.2: Έστω $f = (f_n)$ και $g = (g_n)$ δύο VNP-πλήρεις οικογένειες, η f είναι μονότονη, και $I \subseteq \mathbb{N}$. Ορίζουμε την οικογένεια $h = (h_n)$ με τον εξής τρόπο

$$h_n = \begin{cases} f_n\ , & \alpha\nu\ n \in I \\ g_n\ , & \alpha\nu\ n \notin I \end{cases}$$

Τότε και η h είναι VNP-πλήρης.

Proof Peter Burgisser p.15 24pdf

Παρατήρηση 3.6: Η υπόθεση ότι η (f_n) είναι μονότονη είναι απαραίτητη, διότι αν η f είναι VNP-πλήρης οικογένεια τότε και οι οικογένειες $(f_1, 0, f_2, 0, \dots)$ $\kappa\alpha\iota$ $(0, f_1, 0, f_2, \dots)$ είναι και αυτές πλήρεις, όμως η ανάμειξη τους με $I = \{2, 4, 6, 8, \dots\}$ μας οδηγεί στην μηδενική οικογένεια.

Ο Valiant απέδειξε τα παρακάτω [**12**]

Πρόταση 3.1: Για κάθε πολυώνυμο $f \in K[X_1, \dots, X_n]$ που υπολογίζεται από ένα straight-line πρόγραμμα μεγέθους e, υπάρχει ένας τετραγωνικός πίνακας A μεγέθους $e + 1$ πάνω στο σώμα $K[X_1, \dots, X_n]$ τέτοιο ώστε $f = per(A)$. Ακόμα μπορούμε να υποθέσουμε ότι ο πίνακας A έχει το πολύ $e + 1$ στοιχεία διάφορα του 1.

Θεώρημα 3.9:

Για κάθε σώμα F, $per_F \in VNP_F$. Ισχύει ακόμη ότι για κάθε F, $char(F) \neq 2$,

$$\forall\ \textit{πολυωνυμική οικογένεια}\ \{p_n\}\ \textit{στο}\ VNP_F, \{p_n\}\ \textit{είναι p-projection της}\ \ per_F.$$

Θεώρημα 3.10:

Για κάθε σώμα F, τα πολυώνυμα της ορίζουσας στο det_F μπορούν να υπολογιστούν από αριθμητικά προγράμματα διακλάδωσης πολυωνυμικού μεγέθους.

Επιπλέον για κάθε σώμα F και κάθε πολυωνυμική οικογένεια $\{p_n\}$ που υπολογίζεται από πολυωνυμικού μεγέθους αριθμητικά προγράμματα διακλάδωσης ισχύει ότι το $\{p_n\}$ είναι p-projection του det_F.

Στην πραγματικότητα όλες οι οικογένειες πολυωνύμων στην VP είναι κατά προσέγγιση p-projections της ορίζουσας.

Θεώρημα 3.11:

Έστω C ένα κύκλωμα μεγέθους s που υπολογίζει πολυώνυμα βαθμού d. Τότε υπάρχει ένα άλλο κύκλωμα μεγέθους $s^{O(1)}$ και βάθους $O(\log s + \log d)$ που υπολογίζει τα ίδια πολυώνυμα.

3.3.2 VQP-Πληρότητα

Στην προηγούμενη ενότητα δείξαμε ότι η οικογένεια $\{per_{F,n}\}$ είναι VNP-πλήρης. Τώρα θα παρουσιάσουμε μία άλλη κλάση πολυπλοκότητας την VQP και θα δείξουμε ότι και $\{det_{F,n}\}$ είναι VQP-πλήρης. Οι ορισμοί και οι αποδείξεις των παρακάτω αποτελεσμάτων υπάρχουν στο κεφάλαιο 2 του [**12**].

Ορισμός 3.31:

Θα ονομάζουμε μία συνάρτηση $f : \mathbb{N} \to \mathbb{N}$ ***qp*-φραγμένη** (**quasi-polynomially bounded**) αν-ν

$$\exists\, c > 0 \;\; \tau\acute{\epsilon}\tau o\iota o \;\; \acute{\omega}\sigma\tau\epsilon \;\; f(n) \leq log^{c}(n)\,, \qquad \forall\, n \in \mathbb{N}$$

Ορισμοί 3.32:

Μια p-family $f = (f_n)$ είναι ***qp*-υπολογίσιμη** (***qp*-computable**) αν-ν η πολυπλοκότητα $L(f_n)$ είναι qp-φραγμένη συνάρτηση του n.

Η **κλάση πολυπλοκότητας VQP ή VQP_K** αποτελείται από όλες τις qp-υπολογίσιμες οικογένειες πάνω στο K.

Παρατήρηση 3.7: Είναι εμφανές από τον παραπάνω ορισμό ότι $VP \subseteq VQP$.

Θεώρημα 3.12: Η οικογένεια πολυωνύμων DET είναι VQP-πλήρης ως προς τις qp-προβολές.

Πόρισμα 3.13: Κάθε οικογένεια κυκλωμάτων στην VP_F μπορεί να υπολογισθεί από μία οικογένεια κυκλωμάτων πολυωνυμικού μεγέθους και λογαριθμικού βάθους.

Πόρισμα 3.14:

$$\forall\, n \in \mathbb{N} \;\; \kappa\alpha\iota \;\; \{p_n\} \in VP_F \implies p_n \;\; \epsilon\acute{\iota}\nu\alpha\iota \; projection \; \tau\eta\varsigma \;\; det_{F,m}, \;\; \acute{o}\pi o\upsilon \;\; m = n^{O(\log n)}$$

Με βάση όλα τα παραπάνω ένας τρόπος να διαχωρίσουμε την κλάση VP_F από την VNP_F, αρκεί να δείξουμε ότι η per_F δεν είναι σχεδόν p-projection της det_F.

Για τον αναγνώστη που ενδιαφέρεται να δει την σύνδεση των κλάσεων πολυπλοκότητας του Valiant με τις κλάσεις της κλασικής πολυπλοκότητας και με το BSS model τότε προτείνουμε το [**12**].

3.4 Γνωστά Κάτω Φράγματα για την Permanent

Όπως έχουμε αναφέρει και παραπάνω είναι σημαντική η εύρεση κάτω φραγμάτων για τον υπολογισμό της permanent. Μία ιδέα για την εύρεση κάτω φραγμάτων για τον υπολογισμό της permanent είναι χρησιμοποιήσουμε περιορισμένα κυκλώματα (restricted circuits). Βασικοί περιορισμοί που έχουν χρησιμοποιηθεί είναι οι εξής;

1. μονότονα κυκλώματα (monotone circuits)
2. κυκλώματα σταθερού βάθους (constant depth circuits)
3. πολλυγραμμικές εξισώσεις (multilinear formulas).

3.4.1 Μονότονα Κυκλώματα

Ορισμός 3.33:

Ένα κύκλωμα πάνω στο $\mathbb{Q}$ ή στο $\mathbb{R}$ ονομάζεται **μονότονο** (**monotone circuits**) αν όλες οι σταθερές στο κύκλωμα είναι μη αρνητικές.

Έχει αποδειχθεί από τους Jerrum και Snir [**14**] ότι κάθε μονότονο κύκλωμα που υπολογίζει την permanent είναι εκθετικού μεγέθους.

3.4.2 Κυκλώματα Σταθερού Βάθους

Ορισμός 3.34: Τα **κυκλώματα σταθερού βάθους** (**constant depth circuits**) όπως φανερώνει και η ονομασία τους είναι όλα τα κυκλώματα των οποίων το βάθος είναι μια σταθερά.

Η Permanent (ή και οποιοδήποτε άλλο πολυώνυμο βαθμού $n^{O(1)}$) μπορεί να υπολογιστεί από μία εκθετικού μεγέθους, βάθους 2, οικογένεια κυκλωμάτων και είναι σχετικά εύκολο να δούμε ότι κάθε οικογένεια κυκλωμάτων βάθους 2 που υπολογίζει την Permanent, πρέπει να είναι εκθετικού μεγέθους.

Αν πάμε στην περίπτωση των οικογενειών αριθμητικών κυκλωμάτων βάθους 3 θα παρατηρήσουμε ότι έχει ένα ενδιαφέρον και δεν είναι τόσο τετριμμένη όσο θα περίμενε κανένας. Αυτό συμβαίνει διότι από αριθμητικά κυκλώματα βάθους 3 μπορούν να δημιουργηθούν

1. «αθροίσματα-γινομένων με όρους αθροίσματα» και αντίστοιχα
2. «γινόμενα-αθροισμάτων με όρους γινόμενα»

Η δεύτερη περίπτωση φαίνεται να χρειάζονται εκθετικού μεγέθους κυκλώματα ώστε να υπολογίζουν την Permanent. Παρ' όλα αυτά μπορεί να αποδειχθεί πως η μία πολλαπλασιαστική πύλη είναι στην ουσία περιττή (άχρηστη) οπότε εκπίπτει να είναι ένα κύκλωμα βάθους 2.

Όσο για την πρώτη περίπτωση, αφορά οικογένειες κυκλωμάτων που είναι πιο ισχυρά και μάλιστα ο Ben Or παρατήρησε πως μπορούν να υπολογίσουν αποδοτικά όλα τα συμμετρικά πολυώνυμα βαθμού $n^{O(1)}$ πάνω σε όλα τα σώματα K για τα οποία ισχύει $char(K) = 0$.

Επίσης για σώματα με χαρακτηριστική μηδέν αναφέρουμε το καλύτερο γνωστό κάτω φράγμα είναι των Shpilka και Wigderson [**15**] οι οποίοι απέδειξαν ότι η permanent, υπολογίζεται από

μια οικογένεια κυκλωμάτων μεγέθους $\Omega(n^2)$. Η ιδέα πίσω από την απόδειξη του παραπάνω αποτελέσματος, είναι να θεωρήσουμε τον χώρο που παράγεται από όλες τις μερικές παραγώγους των πολυωνύμων που υπολογίζονται σε κάθε πύλη για δοθέν αριθμητικό κύκλωμα. Τελικά απέδειξαν ότι τα πολυώνυμα που δημιουργούνται στην θύρα εξόδου, ενός αριθμητικού κυκλώματος, «μικρού μεγέθους» και βάθους 3, παράγουν χώρους μικρής διάστασης. Ενώ ο χώρος που παράγεται από τις μερικές παραγώγους των πολυωνύμων της permanent είναι μεγαλύτερης διάστασης.

Αν δουλεύουμε σε πεπερασμένα σώματα τότε βρισκόμαστε σε καλύτερη θέση. Οι Grigoriev και Razborov [**16**] έδειξαν ένα εκθετικό κάτω φράγμα ως προς το μέγεθος των οικογενειών αριθμητικών κυκλωμάτων βάθους 3, που υπολογίζουν την ορίζουσα και την permanent. Στην πραγματικότητα προσπάθησαν να αποδείξουν ότι πολυώνυμα που υπολογίζονται από αριθμητικά κυκλώματα μικρού μεγέθους και βάθους 3 μπορούν να προσεγγιστούν (υπό την έννοια ότι ταυτίζονται στην πλειοψηφία των σημείων του σώματος) από πολυώνυμα χαμηλής τάξης. Έτσι παρατήρησαν ότι η ορίζουσα και η permanent δεν προσεγγίζονται από πολυώνυμα χαμηλής τάξης.

3.4.3 Πολυγραμμικές Εξίσωσεις

Ορισμός 3.35:

Με τον όρο **πολυγραμμικές εξισώσεις** ονομάζουμε αριθμητικά κυκλώματα για τα οποία

i. Κάθε πύλη έχει βαθμό εξερχόμενων ακμών (outdegree) το πολύ 1
ii. Τα πολυώνυμα που υπολογίζονται σε κάθε πύλη είναι πολυγραμμικές συναρτήσεις.

Παρατηρούμε ότι τέτοια αριθμητικά κυκλώματα έχουν πολύ περιορισμένες πύλες πολλαπλασιασμού πιο συγκεκριμένα τα πολυώνυμα στην είσοδο της πολλαπλασιαστικής πύλης δεν πρέπει να περιέχουν καμία κοινή μεταβλητή.

Χρησιμοποιώντας την τεχνική των μερικών παραγώγων και τυχαίων περιορισμών (θέτουμε κάποιες μεταβλητές εισόδου να έχουν τυχαίες τιμές) ο Raz [**17**] απέδειξε ένα κάτω φράγμα του $n^{\Omega(\log n)}$ ως προς το μέγεθος των οικογενειών πολυγραμμικών εξισώσεων που υπολογίζουν την permanent και της ορίζουσας.

Με την χρήση της αλγεβρικής πολυπλοκότητας, αλλά και με της αλγεβρικής γεωμετρίας γίνονται διάφορες προσπάθειες να αποδειχθεί ότι η Permanent ενος πίνακα Α μπορεί να υπολογιστεί ως μία ορίζουσα, ενός πίνακα του οποίου το μέγεθος είναι εκθετικά μεγαλύτερο σε σχέση με του Α.Μια τελευταία δημοσίευση που παρατηρήσαμε είναι η εξης [**18**].

4 ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΤΗΝ PERMANENT

Σε αυτό το κεφάλαιο, όπως φανερά προδίδει και ο τίτλος, θα παρουσιάσουμε συνοπτικά αλγόριθμους που υπάρχουν για την Permanent.

4.1 Αλγόριθμοι Υπολογισμού της Permanent

Από το προηγούμενο κεφάλαιο γίνεται αντιληπτό ότι η permanent δεν υπολογίζεται σε πολυωνυμικό χρόνο, εκτός αν $P = NP$, και ότι είναι τόσης δυσκολίας όσο και η καταμέτρηση των καταστάσεων αποδοχής σε οποιοδήποτε NP-δύσκολο πρόβλημα. Επειδή όμως η Permanent είναι μία υπολογίσιμη συνάρτηση, υπάρχουν αλγόριθμοι, υψηλού υπολογιστικού κόστους, που υπολογίζουν την τιμή της για οποιοδήποτε πίνακα Α. Στην πράξη οι αλγόριθμοι δουλεύουν για πίνακες μικρών διαστάσεων καθώς το υπολογιστικό κόστος αυξάνεται ραγδαία.

Παραθέτουμε τον τύπο του Ryser [**19**], ο οποίος μελετώντας την Permanent βρήκε το ακόλουθο αποτέλεσμα,

Έστω ένας $n \times n$ πίνακας $A = (a_{ij})$ και A_k να είναι ένας πίνακας που προκύπτει από τον Α, αν διαγράψουμε k στήλες, θέτουμε

$$P(A_k) = \sum_{i=1}^{k} \prod_{j=1}^{n} a_{ji} \quad \kappa\alpha\iota \quad \Sigma_k = \sum_{\ell=1}^{\binom{n}{k}} \left(P(A_k)\right)_{\ell}$$

Όπου το Σ_k είναι το άθροισμα όλων των $P(A_k)$ πάνω σε όλες τις δυνατές διαγραφές k στηλών οι οποίες είναι $\binom{n}{k}$ το πλήθος.

Τελικά έχουμε

$$Per(A) = \sum_{k=0}^{n-1} (-1)^k \Sigma_k$$

Ο υπολογισμός αυτός της τιμής της Permanent γίνεται εκτελώντας $O(2^{n-1}n^2)$ απλές πράξεις (προσθέσεις, γινόμενα και αναθέσεις).

4.2 Προσεγγιστικοί Αλγόριθμοι για την Permanent

Το επόμενο λογικό ερώτημα που μπορούμε να κάνουμε για τον υπολογισμό της Permanent, είναι μήπως μπορούμε να θυσιάσουμε ακρίβεια στη λύση και να υπολογίζουμε προσεγγιστικά την τιμή της, σε πολυωνυμικό χρόνο, με έναν παράγοντα σφάλματος.

Οι Jerum και Sinclair στο [**20**] κάνοντας χρήση Μαρκοβιανών αλυσίδων με Monte Carlo δημιούργησαν ένα FPRAS για τον υπολογισμό της Permanent για πίνακες με στοιχεία από το {0,1}. Μετά από κάποια χρόνια γενίκευσαν το αποτέλεσμα τους για πίνακες με θετικά στοιχεία στο [**21**]. Έκτοτε έχουν παρουσιαστεί διάφορα προσεγγιστικά πιθανοτικά σχήματα πολυωνυμικού χρόνου για την Permanent, ενδεικτικά αναφέρουμε τα πιο πρόσφατα [**22**] και [**23**].

Αυτό που παρουσιάζει ενδιαφέρον είναι ότι το πρόβλημα απόφασης, δηλαδή αν ένας διμερής γράφος έχει Perfect Matching, μπορούμε να το απαντήσουμε μετά από πολυωνυμικό χρόνο.

Για να γίνουμε πιο ακριβείς, ένας αλγόριθμος που χρησιμοποιείται για την εύρεση ενός Perfect Matching σε ένα διμερές γράφο $G = (V, E)$ είναι ο αλγόριθμος των Hopcroft-Karp [**24**] που έχει πολυπλοκότητα χείριστου χρόνου $O\left(|E|\sqrt{|V|}\right)$.

Γενικότερα όπως θα δούμε και παρακάτω, σε μετρητικά προβλήματα που ανήκουν στην $TotP$, έχουν βρεθεί ντετερμινιστικά προσεγγιστικά σχήματα πολυωνυμικού χρόνου, όταν το πρόβλημα απόφασης είναι εύκολο. Αλλά για την Permanent έχουν βρεθεί μόνο πιθναοτικοί προσεγγιστικοί αλγόριθμοι.

4.3 Προσπάθεια Κατασκευής Ντετερμινιστικού Προσεγγιστικού Αλγόριθμου

Στα πλαίσια αυτής της εργασίας έγιναν, προσπάθειες κατασκευής ενός ντετερμινιστικού προσεγγιστικού αλγορίθμου για την permanent.

Περιοριστήκαμε σε πίνακες που ανήκουν στο $M_n(\{0,1\})$. Σε αυτούς τους πίνακες γνωρίζουμε ότι η Permanent υπολογίζει τον αριθμό των Cyclic Cover σε μία γενική περίπτωση και τον αριθμό των perfect matchings αν ο πίνακας αναπαριστά έναν διμερή γράφο. Υπενθυμίζουμε ότι αν έχουμε ένα διάνυσμα $\boldsymbol{x} = [x_1, \dots, x_n]$ η νόρμα 1 είναι

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

Έστω ο πίνακας $A = [r_1, r_2, \dots, r_n]^T = [c_1, \dots, c_n]$ όπου r_i, c_i η i-οστή γραμμή και στήλη αντιστοίχως και

$$A = \begin{bmatrix} \alpha_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ \alpha_{n1} & \cdots & a_{nn} \end{bmatrix}$$

Τότε μπορούμε να γράψουμε την permanent του A ως εξής

$$perA = a_{11} perA_{r_1 c_1} + a_{12} perA_{r_1 c_2} + \cdots + a_{1j} perA_{r_1 c_j} + \cdots + a_{1n} perA_{r_1 c_n}$$

Όπου $A_{r_i c_j}$ είναι ο $(n-1) \times (n-1)$ υποπίνακας του A και προκύπτει αν αφαιρέσουμε την i-οστή γραμμή και την j-οστή στήλη. Τα στοιχεία $\alpha_{ij} \in \{0,1\}$ και συνεπώς αν $\alpha_{1j} = 0$ τότε δεν συνεισφέρει στον υπολογισμό της permanent. Άρα

$$perA = \sum_{\substack{a_{1j} \neq 0 \\ 1 \leq j \leq n}} a_{1j} perA_{r_1 c_j} \tag{4.a}$$

Αντίστοιχα επειδή ισχύει ότι $perA = perA^T$ ότι έχουμε πει μέχρι στιγμής για την πρώτη γραμμή ισχύει και για την πρώτη στήλη του πίνακα A. Δηλαδή

$$perA = \sum_{\substack{a_{i1} \neq 0 \\ 1 \leq i \leq n}} a_{i1} perA_{r_i c_1} \tag{4.b}$$

Τώρα παρατηρούμε ότι έχουμε μειώσει τον αριθμό των υποπροβλημάτων που πρέπει να υπολογίσουμε στο $\|r_1\|_1$ ή $\|c_1\|_1$ αντίστοιχα. Με

$$0 \le \|r_1\|_1 \le n \quad και \quad 0 \le \|c_1\|_1 \le n$$

Από τις σχέσεις (4.a) και (4.b) αντί να υπολογίσουμε τον εκάστοτε υποπίνακα θα υποθέτουμε ότι όλοι οι υποπίνακες του A έχουν περίπου την ίδια τιμή, και η permanent είναι

$$perA = \sum_{\substack{a_{1j}\neq 0 \\ 1\le j\le n}} a_{1j} perA_{r_1 c_j} \cong \|r_1\|_1 perA_{r_1 c_1} \ \text{ή} \ perA = \sum_{\substack{a_{i1}\neq 0 \\ 1\le i\le n}} a_{i1} perA_{r_i c_1} \cong \|c_1\|_1 perA_{r_1 c_1}$$

Μια ιδέα για να έναν προσεγγιστικό αλγόριθμο είναι να πούμε ότι

$$perA \cong \min\{\, \|r_1\|_1, \|c_1\|_1 \,\} \cdot perB \tag{4.c}$$

Όπου

$$B = \begin{cases} A_{r_1 c_j}, & αν \ \ \|r_1\|_1 = \min\{\|r_1\|_1, \|c_1\|_1\} \\ A_{r_i c_1}, & αν \ \ \|c_1\|_1 = \min\{\|r_1\|_1, \|c_1\|_1\} \end{cases} \tag{4.d}$$

Ένας αλγόριθμος που δεν δουλεύει αρκετά καλά, γιατί υπάρχουν στιγμιότυπα για τα οποία μπορεί να ξεγελασθεί είναι ο εξής;

Algorithm ApproxPerm

Input A_n

1. Ταξινόμησε τις γραμμές του A με βάση την $\|\cdot\|_1$ κατά αύξουσα σειρά και ονόμασε τον νέο πίνακα B_1.
2. Ταξινόμησε τις στήλες του B_1 με βάση την $\|\cdot\|_1$ κατά αύξουσα σειρά και ονόμασε τον νέο πίνακα B.
   Τώρα παρατηρούμε ότι $perA = perB_1 = perB$ διότι η permanent είναι αναλλοίωτη ως προς την αλλαγή γραμμών ή στηλών.
   Συμβολίζουμε τον $B = (b_{ij})$ και θεωρούμε τα διανύσματα
   $x_i = [b_{ii}, b_{i(i+1)}, \dots, b_{in}] \ \ και \ \ y_i = [b_{ii}, b_{(i+1)i}, \dots, b_{ni}] \ \ για\ κάθε\ 1 \le i \le n$
3. Set apx = 1
4. For $i = 1\ to\ n$-1
5. apx = $\min\{\|x_i\|_1, \|y_i\|_1\} * apx$
6. Endfor
7. **Return** apx

<u>Παρατήρηση 4.1:</u> Αν εξετάσουμε λίγο τον παραπάνω αλγόριθμο γίνεται φανερό πως επηρεάζεται από την διάταξη των στηλών και των γραμμών που έχει ο A, ενώ η permanent παραμένει αναλλοίωτη στην αναδιάταξη στηλών και γραμμών.

<u>Παρατήρηση 4.2:</u> Ο παραπάνω αλγόριθμος υπολογίζει το Optimum όταν όλα τα στοιχεία του πίνακα του A είναι 1.

Το πρόβλημα με αυτόν τον αλγόριθμο είναι ότι μπορεί να ξεγελαστεί εύκολα, διότι αν κάποιο διαγώνιο στοιχείο $b_{ii} = 0$ και τα διανύσματα x_i, y_i που ορίζονται στο i-οστό βήμα του αλγόριθμου είναι $x_i = y_i = 0$, τότε ο αλγόριθμος μπορεί να υπολογίσει εσφαλμένα ότι η permanent είναι μηδέν.

Παράδειγμα 4.1: Για να γίνει αντιληπτό αυτό που εννοούμε παραπάνω έχουμε το εξής παράδειγμα

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Ο αλγόριθμος στο παραπάνω παράδειγμα υπολογίζει μηδέν ενώ η πραγματική τιμή είναι $PerA = Opt = 1$. Διότι στο τελευταίο βήμα ο αλγόριθμος θα «βλέπει τον κάτω δεξιά υποπίνακα $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ με $b_{44} = a_{44} = 0$ και $x_4 = y_4 = 0 \implies \min\{\|x_4\|_1, \|y_4\|_1\} = 0$.

■

Επίσης μία ακόμα ιδέα για να βελτιώσουμε τον παραπάνω αλγόριθμο ώστε είναι να διανύει όχι μόνο την κύρια διαγώνιο του πίνακα αλλά και την δευτερεύουσα διαγώνιο και προς τις δύο κατευθύνσεις. Όμως ακόμα και τότε για μεγάλο πίνακα και με περίεργη διάταξη των διανυσμάτων μπορεί να ξεγελαστεί.

Μπορούμε να αποφύγουμε την περίπτωση του να ξεγελαστεί ο αλγόριθμος μας απαιτώντας τα στοιχεία της κύριας διαγωνίου να είναι 1. Μπορούμε να το πετύχουμε αυτό τρέχοντας τον αλγόριθμο των Hopcroft-Karp για να βρούμε ένα Perfect Matching. Αν αναδιατάξουμε τις γραμμές ή τις στήλες με βάση το Perfect Matching που υπολογίσαμε τότε όλα τα στοιχεία της κύριας διαγωνίου του πίνακα θα είναι 1.

Εφαρμόζοντας την παραπάνω ιδέα μπορούμε να διορθώσουμε την περίπτωση που ο αλγόριθμος επιστρέφει μηδέν, ενώ ο πίνακας A έχει $perA \neq 0$.

Algorithm ApproxPerm2

Input A_n

1. Τρέξε τον αλγόριθμο για να βρεις ένα Perfect Matching
2. If δεν έχει Perfect Matching then $apx = 0$
3. Else
4. Αναδιάταξε τις στήλες με βάση το Perfect Matching
5. Set apx = 1
6. For $i = 1\ to\ n-1$
7. apx = $\min\{\|x_i\|_1, \|y_i\|_1\} * apx$
8. //Σχόλιο: $x_i = [\alpha_{ii}, \alpha_{i(i+1)}, \dots, \alpha_{in}]$ $και$ $y_i = [\alpha_{ii}, \alpha_{(i+1)i}, \dots, \alpha_{ni}]$
9.. Endfor
10. Endif
11. **Return** apx

Για να βελτιώσουμε τον παραπάνω αλγόριθμο μπορούμε να τον κάνουμε να ελέγχει τις διαγώνιους του πίνακα και προς τις δύο κατευθύνσεις.

Παράδειγμα 4.2: Ένα κακό παράδειγμα για το πόσο έξω μπορεί να πέσει η προσέγγιση αυτού του αλγόριθμου είναι η οικογένεια πινάκων A_n που έχουν μορφή

$$A_n = \begin{bmatrix} I_{n-1} & \begin{matrix}1\\ \vdots\\ 1\end{matrix} \\ 1 \cdots 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 0 & 1 \\ \vdots & \ddots & \vdots & 1 \\ 0 & \cdots & 1 & 1 \\ 1 & \cdots & 1 & 1 \end{bmatrix}, \qquad \text{όπου } I_n \text{ ο μοναδιαίος πίνακας } \; n \in \mathbb{N}$$

Είναι εύκολο να δούμε ότι ο πίνακας A_n έχει $Perm(A_n) = n$ και ο αλγόριθμος σε αυτή την περίπτωση υπολογίζει προσέγγιση $apx = 2^{n-1}$. Πράγμα που σημαίνει ότι το σφάλμα μπορεί να είναι και εκθετικό ως προς την πραγματική τιμή.

■

Βέβαια αναφέρουμε ότι αυτή είναι και η χειρότερη προσέγγιση που μπορεί να κάνει ο αλγόριθμος αυτός για την τιμή της permanent. Αν αναδιατάξουμε τον πίνακα φέρνοντας την τελευταία γραμμή πρώτη και την τελευταία στήλη πρώτη τότε δημιουργείται ο εξής πίνακας

$$A_n^{'} = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & & I_{n-1} \\ 1 & & \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 & 1 \\ \vdots & \ddots & \vdots & 1 \\ 1 & \cdots & 1 & 0 \\ 1 & \cdots & 0 & 1 \end{bmatrix}$$

Τώρα παρατηρούμε πως ο αλγόριθμος με είσοδο τον $A_n^{'}$ υπολογίζει προσέγγιση

$$apx = n = perm(A_n^{'}) = perm(A_n)$$

Αυτό είναι μία ένδειξη ότι για «κατάλληλη αναδιάταξη των διανυσμάτων του Α μπορούμε να πετύχουμε προσεγγίσεις κοντά στην πραγματική τιμή του Α.

Παρατήρηση 4.3: Άλλη μία παρατήρηση που πρέπει να κάνουμε είναι ότι αν ο πίνακας C_{n+m} είναι το ευθύ άθροισμα των A_n και B_m και είναι της μορφής

$$C = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix}$$

Αποδεικνύεται ότι η Permanent του C είναι $Perm(C) = Perm(A) \cdot Perm(B)$

Θα θέλαμε να φέρουμε έναν πίνακα Α όσο πιο πολύ μπορούμε στην μορφή που έχει C, δηλαδή

$$A = \begin{bmatrix} A_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & A_n \end{bmatrix} \;\; \text{ώστε} \;\; Perm(A) = \prod_{i=1}^{n} Perm(A_i) = Perm(A_1) \cdot \cdots \cdot Perm(A_n)$$

Βέβαια αυτό που περιγράφουμε δεν είναι απαραίτητα εφικτό για τον τυχαίο πίνακα Α. Συνεπώς η επόμενη ιδέα που είχαμε για να υπολογίσουμε την Permanent ενός πίνακα Α ήταν να τον φέρουμε στην παρακάτω μορφή με αναδιάταξη γραμμών και στηλών.

$$A = \begin{bmatrix} A_1 & & E \\ & \ddots & \\ E^{'} & & A_n \end{bmatrix} \tag{4.e}$$

Όπου E και E' είναι πίνακες που έχουν πολύ λιγότερα 1 σε σχέσει με τους $A_1, \dots, A_n$.
Τα Communities ενός γραφήματος είναι υπογραφήματα ενός γράφου G. Οι κόμβοι που ανήκουν σε ένα community έχουν μεγάλη πιθανότητα να έχουν περισσότερες ακμές με κόμβους που ανήκουν σε αυτό, σε σχέση με κάποιον άλλο κόμβο που δεν ανήκει σε αυτό.

Αν διατάξουμε τον πίνακα μας με βάση τα communities του πίνακα τείνουμε να τον φέρουμε στην μορφή (4.e).

Algorithm ApproxPerm3

Input A_n

1. Τρέξε τον αλγόριθμο για να βρεις τα Communities
4. Αναδιάταξε τις στήλες με βάση τα Communities
5. Set apx = 1
6. For $i = 1\ to\ n - 1$
7. apx = $\min\{\|x_i\|_1, \|y_i\|_1\} * apx$
8. //Σχόλιο: $x_i = [\alpha_{ii}, \alpha_{i(i+1)}, \dots, \alpha_{in}]$ $και$ $y_i = [\alpha_{ii}, \alpha_{(i+1)i}, \dots, \alpha_{ni}]$
9.. Endfor
11. **Return** apx

Δημιουργήσαμε για αυτόν τον αλγόριθμο, όπως και για τους, προηγούμενους μία υλοποίηση χρησιμοποιώντας τον αλγόριθμο του M. E. J. Newman [**25**]. Ο αλγόριθμος αυτός υπολογίζει τα communities κάνει χρήση ιδιοτιμών και ιδιοδιανυσμάτων για εντοπίσει σε έναν πίνακα τα communities, και δεν κάνει εκ των προτέρων υποθέσεις για το μέγεθος και τον αριθμό τους.

Το πρόβλημα με αυτόν τον αλγόριθμο είναι ότι ανακύπτει ξανά το πρόβλημα του να υπολογίσει ο αλγόριθμος μηδέν για έναν πίνακα A, ενώ $Perm(A) \neq 0$. Αναφέρουμε ότι ο αλγόριθμος αυτός έβγαζε τα πιο καλά αποτελέσματα, αν δεν έπεφτε στην παραπάνω περίπτωση.

Με βάση όσα μελετήσαμε στην προσπάθεια που κάναμε παραπάνω για δημιουργήσουμε ένα ντετερμινιστικό προσεγγιστικό σχήμα για την Permanent, προκύπτουν τα ακόλουθα ερωτήματα που θα μπορούσαν να είναι και μελλοντική εργασία.

1. Υπάρχει κάποιος τρόπος ώστε να αποφύγουμε ξανά αυτό το σφάλμα στον αλγόριθμο.
2. Επίσης μπορούμε να αναρωτηθούμε αν έχουμε έναν πίνακα της μορφής

$$A = \begin{bmatrix} A_1 & & E \\ & \ddots & \\ E' & & A_n \end{bmatrix}$$

   Αν ισχύει το εξής

$$Perm(A) \cong \prod_{i=1}^{n} Perm(A_i) + b, \qquad όπου\ b\ μία\ φραγμένη\ ποσότητα$$

3. Ένα τελευταίο πράγμα για το οποίο μπορούμε να αναρωτηθούμε είναι αν η Permanent μπορεί να χρησιμοποιηθεί για να αξιολογεί communities.

5 FPRAS ΚΑΙ FPTAS ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ ΤΟ #KNAPSACK

Το μετρητικό πρόβλημα του Knapsack (#Knapsack) είναι ένα #P-complete πρόβλημα, για το οποίο βρέθηκε αρχικά ένα; FPRAS αλγόριθμος και μετέπειτα ένας FPTAS αλγόριθμος ο οποίος προσεγγίζει εξίσου καλά την λύση του #Knapsack. Ο σκοπός που θα μελετήσουμε αυτούς τους αλγόριθμους, είναι για να κατανοήσουμε πως ο δυναμικός προγραμματισμός ήταν ικανός να αντικαταστήσει την τυχαιότητα στην προκειμένη περίπτωση.

5.1 #P Προβλήματα και Προσεγγιστικοί Αλγόριθμοι

Γενικότερα οι αλγόριθμοι που χρησιμοποιούν τυχαιότητα συνήθως είναι πιο απλοί και έχουν καλύτερους χρόνους επίδοσης σε σχέση με τους αντίστοιχους ντετερμινιστικούς. Παρ' όλα αυτά υπάρχει η εικασία ότι $BPP = P$, το οποίο σημαίνει ότι για πολυπλοκότητα πολυωνυμικού χρόνου, η τυχαιότητα δεν συνεισφέρει στην επιλυσιμότητα ενός προβλήματος. Αυτή η εικασία βασίζεται στο γεγονός ότι είναι λίγα τα προβλήματα για τα οποία δεν έχουμε βρει ντετερμινιστικό αλγόριθμο αλλά έχουμε βρει πολυωνιμικό πιθανοτικό αλγόριθμο. Μερικά χαρακτηριστικά παραδείγματα αλγορίθμων για τα οποία υπήρχε πιθανοτικός αλγόριθμος και αργότερα βρέθηκε και ντετερμινιστικός είναι:

1) Το primality testing το οποίο ήταν ανοικτό πρόβλημα για δεκαετίες ώσπου αποδείχθηκε ότι υπάρχει ντετερμινιστικός αλγόριθμος από το [**26**].
2) Ο έλεγχος πότε ένα πολυώνυμο είναι ταυτοτικά μηδέν.

Όταν πρόκειται για προσεγγιστικούς αλγόριθμους, τότε υπάρχουν ακόμα περισσότερα τέτοια παραδείγματα. Στην περίπτωση των προσεγγιστικών μετρικών προβλημάτων έγινε ευρεία χρήση του Markov Chain Monte Carlo (MCMC) με δοκιμές, μια τεχνική η οποία είναι εγγενώς βασισμένη στην τυχαιότητα και μέχρι στιγμής επιφέρει αξιοθαύμαστα αποτελέσματα.

Γενικότερα υπάρχουν προβλήματα που η λύση τους ξεκινά με τη μέθοδο τυχαίων δοκιμών (random sampling method) και στη συνέχεια η μέτρηση γίνεται με επαναλαμβανόμενες δοκιμές. Τέτοιου είδους προβλήματα είναι τα εξής:

i. #Coloring
ii. #Matchings (πόσα ταιριάσματα υπάρχουν).
iii. Εκτίμηση συναρτήσεων διαμέρισης (partition functions)
iv. Προσέγγιση όγκου ενός κυρτού σώματος (convex body), δηλαδή ενός κυρτού και συμπαγούς συνόλου.

Δεν θεωρείται παράλογο πως αν η είσοδος είναι μιας συγκεκριμένης μορφής να υπάρχει ντετερμινιστικός πολυωνυμικός αλγόριθμος [**26**].

Ένα βασικό αποτέλεσμα ντετερμινιστικών προσεγγιστικών μετρικών αλγορίθμων είναι ο αλγόριθμος του Weitz Count [**27**] που μετράει τα ανεξάρτητα σύνολα τα οποία δέχονται βάρη από μια δραστηριότητα λ για γράφους μέγιστου βαθμού Δ όταν Δ είναι σταθερά και $\lambda < \lambda_u(\Delta)$ όπου $\lambda_u(\Delta)$ είναι το κατώφλι μοναδικότητας για Δ-κανονικά δέντρα. Το αποτέλεσμα αυτό επεκτάθηκε στο πρόβλημα μέτρησης όλων των ταιριασμάτων (matchings)

σε έναν γράφο φραγμένου βαθμού στο [**28**]. Αυτά τα δύο αποτελέσματα είναι δύο παραδείγματα ενός πλήρους πολυωνυμικού προσεγγιστικού σχήματος για $\#P$-complete προβλήματα. Ένας βασικός περιορισμός για αυτά τα αποτελέσματα είναι ότι ο χρόνος εκτέλεσης τους είναι σχετικά μεγάλος.

5.2 Διατύπωση του Προβλήματος Knapsack

Έστω n αριθμημένα αντικείμενα με τα αντίστοιχα βάρη τους $w_1, \dots, w_n$ και $C \in \mathbb{Z}$, ένας ακέραιος να είναι η χωρητικότητα ενός σάκου. Το πρόβλημα λέει πόσα αντικείμενα μπορούμε να βάλουμε στον εν λόγω σάκο, ώστε το συνολικό τους βάρος να μην ξεπερνά τη χωρητικότητα του σάκου.

Συμβολισμοί:

Έστω ένα σύνολο S με $|S|$ θα συμβολίζουμε την πληθικότητα του S, δηλαδή το πλήθος των στοιχείων του.

Θα συμβολίζουμε με $[n]$ το σύνολο των φυσικών αριθμών μέχρι το n. Δηλαδή

$$[n] = \{1,2, \dots, n\}$$

Με $\mathcal{P}([n])$ θα συμβολίζουμε το δυναμοσύνολο του [n], δηλαδή το σύνολο όλων των υποσυνόλων του [n]

$$\mathcal{P}([n]) = \{S : S \subseteq [n]\}$$

Με I θα συμβολίζουμε το $\mathcal{P}(\mathcal{P}([n]))$ το οποίο είναι το δυναμοσύνολο των υποσυνόλων του [n]

$$I = \{S : S \subseteq \mathcal{P}([n])\}$$

Τέλος $\mathbb{R}^+ = [0, +\infty)$ $\kappa\alpha\iota$ $\overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\} = [-\infty, +\infty]$

Ορισμός 5.1: Ένα σύνολο $S \subseteq [n]$ θα είναι **λύση στο Knapsack** πρόβλημα αν $\sum_{i \in S} w_i \leq C$

Δηλαδή το S είναι το σύνολο των στοιχείων που έχουν επιλεγεί για την λύση και $\sum_{i \in S} w_i$ είναι το συνολικό τους βάρος και μικρότερο της χωρητικότητας C.

Ορισμός 5.2: Για να ορίσουμε τυπικά το σύνολο των εφικτών λύσεων σε αυτό το πρόβλημα θα χρησιμοποιήσουμε μια συνάρτηση $f : [n] \times \mathbb{R}^+ \to I$ με τύπο

$$f(i, c) = \left\{ S \subseteq [i] : S \neq \emptyset \ \kappa\alpha\iota \ \sum_{i \in S} w_i \leq C \right\}$$

Δηλαδή η εικόνα της f είναι ένα σύνολο, που αποτελείται από σύνολα δεικτών των αντικειμένων του προβλήματος Knapsack. Τα σύνολα που ανήκουν στο $f(i, c)$ θα αποτελούσαν λύσεις αν έχουμε λάβει υπόψιν μόνο τα i πρώτα στοιχεία του προβλήματος και έχουμε χωρητικότητα c.

Ορισμός 5.3: Ορίζουμε το **σύνολο των λύσεων ενός Knapsack προβλήματος** με n αντικείμενα και χωρητικότητας $C \in \mathbb{R}$ να είναι το

$$f(n,C) = \left\{ S \subseteq [n] : S \neq \emptyset \ και \ \sum_{i \in S} w_i \leq C \right\} \cup \{0\}$$

Ορισμός 5.4: Επίσης με $|f(n,C)|$ θα συμβολίζουμε το **πλήθος των λύσεων του προβλήματος Knapsack**.

Σε ένα πρόβλημα Knapsack θέλουμε να έχουμε και την λύση να μην βάλουμε τίποτα στον σάκο. Γι αυτό τον λόγο προσθέτουμε το 0 και θεωρούμε ότι $w_0 = 0$ ώστε να αντιπροσωπεύσουμε αυτή την λύση.

Παρατηρήσεις 5.1:

1) $0 \leq |f(n,C)| \leq 2^n$. Όλα τα υποσύνολα του συνόλου $\{1, \dots, n\}$ είναι 2^n το πλήθος, συμπεριλαμβανομένου του κενού, εμείς το κενό σύνολο δεν το μετράμε στο $|f(n,c)|$, αλλά προσθέτοντας τη μηδενική λύση προκύπτει η άνωθεν ανισότητα.
2) Αν $|f(n,c)| = 0 \Longrightarrow f(n,c) = \emptyset$ δηλαδή δεν υπάρχει εφικτή λύση στο πρόβλημα.

5.3 Ένα FPRAS για το #Knapsack

Αρχικά προτάθηκε πιθανοτικός αλγόριθμος, υποεκθετικού χρόνου, για το πρόβλημα μέτρησης λύσεων του Knapsack, ο οποίος ήταν βασισμένος σε σχεδόν-ομοιόμορφη δειγματοληψία εφικτών λύσεων, με τυχαίο μονοπάτι από τους Bayati, Gamarnik, Katz, Nair, Tetali στο [**28**]. Έπειτα ο Morris και ο Sinclair [**29**] βελτίωσαν τον παραπάνω αλγόριθμο χρησιμοποιώντας Μαρκοβιανές αλυσίδες και παρουσίασαν έναν FPRAS (fully polynomial randomized approximation scheme) αλγόριθμο.

Ο Dyer [**30**] επανήλθε με μία τελείως διαφορετική προσέγγιση στο πρόβλημα κατά την οποία χρησιμοποίησε δυναμικό προγραμματισμό σε συνδυασμό με δειγματοληψία απόρριψης και παρουσίασε και αυτός έναν FPRAS αλγόριθμο. Αν και είναι πιο απλή πιθανοτική μέθοδος, η τυχαιότητα παίζει σημαντικό ρόλο σε αυτόν τον αλγόριθμο διότι χωρίς τυχαιότητα θα είχε έναν παράγοντα προσέγγισης n. (Όπου n είναι το πλήθος των στοιχείων του Knapsack προβλήματος).

Σε αυτή την παράγραφο θα μελετήσουμε τον FPRAS αλγόριθμο του Dyer [**30**]. Αρχικά ο Dyer παρουσίασε έναν αλγόριθμο ψευδο-πολυωνυμικού χρόνου για το #Knapsack, του οποίου ο χρόνος εκτέλεσης εξαρτάται από τη χωρητικότητα του σάκου C.

Η αναδρομή του αλγορίθμου ακολουθεί την παρακάτω ιδέα: είτε συμπεριλαμβάνουμε το j-οστό αντικείμενο στη λύση του Knapsack και μειώνουμε την χωρητικότητα του σάκου κατά w_j ή δεν το συμπεριλαμβάνουμε στη λύση και η χωρητικότητα μένει αμετάβλητη.

Άρα αναδρομικά, ο αριθμός των λύσεων όταν έχουμε συμπεριλάβει τα j πρώτα αντικείμενα είναι το άθροισμα των λύσεων που περιέχουν το w_j συν τον αριθμό των λύσεων που δεν περιέχουν το w_j.

$$F(j,k) = F(j-1,k) + F\left(j-1,k-w_j\right), \qquad \alpha\nu\ j \geq 1$$
$$F(0,k) = \begin{cases} 1, & \alpha\nu\ k \geq 0 \\ 0, & \alpha\nu\ k < 0 \end{cases} \tag{5.b}$$

Στην παραπάνω αναδρομή η συνάρτηση $F(i,c) = |f(i,c)|$ του ορισμού **5.2** η οποία μετράει τις εφικτές λύσεις. Κάνοντας χρήση δυναμικού προγραμματισμού υπολογίζουμε το $F(n,C)$ σε χρόνο $O(nC)$.

Επειδή δεν θέλουμε ο χρόνος εκτέλεσης να εξαρτάται από το C, κανονικοποιούμε τα βάρη ώστε το C είναι μια ποσότητα πολυωνυμική ως προς το n. Με την κανονικοποίηση τα βάρη θα στρογγυλοποιηθούν προς τα κάτω ώστε να παραμείνουν ακέραιοι και αυτό θα έχει σαν αποτέλεσμα η πληθικότητα του συνόλου των εφικτών λύσεων να αυξηθεί. Με την στρογγυλοποίηση που θα κάνουμε οι λύσεις θα αυξηθούν το πολύ κατά $(n+1)$.

Κανονικοποίηση:

Χωρίς βλάβη της γενικότητας μπορούμε να θεωρήσουμε ότι τα βάρη είναι διατεταγμένα δηλαδή $0 \leq w_1 \leq w_2 \leq \cdots \leq w_n \leq C$.

Έστω τώρα τα κανονικοποιημένα βάρη με τον εξής τρόπο $w_i^{'} = \left\lfloor \frac{n^2}{W} w_i \right\rfloor$ και το νέο άθροισμα $C^{'} = n^2$ Έστω S να είναι μια λύση για το #Knapsack, χρησιμοποιώντας δυναμικό προγραμματισμό μπορούμε να υπολογίσουμε το $|S^{'}|$ σε χρόνο $O(n^3)$.

Παρατηρούμε ότι αν $T \in S$, δηλαδή το T είναι μία εφικτή λύση για το πρόβλημα Knapsack τότε

$$\sum_{i \in T} w_i^{'} \leq \frac{n^2}{C} \sum_{i \in T} w_i \leq \frac{n^2}{C} C \leq n^2$$

Συνεπώς το T είναι και αυτό λύση στο κανονικοποιημένο πρόβλημα Knapsack $\Longrightarrow S \subseteq S^{'}$.

Συγκρίνοντας το $|S|$ με το $|S^{'}|$:

Εφόσον έχουμε αποδείξει ότι $S \subseteq S^{'}$ τότε $|S| \leq |S^{'}|$.

Αν υπάρχει ολική συνάρτηση $g : S^{'} \to S$, η οποία είναι επί και έχει την ιδιότητα για κάθε $T \in S$ υπάρχουν το πολύ $(n+1)$ σύνολα στο $S^{'}$ τα οποία απεικονίζονται στο T, τότε μπορούμε να συμπεράνουμε ότι

$$|S^{'}| \leq (n+1)\,|S|$$

Επομένως αν υπάρχει μια τέτοια συνάρτηση g, τότε σε χρόνο $O(n^3)$ μπορούμε να υπολογίσουμε το $|S^{'}|$ το οποίο ικανοποιεί την ανισότητα

$$\frac{|S^{'}|}{n+1} \leq |S| \leq |S^{'}| \tag{5.c}$$

Το μόνο που απομένει είναι να ορίσουμε την εν λόγω συνάρτηση g, επιβεβαιώνοντας έτσι την ύπαρξη της.

Για κάθε $T \in S$ θέτουμε $g(T) := T$

Τώρα για $T \in S' \backslash S$, θέτουμε K να είναι ο μεγαλύτερος φυσικός k τέτοιος ώστε $w_k \leq C/n$ και μάλιστα ισχύει ότι $T \nsubseteq [K]$. Πράγματι αφού κάθε $T' \subseteq$ [K] είναι και λύση του αρχικού προβλήματος Knapsack (δηλαδή $T' \in S$) εφόσον

$$\sum_{i \in T'} w_i \leq \sum_{i \in T'} \frac{C}{n} \leq C \quad \Longrightarrow \quad T' \in S$$

Επειδή λοιπόν $T \nsubseteq [K]$, θέτουμε ℓ να είναι ο δείκτης του πιο βαριού αντικειμένου στο T με $\ell > K$. Τώρα ορίζουμε

$$g(T) := T \backslash \{\ell\}$$

Πρέπει να αποδείξουμε ότι $g(T) \in S$.

<u>Συμβολισμός:</u> Θα συμβολίζουμε το σφάλμα στρογγυλοποίησης με

$$\delta_i = w_i \frac{n^2}{C} - w_i' \quad \Longrightarrow \quad w_i = \frac{C}{n^2}(w_i' + \delta_i), \qquad \mu\varepsilon \ 0 \leq \delta_i \leq 1$$

Έτσι έχουμε

$$\sum_{i \in f(T)} w_i = \frac{C}{n^2} \sum_{i \in f(T)} (w_i' + \delta_i) = \frac{C}{n^2} \left(\left(\sum_{i \in T} w_i \right) - w_\ell' + \left(\sum_{i \in T} \delta_i \right) - \delta_\ell \right) \leq$$

$$\leq \frac{C}{n^2} \left(\left(\sum_{i \in T} w_i' \right) + n \right) - w_\ell \leq \frac{C}{n^2} \sum_{i \in T} w_i' \ \left(\varepsilon\varphi\acute{o}\sigma o\nu \ w_\ell > \frac{C}{n} \right) \ \leq$$

$$\leq \frac{C}{n^2}(n^2) \ \ (\alpha\varphi o\acute{u} \, T \in S') \ = C$$

Επομένως η $g(T) \in S$ και είναι καλά ορισμένη με τύπο

$$g(T) = \begin{cases} T, & \alpha\nu \ T \in S \\ T \backslash \{\ell\}, & \alpha\nu \ T \in S' \backslash S \end{cases}, \qquad \acute{o}\pi o\upsilon \ w_\ell = \max\{w_i : i \in T\}$$

Ας δούμε τώρα πόσα στοιχεία απεικονίζονται σε ένα στοιχείο $T_0 \in S$ μέσω της g που ορίσαμε. Έστω ένα σύνολο $T_0 \in S \subseteq S'$ και θέτουμε $w_0 = \max\{w_i : i \in T_0\}$.

Το T_0 απεικονίζεται στον εαυτό του μέσω της g.

Ακόμη για οποιοδήποτε σύνολο $T_0 \cup \{\ell\} \in S'$ τέτοιο ώστε $w_\ell > w_0$, $\ell \in [n]$ ισχύει ότι

$$g(T_0 \cup \{\ell\}) = T_0$$

Άρα στην χειρότερη περίπτωση το πολύ $(n+1)$ στοιχεία του $S^{'}$ απεικονίζονται στο $T_0 \in S$. Το οποίο αποδεικνύει ότι ισχύει η σχέση (5.c).

Για να βελτιώσουμε τον παράγοντα προσέγγισης, θα χρησιμοποιήσουμε ομοιόμορφη δειγματοληψία από το S' και μετά θα χρησιμοποιήσουμε αυτές τις εκτιμήσεις για να υπολογίσουμε το $|S|$.

<u>Δειγματοληψία:</u>

Θα εκμεταλλευτούμε την αναδρομή (5.b) για να κάνουμε ομοιόμορφη δειγματοληψία από το $S^{'}$. Θα ξεκινήσουμε με ένα άδειο σύνολο και θα προσθέτουμε αντικείμενα σε αυτό με βάση την συνάρτηση κατανομής πιθανότητας που ορίζει η συνάρτηση F. Άρα ο αλγόριθμος δειγματοληψίας είναι ο εξής:

1. Let $T := \emptyset,\ j := n \quad και \quad k := n^2$
2. While $j > 0$ do:
   (a) With probability $F\left(j-1, k-w_j^{'}\right)/F(j,k)$:
   Set $T = T \cup \{j\}\ and\ set\ k = k - w_j^{'}$
   (b) Set $j = j - 1$

Είναι εύκολο να δούμε ότι ο παραπάνω αλγόριθμος κάνει ομοιόμορφη δειγματοληψία από το $S^{'}$. Δεδομένου ότι έχει υπολογιστεί η F ο παραπάνω αλγόριθμος παράγει σε χρόνο $O(n)$ ένα δείγμα. Συνεπώς για t δείγματα θα χρειαστεί χρόνος $O(n^3)$ για να υπολογιστεί η F και μετά χρόνος $O(tn)$ για να παραχθούν t τυχαία δείγματα από το $S^{'}$, άρα ο συνολικός χρόνος που χρειάζεται είναι της τάξης $O(n^3 + tn)$.

<u>Μέτρηση:</u>

Έστω $p := |S|/|S^{'}|$, βλέπουμε ότι $1/n \leq p \leq 1$. Θα χρησιμοποιήσουμε την δειγματοληψία για να εκτιμήσουμε το p. Παράγονται $N = \frac{4n}{\varepsilon^2}\log(2/\delta)$ τυχαία σύνολα από το $S^{'}$. Για $i = 1, \dots, N$ θέτουμε

$$X_i = \begin{cases} 1, & αν\ το\ i\ οστό\ δείγμα\ ανήκει\ στο\ S \\ 0, & αλλιώς \end{cases}$$

Έστω τώρα $X = \sum_{i=1}^{N} X_i$ και $Z = |S^{'}|\frac{X}{N}$, παρατηρούμε ότι

$$\mu = E[Z] = \frac{|S^{'}|}{N}\sum_i E[X_i] = |S^{'}|\, p = |S|$$

Από το φράγμα του Chernoff έχουμε

$$\Pr[\,|X-\mu| \geq \varepsilon\mu\,] \leq 2\exp\left(-\frac{\mu\,\varepsilon^2}{4}\right) \leq \exp\left(-\frac{p\,N\,\varepsilon^2}{4}\right) \leq \frac{1}{\delta}$$

Αν το $X \in [\mu(1-\varepsilon), \mu(1+\varepsilon)] = pN[1-\varepsilon, 1+\varepsilon]$, τότε και $Z \in |S|[1-\varepsilon, 1+\varepsilon]$.

Αποτέλεσμα: Τελικά έχουμε έναν FPRAS αλγόριθμο που να υπολογίζει το $|S|$ σε χρόνο

$$O\left(n^3 + \left(\frac{n}{\varepsilon}\right)^2 \log\left(\frac{1}{\delta}\right)\right)$$

Συνοψίζοντας: Η ιδέα του Dyer για να επιλύσει το πρόβλημα ήταν η εξής:

«Αν μειώσουμε τη χωρητικότητα του προβλήματος πολλαπλασιάζοντας με ένα συντελεστή ο οποίος είναι πολυωνιμκού μεγέθους ως προς n, και ταυτόχρονα πολλαπλασιάσουμε και τα βάρη με τον ίδιο συντελεστή στρογγυλοποιώντας προς τα κάτω όπου χρειάζεται, τότε μπορούμε να μετρήσουμε τις λύσεις του νέου προβλήματος που δημιουργήθηκε αποδοτικά με χρήση ψευδο-πολυωνυμικού αλγορίθμου δυναμικού προγραμματισμού.»

Ο αλγόριθμος του Dyer τρέχει σε χρόνο $O\left(n^3 + \left(\frac{n}{\varepsilon}\right)^2 \log\left(\frac{1}{\delta}\right)\right)$

Αν χρησιμοποιηθεί τυχαιοποιημένη στρογγυλοποίηση τότε ο παραπάνω αλγόριθμος βελτιώνεται και εκτελείται σε χρόνο

$$O\left(\left(n^{2,5}\sqrt{\log\,(1/\varepsilon)} + \frac{n^2}{\varepsilon^2}\right)\log\left(\frac{1}{\delta}\right)\right)$$

5.4 Ένα FPTAS για το #Knapsack

Ο αλγόριθμος που θα παρουσιάσουμε είναι των D. Stefanakovic, S. Vampela και E. Vigoda [**31**]. Και αυτός αλγόριθμος, όπως και ο FPRAS αλγόριθμος, είναι δυναμικού προγραμματισμού αλλά είναι εμπνευσμένος από έναν ψευδο-πολυωνυμικό αλγόριθμο για το πρόβλημα απόφασης/βελτιστοποίησης του Knapsack. Όπως είδαμε η πολυπλοκότητα του αλγορίθμου απόφασης του Knapsack είναι $O(nC)$, όπου C είναι η χωρητικότητα του προβλήματος.

Για να αποφευχθεί η χρήση τυχαιότητας, χρησιμοποιείται ένα πιο εύστροφο δυναμικό πρόγραμμα, για παράδειγμα αντί να γίνει έλεγχος για όλες τις ακέραιες χωρητικότητες $1, 2, \ldots, C$, τι θα γινόταν αν υπολογίζαμε τις χωρητικότητες που αυξάνονται με βάση κάποια γεωμετρική σειρά; Αυτό θα είχε σαν αποτέλεσμα το μέγεθος του πίνακα στο πρόγραμμα από nC να μειωθεί σε $n\,\log C$, αλλά το πρόβλημα που προκύπτει είναι ότι όταν η χωρητικότητα πολλαπλασιάζεται με έναν παράγοντα της τάξης $\left(1+\frac{n}{2^n}\right)$ τότε ο αριθμός των λύσεων αλλάζει με έναν σταθερό παράγοντα, κάτι που μπορεί αν οδηγήσει σε μεγάλη ανακρίβεια.

Αντ' αυτού στον αλγόριθμο που θα παρουσιάσουμε,ο πίνακας αριθμείται με τους δείκτες i, j. Όπου $1 \leq i \leq n$ και συμβολίζει τα i πρώτα αντικείμενα του προβλήματος Knapsack και j είναι ο αριθμός των λύσεων του υποπροβλήματος για τα i αντικείμενα. Τα στοιχεία του

πίνακα θα είναι η ελάχιστη χωρητικότητα που μπορεί να υπάρξει με τους συγκεκριμένους δείκτες.

Τώρα θα παρουσιάσουμε πιο εμπεριστατομμένα όλα όσα αναφέραμε παραπάνω για τον αλγόριθμο.

Ορισμός 5.5: Ορίζουμε την **συνάρτηση** $\tau : \{0, \dots, n\} \times \mathbb{R}^+ \to \overline{\mathbb{R}}$ με

$$\tau(i,a) = \begin{cases} \inf\{c \in \mathbb{R} : |f(i,c)| \geq \lceil a \rceil\}, & \alpha\nu \ \alpha \leq 2^i \\ +\infty \quad , & \alpha\nu \ \alpha > 2^i \end{cases}$$

Δηλαδή το $\tau(i,a)$ είναι το μικρότερο δυνατό $c \in \mathbb{R}$ ώστε το πρόβλημα Knapsack με τα αντικείμενα $1,2,\dots,i$ και χωρητικότητα c να έχει τουλάχιστον α το πλήθος λύσεις.

Η ελάχιστη χωρητικότητα που υπολογίζει η παραπάνω συνάρτηση είναι το μικρότερο άθροισμα που μπορούν σχηματίσουν κάποια από τα $w_1, \dots, w_i$ ώστε το πρόβλημα να έχει τουλάχιστον α λύσεις.

Παρατήρηση 5.2: Επειδή $\lceil a \rceil \geq \alpha$ *για κάθε* $\alpha \in \mathbb{R}^+$ τότε αν το πρόβλημα έχει ακριβώς $\lceil a \rceil$ λύσεις τότε θα έχει και τουλάχιστον α λύσεις

Ουσιαστικά η βασική ιδέα που οδήγησε σε ένα ντετερμινιστικό πολυωνιμικό σχήμα για το #Knapsack ήταν να κοιτάξουμε το πρόβλημα ανάποδα, δηλαδή να μπορούμε να βρούμε ποια θα είναι η ελάχιστη δυνατή χωρητικότητα του σάκου ώστε να έχουμε τουλάχιστον α λύσει

Παράδειγμα 5.1: Έστω το πρόβλημα Knapsack με τρία αντικείμενα με βάρη $w_1 = 1$, $w_2 = 5$, $w_3 = 3$. Όλα τα δυνατά αθροίσματα που μπορούν να προκύψουν από τα w_1, w_2, w_3, αν τα ταξινομήσουμε, είναι

$$0, 1, 3, 4, 5, 6, 8, 9$$

Σε κάθε δυνατό άθροισμα αντιστοιχεί και μία λύση ξεκινώντας από το μηδέν διότι θεωρούμε και ως λύση, να μην μπει τίποτα στον σάκο. Άρα $\tau(3,7) = 8$ διότι για χωρητικότητα μικρότερη του 8 θα έχουμε λιγότερες από 7 λύσεις. Αντίστοιχα όλα τα δυνατά αθροίσματα που μπορούν να προκύψουν από τα w_1, w_2 είναι

$$0, 1, 5, 6$$

Και με αντίστοιχο συλλογισμό με τον παραπάνω έχουμε ότι $\tau(2,3) = 5$.

■

Βλέπουμε ότι για $i = 0$ μπορούμε να υπολογίσουμε εύκολα την τ για κάθε $\alpha \in \mathbb{R}^+$

$$\tau(0,\alpha) = \begin{cases} -\infty, & \alpha\nu \quad \alpha = 0 \\ 0 \ , & \alpha\nu \quad 0 < \alpha \leq 1 \\ +\infty, & \alpha\nu \quad \alpha > 1 \end{cases}$$

Ουσιαστικά αν σε ένα πρόβλημα Knapsack δεν έχουμε αντικείμενα, τότε η μόνη λύση που έχουμε είναι να μην βάλουμε τίποτα στον σάκο. Η συνάρτηση που ορίσαμε υπολογίζει την ελάχιστη δυνατή χωρητικότητα, ανάλογα με το πλήθος των λύσεων.

Πράγματι: Αν $\alpha = 0$ τότε από την παρατήρηση **5.1** και τον ορισμό **5.3** έχουμε ότι

$$\alpha = 0 \Longrightarrow f(0,c) = \emptyset \Longrightarrow \{0\} \notin f(0,c) \Longrightarrow c < 0$$

$$\alpha\varphi o\acute{u}\ \alpha\nu\ c \geq 0\ \tau o\ \{0\}\ \theta\alpha\ \acute{\alpha}\nu\eta\kappa\varepsilon\ \sigma\tau o\ f(0,c) \Longrightarrow |f(0,c)| \geq 1\ \acute{\alpha}\tau o\pi o$$

Άρα $\tau(0,0) = \inf\{c \in \mathbb{R} : |f(0,c)| = 0\} = \inf\{c \in \mathbb{R} : c < 0\} = -\infty$

Αν $0 < \alpha \leq 1$ τότε $\lceil \alpha \rceil = 1$ και μάλιστα $f(0,c) = \{\{0\}\} \Longrightarrow c \geq 0$

$$\tau(0,a) = \inf\{c \in \mathbb{R} : |f(0,c)| = 1\} = \inf\{c \in \mathbb{R} : c \geq 0\} = 0$$

Τέλος για $\alpha > 1$ προκύπτει άμεσα από τον ορισμό **5.5** ■

Ορισμός 5.6: Ο **αριθμός των λύσεων ενός Knapsack προβλήματος** μπορεί να οριστεί και ως

$$Z = \max\{a \in [0,2^n] : \tau(n,a) \leq C\}$$

Παρατήρηση 5.3: Έστω $i \in [n]$ και $\alpha, \alpha' \in \mathbb{R}^+$ $\mu\varepsilon\ \alpha \leq \alpha'$ τότε $\tau(i,a) \leq \tau(i,a')$.

Δηλαδή για έναν σταθερό αριθμό αντικειμένων, αν έχουμε περισσότερες λύσεις στο πρόβλημα τότε και η ελάχιστη δυνατή χωρητικότητα του σάκου θα είναι μεγαλύτερη.

Λήμμα 5.1: Για κάθε $i \in [n]$ και $\gamma\iota\alpha\ \kappa\acute{\alpha}\theta\varepsilon\ \alpha \in [0,2^i]$, η τ που ορίσαμε ικανοποιεί την παρακάτω αναδρομή

$$\tau(i,a) = \min_{k \in [0,1]} \max \begin{cases} \tau(i-1,ka) \\ \tau(i-1,(1-k)a) + w_i \end{cases}$$

Παρατήρηση 5.4: Από τον ορισμό **5.5** κάνουμε στρογγυλοποίηση στον αριθμό των λύσεων και αφού $k \in (0,1)$ τότε το ελάχιστο στην παραπάνω αναδρομή, ερευνάται για όλες τις ακέραιες τιμές του $[0,2^i]$ αφού $\alpha \in [0,2^i]$. Πράγμα που καθιστά την τ δύσκολα υπολογίσιμη. Το σκεπτικό είναι ότι θα υπολογίσουμε μία συνάρτηση που προσεγγίζει την τ.

Έτσι τώρα θα προχωρήσουμε σε μία συνάρτηση που προσεγγίζει την τ αποδοτικά. Θα ορίσουμε μία συνάρτηση όμοια με την τ αλλά το δεύτερο όρισμα της, αντί να παίρνει τιμές στους θετικούς πραγματικούς αριθμούς θα παίρνει τιμές από το σύνολο τιμών μιας γεωμετρικής σειράς.

$$\text{Έστω} \quad Q := 1 + \frac{\varepsilon}{n+1} \quad \kappa\alpha\iota\ s := \lceil n \log_Q 2 \rceil$$

Η συνάρτηση $T : \{0, \dots, n\} \times \{0, \dots, s\} \to [0, +\infty]$ ορίζεται αναδρομικά ως εξής

$$T(i,j) = \min_{k\in[0,1]} \max \begin{cases} T\left(i-1, \lfloor j + \ln_Q k \rfloor\right) \\ T\left(i-1, \lfloor j + \ln_Q (1-k) \rfloor\right) + w_i \end{cases} \quad (5.d)$$

Για να ακριβολογούμε η T ορίζεται από τον ακόλουθο αλγόριθμο **CountKnapsack**

Count Knapsack

Input: **Integers** $w_1, w_2, \dots, w_n, C$ ***and*** $\varepsilon > 0$

1. **Set** $T[0,0] = 0$ **and** $T[0,j] = \infty$ **for** $j > 0$
2. **Set** $Q = \left(1 + \varepsilon/(n+1)\right)$ **and** $s = \lceil n \log_Q 2 \rceil$
3. **For** $i = 1 \to n$, ***for*** $j = 0 \to s$, **set**

$$T(i,j) = \min_{k\in[0,1]} \max \begin{cases} T\left(i-1, \lfloor j + \ln_Q k \rfloor\right) \\ T\left(i-1, \lfloor j + \ln_Q (1-k) \rfloor\right) + w_i \end{cases}$$

where by convention, $T[i-1,k] = 0$ **for** $k < 0$

4. **Let**

$$j' := \max\{ j : T[n,j] \le C \}$$

5. **Output** $Z' := Q^{j'+1}$

Το ελάχιστο στον αναδρομικό τύπο (5.d) ή και στην αναδρομή του αλγορίθμου, υπολογίζεται αποδοτικά παρόλο που αφορά όλο το διάστημα [0,1].

Από την παρατήρηση **5.4** αναφέραμε ότι έχει νόημα να υπολογίζεται η τ (επομένως και η T) μόνο όταν το δεύτερο όρισμα είναι ακέραιος. Τώρα όμως, επειδή εμείς ελέγχουμε το σύνολο των λύσεων $[0,2^i]$ μόνο στα σημεία που υπολογίζει η γεωμετρική πρόοδος Q^i (αφού τα στρογγυλοποιήσουμε), έχουμε επί της ουσίας τόσα σημεία ώστε να μπορεί να υπολογιστεί η T αποδοτικά με κάποιο σφάλμα προσέγγισης.

Ο λόγος τώρα που η T προσεγγίζει την τ εκφράζεται με το παρακάτω λήμμα.

Λήμμα 5.2:

Έστω $i \ge 1$ και υποθέτουμε ότι για όλα τα $j \in \{0, \dots, s\}$ η $T[i-1,j]$ ικανοποιεί την

$$\tau\left(i-1, Q^{j-(i-1)}\right) \le T[i-1,j] \le \tau\left(i-1, Q^j\right)$$

Τότε και για όλα τα $j \in \{0, \dots, s\}$ η $T[i,j]$ υπολογιζόμενη από τον τύπο (1) ικανοποιεί

$$\tau\left(i, Q^{j-i}\right) \le T[i,j] \le \tau\left(i, Q^j\right)$$

Τώρα είμαστε έτοιμοι να αποδείξουμε ότι η έξοδος Z', του αλγορίθμου CountKnapsack θα είναι κατά έναν παράγοντα $(1 \pm \varepsilon)$ πολλαπλάσιο του Z .

Θεώρημα 5.1: Έστω ένα πρόβλημα knapsack n στοιχείων, με τα αντίστοιχα βάρη τους $w_1, \dots, w_n$ και χωρητικότητα C. Έστω ακόμη Z να είναι το πλήθος των λύσεων του knapsack προβλήματος. Υπάρχει ντετερμινιστικός αλγόριθμος ώστε για κάθε $\varepsilon \in (0,1)$ που δίνει στην έξοδο Z' τέτοιο ώστε

$$(1-\varepsilon)Z \leq Z' \leq Z, \qquad \sigma\varepsilon\ \chi\rho\acute{o}\nu o\ \ O\left(\frac{n^3}{\varepsilon}\log\left(\frac{n}{\varepsilon}\right)\right)$$

*Οι αποδείξεις για τα λήμματα **5.1**, **5.2** και του θεωρήματος **5.1** υπάρχουν στο* [**31**].

Παρατηρούμε ότι ο χρόνος εκτέλεσης συναγωνίζεται τον χρόνο εκτέλεσης του αλγορίθμου του Dyer και είναι ενδιαφέρον ότι στον παρόντα ντετερμινιστικό αλγόριθμο έχουμε γραμμική εξάρτηση από το ε^{-1}, ενώ στον αλγόριθμό του Dyer [**30**] αλλά και σε προγενέστερους αλγόριθμους όπως των Morris Sinclair [**29**] και των Dyer, Vazirani et.al [**32**] έχουμε χρόνο εκτέλεσης που εξαρτάται από το ε^{-2}.

6 ΑΝΑΠΤΥΞΗ FPTAS ΑΛΓΟΡΙΘΜΟΥ ΤΟΥ ΣΕ ΑΛΛΑ #P ΠΡΟΒΛΗΜΑΤΑ

Σε αυτό το κεφάλιαο θα ορίσουμε μετρητικά προβλήματα και θα προσπαθήσουμε να φτιάξουμε ένα πολυωνυμικό προσεγγιστικό σχήμα που να τα επιλύει. Επειδή έχει παρατηρηθεί ότι οι μετρητικές εκδοχές NP-complete προβλημάτων είναι #P-complete, θα ορίσουμε μετρητικά προβλήματα που έχουν εύκολο πρόβλημα απόφασης και θα προσπαθήσουμε να βρούμε έναν προσεγγιστικό αλγόριθμο που να τα επιλύει.

Όταν λέμε το πρόβλημα απόφασης θα είναι εύκολο εννοούμε να επιλύεται σε πολυωνυμικό χρόνο από μια μη ντετερμινιστική μηχανη Turing. Τέλος αναφέρουμε ότι είναι ανοικτό πρόβλημα ότι τα NP-complete προβλήματα έχουν αντίστοιχο μετρητικό πρόβλημα #P-complete.

6.1 Συμβολισμοί:

Έστω ένα σύνολο S με $|S|$ θα συμβολίζουμε την πληθικότητα του S, δηλαδή το πλήθος των στοιχείων του.

Θα συμβολίζουμε με $[n]$ το σύνολο των φυσικών αριθμών μέχρι το n. Δηλαδή

$$[n] = \{1,2, \dots, n\}$$

Με $\mathcal{P}([n])$ θα συμβολίζουμε το δυναμοσύνολο του [n], δηλαδή το σύνολο όλων των υποσυνόλων του [n]

$$\mathcal{P}([n]) = \{S : S \subseteq [n]\}$$

Τέλος με I θα συμβολίζουμε το $\mathcal{P}(\mathcal{P}([n]))$ το οποίο είναι το δυναμοσύνολο των υποσυνόλων του [n]

$$I = \{S : S \subseteq \mathcal{P}([n])\}$$

6.2 #M-Element-Knapsack

6.2.1 Ορισμός Προβλήματος M-Element-Knapsack:

Ορισμός 6.1: Έστω ένα σύνολο αριθμημένων αντικειμένων $[n]$ τα οποία έχουν βάρη $w_1, \dots, w_n$ αντίστοιχα με $w_i \in \mathbb{N}\ για\ 1 \leq i \leq n$, M ένας φυσικός αριθμός και C η χωρητικότητα του σάκου. Υπάρχουν M το πλήθος αντικείμενα που μπορούν να μπουν στον σάκο, δηλαδή υπάρχει M-άδα αντικειμένων χωρίς το συνολικό τους βάρος να ξεπερνά το C.

Ορισμός 6.2: **Μία λύση** στο παραπάνω πρόβλημα είναι ένα σύνολο $S \subseteq [n]$ τέτοιο ώστε $|S| = m$ και $\sum_{i \in S} w_i \leq C$

Κατά αντιστοιχεία με το Knapsack θα ορίσουμε το σύνολο των εφικτών λύσεων σε αυτό το πρόβλημα θα χρησιμοποιήσουμε μια συνάρτηση $f : [n] \times [M] \times \mathbb{R}^+ \to I$ με τύπο

$$f(i,m,c) = \left\{ S \subseteq [i] : |S| = m \ \ και \ \sum_{i \in S} w_i \leq c \right\}$$

Η εικόνα της f είναι ένα σύνολο, που αποτελείται από σύνολα φυσικών αριθμών. Τα σύνολα που ανήκουν στο $f(i,m,c)$ θα αποτελούσαν λύσεις αν λάμβάναμε υπόψιν τα i-πρώτα στοιχεία του προβλήματος, είχαμε χωρητικότητα c και μας ενδιέφερε να βάλουμε ακριβώς m αντικείμενα στον σάκο.

Ορισμός 6.3: Συνεπώς στο πρόβλημα M-Element-Knapsack το **σύνολο των εφικτών λύσεων** θα είναι το

$$f(n,M,C) = \left\{ S \subseteq [n] : |S| = M \ \ και \ \sum_{i \in S} w_i \leq C \right\}$$

Παρατηρούμε πως το σύνολο των εφικτών λύσεων μοιάζει με το σύνολο των λύσεων του Knapsack (Ορισμός ~) μόνο που έχει έναν επιπλέον περιορισμό στην πληθικότητα των υποσυνόλων του $[n]$.

6.2.2 Ορισμός Προβλήματος #M-Element-Knapsack

Ορισμός 6.4: Έστω ένα σύνολο αριθμημένων αντικειμένων $[n]$ τα οποία έχουν βάρη $w_1, \dots, w_n$ αντίστοιχα με $w_i \in \mathbb{N}\ για\ 1 \leq i \leq n$, M ένας φυσικός αριθμός και C η χωρητικότητα του σάκου. Πόσες M-άδες αντικειμένων υπάρχουν χωρίς το συνολικό τους βάρος να ξεπερνά το C.

Ορισμός 6.5: Η **λύση στο #M-Element-Knapsack** θα είναι πληθικότητα του $f(n,M,C)$, $|f(n,M,C)|$.

Θέλουμε να τονίσουμε ότι η εικόνα της f είναι ένα σύνολο και με τον συμβολισμό $|f(n,M,C)|$ εννούμε την πληθικότητα της εικόνας της f και όχι κάποιο μέτρο συνάρτησης.

Παρατήρηση 6.1: Αν αυξηθεί η χωρητικότητα c, τότε ο αρθμός των λύσεων αυξάνεται ή παραμένει σταθερός. Αντίστοιχα αν η χωρητικότητα c μειωθεί, τότε ο αρθμός των λύσεων θα μειωθεί ή παραμένει σταθερός

6.2.3 Αλγόριθμος Επίλσυης

Θα τροποποιήσουμε τον αλγόριθμο των Daniel Stefankovic et al. [**31**] ώστε να δημιουργήσουμε ένα πολυωνιμικό προσεγγιστικό σχήμα για το #M-Element-Knapsack.

Ορισμός 6.6: Θα ορίσουμε με ανάλογο τρόπο όπως στον ορισμό [**5.5**] μία συνάρτηση

$$\tau : \{0, \dots, M\} \times \{0, \dots, n\} \times \mathbb{R}^+ \to \overline{\mathbb{R}} \quad με$$

$$\tau(m,i,a) = \begin{cases} -\infty & , \quad αν\ \alpha = 0 \ ή \ m > i \\ \inf\{c \in \mathbb{R} : |f(i,m,c)| \geq \lceil a \rceil\}, & αν\ \alpha \leq \binom{i}{m} \ \ και \ \ m \leq i \\ +\infty & , \quad αλλιώς \end{cases}$$

Δηλαδή η $\tau(m,i,a)$ υπολογίζει την ελάχιστη χωρητικότητα ώστε να έχουμε τουλάχισοτν α το πλήθος m-άδες ως λύσεις, χρησιμοποιώντας τα i-πρώτα αντικείμενα του προβλήματος.

Η ελάχιστη δυνατή χωρητικότητα c που επιδιώκουμε να βρούμε είναι ένα άθροισμα m-όρων από τα $w_1, \dots, w_i$ σε κάθε βήμα και είναι φραγμένη αφού

$$0 \leq c \leq \sum_{i=1}^{n} w_i$$

Οι τιμές $\pm\infty$ δηλώνουν ότι δεν μπορούμε να έχουμε εφικτές λύσεις για την αντίστοιχη είσοδο. Είναι σημαντικό να παρατηρήσουμε ότι ο αριθμός των λύσεων θα πρέπει να είναι φυσικός αριθμός. Εμείς ορίσαμε την συνάρτηση τ έτσι ώατε το α να είναι ένας μη αρνητικός πραγματικός αριθμός και αυτό το επιδιώξαμε διότι η προσέγγιση θα γίνει στον αριθμό των λύσεων α. Επειδή ο α είναι εκθετικού μεγέθους ως προς n ($\alpha \in [0,2^n]$) στο M-Element-Knapsack, θα τον προσεγγίσουμε με τη χρήση ενός λογαρίθμου. Δηλαδή θα προσπελάσουμε το διάστημα των λύσεων χρησιμοποιώντας κάποια γεωμετρική σειρά ώστε να πάρουμε ένα πολυωνυμικό και προσεγγιστικό σχήμα.

Ο **αριθμός των λύσεων του M-Element-Knapsack** μπορεί να οριστεί και ως

$$Z = \max\{a \in [0,2^n] : \tau(M,n,a) \leq C\}$$

<u>Πρόταση 6.1:</u> Έστω $i \in [n]$, $0 \leq m \leq i$ και $\alpha, \alpha' \in \mathbb{R}^+$ $\mu\varepsilon$ $\alpha \leq \alpha'$ τότε

$$\tau(m,i,a) \leq \tau(m,i,a')$$

Δηλαδή για σταθερό $i \in [n]$ και $0 \leq m \leq i$, για να έχουμε περισσότερες λύσεις στο πρόβλημα τότε και η ελάχιστη δυνατή χωρητικότητα του σάκου θα πρέπει να είναι μεγαλύτερη .

<u>Απόδειξη:</u>

Έστω λοιπόν $i \in [n]$, $0 \leq m \leq i$ και $\alpha, \alpha' \in \mathbb{R}^+$ $\mu\varepsilon$ $\alpha \leq \alpha'$ τότε $\lceil\alpha\rceil \leq \lceil\alpha'\rceil$. Διακρίνουμε τις εξής περιπτώσεις

Αν $\lceil\alpha\rceil = \lceil\alpha'\rceil$ τότε είναι προφανές ότι $\tau(m,i,a) = \tau(m,i,a')$

Αν $\lceil\alpha\rceil < \lceil\alpha'\rceil$ τότε

Αν $\binom{i}{m} < \lceil\alpha\rceil < \lceil\alpha'\rceil$ τότε $\tau(m,i,a) = \tau(m,i,a') = +\infty$

Αν $\lceil\alpha\rceil < \binom{i}{m} < \lceil\alpha'\rceil$ τότε $\tau(m,i,a) = c \in \mathbb{R}$, ενώ $\tau(m,i,a') = +\infty$. Άρα

$$\tau(m,i,a) = c < +\infty = \tau(m,i,a')$$

Αν $\lceil\alpha\rceil < \lceil\alpha'\rceil < \binom{i}{m}$, τότε θέτουμε

$$\tau(m,i,a) = c \quad \kappa\alpha\iota \quad \tau(m,i,a') = c'$$

Από *παρατήρηση* **6.1** για μεγαλύτερο c έχουμε περισσότερες λύσεις. Άρα αν $c' < c$ τότε θα έπρεπε και $|f(i,m,c')| \leq |f(i,m,c)|$, και από τον ορισμό **6.6** του $\tau(m,i,a')$ προκύπτει ότι

$$\lceil \alpha \rceil < \lceil \alpha' \rceil \leq |f(i,m,c')| \leq |f(i,m,c)| \implies |f(i,m,c')| \geq \lceil \alpha \rceil$$

Και θα έπρεπε $\tau(m,i,a) = \inf\{c \in \mathbb{R} : |f(i,m,c)| \geq \lceil a \rceil\} = c'$. Άτοπο και άρα $c \leq c' \Leftrightarrow \tau(m,i,a) \leq \tau(m,i,a')$

Τελικά δείξαμε σε κάθε περίπτωση ότι για $\alpha \leq \alpha'$ προκύπτει $\tau(m,i,a) \leq \tau(m,i,a')$

■

Αυτό που λέει παραπάνω πρόταση είναι ότι η τ που ορίσαμε είναι μονοτονική ως προς τον αριθμό των λύσεων α.

Πρόταση 6.2: Η τ ικανοποιεί την παρακάτω αναδρομή για κάθε $i \in [n]$, $\forall\, m \in [M]$ και $\alpha \in \mathbb{R}^+$

$$\tau(m,i,a) = \min_{k \in [0,1]} \max \begin{cases} \tau(m-1, i-1, ka) + w_i \\ \tau(m, i-1, (1-k)a) \end{cases}$$

Για να γίνει καλύτερα αντιληπτό το maximum παραθέτουμε ότι το πρόβλημα σε κάθε βήμα θα έχει $(1-k)a$ πλήθος λύσεων που δεν περιέχουν το w_i και ka λύσεις που περιέχουν το w_i.

Παρατηρούμε ακόμα ότι για να υπολογίσει κάποιος το minimum στην αναδρομή θα πρέπει να ανατρέξει σε όλα τα $k \in [0,1]$. Στην πραγματικότητα εκεί που παρατηρείται αλλαγή στις τιμές της τ είναι για τιμές της μορφής $k = r/a$, για $r \in \mathbb{Z}$ με $0 \leq r \leq a$.

Απόδειξη:

Έστω λοιπόν $i \in [n]$, $m \in [M]$ και $\alpha \in \mathbb{R}^+$ και κάποιο $k \in [0,1]$. Για να αποδείξουμε την αναδρομή θεωρούμε ότι έχουμε υπολογίσει μέχρι τα $\tau(m', i', a)$, $0 \leq m' \leq m$, $0 \leq i' \leq i-1$ *και* $0 \leq \alpha < 2^n$.

Θα αποδείξουμε αρχικά ότι

$$\tau(m,i,a) \leq \min_{k \in [0,1]} \max \begin{cases} \tau(m-1, i-1, ka) + w_i \\ \tau(m, i-1, (1-k)a) \end{cases}$$

Θέτουμε

$$B = \max\{\, \tau(m-1, i-1, ka) + w_i \;,\; \tau(m, i-1, (1-k)a) \,\}$$

Από τον ορισμό της τ έχουμε ότι υπάρχουν τουλάχιστον ka λύσεις (στις οποίες θα συμπεριλάβουμε το i-στοιχείο) που περιέχουν $m-1$ το πλήθος αντικείμενα με κάποια από τα βάρη $w_1, \dots, w_{i-1}$ και χωρητικότητα

$$\tau(m-1, i-1, ka) \leq B - w_i \Leftrightarrow \tau(m-1, i-1, ka) + w_i \leq B \qquad (6.a)$$

Αντίστοιχα υπάρχουν $(1-k)a$ λύσεις (στις οποίες δε θα συμπεριλάβουμε το i-στοιχείο) που περιέχουν m το πλήθος αντικείμενα με κάποια από τα βάρη $w_1, \dots, w_{i-1}$ και χωρητικότητα $\tau(m, i-1, (1-k)a)$.

$$\textit{Παρατηρούμε ότι}\ \ \tau(m, i-1, (1-k)a) \leq B \tag{6.b}$$

Συνδιάζοντας τις (6.a) και (6.b) περιπτώσεις υπάρχουν τουλάχιστον α λύσεις με m στοιχεία και συμπεριλαμβάνουν κάποια από τα βάρη $w_1, \dots, w_i$ με χωρητικότητα

$$\tau(m, i, a) \leq B$$

Ο συνδιασμός των περιπτώσεων (6.a) και (6.b) δεν μας οδηγεί στο να μετράμε κάποια λύση δύο φορές διότι από την (6.a) μετράμε όλες τις m-άδες που περιέχουν το w_i ενώ στην περίπτωση (6.b) μετράμε τις m-άδες που δεν περιέχουν το w_i.

Εφόσον για το τυχαίο $k \in [0,1]$ που επιλέξαμε αρχικά ισχύει $\tau(m, i, a) \leq B$ τότε και

$$\tau(m, i, a) \leq \min_{k\in[0,1]} \max \begin{cases} \tau(m-1, i-1, ka) + w_i \\ \tau(m, i-1, (1-k)a) \end{cases} \tag{6.c}$$

Μένει να δείξουμε ότι

$$\tau(m, i, a) \geq \min_{k\in[0,1]} \max \begin{cases} \tau(m-1, i-1, ka) + w_i \\ \tau(m, i-1, (1-k)a) \end{cases}$$

Έστω τώρα ότι έχουμε α το πλήθος m-άδες μέχρι το στοιχείο i με ελάχιστη δυνατή χωρητιότητα $c_0 = \tau(m, i, a)$. Τότε

$$\alpha_0 + \alpha_1 = \alpha$$

Με α_1 να είναι οι m-άδες που συμπεριλαμβάνουν το i-οστό στοιχείο και α_0 οι m-άδες που δεν το συμπεριλαμβάνουν.

Θέτουμε $\beta = \frac{\alpha_1}{\alpha}$ να είναι το ποσοστό των λύσεων που περιέχουν το w_i και έχουμε ότι

$$\tau(m, i-1, (1-\beta)\alpha) \leq c_0 \quad \textit{και}\ \ \tau(m-1, i-1, \beta\alpha) \leq c_0 - w_i$$

και έτσι προκύπτει ότι

$$\max\{\, \tau(m, i-1, (1-\beta)\alpha)\, , \tau(m-1, i-1, \beta\alpha) + w_i \,\} \leq c_0$$

Άρα και

$$\tau(m, i, a) \geq \min_{k\in[0,1]} \max \begin{cases} \tau(m-1, i-1, ka) + w_i \\ \tau(m, i-1, (1-k)a) \end{cases} \tag{6.d}$$

Τελικά από (6.c) και (6.d) προκύπτει και η ισότητα, δηλαδή

$$\tau(m,i,a) = \min_{k\in[0,1]} \max \begin{cases} \tau(m-1,i-1,ka) + w_i \\ \tau(m,i-1,(1-k)a) \end{cases}$$

■

Θα ορίσουμε μία συνάρτηση η οποία θα προσεγγίζει τον αριθμό των λύσεων του M-Element-Knapsack με αποδοτικό τρόπο.

Θα ορίσουμε μία συνάρτηση όμοια με την τ αλλά το τρίτο όρισμα της, αντί να παίρνει τιμές στους θετικούς πραγματικούς αριθμούς θα παίρνει τιμές από το σύνολο τιμών μιας γεωμετρικής σειράς. Έστω

$$Q := 1 + \frac{\varepsilon}{n+1} \quad \kappa\alpha\iota \;\; s := \lceil n \log_Q 2 \rceil$$

Η συνάρτηση $T : \{0,\dots,M\} \times \{0,\dots,n\} \times \{0,\dots,s\} \to [0,+\infty]$ ορίζεται αναδρομικά ως εξής

$$T(m,i,j) = \min_{k\in[0,1]} \max \begin{cases} T(\, m-1\,, i-1\,, \lfloor j + \ln_Q k \rfloor\,) + \; w_i \\ T(\, m, i-1, \lfloor j + \ln_Q \;(1-k)\,\rfloor\,) \end{cases}, \qquad (6.e)$$

Για να ακριβολογούμε η T ορίζεται από τον ακόλουθο αλγόριθμο **CountMKnapsack**

CountMKnapsack

Input: Integers $w_1, w_2, \dots, w_n, C, M \quad and\ \varepsilon > 0$

1. Set $T[0,i,1] = 0$ for $i \geq 0$ and $T[0,i,0] = -\infty$ for $i \geq 0$
2. Set $T[0,i,j] = \infty$ for $i,j \geq 0$ and $T[1,i,0] = -\infty$ for $i \geq 0$
3. Set $T[1,0,j] = -\infty$ for $j \geq 0$

Γεμίζουμε τον πίνακα για m=0 (όλο) και για m=1 (την πρώτη γραμμή και στήλη)

4. Set Q=1+(ε/(n+1)) and s=$\lceil n \log_Q 2 \rceil$
5. for $m = 1 \to M$, for $i = 1 \to n$, for $j = 0 \to s$,
6. if $\left(m > i \;\; or \;\; j > \binom{i}{m} \right)$ then
7. if$(m > i)$ then
8. $T[m,i,j] = -\infty$
9. $else$
10. $T[m,i,j] = \infty$
11. $else$
12. $$T(m,i,j) = \min_{k\in[0,1]} \max \begin{cases} T(\, m-1\,, i-1\,, \lfloor j + \ln_Q k \rfloor\,) + \; w_i \\ T(\, m, i-1, \lfloor j + \ln_Q \;(1-k)\,\rfloor\,) \end{cases}$$

Σε περίπτωση που προκύψει $k < 0$ θα θεωρούμε $T[i-1,k] = -\infty$

13. $j' = \max\{j : T[M,n,j] \leq C\}$

14. Output $Z' := Q^{j'+1}$

Ο παραπάνω αλγόριθμος δημιουργεί έναν τρισδιάστατο πίνακα μεγέθους $M \times n \times s$ και σε κάθε κελί αποθηκεύει την ελάχιστη δυνατή χωρητικότητα που έχουν τα αθροίσματα m στοιχείων και συμπεριλαμβάνουν κάποια από τα $w_1, \dots, w_i$.

Παρατήρηση 6.2: Αν έχουμε i στοιχεία στο M-Element-Knapsack με βάρη $w_1, \dots, w_i$ και αναζητούμε όλες τις εφικτές λύσεις με m στοιχεία, τότε αυτες θα είναι το πολύ $\binom{i}{m}$ το πλήθος.

Επισημαίνουμε ότι το minimum στην αναδρομή (6.e) τυπικά αφορά όλο το διάστημα [0,1], ο αλγόριθμος όμως υπολογίζει το ελάχιστο για κάποιες ακέραιες τιμές (ούτος ή άλλος μόνο σε ακέραιες τιμές παρατηρείται αλλαγή) της T καθώς προσεγγίζει την πραγματική λύση του #M-Element-Knapsack.

Ο λόγος τώρα που η T προσεγγίζει την τ εκφράζεται με το παρακάτω λήμμα.

Λήμμα 6.1:

Έστω $i \geq 1,\ 0 \leq m \leq i$ και υποθέτουμε ότι για όλα τα $j \in \{0, \dots, s\}$ η $T[m, i-1, j]$ ικανοποιεί την

$$\tau\left(m, i-1, Q^{j-(i-1)}\right) \leq T[m, i-1, j] \leq \tau\left(m, i-1, Q^{j}\right)$$

Τότε και για όλα τα $j \in \{0, \dots, s\}$ η $T[m,i,j]$ υπολογιζόμενη από τον τύπο (6.e) ικανοποιεί την διπλή ανίσοση

$$\tau\left(m, i, Q^{j-i}\right) \leq T[m, i, j] \leq \tau\left(m, i, Q^{j}\right)$$

Απόδειξη:

Έστω $i \geq 1,\ 0 \leq m \leq i$ και $j \in \{0, \dots, s\}$. Από την υπόθεση του λήμματος έχουμε ότι

$$T\left[m-1, i-1, \lfloor j + \ln_Q k \rfloor\right] \geq \tau\left(m-1, i-1, Q^{\lfloor j+\ln_Q k \rfloor - (i-1)}\right) \geq \geq \tau(m-1, i-1, kQ^{j-i}), \quad (6.f)$$

$$T\left[m, i-1, \lfloor j + \ln_Q (1-k) \rfloor\right] \geq \tau\left(m, i-1, Q^{\lfloor j+\ln_Q (1-k) \rfloor - (i-1)}\right) \geq \geq \tau(m, i-1, (1-k)Q^{j-i}), \quad (6.g)$$

Συνδιάζοντας τις (6.f) και (6.g) αλλά και τις συναρτήσεις min και max έχουμε ότι

$$T[m,i,j] = \left(\min_{k \in [0,1]} \max \begin{Bmatrix} T\left(m-1, i-1, \lfloor j + \ln_Q k \rfloor\right) + w_i \\ T\left(m, i-1, \lfloor j + \ln_Q (1-k) \rfloor\right) \end{Bmatrix} \right) \geq$$

$$\geq \left(\min_{k\in[0,1]} \max \begin{cases} \tau\left(m-1, i-1, kQ^{j-i}\right) + w_i \\ \tau\left(m, i-1, (1-k)Q^{j-i}\right) \end{cases} \right) = \tau\left(m, i, Q^{j-i}\right)$$

Και έτσι παίρνουμε το κάτω φράγμα το όποιο είναι

$$\tau\left(m\,, i\,, Q^{j-i}\right) \leq T[m\,, i\,, j]$$

Μένει να αποδείξουμε το άνω φράγμα της ανίσοσης στο λήμμα μας. Από την πρόταση 1 και από την υπόθεση του λήμματος έχουμε ότι

$$T\left[m-1\,, i-1\,, \lfloor j + \ln_Q k \rfloor\right] \leq \tau\left(m-1\,, i-1\,, Q^{\lfloor j+\ln_Q k\rfloor}\right) \leq \tau(m-1\,, i-1\,, kQ^j\,), \tag{6.h}$$

$$\begin{aligned} T\left[m\,, i-1\,, \lfloor j + \ln_Q (1-k)\rfloor\right] &\leq \tau\left(m\,, i-1\,, Q^{\lfloor j+\ln_Q (1-k)\rfloor}\right) \leq \\ &\leq \tau(m-1\,, i-1\,, (1-k)Q^j\,), \end{aligned} \tag{6.i}$$

Συνδιάζοντας τις (6.h) και (6.i) και χρησιμοποιώντας τις minimum και maximum έχουμε

$$T[m,i,j] = \left(\min_{k\in[0,1]} \max \begin{cases} T\left(\, m-1\,, i-1\,, \lfloor\, j + \ln_Q k \rfloor\,\right) + \, w_i \\ T\left(\, m, i-1, \lfloor j + \ln_Q \,(1-k)\,\rfloor\,\right) \end{cases} \right) \leq$$

$$\leq \left(\min_{k\in[0,1]} \max \begin{cases} \tau\left(\, m-1\,, i-1\,, kQ^j\,\right) + \, w_i \\ \tau\left(\, m, i-1, (1-k)Q^j\,\right) \end{cases} \right) = \tau\left(i, Q^j\right)$$

Αποδεικνύοντας έτσι και το άνω φράγμα της σχέσης προς απόδειξη.

■

Τώρα είμαστε έτοιμοι να αποδείξουμε ότι η έξοδος Z', του αλγορίθμου CountMKnapsack θα είναι κατά έναν παράγοντα $(1 \pm \varepsilon)$ πολλαπλάσιο του Z .

<u>Θεώρημα 6.1:</u> Έστω ένα πρόβλημα M-Element-Knapsack n στοιχείων, με τα αντίστοιχα βάρη τους $w_1, \ldots, w_n$ και χωρητικότητα C. Έστω ακόμη Z να είναι το πλήθος των λύσεων του M-Element-knapsack προβλήματος. Υπάρχει ντετερμινιστικός αλγόριθμος ώστε για κάθε $\varepsilon \in (0,1)$ που δίνει στην έξοδο Z' τέτοιο ώστε

$$(1-\varepsilon)Z \leq Z' \leq (1+\varepsilon)Z, \qquad \sigma\varepsilon\ \chi\rho\acute{o}\nu o\ \ O\left(\frac{n^4}{\varepsilon}\log\left(\frac{n}{\varepsilon}\right)\right)$$

<u>Απόδειξη:</u>

Έχουμε ορίσει $j' = \max\{\, j : T[\, M, n\,, j] \leq C\}$ και από το *<u>λήμμα 1</u>* προκύπτουν τα εξής:

1) Η προσεγγιστική λύση Z' δεν υποτιμά ποτέ το Z αφού

$$C \;<\; T\left[M, n\,, j'+1\right] \;\leq\; \tau\left(M\,, n\,, Q^{j'+1}\right)$$

πράγμα που σημαίνει ότι υπάρχουν το πολύ $Q^{j'+1}$ λύσεις

2) Ισχύει επίσης ότι

$$\tau\left(M, n, Q^{j'-n}\right) \leq T[M, n, j'] \leq C$$

που σημαίνει ότι έχουμε τουλάχιστον $Q^{j'-n}$ λύσεις

Άρα

$$\frac{Z'}{Z} \leq \frac{Q^{j'+1}}{Q^{j'-n}} = Q^{n+1} = \left(1 + \frac{\varepsilon}{n+1}\right)^{n+1} \leq e^{\varepsilon}, \qquad (6.j)$$

Παρατηρούμε ότι

$$e^{-\varepsilon} \geq 1 - \varepsilon \quad και \quad e^{\varepsilon} \leq (1 + \varepsilon)$$

Για την ακρίβεια η δεύτερη ανίσοση δεν ισχύει αυστηρά απλά θέλουμε να υποδηλώσουμε ότι το e^{ε} φράσεται από το (1+ε) για μικρά ε.

Τελικά από την (6.j) αποδεικνύεται το ζητούμενο του θεωρήματος **6.1**.

6.2.4 Χρόνος Εκτέλεσης

Όπως έχουμε αναφέρει και παραπάνω στον αναδρομικό τύπο (6.e6.f) εντοπίζουμε το ελάχιστο στοιχείο ψάχνοντας τιμές σε ένα διακριτό σύνολο τιμών S, όπου το τρίτο όρισμα της T θα πάρει ακέραια τιμή. Για $j \in \{0,1, \dots, s\}$ το σύνολο S είναι της μορφής $S = S_1 \cup S_2$ με

$$S_1 = \{Q^{-j}, \dots, Q^0\} \quad και \quad S_2 = \{1 - Q^0, \dots, 1 - Q^{-j}\}$$

Συνεπώς ένα κελί $T[m, i, j]$ υπολογίζονται σε $O(s)$ χρόνο και ο αλγόριθμος γεμίζει έναν πίνακα μεγέθους $n \times m \times s$ με $m = O(n)$. Δηλαδή όλα τα κελιά του πίνακα είναι τάξης $O(n^2 s)$. Ακόμη προκύπτει ότι το $s = O\left(\frac{n^2}{\varepsilon}\right)$ συνεπώς ένας αφελής αλγόριθμος μπορεί αν υλοποιειθεί σε χρόνο $O(n^2 s^2) = O\left(\frac{n^6}{\varepsilon^2}\right)$.

Όμως στην πρόταση **6.2**, αποδείξαμε ότι η $\tau(m, i, a)$ είναι μονοτονική ως προς α. Θα εκεμεταλευτούμε αυτήν την ιδιότητα για να βελτιώσουμε την πολυπλοκότητα που μπορεί να έχει ο αλγόριθμος.

Εφόσον η $\tau(m, i, a)$ είναι μονοτονική και επειδή η T είναι ένας περιορισμός της τ αποδεικνύεται ότι και η T θα είναι μονοτονική ως προς α.

Έτσι θα ισχύει ότι καθώς αυξάνεται το $k \in [0,1]$:

- ο πρώτος όρος στο maximum της (3), δηλαδή ο $T(m-1, i-1, \lfloor j + \ln_Q k \rfloor) + w_i$ θα αυξάνεται
- ενώ ο όρος $T(m, i-1, \lfloor j + \ln_Q(1-k) \rfloor)$ θα μειώνεται

Άρα το ελάχιστο των maximum αυτών των δύο ποσοτήτων θα επιτυγχάνεται είτε όταν $k \in \{0,1\}$ ή για $k \in (0,1)$ ώστε για οποιοδήποτε $k' < k$ να ισχύει

$$T[m, i-1, \lfloor j + \ln_Q(1-k') \rfloor] < T(m-1, i-1, \lfloor j + \ln_Q k \rfloor) + w_i$$

και για οποιοδήποτε $k' > k$ ακόμη

$$T\left[\, m\,, i-1\,, \left\lfloor j + \ln_Q\left(1-k'\right)\right\rfloor\right] \geq T\left(\, m-1\,, i-1\,, \left\lfloor j + \ln_Q\, k\right\rfloor\,\right) +\, w_i$$

Δυστυχώς δεν μπορούμε να έχουμε το σύνολο S ταξινομημένο, αλλά έχουμε τα S_1 και S_2 ταξινομημένα. Άρα μπορούμε να κάνουμε δυαδική αναζήτηση στο S_1 ώστε να βρούμε ένα k_1 ώστε $k_1 \in \{0,1\}$ ή να έχει την ιδιότητα που περιγράψάμε παραπάνω. Κατά αντίστοιχο τρόπο κάνουμε δυαδική αναζήτηση στο S_2 και υπολογίζουμε ένα k_2. Τελικά συγκρίνουμε τις τιμές της T για k_1, k_2 και κρατάμε αυτή με την μικρότερη τιμή.

Με αυτόν τον τρόπο γεμίζουμε ένα κελί $T[\, m\,, i\,, j\,]$ σε χρόνο $O(\log s)$, δηλαδή ο χρόνος που θα χρειαστεί να γεμίσει ο πίνακας θα είναι της τάξης

$$O(n^2 s \log s) = O\left(\frac{n^4}{\varepsilon}\log\frac{n}{\varepsilon}\right)$$

Έτσι αποδεικνύεται πλήρως το θεώρημα **6.1**.

■

6.3 Πρόβλημα #Factor-C Sum Partition

6.3.1 Ορισμός Προβλήματος Factor-C Sum Partition

Έστω ένα σύνολο από αξίες $A = \{\alpha_1, \dots, \alpha_n\}$. Μπορούμε να διαμερίσουμε το A σε δύο υποσύνολα έτσι ώστε η συνολική αξία του ενός να είναι C-φορές μεγαλύτερη από τη συνολική αξία του άλλου; Πιο τυπικά έχουμε:

Ορισμός 6.7: Έστω ένα πολυσύνολο $A = \{\alpha_1, \dots, \alpha_n\}$, με $\alpha_1, \dots, \alpha_n \in \mathbb{N}$ και C μία παράμετρος τέτοια ώστε $0 < C \leq 1$. Ονομάζουμε το σύνολο των δεικτών των στοιχείων του A με S και ισχύει ότι $S = [n]$. Εξετάζουμε αν υπάρχει διαμέριση του S, δηλαδή υπάρχουν $S_1, S_2 \subseteq S$ με $S_1, S_2 \neq \emptyset$, $S_1 \cup S_2 = S$ και $S_1 \cap S_2 = \emptyset$, ώστε να ισχύει η εξής ιδιότητα:

$$\sum_{i \in S_1} \alpha_i \leq C \cdot \sum_{j \in S_2} a_j, \tag{6.k}$$

Εναλλάκτικά η (6.k) μπορεί να γραφεί

$$\frac{\sum_{i \in S_1} \alpha_i}{\sum_{j \in S_2} a_j} \leq C \tag{6.l}$$

Σημειώνουμε ότι ένα πολυσύνολο από θετικούς ακεραίους είναι ένα σύνολο στο οποίο ένας αριθμός μπορεί να επαναλαμβάνεται. Για παράδειγμα το $\{1, 2, 3, 3, 5, 5, 1, 6, 7, 2, 25\}$ *είναι ένα πολυσύνολο, ενώ το στα σύνολα δεν έχουμε επαναλήψεις ίδιων στοιχείων.*

Ακόμη παρατηρούμε ότι το πρόβλημα Factor-C Sum Partition είναι ένα πρόβλημα απόφασης, και μπορεί να λυθεί σε πουλωνυμικό χρόνο. Πράγματι, ταξινομούμε τα στοιχεία του A σε αύξουσα σειρά και ελέγχουμε αν το

$$\frac{\alpha_1}{\sum_{i=2}^{n} a_i} \leq C$$

Αν ισχύει η παραπάνω ανίσοση τότε τα $S_1 = \{1\}$, $S_2 = \{2, \dots, n\}$ αποτελούν λύση για το Factor-C Sum Partition. Αν

$$\frac{\alpha_1}{\sum_{i=2}^{n} a_i} > C$$

Τότε το πρόβλημα δεν έχει λύση.

6.3.2 Ορισμός Προβλήματος #Factor-C Sum Partition

Ορισμός 6.8: Έστω πολυσύνολο $A = \{\alpha_1, \dots, \alpha_n\}$, με σύνολο δεικτών το $S = [n]$, τα $\alpha_1, \dots, \alpha_n \in \mathbb{N}$ και C μία παράμετρος τέτοια ώστε $0 < C \leq 1$. Πόσα ζεύγη συνόλων $S_1, S_2 \subseteq S$ υπάρχουν με $S_1, S_2 \neq \emptyset$, $S_1 \cup S_2 = S$ και $S_1 \cap S_2 = \emptyset$ και να ισχύει η εξής ιδιότητα;

$$\sum_{i \in S_1} \alpha_i \leq C \cdot \sum_{j \in S_2} a_j, \tag{6.m}$$

Ή εναλλακτικά πόσες διαμερίσεις του S υπάρχουν ώστε

$$\frac{\sum_{i \in S_1} \alpha_i}{\sum_{j \in S_2} a_j} \leq C \tag{6.n}$$

6.3.3 Αλγόριθμος Επίλυση του #Factor-C Sum Partition

Θα επιλύσουμε το #Factor-C Sum Partition χρησιμοποιώντας τον προσεγγιστικό αλγόριθμο επίλυσης του #Knapsack που παρουσιάσαμε στο προηγούμενο κεφάλαιο. Το #Factor-C Sum Partition μπορεί να λυθεί με τον εξής FPTAS αλγόριθμο.

CountFactorCSumPartition

Input: Integers $\alpha_1, \dots, \alpha_n$ **and** $\varepsilon > 0$ **and** $0 < C \leq 1$

1. **Set** $W = a_1 + a_2 + \cdots + a_n$
2. **Set** $C' = (C * W)/(1 + C)$
3. **Set** Z = CountKnapsack $(a_1, \dots, a_n, C', \varepsilon)$
4. **Output** Z

Απόδειξη Ορθότητας Αλγορίθμου CountFactorCSumPartition:

Έχουμε λοιπόν δοσμένα από το πρόβλημα, ένα πολυσύνολο $A = \{\alpha_1, \dots, \alpha_n\}$ και μία παράμετρο C με $0 < C \leq 1$, το σύνολο S του προβλήματος είναι το $S = [n]$.

Θέτουμε $W = \sum_{i=1}^{n} a_i$ και επειδή τα S_1, S_2 που αναζητούμε αποτελούν μία διαμέριση του S έχουμε ότι

$$\sum_{i \in S_1} \alpha_i + \sum_{j \in S_2} a_j = W$$

Και έτσι μπορούμε να καταλήξουμε στη σχέση

$$\sum_{j \in S_2} a_j = W - \sum_{i \in S_1} \alpha_i \tag{6.o}$$

Αντικαθιστώντας στη σχέση (6.n) την (6.o) έχουμε την

$$\frac{\sum_{i \in S_1} \alpha_i}{W - \sum_{i \in S_1} \alpha_i} \leq C \tag{6.p}$$

Από την παραπάνω σχέση γίνεται αντιληπτό ότι για να υπολογίσουμε μία λύση του Factor-C Sum Partition χρειάζεται να βρούμε ένα υποσύνολο του S, αντί δύο, που να την ικανοποιεί. Έχουμε βέβαια υπόψιν μας ότι το S_2 μπορεί να υπολογισθεί ως συμπλήρωμα του S, δηλαδή $S_2 = S \backslash S_1$. Άρα για το #Factor-C Sum Partition επιθυμούμε να βρούμε τον αριθμό των υποσυνόλων του S, ώστε το αντίστοιχο άθροισμα των στοιχείων του A να ικανοποιεί την (6.p).

Ονομάζουμε C' το άθροισμα των στοιχείων του A που δημιουργείται από ένα υποσύνολο δεικτών του S. Παρατηρούμε ότι για διάφορα υποσύνολα του S θα έχουμε και διαφορετικό C' και το C' θα λαμβάνει μέγιστη τιμή, ενώ ταυτόχρονα ικανοποιεί την (6.p) όταν ισχύει η παρακάτω ισότητα

$$\frac{C^{'}}{W - C'} = C \quad \Leftrightarrow \quad C^{'} = \frac{WC}{1 + C}$$

Το W είναι μια σταθερά σε κάθε στιγμιότυπο του προβλήματος και έτσι καθώς το $C^{'}$ αυξάνεται, το $W - C^{'}$ μειώνεται. Οπότε όταν το C' είναι μεγαλύτερο του $WC/(1 + C)$ η (6.p) δεν ικανοποιείται και άρα η διαμέριση των στοιχείων του A δεν αποτελεί λύση. Αντίθετα αν το C' είναι μικρότερο του $WC/(1 + C)$ η (6.p) ικανοποιείται. Συνεπώς οποιοδήποτε υποσύνολο με C' το οποίο είναι μικρότερο ή ίσο της ποσότητας $WC/(1 + C)$ υποδεικνύει μία διαμέριση που είναι λύση του προβλήματος.

Δηλαδή ψάχνουμε να βρούμε τον αριθμό των υποσυνόλων του S ώστε αν αθροίσουμε τα στοιχεία του A το άθροισμα να είναι μικρότερο ή ίσο του C'. Αν τώρα αν ονομάσουμε το S ένα σύνολο αντικειμένων και αν πούμε ότι το A είναι το αύνολο με τα αντίστοιχα βάρη των οποίων το άθροισμα πρέπει να είναι μικρότερο ή ίσο του C', τότε βλέπουμε ότι έχουμε ακριβώς το #Knapsack αντί του #Factor-C Sum Partition.

Συνεπώς αρκεί να τρέξουμε τον FPTAS αλγόριθμο του #Knapsack έχοντας για βάρη τα $\alpha_1, \dots, \alpha_n$ και χωρητικότητα την μέγιστη τιμή του C' η οποία όπως αναφέραμε παραπάνω είναι $C^{'} = WC/(1 + C)$.

■

Ο αλγόριθμος **CountFactorCSumPartition** θα έχει τον ίδιο χρόνο εκετέλεσης με τον **CountKnapack** που παρουσιάσαμε στο προηγούμενο κεφάλαιο και θα είναι ίδιας ακρίβειας.

6.4 Πρόβλημα Διαμοίρασης Μονομερούς Ικανοποίησης για Δύο Άτομα (Biased Share 2-Partition Problem)

6.4.1 Ορισμός Προβλήματος

Έστω ότι έχουμε n αντικείμενα και τα μοιράζουμε σε δύο σε δικαιούχους A και B. Για τον A η χρησιμότητα των αντικειμένων δίνεται από το σύνολο $A = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, ενώ για τον B από το $B = \{b_1, \dots, b_n\}$. Υπάρχει μοιρασιά των αντικειμένων ώστε μόνο ο ένας να μην ζηλεύει τον άλλο; Δηλαδή υπάρχει μοιρασιά κατά την οποία τουλάχιστον ο ένας από τους δύο να είναι ευχαρηστημένος με τα αντικείμενα που θα πάρει; Αν χρησιμοποιήσουμε και την γλώσσα των μαθηματικών έχουμε το παρακάτω.

Ορισμός 6.9: Έστω $n \in \mathbb{N}, S = [n]$ και $A = \{a_1, \dots, a_n\} \subseteq \mathbb{N},\ B = \{b_1, \dots b_n\} \subseteq \mathbb{N}$ και

$$W = \{\ (a_i, b_i) \subseteq A \times B : 1 \leq i \leq n\} = \{\ (a_1, b_1), \dots, (a_n, b_n)\ \}$$

Το ερώτημα είναι αν υπάρχει διαμέριση του S, δηλαδή υπάρχουν $S_1, S_2 \subseteq S$ με $S_1, S_2 \neq \emptyset$, $S_1 \cup S_2 = S$ και $S_1 \cap S_2 = \emptyset$, ώστε να ισχύει η εξής ιδιότητα:

$$\sum_{i \in S_1} \alpha_i \geq \sum_{i \in S_1} b_i \quad \kappa\alpha\iota \quad \sum_{i \in S_2} b_i \geq \sum_{i \in S_2} a_i \tag{6.q}$$

Ορισμός 6.10: Θα λέμε ότι η μοιρασιά θα είναι **δίκαιη** (**envy free**) αν ισχύει

$$\sum_{i \in S_1} \alpha_i \geq \sum_{i \in S_2} \alpha_i \quad \kappa\alpha\iota \quad \sum_{i \in S_2} b_i \geq \sum_{i \in S_1} b_i$$

Με βάση το παραπάνω θα λέμε ότι **ο Α δεν ζηλεύει τον Β** αν ισχύει

$$\sum_{i \in S_1} \alpha_i \geq \sum_{i \in S_2} \alpha_i$$

Ισχυρισμός 6.1: Αν ισχύει η σχέση (6.q) τότε τουλάχιστον ένας από τους δύο δικαιούχους δεν θα ζηλεύει τον άλλο.

Απόδειξη: Έστω ότι ισχύει η σχέση (6.q),

Αν $\sum_{i \in S_2} b_i \geq \sum_{i \in S_1} b_i$ τότε είναι άμεσο ότι ο Β δεν ζηλεύει τον Α

Αν $\sum_{i \in S_2} b_i \leq \sum_{i \in S_1} b_i \ \Leftrightarrow\ \sum_{i \in S_1} b_i \geq \sum_{i \in S_2} b_i$ και επειδή ισχύει η (6.q) έχουμε ότι

$$\sum_{i \in S_1} \alpha_i \geq \sum_{i \in S_1} b_i \geq \sum_{i \in S_2} b_i \geq \sum_{i \in S_2} a_i \ \Longrightarrow\ \sum_{i \in S_1} \alpha_i \geq \sum_{i \in S_2} \alpha_i$$

Που σημαίνει ότι ο Α δεν ζηλεύει τον Β.

■

Στο παραπάνω πρόβλημα απόφασης μπορούμε να απαντήσουμε εύκολα. Πράγματι χωρίς βλάβη της γενικότητας υποθέτουμε ότι

$$\sum_{i=1}^{n} a_i \leq \sum_{i=1}^{n} b_i \tag{6.r}$$

Αν υπάρχει $i_0 \in \mathbb{N}$ με $1 \leq i_0 \leq n$ ώστε να ισχύει

$$\alpha_{i_0} \geq b_{i_0} \Longrightarrow\ -\alpha_{i_0} \leq -b_{i_0} \tag{6.s}$$

Αν προσθέσουμε κατά μέλη τις **(6.r)** και **(6.s)** προκύπτει ότι

$$\sum_{i \in S \setminus S_1} a_i \leq \sum_{i\ \in S \setminus S_1} b_i$$

Πουσημαίνει ότι τα σύνολα $S_1 = \{i_0\}$, $S_2 = S \backslash S_1$ αποτελούν λύση στο πρόβλημα. Αν δεν υπάρχει i_0 ώστε $\alpha_{i_0} \geq b_{i_0}$ τότε δεν υπάρχει διαμέριση που να ικανοποιεί τις ανισόσεις της **(6.q)**.

6.4.2 Ορισμός #Biased Share 2-Partition Problem

Στο μετρητικό πρόβλημα μας ενδιαφέρουν πόσες λύσεις υπάρχουν στο πρόβλημα διαμοίρασης μονομερούς ικανοποίησης για δύο άτομα.

Ορισμός 6.11: Έστω $n \in \mathbb{N}$, $S = [n]$ και ένα σύνολο $W = \{ (a_i, b_i) \subseteq \mathbb{N} \times \mathbb{N} : 1 \leq i \leq n\}$. Πόσες διαμερίσεις του S, δηλαδή πόσα ζεύγη $S_1, S_2 \subseteq S$ με $S_1, S_2 \neq \emptyset$, $S_1 \cup S_2 = S$ και $S_1 \cap S_2 = \emptyset$, υπάρχουν ώστε να ισχύει η εξής ιδιότητα:

$$\sum_{i \in S_1} \alpha_i \geq \sum_{i \in S_1} b_i \quad \kappa\alpha\iota \quad \sum_{i \in S_2} b_i \geq \sum_{i \in S_2} a_i \tag{6.t}$$

Ισχυρισμός 6.2: Η λύση του #Biased Share 2-Partition Problem με είσοδο $\{(\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n)\}$ ταυτίζεται με το άθροισμα των λύσεων του #M-Element Knapsack με είσοδο $w_1, \dots, w_n$, χωρητικότητα mβ και πληθικότητα m, για όλα τα $m \in [n-1]$.

Απόδειξη:
Χωρίς βλάβη της γενικότητας θεωρούμε ότι

$$\sum_{i=1}^{n} a_i \geq \sum_{i=1}^{n} b_i \tag{6.u}$$

Αν δεν ισχυέι το παραπάνω απλά αντιστρέφουμε το ρόλο των α_i με τα b_i και συνεχίζουμε θεωρώντας δεδομένη την (6.u). Τώρα παρατηρούμε την εξής ιδιότητα, αν υπάρχει ένα $S_1 \subseteq S$ για το οποίο ισχύει

$$\sum_{S_1} b_i > \sum_{S_1} a_i \iff -\sum_{S_1} a_i > -\sum_{S_1} b_i \tag{6.v}$$

Τότε αν προσθέσουμε την (6.v) από την (6.u)

$$\sum_{i=1}^{n} a_i - \sum_{S_1} \alpha_i \geq \sum_{i=1}^{n} b_i - \sum_{S_1} b_i \iff \sum_{S \backslash S_1} a_i \geq \sum_{S \backslash S_1} b_i \tag{6.w}$$

Το οποίο σημαίνει ότι αν ένα $S_1 \subset S$ ικανοποιεί την (6.v), τότε μαζί με το συμπλήρωμα του θα αποτελούν λύση στο Biased Share 2-Partition.

Το αντίστροφο, δηλαδή ότι κάθε λύση του Biased Share 2 – Partition S_1, S_2 μπορούμε να την αντιστοιχίσουμε στο σύνολο S_2 το οποίο ικανοποιεί την σχέση (6.v). Με αποτέλεσμα οι δύο περιγραφές να ταυτίζονται.

Δηλαδή ο αριθμός των λύσεων του Biased Share 2-Partition ταυτίζεται με το πλήθος των υποσυνόλων του S για τα οποία ισχύουν οι παρακάτω ισοδυναμίες

$$\sum_{S_1} b_i \geq \sum_{S_1} a_i \;\Leftrightarrow\; \sum_{S_1} (b_i - \alpha_\iota) \geq 0 \;\Leftrightarrow\; \sum_{S_1} (a_i - b_\iota) \leq 0 \qquad (6.x)$$

Επίσης παρατηρούμε ότι το πλήθος των στοιχείων ενός συνόλου $S_1 \subset S$, θα πρέπει να ισχύει ότι

$$1 \leq |S_1| \leq n - 1$$

Αυτό συμβαίνει διότι ως λύση του προβλήματος αναζητούμε δύο μη κενά και ξένα μεταξύ τους σύνολα που να διαμερίζουν το S. Θέτουμε

$$b = \max_{1 \leq i \leq n} b_i$$

Τώρα βλέπουμε ότι για κάθε $1 \leq i \leq n$

$$b \geq b_i \Leftrightarrow\; b - b_i \geq 0 \Leftrightarrow a_i + b - b_i \geq 0 \Leftrightarrow \sum_{S_1} (\alpha_i - b_i + b) \geq 0 \;\; \forall\, S_1 \qquad (6.y)$$

Ακόμη αν δούμε την τρίτη σχέση στην ισοδυναμία (6.x) μπορούμε να την γράψουμε και ως

$$\sum_{S_1} (\alpha_i - b_i) + |S_1| b \leq |S_1| b \;\Leftrightarrow\; \sum_{S_1} (\alpha_i - b_i + b) \leq |S_1| b \qquad (6.z)$$

Άρα τα σύνολα S_1 για τα οποία ικανοποιείται η παραπάνω ανίσοση, μαζί με το συμπλήρωμα τους $S \backslash S_1$, αποτελούν λύση του προβλήματος Biased Share 2-Partition.

■

6.4.3 Αλγόριθμος Επίλυσης του #Biased Share 2-Partition

Άρα τώρα μπορούμε να δημιουργήσουμε έναν FPTAS αλγόριθμο επίλυσης του #Biased Share 2-Partition, χρησιμοποιώντας τον αλγόριθμο CountMKnapsack.

CountBiasedSharePartition

Input Integers $a_1, \dots, , a_n, b_1, \dots, b_n, n$ **and** $\varepsilon > 0$

1. **Set** $Sa = a_1 + \cdots + a_n$
2. **Set** $Sb = b_1 + \cdots + b_n$
3. **If** $Sa \geq Sb$
4. **Set** $b = \max(b_1, \dots, b_n)$
5. **Set** Y=0
6. **For** $i = 1 \rightarrow n$
7. Set $w_i = a_i - b_i + b$
8. **Endfor**

9. **For** $M = 1 \rightarrow (n-1)$
10. $Y = Y + CountMKnapsack(w_1, \dots, w_n, Mb, M, \varepsilon)$
11. **Endfor**
12. **Else**
13. **Set** $b = \max(a_1, \dots, a_n)$
14. **Set** Y=0
15. **For** $i = 1 \rightarrow n$
16. **Set** $w_i = b_i - a_i + b$
17. **Endfor**
18. **For** $M = 1 \rightarrow (n-1)$
19. $Y = Y + CountMKnapsack(w_1, \dots, w_n, Mb, M, \varepsilon)$
20. **Endfor**
21. **Endif**
22. **Output** Y

Ο αλγόριθμος υπολογίζει αρχικά τα αθροισματα των στοιχείων a_i και b_i αντίστοιχα και αν ισχύει

$$\sum_{i=1}^{n} a_i \geq \sum_{i=1}^{n} b_i$$

Όπως υποθέσαμε και στην (6.u), τότε τρέχουμε τον FPTAS αλγόριθμο για το #M-Element Knapsack, έχοντας για βάρη στοιχεία της μορφής

$$w_i = a_i - b_i + b$$

Που όπως υποδηλώνει η (6.y) είναι θετικοί ακέραιοι αριθμοί. Επίσης αν ικανοποιείται η (6.z) τότε εναλλακτικά μπορούμε να γράψουμε

$$\sum_{S_1} w_i \leq |S_1| b$$

Άρα αναζητούμε όλα τα υποσύνολα $S_1 \subset S$ των οποίων το άθροισμα των w_i δεν ξεπερνά το $|S_1|b$. Το μέγεθος του S_1 μπορεί να είναι από 1 εως $n-1$, έτσι αν τρέχαμε έναν αλγόριθμο που μετρά τις λύσεις του M-Element Knapsack για όλα τα δυνατά M, τότε θα είχαμε τον αριθμό των λύσεων στο πρόβλημα του #Biased Share 2-Partition.

Επειδή ισχύει ο ισχυρισμός 6.2 και επειδή στην πραγματικότητα τρέχουμε ένα FPTAS, τότε όταν ο αλγόριθμος επιστρέφει μία τιμή y, γι αυτήν ισχύει

$$y = y_1 + \cdots + y_{n-1} \leq (1+\varepsilon)Opt_1 + \cdots + (1+\varepsilon)Opt_{n-1} =$$

$$= (1+\varepsilon)Opt$$

Όπου Opt_m είναι η ακριβής λύση του #M-Element Knapsack όταν αναζητούμε το πλήθος των συνόλων με πληθικότητα m και Opt είναι η ακριβής λύση του #Biased Partition Problem.

6.4.4 Χρόνος Εκτέλεσης

Στον αλγόριθμο του #Biased Share 2-Partition θέτουμε κάποια κάποια διανύσματα και υπολογίζουμε κάποια αθροίσματα τα οποία είναι πολυπλοκότητας O(n). Αλλά αυτό που επιρεάζει ασυμπτωτικά την πολυπλοκότητα του FPTAS που κατασκευάσαμε παραπάνω για το #Biased Share 2-Partition είναι το γεγονός ότι τρέχουμε τον FPTAS αλγόριθμο για το #M-Element Knapsack ο οποίος σε μία εκτέλεση έχει πολυπλοκότητα χρόνου $O\left(\frac{n^4}{\varepsilon}\log\frac{n}{\varepsilon}\right)$.

Άρα επειδή εμείς τρέχουμε αυτόν τον αλγόριθμο $n-1$ φορές τότε η χρονική πολυπλοκότητα του FPTAS για το #Biased Share 2-Partition είναι

$$O\left(\frac{n^5}{\varepsilon}\log\frac{n}{\varepsilon}\right)$$

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Για επιπλέον μελέτη υπάρχουν τα παρακάτω ερωτήματα που θέλουμε να θέσουμε με βάση όσα είδαμε στην παρούσα εργασία.

Η εργασία [**33**] ασχολείται με προβλήματα που ανήκουν στην TotP, αλλά δεν μπορούν να προσεγγιστούν οι λύσεις τους με έναν FPRAS αλγόριθμο. Η permanent όπως είδαμε, ανήκει στην TotP και έχει FPRAS αλγόριθμο. Αυτό που μπορούμε να αναρωτηθούμε είναι αν μπορεί να προσεγγιστεί η τιμή της από έναν FPTAS αλγόριθμο. Μια σχετική εργασία που ασχολείται με υποκλάσεις της TotP που έχουν FPRAS ή FPTAS αλγόριθμο είναι η εξής [**34**]. Θα ήταν καλό να δούμε αν η permanent ανήκει σε κάποια από αυτες τις υποκλάσεις ώστε να μπορέσουμε να αποφανθούμε αν υπάρχει FPTAS αλγόριθμος για την permanent.

Ακόμη ένα καλύτερο ερώτημα που μπορούμε να κάνουμε είναι είναι το εξής «ποια είναι η ιδιότητα που πρέπει να έχουν τα προβλήματα, ώστε να επιδέχονται ντετερμινιστικούς αλγορίθμους».

Όσον αφοαρά τον ντετερμιινιστικό προσεγγιστικό αλγόριθμο για την permanent, μπορούμε να συνεχίσουμε τον συλλογισμό μας, και να αναρωτηθούμε, αν τα communities μπορούν τελικά να χρησιμοποιηθούν για να δημιουργηθεί ντετερμινιστικό προσεγγιστικό σχήμα.

Μία ακόμη σκέψη που μπορούμε να κάνουμε είναι η ανάποδη από την παραπάνω. Δηλαδή αν η Permanent μπορεί να αποτελεί κάποιο μέτρο αξιολόγισης των Communities.

Τέλος αξίζει να αναρωτηθούμε αν μπορεί να χρησιμοποιηθεί ο αλγόριθμος του #Biased-Share 2 Partition ώστε να υπολογίσουμε το πλήθος των διαμερίσεων που και οι δύο δικαιούχοι είναι ευχαρηστιμένοι και με τον συνήθη ορισμό του δίκαιου (envy free).

8 ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] John B. Fraleigh, *A first course in Abstract Algebra*.: Addison-Wesley Publishing Company Inc., 1989.

[2] Sheldon Axler, *Linear Algebra Done*. Berlin, New York: Springer-Verlag, 1997.

[3] Ν. Καδιανάκης and Σ. Καρανάσιος, *Γραμμική Άλγεβρα και Αναλυτική Γεωμετρία*. Αθήνα: Καδιανάκης, Ν.; Καρανάσιος, Σ.

[4] Ανάργυρος Γ. Φελούρης, *Γραμμική Άλγεβρα και Αναλτική Γεωμετρία*. Αθήνα: Ανάργυρος Γ. Φελούρης, 2009.

[5] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*. Baltimore and London: The Johns Hopkins University Press, 1996.

[6] Henryk Minc, *Permanents*. Massachusetts: Addison-Wesley, 1978.

[7] L. G. Valiant, "The Complexity of Computing the Permanent," *Theooritical Computer Science 8*, pp. 189-201, 1979.

[8] Aggelos Kiayias, Aris Pagourtzis, Kiron Sharma, and Stathis Zachos, "Acceptor-Definable Counting Classes," *In: Manolopoulos Y., Evripidou S., Kakas A.C. (eds) Advances in Informatics. PCI 2001. Lecture Notes in Computer Science*, vol. 2563, pp. 453-463, Springer 2003.

[9] Aris Pagourtzis and Stathis Zachos, "The Complexity of Counting Functions with Easy Decision Version," in *Proceedings of 31st International Symposium on Mathematical Foundations of Computer Science (MFCS 2006) Lecture Notes in Computer Science 4162*, Springer-Verlag, pp. 741-752.

[10] Αγγελική Χαλκή, *Counting Below #P : Classes, problems and descriptive complexity*. Αθήνα: Διπλωματική ΜΠΛΑ http://mpla.math.uoa.gr/media/theses/msc/A.%20Chalki.pdf, 2016.

[11] E. Allender and M. Ogihara, "Relationships among PL, #L, and the Determininant," *In 9th Annual Structure in Complexity Theory Conference*, pp. 269-279, 1994.

[12] Burgeser Peter, *Completeness and reduction in Algebraic Complexity Theory*. New York: Springer-Verlag Berlin Heidelberg GmbH, 2000.

[13] Manindra Agrawal, "Determinant Versus Permanent," *Mathematics Subject Classification*, p. Primary 68Q17; Secondary 68W30, 2000.

[14] M. Jerrum and M. Snir, "Some exact complexity results for straight-line computations over semirings," *Journal of the ACM*, pp. 29(3):874–897, 1982.

[15] A. Shpilka and A. Wigderson, "Depth-3 arithmetic formulae over fields of charecteristic zero," *In Proceedings of the Conference on Computational Complexity*, pp. pages 79-96, 1999.

[16] D. Grigoriev and A. Razbrov, "Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields," *Applicable Algebra in Engineering, Communication and Computing, 10(6)*, pp. 467-487, 2000.

[17] Ran Raz, "Multi-linear formulas for permanent and determinant and of super-polynomial size," *In Proceedings of Annual ACM Symposium on the Theory of Computing*, pp. 633-641, 2004.

[18] Joseph M. Landsberg and Nicolas Ressayre, "Permanent v. Determinant: An Exponential

Lower Bound Assuming Symmetry," in *ITCS*, Cambridge, MA, USA, 2016, pp. 29-35.

[19] Herbert John Ryser, *Combinatorial Mathematics, The Carus Mathematical Monographs, Vol. 14.*: Mathematical Association of America, 1963, vol. 14.

[20] Mark Jerrum and Alistair Sinclair, "Approximating the Permanent," *SIAM J. COMPUT* , vol. 18, no. 6, pp. 1149-1178, December 1989.

[21] Mark Jerum, Alistair Sinclair, and Eric Vigoda, "A Polynomial-Time Approximation Algorithm for the Permanent with Nonnegative Entries," *Journal of the ACM*, vol. 51, no. 4, pp. 671-697, July 2004.

[22] Michael Chertkov and Adam Yedidia, "Approximating the Permanent with Fractional Belief Propagation," *Journal of Machine Learning Research*, vol. 14, pp. 2029-2066, July 2013.

[23] Cristopher Moore and Alexander Russel, "Approximating the Permanent via nonabelian determinants," *SIAM J. COMPUT.*, vol. 41, no. 2, pp. 332-355, 2012.

[24] John E. Hopcroft and Richard M. Karp, "An n^5/2 algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing*, vol. 2, no. 4, pp. 225-231, 1973.

[25] Mark E. J. Newman, "Modularity and Community Structure in Networks," *PNAS*, vol. 103, no. 23, pp. 8577-8582, June 2006.

[26] Manindra Agrawal, N. Kayal, and N. Saxena, "PRIMES is in P," *Annals of Mathematics 160(2)*, pp. 781-793, 2004.

[27] D. Weitz, "Counting Independent sets up to the three threshold," *In Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 140-149, 2006.

[28] M. Bayati, D. Gamarnik, D. Katz, C. Nair, and P. Tetali, "Simple deterministic approximation algorithms for counting matchings," *In Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 122-127, 2007.

[29] B. Morris and A. Sinclair., "Random walks on truncated cubes and sampling 0-1 knapsack solutions.," *SIAM Journal on Computing 34(1)*, pp. 195-226, 2004.

[30] M. Dyer, "Approximate counting by dynamic programming," *In Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing (STOC)*, pp. 693-699, 2003.

[31] Daniel Stefankovic, Santosh Vempala, and Eric Vigoda, "A Deterministic Polynomial-time Approximation Scheme for Counting Knapsack Solution," *SIAM J Comp*, vol. 41, no. 2, pp. 356-366, 2012.

[32] M. Dyer et al., "A mildly exponential time algorithm for approximating the number of solutions to a multidimensional knapsack problem.," *Combinatorics, Probability and Computing 2(3)*, pp. 271-284, 1993.

[33] Eleni Bakali, Aggeliki Chalki, Aris Pagourtzis, Petros Pantavos, and Stathis Zachos, "Completeness Results for Counting Problems with Easy Decision," *In Proc. of Algorithms and Complexity, 10th International Conference, (CIAC 2017)*, pp. 55-66, 2017.

[34] Evangelos Bampas, Andreas - Nikolas Göbel, Aris Pagourtzis, and Aris Tentes, "On the connection between interval size functions and path," *Computational Complexity*, vol. 26, no. 2, pp. 421-467, 2017.