ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Προσεγγιστικοί Αλγόριθμοι για την Κάλυψη Πολυγωνικών Περιοχών

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Ευριπίδη Α. Μάρκου

Πτυχιούχου Φυσικού Πανεπιστημίου Ιωαννίνων (1993)

Αθήνα, Ιούνιος 2003

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Προσεγγιστικοί Αλγόριθμοι για την Κάλυψη Πολυγωνικών Περιοχών

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

του

Ευριπίδη Α. Μάρκου

Συμβουλευτική Επιτροπή: Ευστάθιος Ζάχος
Φώτω Αφράτη
Αντρέας Σταφυλοπάτης

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 12η Ιουνίου 2003.

....	....	....
Ευστάθιος Ζάχος Καθηγητής Ε.Μ.Π.	Φώτω Αφράτη Καθηγήτρια Ε.Μ.Π.	Αντρέας Σταφυλοπάτης Καθηγητής Ε.Μ.Π.
....	....	....
Τίμος Σελλής Καθηγητής Ε.Μ.Π.	Ιωάννης Εμίρης Αν. Καθηγητής ΕΚΠΑ	Αλέξανδρος Παπαϊωάννου Επίκ. Καθηγητής Ε.Μ.Π.
	....	
	Νικόλαος Παπασπύρου Λέκτορας Ε.Μ.Π.	

Αθήνα, Ιούνιος 2003

....
Ευριπίδης Α. Μάρκου
Διδάκτωρ Ε.Μ.Π.

Περίληψη

Η ανάπτυξη των ασύρματων δικτύων επικοινωνίας (κινητά τηλέφωνα, κλπ) έχει δημιουργήσει πολλά προβλήματα έρευνας: ελαχιστοποίηση του αριθμού των σταθμών, ελαχιστοποίηση των χρησιμοποιούμενων συχνοτήτων, κλπ. Δύο σημεία μπορούν να επικοινωνήσουν αν καλύπτονται (δηλαδή επικοινωνούν) με ένα σταθμό. Είναι γνωστό το πρόβλημα της τοποθέτησης ενός ελάχιστου αριθμού σταθμών έτσι ώστε όλα τα σημεία να καλύπτονται. Μια παραλλαγή αυτού του προβλήματος είναι: πώς μπορούμε να τοποθετήσουμε ένα δεδομένο αριθμό σταθμών έτσι ώστε να καλύπτεται ένας μέγιστος αριθμός σημείων; Η τοπολογία της περιοχής καθορίζει εάν η επικοινωνία μεταξύ δύο σημείων είναι δυνατή ή όχι. Το μοντέλο θα πρέπει να κρατά τα χαρακτηριστικά της περιοχής. Γραφήματα, τεραίν, πολύγωνα με ή χωρίς τρύπες, κλπ, έχουν χρησιμοποιηθεί σαν μοντέλα.

Εδώ μελετούμε μια σειρά από προβλήματα κάλυψης πολυγώνων με ή χωρίς τρύπες. Ειδικότερα, εξετάζονται περιπτώσεις στις οποίες ο στόχος είναι η κάλυψη μιας μέγιστης αξίας από δεδομένες περιοχές με αξία που βρίσκονται πάνω στην περίμετρο ή στο εσωτερικό του πολυγώνου χρησιμοποιώντας έναν αριθμό από φύλακες με δεδομένο ανώτατο συνολικό κόστος. Οι φύλακες (σταθμοί) τοποθετούνται πάνω στην περίμετρο του πολυγώνου.

Αποδεικνύουμε ότι όλες αυτές οι περιπτώσεις είναι NP-hard. Ειδικότερα, προτείνουμε προσεγγιστικούς αλγόριθμους πολυωνυμικού χρόνου με σταθερούς παράγοντες προσέγγισης. Αποδεικνύουμε επίσης ότι όλες αυτές οι περιπτώσεις είναι APX-hard και συνεπώς APX-complete: κατασκευάζουμε μια αναγωγή διατήρησης χάσματος. Δείχνουμε ότι με λίγες μετατροπές μπορούμε να εφαρμόσουμε αυτήν την αναγωγή σε μια σειρά προβλημάτων υπολογιστικής γεωμετρίας.

Γίνεται επίσης μελέτη κάποιων προβλημάτων σε γράφους αποδεικνύοντας ότι αυτά τα προβλήματα είναι ευκολότερα σε γράφους ορατότητας από ότι σε γενικούς γράφους (πχ. Maximum Clique Cover).

Κατά τη διάρκεια της μελέτης των παραπάνω προβλημάτων, εισάγαμε τα εξής νέα στοιχεία: α) **βάρη** ή αξίες σε τμήματα της περιμέτρου του πολυγώνου και σε περιοχές στο εσωτερικό του πολυγώνου, β) **κόστος** σε κάθε υποψήφια θέση φύλακα (σταθμού), γ) την χρήσιμη έννοια της **επίβλεψης (watching)** ενός συνόλου σημείων, τμημάτων, ή περιοχών του πολυγώνου και δ) έναν τρόπο κατακερματισμού της περιμέτρου και του εσωτερικού του πολυγώνου ως προς την ορατότητα. Τέλος, ορίζουμε μια ενδιαφέρουσα παραλλαγή του προβλήματος SET COVER όπου τα στοιχεία των συνόλων έχουν διαστάσεις.

Abstract

The development of wireless communication technology (mobile phones, etc) created a number of research problems: minimization of the number of transmitting antennas, minimization of the number of used frequencies, etc. Two points can communicate if they are covered by (i.e. can communicate with) an antenna. The well known problem is how to place stations so that all points are covered (visible) and the number of stations is minimum. A variation is: a number of stations is given, and we are asked to cover as many points as possible. Whether communication between two points is possible or blocked (direct point to point communication) it depends on the area topology. So the model must keep the properties of the topology. Graphs, terrains, polygons with (or without) holes, etc, have been used as models.

A number of covering (or guarding) problems in a polygon with (or without) holes are investigated here. More specifically, a bunch of cases are examined where the goal is to guard (cover) a maximum weight of given valuable areas on the boundary or in the interior of the polygon using a number of guards (stations) with costs within a given budget. The guards (stations) are placed on the boundary of the polygon.

We prove here that all these cases are NP-hard. Moreover, polynomial - time algorithms are proposed which approximate optimal solutions within constants. We also prove that all these cases are APX-hard and thus APX-complete: A gap-preserving reduction (which is a relatively new concept), is presented. Furthermore it is shown that this reduction can be applied in a number of computational geometry problems.

A further investigation and characterization of some graph problems has been done showing that in visibility graphs they are easier than in general graphs (e.g. Maximum Clique Cover).

While investigating the above problems, new concepts that have been introduced are: a) **weights** or values on pieces of the polygon's boundary and areas of the polygon's interior, b) **costs** on candidate guard (station) positions in the polygon, c) the useful and promising concept of **watching** a set of points or line segments or areas as opposed to completely **overseeing** or covering it and d) a way to discretize the boundary and the interior of the polygon with respect to visibility.

Finally, an interesting case of the SET COVER problem has been introduced, where the elements of the universe that must be covered have dimensions and are not merely points.

Ευχαριστίες

Πρώτα από όλους θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή Στάθη Ζάχο για τις πολλές στιγμές που ήταν δίπλα μου σαν δάσκαλος αλλά και σαν φίλος.

Αισθάνομαι πολύ τυχερός που μαθήτευσα δίπλα σε έναν άνθρωπο που αγαπά τόσο πολύ να διδάσκει. Το πάθος του για την έρευνα και οι πλούσιες γνώσεις που μου μετέδιδε έπαιξαν και παίζουν αποφασιστικό ρόλο στον καθορισμό και την επίτευξη των στόχων μου. Η επιμονή του στην μαθηματική κομψότητα και σαφήνεια συνέβαλαν αποφασιστικά στη διαμόρφωση της διδακτορικής διατριβής και στην ποιότητα των αποτελεσμάτων της. Η πολύτιμη φιλία του και η πίστη του σε εμένα μου έδιναν κουράγιο και με στήριζαν τις δύσκολες στιγμές.

Ευχαριστίες ανήκουν στον καθηγητή Τίμο Σελλή για τη βοήθεια που μου προσέφερε απλόχερα όποτε του το ζήτησα όλα αυτά τα χρόνια.

Ευχαριστώ τον επίκουρο καθηγητή της ΣΕΜΦΕ Αλέξη Παπαϊωάννου για τις ωραίες συζητήσεις που είχαμε αλλά και για το ενδιαφέρον που μου ενέπνευσε για διάφορα θέματα έρευνας.

Ευχαριστώ επίσης τον αναπληρωτή καθηγητή του ΕΚΠΑ Γιάννη Εμίρη για τη συμβολή του ιδιαίτερα στο τελευταίο στάδιο της διατριβής και για τα σχόλια και τις παρατηρήσεις του.

Θέλω ακόμη να ευχαριστήσω την καθηγήτρια Φώτω Αφράτη, τον καθηγητή Αντρέα Σταφυλοπάτη και τον λέκτορα Νίκο Παπασπύρου για το ενδιαφέρον τους, την ενθάρρυνσή τους και τις χρήσιμες παρατηρήσεις τους.

Ιδιαίτερα ουσιαστική για την εκπόνηση της διατριβής ήταν η συνεχής συνεργασία μου με τον υποψήφιο διδάκτορα του Ε.Μ.Π. Χριστόδουλο Φραγκουδάκη. Χωρίς τη βοήθειά του και τη φιλία του όλα θα ήταν πολύ πιο δύσκολα (αν όχι αδύνατα) και σίγουρα πολύ πιο άσχημα.

Αποφασιστικής σημασίας ήταν επίσης η συνεργασία που είχα με το Δρα Αρη Παγουρτζή, τον Χρήστο Νομικό, λέκτορα του Πανεπιστημίου Ιωαννίνων και την Κατερίνα Ποτίκα, υποψήφια διδάκτορα του Ε.Μ.Π..

Ευχαριστώ επίσης την Δρα Έλπη Τζαφέστα για τις χρήσιμες παρατηρήσεις, τα σχόλια και τις ενδιαφέρουσες συζητήσεις που είχαμε.

Εκφράζω τις ευχαριστίες μου επίσης στις/στους παρακάτω υποψήφιες/υποψήφιους διδάκτορες του Ε.Μ.Π για την βοήθεια που μου προσέφεραν όλα αυτά τα χρόνια: Πέτρο Πετρόπουλο, Αντώνη Καβαρνό, Όλγα Τσιαντούλη, Πέτρο Ποτίκα, Πάνο Χείλαρη και Νίκο Λεονάρδο.

Στα τελευταία στάδια της διατριβής με βοήθησαν πολύ η κατανόηση, η διευκόλυνση και η συμπαράσταση που μου προσέφεραν οι Διευθυντές του Εργαστηρίου Προσωπικών Υπολογιστών της Σχολής ΕΜΦΕ στο οποίο εργάζομαι, αναπληρωτής καθηγητής Κώστας Φαράκος και επίκουρος καθηγητής Γιάννης Κολέτσος και ο πρόεδρος της Σχολής καθηγητής Γεράσιμος Σπαθής τους οποίους και ευχαριστώ.

Πολύ σημαντικό ρόλο έπαιξε η γνωριμία μου με πολλούς φοιτητές και φοιτήτριες του Ε.Μ.Π. από τους οποίους έμαθα πολλά πράγματα ομορφαίνοντας όλα αυτά τα χρόνια και τους ευχαριστώ ιδιαίτερα.

Σε προσωπικό επίπεδο θέλω να ευχαριστήσω τους γονείς μου Θανάση και Σπυριδούλα και τα αδέρφια μου Κωστή και Σπύρο για την στήριξη και την αγάπη τους όλα αυτά τα χρόνια. Ιδιαίτερα ευχαριστώ την γιαγιά μου Δέσποινα η οποία με τη ζωή της μου δίδαξε την υπομονή αλλά και την επιμονή και προσήλωση στους στόχους μου. Τα λόγια της και η ανάμνησή της με συντρόφευαν και με καθοδηγούσαν σε όλη τη διάρκεια αυτής της προσπάθειας.

Αισθάνομαι ότι αυτή η πορεία δεν θα είχε αρχίσει και είμαι σίγουρος ότι δεν θα έφτανε ποτέ στο τέλος της χωρίς τη συνεχή παρουσία της γυναίκας μου Έλσας. Την ευχαριστώ πολύ για την υπομονή της και τη στήριξη που μου παρείχε, αλλά πιο πολύ την ευχαριστώ για την αγάπη της η οποία ήταν η κινητήριος δύναμη για μένα.

Ευριπίδης Μάρκου

Ιούνιος, 2003

στην Έλσα

Περιεχόμενα

Εισαγωγή

0.1 Εφαρμογές

Τις τελευταίες δεκαετίες όλοι οι τομείς που ασχολούνται με την πληροφορία παρουσιάζουν ραγδαία εξέλιξη. Από τη μία υπάρχει μια συνεχής ανάπτυξη της τεχνολογίας και από την άλλη προκύπτουν συνεχώς καινούριες ανάγκες που πρέπει να ικανοποιηθούν. Μερικές από αυτές είναι: φθηνή, αξιόπιστη και γρήγορη μετάδοση της πληροφορίας, ταξινόμηση και αποθήκευση με τρόπο τέτοιο ώστε να είναι εύκολη και γρήγορη η ανάκτησή της, φιλτράρισμα, αξιολόγηση και επεξεργασία, εξασφάλιση του απόρρητου της επικοινωνίας, κλπ.

Η δημιουργία ενός **δικτύου επικοινωνίας** το οποίο να μπορεί να καλύπτει όσο το δυνατόν μεγαλύτερο αριθμό αιτημάτων για επικοινωνία με φθηνό και αξιόπιστο τρόπο είναι ένα από τα ζητούμενα που προσπαθεί να καλύψει η τεχνολογία.

Οι ανάγκες οδήγησαν στην εξέλιξη των **ενσύρματων δικτύων επικοινωνίας** (οπτικά δίκτυα), αλλά και των **ασύρματων** (κινητά τηλέφωνα). Παράγοντες όπως η ανεξαρτησία, το κόστος και η αξιοπιστία που προσφέρουν τα ασύρματα δίκτυα θα είναι καθοριστικοί για την βαθμιαία κατάργηση μεγάλου μέρους των ενσύρματων δικτύων.

Η μεγάλη ζήτηση για φθηνή και αξιόπιστη ασύρματη επικοινωνία έχει δημιουργήσει πολλά προβλήματα έρευνας με τα οποία ασχολούνται αρκετοί επιστήμονες. Πολλές εταιρείες επίσης δραστηριοποιούνται στο χώρο των ασύρματων δικτύων επικοινωνίας. Η δημιουργία, συντήρηση, επέκταση και λειτουργία ενός ασύρματου δικτύου επικοινωνίας έχει πολύ μεγάλα έξοδα αλλά συγχρόνως και προσδοκίες για μεγάλα έσοδα. Η **έξυπνη δημιουργία** του δικτύου, δηλαδή με τρόπο τέτοιο ώστε να ελαχιστοποιούνται τα έξοδα είναι μια καθοριστική παράμετρος για το μέλλον μιας εταιρείας.

Ένα δίκτυο ασύρματης επικοινωνίας αποτελείται από ένα σύνολο **σταθμών - πομποδεκτών** οι οποίοι εκπέμπουν και λαμβάνουν πληροφορίες. Τα χαρακτηριστικά του κάθε σταθμού και η ποιότητά του καθορίζουν και την ποιότητα ολόκληρου του δικτύου. Το κόστος κατασκευής και συντήρησης του δικτύου, εξαρτάται από το πλήθος των σταθμών του.

Τα αρχικά επενδυτικά κεφάλαια συνήθως καθορίζουν την περιοχή που αρχικά πρέπει να καλύπτει το δίκτυο, η οποία στη συνέχεια διευρύνεται.

Οι σταθμοί του δικτύου μπορεί να βρίσκονται σε δορυφόρους ή μπορεί να είναι επίγειοι. Εμείς θα ασχοληθούμε με τους επίγειους σταθμούς.

Μερικοί από τους παράγοντες που πρέπει να συνυπολογιστούν προκειμένου να κατασκευαστεί ένας σταθμός είναι:

- το κόστος του σταθμού,
- το κόστος της ενοικίασης της περιοχής στην οποία θα τοποθετηθεί,
- παράπονα των κατοίκων για έκθεση στην ακτινοβολία
- γεωγραφικές ιδιαιτερότητες

Ένας στόχος είναι η κάλυψη μιας γεωγραφικής περιοχής τοποθετώντας τον ελάχιστο αριθμό σταθμών.

Επειδή όμως τα χρηματικά κεφάλαια είναι δεδομένα αλλά και για άλλους λόγους που θα εξηγήσουμε παρακάτω, μια ρεαλιστικότερη αντιμετώπιση του προβλήματος είναι: η τοποθέτηση δεδομένου αριθμού σταθμών έτσι ώστε να καλύπτεται όσο το δυνατόν μεγαλύτερο μέρος της περιοχής. Μάλιστα η περιοχή μπορεί να έχει σημεία με διαφορετική **προτεραιότητα κάλυψης**. Για παράδειγμα μπορεί στην περιοχή προς κάλυψη να υπάρχει μια λίμνη την οποία μας ενδιαφέρει να καλύψουμε *αφού* έχουμε καλύψει την υπόλοιπη περιοχή. Επίσης μπορεί να είναι ανάγκη να καλύψουμε κατά προτεραιότητα περιοχές με μεγάλο πληθυσμό (αστικά κέντρα, κοκ).

Μετά λοιπόν και τις παραπάνω παρατηρήσεις ένα ρεαλιστικό πρόβλημα είναι το εξής: με δεδομένο ένα **όριο κόστους** που δεν πρέπει να ξεπεραστεί να τοποθετηθούν σταθμοί σε μια δεδομένη περιοχή με τέτοιο τρόπο ώστε να καλύπτεται ένα μέγιστο μέρος της περιοχής, λαμβάνοντας υπόψιν τις προτεραιότητες κάλυψης.

Επιπλέον θα θέλαμε η επέκταση του δικτύου να γίνεται με τέτοιο τρόπο ώστε αυτό να παραμένει βέλτιστο. Δηλαδή θα θέλαμε η τοπική βελτιστότητα του δικτύου να οδηγεί στη συνολική βελτιστότητα. Με άλλα λόγια, θέλουμε όταν αποφασίζεται η επέκταση του δικτύου, να παραμένει βέλτιστο χωρίς να χρειάζεται να καταργηθούν κάποιοι σταθμοί.

Ανάλογα με τον τρόπο επικοινωνίας μπορούμε να έχουμε δίκτυα στα οποία:

- δύο σημεία για να επικοινωνήσουν μεταξύ τους πρέπει κάθε ένα από αυτά να μπορεί να επικοινωνήσει με κάποιο σταθμό του δικτύου (**point to station communication**)
- δύο σημεία επικοινωνούν μεταξύ τους χωρίς τη μεσολάβηση κάποιου σταθμού (**point to point communication**)

Στην πρώτη περίπτωση, ένα παράδειγμα είναι το δίκτυο κινητής τηλεφωνίας. Η επικοινωνία μεταξύ δύο σημείων γίνεται εάν κάθε σημείο επικοινωνεί με κάποιο σταθμό.

Στη δεύτερη περίπτωση, ένα παράδειγμα είναι η επικοινωνία με walkie talkie. Σε αυτήν την περίπτωση το δίκτυο δεν έχει σταθμούς.

Και στις δύο περιπτώσεις η επικοινωνία γίνεται σε υψηλές συχνότητες της τάξης των 900 MHz, 1800 MHz και 1900 MHz. Μάλιστα υπάρχει η τάση είναι να αυξηθεί και άλλο η συχνότητα επικοινωνίας σε GHz.

Σε αυτές τις συχνότητες τα προβλήματα απώλειας ενέργειας από ανάκλαση και διάθλαση είναι υπολογίσιμα. Τα προβλήματα αυτά εξαφανίζονται αν απαιτήσουμε η επικοινωνία μεταξύ ενός σημείου και ενός σταθμού να είναι δυνατή εάν ο σταθμός είναι ορατός από το σημείο. Έτσι λοιπόν για να είναι δυνατή η επικοινωνία μεταξύ δύο σημείων θα πρέπει: είτε κάθε ένα από αυτά να **βλέπει** κάποιο σταθμό (point to station) ή να **βλέπονται** μεταξύ τους (point to point).

Το αν δύο σημεία βλέπονται εξαρτάται από την τοπολογία της γεωγραφικής περιοχής. Το μοντέλο λοιπόν που θα χρησιμοποιήσουμε πρέπει να διατηρεί τις ιδιότητες της γεωγραφικής περιοχής που αφορούν την ορατότητα.

Σαν γεωμετρικό μοντέλο της περιοχής χρησιμοποιούμε ένα **απλό πολύγωνο**. Η γεωγραφική περιοχή προς κάλυψη αναπαριστάται με το εσωτερικό και την περίμετρο του πολυγώνου.

Δύο σημεία μέσα στο πολύγωνο ή πάνω στην περίμετρο του είναι ορατά το ένα στο άλλο, αν και μόνο αν υπάρχει μια ευθεία γραμμή που τα ενώνει η οποία βρίσκεται ολόκληρη μέσα στο πολύγωνο.

Διάφορα προβλήματα κάλυψης πολυγώνων έχουν μελετηθεί τις τελευταίες τρεις δεκαετίες. Το 1976 κατά τη διάρκεια ενός συνεδρίου ο Victor Klee έθεσε το ακόλουθο πρόβλημα γνωστό και ως **πρόβλημα του μουσείου** (**art gallery**): πόσοι φύλακες χρειάζονται για να βλέπουν κάθε σημείο στο εσωτερικό μιας γκαλερί;

Το μαθηματικό μοντέλο του προβλήματος είναι ένα πολύγωνο το οποίο αντιστοιχεί στο πάτωμα του μουσείου (κάτοψη). Τα προβλήματα που έχουν ερευνηθεί μπορούν να ταξινομηθούν με βάση διάφορα κριτήρια όπως:

- πού επιτρέπεται να τοποθετηθούν φύλακες (στις κορυφές του πολυγώνου, στις πλευρές του πολυγώνου ή οπουδήποτε μέσα στο πολύγωνο)
- τι θέλουμε να φρουρήσουμε (την περίμετρο του πολυγώνου, το εσωτερικό του πολυγώνου)
- εάν το πολύγωνο είναι απλό, ορθογώνιο, κλπ

Έχουν βρεθεί πολλά πάνω και κάτω όρια για τον αριθμό των φυλάκων που είναι αναγκαίοι για την φύλαξη του πολυγώνου. Λιγότερη δουλειά έχει γίνει για τη μελέτη της υπολογιστικής πολυπλοκότητας που έχει η εύρεση καλών θέσεων για τους φύλακες. Τέλος ελάχιστη δουλειά έχει γίνει όσον αφορά την τοποθέτηση δεδομένου αριθμού φυλάκων έτσι ώστε να φυλάσσεται ένα μέγιστο μέρος του πολυγώνου. Περισσότερες λεπτομέρειες υπάρχουν στα [41, 46, 48].

Τα προβλήματα, λοιπόν με τα οποία ασχολούμαστε σε αυτήν την εργασία είναι παραλλαγές του προβλήματος του μουσείου. Μερικές από αυτές τις παραλλαγές ήταν ήδη ορισμένες, ενώ άλλες ορίσαμε εμείς με σκοπό την πιο ρεαλιστική εφαρμογή σε ασύρματα δίκτυα.

Για αυτό το λόγο, σε όλη τη διάρκεια της εργασίας οι όροι που χρησιμοποιούμε στα προβλήματα είναι άλλοτε δανεισμένοι από την ορολογία της γκαλερί (**φύλακες, έργα τέχνης**) και άλλοτε από τα ασύρματα δίκτυα (**πομποδέκτες-σταθμοί, αναμεταδότες**). Η χρησιμοποίηση πότε της μιας και πότε της άλλης ορολογίας γίνεται για λόγους ιστορικούς (αρχικά το πρόβλημα τέθηκε σαν πρόβλημα μουσείου όπως είπαμε) αλλά και για λόγους υπενθύμισης των εφαρμογών στον αναγνώστη.

Θα εξετάσουμε:

- πώς μπορούμε να τοποθετήσουμε φύλακες (ή σταθμούς) σε ένα πολύγωνο έτσι ώστε να καλύπτουμε ένα μέγιστο μήκος της περιμέτρου του,
- πώς μπορούμε να τοποθετήσουμε φύλακες (ή σταθμούς) σε ένα πολύγωνο του οποίου η περίμετρος έχει αξίες έτσι ώστε να καλύπτουμε τη μέγιστη αξία,
- πώς μπορούμε να καλύψουμε ένα μέγιστο εμβαδόν του εσωτερικού του πολυγώνου.

Εισάγουμε την έννοια της **επίβλεψης**: ένα σύνολο σημείων A **επιβλέπεται** από κάποιο σημείο p εάν τουλάχιστον ένα υποσύνολο του A είναι ορατό από το p. Η έννοια της επίβλεψης όπως θα δούμε έχει εφαρμογές στην φρούρηση της γκαλερί αν θέλουμε να εγκαταστήσουμε ένα σύστημα συναγερμού.

Εισάγουμε επίσης την έννοια του **κόστους**: κάθε υποψήφια θέση φύλακα (ή σταθμού) έχει ένα κόστος. Η τοποθέτηση των σταθμών θέλουμε να μην ξεπερνά ένα **όριο κόστους**.

Μελετούμε επίσης πώς μπορούμε να εγκαταστήσουμε ένα δεδομένο αριθμό δικτύων (point to point communication) έτσι ώστε να καλύπτεται το μεγαλύτερο μέρος του πολυγώνου.

Για όλα τα παραπάνω προβλήματα δίνουμε πολυωνυμικού χρόνου προσεγγιστικούς αλγόριθμος που πετυχαίνουν σταθερούς προσεγγιστικούς παράγοντες. Αποδεικνύουμε επίσης αποτελέσματα μη-προσεγγισιμότητας για αυτά τα προβλήματα.

Μερικές από τις μεθόδους μας εφαρμόζονται σε γνωστά γραφοθεωρητικά προβλήματα για τα οποία προκύπτουν καινούρια αποτελέσματα.

Τέλος, ορίζουμε και μελετούμε ένα καινούριο πρόβλημα που προέκυψε κατά τη διάρκεια της μελέτης των προβλημάτων. Το ονομάζουμε πρόβλημα **γενικευμένης κάλυψης συνόλων** και φαίνεται αρκετά ενδιαφέρον και με πολλές εφαρμογές.

Τα ανοιχτά προβλήματα που προτείνουμε εκτός από θεωρητικό ενδιαφέρον, έχουν πολλές εφαρμογές και γι' αυτό πιστεύουμε ότι η έρευνα πάνω σ' αυτά θα ενταθεί τα επόμενα χρόνια.

Στις επόμενες ενότητες αυτού του κεφαλαίου δίνουμε:

- ορισμούς που θα χρειαστούμε (ενότητα 0.2),
- μια εισαγωγή στη θεωρία της προσεγγισιμότητας και στις έννοιες που θα χρησιμοποιήσουμε (ενότητα 0.3),
- γνωστά αποτελέσματα σχετικά με ελαχιστοποίηση αριθμού φυλάκων (ενότητα 0.4) και
- περίληψη των αποτελεσμάτων της εργασίας σχετικά με μεγιστοποίηση της κάλυψης (ενότητα 0.5)

0.2 Ορισμοί

0.2.1 Πολύγωνα και ορατότητα

Ορισμός 1 *Μια διατεταγμένη ακολουθία σημείων* $p_1, p_2, ..., p_n, n \geq 3$ *στο επίπεδο μαζί με τις ευθείες που ενώνουν το* p_i *με το* $p_{i+1}, i = 1, ..., n-1$, *ονομάζεται* **πολυγωνική αλυσίδα**. *Τα* $p_1, p_2, ..., p_n$ *καλούνται* **κορυφές** *της αλυσίδας ενώ τα* (p_i, p_{i+1}) **ακμές** *ή* **πλευρές**. *Η πολυγωνική αλυσίδα καλείται* **κλειστή** *αν* $p_1 = p_n$ *και* **ανοικτή** *αλλιώς.*

Ορισμός 2 *Μια πολυγωνική αλυσίδα καλείται* **απλή** *αν τα μόνα σημεία τομής μεταξύ των ακμών είναι τα κοινά άκρα των διαδοχικών ακμών. Μια απλή κλειστή πολυγωνική αλυσίδα* T *χωρίζει το επίπεδο σε δύο μέρη: το εσωτερικό του* T *και το εξωτερικό του* T.

Ορισμός 3 *Το εσωτερικό μιας απλής κλειστής πολυγωνικής αλυσίδας* T, *μαζί με το* T, *ονομάζεται* **απλό πολύγωνο χωρίς τρύπες**. *Το σύνορο* δT *το ονομάζουμε* **περίμετρο** *του πολυγώνου.*

Ορισμός 4 *Η ένωση ενός πεπερασμένου αριθμού από απλά πολύγωνα ονομάζεται πολύγωνο. Ένα πολύγωνο* T *ονομάζεται* **συνδεδεμένο** *αν για κάθε δύο σημεία του* T *υπάρχει μια πολυγωνική αλυσίδα που ανήκει στο* T *και τα ενώνει, Ένα συνδεδεμένο πολύγωνο* T *ονομάζεται* **απλά συνδεδεμένο** *αν κάθε πολυγωνική αλυσίδα που ενώνει δύο σημεία της περιμέτρου του* T *και δεν περνά από κανένα άλλο σημείο της περιμέτρου χωρίζει το* T. *Ένα συνδεδεμένο πολύγωνο που δεν είναι απλά συνδεδεμένο, ονομάζεται* **απλό πολύγωνο με τρύπες**.

Για απλότητα θα χρησιμοποιούμε τον όρο πολύγωνο για ένα συνδεδεμένο πολύγωνο με ή χωρίς τρύπες.

Ορισμός 5 *Έστω* T *ένα πολύγωνο και έστω* a, b *σημεία του* T. *Λέμε ότι τα σημεία* a, b *είναι* **ορατά** *το ένα στο άλλο, αν η ευθεία γραμμή που τα ενώνει βρίσκεται ολόκληρη μέσα στο* T. *Λέμε επίσης πως το σημείο* a **βλέπει** *το* b, *το* a *είναι* **ορατό** *από το* b *ή τα σημεία* a, b **φαίνονται** *μεταξύ τους. Ένα σύνολο* S *από σημεία του* T *είναι* **ορατό** *από ένα σημείο* a *αν κάθε σημείο του* S *είναι ορατό από το* a.

Ορισμός 6 *Έστω* T *ένα πολύγωνο,* a, b *σημεία του* T, S *και* Q *σύνολα σημείων του* T. *Λέμε ότι το* a **περιφρουρεί** *ή* **βλέπει** *ή* **καλύπτει** *το* S *αν κάθε σημείο του* S *φαίνεται από το* a. *Λέμε ότι το* S **περιφρουρεί** *ή* **καλύπτει** *το* Q *αν για κάθε σημείο του* Q *υπάρχει ένα τουλάχιστον σημείο του* S *που το βλέπει.*

Ορισμός 7 *Λέμε ότι ένα σημείο* a **επιβλέπει** *ή* **καλύπτει μερικώς** *το σύνολο* S *αν υπάρχει τουλάχιστον ένα σημείο του* S *που φαίνεται από το* a. *Λέμε ότι το* S *επιβλέπει ή καλύπτει μερικώς το* Q *εάν υπάρχει τουλάχιστον ένα σημείο του* Q *που φαίνεται από τουλάχιστον ένα σημείο του* S.

Η έννοια της περιφρούρησης είναι συμμετρική για δύο σημεία αλλά όχι για σύνολα σημείων. Ένα σύνολο σημείων S μπορεί να βλέπει το σύνολο Q αλλά μπορεί να μην φαίνονται όλα τα σημεία του S από το Q.

Η έννοια της επίβλεψης είναι συμμετρική. Αν ένα σύνολο σημείων S επιβλέπει το Q τότε και το S επιβλέπεται από το Q.

Ορισμός 8 *Έστω ένα πολύγωνο P. Καλούμε γράφο ορατότητας $V_G(P)$ το γράφημα που έχει έναν κόμβο για κάθε κορυφή του πολυγώνου και ακμή ανάμεσα σε δύο κόμβους, αν και μόνο αν οι αντίστοιχες κορυφές του πολυγώνου φαίνονται μεταξύ τους.*

Το πρόβλημα της αναγνώρισης ενός γράφου ορατότητας είναι στο PSPACE ([19]). Για λεπτομέρειες σχετικά με την αναπαράσταση πολυγώνων με γράφους δες [42].

0.2.2 Προβλήματα τοποθέτησης φυλάκων (ή σταθμών)

Ορισμός 9 *Έστω ένα πολύγωνο P χωρίς τρύπες. Ο στόχος του προβλήματος* MINIMUM VERTEX GUARD *είναι να τοποθετήσουμε στις κορυφές του πολυγώνου έναν ελάχιστο αριθμό φυλάκων έτσι ώστε το εσωτερικό και η περίμετρος του πολυγώνου να είναι ορατά από τους φύλακες.*

Εάν η τοποθέτηση των φυλάκων γίνεται στις πλευρές του πολυγώνου (ένας φύλακας καταλαμβάνει μια ολόκληρη πλευρά) τότε το πρόβλημα καλείται MINIMUM EDGE GUARD. Εάν η τοποθέτηση των φυλάκων γίνεται στο εσωτερικό του πολυγώνου, τότε το πρόβλημα καλείται MINIMUM POINT GUARD.

Τα παραπάνω προβλήματα μπορούν να οριστούν με σκοπό μόνο η περίμετρος του πολυγώνου να είναι ορατή από τους φύλακες. Σ' αυτήν την περίπτωση τα προβλήματα ονομάζονται αντίστοιχα MINIMUM VERTEX GUARD (BOUNDARY RESTRICTED), MINIMUM EDGE GUARD (BOUNDARY RESTRICTED) και MINIMUM POINT GUARD (BOUNDARY RESTRICTED).

Ορισμός 10 *Δίνεται ένα πολύγωνο P χωρίς τρύπες και ένας ακέραιος $k > 0$. Έστω $L(b)$ το ευκλείδειο μήκος ενός τμήματος b. Ο στόχος του προβλήματος* MAXIMUM LENGTH VERTEX GUARD *είναι να τοποθετήσουμε k φύλακες (σταθμούς), στις κορυφές του πολυγώνου με τέτοιο τρόπο ώστε το ευκλείδειο μήκος του μέρους της περιμέτρου του P που είναι ορατό από τους φύλακες να είναι μέγιστο.*

Ομοίως όπως και παραπάνω, αν η τοποθέτηση των φυλάκων γίνεται στις πλευρές το πρόβλημα καλείται MAXIMUM LENGTH EDGE GUARD, ενώ αν γίνεται στο εσωτερικό του πολυγώνου καλείται MAXIMUM LENGTH POINT GUARD.

Ορισμός 11 *Έστω ένα πολύγωνο P χωρίς τρύπες και ένας ακέραιος $k > 0$. Ο στόχος του προβλήματος* MAXIMUM AREA VERTEX GUARD *είναι να τοποθετήσουμε k φύλακες στις κορυφές του πολυγώνου έτσι ώστε το εσωτερικό τμήμα του πολυγώνου που είναι ορατό από τους φύλακες να είναι μέγιστο.*

Αντίστοιχα ορίζονται τα προβλήματα Maximum Area Edge Guard και Maximum Area Point Guard για τοποθέτηση φυλάκων σε πλευρές και στο εσωτερικό του πολυγώνου.

Ορισμός 12 *Έστω ένα πολύγωνο P χωρίς τρύπες και ένας ακέραιος $k > 0$. Η περίμετρος του πολυγώνου είναι χωρισμένη σε ξένα μεταξύ τους διαστήματα στα οποία έχουν αποδοθεί μη-αρνητικά βάρη. Ο στόχος του προβλήματος* Maximum Value Vertex Guard *είναι να τοποθετηθούν το πολύ k φύλακες (σταθμοί) στις κορυφές του πολυγώνου έτσι ώστε το συνολικό βάρος των τμημάτων που φαίνονται να είναι μέγιστο.*

Αντίστοιχα ορίζονται τα προβλήματα Maximum Value Edge Guard και Maximum Value Point Guard για τοποθέτηση φυλάκων σε πλευρές και στο εσωτερικό του πολυγώνου.

Τα τρία παραπάνω προβλήματα μπορούν να οριστούν και με μια διαφορετική έννοια ορατότητας. Όταν ο στόχος είναι οι φύλακες να επιβλέπουν (watch) τα τμήματα και όχι να τα περιφρουρούν τότε έχουμε αντίστοιχα τα προβλήματα Maximum Value Vertex Guard (Watching Version), Maximum Value Edge Guard (Watching Version) και Maximum Value Point Guard (Watching Version).

Ορισμός 13 *Έστω ένα πολύγωνο P χωρίς τρύπες, ένας ακέραιος $k > 0$ και ένα σύνολο D από πίνακες. Κάθε πίνακας $d \in D$ έχει ένα μήκος $l(d)$ και μια αξία $c(d)$. Ο σκοπός του προβλήματος* Maximum Value Vertex Guard with Painting Placement *είναι να τοποθετηθούν k φύλακες στις κορυφές του πολυγώνου και πίνακες στις πλευρές του πολυγώνου (δηλαδή στους τοίχους της γκαλερί) έτσι ώστε η συνολική αξία των πινάκων που φαίνονται να είναι μέγιστη.*

Αντίστοιχα ορίζονται τα προβλήματα Maximum Value Edge Guard with Painting Placement και Maximum Value Point Guard with Painting Placement για τοποθέτηση φυλάκων σε πλευρές και στο εσωτερικό του πολυγώνου.

Στα τρία παραπάνω προβλήματα η λέξη 'πίνακες' στους ορισμούς δεν περιορίζει φυσικά τις εφαρμογές αλλά γίνεται περισσότερο για ιστορικούς λόγους (τα προβλήματα είναι παραλλαγές του προβλήματος του μουσείου). Πολλές φορές, αντί για φύλακες και πίνακες χρησιμοποιούμε τις λέξεις **'πομποδέκτες'** και **'αναμεταδότες'** αντίστοιχα δανεισμένες από την ορολογία των δικτύων.

Σε όλα τα παραπάνω προβλήματα μπορούμε να έχουμε αντί του ακεραίου k ένα διαθέσιμο ποσό B και σε κάθε υποψήφια θέση φύλακα ένα κόστος το οποίο πρέπει να πληρώσουμε προκειμένου να τοποθετήσουμε εκεί φύλακα. Έτσι λοιπόν τα αντίστοιχα προβλήματα που προκύπτουν είναι τα εξής: Budgeted Maximum Length Vertex Guard, Budgeted Maximum Length Edge Guard, Budgeted Maximum Length Point Guard, Budgeted Maximum Area Vertex Guard, Budgeted Maximum Area Edge Guard, Budgeted Maximum Area Point Guard, Budgeted Maximum Value Vertex Guard, Budgeted Maximum Value Edge Guard, Budgeted Maximum Value Point

Guard, Budgeted Maximum Value Vertex Guard (watching version), Budgeted Maximum Value Edge Guard (watching version), Budgeted Maximum Value Point Guard (watching version), Budgeted Maximum Value Vertex Guard with Painting Placement, Budgeted Maximum Value Edge Guard with Painting Placement, Budgeted Maximum Value Point Guard with Painting Placement.

Τέλος όλα τα παραπάνω προβλήματα μπορούν να οριστούν σε πολύγωνα με τρύπες.

0.2.3 Αλλα προβλήματα

Ορισμός 14 *Έστω ένα πολύγωνο P χωρίς τρύπες. Ο στόχος του προβλήματος* Minimum Convex Decomposition *είναι ο κατακερματισμός του πολυγώνου στον ελάχιστο αριθμό από μη-επικαλυπτόμενα κυρτά πολύγωνα.*

Ορισμός 15 *Έστω ένα πολύγωνο P χωρίς τρύπες. Ο στόχος του προβλήματος* Minimum Convex Cover *είναι ο η κάλυψη του πολυγώνου από έναν ελάχιστο αριθμό πιθανά επικαλυπτόμενων κυρτών πολυγώνων που βρίσκονται μέσα στο P.*

Ορισμός 16 *Έστω ένα πολύγωνο P χωρίς τρύπες. Ο στόχος του προβλήματος* Minimum Clique Partition On Visibility Graph *είναι ο χωρισμός του γράφου ορατότητας του P σε έναν ελάχιστο αριθμό από κλίκες.*

Ορισμός 17 *Έστω ένα πολύγωνο P χωρίς τρύπες με βάρη στις κορυφές του. Ο στόχος του προβλήματος* Maximum Weighted Clique On Visibility Graph *είναι η εύρεση μιας κλίκας μέγιστου βάρους στον visibility γράφο του P.*

Ορισμός 18 *Έστω πολύγωνο P χωρίς τρύπες, με βάρη στις κορυφές του και ένας ακέραιος $k > 0$. Ο σκοπός του προβλήματος* Maximum Weight In k Cliques *είναι να βρούμε k κλίκες στο γράφο ορατότητας $V_G(P)$ έτσι ώστε το βάρος των κορυφών που ανήκουν στις κλίκες να είναι μέγιστο.*

Ορισμός 19 *Έστω U ένα σύνολο με στοιχεία και C μια συλλογή από υποσύνολα του U τέτοια ώστε $\cup_{S_i \in C} S_i = U$. Ο στόχος του προβλήματος* Set Cover *είναι να βρεθεί μια συλλογή $C' \subset C$ με ελάχιστο πληθικό αριθμό τέτοια ώστε $\cup_{S_i \in C'} S_i = U$*

Ορισμός 20 *Δίνεται ένα σύνολο U το οποίο περιέχει στοιχεία με βάρη, ένας ακέραιος $k > 0$ και μια συλλογή C από υποσύνολα του U. Ο στόχος του προβλήματος* Maximum Coverage *είναι να βρεθούν k σύνολα $S_i \in C$ έτσι ώστε το $\cup S_i$ να έχει μέγιστο συνολικό βάρος.*

Ορισμός 21 *Δίνεται ένα σύνολο U το οποίο περιέχει στοιχεία με βάρη, ένας ακέραιος $B > 0$ και μια συλλογή C από υποσύνολα του U σε καθένα από τα οποία έχει αντιστοιχηθεί ένα κόστος. Ο στόχος του προβλήματος* Budgeted Maximum Coverage *είναι να βρεθεί μια συλλογή από σύνολα $C' \subset C$ τέτοια ώστε το συνολικό κόστος των συνόλων της C' να μην υπερβαίνει τον προϋπολογισμό B και το συνολικό βάρος των στοιχείων που περιέχονται στα σύνολα της C' να είναι μέγιστο.*

Ορισμός 22 *Έστω σακίδιο με χωρητικότητα M και ένα σύνολο S από n αντικείμενα. Κάθε αντικείμενο i έχει ένα μέγεθος $s(i)$ και μια αξία $p(i)$. Ο σκοπός του προβλήματος* Knapsack *είναι η εύρεση ενός συνόλου $U \subset S$ με μέγιστη αξία τέτοιο ώστε τα αντικείμενά του να χωρούν στο σακίδιο.*

Ορισμός 23 *Έστω ένα ζεύγος (B, S), όπου B είναι ένα σύνολο από m σάκους και S ένα σύνολο από n αντικείμενα. Κάθε σάκος $j \in B$ έχει μια χωρητικότητα $w(j)$ και κάθε αντικείμενο i έχει ένα μέγεθος $s(i)$ και μια αξία $p(i)$. Ο σκοπός του προβλήματος* Multiple Knapsack *είναι η εύρεση ενός συνόλου $U \subset S$ με μέγιστη αξία τέτοιο ώστε τα αντικείμενά του να χωρούν στους σάκους του B.*

Ορισμός 24 *Έστω U μια συλλογή αντικειμένων τα οποία έχουν αξία. Κάθε αντικείμενο είναι ένα σύνολο δομικών στοιχείων. Δίνονται επίσης μια συλλογή C από σύνολα τα οποία περιέχουν δομικά στοιχεία και ένας ακέραιος $k > 0$. Ο σκοπός του προβλήματος* Generalized Set Cover *είναι να βρεθούν k σύνολα $S_i \in C$ έτσι ώστε η αξία των αντικειμένων που καλύπτονται να είναι μέγιστη. Ένα αντικείμενο καλύπτεται όταν έχουμε πάρει όλα τα δομικά στοιχεία του.*

Ορισμός 25 *Έστω Φ μια λογική έκφραση σε κανονική συζευκτική μορφή (CNF). Το πρόβλημα απόφασης* SAT *είναι: υπάρχει απονομή αλήθειας που να ικανοποιεί την Φ;*

Ορισμός 26 *Έστω Φ μια λογική έκφραση σε CNF, όπου κάθε πρόταση (clause) αποτελείται από το πολύ 3 literals. Ο σκοπός του προβλήματος* Max3Sat *είναι να βρεθεί μια απονομή αλήθειας στις μεταβλητές της Φ έτσι ώστε ο αριθμός των προτάσεων που ικανοποιούνται να είναι μέγιστος.*

Ορισμός 27 *Έστω G ένα γράφημα. Ο σκοπός του προβλήματος* Max Clique *είναι η εύρεση μιας μέγιστης κλίκας του G.*

Ορισμός 28 *Έστω Φ μια λογική έκφραση σε CNF, όπου κάθε clause αποτελείται από το πολύ 3 literals και η κάθε μεταβλητή εμφανίζεται το πολύ σε 5 clauses. Ο σκοπός του προβλήματος* Max-5-Occurence-3-Sat *είναι να βρεθεί μια απονομή αλήθειας στις μεταβλητές της Φ έτσι ώστε ο αριθμός των clauses που ικανοποιούνται να είναι μέγιστος.*

0.3 Προσεγγισιμότητα

0.3.1 Προσεγγιστικοί αλγόριθμοι

Θα κάνουμε μια εισαγωγή στη θεωρία της προσεγγισιμότητας περιγράφοντας έννοιες που χρησιμοποιούμε στα επόμενα κεφάλαια και αφορούν τους προσεγγιστικούς αλγόριθμους, τις **αναγωγές διατήρησης χάσματος** (**gap-preserving**) και τις κλάσεις προσέγγισης. Λεπτομέρειες για αυτές τις έννοιες υπάρχουν στα [26, 2].

Έστω Π ένα πρόβλημα βελτιστοποίησης και έστω I ένα στιγμιότυπο του Π. Έστω $\Sigma(I)$ το σύνολο των εφικτών λύσεων του $\Pi(I)$. Έστω $OPT(I)$ το μέγεθος μιας βέλτιστης λύσης του $\Pi(I)$. Εάν το Π είναι πρόβλημα μεγιστοποίησης:

$$OPT(I) = max_{SOL \in \Sigma(I)} |SOL|$$

Εάν το Π είναι πρόβλημα ελαχιστοποίησης:

$$OPT(I) = min_{SOL \in \Sigma(I)} |SOL|$$

Για παράδειγμα, εάν το Π είναι το πρόβλημα Max3Sat, τότε ένα στιγμιότυπο I του προβλήματος θα είναι m clauses, το καθένα από τα οποία έχει το πολύ 3 literals. Εάν το στιγμιότυπο έχει n μεταβλητές τότε με κάθε απονομή αλήθειας στις n μεταβλητές έχουμε ένα υποσύνολο SOL από clauses που αληθεύουν. Το σύνολο των εφικτών λύσεων $\Sigma(I)$ είναι το σύνολο από όλα τα δυνατά SOL (κάθε SOL προκύπτει από μια διαφορετική απονομή αλήθειας). ᾿ρα λοιπόν σε αυτή την περίπτωση το OPT(I) είναι ο μέγιστος αριθμός από clauses που αληθεύουν με κάποια απονομή αλήθειας.

Έστω Α ένας **προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου** για το πρόβλημα Π. Ο αλγόριθμος Α θα πρέπει για κάθε στιγμιότυπο I να επιστρέφει μια λύση $SOL(A(I))$, η οποία φυσικά ανήκει στο $\Sigma(I)$. Λέμε ότι ο προσεγγιστικός αλγόριθμος Α πετυχαίνει προσέγγιση $R_A(I)$ για το πρόβλημα Π αν για όλα τα στιγμιότυπα I του Π ισχύει:

$$R_{A(I)} \geq max(\frac{OPT(I)}{|SOL(A(I))|}, \frac{|SOL(A(I))|}{OPT(I)})$$

Ο παράγοντας προσέγγισης $R_A(I)$, ορισμένος όπως παραπάνω, είναι πάντα ≥ 1 ανεξάρτητα από το εάν το Π είναι πρόβλημα μεγιστοποίησης ή ελαχιστοποίησης.

Λέμε ότι ένα πρόβλημα βελτιστοποίησης Π δεν μπορεί να προσεγγιστεί με παράγοντα προσέγγισης $R(I)$ (ή μικρότερο), εάν για κάθε προσεγγιστικό αλγόριθμο πολυωνυμικού χρόνου Α για το Π, υπάρχει κάποιο στιγμιότυπο I του Π τέτοιο ώστε:

$$R(I) < R_A(I)$$

Τέτοια αποτελέσματα μη - προσεγγισιμότητας τα αποδεικνύουμε συνήθως θεωρώντας ότι $P <> NP$. Μερικά αποτελέσματα μη - προσεγγισιμότητας για να αποδειχθούν χρειάζονται κάπως πιο δυνατές υποθέσεις όπως $NP <> TIME(n^{O(\log \log n)})$ (quasi NP-hardness) οι οποίες πιστεύεται ότι είναι αλήθεια.

0.3.2 Προβλήματα με εγγύηση

Τα προβλήματα με εγγύηση (promise ή canonical προβλήματα) συνδέουν τα προβλήματα βελτιστοποίησης με τα προβλήματα απόφασης. Σε ένα πρόβλημα με εγγύηση, δίνεται ένα σύνολο στιγμιοτύπων ενός προβλήματος βελτιστοποίησης καθώς και μια εγγύηση ότι η βέλτιστη λύση $OPT(I)$ είναι τουλάχιστον ίση με $U(I)$ ή μικρότερη από $L(I)$. Δηλαδή το $OPT(I)$ δε μπορεί να είναι ανάμεσα στα $L(I)$ και $U(I)$.

$$OPT(I) \geq U(I) \tag{1}$$

ή

$$OPT(I) < L(I) \tag{2}$$

Τα $U(I)$ και $L(I)$ εξαρτώνται από το στιγμιότυπο I και προφανώς $L(I) < U(I)$.

Ο στόχος του προβλήματος με εγγύηση είναι να αποφασίσουμε ποια από τις δύο περιπτώσεις ισχύει.

Για παράδειγμα έστω Π είναι το πρόβλημα MAX3SAT. Έστω I ένα στιγμιότυπο του MAX3SAT με m clauses. Αν $\epsilon > 0$ είναι μια μικρή σταθερά και $U(I) = m$ και $L(I) = (1-\epsilon)m$, τότε το πρόβλημα με εγγύηση MAX3SAT είναι να αποφασίσουμε αν:

$$OPT(I) \geq m$$

ή

$$OPT(I) < (1-\epsilon)m$$

Θεώρημα 0.1 *([2]) Το πρόβλημα με εγγύηση* MAX3SAT *με παραμέτρους $U(I) = m$ και $L(I) = (1-\epsilon)m$ για κάποιο $\epsilon > 0$, όπου m είναι ο αριθμός των clauses του στιγμιότυπου I, είναι NP-hard να αποφασιστεί.*

Τονίζουμε ότι ένα πρόβλημα με εγγύηση περιλαμβάνει μόνο εκείνα τα στιγμιότυπα του αντίστοιχου προβλήματος βελτιστοποίησης, για τα οποία ισχύουν οι ιδιότητες 1 ή 2. Για την ακρίβεια τα στιγμιότυπα του προβλήματος με εγγύηση για τα οποία ισχύει η ιδιότητα 1 είναι μετασχηματισμοί στιγμιοτύπων του προβλήματος απόφασης SAT με απάντηση 'ΝΑΙ'. Ενώ τα στιγμιότυπα του προβλήματος με εγγύηση για τα οποία ισχύει η ιδιότητα 2 είναι μετασχηματισμοί στιγμιοτύπων του προβλήματος απόφασης SAT με απάντηση 'ΟΧΙ'.

Γενικώς δηλαδή τα προβλήματα με εγγύηση είναι προβλήματα της μορφής που περιγράψαμε για τα οποία έχει αποδειχθεί ότι είναι NP-hard να αποφασισθούν. Δηλαδή, εκτός αν $P = NP$, δεν υπάρχει πολυωνυμικός αλγόριθμος που να αποφασίζει αν $OPT(I) \geq U(I)$ ή $OPT(I) < L(I)$. Αυτό όμως σημαίνει ότι δε μπορεί να υπάρχει προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου που να δίνει παράγοντα προσέγγισης

$$R(I) = \frac{U(I)}{L(I)}$$

Ας δούμε γιατί:

Έστω ότι το Π είναι πρόβλημα μεγιστοποίησης. Έστω A ο προσεγγιστικός αλγόριθμος με $R_{A(I)} = R$.

$$R_{A(I)} \geq \frac{OPT(I)}{SOL(A(I))} \rightarrow OPT(I) \leq \frac{U(I)}{L(I)} SOL$$

Ισχύει επίσης:

$$OPT(I) \geq U(I)$$

ή
$$OPT(I) < L(I)$$

Εφαρμόζουμε τον προσεγγιστικό αλγόριθμο A στο $\Pi(I)$.

1. Έστω $SOL(A(I)) < L(I)$
$$\rightarrow OPT(I) < U(I) \rightarrow OPT(I) < L(I)$$

2. Έστω $SOL(A(I)) \geq L(I)$
$$\rightarrow OPT(I) \geq L(I) \rightarrow OPT(I) \geq U(I)$$

Δηλαδή αν υπήρχε προσεγγιστικός αλγόριθμος με παράγοντα προσέγγισης $R(I) = \frac{U(I)}{L(I)}$, το NP-hard πρόβλημα με εγγύηση Π θα μπορούσε να αποφασισθεί. Συνεπώς δεν υπάρχει προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου με παράγοντα προσέγγισης
$$R(I) = \frac{U(I)}{L(I)} = \frac{1}{1-\epsilon}$$
για το πρόβλημα Max3Sat αφού αν υπήρχε αυτός ο αλγόριθμος θα μπορούσε να χρησιμοποιηθεί για να αποφασισθεί το παραπάνω πρόβλημα με εγγύηση.

Αλλα αποτελέσματα για promise προβλήματα είναι:

Θεώρημα 0.2 *([2]) Το πρόβλημα με εγγύηση* Max-5-Occurence-3-Sat *με παραμέτρους $U(I) = m$ και $L(I) = (1-\epsilon)m$ για κάποιο $\epsilon > 0$, όπου m είναι ο αριθμός των clauses του στιγμιότυπου I, είναι NP-hard να αποφασιστεί.*

Θεώρημα 0.3 *([20]) Το promise πρόβλημα* Set Cover *με παραμέτρους $U(I) = k(1-\epsilon)\ln n$ και $L(I) = k$ για οποιοδήποτε $\epsilon > 0$, είναι NP-hard να αποφασιστεί.*

Δηλαδή δεν υπάρχει προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου με παράγοντα προσέγγισης
$$R(I) = \frac{U(I)}{L(I)} = (1-\epsilon)\ln n$$
για το πρόβλημα Set Cover αφού αν υπήρχε αυτός ο αλγόριθμος θα μπορούσε να χρησιμοποιηθεί για να αποφασισθεί το παραπάνω promise πρόβλημα.

Θεώρημα 0.4 *([23]) Το promise πρόβλημα* Max Clique *με παραμέτρους $U(I) = k$ και $L(I) = \frac{k}{n^{\frac{1}{2}-\epsilon}}$ για κάποιο $\epsilon > 0$, είναι NP-hard να αποφασιστεί.*

Αυτό σημαίνει ότι δεν υπάρχει προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου με παράγοντα προσέγγισης
$$R(I) = \frac{U(I)}{L(I)} = n^{\frac{1}{2}-\epsilon}$$
για το πρόβλημα Max Clique αφού αν υπήρχε αυτός ο αλγόριθμος θα μπορούσε να χρησιμοποιηθεί για να αποφασισθεί το παραπάνω promise πρόβλημα.

Τα παραπάνω θεωρήματα αποδείχθηκαν χρησιμοποιώντας το PCP θεώρημα. Για περισσότερες λεπτομέρειες δες [1, 2].

0.3.3 Αναγωγές διατήρησης χάσματος

Η αναγωγή διατήρησης χάσματος (gap - preserving reduction) προτάθηκε στο [2] και χρησιμοποιείται για την απόδειξη αποτελεσμάτων μη - προσέγγισης.

Έστω Π και Π′ δύο προβλήματα βελτιστοποίησης. Μια αναγωγή διατήρησης χάσματος είναι ένας αλγόριθμος πολυωνυμικού χρόνου ο οποίος μετασχηματίζει ένα στιγμιότυπο I του προβλήματος με εγγύηση Π με πάνω όριο $U(I)$ και κάτω όριο $L(I)$ σε ένα στιγμιότυπο I' του προβλήματος Π′ με πάνω όριο $U'(I')$ και κάτω όριο $L'(I')$. Η αναγωγή γίνεται με τέτοιο τρόπο ώστε να ισχύει:

$$OPT(I) \geq U(I) \rightarrow OPT(I') \geq U'(I') \tag{3}$$

$$OPT(I) < L(I) \rightarrow OPT(I') < L'(I') \tag{4}$$

Έτσι αν το promise πρόβλημα Π είναι NP-hard να αποφασισθεί τότε και το Π′ είναι NP-hard να αποφασισθεί. Ας δούμε γιατί:

Ας υποθέσουμε ότι υπάρχει ένας αλγόριθμος πολυωνυμικού χρόνου που αποφασίζει το Π′ με παραμέτρους $U'(I')$ και $L'(I')$. Ας υποθέσουμε χωρίς βλάβη της γενικότητας ότι ο αλγόριθμος αποφασίζει ότι ισχύει $OPT(I') \geq U'(I')$. Συνεπώς το $OPT(I') < L'(I')$ είναι ψευδές και άρα με αντιθετοαντιστροφή το $OPT(I) < L(I)$ είναι ψευδές, συνεπώς $OPT(I) \geq U(I)$, δηλαδή μπορούμε να αποφασίσουμε για το Π. Η συνέπεια της αναγωγής διατήρησης χάσματος είναι ότι δε μπορεί να υπάρχει πολυωνυμικού χρόνου προσεγγιστικός αλγόριθμος που να πετυχαίνει προσέγγιση

$$R'(I') = \frac{U'(I')}{L'(I')}$$

για το πρόβλημα Π′. **Παρατηρήσεις**:

- Όπως στις περισσότερες γνωστές αναγωγές, έτσι και στην αναγωγή διατήρησης χάσματος υπάρχει αντιστοίχηση στις λύσεις. Δηλαδή μια λύση του I' μεγέθους τουλάχιστον c' μπορεί να μετασχηματισθεί σε πολυωνυμικό χρόνο σε μια λύση του I μεγέθους τουλάχιστον c.

- Τα προβλήματα Π και Π′ μπορεί να είναι προβλήματα μεγιστοποίησης ή ελαχιστοποίησης με οποιοδήποτε συνδυασμό.

- Η ιδιότητες 3 και 4 ισχύουν για ένα υποσύνολο των δυνατών στιγμιοτύπων του προβλήματος Π. Συνεπώς η gap-preserving αναγωγή για τα στιγμιότυπα I του Π για τα οποία ισχύει $L(I) \leq OPT(I) < U(I)$ μπορεί να συμπεριφέρεται με οποιοδήποτε τρόπο. Για αυτό το λόγο, ένας προσεγγιστικός αλγόριθμος για το Π′ και μια gap-preserving αναγωγή από το Π στο Π′ **δεν** σηματοδοτούν την ύπαρξη προσεγγιστικού αλγόριθμου για το Π. Με αυτή την έννοια η gap-preserving αναγωγή είναι πιο αδύνατη από την **L - αναγωγή** ([43]) η οποία ορίζεται για όλα τα στιγμιότυπα του Π. Δηλαδή μια L - αναγωγή από το Π στο Π′ σε συνδυασμό με έναν προσεγγιστικό αλγόριθμο του Π′ μας εξασφαλίζει έναν προσεγγιστικό αλγόριθμο για το Π. Η gap-preserving αναγωγή λοιπόν είναι συνήθως πιο εύκολη από την L-αναγωγή και χρησιμοποιείται για να δείχνουμε αρνητικά αποτελέσματα προσεγγισιμότητας. Μια L-αναγωγή είναι και gap - preserving, το αντίθετο όμως δεν ισχύει. Για μια περιγραφή της L - αναγωγής δες [43].

- Η αναγωγή διατήρησης χάσματος μπορεί να παράγει ένα μικρότερο ή μεγαλύτερο χάσμα στο Π' από ότι το χάσμα του Π. Δηλαδή δεν διατηρεί αναγκαστικά το μέγεθος του χάσματος, διατηρεί όμως χάσμα.

0.3.4 Κλάσεις Προσέγγισης

Τα υπολογιστικά προβλήματα ταξινομούνται σε κλάσεις ανάλογα με το χρόνο που χρειάζεται κάποιος αλγόριθμος για να τα επιλύσει.

Στις κλάσεις προσέγγισης, ταξινομούμε προβλήματα βελτιστοποίησης των οποίων τα προβλήματα απόφασης είναι NP-hard να επιλυθούν. Η ταξινόμηση γίνεται με κριτήριο τον παράγοντα προσέγγισης που έχει κάποιος πολυωνυμικός αλγόριθμος που τα προσεγγίζει. Η ιεραρχία αυτή είναι σχετικά καινούρια και είναι αποτέλεσμα περισσότερο των αποδεικτικών μεθόδων που έχουμε στη διάθεσή μας και αντανακλά λιγότερο ίσως την πραγματικότητα. Η ιεραρχία που έχει προκύψει ως τώρα φαίνεται στο σχήμα 1.

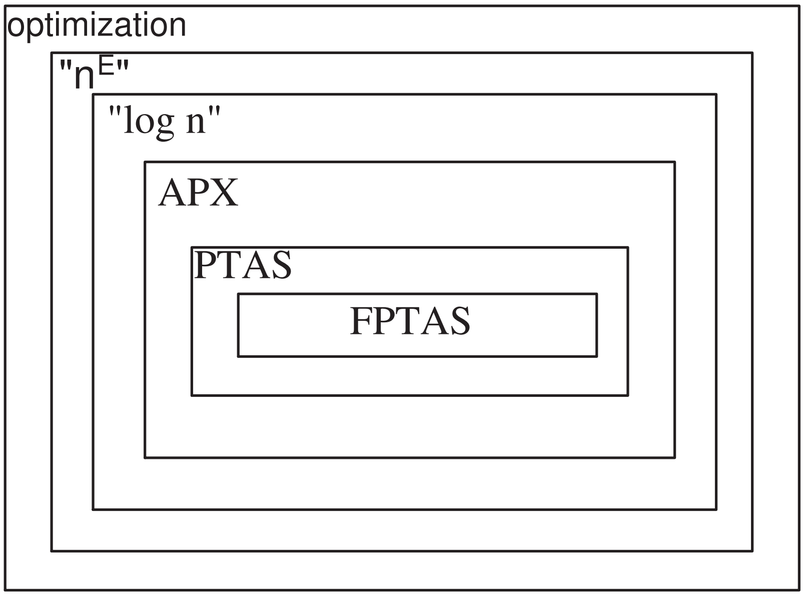
Ένα πρόβλημα ανήκει στην κλάση n^{ϵ} αν υπάρχει $\epsilon > 0$ έτσι ώστε το πρόβλημα να μπορεί να προσεγγιστεί από έναν αλγόριθμο πολυωνυμικού χρόνου με παράγοντα προσέγγισης $O(n^{\epsilon})$, όπου n είναι το μέγεθος του στιγμιότυπου του προβλήματος. Λέμε ότι ένα πρόβλημα βελτιστοποίησης είναι n^{ϵ}-hard, αν υπάρχει $\epsilon > 0$ για το οποίο δεν υπάρχει πολυωνυμικού χρόνου προσεγγιστικός αλγόριθμος με προσέγγιση $R(I) \in \Theta(n^{\epsilon})$. Λέμε ότι το πρόβλημα βελτιστοποίησης είναι n^{ϵ} - complete αν ανήκει στην κλάση n^{ϵ} και είναι n^{ϵ} - hard. Ένα τέτοιο πρόβλημα είναι το Max Clique.

Η κλάση $\log n$ είναι υποκλάση της n^{ϵ} και περιλαμβάνει εκείνα τα προβλήματα για τα οποία υπάρχει αλγόριθμος πολυωνυμικού χρόνου που πετυχαίνει προσέγγιση με παράγοντα $O(\log n)$.

Λέμε ότι ένα πρόβλημα βελτιστοποίησης είναι $\log n$ - hard αν δεν υπάρχει πολυωνυμικού χρόνου προσεγγιστικός αλγόριθμος που να πετυχαίνει παράγοντα προσέγγισης $R(I) \in \Theta(\log n)$. Ένα πρόβλημα βελτιστοποίησης είναι $\log n$ - complete αν ανήκει στην κλάση $\log n$ και είναι $\log n$ - hard. Ένα τέτοιο πρόβλημα είναι το Set Cover.

Ένα πρόβλημα ανήκει στην κλάση APX (η οποία είναι υποκλάση της $\log n$) αν υπάρχει πολυωνυμικού χρόνου προσεγγιστικός αλγόριθμος που πετυχαίνει παράγοντα $1+\epsilon$ για κάποιο $\epsilon > 0$. Λέμε ότι ένα πρόβλημα είναι APX-hard αν υπάρχει $\epsilon > 0$ τέτοιο ώστε κανένας πολυωνυμικού χρόνου προσεγγιστικός αλγόριθμος δεν μπορεί να πετύχει παράγοντα προσέγγισης $1 + \epsilon$. Ένα πρόβλημα που ανήκει στην κλάση APX και είναι APX-hard, ονομάζεται APX-complete. Ένα τέτοιο πρόβλημα είναι το Max3Sat.

Η κλάση PTAS είναι υποκλάση της APX και περιέχει εκείνα τα προβλήματα για τα οποία υπάρχει πολυωνυμικό σχήμα. Δηλαδή $\forall \epsilon > 0$ υπάρχει πολυωνυμικού χρόνου προσεγγιστικός αλγόριθμος που πετυχαίνει παράγοντα προσέγγισης $1 + \epsilon$.



Σχήμα 1: Ταξινόμηση προβλημάτων σε κλάσεις προσέγγισης. Συμβολίζουμε με: “n^ϵ” την κλάση των προβλημάτων με $O(n^\epsilon)$ παράγοντα προσέγγισης, “$\log n$” την κλάση των προβλημάτων με $O(\log n)$ παράγοντα προσέγγισης, APX την κλάση των προβλημάτων με σταθερό παράγοντα προσέγγισης και PTAS και FPTAS τις κλάσεις των προβλημάτων με παράγοντα προσέγγισης οσοδήποτε κοντά στο 1.

Η χρονική πολυπλοκότητα του αλγορίθμου εξαρτάται εκθετικά από το $\frac{1}{\epsilon}$. Εάν η χρονική πολυπλοκότητα του αλγορίθμου εξαρτάται πολυωνυμικά από το $\frac{1}{\epsilon}$ τότε το ονομάζουμε FPTAS. Ένα πρόβλημα που ανήκει στην κλάση PTAS και δεν ανήκει στην κλάση FPTAS είναι το Multiple Knapsack ενώ το Knapsack ανήκει στην κλάση FPTAS.

Έχουν προταθεί και άλλες κλάσεις, όπως για παράδειγμα η κλάση $2^{\log^{1-\gamma} n}$ η οποία είναι ανάμεσα στην κλάση $\log n$ και στην κλάση n^ϵ ([2]). Επίσης στο [2] οι κλάσεις προτείνονται λίγο διαφορετικά. Για παράδειγμα σαν κλάση I ορίζεται η κλάση των προβλημάτων που είναι APX-hard. Όπως είπαμε και παραπάνω, οι κλάσεις που έχουν προκύψει ως τώρα είναι περισσότερο αποτέλεσμα των αποδεικτικών μεθόδων.

Αφήσαμε για το τέλος την περιγραφή της κλάσης MAXSNP. Η κλάση MAXSNP προτάθηκε στο [43] και περιλαμβάνει τα προβλήματα που ανήκουν στην κλάση APX. Ένα πρόβλημα είναι complete στην κλάση MAXSNP αν όλα τα προβλήματα της κλάσης ανάγονται με L-αναγωγές σ' αυτό.

Η κλάση MAXSNP και το πρόβλημα Max3Sat, το οποίο ήταν ιστορικά το πρώτο complete πρόβλημα της κλάσης έπαιξαν ρόλο στην ανακάλυψη του PCP θεωρήματος ([1]). Η απόδειξη του PCP θεωρήματος έκανε δυνατή την κατασκευή των προβλημάτων με εγγύηση.

Για να δείξουμε ότι ένα πρόβλημα δεν έχει προσεγγιστικό παράγοντα R, αρκεί να κατασκευάσουμε μια αναγωγή διατήρησης χάσματος από ένα πρόβλημα εγγύησης στο πρόβλημά μας και έπειτα να 'μετρήσουμε' το χάσμα.

Οι κλάσεις προσέγγισης και τα αποτελέσματα μη - προσεγγισιμότητας είναι απαραίτητα για να κρίνουμε αν ένας προσεγγιστικός αλγόριθμος πετυχαίνει την καλύτερη δυνατή προσέγγιση για κάποιο πρόβλημα.

0.4 Ελαχιστοποίηση φυλάκων - Γνωστά αποτελέσματα

Ένα από τα πιο γνωστά αποτελέσματα που έχουν σχέση με τη φύλαξη πολυγώνων είναι το Art Gallery Theorem το οποίο λέει ότι $\lfloor \frac{n}{3} \rfloor$ φύλακες αρκούν πάντα (κάποιες φορές μπορεί να είναι και αναγκαίοι) έτσι ώστε να περιφρουρείται ένα πολύγωνο με n κορυφές [9].

Το πρόβλημα περιφρούρησης ενός πολυγώνου χρησιμοποιώντας τον ελάχιστο αριθμό φυλάκων είναι NP-hard. Δηλαδή ανεξάρτητα από το εάν οι φύλακες επιτρέπεται να τοποθετηθούν στις κορυφές στις πλευρές ή στο εσωτερικό και ανεξάρτητα από το αν στόχος είναι η φρούρηση του εσωτερικού ή της περιμέτρου του πολυγώνου με ή χωρίς τρύπες, το πρόβλημα είναι NP-hard [32].

Προσεγγιστικοί αλγόριθμοι είναι γνωστοί [22] για τα προβλήματα Minimum Vertex Guard και Minimum Edge Guard για πολύγωνα (εσωτερικό ή περίμετρο) με ή χωρίς τρύπες με προσεγγιστικούς παράγοντες της τάξης $O(\log n)$. Τα προβλήματα Minimum Vertex Guard (Boundary Restricted), Minimum Edge Guard (Boundary Restricted) και Minimum Point Guard (Boundary Restricted) για πολύγωνα χωρίς τρύπες είναι APX-hard [32, 15, 16, 18].

Τα προβλήματα Minimum Vertex Guard (Boundary Restricted), Minimum Edge Guard (Boundary Restricted) και Minimum Point Guard (Boundary Restricted) για πολύγωνα με τρύπες είναι $\log n$-hard ([18]). Μάλιστα, ο καλύτερος γνωστός προσεγγιστικός αλγόριθμος για το πρόβλημα Minimum Point Guard (Boundary Restricted) για πολύγωνα με ή χωρίς τρύπες πετυχαίνει προσεγγιστικό παράγοντα $\Theta(n)$.

Στο [12] προτάθηκε το εξής πρόβλημα: έστω ένας αριθμός από σημειακούς θησαυρούς σε ένα πολύγωνο P. Ποιος είναι ο ελάχιστος αριθμός φυλάκων που απαιτούνται ώστε όλοι οι θησαυροί να φυλάσσονται. Στο [12] αποδείχθηκε ότι το πρόβλημα είναι NP-hard και προτάθηκαν κάποιοι ευριστικοί αλγόριθμοι.

Στο [5] αναθέτονται βάρη στους θησαυρούς και μελετούν την περίπτωση της τοποθέτησης **ενός** φύλακα έτσι ώστε το άθροισμα των βαρών των θησαυρών που φυλάσσονται να είναι μέγιστο. Survey [32, 41, 46, 48, 15, 22].

Στο [31] εξετάζεται η περιφρούρηση των πλευρών του πολυγώνου.

Το πρόβλημα Minimum Convex Cover είναι NP-hard ([11]). Μπορεί να προσεγγιστεί με παράγοντα $\log n$ ([15]) ενώ έχει αποδειχθεί ότι είναι APX-hard.

Το πρόβλημα κατακερματισμού ενός πολυγώνου χωρίς τρύπες σε ένα ελάχιστο αριθμό από μη-επικαλυπτόμενα κυρτά πολύγωνα (Minimum Convex Decomposi-

TION) είναι στο P ([6]). Το πρόβλημα γίνεται NP-hard μόνο αν το πολύγωνο έχει τρύπες ([33]).

Το πρόβλημα Maximum Weighted Clique On Visibility Graph ανήκει στο P αν το πολύγωνο δεν έχει τρύπες ([3, 13, 14, 17]). Εάν το πολύγωνο έχει τρύπες τότε το πρόβλημα δεν προσεγγίζεται με παράγοντα $\frac{n^{\frac{1}{6}-\gamma}}{4}$ για οποιοδήποτε $\gamma > 0$, εκτός αν P=NP ([17]).

Το πρόβλημα Minimum Clique Partition On Visibility Graph σε πολύγωνα χωρίς τρύπες έχει αποδειχθεί ότι είναι APX-hard, ενώ έχει βρεθεί προσεγγιστικός αλγόριθμος με παράγοντα $O(\log n)$ ([17]). Σε πολύγωνα με τρύπες έχει βρεθεί ότι το πρόβλημα δε μπορεί να προσεγγιστεί με παράγοντα καλύτερο από $O(n^{\epsilon})$ για κάποιο $\epsilon > 0$ ([17]).

Η τεχνική greedy που χρησιμοποιούμε για την εύρεση προσεγγιστικών αλγορίθμων για τα προβλήματα Maximum Length Guard και Maximum Value Guard έχει χρησιμοποιηθεί για την εύρεση προσεγγιστικού αλγορίθμου για το πρόβλημα Maximum Coverage ([40, 10, 49, 26]). Η τεχνική greedy που χρησιμοποιούμε για την προσέγγιση των προβλημάτων Budgeted Maximum Length Vertex Guard και Budgeted Maximum Value Vertex Guard έχει χρησιμοποιηθεί στην προσέγγιση του προβλήματος Budgeted Maximum Coverage ([30]).

Τέλος, το πρόβλημα Max-5-Occurence-3-Sat έχει αποδειχθεί ότι είναι APX-complete ([43]).

0.5 Μεγιστοποίηση κάλυψης - Αποτελέσματα

Η διατριβή εμπίπτει στην περιοχή της Θεωρητικής Πληροφορικής και ειδικότερα στην περιοχή των προσεγγιστικών αλγορίθμων για προβλήματα υπολογιστικής γεωμετρίας.

Στο πρώτο κεφάλαιο περιγράφουμε μια διαδικασία κατακερματισμού του πολυγώνου. Ειδικότερα στην ενότητα 1 του κεφαλαίου δίνουμε ένα τρόπο κατακερματισμού της περιμέτρου του πολυγώνου ως προς την ορατότητα. Αυτή τη διαδικασία (Finest Visibility Segmentation) θα καλούμε συχνά στους αλγορίθμους των επόμενων κεφαλαίων. Στη δεύτερη ενότητα περιγράφουμε μια διαδικασία κατακερματισμού του εσωτερικού του πολυγώνου ως προς την ορατότητα την οποία θα χρησιμοποιήσουμε στο κεφάλαιο 5. Τα αποτελέσματα της ενότητας 1 του κεφαλαίου έχουν δημοσιευθεί στο [35].

Στο δεύτερο κεφάλαιο ασχολούμαστε με το πρόβλημα της περιφρούρησης μέγιστου μήκους της περιμέτρου ενός πολυγώνου με k φύλακες (Maximum Length Guard). Αποδεικνύουμε ότι το πρόβλημα είναι NP-hard και δίνουμε αλγορίθμους πολυωνυμικού χρόνου που πετυχαίνουν σταθερή προσέγγιση ως προς τις βέλτιστες λύσεις για τις περιπτώσεις τοποθέτησης των φυλάκων σε κορυφές, ή πλευρές καθώς και για την περίπτωση πολυγώνων με τρύπες.

Στο κεφάλαιο 3 μελετούμε το πρόβλημα της κάλυψης μέγιστης αξίας της περιμέτρου (Maximum Value Guard). Αποδεικνύουμε ότι το πρόβλημα είναι NP-hard και δίνουμε αλγόριθμους πολυωνυμικού χρόνου για τις περιπτώσεις τοποθέτησης φυλάκων σε κορυφές ή πλευρές και για την περίπτωση πολύγωνων με τρύπες. Όλοι οι αλγόριθμοι που δίνουμε πετυχαίνουν σταθερό προσεγγιστικό παράγοντα. Στο ίδιο κεφάλαιο μελετούμε το πρόβλημα της επίβλεψης ή μερικής κάλυψης μέγιστης αξίας. Αποδεικνύουμε ότι το πρόβλημα παραμένει NP-hard και δίνουμε προσεγγιστικούς αλγορίθμους για τις περιπτώσεις φυλάκων - κορυφών, φυλάκων - πλευρών και πολυγώνων με τρύπες. Αποδεικνύουμε επίσης πως τα προβλήματα αυτού του κεφαλαίου δε μπορεί να έχουν προσεγγιστικά σχήματα πολυωνυμικού χρόνου. Δηλαδή τα προβλήματα αυτά δε μπορεί να ανήκουν στην κλάση FPTAS. Τα αποτελέσματα των κεφαλαίων 2 και 3 έχουν δημοσιευθεί στο [35].

Στο κεφάλαιο 4 μελετούμε το πρόβλημα περιφρούρησης μέγιστης αξίας με παράμετρο την τοποθέτηση των έργων τέχνης (Maximum Value Guard with Painting Placement). Δηλαδή εκτός από την τοποθέτηση των φυλάκων ζητάμε επίσης να τοποθετήσουμε τα δεδομένα έργα τέχνης πάνω στην περίμετρο έτσι ώστε να έχουμε περιφρούρηση μέγιστης αξίας. Δίνουμε προσεγγιστικούς αλγόριθμους που χρησιμοποιούν ένα πολυωνυμικό σχήμα για το πρόβλημα Multiple Knapsack. Αποδεικνύουμε ότι πετυχαίνουν σταθερούς παράγοντες προσέγγισης για τις περιπτώσεις φυλάκων - κορυφών, φυλάκων - πλευρών και πολυγώνων με τρύπες.

Στο κεφάλαιο 5 ορίζουμε το πρόβλημα της κάλυψης μέγιστου μήκους της περιμέτρου με όριο κόστους πομποδεκτών (Budgeted Maximum Length Vertex Guard). Το πρόβλημα του κεφαλαίου 2 (Maximum Length Vertex Guard) είναι ειδική περίπτωση του προβλήματος αυτού και έτσι το πρόβλημα Budgeted Maximum Length Vertex Guard είναι NP-hard. Ο αλγόριθμος όμως του προβλήματος Maximum Length Vertex Guard δεν οδηγεί αναγκαστικά σε προσεγγιστική λύση του προβλήματος Budgeted Maximum Length Vertex Guard. Δίνουμε έναν καινούριο αλγόριθμο και αποδεικνύουμε ότι πετυχαίνει σταθερό προσεγγιστικό παράγοντα για τις περιπτώσεις τοποθέτησης πομποδεκτών σε κορυφές και πλευρές, όπως επίσης και για την περίπτωση πολυγώνων με τρύπες.

Ομοίως, το πρόβλημα του κεφαλαίου 6, δηλαδή η κάλυψη μέγιστης αξίας με άνω όριο κόστους πομποδεκτών (Budgeted Maximum Value Vertex Guard) είναι πιο γενικό από το πρόβλημα Maximum Value Vertex Guard του κεφαλαίου 3. Δίνουμε και για αυτό το πρόβλημα πολυωνυμικού χρόνου προσεγγιστικούς αλγόριθμους.

Στο κεφάλαιο 7 εκτός από την τοποθέτηση των πομποδεκτών, κάνουμε και βέλτιστη τοποθέτηση αναμεταδοτών (Budgeted Maximum Value Vertex Guard with Painting Placement). Το πρόβλημα είναι γενίκευση του προβλήματος Maximum Value Guard with Painting Placement του κεφαλαίου 4.

Με το κεφάλαιο 8 κλείνει η ενότητα της παρουσίασης προσεγγιστικών αλγορίθμων για προβλήματα σε πολύγωνα, μελετώντας πως μπορούμε να καλύψουμε ένα μέγιστο εμβαδόν από το εσωτερικό του πολυγώνου (Maximum Area Vertex Guard).

Στο κεφάλαιο 9 ξεκινά μια νέα ενότητα, αυτή των αποτελεσμάτων μη - προσεγγισιμότητας. Υπενθυμίζουμε κάποια στοιχεία από τη θεωρία προσεγγισιμότητας και έπειτα περιγράφουμε μια αναγωγή διατήρησης χάσματος (gap preserving reduction) από το πρόβλημα Max-5-Occurence-3-Sat στο πρόβλημα Maximum Value Vertex Guard, αποδεικνύοντας ότι το πρόβλημα Maximum Value Vertex Guard είναι APX-hard.

Στη συνέχεια, κάνοντας τις κατάλληλες τροποποιήσεις στην προηγούμενη αναγωγή αποδεικνύουμε στο κεφάλαιο 10 ότι τα προβλήματα των κεφαλαίων 2 - 8 είναι APX-complete. Είναι γνωστό ([43]) ότι το πρόβλημα MAX-5-OCCURENCE-3-SAT είναι APX-complete. Οι αναγωγές αυτές χρησιμοποιούν στο κατασκευαστικό μέρος κάποιες ιδέες που χρησιμοποιήθηκαν στα [32, 16] για τις αποδείξεις NP-hardness και APX-hardness αντίστοιχα του προβλήματος MINIMUM VERTEX GUARD. Με το κεφάλαιο 10 κλείνει η ενότητα των αποτελεσμάτων μη - προσεγγισιμότητας. Τα αποτελέσματα των κεφαλαίων 9 και 10 έχουν δημοσιευθεί στο [36].

Στο κεφάλαιο 11 ασχολούμαστε με προβλήματα σε γραφήματα. Συγκεκριμένα, μελετούμε το πρόβλημα MAXIMUM WEIGHT IN k CLIQUES σε γράφους ορατότητας. Χρησιμοποιούμε τη μέθοδο του κεφαλαίου 2 και περιγράφουμε έναν προσεγγιστικό αλγόριθμο με σταθερό παράγοντα προσέγγισης. Τα αποτελέσματα του κεφαλαίου 11 έχουν δημοσιευθεί στο [35].

Στο κεφάλαιο 12 ορίζουμε ένα ενδιαφέρον πρόβλημα με εφαρμογές σε πολλές περιοχές. Το πρόβλημα αυτό ονομάζεται GENERALIZED SET COVER.

Τα αποτελέσματα των κεφαλαίων 4, 5, 6, 7, 8 έχουν σταλεί για δημοσίευση στα [37, 38, 39].

Τέλος στο κεφάλαιο 13 κάνουμε μια ανακεφαλαίωση των αποτελεσμάτων της διατριβής και προτείνουμε μια σειρά από προβλήματα για παραπέρα έρευνα.

Κεφάλαιο 1

Κατακερματισμός του πολυγώνου

Θα περιγράψουμε εδώ μια μέθοδο με την οποία μπορούμε να κατακερματίσουμε την περίμετρο του πολυγώνου ως προς την ορατότητα. Στη συνέχεια θα δείξουμε πως μπορούμε να γενικεύσουμε αυτή τη μέθοδο για τον κατακερματισμό του εσωτερικού του πολυγώνου ως προς την ορατότητα. Τη μέθοδο αυτή θα χρησιμοποιήσουμε στα επόμενα κεφάλαια.

1.1 Η περίμετρος του πολυγώνου

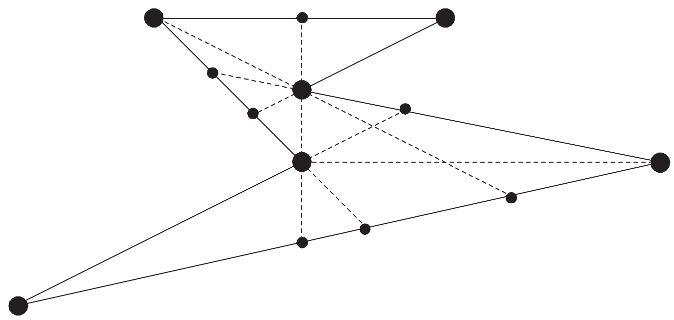
Έστω ένα πολύγωνο P με n κορυφές. Για αυτό το πολύγωνο μπορούμε να κατασκευάσουμε ένα γράφημα $G(V, E)$ όπου το σύνολο V είναι το σύνολο των κορυφών του πολυγώνου και $(v, u) \in E$ αν και μόνο αν η κορυφή v του πολυγώνου βλέπει την κορυφή u. Αυτός ο γράφος ονομάζεται γράφος ορατότητας (visibility graph).

Για κάθε ακμή $(v, u) \in E$, προεκτείνουμε το ευθύγραμμο τμήμα (v, u) όσο η προέκταση παραμένει στο εσωτερικό του πολυγώνου και έως ότου αυτή τμήσει την περίμετρο του πολυγώνου. Το σύνολο των σημείων τομής το ονομάζουμε FVS. Το σύνολο αυτό περιέχει και όλες τις κορυφές του πολυγώνου (σχήμα 1.1).

Πρόταση 1.1 *Έστω $v \in V(P)$ μια τυχαία κορυφή του πολυγώνου. Έστω a, b δύο διαδοχικά (πάνω στην περίμετρο του πολυγώνου) σημεία που ανήκουν στο σύνολο FVS. Το τμήμα της περιμέτρου (a, b) είναι ορατό από την κορυφή v αν και μόνο αν επιβλέπεται από την κορυφή v.*

Απόδειξη. Εάν ένα τμήμα είναι ορατό από κάποια κορυφή τότε προφανώς επιβλέπεται.

Ας υποθέσουμε ότι ένα τμήμα FVS (a, b) επιβλέπεται από μια κορυφή v αλλά δεν είναι ορατό ολόκληρο. Έστω ότι το σύνολο των σημείων του (a, b) που φαίνονται από την κορυφή v είναι (c, d), όπως στο σχήμα 1.2. Το σύνολο (c, d) υπάρχει αφού το (a, b) επιβλέπεται από την κορυφή v. Εφόσον το v δεν βλέπει αριστερά του c, σημαίνει ότι υπάρχει κορυφή u_1, που εμποδίζει την ορατότητα του v. Για τον ίδιο λόγο υπάρχει κορυφή u_2 που εμποδίζει την ορατότητα του v δεξιά του d. Όμως τότε



Σχήμα 1.1: Χωρίζοντας την περίμετρο του πολυγώνου σε τμήματα με άκρα στο σύνολο FVS

από την κατασκευή του κατακερματισμού, τα c, d θα έπρεπε να ανήκουν στο σύνολο FVS και συνεπώς τα a, b δεν θα ήταν διαδοχικά πάνω στην περίμετρο. ΑΤΟΠΟ. □

Οι συνέπειες της παραπάνω πρότασης είναι:

- Οποιοδήποτε FVS τμήμα της περιμέτρου του πολυγώνου είτε θα είναι ορατό και θα επιβλέπεται από κάποια κορυφή είτε όχι.
- Ένα οποιοδήποτε τμήμα της περιμέτρου του πολυγώνου που ορίζεται από διαδοχικά (πάνω στην περίμετρο) σημεία του συνόλου FVS και είναι ορατό από μια κορυφή v, δεν μπορεί να περιέχει γνήσιο υποτμήμα ορατό από κάποια άλλη κορυφή.

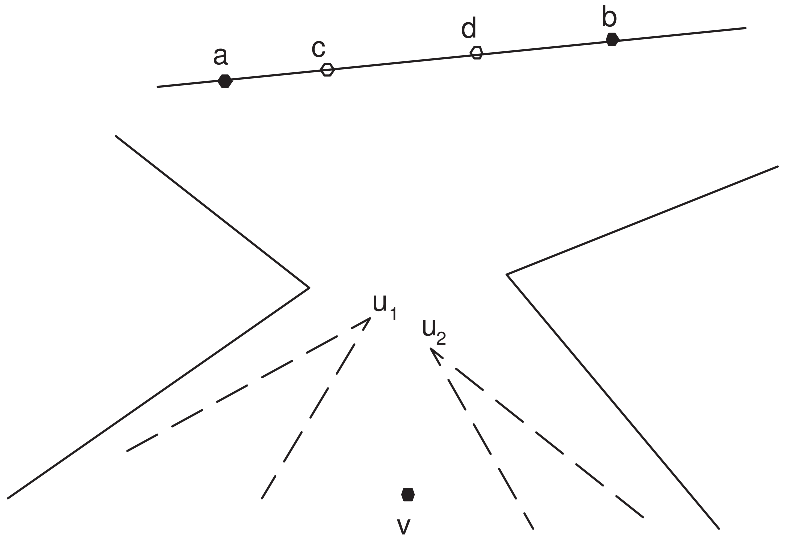
Εάν το πολύγωνο έχει n κορυφές, ο αριθμός των ακμών του γράφου ορατότητας είναι $O(n^2)$. Συνεπώς ο αριθμός των σημείων τομής των $O(n^2)$ ακμών με τις $O(n)$ πλευρές του πολυγώνου είναι $O(n^2)$, αφού οι προεκτάσεις μιας ακμής του γράφου ορατότητας έχουν το πολύ 2 σημεία τομής με την περίμετρο του πολυγώνου. Την κατασκευή αυτή την ονομάζουμε **Finest Visibility Segmentation**.

Πρόταση 1.2 *Έστω e μια πλευρά του πολυγώνου. Τα τμήματα FVS που επιβλέπονται από την πλευρά e είναι ακριβώς αυτά που φαίνονται από την πλευρά e.*

Απόδειξη. Αν ένα τμήμα FVS φαίνεται από μια πλευρά e, τότε προφανώς επιβλέπεται.

Έστω ότι επιβλέπεται και δεν φαίνεται ολόκληρο από την πλευρά e. Δηλαδή σημαίνει ότι υπάρχει μια πλευρά e' που εμποδίζει την ορατότητα κάποιου υποτμήματος του τμήματος FVS από την πλευρά e.

Όμως τότε σημαίνει όπως φαίνεται στο σχήμα 1.3 ότι τα a, b δεν είναι διαδοχικά (πάνω στην περίμετρο) FVS σημεία. ΑΤΟΠΟ.



Σχήμα 1.2: Ένα τμήμα FVS (a, b) φαίνεται από την κορυφή v αν και μόνο αν επιβλέπεται από την v

Συνεπώς εάν κάποιο FVS τμήμα περιφρουρείται ή επιβλέπεται από κάποιο σημείο της πλευράς, τότε προφανώς επιβλέπεται και περιφρουρείται από την πλευρά.

□

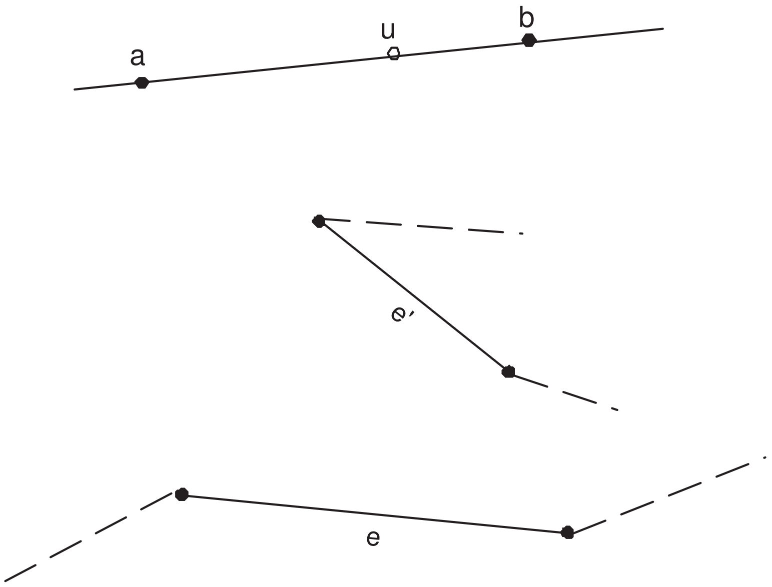
Οι συνέπειες της προηγούμενης πρότασης είναι ανάλογες με τις συνέπειες της πρότασης 1.1.

- Οποιοδήποτε FVS τμήμα της περιμέτρου του πολυγώνου είτε θα είναι ορατό και θα επιβλέπεται από κάποια πλευρά είτε όχι.
- Ένα οποιοδήποτε τμήμα της περιμέτρου του πολυγώνου που ορίζεται από διαδοχικά (πάνω στην περίμετρο) σημεία του συνόλου FVS και είναι ορατό από μια πλευρά e, δεν μπορεί να περιέχει γνήσιο υποτμήμα ορατό από κάποια άλλη πλευρά.

Αν θέλουμε να βρούμε τα FVS τμήματα που επιβλέπει και περιφρουρεί μια πλευρά e εργαζόμαστε ως εξής: για κάθε ένα FVS τμήμα παίρνουμε ένα σημείο στο εσωτερικό του και εξετάζουμε αν υπάρχει ευθεία που ενώνει αυτό το σημείο με ένα σημείο στο εσωτερικό της e. Εάν ναι, τότε το τμήμα FVS επιβλέπεται και περιφρουρείται από την e. Εάν όχι, τότε σημαίνει ότι υπάρχει σημείο του FVS τμήματος που δεν φαίνεται από την e. Συνεπώς σ' αυτή την περίπτωση το FVS τμήμα δεν περιφρουρείται ούτε επιβλέπεται από την e.

Η παραπάνω κατασκευή μπορεί να εφαρμοστεί χωρίς καμμία αλλαγή και σε πολύγωνα με τρύπες.

Η χρονική πολυπλοκότητα της εύρεσης των τμημάτων που επιβλέπονται (και είναι ορατά) από μια πλευρά είναι $O(n^2)$.



Σχήμα 1.3: Ένα τμήμα FVS (a, b) φαίνεται από την πλευρά e αν και μόνο αν επιβλέπεται από την e

1.2 Το εσωτερικό του πολυγώνου

Θα δούμε τώρα πως μπορούμε να κατακερματίσουμε με ανάλογο τρόπο το εσωτερικό του πολυγώνου.

Μπορούμε να γενικεύσουμε την κατασκευή της προηγούμενης ενότητας ως εξής:

Παίρνουμε τις προεκτάσεις των ακμών του γράφου ορατότητας, όσο αυτές παραμένουν μέσα στο πολύγωνο και βάζουμε στο σύνολο FVS όλα τα **σημεία τομής** μεταξύ αυτών των προεκτάσεων. Ορίζουμε σαν επιφάνειες FVS στο εσωτερικό του πολυγώνου, τις επιφάνειες που δεν περιέχουν κανένα FVS σημείο μέσα.

Ο αριθμός των επιφανειών FVS είναι $O(n^4)$. Είναι προφανές από την κατασκευή ότι δεν μπορεί να υπάρχει προέκταση ακμής του γράφου ορατότητας εντός του πολυγώνου που να τέμνει κάποια επιφάνεια FVS.

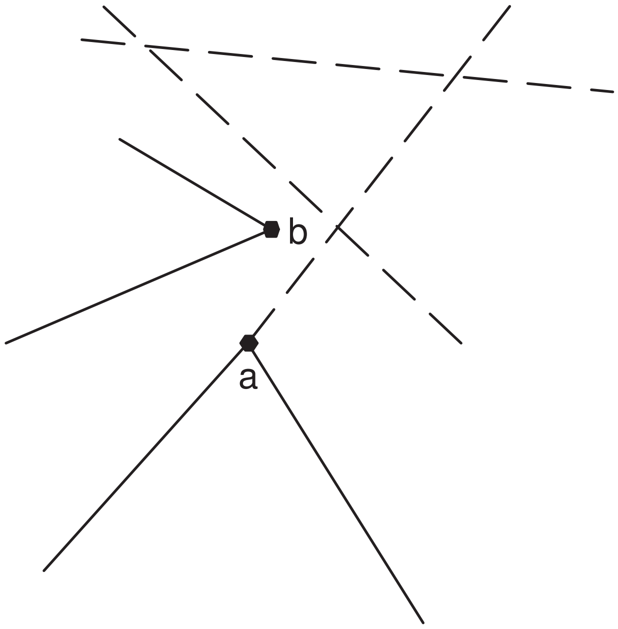
Πρόταση 1.3 *Μια επιφάνεια FVS είναι ορατή από μια κορυφή του πολυγώνου αν και μόνο αν επιβλέπεται από αυτήν.*

Απόδειξη. Εάν μια επιφάνεια FVS είναι ορατή από μια κορυφή, προφανώς επιβλέπεται από αυτήν.

Αντίστροφα, εάν επιβλέπεται, ας υποθέσουμε ότι δεν είναι ορατή. Δηλαδή υπάρχει κάποια άλλη κορυφή b που εμποδίζει την ορατότητα της κορυφής a όπως στο σχήμα 1.4

Όμως τότε η προέκταση του τμήματος (a, b) τέμνει την επιφάνεια FVS. ΑΤΟΠΟ. □

Οι συνέπειες της παραπάνω πρότασης είναι:



Σχήμα 1.4: Μια επιφάνεια FVS είναι ορατή από την κορυφή a αν και μόνο αν επιβλέπεται από την a

- Οποιαδήποτε επιφάνεια FVS του πολυγώνου είτε θα είναι ορατή και θα επιβλέπεται από κάποια κορυφή είτε όχι.
- Μια οποιαδήποτε επιφάνεια FVS του πολυγώνου που είναι ορατή από μια κορυφή v, δεν μπορεί να περιέχει γνήσια υποπεριοχή ορατή από κάποια άλλη κορυφή.

Για παρόμοιους λόγους με τους παραπάνω ισχύει:

Πρόταση 1.4 *Μια επιφάνεια FVS είναι ορατή από μια πλευρά του πολυγώνου αν και μόνο αν επιβλέπεται από αυτήν.*

Οι συνέπειες της πρότασης είναι:

- Οποιαδήποτε επιφάνεια FVS του πολυγώνου είτε θα είναι ορατή και θα επιβλέπεται από κάποια πλευρά είτε όχι.
- Μια οποιαδήποτε επιφάνεια FVS του πολυγώνου που είναι ορατή από μια πλευρά e, δεν μπορεί να περιέχει γνήσια υποπεριοχή ορατή από κάποια άλλη πλευρά.

Η εύρεση των επιφανειών FVS που περιφρουρούνται και επιβλέπονται από κάποια πλευρά μπορεί να γίνει σε πολυωνυμικό χρόνο με τρόπο ανάλογο του τρόπου εύρεσης των περιφρουρούμενων τμημάτων FVS από μια πλευρά.

Κεφάλαιο 2

Περιφρούρηση μέγιστου μήκους με k φύλακες

2.1 Φύλακες στις κορυφές

Έστω ένα πολύγωνο P χωρίς τρύπες. Ζητάμε να καλύψουμε ένα μέγιστο μέρος της περιμέτρου του πολυγώνου, χρησιμοποιώντας το πολύ k φύλακες (σταθμούς). Οι φύλακες επιτρέπεται να τοποθετηθούν πάνω στις κορυφές του πολυγώνου.

Θα αποδείξουμε ότι το Maximum Length Vertex Guard είναι NP-hard, ανάγοντας το πρόβλημα απόφασης Minimum Vertex Guard (Boundary Restricted) στο πρόβλημα απόφασης Maximum Length Vertex Guard.

Ορισμός 29 *Στο πρόβλημα απόφασης* Minimum Vertex Guard (Boundary Restricted) *δίνεται ένα πολύγωνο P με μήκος περιμέτρου $L(P)$ και ένας ακέραιος $k > 0$. Το ζητούμενο είναι: μπορούμε να τοποθετήσουμε το πολύ k φύλακες στις κορυφές του πολυγώνου έτσι ώστε το μήκος της περιμέτρου του πολυγώνου που φαίνεται από τους φύλακες να είναι $L(P)$;*

Ορισμός 30 *Στο πρόβλημα απόφασης* Maximum Length Vertex Guard *δίνεται ένα πολύγωνο P, ένας ακέραιος $k > 0$ και ένας ακέραιος B. Το ζητούμενο είναι να τοποθετήσουμε στις κορυφές του P, k φύλακες έτσι ώστε το ευκλείδειο μήκος της περιμέτρου του πολυγώνου που φαίνεται από τους k φύλακες να είναι $\geq B$.*

Πρόταση 2.1 *Το πρόβλημα* Maximum Length Vertex Guard *είναι NP-hard.*

Απόδειξη. Έστω ένα στιγμιότυπο του προβλήματος Minimum Vertex Guard (Boundary Restricted): Δίνεται πολύγωνο P, ένας ακέραιος $k > 0$ και ζητάμε να βρούμε αν μπορούμε να καλύψουμε όλη την περίμετρο του πολυγώνου, τοποθετώντας στις κορυφές του πολυγώνου $\leq k$ φύλακες. Κατασκευάζουμε το εξής στιγμιότυπο του προβλήματος απόφασης Maximum Length Vertex Guard: παίρνουμε το ίδιο πολύγωνο P, τον ακέραιο $k > 0$ και έναν ακέραιο B που ισούται με το ευκλείδειο μήκος όλης της περιμέτρου του πολυγώνου. Η ερώτηση είναι: μπορούμε να τοποθετήσουμε k φύλακες στις κορυφές του πολυγώνου έτσι ώστε το μέρος της περιμέτρου του πολυγώνου που φαίνεται από τους φύλακες να έχει μήκος $\geq B$; Αν η απάντηση στο δεύτερο πρόβλημα είναι 'ΝΑΙ' ('ΟΧΙ'), τότε η απάντηση στο πρώτο πρόβλημα είναι 'ΝΑΙ' ('ΟΧΙ'). □

Θα περιγράψουμε τώρα έναν αλγόριθμο πολυωνυμικού χρόνου ο οποίος προσεγγίζει το πρόβλημα Maximum Length Vertex Guard κατά ένα σταθερό προσεγγιστικό παράγοντα.

Αρχικά κάνουμε τον κατακερματισμό της περιμέτρου του πολυγώνου όπως περιγράφτηκε στην ενότητα 1 του κεφαλαίου 1. Τα FVS σημεία είναι άκρα τμημάτων με την ακόλουθη ιδιότητα: για κάθε κορυφή $v \in V(P)$, ένα τμήμα (a, b) το οποίο ορίζεται από διαδοχικά FVS σημεία a, b είναι ορατό από το v αν και μόνο αν επιβλέπεται από το v.

Ο αλγόριθμος 1 λοιπόν βρίσκει σε πολυωνυμικό χρόνο τα $O(n^2)$ FVS σημεία και τμήματα. Μετά βρίσκει για κάθε κορυφή v του πολυγώνου, το σύνολο των FVS τμημάτων που φαίνονται από αυτή το οποίο σύνολο καλούμε $E'(v)$. Με τον υπολογισμό των συνόλων $E'(v)$ τελειώνει το προκαταρκτικό μέρος του αλγορίθμου.

Στο κυρίως μέρος του αλγορίθμου 1, σε κάθε βήμα τοποθετεί φύλακα στην κορυφή που βλέπει τα τμήματα FVS με το συνολικό μεγαλύτερο μήκος και τα οποία δεν φαίνονταν μέχρι τώρα.

Algorithm 1 MaxLegthVertexGuards (* greedy *)

```
compute FVS points
for every v ∈ V(P) do
  compute E'(v)
end for
SOL ← ∅
for i = 1 to k do
  select v ∈ V that maximizes L(SOL ∪ E'(v))
  SOL ← SOL ∪ E'(v)
end for
return L(SOL)
```

Θα αποδείξουμε ότι ο αλγόριθμος 1 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει σταθερό παράγοντα προσέγγισης. Θα μας χρειαστούν δύο λήμματα τα οποία αποδεικνύουμε στη συνέχεια.

Έστω OPT η συλλογή των συνόλων των τμημάτων FVS που ανήκουν στη βέλτιστη λύση και SOL η συλλογή που μας επέστρεψε ο αλγόριθμος 1. Έστω $L(OPT)$ το συνολικό μήκος των τμημάτων FVS που ανήκουν στη βέλτιστη λύση και $L(SOL)$ το συνολικό μήκος των τμημάτων FVS που επιστρέφει ο αλγόριθμος 1. Έστω S_i το i-οστό σύνολο των τμημάτων που επιλέγεται από τον αλγόριθμο στο βήμα i. Θεωρούμε ότι

$$L(S_i) = L(\cup_{m=1}^{i} S_m) - L(\cup_{m=1}^{i-1} S_m)$$

Δηλαδή το σύνολο S_i δεν περιέχει FVS τμήματα που έχουν ήδη καλυφθεί από προηγούμενα σύνολα.

Έστω ότι τα $S_1, S_2, ..., S_{l-1}$ είναι σύνολα που ανήκουν και στη βέλτιστη λύση και το S_l είναι το πρώτο σύνολο που συναντάμε το οποίο ανήκει στη λύση που μας δίνει ο αλγόριθμος, όχι όμως και στη βέλτιστη. Συνεπώς:

$$L(OPT) = L(\cup_{i=1}^{l-1} S_i) + L(\cup_{i=l}^{k} S_i')$$

όπου S_i' είναι τα υπόλοιπα σύνολα που περιέχει η βέλτιστη λύση.

Λήμμα 2.1 *Μετά από l βήματα του αλγορίθμου 1 ισχύει:*

$$L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i) \geq \frac{L(OPT) - L(\cup_{i=1}^{l-1} S_i)}{k}, l = 1, 2, ..., k$$

Απόδειξη. Ισχύει ότι:

$$L(S_{i-1}) \geq L(S_i), i = 2, ..., k$$

αφού έτσι κάνει την επιλογή ο αλγόριθμος. Το S_l που επιλέγει ο αλγόριθμος έχει:

$$L(S_l) \geq L(S_i'), i = l, l+1, ..., k$$

Δηλαδή έχει το μεγαλύτερο μήκος από όλα τα υπόλοιπα σύνολα που έχει η βέλτιστη λύση και δεν έχει επιλέξει ο αλγόριθμος. Συνεπώς:

$$L(OPT) \leq L(\cup_{i=1}^{l-1} S_i) + kL(S_l)$$

Ισχύει επίσης:

$$L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i) = L(S_l)$$

Από τις δύο παραπάνω σχέσεις:

$$L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i) \geq \frac{L(OPT) - L(\cup_{i=1}^{l-1} S_i)}{k}$$

□

Λήμμα 2.2 *Μετά από l βήματα του αλγορίθμου 1 ισχύει:*

$$L(\cup_{i=1}^{l} S_i) \geq (1 - (1 - \frac{1}{k})^l) L(OPT), l = 1, ..., k$$

Απόδειξη.

Θα το αποδείξουμε με επαγωγή στο l.

Από την αρχή του περιστερώνα, ισχύει ότι από τα σύνολα που έχει η βέλτιστη λύση υπάρχει τουλάχιστον ένα με μήκος τουλάχιστον

$$\frac{L(OPT)}{k}$$

Το μεγαλύτερο μήκος το έχει το πρώτο σύνολο που επιλέγει ο αλγόριθμος. Συνεπώς:

$$L(S_1) \geq \frac{L(OPT)}{k}$$

Έστω ότι ισχύει για $l-1$, δηλαδή

$$L(\cup_{i=1}^{l-1} S_i) \geq (1-(1-\frac{1}{k})^{l-1})L(OPT)$$

Έχουμε:

$$L(\cup_{i=1}^{l} S_i) = L(\cup_{i=1}^{l-1} S_i) + (L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i))$$

Από το προηγούμενο λήμμα ισχύει:

$$L(\cup_{i=1}^{l} S_i) \geq L(\cup_{i=1}^{l-1} S_i) + \frac{L(OPT) - L(\cup_{i=1}^{l-1} S_i)}{k} \rightarrow$$

$$\rightarrow L(\cup_{i=1}^{l} S_i) \geq L(\cup_{i=1}^{l-1} S_i)(1-\frac{1}{k}) + \frac{L(OPT)}{k}$$

Από την επαγωγική υπόθεση:

$$L(\cup_{i=1}^{l} S_i) \geq (1-(1-\frac{1}{k})^{l-1})L(OPT)(1-\frac{1}{k}) + \frac{L(OPT)}{k} \rightarrow$$

$$L(\cup_{i=1}^{l} S_i) \geq (1-(1-\frac{1}{k})^{l})L(OPT)$$

□

Θεώρημα 2.1 *Ο αλγόριθμος 1 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* $\frac{1}{1-\frac{1}{e}} = 1.58$ *του προβλήματος* MAXIMUM LENGTH VERTEX GUARD.

Απόδειξη. Από το λήμμα προηγούμενο λήμμα, θέτοντας $l = k$ έχουμε:

$$L(\cup_{i=1}^{k} S_i) \geq (1-(1-\frac{1}{k})^{k})L(OPT)$$

Ισχύει ότι:

$$lim_{k\rightarrow\infty}(1-(1-\frac{1}{k})^{k}) = 1-\frac{1}{e}$$

Ο όρος $(1-(1-\frac{1}{k})^{k})$ φθίνει, συνεπώς:

$$1-(1-\frac{1}{k})^k \geq 1-\frac{1}{e} > 0,632$$

άρα

$$L(SOL) > (1-\frac{1}{e})L(OPT)$$

επομένως ο αλγόριθμος πετυχαίνει προσεγγιστικό παράγοντα $\frac{1}{1-\frac{1}{e}} = 1.58$ του προβλήματος Maximum Length Vertex Guard.

□

2.2 Φύλακες - πλευρές

Στην περίπτωση που έχουμε φύλακες που καταλαμβάνουν ολόκληρες πλευρές του πολυγώνου τότε το πρόβλημα συνεχίζει να είναι NP-hard. Μπορούμε να ανάγουμε το πρόβλημα απόφασης Minimum Edge Guard (Boundary Restricted) στο Maximum Length Edge Guard όπως κάναμε στην πρόταση 2.1.

Πρόταση 2.2 *Το πρόβλημα* Maximum Length Edge Guard *είναι NP-hard.*

Ένας φύλακας - πλευρά μπορεί να αναπαριστά ένα φύλακα που κινείται από το ένα άκρο της πλευράς στο άλλο. Ο αλγόριθμος 2 που προσεγγίζει αυτό το πρόβλημα είναι παρόμοιος με τον αλγόριθμο 1 με τη διαφορά ότι υπολογίζει αρχικά το σύνολο $E'(e)$ το οποίο περιλαμβάνει τα FVS τμήματα που φαίνονται από την πλευρά e.

Για τον υπολογισμό κάθε συνόλου $E'(e)$ ο αλγόριθμος 2 εξετάζει όλα τα τμήματα FVS όπως αυτό περιγράφεται στην ενότητα 1 του κεφαλαίου 1.

Algorithm 2 MaxLegthEdgeGuards (* greedy *)

```
compute FVS points
for every e ∈ E(P) do
   compute E'(e)
end for
SOL ← ∅
for i = 1 to k do
   select e ∈ E that maximizes L(SOL ∪ E'(e))
   SOL ← SOL ∪ E'(e)
end for
return L(SOL)
```

Θεώρημα 2.2 *Ο αλγόριθμος 2 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* $\frac{1}{1-\frac{1}{e}} = 1.58$ *του προβλήματος* Maximum Length Edge Guard.

Απόδειξη. Βλέπε θεώρημα 2.1. □

2.3 Πολύγωνα με τρύπες

Τα αποτελέσματα των παραγράφων 2.1 και 2.2 ισχύουν και στην περίπτωση που το δεδομένο πολύγωνο έχει τρύπες. Σε αυτήν την περίπτωση το σύνολο των κορυφών του πολυγώνου περιλαμβάνει και τις κορυφές από τις τρύπες, το σύνολο των πλευρών του πολυγώνου περιλαμβάνει και τις πλευρές από τις τρύπες και η περίμετρος του πολυγώνου περιλαμβάνει και τις περιμέτρους από τις τρύπες.

Τα προβλήματα Maximum Length Vertex Guard και Maximum Length Edge Guard στην περίπτωση πολυγώνου με τρύπες είναι NP-hard καθώς η λύση τους θα οδηγούσε σε λύση των προβλημάτων Minimum Vertex Guard (Boundary Restricted) και Minimum Edge Guard (Boundary Restricted) αντίστοιχα σε πολύγωνα με τρύπες. Οι αποδείξεις είναι όμοιες με αυτές των προτάσεων 2.1 και 2.2.

Πρόταση 2.3 *Το πρόβλημα* Maximum Length Vertex Guard *σε πολύγωνα με τρύπες είναι NP-hard.*

Πρόταση 2.4 *Το πρόβλημα* Maximum Length Edge Guard *σε πολύγωνα με τρύπες είναι NP-hard.*

Οι αλγόριθμοι 1 και 2 μπορούν να χρησιμοποιηθούν στα προβλήματα Maximum Length Vertex Guard και Maximum Length Edge Guard αντίστοιχα για πολύγωνα με τρύπες πετυχαίνοντας προσεγγιστικό παράγοντα 1.58 ως προς τις αντίστοιχες βέλτιστες λύσεις.

Η περίπτωση των πολυγώνων με τρύπες είναι ενδιαφέρουσα και αρκετά ρεαλιστική. Οι τρύπες μπορούν να αναπαριστούν εμπόδια τα οποία περιορίζουν την ορατότητα σημείων του πολυγώνου. Επίσης μας δίνουν την ευκαιρία τοποθέτησης φυλάκων στο εσωτερικό του πολυγώνου σε προκαθορισμένα σημεία (στις κορυφές ή πλευρές των τρυπών). Σημειώνουμε επίσης πως τα προβλήματα σε πολύγωνα με τρύπες είναι αρκετά πιο γενικά. Ενδεικτικό του τελευταίου είναι πως το πρόβλημα Minimum Vertex Guard σε πολύγωνα με τρύπες είναι $\log n$ -hard ([18]). Δηλαδή δεν υπάρχει αλγόριθμος πολυωνυμικού χρόνου που να πετυχαίνει σταθερή προσέγγιση για αυτό το πρόβλημα.

Κεφάλαιο 3

Κάλυψη μέγιστης αξίας με k φύλακες

Έστω ένα πολύγωνο P. Η περίμετρος του είναι χωρισμένη σε ξένα μεταξύ τους διαστήματα στα οποία έχουν αποδοθεί βάρη (σχήμα 3.1). Ο στόχος είναι να καλύψουμε το μέγιστο δυνατό βάρος τοποθετώντας το πολύ k σταθμούς.

3.1 Φύλακες σε κορυφές

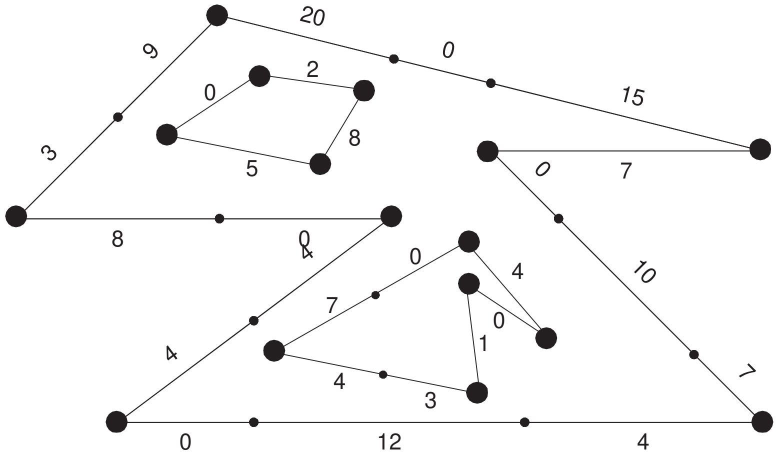
Σε αυτή την παράγραφο εξετάζουμε το πρόβλημα σε πολύγωνα χωρίς τρύπες. Έχουμε επίσης τη δυνατότητα να τοποθετούμε φύλακες μόνο στις κορυφές του πολυγώνου.

θα αποδείξουμε ότι το πρόβλημα Maximum Value Vertex Guard είναι NP-hard ανάγοντας το πρόβλημα απόφασης Minimum Vertex Guard (Boundary Restricted) στο πρόβλημα απόφασης Maximum Value Vertex Guard.

Πρόταση 3.1 *Το πρόβλημα* Maximum Value Vertex Guard *είναι NP-hard.*

Απόδειξη. Έστω ένα πολύγωνο P και ένας αριθμός $k > 0$. Αυτό είναι ένα στιγμιότυπο του προβλήματος απόφασης Minimum Vertex Guard που ολοκληρώνεται με την ερώτηση: Μπορεί να καλυφθεί όλη η περίμετρος χρησιμοποιώντας το πολύ k φύλακες;

Για να κατασκευάσουμε το στιγμιότυπο του προβλήματος απόφασης Maximum Value Vertex Guard, χρησιμοποιούμε το ίδιο πολύγωνο P. Χωρίζουμε την περίμετρο του πολυγώνου σε διαστήματα χρησιμοποιώντας την FVS κατασκευή. Κάθε πλευρά του πολυγώνου χωρίζεται έτσι σε μη-επικαλυπτόμενα διαστήματα. Κάθε τέτοιο διάστημα επιβλέπεται αν και μόνο αν είναι ορατό από μια κορυφή. Αναθέτουμε βάρος 1 σε κάθε τέτοιο διάστημα. Έστω T το συνολικό βάρος. Λαμβάνοντας υπόψιν ότι κάθε κομμάτι της περιμέτρου είναι ορατό από κάποια κορυφή έχουμε: η περίμετρος του πολυγώνου μπορεί να καλυφθεί με k φύλακες αν και μόνο αν στο νέο πολύγωνο με βάρη μπορούμε να τοποθετήσουμε k φύλακες που βλέπουν βάρος T. □



Σχήμα 3.1: Ένα πολύγωνο με βάρη σε τμήματα των πλευρών

Στη συνέχεια δίνουμε έναν αλγόριθμο πολυωνυμικού χρόνου ο οποίος πετυχαίνει σταθερό παράγοντα προσέγγισης για το πρόβλημα Maximum Value Vertex Guard.

Αρχικά ο αλγόριθμος 3 υπολογίζει για κάθε κορυφή $v \in V(P)$ το σύνολο $S(v)$ των διαστημάτων που φαίνονται από την κορυφή v. Αυτό μπορεί να γίνει πολύ εύκολα καθώς ένα τμήμα φαίνεται ολόκληρο από μια κορυφή αν και μόνο αν και τα δύο άκρα του φαίνονται από την κορυφή. Συνεπώς για κάθε κορυφή αρκεί να εξεταστεί μια φορά η περίμετρος του πολυγώνου. Έπειτα από τον υπολογισμό των συνόλων $S(v)$ ο αλγόριθμος 3 επιλέγει σε κάθε βήμα να τοποθετήσει φύλακα στην κορυφή που προκαλεί τη μεγαλύτερη αύξηση βάρους στη λύση.

Algorithm 3 MaxValueVertexGuards (* greedy *)

for every $v \in V(P)$ **do**
  compute $S(v)$
end for
$SOL \leftarrow \emptyset$
for $i = 1$ to k **do**
  select $v \in V$ that maximizes `Weight`$(SOL \cup S(v))$
  $SOL \leftarrow SOL \cup S(v)$
end for
return `Weight`(SOL)

Θεώρημα 3.1 *Ο αλγόριθμος 3 τρέχει σε πολυωνυμικό χρόνο και προσεγγίζει τη βέλτιστη λύση του προβλήματος* Maximum Value Vertex Guard *με παράγοντα προσέγγισης* $\frac{1}{1-\frac{1}{e}} = 1.58$.

Απόδειξη. Βλέπε θεώρημα 2.1 αλλάζοντας το L με Weight. □

Τονίζουμε ότι στο πρόβλημα που εξετάσαμε θέλουμε για κάθε διάστημα που ανήκει στη λύση να υπάρχει τουλάχιστον ένας φύλακας που το βλέπει ολόκληρο. Το $S(v)$ περιέχει τα τμήματα που φαίνονται ολόκληρα από την κορυφή v. Στο κεφάλαιο 12 θα δούμε μια παραλλαγή του προβλήματος αυτού στην οποία δε μας ενδιαφέρει κάθε διάστημα που ανήκει στη λύση να φαίνεται ολόκληρο από κάποιο φύλακα. Για παράδειγμα στο σχήμα 3.2 το τμήμα (a, b) δεν φαίνεται από την κορυφή 1 ολόκληρο. Εάν όμως υπάρχουν φύλακες στις κορυφές 1 και 2 τότε ολόκληρο το (a, b) φαίνεται.

Παρακάτω αποδεικνύουμε ένα αρνητικό αποτέλεσμα για το πρόβλημα Maximum Value Vertex Guard.

Θεώρημα 3.2 *Το πρόβλημα* Maximum Value Vertex Guard *δεν επιδέχεται FPTAS εκτός αν P=NP.*

Απόδειξη. Έστω ότι υπάρχει FPTAS για αυτό το πρόβλημα. Θα αποδείξουμε ότι χρησιμοποιώντας αυτό το FPTAS μπορούμε να λύσουμε σε πολυωνυμικό χρόνο το πρόβλημα απόφασης Minimum Vertex Guard (Boundary Restricted).

Ας υποθέσουμε λοιπόν ότι υπάρχει ένας αλγόριθμος ο οποίος πετυχαίνει προσέγγιση $1 - \epsilon$, $\forall \epsilon > 0$ σε πολυωνυμικό χρόνο ως προς την είσοδο και το $\frac{1}{\epsilon}$ (βλ. παρ. 0.3.4). Δηλαδή η λύση που μας δίνει είναι:

$$SOL > (1 - \epsilon)OPT$$

Ας θυμηθούμε επίσης το πρόβλημα απόφασης Minimum Vertex Guard (Boundary Restricted): Μας δίνεται ένα πολύγωνο P και ένας ακέραιος $k > 0$. Θέλουμε να αποφασίσουμε αν μπορούμε να τοποθετήσουμε το πολύ k φύλακες στις κορυφές του πολυγώνου και να βλέπουμε όλη την περίμετρό του.

Μετασχηματίζουμε αυτό το στιγμιότυπο του προβλήματος Minimum Vertex Guard σε στιγμιότυπο του προβλήματος Maximum Value Vertex Guard ως εξής:

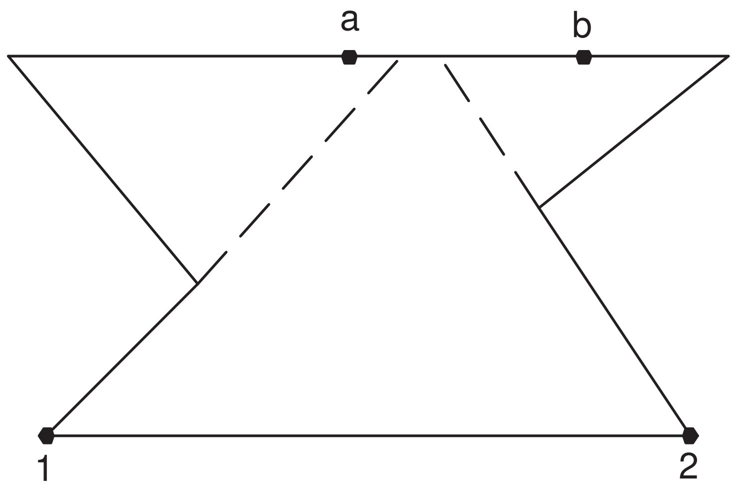
Παίρνουμε το ίδιο πολύγωνο P και χωρίζουμε την περίμετρό του σε $O(n^2)$ τμήματα με την FVS κατασκευή. Βάζουμε βάρος 1 σε κάθε τμήμα. Ρωτάμε: με k φύλακες τοποθετημένους σε κορυφές του πολυγώνου ποια είναι η μέγιστη αξία της περιμέτρου που μπορούμε να καλύψουμε; Εφαρμόζουμε το FPTAS με $\epsilon = \frac{1}{|FVS|}$.

Έστω SOL είναι η λύση που μας επιστρέφει το FPTAS και OPT μια βέλτιστη λύση.

- Αν $SOL = |FVS|$ τότε $OPT = |FVS|$, δηλαδή με k φύλακες μπορούμε να βλέπουμε όλη την περίμετρο, συνεπώς η απάντηση στο πρόβλημα απόφασης Minimum Vertex Guard είναι 'NAI'.

- Αν $SOL \leq |FVS| - 1$ τότε: από την ύπαρξη του FPTAS ισχύει:

$$SOL > (1 - \epsilon)OPT \rightarrow (1 - \frac{1}{|FVS|})OPT < SOL$$



Σχήμα 3.2: Το τμήμα (a, b) δεν φαίνεται ολόκληρο από την κορυφή a ούτε από την κορυφή b

και από την παραπάνω σχέση:

$$(1 - \frac{1}{|FVS|})OPT < |FVS| - 1 \rightarrow OPT < |FVS|$$

δηλαδή με k φύλακες δεν μπορεί να καλυφθεί όλη η περίμετρος, συνεπώς η απάντηση στο πρόβλημα απόφασης Minimum Vertex Guard είναι 'ΟΧΙ'.

Δηλαδή βλέπουμε ότι χρησιμοποιώντας το FPTAS μπορούμε σε χρόνο

$$Poly(n + \epsilon) = Poly(n)$$

να λύσουμε ένα NP-hard πρόβλημα απόφασης. Συνεπώς, εκτός αν $P = NP$, δε μπορεί να υπάρχει FPTAS για το πρόβλημα Maximum Value Vertex Guard. □

3.2 Φύλακες - πλευρές

Εξετάζουμε τώρα την περίπτωση στην οποία οι φύλακες καταλαμβάνουν ολόκληρες πλευρές.

Πρόταση 3.2 *Το* Maximum Value Edge Guard *είναι NP-hard.*

Απόδειξη. Η απόδειξη είναι όμοια με εκείνη της πρότασης 3.1 με τη διαφορά ότι η αναγωγή γίνεται από το πρόβλημα Minimum Edge Guard (Boundary Restricted). Κάθε πλευρά του πολυγώνου χωρίζεται σε μη-επικαλυπτόμενα διαστήματα χρησιμοποιώντας την FVS κατασκευή. Κάθε τέτοιο διάστημα επιβλέπεται αν και μόνο αν είναι ορατό από μια πλευρά. Τα υπόλοιπα είναι παρόμοια όπως στην πρόταση 3.1. □

Ο αλγόριθμος 4 είναι παρόμοιος με τον αλγόριθμο 2. Ο αλγόριθμος 4 αρχικά υπολογίζει για κάθε πλευρά e το σύνολο $S(e)$ των διαστημάτων που φαίνονται από την e. Ένας τρόπος για να γίνει αυτό είναι μέσω του υπολογισμού των FVS σημείων. Ένα από τα δεδομένα διαστήματα φαίνεται ολόκληρο από μια πλευρά, αν και μόνο αν όλα τα FVS διαστήματα που ανήκουν στο δεδομένο διάστημα επιβλέπονται από την πλευρά.

Algorithm 4 MaxValueEdgeGuards (* greedy *)

```
compute FVS points
for every e ∈ E(P) do
  compute E'(e)
end for
SOL ← ∅
for i = 1 to k do
  select e ∈ E that maximizes Weight(SOL ∪ E'(e))
  SOL ← SOL ∪ E'(e)
end for
return Weight(SOL)
```

Θεώρημα 3.3 *Ο αλγόριθμος 4 τρέχει σε πολυωνυμικό χρόνο και προσεγγίζει τη βέλτιστη λύση του προβλήματος* Maximum Value Edge Guard *με παράγοντα προσέγγισης* $\frac{1}{1-\frac{1}{e}} = 1.58$.

Απόδειξη. Βλέπε θεώρημα 2.2 αλλάζοντας το L με Weight. □

Θεώρημα 3.4 *Το πρόβλημα* Maximum Value Edge Guard *δεν επιδέχεται FPTAS εκτός αν P=NP.*

Απόδειξη. Η απόδειξη είναι όμοια με την απόδειξη του θεωρήματος 3.2. Αποδεικνύουμε ότι δεδομένου ενός FPTAS για το πρόβλημα Maximum Value Edge Guard, λύνουμε σε πολυωνυμικό χρόνο το πρόβλημα Minimum Edge Guard (Boundary Restricted). □

3.3 Πολύγωνα με τρύπες

Τα αποτελέσματα των παραγράφων 3.1 και 3.2 ισχύουν και σε πολύγωνα με τρύπες. Η μόνη διαφορά είναι ότι στην περίπτωση της τοποθέτησης φυλάκων σε κορυφές, ο τρόπος υπολογισμού των συνόλων $S(v)$ γίνεται μέσω της FVS κατασκευής. Δηλαδή μια κορυφή βλέπει ολόκληρο κάποιο από τα δεδομένα διαστήματα αν και μόνο αν βλέπει όλα τα FVS διαστήματα που ανήκουν σε αυτό. Συνεπώς:

Πρόταση 3.3 *Το πρόβλημα* Maximum Value Vertex Guard *σε πολύγωνα με τρύπες είναι NP-hard*

Απόδειξη. Βλέπε πρόταση 3.1. □

Πρόταση 3.4 *Το πρόβλημα* Maximum Value Edge Guard *σε πολύγωνα με τρύπες είναι NP-hard*

Απόδειξη. Βλέπε πρόταση 3.2 □

Οι αλγόριθμοι 3 και 4 μπορούν να χρησιμοποιηθούν και στην περίπτωση που τα πολύγωνα έχουν τρύπες πετυχαίνοντας τον ίδιο προσεγγιστικό παράγοντα.

3.4 Επίβλεψη (μερική κάλυψη)

Μια ενδιαφέρουσα παραλλαγή των προβλημάτων των παραγράφων 3.1 - 3.3 είναι η εξής: μας ενδιαφέρει μόνο η επίβλεψη των διαστημάτων από φύλακες και όχι η πλήρης κάλυψή τους. Θυμίζουμε ότι ένα τμήμα επιβλέπεται από ένα σημείο, αν ένα υποτμήμα του φαίνεται από το σημείο. Η έννοια της επίβλεψης, εκτός από χρήσιμη στην περιγραφή της ιδιότητας των FVS τμημάτων, είναι ενδιαφέρουσα καθώς νέες ρεαλιστικές εφαρμογές προκύπτουν.

Σε πολύγωνα με ή χωρίς τρύπες λοιπόν, ισχύουν:

Πρόταση 3.5 *Το* Maximum Value Vertex Guard *(watching version) είναι NP-hard.*

Απόδειξη. Βλέπε πρόταση 3.1 □

Πρόταση 3.6 *Το* Maximum Value Edge Guard *(watching version) είναι NP-hard.*

Απόδειξη. Βλέπε πρόταση 3.2 □

Algorithm 5 MaxValueVertexGuards (watching) (* greedy *)

compute FVS points
for every $v \in V(P)$ **do**
  compute $R(v)$
end for
$SOL \leftarrow \emptyset$
for $i = 1$ to k **do**
  select $v \in V$ that maximizes $\texttt{Weight}(SOL \cup R(v))$
  $SOL \leftarrow SOL \cup R(v)$
end for
return $\texttt{Weight}(SOL)$

Algorithm 6 MaxValueEdgeGuards (watching) (* greedy *)

compute FVS points
for every $e \in E(P)$ **do**
  compute $R(e)$
end for
$SOL \leftarrow \emptyset$
for $i = 1$ to k **do**
  select $e \in E$ that maximizes `Weight`$(SOL \cup R(e))$
  $SOL \leftarrow SOL \cup R(e)$
end for
return `Weight`(SOL)

Οι αλγόριθμοι 5 και 6 είναι παρόμοιοι με τους 3 και 4, με τη διαφορά ότι για τον υπολογισμό των συνόλων $R(v)$ και $R(e)$ κάνουμε τα εξής:

Χρησιμοποιούμε την FVS κατασκευή. Ένα δεδομένο διάστημα (a, b) επιβλέπεται από μια κορυφή, αν και μόνο αν τουλάχιστον ένα από τα FVS διαστήματα που ανήκει στο (a, b) επιβλέπεται από την κορυφή. Επίσης το (a, b) επιβλέπεται από μια πλευρά e αν έστω ένα από τα FVS διαστήματα που ανήκουν στην (a, b) επιβλέπεται από την e.

Με βάση τα παραπάνω ο υπολογισμός των συνόλων $R(v)$ και $R(e)$ εξετάζει την περίμετρο για κάθε v και e αντίστοιχα.

Θεώρημα 3.5 *Ο αλγόριθμος 5 τρέχει σε πολυωνυμικό χρόνο και προσεγγίζει τη βέλτιστη λύση του προβλήματος* Maximum Value Vertex Guard *(watching version) με παράγοντα προσέγγισης* $\frac{1}{1-\frac{1}{e}} = 1.58$.

Απόδειξη. Βλέπε θεώρημα 3.1 □

Θεώρημα 3.6 *Ο αλγόριθμος 6 τρέχει σε πολυωνυμικό χρόνο και προσεγγίζει τη βέλτιστη λύση του προβλήματος* Maximum Value Edge Guard *(watching version) με παράγοντα προσέγγισης* $\frac{1}{1-\frac{1}{e}} = 1.58$.

Απόδειξη. Βλέπε θεώρημα 3.3 □

Λόγω της ιδιότητας που έχουν τα FVS τμήματα να περιφρουρούνται αν και μόνο αν επιβλέπονται, μπορούμε να αποδείξουμε ότι με τον ίδιο τρόπο που αποδείξαμε το θεώρημα 3.2 αποδεικνύεται:

Θεώρημα 3.7 *Το πρόβλημα* Maximum Value Vertex/Edge Guard *(watching version) δεν επιδέχεται FPTAS εκτός αν P=NP.*

Κεφάλαιο 4

Βέλτιστη τοποθέτηση φυλάκων και έργων τέχνης

Εξετάζουμε τώρα μια νέα παραλλαγή του προβλήματος του κεφαλαίου 3. Δίνεται το πολύγωνο P και ένα σύνολο διαστημάτων με μήκη και βάρη. Δηλαδή η περίμετρος του P δεν είναι κατακερματισμένη. Με άλλα λόγια τα διαστήματα δεν είναι τοποθετημένα στην περίμετρο. Ένας επιπλέον στόχος εκείνων της παραγράφου 3.1 είναι και ο κατακερματισμός της περιμέτρου ή αλλιώς η τοποθέτηση των διαστημάτων στην περίμετρο.

4.1 Φύλακες σε κορυφές

Σαν συνέπεια της πρότασης 3.1 του κεφαλαίου 3 ισχύει:

Πρόταση 4.1 *Το* Maximum Value Vertex Guard with Painting Placement *είναι NP-hard.*

Παρακάτω φαίνεται ο αλγόριθμος 7 που προσεγγίζει το πρόβλημα Maximum Value Vertex Guard with Painting Placement.

Στον αλγόριθμο 7 χρησιμοποιούμε το πρόβλημα Multiple Knapsack που ξέρουμε ότι είναι NP-hard.

Είναι γνωστό ότι υπάρχει πολυωνυμικού χρόνου αλγόριθμος που προσεγγίζει το πρόβλημα Multiple Knapsack με παράγοντα προσέγγισης 2 ([47]). Μάλιστα έχει αποδειχθεί το επόμενο θεώρημα:

Θεώρημα 4.1 *[8] Υπάρχει PTAS για το πρόβλημα* Multiple Knapsack.

Ο αλγόριθμος 7 δουλεύει ως εξής: αρχικά υπολογίζει τα FVS σημεία του πολυγώνου μέσω της FVS κατασκευής. Από αυτά υπολογίζει για κάθε κορυφή $v \in V(P)$ το σύνολο $E'(v)$ των τμημάτων FVS που είναι ορατά από την κορυφή v. Στο κυρίως μέρος του, κατασκευάζει για κάθε κορυφή στην οποία δεν έχει τοποθετηθεί ακόμη φύλακας, ένα σύνολο από τμήματα $E'(v) - SOL \cap E'(v)$. Τα τμήματα αυτά

Algorithm 7 MaxValueVertexGuardWithPaintingPlacement (* greedy *)

compute FVS points
for every $v \in V(P)$ **do**
  compute $E'(v)$
end for
$SOL \leftarrow \emptyset$
for $i = 1$ to k **do**
  select $v \in V$ that maximizes $W(\texttt{multknap}(E'(v) - SOL \cap E'(v), D))$
  `update`D
end for
return $W(SOL)$

είναι τα τμήματα που φαίνονται από την κορυφή v μείον τα τμήματα εκείνα στα οποία έχουν ήδη τοποθετηθεί πίνακες. Κατόπιν παίρνει αυτό το σύνολο τμημάτων με τα μήκη τους μαζί με το σύνολο των πινάκων D που δεν έχουν ακόμα τοποθετηθεί και λύνει το πρόβλημα Multiple Knapsack. Δηλαδή θεωρεί το σύνολο $E'(v) - SOL \cap E'(v)$ ένα σύνολο σάκων με χωρητικότητες τα μήκη τους και τοποθετεί σ' αυτούς όσο μεγαλύτερο συνολικό βάρος από πίνακες μπορεί. Από όλες τις κορυφές επιλέγει εκείνη για την οποία τοποθετούνται στην περίμετρο οι πίνακες με το μεγαλύτερο συνολικό βάρος. Παρατηρούμε ότι η τοποθέτηση φύλακα με αυτόν τον τρόπο προκαλεί μια αύξηση στη συνολική λύση κατά έναν παράγοντα $1 - \epsilon$ της μέγιστης δυνατής αύξησης, λόγω του PTAS του Multiple Knapsack. Η λύση SOL, δηλαδή το σύνολο των τμημάτων με τους ήδη τοποθετημένους πίνακες, αυξάνει με την τοποθέτηση των νέων πινάκων όπως αυτό γίνεται από τη λύση του προβλήματος Multiple Knapsack.

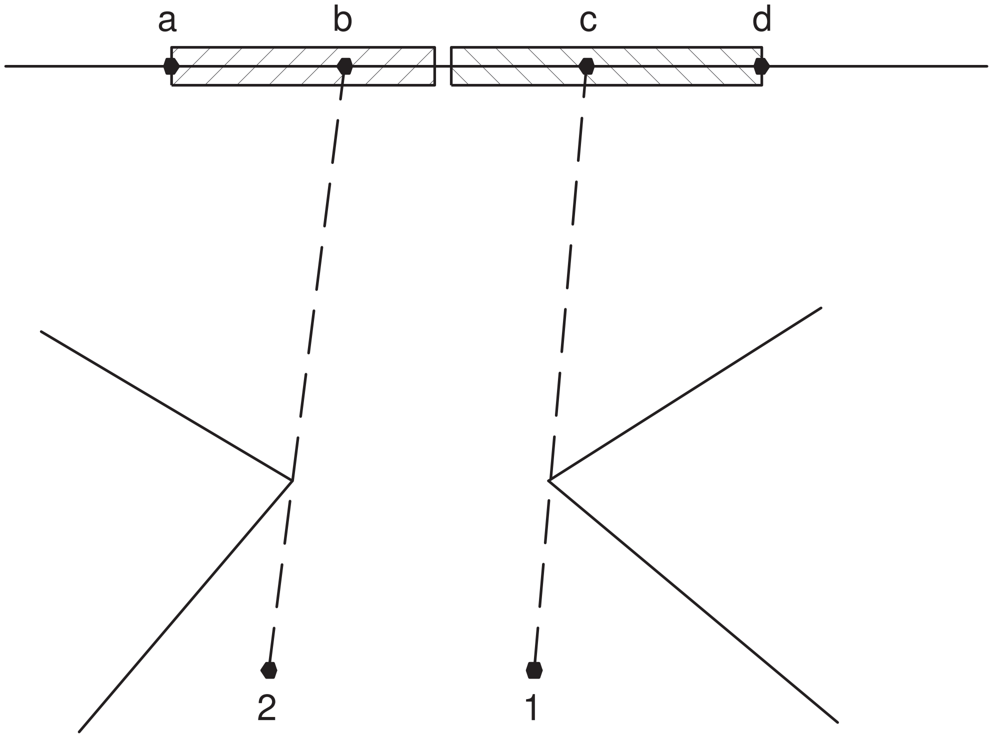
Τονίζουμε ότι στο σύνολο SOL δεν προστίθενται τα τμήματα $E'(v)$ που βλέπει η κορυφή v αλλά προστίθενται μόνο τα υποτμήματα των $E'(v)$ που καταλαμβάνουν οι πίνακες που τοποθετεί το Multiple Knapsack. Ο λόγος για τον οποίο η αύξηση του συνόλου SOL γίνεται με αυτόν τον τρόπο, φαίνεται στο σχήμα 4.1.

Ας υποθέσουμε ότι στην κορυφή 1 έχει ήδη τοποθετηθεί φύλακας από τον αλγόριθμο 7 σε προηγούμενο βήμα. Έστω ότι ένας πίνακας που είχε τοποθετηθεί σε εκείνο το βήμα ήταν αυτός που φαίνεται να καλύπτει το FVS τμήμα (a, b) και λίγο από το τμήμα (b, c). Εάν τότε είχε προστεθεί στη λύση ολόκληρο το τμήμα (b, c), τότε από την κορυφή 2 δεν θα μπορούσε να φαίνεται τοποθετημένος ο πίνακας που καλύπτει το τμήμα (c, d) και λίγο από το (b, c). Με άλλα λόγια, τα νέα τμήματα που φαίνονται από την κορυφή 2 θα υπολογίζονταν λιγότερα, χωρίς λόγο.

Τέλος ο αλγόριθμος ενημερώνει το σύνολο των πινάκων D, δηλαδή αφαιρεί εκείνους τους πίνακες που τοποθετήθηκαν.

Θα αποδείξουμε τώρα ότι ο αλγόριθμος 7 προσεγγίζει το πρόβλημα Maximum Value Vertex Guard with Painting Placement κατά ένα σταθερό προσεγγιστικό παράγοντα.

Έστω OPT η συλλογή των συνόλων των πινάκων σε μια βέλτιστη λύση και SOL η συλλογή των πινάκων που μας επέστρεψε ο αλγόριθμος. Έστω $W(OPT)$



Σχήμα 4.1:

και $W(SOL)$ οι αξίες των πινάκων της βέλτιστης λύσης και της λύσης του αλγόριθμου αντίστοιχα. Έστω P_i το i-οστό σύνολο των πινάκων που επιλέγεται από τον αλγόριθμο στο βήμα i. Θεωρούμε ότι

$$W(P_i) = W(\cup_{m=1}^{i} P_m) - W(\cup_{m=1}^{i-1} P_m)$$

Δηλαδή ότι το σύνολο P_i δεν περιέχει πίνακες που έχουν ήδη καλυφθεί σε προηγούμενα βήματα.

Αν P_i' είναι το σύνολο με τη μέγιστη αξία στο βήμα i, ισχύει ότι:

$$W(P_i) \geq \alpha W(P_i')$$

Δηλαδή στο βήμα i, ο κόμβος i που επιλέγεται από τον αλγόριθμο μπορεί να μην βλέπει τη μέγιστη δυνατή αξία από τους πίνακες που έχουν απομείνει, βλέπει όμως σίγουρα ένα κλάσμα αυτής που εξαρτάται από την προσέγγιση του Multiple Knapsack($\frac{1}{\alpha}$).

Λήμμα 4.1 *Μετά από l βήματα του αλγορίθμου 7 ισχύει:*

$$W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i) \geq \frac{\alpha}{k}(W(OPT) - W(\cup_{i=1}^{l-1} P_i)), l = 1, 2, ..., k$$

Απόδειξη. Η αξία των πινάκων που δεν έχουν καλυφθεί από τα $l-1$ σύνολα του αλγόριθμου, καλύπτεται από τα k σύνολα του OPT. Συνεπώς σύμφωνα με την αρχή του περιστερώνα, κάποιο από τα σύνολα του OPT (διαφορετικό από τα $P_i, i = 1, ..., l-1$), θα πρέπει να καλύπτει αξία τουλάχιστον:

$$\frac{W(OPT) - W(\cup_{i=1}^{l-1} P_i)}{k}$$

Το $W(P_l')$ που είναι το επόμενο στην ακολουθία της λύσης του αλγόριθμου, καλύπτει τη μεγαλύτερη δυνατή αξία από τους πίνακες που έχουν απομείνει, συνεπώς:

$$W(P_l') \geq \frac{W(OPT) - W(\cup_{i=1}^{l-1} P_i)}{k}$$

Ισχύει επίσης:

$$W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i) = W(P_l) \geq \alpha W(P_l')$$

Συνεπώς:

$$W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i) \geq \frac{\alpha}{k}(W(OPT) - W(\cup_{i=1}^{l-1} P_i))$$

□

Λήμμα 4.2 *Μετά από l βήματα του αλγορίθμου 7 ισχύει:*

$$W(\cup_{i=1}^{l} P_i) \geq (1 - (1 - \frac{\alpha}{k})^l) W(OPT), l = 1, ..., k$$

Απόδειξη.

Θα το αποδείξουμε με επαγωγή στο l.

Το πρώτο σύνολο που επιλέγει ο αλγόριθμος έχει αξία $W(P_1)$. Ισχύει:

$$W(P_1) \geq \alpha W(P_1')$$

Το $W(P_1')$ έχει τη μέγιστη δυνατή αξία από τα σύνολα που επιλέγει ο OPT. Συνεπώς από την αρχή του περιστερώνα:

$$W(P_1') \geq \frac{W(OPT)}{k} \rightarrow W(P_1) \geq \frac{\alpha}{k} W(OPT)$$

Έστω ότι ισχύει για $l-1$, δηλαδή

$$W(\cup_{i=1}^{l-1} P_i) \geq (1 - (1 - \frac{\alpha}{k})^{l-1}) W(OPT)$$

Έχουμε:

$$W(\cup_{i=1}^{l} P_i) = W(\cup_{i=1}^{l-1} P_i) + (W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i))$$

Από το προηγούμενο λήμμα ισχύει:

$$W(\cup_{i=1}^{l} P_i) \geq W(\cup_{i=1}^{l-1} P_i) + \frac{\alpha}{k}(W(OPT) - W(\cup_{i=1}^{l-1} P_i)) \rightarrow$$

$$W(\cup_{i=1}^{l} P_i) \geq W(\cup_{i=1}^{l-1} P_i)(1 - \frac{\alpha}{k}) + \frac{\alpha}{k} W(OPT)$$

Από την επαγωγική υπόθεση:

$$W(\cup_{i=1}^{l} P_i) \geq (1 - (1 - \frac{\alpha}{k})^{l-1}) W(OPT)(1 - \frac{\alpha}{k}) + \frac{\alpha}{k} W(OPT) \rightarrow$$

$$W(\cup_{i=1}^{l} P_i) \geq (1 - (1 - \frac{\alpha}{k})^{l}) W(OPT)$$

□

Θεώρημα 4.2 *Ο αλγόριθμος 7 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* $\frac{1}{1-\frac{1}{e^\alpha}}$ *για το πρόβλημα* MAXIMUM VALUE VERTEX GUARD WITH PAINTING PLACEMENT, *όπου* $\frac{1}{\alpha}$ *ο προσεγγιστικός παράγοντας του* MULTIPLE KNAPSACK.

Απόδειξη. Από το προηγούμενο λήμμα, θέτοντας $l = k$ έχουμε:

$$W(\cup_{i=1}^{k} P_i) \geq (1 - (1 - \frac{\alpha}{k})^{k}) W(OPT)$$

Ισχύει ότι:

$$lim_{k \to \infty} (1 - (1 - \frac{\alpha}{k})^{k}) = 1 - \frac{1}{e^\alpha}$$

Ο όρος $(1 - (1 - \frac{\alpha}{k})^k)$ φθίνει, συνεπώς:

$$1 - (1 - \frac{\alpha}{k})^k \geq 1 - \frac{1}{e^\alpha}$$

άρα

$$W(SOL) > (1 - \frac{1}{e^\alpha}) W(OPT)$$

Δηλαδή ο αλγόριθμος προσεγγίζει το πρόβλημα με παράγοντα $\frac{1}{1-\frac{1}{e^\alpha}}$. Λόγω της ύπαρξης PTAS για το πρόβλημα MULTIPLE KNAPSACK, το $\alpha \rightarrow 1$ και συνεπώς $\frac{1}{1-\frac{1}{e^\alpha}} \rightarrow 1.58$. □

Algorithm 8 MaxValueEdgeGuardWithPaintingPlacement (* greedy *)

```
compute FVS points
for every e ∈ E(P) do
  compute E'(e)
end for
SOL ← ∅
for i = 1 to k do
  select e ∈ E that maximizes W(multknap(E'(e) − SOL ∩ E'(e), D))
  updateD
end for
return W(SOL)
```

4.2 Φύλακες - πλευρές

Σαν συνέπεια της πρότασης 3.2 του κεφαλαίου 3 ισχύει:

Πρόταση 4.2 *Το* MAXIMUM VALUE EDGE GUARD WITH PAINTING PLACEMENT *είναι NP-hard.*

Παρακάτω φαίνεται ο αλγόριθμος 8 ο οποίος προσεγγίζει το πρόβλημα MAXIMUM VALUE EDGE GUARD WITH PAINTING PLACEMENT.

Παρόμοια λοιπόν με το θεώρημα 4.2 ισχύει:

Θεώρημα 4.3 *Ο αλγόριθμος 8 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* 1.58 *για το πρόβλημα* MAXIMUM VALUE EDGE GUARD WITH PAINTING PLACEMENT.

4.3 Πολύγωνα με τρύπες

Παρόμοια αποτελέσματα ισχύουν και στην περίπτωση πολυγώνων με τρύπες. Οι αλγόριθμοι 7 και 8 μπορούν να εφαρμοστούν και σε πολύγωνα με τρύπες πετυχαίνοντας την ίδια προσέγγιση.

Κεφάλαιο 5

Κάλυψη μέγιστου μήκους με όριο κόστους πομποδεκτών

5.1 Πομποδέκτες σε κορυφές

Έστω ένα πολύγωνο P χωρίς τρύπες και ένας αριθμός $B > 0$. Κάθε κορυφή v_i του P έχει ένα κόστος c_i. Θέλουμε να τοποθετήσουμε φύλακες στις κορυφές του πολυγώνου έτσι ώστε:

- $\sum c_i$ των κορυφών που έχουν φύλακα να είναι $\leq B$
- Το μήκος της περιμέτρου του πολυγώνου που φαίνεται από τους φύλακες να είναι μέγιστο.

Αυτή η περίπτωση είναι γενίκευση της περίπτωσης όπου κάθε κόμβος έχει κόστος 1 την οποία εξετάσαμε στο κεφάλαιο 2.

Πρόταση 5.1 *Το πρόβλημα* BUDGETED MAXIMUM LENGTH VERTEX GUARD *είναι NP-hard.*

Απόδειξη. Είναι γενίκευση του προβλήματος MAXIMUM LENGTH VERTEX GUARD το οποίο είναι NP-hard (βλ. κεφ 2 πρόταση 2.1). □

Ο αλγόριθμος 1 που δώσαμε στο κεφ. 2 για το πρόβλημα MAXIMUM LENGTH VERTEX GUARD όπως φαίνεται εύκολα δεν οδηγεί αναγκαστικά σε εύρεση προσεγγιστικής λύσης με σταθερό παράγοντα. Δίνουμε λοιπόν εδώ ένα νέο αλγόριθμο που πετυχαίνει σταθερό προσεγγιστικό παράγοντα για το πρόβλημα BUDGETED MAXIMUM LENGTH VERTEX GUARD.

Οι προκαταρκτικοί υπολογισμοί είναι ίδιοι όπως και στον αλγόριθμο 1 του κεφ. 2. Δηλαδή κατακερματίζουμε την περίμετρο του πολυγώνου χρησιμοποιώντας την **FVS** κατασκευή και υπολογίζουμε τα σύνολα $E'(v)$ για κάθε $v \in V(P)$. Ο νέος αυτός greedy αλγόριθμος σε κάθε βήμα επιλέγει μια κορυφή σαν υποψήφια για την τοποθέτηση φύλακα. Το κριτήριο με το οποίο επιλέγει την κορυφή είναι η μεγιστοποίηση του λόγου του μήκους του καινούριου μέρους της περιμέτρου που φαίνεται από αυτήν την κορυφή προς το κόστος της κορυφής. Εάν αυτό το κόστος μπορεί να καλυφθεί από τον δεδομένο προϋπολογισμό, τότε ο αλγόριθμος τοποθετεί φύλακα στην κορυφή. Σε αντίθετη περίπτωση η κορυφή απορρίπτεται. Με αυτόν τον τρόπο εξετάζονται μία - μία όλες οι κορυφές του πολυγώνου.

Τέλος ο αλγόριθμος 9 συγκρίνει το συνολικό μήκος που φαίνεται με βάση τις επιλογές που έχει κάνει με το μέγιστο μήκος που μπορεί να φαίνεται από μια κορυφή και επιστρέφει σαν λύση το μεγαλύτερο από αυτά τα δύο.

Algorithm 9 BudgetedMaxLegthVertexGuards (* greedy *)

```
compute FVS points
for every v ∈ V(P) do
  compute E'(v)
end for
SOL ← ∅
G ← ∅
C_TOT ← 0
M ← V
repeat
  select v_i ∈ V that maximizes L(G∪E'(v_i))/c_i
  if C_TOT + c_i ≤ B then
    G ← G ∪ E'(v_i)
    C_TOT ← C_TOT + c_i
  end if
  M ← M − {v_i}
until M = ∅
select v_t ∈ V that maximizes L(E'(v_t))
if L(G) ≥ L(E'(v_t)) then
  SOL ← G
else
  SOL ← E'(v_t)
end if
return L(SOL)
```

Παρακάτω αποδεικνύουμε ότι ο αλγόριθμος 9 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει σταθερό προσεγγιστικό παράγοντα για το πρόβλημα Budgeted Maximum Length Vertex Guard.

Έστω OPT η συλλογή των συνόλων των τμημάτων FVS που ανήκουν στη βέλτιστη λύση και SOL η συλλογή που μας επέστρεψε ο αλγόριθμος 9. Έστω r ο αριθμός των επαναλήψεων που εκτελέστηκαν στον αλγόριθμο μέχρι που το πρώτο σύνολο S_{l+1} από τη συλλογή OPT εμφανίστηκε αλλά δεν προστέθηκε στη SOL γιατί η πρόσθεσή του θα οδηγούσε σε κόστος μεγαλύτερο του B. Θεωρούμε:

$$L(S_k) = L(\cup_{i=1}^{k} S_i) - L(\cup_{i=1}^{k-1} S_i)$$

Δηλαδή το σύνολο S_k δεν περιέχει τυχόν τμήματα που έχουμε ήδη καλύψει. Μερικές φορές για ευκολία καλούμε κόστος ενός συνόλου με τμήματα το κόστος της κορυφής που τα βλέπει.

Λήμμα 5.1 *Μετά από l βήματα του αλγορίθμου 9 ισχύει:*

$$L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i) \geq \frac{c_l}{B}(L(OPT) - L(\cup_{i=1}^{l-1} S_i))$$

Απόδειξη. Για κάθε σύνολο $S_k \in OPT - \cup_{i=1}^{l-1} S_i$ ισχύει ότι

$$\frac{L(S_k)}{c_k} \leq \frac{L(S_l)}{c_l}$$

το συνολικό κόστος των συνόλων στο $OPT - \cup_{i=1}^{l-1} S_i$ είναι το πολύ B. Το συνολικό μήκος των τμημάτων που ανήκουν στα σύνολα $OPT - \cup_{i=1}^{l-1} S_i$ είναι:

$$L(\cup_{i=m}^{k} S_i')$$

όπου S_i' είναι τα σύνολα που ανήκουν στη βέλτιστη λύση. Αφού λοιπόν

$$\frac{L(S_m')}{c_m} \leq \frac{L(S_l)}{c_l}$$

ισχύει:

$$L(\cup_{i=m}^{k} S_i') \leq c_m \frac{L(S_l)}{c_l} + c_{m+1} \frac{L(S_l)}{c_l} + ... + c_{m+k} \frac{L(S_l)}{c_l} \leq B \frac{L(S_l)}{c_l}$$

συνεπώς

$$L(OPT) - L(\cup_{i=1}^{l-1} S_i) \leq B \frac{L(S_l)}{c_l}$$

ισχύει επίσης

$$L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i) = L(S_l)$$

επομένως:

$$L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i) \geq \frac{c_l}{B}(L(OPT) - L(\cup_{i=1}^{l-1} S_i))$$

□

Λήμμα 5.2 *Μετά από l βήματα του αλγορίθμου 9 ισχύει:*

$$L(\cup_{i=1}^{l} S_i) \geq (1 - \prod_{k=1}^{l} (1 - \frac{c_k}{B}))L(OPT)$$

Απόδειξη. Θα το αποδείξουμε με επαγωγή στο l. Αρχικά αποδεικνύουμε ότι υπάρχει τουλάχιστον ένα σύνολο του οποίου ο λόγος του μήκους των τμημάτων του προς το κόστος του είναι τουλάχιστον $\frac{L(OPT)}{B}$.

Το παραπάνω ισχύει, διότι αν υποθέσουμε πως δεν ισχύει, τότε το συνολικό μήκος των τμημάτων που ανήκουν στη βέλτιστη λύση είναι:

$$L(\cup_{i=1}^{t} S_i) \leq L(S_1) + ... + L(S_t) =$$

$$c_1 \frac{L(S_1)}{c_1} + ... + c_t \frac{L(S_t)}{c_t}$$

και λόγω της υπόθεσης που κάναμε:

$$c_1 \frac{L(S_1)}{c_1} + ... + c_t \frac{L(S_t)}{c_t} <$$

$$c_1 \frac{L(OPT)}{B} + ... + c_t \frac{L(OPT)}{B} \leq B \frac{L(OPT)}{B} = L(OPT)$$

δηλαδή

$$L(\cup_{i=1}^{t} S_i) < L(OPT)$$

ΑΤΟΠΟ. Συνεπώς υπάρχει τουλάχιστον ένα σύνολο με λόγο μήκους προς κόστος τουλάχιστον $\frac{L(OPT)}{B}$. Το πρώτο σύνολο που επιλέγει ο αλγόριθμος έχει το μέγιστο λόγο μήκους προς κόστος, συνεπώς:

$$L(S_1) \geq \frac{c_1}{B} L(OPT)$$

έστω ότι ισχύει για $l-1$, δηλαδή:

$$L(\cup_{i=1}^{l-1} S_i) \geq (1 - \prod_{k=1}^{l-1} (1 - \frac{c_k}{B})) L(OPT)$$

έχουμε:

$$L(\cup_{i=1}^{l} S_i) = L(\cup_{i=1}^{l-1} S_i) + (L(\cup_{i=1}^{l} S_i) - L(\cup_{i=1}^{l-1} S_i))$$

από το προηγούμενο λήμμα:

$$L(\cup_{i=1}^{l} S_i) \geq L(\cup_{i=1}^{l-1} S_i) + \frac{c_l}{B}(L(OPT) - L(\cup_{i=1}^{l-1} S_i)) \rightarrow$$

$$L(\cup_{i=1}^{l} S_i) \geq L(\cup_{i=1}^{l-1} S_i)(1 - \frac{c_l}{B}) + \frac{c_l}{B} L(OPT)$$

από την επαγωγική υπόθεση:

$$L(\cup_{i=1}^{l} S_i) \geq (1 - \prod_{k=1}^{l-1} (1 - \frac{c_k}{B})) L(OPT)(1 - \frac{c_l}{B}) + \frac{c_l}{B} L(OPT) \rightarrow$$

$$L(\cup_{i=1}^{l} S_i) \geq (1 - \prod_{k=1}^{l} (1 - \frac{c_k}{B})) L(OPT)$$

□

Θεώρημα 5.1 *Ο αλγόριθμος 9 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* $\frac{2}{1-\frac{1}{e}} = 3.16$ *για το πρόβλημα* Budgeted Maximum Length Vertex Guard.

Απόδειξη. Από το προηγούμενο λήμμα ισχύει:

$$L(\cup_{i=1}^{l+1} S_i) \geq (1 - \prod_{k=1}^{l+1}(1 - \frac{c_k}{B}))L(OPT)$$

$$\geq (1 - \prod_{k=1}^{l+1}(1 - \frac{c_k}{\sum_{i=1}^{l+1} c_i}))L(OPT), \alpha\phi o \sum_{i=1}^{l+1} c_i > B$$

Η συνάρτηση

$$(1 - \prod_{i=1}^{n}(1 - \frac{a_i}{A}))$$

όπου $\sum_{i=1}^{n} a_i = A$ έχει minimum όταν $a_1 = a_2 = ... = a_n = \frac{A}{n}$. Συνεπώς

$$L(\cup_{i=1}^{l+1} S_i) \geq (1 - (1 - \frac{1}{l+1})^{l+1})L(OPT) \geq (1 - \frac{1}{e})L(OPT)$$

ισχύει $L(S_{l+1}) \leq L(S_t)$, αφού το $L(S_t)$ είναι το μέγιστο μήκος που μπορεί να δει μια κορυφή άρα

$$L(\cup_{i=1}^{l} S_i) + L(S_t) \geq L(\cup_{i=1}^{l} S_i) + L(S_{l+1}) \geq (1 - \frac{1}{e})L(OPT)$$

συνεπώς μια από τις δύο ποσότητες $L(\cup_{i=1}^{l} S_i)$ ή $L(S_t)$ είναι μεγαλύτερη ή ίση του

$$\frac{1}{2}(1 - \frac{1}{e})L(OPT)$$

Επομένως ο αλγόριθμος προσεγγίζει το πρόβλημα πετυχαίνοντας παράγοντα $\frac{2}{1-\frac{1}{e}} = 3.16$.

□

5.2 Πομποδέκτες - πλευρές

Η περίπτωση τοποθέτησης φυλάκων σε ακμές είναι γενίκευση του προβλήματος Maximum Length Edge Guard που εξετάσαμε στην παράγραφο 2.2 του κεφ. 2.

Πρόταση 5.2 *Το πρόβλημα* Budgeted Maximum Length Edge Guard *είναι NP-hard.*

Απόδειξη. Η περίπτωση όπου κάθε κορυφή έχει κόστος 1 είναι NP-hard (βλ. κεφ. 2 πρόταση 2.2). □

Ο αλγόριθμος 10 που προσεγγίζει το πρόβλημα Budgeted Maximum Length Edge Guard αρχικά υπολογίζει τα σύνολα $E'(e)$ για κάθε e όπως ακριβώς στην παράγραφο 2.2 του κεφ. 2. Το κυρίως μέρος του είναι παρόμοιο με εκείνο του αλγορίθμου 9 με τη διαφορά ότι τώρα οι φύλακες τοποθετούνται σε πλευρές.

Algorithm 10 BudgetedMaxLegthEdgeGuards (* greedy *)

compute FVS points
for every $e \in E(P)$ **do**
  compute $E'(e)$
end for
$SOL \leftarrow \emptyset$
$G \leftarrow \emptyset$
$C_{TOT} \leftarrow 0$
$M \leftarrow E$
repeat
  select $e_i \in E$ that maximizes $\frac{L(G \cup E'(e_i))}{c_i}$
  if $C_{TOT} + c_i \leq B$ **then**
    $G \leftarrow G \cup E'(e_i)$
    $C_{TOT} \leftarrow C_{TOT} + c_i$
  end if
  $M \leftarrow M - \{e_i\}$
until $M = \emptyset$
select $e_t \in E$ that maximizes $L(e_t)$
if $L(G) \geq L(e_t)$ **then**
  $SOL \leftarrow G$
else
  $SOL \leftarrow \{e_t\}$
end if
return $L(SOL)$

Θεώρημα 5.2 *Ο αλγόριθμος 10 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* $\frac{2}{1-\frac{1}{e}} = 3.16$ *για το πρόβλημα* Budgeted Maximum Length Edge Guard.

5.3 Πολύγωνα με τρύπες

Τα αποτελέσματα των παραγράφων 5.1 και 5.2 ισχύουν και στην περίπτωση πολυγώνων με τρύπες.

Πρόταση 5.3 *Το πρόβλημα* Budgeted Maximum Length Vertex Guard *σε πολύγωνα με τρύπες είναι NP-hard.*

Πρόταση 5.4 *Το πρόβλημα* Budgeted Maximum Length Edge Guard *είναι NP-hard.*

Οι αλγόριθμοι 9 και 10 μπορούν να χρησιμοποιηθούν στα προβλήματα που αφορούν πολύγωνα με τρύπες πετυχαίνοντας παράγοντα $\frac{2}{1-\frac{1}{e}} = 3.16$.

Για μια συζήτηση σχετικά με το ενδιαφέρον που έχουν τα πολύγωνα με τρύπες στα συγκεκριμένα προβλήματα βλ. κεφ 2 παρ. 2.3.

Κεφάλαιο 6

Κάλυψη μέγιστης αξίας με όριο κόστους πομποδεκτών

6.1 Πομποδέκτες σε κορυφές

Έστω ένα πολύγωνο P και ένας αριθμός $B > 0$. Η περίμετρος του P είναι χωρισμένη σε ξένα μεταξύ τους διαστήματα στα οποία έχουν αποδοθεί βάρη. Κάθε κορυφή v_i του P έχει ένα κόστος c_i. Θέλουμε να τοποθετήσουμε φύλακες στις κορυφές του πολυγώνου έτσι ώστε:

- $\sum c_i$ των κορυφών που έχουν φύλακα να είναι $\leq B$
- Το συνολικό βάρος των διαστημάτων που φαίνονται από τους φύλακες να είναι μέγιστο.

Αυτό το πρόβλημα είναι γενίκευση του προβλήματος Maximum Value Vertex Guard που εξετάσαμε στο κεφάλαιο 3.

Πρόταση 6.1 *Το πρόβλημα* Budgeted Maximum Value Vertex Guard *είναι NP-hard.*

Απόδειξη. Είναι γενίκευση του προβλήματος Maximum Value Vertex Guard το οποίο είναι NP-hard (βλ. κεφ. 3, πρόταση 3.1) □

Ο αλγόριθμος 3 που δώσαμε για το πρόβλημα Maximum Value Vertex Guard στο κεφάλαιο 3, φαίνεται εύκολα πως δεν οδηγεί αναγκαστικά σε εύρεση προσεγγιστικής λύσης με σταθερό παράγοντα του προβλήματος Budgeted Maximum Value Vertex Guard.

Ο νέος αλγόριθμος που δίνουμε εδώ για το πρόβλημα Budgeted Maximum Value Vertex Guard έχει τους ίδιους προκαταρκτικούς υπολογισμούς με τον αλγόριθμο 3 του κεφ. 3 (υπολογισμό των συνόλων $S(v)$). Στο υπόλοιπο μέρος του είναι παρόμοιος με τον αλγόριθμο 9 του κεφ. 5 με τη διαφορά ότι αντί για L τώρα έχουμε Weight.

Algorithm 11 BudgetedMaxValueVertexGuards (* greedy *)

for every $v \in V(P)$ **do**
  compute $S(v)$
end for
$SOL \leftarrow \emptyset$
$G \leftarrow \emptyset$
$C_{TOT} \leftarrow 0$
$M \leftarrow V$
repeat
  select $v_i \in V$ that maximizes $\frac{W(G \cup S(v_i))}{c_i}$
  if $C_{TOT} + c_i \leq B$ **then**
    $G \leftarrow G \cup S(v_i)$
    $C_{TOT} \leftarrow C_{TOT} + c_i$
  end if
  $M \leftarrow M - \{v_i\}$
until $M = \emptyset$
select $v_t \in V$ that maximizes $W(S(v_t))$
if $W(G) \geq W(S(v_t))$ **then**
  $SOL \leftarrow G$
else
  $SOL \leftarrow S(v_t)$
end if
return $W(SOL)$

Θυμίζουμε ότι στο σύνολο $S(v)$ ανήκουν εκείνα τα δεδομένα διαστήματα, τα οποία φαίνονται ολόκληρα από την κορυφή v. Για μια περιγραφή του αλγορίθμου 11 δες αλγόριθμο 9 του κεφ. 5 βάζοντας αντί για μήκος το βάρος.

Θεώρημα 6.1 *Ο αλγόριθμος 11 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* $\frac{2}{1-\frac{1}{e}} = 3.16$ *για το πρόβλημα* BUDGETED MAXIMUM VALUE VERTEX GUARD.

Απόδειξη. Βλέπε απόδειξη θεωρήματος 5.1, αλλάζοντας το Length με Weight. □

6.2 Πομποδέκτες - πλευρές

Το πρόβλημα BUDGETED MAXIMUM VALUE EDGE GUARD είναι γενίκευση του προβλήματος MAXIMUM VALUE EDGE GUARD του κεφ. 3.

Πρόταση 6.2 *Το πρόβλημα* BUDGETED MAXIMUM VALUE EDGE GUARD *είναι NP-hard.*

Απόδειξη. Η περίπτωση όπου κάθε πλευρά έχει κόστος 1 είναι NP-hard (βλ. κεφ. 3 πρόταση 3.2). □

Ο αλγόριθμος 12 που πετυχαίνει παράγοντα προσέγγισης $\frac{2}{1-\frac{1}{e}} = 3.16$ είναι παρόμοιος με τον προηγούμενο.

6.3 Πολύγωνα με τρύπες

Παρόμοια αποτελέσματα ισχύουν σε πολύγωνα με τρύπες στα οποία μπορούν να εφαρμοστούν οι ίδιοι αλγόριθμοι. Προφανώς το πρόβλημα BUDGETED MAXIMUM VALUE VERTEX/EDGE GUARD δεν επιδέχεται FPTAS λόγω του θεωρήματος 3.2 του κεφ. 3.

6.4 Μερική κάλυψη

Ανάλογα αποτελέσματα έχουμε και στην περίπτωση που ο σκοπός είναι η επίβλεψη αντί της κάλυψης των διαστημάτων.

Πρόταση 6.3 *Το* BUDGETED MAXIMUM VALUE VERTEX/EDGE GUARD *(watching version) είναι NP-hard.*

Απόδειξη. Βλέπε κεφ. 3 πρόταση 3.1 και πρόταση 3.2. □

Οι προσεγγιστικοί αλγόριθμοι 13 και 14 είναι παρόμοιοι με τους προηγούμενους χρησιμοποιώντας τους προκαταρκτικούς υπολογισμούς των αλγορίθμων 5 και 6 του κεφ. 3.

Τα αποτελέσματα αυτά ισχύουν και σε πολύγωνα με τρύπες.

Λόγω του θεωρήματος 3.2του κεφ. 3, τα προβλήματα BUDGETED MAXIMUM VALUE VERTEX/EDGE GUARD (watching version) δεν επιδέχονται FPTAS.

Algorithm 12 BudgetedMaxValueEdgeGuards (* greedy *)

compute FVS points
for every $e \in E(P)$ **do**
  compute $S(e)$
end for
$SOL \leftarrow \emptyset$
$G \leftarrow \emptyset$
$C_{TOT} \leftarrow 0$
$M \leftarrow E)$
repeat
  select $e_i \in E$ that maximizes $\frac{W(G \cup S(e_i))}{c_i}$
  if $C_{TOT} + c_i \leq B$ **then**
    $G \leftarrow G \cup S(e_i)$
    $C_{TOT} \leftarrow C_{TOT} + c_i$
  end if
  $M \leftarrow M - \{e_i\}$
until $M = \emptyset$
select $e_t \in E$ that maximizes $W(S(e_t))$
if $W(G) \geq W(S(e_t))$ **then**
  $SOL \leftarrow G$
else
  $SOL \leftarrow S(e_t)$
end if
return $W(SOL)$

Algorithm 13 BudgetedMaxValueVertexGuards (watching) (* greedy *)

compute FVS points
for every $v \in V(P)$ **do**
  compute $R(v)$
end for
$SOL \leftarrow \emptyset$
$G \leftarrow \emptyset$
$C_{TOT} \leftarrow 0$
$M \leftarrow V$
repeat
  select $v_i \in V$ that maximizes $\frac{W(G \cup R(v_i))}{c_i}$
  if $C_{TOT} + c_i \leq B$ **then**
    $G \leftarrow G \cup R(v_i)$
    $C_{TOT} \leftarrow C_{TOT} + c_i$
  end if
  $M \leftarrow M - \{v_i\}$
until $M = \emptyset$
select $e_t \in E$ that maximizes $W(R(v_t))$
if $W(G) \geq W(R(v_t))$ **then**
  $SOL \leftarrow G$
else
  $SOL \leftarrow R(v_t)$
end if
return $W(SOL)$

Algorithm 14 BudgetedMaxValueEdgeGuards (watching) (* greedy *)

compute FVS points
for every $e \in E(P)$ **do**
  compute $R(e)$
end for
$SOL \leftarrow \emptyset$
$G \leftarrow \emptyset$
$C_{TOT} \leftarrow 0$
$M \leftarrow E$
repeat
  select $e_i \in E$ that maximizes $\frac{W(G \cup R(e_i))}{c_i}$
  if $C_{TOT} + c_i \leq B$ **then**
    $G \leftarrow G \cup R(e_i)$
    $C_{TOT} \leftarrow C_{TOT} + c_i$
  end if
  $M \leftarrow M - \{e_i\}$
until $M = \emptyset$
select $e_t \in E$ that maximizes $W(R(e_t))$
if $W(G) \geq W(R(e_t))$ **then**
  $SOL \leftarrow G$
else
  $SOL \leftarrow R(e_t)$
end if
return $W(SOL)$

Κεφάλαιο 7

Πομποδέκτες και αναμεταδότες

7.1 Πομποδέκτες σε κορυφές

Σε αυτό το κεφάλαιο εξετάζουμε τη γενίκευση του προβλήματος του κεφ. 4. Τώρα κάθε κορυφή έχει ένα κόστος και ζητάμε να τοποθετήσουμε φύλακες στις κορυφές χωρίς να ξεπεράσουμε έναν δεδομένο προϋπολογισμό B.

Το πρόβλημα είναι NP-hard σαν συνέπεια της πρότασης 7.1 του κεφ. 4.

Πρόταση 7.1 *Το πρόβλημα* Budgeted Maximum Value Vertex Guard with Painting Placement *είναι NP-hard.*

Ο αλγόριθμος που προσεγγίζει το πρόβλημα φαίνεται παρακάτω.

Ο αλγόριθμος 15 χρησιμοποιεί τη μέθοδο του προηγούμενου κεφαλαίου, εφαρμόζοντας σε κάθε βήμα τον αλγόριθμο που προσεγγίζει το πρόβλημα Multiple Knapsack, όπως στο κεφάλαιο 4. Παρακάτω αποδεικνύουμε την προσέγγιση που πετυχαίνει ο αλγόριθμος.

Έστω OPT η συλλογή των συνόλων των πινάκων που ανήκουν στη βέλτιστη λύση και SOL η συλλογή των πινάκων που μας επέστρεψε ο αλγόριθμος 15. Έστω r ο αριθμός των επαναλήψεων που εκτελέστηκαν στον αλγόριθμο μέχρι που το πρώτο σύνολο P_{l+1} από τη συλλογή OPT εμφανίστηκε αλλά δεν προστέθηκε στη SOL γιατί η πρόσθεσή του θα οδηγούσε σε κόστος μεγαλύτερο του B. Θεωρούμε:

$$W(P_k) = W(\cup_{i=1}^{k} P_i) - W(\cup_{i=1}^{k-1} P_i)$$

Δηλαδή το σύνολο P_k δεν περιέχει τυχόν πίνακες που έχουμε ήδη καλύψει. Ισχύει ότι:

$$W(P_i) \geq \alpha W(P_i')$$

όπου P_i' είναι το σύνολο με τη μέγιστη δυνατή αξία στο βήμα i. Δηλαδή στο βήμα i, ο κόμβος i που επιλέγεται από τον αλγόριθμο μπορεί να μην βλέπει τη μέγιστη δυνατή αξία από τους πίνακες που έχουν απομείνει, βλέπει όμως σίγουρα ένα κλάσμα αυτής που εξαρτάται από την προσέγγιση του Multiple Knapsack($\frac{1}{\alpha}$).

Algorithm 15 BudgetedMaxValueVertexGuardWithPaintingPlacement (* greedy *)

compute FVS points
for every $v \in V(P)$ **do**
  compute $E'(v)$
end for
$SOL \leftarrow \emptyset$
$G \leftarrow \emptyset$
$C_{TOT} \leftarrow 0$
$M \leftarrow V$
repeat
  select $v_i \in V$ that maximizes $\frac{W(multknap(E'(v_i)-G \cap E'(v_i),D))}{c_i}$
  if $C_{TOT} + c_i \leq B$ **then**
    $G \leftarrow G \cup L(P_i)$ $\{P_i$ is the set of the new overseen paintings$\}$
    $D \leftarrow D - P_i$
    $C_{TOT} \leftarrow C_{TOT} + c_i$
  end if
  $M \leftarrow M - \{v_i\}$
until $M = \emptyset$
select $v_t \in V$ that maximizes $W(multknap(E'(v_t), D))$
if $W(G) \geq W(multknap(E'(v_t), D))$ **then**
  $SOL \leftarrow G$
else
  $SOL \leftarrow multknap(E'(v_t), D)$
end if
return $W(SOL)$

Λήμμα 7.1 *Μετά από l βήματα του αλγόριθμου 15 ισχύει:*

$$W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i) \geq \frac{c_l a}{B}(w(OPT) - W(\cup_{i=1}^{l-1} P_i))$$

Απόδειξη. Για κάθε σύνολο $P_k \in OPT - \cup_{i=1}^{l-1} P_i$ ισχύει ότι

$$\frac{W(P_k)}{c_k} \leq \frac{W(P_l)}{c_l}$$

το συνολικό κόστος των συνόλων στο $OPT - \cup_{i=1}^{l-1} P_i$ είναι το πολύ B.

Το συνολικό βάρος των πινάκων που ανήκουν στα σύνολα $OPT - \cup_{i=1}^{l-1} P_i$ είναι:

$$B\frac{W(\cup_{i=1}^{l} P_i)}{c_l}$$

άρα

$$W(OPT) - W(\cup_{i=1}^{l-1} P_i) \leq B\frac{W(P_l)}{c_l}$$

$$W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i) \geq aW(P_l)$$

συνεπώς

$$W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i) \geq \frac{c_l a}{B}(W(OPT) - W(\cup_{i=1}^{l-1} P_i))$$

□

Λήμμα 7.2 *Μετά από l βήματα του αλγορίθμου 15 ισχύει:*

$$W(\cup_{i=1}^{l} P_i) \geq (1 - \prod_{k=1}^{l}(1 - \frac{ac_k}{B}))W(OPT)$$

Απόδειξη. Θα το αποδείξουμε με επαγωγή στο l. Ισχύει

$$W(P_1) \geq \frac{ac_1}{B}W(OPT)$$

Έστω ότι ισχύει για $l-1$, δηλαδή:

$$W(\cup_{i=1}^{l-1} P_i) \geq (1 - \prod_{k=1}^{l-1}(1 - \frac{ac_k}{B}))W(OPT)$$

έχουμε:

$$W(\cup_{i=1}^{l} P_i) = W(\cup_{i=1}^{l-1} P_i) + (W(\cup_{i=1}^{l} P_i) - W(\cup_{i=1}^{l-1} P_i))$$

από το προηγούμενο λήμμα:

$$W(\cup_{i=1}^{l} P_i) \geq W(\cup_{i=1}^{l-1} P_i) + \frac{ac_l}{B}(W(OPT) - W(\cup_{i=1}^{l-1} P_i)) \rightarrow$$

$$W(\cup_{i=1}^{l} P_i) \geq W(\cup_{i=1}^{l-1} P_i)(1 - \frac{ac_l}{B}) + \frac{ac_l}{B} W(OPT)$$

από την επαγωγική υπόθεση:

$$W(\cup_{i=1}^{l} P_i) \geq (1 - \prod_{k=1}^{l-1}(1 - \frac{ac_k}{B}))W(OPT)(1 - \frac{ac_l}{B}) + \frac{ac_l}{B} W(OPT) \rightarrow$$

$$W(\cup_{i=1}^{l} P_i) \geq (1 - \prod_{k=1}^{l}(1 - \frac{ac_k}{B}))W(OPT)$$

□

Θεώρημα 7.1 *Ο αλγόριθμος 15 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* $\frac{2}{1-\frac{1}{e^a}} \rightarrow \frac{2}{1-\frac{1}{e}} = 3.16$ *για το πρόβλημα* Budgeted Maximum Value Vertex Guard with Painting Placement.

Απόδειξη. Από το προηγούμενο λήμμα ισχύει:

$$W(\cup_{i=1}^{l+1} P_i) \geq (1 - \prod_{k=1}^{l+1}(1 - \frac{ac_k}{B}))W(OPT)$$

$$\geq (1 - \prod_{k=1}^{l+1}(1 - \frac{ac_k}{\sum_{i=1}^{l+1} c_i}))W(OPT), \alpha\phi o \sum_{i=1}^{l+1} c_i > B$$

η συνάρτηση

$$(1 - \prod_{i=1}^{n}(1 - \frac{a_i}{A}))$$

όπου $\sum_{i=1}^{n} a_i = A$ έχει minimum όταν $a_1 = a_2 = ... = a_n = \frac{A}{n}$. Συνεπώς

$$W(\cup_{i=1}^{l+1} P_i) \geq (1 - (1 - \frac{a}{l+1})^{l+1})W(OPT) \geq (1 - \frac{1}{e^a})W(OPT)$$

ισχύει $W(P_{l+1}) \leq W(P_t)$, άρα

$$W(\cup_{i=1}^{l} P_i) + W(P_t) \geq W(\cup_{i=1}^{l} W_i) + W(P_{l+1}) \geq (1 - \frac{1}{e^a})W(OPT)$$

συνεπώς μια από τις δύο ποσότητες $W(\cup_{i=1}^{l} P_i)$ ή $W(P_t)$ είναι μεγαλύτερη ή ίση του

$$\frac{1}{2}(1 - \frac{1}{e^a})W(OPT)$$

Δηλαδή ο αλγόριθμος πετυχαίνει παράγοντα προσέγγισης $\frac{2}{1-\frac{1}{e^a}} \rightarrow \frac{2}{1-\frac{1}{e}} = 3.16$ για το πρόβλημα Budgeted Maximum Value Vertex Guard with Painting Placement.

□

Algorithm 16 BudgetedMaxValueEdgeGuardWithPaintingPlacement (* greedy *)

compute FVS points
for every $e \in E(P)$ **do**
  compute $E'(e)$
end for
$SOL \leftarrow \emptyset$
$G \leftarrow \emptyset$
$C_{TOT} \leftarrow 0$
$M \leftarrow E$
repeat
  select $e_i \in E$ that maximizes $\frac{W(multknap(E'(e_i)-G\cap E'(e_i),D))}{c_i}$
  if $C_{TOT} + c_i \leq B$ **then**
    $G \leftarrow G \cup L(P_i)$ $\{P_i$ is the set of the new overseen paintings$\}$
    $D \leftarrow D - P_i$
    $C_{TOT} \leftarrow C_{TOT} + c_i$
  end if
  $M \leftarrow M - \{e_i\}$
until $M = \emptyset$
select $e_t \in E$ that maximizes $W(multknap(E'(e_t), D))$
if $W(G) \geq W(multknap(E'(e_t), D))$ **then**
  $SOL \leftarrow G$
else
  $SOL \leftarrow multknap(E'(e_t), D)$
end if
return $W(SOL)$

7.2 Πομποδέκτες - πλευρές

Παρόμοια αποτελέσματα με αυτά της προηγούμενης παραγράφου, έχουμε και στην περίπτωση της τοποθέτησης φυλάκων στις πλευρές. Ο προσεγγιστικός αλγόριθμος 16 δίνεται παρακάτω.

Επίσης όπως και στις προηγούμενες περιπτώσεις οι αλγόριθμοι αυτοί μπορούν να εφαρμοστούν και σε πολύγωνα με τρύπες πετυχαίνοντας την ίδια προσέγγιση.

Κεφάλαιο 8

Η κάλυψη της επιφάνειας του πολυγώνου

8.1 Πομποδέκτες σε κορυφές του πολυγώνου

Θα εξετάσουμε τώρα το πρόβλημα Maximum Area Vertex Guard. Έστω ένα πολύγωνο P και ένας αριθμός $k > 0$. Ο σκοπός είναι να τοποθετήσουμε k πομποδέκτες στις κορυφές του πολυγώνου έτσι ώστε να καλύπτεται το μεγαλύτερο μέρος του εσωτερικού του.

Κάνουμε έναν κατακερματισμό του εσωτερικού του πολυγώνου όπως στο κεφάλαιο 1 ενότητα 2.

Οποιαδήποτε FVS περιοχή a που ορίζεται από τις τεμνόμενες προεκτάσεις των ακμών του $V_G(P)$ είναι ορατή από κάποια κορυφή $v \in V(P)$ αν και μόνο αν επιβλέπεται από τη v.

Έτσι λοιπόν αν στον αλγόριθμο 1 αντικαταστήσουμε, όπου $E'(v_i)$ το $A'(v_i)$, το οποίο θα περιέχει τις επιφάνειες που φαίνονται από το v_i και όπου L με S το οποίο υπολογίζει το εμβαδόν, τότε έχουμε έναν προσεγγιστικό αλγόριθμο με την ίδια προσέγγιση όπως και πριν. Ο αλγόριθμος 17 φαίνεται παρακάτω.

Algorithm 17 MaxAreaVertexGuards (* greedy *)

compute FVS points
for every $v \in V(P)$ **do**
  compute $A'(v)$
end for
$SOL \leftarrow \emptyset$
for $i = 1$ to k **do**
  select $v \in V$ that maximizes $S(SOL \cup A'(v))$
  $SOL \leftarrow SOL \cup A'(v)$
end for
return $A(SOL)$

Παρόμοιος αλγόριθμος με τον αλγόριθμο 9 μπορεί να εφαρμοστεί στην περίπτωση του προβλήματος Budgeted Maximum Area Vertex Guard.

8.2 Πομποδέκτες - πλευρές

Ξέρουμε από την ενότητα 2 του κεφ. 1 ότι: μια οποιαδήποτε FVS περιοχή περιφρουρείται αν και μόνο αν επιβλέπεται από κάποια πλευρά. Συνεπώς τα αποτελέσματα της ενότητας 2 του κεφαλαίου 2 εφαρμόζονται με λίγες αλλαγές και εδώ.

Ο αλγόριθμος 18 προσεγγίζει το πρόβλημα Maximum Area Edge Guard με παράγοντα προσέγγισης 1.58.

Algorithm 18 MaxAreaEdgeGuards (* greedy *)

compute FVS points
for every $e \in E(P)$ **do**
  compute $A'(e)$
end for
$SOL \leftarrow \emptyset$
for $i = 1$ to k **do**
  select $e \in E$ that maximizes $S(SOL \cup A'(e))$
  $SOL \leftarrow SOL \cup A'(e)$
end for
return $A(SOL)$

Παρόμοιος αλγόριθμος με τον αλγόριθμο 10 μπορεί να εφαρμοστεί στην περίπτωση του προβλήματος Budgeted Maximum Area Edge Guard.

Κεφάλαιο 9

Μη - προσεγγισιμότητα - Αναγωγές

9.1 Θεωρία μη προσεγγισιμότητας

Προκειμένου να αξιολογήσουμε την ποιότητα των προσεγγιστικών αλγορίθμων που δώσαμε στα προηγούμενα κεφάλαια μας χρειάζονται κάποια αποτελέσματα μη - προσεγγισιμότητας για τα προβλήματα που εξετάσαμε.

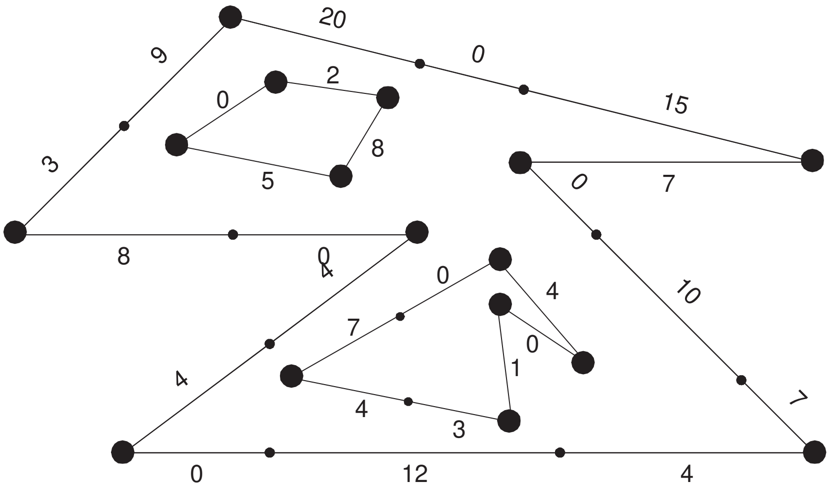
Ένας προσεγγιστικός αλγόριθμος σε συνδυασμό με ένα αποτέλεσμα μη - προσεγγισιμότητας είναι δυνατόν να χαρακτηρίζει ένα πρόβλημα complete σε κάποια κλάση προσέγγισης. Σε αυτήν την περίπτωση ο προσεγγιστικός αλγόριθμος θεωρείται **καλός** ή **σφιχτός** ή **βέλτιστος** ως προς την προσεγγισιμότητα. Το δε αποτέλεσμα μη - προσεγγισιμότητας θεωρείται επίσης **σφιχτό** ή **βέλτιστο**.

Σε μερικά βασικά και σημαντικά προβλήματα η έρευνα συνεχίζεται μερικές φορές και μετά την ταξινόμησή τους σε κλάσεις. Δηλαδή πέρα από το ταίριασμα της τάξης του προσεγγιστικού παράγοντα (σταθερός, λογαριθμικός, κλπ), μερικές φορές προσπαθούμε να ταιριάξουμε όρους χαμηλής τάξης. Για παράδειγμα το πρόβλημα Max Clique δεν είναι προσεγγίσιμο με παράγοντα $n^{1-\epsilon}, \forall \epsilon > 0$ ([23]), ενώ προσεγγίζεται (με ένα προφανή αλγόριθμο) με παράγοντα n.

Σε αυτές τις περιπτώσεις η βελτιστότητα του αλγορίθμου και του αποτελέσματος μη - προσεγγισιμότητας είναι πιο ισχυρή. Για τα APX-complete προβλήματα, ισχυρή βελτιστότητα (strong optimality) σημαίνει η εύρεση μιας σταθεράς $\epsilon > 0$, έτσι ώστε ο προσεγγιστικός αλγόριθμος να πετυχαίνει παράγοντα προσέγγισης $1+\epsilon$ και από την άλλη πλευρά η απόδειξη ότι δεν μπορεί να υπάρχει πολυωνυμικού χρόνου προσεγγιστικός αλγόριθμος που να πετυχαίνει παράγοντα $1+\epsilon-\gamma, \forall \gamma > 0$.

Ένα παράδειγμα τέτοιου προβλήματος για το οποίο υπάρχουν αποτελέσματα ισχυρής βελτιστότητας είναι το MaxExact3SAT ([24]) το οποίο ορίζεται όπως το Max3Sat με τη διαφορά ότι κάθε clause περιέχει ακριβώς 3 literals.

Σε αυτό το κεφάλαιο θα δούμε ένα αποτέλεσμα μη-προσεγγισιμότητας για το πρόβλημα Maximum Value Vertex Guard.



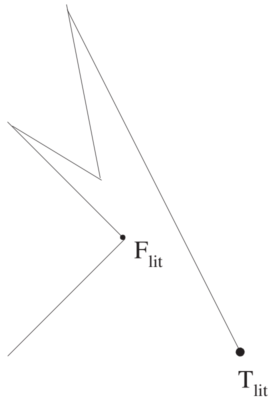
Σχήμα 9.1: ένα πολύγωνο με βάρη

9.2 Maximum Value Vertex Guard (3.1)

Έστω ένα πολύγωνο χωρίς τρύπες όπως στο σχήμα 9.1. Η περίμετρός του είναι χωρισμένη σε διαστήματα με βάρη. Θεωρούμε ότι τα διαστήματα αυτά δεν περιέχουν τα άκρα (ανοιχτά διαστήματα). Το πολύγωνο μπορεί να αναπαριστά το περίγραμμα των τοίχων μιας γκαλερί (κάτοψη). Τα διαστήματα με τα βάρη αναπαριστούν έργα τέχνης κρεμασμένα στους τοίχους της γκαλερί. Τα βάρη αναπαριστούν τις αξίες των έργων τέχνης. Ο σκοπός είναι να τοποθετήσουμε το πολύ k φύλακες πάνω στους κόμβους του πολυγώνου, έτσι ώστε το συνολικό βάρος του τμήματος της περιμέτρου που είναι ορατό από τους φύλακες να είναι το μέγιστο δυνατό.

Μια άλλη εφαρμογή του προβλήματος είναι στα ασύρματα δίκτυα επικοινωνίας. Τα διαστήματα με τα βάρη αναπαριστούν κατοικημένες περιοχές με τους πληθυσμούς τους, ενώ το πολύγωνο μοντελοποιεί το γεωγραφικό χώρο που βρίσκονται οι περιοχές. Θέλουμε να τοποθετήσουμε k σταθμούς στις περιοχές, έτσι ώστε ο περισσότερος δυνατός πληθυσμός να μπορεί να επικοινωνήσει. Το γεγονός ότι τοποθετούμε τους σταθμούς στο περίγραμμα και όχι μέσα στο πολύγωνο μπορεί να υπαγορεύεται π.χ. από το γεγονός ότι οι περιοχές βρίσκονται στις παρυφές κάποιας λίμνης, και τεχνικώς μας συμφέρει να κατασκευάσουμε τους σταθμούς στην ξηρά. Επίσης μπορεί να προκύπτει από το ότι οι περιοχές βρίσκονται ενδεχομένως σκαρφαλωμένες σε πλαγιές βουνών και κάποιος σταθμός τοποθετημένος στο εσωτερικό της περιοχής θα είχε μειωμένη εμβέλεια λόγω χαμηλού ύψους.

Θα αποδείξουμε ότι το πρόβλημα της τοποθέτησης φυλάκων σε πολύγωνο χωρίς τρύπες είναι APX-hard. Θα κάνουμε μια αναγωγή από το πρόβλημα Max-5-Occurence-3-Sat και θα αποδείξουμε ότι η αναγωγή αυτή είναι αναγωγή διατήρησης χάσματος.



Σχήμα 9.2: ένα περίγραμμα literal

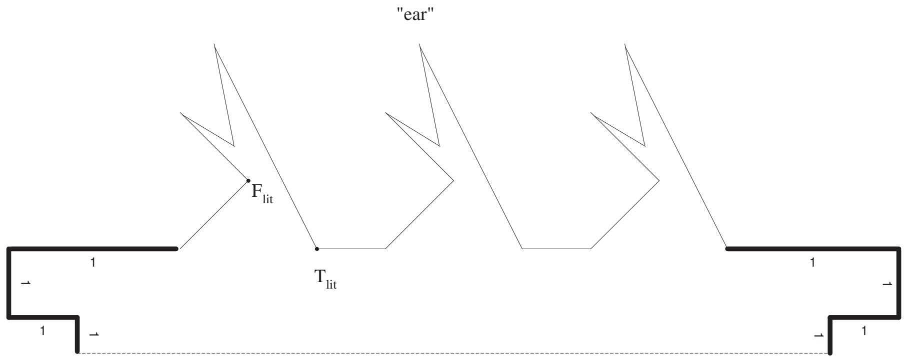
9.2.1 Κατασκευή

Η κατασκευή που κάνουμε μοιάζει με αυτή που χρησιμοποιήθηκε στο [32] για την απόδειξη ότι το πρόβλημα τοποθέτησης ελάχιστου αριθμού φυλάκων σε πολύγωνο (Minimum Vertex Guard) είναι NP-hard. Επίσης χρησιμοποιήθηκε στο [8] για να αποδείξει ότι το Minimum Vertex Guard είναι APX-hard. Οι διαφορές του προβλήματος που εξετάζουμε εδώ είναι ότι έχει βάρη και ότι προσπαθούμε να φρουρήσουμε μέρος της περιμέτρου του πολυγώνου. Αυτές οι διαφορές του έχουν συνέπεια διαφορές:

- στην κατασκευή (τοποθέτηση κατάλληλων βαρών),
- στην μετατροπή μιας εφικτής λύσης του προβλήματος Maximum Value Vertex Guard σε εφικτή λύση του προβλήματος Max-5-Occurence-3-Sat (μετακινούμε τους φύλακες έτσι ώστε να βλέπουν τουλάχιστον τόσο συνολικό βάρος, όσο έβλεπαν και πριν και έτσι ώστε να μας οδηγούν σε μια εφικτή λύση για το πρόβλημα ικανοποιησιμότητας),
- στη διαδικασία της ανάλυσης (προκειμένου να αποδείξουμε ότι μετά τη μετακίνηση των φυλάκων το συνολικό βάρος των τμημάτων της περιμέτρου που φαίνονται είναι τουλάχιστον όσο και πριν) και
- στη διαδικασία απόδειξης ότι $R \geq 1 + \epsilon'$

Για κάθε literal που υπάρχει στο στιγμιότυπο του προβλήματος Max-5-Occurence-3-Sat κατασκευάζουμε ένα περίγραμμα σαν αυτό που φαίνεται στο σχήμα 9.2. Το περίγραμμα αυτό το κατασκευάζουμε έτσι ώστε για να είναι ολόκληρο ορατό από έναν μόνο φύλακα τοποθετημένο σε κορυφή, αυτός θα πρέπει να βρίσκεται είτε στην κορυφή F_{lit} είτε στην κορυφή T_{lit}. Ονομάζουμε αυτό το περίγραμμα **αυτί** ή **περίγραμμα literal**.

Σχήμα 9.3: ένα περίγραμμα clause



Σχήμα 9.4: ένα περίγραμμα clause με τρία literal περιγράμματα

Για κάθε clause που υπάρχει στο στιγμιότυπο του προβλήματος Max-5-Occurence-3-Sat κατασκευάζουμε ένα περίγραμμα σαν αυτό που φαίνεται στο σχήμα 9.3. Το περίγραμμα αυτό το κατασκευάζουμε έτσι ώστε για να είναι ολόκληρο ορατό από έναν φύλακα, αυτός θα πρέπει να βρίσκεται μέσα στην περιοχή που φαίνεται στο σχήμα 9.3. Ονομάζουμε αυτό το περίγραμμα, **περίγραμμα clause**.

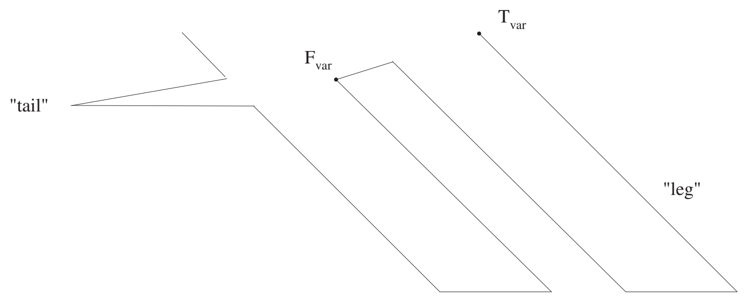
Το περίγραμμα clause με τρία αυτιά που έχει προκύψει από ένα clause με τρία literals φαίνεται στο σχήμα 9.4.

Για κάθε variable που υπάρχει στο στιγμιότυπο του προβλήματος ικανοποιησιμότητας, κατασκευάζουμε το περίγραμμα που φαίνεται στο σχήμα 9.5. Ονομάζουμε το περίγραμμα αυτό, **variable περίγραμμα**, την αριστερή εσοχή **ουρά** και τα δύο κάτω μέρη **πόδια**.

Θα προσθέσουμε επίσης στα variable περιγράμματα κάποια **περιγράμματα ακίδες** τα οποία περιγράφονται παρακάτω.

Τέλος προσθέτουμε ένα αυτί στην πάνω αριστερή γωνία του πολυγώνου και η κατασκευή έχει ολοκληρωθεί. Ο προσανατολισμός των ποδιών των variable περιγραμμάτων είναι τέτοιος, έτσι ώστε να φαίνονται από ένα φύλακα τοποθετημένο στην κορυφή W. Ένα πολύγωνο που έχει κατασκευαστεί όπως περιγράφτηκε παραπάνω και έχει προκύψει από ένα στιγμιότυπο του προβλήματος Max-5-Occurence-3-Sat φαίνεται στο σχήμα 9.6.

Θα περιγράψουμε τώρα την κατασκευή των **περιγραμμάτων ακίδων**. Για κάθε εμφάνιση ενός literal στην λογική έκφραση, δηλαδή για κάθε literal περίγραμμα που έχουμε προσθέσει στο πολύγωνο, προσθέτουμε δύο περιγράμματα ακίδες στο αντίστοιχο περίγραμμα variable ως εξής: Εάν το περίγραμμα literal αντιστοιχεί σε



Σχήμα 9.5: ένα περίγραμμα variable

θετικό (αρνητικό) literal στη λογική έκφραση προσθέτουμε τα δύο περιγράμματα ακίδες έτσι όπως φαίνεται στο σχήμα 9.7 (9.8). Το περίγραμμα ακίδα το οποίο φαίνεται από την κορυφή $F_{lit}(T_{lit})$ ονομάζεται FALSE (TRUE) ακίδα. Παρατηρήστε στα σχήματα 9.7 και 9.8 ότι τα περιγράμματα ακίδες είναι κατασκευασμένα έτσι ώστε η βάση του FALSE περιγράμματος ακίδας είναι το τμήμα (a, F_{lit}), ενώ η βάση του TRUE περιγράμματος ακίδας είναι το τμήμα (T_{lit}, b) και όχι (T_{lit}, c). Ο σκοπός που κατασκευάστηκαν έτσι είναι για να μην υπάρχει κορυφή στο clause περίγραμμα (δες σχήμα 9.6) από την οποία να φαίνονται περισσότερα από ένα περιγράμματα ακίδες.

Από τον τρόπο κατασκευής του πολυγώνου προκύπτουν τα εξής:

Τρεις φύλακες είναι αναγκαίοι και αρκούν για να φυλάξουν ένα περίγραμμα αυτί, το αντίστοιχο variable περίγραμμα (δύο πόδια και μια ουρά) και τα αντίστοιχα περιγράμματα ακίδες:

Ο ένας από αυτούς τους φύλακες πρέπει να τοποθετηθεί στην κορυφή W και από εκεί βλέπει τα πόδια του variable περιγράμματος.

Οι υπόλοιποι δύο τοποθετούνται στις κορυφές ως εξής:

- $\{F_{var}, F_{lit}\}$, ή $\{T_{var}, T_{lit}\}$ εάν πρόκειται για ακίδες που αντιστοιχούν σε θετικά literals και

- $\{F_{var}, T_{lit}\}$, ή $\{T_{var}, F_{lit}\}$ εάν πρόκειται για ακίδες που αντιστοιχούν σε αρνητικά literals

Η κατασκευή ολοκληρώνεται με την ανάθεση των βαρών. Αναθέτουμε βάρος 8 σε κάθε ακμή του πολυγώνου που κατασκευάσαμε, εκτός από τις 'φθηνές' ακμές των clause περιγραμμάτων που είναι τονισμένες στο σχήμα 9.4, στις οποίες αναθέτουμε βάρος 1.

Παίρνουμε σαν αριθμό διαθέσιμων φυλάκων $k = l+n+1$, όπου l είναι ο αριθμός των εμφανίσεων των literals και n είναι ο αριθμός των μεταβλητών της λογικής έκφρασης.

Έτσι έχουμε κατασκευάσει ένα στιγμιότυπο του προβλήματος Maximum Value Vertex Guard.

9.2.2 Μετασχηματισμός μιας εφικτής λύσης

Έστω μια απονομή αλήθειας της λογικής έκφρασης. Θα περιγράψουμε μια τοποθέτηση φυλάκων που αντιστοιχεί στη δεδομένη απονομή αλήθειας. Τοποθετούμε $k = l + n + 1$ φύλακες στις κορυφές του πολυγώνου που κατασκευάσαμε στην προηγούμενη ενότητα ως εξής:

- τοποθετούμε ένα φύλακα σε κάθε περίγραμμα variable στην κορυφή $F_{var}(T_{var})$, εάν η τιμή αλήθειας της αντίστοιχης μεταβλητής είναι FALSE (TRUE),
- τοποθετούμε ένα φύλακα σε κάθε literal περίγραμμα στην κορυφή $F_{lit}(T_{lit})$ εάν η αποτίμηση του αντίστοιχου literal στη λογική έκφραση είναι FALSE (TRUE),
- τέλος τοποθετούμε ένα φύλακα στο πρόσθετο περίγραμμα αυτί στην κορυφή W.

Με αυτό τον τρόπο τοποθέτησης των φυλάκων είναι εύκολο να δει κανείς ότι κάθε περίγραμμα literal φαίνεται από κάποιον φύλακα. Επίσης κάθε περίγραμμα variable φαίνεται από κάποιον φύλακα. Ομοίως φαίνονται όλα τα περιγράμματα ακίδες. Επίσης τα πόδια από τα variable περιγράμματα φαίνονται από τον φύλακα που έχει τοποθετηθεί στην κορυφή W. Οι μόνες ακμές που δεν θα φαίνονται είναι οι ακμές από τα clause περιγράμματα τα οποία αντιστοιχούν σε clauses που δεν ικανοποιούνται από τη δεδομένη απονομή αλήθειας.

Αντίστροφα, δεδομένης μιας τοποθέτησης από $k = l + n + 1$ φύλακες στο κατασκευασμένο πολύγωνο της προηγούμενης ενότητας, το οποίο αποτελεί στιγμιότυπο του Maximum Value Vertex Guard προβλήματος, θα περιγράψουμε μια απονομή αλήθειας για το αρχικό στιγμιότυπο του Max-5-Occurence-3-Sat προβλήματος. Αρχικά αλλάζουμε την τοποθέτηση των φυλάκων ως εξής:

- τοποθετούμε ακριβώς ένα φύλακα σε κάθε variable περίγραμμα σε κάποια από τις κορυφές F_{var} ή T_{var},
- τοποθετούμε ακριβώς ένα φύλακα σε κάθε literal περίγραμμα στην κορυφή $F_{lit}(T_{lit})$ εάν το αντίστοιχο TRUE (FALSE) περίγραμμα ακίδα φαίνεται από τον φύλακα που έχει τοποθετηθεί στο περίγραμμα variable,
- τοποθετούμε έναν φύλακα στην κορυφή W του πρόσθετου αυτιού.

Ακολουθεί η περιγραφή με περισσότερες λεπτομέρειες: Δεδομένης μιας τοποθέτησης από $k = l + n + 1$ φύλακες, έστω ότι η συνολική αξία των τμημάτων που φαίνονται είναι B. Θα αλλάξουμε την τοποθέτηση των φυλάκων έτσι ώστε η αξία των τμημάτων που φαίνονται να είναι τουλάχιστον B, και έτσι ώστε με την εξαίρεση κάποιων 'φτηνών' ακμών με βάρη 1, η νέα τοποθέτηση φυλάκων πετυχαίνει:

α) να βλέπει τις ακμές του πολυγώνου και β) 'συνεπή' τοποθέτηση φυλάκων σε δύο κορυφές από τις 4 $F_{lit}, T_{lit}, F_{var}, T_{var}$, για κάθε literal περίγραμμα. Ακολουθεί η νέα τοποθέτηση των φυλάκων:

- βήμα 1: τοποθετούμε ένα φύλακα στην κορυφή w του πρόσθετου περιγράμματος αυτιού,
- βήμα 2: για κάθε περίγραμμα μεταβλητής:
  - i) εάν υπάρχει μόνο ένας φύλακας στο περίγραμμα, τοποθετημένος σε κορυφή από την οποία φαίνεται κάποιο περίγραμμα ακίδας, τον τοποθετούμε στην κορυφή $F_{var}(T_{var})$ αν από την κορυφή $F_{var}(T_{var})$ φαίνεται η ίδια ακίδα
  - ii) σε όλες τις άλλες περιπτώσεις (δεν υπάρχουν τοποθετημένοι φύλακες στο περίγραμμα, υπάρχει ένας φύλακας που δεν βλέπει κανένα περίγραμμα ακίδα, υπάρχουν περισσότεροι από ένας φύλακες), τοποθετούμε ένα φύλακα στην κορυφή $F_{var}(T_{var})$ εάν από την κορυφή $F_{var}(T_{var})$ ο φύλακας μπορεί να δει περισσότερα FALSE περιγράμματα ακίδες από ότι από την κορυφή $T_{var}(F_{var})$
- βήμα 3: για κάθε literal περίγραμμα, τοποθετούμε ένα φύλακα στην κορυφή $F_{lit}(T_{lit})$ εάν το αντίστοιχο FALSE (TRUE) περίγραμμα ακίδας δεν φαίνεται από τον φύλακα ο οποίος έχει τοποθετηθεί στο variable περίγραμμα.

Δηλαδή η πιθανή μετακίνηση των φυλάκων στα περιγράμματα variable έχει οδηγήσει σε μετακίνηση των φυλάκων που είχαν τοποθετηθεί στα literal περιγράμματα.

Θα αποδείξουμε στην επόμενη ενότητα (βλ. Λήμμα 10.2) ότι το συνολικό βάρος των ακμών που φαίνονται από τους φύλακες μετά τη νέα τοποθέτηση είναι τουλάχιστον B.

Τώρα μπορούμε να περιγράψουμε μια απονομή αλήθειας ως εξής: ανάθεσε τιμή TRUE (FALSE) σε μια variable της λογικής έκφρασης εάν το αντίστοιχο variable περίγραμμα έχει φύλακα τοποθετημένο στην κορυφή $T_{var}(F_{var})$.

9.2.3 Ανάλυση της αναγωγής

Έστω I ένα στιγμιότυπο του προβλήματος MAX-5-OCCURENCE-3-SAT με n variables, l εμφανίσεις από literals και m clauses ($l \leq 3m$). Έστω I' ένα στιγμιότυπο του προβλήματος MAXIMUM VALUE VERTEX GUARD (το οποίο κατασκευάστηκε όπως στην ενότητα 1.1) με $k = l + n + 1$. Έστω M το συνολικό βάρος όλων των ακμών του πολυγώνου.

Λήμμα 9.1 *Αν $OPT(I) = m$ τότε $OPT(I') = M$*

Απόδειξη. Έστω λοιπόν ότι υπάρχει μια απονομή αλήθειας που ικανοποιεί και τα m clauses. Εάν τοποθετήσουμε $l+n+1$ φύλακες στο πολύγωνο όπως στην ενότητα 10.1.2, τότε είναι εύκολο να δούμε ότι το σύνολο του περιγράμματος του πολυγώνου φαίνεται από τους φύλακες. Έτσι το συνολικό βάρος των ακμών που φαίνονται είναι M. □

Λήμμα 9.2 *Αν* $OPT(I') \geq M - 8\epsilon m$ *τότε* $OPT(I) \geq m(1-\epsilon)$

Απόδειξη. Έστω $\epsilon > 0$ και μια τοποθέτηση από $l+n+1$ φύλακες στο στιγμιότυπο I' έτσι ώστε το συνολικό βάρος των τμημάτων του περιγράμματος του πολυγώνου που φαίνονται από τους φύλακες είναι τουλάχιστον $M - 8\epsilon m$. Αν μετακινήσουμε τους φύλακες όπως περιγράφτηκε στην ενότητα 10.1.2, τότε οι $k = l+n+1$ φύλακες βλέπουν το περίγραμμα ολόκληρο, εκτός από κάποιες 'φθηνές' ακμές. Θα αποδείξουμε ότι το συνολικό βάρος των ακμών του περιγράμματος που φαίνονται μετά τη μετακίνηση των φυλάκων είναι τουλάχιστον όσο και πριν. Δηλαδή τουλάχιστον $M - 8\epsilon m$.

Παρατηρούμε ότι αν τοποθετήσουμε ένα φύλακα στην κορυφή F_{lit} ή T_{lit} από ένα literal περίγραμμα το οποίο πριν δεν φυλασσόταν τότε σίγουρα αυξάνουμε το συνολικό βάρος κατά 16 τουλάχιστον αφού οι ακμές (F_{lit}, d) και (d, e) δε μπορεί να φαίνονται από φύλακα εκτός του literal περιγράμματος.

Ομοίως, ένας φύλακας τοποθετημένος στην κορυφή F_{var} ή T_{var} ενός variable περιγράμματος το οποίο πριν δεν είχε φύλακες, ασφαλώς αυξάνει το συνολικό βάρος των ακμών που φαίνονται κατά τουλάχιστον 16 (είναι το βάρος των δύο ακμών της ουράς του variable περιγράμματος).

Θα συζητήσουμε τώρα δύο περιπτώσεις που αφορούν τοποθέτηση φυλάκων στα αυτιά των literal περιγραμμάτων.

- Έστω ότι κατά την αρχική τοποθέτηση υπήρχαν δύο φύλακες στις κορυφές T_{lit} και F_{lit} ενός αυτιού και μετά τη μετακίνηση, ο μοναδικός φύλακας τοποθετήθηκε στην κορυφή F_{lit}. Το συνολικό βάρος μπορεί να έχει μειωθεί κατά το πολύ 8 (διότι τώρα μπορεί να μη φαίνονται οι φθηνές ακμές που φαίνονταν από το T_{lit}), όμως αυξήθηκε σίγουρα κατά 16 (διότι ο φύλακας που έφυγε τοποθετήθηκε σε μια μέχρι τώρα αφύλακτη περιοχή).

- Έστω ότι κατά την αρχική τοποθέτηση υπήρχε ένας φύλακας στην κορυφή T_{lit} ενός αυτιού (και δεν υπήρχε άλλος στο αυτί) και μετά τη μετακίνηση ο φύλακας αυτός βρίσκεται στην κορυφή F_{lit} του αυτιού:

  – Εάν το αντίστοιχο FALSE περίγραμμα ακίδα δεν φαινόταν κατά την αρχική τοποθέτηση των φυλάκων (από ένα φύλακα δηλαδή στο variable περίγραμμα), το συνολικό βάρος μετά τη μετακίνηση μπορεί να έχει μειωθεί κατά το πολύ 8 (αφού τώρα μπορεί να μην φαίνονται οι φθηνές πλευρές) αλλά έχει αυξηθεί κατά τουλάχιστον 16 (αφού το FALSE περίγραμμα ακίδα τώρα φαίνεται από τον φύλακα στην κορυφή F_{lit}).

  – Εάν το αντίστοιχο FALSE περίγραμμα ακίδα, φαινόταν κατά την αρχική τοποθέτηση των φυλάκων (από ένα φύλακα g στο variable περίγραμμα) τότε σύμφωνα με το βήμα 3 στην μετακίνηση της ενότητας 10.1.2, στο περίγραμμα variable δεν μπορεί να υπήρχε μόνο ένας φύλακας, αφού αν ήταν έτσι θα έβλεπε αυτός το FALSE περίγραμμα ακίδας και με βάση το βήμα 3 θα έπρεπε να τοποθετήσουμε φύλακα στο T_{lit} και όχι στο F_{lit}. Άρα πρόκειται για την περίπτωση στην οποία κατά την αρχική τοποθέτηση, υπάρχουν τουλάχιστον 2 φύλακες στο variable περίγραμμα, και μετά τη μετακίνηση ο φύλακας g έχει μετακινηθεί και τοποθετηθεί σε άλλο περίγραμμα γιατί σύμφωνα με το βήμα 2)ii) της ενότητας 10.1.2 υπήρχε άλλος φύλακας που έβλεπε τα περισσότερα FALSE περιγράμματα ακίδες στο περίγραμμα variable. Αυτό σημαίνει ότι ο φύλακας g έβλεπε το πολύ 2 FALSE περιγράμματα ακίδες, διότι ένα περίγραμμα variable μπορεί να έχει το πολύ 5 FALSE περιγράμματα ακίδες (μια μεταβλητή εμφανίζεται το πολύ σε 5 literals). Έτσι λοιπόν για κάθε variable περίγραμμα, φύλακες έχουν μετακινηθεί από την κορυφή T_{lit} στην F_{lit} σε το πολύ 2 literal περιγράμματα. Αρα το συνολικό βάρος έχει μειωθεί κατά το πολύ 16 (διότι μπορεί μετά τη μετακίνηση να μην φαίνονται οι φθηνές πλευρές από 2 clause περιγράμματα) αλλά έχει αυξηθεί κατά τουλάχιστον 16 (αφού τουλάχιστον ένας φύλακας έχει τώρα τοποθετηθεί σε ένα αφύλακτο ως τώρα περίγραμμα).

Υπενθυμίζουμε, ότι μετά τη μετακίνηση μόνο 'φθηνές' ακμές μπορεί να μην φαίνονται. Δεν υπάρχει περίπτωση δηλαδή μετά τη μετακίνηση που έγινε όπως στην ενότητα 10.1.2 να μην φαίνεται κάποιο περίγραμμα ακίδα, κλπ. Μετά λοιπόν τη μετακίνηση έχουμε πετύχει να βλέπουμε όλο το περίγραμμα του πολυγώνου εκτός ίσως από ϵm clause περιγράμματα. Αυτό σημαίνει ότι αν κατασκευάσουμε μια απονομή αλήθειας για το στιγμιότυπο I όπως στην ενότητα 10.1.2, αυτή η απονομή αλήθειας θα αφήνει μη-ικανοποιήσιμα, ακριβώς αυτά τα ϵm clauses που αντιστοιχούν στα ϵm clause περιγράμματα που δεν καλύπτονται από φύλακες στο I'. □

Από το λήμμα 10.1 και την αντιθετοαντιστροφή του λήμματος 10.2 προκύπτει το παρακάτω θεώρημα.

Θεώρημα 9.1 *Έστω I ένα στιγμιότυπο του προβλήματος* Max-5-Occurence-3-Sat *με n variables, m clauses και $l \leq 3m$ εμφανίσεις από literals. Έστω I' το στιγμιότυπο του προβλήματος* Maximum Value Vertex Guard *(το οποίο κατασκευάστηκε όπως στην ενότητα 10.1.1) με $k = l+n+1$. Έστω M το συνολικό βάρος όλων των ακμών του πολυγώνου. Τότε:*

$$OPT(I) = m \rightarrow OPT(I') = M$$

$$OPT(I) \leq m(1-\epsilon) \rightarrow OPT(I') \leq M - 8\epsilon m$$

Δηλαδή η αναγωγή αυτή είναι αναγωγή διατήρησης χάσματος.

Στο [1] και [2] έχει αποδειχθεί ότι το πρόβλημα Max-5-Occurence-3-Sat με παραμέτρους m και $(1-\epsilon)m$ για κάποιο $\epsilon > 0$, όπου m είναι ο αριθμός των clauses στο στιγμιότυπο I, είναι NP-hard να αποφασισθεί.

Αρα μέσω της gap-preserving αναγωγής που περιγράφτηκε παραπάνω, ισχύει ότι εκτός αν $P = NP$, δε μπορεί να υπάρχει προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου για το πρόβλημα Maximum Value Vertex Guard, ο οποίος να πετυχαίνει προσεγγιστικό παράγοντα

$$\frac{M}{M - 8\epsilon m}$$

.

Εάν συμβολίσουμε:

- με V το συνολικό βάρος ενός variable περιγράμματος (πόδια, ουρά, ακμές στα πόδια ανάμεσα σε ακίδες, συν μια πλευρά η οποία ενώνει το περίγραμμα variable με το επόμενο περίγραμμα variable προς τα δεξιά), έχουμε $104 \leq V \leq 168$
- με L το συνολικό βάρος ενός literal περιγράμματος, $L = 40$
- με S το συνολικό βάρος ενός περιγράμματος ακίδα, $S = 16$
- με C το συνολικό βάρος ενός clause περιγράμματος χωρίς τα αυτιά, συν μια πλευρά που συνδέει το περίγραμμα clause με το επόμενο περίγραμμα clause προς τα δεξιά, τότε $16 \leq C \leq 32$
- με E το συνολικό βάρος του πρόσθετου περιγράμματος αυτιού συν τις υπόλοιπες πλευρές του περιγράμματος του πολυγώνου, $E = 80$

Έχουμε:

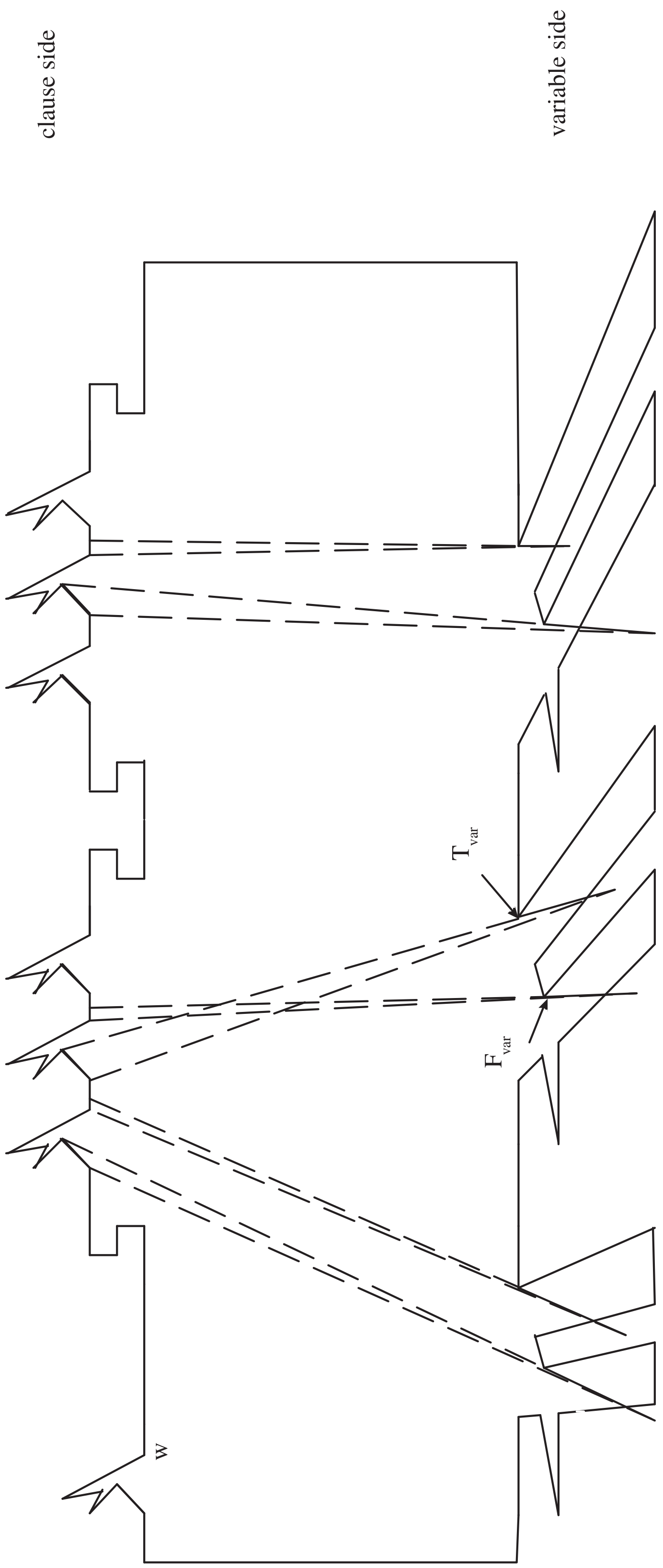
$$M \leq 3mV + 3mL + 6mS + mC + E \leq m(3V + 3L + 6S + C + E) \leq m\epsilon'$$

Μετά από μερικούς υπολογισμούς έχουμε:

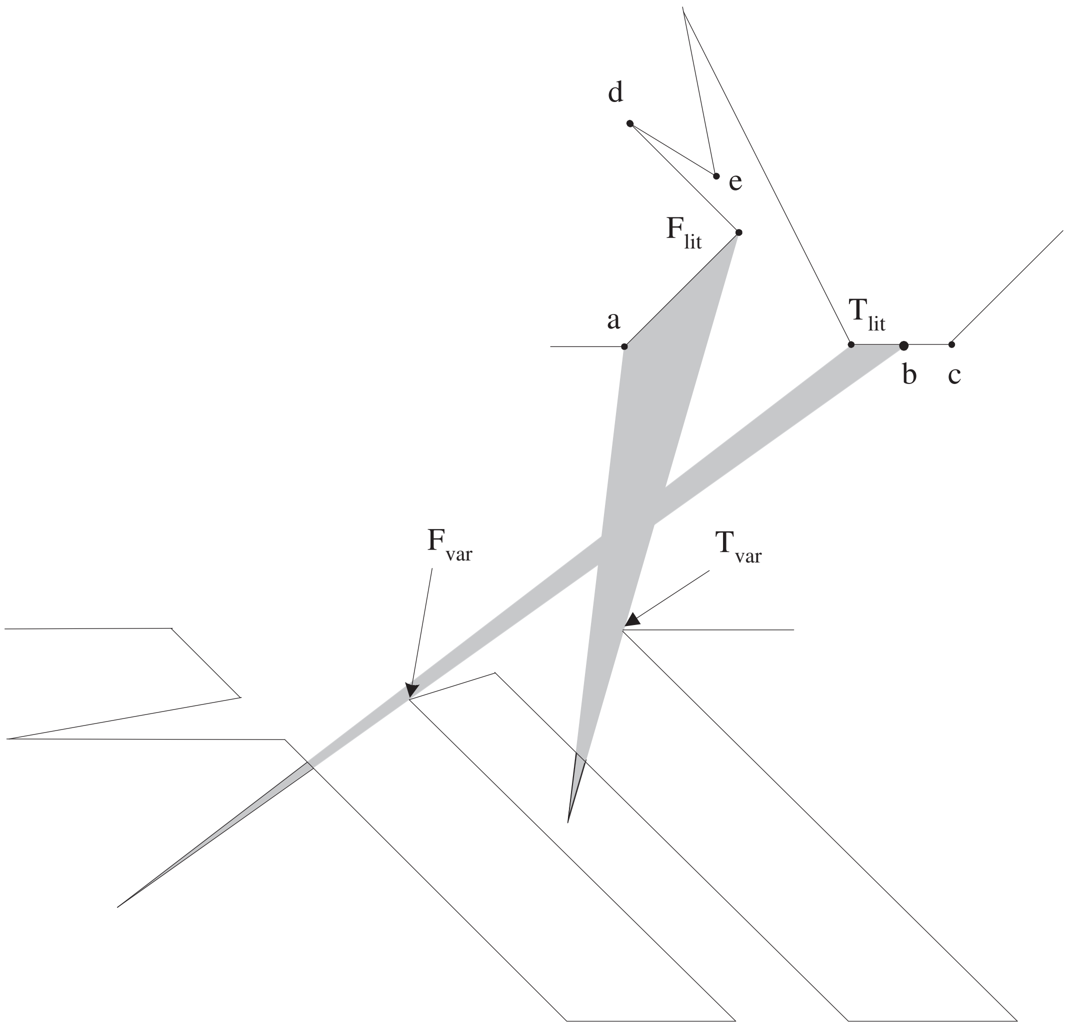
$$\frac{M}{M - 8\epsilon m} = \frac{1}{1 - \frac{8\epsilon m}{M}} \geq \frac{1}{1 - \frac{8\epsilon}{3V+3L+6S+C+E}}$$

$$\geq \frac{1}{1 - \frac{8\epsilon}{\epsilon'}} \geq 1 + \frac{\frac{8\epsilon}{\epsilon'}}{1 - \frac{8\epsilon}{\epsilon'}} \geq 1 + \frac{8\epsilon}{\epsilon'} \geq 1 + \epsilon''$$

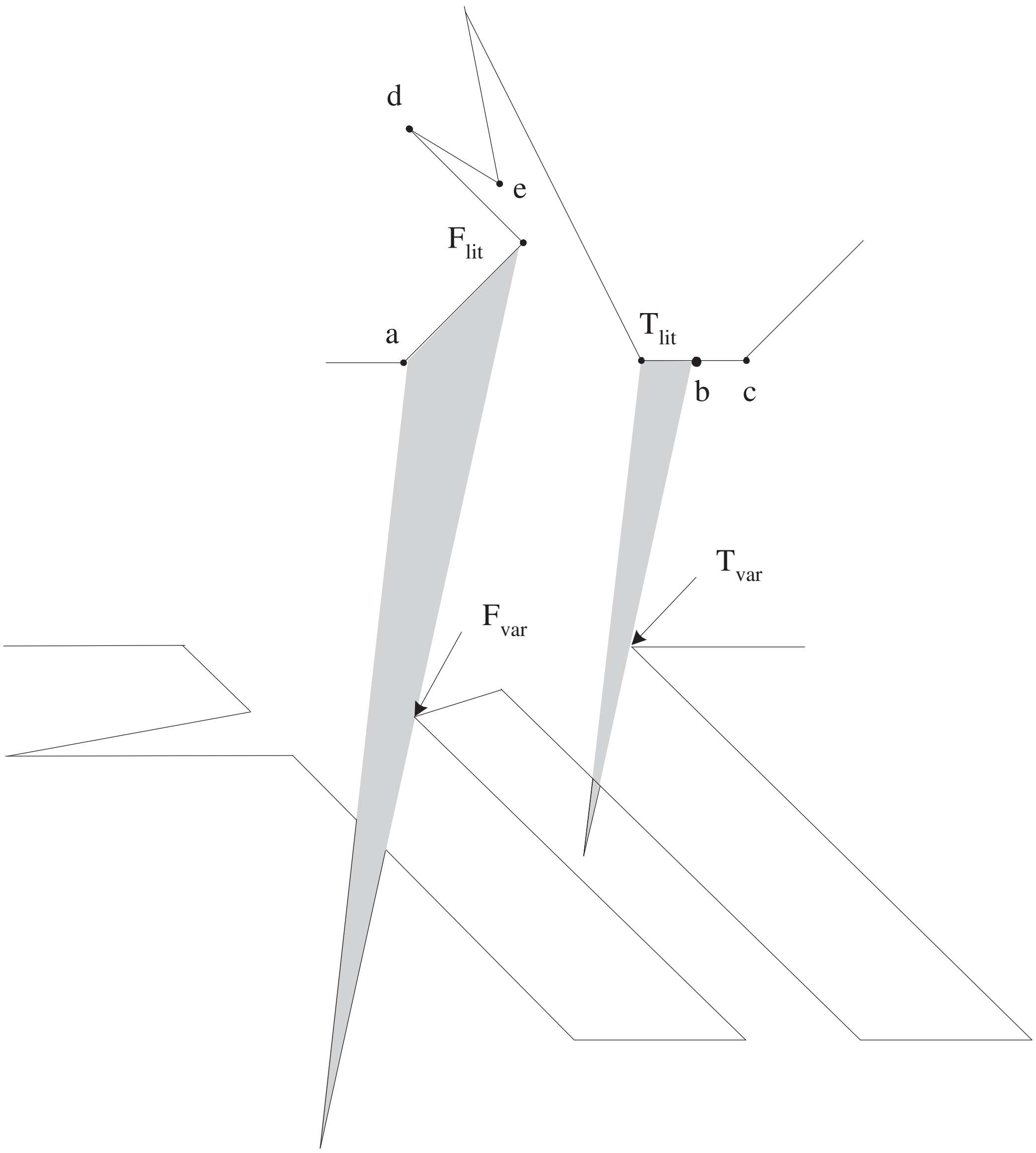
Θεώρημα 9.2 *Το πρόβλημα* Maximum Value Vertex Guard *για πολύγωνα χωρίς τρύπες είναι APX-hard.*



Σχήμα 9.6: προκύπτον πολύγωνο



Σχήμα 9.7: δύο περιγράμματα ακίδες που αντιστοιχούν στην εμφάνιση ενός θετικού literal σε ένα clause. Τα δύο περιγράμματα ακίδες και το αυτί φαίνονται από δύο φύλακες τοποθετημένους π.χ. στο σημείο F_{var} (βλέπει το αριστερό περίγραμμα ακίδα και την ουρά) και στο σημείο F_{lit} (βλέπει το δεξιό περίγραμμα ακίδα και το αυτί).



Σχήμα 9.8: δύο περιγράμματα ακίδες που αντιστοιχούν στην εμφάνιση ενός αρνητικού literal σε ένα clause. Τα δύο περιγράμματα ακίδες και το αυτί φαίνονται από δύο φύλακες τοποθετημένους π.χ. στο σημείο T_{var} (βλέπει το δεξιό περίγραμμα ακίδα και την ουρά) και στο σημείο F_{lit} (βλέπει το αριστερό περίγραμμα ακίδα και το αυτί).

Κεφάλαιο 10

Περισσότερα APX-complete προβλήματα

Σε αυτό το κεφάλαιο θα δούμε πως μπορούμε να εφαρμόσουμε την αναγωγή του προηγούμενου κεφαλαίου για να αποδείξουμε αρνητικά αποτελέσματα προσεγγισιμότητας για τα προβλήματα των κεφαλαίων 2-7.

10.1 Maximum Value Edge Guard (3.2)

Ας εξετάσουμε την περίπτωση στην οποία οι φύλακες επιτρέπεται να τοποθετούνται μόνο σε πλευρές (ένας φύλακας καταλαμβάνει μια ολόκληρη πλευρά).

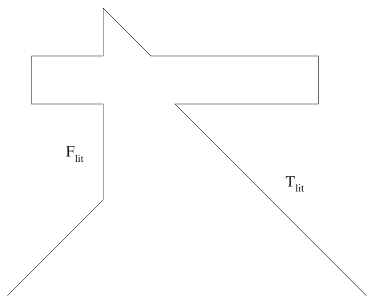
Πρόταση 10.1 *Το πρόβλημα* Maximum Value Edge Guard *είναι APX-hard.*

Απόδειξη. Θα χρησιμοποιήσουμε την αναγωγή του προηγούμενου κεφαλαίου κάνοντας τις εξής αλλαγές: το literal pattern του πολυγώνου είναι αυτό που φαίνεται στο σχήμα 10.1. Το variable pattern είναι αυτό που φαίνεται στο σχήμα 10.2. Το θεώρημα 9.1 ισχύει και εδώ με τη μόνη διαφορά ότι οι φύλακες τώρα καταλαμβάνουν πλευρές και όχι κόμβους. □

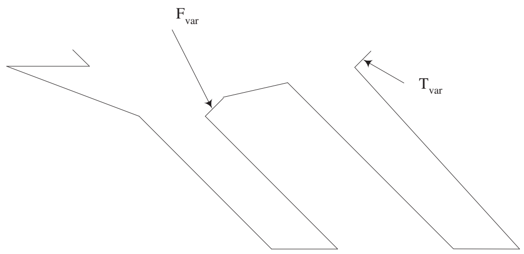
10.2 Maximum Value Guard watching (3.4)

Ας δούμε τι γίνεται στην περίπτωση που ο σκοπός μας είναι να τοποθετήσουμε φύλακες που να επιβλέπουν τη μέγιστη αξία και όχι να την καλύπτουν. Δηλαδή ο φύλακας είναι ανάγκη να βλέπει τουλάχιστον ένα κομμάτι του πίνακα και όχι αναγκαστικά ολόκληρο.

Πρόταση 10.2 *Το πρόβλημα* Maximum Value Vertex Guard *(watching version) είναι APX-hard.*



Σχήμα 10.1: Ένα literal pattern για φύλακες - πλευρές



Σχήμα 10.2: Ένα variable pattern για φύλακες - πλευρές

Απόδειξη. Θα περιγράψουμε μια αναγωγή από το πρόβλημα Max-5-Occurence-3-Sat στο πρόβλημα Maximum Value Vertex Guard (watching version). Αρχικά κατασκευάζουμε το πολύγωνο όπως στην ενότητα 10.1.1.

Κάθε πλευρά μέσα σε ένα clause pattern έχει διαιρεθεί σε $O(n)$ FVS τμήματα αφού μπορεί να επιβλέπεται μόνο από σημεία στα variable patterns. Έστω $\delta > 0$ ένας ακέραιος τέτοιος ώστε ο αριθμός των τμημάτων σε οποιαδήποτε (προηγούμενη) φθηνή πλευρά είναι το πολύ δn. Θέτουμε αξία 1 σε κάθε τμήμα το οποίο ανήκει σε μια (προηγούμενα) φθηνή πλευρά του clause pattern. Θέτουμε αξία $8\delta n$ σε οποιοδήποτε άλλο τμήμα. Τώρα μπορούμε να επαληθεύσουμε ότι το θεώρημα 9.1 ισχύει. □

Πρόταση 10.3 *Το πρόβλημα* Maximum Value Edge Guard *(watching version) είναι APX-hard.*

Απόδειξη. Ομοίως ανάγουμε το πρόβλημα Max-5-Occurence-3-Sat στο Maximum Value Edge Guard όπως και στην προηγούμενη πρόταση με τη διαφορά ότι στην αρχική κατασκευή του πολυγώνου χρησιμοποιούμε σαν literal και variable pattern εκείνα των σχημάτων 10.1 και 10.2. □

10.3 Maximum Length Guard (2.1 / 2.2)

Πρόταση 10.4 *Τα προβλήματα* Maximum Length Vertex Guard *και* Maximum Length Edge Guard *είναι APX-hard.*

Απόδειξη. Ανάγουμε πάλι όπως και πριν το πρόβλημα Max-5-Occurence-3-Sat στα Maximum Length Vertex Guard και Maximum Length Edge Guard. Χρησιμοποιούμε στην κατασκευή τα αντίστοιχα literal και variable patterns όπως και πριν. Αυτή τη φορά κατασκευάζουμε το πολύγωνο έτσι ώστε κάθε (προηγούμενα) φθηνή πλευρά ενός clause pattern να είναι 8 φορές μικρότερη σε μήκος από κάθε άλλη πλευρά του πολυγώνου. Το θεώρημα 9.1 ισχύει και εδώ. □

10.4 Maximum Value Vertex/Edge Guard with Painting Placement (4.1 / 4.2)

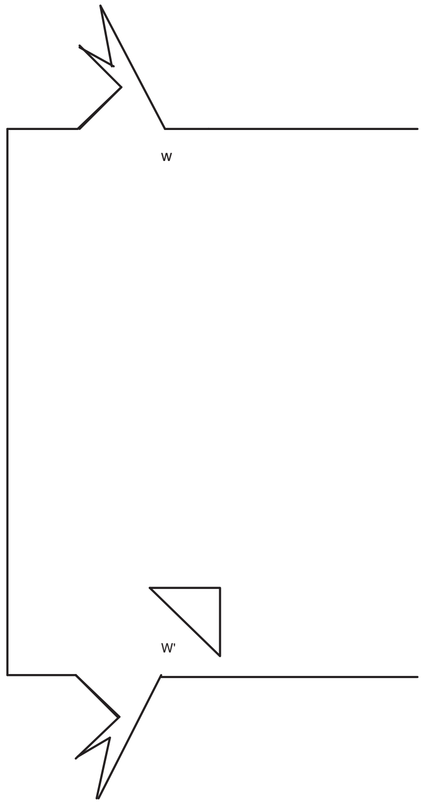
Ας δούμε τώρα την περίπτωση στην οποία οι πίνακες που πρέπει να φρουρήσουμε δεν έχουν ακόμη τοποθετηθεί στους τοίχους του μουσείου (δηλαδή στην περίμετρο του πολυγώνου). Δηλαδή μας δίνονται οι διαστάσεις των πινάκων με αξία και μας ζητούν να τοποθετήσουμε και τους πίνακες και τους φύλακες έτσι ώστε η μέγιστη δυνατή αξία να φυλάσσεται.

Πρόταση 10.5 *Τα προβλήματα* Maximum Value Vertex Guard with Painting Placement *και* Maximum Value Edge Guard with Painting Placement *είναι APX-hard.*

Απόδειξη. Δεδομένου ενός στιγμιότυπου του προβλήματος Max-5-Occurence-3-Sat μπορούμε να κατασκευάσουμε ένα στιγμιότυπο του προβλήματος Maximum Value Vertex Guard with Painting Placement ή Maximum Value Edge Guard with Painting Placement ως εξής: κατασκευάζουμε το κατάλληλο πολύγωνο για την περίπτωση τοποθέτησης φυλάκων σε κορυφές ή πλευρές. Κατασκευάζουμε τις (προηγούμενα) φθηνές ακμές των clause patterns λίγο μικρότερες σε μήκος από οποιαδήποτε άλλη πλευρά. Για κάθε πλευρά e βάζουμε στο σύνολο των πινάκων ένα πίνακα με μήκος ίδιο με της e. Τέλος βάζουμε αξία 8 σε κάθε πίνακα εκτός από εκείνους τους πίνακες που τους φτιάξαμε για να ταιριάζουν στις μικρότερες πλευρές (πρώην φθηνές) των clause patterns. Σε αυτούς τους τελευταίους πίνακες βάζουμε αξία 1. Το θεώρημα 9.1 ισχύει. □

10.5 Πολύγωνα με τρύπες (2.3 / 3.3 / 4.3)

Όλα αυτά τα προβλήματα μπορούμε να τα ορίσουμε σε πολύγωνα με τρύπες. Τα πολύγωνα με τρύπες είναι χρήσιμα γιατί μας δίνουν την ευκαιρία να μοντελοποιήσουμε την πραγματικότητα πιο πιστά. Για παράδειγμα οι τρύπες μπορούν να αναπαριστούν φυσικά εμπόδια. Επίσης χάρη στις τρύπες μπορούμε να τοποθετήσουμε φύλακες στο



Σχήμα 10.3: Το αριστερό μέρος του πολυγώνου με την τρύπα

εσωτερικό του πολυγώνου πάνω στους κόμβους και τις πλευρές των τρυπών (δηλαδή σε προκαθορισμένες περιοχές).

Θυμίζουμε στον αναγνώστη ότι τα προβλήματα Minimum Vertex Guard και Minimum Edge Guard σε πολύγωνα με τρύπες είναι $\log n$-hard. Πιο συγκεκριμένα, δεν υπάρχει προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου ο οποίος να μπορεί να εγγυηθεί προσεγγιστικό παράγοντα $\frac{1-\epsilon}{12} \ln n$ για οποιοδήποτε $\epsilon > 0$, εκτός αν $NP \subseteq TIME(n^{O(\log \log n)})$ ([18, 15]).

Πρόταση 10.6 *Τα προβλήματα:* Maximum Value Vertex Guard, Maximum Value Edge Guard, Maximum Value Vertex Guard *watching version,* Maximum Value Edge Guard *watching version,* Maximum Length Vertex Guard, Maximum Length Edge Guard, Maximum Value Vertex Guard with Painting Placement *και* Maximum Value Edge Guard with Painting Placement *σε πολύγωνα με τρύπες είναι APX-hard.*

Απόδειξη. Ανάγουμε το Max-5-Occurence-3-Sat σε καθένα από τα παραπάνω προβλήματα. Στο κατασκευαστικό μέρος της κάθε αναγωγής προσθέτουμε μια τρύπα και ένα “αυτί” στην αριστερή κάτω γωνία του πολυγώνου, όπως φαίνεται στο σχήμα 10.3. Το θεώρημα 9.1 ισχύει για κάθε αναγωγή. □

10.6 Όριο συνολικού κόστους (5 / 6 / 7)

Τα προβλήματα των κεφαλαίων 2, 3, 4 αποτελούν ειδικές περιπτώσεις των προβλημάτων με όριο συνολικού κόστους που εξετάσαμε στα κεφάλαια 5, 6, 7. όπου κάθε υποψήφια θέση φύλακα έχει κόστος 1.

Πρόταση 10.7 *Τα προβλήματα:* Budgeted Maximum Value Vertex Guard, Budgeted Maximum Value Edge Guard, Budgeted Maximum Value Vertex Guard *watching version,* Budgeted Maximum Value Edge Guard *watching version,* Budgeted Maximum Length Vertex Guard, Budgeted Maximum Length Edge Guard, Budgeted Maximum Value Vertex Guard with Painting Placement *και* Budgeted Maximum Value Edge Guard with Painting Placement *σε πολύγωνα με ή χωρίς τρύπες είναι APX-hard.*

10.7 APX-completeness

Από τα αποτελέσματα του κεφαλαίου αυτού και των κεφαλαίων 2, 3, 4, 5, 6, 7 προκύπτουν ότι:

Θεώρημα 10.1 *Τα προβλήματα:* Maximum Value Vertex Guard, Maximum Value Edge Guard, Maximum Value Vertex Guard *watching version,* Maximum Value Edge Guard *watching version,* Maximum Length Vertex Guard, Maximum Length Edge Guard, Maximum Value Vertex Guard with Painting Placement *και* Maximum Value Edge Guard with Painting Placement *σε πολύγωνα με ή χωρίς τρύπες είναι APX-complete.*

Θεώρημα 10.2 *Τα προβλήματα:* Budgeted Maximum Value Vertex Guard, Budgeted Maximum Value Edge Guard, Budgeted Maximum Value Vertex Guard *watching version,* Budgeted Maximum Value Edge Guard *watching version,* Budgeted Maximum Length Vertex Guard, Budgeted Maximum Length Edge Guard, Budgeted Maximum Value Vertex Guard with Painting Placement *και* Budgeted Maximum Value Edge Guard with Painting Placement *σε πολύγωνα με ή χωρίς τρύπες είναι APX-complete.*

Κεφάλαιο 11

Γραφοθεωρητικά προβλήματα

11.1 Διαμέριση του πολυγώνου σε k κλίκες

Σε αυτό το κεφάλαιο θα εξετάσουμε ένα πρόβλημα σε πολύγωνα χωρίς τρύπες. Το πρόβλημα αυτό όπως θα δούμε μπορούμε να το διατυπώσουμε σε γραφήματα. Θα εφαρμόσουμε τη μέθοδο των προηγούμενων κεφαλαίων.

Έστω ένα πολύγωνο P χωρίς τρύπες, με βάρη στις κορυφές του και ένας ακέραιος $k > 0$. Ζητάμε k κυρτά υποπολύγωνα C_i του P (πιθανώς επικαλυπτόμενα) με $V(C_i) \subseteq V(P)$ έτσι ώστε το βάρος των κορυφών των υποπολυγώνων να είναι μέγιστο.

Αυτό το πρόβλημα, εκτός από θεωρητικό ενδιαφέρον, έχει εφαρμογές σε ασύρματα δίκτυα επικοινωνίας:

Έστω ένας αριθμός από πόλεις μαζί με τους πληθυσμούς τους. Αυτές οι πόλεις μπορούν να μοντελοποιηθούν σαν κορυφές με βάρη ενός απλού πολυγώνου χωρίς τρύπες. Ζητάμε να σχεδιάσουμε το πολύ k ασύρματα δίκτυα, έτσι ώστε να μπορεί να επικοινωνεί ένας μέγιστος αριθμός ανθρώπων και όλες οι πόλεις που βρίσκονται στο ίδιο δίκτυο να μπορούν να επικοινωνούν απ' ευθείας και όχι μέσω κάποιου σταθμού.

Αντί για κυρτά υποπολύγωνα είναι ισοδύναμο να ψάχνουμε για κλίκες στο γράφο ορατότητας του πολυγώνου.

Ένα κυρτό υποπολύγωνο C ενός πολυγώνου P έχει γράφο ορατότητας που είναι υπογράφημα του γράφου ορατότητας του P.Μάλιστα αυτό το υπογράφημα είναι κλίκα.

Μέσω του γράφου ορατότητας λοιπόν, μπορούμε να περάσουμε από προβλήματα υπολογιστικής γεωμετρίας σε προβλήματα γραφημάτων. Μπορούμε επίσης να αντιμετωπίσουμε διάφορα προβλήματα γραφημάτων με πολύγωνα. Δυστυχώς το μόνο που ξέρουμε για το πρόβλημα της αναγνώρισης ενός γράφου ορατότητας, είναι ότι βρίσκεται στο PSPACE [19]. Δηλαδή δεν είναι γνωστός ένας γρήγορος αλγόριθμος που να αποφασίζει αν ένας δεδομένος γράφος είναι γράφος ορατότητας για κάποιο πολύγωνο ή όχι.

Συνεπώς για να μπορέσουμε να αντιμετωπίσουμε ένα πρόβλημα γραφημάτων χρησιμοποιώντας πολύγωνα θα πρέπει να ξέρουμε ότι το γράφημα είναι γράφος ορατότητας κάποιου πολυγώνου. Συνήθως μας δίνουν το αντίστοιχο πολύγωνο.

Για παράδειγμα το πρόβλημα της εύρεσης μιας μέγιστης κλίκας σε ένα γράφο είναι NP-hard, ενώ δεν προσεγγίζεται ούτε καν με λογαριθμικό παράγοντα. Αν όμως ο γράφος είναι γράφος ορατότητας, τότε το πρόβλημα ανήκει στο P. Δηλαδή μπορούμε να βρούμε ένα μέγιστο κυρτό υποπολύγωνο του αρχικού κυρτού πολυγώνου σε πολυωνυμικό χρόνο.

Παρόμοια προβλήματα με αυτό που μελετούμε εδώ είναι:

Το πρόβλημα του κατακερματισμού του πολυγώνου σε ένα ελάχιστο αριθμό από κυρτά πολύγωνα που βρίσκονται μέσα στο πολύγωνο. Εάν θέλουμε τα κυρτά πολύγωνα να μην επικαλύπτονται, τότε το πρόβλημα που ονομάζεται Minimum Convex Decomposition μπορεί να λυθεί σε πολυωνυμικό χρόνο ([6, 7]). Εάν τα κυρτά πολύγωνα επιτρέπεται να επικαλύπτονται, τότε το πρόβλημα ονομάζεται Minimum Convex Cover και είναι APX-hard, ενώ είναι γνωστός μόνο λογαριθμικής προσέγγισης αλγόριθμος ([15]).

Το πρόβλημα Maximum Weight In k Cliques είναι NP-hard, καθώς το αντίστοιχο πρόβλημα απόφασης είναι NP-hard.

Πρόταση 11.1 *Το πρόβλημα* Maximum Weight In k Cliques *είναι NP-hard.*

Απόδειξη. Θα ανάγουμε το πρόβλημα απόφασης Minimum Clique Partition On Visibility Graph στο πρόβλημα απόφασης Maximum Weight In k Cliques. Έστω λοιπόν ότι μας δίνεται ένας γράφος ορατότητας και ένας ακέραιος k και μας ρωτούν αν μπορεί ο γράφος να χωριστεί σε το πολύ k κλίκες. Αυτό είναι ένα στιγμιότυπο του προβλήματος απόφασης Minimum Clique Partition On Visibility Graph.

Παίρνουμε τον ίδιο γράφο και βάζουμε βάρος 1 σε κάθε του κορυφή. Ρωτάμε: με το πολύ k κλίκες μπορούμε να καλύψουμε βάρος $\geq n$; (όπου n ο αριθμός των κορυφών). Αυτό είναι ένα στιγμιότυπο του προβλήματος απόφασης Maximum Weight In k Cliques.

Αν η απάντηση του προβλήματος απόφασης Maximum Weight In k Cliques είναι 'ΝΑΙ', τότε σημαίνει ότι ο γράφος μπορεί να χωριστεί σε k κλίκες, δηλαδή η απάντηση του προβλήματος απόφασης Minimum Clique Partition On Visibility Graph είναι 'ΝΑΙ'.

Αν η απάντηση του προβλήματος απόφασης Maximum Weight In k Cliques είναι 'ΟΧΙ', τότε και η απάντηση του προβλήματος απόφασης Minimum Clique Partition On Visibility Graph είναι 'ΟΧΙ'.

Ξέρουμε ότι το Minimum Clique Partition On Visibility Graph είναι NP-hard, συνεπώς το πρόβλημα Maximum Weight In k Cliques είναι NP-hard. □

Θα περιγράψουμε τώρα έναν αλγόριθμο πολυωνυμικού χρόνου ο οποίος προσεγγίζει κατά ένα σταθερό παράγοντα το πρόβλημα Maximum Weight In k Cliques. Ο αλγόριθμος αυτός είναι ένας greedy επαναληπτικός αλγόριθμος ο οποίος σε κάθε βήμα βρίσκει μια κλίκα μέγιστου βάρους στο γράφο ορατότητας.

Το πρόβλημα της εύρεσης μιας κλίκας μέγιστου βάρους σε ένα γενικό γράφο είναι NP-hard. Σε ένα γράφο ορατότητας όμως, το πρόβλημα Maximum Weighted Clique On Visibility Graph ανήκει στο P ([3, 13, 14]).

Ακολουθεί ο προσεγγιστικός αλγόριθμος που προσεγγίζει το πρόβλημα Maximum Weight In k Cliques.

Algorithm 19 MaximumWeightinkCliques (* greedy *)

$SOL \leftarrow \emptyset$;
for $i = 1$ to k **do**
  $S_i \leftarrow$ MaxWeightedClique(V_G)
  $SOL \leftarrow SOL \cup V(S_i)$
  $W(V(S_i)) \leftarrow 0$
end for
return $W(SOL)$

Ο greedy αλγόριθμος 19 δουλεύει ως εξής: σε κάθε βήμα βρίσκει μια κλίκα μεγίστου βάρους χρησιμοποιώντας κάποιον από τους γνωστούς αλγόριθμους. Βάζει τις κορυφές της κλίκας στο σύνολο της λύσης και μηδενίζει τα βάρη τους. Ο αλγόριθμος 19 λοιπόν σε κάθε βήμα προκαλεί μια μέγιστη αύξηση στη συνολική λύση, όπως ακριβώς και στο πρόβλημα Maximum Value Vertex Guard.

Θεώρημα 11.1 *Ο αλγόριθμος 19 τρέχει σε πολυωνυμικό χρόνο και πετυχαίνει προσέγγιση* 1.58 *της βέλτιστης λύσης του προβλήματος* Maximum Weight In k Cliques.

Θεώρημα 11.2 *Το πρόβλημα* Maximum Weight In k Cliques *δεν επιδέχεται FPTAS, εκτός αν* $P = NP$.

Απόδειξη. Έστω ότι υπάρχει FPTAS για αυτό το πρόβλημα. Θα αποδείξουμε ότι χρησιμοποιώντας αυτό το FPTAS μπορούμε να λύσουμε σε πολυωνυμικό χρόνο το πρόβλημα απόφασης Minimum Clique Partition On Visibility Graph.

Ας υποθέσουμε λοιπόν ότι υπάρχει ένας αλγόριθμος ο οποίος πετυχαίνει προσέγγιση $1-\epsilon$, $\forall \epsilon > 0$ σε πολυωνυμικό χρόνο ως προς την είσοδο και το $\frac{1}{\epsilon}$.

Ας θυμηθούμε επίσης το πρόβλημα απόφασης Minimum Clique Partition On Visibility Graph: Έχουμε ένα γράφο ορατότητας με n κόμβους και έναν ακέραιο $k > 0$. Θέλουμε να αποφασίσουμε αν ο γράφος μπορεί να χωριστεί σε k κλίκες.

Μετασχηματίζουμε αυτό το στιγμιότυπο του προβλήματος Minimum Clique Partition On Visibility Graph σε στιγμιότυπο του προβλήματος Maximum Weight In k Cliques ως εξής:

Παίρνουμε τον ίδιο γράφο ορατότητας και βάζουμε βάρος 1 σε κάθε κόμβο. Ρωτάμε: με k κλίκες ποιο είναι το μέγιστο βάρος κόμβων που μπορούμε να καλύψουμε; Εφαρμόζουμε το FPTAS με $\epsilon = \frac{1}{n}$.

Έστω SOL είναι η λύση που μας επιστρέφει το FPTAS και OPT μια βέλτιστη λύση.

- Αν $SOL = n$ τότε $OPT = n$, δηλαδή με k κλίκες μπορούν να καλυφθούν όλοι οι κόμβοι, συνεπώς η απάντηση στο πρόβλημα απόφασης Minimum Clique Partition On Visibility Graph είναι 'ΝΑΙ'.

- Αν $SOL \leq n-1$ τότε: από την ύπαρξη του FPTAS ισχύει:

  $$SOL > (1-\epsilon)OPT \rightarrow (1-\frac{1}{n})OPT < SOL$$

  και από την παραπάνω σχέση:

  $$(1-\frac{1}{n})OPT < n-1 \rightarrow OPT < n$$

  δηλαδή με k κλίκες δεν μπορούν να καλυφθούν όλοι οι κόμβοι, συνεπώς η απάντηση στο πρόβλημα απόφασης Minimum Clique Partition On Visibility Graph είναι 'ΟΧΙ'.

Δηλαδή βλέπουμε ότι χρησιμοποιώντας το FPTAS μπορούμε σε χρόνο

$$Poly(n+\epsilon) = Poly(n)$$

να λύσουμε ένα NP-hard πρόβλημα απόφασης. Συνεπώς, εκτός αν $P = NP$, δε μπορεί να υπάρχει FPTAS για το πρόβλημα Maximum Weight In k Cliques. $\square$

Κεφάλαιο 12

Καινούρια Προβλήματα

12.1 Ορατότητα πολυγώνων

Στην περίπτωση του προβλήματος περιφρούρησης μέγιστης αξίας της περιμέτρου ενός πολυγώνου από δεδομένο αριθμό φυλάκων (Maximum Value Vertex Guard), είχαμε θέσει τον περιορισμό κάθε τμήμα (πίνακας) που ανήκει στο σύνολο των λύσεων να φαίνεται ολόκληρο από τουλάχιστον ένα φύλακα.

Τι γίνεται όμως στην περίπτωση που άρουμε αυτόν τον περιορισμό;

Με άλλα λόγια στο σύνολο λύσης μπορούμε να έχουμε τμήματα τα οποία δεν φαίνονται ολόκληρα από ένα φύλακα αλλά από περισσότερους.

Για παράδειγμα στο σχήμα 3.2 του κεφαλαίου 3 το τμήμα (a, b) δε φαινόταν ολόκληρο από την κορυφή 1 ή 2. Συνεπώς σε αυτό το κομμάτι του πολυγώνου, εάν ο αλγόριθμος 3 που δώσαμε στην ενότητα 3.1 επέλεγε να τοποθετήσει φύλακες στις κορυφές 1 και 2 (στο σχήμα 3.2 φαίνεται ένα κομμάτι του πολυγώνου) τότε το τμήμα (a, b) δεν θα το περιφρουρούσε ολόκληρο ένας φύλακας. Εάν λοιπόν τώρα μας ενδιαφέρει απλά το τμήμα (a, b) να περιφρουρείται, τότε ακόμη και αν υπήρχαν φύλακες μόνο στις κορυφές 1 και 2, το (a, b) περιφρουρείται.

Ας δούμε πρώτα την περίπτωση στην οποία ενδιαφερόμαστε για τον ελάχιστο αριθμό φυλάκων που χρειάζονται για να φρουρούν έναν αριθμό δεδομένων τμημάτων στην περίμετρο ενός πολυγώνου (δηλαδή όχι αναγκαστικά όλη την περίμετρο).

Αν κάνουμε την FVS κατασκευή, χωρίζουμε την περίμετρο του πολυγώνου σε $O(n^2)$ τμήματα με τις γνωστές ιδιότητες. Αν εξαιρέσουμε εκείνα τα FVS τμήματα τα οποία δεν έχουν σημεία τομής (εκτός ίσως των άκρων τους) με τα δεδομένα τμήματα (πίνακες), τότε το πρόβλημα της φύλαξης όλων των δεδομένων τμημάτων είναι ισοδύναμο με το πρόβλημα της φύλαξης όλων των FVS τμημάτων. Δηλαδή όταν ενδιαφερόμαστε για την ακριβή λύση, όλα τα δεδομένα τμήματα φυλάσσονται αν και μόνο αν φυλάσσονται όλα τα FVS τμήματα.

Συνεπώς το πρόβλημα και σ' αυτήν την περίπτωση είναι NP-hard. Υποστηρίζουμε ότι το πρόβλημα είναι επίσης APX-hard, ενώ αν εφαρμόσουμε τον αλγόριθμο που διαλέγει κάθε φορά να τοποθετήσει φύλακα στον κόμβο εκείνο από τον οποίο φαίνονται τα περισσότερα FVS τμήματα, πετυχαίνουμε $\log n$ προσέγγιση.

Ας δούμε τώρα τι γίνεται στην περίπτωση που μας δίνεται ένας αριθμός τμημάτων με αξία και θέλουμε, χρησιμοποιώντας έναν δοσμένο αριθμό k φυλάκων, να φρουρούμε τη μέγιστη αξία.

Ο αλγόριθμος που δώσαμε στο κεφάλαιο 3 θα έκανε σπατάλη φυλάκων όπως είδαμε και προηγουμένως. Μάλιστα, σε αυτή την περίπτωση δεν φαίνεται να δουλεύει κάποιος greedy αλγόριθμος, αφού η αξία που περιφρουρείται σε κάποιο βήμα με την τοποθέτηση ενός επιπλέον φύλακα μπορεί να είναι πολύ μεγαλύτερη από την αξία που θα περιφρουρούσε ο φύλακας αν είχε προστεθεί σε άλλο βήμα.

Το πρόβλημα βέβαια παραμένει APX-hard.

12.2 Generalized Set Cover

Από τη μελέτη του προηγούμενου προβλήματος οδηγηθήκαμε στον ορισμό ενός γενικότερου προβλήματος σε σύνολα:

Έστω U μια συλλογή αντικειμένων τα οποία έχουν αξία. Κάθε αντικείμενο είναι ένα σύνολο δομικών στοιχείων. Δίνονται επίσης μια συλλογή C από σύνολα τα οποία περιέχουν δομικά στοιχεία και ένας ακέραιος $k > 0$. Ο σκοπός του προβλήματος Generalized Set Cover είναι να βρεθούν k σύνολα $S_i \in C$ έτσι ώστε η αξία των αντικειμένων που καλύπτονται να είναι μέγιστη. Ένα αντικείμενο καλύπτεται όταν έχουμε πάρει όλα τα δομικά στοιχεία του.

Η διαφορά από το κλασσικό πρόβλημα κάλυψης συνόλων Set Cover είναι ότι τώρα ένα σύνολο μπορεί να καλύπτει μέρος από κάποιο αντικείμενο και όχι ολόκληρο.

Πρώτη περίπτωση: Ζητάμε να βρούμε τον ελάχιστο αριθμό συνόλων έτσι ώστε να καλύπτουμε όλη τη συλλογή.

Δεύτερη περίπτωση: Ζητάμε να βρούμε τη μέγιστη αξία από αντικείμενα που μπορούμε να καλύψουμε παίρνοντας ένα δεδομένο αριθμό συνόλων.

Στην πρώτη περίπτωση, εάν επιλέγουμε κάθε φορά το σύνολο που καλύπτει τα περισσότερα δομικά στοιχεία οδηγούμαστε σε $\log n$ προσέγγιση για τον αριθμό των φυλάκων.

Στην δεύτερη περίπτωση, εάν παίρνουμε κάθε φορά εκείνο το σύνολο που μεγιστοποιεί τα δομικά στοιχεία, δεν φτάνουμε αναγκαστικά σε κάποια σταθερή προσέγγιση του αριθμού ή της αξίας των αντικειμένων.

Κεφάλαιο 13

Συμπεράσματα και Ανοιχτά Προβλήματα

Σε αυτή την εργασία περιγράψαμε μεθόδους κατακερματισμού ενός πολυγώνου, δώσαμε προσεγγιστικούς αλγόριθμους πολυωνυμικού χρόνου για διάφορα προβλήματα μεγιστοποίησης της κάλυψης σε πολυγωνικές περιοχές, αποδείξαμε ότι αυτά τα προβλήματα είναι APX-hard και τέλος εφαρμόσαμε παρόμοιες greedy μεθόδους προσέγγισης σε προβλήματα γραφημάτων.

Αρχικά περιγράψαμε τρόπους κατακερματισμού της περιμέτρου και του εσωτερικού του πολυγώνου ως προς την ορατότητα.

Στη συνέχεια ορίσαμε το πρόβλημα της περιφρούρησης μέγιστου μήκους της περιμέτρου ενός πολυγώνου με k φύλακες (Maximum Length Vertex Guard). Αποδείξαμε ότι το πρόβλημα είναι NP-hard και δώσαμε αλγόριθμους πολυωνυμικού χρόνου που πετυχαίνουν σταθερή προσέγγιση ως προς τις βέλτιστες λύσεις για τις περιπτώσεις τοποθέτησης των φυλάκων σε κορυφές, ή πλευρές καθώς και για την περίπτωση πολύγωνων με τρύπες.

Έπειτα ορίσαμε και μελετήσαμε το πρόβλημα της κάλυψης μέγιστης αξίας της περιμέτρου (Maximum Value Vertex Guard). Αποδείξαμε ότι το πρόβλημα είναι NP-hard και δώσαμε αλγόριθμους πολυωνυμικού χρόνου για τις περιπτώσεις τοποθέτησης φυλάκων σε κορυφές ή πλευρές και για την περίπτωση πολύγωνων με τρύπες. Όλοι οι αλγόριθμοι που δώσαμε πετυχαίνουν σταθερό προσεγγιστικό παράγοντα. Ορίσαμε επίσης την έννοια της επίβλεψης έργων τέχνης και μελετήσαμε το πρόβλημα της επίβλεψης ή μερικής κάλυψης μέγιστης αξίας. Αποδείξαμε ότι το πρόβλημα παραμένει NP-hard και δώσαμε προσεγγιστικούς αλγορίθμους για τις περιπτώσεις φυλάκων - κορυφών, φυλάκων - πλευρών και πολυγώνων με τρύπες. Αποδείξαμε επίσης πως τα προβλήματα αυτά δεν έχουν προσεγγιστικά σχήματα πολυωνυμικού χρόνου. Δηλαδή τα προβλήματα αυτά δεν ανήκουν στην κλάση FPTAS (αλλά ούτε και στην κλάση PTAS όπως δείξαμε στα κεφάλαια 9 και 10).

Στη συνέχεια μελετήσαμε το πρόβλημα περιφρούρησης μέγιστης αξίας με παράμετρο την τοποθέτηση των έργων τέχνης (Maximum Value Vertex Guard with Painting Placement). Δώσαμε προσεγγιστικούς αλγόριθμους που χρησιμοποιούν ένα πολυωνυμικό σχήμα για το πρόβλημα (Multiple Knapsack). Αποδείξαμε ότι πετυχαίνουν σταθερούς παράγοντες προσέγγισης για τις περιπτώσεις φυλάκων - κορυφών, φυλάκων - πλευρών και πολυγώνων με τρύπες.

Ορίσαμε επίσης το πρόβλημα της κάλυψης μέγιστου μήκους της περιμέτρου με όριο κόστους πομποδεκτών (Budgeted Maximum Length Vertex Guard). Δώσαμε έναν προσεγγιστικό αλγόριθμο και αποδείξαμε ότι πετυχαίνει σταθερό προσεγγιστικό παράγοντα για τις περιπτώσεις τοποθέτησης πομποδεκτών σε κορυφές και πλευρές, όπως επίσης και για την περίπτωση πολυγώνων με τρύπες.

Μετά ορίσαμε το πρόβλημα της κάλυψης μέγιστης αξίας με άνω όριο κόστους πομποδεκτών (Budgeted Maximum Value Vertex Guard). Δώσαμε και για αυτό το πρόβλημα πολυωνυμικού χρόνου προσεγγιστικούς αλγόριθμους.

Κάναμε επίσης εκτός από την τοποθέτηση των πομποδεκτών, και βέλτιστη τοποθέτηση αναμεταδοτών (Budgeted Maximum Value Vertex Guard with Painting Placement).

Επεκτείναμε μερικά από τα αποτελέσματά μας μελετώντας πως μπορούμε να καλύψουμε ένα μέγιστο εμβαδόν από το εσωτερικό του πολυγώνου (Maximum Area Edge Guard).

Στη συνέχεια περιγράψαμε μια αναγωγή διατήρησης χάσματος (gap preserving reduction) από το πρόβλημα Max-5-Occurence-3-Sat στο πρόβλημα Maximum Value Vertex Guard, αποδεικνύοντας ότι το πρόβλημα Maximum Value Vertex Guard είναι APX-hard.

Κάνοντας τις κατάλληλες τροποποιήσεις στην προηγούμενη αναγωγή αποδείξαμε ότι όλα τα προβλήματα που μελετήσαμε είναι APX-complete.

Τέλος εξετάσαμε ένα πρόβλημα σε γραφήματα (Maximum Weight In k Cliques) για το οποίο δώσαμε προσεγγιστικούς αλγορίθμους και ορίσαμε ένα πολύ ενδιαφέρον πρόβλημα με σύνολα (Generalized Set Cover).

Σύμφωνα με τα αποτελέσματα αυτής της εργασίας, η ταξινόμηση των προβλημάτων κάλυψης πολυγώνων συμπληρώνεται όπως φαίνεται στο σχήμα 13.1.

Τα προβλήματα ελαχιστοποίησης σε πολύγωνα με τρύπες (συμβολίζουμε MIN (holes) στο σχήμα 13.1) είναι γνωστό ότι είναι $\log n$-hard και προσεγγίζονται με παράγοντα $\log n$ (άρα $\log n$-complete).

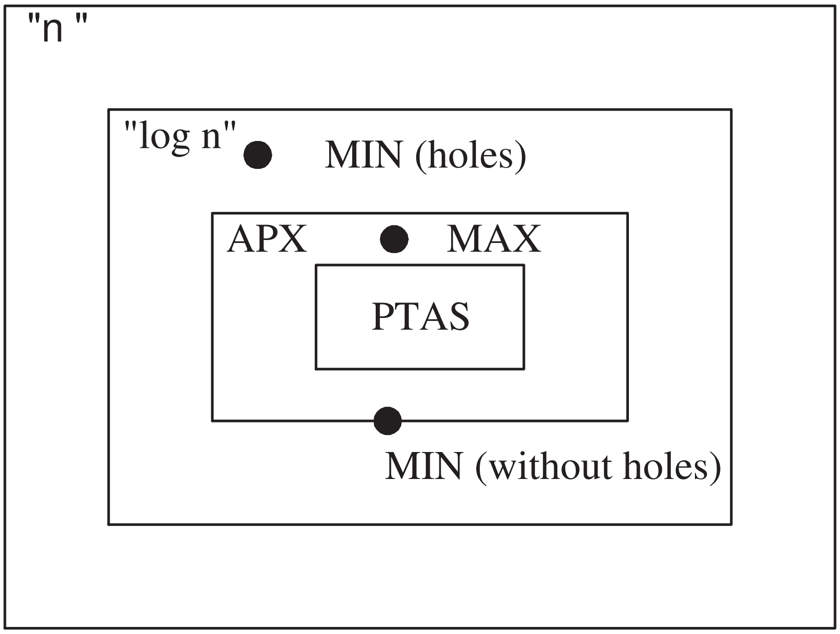
Τα προβλήματα ελαχιστοποίησης σε πολύγωνα χωρίς τρύπες (συμβολίζουμε MIN (without holes) στο σχήμα 13.1) είναι γνωστό ότι είναι APX-hard και προσεγγίζονται με παράγοντα $\log n$. Δηλαδή δεν είναι γνωστό αν είναι APX-complete ή $\log n$-complete. Γι' αυτό στο σχήμα 13.1 είναι ζωγραφισμένα πάνω στο όριο της κλάσης APX και $\log n$.

Σε αυτή την εργασία αποδείξαμε ότι τα προβλήματα μεγιστοποίησης σε πολύγωνα με ή χωρίς τρύπες (συμβολίζουμε MAX στο σχήμα 13.1) έχουν προσεγγιστικούς αλγόριθμους με σταθερούς παράγοντες προσέγγισης και είναι APX-hard. Συνεπώς είναι APX-complete.

Τα ανοιχτά προβλήματα που προκύπτουν είναι:

- Έχουμε επιτύχει τις βέλτιστες προσεγγισιμότητες για τα προβλήματα που μελετήσαμε;

- Πως μεταβάλλεται η προσεγγισιμότητα αν οι πιθανές θέσεις των φυλάκων είναι σε σημεία στις πλευρές ή στο εσωτερικό του πολυγώνου;



Σχήμα 13.1: Ταξινόμηση των Art Gallery προβλημάτων σε κλάσεις προσέγγισης. Συμβολίζουμε με: "n^ϵ" την κλάση των προβλημάτων με $O(n^\epsilon)$ παράγοντα προσέγγισης, "$\log n$" την κλάση των προβλημάτων με $O(\log n)$ παράγοντα προσέγγισης, APX την κλάση των προβλημάτων με σταθερό παράγοντα προσέγγισης και PTAS την κλάση των προβλημάτων με παράγοντα προσέγγισης οσοδήποτε κοντά στο 1.

- Πως μπορούμε να τοποθετήσουμε βέλτιστα εκτός από τους πομποδέκτες, τους αναμεταδότες στο εσωτερικό του πολυγώνου;
- Αποτέλεσμα μη-προσεγγισιμότητας για το πρόβλημα Maximum Weight In k Cliques.
- Μελέτη του καινούριου προβλήματος Generalized Set Cover.

Βιβλιογραφία

[1] Arora, S., Probabilistic Checking of Proofs and the Hardness of Approximation Problems, PhD thesis, Berkeley, 1994.

[2] Arora, S., Lund, C., Hardness of Approximations; in: Approximation Algorithms for NP-Hard Problems (Dorit Hochbaum ed.), pp. 399-446, PWS Publishing Company, 1996.

[3] Avis, D., Rappaport, D., Computing the largest empty convex subset of a set of points, Proc. 1st Ann. ACM Symposium Computational Geometry, pp. 161-167, 1985.

[4] Baker, B. S., Approximation algorithms for NP-complete problems on planar graphs, J. ACM 41, 153-180, 1994.

[5] Carlsson S., Jonsson H., Guarding a Treasury, Proc. 5th Canadian Conf. Computational Geometry, pp. 85-90, 1993.

[6] Chazelle, B., Dobkin, D., Decomposing a polygon into its convex parts, Proc. 11th Annual ACM Symp. Theory Comput., pp 38-48, 1979.

[7] Chazelle, B., Dobkin, D., Optimal convex decompositions, Computational Geometry, North-Holland, Netherlands, 1985.

[8] Chekuri, C., Khanna, S., A PTAS for the Multiple Knapsack problem, Proc. 11th ACM Symp on Discrete Algorithms, 213-222, 2000.

[9] Chvatal, V., A combinatorial theorem in plane geometry, J. Combinatorial Theory Ser. B, 13(2):395-398, 1975.

[10] Conforti, M., Cornuejols, G., Submodular functions, matroids and the greedy algorithm: tight worst case bounds and some generalizations of the Rado-Edmonds theorem, Discrete Applied Mathematics 7:257-275, 1984.

[11] Culberson, J., and Reckhow,. R., Covering Polygons is hard, Proc. 29th Symposium on Foundations of Computer Science, pp. 601 - 611, 1988.

[12] Deneen L. L., Joshi S., Treasures in an art gallery, Proc. 4th Canadian Conf. Computational Geometry, pp. 17-22, 1992.

[13] Dobkin, D., Edelsbrunner, H., Overmars, H., Searching for Empty Convex Polygons, Algorithmica 5, pp. 561-571, 1990.

[14] Edelsbrunner, H., Guibas, L., Topologically sweeping an arrangement, J. Comput. System Sci. 38, pp. 165-194, 1989.

[15] Eidenbenz, S., (In-)Approximability of Visibility Problems on Polygons and Terrains, PhD Thesis, ETH Zurich, 2000.

[16] Eidenbenz, S., Inapproximability Results for Guarding Polygons without Holes, Lecture notes in Computer Science, Vol. 1533 (ISAAC'98), p. 427-436, 1998.

[17] Eidenbenz, S. and Stamm, C., Maximum Clique and Minimum Clique Partition in Visibility Graphs, LNCS, 2000.

[18] Eidenbenz, S., Stamm, C., Widmayer, P., Inapproximability of some Art Gallery Problems, Proc. 10th Canadian Conf. Computational Geometry (CCCG'98), pp. 64-65, 1998.

[19] Everett, H., Visibility graph recognition, *Report 231/90, Dept. Comput. Sci., Univ. Toronto*, Toronto, ON, 1990, Ph.D. Thesis.

[20] Feige, U., A threshold of $\ln n$ for Approximating Set Cover; Journal of the ACM, Vol. 45 No. 4, pp. 634 - 652, 1998.

[21] Fisk, S., A short proof of Chvátal's watchman theorem, *J. Combin. Theory Ser. B* **24**, 374, 1978.

[22] Ghosh, S., Approximation algorithms for Art Gallery Problems, Proc. of the Canadian Information Processing Society Congress, pp. 429-434, 1987.

[23] Hastad, J., Clique is hard to approximate within $n^{1-\epsilon}$, Proc. of the 37th Symposium on Foundations of Computer Science, IEEE, pp. 627 - 637, 1996.

[24] Hastad, J., Some optimal inapproximability results, Proc. of the 29th Annual ACM Symposium on Theory of Computing 1 - 10, 1997.

[25] Hochbaum, D., Approximation Algorithms for NP-Hard Problems, PWS, 1996.

[26] Hochbaum, D., Approximating Covering and Packing Problems in: Approximation Algorithms for NP-Hard Problems (Dorit Hochbaum ed.), pp. 94-143, PWS Publishing Company, 1996.

[27] Hunt III, H. B., Marathe, M. V., Radhakrishnan, V., Ravi, S. S., Rosenkrantz, D. J., and Stearns, R. E., Nc-approximation schemes for NP- and PSPACE-hard problems for geometric graphs, J. Algorithms 26, 238-274, 1998.

[28] Johnson, D. S., Approximation algorithms for combinatorial problems, J. Comput. System Sci. 9, 256-278, 1974.

[29] Keil, M., Snoeyink J., On the time bound for convex decompositions of simple polygons, *10th Canadian Conference on Computational Geometry*, 1998.

[30] Khuller, S., Moss, A., Naor, J., The Budgeted Maximum Coverage Problem, Information Processing Letters 70(1): 39-45 (1999).

[31] Laurentini A., Guarding the walls of an art gallery, The Visual Computer Journal, (1999) 15:265-278.

[32] Lee, D., Lin, A., Computational complexity of art gallery problems, IEEE Trans. Inform. Theory 32, 276-282, 1986.

[33] Lingas, A., The Power of Non-Rectilinear Holes, Proc. 9th Colloquium on Automata, Languages and Programming, pp. 445 - 454, 1999.

[34] Lund, C., Yannakakis, M., On the hardness of approximating minimization problems, *Proc. 25th ACM Symp. on the Theory of Computing*, 286-295, 1993.

[35] Markou, E., Fragoudakis, C., Zachos, S., Approximating Visibility Problems within a constant, 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks, Rome pp. 91-103, 2002.

[36] Markou, E., C., Zachos, S., Fragoudakis, C., Maximizing the Guarded Boundary of an Art Gallery is APX-complete, 5th Italian Conference on Algorithms and Complexity, Rome, pp. 2003.

[37] Markou, E., C., Zachos, S., Fragoudakis, C., Budgeted Coverage of a Maximum Part of a Polygonal Area, 2003 (submitted).

[38] Markou, E., C., Zachos, S., Fragoudakis, C., Optimal placement of tranceivers and broadcasters in a polygon, 2003 (submitted).

[39] Markou, E., C., Zachos, S., Fragoudakis, C., Maximizing the Guarded Interior of a polygonal area, 2003 (submitted).

[40] Nemhauser, G., Wolsey, L., Maximizing submodular set functions: formulations and analysis of algorithms. In Studies of Graphs and Discrete Programming North-Holland, Amsterdam, 279-301, 1972.

[41] O'Rourke, J., Art Gallery Theorems and Algorithms, Oxford Univ. Press, New York, 1987.

[42] O'Rourke, J., Streinu, I., The Vertex-Edge Visibility Graph of a Polygon, 1997.

[43] Papadimitriou, C. H., and Yannakakis, M., Optimization, approximation, and complexity classes, J. Comput. System Sci. 43, 425-440, 1991.

[44] Paz, A., Moran, S., Non Deterministic polynomial optimization problems and their approximations, *Theoretical Comput. Sci.* **15**, 251-277, 1981.

[45] Raz, R., and Safra, S., A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP, Proc. 29th Ann. ACM Symp. on Theory of Comp., ACM, 475-484,1997.

[46] Shermer, T., Recent results in Art Galleries, Proc. of the IEEE, 1992.

[47] Shmoys, D., Tardos, E., An approximation algorithm for the generalized assignment problem, Mathematical Programming A, 62:461-74, 1993.

[48] Urrutia, J., Art gallery and Illumination Problems, Handbook on Comput. Geometry, 1998.

[49] Vohra, R., Hall, N., A probabilistic analysis of the maximal covering location problem, Discrete Applied Mathematics 43:175-183, 1993.

Γλωσσάρι

γράφος ορατότητας

προβλήματα με εγγύηση

αναγωγή διατήρησης χάσματος

περιφρούρηση

επίβλεψη

finest visibility segmentation

πομποδέκτης **(tranceiver)**

αναμεταδότης **(broadcaster)**

ασύρματα δίκτυα επικοινωνίας

point to point communication

point to station communication

art gallery

κάλυψη

μερική κάλυψη

φύλακας σε κορυφή

φύλακας-πλευρά

Δημοσιεύσεις

Coloring Paths In A Tree Nomikos, C., Zachos, S., Markou, E., 6th Panhellenic Conference in Computer Science, Athens, pp. 578 - 588, 1997.

Approximating Visibility Problems within a constant Markou, E., Fragoudakis, C., Zachos, S., 3rd Workshop on Approximation and Randomization Algorithms in Communication Networks, Rome pp. 91-103, 2002.

Maximizing the Guarded Boundary of an Art Gallery is APX-complete Markou, E., Zachos, S., Fragoudakis, C., 5th Italian Conference on Algorithms and Complexity, Rome, pp. 24 - 35, 2003.

Budgeted Coverage of a Maximum Part of a Polygonal Area Markou, E., Zachos, S., Fragoudakis, C., 1st Balkan Conference on Informatics, Thessaloniki, 2003 (accepted).

Optimal placement of tranceivers and broadcasters in a polygon Markou, E., Fragoudakis, C., Zachos, S., 2003 (submitted).

Maximizing the Guarded Interior of a polygonal area Markou, E., Fragoudakis, C., Zachos, S., 2003 (submitted).