



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΗ ΕΠΙΣΤΗΜΗ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Προσεγγιστικοί αλγόριθμοι για το Generalized Min-Sum
Set Cover**

Θεόδωρος Α. Δημητρακόπουλος

ΕΠΙΒΛΕΠΩΝ: Σταύρος Κολλιόπουλος, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2014

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Προσεγγιστικοί αλγόριθμοι για το Generalized Min-Sum Set
Cover

Θεόδωρος Α. Δημητρακόπουλος

A.M.: M1220

ΕΠΙΒΛΕΠΩΝ: Σταύρος Κολλιόπουλος, Αναπληρωτής Καθηγητής

**ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: Βασίλειος Ζησιμόπουλος, Καθηγητής ΕΚΠΑ
Σταύρος Κολλιόπουλος, Αναπληρωτής Καθηγητής ΕΚΠΑ**

Ιανουάριος 2014

ΠΕΡΙΛΗΨΗ

Στην παρούσα εργασία, μελετάμε προσεγγιστικούς αλγόριθμους, βασισμένους στο γραμμικό προγραμματισμό, για το Generalized Min-Sum Set Cover ή Multiple Intents Re-Ranking. Το πρόβλημα παρουσιάστηκε για πρώτη φορά από τους Yossi Azar, Iftah Gamzu και Xiaoxin Yin [2]. Θα ασχοληθούμε με τρεις αλγόριθμους για το συγκεκριμένο πρόβλημα εξετάζοντας τους, με την σειρά που αυτοί δημοσιεύτηκαν. Θα δούμε τον τρόπο που βελτιώνονται οι τεχνικές των αλγορίθμων σε σχέση με τους προηγούμενους και πως με αυτόν τον τρόπο επιτυγχάνεται όλο και καλύτερη εγγύηση απόδοσης.

Στο πρώτο κεφάλαιο παρουσιάζουμε κάποιες εισαγωγικές έννοιες, για την πολυπλοκότητα αλγορίθμων, τους προσεγγιστικούς αλγόριθμους και τον γραμμικό προγραμματισμό, απαραίτητες για την σαφή κατανόηση όσων αναφερθούν στην συνέχεια. Στο δεύτερο κεφάλαιο, ασχολούμαστε αναλυτικότερα με το Generalized Min-Sum Set Cover ορίζοντας το τυπικά ως πρόβλημα. Στο τρίτο κεφάλαιο ξεκινάμε μελετώντας τον πρώτο προσεγγιστικό αλγόριθμο των Nikhil Bansal, Anupam Gupta και Ravishankar Krishnaswamy [3] που έχει εγγύηση απόδοσης 485. Στο τέταρτο κεφάλαιο συνεχίζουμε με την δουλειά των Martin Skutella και David P. Williamson [41] που βελτίωσαν τον προηγούμενο αλγόριθμο κατεβάζοντας την εγγύηση απόδοσης από 485 σε 28. Τέλος, στο πέμπτο κεφάλαιο βλέπουμε τον πιο πρόσφατο αλγόριθμο από τους Sungjin Im, Maxim Sviridenko και Ruben van der Zwaan [20] που βελτίωσαν ακόμα περισσότερο την εγγύηση απόδοσης, ρίχνοντάς τη στο 12.4.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Προσεγγιστικοί Αλγόριθμοι

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: γραμμικός προγράμματος, generalized min-sum set cover, εγγύηση απόδοσης, α -points, μητροειδές

ABSTRACT

In this paper, we analyse three approximation algorithms, based on linear programming, for the Generalized Min-Sum Set Cover or Multiple Intents Re-Ranking. The problem was introduced, for the first time, by Yossi Azar, Iftah Gamzu and Xiaoxin Yin [2]. We will deal with three algorithms for this problem, examining them in the order in which they were published. We will perceive how, compared to the previous one, each algorithm has improved technically and how, each time, a better performance guarantee is achieved.

In the first chapter some introductory concepts are presented –concerning the algorithms' complexity, approximation algorithms and the linear programming– which are necessary for the distinct understanding of those further down. The second chapter is dealing, extensively, with the Generalized Min-Sum Set Cover, defining it as a problem. In the third chapter, we begin the study of the first approximate algorithm of Nikhil Bansal, Anupam Gupta and Ravishankar Krishnaswamy [3], which has a 485 performance guarantee. In the fourth chapter we continue with the work of Martin Skutella and David p. Williamson [41] that improved the previous algorithm by lowering the performance guarantee from 485 to 28. Finally, in the fifth chapter, we see the latest algorithm, from the Sungjin Im, Maxim Sviridenko and Ruben van der Zwaan [20], that improved the performance guarantee even more, dropping it to 12.4.

SUBJECT AREA: Approximation Algorithms

KEYWORDS: linear programming, generalized min-sum set cover, performance guarantee, α -points, matroid

Στην Ντίνα, τον Τούλη και την Ερατώ.

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ θερμά όλους όσους συνέβαλαν στην υλοποίηση της παρούσας διπλωματικής εργασίας. Κυρίως, θα ήθελα να απευθύνω τις ευχαριστίες μου στον καθηγητή κ. Σταύρο Κολλιόπουλο για την πολύτιμη βοήθεια την οποία μου πρόσφερε μέσα από την καθοδήγηση του και την εμπιστοσύνη που μου έδειξε, καθώς και για την διαρκή υπομονή του έως ότου η εργασία ολοκληρωθεί. Επίσης, θα ήθελα να αποδώσω, μαζί με τις ευχαριστίες, το σεβασμό μου στα μέλη της οικογενείας μου. Ανέκαθεν με στήριξαν, μεριμνώντας για την καλύτερη δυνατή μόρφωση που θα μπορούσα να λάβω. Τέλος, ένα μεγάλο ευχαριστώ στους φίλους συμφοιτητές μου, που απέκτησα κατά την διάρκεια των σπουδών.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ	10
Πολυπλοκότητα αλγορίθμων	10
Ασυμπτωτική ανάλυση	10
Αποδοτικοί και μη αλγόριθμοι	11
Υπολογιστική πολυπλοκότητα	11
Προβλήματα βελτιστοποίησης και προβλήματα απόφασης	12
<i>P</i> και <i>NP</i>	13
Αναγωγές και Πληρότητα	13
Προσεγγιστικοί αλγόριθμοι	14
Δυσκολία προσέγγισης (hardness of approximation)	16
Πιθανοτικοί προσεγγιστικοί αλγόριθμοι	16
Πρόβλημα γραμμικού προγραμματισμού	16
Μητροειδή (matroids)	18
Για περαιτέρω ανάγνωση	20
ΤΟ ΠΡΟΒΛΗΜΑ	21
Set Cover και Hitting Set	21
Min-Sum Set Cover και Min-Latency Set Cover	23
Generalized Min-Sum Set Cover	25
Preemptive Generalized Min-Sum Set Cover	26
Ο ΠΡΩΤΟΣ ΑΛΓΟΡΙΘΜΟΣ ΜΕ ΣΤΑΘΕΡΗ ΕΓΓΥΗΣΗ ΑΠΟΔΟΣΗΣ	28
Χαλαρωμένα γραμμικά προγράμματα (LP relaxations)	28
Separation oracle (μαντείο διαχωρισμού)	31
Ο αλγόριθμος	31
Ανάλυση του αλγορίθμου	32

Ο ΔΕΥΤΕΡΟΣ ΑΛΓΟΡΙΘΜΟΣ	36
Ο αλγόριθμος	36
Ανάλυση του αλγορίθμου	36
Ο ΤΡΙΤΟΣ ΑΛΓΟΡΙΘΜΟΣ	43
Ένας 2-προσεγγιστικός αλγόριθμος για το Preemptive Generalized Min-Sum Set Cover	43
Από το preemptive στο non-preemptive	61
ΣΥΜΠΕΡΑΣΜΑΤΑ	71
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	73
ΣΥΝΤΜΗΣΕΙΣ, ΑΡΚΤΙΚΟΛΕΞΑ ΚΑΙ ΑΚΡΩΝΥΜΙΑ	74
ΑΝΑΦΟΡΕΣ	75

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1:	Στο διμερές γράφημα παρατηρούμε τη σχέση μεταξύ Set Cover και Hitting Set.	23
Σχήμα 2:	Στην πρώτη γραμμή, έχουμε την μη ακέραια διάταξη. Στη δεύτερη γραμμή, την ακέραια διάταξη που κατασκευάστηκε από τις μεταθέσεις $F_{1,1}, F_{2,2}, F_{3,3}$. Στη τρίτη την διάταξη που κατασκευάστηκε με τις $F_{3,1}, F_{1,2}, F_{2,3}$. Και τέλος, στην τέταρτη, την διάταξη που κατασκευάστηκε με τις $F_{2,1}, F_{3,2}, F_{1,3}$	47
Σχήμα 3:	Αναπαράσταση του γραφήματος G στον οποίο θέλουμε να βρούμε την ροή $s-t$ με το μέγιστο κόστος.	54
Σχήμα 4:	Τα βήματα του αλγορίθμου.	59
Σχήμα 5:	Γραφική αναπαράσταση του $\tilde{C}_S(\lambda)$. Το j -οστό ορθογώνιο αντιστοιχεί στη μετάθεση F_j , έχει μήκος $y_S^{F_j}$ και ύψος $C_S^{F_j}$	60
Σχήμα 6:	Γραφική απεικόνιση της κατασκευής της διάταξης \tilde{x} από την x , με παραμέτρους $r = 2$, $\rho = 1$ και $Q = 2$ και α -τιμές $\alpha_a = \frac{1}{2}$, $\alpha_b = \frac{1}{4}$, $\alpha_c = 1$ και $\alpha_d = \frac{1}{2}$	64

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό γίνεται συνοπτική παρουσίαση ορισμένων εισαγωγικών εννοιών, που αφορούν την πολυπλοκότητα αλγορίθμων, την υπολογιστική πολυπλοκότητα, το γραμμικό προγραμματισμό, τους προσεγγιστικούς αλγορίθμους, καθώς και άλλων εννοιών απαραίτητων για την κατανόηση της εργασίας.

Πολυπλοκότητα αλγορίθμων

Κάθε αλγόριθμος απαιτεί πόρους. Για ένα πρόβλημα είμαστε σε θέση να κατασκευάσουμε πολλούς αλγορίθμους. Καθένας από τους αλγορίθμους έχει διαφορετικές απαιτήσεις, δεν είναι όμως, βάσει των πόρων που χρησιμοποιούν, όλοι εξίσου αποδοτικοί. Για να ταξινομήσουμε την αποδοτικότητα ενός αλγορίθμου επιλέγουμε σαν μέτρο σύγκρισης τους πόρους, που είναι ανεξάρτητοι γλώσσας προγραμματισμού και υπολογιστικής μηχανής. Οι πόροι αυτοί είναι ο αριθμός των εντολών που εκτελεί ο αλγόριθμος (βήματα ή χρόνος εκτέλεσης) και ο αριθμός των μονάδων μνήμης (χώρος), συνήθως των bits, που χρησιμοποιεί κατά την εκτέλεση του.

Το πλήθος των εντολών που θα εκτελέσει ένας αλγόριθμος για συγκεκριμένη είσοδο εξαρτάται από πολλούς παράγοντες. Για παράδειγμα, τα βήματα που θα κάνει ένας αλγόριθμος για την εύρεση του κυρτού περιβλήματος ενός συνόλου σημείων στο χώρο μπορούν να εξαρτώνται από το πως είναι κατανομημένα αυτά τα σημεία στο χώρο. Ωστόσο, υπολογίζουμε το χρόνο εκτέλεσης ενός αλγορίθμου μόνο συναρτήσει του αριθμού των bits που χρειάστηκαν για την αποθήκευση των δεδομένων της εισόδου.

Ορισμός 1.1 (Μέγεθος ή μήκος εισόδου ενός αλγορίθμου). *Αλγόριθμος A με είσοδο I . Καλούμε μέγεθος εισόδου του A , που συμβολίζεται με $|I|$, τον αριθμό των bits χρειάζονται για αποθηκεύσουμε την είσοδο I χρησιμοποιώντας δυαδική κωδικοποίηση.*

Ακόμα και για διαφορετικές εισόδους με ίδιο μέγεθος, οι χρόνοι εκτέλεσης ενός αλγορίθμου μπορεί να ποικίλουν. Τις περισσότερες φορές, μας ενδιαφέρει η χειρότερη περίπτωση όταν, δηλαδή, ο αλγόριθμος έκανε τον περισσότερο χρόνο για να εκτελεστεί, καθώς είναι ένα άνω φράγμα για το πως συμπεριφέρεται ο αλγόριθμος για οποιαδήποτε είσοδο.

Ορισμός 1.2 (Χρονική Πολυπλοκότητα). *Χρονική πολυπλοκότητα ενός αλγορίθμου είναι μια συνάρτηση $f(n)$ η οποία επιστρέφει το πλήθος των εντολών που εκτελεί ο αλγόριθμος, στη χειρότερη περίπτωση, όταν έχει είσοδο μεγέθους n .*

Ορισμός 1.3 (Πολυπλοκότητα Χώρου). *Πολυπλοκότητα χώρου ενός αλγορίθμου είναι μια συνάρτηση $f(n)$ η οποία επιστρέφει τον αριθμό των bits μνήμης που χρησιμοποιεί ο αλγόριθμος, στη χειρότερη περίπτωση, όταν έχει είσοδο μεγέθους n .*

Ασυμπτωτική ανάλυση

Είναι γεγονός ότι οι συναρτήσεις που εκφράζουν τον ακριβή χρόνο εκτέλεσης ενός αλγορίθμου είναι συχνά πολύπλοκες. Συνήθως, μας αρκεί απλώς μια εκτίμηση του χρόνου εκτέλεσης για μεγάλες εισόδους. Για αυτό το λόγο, εφαρμόζουμε την ασυμπτωτική ανά-

λυση, η οποία μας δείχνει πως μεταβάλλεται η τιμή της συνάρτησης για μεγάλες τιμές εισόδου.

Σε μια συνάρτηση $f(n)$ που αντιπροσωπεύει το χρόνο εκτέλεσης ενός αλγορίθμου, για παράδειγμα, αν $f(n) = 2n^4 - 5n^2 + n + 7$ λέμε ότι $f(n) = O(n^4)$. Ο συμβολισμός O καλείται *big O*, κεφαλαίο όμικρον. Δείχνει ότι όσο μεγαλώνει η τιμή του n , η f δεν θα ξεπεράσει τη n^4 ή όπως λέμε η f φράσσεται από την n^4 .

Ορισμός 1.4 (*big O*, κεφαλαίο όμικρον). $f(n) = O(g(n))$ σημαίνει ότι υπάρχουν θετικές σταθερές c και n_0 τέτοιες ώστε, για κάθε ακέραιο $n \geq n_0$, $f(n) \leq cg(n)$.

Υπάρχουν και άλλοι ασυμπτωτικοί συντελεστές

Ορισμός 1.5 (*big Ω* , κεφαλαίο ωμέγα). $f(n) = \Omega(g(n))$ σημαίνει ότι υπάρχουν θετικές σταθερές c και n_0 τέτοιες ώστε, για κάθε ακέραιο $n \geq n_0$, $f(n) \geq cg(n)$.

Ορισμός 1.6 (*big Θ* , κεφαλαίο θήτα). $f(n) = \Theta(g(n))$ σημαίνει ότι υπάρχουν θετικές σταθερές c_1, c_2 και n_0 τέτοιες ώστε για κάθε ακέραιο $n \geq n_0$, $c_1g(n) \geq f(n) \geq c_2g(n)$.

Από τους ορισμούς βλέπουμε ότι η $f(n)$ του παραπάνω παραδείγματος είναι $f(n) = \Theta(n^4)$ αφού $f(n) = O(n^4)$ αλλά και $f(n) = \Omega(n^4)$.

Αποδοτικοί και μη αλγόριθμοι

Για να αξιολογήσουμε ποιοι αλγόριθμοι είναι αποδοτικοί ή μη, έχουμε τον εξής ορισμό.

Ορισμός 1.7 (Αποδοτικός αλγόριθμος). Ένας αλγόριθμος με χρόνο εκτέλεσης στη χειρότερη περίπτωση $f(n)$ θεωρούμε ότι είναι αποδοτικός αν:

$$f(n) = O(n^k) \quad \text{όπου } k \text{ σταθερά με } k > 0.$$

Αν διαθέτει, δηλαδή, αυτό που ονομάζουμε πολυωνυμική πολυπλοκότητα. Μερικά παραδείγματα πολυωνυμικής πολυπλοκότητας είναι τα εξής: \sqrt{n} , n , $n \log n$, n^2 , $n^2 \log n$, n^3 , ... Απεναντίας όσοι αλγόριθμοι έχουν υποεκθετική πολυπλοκότητα $n^{\log n}$, εκθετική πολυπλοκότητα e^n , 2^n ή ακόμα χειρότερα υπερεκθετική πολυπλοκότητα $n!$, n^n θεωρούνται ασύμφοροι.

Το πρόβλημα με τους συγκεκριμένους είναι ότι για μεγάλες εισόδους η εκτέλεση τους παίρνει υπερβολικό χρόνο. Για παράδειγμα ένας αλγόριθμος με χρόνο εκτέλεσης $O(n!)$ για $n = 100$ θα εκτελέσει 100! βήματα, που είναι ένας αστρονομικά μεγάλος αριθμός. Ακόμα και ο πιο γρήγορος υπολογιστής θα χρειαζόταν υπερβολικά μεγάλο χρονικό διάστημα για να τον εκτελέσει. Είναι προφανές, λοιπόν, ότι αλγόριθμοι σαν και αυτόν θα πρέπει να αποφεύγονται.

Υπολογιστική πολυπλοκότητα

Υπάρχουν προβλήματα για τα οποία έχουμε κατασκευάσει αποδοτικούς, πολυωνυμικούς αλγόριθμους και προβλήματα για τα οποία, μέχρι τώρα, δεν έχουμε καταφέρει να κατα-

σκευάσουμε. Πάνω σε αυτό προκύπτει το ερώτημα: είμαστε σε θέση να κατασκευάσουμε κάποιον αποδοτικό αλγόριθμο για κάθε πρόβλημα ή υπάρχουν κάποια προβλήματα για τα οποία κάθε απόπειρα εύρεσης γρήγορου αλγόριθμου θα αποτύχει?

Με το συγκεκριμένο ερώτημα ασχολείται η υπολογιστική πολυπλοκότητα, η οποία κατηγοριοποιεί τα προβλήματα σε κλάσεις, τις λεγόμενες κλάσεις πολυπλοκότητας. Για ορισμένες από αυτές τις κλάσεις ορίζει προβλήματα-αντιπροσώπους. Το πρόβλημα-αντιπρόσωπος εκφράζει όλη την κλάση, μελετώντας την συμπεριφορά του βγάζουμε συμπεράσματα σχετικά με όλα τα προβλήματα της κλάσης.

Οι κλάσεις πολυπλοκότητας με το μεγαλύτερο πρακτικό ενδιαφέρον είναι η P και η NP , οι οποίες για τυπικούς λόγους έχουμε ορίσει να περιέχουν προβλήματα απόφασης. Για καλύτερη κατανόηση, επομένως, των κλάσεων P και NP είναι αναγκαία μια αναφορά στα προβλήματα απόφασης καθώς και σε μια άλλη κατηγορία προβλημάτων, τα προβλήματα βελτιστοποίησης.

Πρόβλημα βελτιστοποίησης και προβλήματα απόφασης

Σε ένα πρόβλημα βελτιστοποίησης αναζητούμε την βέλτιστη λύση (ελάχιστη ή μέγιστη) μεταξύ του συνόλου των δυνατών λύσεων. Αν ζητάμε την ελάχιστη λύση τότε λέμε ότι το πρόβλημα είναι πρόβλημα ελαχιστοποίησης ενώ αν ζητάμε τη μέγιστη τότε λέμε ότι είναι πρόβλημα μεγιστοποίησης. Για να βρούμε τη βέλτιστη λύση ορίζουμε μια συνάρτηση f (συνάρτηση κόστους) η οποία αντιστοιχεί σε κάθε λύση ένα κόστος. Η βέλτιστη λύση είναι αυτή στην οποία η συνάρτηση παρουσιάζει ολικό ακρότατο.

Ορισμός 1.8 (Συνάρτηση κόστους). *Πρόβλημα Π με σύνολο D_Π που περιέχει όλες τις πιθανές εισόδους (στιγμιότυπα) για αυτό το πρόβλημα. Για κάθε στιγμιότυπο $I \in D_\Pi$ υπάρχει ένα σύνολο από λύσεις $S(I)$. Ορίζουμε τη συνάρτηση κόστους $f(I, s)$ που για συγκεκριμένο στιγμιότυπο $I \in D_\Pi$ και λύση $s \in S(I)$ επιστρέφει το κόστος της s που είναι ένας θετικός πραγματικός αριθμός.*

Ορισμός 1.9 (Πρόβλημα βελτιστοποίησης). *Σε ένα πρόβλημα βελτιστοποίησης (ελαχιστοποίησης ή μεγιστοποίησης) με είσοδο I και συνάρτηση κόστους $f(I, s)$, ζητείται λύση $s' \in S(I)$, την οποία συμβολίζουμε με $OPT(I)$, τέτοια ώστε $f(I, s') = \min_s \{f(I, s)\}$ αν πρόκειται για πρόβλημα ελαχιστοποίησης ή $f(I, s') = \max_s \{f(I, s)\}$ αν πρόκειται για πρόβλημα μεγιστοποίησης.*

Μια άλλη κατηγορία προβλημάτων είναι τα προβλήματα απόφασης (ύπαρξης), σε αυτά θέτουμε ένα ερώτημα και η απάντηση που παίρνουμε είναι ΝΑΙ ή ΟΧΙ ή εναλλακτικά 1 ή 0. Σε κάθε πρόβλημα απόφασης αντιστοιχίζουμε ένα σύνολο από αποδεκτές λύσεις. Ένας αλγόριθμος που επιλύει ένα τέτοιο πρόβλημα, ελέγχει αν η είσοδο που παίρνει, ανήκει στο σύνολο των αποδεκτών λύσεων αυτού του προβλήματος. Αν ανήκει θα απαντήσει στην έξοδο ΝΑΙ ενώ σε αντίθετη περίπτωση ΟΧΙ.

Ορισμός 1.10 (Πρόβλημα απόφασης). *Σε ένα πρόβλημα απόφασης Π με είσοδο το στιγμιότυπο x , καλούμαστε να αποφασίσουμε αν $x \in S_\Pi$ όπου το S_Π είναι το σύνολο όλων των*

αποδεκτών λύσεων του Π .

Σε κάθε πρόβλημα ελαχιστοποίησης (μεγιστοποίησης) Π_0 αντιστοιχεί ένα πρόβλημα απόφασης Π_0 . Αυτό συμβαίνει αν στην είσοδο του προβλήματος βελτιστοποίησης μαζί με το στιγμιότυπο I προσθέσουμε και έναν αριθμό $K \in \mathbb{N}$ και ζητάμε στην έξοδο αν υπάρχει s' έτσι ώστε $f(I, s') \leq K$. Ρωτάμε, δηλαδή, αν υπάρχει λύση με κόστος το πολύ K (ή $f(I, s') \geq K$ για πρόβλημα μεγιστοποίησης).

P* και *NP

Αφού ορίσαμε τα προβλήματα απόφασης συνεχίζουμε ορίζοντας τις κλάσεις P και NP .

Ορισμός 1.11 (Κλάση P). *Η κλάση P περιέχει όλα τα προβλήματα απόφασης, τα οποία επιλύονται με πολυωνυμικό αλγόριθμο.*

Ορισμός 1.12 (Κλάση NP). *Καλούμε πιστοποιητικό y μια απόδειξη (με μέγεθος πολυωνυμικό σε σχέση με το μέγεθος της εισόδου) ότι μια συγκεκριμένη είσοδος x ενός προβλήματος απόφασης Π θα είναι αποδεκτή ($x \in S_\Pi$). Η κλάση NP περιέχει όλα τα προβλήματα απόφασης, για τα οποία, αν μας δώσουν ένα τέτοιο ζευγάρι (y, x) , είμαστε σε θέση να επαληθεύσουμε, σε πολυωνυμικό χρόνο, αν όντως η είσοδος x θα είναι αποδεκτή.*

Είναι προφανές ότι $P \subseteq NP$, εφόσον η επαλήθευση ενός προβλήματος που ανήκει στην κλάση P μπορεί να γίνει με τον ίδιο αλγόριθμο που επιλύει το πρόβλημα, καθώς ο αλγόριθμος είναι πολυωνυμικός.

Ωστόσο, παραμένει ανοικτό ερώτημα αν $P = NP$ ή $P \neq NP$. Το συγκεκριμένο ερώτημα είναι ένα από τα σημαντικότερα, αν όχι το σημαντικότερο ανοικτό πρόβλημα, της επιστήμης των υπολογιστών και αυτό γιατί η κλάση NP περιέχει και προβλήματα για τα οποία δεν έχουμε καταφέρει ακόμα να κατασκευάσουμε πολυωνυμικό αλγόριθμο. Για αυτά τα προβλήματα της κλάσης NP , (πολλά από τα οποία έχουν πρακτικό ενδιαφέρον, αφού συναντάμε, τα ίδια ή παραλλαγές τους, πολύ συχνά στη πληροφορική και στη καθημερινή ζωή) θα θέλαμε να γνωρίζουμε αν το ότι δεν έχουμε κατασκευάσει πολυωνυμικό αλγόριθμο οφείλεται σε δική μας αδυναμία ή στο ότι για τα συγκεκριμένα προβλήματα δεν είναι δυνατόν να κατασκευαστεί ένας τέτοιος αλγόριθμος.

Αν $P = NP$, τότε οποιοδήποτε πρόβλημα μπορούμε να επαληθεύσουμε σε πολυωνυμικό χρόνο μπορούμε και να το επιλύσουμε σε πολυωνυμικό χρόνο. Ενώ εάν $P \neq NP$, για μια μεγάλη κατηγορία προβλημάτων της κλάσης NP για τα οποία δεν έχουμε ακόμα κάποιον πολυωνυμικό αλγόριθμο, τα NP -complete, θα γνωρίζουμε με βεβαιότητα ότι δεν μπορεί να κατασκευαστεί πολυωνυμικός αλγόριθμος που να τα επιλύει.

Αναγωγές και Πληρότητα

Όταν χρησιμοποιούμε έναν αλγόριθμο που λύνει ένα πρόβλημα A για να λύσουμε ένα άλλο πρόβλημα B , κάνουμε αναγωγή. Παραδείγματος χάριν, έστω ότι έχουμε ένα αλγόριθμο που κάνει μόνο πρόσθεση $a + b$. Αν θέλουμε να κάνουμε την αφαίρεση $a - b$, μπορούμε να

αλλάξουμε το πρόσημο και να προσθέσουμε $a + (-b)$. με αυτό τον τρόπο κάναμε αναγωγή από την αφαίρεση στην πρόσθεση, την οποία συμβολίζουμε με *αφαίρεση* \propto *πρόσθεση*.

Ορισμός 1.13 (Αναγωγή πολυωνυμικού χρόνου). Έχουμε αναγωγή πολυωνυμικού χρόνου από ένα πρόβλημα A στο πρόβλημα B και τη συμβολίζουμε με $A \propto B$ αν υποθέτοντας ότι υπάρχει αλγόριθμος που επιλύει το B τότε μπορούμε να κατασκευάσουμε πολυωνυμικό χρόνο αλγόριθμο που επιλύει το A .

Ένα complete πρόβλημα (πλήρες, αντιπρόσωπος κλάσης) έχει το εξής χαρακτηριστικό που το καθιστά σημαντικό. Ο αλγόριθμος για την επίλυση αυτού του προβλήματος μπορεί να χρησιμοποιηθεί για την επίλυση όλων των προβλημάτων της κλάσης. Υπάρχει, δηλαδή, αναγωγή από κάθε πρόβλημα αυτής της κλάσης στο πλήρες πρόβλημα.

Ορισμός 1.14 (C -complete (C -πλήρες)). Πρόβλημα Π και C κλάση πολυπλοκότητας, λέμε ότι Π είναι C -complete αν:

1. $\Pi \in C$
2. Για κάθε πρόβλημα $L \in C$ έχουμε αναγωγή πολυωνυμικού χρόνου $\Pi \propto C$

Αν για το Π ισχύει μόνο το 2 τότε λέμε ότι το Π είναι C -hard (C -δύσκολο).

Για την κλάση NP έχουμε βρει πολλά NP -complete προβλήματα. Σχεδόν όλα τα προβλήματα απόφασης για τα οποία δεν έχουμε καταφέρει ακόμα να βρούμε πολυωνυμικό αλγόριθμο έχουμε δείξει ότι είναι NP -complete. Αν καταφέρουμε να βρούμε έστω για ένα από αυτά τα προβλήματα πολυωνυμικό αλγόριθμο, θα μπορούσαμε χρησιμοποιώντας τον να κατασκευάσουμε πολυωνυμικό αλγόριθμο για όλα τα προβλήματα της κλάσης NP . Θα είχαμε αποδείξει ότι $P = NP$.

Επειδή όμως παραμένει ανοιχτό ερώτημα αν $P = NP$, αν για ένα πρόβλημα δεν μπορούμε να βρούμε ένα γρήγορο αλγόριθμο, μπορούμε να προσπαθήσουμε να δείξουμε ότι το πρόβλημα είναι C -complete ή C -hard. Οπότε, θεωρώντας ότι $P \neq NP$, μπορούμε να εγκαταλείψουμε την προσπάθεια εύρεσης πολυωνυμικού αλγορίθμου.

Προσεγγιστικοί αλγόριθμοι

Δεδομένου ότι $P \neq NP$, αν ένα πρόβλημα βελτιστοποίησης είναι NP -hard, η εύρεση της βέλτιστης λύσης απαιτεί εξαντλητικό ψάξιμο του χώρου όλων των δυνατών λύσεων και είναι υπερβολικά χρονοβόρα. Στην πράξη, όμως, η βέλτιστη λύση μπορεί να μην είναι πάντοτε απαραίτητη. Μπορεί να μας αρκεί μια λύση που να προσεγγίζει τη βέλτιστη, την οποία μπορούμε να βρούμε γρήγορα. Για αυτό το λόγο έχουμε τους προσεγγιστικούς αλγορίθμους, αλγορίθμους γρήγορους, οι οποίοι βρίσκουν αποδεδειγμένα λύσεις κοντά στη βέλτιστη. Είμαστε σε θέση να ορίσουμε το πόσο κοντά είναι η εκάστοτε λύση που βρήκαμε στη βέλτιστη και αυτό αποτελεί το κριτήριο για το πόσο καλός είναι ένας προσεγγιστικός αλγόριθμος.

Το μέσο που έχουμε ορίσει ώστε να μετράμε πόσο καλά προσεγγίζουν οι λύσεις, τις οποίες επιστρέφει ένας αλγόριθμος, τη βέλτιστη λύση είναι η εγγύηση απόδοσης (performance

guarantee).

Ορισμός 1.15 (Απόλυτη εγγύηση απόδοσης (absolute performance guarantee)). Ένας A_{Π} αλγόριθμος, για ένα πρόβλημα ελαχιστοποίησης (μεγιστοποίησης) Π με σύνολο στιγμιοτύπων D_{Π} , έχει απόλυτη εγγύηση απόδοσης $c \in \mathbb{R}^+$ αν για κάθε είσοδο $I \in D_{\Pi}$ επιστρέφει στην έξοδο λύση s με κόστος $A(I)$ τέτοιο ώστε $|A(I) - OPT(I)| \leq c$ (ή $|A(I) - OPT(I)| \geq c$ αν το πρόβλημα είναι μεγιστοποίησης).

Η απόλυτη εγγύηση απόδοσης, αν και χρησιμοποιείτε για ορισμένα προβλήματα, είναι ακατάλληλη για τα περισσότερα NP -hard προβλήματα. Ως επί το πλείστον, χρησιμοποιούμε την σχετική εγγύηση απόδοσης.

Ορισμός 1.16 (Εγγύηση απόδοσης (performance guarantee) ή σχετική εγγύηση απόδοσης (relative performance guarantee)). Ένας A_{Π} αλγόριθμος, για ένα πρόβλημα ελαχιστοποίησης (μεγιστοποίησης) Π με σύνολο στιγμιοτύπων D_{Π} , έχει εγγύηση απόδοσης ρ αν για κάθε είσοδο $I \in D_{\Pi}$ επιστρέφει στην έξοδο λύση s με κόστος $A(I)$ τέτοιο ώστε $\frac{A(I)}{OPT(I)} \leq \rho$ (ή $\frac{A(I)}{OPT(I)} \geq \rho$ αν το πρόβλημα είναι μεγιστοποίησης).

Για ένα πρόβλημα ελαχιστοποίησης $\rho \geq 1$ ενώ για ένα πρόβλημα μεγιστοποίησης $\rho \in [0, 1]$. Το ρ μας δείχνει πόσο καλά ο αλγόριθμος προσεγγίζει τη βέλτιστη λύση. Όσο πιο κοντά είναι το ρ στο 1 τόσο καλύτερη είναι η προσέγγιση.

Ορισμός 1.17 (ρ -προσεγγιστικός αλγόριθμος). Έστω A_{Π} αλγόριθμος πολυωνυμικής πολυπλοκότητας, για ένα πρόβλημα ελαχιστοποίησης (μεγιστοποίησης) Π με σύνολο στιγμιοτύπων D_{Π} . Για ένα πραγματικό αριθμό $\epsilon > 0$ ορίζουμε $\rho = 1 + \epsilon$ (ή $\rho = 1 - \epsilon$ αν το πρόβλημα είναι μεγιστοποίησης). Λέμε ότι ο A_{Π} είναι ρ -προσεγγιστικός αλγόριθμος αν για κάθε είσοδο $I \in D_{\Pi}$ επιστρέφει στην έξοδο λύση s με κόστος $A(I)$ τέτοιο ώστε $|A(I) - OPT(I)| \leq \epsilon OPT(I)$

Το ρ είναι εγγύηση απόδοσης (performance guarantee) για τον αλγόριθμο A_{Π} .

Ορισμός 1.18 (Κλάση APX). Η κλάση πολυπλοκότητας APX περιέχει όλα τα προβλήματα βελτιστοποίησης που έχουν πολυωνυμικούς προσεγγιστικούς αλγόριθμους με σταθερή εγγύηση απόδοσης (performance guarantee).

Ορισμός 1.19 (Προσεγγιστικό σχήμα (approximation scheme)). Για ένα πρόβλημα ελαχιστοποίησης (μεγιστοποίησης) Π , προσεγγιστικό σχήμα είναι ένα σύνολο προσεγγιστικών $(1 + \epsilon)$ -αλγορίθμων A_{ϵ} (ή $(1 - \epsilon)$ -αλγορίθμων A_{ϵ} για πρόβλημα μεγιστοποίησης) ένας για κάθε ϵ με $0 < \epsilon < 1$.

Έχουμε και ένα άλλο ισοδύναμο ορισμό για το προσεγγιστικό σχήμα.

Ορισμός 1.20 (Προσεγγιστικό σχήμα (approximation scheme)). Προσεγγιστικό σχήμα καλούμε ένα αλγόριθμο A , για ένα πρόβλημα ελαχιστοποίησης (μεγιστοποίησης) Π με σύνολο στιγμιοτύπων D_{Π} , που δέχεται στην είσοδο $I \in D_{\Pi}$ και ϵ και επιστρέφει λύση s με κόστος $(1 + \epsilon)OPT(I)$ (ή $(1 - \epsilon)OPT(I)$ για πρόβλημα μεγιστοποίησης) μια για κάθε $0 < \epsilon < 1$.

Ορισμός 1.21 ($PTAS$ (polynomial-time approximation scheme)). Για ένα πρόβλημα βελτιστοποίησης Π , $PTAS$ είναι ένα προσεγγιστικό σχήμα του οποίου οι αλγόριθμοι A_{ϵ} τρέχουν σε χρόνο πολυωνυμικό στο μέγεθος τις εισόδου για κάθε σταθερό ϵ .

Ένα πρόβλημα που έχει *PTAS* μπορούμε, μέσα σε πολυωνυμικό χρόνο, να το προσεγγίσουμε με όση ακρίβεια επιθυμούμε, ωστόσο αυτό ενδέχεται να έχει ως συνέπεια τής αύξηση του χρόνου εκτέλεσης. Ορισμένες φορές μάλιστα ο χρόνος εκτέλεσης μπορεί να αυξηθεί υπερβολικά. Ένα *PTAS*, παραδείγματος χάριν, με χρόνο εκτέλεσης $O(n^{2/\epsilon})$ αν και τρέχει σε πολυωνυμικό χρόνο στο n είναι εκθετικός στο $\frac{1}{\epsilon}$ και για ϵ κοντά στη μονάδα ο εκθέτης του πολυωνύμου γίνεται υπερβολικά μεγάλος. Για αυτό το λόγο έχουμε ορίσει τα *FPTAS*.

Ορισμός 1.22 (*FPTAS* (fully polynomial-time approximation scheme)). Για ένα πρόβλημα βελτιστοποίησης Π , *FPTAS* είναι ένα προσεγγιστικό σχήμα του οποίου οι αλγόριθμοι τρέχουν σε χρόνο πολυωνυμικό στο μέγεθος τις εισόδου και στο $\frac{1}{\epsilon}$ για κάθε σταθερό ϵ .

Δυσκολία προσέγγισης (hardness of approximation)

Πολλά προβλήματα είναι *NP-hard* ακόμα και να τα προσεγγίσουμε πέρα από μια συγκεκριμένη εγγύηση απόδοσης (όχι απαραίτητα σταθερή). Δηλαδή όχι μόνο δεν υπάρχει αποδοτικός αλγόριθμος για πρόβλημα αλλά ούτε και προσεγγιστικός αλγόριθμος με καλύτερη εγγύηση απόδοσης εκτός εάν $P = NP$. Ωστόσο πολλές αποδείξεις για την δυσκολία προσέγγισης βασίζονται σε άλλες υποθέσεις, μια υπόθεση που έχει χρησιμοποιηθεί εκτενώς είναι η Unique Games Conjecture (UGC) [24].

Στο πρόβλημα Unique Games έχουμε περιορισμούς της μορφής $x_i - x_j \equiv c_{ij} \pmod k$ και θέλουμε να βρούμε τα x που ικανοποιούν τους περισσότερους. Η UGC ισχυρίζεται πως όταν έχουμε εισόδους με πολύ μεγάλα k , για κάθε $\epsilon > 0$ είναι *NP-hard* ξεχωρίζουμε αν υπάρχει μια ανάθεση τιμών στα x που να ικανοποιεί το $(100 - \epsilon)\%$ των περιορισμών ή αν δεν υπάρχει ούτε μια ανάθεση τιμών που να ικανοποιεί περισσότερο από το $\epsilon\%$ των περιορισμών.

Πιθανοτικοί προσεγγιστικοί αλγόριθμοι

Ένας πιθανοτικός αλγόριθμος, εκτός από την κύρια είσοδο, δέχεται επιπλέον, σαν βοηθητική είσοδο μια ακολουθία από τυχαία bits τα οποία χρησιμοποιεί ώστε να πάρει οριζόμενες τυχαίες αποφάσεις κατά την διάρκεια της εκτέλεσης του. Έτσι, διαδοχικές εκτελέσεις ενός τέτοιου αλγορίθμου ενδέχεται να έχουν διαφορετική συμπεριφορά και έξοδο ακόμα και για την ίδια κύρια είσοδο. Σε ένα πιθανοτικό προσεγγιστικό αλγόριθμο ο χρόνος εκτέλεσης και η εγγύηση απόδοσης γίνονται τυχαίες μεταβλητές και για να εξάγουμε συμπεράσματα είναι απαραίτητο να κατανοήσουμε τις κατανομές αυτών των τυχαίων μεταβλητών.

Πρόβλημα γραμμικού προγραμματισμού

Το πρόβλημα του γραμμικού προγραμματισμού (Linear Programming LP) είναι ένα πρόβλημα βελτιστοποίησης. Θέλουμε να υπολογίσουμε το διάνυσμα x που μεγιστοποιεί την συνάρτηση κόστους z (αντικειμενική συνάρτηση) δεδομένων κάποιων γραμμικών σχέσεων (περιορισμών) τις οποίες εκφράζουμε με ένα πίνακα A και περιλαμβάνουν το x .

Πολλά προβλήματα τα μετασχηματίζουμε σε προβλήματα γραμμικού προγραμματισμού καθώς για το γραμμικό προγραμματισμό έχουμε αναπτύξει αποδοτικούς αλγόριθμους, όπως τον interior-point και τον ellipsoid.

Ορισμός 1.23 (Linear Programming).

ΕΙΣΟΔΟΣ: πίνακας $A \in \mathbb{R}^{m \times n}$ και διανύσματα $b \in \mathbb{R}^m, c \in \mathbb{R}^n$

ΕΞΟΔΟΣ: διανύσματα πραγματικών μεταβλητών $x \in \mathbb{R}^n$ τέτοιο ώστε να μεγιστοποιείται η $z = c^T x$ και να τηρούνται οι περιορισμοί $Ax \leq b$ και $x \geq 0$.

Ισοδύναμη γραφή, με την μορφή αθροισμάτων, είναι η παρακάτω:

$$\begin{aligned} \min \sum_{j=1}^n c_j x_j \\ \text{τηρώντας τα } \sum_{j=1}^n a_{i,j} x_j \leq b_i \quad & i = 1, \dots, m \\ x_j \geq 0 \quad & j = 1, \dots, n \end{aligned}$$

Τα στιγμιότυπα για το πρόβλημα του γραμμικού προγραμματισμού τα αποκαλούμε γραμμικά προγράμματα.

Κάθε x που ικανοποιεί την $Ax \leq b$ είναι μια εφικτή λύση (feasible solution). Υπάρχει το ενδεχόμενο η $Ax \leq b$ να μην ικανοποιείται για κανένα x . Τότε λέμε ότι το πρόγραμμα είναι ανέφικτο. Αν πάλι η z μπορεί να πάρει οποιαδήποτε τιμή τότε λέμε ότι το πρόγραμμα είναι μη φραγμένο, ενώ σε αντίθετη περίπτωση λέμε ότι είναι φραγμένο. Σε περίπτωση που το πρόγραμμα δεν είναι ούτε ανέφικτο, ούτε μη φραγμένο τότε η τιμή της αντικειμενικής συνάρτησης z είναι πεπερασμένη.

Το LP μπορούμε να το περιγράψουμε και με άλλες ισοδύναμες μορφές. Για παράδειγμα, μπορούμε αντί για πρόβλημα μεγιστοποίησης να το αντιμετωπίσουμε σαν πρόβλημα ελαχιστοποίησης, αν ζητάμε το x που να ελαχιστοποιεί την $z = c^T x$ δεδομένου των περιορισμών $Ax \geq b$ και $x \geq 0$.

Υπάρχουν επίσης και άλλες εκδοχές του προβλήματος όπως αυτή που δεν περιλαμβάνει τον περιορισμό $x \geq 0$. Μια άλλη εκδοχή είναι ο ακέραιος γραμμικός προγραμματισμός (integer linear programming ILP) όπου το x θέλουμε να είναι διάνυσμα ακεραίων μεταβλητών. Το ILP ως πρόβλημα απόφασης είναι NP-complete σε αντίθεση με το LP.

1.4.1 Δυϊσμός

Για ένα γραμμικό πρόγραμμα P της μορφής:

Βρες το διανύσματα x που μεγιστοποιεί τη $z = c^T x$ και να ικανοποιεί τους περιορισμούς $Ax \leq b$ και $x \geq 0$.

Υπάρχει ένα αντίστοιχο δυϊκό γραμμικό πρόγραμμα DP:

Βρες το διανύσματα y που ελαχιστοποιεί τη $w = b^T y$ και να ικανοποιεί τους περιορισμούς $A^T y \leq c$ και $y \geq 0$.

Η αντικειμενική συνάρτηση z του DP φράσσεται από την αντικειμενική συνάρτηση w του P. Πιο συγκεκριμένα έχουμε τα ακόλουθα θεωρήματα.

Ορισμός 1.24 (Ασθενής δυϊσμός ((weak duality)). Αν x' είναι εφικτή λύση για το P και y' μια εφικτή λύση του DP τότε ισχύει $c^T x' \leq b^T y'$.

Ορισμός 1.25 (Ισχυρός δυϊσμός ((strong duality)). Έστω ότι τα P και DP δεν είναι ανέφικτα z' είναι η βέλτιστη τιμή τις αντικειμενικής συνάρτησης του P και w' είναι η βέλτιστη τιμή τις αντικειμενικής συνάρτησης του DP τότε ισχύει ότι $z' = w'$.

Αν ένα LP είναι εφικτό και φραγμένο μπορούμε να εκμεταλλευτούμε τον ισχυρό δυϊσμό και να βρούμε την βέλτιστη εφικτή λύση x' από την αντίστοιχη βέλτιστη εφικτή λύση y' του DLP.

1.4.2 Γεωμετρική προσέγγιση

Οι γραμμικοί περιορισμοί του LP σχηματίζουν ένα κυρτό πολύεδρο P διάστασης $rank(A)$ στο R^n . Η συνάρτηση z είναι μια συνάρτηση πάνω στο κυρτό περίβλημα αυτού του πολύεδρου. Καλούμαστε να βρούμε το σημείο $x' \in P$ που μεγιστοποιεί την z .

Όταν το πρόγραμμα είναι ανέφικτο, το P είναι κενό, ενώ όταν το πρόβλημα είναι μη φραγμένο, το P δεν φράσσεται προς την κατεύθυνση του c , με αποτέλεσμα η z να μην έχει ολικό μέγιστο και το γραμμικό πρόγραμμα να έχει άπειρες λύσεις.

Οι κόμβοι του κυρτού περιβλήματος του P καλούνται βασικές λύσεις. Μια από τις βασικές λύσεις είναι και η βέλτιστη. Επειδή $rank(A) \leq m \leq n$ το πολύεδρο μπορεί να είναι μικρότερης διάστασης από τον χώρο στο οποίο βρίσκεται. Συμπεραίνουμε λοιπόν ότι κάθε βασική λύση θα έχει το πολύ m μη μηδενικές μεταβλητές.

Μητροειδή (matroids)

Τα μητροειδή γενικεύουν την έννοια της γραμμική ανεξαρτησίας που συναντάμε στους διανυσματικούς χώρους. Τα βρίσκουμε σε πολλούς τομείς των μαθηματικών και της πληροφορικής όπως τη γραμμική άλγεβρα, τη θεωρία γραφημάτων και τους άπληστους αλγορίθμους (μια κατηγορία αλγορίθμων για προβλήματα βελτιστοποίησης, το βασικό τους χαρακτηριστικό είναι πως όταν χρειαστεί να κάνουν μια αμετάκλητη επιλογή, διαλέγουν αυτή που φαίνεται καλύτερη με βάση την τρέχουσα κατάσταση).

Ορισμός 1.26 (Μητροειδές (matroid)). Ένα μητροειδές M είναι ένα ζευγάρι (U, I) , όπου U είναι ένα σύμπαν από στοιχεία και $I \subseteq 2^U$ μια συλλογή από υποσύνολα του U που έχει τις παρακάτω ιδιότητες

1. $\emptyset \in I$.
2. Αν $X \in I$ και $Y \subseteq X$ τότε $Y \in I$.
3. Αν $X, Y \in I$ και $|X| < |Y|$ τότε $\exists y \in Y \setminus X$ τέτοιο ώστε $X + y \in I$.

Παράδειγμα (Uniform μητροειδές). Έστω $M = (U, I)$ με $U = [n] = \{1, 2, \dots, n\}$ και $I = \{I : I \subseteq [n], |I| \leq r\}$ όπου r θετικός ακέραιος. Τότε το $M = (U, I)$ είναι Uniform μητροειδές και συμβολίζεται με $U^{r,n}$. Για $n = 4, r = 2$ έχουμε:

$$U^{2,4} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}.$$

Παράδειγμα (Partition μητροειδές). Έστω U_1, U_2, \dots, U_n μια διαμέριση του U και k_1, k_2, \dots, k_n θετικοί ακέραιοι. Έστω $I = \{I : I \subseteq U, |I \cap U_i| \leq k_i \text{ για κάθε } 1 \leq i \leq n\}$ τότε το $M = (U, I)$ είναι partition μητροειδές.

Παράδειγμα (Laminar μητροειδές). Έστω F μια laminar οικογένεια του U ($X, Y \in F$ τότε $X, Y \subseteq U$ και ή $X \cap Y = \emptyset$ ή $X \subseteq Y$ ή $Y \subseteq X$) τέτοια ώστε κάθε $x \in U$ να βρίσκεται σε ένα τουλάχιστον $X \in F$. Επιπλέον, σε κάθε $X \in F$ αντιστοιχεί ένας θετικός ακέραιος $k(X)$. Έστω $I = \{I : I \subseteq U, |I \cap X| \leq k(X) \forall X \in F\}$ τότε το $M = (U, I)$ είναι laminar μητροειδές.

Ορισμός 1.27 (Βάση μητροειδούς (matroid base)). Έστω μητροειδές M , μια βάση $B \in I$ του M είναι το σύνολο με τον μεγαλύτερο πληθικό αριθμό.

Ορισμός 1.28 (Τάξη μητροειδούς (matroid rank)). Έστω μητροειδές M και B μια βάση του. Η τάξη r του M ορίζεται ως $r = |B|$.

1.5.1 Πολύτοπα μητροειδών και πολυμητροειδή

Έστω $A \subset [n]$, ορίζουμε το διάνυσμα

$$\mathbf{e}_A = \sum_{i \in A} \mathbf{e}_i$$

όπου $\mathbf{e}_i \in \mathbb{R}^n$ το μοναδιαίο διάνυσμα με 1 στην θέση του i . Για παράδειγμα αν $A = \{1, 3, 4\} \subset [5]$ τότε έχουμε $\mathbf{e}_A = (1, 0, 1, 1, 0) \in \mathbb{R}^5$.

Ορισμός 1.29 (Πολύτοπο μητροειδούς (matroid polytope)). Έστω μητροειδές M και B_M μια βάση του. Το πολύτοπο του μητροειδούς M είναι

$$P_M = \text{conv}(\{\mathbf{e}_B | B \in B_M\})$$

όπου το $\text{conv}(X)$ ορίζει το κυρτό περίβλημα του συνόλου X .

Παράδειγμα. Το πολύτοπο του $U^{2,4} = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$ είναι το

$$P_{U^{2,4}} = \text{conv}((1, 1, 0, 0), (1, 0, 1, 0), (1, 0, 0, 1), (0, 1, 1, 0), (0, 1, 0, 1), (0, 0, 1, 1)).$$

Μια συνάρτηση $\psi : 2^{[n]} \rightarrow \mathbb{R}$ λέγεται *submodular* αν $\psi(X \cap Y) + \psi(X \cup Y) \leq \psi(X) + \psi(Y)$ για κάθε $X, Y \subseteq [n]$ και *non-decreasing* αν $\psi(X) \leq \psi(Y)$ για κάθε $X \subseteq Y \subseteq [n]$.

Ορισμός 1.30 (Συνάρτηση τάξης πολυμητροειδούς (polymatroid rank function)). *Μια συνάρτηση $\psi : 2^{[n]} \rightarrow \mathbb{R}$ είναι συνάρτηση τάξης πολυμητροειδούς αν είναι submodular, non-decreasing και $\psi(\emptyset) = 0$.*

Ορισμός 1.31 (Πολυμητροειδές (Polymatroid)). *Ένα πολυμητροειδές που ορίζεται από μια συνάρτηση τάξης ψ , είναι το κυρτό πολύεδρο:*

$$P_\psi = \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i \in A} x_i \leq \psi(A), \forall A \subseteq [n], \mathbf{x} \geq \mathbf{0} \right\}.$$

Για περαιτέρω ανάγνωση

Στο παρόν κεφάλαιο, αναφερθήκαμε, εν τάχει, σε ορισμένες έννοιες, απαραίτητες για μια σαφή κατανόηση της εργασίας. Κλείνοντας την συνοπτική παρουσίαση, παρατίθενται μερικά συγγράμματα, τα οποία ενδείκνυνται για περαιτέρω μελέτη των συγκεκριμένων εννοιών, προς τυχόν ενδιαφερόμενους. Πολλά από τα παρακάτω συγγράμματα έχουν μεταφραστεί στα ελληνικά.

Για την πολυπλοκότητα αλγορίθμων [25, 12, 8].

Για την υπολογιστική πολυπλοκότητα [15, 39, 33, 28].

Για τους προσεγγιστικούς αλγόριθμους [1, 19, 42].

Για τους πιθανοτικούς αλγόριθμους [31, 25, 30].

Για το γραμμικό προγραμματισμό [36, 5, 11, 10].

Για τα μητροειδή [35, 22].

ΤΟ ΠΡΟΒΛΗΜΑ

Το πρόβλημα με το οποίο θα ασχοληθούμε είναι το Generalized Min-Sum Set Cover ή Multiple Intents Re-Ranking [2]. Σε αυτό το NP -hard πρόβλημα, μας δίνετε ένα σύμπαν U που αποτελείται από n στοιχεία και μια συλλογή από υποσύνολα \mathbf{S} του σύμπαντος U . Το κάθε υποσύνολο $S_i \in \mathbf{S}$ έχει μια απαραίτητη προϋπόθεση κάλυψης $K(S_i)$. Στόχος είναι να διαλέξουμε διαδοχικά το ένα στοιχείο μετά το άλλο τοποθετώντας τα n στοιχεία με τέτοια σειρά ώστε ο μέσος χρόνος κάλυψης των συνόλων να ελαχιστοποιηθεί, όπου ο χρόνος κάλυψης ενός συνόλου S_i είναι η πρώτη φορά που $K(S_i)$ στοιχεία έχουν επιλεγεί.

Δεν θα προχωρήσουμε απευθείας στον τυπικό ορισμό του προβλήματος αλλά, για την καλύτερη κατανόηση του, θα δούμε πρώτα απλούστερα παρόμοια προβλήματα και κάποιες ειδικές του περιπτώσεις.

Set Cover και Hitting Set

Το Generalized Min-Sum Set Cover μοιάζει πολύ με το γνωστό NP -complete πρόβλημα Set Cover και ακόμα περισσότερο στην ισοδύναμη παραλλαγή του το Hitting Set.

Ορισμός 2.1 (Set Cover).

ΕΙΣΟΔΟΣ: Ένα σύνολο $U = \{e_1, e_2, \dots, e_n\}$ και μια συλλογή από υποσύνολα $\mathbf{S} = \{S_1, S_2, \dots, S_m\}$ τέτοια ώστε $S_i \subseteq U, i = 1, \dots, m$.

ΕΞΟΔΟΣ: Το μικρότερο υποσύνολο $C \subseteq \mathbf{S}$ τέτοιο ώστε $\bigcup_{S_i \in C} S_i = U$.

Στο Set Cover θέλουμε να διαλέξουμε τα λιγότερα υποσύνολα από το \mathbf{S} , έτσι ώστε μέσα σε αυτά τα υποσύνολα να βρίσκονται όλα τα στοιχεία του U τουλάχιστον μια φορά το καθένα.

Παράδειγμα. Για είσοδο τα:

$$U = \{a, b, c, d, e\} \text{ και } \mathbf{S} = \{\{a, b, c\}, \{b, d\}, \{c, d\}, \{d, e\}\}$$

η έξοδος είναι:

$$C = \{\{a, b, c\}, \{d, e\}\}.$$

Το Set Cover όπως και το Hitting Set είναι από τα πρώτα προβλήματα που αποδείχτηκε ότι είναι NP -complete από τον Richard M. Karp το 1972 [23]. Ο απλός άπληστος αλγόριθμος, που ξεκινά με $C = \emptyset$ και κάθε φορά επιλέγει το σύνολο που έχει τα περισσότερα στοιχεία του U , τα οποία δεν έχουν καλυφθεί από τα σύνολα που υπάρχουν ήδη στο C , έχει εγγύηση απόδοση $H(n)$ (David S. Johnson 1974 [21]). Όπου $H(n)$ είναι αρμονική σειρά του n :

$$H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{k=1}^n \frac{1}{k} \leq \ln n + 1$$

Οι C. Lund και M. Yannakakis [29] απέδειξαν το 1994 ότι δεν υπάρχει πολυωνυμικός προσεγγιστικός αλγόριθμος για το Set Cover με εγγύηση απόδοσης μικρότερη από $\frac{1}{2} \log_2 n \approx$

$0.72 \ln n$, εκτός εάν το NP έχει αλγορίθμους με χρόνο εκτέλεσης $n^{O(\log \log n)}$, η οποία είναι πιο αδύναμη από την υπόθεση: εκτός εάν $P = NP$, που επιτρέπει στο NP να έχει και πολυωνυμικούς αλγορίθμους. Ο Uriel Feige το 1998 [13] βελτίωσε το φράγμα σε $(1 - \epsilon) \ln n$ για κάθε $\epsilon > 0$, υπό την ίδια υπόθεση. Αυτό σημαίνει ότι, ουσιαστικά, ο απλός άπληστος αλγόριθμος είναι και ο καλύτερος που μπορούμε να έχουμε για το Set Cover.

Να σημειωθεί πως τα αποτελέσματα που ισχύουν για το Set Cover ισχύουν και για το Hitting Set αφού όπως θα δούμε αυτά τα δυο προβλήματα είναι ισοδύναμα.

Ορισμός 2.2 (Hitting Set).

ΕΙΣΟΔΟΣ: Ένα σύνολο $U = \{e_1, e_2, \dots, e_n\}$ και μια συλλογή από υποσύνολα $\mathbf{S} = \{S_1, S_2, \dots, S_m\}$ τέτοια ώστε $S_i \subseteq U, i = 1, \dots, m$.

ΕΞΟΔΟΣ: Το μικρότερο υποσύνολο $H \subseteq U$ τέτοιο ώστε $H \cap S_i \neq \emptyset, i = 1, \dots, m$.

Στο Hitting Set θέλουμε να βρούμε το μικρότερο σύνολο από στοιχεία $H \subseteq U$, ώστε κάθε υποσύνολο $S_i \in \mathbf{S}$ να έχει τουλάχιστον ένα κοινό στοιχείο με το H .

Ας χρησιμοποιήσουμε ξανά την ίδια είσοδο που είχαμε πιο πριν για το Set Cover, αυτή τη φορά για το Hitting Set. Θα είναι πολύ εύκολο να διαπιστώσουμε την σχέση που έχουν αυτά τα δυο προβλήματα.

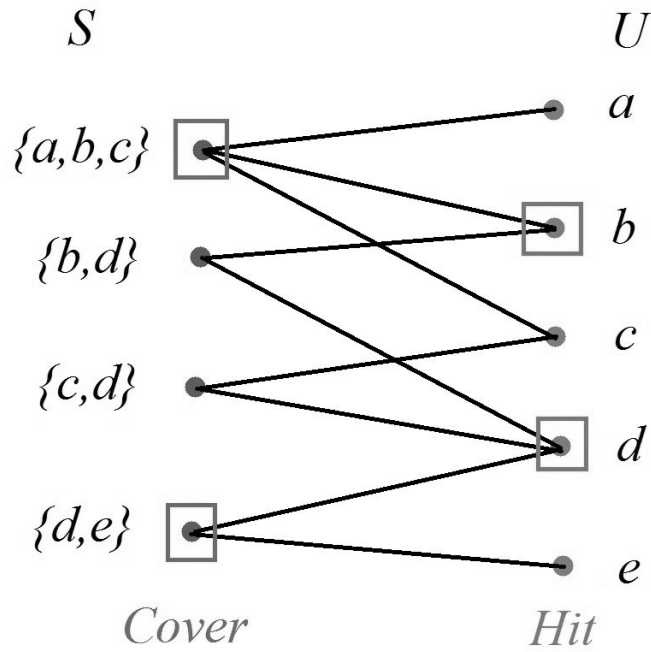
Παράδειγμα. Για είσοδο τα:

$$U = \{a, b, c, d, e\} \text{ και } \mathbf{S} = \{\{a, b, c\}, \{b, d\}, \{c, d\}, \{d, e\}\}$$

η έξοδος είναι:

$$H = \{b, d\}.$$

Από το ένα πρόβλημα πάμε πολύ εύκολα στο άλλο. Απλώς μετατρέποντας τα σύνολα σε στοιχεία και τα στοιχεία σε σύνολα. Τα νέα σύνολα περιέχουν σαν στοιχεία τα σύνολα στα οποία ανήκαν προηγουμένως. Ας φανταστούμε την σχέση συνόλων-στοιχείων σαν ένα διμερές γράφημα όπως στο παρακάτω σχήμα.



Σχήμα 1: Στο διμερές γράφημα παρατηρούμε τη σχέση μεταξύ Set Cover και Hitting Set.

Στη μια πλευρά, οι κόμβοι αντιπροσωπεύουν τα σύνολα και στην άλλη, τα στοιχεία. Ακμές ενώνουν το κάθε σύνολο με τα στοιχεία τα οποία περιέχει. Θέλουμε να βρούμε τους λιγότερους κόμβους της μιας πλευράς που συνδέονται με όλους τους κόμβους της άλλης. Όταν διαλέγουμε κόμβους από την πλευρά των συνόλων πρόκειται για το Set Cover ενώ όταν διαλέγουμε κόμβους από την πλευρά των στοιχείων πρόκειται για το Hitting Set.

Min-Sum Set Cover και Min-Latency Set Cover

Σειρά έχει να δούμε τις δυο ειδικές περιπτώσεις του Generalized Min-Sum Set Cover. Το Min-Sum Set Cover μοιάζει στο Hitting Set, και στα δυο έχουμε ένα σύμπαν U με n στοιχεία και μια συλλογή από υποσύνολα $\mathbf{S} = \{S_1, S_2, \dots, S_m\}$. Η διαφορά είναι ότι αντί να διαλέγουμε στοιχεία από το U , τα βάζουμε όλα σε μια σειρά. Πιο συγκεκριμένα, επιλέγουμε ένα στοιχείο για κάθε χρονική μονάδα $t \in [n]$, αν ένα στοιχείο e επιλεγθεί για τη χρονική μονάδα t , ο χρόνος κάλυψης του είναι $Cov(e) = t$. Αναλόγως, χρόνος κάλυψης ενός συνόλου S είναι $Cov(S) = \min_{e \in S} Cov(e)$. Ένα σύνολο, δηλαδή, είναι καλυμμένο όταν ένα στοιχείο του έχει επιλεγθεί. Ο στόχος είναι να διαλέξουμε τα στοιχεία με τέτοια σειρά (αναθέτοντας τους t) ώστε να ελαχιστοποιηθεί το $\sum_{S \in \mathbf{S}} Cov(S)$.

Ορισμός 2.3 (Min-Sum Set Cover).

ΕΙΣΟΔΟΣ: Ένα σύνολο $U = \{e_1, e_2, \dots, e_n\}$ και μια συλλογή από υποσύνολα $\mathbf{S} = \{S_1, S_2, \dots, S_m\}$ τέτοια ώστε $S_i \subseteq U, i = 1, \dots, m$.

ΕΞΟΔΟΣ: Μια διάταξη A των στοιχείων του U , τέτοια ώστε να ελαχιστοποιείται το $\sum_{S \in \mathbf{S}} Cov(S)$, όπου $Cov(e) = t$ η θέση του στοιχείου e στην διάταξη και $Cov(S) = \min_{e \in S} Cov(e)$.

Παράδειγμα. Για είσοδο τα:

$$U = \{a, b, c, d, e\}$$

$$\mathbf{S} = \{S_1 = \{a, b, c\}, S_2 = \{b, d\}, S_3 = \{c, d\}, S_4 = \{d, e\}\}$$

η έξοδος είναι:

$$A = \begin{pmatrix} d & c & a & b & e \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

Από την σειρά με την οποία βάλαμε τα στοιχεία του U προκύπτει ο χρόνος κάλυψης του κάθε στοιχείου:

$$\text{Cov}(a) = 3, \text{Cov}(b) = 4, \text{Cov}(c) = 2, \text{Cov}(d) = 1, \text{Cov}(e) = 5$$

και ο χρόνος κάλυψης κάθε σύνολου:

$$\text{Cov}(S_1) = \min\{3, 4, 2\} = 2, \text{Cov}(S_2) = 1, \text{Cov}(S_3) = 1, \text{Cov}(S_4) = 1$$

ο μέσος χρόνος κάλυψης είναι:

$$\sum_{S \in \mathbf{S}} \text{Cov}(S) = 5.$$

Για αυτό το πρόβλημα ο άπληστος αλγόριθμος, ο οποίος διαλέγει κάθε φορά το στοιχείο που καλύπτει το μεγαλύτερο αριθμό από ακάλυπτα σύνολα, είναι 4-προσεγγιστικός, [4, 14], και είναι ο καλύτερος που μπορούμε να έχουμε, εκτός αν $P = NP$ [14].

Το Min-Latency Set Cover είναι ίδιο με το Min-Sum Set Cover. Το μόνο που αλλάζει είναι ο τρόπος με τον οποίο υπολογίζουμε το χρόνο κάλυψης ενός συνόλου S . Αυτή τη φορά είναι $\text{Cov}(S) = \max_{e \in S} \text{Cov}(e)$, ο χρόνος, δηλαδή, στον οποίο όλα τα στοιχεία του συνόλου έχουν επιλεγθεί.

Ορισμός 2.4 (Min-Latency Set Cover).

ΕΙΣΟΔΟΣ: Ένα σύνολο $U = \{e_1, e_2, \dots, e_n\}$ και μια συλλογή από υποσύνολα $\mathbf{S} = \{S_1, S_2, \dots, S_m\}$ τέτοια ώστε $S_i \subseteq U, i = 1, \dots, m$.

ΕΞΟΔΟΣ: Μια διάταξη A των στοιχείων του U , τέτοια ώστε να ελαχιστοποιείται το $\sum_{S \in \mathbf{S}} \text{Cov}(S)$, όπου $\text{Cov}(e) = t$ η θέση του στοιχείου e στην διάταξη και $\text{Cov}(S) = \max_{e \in S} \text{Cov}(e)$.

Παράδειγμα. Για είσοδο τα:

$$U = \{a, b, c, d, e\}$$

$$\mathbf{S} = \{S_1 = \{a, b, c\}, S_2 = \{b, d\}, S_3 = \{c, d\}, S_4 = \{d, e\}\}$$

η έξοδος είναι:

$$A = \begin{pmatrix} b & c & d & e & a \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

ο χρόνος κάλυψης κάθε στοιχείου:

$$\text{Cov}(a) = 5, \text{Cov}(b) = 1, \text{Cov}(c) = 2, \text{Cov}(d) = 3, \text{Cov}(e) = 4$$

ο χρόνος κάλυψης κάθε σύνολου:

$$\text{Cov}(S_1) = \max\{5, 1, 2\} = 5, \text{Cov}(S_2) = 3, \text{Cov}(S_3) = 3, \text{Cov}(S_4) = 4$$

ο μέσος χρόνος κάλυψης:

$$\sum_{S \in \mathbf{S}} \text{Cov}(S) = 15$$

Το Min-Latency Set Cover είναι ισοδύναμο με το Single Machine Precedence Constrained Scheduling [26, 27], για το οποίο υπάρχουν πολύ γνωστοί 2-προσεγγιστικοί αλγόριθμοι [7, 22]. Σε αυτό το πρόβλημα έχουμε n εργασίες j_1, j_2, \dots, j_n και θέλουμε να τις αναθέσουμε σε μια μηχανή για να τις εκτελέσει. Η μηχανή μπορεί να εκτελέσει μόνο μια εργασία την φορά. Κάθε εργασία έχει διάρκεια p_j και βαρύτητα w_j . Η εργασίες έχουν μεταξύ τους σχέσεις προτεραιότητας, τις οποίες μπορούμε να φανταστούμε σαν ένα κατευθυνόμενο γράφημα $G = (E, V)$, όπου η ακμή $(v, u) \in E$ σημαίνει πως η εργασία v πρέπει να έχει τελειώσει προτού η u αρχίσει. Σκοπός είναι να αναθέσουμε τις εργασίες με τέτοιο τρόπο ώστε να ελαχιστοποιηθεί το $\sum_{j=1}^n w_j C_j$, όπου C_j είναι η χρονική στιγμή κατά την οποία η εργασία j ολοκληρώθηκε. Για να καταλάβετε την ισοδυναμία που έχει με το Min-Latency Set Cover αναλογιστείτε την περίπτωση που έχουμε δυο διαφορετικού τύπου εργασίες J_a και J_b . Όλες οι εργασίες $j \in J_a$ έχουν $w_j = 0$ και $p_j = 1$ (αντιστοιχούν στα στοιχεία e_1, e_2, \dots, e_n) και οι εργασίες $j \in J_b$ έχουν $w_j = 1, p_j = 0$ (αντιστοιχούν στα σύνολα $S_j \subseteq J_a$). Οι σχέσεις προτεραιότητας είναι οι εξής, κάθε εργασία $j \in J_b$ που αντιστοιχεί σε ένα σύνολο πρέπει να εκτελεστεί αφότου εκτελεστούν όλες οι εργασίες $j \in S_j \subseteq J_a$ που αντιστοιχούν στα στοιχεία τα οποία ανήκουν σε αυτό το σύνολο. Έχει αποδειχθεί ότι δεν υπάρχει προσεγγιστικός αλγόριθμος για το Single Machine Precedence Constrained Scheduling, και κατ' επέκταση ούτε για το Min-Latency Set Cover, με εγγύηση απόδοσης καλύτερη από $2 - \epsilon, \epsilon > 0$ δεδομένου ότι ισχύει η UGC [24].

Generalized Min-Sum Set Cover

Το Generalized Min-Sum Set Cover είναι η γενίκευση των δύο προηγούμενων προβλημάτων. Εδώ, κάθε σύνολο $S \in \mathbf{S}$ έχει επίσης μια προϋπόθεση κάλυψης $K(S) \in \{1, 2, \dots, |S|\}$ και ο χρόνος κάλυψης είναι η χρονική μονάδα στην οποία $K(S)$ στοιχεία από το S επιλέγονται

$$\text{Cov}(S) = \min\{t : |\{e \in S : \text{Cov}(e) \leq t\}| = K(S)\}.$$

Ο στόχος παραμένει να ελαχιστοποιήσουμε το $\sum_{S \in \mathbf{S}} \text{Cov}(S)$.

Ορισμός 2.5 (Generalized Min-Sum Set Cover).

ΕΙΣΟΔΟΣ: Ένα σύνολο $U = \{e_1, e_2, \dots, e_n\}$ και μια συλλογή από υποσύνολα $\mathbf{S} = \{S_1, S_2, \dots, S_m\}$ τέτοια ώστε $S_i \subseteq U, i = 1, \dots, m$, όπως και έναν ακέραιο $K(S) \in \{1, 2, \dots, |S|\}$ για κάθε $S \in \mathbf{S}$.

ΕΞΟΔΟΣ: Μια διάταξη A των στοιχείων του U , τέτοια ώστε να ελαχιστοποιείται το $\sum_{S \in \mathbf{S}} \text{Cov}(S)$,

όπου $Cov(e) = t$ η θέση του στοιχείου e στην διάταξη και $Cov(S) = \min\{t : |e \in S : Cov(e) \leq t| = K(S)\}$.

Παρατηρήστε ότι:

- Για $K(S) = 1$ για κάθε $S \in \mathbf{S}$, προκύπτει το Min-Sum Set Cover
- Για $K(S_i) = |S_i|$, $i = 1, \dots, m$ έχουμε το Min-Latency Set Cover.

Επειδή το Generalized Min-Sum Set Cover είναι γενίκευση του Min-Sum Set Cover, το 4 είναι ένα κάτω φράγμα για την εγγύηση απόδοση που μπορεί να πετύχει προσεγγιστικός αλγόριθμος που το επιλύει.

Το πρόβλημα παρουσιάστηκε και μελετήθηκε για πρώτη φορά από τους Yossi Azar, Iftah Gamzu και Xiaoxin Yin [2], οι οποίοι έδωσαν και έναν άπληστο $O(\ln r)$ -προσεγγιστικό αλγόριθμο για το πρόβλημα, όπου r είναι το μέγεθος του μεγαλύτερου συνόλου στο \mathbf{S} . Στη παρούσα εργασία, θα μελετήσουμε προσεγγιστικούς αλγόριθμους με σταθερή εγγύηση απόδοσης για το Generalized Min-Sum Set Cover, βασισμένους στο γραμμικό προγραμματισμό, με την σειρά που αυτοί παρουσιάστηκαν. Θα δούμε με ποιον τρόπο, βελτιώνονται οι τεχνικές των προηγούμενων αλγορίθμων και επιτυγχάνεται όλο και καλύτερη εγγύηση απόδοσης.

Preemptive Generalized Min-Sum Set Cover

Στη preemptive εκδοχή του προβλήματος επιτρέπεται να τεμαχίσουμε τα στοιχεία e_i , $i \in [n]$ σε μικρότερα κομμάτια (υποθέτουμε ότι ένα στοιχείο έχει μήκος μια χρονική μονάδα). Έτσι, μπορούμε να φτιάξουμε μια μη ακέραια διάταξη με τα στοιχεία στο διάστημα $[0, n]$. Πιο τυπικά, ορίζουμε μια συνάρτηση $x_e(t) : [0, n] \rightarrow \{0, 1\}$, έτσι ώστε $\int_{t=0}^n x_e(t) dt = 1$ για κάθε $e \in [n]$ και $\sum_{e \in [n]} x_e(t) = 1$ για κάθε $t \in [0, n]$. Η $x_e(t)$ δείχνει αν το στοιχείο e έχει επιλεγεί την χρονική στιγμή t ή όχι. Οι περιορισμοί εξασφαλίζουν ότι κάθε χρονική στιγμή θα είναι επιλεγμένο μόνο ένα στοιχείο και πως το άθροισμα όλων των κομματιών του κάθε στοιχείου στην διάταξη θα δίνει μονάδα. Επιπλέον, ο περιορισμός $\int_{t=0}^n x_e(t) dt = 1$ εξασφαλίζει ότι τα κομμάτια που είναι χωρισμένο το κάθε στοιχείο είναι συμπαγή και είναι σαφές πως ακριβώς είναι μέσα σε κάθε χρονική μονάδα. Ο χρόνος κάλυψης ενός συνόλου $cov(S)$ είναι η πρώτη χρονική στιγμή όπου $\int_{\tau=0}^t \sum_{e \in S} x_e(\tau) d\tau \geq K(S)$. Παρατηρήστε ότι αυτή τη φορά ο χρόνος κάλυψης $cov(S)$ δεν είναι απαραίτητα ακέραιος. Ο στόχος παραμένει να ελαχιστοποιηθεί το $\sum_{S \in \mathbf{S}} Cov(S)$.

Παράδειγμα. Για το στιγμιότυπο:

$$U = \{a, b, c, d\}, \text{ με } \mathbf{S} = \{S_1 = \{a, b\}, S_2 = \{c, d\}\} \text{ και } K(S_1) = K(S_2) = 1$$

μια πιθανή διάταξη είναι εξής:

$$\begin{array}{cccccc} \left[\frac{1}{4}, \frac{3}{4}\right], & \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\right], & [1], & \left[\frac{3}{4}, \frac{1}{4}\right], \\ a & c & b & c & a & d & b & a \end{array}$$

την οποία θα συμβολίζουμε και ως

$$(x_e(t))_{e \in [n]} = \frac{1}{4}a, \frac{3}{4}c, \frac{1}{4}b, \frac{1}{4}c, \frac{1}{2}a, 1d, \frac{3}{4}b, \frac{1}{4}a,$$

ο χρόνος κάλυψης των συνόλων είναι:

$$\text{Cov}(S_1) = 2, \text{Cov}(S_2) = 1 + \frac{1}{2}$$

ο μέσος χρόνος κάλυψης:

$$\sum_{S \in \mathbf{S}} \text{Cov}(S) = \frac{7}{2}$$

Το Preemptive Generalized Min-Sum Set Cover για $K(S_i) = |S_i|$, $i = 1, \dots, m$ είναι ισοδύναμο με το Single Machine Scheduling with Precedence Constraints and Preemptions, το οποίο δεν είναι παρά το Single Machine Precedence Constrained Scheduling που είδαμε προηγουμένως, όπου μας επιτρέπεται, επιπλέον, να τεμαχίσουμε τις εργασίες. Ωστόσο, ότι μπορούμε να κόψουμε τις εργασίες δεν βελτιώνει την ποιότητα της λύσης του προβλήματος. Οποιαδήποτε, δηλαδή, λύση της preemptive εκδοχής μετασχηματίζεται σε μια λύση της non-preemptive εκδοχής με το ίδιο κόστος. Άρα, μπορούμε να πούμε πως δεν υπάρχει $(2 - \epsilon)$ -προσεγγιστικός αλγόριθμος, για το Preemptive Generalized Min-Sum Set Cover για κάθε $\epsilon > 0$, δεδομένου ότι ισχύει η UGC [24].

Ο ΠΡΩΤΟΣ ΑΛΓΟΡΙΘΜΟΣ ΜΕ ΣΤΑΘΕΡΗ ΕΓΓΥΗΣΗ ΑΠΟΔΟΣΗΣ

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα αποτελέσματα της εργασίας των Nikhil Bansal, Anuram Gupta και Ravishankar Krishnaswamy [3] οι οποίοι είναι οι πρώτοι που έδειξαν ότι το Generalized Min-Sum Set Cover έχει αλγόριθμο με σταθερή εγγύηση απόδοσης. Ο αλγόριθμος τους είναι 485-προσεγγιστικός, δηλαδή μια λύση που θα δώσει, ενδέχεται να έχει έως και 485 φορές μεγαλύτερο κόστος από την βέλτιστη λύση. Όπως καταλαβαίνετε ο αλγόριθμος δεν είναι ικανοποιητικός. Συνήθως βλέπουμε προσεγγιστικούς αλγόριθμους με εγγύηση απόδοσης 2 ή ακόμα μικρότερη. Ωστόσο, ήταν ο πρώτος με σταθερή εγγύηση απόδοσης για το πρόβλημα και πάνω στις ιδέες του βασίστηκαν όλοι οι άλλοι αλγόριθμοι που θα δούμε στην συνέχεια. Επιπλέον, είναι απλός και εκμεταλλεύεται όλες τις βασικές τεχνικές που βρίσκουμε στους προσεγγιστικούς αλγόριθμους που χρησιμοποιούν γραμμικό προγραμματισμό.

Χαλαρωμένα γραμμικά προγράμματα (LP relaxations)

Μια συνηθισμένη στρατηγική στους προσεγγιστικούς αλγόριθμους είναι να περιγράψουμε το πρόβλημα που θέλουμε να λύσουμε σαν ένα πρόβλημα γραμμικού προγραμματισμού (Χαλαρωμένο γραμμικό πρόγραμμα LP relaxation). Το πλεονέκτημα είναι πως μπορούμε να λύσουμε αυτό το LP σε πολυωνυμικό χρόνο και να πάρουμε μια μη ακέραια προσέγγιση της λύσης, πάνω στην οποία μπορούμε να βασιστούμε για να εξάγουμε την ακέραια τελική προσέγγιση.

3.1.1 Το γραμμικό πρόγραμμα A (LP A)

Έστω $[n] = \{1, 2, \dots, n\}$ όπου $n = |U|$, ο αριθμός των στοιχείων στο σύμπαν. Ορίζουμε τις μεταβλητές x_{et} που δείχνει αν το στοιχείο $e \in U$ θα επιλεγεί τη χρονική μονάδα $t \in [n]$ και y_{St} που δείχνει αν το σύνολο S έχει καλυφθεί πριν τη χρονική μονάδα $t \in [n]$. Το LP A είναι το ακόλουθο:

$$\min \sum_{1 \leq t \leq |U|} \sum_{S \in \mathbf{S}} (1 - y_{St})$$

$$\sum_{e \in U} x_{et} = 1 \quad \forall t \in [n] \quad (1)$$

$$\sum_{t \in [n]} x_{et} = 1 \quad \forall e \in U \quad (2)$$

$$\sum_{e \in S} \sum_{t' < t} x_{et'} \geq K(S) \cdot y_{St} \quad \forall S \in \mathbf{S}, \forall t \in [n] \quad (3)$$

$$x_{et}, y_{St} \in [0, 1] \quad \forall e \in U, S \in \mathbf{S}, t \in [n] \quad (4)$$

Οι περιορισμοί (1) απαιτούν πως μόνο ένα στοιχείο μπορεί να επιλεγεί σε μια χρονική μονάδα και οι περιορισμοί (2) απαιτούν ότι κάθε στοιχείο πρέπει να έχει επιλεγεί μόνο μια φορά. Οι περιορισμοί (3) δεν επιτρέπουν σε ένα σύνολο S να καλυφθεί αν δεν έχουν

τοποθετηθεί τουλάχιστον $K(S)$ από τα στοιχεία του στη διάταξη, αν δηλαδή $y_{St} = 1$ (το S έχει καλυφθεί πριν τη χρονική μονάδα t), τότε στις χρονικές μονάδες $t' < t$ θα πρέπει να έχουν επιλεγθεί τουλάχιστον $K(S)$ στοιχεία $K(S)$ από το S . Αν x_{et} και y_{St} παίρνουν τιμές μόνο 0 ή 1. Όσο $y_{St} = 0$ το σύνολο S θα συνεχίσει να προσφέρει στο κόστος της αντικειμενικής συνάρτησης αφού ο όρος $(1 - y_{St})$ θα είναι ίσος με 1. Όταν $y_{St} = 1$ και ο όρος $(1 - y_{St})$ γίνει 0, το σύνολο θα σταματήσει να επιβαρύνει το κόστος της αντικειμενικής. Κάθε ακέραια λύση του προβλήματος ικανοποιεί τους περιορισμούς και το κόστος της αντικειμενικής συνάρτησης για μια βέλτιστη λύση θα είναι το μικρότερο. Επειδή, όμως, τα x_{et} και y_{St} παίρνουν τιμές μεταξύ του 0 και του 1 το κόστος της λύσης του LP A ($LPOpt$) θα είναι μικρότερο ή ίσο του κόστους μιας βέλτιστης λύσης Opt .

Λήμμα 3.1. *Το κόστος της λύσης του LP A ($LPOpt$) είναι το πολύ ο χρόνος κάλυψης όλων των συνόλων μιας βέλτιστης λύσης (Opt).*

Το LP A είναι ένα έγκυρο LP relaxation για το Generalized Min-Sum Set Cover ωστόσο αυτό δεν είναι αρκετό. Αν LPe είναι το κόστος της αντικειμενικής συνάρτησης για μια βέλτιστη λύση του προβλήματος και $LPOpt$ το κόστος της βέλτιστης λύσης του γραμμικού προγράμματος. Ο λόγος μεταξύ του LPe και $LPOpt$ λέγεται integrality gap. Ένα LP για να είναι χρήσιμο πρέπει να έχει σταθερό LP integrality gap για κάθε στιγμιότυπο I . Σε αντίθετη περίπτωση η διάφορα του κόστους των δυο λύσεων μπορεί να είναι αυθαίρετα μεγάλη. Κάτι τέτοιο δεν ισχύει για το LP A και θα το δείξουμε με ένα αντιπαράδειγμα. Ας υποθέσουμε ότι έχουμε το στιγμιότυπο I

$$U = \{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_l\}$$

$$I \text{ σύνολα της μορφής: } S_i = \{a_1, a_2, \dots, a_n, b_i\} \quad i = 1, 2, \dots, l$$

$$\text{και } K(S) = (n + 1) \quad \forall S \in \mathbf{S}.$$

Παρατηρήστε πως το I είναι ένα στιγμιότυπο του Min-Latency Set Cover. Μια ακέραια βέλτιστη λύση είναι να επιλέξουμε με την σειρά πρώτα τα a_1, a_2, \dots, a_n και με τα b_1, b_2, \dots, b_l

$$\begin{pmatrix} a_1, & a_2, & \dots & a_n, & b_1, & b_2, & \dots & b_l \\ 1, & 2, & \dots & n, & n+1, & n+2, & \dots & n+l \end{pmatrix}$$

Όταν επιλέξουμε τα a_1, a_2, \dots, a_n όλα τα σύνολα είναι ακάλυπτα και το καθένα έχει συνεισφέρει n στην αντικειμενική. Μόλις αρχίσουμε να βάζουμε τα b_1, b_2, \dots, b_l καλύπτουμε ένα σύνολο την φορά μέχρι να τα καλύψουμε όλα. Το κόστος αυτής της ακέραιας λύσης θα είναι:

$$(n + 1) + (n + 2) + \dots + (n + l) = \Theta(nl + l^2).$$

Για το ίδιο στιγμιότυπο κατασκευάζουμε και μια μη ακέραια λύση που σέβεται του περιορισμούς του LP A. Η πραγματική λύση είναι ίδια με την ακέραια, διαφέρει όμως στο κόστος.

Ορίζουμε τα x_{et} σαν:

$$x_{a,t} = 1 \text{ για } t \in [n] \text{ και } x_{b_t(n+1)} = 1 \text{ για } t \in [l]$$

και τα y_{St} σαν:

$$y_{St} = \min \left(1, \frac{1}{n+1} \sum_{e \in S} \sum_{t' < t} x_{et'} \right) \quad \forall S \in \mathbf{S}, \forall t \in [n].$$

Στα πρώτα n βήματα που επιλέγουμε τα a_1, a_2, \dots, a_n , το κάθε σύνολο θα έχει κόστος:

$$1 + \left(1 - \frac{1}{n+1}\right) + \left(1 - \frac{2}{n+1}\right) + \dots + \left(1 - \frac{n-1}{n+1}\right) \approx \frac{n}{2} = \Theta(n)$$

Στη συνέχεια, επιλέγουμε τα b_1, b_2, \dots, b_l , καλύπτοντας ένα σύνολο την φορά, όμως αυτή την φορά κάθε σύνολο είναι ακάλυπτο μόνο κατά ένα ποσοστό $\frac{1}{n+1}$.

$$\frac{1}{n+1} \sum_{i=1}^l i = \Theta(nl + l^2)$$

Αθροίζοντας, το συνολικό κόστος της πραγματικής λύσης είναι: $\Theta \left(nl + \frac{1}{n+1} (nl + l^2) \right)$. Αν θέσουμε $n = \sqrt{l}$ το κόστος ακέραιας λύσης γίνεται $O(l^2)$ ενώ το κόστος πραγματικής λύσης γίνεται $O(l^{\frac{3}{2}})$. Ο λόγος του κόστους των δυο λύσεων είναι $\Omega(\sqrt{l})$. Εμείς θέλουμε όμως να είναι σταθερός, δηλαδή $O(1)$. Αυτός είναι και ο λόγος που απορρίπτουμε το LP A.

3.1.2 Το γραμμικό πρόγραμμα B (LP B)

Στο LP B αντιμετωπίζουμε με επιτυχία το πρόβλημα που προέκυψε στο LP A, προσθέτοντας περισσότερους περιορισμούς. Το LP B είναι το ακόλουθο:

$$\begin{aligned} \min & \sum_{1 \leq t \leq |U|} \sum_{S \in \mathbf{S}} (1 - y_{St}) \\ & \sum_{e \in U} x_{et} = 1 & \forall t \in [n] \\ & \sum_{t \in [n]} x_{et} = 1 & \forall e \in U \\ & \sum_{e \in S \setminus A} \sum_{t' < t} x_{et'} \geq (K(S) - |A|) \cdot y_{St} & \forall S \in \mathbf{S}, \forall A \subseteq \mathbf{S}, \forall t \in [n] \\ & x_{et}, y_{St} \in [0, 1] & \forall e \in U, S \in \mathbf{S}, t \in [n] \end{aligned} \quad (5)$$

Το LP B είναι ίδιο με το LP A εκτός από την γραμμή (5). Οι περιορισμοί στην (5) απαιτούν ότι αν $y_{St} = 1$ (το S έχει καλυφθεί τη χρονική μονάδα t), τότε για κάθε υποσύνολο στοιχείων $A \subseteq S$, τουλάχιστον $K(S) - |A|$ στοιχεία θα να έχουν επιλεγεί από το σύνολο S/A πριν τη

t . Οι επιπλέον περιορισμοί δεν επιτρέπουν στο y_{St} να μικραίνει κατά $\frac{1}{(n-1)}$ κάθε φορά που καλύπτεται ένα στοιχείο του S .

Για την ειδική περίπτωση $K(S) = |S|$, για κάθε $S \in \mathbf{S}$ που προκύπτει το Min-Latency Set Cover το LP B, έχει αποδειχθεί πως έχει integrality gap 2 [18]. Αντίστοιχα, και για την άλλη ακραία περίπτωση, το Min-Sum Set Cover έχει αποδειχθεί πως το LP B έχει integrality gap 4 [14].

Separation oracle (μαντείο διαχωρισμού)

Το Γραμμικό πρόγραμμα B έχει εκθετικό πλήθος περιορισμών. Ωστόσο, μπορεί να λυθεί αποδοτικά με την μέθοδο των ελλειψοειδών (ellipsoid method) [17], αρκεί, σε πολυωνυμικό χρόνο, να μπορούμε να επαληθεύσουμε αν μια υποψήφια λύση (x, y) ικανοποιεί όλους τους περιορισμούς (separation oracle). Πράγματι, για οποιαδήποτε μη ακέραια λύση (x, y) , οι (1),(2) και (4) μπορούν να επαληθευθούν σε $O(mn + n^2)$. Τι γίνεται όμως με τους (5) που είναι εκθετικού πλήθους? Για κάθε σύνολο $S \in \mathbf{S}$, κάθε χρονική μονάδα t και $a < K(S)$. Για να επιβεβαιωθούν οι (5), θέλουμε να ελέγξουμε την ακόλουθη προϋπόθεση.

$$\min_{A:|A|=a} \sum_{e \in S/A} \sum_{t' < t} x_{et'} \geq (K(S) - a) \cdot y_{St} \quad (6)$$

Για κάθε σύνολο A μεγέθους a , η αριστερή μεριά της παραπάνω ανισότητας μπορεί να γραφεί ως:

$$\sum_{e \in S} \sum_{t' < t} x_{et'} - \sum_{e \in A} \sum_{t' < t} x_{et'}. \quad (7)$$

Αν η (7) ισχύει όταν το A είναι το σύνολο των a στοιχείων με τις μεγαλύτερες τιμές του

$$\sum_{t' < t} x_{et'}$$

τότε θα ισχύει και για κάθε άλλο σύνολο A . Άρα μπορούμε να επαληθεύσουμε τους (5) ελέγχοντας μόνο τους $O(mn^2)$ από τους $O(mn \cdot 2^n)$ περιορισμούς.

Ο αλγόριθμος

Ο αλγόριθμος έχει $O(\log n)$ επαναλήψεις, με την επανάληψη i να εκτελείται στο διάστημα $[1, 2^i]$. Σε κάθε επανάληψη i κάνει στρογγυλοποίηση (rounding), στη βέλτιστη λύση (x^*, y^*) του γραμμικού προγράμματος B, μόνο για το διάστημα $[1, 2^i]$ και στη συνέχεια κατασκευάζει το σύνολο O_i . Η ρουτίνα στρογγυλοποίησης για την i επανάληψη που δημιουργεί το σύνολο O_i είναι η ακόλουθη:

- 1 **θέσε** $t_i = 2^i$
- 2 **θέσε** $z_{e,i} \leftarrow \sum_{t' < t_i} x_{et'}^* \quad \forall e \in U$
- 3 **θέσε** $p_{e,i} \leftarrow \min(1, 8z_{e,i}) \quad \forall e \in U$
- 4 **διάλεξε** κάθε ένα στοιχείο $e \in U$ με πιθανότητα $p_{e,i}$
- 5 **θέσε** O_i το σύνολο με τα επιλεγμένα στοιχεία
- 6 **αν** $|O_i| > 16 \cdot 2^i$ **τότε** κράτα μόνο $16 \cdot 2^i$ στοιχεία
- 7 **επέστρεψε** O_i

Η έξοδος του αλγορίθμου είναι η παράθεση των λύσεων κάθε επανάληψης

$$O_1, O_2, \dots, O_{\lceil \log n \rceil}$$

κρατώντας μόνο την πρώτη εμφάνιση κάθε στοιχείου (για την διευκόλυνση την ανάλυσης του αλγορίθμου θα υποθέτουμε ότι ένα στοιχείο μπορεί να εμφανιστεί πολλές φορές). Προσέξτε πως τα O_i δεν είναι διατάξεις, τα στοιχεία τους είναι σε τυχαία σειρά.

Ανάλυση του αλγορίθμου

Οι συγγραφείς του [3] θέλησαν η ανάλυση να είναι απλή και κατανοητή, αφού πρωταρχικός σκοπός τους ήταν να δείξουν ότι το Preemptive Generalized Min-Sum Set Cover έχει αλγόριθμο με σταθερή εγγύηση απόδοσης. Για αυτό το λόγο δεν προχώρησαν στην βελτιστοποίηση των αποτελεσμάτων.

Ξεκινάμε την ανάλυση με την παρατήρηση ότι:

$$y_{St}^* \geq y_{St'}^* \quad \text{για κάθε σύνολο } S \in \mathbf{S} \text{ και } 1 \leq t' \leq t \leq n \quad (8)$$

Στη μη ακέραια λύση (x^*, y^*) τα y_{St}^* και τα x_{et}^* δεν πάνε απότομα από 0 στο 1 αλλά σταδιακά, αυξάνουν όσο μεγαλώνει το t . Το y_{St}^* εξαρτάται από τα x_{et}^* , $e \in S$ εξαιτίας των περιορισμών (5), όσο μεγαλώνει το t ένα μεγαλώνει και το άλλο.

Για κάθε σύνολο $S \in \mathbf{S}$, το t_S^* ορίζεται ως η τελευταία χρονική μονάδα t τέτοια ώστε $y_{St}^* \leq \frac{1}{2}$. Το ακόλουθο συμπέρασμα έπεται από το γεγονός ότι ένα σύνολο S συνεισφέρει στην αντικειμενική τουλάχιστον $(1 - y_{St}^*) \geq \frac{1}{2}$ για κάθε χρονική μονάδα $t \in [1, t_S^*]$.

$$LPOpt \geq \frac{1}{2} \sum_{S \in \mathbf{S}} t_S^* \quad (9)$$

Η (9) είναι ένα κάτω φράγμα για το κόστος την βέλτιστης λύσης $LPOpt$ του (LP B). Το κόστος της αντικειμενικής είναι τουλάχιστον $\frac{1}{2} \sum_{S \in \mathbf{S}} t_S^*$ όταν έχουμε αθροίσει τους ορούς $(1 - y_{St}^*)$, μόνο για $y_{St}^* \leq \frac{1}{2}$ για κάθε S . Η $LPOpt$ όμως μετράει τους όρους μέχρι τα y_{St}^* να γίνουν 0 για όλα τα $S \in \mathbf{S}$.

Λήμμα 3.2. Για ένα σύνολο S και μια επανάληψη i τέτοια ώστε $t_S^* \in [1, t_i)$, η πιθανότητα

$K(S)$ στοιχεία του S να μην έχουν επιλεγεί στο βήμα 4 της επανάληψης i είναι το πολύ $e^{-9/8}$.

Απόδειξη. Για ένα σύνολο S ορίζουμε το υποσύνολο $S_g = \{e \in S \mid z_{e,i} \geq 1/8\}$. Στο βήμα 3 της ρουτίνας στρογγυλοποίησης, όλα στοιχεία που ανήκουν στο S_g θα έχουν $p_{e,i} = 1$ και θα επιλεχθούν σίγουρα στο βήμα 4, ενώ τα στοιχεία $e \in S \setminus S_g$ επιλέγονται ανεξάρτητα με πιθανότητα $p_{e,i} = 8z_{e,i} < 1$ το καθένα. Επομένως, αν $|S_g| \geq K(S)$ είναι προφανές ότι το Λήμμα 3.2 ισχύει. Τι γίνεται στην περίπτωση που $|S_g| < K(S)$; Έστω σύνολο S και επανάληψη i , ώστε $t_S^* \in [1, t_i)$. Αφού t_S^* ήταν η τελευταία χρονική μονάδα που $y_{St}^* \leq \frac{1}{2}$ τότε ισχύει ότι $y_{St_i}^* > \frac{1}{2}$. Αν θέσουμε $A = S_g$ στην (5) έχουμε:

$$\begin{aligned} \sum_{e \in S \setminus S_g} z_{e,i} &= \sum_{e \in S \setminus S_g} \sum_{t' < t_i} x_{et'}^* \geq (K(S) - |S_g|) \cdot y_{St_i}^* \\ &\geq \frac{1}{2}(K(S) - |S_g|). \end{aligned}$$

Ως εκ τούτου ο αναμενόμενος αριθμός στοιχείων του $S \setminus S_g$ που επιλέγονται στην επανάληψη i είναι:

$$\sum_{e \in S \setminus S_g} 8z_{e,i} \geq 4(K(S) - |S_g|).$$

Αφού το κάθε στοιχείο επιλέγεται ανεξάρτητα, μπορούμε να χρησιμοποιήσουμε το ακόλουθο Chernoff bound [31] (θεώρημα 4.2): Αν X_1, X_2, \dots, X_n είναι ανεξάρτητες τυχαίες μεταβλητές που περνούν τιμές 0 ή 1, με $X = \sum_i X_i$ και $\mathbb{E}[X] = \mu$ τότε ισχύει:

$$Pr[X < \mu(1 - \beta)] \leq \exp\left(-\frac{\beta^2}{2}\mu\right).$$

Έχουμε $\mu \geq 4(K(S) - |S_g|) \geq 4$, αν διαλέξουμε $\beta = \frac{3}{4}$ μπορούμε να φράξουμε την πιθανότητα λιγότερα από $(K(S) - |S_g|)$ στοιχεία να επιλεχθούν από το σύνολο S/S_g , με $\exp\left(-\frac{(3/4)^2}{2} \cdot 4\right) = e^{-9/8}$. Αφού τα στοιχεία στο S_g επιλέγονται με πιθανότητα ίση με 1 έπεται ότι η πιθανότητα, λιγότερα από $K(S)$ στοιχεία να επιλεχθούν από το S , είναι και αυτή, το πολύ $e^{-9/8}$. \square

Λήμμα 3.3. Η πιθανότητα ότι θα απορριφθούν στοιχεία στο βήμα 6 είναι το πολύ e^{-6} .

Απόδειξη. Για να αποδείξουμε το Λήμμα 3.3 χρησιμοποιούμε την ανισότητα [6] (Θεώρημα 1, Παρατήρηση 3): Αν X_1, X_2, \dots, X_n είναι ανεξάρτητες τυχαίες μεταβλητές που περνούν τιμές 0 ή 1 με $X = \sum_i X_i$ και $\mathbb{E}[X] = \mu$ τότε:

$$Pr[X > \mu + \beta] \leq \exp\left(-\frac{\beta^2}{2\mu + 2\beta/3}\right). \quad (10)$$

Η πιθανότητα με την οποία ένα στοιχείο e επιλέγεται να μπει στο O_i είναι το πολύ 8 φορές το ποσοστό του στοιχείου που έχει επιλεγεί στο διάστημα $[1, 2^i)$, στην μη ακέραια λύση. Ο αναμενόμενος αριθμός στοιχείων που θα μπουκ στο O_i είναι το πολύ $8 \cdot 2^i$. Αν αν θέσουμε

$\beta = 8 \cdot 2^i$ και $\mu \leq 8 \cdot 2^i$ στην ανισότητα (10), έχουμε πως η πιθανότητα, να επιλεγθούν περισσότερα από $16 \cdot 2^i$ στοιχεία στο στάδιο 4 είναι το πολύ $\exp\left(-\frac{64 \cdot 2}{2\mu + 2\beta/3}\right) \leq e^{-6}$. \square

Σειρά έχει να φράξουμε τον χρόνο κάλυψης που επιστρέφει ο αλγόριθμος για ένα σύνολο S .

Θεώρημα 3.1 ([3]). *Ο αναμενόμενος χρόνος κάλυψης ενός συνόλου S είναι το πολύ $O(1)t_S^*$.*

Έστω $\text{Cov}^{\text{alg}}(S)$ ο χρόνος κάλυψης του συνόλου S , που περνούμε από την διάταξη την οποία δίνει σαν έξοδο ο αλγόριθμος. Για ευκολία στην ανάλυση, θεωρούμε πως ένα σύνολο S είναι καλυμμένο κατά την επανάληψη i μόνο αν:

1. $t_S^* \in [1, t_i]$ και
2. δεν έχουν απορριφθεί στοιχεία του S κατά το βήμα 6.

Να σημειωθεί πως, αν το σύνολο S καλύπτεται χωρίς να προκύπτουν τα δυο παραπάνω κριτήρια, ο χρόνος κάλυψης του μόνο βελτιώνεται. Ορίζουμε το E_{iS} ως το γεγονός, το σύνολο S να καλύπτεται για πρώτη φορά κατά την επανάληψη i και ταυτόχρονα να πληρούνται και οι δυο παραπάνω προϋποθέσεις. Αφού το S έχει καλυφθεί στην i επανάληψη, ο χρόνος κάλυψης του είναι το πολύ $\sum_{j=1}^i 16 \cdot 2^j \leq 2 \cdot 16 \cdot 2^i$. Επιπλέον, γνωρίζουμε ότι όλα τα σύνολα θα έχουν καλυφθεί μέχρι το στάδιο $\lceil \log n \rceil$ γιατί οι περιορισμοί (1) και (2) επιβάλλουν ότι όλα τα στοιχεία θα έχουν επιλεγθεί πλήρως όταν $t = n$. Έτσι καταλήγουμε πως ο αναμενόμενος χρόνος κάλυψης ενός συνόλου S θα είναι:

$$E \left[\text{Cov}^{\text{alg}}(S) \right] \leq \sum_{i=\lceil \log t_S^* \rceil}^{\lceil \log n \rceil + 1} (2 \cdot 16 \cdot 2^i) \times \text{Pr}[E_{iS}]. \quad (11)$$

Το γεγονός E_{iS} , το σύνολο S να καλύπτετε για πρώτη φορά κατά την i επανάληψη είναι γνήσιο υποσύνολο του γεγονότος, το S δεν είναι καλυμμένο στα στάδια $\lceil \log t_S^* \rceil, (\lceil \log t_S^* \rceil + 1), \dots, (i - 2)$ και $(i - 1)$, αλλά, το γεγονός, το σύνολο S να μην έχει καλυφθεί κατά την i επανάληψη πραγματοποιείται μόνο αν

1. $K(S)$ στοιχεία του S δεν έχουν επιλεγεί για το O_i , ή
2. να έχουν απορριφθεί στοιχεία από το O_i κατά το βήμα 6.

Το γεγονός 1. βάση του Λήμματος 3.2 πραγματοποιείται με πιθανότητα $e^{9/8}$ ενώ το γεγονός 2. πραγματοποιείται με πιθανότητα e^{-6} όπως είδαμε στο Λήμμα 2. Επομένως, η πιθανότητα σε μια συγκεκριμένη επανάληψη το S να μην είναι καλυμμένο είναι το πολύ $e^{9/8} + e^{-6} < e^{-1}$ όποτε έχουμε

$$\text{Pr}[E_{iS}] \leq \exp(-(i - \lceil \log t_S^* \rceil)).$$

Αντικαθιστώντας στην ανισότητα (11) παίρνουμε:

$$\begin{aligned}
 E \left[\text{Cov}^{\text{alg}}(S) \right] &\leq \sum_{i=\lceil \log t_S^* \rceil}^{\lceil \log n \rceil} (2 \cdot 16 \cdot 2^i) \times e^{-(i-\lceil \log t_S^* \rceil)} \\
 &= 32 \cdot 2^{\lceil \log t_S^* \rceil} \times \sum_{i=\lceil \log t_S^* \rceil}^{\lceil \log n \rceil} (2/e)^{(i-\lceil \log t_S^* \rceil)} \\
 &\leq \left(64 \cdot \frac{e}{e-2} \right) \times t_S^*.
 \end{aligned}$$

□

Εξαιτίας της γραμμικότητας της μέσης τιμής, ο χρόνος κάλυψης όλων των συνόλων μαζί, είναι το πολύ $\frac{64e}{e-2} \sum_S t_S^*$ έτσι από την (9) καταλήγουμε ότι

$$\frac{128e}{e-2} \text{LPOpt} \leq 485 \text{LPOpt}.$$

Ο αναμενόμενος χρόνος κάλυψης που θα δώσει ο αλγόριθμος είναι το πολύ 485 φορές μεγαλύτερος από την λύση του γραμμικού προγράμματος LPOpt. Παρατηρήστε πως ο αναμενόμενος χρόνος κάλυψης φράσσεται από την LPOpt και όχι από την βέλτιστη λύση Opt.

Ο ΔΕΥΤΕΡΟΣ ΑΛΓΟΡΙΘΜΟΣ

Οι Martin Skutella και David P. Williamson [41] το 2011 βελτίωσαν τον αλγόριθμο κατεβάζοντας την εγγύηση απόδοσης από 485 σε 28. Κράτησαν το ίδιο γραμμικό πρόγραμμα (LP B) και χρησιμοποιούν πολλές ιδέες από την ανάλυση των Nikhil Bansal, Anupam Gupta και Ravishankar Krishnaswamy [3]. Η βασική διάφορα, από την οποία και προέρχεται η βελτίωση της απόδοσης, είναι τα α -points [40]. Σε αυτό το κεφάλαιο θα μελετήσουμε τα αποτελέσματα της δημοσίευσης [41].

Ο αλγόριθμος

Ο αλγόριθμος δεν κάνει πλέον επαναλήψεις. Έτσι, δεν αντιμετωπίζει το πρόβλημα των επανεμφάνισων ή της υπερχειλίσης του σύνολου O_i και την απόρριψη στοιχείων. Χωρίζεται σε δυο φάσεις. Στη πρώτη φάση φουσκώνει την λύση του γραμμικού προγράμματος. Έστω (x^*, y^*) μια βέλτιστη λύση που δίνει το LP B και $Q > 0$ μια σταθερά που θα ορίσουμε αργότερα. Ξεκινάει υπολογίζοντας τα:

$$x = Q \cdot x^* \quad (12)$$

και στη συνέχεια, για $t = 1$ έως $\lfloor \frac{n}{2} \rfloor$ υπολογίζει τα:

$$x_{e,2t} = x_{e,2t} + x_{e,t}. \quad (13)$$

Με αυτόν τον τρόπο φουσκώνει τις τιμές των $x_{e,t}$ όσο το t πλησιάζει το n . Στη δεύτερη φάση, για κάθε $e \in U$ διαλέγει τυχαία και ανεξάρτητα ένα $\alpha_e \in [0, 1]$. και υπολογίζει το t_{e,α_e} που ορίζεται ως η πρώτη χρονική μονάδα t στην οποία $\sum_{t'=1}^t x_{e,t'} \geq \alpha_e$. Η τελική λύση είναι η διάταξη με τα στοιχεία $e \in U$ σε μη φθίνουσα σειρά βάσει των αντίστοιχων t_{e,α_e} . Σε περίπτωση ισότητας δυο ή περισσότερων στοιχείων τα ταξινομεί στην τύχη.

Για (x^*, y^*) μια βέλτιστη λύση του LP B και Q σταθερά, ο αλγόριθμος κάνει συνοπτικά τα εξής βήματα:

- 1 **υπολόγισε** τα $x = Q \cdot x^*$
- 2 **θέσε** $x_{e,2t} = x_{e,2t} + x_{e,t}$ για $t = 1$ έως $\lfloor \frac{n}{2} \rfloor$
- 3 **διάλεξε** τυχαία και ανεξάρτητα $\alpha_e \in [0, 1] \quad \forall e \in U$
- 4 **υπολόγισε** τα $t_{e,\alpha_e} \quad \forall e \in U$
- 5 **ταξινόμησε** τα στοιχεία e σε μη φθίνουσα σειρά βάση των t_{e,α_e}
- 6 **επέστρεψε** την διάταξη με τα ταξινομημένα στοιχεία e

Ανάλυση του αλγορίθμου

Ξεκινάμε την ανάλυση με το παρακάτω λήμμα που αφορά την πρώτη φάση του αλγορίθμου, δηλαδή τα βήματα 1 και 2.

Λήμμα 4.1. Για κάθε $t \in [n]$

$$\sum_{t'=1}^t \sum_{e \in U} x_{e,t'} \leq 2 \cdot Q \cdot t. \quad (14)$$

Επιπλέον, για κάθε $e \in U$ και $t \leq \lfloor \frac{n}{2} \rfloor$

$$\sum_{t'=t+1}^{2t} x_{e,t'} \geq Q \sum_{t'=1}^t x_{e,t'}^*, \quad (15)$$

και για κάθε $t \in [n]$

$$\sum_{t'=1}^t x_{e,t'} \geq Q \sum_{t'=1}^t x_{e,t'}^*. \quad (16)$$

Απόδειξη. Αρχικά, ας δούμε πόσο πολύ φουσκώνει ένα x^* κατά την πρώτη φάση του αλγορίθμου εξαιτίας των (12) και (13). Παρατηρήστε πως:

$$x_{e,t'} = Q \sum_{t'' \in I(t')} x_{e,t''}^* \quad (17)$$

όπου

$$I(t') = \{t'' : t' = 2^i \cdot t'' \text{ για κάποια } i \geq 0\}$$

Το $I(t')$ είναι στην ουσία το υποσύνολο των χρονικών μονάδων t'' στις οποίες τα $x_{e,t''}^*$ συνεισφέρουν στο $x_{e,t'}$. Για μια επιλεγμένη χρονική μονάδα $t \in [n]$ και $t'' \leq t$, ορίζουμε αντίστοιχα και το $J(t'')$ να είναι το υποσύνολο όλων των $t' \leq t$ στις οποίες το $x_{e,t''}^*$ συνεισφέρει στα $x_{e,t'}$.

$$J(t'') = \{t' < t : t' \in I(t'')\}$$

Από τους ορισμούς των $I(t')$ και $J(t'')$ έχουμε

$$\sum_{t'=1}^t |I(t')| = \sum_{t''=1}^t |J(t'')|,$$

που είναι το σύνολο όλων των προσθετέων μέχρι και την χρονική μονάδα t . Επίσης προσέξτε πως:

$$|J(t'')| = 1 + \left\lfloor \log \frac{t}{t''} \right\rfloor.$$

Αφού τα t' στα οποία συνεισφέρει το t'' είναι της μορφής $2^i t'' \leq t$. Επομένως

$$\begin{aligned} \frac{1}{Q} \sum_{t'=1}^t \sum_{e \in U} x_{e,t'} &= \sum_{t'=1}^t \sum_{t'' \in I(t')} \underbrace{\sum_{e \in U} x_{e,t''}^*}_{=1} = \sum_{t'=1}^t |I(t')| = \sum_{t''=1}^t |J(t'')| \\ &= t + \sum_{t''=1}^t \left\lfloor \log \left(\frac{t}{t''} \right) \right\rfloor \leq t + \int_0^t \left\lfloor \log \left(\frac{t}{\theta} \right) \right\rfloor d\theta \\ &= t + \sum_{i=0}^{\infty} \int_{t/2^{i+1}}^{t/2^i} \left\lfloor \log \left(\frac{t}{\theta} \right) \right\rfloor d\theta = t + \sum_{i=0}^{\infty} \frac{t}{2^{i+1}} \cdot i = 2t. \end{aligned}$$

Έτσι ολοκληρώσαμε την απόδειξη της (14). Η (15) έπεται από την (17), απλώς προσέξτε πως για κάθε $t'' \in \{1, \dots, t\}$ υπάρχει μια $t' \in \{t+1, \dots, 2t\}$ τέτοια ώστε $t'' \in I(t')$. Τέλος, και η (16) έπεται από την (17) αφού $t' \in I(t')$ για κάθε t' . \square

Η δεύτερη φάση του αλγορίθμου είναι η φάση των α -points. Θυμίζουμε πως ο αλγόριθμος διαλέγει τυχαία και ανεξάρτητα ένα $\alpha_e \in [0, 1]$ για κάθε $e \in U$. Στη συνέχεια, υπολογίζει τα t_{e,α_e} που δηλώνουν την πρώτη χρονική μονάδα t στην οποία $\sum_{t'=1}^t x_{e,t'} \geq \alpha_e$. Η τελική λύση είναι μια ταξινόμηση των στοιχείων $e \in U$ σε μη φθίνουσα διάταξη βάση των t_{e,α_e} .

Έστω $S \in \mathbf{S}$ και $\alpha \in (0, 1)$ ορίζουμε το $t_{S,\alpha}$ να είναι η τελευταία χρονική μονάδα t στην οποία το $y_{S,t}^* < \alpha$. Στην ανάλυση του πρώτου αλγορίθμου ο Bansal και η ομάδα του για αποδείξουν το κάτω φράγμα της *LPort* στην (9) είχαν επιλέξει $\alpha = \frac{1}{2}$. Το κόστος με το οποίο ένα σύνολο S θα επιβαρύνει την αντικειμενική συνάρτηση είναι:

$$C_S^{LP} = \sum_{t \in [n]} (1 - y_{S,t}^*) = \int_0^1 t_{S,\alpha} d\alpha. \quad (18)$$

Η 18 ισχύει αφού στο $\int_0^1 t_{S,\alpha} d\alpha$ μετράμε πόσες φορές, δηλαδή για πόσες χρονικές μονάδες $(1 - y_{S,t}^*) = \alpha$ για όλα τα $\alpha \in [0, 1]$.

Σειρά έχει να φράξουμε την πιθανότητα, να έχουμε λιγότερα από $K(S)$ στοιχεία από S με $t_{e,\alpha_e} \leq t_{S,\alpha}$, σε σχέση με τα Q και α .

Λήμμα 4.2. Έστω $Q \cdot \alpha \geq 1$, ορίζουμε p τέτοιο ώστε

$$p = \exp \left(-\frac{1}{2} \cdot \left(1 - \frac{1}{Q \cdot \alpha} \right)^2 \cdot Q \cdot \alpha \right) \leq 1$$

Τότε για ένα ακέραιο $i \geq 0$

$$Pr[\{e \in S : t_{e,\alpha_e} \leq 2^i \cdot t_{S,\alpha}\} < K(S)] \leq p^{i+1}$$

Απόδειξη. Η απόδειξη του λήμματος ακολουθεί εν μέρει την ανάλυση του Λήμματος 3.2

του πρώτου αλγορίθμου. Ορίζουμε το σύνολο:

$$A = \left\{ e \in S : \sum_{t' < 2^i \cdot t_{S,\alpha}} x_{e,t'} \geq 1 \right\},$$

που είναι το αντίστοιχο του συνόλου S_g στην ανάλυση πρώτου αλγορίθμου. Είναι προφανές πως για κάθε $e \in A$ ισχύει ότι $Pr[t_{e,\alpha_e} \leq 2^i \cdot t_{S,\alpha}] = 1$ αφού τα στοιχεία που ανήκουν στο A έχουν επιλεγθεί πλήρως πριν ή κατά την χρονική μονάδα $t_{S,\alpha}$. Από την (5) του γραμμικού προγράμματός B παίρνουμε:

$$\sum_{e \in S \setminus A} \sum_{t' \leq t_{S,\alpha}} x_{e,t'}^* \geq (K(S) - |A|) \cdot y_{S,1+t_{S,\alpha}}^* \geq (K(S) - |A|) \cdot \alpha.$$

Έτσι από την (16) έχουμε:

$$\sum_{e \in S \setminus A} \sum_{t' \leq t_{S,\alpha}} x_{e,t'} \geq (K(S) - |A|) \cdot Q \cdot \alpha.$$

Τέλος, έπεται από αναγωγή στο i και τις (16) και (15) ότι:

$$\sum_{e \in S \setminus A} \sum_{t' \leq 2^i \cdot t_{S,\alpha}} x_{e,t'} \geq (i+1) \cdot (K(S) - |A|) \cdot Q \cdot \alpha.$$

Για κάθε $e \in S \setminus A$ ορίζουμε την τυχαία μεταβλητή X_e που είναι 1 αν $t_{e,\alpha_e} \leq 2^i \cdot t_{S,\alpha}$ και σε αντίθετη περίπτωση 0. Προσέξτε πως $Pr[X_e = 1] = \sum_{t' \leq 2^i \cdot t_{S,\alpha}} x_{e,t'}$, όσο μεγαλύτερο ποσοστό του e έχει επιλεγθεί τόσο μεγαλύτερη είναι η πιθανότητα να συμβεί το ενδεχόμενο $X_e = 1$. Έστω $X = \sum_{e \in S \setminus A} X_e$ και $\mu = E[X]$ τότε έχουμε:

$$\mu = E[X] = \sum_{e \in S \setminus A} \sum_{t' \leq 2^i \cdot t_{S,\alpha}} x_{e,t'} \geq (i+1) \cdot (K(S) - |A|) \cdot Q \cdot \alpha.$$

Να σημειωθεί πως αν $|A| \geq K(S)$ τότε $Pr[\{e \in S : t_{e,\alpha_e} \leq 2^i \cdot t_{S,\alpha}\} < K(S)] = 0$ για αυτό

Θεωρούμε πως $|A| < K(S)$. Οπότε

$$\begin{aligned}
 Pr[\{\mathbf{e} \in S : t_{e,\alpha_e} \leq 2^i \cdot t_{S,\alpha}\} < K(S)] &= \\
 &= Pr[\{\mathbf{e} \in S \setminus A : t_{e,\alpha_e} \leq 2^i \cdot t_{S,\alpha}\} < K(S) - |A|] \\
 &= Pr[X < K(S) - |A|] \\
 &\leq Pr\left[X < \frac{\mu}{(i+1) \cdot Q \cdot \alpha}\right] = Pr\left[X < \mu \left(1 - \left(1 - \frac{1}{(i+1) \cdot Q \cdot \alpha}\right)\right)\right] \\
 &\leq \exp\left(-\frac{1}{2} \left(1 - \frac{1}{(i+1) \cdot Q \cdot \alpha}\right)^2 \cdot \mu\right) \tag{19}
 \end{aligned}$$

$$\begin{aligned}
 &\leq \exp\left(-\frac{1}{2} \left(1 - \frac{1}{(i+1) \cdot Q \cdot \alpha}\right)^2 \cdot (i+1) \cdot Q \cdot \alpha\right) \\
 &\leq \exp\left(-\frac{1}{2} \left(1 - \frac{1}{Q \cdot \alpha}\right)^2 \cdot (i+1) \cdot Q \cdot \alpha\right) = p^{i+1}. \tag{20}
 \end{aligned}$$

Στην (19) χρησιμοποιούμε το Chernoff bound [31] (κεφαλαίο 4.1)

$$Pr[X < -\mu \cdot (1 - \beta)] \leq \exp\left(-\frac{1}{2} \cdot \beta^2 \cdot \mu\right)$$

και στην (20) το γεγονός ότι η παρακάτω ανισότητα:

$$-\left(1 - \frac{1}{(i+1) \cdot Q \cdot \alpha}\right)^2 \leq -\left(1 - \frac{1}{Q \cdot \alpha}\right)^2$$

ισχύει για $i \geq 0$ και $Q \cdot \alpha \geq 1$. □

Το κόστος με το οποίο ένα σύνολο S επιβαρύνει την αντικειμενική συνάρτηση, όπως αυτό ορίστηκε στην (18), είναι C_S^{LP} . Το συνολικό κόστος της αντικειμενικής συνάρτησης του LP B είναι $OPT_{LP} = \sum_{S \in \mathcal{S}} C_S^{LP}$. Έστω C_S μια τυχαία μεταβλητή που δίνει την θέση του $K(S)$ -οστού στοιχείου ενός συνόλου S στη διάταξη που κατασκευάζει ο αλγόριθμος. Το συνολικό κόστος αυτής της διάταξη θα είναι $\sum_{S \in \mathcal{S}} C_S$. Προχωρούμε λοιπόν στο ακόλουθο λήμμα:

Λήμμα 4.3. Για Q και α τέτοια ώστε $p < \frac{1}{2}$ τότε:

$$E\left[\sum_{S \in \mathcal{S}} C_S\right] \leq \frac{2Q}{1-\alpha} \cdot \frac{1-p}{1-2p} \cdot OPT_{LP} + OPT.$$

Απόδειξη. Έστω t_S η πρώτη χρονική μονάδα στην οποία $|\{\mathbf{e} \in S : t_{e,\alpha_e} \leq t_S\}| \geq K(S)$. Τότε από το Λήμμα 4.2 ξέρουμε ότι η πιθανότητα, ότι $t_S > t_{S,\alpha}$ είναι το πολύ p αφού το ενδεχόμενο $t_S \leq t_{S,\alpha}$ είναι συμπληρωματικό του $|\{\mathbf{e} \in S : t_{e,\alpha_e} \leq 2^i \cdot t_{S,\alpha}\}| < K(S)$. Άρα η πιθανότητα, ότι $t_{S,\alpha} < t_S < 2 \cdot t_{S,\alpha}$ είναι και αυτή το πολύ p . Με την ίδια λογική, η πιθανότητα, ότι $2 \cdot t_{S,\alpha} < t_S < 4 \cdot t_{S,\alpha}$ είναι το πολύ p^2 η πιθανότητα, ότι $4 \cdot t_{S,\alpha} < t_S < 8 \cdot t_{S,\alpha}$ είναι το

πολύ p^3 κι ούτω καθ' εξής, έτσι έχουμε ότι:

$$E[t_S] \leq t_{S,\alpha} + t_{S,\alpha} \sum_{i=0}^{\infty} 2^i \cdot p^{i+1} = t_{S,\alpha} \cdot \left(1 + \frac{p}{1-2p}\right) = t_{S,\alpha} \cdot \frac{1-p}{1-2p} \quad (21)$$

Για όλες χρονικές μονάδες $t \leq t_{S,\alpha}$ ισχύει ότι $1 - y_{S,t}^* > 1 - \alpha$ όποτε $C_S^{LP} \geq t_{S,\alpha} \cdot (1 - \alpha)$ η $t_{S,\alpha} \leq \frac{C_S^{LP}}{(1-\alpha)}$. Επομένως:

$$E[t_S] \leq C_S^{LP} \cdot \frac{1}{1-\alpha} \cdot \frac{1-p}{1-2p}.$$

Παρατηρήστε ότι $C_S \leq |\{e \in U \setminus S : t_{e,\alpha_e} \leq t_S\}| + K(S)$. Με το ίδιο σκεπτικό που αναπτύξαμε προηγουμένως, για ένα στοιχείο $e \notin S$ και μια χρονική μονάδα t η πιθανότητα, ότι $t_{e,\alpha_e} \leq t$ είναι $\min\{1, \sum_{t' \leq t} x_{e,t'}\}$. Χρησιμοποιώντας την (14) έχουμε:

$$E[|\{e \in U \setminus S : t_{e,\alpha_e} \leq t\}|] \leq \sum_{e \in U} \sum_{t' \leq t} x_{e,t'} \leq 2Q \cdot t.$$

Έτσι καταλήγουμε στην:

$$E[C_S] \leq 2Q \cdot E[t_S] + K(S) \leq \frac{2Q}{1-\alpha} \cdot \frac{1-p}{1-2p} \cdot C_S^{LP} + K(S). \quad (22)$$

Από τη οποία προκύπτει ότι:

$$E\left[\sum_{S \in \mathcal{S}} C_S\right] \leq \frac{2Q}{1-\alpha} \cdot \frac{1-p}{1-2p} \cdot OPT_{LP} + OPT.$$

Αφού σε κάθε λύση $\sum_{S \in \mathcal{S}} K(S) \leq OPT$. □

Σειρά έχει να βρούμε τις τιμές των σταθερών που ορίσαμε προηγουμένως έτσι ώστε να έχουμε την καλύτερη εγγύηση απόδοσης. Αν θέσουμε $\alpha = \frac{1}{2}$ (όπως έκανε η ομάδα του Nikhil Bansal) και $Q = 10.05$ τότε $p = 0.1995$ τότε

$$\frac{2Q}{1-\alpha} \cdot \frac{1-p}{1-2p} + 1 \approx 54.54$$

και έχουμε εγγύηση απόδοσης περίπου 55. Ωστόσο θα κάναμε καλύτερα αν επιλέξουμε το α και το Q στην τύχη.

Θεώρημα 4.1 ([41]). *Αν διαλέξουμε α ανεξάρτητα και τυχαία από $(0, 1)$ σύμφωνα με την συνάρτηση κατανομής $f(\alpha) = 2 \cdot \alpha$ και θέσουμε $Q = \frac{z}{\alpha}$ για κάποια κατάλληλα επιλεγμένη σταθερά z τότε ο αλγόριθμος έχει εγγύηση απόδοσης μικρότερη από 27.78.*

Απόδειξη. Προσέξτε πως το $\alpha \cdot Q$ είναι ίσο με τη σταθερά z και το $p = \exp\left(-\frac{1}{2} \left(1 - \frac{1}{2}\right)^2 \cdot z\right)$ εξαρτάται μόνο από την z επομένως είναι και αυτό σταθερό. Στην απόδειξη του Λήμματος 4.3 βρήκαμε φράγματα για στις αναμενόμενες τιμές των t_S και C_S , υποθέτοντας ότι τα α και

Q είναι σταθερά. Ορίζουμε αυτές τις δεσμευμένες αναμενόμενες τιμές με E_α έτσι έχουμε:

$$E_\alpha[t_S] \leq t_{S,\alpha} \cdot \frac{1-p}{1-2p} \quad \text{λόγω της (21),}$$

$$E_\alpha[C_S] \leq 2Q \cdot E_\alpha[t_S] + K(S) \quad \text{λόγω της (22).}$$

Εφαρμόζοντας τον τύπο για την δεσμευμένη αναμενόμενη τιμή

$$\int_B E(X|Y=y) \cdot Pr(Y=y) dy$$

(όπου B το πεδίο τιμών του y) μαζί με την (18) καταλήγουμε ότι:

$$\begin{aligned} E[C_S] &= \int_0^1 E_\alpha[C_S] \cdot f(\alpha) d\alpha \\ &\leq \int_0^1 2\alpha \cdot 2Q \cdot t_{S,\alpha} \cdot \frac{1-p}{1-2p} d\alpha + K(S) \\ &= 4z \cdot \frac{1-p}{1-2p} \int_0^1 t_{S,\alpha} d\alpha + K(S) \\ &= 4z \cdot \frac{1-p}{1-2p} \cdot C_S^{LP} + K(S). \end{aligned}$$

Επομένως παίρνουμε:

$$E \left[\sum_{S \in \mathcal{S}} C_S \right] \leq 4z \cdot \frac{1-p}{1-2p} \cdot OPT_{LP} + OPT \leq \left(1 + 4z \cdot \frac{1-p}{1-2p} \right) \cdot OPT.$$

Αν θέσουμε $z = 5.03$ τότε $p \approx 0.1990$ και η εγγύηση απόδοσης είναι λιγότερη από 27.78

□

Ο ΤΡΙΤΟΣ ΑΛΓΟΡΙΘΜΟΣ

Πρόσφατα οι Sungjin Im, Maxim Sviridenko και Ruben van der Zwaan [20] βελτίωσαν ακόμα περισσότερο την εγγύηση απόδοσης, ρίχνοντάς τη στο 12.4. Ο αλγόριθμος τους διαφέρει πολύ σε σχέση με τους δυο προηγούμενους [3, 41] που, όπως είδαμε, παρουσιάζαν πολλά κοινά σημεία. Κυρία διάφορα είναι πως ασχολούνται με την preemptive εκδοχή του Generalized Min-Sum Set Cover, για την οποία κατασκευάζουν έναν 2-προσεγγιστικό αλγόριθμο. Στη συνέχεια, παρουσιάζουν μια μέθοδο, με την οποία μπορούμε να μετατρέψουμε κάθε λύση του preemptive σε μια λύση του non-preemptive, που έχει κόστος το πολύ 6.2 φορές μεγαλύτερο. Συνδυάζοντας τα παραπάνω περνούμε έναν 12.4-προσεγγιστικό αλγόριθμο για το non-preemptive πρόβλημα. Παρακάτω παρουσιάζονται τα αποτελέσματα της εργασίας [20].

Ένας 2-προσεγγιστικός αλγόριθμος για το Preemptive Generalized Min-Sum Set Cover

Ας θυμηθούμε, εν τάχει, την preemptive εκδοχή του προβλήματος. Στο Preemptive Generalized Min-Sum Set Cover μπορούμε να τεμαχίσουμε τα στοιχεία e_i , $i \in [n]$ σε μικρότερα κομμάτια και να φτιάξουμε μια διάταξη με τα κομματιασμένα στοιχεία στο διάστημα $[0, n]$. Για να το περιγράψουμε αυτό, ορίζουμε τη συνάρτηση $x_e(t) : [0, n] \rightarrow \{0, 1\}$, που δείχνει αν το στοιχείο e έχει επιλεχτεί την χρονική στιγμή t ή όχι. Για την $x_e(t)$ ισχύει ότι, $\int_{t=0}^n x_e(t) dt = 1$ για κάθε $e \in [n]$ και $\sum_{e \in [n]} x_e(t) = 1$ για κάθε $t \in [0, n]$. Ο χρόνος κάλυψης ενός συνόλου $\text{cov}(S)$ είναι η πρώτη χρονική στιγμή όπου $\int_{\tau=0}^t \sum_{e \in S} x_e(\tau) d\tau \geq K(S)$ και στόχος μας είναι να ελαχιστοποιηθεί το $\sum_{S \in \mathcal{S}} \text{Cov}(S)$.

Όταν μιλάμε για την preemptive εκδοχή χρειάζεται να γίνει σαφής η διάφορα μεταξύ χρονικής μονάδας και χρονικής στιγμής. Χρονική μονάδα είναι ένας ακέραιος $t \in [0, n]$ και εκφράζει το χρονικά διάστημα $(t - 1, t]$. Χρονική στιγμή, από την άλλη, είναι ένας πραγματικός αριθμός $t \in \mathbb{R}$ που ανήκει στο διάστημα $[0, n]$. Αν και τις ορίζουμε και τις δύο με t , θα είναι σαφές αν και πότε μιλάμε για χρονική μονάδα ή χρονική στιγμή.

Προχωράμε στο βασικό θεώρημα με την απόδειξη του οποίου θα ασχοληθούμε σε αυτό το κεφαλαίο.

Θεώρημα 5.1 ([20]). *Υπάρχει 2-προσεγγιστικός πιθανοτικός αλγόριθμος για το Preemptive Generalized Min-Sum Set Cover.*

Ένας 2-προσεγγιστικός αλγόριθμος είναι ότι καλύτερο μπορούμε να έχουμε για το συγκεκριμένο πρόβλημα, αφού όπως είδαμε προηγουμένως, δεν μπορεί να υπάρξει αλγόριθμος με μικρότερη εγγύηση απόδοσης, δεδομένου ότι ισχύει η UGC [24].

5.1.1 Το γραμμικό πρόγραμμα με τις μεταθέσεις

Για κάθε $S \in \mathbf{S}$ μια έγκυρη μετάθεση είναι μια αντιστοίχιση των στοιχείων του S σε χρονικές

μονάδες. Αυτή η αντιστοίχιση μπορεί να περιγραφεί με μια συνάρτηση $f_S : S \rightarrow [n]$. Θα εκφράζουμε μια έγκυρη μετάθεση σαν ένα σύνολο διμερών σχέσεων $F = \{(e, f_S(e)) | e \in S\}$. Ορίζουμε το $\mathbf{F}(S)$ ως το σύνολο με όλες τις έγκυρες μεταθέσεις του S . Έστω C_S^F ο χρόνος κάλυψης του S αν χρησιμοποιήσουμε τη μετάθεση F , δηλαδή η πρώτη χρονική στιγμή στην οποία ισχύει ότι $|f_S^{-1}([t'])| \geq K(S)$.

Η ομάδα του Sungjin Im χρησιμοποιεί ένα καινούριο γραμμικό πρόγραμμα πολύ διαφορετικό από το (LP B). Οι συντελεστές της αντικείμενης συνάρτησης είναι τα C_S^F και οι μεταβλητές είναι τα y_S^F . Ένα y_S^F δείχνει το ποσοστό της μετάθεσης F με το οποίο τα στοιχεία του S έχουν επιλεγεί στην τελική διάταξη. Για παράδειγμα αν $y_S^F = 1$ τα στοιχεία του S έχουν επιλεγεί μόνο βάση της μετάθεσης F .

Ξεκινάμε πρώτα από το ακέραιο γραμμικό πρόγραμμα ILP, όπου οι μεταβλητές του περνούν τιμές μόνο 0 ή 1 και λύνει την non-preemptive εκδοχή.

$$\min \sum_{S \in \mathbf{S}} \sum_{F \in \mathbf{F}(S)} C_S^F y_S^F$$

$$\sum_{e \in U} x_{e,t} = 1 \quad \forall t \in [n] \quad (23)$$

$$\sum_{t \in [n]} x_{e,t} = 1 \quad \forall e \in [n] \quad (24)$$

$$\sum_{F \in \mathbf{F}(S)} y_S^F = 1 \quad \forall S \in \mathbf{S} \quad (25)$$

$$\sum_{F \in \mathbf{F}(S), (e,t) \in F} y_S^F = x_{e,t} \quad \forall e, t \in [n], S : e \in S \quad (26)$$

$$x_{e,t} \in \{0, 1\} \quad \forall e, t \in [n]$$

$$y_S^F \in \{0, 1\} \quad \forall S \in \mathbf{S}, F \in \mathbf{F}(S)$$

Το $x_{e,t}$ δείχνει το ποσοστό του στοιχείου $e \in S$ που έχει επιλεγεί τη χρονική μονάδα $t \in [n]$. Οι περιορισμοί (23) και (24) είναι ίδιοι με τους αντιστοιχούς περιορισμούς (1), (2) του LP B. Οι (23) απαιτούν ότι μόνο ένα στοιχείο μπορεί να επιλεγεί σε μια χρονική μονάδα και οι (24) απαιτούν ότι κάθε στοιχείο πρέπει να έχει επιλεγεί μόνο μια φορά. Οι περιορισμοί (25) επιβάλλουν ότι κάθε σύνολο έχει μια μοναδική μετάθεση βάσει της οποίας είναι κατανεμημένα τα στοιχεία του. Τέλος, οι περιορισμοί (26) συνδέουν τα στοιχεία με τις μεταθέσεις. Το ποσοστό που ένα στοιχείο έχει καλυφθεί κατά την χρονική μονάδα t είναι ίσο με το συνολικό ποσοστό όλων των μεταθέσεων που το αντιστοιχίζουν στην t .

Το μη ακέραιο γραμμικό πρόγραμμα LP_{primal} είναι ίδιο με το ILP, μόνο που τώρα οι μετα-

βλητές μπορούν να πάρουν τιμές μεταξύ του 0 και του 1.

$$\min \sum_{S \in \mathbf{S}} \sum_{F \in \mathbf{F}(S)} C_S^F y_S^F$$

(23), (24), (25) και (26)

$$1 \geq x_{e,t} \geq 0$$

$$\forall e, t \in [n]$$

$$1 \geq y_S^F \geq 0$$

$$\forall S \in \mathbf{S}, F \in \mathbf{F}(S)$$

5.1.2 Η εγκυρότητα του νέου LP

Το LP_{primal} είναι ένα έγκυρο LP relaxation για το Generalized Min-Sum Set Cover. Ωστόσο, δεν είναι προφανές ότι είναι έγκυρο LP relaxation και για την preemptive εκδοχή του προβλήματος. Για την απόδειξη, χρησιμοποιούμε δυο διαφορετικά είδη μη ακεραίων διατάξεων. Η πρώτη είναι η *συνεχής* διάταξη. Αυτή ορίζεται από την συνάρτηση αντιστοίχισης $x_e(t) : [0, n] \rightarrow \{0, 1\}$, $e \in [n]$ για την οποία ισχύουν $\sum_{e \in [n]} x_e(t) = 1$ για κάθε $t \in [0, n]$ και $\int_{\tau=0}^n x_e(\tau) d\tau = 1$ για κάθε $e \in [n]$. Θα λέμε ότι η $x_e(t)$, $e \in [n]$ είναι μια εφικτή συνεχής διάταξη εάν οι παραπάνω περιορισμοί ικανοποιούνται. Θυμηθείτε πως όταν ορίζαμε το preemptive πρόβλημα, ο χρόνος κάλυψης $con(S)$ ενός συνόλου S (από εδώ και πέρα θα τον συμβολίζουμε και ως C_S) έχει οριστεί βάσει της συνεχούς διάταξης ως η πρώτη χρονική στιγμή κατά την οποία $\int_{\tau=0}^t \sum_{e \in S} x_e(\tau) d\tau \geq K(S)$. Τη δεύτερη διάταξη θα την αποκαλούμε ως *μη-συνεχής*. Η μη-συνεχής διάταξη ορίζεται από την συνάρτηση αντιστοίχισης $x_{e,t}$, $e, t \in [n]$ για την οποία ισχύει ότι, $\sum_{e \in [n]} x_{e,t} = 1$, $\sum_{t \in [n]} x_{e,t} = 1$ και $0 \leq x_{e,t} \leq 1$ για κάθε $e, t \in [n]$. Αν οι παραπάνω περιορισμοί ικανοποιούνται, τότε η $x_{e,t}$, $e, t \in [n]$ θα είναι μια εφικτή μη-συνεχής διάταξη. Να σημειωθεί ότι η μη-συνεχής εκδοχή δεν είναι ιδανική γιατί δεν μας δείχνει πως είναι καταμεμημένα τα κομμάτια των στοιχείων μέσα σε κάθε χρονική μονάδα και δεν μπορούμε να την χρησιμοποιήσουμε για να υπολογίσουμε τον χρόνο κάλυψης. Χρησιμοποιείται, όμως, στο LP_{primal} και είναι μια χαλάρωση της συνεχής διάταξης.

Θεώρημα 5.2. Για μια εφικτή συνεχής διάταξη $x_e(t)$, $e \in [n]$ το C_S εκφράζει το χρόνο κάλυψης του συνόλου S σε αυτή την διάταξη. Για κάθε $e, t \in [n]$ ορίζουμε την $x_{e,t} = \int_{\tau=t-1}^t x_e(\tau) d\tau$. Η $x_{e,t}$ ικανοποιεί τους περιορισμούς (23) και (24). Επίσης, υπάρχουν y_S^F τα οποία ικανοποιούν τους περιορισμούς (25) και (26), επιπλέον ισχύει ότι:

$$\sum_{S \in \mathbf{S}} \sum_{F \in \mathbf{F}(S)} C_S^F y_S^F \leq \sum_{S \in \mathbf{S}} C_S \quad (27)$$

Το πρώτο μέρος του Θεωρήματος 5.2, ότι η $x_{e,t}$ ικανοποιεί τους περιορισμούς (23) και (24), αποδεικνύεται εύκολα από τις ιδιότητες της συνεχής διάταξης και από τον τρόπο με τον

οπείο ορίσαμε τη $x_{e,t}$. Συγκεκριμένα για τον περιορισμό (23) έχουμε:

$$\sum_{e \in [n]} x_{e,t} = \sum_{e \in [n]} \int_{\tau=t-1}^t x_e(\tau) d\tau = \int_{\tau=t-1}^t \sum_{e \in [n]} x_e(\tau) d\tau = \int_{\tau=t-1}^t 1 d\tau = 1$$

και για τον (24)

$$\sum_{t \in [n]} x_{e,t} = \sum_{t \in [n]} \int_{\tau=t-1}^t x_e(\tau) d\tau = \int_{\tau=0}^n x_e(\tau) d\tau = 1.$$

Για το δεύτερο μέρος του Θεωρήματος, δεν είναι δύσκολο να βρούμε y_S^F που να ικανοποιούν τους (23) έως (26). Παρόλα αυτά, δεν είναι σίγουρο αν ικανοποιούν την (27).

Πρόταση 5.1. Υπάρχουν y_S^F που να ικανοποιούν τους περιορισμούς (23) έως (26) αλλά δεν ικανοποιούν με σταθερό λόγο την (27).

Απόδειξη. Για την απόδειξη θα χρησιμοποιήσουμε το εξής απλό αντιπαράδειγμα. Για είσοδο το στιγμιότυπο:

$$\mathbf{S} = \{S_1 = \{e_1, e_2, e_3\}, S_2 = \{e_4, e_5, e_6\}, S_3 = \{e_7, e_8, e_9\}\}$$

$$\text{και } K(S_1) = K(S_2) = K(S_3) = 1,$$

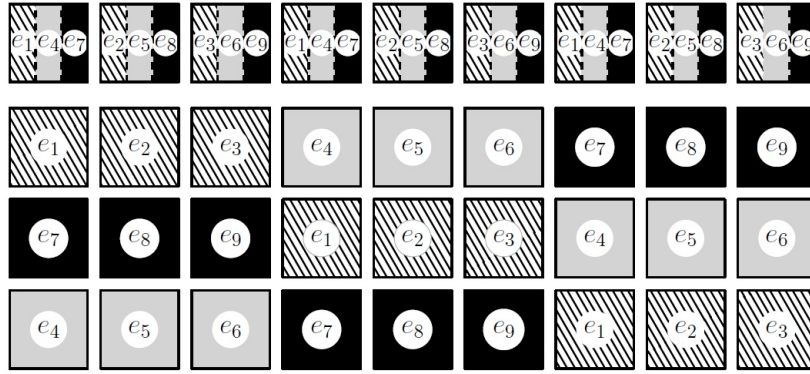
κατασκευάζουμε την μη ακέραια διάταξη:

- $x_{e_1,t} = x_{e_2,t+1} = x_{e_3,t+2} = \frac{1}{3}$ για $t = 1, 4, 7$.
- $x_{e_4,t} = x_{e_5,t+1} = x_{e_6,t+2} = \frac{1}{3}$ για $t = 1, 4, 7$.
- $x_{e_7,t} = x_{e_8,t+1} = x_{e_9,t+2} = \frac{1}{3}$ για $t = 1, 4, 7$.

Τα S_1, S_2, S_3 στην παραπάνω διάταξη έχουν χρόνο κάλυψης το πολύ 3,

$$C_{S_1} = 2 + \frac{1}{3}, C_{S_2} = 2 + \frac{2}{3}, C_{S_3} = 3 \text{ και } \sum_{S \in \mathbf{S}} C_S < 9.$$

Επιλέγουμε τις μεταθέσεις F_S και τα αντίστοιχα y_S^F που ικανοποιούν τους (23) έως (26).



Σχήμα 2: Στην πρώτη γραμμή, έχουμε την μη ακέραια διάταξη. Στη δεύτερη γραμμή, την ακέραια διάταξη που κατασκευάστηκε από τις μεταθέσεις $F_{1,1}, F_{2,2}, F_{3,3}$. Στη τρίτη την διάταξη που κατασκευάστηκε με τις $F_{3,1}, F_{1,2}, F_{2,3}$. Και τέλος, στην τέταρτη, την διάταξη που κατασκευάστηκε με τις $F_{2,1}, F_{3,2}, F_{1,3}$.

- $F_{1,1} = \{(e_1, 1)(e_2, 2)(e_3, 3)\}, F_{1,2} = \{(e_1, 4)(e_2, 5)(e_3, 6)\},$
 $F_{1,3} = \{(e_1, 7)(e_2, 8)(e_3, 9)\}$ και $y_{S_1}^{F_{1,1}} = y_{S_1}^{F_{1,2}} = y_{S_1}^{F_{1,3}} = \frac{1}{3}.$
- $F_{2,1} = \{(e_4, 1)(e_5, 2)(e_6, 3)\}, F_{2,2} = \{(e_4, 4)(e_5, 5)(e_6, 6)\},$
 $F_{2,3} = \{(e_4, 7)(e_5, 8)(e_6, 9)\}$ και $y_{S_2}^{F_{2,1}} = y_{S_2}^{F_{2,2}} = y_{S_2}^{F_{2,3}} = \frac{1}{3}.$
- $F_{3,1} = \{(e_7, 1)(e_8, 2)(e_9, 3)\}, F_{3,2} = \{(e_7, 4)(e_8, 5)(e_9, 6)\},$
 $F_{3,3} = \{(e_7, 7)(e_8, 8)(e_9, 9)\}$ και $y_{S_3}^{F_{3,1}} = y_{S_3}^{F_{3,2}} = y_{S_3}^{F_{3,3}} = \frac{1}{3}.$

Οι παραπάνω F_S και y_S^F μας δίνουν:

$$\begin{aligned}
 \sum_{S \in \mathbf{S}} \sum_{F \in \mathbf{F}(S)} c_S^F y_S^F &= \\
 &= c_{S_1}^{F_{1,1}} y_{S_1}^{F_{1,1}} + c_{S_2}^{F_{2,1}} y_{S_2}^{F_{2,1}} + c_{S_3}^{F_{3,1}} y_{S_3}^{F_{3,1}} + c_{S_1}^{F_{1,2}} y_{S_1}^{F_{1,2}} + c_{S_2}^{F_{2,2}} y_{S_2}^{F_{2,2}} + \\
 &\quad + c_{S_3}^{F_{3,2}} y_{S_3}^{F_{3,2}} + c_{S_1}^{F_{1,3}} y_{S_1}^{F_{1,3}} + c_{S_2}^{F_{2,3}} y_{S_2}^{F_{2,3}} + c_{S_3}^{F_{3,3}} y_{S_3}^{F_{3,3}} \\
 &= 1 \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + 4 \cdot \frac{1}{3} + 4 \cdot \frac{1}{3} + \\
 &\quad + 4 \cdot \frac{1}{3} + 7 \cdot \frac{1}{3} + 7 \cdot \frac{1}{3} + 7 \cdot \frac{1}{3} \\
 &= 1 + 4 + 7 = 12.
 \end{aligned}$$

Όπως βλέπουμε, για το παράδειγμα μας, η (27) δεν ισχύει. Όχι μόνο αυτό, αλλά ακολουθώντας την ίδια ιδέα, μπορούμε να προσαρμόσουμε το στιγμιότυπο κάνοντας την διάφορα οσοδήποτε μεγάλη. □

Θέλουμε να δείξουμε ότι όντως υπάρχουν καλά y_S^F που ικανοποιούν την (27). Για να το πετύχουμε αυτό, θα δείξουμε πως μπορούμε να κατασκευάσουμε y_S^F για τα οποία ισχύει η ανισότητα

$$\sum_{F \in \mathbf{F}} c_S^F y_S^F \leq C_S \quad (28)$$

για κάθε $S \in \mathbf{S}$. Έτσι, αφού, η (28) θα ισχύει για όλα τα σύνολα, θα ισχύει και η (27). Δεδομένου ότι, οι τιμές των y_S^F , για ένα συγκεκριμένο σύνολο, S δεν επηρεάζουν τις τιμές των y_S^F για τα υπόλοιπα σύνολα, θα ασχοληθούμε, για κάθε ένα σύνολο ξεχωριστά, βρίσκοντας y_S^F που ικανοποιούν τους περιορισμούς (25) και (26) όπως και την (28).

Για να το πετύχουμε αυτό χρησιμοποιούμε την θεωρία των μητροειδών που είδαμε συνοπτικά στο εισαγωγικό κεφάλαιο. Ορίζουμε δύο μητροειδή τα M_1 και M_2 , έτσι ώστε κάθε υποσύνολο που βρίσκεται στην τομή των M_1, M_2 και είναι επιπλέον βάση του M_1 , να αντιστοιχεί σε μια εφικτή μετάθεση $F \in \mathbf{F}$. Δείχνουμε ότι το διάνυσμα $x_{e,t}$, $e \in S, t \in [n]$ βρίσκεται στην τομή των πολυτόπων των μητροειδών M_1, M_2 . Δεδομένου ότι, η τομή των πολυτόπων των μητροειδών είναι ακέραια [35, 38, 9], μπορούμε να εκφράσουμε το $x_{e,t}$ σαν κυρτό γραμμικό συνδυασμό των ακέραιων σημείων που βρίσκονται στην τομή των πολυτόπων των M_1 και M_2 . Εξαιτίας του τρόπου με τον οποίο κατασκευάσαμε τα M_1, M_2 , κάθε ακέραιο σημείο αντιστοιχεί σε μια $F \in \mathbf{F}$. Έτσι, εξασφαλίζουμε ότι τα y_S^F ικανοποιούν του περιορισμούς (25) και (26). Στη συνέχεια, ολοκληρώνουμε την ανάλυση δείχνοντας ότι αυτά τα y_S^F ικανοποιούν και την (28).

Το πρώτο μητροειδές M_1 εκφράζει την απαίτηση κάθε στοιχείο του S να αντιστοιχεί το πολύ σε μια χρονική μονάδα. Για ένα συγκεκριμένο σύνολο $S \in \mathbf{S}$ ορίζουμε το μητροειδές $M_1 = (U, \mathbf{I}(M_1))$ με το συμβάν $U = \{(e, t) | e \in S, t \in [n]\}$ και την συλλογή από υποσύνολα $\mathbf{I}(M_1)$ για την οποία ισχύει ότι: $A \in \mathbf{I}(M_1)$ αν και μόνο αν $e \in S, |A \cap \{(e, t) : t \in [n]\}| \leq 1$. Το U αποτελείται από όλα τα δυνατά ζευγάρια μεταξύ στοιχείων του S και χρονικών μονάδων. Κάθε υποσύνολο A έχει το πολύ ένα ζευγάρι για κάθε στοιχείο $e \in S$. Αν χωρίσουμε τα ζευγάρια του U βάσει των κοινών στοιχείων στα σύνολα $U_{e1}, U_{e2}, \dots, U_{en}$ και θέσουμε $k_{U_{e1}} = k_{U_{e2}} = \dots = k_{U_{en}} = 1$ παρατηρούμε ότι το M_1 είναι ένα partition μητροειδές. Για αυτό το λόγο το πολυμητροειδές $P(M_1)$ του M_1 είναι:

$$\begin{aligned} \sum_{t \in [n]} x_{e,t} &\leq 1 && \forall e \in S \\ x_{e,t} &\geq 0 && \forall e \in S, t \in [n]. \end{aligned}$$

Πρόταση 5.2. Το διάνυσμα $x = (x_{e,t})$, $e \in S, t \in [n]$ βρίσκεται μέσα στο πολύτοπο $P(M_1)$ Επιπλέον, ισχύει ότι $\sum_{e \in S, t \in [n]} x_{e,t} = |S|$, που σημαίνει ότι το x ανήκει σε μια βάση του πολυμητροειδούς του M_1 .

Απόδειξη. Από τον ορισμό του $x_{e,t}$ έχουμε πως για κάθε $S \in \mathbf{S}$ ισχύει:

$$\sum_{t \in [n]} x_{e,t} = \sum_{t=1}^n \int_{\tau=t-1}^t x_e(\tau) d\tau = \int_{\tau=0}^n x_e(\tau) d\tau = 1.$$

□

Το δεύτερο μητροειδές M_2 είναι πιο περίπλοκο. Εκφράζει την απαίτηση, σε κάθε χρονική μονάδα να αντιστοιχεί το πολύ ένα στοιχείο του S . Επιπλέον, εκφράζει ότι το πολύ $K(S)$ στοιχεία μπορούν να επιλεγούν κατά τις πρώτες $C - 1$ χρονικές μονάδες και το πολύ $|S| - K(S)$ στοιχεία επιλέγονται κατά τις χρονικές μονάδες $C + 1, C + 2, \dots, n$, όπου C ένας ακέραιος τέτοιος ώστε $C - 1 < C_S \leq C$. Το μητροειδές $M_2 = (U, I(M_2))$ ορίζεται από το συμβάν $U = \{(e, t) | e \in S, t \in [n]\}$, που είναι το ίδιο με του M_1 , και την συλλογή από υποσύνολα $I(M_2)$ για την οποία ισχύει ότι $A \in I(M_2)$, αν και μόνο αν, το A ικανοποιεί τα παρακάτω:

- Για κάθε χρονική μονάδα $t \in [n]$, $|A \cap \{(e, t) | e \in S\}| \leq 1$.
- $|A \cap \{(e, t) | e \in S, 1 \leq t \leq C - 1\}| \leq K(S)$.
- $|A \cap \{(e, t) | e \in S, C + 1 \leq t \leq n\}| \leq |S| - K(S)$.

Αν χωρίσουμε το U σε σύνολα των οποίων τα ζευγάρια έχουν κοινές χρονικές μονάδες $U_{t_1}, U_{t_2}, \dots, U_{t_n}$, τότε κάθε υποσύνολο A θα περιέχει το πολύ ένα ζευγάρι από το καθένα, δηλαδή $k_{U_{t_1}} = k_{U_{t_2}} = \dots = k_{U_{t_n}} = 1$. Επιπλέον, χωρίζουμε το U στο σύνολο U_{C-1} που έχει όλα τα ζευγάρια με χρονικές μονάδες $t = 1, 2, \dots, C - 1$ και στο σύνολο U_{C+1} που έχει όλα τα ζευγάρια με χρονικές μονάδες $t = C + 1, C + 2, \dots, n$. Οι δύο τελευταίες ανισότητες φράσσουν τον αριθμό των ζευγαριών, που μπορεί να περιέχει ένα A , από αυτά τα σύνολα. Έτσι, έχουμε $k_{U_{C-1}} = K(S)$ και $k_{U_{C+1}} = |S| - K(S)$ αντίστοιχα. Από τα παραπάνω καταλήγουμε ότι το M_2 είναι ένα laminaar μητροειδές. Το πολυμητροειδές $P(M_2)$ του M_2 ορίζεται ως εξής:

$$\begin{aligned} \sum_{e \in S} x_{e,t} &\leq 1 && \forall t \in [n] \\ \sum_{t=1}^{C-1} \sum_{e \in S} x_{e,t} &\leq K(S) \\ \sum_{t=C+1}^n \sum_{e \in S} x_{e,t} &\leq |S| - K(S) \\ x_{e,t} &\geq 0 && \forall e \in S, t \in [n]. \end{aligned}$$

Πρόταση 5.3. Το διάνυσμα $x = (x_{e,t})$ βρίσκεται μέσα στο πολύτοπο $P(M_2)$.

Απόδειξη. Ξεκινάμε, αποδεικνύοντας ότι το $x_{e,t}$ ικανοποιεί την πρώτη ανισότητα. Από τον

ορισμό του $x_{e,t}$ έχουμε ότι:

$$\sum_{e \in S} x_{e,t} = \sum_{e \in S} \int_{\tau=t-1}^t x_e(\tau) d\tau = \int_{\tau=t-1}^t \sum_{e \in S} x_e(\tau) d\tau \leq \int_{\tau=t-1}^t 1 d\tau = 1$$

Σειρά έχει η δεύτερη ανισότητα. Θυμηθείτε ότι $C - 1 < C_S \leq C$.

$$\sum_{t=1}^{C-1} \sum_{e \in S} x_{e,t} = \int_{\tau=0}^{C-1} \sum_{e \in S} x_e(\tau) d\tau \leq \int_{\tau=0}^{C_S} \sum_{e \in S} x_e(\tau) d\tau = K(S)$$

Η τελευταία ανισότητα έπεται από τον ορισμό του C_S .

$$\begin{aligned} \sum_{t=C+1}^n \sum_{e \in S} x_{e,t} &= \int_{\tau=C}^n \sum_{e \in S} x_e(\tau) d\tau \leq \int_{\tau=C_S}^n \sum_{e \in S} x_e(\tau) d\tau \\ &= |S| - \int_{\tau=0}^{C_S} \sum_{e \in S} x_e(\tau) d\tau = |S| - K(S). \end{aligned}$$

□

Η τομή δύο πολυμητροειδων είναι ένα ακέραιο πολύτοπο, δηλαδή, τα σημεία του κυρτού του περιβλήματος είναι ακέραια, εφόσον το $(x_{e,t})$ βρίσκεται μέσα στην τομή των πολυτόπων $P(M_1)$, και $P(M_2)$ μπορούμε να το γράψουμε σαν γραμμικό συνδυασμό των σημείων του κυρτού περιβλήματος του $P(M_1) \cap P(M_2)$, τα οποία είναι ακέραια. Κάθε σημείο του κυρτού περιβλήματος αντιστοιχεί σε ένα σύνολο που ανήκει στο $I(M_1) \cap I(M_2)$. Όλα αυτά τα σύνολα έχουν μέγεθος το πολύ $|S|$ εξαιτίας των ιδιοτήτων του M_1 . Για την ακρίβεια, θα έχουν μέγεθος ακριβώς $|S|$, κι αυτό γιατί $\sum_{e \in S} \sum_{t \in [n]} x_{e,t} = |S|$. Από τις ιδιότητες του M_1 και την πρώτη ιδιότητα του M_2 , καταλήγουμε ότι κάθε ακέραιο σημείο του κυρτού περιβλήματος αντιστοιχεί σε μια έγκυρη μετάθεση $F \in \mathbf{F}'(S)$. Έτσι, έχουμε το ακόλουθο λήμμα:

Λήμμα 5.1. Υπάρχει $\mathbf{F}'(S) \subseteq \mathbf{F}(S)$ και θετικές σταθερές θ_S^F , $F \in \mathbf{F}'(S)$ τέτοιες ώστε να ικανοποιούνται τα:

- $\sum_{F \in \mathbf{F}'(S)} \theta_S^F = 1$
- Για κάθε $e \in S, t \in [n]$, $x_{e,t} = \sum_{F \in \mathbf{F}'(S)} \theta_S^F \cdot 1[(e, t) \in F]$

όπου η μεταβλητή $1[(e, t) \in F]$ είναι ίση με 1 αν και μόνο αν $(e, t) \in F$ και 0 σε αντίθετη περίπτωση.

Θέτουμε $y_S^F = \theta_S^F$ για κάθε σύνολο $F \in \mathbf{F}'(S)$ και $y_S^F = 0$ για κάθε $F \in \mathbf{F}(S) \setminus \mathbf{F}'(S)$. Τα $x_{e,t}$ και y_S^F ικανοποιούν τους περιορισμούς (25) και (26).

Μένει λοιπόν να δείξουμε ότι τα y_S^F ικανοποιούν και την (28). Για την απόδειξη αυτή, η

δεύτερη και τρίτη ιδιότητα του M_2 παίζουν σημαντικό ρόλο. Κάνουμε την εξής παρατήρηση.

Λήμμα 5.2. Για κάθε $F \in \mathbf{F}'(S)$ ισχύει, ακριβώς, ένα από τα παρακάτω:

- $|F \cap \{(e, t) | e \in S, 1 \leq t \leq C - 1\}| = K(S)$
- $|F \cap \{(e, t) | e \in S, 1 \leq t \leq C - 1\}| = K(S) - 1$ και $(e, C) \in F$ για κάποιο $e \in S$

Απόδειξη. Ισχύει ότι $|F| = |S|$, δηλαδή, ο αριθμός των ζευγαριών σε μια μετάθεση είναι ίσος με τον αριθμό των στοιχείων του σύνολου S . Από την τρίτη ιδιότητα του M_2 , έχουμε ότι:

$$N_{\leq C+1} = |F \cap \{(e, t) | e \in S, C + 1 \leq t \leq n\}| \leq |S| - K(S).$$

Άρα:

$$N_{\leq C} = |F \cap \{(e, t) | e \in S, C \leq t \leq n\}| \leq |S| - K(S) + 1.$$

Επιπλέον έχουμε:

$$N_{\leq C-1} = |F \cap \{(e, t) | e \in S, 1 \leq t \leq C - 1\}| \geq K(S) - 1.$$

Εξαιτίας της δεύτερης ιδιότητας του M_2 , ισχύει ότι $N_{\leq C-1} \leq K(S)$. Επομένως, έχουμε δυο περιπτώσεις ή $N_{\leq C-1} = K(S)$ ή $N_{\leq C-1} = K(S) - 1$. Στην δεύτερη περίπτωση, αφού $N_{\geq C-1} \leq |S| - K(S)$ καταλήγουμε στο ότι το ζευγάρι (e, C) για κάποιο $e \in S$ ανήκει στην F . \square

Χρησιμοποιώντας το Λήμμα 5.2 μπορούμε πλέον να αποδείξουμε ότι το νέο γραμμικό πρόγραμμα είναι ένα έγκυρο LP relaxation της preemptive εκδοχής του προβλήματος.

Απόδειξη. (Απόδειξη του Θεωρήματος 5.2) Χωρίζουμε το $\mathbf{F}'(S)$ στα $\mathbf{F}'_1(S)$ και $\mathbf{F}'_2(S)$ όπου το $\mathbf{F}'_1(S)$ έχει όλες τις μεταθέσεις, για τις οποίες ισχύει η πρώτη περίπτωση του προηγούμενου λήμματος και $\mathbf{F}'_2(S) = \frac{\mathbf{F}'(S)}{\mathbf{F}'_1(S)}$. Έστω $\theta' = \sum_{F \in \mathbf{F}'_2(S)} \theta_S^F$. Παρατηρήστε πως για κάθε $F \in \mathbf{F}'_1(S)$ ισχύει ότι $C_S^F \leq C - 1$ και για κάθε $F \in \mathbf{F}'_2(S)$, $C_S^F = C$. Δηλαδή, το σύνολο S δεν καλύπτεται νωρίτερα από την χρονική μονάδα $C - 1$ για το $(1 - \theta')$ ποσοστό των μεταθέσεων του $\mathbf{F}'(S)$ και καλύπτεται ακριβώς την χρονική μονάδα C για το ποσοστό θ' των μεταθέσεων του $\mathbf{F}'(S)$. Οπότε έχουμε ότι:

$$\begin{aligned} \sum_{F \in \mathbf{F}(S)} c_S^F y_S^F &= \sum_{F \in \mathbf{F}'(S)} c_S^F \theta_S^F = \sum_{F \in \mathbf{F}'_1(S)} c_S^F \theta_S^F + \sum_{F \in \mathbf{F}'_2(S)} c_S^F \theta_S^F \\ &\leq (1 - \theta')(C - 1) + \theta' C = C - 1 + \theta' \end{aligned} \quad (29)$$

Τώρα, στόχος είναι να βρούμε ένα πάνω φράγμα για το θ' . Από τον ορισμό του C_S και το

γεγονός ότι $\sum_{e \in S} x_e(\tau) \leq 1$ για κάθε τ , γνωρίζουμε ότι:

$$\begin{aligned} \int_{\tau=0}^{C-1} \sum_{e \in S} x_e(\tau) d\tau &= \int_{\tau=0}^{C_S} \sum_{e \in S} x_e(\tau) d\tau - \int_{\tau=C-1}^{C_S} \sum_{e \in S} x_e(\tau) d\tau \\ &\geq K(S) - (C_S - (C - 1)) \end{aligned} \quad (30)$$

Επιπλέον ισχύει ότι:

$$\begin{aligned} \int_{\tau=0}^{C-1} \sum_{e \in S} x_e(\tau) d\tau &= \sum_{t=1}^{C-1} \sum_{e \in S} x_{e,t} \quad [\text{από τον ορισμό του } x_{e,t}] \\ &= \sum_{t=1}^{C-1} \sum_{e \in S} \sum_{F \in \mathbf{F}'(S): (e,t) \in F} y_S^F \quad [\text{το } x \text{ σαν κυρτός γραμμικός συνδυασμός των } y_S^F] \\ &= \sum_{F \in \mathbf{F}'_1(S)} y_S^F \sum_{e \in S} \sum_{\tau=1}^{C-1} 1[(e, \tau) \in F] + \sum_{F \in \mathbf{F}'_2(S)} y_S^F \sum_{e \in S} \sum_{\tau=1}^{C-1} 1[(e, \tau) \in F] \\ &= \sum_{F \in \mathbf{F}'_1(S)} \theta_S^F \cdot K(S) + \sum_{F \in \mathbf{F}'_2(S)} \theta_S^F \cdot (K(S) - 1) \\ &= (1 - \theta') \cdot K(S) + \theta' \cdot (K(S) - 1) = K(S) - \theta' \end{aligned} \quad (31)$$

Από τις (30) και (31) περνούμε ότι $\theta' \leq C_S - (C - 1)$ και συνδυάζοντάς το με την (29) ολοκληρώνουμε την απόδειξη του Θεωρήματος 5.2. \square

5.1.3 Separation oracle (μαντείο διαχωρισμού)

Αφού το γραμμικό πρόγραμμα $\text{LP}_{\text{Primal}}$ έχει, στην αντικειμενική του συνάρτηση, ένα y_S^F για κάθε πιθανή μετάθεση F των στοιχείων του S για κάθε $S \in \mathbf{S}$. Το $\text{LP}_{\text{Primal}}$ έχει εκθετικό πλήθος μεταβλητών. Για αυτό το λόγο θα λύσουμε, το δυϊκό του $\text{LP}_{\text{Primal}}$, το LP_{dual} .

$$\begin{aligned} \max \sum_{t \in [n]} \alpha_t + \sum_{e \in [n]} \beta_e + \sum_{S \in \mathbf{S}} \gamma_S \\ \alpha_t + \beta_e - \sum_{S: e \in S} \delta_{eS} \leq 0 \quad \forall e, t \in [n] \end{aligned} \quad (32)$$

$$\gamma_S + \sum_{(e,t) \in F} \delta_{eS} \leq C_S^F \quad \forall S \in \mathbf{S}, F \in \mathbf{F}(S) \quad (33)$$

Τώρα, όμως, έχουμε να αντιμετωπίσουμε εκθετικό πλήθος περιορισμών. Μπορούμε ωστόσο, όπως είδαμε, να λύσουμε το LP_{dual} με την μέθοδο των ελλειψοειδών [17], αρκεί να έχουμε ένα separation oracle, που να ελέγχει αν μια υποψήφια λύση ικανοποιεί όλους τους περιορισμούς. Οι περιορισμοί (32) μπορούν να ελεγχθούν γρήγορα αφού είναι μόνο n^2 . Για αυτό θα ασχοληθούμε με τους περιορισμούς (33). Χρειαζόμαστε ένα αλγόριθμο που να δέχεται τα γ_S και δ_{eS} και να βρίσκει σε πολυωνυμικό χρόνο, αν υπάρχουν, $S \in \mathbf{S}$ και $F \in \mathbf{F}(S)$ που να παραβιάζουν τους περιορισμούς (33).

Θα μοντελοποιήσουμε αυτό το πρόβλημα σαν το Minimum Cost s-t Flow Problem. Σε αυτό το πρόβλημα, μας δίνεται ένας κατευθυνόμενο γράφημα $G = (V, E)$ (δίκτυο ροής) με δυο διακεκριμένους κόμβους, την πηγή $s \in V$ και τον προορισμό $t \in V$. Κάθε ακμή $(u, v) \in E$ έχει μια χωρητικότητα $c(u, v) \in \mathbb{Q}_+$ και ένα κόστος $k(u, v) \in \mathbb{Q}$. Επιπλέον, μας δίνεται μια τιμή $\varphi \in \mathbb{Q}_+$. Μια ροή f με τιμή φ πάνω στο γράφημα $G = (V, E)$ είναι μια συνάρτηση $f: E \rightarrow \mathbb{Q}_+$ για την οποία ισχύουν:

$$\begin{aligned} \text{[περιορισμός χωρητικότητας]} \quad & 0 \leq f(u, v) \leq c(u, v), \\ \text{[διαγώνια συμμετρία]} \quad & f(u, v) = -f(v, u), \\ \text{[διατήρηση ροής]} \quad & \sum_{w \in V} f(u, w) = 0 \text{ για κάθε } u \neq s, t, \\ \text{[απαιτούμενη ροή]} \quad & \sum_{w \in V} f(s, w) = \varphi \text{ και } \sum_{w \in V} f(w, t) = \varphi. \end{aligned}$$

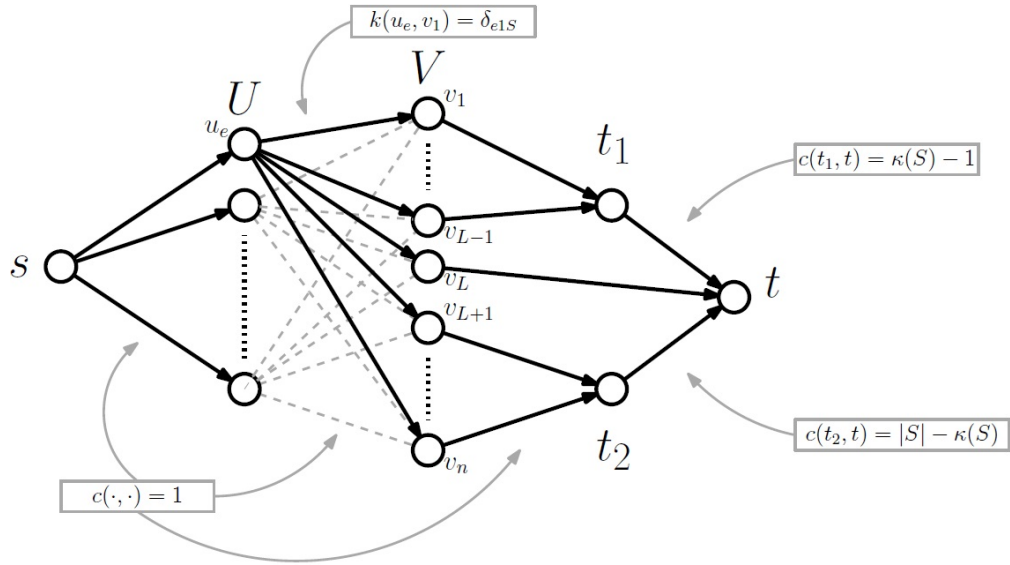
Το ζητούμενο του προβλήματος, είναι να στείλουμε μια ροή f με τιμή φ από το s στο t με όσο το δυνατόν μικρότερο κόστος $\sum_{(u,v) \in E} f(u, v) \cdot k(u, v)$.

Αν η τιμή φ και οι χωρητικότητες των ακμών $c(u, v) \in E$, $(u, v) \in E$ είναι ακέραιες, τότε μπορούμε σε πολυωνυμικό χρόνο να ελέγξουμε αν υπάρχει ροή από το s στο t με τιμή φ . Επιπλέον, αν μια τέτοια ροή υπάρχει, δηλαδή, το πρόβλημα έχει εφικτή λύση, τότε η βέλτιστη ροή s-t με τιμή φ θα είναι ακέραια και μπορούμε να την υπολογίσουμε σε πολυωνυμικό χρόνο [35, 16].

Σειρά έχει να δείξουμε πως θα κάνουμε αυτή την μοντελοποίηση. Είναι πιο βολικό να ασχοληθούμε με το ισοδύναμο πρόβλημα Maximum Cost s-t Flow Problem, όπου θέλουμε να μεγιστοποιήσουμε το κόστος $\sum_{(u,v) \in E} f(u, v) \cdot k(u, v)$ αντί να το ελαχιστοποιήσουμε.

Διαλέγουμε ένα σύνολο S και ένα ακέραιο $L \in [n]$. Θα βρούμε τον τρόπο να επαληθεύσουμε αν παραβιάζονται οι περιορισμοί (33) που αφορούν το σύνολο S και της μεταθέσεις $F \in \mathbf{F}$ με $C_S^F = L$. Αρχικά, κατασκευάζουμε ένα πλήρες διμερές γράφημα $G_L = (U, V, E)$ όπου η πλευρά U έχει ένα κόμβο u_e για κάθε στοιχείο $e \in S$ και η πλευρά έχει V ένα κόμβο v_i για κάθε χρονική μονάδα $i \in [n]$. Η κάθε ακμή $\varepsilon = (u_e, v_i) \in E$ έχει κόστος $k(\varepsilon) = \delta_{e \in S}$ και χωρητικότητα $c(\varepsilon) = 1$. Στη συνέχεια, προσθέτουμε στον G_L , την πηγή s και την συνδέουμε με όλους τους κόμβους της πλευράς U . Οι ακμές ε που συνδέουν την πηγή s με την U έχουν κόστος $k(\varepsilon) = 0$ και χωρητικότητα $c(\varepsilon) = 1$. Υπάρχουν δύο ενδιάμεσοι προορισμοί t_1 και t_2 που συνδέονται με τον τελικό προορισμό t . Οι κόμβοι v_1, v_2, \dots, v_{L-1} στην πλευρά V , συνδέονται με τον ενδιάμεσο προορισμό t_1 ενώ οι κόμβοι $v_{L+1}, v_{L+2}, \dots, v_n$ συνδέονται με τον άλλο ενδιάμεσο προορισμό t_2 . Όλες οι ακμές ε που ενώνουν την πλευρά V και τους ενδιάμεσους προορισμούς t_1, t_2 έχουν κόστος $k(\varepsilon) = 0$ και χωρητικότητα $c(\varepsilon) = 1$. Οι ακμές $\varepsilon' = (t_1, t)$ και $\varepsilon'' = (t_2, t)$ έχουν χωρητικότητα $c(\varepsilon') = K(S) - 1$ και $c(\varepsilon'') = |S| - K(S)$ αντίστοιχα, ενώ η ε' όπως και η ε'' έχουν μηδενικό κόστος. Ο κόμβος v_L είναι ο μόνος που συνδέεται απευθείας με τον τελικό προορισμό t . Η ακμή (v_L, t) έχει $c(v_L, t) = 1$ και μηδενικό κόστος. Με αυτόν τον τρόπο κατασκευάσαμε το γράφημα $G = (V, E)$. Ο στόχος είναι να

βρούμε την ροή $s-t$ με τιμή $\varphi = |S|$ που έχει το μέγιστο κόστος.



Σχήμα 3: Αναπαράσταση του γραφήματος G στον οποίο θέλουμε να βρούμε την ροή $s-t$ με το μέγιστο κόστος.

Από τον τρόπο που κατασκευάσαμε τον G , κάθε ακέραια ροή f από το s στο t , με τιμή $\varphi = |S|$ πάνω στο γράφημα G , αντιστοιχεί σε μια έγκυρη μετάθεση F των στοιχείων του συνόλου S , για την οποία ισχύει $C_S^F = L$ και αντίστροφα. Ως εκ τούτου, αν η ροή f με το μεγαλύτερο κόστος, έχει κόστος μεγαλύτερο του $L - \gamma_S$, οι περιορισμοί (33) παραβιάζονται για το $S \in \mathbf{S}$ και τη $F \in \mathbf{F}$ που αντιστοιχούν στην ροή f . Επίσης, ισχύει και το αντίθετο. Αν η ροή f με το μεγαλύτερο κόστος, έχει κόστος μικρότερο ή ίσο του $L - \gamma_S$ δεν υπάρχει μετάθεση $F \in \mathbf{F}$ με $C_S^F = L$ που να παραβιάζει τους περιορισμούς (33).

Το separation oracle κατασκευάζει ένα γράφημα G_S για κάθε $S \in \mathbf{S}$ και κάθε $L \in [n]$ και ελέγχει αν υπάρχει ροή $s-t$ με τιμή $|S|$. Έτσι, με $\mathbf{S} \cdot n$ πολυωνυμικής πολυπλοκότητας ελέγχους, μπορεί να επαληθεύσει αν μια υποψηφία λύση ικανοποιεί τους (33).

5.1.4 Σύγκριση του νέου LP με το LP B

Το νέο γραμμικό πρόγραμμα LP_{Primal} , είναι καλύτερο LP relaxation από το LP B, για την non-preemptive όπως και για την preemptive εκδοχή του προβλήματος. Πριν όμως απο-

δείξουμε κάτι τέτοιο, ας θυμηθούμε το LP B.

$$\min \sum_{1 \leq t \leq |U|} \sum_{S \in \mathbf{S}} (1 - z_{S,t})$$

$$\sum_{e \in U} x_{et} = 1 \quad \forall t \in [n] \quad (34)$$

$$\sum_{t \in [n]} x_{et} = 1 \quad \forall e \in U \quad (35)$$

$$\sum_{e \in S \setminus A} \sum_{t' < t} x_{et'} \geq (K(S) - |A|) \cdot z_{S,t} \quad \forall S \in \mathbf{S}, \forall A \subseteq S, \forall t \in [n] \quad (36)$$

$$x_{et}, z_{S,t} \in [0, 1] \quad \forall e \in U, S \in \mathbf{S}, t \in [n]$$

Προσέξτε πως αλλάξαμε την μεταβλητή $y_{S,t}$ σε $z_{S,t}$ ώστε να μην υπάρχει σύγχυση με την μεταβλητή y_S^F του LP_{Primal} . Οι περιορισμοί (34) και (35) του LP B είναι ίδιοι με τους περιορισμούς (23) και (24) του LP_{Primal} . Στο LP B η αντικειμενική συνάρτηση μετράει το ποσοστό του κάθε συνόλου S που έχει καλυφθεί ανά χρονική μονάδα $t \in [n]$, ενώ στο LP_{Primal} το ποσοστό που οι μεταθέσεις $F \in \mathbf{F}$ των στοιχείων του κάθε συνόλου $S \in \mathbf{S}$ συνεισφέρουν ώστε να σχηματιστεί η τελική λύση.

Το LP B αν και χρησιμοποιήθηκε για την non-preemptive εκδοχή, είναι έγκυρο LP relaxation και για τη preemptive εκδοχή του προβλήματος. Έχουμε λοιπόν το ακόλουθο λήμμα.

Πρόταση 5.4. *Το LP B είναι ένα έγκυρο LP relaxation για το Preemptive Generalized Min-Sum Set Cover.*

Απόδειξη. Για κάθε preemptive συνεχής διάταξη $x_e(t)$, $e \in [n]$, $t \in [0, n]$. Ορίζουμε την $x_{e,t} = \int_{\tau=t-1}^t x_e(\tau) d\tau$. Όπως είδαμε στο Θεώρημα 5.2 η $x_{e,t}$ ικανοποιεί τους περιορισμούς (34) και (35). Για κάθε σύνολο $S \in \mathbf{S}$ και $t \in [n]$. Ορίζουμε την $z_{S,t} = \min_{A \subseteq S, |A| < K(S)} \frac{\sum_{e \in S \setminus A} \sum_{t' < t} x_{e,t'}}{K(S) - |A|}$. Από τον τρόπο που ορίσαμε την $z_{S,t}$, είναι προφανές ότι ικανοποιεί τους περιορισμούς (34).

Τώρα απομένει να δείξουμε ότι για τις μεταβλητές $x_{e,t}$, $z_{S,t}$ που ορίσαμε, το κόστος της αντικειμενικής συνάρτησης του LP B είναι μικρότερο από το κόστος της συνεχής διάταξη $x_e(t)$, που είναι και η βέλτιστη λύση. Για να αποδείξουμε κάτι τέτοιο αρκεί να δείξουμε ότι για κάθε $S \in \mathbf{S}$, ισχύει $\sum_{t \in [n]} (1 - z_{S,t}) \leq C_S^P$, όπου το C_S^P είναι ο χρόνος κάλυψης του S στην συνεχή διάταξη $x_e(t)$. Παρατηρήστε ότι για κάθε χρονική μονάδα $t \geq \lfloor C_S^P \rfloor + 2$, $(1 - z_{S,t}) = 0$, γιατί τουλάχιστον $K(S)$ στοιχεία του S έχουν επιλεγεί πλήρως κατά την χρονική μονάδα $\lfloor C_S^P \rfloor + 1$. Τώρα, κατά την χρονική μονάδα $\lfloor C_S^P \rfloor + 1$. Έστω $S' \subseteq S$ είναι το υποσύνολο των $K(S)$ στοιχείων που καλύπτονται πλήρως από την χρονική στιγμή $\lfloor C_S^P \rfloor$ μέχρι και τη χρονική στιγμή C_S^P . Επομένως, για κάθε στοιχείο $e \in S'$ έχουμε $\int_{\tau=0}^{\lfloor C_S^P \rfloor} x_e(\tau) d\tau \geq 1 - (C_S^P - \lfloor C_S^P \rfloor)$. Παίρνουμε $A = S'$, στην χειρότερη περίπτωση που $|A| = K(S) - 1$ έχουμε $y_{S, \lfloor C_S^P \rfloor} \geq 1 - (C_S^P - \lfloor C_S^P \rfloor)$. Τέλος, για κάθε χρονική μονάδα $t \leq \lfloor C_S^P \rfloor$ το S συνεισφέρει το

πολύ 1 στην αντικειμενική συνάρτηση του LP B. Αθροίζοντας για όλες χρονικές μονάδες $t \in [n]$ περνούμε $\sum_{t \in [n]} \leq \lfloor C_S^P \rfloor + (1 - y_{S, \lfloor C_S^P \rfloor}) \leq C_S^P$. \square

Σειρά έχει να δείξουμε ότι το LP_{Primal} είναι ισχυρότερο κάτω φράγμα από το LP B για το preemptive πρόβλημα. Αρχικά, θα κατασκευάσουμε στιγμιότυπα για τα οποία η αντικειμενική συνάρτηση του LP_{Primal} έχει κόστος αυστηρά μεγαλύτερο από αυτήν του LP B.

Πρόταση 5.5. *Για κάθε $\varepsilon > 0$, υπάρχει στιγμιότυπο για το οποίο το κόστος της αντικειμενικής συνάρτησης του LP_{Primal} είναι μεγαλύτερο του κόστους της αντικειμενικής του LP B κατά ένα παράγοντα μεγαλύτερο του $2 - \varepsilon$.*

Απόδειξη. Για ένα στιγμιότυπο, το οποίο έχει ένα σύνολο $S = [n]$ και $K(S) = n$, η αντικειμενική του LP_{Primal} θα έχει κόστος σίγουρα μεγαλύτερο ή ίσο του n . Αυτό θα συμβεί εξαιτίας του περιορισμού (25), αφού όλες οι μεταθέσεις $F \in \mathbf{F}(S)$ του S έχουν $C_S^F \geq n$. Από την άλλη, για το LP B, αν υποθέσουμε ότι όλα τα στοιχεία έχουν κατανεμηθεί με ίσο τρόπο μέσα σε κάθε χρονική μονάδα, δηλαδή, έχουμε $x_{e,t} = \frac{1}{n}$ για κάθε $e, t \in [n]$. Αφού το $z_{S,t} = \frac{t-1}{n}$ ικανοποιεί τους περιορισμούς, το LP B θα έχει αντικειμενική με κόστος $\sum_{t=1}^n (1 - \frac{t-1}{n}) = \frac{n+1}{2}$ που είναι δυο φορές μικρότερο από το κόστος της αντικειμενικής του LP_{Primal} για το ίδιο στιγμιότυπο. \square

Ολοκληρώνουμε την απόδειξη αποδεικνύοντας ότι για κάθε στιγμιότυπο το LP_{Primal} είναι ισχυρότερο του LP B. Αυτό το εκφράζουμε με το παρακάτω λήμμα:

Λήμμα 5.3. *Για κάθε στιγμιότυπο, το κόστος της αντικειμενικής συνάρτησης του LP_{Primal} είναι μεγαλύτερο ή ίσο του κόστους της αντικειμενικής του LP B.*

Απόδειξη. Έστω ένα στιγμιότυπο και x^*, y^* μια βέλτιστη λύση που δίνει το LP_{Primal} . Ορίζουμε το z ως εξής:

$$y_{S,t} = \sum_{F \in \mathbf{F}(S) \text{ και } C_S^F < t} y_S^{*F}.$$

Θέλουμε να δείξουμε ότι τα x^*, z ικανοποιούν τους περιορισμούς του LP B. Θα επικεντρωθούμε στους περιορισμούς (36) αφού οι υπόλοιποι είναι προφανές ότι ικανοποιούνται.

Έστω ένα σύνολο S και $A \subseteq S$. Τότε έχουμε:

$$\begin{aligned}
 \sum_{e \in S \setminus A} \sum_{t' < t} x_{e,t'}^* &= \sum_{e \in S \setminus A} \sum_{t' < t} \sum_{F \in \mathbf{F}(S), (e,t') \in F} y_S^{*F} \quad [\text{από τους περιορισμούς (26)}] \\
 &= \sum_{F \in \mathbf{F}(F)} y_S^{*F} \sum_{e \in S \setminus A} \sum_{t' < t} 1[(e,t') \in F] \\
 &\geq \sum_{F \in \mathbf{F}(F) \text{ και } C_S^F < t} y_S^{*F} \sum_{e \in S \setminus A} \sum_{t' < t} 1[(e,t') \in F] \\
 &\geq \sum_{F \in \mathbf{F}(F) \text{ και } C_S^F < t} y_S^{*F} (K(S) - |A|) \\
 &= (K(S) - |A|) \cdot z_{S,t}.
 \end{aligned}$$

Η τελευταία ανισότητα ισχύει γιατί για κάθε $t \in [n]$, $S \in \mathbf{S}$ και $F \in \mathbf{F}(S)$ με $C_S^F < t$ τουλάχιστον $K(S)$ στοιχεία του S έχουν επιλεχθεί πριν την χρονική μονάδα t κατά την μετάθεση F . Το $\text{LP}_{\text{Primal}}(x^*, y^*)$ εκφράζει το κόστος της αντικειμενικής του $\text{LP}_{\text{Primal}}$ για την βέλτιστη λύση x^*, y^* . Ανάλογα, το $\text{LP B}(x^*, z)$ εκφράζει το κόστος της αντικειμενικής του LP B για την λύση x^*, z . Θα δείξουμε ότι:

$$\text{LP}_{\text{Primal}}(x^*, y^*) = \text{LP B}(x^*, z).$$

Αφού τα x^*, y^* είναι μια βέλτιστη λύση του $\text{LP}_{\text{Primal}}$ και τα x^*, z είναι μια εφικτή λύση για το LP B , θα έχουμε αποδείξει το Λήμμα 5.3. Έστω ένα σύνολο $S \in \mathbf{S}$ και μια μετάθεση $F \in \mathbf{F}(S)$. Δεδομένου ότι, το C_S^F προσθέτει ένα στο κόστος της αντικειμενικής για κάθε χρονική μονάδα $t \leq C_S^F$ έχουμε:

$$\begin{aligned}
 \text{LP}_{\text{Primal}}(x^*, y^*) &= \sum_{S \in \mathbf{S}} \sum_{F \in \mathbf{F}(S)} C_S^F y_S^{*F} \\
 &= \sum_{S \in \mathbf{S}} \sum_{1 \leq t \leq n} \sum_{F \in \mathbf{F}(S): t \leq C_S^F} y_S^{*F} \\
 &= \sum_{1 \leq t \leq n} \sum_{S \in \mathbf{S}} \left(1 - \sum_{F \in \mathbf{F}(S): t > C_S^F} y_S^{*F} \right) \quad [\text{από τους περιορισμούς (25)}] \\
 &= \sum_{1 \leq t \leq n} \sum_{S \in \mathbf{S}} (1 - z_{S,t}) \\
 &= \text{LP B}(x^*, z).
 \end{aligned}$$

□

5.1.5 Ο αλγόριθμος

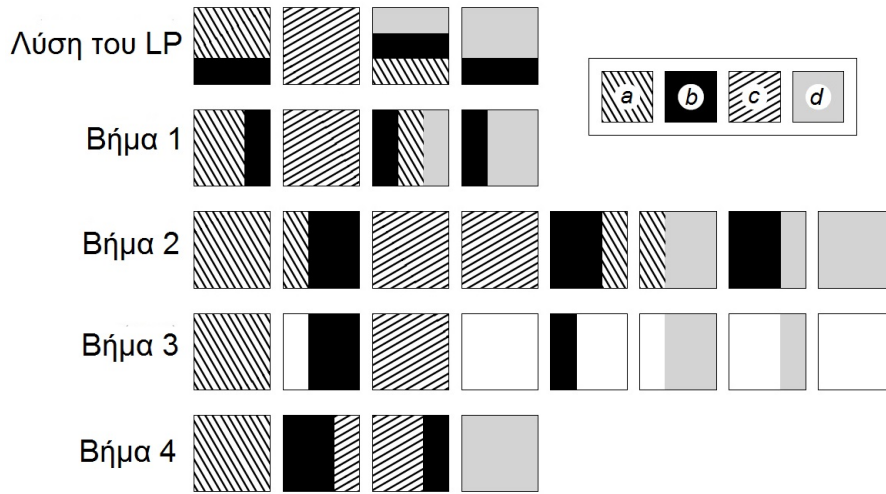
Έστω $x_{e,t}$ και y_S^F μια βασική βέλτιστη λύση του γραμμικού προβλήματος $\text{LP}_{\text{Primal}}$. Γνωρίζουμε ότι θα έχει το πολύ $2n + m + n^2m$ μη μηδενικές μεταβλητές, όσες και ο αριθμός των περιορισμών (23) έως (26). Ο αλγόριθμος από αυτή την βασική βέλτιστη λύση, κατα-

σκευάζει μια νέα διάταξη με την βοήθεια της παραμέτρου $\lambda \in (0, 1]$, η λ διαλέγεται τυχαία από το $(0, 1]$ βάση της συνάρτησης κατανομής $f(v) = 2v$. Αυτό το πετυχαίνει σταδιακά μέσα σε τέσσερα βήματα

Στο πρώτο βήμα κατασκευάζει μια τυχαία συνεχή διάταξη $x_e(t)$, $e \in [n]$, $t \in [0, n]$ από τα $x_{e,t}$, $e, t \in [n]$, τέτοια ώστε για κάθε $e, t \in [n]$, $\int_{\tau=t-1}^t x_e(\tau) d\tau = x_{e,t}$. Αυτό, για παράδειγμα, μπορεί να γίνει αναθέτοντας με τυχαία σειρά, μέσα σε κάθε χρονοθυρίδα t , ένα συμπαγή ποσοστό $x_{e,t}$ του κάθε στοιχείου e . Αυτή την συνεχή διάταξη $x_e(t)$ θα την συμβολίζουμε με σ . Στο δεύτερο βήμα, από την σ κατασκευάζει την νέα διάταξη $\sigma(\lambda)$ φουσκώνοντας όλα τα συμπαγή κομμάτια κατά ένα παράγοντα $\frac{1}{\lambda}$. Με άλλα λόγια, αυτό που κάνει ο αλγόριθμος είναι να αντιστοιχεί κάθε χρονική στιγμή τ στην χρονική στιγμή $\frac{\tau}{\lambda}$. Για κάθε στοιχείο $e \in [n]$ η $\tau_e \in [1, \frac{n}{\lambda}]$ είναι η πρώτη χρονική στιγμή κατά την οποία έχει επιλεγθεί ακριβώς μια μονάδα του στοιχείου (από το συνολικό $\frac{1}{\lambda}$). Στο τρίτο βήμα, ο αλγόριθμος για κάθε στοιχείο e μετά την τ_e θα απορρίψει όλο το υπόλοιπο ποσοστό του e , έτσι ώστε όλα τα στοιχεία στη $\sigma(\lambda)$ να έχουν μοναδιαίο μέγεθος. Αυτό έχει σαν συνέπεια να δημιουργηθούν κενά στην διάταξη και σε κάποια χρονικά διαστήματα να μην επιλέγεται κανένα στοιχείο. Στο τελικό βήμα, ο αλγόριθμος αφαιρεί αυτά τα κενά σπρώχνοντας όλα τα κομμάτια των στοιχείων στα αριστερά ώστε η τελική διάταξη $\sigma(\lambda)$ να έχει μήκος n . Η τελική συνεχής διάταξη $\sigma(\lambda)$ θα συμβολίζετε και $x_e^{(\lambda)}(t)$, $e \in [n]$, $t \in [0, n]$. Η τεχνική του αλγορίθμου που μόλις περιγράψαμε, έχει χρησιμοποιηθεί και άλλες φορές για την υλοποίηση προσεγγιστικών αλγορίθμων για προβλήματα χρονοδρομολόγησης (scheduling problems) [34, 37].

Ο Αλγόριθμος παίρνει τα $x_{e,t}$ και y_S^F που θα δώσει το LP_{Primal} και κατασκευάζει την τελική διάταξη $\sigma(\lambda)$ κάνοντας τα ακόλουθα βήματα:

- 1 **κατασκευάσε** μια τυχαία συνεχής διάταξη σ από τα $x_{e,t}$
- 2 **διαλέξε** τυχαία ένα λ από το $(0, 1]$ βάση της συνάρτησης κατανομής $f(v) = 2v$
κατασκευάσε την $\sigma(\lambda)$ φουσκώνοντας τα κομμάτια της σ κατά $\frac{1}{\lambda}$
- 3 **κόψε** το ποσοστό του κάθε στοιχείου e μόλις υπερβεί την μονάδα
- 4 **σπρώξε** την διάταξη ώστε να κλείσουν τα κενά και να έχει μήκος n
επέστρεψε την διάταξη $\sigma(\lambda)$



Σχήμα 4: Τα βήματα του αλγορίθμου.

Για να καταλάβουμε πως ακριβώς δουλεύει ο αλγόριθμος, ας υποθέσουμε, για παράδειγμα, ότι έχουμε ένα στιγμιότυπο με τέσσερα στοιχεία $\{a, b, c, d\}$.

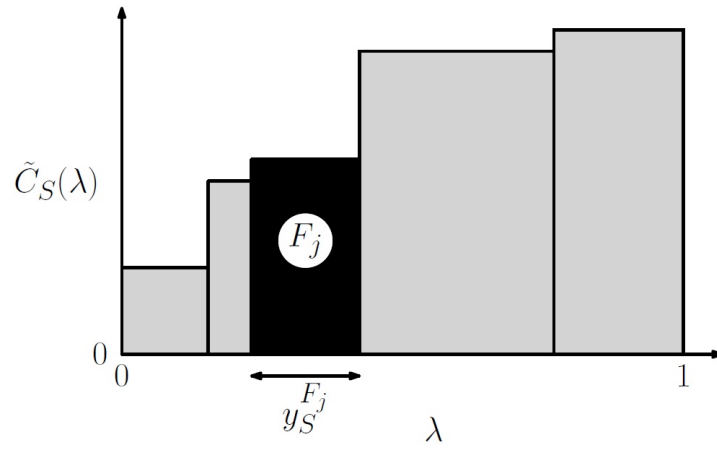
Η λύση που δίνει το LP_{Primal} είναι:

$$x_{a,1} = \frac{2}{3}, x_{b,1} = \frac{1}{3}, x_{c,2} = 1, x_{d,3} = \frac{1}{3}, x_{b,3} = \frac{1}{3}, x_{a,3} = \frac{1}{3}, x_{d,4} = \frac{2}{3}, x_{b,4} = \frac{1}{3},$$

Αρχικά, κατασκευάζει μια συνεχή διάταξη κάνοντας τα κομμάτια των στοιχείων συμπαγή και βάζοντας τα τυχαία μέσα σε κάθε χρονική μονάδα. Για παράδειγμα, στην τρίτη χρονική μονάδα στοιχεία τοποθετήθηκαν τυχαία στην σειρά b, a, c και το καθένα πιάνει το $\frac{1}{3}$ της χρονοθυρίδας. Στη συνέχεια, επιμηκύνει την διάταξη κατά ένα παράγοντα δύο ($\lambda = \frac{1}{2}$). Στο επόμενο βήμα, κόβει και απορρίπτει κάθε στοιχείο μόλις αυτό συμπληρώσει την μονάδα. Στο τέλος, σπρώχνει την διάταξη στα αριστερά ώστε να κλείσουν τα κενά. Το παραπάνω σχήμα δίνει μια γραφική αναπαράσταση του πως δουλεύει ο αλγόριθμος για το στιγμιότυπο του παραδείγματος.

5.1.6 Ανάλυση του αλγορίθμου

Ξεκινάμε την ανάλυση με κάποιους απαραίτητους ορισμούς. Για ένα σύνολο $S \in \mathbf{S}$, ορίζουμε το C_S^{LP} να εκφράζει τον χρόνο κάλυψης του συνόλου S που παίρνουμε από την βασική βέλτιστη λύση του LP_{Primal} , δηλαδή, $C_S^{LP} = \sum_{F \in \mathbf{F}(S)} C_S^F y_S^F$ και το $C_S(\lambda)$ να εκφράζει τον χρόνο κάλυψης του S που δίνει η τελική διάταξη $\sigma(\lambda)$. Οργανώνουμε όλες τις μεταθέσεις $F \in \mathbf{F}(S)$ με $y_S^F > 0$ σε αύξουσα σειρά βάση του C_S^F . Έστω F_1, F_2, \dots, F_k μια τέτοια σειρά. Ορίζουμε το $\tilde{C}_S(\lambda) = C_S^{F_j}$ όπου $\sum_{i=1}^{j-1} y_S^{F_i} < \lambda$ και $\sum_{i=1}^j y_S^{F_i} \geq \lambda$, (δείτε στο παρακάτω σχήμα για γραφική αναπαράσταση). Έστω μια συνάρτηση $1[\varphi]$ όπου $1[\varphi] = 1$ αν και μόνο αν το φ αληθεύει και 0 σε αντίθετη περίπτωση.



Σχήμα 5: Γραφική αναπαράσταση του $\tilde{C}_S(\lambda)$. Το j -οστό ορθογώνιο αντιστοιχεί στη μετάθεση F_j , έχει μήκος $y_S^{F_j}$ και ύψος $C_S^{F_j}$.

Λήμμα 5.4. Για κάθε $S \in \mathbf{S}$ και $0 < \lambda \leq 1$, έχουμε $C_S(\lambda) \leq \frac{1}{\lambda} \cdot \tilde{C}_S(\lambda)$.

Απόδειξη. Για να απλοποιήσουμε την απόδειξη, υποθέτουμε ότι υπάρχει j τέτοιο ώστε $\sum_{1 \leq l \leq j} y_S^{F_l} = \lambda$. Σε αντίθετη περίπτωση, για το μικρότερο j για το οποίο ισχύει $\sum_{1 \leq l \leq j} y_S^{F_l} > \lambda$ ορίζουμε δύο αντίγραφα F_j και F_j' της μετάθεσης F_j , μόνο που $y_S^{F_j} = \lambda - \sum_{1 \leq l \leq j-1} y_S^{F_l}$ και $y_S^{F_j'} = \lambda - \sum_{1 \leq l \leq j} y_S^{F_l} - \lambda$. Οι μεταθέσεις F_j και F_j' είναι ίδιες με την F_j και περνούμε $\sum_{1 \leq l \leq j-1} (y_S^{F_l}) + y_S^{F_j} = \lambda$.

Η παρακάτω ανισότητα δείχνει ότι ο χρόνος κάλυψης $C_S(\lambda)$ του σύνολου S στην διάταξη $\sigma(\lambda)$ είναι μικρότερος η ίσος του $\tilde{C}_S(\lambda)/\lambda$.

$$\int_{\tau=0}^{\tilde{C}_S(\lambda)/\lambda} \sum_{e \in S} x_e^{(\lambda)}(\tau) d\tau \geq K(S). \quad (37)$$

Αφού για κάθε $e \in S$ έχουμε:

$$\int_{\tau=0}^{\tilde{C}_S(\lambda)/\lambda} x_e^{(\lambda)}(\tau) d\tau \geq \min \left\{ 1, \frac{1}{\lambda} \cdot \int_{\tau=0}^{\tilde{C}_S(\lambda)} x_e(\tau) d\tau \right\} \geq \min \left\{ 1, \frac{1}{\lambda} \cdot \sum_{t \leq \lfloor \tilde{C}_S(\lambda) \rfloor} x_{e,t} \right\}$$

και επειδή το $\tilde{C}_S(\lambda)$ είναι εξ ορισμού ακέραιος για κάθε $\lambda \in (0, 1]$. Για να αποδείξουμε ότι η (37) ισχύει αρκεί, να δείξουμε ότι ισχύει η ανισότητα:

$$\sum_{e \in S} \min \left\{ \lambda, \sum_{t \leq \tilde{C}_S(\lambda)} x_{e,t} \right\} \geq \lambda \cdot K(S) \quad (38)$$

Αναλύοντας την (38) παίρνουμε:

$$\begin{aligned}
 \sum_{e \in S} \min \left\{ \lambda, \sum_{t \leq \tilde{C}_S(\lambda)} x_{e,t} \right\} &\geq \sum_{e \in S} \min \left\{ \lambda, \sum_{t \leq \tilde{C}(\lambda)} \sum_{l=1}^j y_S^{F_l} \cdot 1[(e, t) \in F_l] \right\} \\
 &= \sum_{e \in S} \min \left\{ \lambda, \sum_{l=1}^j y_S^{F_l} \cdot 1[(e, t) \in F_l \text{ για κάποια } t \leq \tilde{C}(\lambda)] \right\} \\
 &= \sum_{e \in S} \sum_{l=1}^j y_S^{F_l} \cdot 1[(e, t) \in F_l \text{ για κάποια } t \leq \tilde{C}(\lambda)] \\
 &= \sum_{l=1}^j y_S^{F_l} \sum_{e \in S} 1[(e, t) \in F_l \text{ για κάποια } t \leq \tilde{C}(\lambda)] \\
 &\geq \sum_{l=1}^j y_S^{F_l} \cdot K(S) = \lambda \cdot K(S).
 \end{aligned}$$

Η πρώτη ανισότητα έπεται από τους περιορισμούς (26). Η δεύτερη ισότητα ισχύει επειδή $\sum_{j=1}^j y_S^{F_l} = \lambda$. Η τελευταία ανισότητα ισχύει επειδή για κάθε F_l με $l \leq j$ έχουμε $C_S^{F_l} \leq \tilde{C}_S(\lambda)$. \square

Το ακόλουθο λήμμα αποδεικνύεται εύκολα από τον ορισμό του $\tilde{C}_S(\lambda)$.

Λήμμα 5.5. Για κάθε $\tilde{C}_S(\lambda)$, έχουμε $S \in \mathbf{S}$, $\int_{\lambda=0}^1 \tilde{C}_S(\lambda) d\lambda = C_S^{LP}$.

Απόδειξη. (Απόδειξη του Θεωρήματος 5.1) Ολοκληρώνουμε υπολογίζοντας τον αναμενόμενο χρόνο κάλυψης της διάταξης $\sigma(\lambda)$ που επιστρέφει ο αλγόριθμος.

$$\begin{aligned}
 E[C_S(\lambda)] &= \int_{\lambda=0}^1 C_S(\lambda) \cdot 2\lambda d\lambda \quad [\text{εξ ορισμού}] \\
 &\leq \int_{\lambda=0}^1 \frac{1}{\lambda} \cdot \tilde{C}_S(\lambda) \cdot 2\lambda d\lambda \quad [\text{από το Λήμμα 5.4}] \\
 &= 2 \int_{\lambda=0}^1 \tilde{C}_S(\lambda) d\lambda = 2C_S^{LP} \quad [\text{από το Λήμμα 5.5}]
 \end{aligned}$$

\square

Μπορούμε να μετατρέψουμε τον αλγόριθμο από πιθανοτικό σε διαδικαστικό. Η συνάρτηση $\tilde{C}_S(\lambda)$ είναι piecewise constant function δηλαδή είναι τοπικά σταθερή και σχηματίζει σκαλοπάτια. Επειδή, η λύση που περνούμε είναι βασική και βέλτιστη, θα έχουμε πολυωνυμικό αριθμό μη μηδενικών μεταβλητών Y_S^F για κάθε S . Άρα, και ο αριθμός των σκαλοπατιών θα είναι πολυωνυμικός, αφού κάθε σκαλοπάτι εκφράζει μια F με $Y_S^F \geq 0$. Αυτό σημαίνει ότι, θα υπάρχουν πολυωνυμικό πλήθος πιθανών λ που χρειάζεται να ελέγξουμε, ένα από τα οποία θα μας δίνει την επιθυμητή απόδοση.

Από το preemptive στο non-preemptive

Σε αυτή τη ενότητα θα δούμε πως πάμε από μια preemptive λύση σε μια non-preemptive. Προσέξτε πως, αν για ένα στιγμιότυπο μπορούμε να μετατρέψουμε μια preemptive λύση του σε μια non-preemptive πληρώνοντας ένα λόγο η , θα είχαμε ένα 2η -προσεγγιστικό αλγόριθμο για το Generalized Min-Sum Set Cover. Ο αλγόριθμος που μετατρέπει την preemptive λύση του σε non-preemptive έχει πολλά κοινά σημεία με τον αλγόριθμο με τα α -points των Martin Skutella και David P. Williamson [41] που είδαμε νωρίτερα. Τα πλεονεκτήματα είναι ότι ξεκινάει από την preemptive λύση και ο τρόπος με τον οποίο φουσκώνει την λύση είναι πιο πολύπλοκος και ευρηματικός.

Θεώρημα 5.3 ([20]). *Για κάθε preemptive διάταξη που έχει κόστος C , υπάρχει μια non-preemptive διάταξη με αναμενόμενο κόστος το πολύ $6.2C$. Επιπλέον, τη συγκεκριμένη non-preemptive διάταξη μπορούμε να την βρούμε σε πολυωνυμικό χρόνο.*

Συνδυάζοντας το Θεώρημα 5.1 και το Θεώρημα 5.3 καταλήγουμε

Θεώρημα 5.4 ([20]). *Υπάρχει 12.4-προσεγγιστικός αλγόριθμος για το Generalized Min-Sum Set Cover.*

Το Θεώρημα 5.3 μας δίνει ένα πάνω φράγμα για το η , επίσης, ξέρουμε ότι δεν μπορεί να πάρει τιμή μικρότερη του 2. Αυτό θα σήμαινε ότι θα είχαμε αλγόριθμο με εγγύηση απόδοσης μικρότερη του 4 για το Generalized Min-Sum Set Cover, πράγμα αδύνατο, εκτός αν $P = NP$ (επειδή είναι γενίκευση του Min-Sum Set Cover).

Έστω P μια preemptive λύση στην οποία αντιστοιχεί η συνάρτηση $x_e(t)$, $e \in [n]$, $t \in [0, n]$. Η $x_e(t) \in \{0, 1\}$ μας δείχνει αν το στοιχείο e έχει ανατεθεί στην χρονική στιγμή t . Θα βοηθούσε, αν επεκτείναμε το πεδίο ορισμού της $x_e(t)$ σε $[0, \infty)$, θέτοντας $x_e(t) = 0$ για κάθε $t > n$. Ορίζουμε το C_S^P ως το χρόνο κάλυψης του συνόλου S που παίρνουμε από τη preemptive λύση P με συνάρτηση $x_e(t)$. Θέλουμε ο αλγόριθμος να φτιάχνει μια non-preemptive διάταξη R στην οποία το S καλύπτεται τη χρονική μονάδα C_S^R , έτσι ώστε $E[C_S^R] = O(1)C_S^P$. Ο αλγόριθμος μετατροπής της preemptive λύσης, όπως και ο αλγόριθμος των Martin Skutella και David P. Williamson, χωρίζεται σε δύο φάσεις. Στην πρώτη φάση φουσκώνει την αρχική διάταξη, ενώ στην δεύτερη επιλεγεί τυχαία τα α -points. και τα χρησιμοποιεί για να εξάγει την τελική διάταξη.

Ας υποθέσουμε ότι, ο αλγόριθμος παίρνει στην είσοδο τη διάταξη $\sigma = \sigma_0$ με συνάρτηση x . Θα ορίσουμε και θα περιγράψουμε τον τρόπο κατασκευής κάποιων διατάξεων που δημιουργεί ο αλγόριθμος κατά το πρώτο μέρος. Ξεκινάμε, ορίζοντας τη νέα συνάρτηση (και μη ακέραια διάταξη) \tilde{x} που παίρνουμε από την x με τον ακόλουθο τρόπο. Για κάθε ακέραιο $i \geq 1$ “πιέζουμε” την σ ώστε να απλωθεί κατά ένα παράγοντα r_i όπου $r \geq 1$ μια σταθερά που θα ορίσουμε αργότερα. Τη νέα διάταξη την συμβολίζουμε με σ_i . Πιο τυπικά, η σ_i προέρχεται από την $x_e^{(i)}(t)$ για την οποία ισχύει ότι $x_e^{(i)}(r^i t) = \frac{1}{r^i} x_e(t)$ για όλα τα $e \in [n]$. Η $x_e^{(i)}(t)$ μπορεί πλέον να πάρει και τιμές μεταξύ του 0 και του 1. Προσέξτε, επίσης, πως

η σ_i είναι “πιεσμένη” και ο συνολικός όγκος κάθε συμπαγούς κομματιού είναι ίδιος με το αντίστοιχο κομμάτι στη σ , δηλαδή η σ_i αναθέτει το ίδιο ποσοστό του στοιχείου e στο χρονικό κομμάτι $[r^i t, r^i(t + dt)]$ που η σ αναθέτει στο χρονικό κομμάτι $[t, t + dt]$. Έστω $Q \geq 0$ και $\rho \geq 0$ παράμετροι που θα επιλέξουμε αργότερα, ορίζουμε την $\tilde{x}_e(t)$ ως εξής:

$$\tilde{x}_e(t) = Q \left(x_e(t) + \rho \sum_{i=1}^{\infty} x_e^{(i)}(t) \right).$$

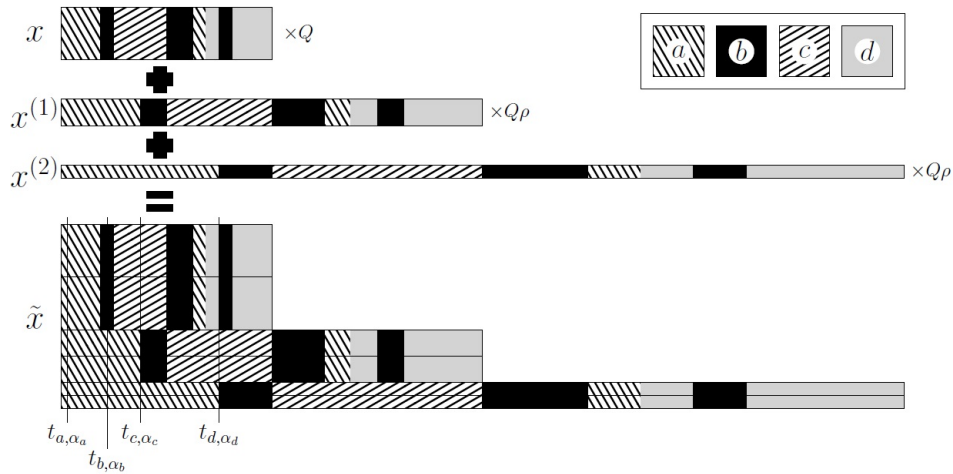
Η $\tilde{x}_e(t)$ ενδέχεται να αντιστοιχεί σε μια μη εφικτή διάταξη γιατί μπορεί να αναθέσει ένα στοιχείο με ποσοστό μεγαλύτερο της μονάδας ή ακόμα και να επιλέξει παραπάνω από ένα στοιχείο την ίδια χρονική στιγμή. Τη (μη εφικτή) διάταξη που αντιστοιχεί στη $\tilde{x}_e(t)$ θα την συμβολίζουμε με $\tilde{\sigma}$. Για την $\tilde{x}_e(t)$ έχουμε το ακόλουθο λήμμα.

Λήμμα 5.6. Για κάθε $t \in (0, \infty]$ ισχύει ότι:

$$\int_{\tau=0}^t \sum_{e \in [n]} \tilde{x}_e(\tau) d\tau \leq Q \left(1 + \frac{\rho}{r-1} \right) t.$$

Απόδειξη. Η ανισότητα έπεται από το ορισμό του \tilde{x} και το γεγονός ότι κάθε σ_i , $i \geq 0$ έχει αναθέσει τα στοιχεία με ποσοστό το πολύ $\frac{t}{r^i}$ μέχρι την χρονική στιγμή t □

Στο δεύτερο μέρος ο αλγόριθμος για κάθε $e \in [n]$, επιλέγει τυχαία τα $\alpha_e \in [0, 1]$. Έστω t_{e,α_e} η πρώτη χρονική στιγμή t κατά την οποία $\int_{\tau=0}^t \tilde{x}_e(\tau) d\tau \geq \alpha_e$. Η σ^l είναι η μη εφικτή ακέραια διάταξη όπου κάθε στοιχείο e διαλέγεται μόνο την χρονική στιγμή t_{e,α_e} . Για την σ^l , όταν λέμε ότι το στοιχείο e έχει ανατεθεί την χρονική στιγμή t_{e,α_e} , αγνοούμε το γεγονός ότι κάθε στοιχείο e χρειάζεται μια χρονική μονάδα για να καλυφθεί πλήρως. Αυτό θα ληφθεί υπόψη στη τελική διάταξη. Αντίθετα, θεωρούμε το στοιχείο e πως έχει ανατεθεί μόνο τη χρονική στιγμή t_{e,α_e} . Με το ίδιο σκεπτικό, ορίζουμε το χρόνο κάλυψης C_S^l του S στην σ^l ως τη πρώτη χρονική μονάδα t τέτοια ώστε $|\{e \in S | t_{e,\alpha_e} \leq t\}| \geq K(S)$. Η τελική διάταξη είναι η σ^R και ο αλγόριθμος την κατασκευάζει επιλέγοντας τα στοιχεία σε αύξουσα σειρά βάση των αντιστοίχων t_{e,α_e} . Σε περίπτωση που ένα ή περισσότερα t_{e,α_e} είναι ίσα διαλέγει στην τύχη.



Σχήμα 6: Γραφική απεικόνιση της κατασκευής της διάταξης \tilde{x} από την x , με παραμέτρους $r = 2$, $\rho = 1$ και $Q = 2$ και α -τιμές $\alpha_a = \frac{1}{2}$, $\alpha_b = \frac{1}{4}$, $\alpha_c = 1$ και $\alpha_d = \frac{1}{2}$.

Θα δείξουμε τον τρόπο κατασκευής της \tilde{x} (πρώτη φάση), όπως και τον υπολογισμό των t_{e,α_e} και την κατασκευή της σ^R (δεύτερη φάση), με ένα παράδειγμα. Δείτε το παραπάνω σχήμα για γραφική απεικόνιση. Έχουμε ένα στιγμιότυπο με τέσσερα στοιχεία $\{a, b, c, d\}$ και την preemptive διάταξη

$$(x_e(t))_{e \in [n]} = \frac{3}{4}a, \frac{1}{4}b, 1c, \frac{1}{2}b, \frac{1}{4}a, \frac{1}{4}d, \frac{1}{4}b, \frac{3}{4}d$$

Αν πάρουμε $Q = 2$, $\rho = 1$ και $r = 2$. Φτιάχνουμε τις $x^{(i)}$ πατώντας και απλώνοντας την x κάθε φορά, κατά ένα παράγοντα r^i . Στη συνέχεια, προσθέτουμε Q αντίγραφα του $x^{(0)} = x$ και $Q\rho$ αντίγραφα για κάθε $x^{(i)}$ με $i \geq 1$ το ένα πάνω στο άλλο σχηματίζοντας την “ψήλη” διάταξη $\tilde{\sigma}$ με συνάρτηση \tilde{x} . Αν διαλέξουμε για α -points τα

$$\alpha_a = \frac{1}{2}, \alpha_b = \frac{1}{4}, \alpha_c = 1, \alpha_d = \frac{1}{2}.$$

Παίρνουμε

$$t_{a,\alpha_a} = \frac{1}{8}, t_{b,\alpha_b} = \frac{7}{8}, t_{c,\alpha_c} = \frac{3}{2}, t_{d,\alpha_d} = 3.$$

Έτσι, η τελική λύση θα είναι:

$$\sigma^R = \begin{pmatrix} a, & b, & c, & d, \\ 1, & 2, & 3, & 4, \end{pmatrix}$$

Τώρα, μπορούμε να δούμε συνοπτικά τα βήματα του αλγορίθμου μετατροπής. Δέχεται μια preemptive λύση P με διάταξη $\sigma = \sigma_0$ στην οποία αντιστοιχεί η συνάρτηση $x_e(t)$ και τις σταθερές Q , ρ και r και κάνει τα εξής βήματα:

- 1 **υπολόγισε** την $\tilde{\sigma}$ με $\tilde{x}_e(t) = Q \left(x_e(t) + \rho \sum_{i=1}^{\infty} x_e^{(i)}(t) \right)$
- 2 **διάλεξε** τυχαία και ανεξάρτητα $\alpha_e \in [0, 1] \quad \forall e \in [n]$
- 3 **υπολόγισε** τα $t_{e,\alpha_e} \quad \forall e \in U$
- 4 **ταξινόμησε** τα στοιχεία e σε μη φθίνουσα σειρά βάση των t_{e,α_e}
- 5 **επέστρεψε** την non-preemptive διάταξη σ^R με τα ταξινομημένα στοιχεία e

Ξεκινάμε την ανάλυση του αλγορίθμου μετατροπής, η οποία έχει ακριβώς την ίδια φιλοσοφία με την ανάλυση του δεύτερου αλγορίθμου. Έστω $p_{S,i}$ η πιθανότητα ότι στη διάταξη σ^i το σύνολο S δεν έχει καλυφθεί μέχρι και την χρονική στιγμή $r^i \cdot C_S^P$, δηλαδή

$$p_{S,i} = Pr[|\{e \in S | t_{e,\alpha_e} \leq r^i \cdot C_S^P\}| < K(S)].$$

Στο ακόλουθο λήμμα, φράσσουμε την $p_i = \max_{S \in \mathcal{S}} \{p_{S,i}\}$.

Λήμμα 5.7. Έστω $Q \geq 1$, $r > 1$, και $\rho \leq 1$ τότε $\max_{S \in \mathcal{S}} \{p_{S,i}\} = p_i \leq \max\{K_{1i}, K_{2i}, K_{3i}\}$ όπου:

$$\begin{aligned} K_{1i} &= \exp(-Q(1 + \rho i)), \\ K_{2i} &= \exp(-2Q(1 + \rho i)) + (2Q(1 + \rho i)) \exp(-2Q(1 + \rho i) + 1), \\ K_{3i} &= \exp\left(-1.5Q \left(\left(1 - \frac{1}{(1 + \rho i)Q}\right)^2 (1 + \rho i) \right)\right). \end{aligned}$$

Απόδειξη. Για ένα σύνολο S και ένα συγκεκριμένο $i \geq 0$, ορίζουμε το σύνολο:

$$A = \left\{ e \in S : \int_{\tau=0}^{r^i \cdot C_S^P} \tilde{x}_e(\tau) d\tau \geq 1 \right\}.$$

Παρατηρήστε ότι κάθε στοιχείο $e \in A$ δεν έχει επιλεγθεί νωρίτερα από την χρονική στιγμή $r^i \cdot C_S^P$ στη διάταξη σ^i . Από τον ορισμό του C_S^P έχουμε ότι $\sum_{e \in S} \int_{\tau=0}^{C_S^P} x_e(\tau) d\tau \geq K(S)$. Ως εκ τούτου, συμπεραίνουμε ότι για κάθε $A \subseteq S$:

$$\sum_{e \in S \setminus A} \int_{\tau=0}^{C_S^P} x_e(\tau) d\tau \geq K(S) - |A|.$$

Από τον ορισμό του \tilde{x} και την παρατήρηση ότι για κάθε $i \geq 0$ κάθε διάταξη σ^i επιλέγει τα στοιχεία $e \in S \setminus A$, με ποσοστό τουλάχιστον $K(S) - |A|$, μέχρι τη χρονική στιγμή $\sigma^i \cdot C_S^P$, έχουμε ότι για κάθε $i \geq 0$

$$\sum_{e \in S \setminus A} \int_{\tau=0}^{r^i \cdot C_S^P} \tilde{x}_e(\tau) d\tau \geq Q(K(S) - |A|)(1 + \rho i).$$

Για κάθε $e \in S \setminus A$ έχουμε μια τυχαία μεταβλητή X_e , τέτοια ώστε $X_e = 1$ αν και μόνο αν

$t_{e,\alpha_e} \leq r^i \cdot C_S^P$. Παρατηρήστε ότι οι X_e είναι ανεξάρτητες μεταξύ τους, αφού η τιμή των X_e καθορίζεται από τα α_e , τα οποία επιλέγονται τυχαία και ανεξάρτητα. Έστω $X = \sum_{e \in S \setminus A} X_e$ και $\mu = E[X]$. Παρατηρώντας ότι $Pr[X_e = 1] = \int_{\tau=0}^{r^i \cdot C_S^P} \tilde{x}_e(\tau) d\tau$, προκύπτει ότι

$$\mu = E[X] = \sum_{e \in S \setminus A} \int_{\tau=0}^{r^i \cdot C_S^P} \tilde{x}_e(\tau) d\tau \geq Q(K(S) - |A|)(1 + \rho i)$$

Προσέξτε ότι $p_{S,i} \leq Pr[X < K(S) - |A|]$. Ξεχωρίζουμε κάποιες περιπτώσεις, ανάλογα με την τιμή του $K(S) - |A|$. Υποθέσουμε φυσικά ότι $K(S) > |A|$, αφού διαφορετικά $p_{S,i} = 0$.

Περίπτωση (α): $K(S) - |A| = 1$. Χρησιμοποιώντας το γεγονός ότι οι X_e είναι ανεξάρτητες μεταξύ τους, έχουμε:

$$\begin{aligned} Pr[X < K(S) - |A|] &= Pr[X = 0] \\ &= \prod_{e \in S \setminus A} (1 - Pr[X_e = 1]) \\ &\leq \exp\left(-\sum_{e \in S \setminus A} Pr[X_e = 1]\right) = \exp(-\mu) \\ &\leq \exp(-Q(1 + \rho i)) \end{aligned}$$

Περίπτωση (β): $K(S) - |A| = 2$.

$$\begin{aligned} &Pr[X < K(S) - |A|] \\ &= Pr[X = 0] + Pr[X = 1] \\ &\leq \exp(-\mu) + \sum_{e' \in S \setminus A} Pr[X_{e'} = 1] \prod_{e \in S \setminus (A \setminus \{e'\})} (1 - Pr[X_e = 1]) \\ &\leq \exp(-2Q(1 + \rho i)) + \sum_{e' \in S \setminus A} Pr[X_{e'} = 1] \exp\left(-\sum_{e \in S \setminus (A \setminus \{e'\})} Pr[X_e = 1]\right) \\ &\leq \exp(-2Q(1 + \rho i)) + \mu \cdot \exp(-\mu + 1) \\ &\leq \exp(-2Q(1 + \rho i)) + 2Q(1 + \rho i) \exp(-2Q(1 + \rho i) + 1) \end{aligned}$$

Περίπτωση (γ): $K(S) - |A| \geq 3$. Αυτή τη περίπτωση την αντιμετωπίζουμε όπως κάναμε στην ανάλυση του δεύτερου αλγορίθμου. Χρησιμοποιούμε και το ίδιο Chernoff bound [31] $Pr[X < -\mu(1 - \beta)] \leq \exp\left(-\frac{1}{2}\beta^2\mu\right)$ (στη δεύτερη ανισότητα). Η κύρια διαφορά είναι ότι εδώ εκμεταλλευόμαστε το γεγονός ότι $K(S) - |A| \geq 3$, πράγμα που βοηθά να αποκτήσουμε

ένα καλύτερο φράγμα.

$$\begin{aligned}
 \Pr[X < K(S) - |A|] &\leq \Pr\left[X < \frac{\mu}{Q(1 + \rho i)}\right] \\
 &= \Pr\left[X < \mu \left(1 - \left(1 - \frac{1}{Q(1 + \rho i)}\right)\right)\right] \\
 &\leq \exp\left(-\frac{1}{2} \left(1 - \frac{1}{Q(1 + \rho i)}\right)^2 \cdot \mu\right) \\
 &\leq \exp\left(-\frac{1}{2} \left(1 - \frac{1}{Q(1 + \rho i)}\right)^2 \cdot 3Q(1 + \rho i)\right)
 \end{aligned}$$

Διαλέγοντας το μέγιστο από τις τρεις παραπάνω περιπτώσεις ολοκληρώνουμε την απόδειξη. \square

Στο ακόλουθο λήμμα, φράσσουμε τον αναμενόμενο χρόνο κάλυψης της τελική διάταξης σ^R .

Λήμμα 5.8. Για κάθε $\rho \leq 1$, $r > 1$, $Q \geq 1$, έχουμε ότι:

$$E\left[\sum_{S \in \mathcal{S}} C_S^R\right] \leq Q \left(1 + \frac{\rho}{r-1}\right) \left(1 + (r-1) \sum_{i=0}^{\infty} r^i \rho_i\right) \sum_{S \in \mathcal{S}} C_S^P + \sum_{S \in \mathcal{S}} K(S).$$

Απόδειξη. Πριν δώσουμε, όμως, μια τυπική απόδειξη, δίνουμε μια απλή επεξήγηση για το πώς θα πάρουμε το φράγμα. Πρώτον, η απώλεια του παράγοντα $Q \left(1 + \frac{\rho}{r-1}\right)$ προέρχεται από ισοπέδωση, της “ψήλης” διάταξης $\tilde{\sigma}$, η οποία έχει ψηλώσει εξαιτίας των πολλαπλών διατάξεων σ_i που τοποθετούνται η μια πάνω στην άλλη. Δώσαμε ένα φράγμα για το ύψος της $\tilde{\sigma}$ στο Λήμμα 5.6. Ο άλλος όρος $\left(1 + (r-1) \sum_{i=0}^{\infty} r^i \rho_i\right)$ προέρχεται από τον ορισμό της ρ_i . Το σύνολο S καλύπτεται μετά την χρονική στιγμή $r^i \cdot C_S^P$ με πιθανότητα τουλάχιστον $1 - \rho_i$. Ο τελικός όρος προκύπτει από την ακόλουθη μεθοδολογία. Μετράμε ξεχωριστά για κάθε σύνολο S τον αναμενόμενο αριθμό από στοιχεία που έχουν επιλεχθεί πριν την C_S^I στη τελική διάταξη σ^R και στο τέλος αθροίζουμε.

Θυμηθείτε ότι, η C_S^I είναι η πρώτη χρονική μονάδα t τέτοια ώστε $|\{e \in S : t_{e, \alpha_e} \leq t\}| \geq K(S)$. Πρώτα θα βρούμε ένα άνω φράγμα για την $E[C_S^I]$. Από τον ορισμό του ρ_i , έχουμε $\Pr[C_S^I > r^{i-1} \cdot C_S^P] \leq \rho_{i-1}$ για κάθε $i \geq 1$. Επομένως έπεται:

$$\begin{aligned}
 E[C_S^I] &\leq \left((1 - \rho_0) + \sum_{i=1}^{\infty} r^i (\rho_{i-1} - \rho_i) \right) C_S^P \\
 &= \left(1 + (r-1) \sum_{i=1}^{\infty} r^i \rho_i \right) C_S^P
 \end{aligned}$$

Προσέξτε πως C_S^I δεν είναι απαραίτητα ο χρόνος κάλυψης C_S^R του S στη τελική διάταξη σ^R . Αυτό διότι ενδέχεται να έχουμε επιλεχθεί περισσότερα από C_S^I στοιχεία μέχρι τη στιγμή C_S^I .

Αφού η σ^l είναι μη εφικτή ακέραια διάταξη και μπορεί να αναθέσει πολλά στοιχεία στην ίδια χρονοθυρίδα. Μετρώντας τον αριθμό των στοιχείων που επιλέγονται μέχρι και την χρονική μονάδα C_S^l παίρνουμε τη ακόλουθη σχέση μεταξύ C_S^R και C_S^l . Αυτά τα στοιχεία υπολογίζονται με διαφορετικό τρόπο, ανάλογα με το αν ανήκουν στο σε S ή όχι.

Έχουμε τα $K(S)$ στοιχεία που ανήκουν στο S και όλα όσα δεν ανήκουν στο S άλλα έχουν $t_{e,\alpha_e} \leq C_S^l$ και για αυτό έχουν επιλεχθεί.

$$C_S^R \leq |\{e \in [n] \setminus S \mid t_{e,\alpha_e} \leq C_S^l\}| + K(S)$$

Εφόσον, για κάθε $e \notin S$ και χρονική στιγμή $t \in \mathbb{R}_+$, $Pr[t_{e,\alpha_e} \leq t] = \min \left\{ 1, \int_{\tau=0}^t \tilde{x}_e(\tau) d\tau \right\}$, από το Λήμμα 5.6, έχουμε:

$$E[|\{e \in [n] \setminus S : t_{e,\alpha_e} \leq t\}|] \leq \sum_{e \in [n]} \int_{\tau=0}^t \tilde{x}_e(\tau) d\tau \leq Q \left(1 + \frac{\rho}{r-1} \right) t$$

ως εκ τούτου

$$\begin{aligned} & E[|\{e \in [n] \setminus S \mid t_{e,\alpha_e} \leq C_S^l\}|] \\ &= \int_{\tau=0}^{\infty} (E[|\{e \in [n] \setminus S : t_{e,\alpha_e} \leq \tau\}|] \cdot Pr[C_S^l = \tau]) d\tau \\ &\leq \int_{\tau=0}^{\infty} \left(Q \left(1 + \frac{\rho}{r-1} \right) \tau \cdot Pr[C_S^l = \tau] \right) d\tau \\ &\leq Q \left(1 + \frac{\rho}{r-1} \right) \int_{\tau=0}^{\infty} (\tau \cdot Pr[C_S^l = \tau]) d\tau \\ &= Q \left(1 + \frac{\rho}{r-1} \right) E[C_S^l] \end{aligned}$$

οπότε προκύπτει ότι

$$\begin{aligned} E[C_S^R] &\leq Q \left(1 + \frac{\rho}{r-1} \right) E[C_S^l] + K(S) \\ &\leq Q \left(1 + \frac{\rho}{r-1} \right) \left(1 + (r-1) \sum_{i=1}^{\infty} r^i \rho_i \right) C_S^p + K(S). \end{aligned}$$

Αθροίζοντας για όλα τα σύνολα S ολοκληρώνουμε την απόδειξη. □

Τώρα είμαστε έτοιμοι να ολοκληρώσουμε την απόδειξη του Θεωρήματος 5.3.

Απόδειξη. (Απόδειξη του Θεωρήματος 5.3) Παρατήρησε ότι το $\sum_{S \in \mathcal{S}} K(S)$ είναι ένα κάτω φράγμα του κόστους κάθε preemptive διάταξης, έτσι στο Λήμμα 5.8, αρκεί να δείξουμε ότι:

$$Q \left(1 + \frac{\rho}{r-1} \right) \left(1 + (r-1) \sum_{i=1}^{\infty} r^i \rho_i \right) C_S^p \leq 5.2.$$

Θέτουμε $Q = 2.65$, $r = 1.40$ και $\rho = 0.22$. Εφαρμόζοντας το Λήμμα 5.7, παίρνουμε

$$Q \left(1 + \frac{\rho}{r-1}\right) \left(1 + (r-1) \sum_{i=1}^{30} r^i \rho_i\right) C_S^P \leq 5.13$$

(ο υπολογισμός έγινε αριθμητικά). Για $i > 30$, αποδεικνύουμε τώρα πως:

$$\begin{aligned} r^i \rho_i &\leq r^i \max(K_{1i}, K_{2i}, K_{3i}) \\ &\leq 2 \exp(-Q(1 + \rho i)) r^i \\ &< 1.145 \cdot (e^{-Q\rho} r)^i \\ &< 1.145 \cdot (0.782)^i \end{aligned}$$

Από το Λήμμα 5.7 έχουμε

1. $K_{1i} \leq 2 \exp(-Q(1 + \rho i))$
2. Δεδομένου ότι $2e^{-\frac{z}{\exp(z)}} \leq 1$ για $z \geq 3$ και $Q(1 + \rho i) \geq 2.65(1 + 30 \cdot 0.22) > 10$ προκύπτει ότι:

$$\begin{aligned} K_{2i} &= \exp(-2Q(1 + \rho i)) + 2Q(1 + \rho i) \exp(-2Q(1 + \rho i) + 1) \\ &= \exp(-2Q(1 + \rho i)) + 2e^{\frac{Q(1 + \rho i)}{\exp(Q(1 + \rho i))}} \exp(-Q(1 + \rho i)) \\ &\leq 2 \exp(-Q(1 + \rho i)). \end{aligned}$$

3. Τέλος, από το γεγονός ότι $Q(1 + \rho i) > 10$, γνωρίζουμε:

$$\begin{aligned} K_{3i} &= \exp\left(-1.5Q \left(\left(1 - \frac{1}{(1 + \rho i)Q}\right)^2 (1 + \rho i)\right)\right) \\ &\leq \exp\left(-1.5Q(1 + \rho i) \left(1 - \frac{1}{10}\right)^2\right) \\ &\leq \exp(-1.2Q(1 + \rho i)) \\ &\leq 2 \exp(Q(1 + \rho i)). \end{aligned}$$

Ως εκ τούτου προκύπτει ότι

$$\begin{aligned}
 & Q \left(1 + \frac{\rho}{r-1} \right) \left((r-1) \sum_{i=31}^{\infty} r^i \rho_i \right) \\
 & < Q \left(1 + \frac{\rho}{r-1} \right) \left(1.145(r-1) \sum_{i=31}^{\infty} (0.782)^{31} \right) \\
 & = Q \left(1 + \frac{\rho}{r-1} \right) \left(1.145(r-1)(0.782)^{31} \sum_{i=0}^{\infty} (0.782)^i \right) \\
 & < Q \left(1 + \frac{\rho}{r-1} \right) \left(1.145(r-1)(0.782)^{31} \cdot \frac{1}{1-0.782} \right) < 0.02
 \end{aligned}$$

Πρέπει επίσης να προσθέσουμε μια μονάδα επειδή $\sum_{S \in \mathcal{S}} K(S) \leq \sum_{S \in \mathcal{S}} C_S^P$. Έτσι καταλήγουμε πως ο λόγος που πληρώνουμε για να μετατρέψουμε μια preemptive διάταξη του Generalized Min-Sum Set Cover σε μια non-preemptive είναι το πολύ 6.2. \square

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα προηγούμενα κεφάλαια, μελετήσαμε αναλυτικά τρεις προσεγγιστικούς αλγόριθμους για το Generalized Min-Sum Set Cover. Αντιμετωπίσαμε τη βελτίωση και εξέλιξη των τεχνικών που χρησιμοποιούν, καθώς και την επίπτωση που έχουν οι βελτιώσεις αυτές στην εγγύηση απόδοσης. Συνοψίζοντας, τα κυριότερα σημεία για κάθε αλγόριθμο:

Πρώτος αλγόριθμος [3]

- Δόθηκε από τους Nikhil Bansal, Anuram Gupta και Ravishankar Krishnaswamy το 2010.
- Ο πρώτος με σταθερή εγγύηση απόδοσης για το Generalized Min-Sum Set Cover.
- Εγγύηση απόδοσης 485.

Δεύτερος αλγόριθμος [41]

- Δόθηκε από τους Martin Skutella και David P. Williamson το 2011.
- Οι βασικές ιδέες και το γραμμικό πρόγραμμα είναι ίδια με τον πρώτο αλγόριθμο.
- Χρήση των α -points.
- Εγγύηση απόδοσης 27.78.

Τρίτος αλγόριθμος [20]

- Δόθηκε από τους Sungjin Im, Maxim Sviridenko και Ruben van der Zwaan το 2012.
- 2-προσεγγιστικός αλγόριθμος για το preemptive, ο οποίος είναι και ό,τι καλύτερο μπορούμε να έχουμε, δεδομένου ότι ισχύει η UGC [24].
- Αλγόριθμος που μετατρέπει κάθε preemptive λύση C σε μια non-preemptive, με κόστος το πολύ $6.2C$.
- Συνδυάζοντας τους δύο παραπάνω παίρνουμε ένα 12.4-προσεγγιστικό αλγόριθμο για τη non-preemptive εκδοχή του προβλήματος.

Οι δυο πρώτοι αλγόριθμοι και ο αλγόριθμος που μετατρέπει κάθε preemptive λύση σε μια non-preemptive, έχουν ένα κοινό μοτίβο που συναντάμε πολύ συχνά σε προβλήματα χρονοδρομολόγησης και παρουσιάζουν πολλές ομοιότητες. Τα τρία βασικά κοινά τους χαρακτηριστικά είναι το φούσκωμα της αρχικής λύσης, είτε αυτή είναι λύση του LP είτε είναι preemptive λύση, τα α -points και το καλό υποσύνολο A . Στην ανάλυση κάθε αλγόριθμου, εξάγεται ένα άνω φράγμα για το πόσο φουσκώνει η αρχική λύση, το οποίο χρησιμοποιείται για να βρεθεί το τελικό άνω φράγμα. Η ιδέα των α -points είναι, εν ολίγοις, να διαλέγουμε τυχαία και ανεξάρτητα $\alpha_e \in [0, 1]$ για κάθε $e \in U$ και να υπολογίζουμε τα t_{e,α_e} , τα οποία δεν είναι παρά η πρώτη t στην οποία $\sum_{t'=1}^t x_{e,t'} \geq \alpha_e$. Να σημειωθεί ότι ακόμα και στην ανάλυση του πρώτου αλγόριθμου χρησιμοποιούνται α -points με $\alpha = \frac{1}{2}$ για κάθε $e \in U$. Το καλό υποσύνολο A ενός συνόλου S περιέχει τα στοιχεία που έχουν καλυφθεί με πιθανότητα 1 στην διάταξη που κατασκευάζει ο αντίστοιχος αλγόριθμος. Και στις τρεις περιπτώσεις ορί-

ζεται ένα τέτοιο σύνολο, με σκοπό να βρεθεί ένα άνω φράγμα στον αναμενόμενο χρόνο κάλυψης $\text{Con}^{\text{alg}}(S)$ ενός συνόλου S και, την συνέχεια, στον αναμενόμενο χρόνο κάλυψης όλων των συνόλων μαζί (το οποίο είναι και το ζητούμενο).

Ο τρίτος αλγόριθμος (που μετατρέπει κάθε preemptive λύση σε non-preemptive) είναι και αυτός που δίνει τα καλύτερα αποτελέσματα γιατί ξεκινάει από μια preemptive λύση και όχι από μια λύση που δίνει ένα γραμμικό πρόγραμμα. Κάνει πολύ λεπτομερές και σύνθετο φούσκωμα της αρχικής λύσης. Παίρνει τρεις διαφορετικές περιπτώσεις ανάλογα με το μέγεθος του καλού συνόλου A . Ο αλγόριθμος δεν είναι tight, δηλαδή, παρότι έχει υπολογιστεί πως οι λύσεις που δίνει έχουν κόστος το πολύ $6.2C$, ενδέχεται να έχει ακόμα καλύτερη απόδοση την οποία μπορεί να πάρουμε από μια λεπτομερέστερη ανάλυση του.

Ξέρουμε ότι το κάτω φράγμα για το Generalized Min-Sum Set Cover είναι 4 εκτός αν $P = NP$ επειδή είναι γενίκευση του Min-Sum Set Cover και κληρονομεί το κάτω φράγμα του. Όποτε το πρόβλημα παραμένει ανοικτό. Υπάρχουν πολλές κατευθύνσεις στις οποίες μπορούμε να κοιτάξουμε ώστε να βελτιώσουμε την εγγύηση απόδοσης από 12.4, που έχουμε μέχρι στιγμής, σε 4. Η πρώτη θα ήταν, όπως είπαμε παραπάνω, να γίνει μια λεπτομερέστερη ανάλυση του τρίτου αλγορίθμου ώστε να διαπιστωθεί αν έχει καλύτερη απόδοση. Μια άλλη, είναι να κατασκευαστεί ένας καινούριος αλγόριθμος για το πρόβλημα, ο οποίος να δέχεται μια preemptive λύση με κόστος C και να κατασκευάζει, σε πολυωνυμικό χρόνο, μια non-preemptive λύση με κόστος το πολύ $2C$. Κάτι τέτοιο, ενδεχομένως, να χρειαζόταν μια τελείως διαφορετική προσέγγιση του προβλήματος. Το Generalized Min-Sum Set Cover είναι πρόβλημα χρονοδρομολόγησης αλλά και ένα πρόβλημα κάλυψης, όπως το Set Cover και το Hitting Set. Οι τεχνικές που ακολούθησαν οι τρεις αλγόριθμοι τους οποίους μελετήσαμε, χρησιμοποιούνται με επιτυχία σε προβλήματα χρονοδρομολόγησης, ωστόσο, για το συγκεκριμένο πρόβλημα, ίσως να μην μπορούν να δώσουν καλύτερα αποτελέσματα.

Μια διαφορετική κατεύθυνση, θα ήταν να δούμε ειδικές περιπτώσεις του προβλήματος, όπως την περίπτωση όπου η απαίτηση κάλυψης είναι ίδια για όλα τα σύνολα $K(S_i) = c$ για κάθε $i \in U$. Μπορούμε, επίσης, να ερευνήσουμε αν επηρεάζουν την εγγύηση απόδοσης διάφοροι παράμετροι ξ , όπως, για παράδειγμα, $\xi = K(S)$ ή $\xi = K(S)/|S|$. Θα μπορούσαμε ακόμα δούμε αν η σχέση μεταξύ preemptive και non-preemptive εξαρτάται από κάποια παράμετρο ξ . Ξέρουμε πως για την παράμετρο $\xi = \min_S \{K(S)/|S|\}$, όταν $\xi = 1$ το κόστος της preemptive διάταξης είναι ίσο με της non-preemptive [20].

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

absolute performance guarantee	απόλυτη εγγύηση απόδοσης
approximation algorithm	προσεγγιστικός αλγόριθμος
approximation scheme	προσεγγιστικό σχήμα
asymptotic performance guarantee	ασυμπτωτική εγγύηση απόδοσης
Chernoff bound	φράγμα Chernoff
ellipsoid method	μέθοδος των ελλειψοειδών
feasible solution	εφικτή λύση
hardness of approximation	δυσκολία προσέγγισης
integer linear programming	ακέραιος γραμμικός προγραμματισμός
linear programming	γραμμικός προγραμματισμός
LP relaxation	χαλαρωμένο γραμμικό πρόγραμμα
matroid	μητροειδές
matroid base	βάση μητροειδούς
matroid polytope	πολύτοπο μητροειδούς
matroid rank	τάξη μητροειδούς
performance guarantee	εγγύηση απόδοσης
relative performance guarantee	σχετική εγγύηση απόδοσης
rounding	στρογγυλοποίηση
separation oracle	μαντείο διαχωρισμού
scheduling problem	πρόβλημα χρονοδρομολόγησης
strong duality	ισχυρός δυϊσμός
weak duality	ασθενής δυϊσμός
α -points	α -τιμές

ΣΥΝΤΜΗΣΕΙΣ, ΑΡΚΤΙΚΟΛΕΞΑ ΚΑΙ ΑΚΡΩΝΥΜΙΑ

APX	approximable
DP	dual program
FPTAS	fully polynomial-time approximation scheme
ILP	integer linear programming
LP	linear programming
NP	nondeterministic polynomial time
P	polynomial time
PTAS	polynomial-time approximation scheme
UGC	Unique Games Conjecture

ΑΝΑΦΟΡΕΣ

- [1] Giorgio Ausiello, M. Protasi, A. MarchettiSpaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. SpringerVerlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1999.
- [2] Yossi Azar, Iftah Gamzu, and Xiaoxin Yin. *Multiple intents reranking*. In STOC, pages 669-678, 2009.
- [3] Nikhil Bansal, Anupam Gupta, and Ravishankar Krishnaswamy. *A constant factor approximation algorithm for generalized minsum set cover*. In SODA, pages 1539-1545, 2010.
- [4] Amotz BarNoy, Mihir Bellare, Magnús M. Halldórsson, Hadas Shachnai, and Tami Tamir. On chromatic sums and distributed resource allocation. *Inf. Comput.*, 140(2):183- 202, 1998.
- [5] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1st edition, 1997.
- [6] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *A sharp concentration inequality with applications*. *Random Struct. Algorithms*, 16(3):277-292, 2000.
- [7] C. Chekuri and S. Khanna. *Approximation algorithms for minimizing average weighted completion time*. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. CRC Press, Inc., Boca Raton, FL, USA, 2004.
- [8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms* (3. ed.). MIT Press, 2009.
- [9] William H. Cunningham. *Testing membership in matroid polyhedra*. *J. Comb. Theory, Ser. B*, 36(2):161-188, 1984.
- [10] G.B. Dantzig and Mukund N. Thapa. *Linear Programming 2: Theory and extensions*. Springer series in operations research. Springer-Verlag New York, Inc., 2003.
- [11] George B. Dantzig and Mukund N. Thapa. *Linear programming 1: introduction*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.
- [12] Sanjoy Dasgupta, Christos H. Papadimitriou, and Umesh V. Vazirani. *Algorithms*. McGraw-Hill, 2008.
- [13] Uriel Feige. *A threshold of $\ln n$ for approximating set cover*. *J. ACM*, 45(4):634-652, 1998.
- [14] Uriel Feige, László Lovász, and Prasad Tetali. *Approximating min sum set cover*. *Algorithmica*, 40(4):219-234, 2004.
- [15] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [16] Igor Griva, Stephen G. Nash, and Ariela Sofer. *Linear and Nonlinear Optimization* (2. ed.). SIAM, 2008.
- [17] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization, volume 2 of Algorithms and Combinatorics*. Springer, 1988.
- [18] Leslie A. Hall, David B. Shmoys, and Joel Wein. *Scheduling to minimize average completion time: Off-line and online algorithms*. In SODA, pages 142-151, 1996.
- [19] Dorit S. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., Boston, MA, USA, 1997.
- [20] Sungjin Im, Maxim Sviridenko, and Ruben van der Zwaan. *Preemptive and nonpreemptive generalized min sum set cover*. In STACS, pages 465-476, 2012.
- [21] David S. Johnson. *Approximation algorithms for combinatorial problems*. *J. Comput. Syst. Sci.*, 9(3):256-278, 1974.
- [22] David Karger, Cliff Stein, and Joel Wein. *Scheduling algorithms*, 1997.
- [23] Richard M. Karp. *Reducibility among combinatorial problems*. In *Complexity of Computer Computations*, pages 85-103, 1972.
- [24] Subhash Khot. *On the power of unique 2-prover 1-round games*. In *IEEE Conference on Computational Complexity*, page 25. IEEE Computer Society, 2002.
- [25] Jon M. Kleinberg and Éva Tardos. *Algorithm design*. Addison-Wesley, 2006.
- [26] E. L. Lawler. *Sequencing jobs to minimize total weighted completion time subject to precedence*

- constraints*. Annals of Discrete Mathematics, 2:75-90, 1978.
- [27] J. K. Lenstra and Rinnooy A. H. G. Kan. *Complexity of Scheduling under Precedence Constraints*. Operations Research, 26(1), 1978.
- [28] Harry R. Lewis and Christos H. Papadimitriou. *Elements of the theory of computation (2. ed.)*. Prentice Hall, 1998.
- [29] Carsten Lund and Mihalis Yannakakis. *On the hardness of approximating minimization problems*. J. ACM, 41(5):960-981, 1994.
- [30] Michael Mitzenmacher and Eli Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.
- [31] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [32] James G. Oxley. *Matroid theory*. Oxford University Press, 1992.
- [33] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [34] Maurice Queyranne and Maxim Sviridenko. *A $(2+\epsilon)$ -approximation algorithm for the generalized preemptive open shop problem with minsum objective*. J. Algorithms, 45(2):202-212, 2002.
- [35] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.
- [36] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
- [37] Andreas S. Schulz and Martin Skutella. *Randombased scheduling: New approximations and l_p lower bounds*. In RANDOM, pages 119-133, 1997.
- [38] Paul D. Seymour. *Decomposition of regular matroids*. J. Comb. Theory, Ser. B, 28(3):305- 359, 1980.
- [39] Michael Sipser. *Introduction to the theory of computation*. PWS Publishing Company, 1997.
- [40] Martin Skutella. *Efficient approximation and online algorithms. chapter List scheduling in order of ϵ -points on a single machine, pages 250-291*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [41] Martin Skutella and David P. Williamson. *A note on the generalized minsum set cover problem*. Oper. Res. Lett., 39(6):433-436, 2011.
- [42] Vijay V. Vazirani. *Approximation algorithms*. Springer, 2001.