

Το Πρόβλημα της Ελάχιστης Κάλυψης Ορθογωνίου

Πολυγώνου με s -Αστέρες

Η Μεταπτυχιακή Εργασία Εξειδίκευσης

υποβάλλεται στην ορισθείσα

από τη Συνέλευση

του Τμήματος Μηχανικών Η/Υ και Πληροφορικής

Εξεταστική Επιτροπή

από την

Ειρήνη Γορδίου

ως μέρος των υποχρεώσεων για την απόκτηση του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ

ΣΤΗ ΘΕΩΡΙΑ ΕΠΙΣΤΗΜΗΣ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Πανεπιστήμιο Ιωαννίνων

Πολυτεχνική Σχολή

Ιωάννινα 2021

Εξεταστική επιτροπή:

- **Λεωνίδας Παληός**, Καθηγητής, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων (Επιβλέπων)
- **Λουκάς Γεωργιάδης**, Αναπληρωτής Καθηγητής, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων
- **Χρήστος Νομικός**, Επίκουρος Καθηγητής, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων

ΑΦΙΕΡΩΣΗ

Αφιερώνεται στην οικογένειά μου και στον Μιχάλη.

ΕΥΧΑΡΙΣΤΙΕΣ

Η εκπόνηση της παρούσας διπλωματικής εργασίας δε θα ήταν δυνατή χωρίς τη βοήθεια ορισμένων ανθρώπων, τους οποίους θα ήθελα να ευχαριστήσω. Πρωτίστως, τον επιβλέποντα καθηγητή μου, κύριο Λεωνίδα Παληό, για την επιστημονική καθοδήγηση, την υπομονή και τη συνεχή υποστήριξή του κατά την εκπόνηση της παρούσας εργασίας. Ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στον κ. Γεωργιάδη και στον κ. Νομικό για τη μελέτη της αναφοράς του κειμένου της εργασίας μου και για τα χρήσιμα σχόλιά τους. Επίσης, θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξή τους καθ' όλη τη διάρκεια των σπουδών μου. Τέλος, θα ήθελα να ευχαριστήσω την εταιρεία στην οποία εργάζομαι για την υποστήριξη έως το πέρας των μεταπτυχιακών μου σπουδών.

ΠΕΡΙΕΧΟΜΕΝΑ

Καταλογος Σχηματων	iii
Καταλογος Πινακων	v
Καταλογος Αλγοριθμων	vi
Περιληψη	vii
Extended Abstract	ix
ΚΕΦΑΛΑΙΟ 1 Εισαγωγή	1
1.1 Ορισμοί Βασικών Εννοιών.....	1
1.2 Το Αντικείμενο της Μεταπτυχιακής Εργασίας.....	11
1.3 Σχετικά Ερευνητικά Αποτελέσματα.....	12
1.4 Η Δομή της Μεταπτυχιακής Εργασίας	14
ΚΕΦΑΛΑΙΟ 2 Ο Αλγόριθμος των Motwani, Raghunathan και Saran	15
2.1 Θεωρητικό Υπόβαθρο	15
2.2 Ο Αλγόριθμος Υπολογισμού Ελάχιστου Χρωματισμού σε Ασθενώς Τριγωνικά Γραφήματα.....	19
2.3 Ο Αλγόριθμος των Motwani, Raghunathan και Saran	23
2.4 Παράδειγμα Εφαρμογής του Αλγορίθμου.....	24
2.5 Η Υλοποίηση.....	36
2.6 Παραδείγματα Εκτέλεσης του Προγράμματος.....	47
ΚΕΦΑΛΑΙΟ 3 Αλγόριθμος Κατασκευής Ορθογωνίων Πολυγώνων	80
3.1 Ο Αλγόριθμος Κατασκευής Ορθογωνίων Πολυγώνων.....	80
3.1.1 Ο Αλγόριθμος Υπολογισμού του Πίνακα M	82
3.1.2 Ο Αλγόριθμος Εξαγωγής Ορθογωνίου Πολυγώνου.....	84
3.2 Παράδειγμα Εφαρμογής του Αλγορίθμου.....	85

3.3 Η Υλοποίηση.....	92
ΚΕΦΑΛΑΙΟ 4 Συμπερασματα-Επεκτασεις	95
Βιβλιογραφια	96

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1: Παραδείγματα απλών πολυγώνων και μη απλών πολυγώνων.	2
Σχήμα 1.2: Τα είδη των ακμών και των εσοχών.....	4
Σχήμα 1.3: Κλάσεις ορθογωνίων πολυγώνων.	6
Σχήμα 1.4: Το σημείο y είναι ορατό από το σημείο x	7
Σχήμα 1.5: Παράδειγμα s -ορατότητας και r -ορατότητας.....	8
Σχήμα 1.6: Επέκταση της ακμής εσοχής D	9
Σχήμα 1.7: Διάγραμμα εσοχών (dent diagram) και οι πηγές του.	10
Σχήμα 2.1: Αρχικό πολύγωνο και εσοχές.	24
Σχήμα 2.2: Επέκταση Νότιων και Βόρειων Εσοχών.....	25
Σχήμα 2.3: Επέκταση Δυτικών Εσοχών.....	26
Σχήμα 2.4: Επέκταση Ανατολικών Εσοχών.....	27
Σχήμα 2.5: Κατασκευή dent diagram.....	28
Σχήμα 2.6: Απεικόνιση των περιοχών.	29
Σχήμα 2.7: Απεικόνιση των κόμβων του γραφήματος H_s	31
Σχήμα 2.8: Απεικόνιση των κόμβων του γραφήματος H	33
Σχήμα 2.9: Απεικόνιση των κλικών.....	35
Σχήμα 2.10: Οπτικοποίηση πολυγώνου εισόδου.....	36
Σχήμα 2.11: Οπτικοποίηση πηγών και καταβοθρών, με γκρι και μπλε χρώμα αντι- στοίχως.	37
Σχήμα 2.12: Η ημι-ακμή \vec{e} και η δίδυμή της $\text{Twin}(\vec{e})$	38
Σχήμα 2.13: Οπτικοποίηση πολυγώνου εισόδου.....	49
Σχήμα 2.14: Οπτικοποίηση πηγών και καταβοθρών.....	57
Σχήμα 2.15: Οπτικοποίηση πολυγώνου εισόδου.....	66

Σχήμα 2.16: Οπτικοποίηση πηγών και καταβοθρών.....	73
Σχήμα 3.1: Επέκταση πλευράς κατά E.	81
Σχήμα 3.2: Παράδειγμα δημιουργίας σημείου και ευθύγραμμου τμήματος.....	84
Σχήμα 3.3: Παράδειγμα εφαρμογής του αλγορίθμου.....	86
Σχήμα 3.4: Παράδειγμα εφαρμογής του αλγορίθμου.....	86
Σχήμα 3.5: Απεικόνιση τελικού πολυγώνου	86
Σχήμα 3.6: Παράδειγμα του πίνακα M.....	87
Σχήμα 3.7: Σχηματισμός πολυγώνου	88
Σχήμα 3.8: Αφαίρεση περιττών σημείων.....	91
Σχήμα 3.9: Οπτικοποίηση τελικού πολυγώνου	92

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 2.1: Πίνακας γειτνίασης για το γράφημα H_s	30
Πίνακας 2.2: Πίνακας γειτνίασης για το γράφημα H	32
Πίνακας 2.3: Πίνακας γειτνίασης για το γράφημα H_s	61
Πίνακας 2.4: Πίνακας γειτνίασης για το γράφημα H	62
Πίνακας 2.5: Πίνακας γειτνίασης για το γράφημα H_s	76
Πίνακας 2.6: Πίνακας γειτνίασης για το γράφημα H	77

ΚΑΤΑΛΟΓΟΣ ΑΛΓΟΡΙΘΜΩΝ

Αλγόριθμος Υπολογισμού 2-ζεύγους (G).....	20
Αλγόριθμος W-OPT (G).....	21
Αλγόριθμος W-OPT unweighted (G).	22

ΠΕΡΙΛΗΨΗ

Ειρήνη Γορδίου, Δ.Μ.Σ. στη Θεωρία Επιστήμης των Υπολογιστών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, Αύγουστος 2021

Το Πρόβλημα της Ελάχιστης Κάλυψης Ορθογωνίου Πολυγώνου με s -Αστέρες
Επιβλέπων: Λεωνίδα Παληός, Καθηγητής

Ένα πολύγωνο στο επίπεδο λέγεται ορθογώνιο εάν κάθε ακμή του είναι είτε οριζόντια είτε κατακόρυφη. Δύο σημεία ενός ορθογωνίου πολυγώνου P είναι s -ορατά το ένα από το άλλο εάν μπορούν να συνδεθούν εντός του P με μία κλιμακοειδή διαδρομή, δηλαδή μια μονότονη διαδρομή με εναλλάξ οριζόντια και κατακόρυφα τμήματα (μια τέτοια διαδρομή αν ξεκινήσει με κατεύθυνση προς τα δεξιά και άνω συνεχίζει προς τα δεξιά και άνω έως το τέλος της). Τότε ένα ορθογώνιο πολύγωνο Q λέγεται s -αστέρας αν υπάρχει σημείο του από το οποίο κάθε άλλο σημείο του Q είναι s -ορατό. Τέλος, μια συλλογή S από πολύγωνα αποτελεί κάλυμμα ενός πολυγώνου R αν κάθε σημείο του R ανήκει σε κάποιο πολύγωνο στη συλλογή S (δηλαδή το R ανήκει πλήρως στην ένωση των πολυγώνων στη συλλογή S).

Στην παρούσα μεταπτυχιακή εργασία μελετήσαμε και υλοποιήσαμε τον αλγόριθμο των Motwani, Raghunathan και Saran για τον υπολογισμό ενός καλύμματος ενός γενικού ορθογωνίου πολυγώνου (χωρίς οπές) από το ελάχιστο πλήθος s -αστέρων. Ο αλγόριθμος συνδυάζει ιδιότητες ορατότητας με αποτελέσματα Θεωρίας Γραφημάτων και έχει πολυπλοκότητα χρόνου $O(n^8)$ όπου n είναι το πλήθος κορυφών του δοθέντος ορθογωνίου πολυγώνου. Επιπλέον, παρουσιάζουμε έναν αλγόριθμο για την κατασκευή «τυχαίων» ορθογωνίων πολυγώνων εντός δοθέντος ορθογωνίου παραλληλογράμμου με βάση δοθέντα συντελεστή «πυκνότητας».

Η υλοποίηση των αλγορίθμων πραγματοποιήθηκε στη γλώσσα προγραμματισμού C++, ενώ η οπτικοποίηση των πολυγώνων με βιβλιοθήκες της OpenGL.

EXTENDED ABSTRACT

Eirini Gordiou, M.Sc. in Theory of Computer Science, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, August 2021

The Problem of Covering Orthogonal Polygons with the Minimum Number of s-Stars
Advisor: Leonidas Palios, Professor

A polygon in a plane is called orthogonal if each of its edges is either horizontal or vertical. Two points of an orthogonal polygon are s-visible from each other if they can be connected within P by a staircase path, i.e. a monotone path with alternating horizontal and vertical line segments (such a path if it starts in a right and upward direction keeps proceeding right and upwards until its end). Then an orthogonal polygon Q is called s-star if there is a point from which every other point of Q is s-visible. Finally, a set S of polygons is the cover of a polygon R if every point on R belongs to a polygon in S (i.e. R belongs entirely to the union of the polygons in S).

In the present Master's thesis we study and implement the algorithm of Motwani, Raghunathan and Saran to compute a cover of a simple orthogonal polygon (without holes) consisting of the minimum number of s-stars. The algorithm combines visibility properties with Graph Theory results and has $O(n^8)$ time complexity where n is the number of vertices of the given orthogonal polygon. In addition, we present an algorithm for constructing "random" orthogonal polygons within a given rectangle based on a given "density" coefficient.

The algorithms have been implemented in the C++ programming language, while the polygons have been visualized with OpenGL libraries.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

-
- 1.1 Ορισμοί Βασικών Εννοιών
 - 1.2 Το Αντικείμενο της Μεταπτυχιακής Εργασίας
 - 1.3 Σχετικά Ερευνητικά Αποτελέσματα
 - 1.4 Η Δομή της Μεταπτυχιακής Εργασίας
-

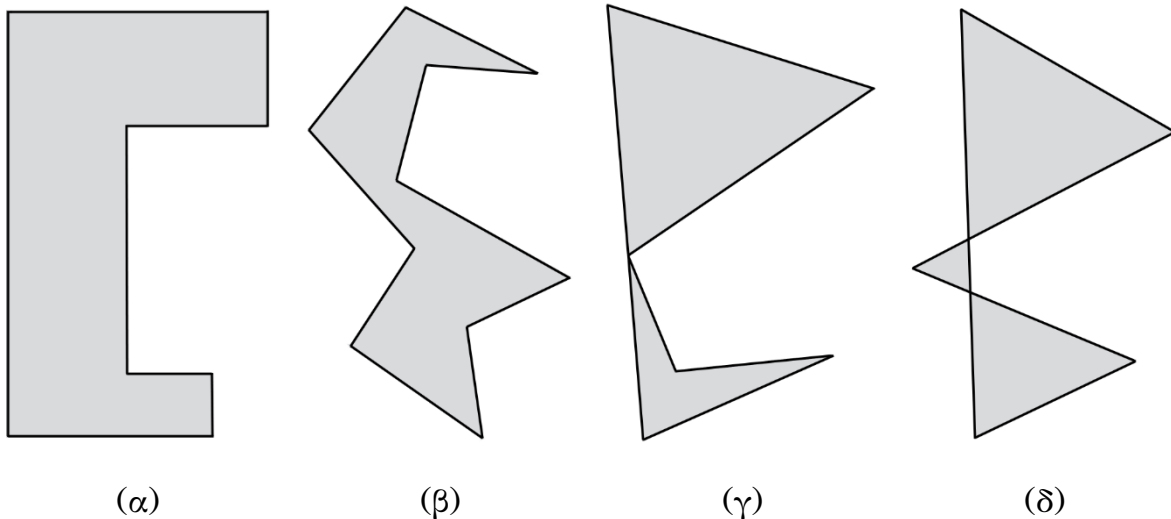
1.1 Ορισμοί Βασικών Εννοιών

Σε αυτήν την ενότητα θα παρουσιάσουμε βασικές έννοιες και ορισμούς που είναι χρήσιμοι για την κατανόηση του προβλήματος που θα μας απασχολήσει στην παρούσα μεταπτυχιακή εργασία. Συγκεκριμένα, τα πολύγωνα που θα ασχοληθούμε ανήκουν στην κατηγορία των κυρτών ορθογώνιων πολυγώνων. Παρακάτω παρουσιάζονται οι ορισμοί του απλού πολυγώνου (simple polygon), του ορθογώνιου πολυγώνου (orthogonal polygon), των εσοχών (dents) και, τέλος, των περιοχών (region) ενός πολυγώνου.

Ένα απλό πολύγωνο (*simple polygon*) είναι η περιοχή του επιπέδου που περικλείεται από ένα πεπερασμένο σύνολο ευθυγράμμων τμημάτων τα οποία σχηματίζουν μία απλή κλειστή πολυγωνική γραμμή. Έστω ότι τα u_1, u_2, \dots, u_n είναι n σημεία στο επίπεδο, και έστω ότι τα $e_1 = u_1u_2, e_2 = u_2u_3, \dots, e_n = u_nu_1$ είναι n ευθύγραμμα τμήματα που συνδέουν τα σημεία. Τότε τα τμήματα αυτά ορίζουν ένα απλό πολύγωνο αν και μόνο αν:

1. Η τομή κάθε ζεύγους διαδοχικών τμημάτων είναι το μοναδικό κοινό τους σημείο: $e_i \cap e_{i+1} = v_{i+1}$ για κάθε $i = 1, \dots, n-1$ και $e_n \cap e_1 = v_1$.
2. Μη διαδοχικά τμήματα δεν τέμνονται: $e_i \cap e_j = \emptyset$ για κάθε $1 \leq i < j - 1$ και $j \leq n$.

Ο λόγος για τον οποίον τα ευθύγραμμα τμήματα σχηματίζουν μια *πολυγωνική γραμμή* είναι γιατί είναι συνδεδεμένα στη σειρά το ένα μετά το άλλο. Ο λόγος για τον οποίον αυτή η γραμμή είναι *κλειστή* είναι γιατί τα τμήματα κλείνουν σε έναν “κύκλο”. Ο λόγος για τον οποίον αυτή η κλειστή γραμμή είναι *απλή* είναι γιατί μη διαδοχικά τμήματα δεν τέμνονται (Σχήμα 1.1).



Σχήμα 1.1: Παραδείγματα απλών πολυγώνων και μη απλών πολυγώνων. Τα πολύγωνα (α) και (β) είναι απλά, ενώ τα (γ) και (δ) δεν είναι.

Κορυφές (vertices) ονομάζονται τα σημεία v_i του πολυγώνου, ενώ *ακμές (edges)* ονομάζονται τα τμήματα e_i του πολυγώνου. Ένα πολύγωνο με n κορυφές έχει και n ακμές.

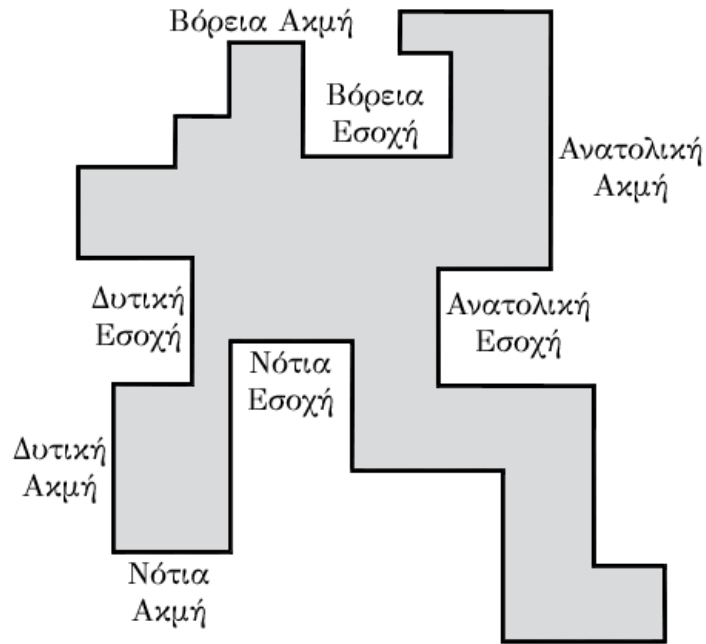
Κάθε απλή, κλειστή πολυγωνική γραμμή και γενικότερα κάθε κλειστή καμπύλη χωρίζει το επίπεδο στο οποίο βρίσκεται σε δύο μέρη:

- Το *εσωτερικό (interior)* που περικλείεται από την κλειστή καμπύλη.
- Το *εξωτερικό (exterior)* που επεκτείνεται γύρω από την κλειστή καμπύλη και προκύπτει αν αφαιρέσω από το επίπεδο το εσωτερικό της κλειστής καμπύλης.

Ως *πολύγωνο (polygon)* ορίζεται η περιοχή του επιπέδου που περικλείεται από μια κλειστή πολυγωνική γραμμή. Συχνά ένα πολύγωνο ορίζεται ως το σύνολο των ευθυγράμμων τμημάτων από τα οποία απαρτίζεται η κλειστή πολυγωνική γραμμή. Το σύνολο των ευθυγράμμων τμημάτων που περικλείουν ένα πολύγωνο P ονομάζεται *σύνορο (boundary)* του P και συμβολίζεται συνήθως με ∂P . Σύμφωνα με τον ορισμό του πολυγώνου, ισχύει ότι $\partial P \subseteq P$.

Τα απλά πολύγωνα διακρίνονται σε κυρτά και μη κυρτά. Ένα πολύγωνο είναι *κυρτό (convex)* αν και μόνο αν για οποιαδήποτε δύο σημεία a, b το ευθύγραμμο τμήμα ab ανήκει εξολοκλήρου στο P . Ο ορισμός αυτός συνεπάγεται ότι για κάθε ακμή ενός κυρτού πολυγώνου η ευθεία που διέρχεται από την ακμή αυτή, αφήνει όλα τα σημεία του πολυγώνου στο ίδιο κλειστό ημιεπίπεδο. Αντίθετα, σε ένα *μη κυρτό* πολύγωνο υπάρχει μια ακμή τέτοια ώστε η ευθεία που διέρχεται από την ακμή αυτή έχει σημεία του πολυγώνου και δεξιά και αριστερά της.

Ορθογώνιο πολύγωνο είναι ένα πολύγωνο του οποίου οι ακμές είναι παράλληλες προς τους άξονες του καρτεσιανού συστήματος συντεταγμένων. Επειδή ένα ορθογώνιο πολύγωνο έχει μόνο είτε οριζόντιες είτε κάθετες ακμές, μπορούμε να χαρακτηρίσουμε τις ακμές του σύμφωνα με τα σημεία του ορίζοντα (Βορράς, Νότος, Ανατολή, Δύση). Πιο συγκεκριμένα, ως Βόρεια ακμή χαρακτηρίζουμε μία ακμή ενός ορθογωνίου πολυγώνου όταν το κάθετο διάνυσμα στην ακμή με φορά προς τα έξω δείχνει προς τον Βορρά. Αντίστοιχα χαρακτηρίζουμε μία ακμή ως Νότια, Ανατολική και Δυτική (Σχήμα 1.2).



Σχήμα 1.2: Τα είδη των ακμών και των εσοχών.

Κυρτό ορθογώνιο πολύγωνο (OCP) είναι ένα πολύγωνο όταν το ευθύγραμμο τμήμα μεταξύ δύο οποιωνδήποτε σημείων του στην ίδια οριζόντια ή στην ίδια κατακόρυφη γραμμή ανήκει στο πολύγωνο.

Ένα από πολύγωνο P λέγεται αστεροειδές (*star polygon*) αν περιέχει κάποιο σημείο q τέτοιο ώστε για κάθε σημείο p του P το ευθύγραμμο τμήμα \overline{pq} να περιέχεται στο P .

Ως εσοχές (*dents*) ενός ορθογωνίου πολυγώνου ορίζονται οι ακμές των οποίων τα άκρα είναι μη-κυρτές κορυφές. Εάν η ακμή που ορίζει μία εσοχή είναι Βόρεια, τότε η εσοχή χαρακτηρίζεται ως Βόρεια εσοχή (N-dent). Αντίστοιχα χαρακτηρίζουμε μία εσοχή ως Νότια (S-dent), Ανατολική (E-dent) και Δυτική (W-dent) (Σχήμα 1.2).

Ένα ορθογώνιο πολύγωνο κλάσης k ($0 \leq k \leq 4$) έχει εσοχές με το πολύ k διαφορετικούς προσανατολισμούς. Επομένως, ισχύουν τα κάτωθι:

- Κλάση 0: δεν έχει καμία εσοχή άρα είναι ένα κυρτό ορθογώνιο πολύγωνο
- Κλάση 1: έχει ένα είδος εσοχών, δηλαδή είτε Βόρειες, είτε Νότιες, είτε Δυτικές, είτε Ανατολικές
- Κλάση 2: έχει δύο είδη εσοχών και χωρίζεται στις εξής δύο υποκατηγορίες:

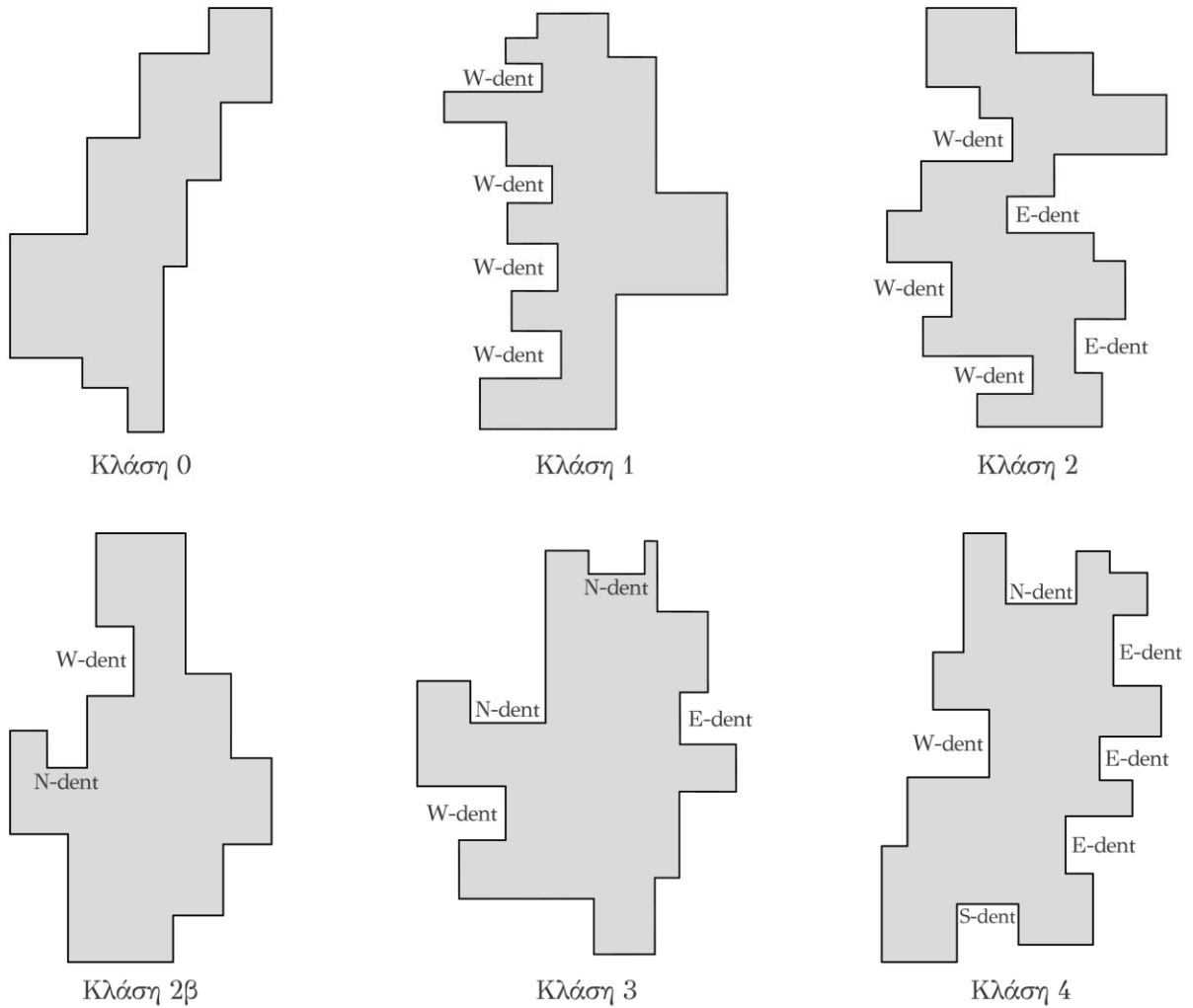
- Κλάση 2α: όπου τα δύο είδη εσοχών είναι «απέναντι», δηλαδή Δυτικές και Ανατολικές, ή Βόρειες και Νότιες
- Κλάση 2β: όπου τα δύο είδη εσοχών είναι κάθετα μεταξύ τους, δηλαδή Δυτικές-Βόρειες, Βόρειες-Ανατολικές, Ανατολικές-Νότιες, Νότιες-Δυτικές
- Κλάση 3: έχει τρία είδη εσοχών
- Κλάση 4: έχει τέσσερα είδη εσοχών

Στο Σχήμα 1.3 που ακολουθεί, παρατηρούμε τις εξής περιπτώσεις: ένα κυρτό ορθογώνιο πολύγωνο κλάσης 0, ένα ορθογώνιο πολύγωνο κλάσης 1 με Δυτικές Εσοχές (W-dents), ένα ορθογώνιο πολύγωνο κλάσης 2α με Δυτικές (W-dents) και Ανατολικές Εσοχές (E-dents), ένα ορθογώνιο πολύγωνο κλάσης 2β με μία Δυτική (W-dent) και μία Βόρεια Εσοχή (N-dent), ένα ορθογώνιο πολύγωνο κλάσης 3 το οποίο δεν έχει Νότιες Εσοχές (S-dents) και, τέλος, ένα ορθογώνιο πολύγωνο κλάσης 4 το οποίο αποτελεί τη γενική περίπτωση ορθογωνίου πολυγώνου.

Μία κλιμακοειδής διαδρομή (*staircase path*) από το x στο y σε ένα πολύγωνο P είναι μία ακολουθία σημείων $(x = x_0, \dots, x_n = y)$ έτσι ώστε να ισχύουν τα εξής:

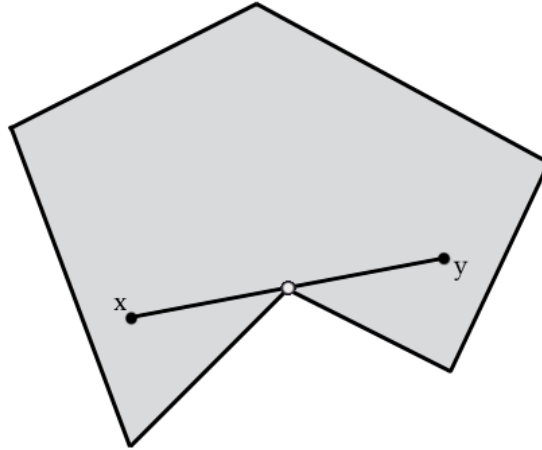
1. Για $1 \leq i \leq n$, το ευθύγραμμο τμήμα μεταξύ των x_{i-1} και x_i είναι είτε οριζόντιο είτε κάθετο και περιέχεται εξ ολοκλήρου στο P .
2. Για $3 \leq i \leq n$ και $d \in \{\text{βορρά, ανατολή, νότο, δύση}\}$, αν το x_{i-2} είναι το d του x_{i-3} τότε το x_i είναι το d του x_{i-1} .

Συνεπώς, μία κλιμακοειδής διαδρομή είναι μία ορθογώνια (x, y) -μονότονη διαδρομή στο P (στο Σχήμα 1.5 τα σημεία p και q συνδέονται με μια κλιμακοειδή διαδρομή).



Σχήμα 1.3: Κλάσεις ορθογωνίων πολυγώνων.

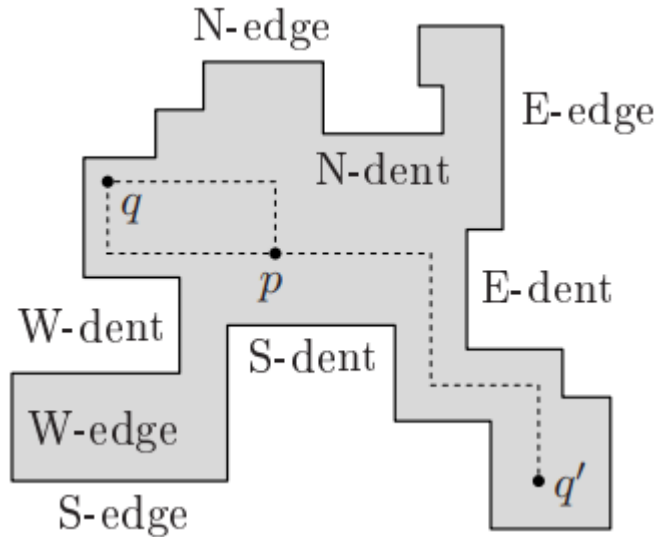
Ορατότητα (visibility) μεταξύ των σημείων x και y (ή το σημείο y είναι ορατό από το x) υπάρχει αν και μόνο αν το κλειστό ευθύγραμμο τμήμα xy δεν τέμνει το εξωτερικό του πολυγώνου $P: xy \subseteq P$. Σημειώνεται ότι ο ορισμός αυτός επιτρέπει την περίπτωση όπου το τμήμα xy έχει κοινά σημεία με το σύνορο του πολυγώνου (Σχήμα 1.4).



Σχήμα 1.4: Το σημείο y είναι ορατό από το σημείο x .

Πιο συγκεκριμένα, δίνονται οι ορισμοί για τις έννοιες: s -ορατότητα, r -ορατότητα, s -αστέρας και r -αστέρας.

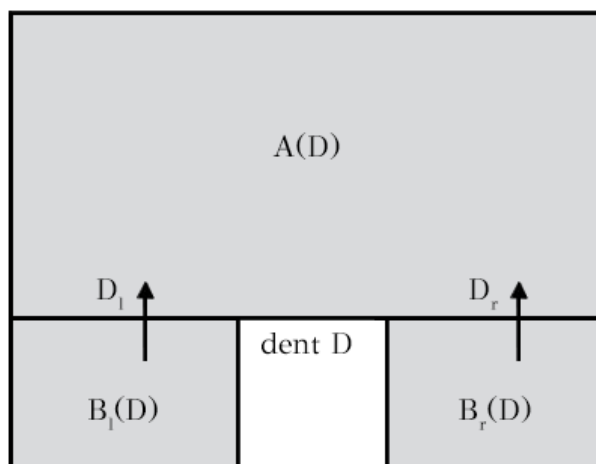
- s -ορατότητα: Έστω δύο σημεία x, y ενός ορθογωνίου πολυγώνου P . Το σημείο x είναι s -ορατό από το σημείο y εάν υπάρχει μια κλιμακοειδής διαδρομή εντός του πολυγώνου P που τα συνδέει. Για παράδειγμα, στο Σχήμα 1.5 το σημείο q είναι s -ορατό από το σημείο p .
- r -ορατότητα: Έστω δύο σημεία x, y ενός ορθογωνίου πολυγώνου P . Το σημείο x είναι r -ορατό από το σημείο y όταν υπάρχει ένα ορθογώνιο παραλληλόγραμμο (με πλευρές παράλληλες στους άξονες του καρτεσιανού συστήματος συντεταγμένων) με τα x, y να βρίσκονται σε δύο απέναντι κορυφές του το οποίο ανήκει στο πολύγωνο P . Για παράδειγμα, στο Σχήμα 1.5 το σημείο q είναι r -ορατό από το σημείο p .
- s -αστέρας: Είναι ένα πολύγωνο P όταν υπάρχει ένα σημείο του από το οποίο όλα τα σημεία του πολυγώνου είναι s -ορατά.
- r -αστέρας: Είναι ένα πολύγωνο P όταν υπάρχει ένα σημείο του από το οποίο όλα τα σημεία του πολυγώνου είναι r -ορατά.



Σχήμα 1.5: Παράδειγμα s-ορατότητας και r-ορατότητας. Το σημείο q' είναι s-ορατό από το σημείο p , ενώ το σημείο q είναι r-ορατό από το σημείο p .

Ένα σύνολο πολυγώνων $S = \{P_1, P_2, \dots, P_k\}$ αποτελεί μία κάλυψη ενός πολυγώνου P εάν η ένωση όλων των πολυγώνων στο S είναι το P . Αν τα κομμάτια αυτού του συνόλου S είναι και ξένα μεταξύ τους (με εξαίρεση τα σύνορά τους) τότε έχουμε μια διαμέριση του πολυγώνου P στο σύνολο S . Στην παρούσα μεταπτυχιακή εργασία θα μελετήσουμε το πρόβλημα της κάλυψης ορθογωνίου πολυγώνου με το ελάχιστο πλήθος s-αστέρων.

Ας θεωρήσουμε μια ακμή εσοχής (dent edge) D , και ας φέρουμε το αντίστοιχο ευθύγραμμο τμήμα εσοχής (dent line) \bar{D} προεκτείνοντας το D και από τα δύο άκρα του τόσο ώστε να περιέχεται στο πολύγωνο P . Αυτός ο ορισμός συνεπάγεται ότι το \bar{D} αποτελεί το μέγιστο τμήμα ευθείας γραμμής που περιέχεται εντελώς στο P και περιέχει την ακμή εσοχής D . Η κατεύθυνση του \bar{D} είναι η ίδια με του D . Το \bar{D} αποτελείται από δύο επιμέρους ευθύγραμμα τμήματα, D_l και D_r , ένα σε κάθε πλευρά του D (Σχήμα 1.6). Το ευθύγραμμο τμήμα εσοχής μιας ακμής εσοχής D χωρίζει το πολύγωνο σε τρία πολυγωνικά τμήματα: τα 2 πολυγωνικά τμήματα $B_l(D)$ και $B_r(D)$ εκατέρωθεν της εσοχής και το $A(D)$ που είναι το υπόλοιπο πολύγωνο (Σχήμα 1.6).



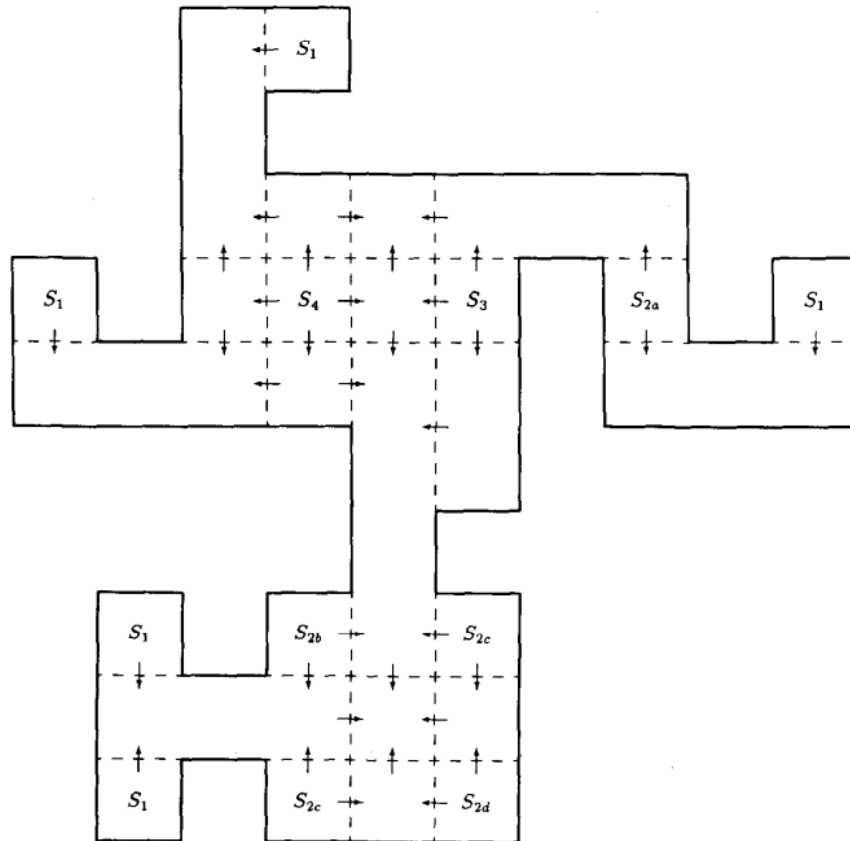
Σχήμα 1.6: Επέκταση της ακμής εσοχής D .

Για κάθε σημείο p και κάθε εσοχή D ενός πολυγώνου P μπορούμε να προσδιορίσουμε μοναδικά αν το p ανήκει στο $B_l(D)$, $B_r(D)$ ή $A(D)$ (Σχήμα 1.6). Μία περιοχή (*region*) u είναι ένα μέγιστο υποσύνολο του P έτσι ώστε για δύο σημεία p και q του u και για κάθε εσοχή D , $p \in Z(D)$ αν και μόνο αν $q \in Z(D)$, όπου το $Z(D)$ είναι ένα από τα $B_l(D)$, $B_r(D)$ ή $A(D)$. Με άλλα λόγια, το σύνολο των σημείων του πολυγώνου τα οποία είναι s -ορατά είναι το ίδιο για όλα τα σημεία κάθε περιοχής.

Για κάθε περιοχή R , ας θεωρήσουμε το σύνολο των σημείων $V(R)$ του πολυγώνου τα οποία είναι s -ορατά από τα σημεία της περιοχής. Λέμε ότι μια περιοχή R κυριαρχείται από μια περιοχή S αν $V(R) \subseteq V(S)$. Αποδεικνύεται ότι για κάθε εσοχή D , τα $B_l(D)$ και $B_r(D)$ κυριαρχούνται από το $A(D)$, ενώ τα $B_l(D)$ και $B_r(D)$ είναι μη συγκρίσιμα (Σχήμα 1.6). Έτσι, από τη διαμέριση ενός ορθογωνίου πολυγώνου σε περιοχές, η οποία προκύπτει από την προσθήκη των ευθυγράμμων τμημάτων εσοχής για κάθε εσοχή, κατασκευάζουμε το διάγραμμα εσοχών (*dent diagram*) [7] προσθέτοντας σε κάθε τμήμα του ευθύγραμμου τμήματος κάθε εσοχής D , ένα βέλος από τα $B_l(D)$ και $B_r(D)$ προς το $A(D)$ (Σχήμα 1.7).

Μια πιο εύχρηστη αναπαράσταση του διαγράμματος εσοχών ενός πολυγώνου είναι μέσω ενός κατευθυνόμενου γραφήματος τοποθετώντας μία κορυφή για κάθε περιοχή και προσθέτοντας μία κατευθυνόμενη ακμή από την κορυφή v στην κορυφή u αν η περιοχή v κυριαρχείται από την περιοχή u . Αποδεικνύεται ότι το προκύπτον κατευθυνόμενο γράφημα είναι μη κυκλικό και για αυτό ονομάζεται κατευθυνόμενο άκυκλο γράφημα περιοχών (*region DAG*). Ως πηγή (*source*) ορίζεται μια περιοχή

που δεν έχει εισερχόμενες ακμές στο γράφημα περιοχών (Σχήμα 1.7). Ομοίως, μία καταβόθρα (*sink*) είναι μια περιοχή που δεν έχει εξερχόμενες ακμές στο γράφημα περιοχών.



Σχήμα 1.7: Διάγραμμα εσοχών (dent diagram) και οι πηγές του, S_i , όπου i το πλήθος των εξερχομένων [1].

Τέλος, ολοκληρώνουμε την ενότητα παραθέτοντας χρήσιμους ορισμούς από την Θεωρία Γραφημάτων. Ένα γράφημα είναι *τριγωνικό* (*chordal*) όταν κάθε κύκλος μήκους τουλάχιστον ίσου με 4, αν υπάρχει, έχει χορδή. Δηλαδή, ένα τριγωνικό γράφημα δεν περιέχει επαγόμενο υπογράφημα ισομορφικό προς άχορδο κύκλο μήκους τουλάχιστον 4. Τα τριγωνικά γραφήματα ανήκουν στην κλάση των τέλειων γραφημάτων. Ένα γράφημα G λέγεται *τέλειο* αν το μέγεθος $\omega(G)$ της μέγιστης κλίμακας του G ισούται με τον χρωματικό αριθμό $\chi(G)$ του G . Σημειώνεται ότι *χρωματικός αριθμός* ενός γραφήματος είναι το ελάχιστο πλήθος χρωμάτων που απαιτούνται

έτσι ώστε κάθε κόμβος να έχει ένα χρώμα και οποιοδήποτε δύο γειτονικοί κόμβοι να έχουν διαφορετικά χρώματα.

Ένα απλό, μη κατευθυνόμενο γράφημα είναι *ασθενώς τριγωνικό* (*weakly triangulated*) εάν ούτε το γράφημα ούτε το συμπλήρωμά του περιέχει επαγόμενο υπογράφημα ισομορφικό προς άχορδο κύκλο μήκους τουλάχιστον 5.

Ένα *κάλυμμα με κλίκες* (clique cover) ενός γραφήματος είναι η διαμέριση του συνόλου κόμβων του γραφήματος σε υποσύνολα κόμβων καθένα από τα οποία επάγει μια κλίκα στο γράφημα. Κάθε γράφημα έχει ένα κάλυμμα με κλίκες, αρκεί να θέσουμε κάθε κόμβο σε ένα ξεχωριστό σύνολο της διαμέρισης. Έτσι, ενδιαφέρον είναι ένα *ελάχιστο κάλυμμα με κλίκες*, δηλαδή ένα κάλυμμα με το ελάχιστο πλήθος υποσυνόλων κόμβων.

1.2 Το Αντικείμενο της Μεταπτυχιακής Εργασίας

Το αντικείμενο της Μεταπτυχιακής Εργασίας είναι η μελέτη του προβλήματος της κάλυψης ορθογωνίου πολυγώνου χωρίς οπές με το ελάχιστο πλήθος s -αστέρων, ειδικότερα η μελέτη και η υλοποίηση του αλγορίθμου των Motwani, Raghunathan και Saran [3] για την επίλυση αυτού του προβλήματος. Ο αλγόριθμος βασίζεται στον υπολογισμό ενός ελάχιστου χρωματισμού σε ασθενώς τριγωνικά γραφήματα με τη βοήθεια του αλγορίθμου των Spinrad και Sritharan [8].

Επιπλέον, παρουσιάζεται ένας αλγόριθμος για την κατασκευή “τυχαίων” ορθογωνίων πολυγώνων εντός ενός ορθογωνίου παραλληλογράμμου. Η κατασκευή αυτή βασίζεται σε μια παράμετρο πυκνότητας $D \in (0,1)$. Ο αλγόριθμος αυτός εκτελείται για περικλείοντα ορθογώνια παραλληλόγραμμα διαφορετικών διαστάσεων και διαφορετικούς παράγοντες πυκνότητας D .

Η υλοποίηση των αλγορίθμων πραγματοποιήθηκε στη γλώσσα προγραμματισμού C++, ενώ η οπτικοποίηση των πολυγώνων έγινε με βιβλιοθήκες της OpenGL.

1.3 Σχετικά Ερευνητικά Αποτελέσματα

Το πρόβλημα της κάλυψης μη ορθογώνιων πολυγώνων από απλούστερες συνιστώσες έχει λάβει ιδιαίτερη προσοχή από πολλούς επιστήμονες. Τα περισσότερα από αυτά τα προβλήματα είναι NP-hard, είτε το πολύγωνο που θα καλυφθεί περιέχει οπές είτε όχι. Έτσι, το ενδιαφέρον στράφηκε στον περιορισμό του προβλήματος με ορθογώνια πολύγωνα. Το 1979 ο Masek [4] ισχυρίστηκε ότι για οποιοδήποτε ορθογώνιο πολύγωνο στο οποίο επιτρέπονται οι οπές, το πρόβλημα της ελάχιστης κάλυψης ορθογωνίων πολυγώνων με ορθογώνια παραλληλόγραμμα είναι NP-complete. Οι Franzblau και Kleitman [5] έδωσαν έναν $O(n^2)$ αλγόριθμο για την κάλυψη ενός κάθετα κυρτού πολυγώνου χωρίς οπές με τον ελάχιστο αριθμό ορθογωνίων παραλληλογράμμων.

Το 1986 ο Keil [6] επινόησε έναν $O(n^2)$ αλγόριθμο για την κάλυψη οριζόντια κυρτών πολυγώνων με τον ελάχιστο αριθμό ορθογώνια κυρτών συνιστωσών και τον ελάχιστο αριθμό ορθογωνίων αστεροειδών συνιστωσών, και συγκεκριμένα r -αστέρων. Οι Culberson και Reckhow [1] απέδειξαν ότι ο αλγόριθμος αυτός είναι ασυμπτωματικά βέλτιστος, δηλαδή για μεγάλες εισόδους εκτελεί στη χειρότερη περίπτωση έναν σταθερό παράγοντα (ανεξάρτητα από το μέγεθος εισόδου) χειρότερο από τον καλύτερο δυνατό αλγόριθμο. Ωστόσο, μπορεί να βελτιωθεί σε $O(n)$ για τον υπολογισμό του αριθμού των πολυγώνων που απαιτούνται για ένα ελάχιστο κάλυμμα. Οι ίδιοι συγγραφείς [7] επινόησαν έναν $O(n^2)$ αλγόριθμο για την κάλυψη ορθογωνίων πολυγώνων κλάσης 2 με τον ελάχιστο αριθμό s -αστέρων.

Έπειτα, ανέπτυξαν [1] έναν αλγόριθμο για την εύρεση ενός ελάχιστου συνόλου ορθογώνια κυρτών πολυγώνων για την κάλυψη ενός ορθογωνίου πολυγώνου χωρίς οπές. Χρησιμοποιώντας dent diagrams, προσδιόρισαν ένα ελάχιστο σύνολο περιοχών έτσι ώστε εάν αυτές οι περιοχές καλύπτονται από οποιοδήποτε σύνολο κυρτών πολυγώνων, τότε το σύνολο μπορεί να επεκταθεί για να καλύψει ολόκληρο το πολύγωνο, χωρίς να αυξήσει τον αριθμό των πολυγώνων στο σύνολο. Πιο συγκεκριμένα, απέδειξαν ότι το πλήθος των ορθογώνια κυρτών πολυγώνων σε ένα ελάχιστο κάλυμμα οποιοδήποτε ορθογωνίου πολυγώνου με εσοχές το πολύ δύο διαφορετικών προσανατολισμών μπορεί να βρεθεί σε γραμμικό χρόνο. Κατασκεύασαν, επίσης, αλγορίθμους τάξης $O(n^2)$ για την ελάχιστη κάλυψη όλων των ορθογωνίων πολυγώνων τέτοιου τύπου.

Επιπροσθέτως, παρουσίασαν μία τεχνική που βασίζεται στην ιδέα της υπογραφής (signature) μερικής κάλυψης, η οποία οδηγεί σε έναν αλγόριθμο πολυωνυμικού χρόνου για την εύρεση του ελάχιστου αριθμού κάλυψης ορθογώνια κυρτών πολυγώνων για ορθογώνια πολύγωνα μεγάλης τάξης με εσοχές και των τεσσάρων προσανατολισμών. Ο αλγόριθμος αυτός είναι αποδοτικός υπό την προϋπόθεση ότι καμία γραμμή εσοχής (dent line) δεν τέμνει κάποια άλλη.

Οι Motwani, Raghunathan και Saran [9] ανέπτυξαν έναν πολυωνυμικό αλγόριθμο για την ελάχιστη κάλυψη πολυγώνων κλάσης 3 με ορθογώνια κυρτά πολύγωνα. Ο αλγόριθμος των Motwani, Raghunathan και Saran [3], στον οποίο βασίζεται η παρούσα εργασία, αφορά την κάλυψη ορθογωνίων πολυγώνων με s -αστέρες (s -stars) για πολύγωνα κλάσης 4. Κύριο εργαλείο της έρευνας αυτής αποτέλεσαν τα γραφήματα ορατότητας (visibility graphs) για περιοχές εντός ενός απλού ορθογωνίου πολυγώνου και απέδειξαν ότι ένα κατάλληλο γράφημα ορατότητας (γράφημα H) είναι ασθενώς τριγωνικό και ότι το πρόβλημα κάλυψης ισοδυναμεί με το πρόβλημα εύρεσης ενός ελάχιστου καλύμματος με κλίκες σε ένα ασθενώς τριγωνικό γράφημα. Χρησιμοποιώντας την ιδιότητα αυτή, επινόησαν πολυωνυμικούς αλγορίθμους για μια κλάση προβλημάτων κάλυψης πολυγώνων που γενικά είναι NP-hard.

Ένα ελάχιστο κάλυμμα με κλίκες για ένα ασθενώς τριγωνικό γράφημα μπορεί να υπολογιστεί σε $O(k^4)$ χρόνο, όπου k το πλήθος των κορυφών του γραφήματος [8]. Από αυτό συνεπάγεται ότι ο αλγόριθμος των Motwani, Raghunathan και Saran [3] έχει πολυπλοκότητα χρόνου $O(n^8)$, όπου n το πλήθος των κορυφών του πολυγώνου. Με βάση τον πιο πρόσφατο αλγόριθμο των Eschen et al. [12] για την κλάση των biclique διαχωρίσιμων γραφημάτων (μιας υπερκλάσης των ασθενώς τριγωνικών γραφημάτων), ένα ελάχιστο κάλυμμα κλικών για ένα ασθενώς τριγωνικό γράφημα με k κόμβους μπορεί να υπολογιστεί σε $O(k^3)$ χρόνο, συνεπώς η πολυπλοκότητα χρόνου του αλγορίθμου των Motwani, Raghunathan και Saran είναι $O(n^6)$.

Τέλος, οι Motwani, Raghunathan και Saran [3] επινόησαν έναν $O(n^3)$ -χρόνου αλγόριθμο για την κάλυψη ορθογωνίων πολυγώνων κλάσης 3 με το ελάχιστο πλήθος s -αστέρων αποδεικνύοντας ότι αυτό το πρόβλημα επιλύεται υπολογίζοντας ένα ελάχιστο κάλυμμα κλικών σε ένα τριγωνικό γράφημα.

1.4 Η Δομής της Μεταπτυχιακής Εργασίας

Η παρούσα Μεταπτυχιακή Εργασία δομείται σε τέσσερα κεφάλαια. Το πρώτο κεφάλαιο αποτελεί την εισαγωγή. Συγκεκριμένα, γίνεται αναφορά σε βασικές έννοιες και ορισμούς σχετικούς με τα πολύγωνα και τις ιδιότητές τους, ενώ στη συνέχεια, παρουσιάζεται το αντικείμενο της Μεταπτυχιακής Εργασίας και σχετικά ερευνητικά αποτελέσματα.

Στο Κεφάλαιο 2 περιγράφουμε αναλυτικά τον αλγόριθμο των Motwani, Raghunathan και Saran [3], τον αλγόριθμο υπολογισμού ελάχιστου χρωματισμού σε ασθενώς τριγωνικά γραφήματα, με χρήση του οποίου υπολογίζεται ένα ελάχιστο κάλυμμα με κλίκες σε ασθενώς τριγωνικά γραφήματα, και η υλοποίησή τους στη γλώσσα προγραμματισμού C++. Ειδικότερα, παρουσιάζουμε το θεωρητικό υπόβαθρο του αλγορίθμου με σχετικά θεωρήματα και λήμματα, την πολυπλοκότητά του και δίνουμε ένα παράδειγμα εφαρμογής του. Για την υλοποίηση, περιγράφουμε τη μορφή της εισόδου και της εξόδου του προγράμματος, αναλύουμε τις δομές δεδομένων και τις βασικές κλάσεις και μεθόδους που χρησιμοποιήσαμε, ενώ επίσης δίνουμε επιλεγμένα παραδείγματα εκτέλεσης του προγράμματος.

Στο Κεφάλαιο 3 παρουσιάζουμε τον αλγόριθμο υπολογισμού “τυχαίων” απλών ορθογωνίων πολυγώνων και περιγράφουμε την υλοποίηση του αλγορίθμου, παραθέτοντας τις βασικότερες κλάσεις του προγράμματος. Ακολουθούν μερικά παραδείγματα εφαρμογής του αλγορίθμου και εκτέλεσης του προγράμματος.

Τέλος, στο Κεφάλαιο 4 συνοψίζουμε τα αποτελέσματα της εργασίας μας και παρουσιάζουμε ενδιαφέρουσες επεκτάσεις.

ΚΕΦΑΛΑΙΟ 2

Ο ΑΛΓΟΡΙΘΜΟΣ ΤΩΝ ΜΟΤWANI, RAGHUNATHAN

ΚΑΙ SARAN

-
- 2.1 Θεωρητικό Υπόβαθρο
 - 2.2 Ο Αλγόριθμος Υπολογισμού Ελάχιστου Χρωματισμού σε Ασθενώς Τριγωνικά Γραφήματα
 - 2.3 Ο Αλγόριθμος των Motwani, Raghunathan και Saran
 - 2.4 Παράδειγμα Εφαρμογής του Αλγορίθμου
 - 2.5 Η Υλοποίηση
 - 2.6 Παραδείγματα Εκτέλεσης του Προγράμματος
-

2.1 Θεωρητικό Υπόβαθρο

Ο αλγόριθμος των Motwani, Raghunathan και Saran [3] βασίζεται στην κατασκευή του διαγράμματος εσοχών (dent diagram) του δοθέντος ορθογωνίου πολυγώνου (Σχήμα 1.7) το οποίο υποδιαιρεί το εσωτερικό του πολυγώνου σε περιοχές. Προκειμένου να επιλύσουν το πρόβλημα της ελάχιστης κάλυψης ορθογωνίου πολυγώνου με s -αστέρες, οι Motwani, Raghunathan και Saran [3] επαναδιατύπωσαν τον ορισμό του αστεροειδούς πολυγώνου (star polygon) ως εξής:

Ορισμός 2.1. Ένα αστεροειδές πολύγωνο (*star polygon*) Q είναι ένα ορθογώνιο πολύγωνο τέτοιο ώστε να υπάρχει ένα σημείο p στο Q με την ιδιότητα το p να βλέπει κάθε σημείο q στο Q .

Για δύο σημεία p, q στο πολύγωνο P , λέμε ότι το p βλέπει ή βλέπει άμεσα το q εάν τα p, q είναι s -ορατά το ένα από το άλλο. Επίσης, λέμε ότι το p βλέπει έμμεσα το q εάν υπάρχει περιοχή $r \in P$, η οποία βλέπει και το p και το q .

Ένα μέγιστο (*maximal*) αστεροειδές πολύγωνο στο P είναι ένα αστεροειδές πολύγωνο που περιέχεται στο P αλλά δεν περιέχεται σε κανένα άλλο αστεροειδές πολύγωνο του P . Κάθε μέγιστος s -αστέρας (*maximal s -star*) είτε περιλαμβάνει κάθε σημείο μιας περιοχής είτε κανένα.

Λήμμα 2.1. Οποιοδήποτε κάλυμμα του P από s -αστέρες μπορεί να γίνει κάλυμμα του P με το ίδιο πλήθος μεγίστων αστεροειδών πολυγώνων.

Ορισμός 2.2. Το πολύγωνο ορατότητας $v(p)$ ενός σημείου $p \in P$ είναι το σύνολο όλων των σημείων του P τα οποία είναι s -ορατά από το p .

Λήμμα 2.2. Το σύνορο μεταξύ $v(p)$ και οποιουδήποτε συνδεδεμένου στοιχείου του $P \setminus v(p)$ είναι ένα ευθύγραμμο τμήμα (οριζόντιο ή κατακόρυφο).

Στον Ορισμό που ακολουθεί, γίνεται συσχέτιση της έννοιας της ορατότητας από διαδρομές κλίμακας με την έννοια των περιοχών.

Ορισμός 2.3. Έστω u, v δύο περιοχές στο πολύγωνο P . Τότε ισχύει ότι $u \cap v$ (η περιοχή u βλέπει την περιοχή v) αν και μόνο αν κάποιο OCP (που περιλαμβάνεται στο P) περιέχει και τις δύο περιοχές, u και v .

Μέσω του ακόλουθου λήμματος, αποδεικνύεται η σχέση ανάμεσα στις κλιμακοειδείς διαδρομές (*staircase paths*) και τα ορθογώνια κυρτά πολύγωνα (OCPs).

Λήμμα 2.3. Για οποιαδήποτε δύο σημεία $u, v \in P$, τα u, v είναι s -ορατά το ένα από το άλλο αν και μόνο αν κάποιο ορθογώνιο κυρτό πολύγωνο (που περιέχεται στο P) περιέχει και τα δύο.

Ένα μέγιστο πολύγωνο κάλυψης (MCP) για ένα απλό ορθογώνιο πολύγωνο P είναι ένα κυρτό ορθογώνιο πολύγωνο που περιέχεται στο P , αλλά δεν περιέχεται σωστά σε κανένα άλλο κυρτό ορθογώνιο πολύγωνο που περιέχεται στο P .

Θεώρημα 2.1. *Δύο σημεία ανήκουν στην ίδια περιοχή αν και μόνο αν ανήκουν στο ίδιο σύνολο μέγιστων πολυγώνων κάλυψης.*

Κάθε μέγιστο πολύγωνο κάλυψης (MCP) αποτελείται από ένα συνδεδεμένο σύνολο περιοχών. Από το Θεώρημα 2.1 αποδεικνύεται ότι η περίμετρος οποιουδήποτε MCP αποτελείται από ακμές του πολυγώνου (τμήματα αυτών) και γραμμές εσοχής (dent lines) έτσι ώστε το εσωτερικό του MCP να είναι πάνω από αυτά. Κατά την υλοποίηση του αλγορίθμου κρίνεται απαραίτητη η κατασκευή του κατευθυνόμενου γραφήματος περιοχών. Οι κόμβοι αυτού του γραφήματος είναι οι περιοχές του πολυγώνου και υπάρχει μια κατευθυνόμενη ακμή από το r_1 στο r_2 εάν μοιράζονται ένα κοινό σύνορο και το r_1 είναι κάτω από το μεταξύ τους dent line.

Λήμμα 2.4. *Το κατευθυνόμενο γράφημα περιοχών είναι άκυκλο.*

Απόδειξη. Δεδομένου ότι κάθε ευθύγραμμο τμήμα εσοχής διασχίζει ολόκληρο το πολύγωνο από ένα σημείο του συνόρου του σε ένα άλλο σημείο του συνόρου του και ότι κάθε κατευθυνόμενη ακμή στο γράφημα περιοχών διασχίζει ένα ευθύγραμμο τμήμα εσοχής, ένας κύκλος στο γράφημα περιοχών πρέπει να περιλαμβάνει δύο ακμές που διασχίζουν το ίδιο ευθύγραμμο τμήμα εσοχής, αλλά σε αντίθετες κατευθύνσεις. Όμως αυτό είναι αδύνατο καθώς όλες οι κατευθυνόμενες ακμές που διασχίζουν ένα δεδομένο ευθύγραμμο τμήμα εσοχής στο γράφημα περιοχών το διασχίζουν την ίδια κατεύθυνση (από κάτω προς τα πάνω). Συνεπώς, το γράφημα περιοχών είναι άκυκλο.

□

Θεώρημα 2.2. *Εάν το C είναι ένα σύνολο μέγιστων πολυγώνων κάλυψης (MCP) που περιλαμβάνει κάθε περιοχή πηγής κάποιου απλού ορθογώνιου πολυγώνου P , τότε το C καλύπτει κάθε περιοχή του P .*

Από το παραπάνω θεώρημα συμπεραίνουμε ότι για την κάλυψη ενός πολυγώνου P με MCPs αρκεί να καλυφθούν οι πηγές του πολυγώνου P .

Από το ακόλουθο Λήμμα καταλήγουμε στο συμπέρασμα ότι προκειμένου να ελέγξουμε εάν δύο περιοχές του P βλέπουν έμμεσα η μία την άλλη, αρκεί να ελέγξουμε αν κάποια καταβόθρα του P βλέπει και τις δύο περιοχές.

Λήμμα 2.5. Έστω M ένα μέγιστο αστεροειδές πολύγωνο στο P . Τότε το M περιέχει μία καταβόθρα του P που βλέπει κάθε περιοχή που καλύπτεται από το M .

Οι Motwani, Raghunathan και Saran [3] όρισαν το γράφημα H ενός απλού ορθογωνίου πολυγώνου κλάσης 4 ως εξής:

Ορισμός 2.4. Το γράφημα H ενός απλού ορθογωνίου πολυγώνου P είναι ένα μη κατευθυνόμενο γράφημα που έχει έναν κόμβο για κάθε περιοχή που ορίζουν στο P τα ευθύγραμμα τμήματα εσοχών ενώ δύο τέτοιοι κόμβοι u, v συνδέονται με ακμή αν υπάρχει περιοχή του P από την οποία είναι s -ορατές οι περιοχές που αντιστοιχούν στους u, v .

Έχοντας ορίσει την έννοια του αστεροειδούς πολυγώνου και του γραφήματος H , παρουσιάζουμε πώς συμβάλλουν στην επίλυση του προβλήματος, μέσω του εξής Θεωρήματος:

Θεώρημα 2.3. Έστω $K = (V(K), E(K))$ μία κλίκα στο γράφημα H . Τότε, οι περιοχές του $V(K)$ μπορούν να καλυφθούν από έναν s -αστέρα στο P . Επομένως, ένα ελάχιστο κάλυμμα με κλίκες του H αντιστοιχεί ακριβώς σε ένα ελάχιστο κάλυμμα του P από αστεροειδή πολύγωνα.

Το θεώρημα προκύπτει από το γεγονός ότι δύο σημεία p, q ενός s -αστέρα βλέπουν έμμεσα το ένα το άλλο. Συνεπώς, κάθε μέγιστος s -αστέρας αντιστοιχεί σε μία κλίκα στο γράφημα H . Από αυτό συνεπάγεται ότι ένα κάλυμμα του P από μέγιστους s -αστέρες αντιστοιχεί σε ένα ελάχιστο κάλυμμα του H με κλίκες.

Το κρίσιμο αποτέλεσμα που οδήγησε στον αλγόριθμο των Motwani, Raghunathan και Saran [3] είναι:

Θεώρημα 2.4. Το γράφημα H ενός απλού ορθογωνίου πολυγώνου P είναι ασθενώς τριγωνικό.

Ο Hayward [10] απέδειξε ότι τα ασθενώς τριγωνικά γραφήματα είναι τέλεια (perfect) από όπου προκύπτει ότι και το γράφημα H είναι τέλειο. Τέλος, οι Hayward, Hoang και Maffray [11] απέδειξαν, επίσης, την ύπαρξη ενός 2-ζεύγους (2-pair) σε κάθε ασθενώς τριγωνικό γράφημα που δεν είναι κλίκα. Παρακάτω διατυπώνεται το Θεώρημα 2-ζευγών (Two-pair theorem).

Θεώρημα 2.5. *Αν το G είναι ένα ασθενώς τριγωνικό γράφημα, τότε κάθε επαγόμενο υπογράφημα του G που δεν είναι κλίκα περιέχει ένα 2-ζεύγος (2-pair).*

2.2 Ο Αλγόριθμος Υπολογισμού Ελάχιστου Χρωματισμού σε Ασθενώς Τριγωνικά Γραφήματα

Για τον υπολογισμό ενός ελάχιστου καλύμματος κλικών ενός ασθενώς τριγωνικού γραφήματος, οι Motwani, Ragunathan και Saran χρησιμοποίησαν τον αντίστοιχο αλγόριθμο των Spinrad και Sritharan [8] για ασθενώς τριγωνικά γραφήματα με βάρη στους κόμβους. Στην εργασία [8], οι Spinrad και Sritharan περιέγραψαν ταχείς αλγορίθμους για την αναγνώριση ασθενώς τριγωνικών γραφημάτων αλλά και για τον υπολογισμό μιας μέγιστης κλικας και ενός ελάχιστου χρωματισμού σε βεβαρημένα ασθενώς τριγωνικά γραφήματα οι οποίοι αν εφαρμοσθούν στο συμπλήρωμα ενός τέτοιου γραφήματος (που είναι επίσης βεβαρημένο ασθενώς τριγωνικό) μας επιτρέπουν να υπολογίσουμε ένα μέγιστο ανεξάρτητο σύνολο και ένα ελάχιστο κάλυμμα κλικών.

Ο αλγόριθμος **W_OPT** των Spinrad και Sritharan [8] με πολυπλοκότητα $O(n^4)$ βασίζεται στον ταχύ υπολογισμό ενός 2-ζεύγους (δύο κόμβοι x, y αποτελούν 2-ζεύγος αν κάθε άχορδο μονοπάτι που τους συνδέει είναι μήκους 2 – συνεπώς οι κόμβοι ενός 2-ζεύγους δεν είναι γειτονικοί) με χρήση του αλγορίθμου **Υπολογισμός_2_ζεύγους** με πολυπλοκότητα $O(n^2)$. Παρακάτω παραθέτουμε αυτούς τους δύο αλγόριθμους.

Αλγόριθμος Υπολογισμός_2_ζεύγους (G)

Είσοδος: Ένα μη κατευθυνόμενο γράφημα G.

Έξοδος: Ένα 2-ζεύγος (x, y), αν υπάρχει.

Για κάθε κόμβο x_i στο $V(G)$ κάνε{

Βήμα 1. /* Υπολογισμός χρήσιμων πληροφοριών σχετικών με τον x_i */

1.1 εκτέλεσε breadth first search από τον κόμβο x_i και για κάθε κόμβο u του G, υπολόγισε την απόσταση $level(u)$ του u από τον x_i ;

1.2 υπολόγισε τις συνεκτικές συνιστώσες H_1, \dots, H_k του γραφήματος $G - (x_i \cup N(x_i))$, δηλ. του γραφήματος που προκύπτει από το G αφού αφαιρέσουμε τον κόμβο x_i , όλους τους γείτονές του και όλες τις προσπίπτουσες ακμές σε αυτούς;

1.3 για κάθε συνεκτική συνιστώσα H_j κάνε

υπολόγισε έναν ελάχιστο διαχωριστή S_j του $V(H_j)$ και του x_i ;

1.4 για κάθε κόμβο y τέτοιον ώστε το $level(y)$ είναι 2 κάνε

Έστω H_k είναι η συνεκτική συνιστώσα στην οποία ανήκει ο y ;
Υπολογίζουμε το πλήθος $deficiency(y)$ των κόμβων του διαχωριστή S_k (της συνιστώσας H_k) στους οποίους ο y δεν είναι γειτονικός;

Βήμα 2. /* αναζήτησε έναν άλλον κόμβο για το 2-ζεύγος με τον x_i */

για κάθε κόμβο y τέτοιον ώστε το $level(y)$ είναι 2 κάνε

{

αν $deficiency(y) = 0$ τότε

{

Τύπωσε ({ x_i, y }); /* ένα 2-ζεύγος */

Σταμάτα.

}

}

}

Αλγόριθμος W-OPT (G)

Είσοδος: Ένα ασθενώς τριγωνικό γράφημα G , με θετικά βάρη w στους κόμβους.

Έξοδος: Μία μέγιστου βάρους κλίκα K_G και ένας ελάχιστου βάρους χρωματισμός f_G .

```
{
  βρες ένα 2-ζεύγος  $\{x, y\}$  στο  $G$ , αν υπάρχει;
  αν το  $G$  δεν έχει 2-ζεύγη τότε /* το  $G$  είναι ένωση πλήρων γραφημάτων */
    υπολόγισε τα  $K_G$  και  $f_G$ ;
  αλλιώς
    {
      Έστω, χωρίς βλάβη της γενικότητας, ότι  $w(x) \leq w(y)$ ;
      αν  $w(x) = w(y)$  τότε
        {
          κατασκεύασε το γράφημα  $J$  ως εξής: /* συγχώνευση  $y$  στον  $x$  */
          πρόσθεσε ακμή που συνδέει τον  $x$  με κάθε κόμβο στο  $N(y)-N(x)$ ;
          διαγραφή  $y$ ;
        }
      αλλιώς /*  $w(x) < w(y)$  */
        {
          κατασκεύασε το γράφημα  $J$  ως εξής: /* συγχώνευση  $y$  στον  $x$  */
          πρόσθεσε ακμή που συνδέει τον  $x$  με κάθε κόμβο στο  $N(y)-N(x)$ ;
          πρόσθεσε μια ακμή μεταξύ των  $x$  και  $y$ ;
           $w(y) \leftarrow w(y) - w(x)$ ;
        }
      }
    }
   $K_J, f_J \leftarrow W-OPT(J)$ ; /* αναδρομικό βήμα */
  αναίρεσε τις πράξεις στο 2-ζεύγος για την κατασκευή των  $K_G, f_G$  από τα  $K_J, f_J$ ;
}
```

Ειδικά, για τον υπολογισμό ενός ελάχιστου χρωματισμού (από τον οποίο υπολογίζουμε ένα ελάχιστο κάλυμμα κλικών στο συμπλήρωμα του γραφήματος) σε ασθενώς τριγωνικά γραφήματα χωρίς βάρη (συνεπώς όλα τα βάρη κόμβων είναι ίσα), ο αλγόριθμος **W-OPT** απλοποιείται στον ακόλουθο αλγόριθμο **W-OPT_unweighted** τον οποίο και χρησιμοποιήσαμε στην υλοποίησή μας.

Αλγόριθμος W-OPT_unweighted (G)

Είσοδος: Ένα ασθενώς τριγωνικό γράφημα G χωρίς βάρη.

Έξοδος: Ένας ελάχιστος βάρους χρωματισμός f_G .

{

Βρες ένα 2-ζεύγος $\{x, y\}$ στο G, αν υπάρχει;

αν το G δεν έχει 2-ζεύγη τότε /* το G είναι ένωση πλήρων γραφημάτων */
 υπολόγισε έναν ελάχιστο χρωματισμό f_G (ο χρωματικός αριθμός είναι ίσος με το πλήθος κόμβων της μέγιστης συνεκτικής συνιστώσας του G. Αντίστοιχα, κάθε κόμβος του G αποτελεί κόμβο-αντιπρόσωπο κάθε κλίκας σε ένα ελάχιστο κάλυμμα κλικών του συμπληρωματικού γραφήματος);

αλλιώς

{

κατασκεύασε το γράφημα J ως εξής: /* συγχώνευση y στον x */

πρόσθεσε ακμή που συνδέει τον x με κάθε κόμβο στο $N(y)-N(x)$;

/ οι κόμβοι x, y ανήκουν στην ίδια χρωματική κλάση ή αντίστοιχα, στην ίδια κλίκα του καλύμματος κλικών του συμπληρωματικού γραφήματος */*

διαγραφή y;

}

$K_J, f_J \leftarrow W-OPT(J)$; /* αναδρομικό βήμα */

}

2.3 Ο Αλγόριθμος των Motwani, Raghunathan και Saran

Με βάση όσα αναφέρθηκαν στην Ενότητα 2.1, ο αλγόριθμος των Motwani, Raghunathan και Saran [3] συνοψίζεται στα εξής:

1. Για κάθε εσοχή, φέρουμε το ευθύγραμμο τμήμα εσοχής, και μετά από ανάθεση των απαιτούμενων φορών μεταξύ περιοχών, κατασκευάζουμε τελικά το διάγραμμα εσοχών (dent diagram).
2. Με βάση το διάγραμμα εσοχών, κατασκευάζουμε γράφημα H_s το οποίο έχει έναν κόμβο για κάθε περιοχή και δύο κόμβοι συνδέονται με ακμή αν οι 2 περιοχές που αντιστοιχούν στους κόμβους είναι s -ορατές η μία από την άλλη.
3. Από το γράφημα H_s κατασκευάζουμε το γράφημα H (Ορισμός 2.4).
4. Εκτελούμε τον αλγόριθμο των Spinrad και Sritharan [12] (βλ. Ενότητα 2.2) για τον υπολογισμό ενός ελάχιστου καλύμματος με κλίκες του γραφήματος H . Το πλήθος κλικών στο ελάχιστο κάλυμμα είναι ίσο με το πλήθος s -αστέρων που απαιτούνται για την κάλυψη του δοθέντος ορθογωνίου πολυγώνου.

Το πλήθος περιοχών στο διάγραμμα εσοχών είναι $O(n^2)$ όπου n το πλήθος κορυφών του δοθέντος ορθογωνίου πολυγώνου P . Συνεπώς, το πλήθος κόμβων τόσο του γραφήματος H_s όσο και του γραφήματος H είναι $O(n^2)$ ενώ το πλήθος ακμών τους είναι $O(n^4)$.

Σύμφωνα με το ακόλουθο λήμμα, προκειμένου να ελεγχθεί αν δύο περιοχές του πολυγώνου P βλέπουν έμμεσα η μία την άλλη, αρκεί να ελεγχθεί αν κάποια καταβόθρα του P βλέπει και τις δύο περιοχές.

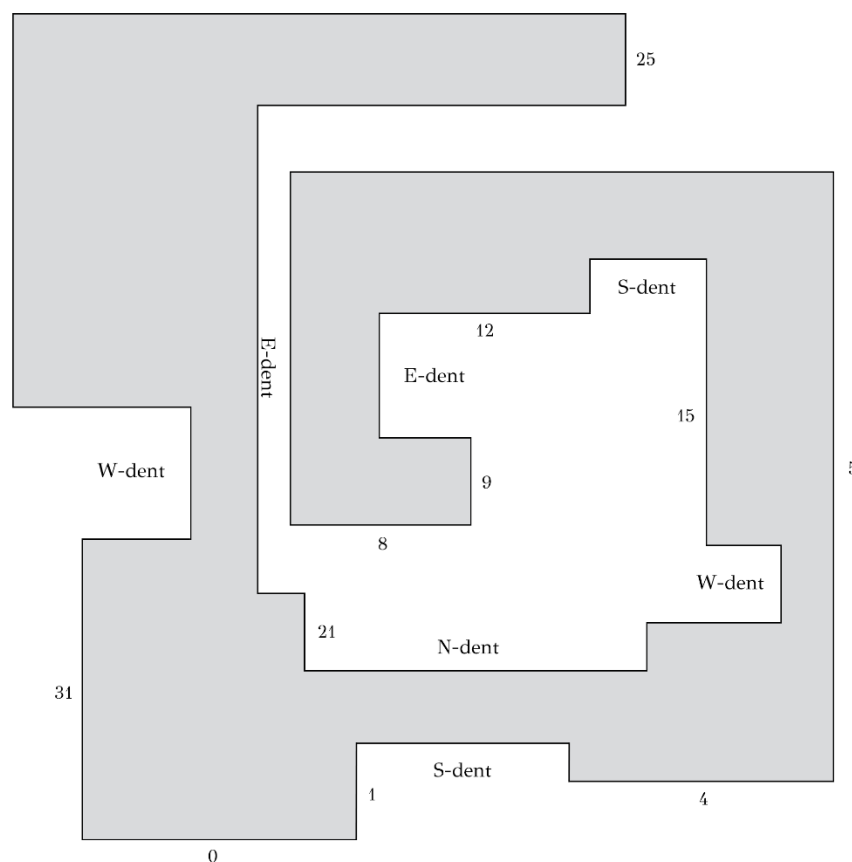
Λήμμα 2.6. *Έστω M ένα μέγιστο αστεροειδές πολύγωνο στο P . Τότε το M περιέχει μια καταβόθρα του P που βλέπει (με s -ορατότητα) κάθε περιοχή που καλύπτεται από το M .*

Για μία περιοχή v και μία καταβόθρα u , μπορούμε να βρούμε σε $O(n)$ χρόνο εάν η περιοχή v είναι s -ορατή από την καταβόθρα u [7]. Επομένως, σε $O(n^5)$ χρόνο, μπορούμε να βρούμε τις περιοχές που βλέπουν οι καταβόθρες του P . Επιπλέον, για κάθε ζεύγος περιοχών, μπορούμε να ελέγξουμε σε γραμμικό χρόνο αν η μία περιοχή βλέπει την άλλη έμμεσα, κατασκευάζοντας το γράφημα H σε $O(n^5)$ χρόνο.

Όπως αναφέραμε στην Ενότητα 2.1, το γράφημα H ενός ορθογωνίου πολυγώνου κλάσης 4 είναι ασθενώς τριγωνικό. Ένα ελάχιστο κάλυμμα με κλίκες (minimum clique cover) για ένα ασθενώς τριγωνικό γράφημα με k κόμβους χρειάζεται $O(k^4)$ χρόνο (Ενότητα 2.2). Συνεπώς, ένα ελάχιστο κάλυμμα με κλίκες για το γράφημα H απαιτεί $O(n^8)$ χρόνο και άρα $O(n^8)$ χρόνο απαιτεί ο υπολογισμός ενός ελαχίστου καλύμματος με s -αστέρες για το δοθέν ορθογώνιο πολύγωνο P .

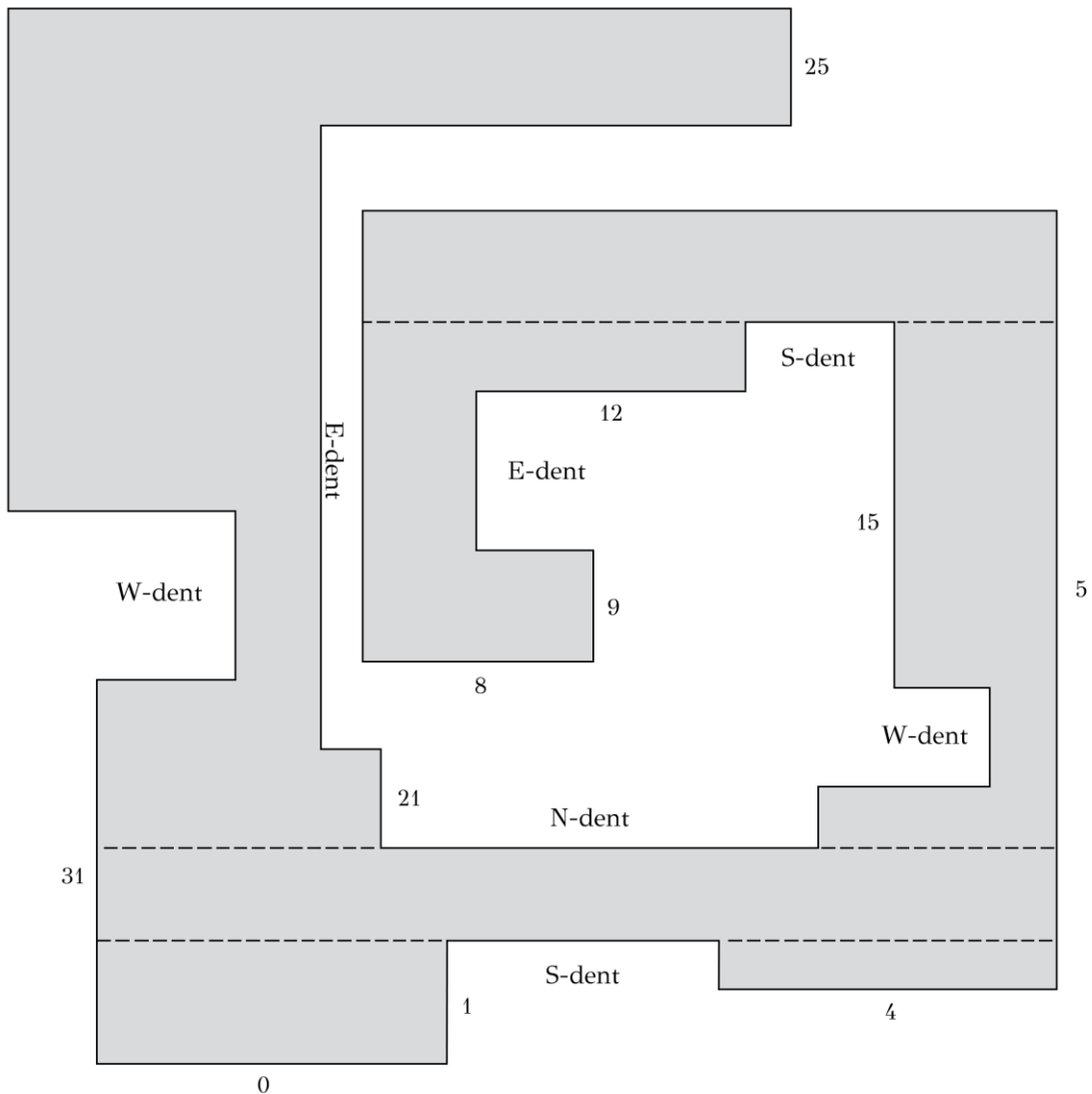
2.4 Παράδειγμα Εφαρμογής του Αλγορίθμου

Ακολουθεί ένα παράδειγμα εφαρμογής του αλγορίθμου με τη βοήθεια σχημάτων. Πιο αναλυτικά, στο Σχήμα 2.1 βλέπουμε ένα ορθογώνιο πολύγωνο κλάσης 4 με μία Βόρεια Εσοχή, δύο Νότιες Εσοχές, δύο Δυτικές Εσοχές και δύο Ανατολικές Εσοχές. Οι αριθμοί που βρίσκονται κάτω από τις ακμές ισοδυναμούν με το αναγνωριστικό (id) κάθε ακμής. Σημειώνεται ότι η προσπέλαση του πολυγώνου ξεκινάει από την ημι-ακμή 0.



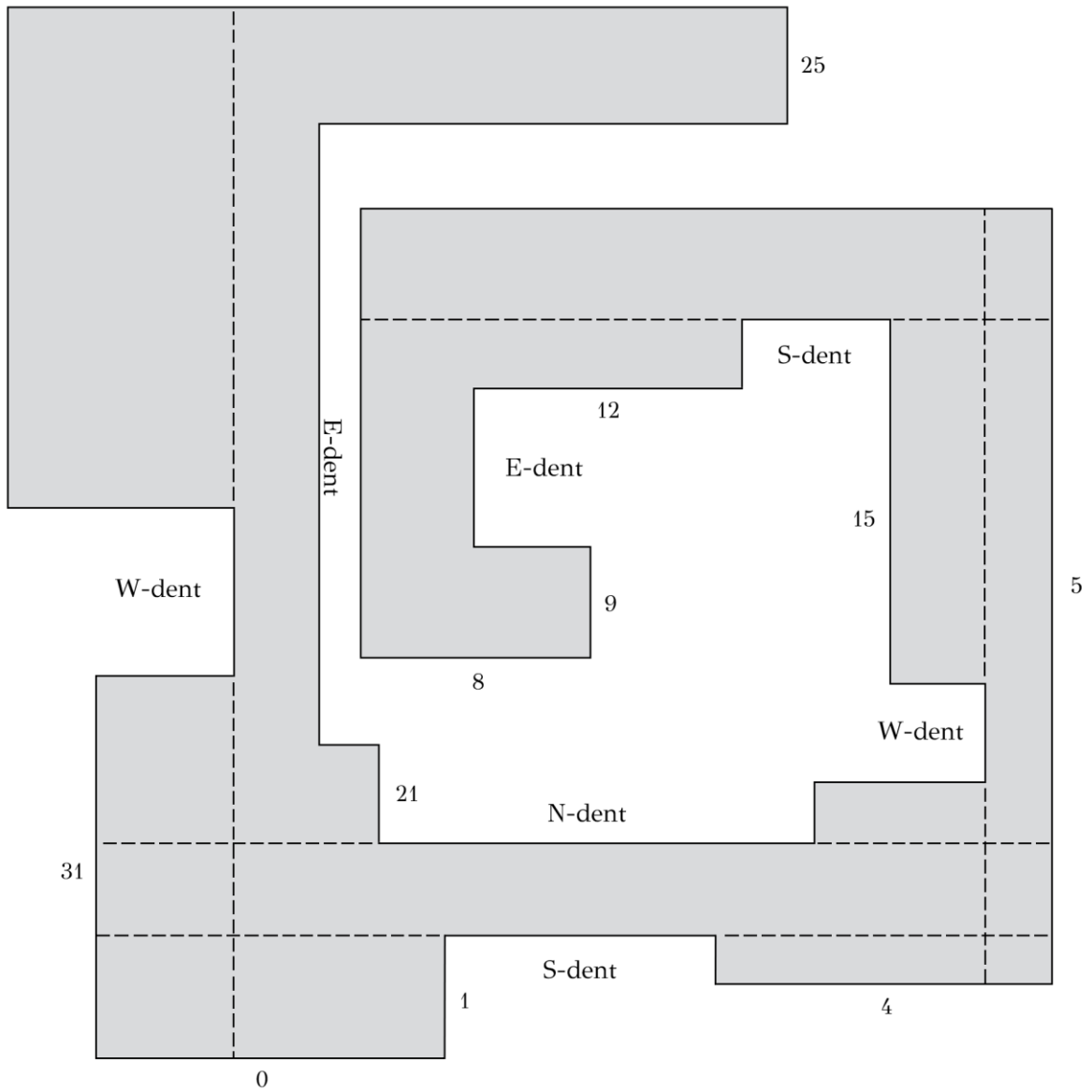
Σχήμα 2.1: Αρχικό πολύγωνο και εσοχές.

Στο Σχήμα 2.2 βρισκόμαστε στο σημείο όπου έχει ολοκληρωθεί η επέκταση της Βόρειας Εσοχής (N-dent) και των δύο Νότιων Εσοχών (S-dents). Κατά την επέκταση των εσοχών δημιουργούνται νέες ακμές οι οποίες συμβολίζονται με διακεκομμένες ακμές. Οι ακμές που τέμνονται από τις νέες ακμές και βρίσκονται στο σύνορο του πολυγώνου χωρίζονται σε δύο επιμέρους ακμές.

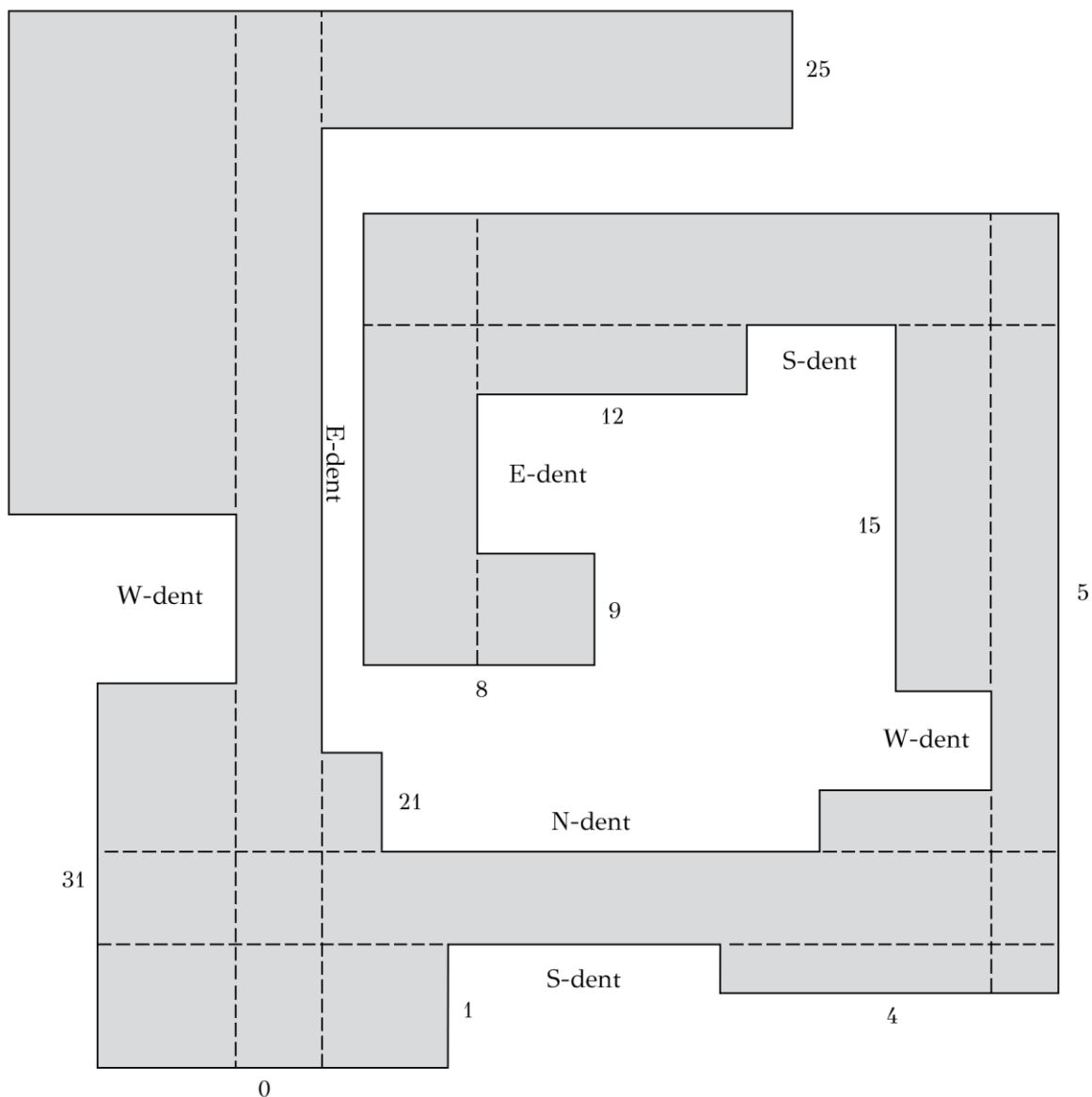


Σχήμα 2.2: Επέκταση Νότιων και Βόρειων Εσοχών.

Στο Σχήμα 2.3 έχουμε ολοκληρώσει την επέκταση και της Δυτικής Εσοχής (W-dent). Η επέκταση όλων των εσοχών του πολυγώνου ολοκληρώνεται στο Σχήμα 2.4 με την επέκταση των δύο Ανατολικών Εσοχών (E-dents).

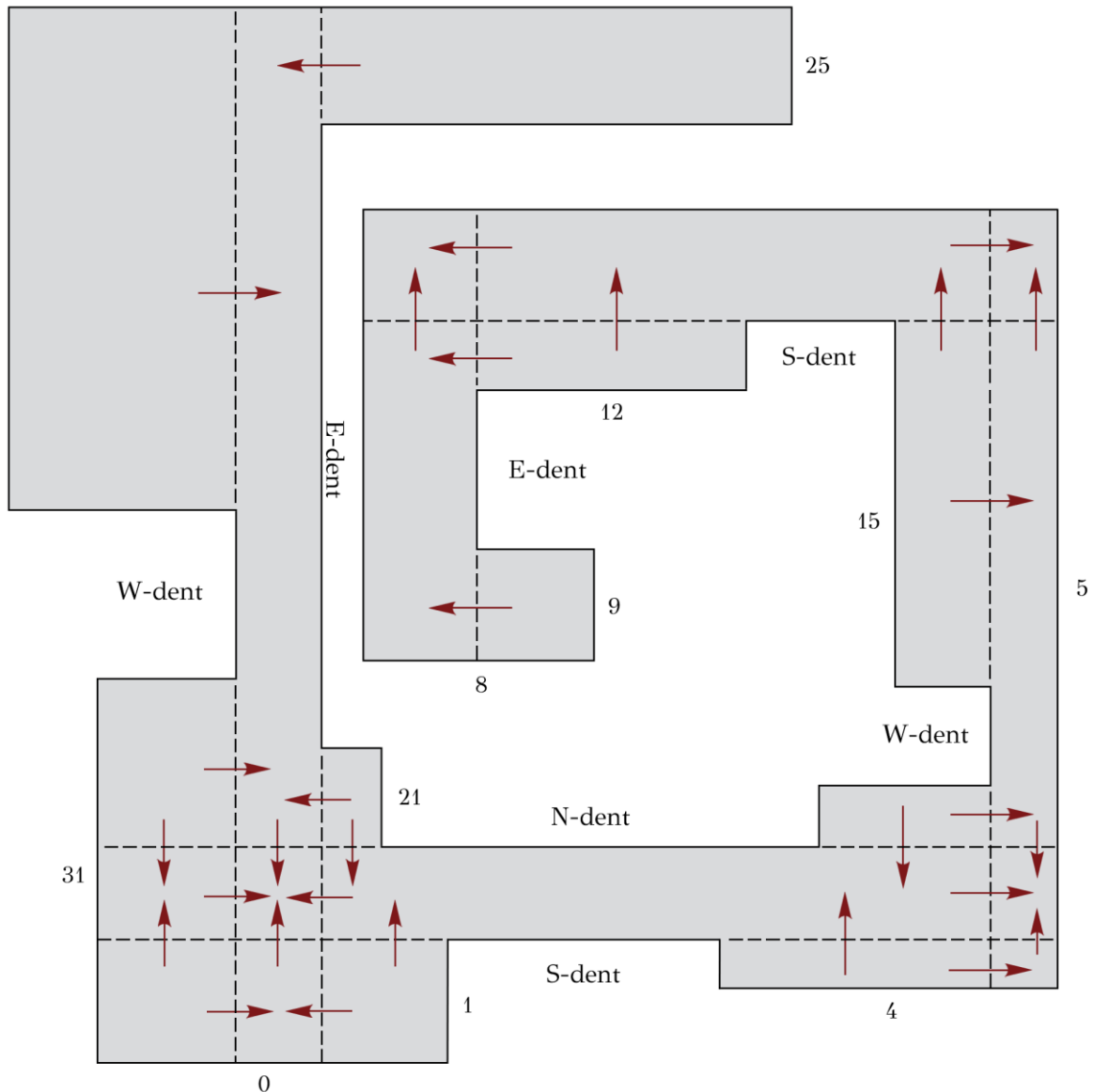


Σχήμα 2.3: Επέκταση Δυτικών Εσοχών.



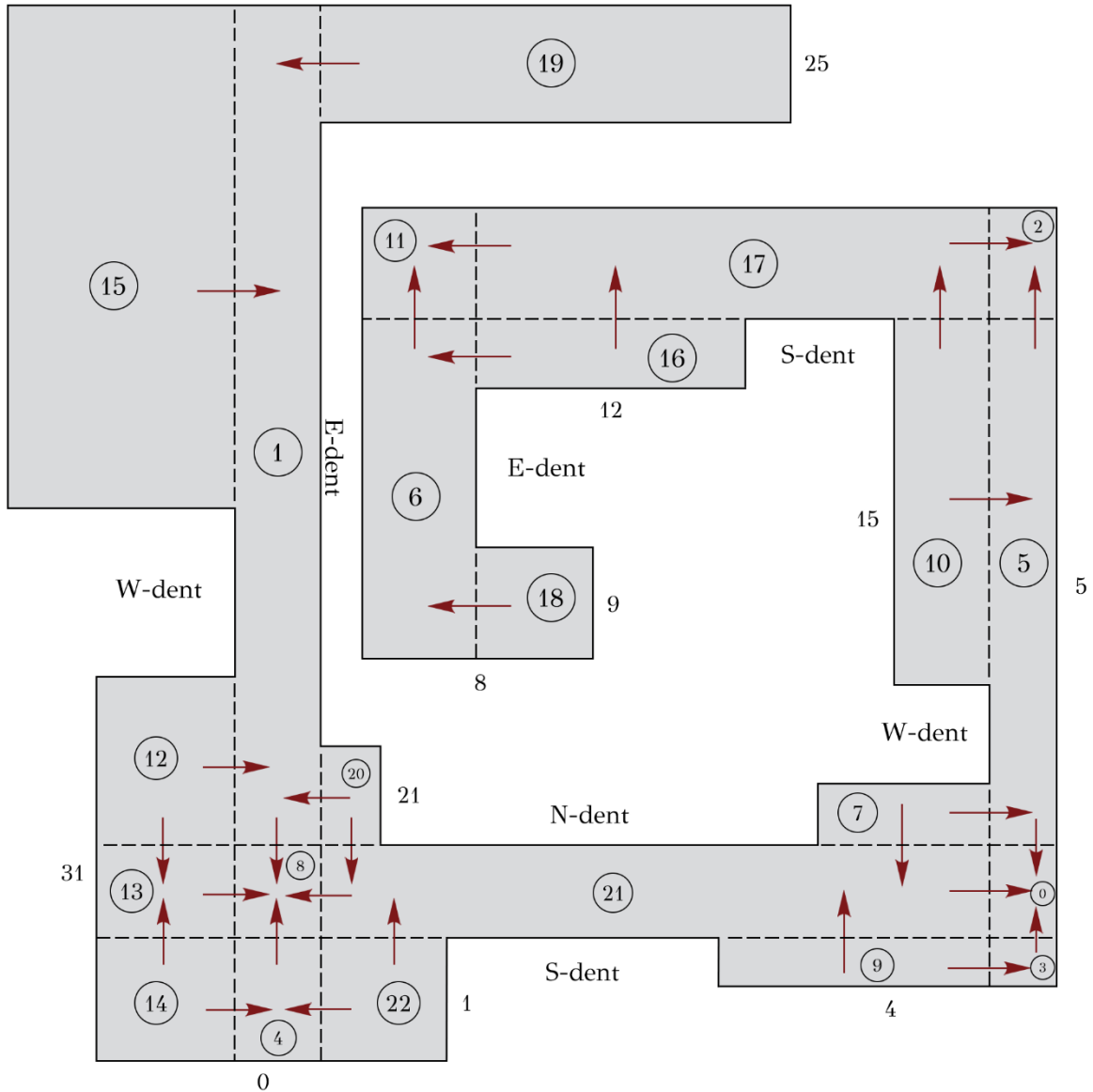
Σχήμα 2.4: Επέκταση Ανατολικών Εσοχών.

Προκειμένου να κατασκευάσουμε το διάγραμμα εσοχών (dent diagram), σχεδιάζουμε μικρά βέλη σε κάθε ακμή που προέρχεται από την επέκταση ενός dent line, από τα αριστερά προς τα δεξιά και από κάτω από την εσοχή προς τα επάνω (Σχήμα 2.5).



Σχήμα 2.5: Κατασκευή dent diagram.

Γνωρίζοντας πλέον όλα τα εισερχόμενα και εξερχόμενα βέλη σε όλες τις περιοχές, μπορούμε να εντοπίσουμε τις πηγές (sources) και τις καταβόθρες (sinks) του πολυγώνου. Πιο συγκεκριμένα, οι περιοχές 7, 9, 10, 12, 14, 15, 16, 18, 19, 20, 22 αποτελούν πηγές του πολυγώνου καθώς όλα τα βέλη είναι εξερχόμενα. Οι καταβόθρες του πολυγώνου είναι οι περιοχές 0, 2, 8 και 11, όπου όλα τα βέλη-κατευθύνσεις των ακμών που απαρτίζουν κάθε περιοχή δείχνουν προς το κέντρο της περιοχής. Στο Σχήμα 2.6 τα αναγνωριστικά (id) των περιοχών που αποτελούν πηγή ή καταβόθρα συμβολίζονται με κύκλο.



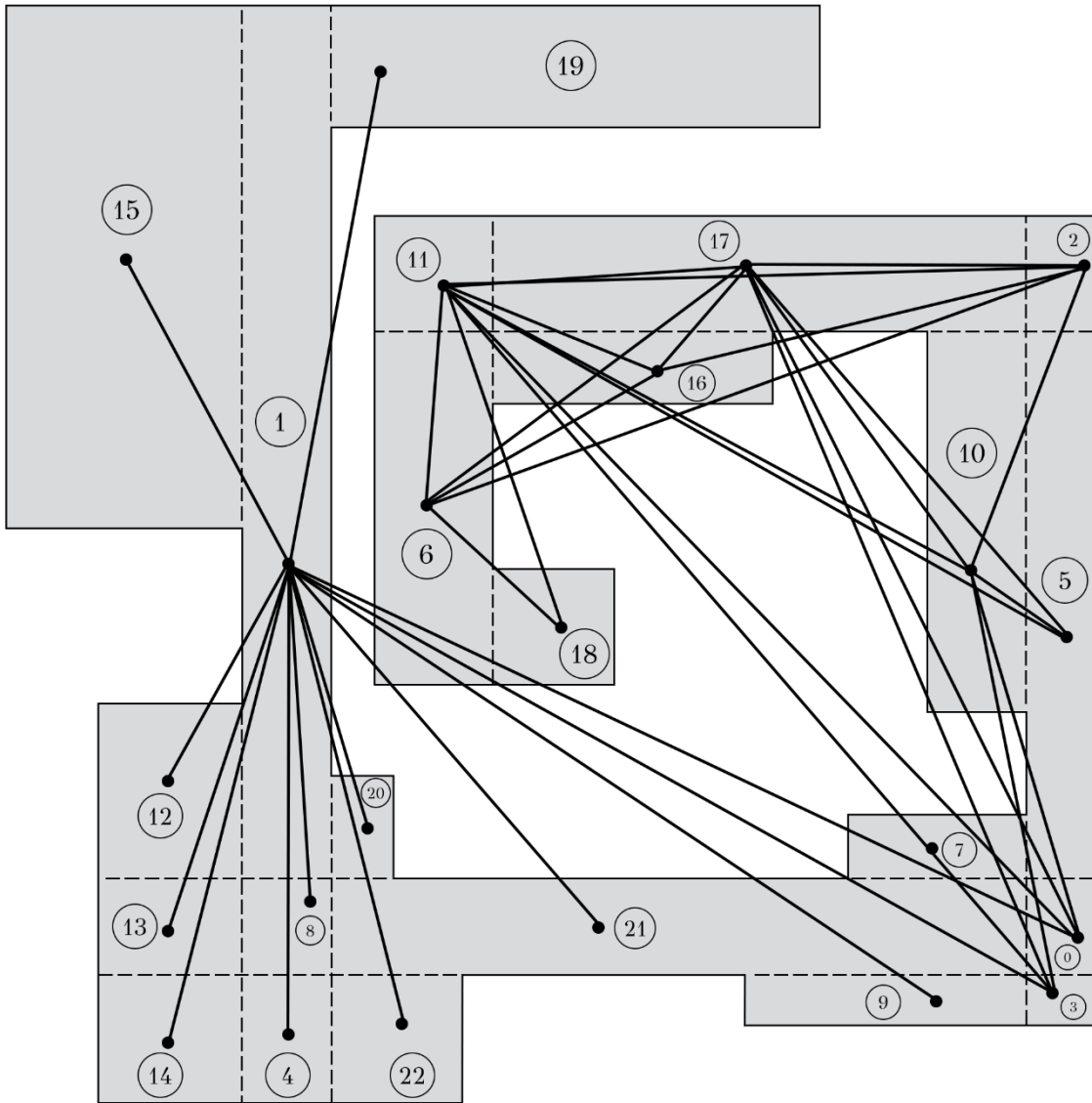
Σχήμα 2.6: Απεικόνιση των περιοχών.

Στη συνέχεια, δημιουργούμε το γράφημα H_s ορίζοντας ως κόμβους του γραφήματος όλες τις περιοχές στις οποίες έχει χωριστεί το πολύγωνο. Δύο τέτοιοι κόμβοι είναι γειτονικοί αν κάθε σημείο της μίας περιοχής s-βλέπει, δηλαδή βλέπει με ένα staircase path, κάθε σημείο της άλλης περιοχής. Για τη δημιουργία του πίνακα γειτνίασης χρησιμοποιούνται τα αποτελέσματα της εκτέλεσης DFS, δηλαδή οι λίστες με τις περιοχές που επισκεπτόμαστε από κάθε περιοχή του πολυγώνου. Ο Πίνακας 2.1 αποτελεί τον πίνακα γειτνίασης του γραφήματος H_s , όπως αυτός προκύπτει από τα ανωτέρω αποτελέσματα της εκτέλεσης DFS. Στο Σχήμα 2.7

απεικονίζονται οι κόμβοι του γραφήματος H_S , ενώ οι κόμβοι 1, 6, 16 και 18 ενώνονται με ακμή με κάθε γειτονικό τους κόμβο. Ομοίως, δημιουργούνται οι ακμές και για τους υπόλοιπους κόμβους του γραφήματος.

Πίνακας 2.1: Πίνακας γειτνίασης για το γράφημα H_S .

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1
1	1	1	0	1	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1
2	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	1	1
3	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	0
4	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1
5	1	0	1	1	1	1	0	1	1	1	1	1	0	1	1	0	0	1	0	0	0	1	1
6	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0
7	1	0	1	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	1
8	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1
9	1	1	1	1	0	1	0	1	1	1	0	0	1	1	0	1	0	0	0	0	1	1	0
10	1	0	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
11	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0
12	1	1	0	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1
13	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1
14	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	0	0	0	1	1	1	1
15	1	1	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1
16	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
17	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0
18	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
19	0	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0
20	1	1	0	1	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1
21	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1
22	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1

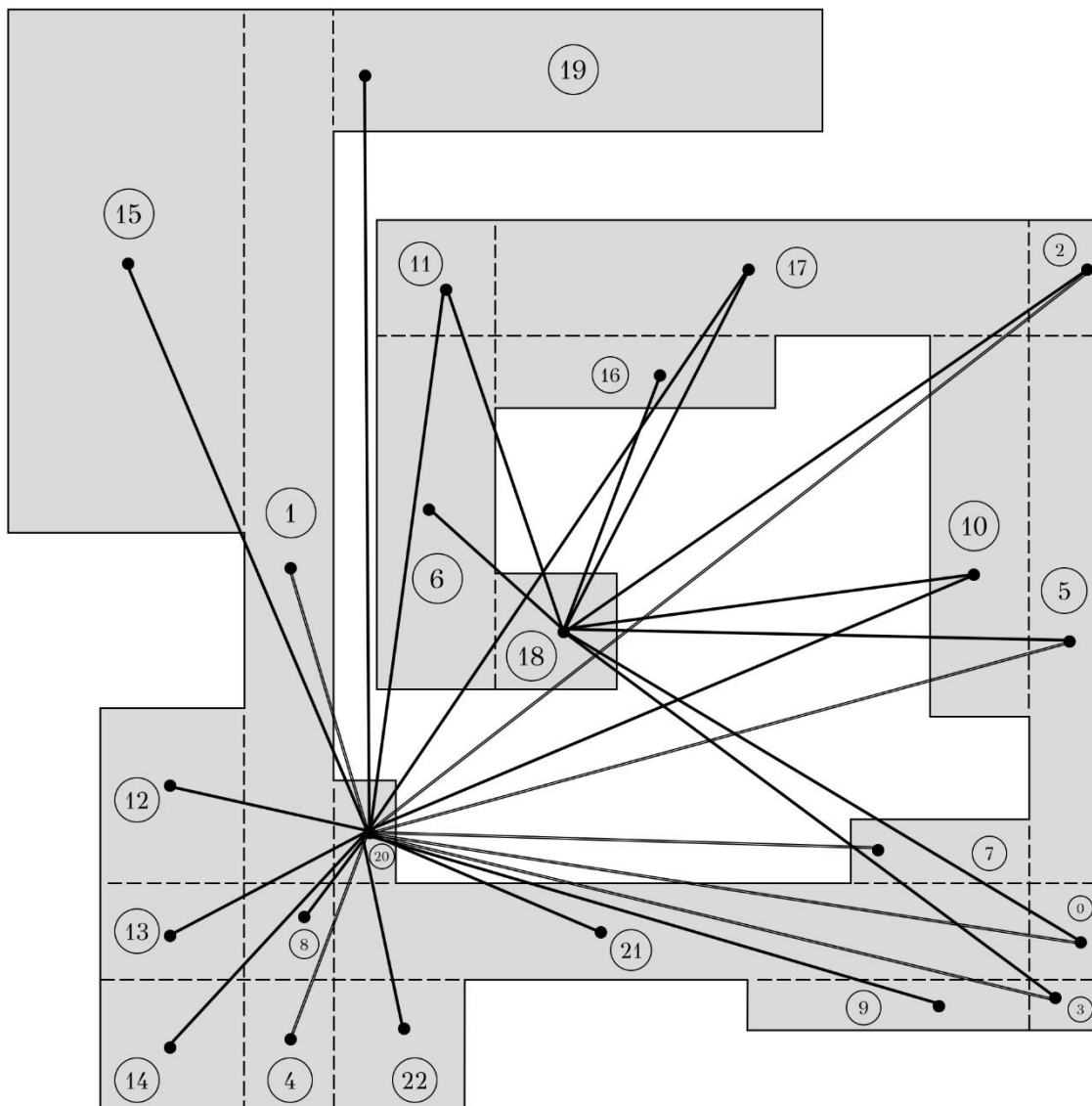


Σχήμα 2.7: Απεικόνιση των κόμβων του γραφήματος H_5 .

Από το γράφημα H_5 κατασκευάζουμε το γράφημα H ορίζοντας ως κόμβους του γραφήματος όλες τις περιοχές του πολυγώνου. Δύο τέτοιοι κόμβοι u, v συνδέονται με ακμή αν υπάρχει κόμβος στο H_5 που είναι γειτονικός, στο γράφημα H_5 , και στον κόμβο u και στον v . Στον Πίνακα 2.2 σημειώνονται με “1” οι γείτονες κάθε κόμβου στο γράφημα H . Στο Σχήμα 2.8 απεικονίζονται οι κόμβοι του γραφήματος H , ενώ ενώνονται με ακμή οι γειτονικοί κόμβοι των κόμβων 18 και 20. Ομοίως, δημιουργούνται οι ακμές και για τους υπόλοιπους κόμβους του γραφήματος.

Πίνακας 2.2: Πίνακας γειτνίασης για το γράφημα H .

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
5	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	0	1	1	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1
7	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
8	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
9	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1
12	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1
15	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1
16	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	0	0	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
18	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0
19	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	0	0	1	1
20	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0



Σχήμα 2.8: Απεικόνιση των κόμβων του γραφήματος H .

Έχοντας κατασκευάσει το γράφημα H , ελέγχουμε για κάθε ζεύγος μη γειτονικών κόμβων εάν είναι 2-ζεύγη (2-pair). Τα 2-ζεύγη για το συγκεκριμένο γράφημα είναι τα εξής:

22 – 21, 22 – 20, 22 – 19, 22 – 15, 22 – 14, 22 – 13, 22 – 12, 22 – 9, 22 – 8, 22 – 7, 22 – 5, 22 – 4, 22 – 3, 22 – 2, 22 – 1, 22 – 0, 18 – 17, 18 – 16, 18 – 11, 18 – 10 και 18 – 6.

Σύμφωνα με τις λίστες γειτνίασης του συμπληρωματικού γραφήματος που ακολουθούν, μπορούμε να αιτιολογήσουμε τη δημιουργία των 2-pair. Για

παράδειγμα, οι κόμβοι 22 και 6 δεν αποτελούν 2-ζεύγος καθώς η διαδρομή που θα ακολουθούσαμε είναι $22 \rightarrow 18 \rightarrow 15 \rightarrow 6$. Συνεπώς, η διαδρομή αυτή έχει απόσταση μεγαλύτερη του 2 και οι κόμβοι 22 και 6 δεν μπορούν να αποτελούν 2-ζεύγος.

0:

1: 6 16 18

2:

3:

4: 18

5:

6: 1 12 15 19 20

7: 18

8: 18

9: 18

10: 19

11: 19

12: 6 16 18

13: 18

14: 18

15: 6 16 18

16: 1 12 15 19 20

17: 19

18: 1 4 7 8 9 12 13 14 15 19 20 21 22

19: 6 10 11 16 17 18

20: 6 16 18

21: 18

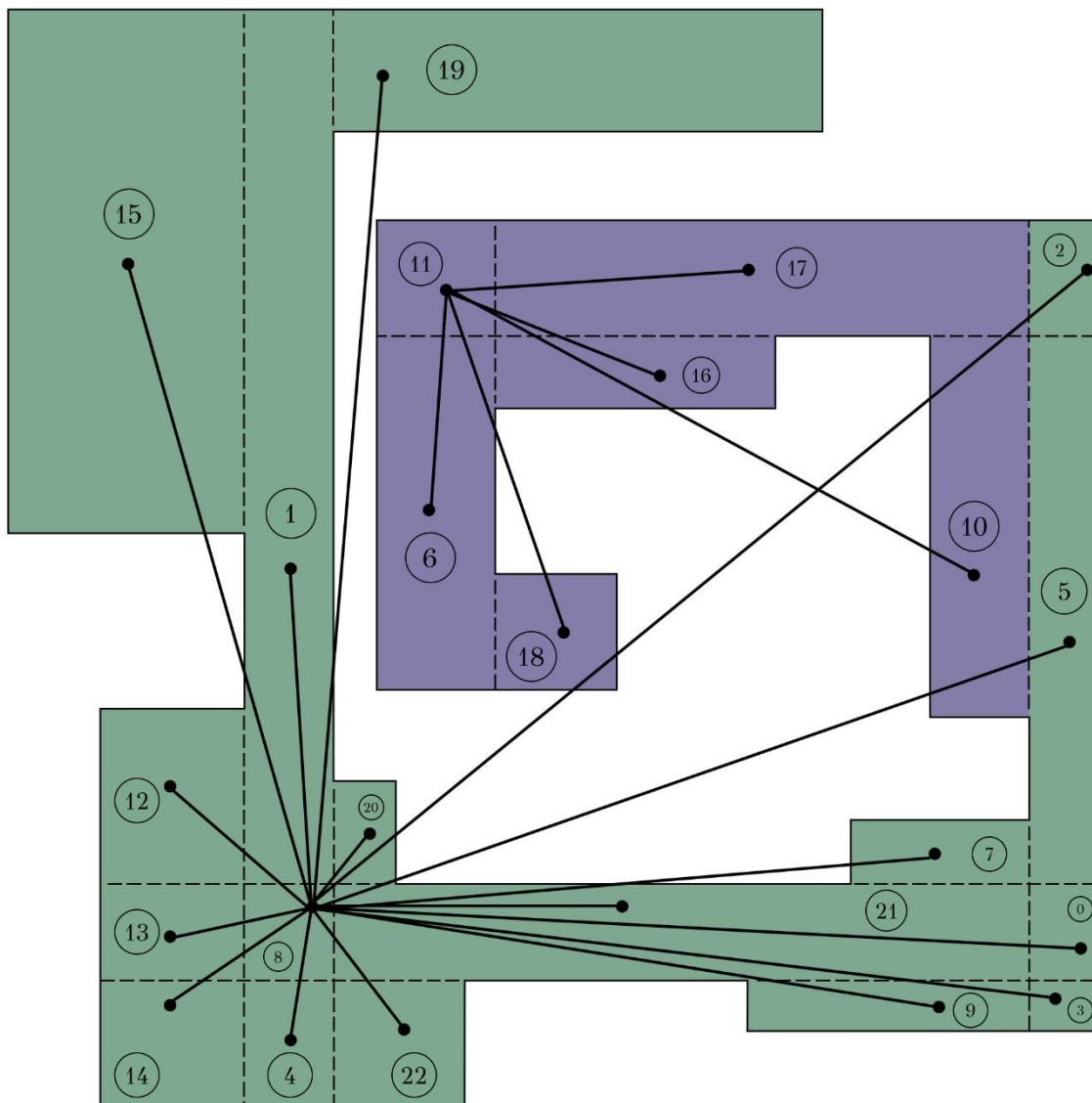
22: 18

Σύμφωνα με τα ανωτέρω αποτελέσματα, καταλήγουμε στις εξής δύο κλίκες:

1) 6 10 11 16 17 18

2) 0 1 2 3 4 5 7 8 9 12 13 14 15 19 20 21 22

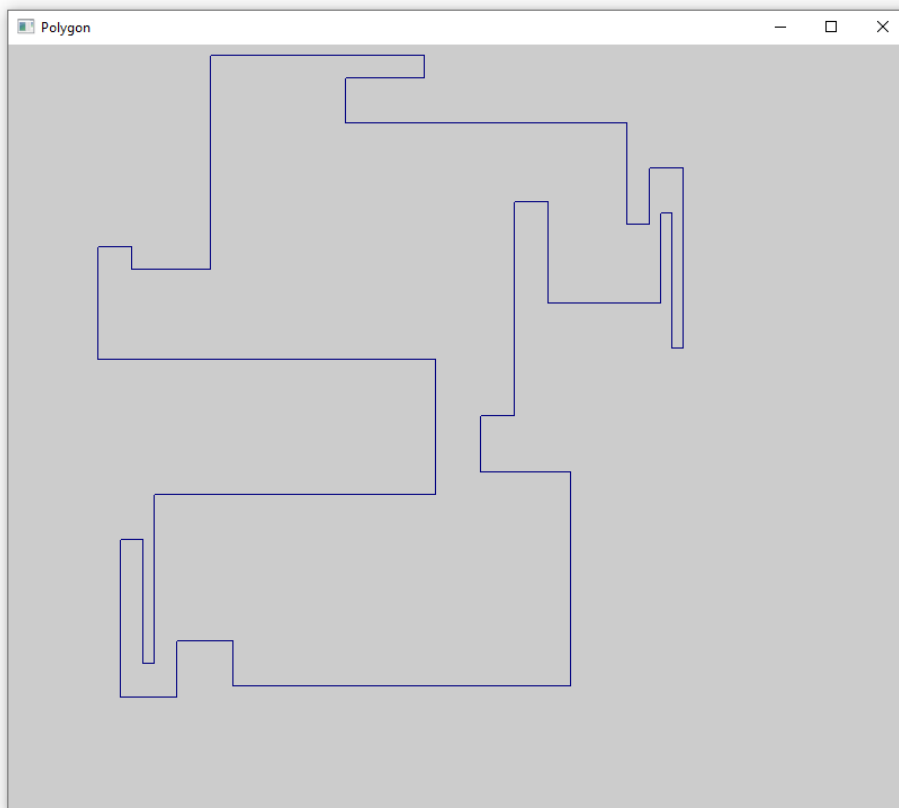
Οι κόμβοι-αντιπρόσωποι κάθε κλίμακας στο κάλυμμα κάλυψης είναι οι καταβόθρες 11 και 8, για την πρώτη και δεύτερη κλίμακα αντιστοίχως. Στο σχήμα που ακολουθεί (Σχήμα 2.9) απεικονίζεται η κλίμακα 6, 10, 11, 16, 17, 18 με μωβ χρώμα και αντιπρόσωπο τον κόμβο 11. Στο ίδιο σχήμα απεικονίζεται, επίσης, η κλίμακα 0, 1, 2, 3, 4, 5, 7, 8, 9, 12, 13, 14, 15, 19, 20, 21, 22 με πράσινο χρώμα και αντιπρόσωπο τον κόμβο 8.



Σχήμα 2.9: Απεικόνιση των κλιμάκων.

2.5 Η Υλοποίηση

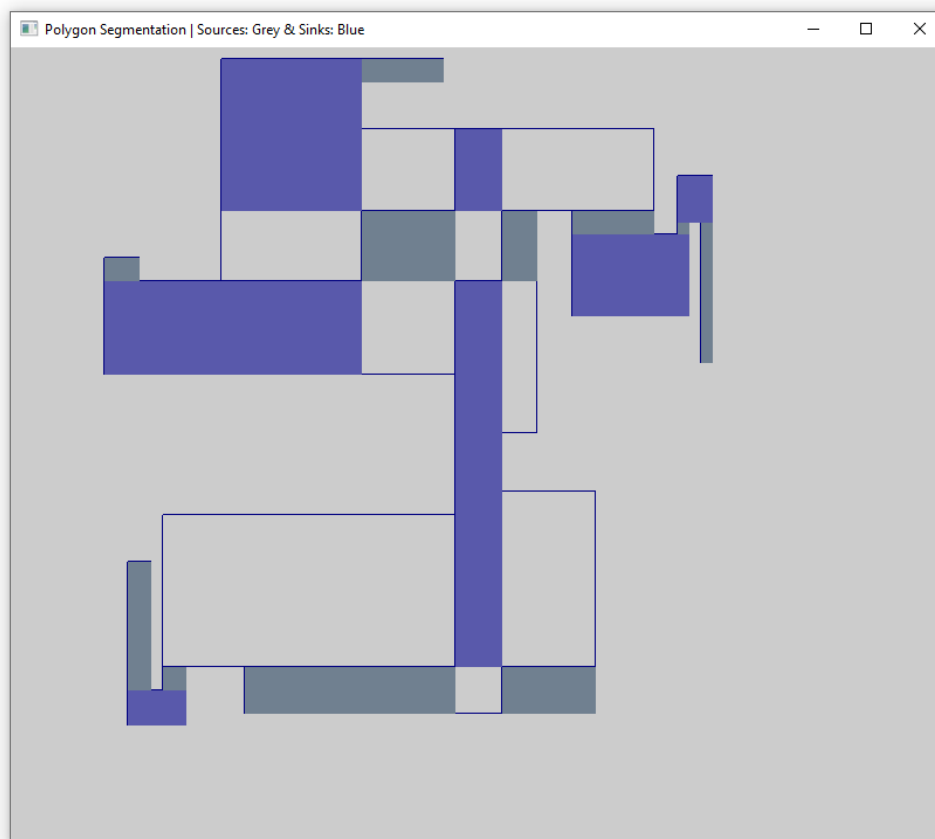
Το πρόγραμμα που υλοποιήσαμε δέχεται ως είσοδο ένα αρχείο που περιέχει, με την ακόλουθη σειρά, το πλήθος των κορυφών και τα ζεύγη των συντεταγμένων όλων των κορυφών ενός πολυγώνου με δεξιόστροφη σειρά εμφάνισής τους κατά μήκος του συνόρου του (έτσι εξασφαλίζουμε και ότι το πολύγωνο θα είναι απλό (δηλαδή χωρίς οπές)). Το ορθογώνιο πολύγωνο που δίνεται ως είσοδος μπορεί να είναι κλάσης 4. Σε περίπτωση ανίχνευσης λαθών εμφανίζονται σχετικά διαγνωστικά μηνύματα. Αυτό μπορεί να συμβεί όταν, για παράδειγμα, δεν μπορεί να ανοίξει ή να διαβαστεί το αρχείο που επιλέξαμε. Πληκτρολογούμε το όνομα του αρχείου και εμφανίζεται το πολύγωνο, πατώντας exit συνεχίζεται η υλοποίηση του προγράμματος (Σχήμα 2.10). Τυπώνονται οι εσοχές που εντοπίζονται στο πολύγωνο καθώς και οι συντεταγμένες κάθε γραμμής εσοχής (dent line). Για κάθε εσοχή τυπώνουμε την ημι-ακμή που συναντάμε και τις νέες κορυφές που προκύπτουν προεκτείνοντας την εσοχή και από τις δύο πλευρές του πολυγώνου.



Σχήμα 2.10: Οπτικοποίηση πολυγώνου εισόδου.

Σημειώνεται ότι κατά την είσοδο ενός αρχείου λαμβάνουμε υπόψιν του κάτωθι περιορισμούς. Οι ημι-ακμές του πολυγώνου θα πρέπει να είναι είτε οριζόντιες είτε κάθετες. Επιπλέον, για απλότητα, απαιτούμε να μην υπάρχει πάνω από μία ακμή στην ίδια οριζόντια ή κατακόρυφη γραμμή. Αλλιώς εμφανίζεται αντίστοιχο μήνυμα στην οθόνη. Οι συντεταγμένες των κορυφών δίνονται με τη φορά του ρολογιού κατά την ωρολογιακή φορά.

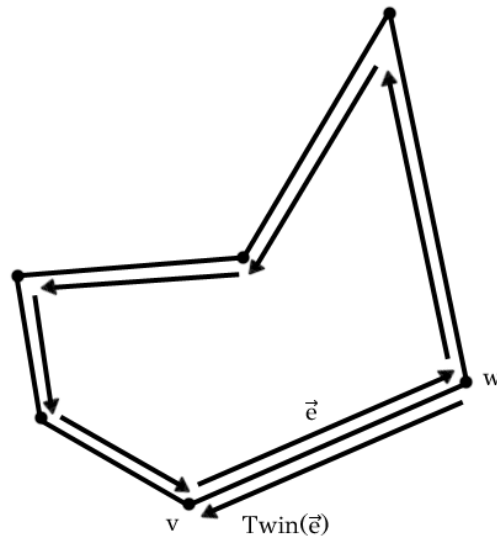
Ακολουθεί η οπτικοποίηση του πολυγώνου μετά τον υπολογισμό των πηγών και των καταβοθρών, των οποίων η απεικόνιση αποτυπώνεται με γκρι και μπλε χρώμα αντιστοίχως (Σχήμα 2.11).



Σχήμα 2.11: Οπτικοποίηση πηγών και καταβοθρών, με γκρι και μπλε χρώμα αντιστοίχως.

Για την αναπαράσταση των περιοχών στο διάγραμμα εσοχών και στο πολύγωνο χρησιμοποιούμε την αναπαράσταση διαμερίσεων του επιπέδου σε περιοχές μέσω της διπλά συνδεδεμένης λίστας ακμών (doubly connected edge list, DCEL).

Σύμφωνα με αυτόν τρόπο αναπαράστασης, κάθε ακμή της διαμέρισης αναπαρίσταται ως ένα ζεύγος δίδυμων ημι-ακμών (half-edge), με αντίθετες κατευθύνσεις (Σχήμα 2.12) και καθεμία τους συσχετίζεται με τη μία από τις δύο περιοχές που είναι γειτονικές στην ακμή. Κάθε ημι-ακμή φέρει δείκτες προς την κορυφή αφετηρίας της, προς την περιοχή f στην οποία ανήκει, προς τη δίδυμή της ημι-ακμή καθώς και προς την προηγούμενη και την επόμενη ημι-ακμή στο σύνορο της περιοχής f .



Σχήμα 2.12: Η ημι-ακμή \vec{e} και η δίδυμή της $\text{Twin}(\vec{e})$.

Από το πρώτο στάδιο της υλοποίησης του αλγορίθμου κρίνεται απαραίτητη η αποθήκευση των κορυφών και των ημι-ακμών που θα επεξεργαστούμε. Λόγω της ανάγκης για συνεχή προσθήκη νέων κορυφών και ημι-ακμών κατά την εκτέλεση του προγράμματος, επιλέχθηκε η αποθήκευσή τους σε vectors από αντικείμενα με την ακόλουθη μορφή.

```
vector<Coordinates> c;
Coordinates::Coordinates(int new_x, int new_y, int new_incident){
    x = new_x;
    y = new_y;
    incidentEdge = new_incident;
}
```

όπου x, y οι συντεταγμένες των κορυφών του πολυγώνου και incidentEdge η ημι-ακμή που έχει ως αφετηρία τη συγκεκριμένη κορυφή.

Για κάθε ημι-ακμή του πολυγώνου αποθηκεύουμε έναν μοναδικό αύξοντα αριθμό που αποτελεί το αναγνωριστικό της (id), την κορυφή αφετηρίας (origin), την περιοχή στην οποία ανήκει (region), την προηγούμενη (previous) και την επόμενη ημι-ακμή (next), τη δίδυμη ημι-ακμή της (twin) και, τέλος, την κατεύθυνση (direction), όπως αυτή προκύπτει από την επέκταση κάθε εσοχής. Αξίζει να σημειωθεί ότι η συγκεκριμένη δομή δεδομένων μας επιτρέπει να τη χειριζόμαστε ως έναν δυναμικό πίνακα, αποθηκεύοντας παράλληλα μεταβλητές διαφορετικών τύπων.

```
vector<HalfEdges> h;  
HalfEdges::HalfEdges(int new_id, int new_origin, int new_region, int  
new_previous, int new_next, int new_twin, string new_direction){  
    id = new_id;  
    origin = new_origin;  
    region = new_region;  
    previous = new_previous;  
    next = new_next;  
    twin = new_twin;  
    direction = new_direction;  
}
```

Κατά την υλοποίηση του προγράμματος χρειάστηκε να αποθηκεύσουμε τις περιοχές του πολυγώνου. Συγκεκριμένα, αποθηκεύουμε το αναγνωριστικό της περιοχής (region_id) και μία ημι-ακμή που πρόσκειται σε αυτήν (component).

```
vector<Regions> r;  
Regions::Regions(int new_region_id, int new_component){  
    region_id = new_region_id;  
    component = new_component;  
}
```

Για την αποθήκευση των εσοχών του πολυγώνου χρησιμοποιήθηκε vector από αντικείμενα (Dents) τα οποία περιέχουν πληροφορίες για τον τύπο της εσοχής (βόρεια, νότια, δυτική, ανατολική) καθώς και για την κορυφή αφετηρίας της εσοχής (dent_origin).

```
vector<Dents> d;
Dents::Dents(int new_dent_type, int new_dent_origin){
    dent_type = new_dent_type;
    dent_origin = new_dent_origin;
}
```

Ομοίως, αποθηκεύσαμε τις πηγές (sources) του πολυγώνου σε vector από αντικείμενα (Sources), αποθηκεύοντας για κάθε πηγή το αναγνωριστικό της περιοχής που αποτελεί πηγή (region) και το πλήθος των εξερχόμενων βελών (outdegree_num). Τα αναγνωριστικά (id) των περιοχών που αποτελούν καταβόθρα για το εξεταζόμενο πολύγωνο, αποθηκεύονται στη λίστα sinks.

```
vector<Sources> sources;
list<int> sinks;
```

Μία από τις βασικές λειτουργίες του προγράμματος είναι η εύρεση των εσοχών του πολυγώνου. Στη συνάρτηση findDents() αποθηκεύουμε στο vector Dents όλες τις εσοχές του πολυγώνου. Πιο συγκεκριμένα, για κάθε νέο dent που βρίσκουμε, αποθηκεύουμε τον τύπο του (1=north, 2=south, 3=west, 4=east) και την κορυφή αφετηρίας του dent line.

```
void findDents(){
    int i2, i3, i4;
    for(int i = 0; i < vertices; i++){
        i2 = i+1;
        i3 = i+2;
        i4 = i+3;
        if(i == vertices-3){
            i4=0;
        }else if(i==vertices-2){
            i3=0;
            i4=1;
        }else if(i==vertices-1){
            i2=0;
            i3=1;
            i4=2;
        }
    }
}
```


Μέσω της συνάρτησης `findSinks()` βρίσκουμε ποιες περιοχές του πολυγώνου αποτελούν και καταβόθρα (sink). Πιο αναλυτικά, για κάθε περιοχή ελέγχουμε την κατεύθυνση όλων των ημι-ακμών που την απαρτίζουν. Οι κατευθύνσεις αυτές έχουν δοθεί κατά την κατασκευή του dent diagram. Σημειώνεται ότι για να αποτελεί μία περιοχή καταβόθρα, θα πρέπει όλα τα βέλη των ημι-ακμών της να δείχνουν προς το εσωτερικό της περιοχής, δηλαδή να είναι περιοχή μηδενικών out-degree.

Η συνάρτηση `return_direction()`, η οποία καλείται στη συνάρτηση `findSinks()`, δέχεται ως είσοδο την τρέχουσα ημι-ακμή και την επόμενη της και επιστρέφει προς τα πού θα κινηθούμε στην επόμενη επανάληψη. Η συγκεκριμένη πληροφορία κρίνεται απαραίτητη καθώς δεν αρκεί να ξέρουμε την κατεύθυνση της ημι-ακμής αλλά και πού βρίσκεται η ημι-ακμή αυτή σε σχέση με την περιοχή. Όταν η κατεύθυνση μίας ημι-ακμής έχει την τιμή `null` σημαίνει ότι βρισκόμαστε σε ημι-ακμή του συνόρου του πολυγώνου. Αντίστοιχα, βρίσκουμε και τις πηγές του πολυγώνου.

```
void findSinks(){
    int halfedge, next_halfedge, component, first_iteration;
    string second_direction;

    for(int i=0; i<=regions_num; i++){
        halfedge = r[i].getComponent();
        next_halfedge = h[halfedge].getNext();
        component = next_halfedge;
        /* add to list sinks the id of the region */
        sinks.push_back(r[i].getRegion());
        first_iteration = 1;
        while((next_halfedge != component) || (first_iteration==1)){
            first_iteration = 0;
            second_direction=return_direction(halfedge,next_halfedge);
            /* If the current half-edge has the direction up or null
             * and the next half-edge goes up or straight.
             */
            if((h[halfedge].getDirection() == "up" ||
h[halfedge].getDirection() == "null") && (second_direction=="up" ||
second_direction=="straight")){
                halfedge = next_halfedge;
```

```

        next_halfedge = h[next_halfedge].getNext();
    /* If the current half-edge has the direction left or null
     * and the next half-edge goes left or straight.
     */
    }else if((h[halfedge].getDirection() == "left" ||
h[halfedge].getDirection() == "null") && (second_direction=="left" ||
second_direction=="straight")){
        halfedge = next_halfedge;
        next_halfedge = h[next_halfedge].getNext();
    /* If the current half-edge has the direction down or null
     * and the next half-edge goes down or straight.
     */
    }else if((h[halfedge].getDirection() == "down" ||
h[halfedge].getDirection() == "null") && (second_direction=="down" ||
second_direction=="straight")){
        halfedge = next_halfedge;
        next_halfedge = h[next_halfedge].getNext();
    /* If the current half-edge has the direction right or
     * null and the next half-edge goes right or straight.
     */
    }else if((h[halfedge].getDirection() == "right" ||
h[halfedge].getDirection() == "null") && (second_direction=="right" ||
second_direction=="straight")){
        halfedge = next_halfedge;
        next_halfedge = h[next_halfedge].getNext();
    }else{
        /* Otherwise, delete the sink from the list
         * and go to the next region.
         */
        sinks.remove(r[i].getRegion());
        num_sinks--;
        break;
    }
    }
}
printSinks(); // Print the final list of sinks
}

```


Μία ακόμη βασική συνάρτηση είναι η DFS(). Συγκεκριμένα, εκτελούμε μονότονο DFS από κάθε περιοχή του πολυγώνου. Για κάθε περιοχή, καλούμε τη συνάρτηση visit() μέσω της οποίας επισκεπτόμαστε, στη συνέχεια, αναδρομικά τις περιοχές προς τις επιτρεπόμενες κατευθύνσεις. Η συνάρτηση visit() δέχεται ως όρισμα το αναγνωριστικό (id) της περιοχής και μία τιμή για κάθε κατεύθυνση που μας επιτρέπει να γνωρίζουμε προς ποιες κατευθύνσεις μπορούμε να κινηθούμε. Για παράδειγμα, εάν έχουμε κινηθεί προς τον Βορρά, τότε δεν μπορούμε να κινηθούμε προς τον Νότο.

```
void visit(int f, int N_flag, int W_flag, int S_flag, int E_flag){
    int halfedge, initial_halfedge, next_halfedge, twin, origin_1, origin_2,
    x1, y1, x2, y2;

    halfedge = r[f].getComponent();
    initial_halfedge = halfedge;
    do{
        if(!(find(visible_regions.begin(), visible_regions.end(), f) !=
        visible_regions.end())){
            visible_regions.push_back(f);           // add it to visible regions
        }

        /* If the half-edge isn't on the border of the polygon */
        if(h[halfedge].getTwin() != -1){
            origin_1 = h[halfedge].getOriginId();
            x1 = c[origin_1].getX();
            y1 = c[origin_1].getY();
            next_halfedge = h[halfedge].getNext();
            origin_2 = h[next_halfedge].getOriginId();
            x2 = c[origin_2].getX();
            y2 = c[origin_2].getY();

            /* If halfedge is north of region */
            if((y1 == y2) && (x1 > x2) && (N_flag == 1)){
                twin = h[halfedge].getTwin();
                visit(h[twin].getRegionId(), 1, W_flag, 0, E_flag);
            }

            if((y1 > y2) && (x1 == x2) && (W_flag == 1)){           // left
                twin = h[halfedge].getTwin();
```

```

        visit(h[twin].getRegionId(), N_flag, 1, S_flag, 0);
    }

    if((y1 == y2) && (x1 < x2) && (S_flag == 1)){ // south
        twin = h[halfedge].getTwin();
        visit(h[twin].getRegionId(), 0, W_flag, 1, E_flag);
    }

    if((y1 < y2) && (x1 == x2) && (E_flag == 1)){ //right
        twin = h[halfedge].getTwin();
        visit(h[twin].getRegionId(), N_flag, 0, S_flag, 1);
    }
}
halfedge = h[halfedge].getNext();
}while(halfedge != initial_halfedge);
}

void DFS(){
    num_visible_regions = 0;

    /* Execute DFS for all regions of the polygon */
    for(int i=0; i<=regions_num; i++){
        visit(i, 1, 1, 1, 1);
        num_visible_regions++;
        cout<<"Regions visited after DFS from region "<<i<<":"<<endl;
        for (int item:visible_regions){
            cout << item << " ";
        }
        cout << endl;
        myVector.push_back(visible_regions);
        visible_regions.clear();
    }
}
}

```

Στη συνέχεια, δημιουργούμε το γράφημα H_s ορίζοντας ως κόμβους του γραφήματος όλες τις περιοχές στις οποίες έχει χωριστεί το πολύγωνο. Δύο τέτοιοι κόμβοι είναι γειτονικοί αν κάθε σημείο της μίας περιοχής s -βλέπει, δηλαδή βλέπει με ένα staircase path, κάθε σημείο της άλλης περιοχής. Για τη δημιουργία του

πίνακα γειτνίασης χρησιμοποιούνται τα αποτελέσματα της εκτέλεσης DFS, δηλαδή τις λίστες με τις περιοχές που επισκεπτόμαστε από κάθε περιοχή.

Η συνάρτηση `min_clique_cover()` βρίσκει ένα ελάχιστο κάλυμμα κλικών για το μη βεβαρημένο γράφημα H , υπολογίζοντας έναν ελάχιστο χρωματισμό στο συμπλήρωμα του γραφήματος. Για τον σκοπό αυτό, καλείται η συνάρτηση `get_two_pair_complement()` η οποία υπολογίζει ένα 2-pair, εάν υπάρχει, στο συμπλήρωμα του υπογραφήματος που παράγεται από τους κόμβους μη μηδενικών τιμών στον πίνακα `in_subgraph[]`. Αξίζει να σημειωθεί ότι σε ένα 2-pair πρέπει να υπάρχει μία διαδρομή μήκους 2 που να συνδέει τους δύο κόμβους του 2-pair.

Μετά τον υπολογισμό του 2-pair, πραγματοποιείται συγχώνευση των δύο κόμβων του 2-pair σε έναν κόμβο. Όσον αφορά το κάλυμμα κλικών, οι κόμβοι που απομένουν όταν δεν υπάρχει 2-pair περιλαμβάνουν έναν κόμβο από κάθε κλίκα. Συνεπώς, το πλήθος τους είναι το μέγεθος του καλύμματος κλικών.

```

/* computes an unweighted min clique cover of the given subgraph */
/* by computing a min coloring in the complement of the subgraph. */
/* At the end, the 1-elements of the array in_subgraph are          */
/* representatives of each of the cliques in the clique cover      */

void min_clique_cover(int in_subgraph[]){
    do{
        if(get_two_pair_complement(in_subgraph) == 0){
            break; /* no two-pair found */
        }

        /* merge y into x */
        in_subgraph[y_item] = 0; /* remove from graph */
        for(int k=v_num-1; k>=0; k--){
            if(in_subgraph[k]==1 /* active vertex and not y */
                && k!=x_item /* not x */
                && adjacency_matrix[y_item][k] == 0){ /* adjacency
in complement */
                /* update neighbors of x */
                adjacency_matrix[x_item][k] = 0;
                /* to preserve symmetry */
                adjacency_matrix[k][x_item] = 0;
            }
        }
    }
}

```

```
    }  
  }while(1);  
  /* clique representatives: all vertices k with in_subgraph[k] = 1 */  
}
```

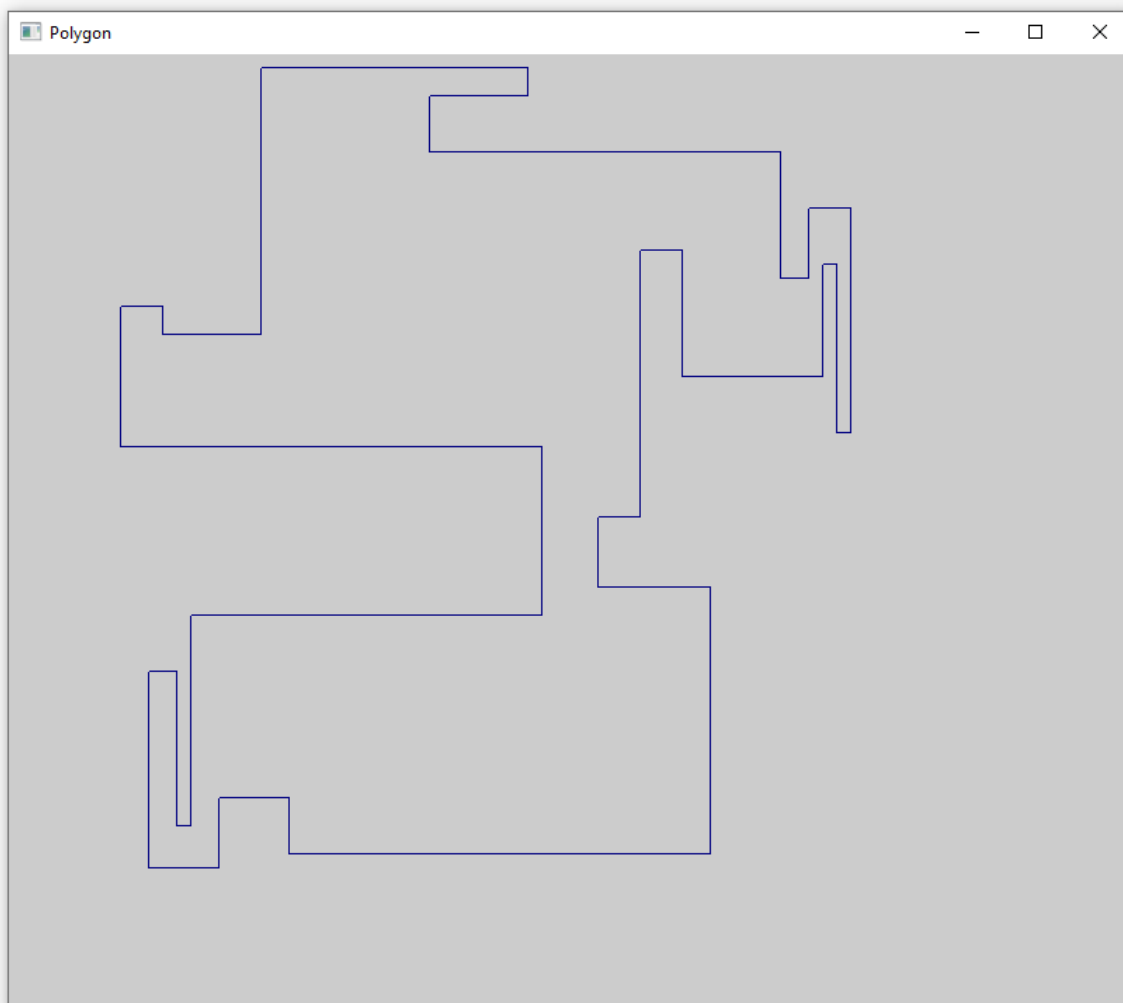
2.6 Παραδείγματα Εκτέλεσης του Προγράμματος

Κατά την εκκίνηση του προγράμματος, εισάγουμε το όνομα του αρχείου εισόδου που περιέχει τις πληροφορίες του πολυγώνου που θα επεξεργαστούμε. Το αρχείο αυτό περιέχει το πλήθος των κορυφών στην πρώτη γραμμή και στη συνέχεια τα ζεύγη των κορυφών του πολυγώνου που ενώνονται με ακμή. Οι συντεταγμένες των κορυφών δίνονται με τη φορά του ρολογιού κατά την ωρολογιακή φορά. Έστω ότι το αρχείο εισόδου είναι το εξής:

```
40  
200 110  
500 110  
500 300  
420 300  
420 350  
450 350  
450 540  
480 540  
480 450  
580 450  
580 530  
590 530  
590 410  
600 410  
600 570  
570 570  
570 520  
550 520
```

550 610
300 610
300 650
370 650
370 670
180 670
180 480
110 480
110 500
80 500
80 400
380 400
380 280
130 280
130 130
120 130
120 240
100 240
100 100
150 100
150 150
200 150

Στο Σχήμα 2.13 απεικονίζεται το πολύγωνο εισόδου, όπως ακριβώς εμφανίζεται κατά την εκτέλεση του προγράμματος.



Σχήμα 2.13: Οπτικοποίηση πολυγώνου εισόδου.

Στη συνέχεια, εκτελούνται οι απαραίτητοι υπολογισμοί για την ανίχνευση εσοχών στο πολύγωνο και τυπώνονται οι εσοχές και τα αντίστοιχα dent line κάθε εσοχής.

E-dent found!

The E-dent consists of:

1: (420,300)

2: (420,350)

S-dent found!

The S-dent consists of:

1: (450,540)

2: (480,540)

S-dent found!

The S-dent consists of:

1: (580,530)

2: (590,530)

N-dent found!

The N-dent consists of:

1: (570,520)

2: (550,520)

E-dent found!

The E-dent consists of:

1: (300,610)

2: (300,650)

N-dent found!

The N-dent consists of:

1: (180,480)

2: (110,480)

W-dent found!

The W-dent consists of:

1: (380,400)

2: (380,280)

N-dent found!

The N-dent consists of:

1: (130,130)

2: (120,130)

S-dent found!

The S-dent consists of:

1: (150,150)

2: (200,150)

Σύμφωνα με τα ανωτέρω αποτελέσματα, στο συγκεκριμένο πολύγωνο υπάρχουν τρεις Βόρειες Εσοχές (N-dents), τρεις Νότιες Εσοχές (S-dents), δύο Ανατολικές Εσοχές (E-dents) και μία Δυτική Εσοχή (W-dent). Γίνεται ταξινόμηση των εσοχών έτσι ώστε να επεξεργαστούν, στη συνέχεια, με προκαθορισμένη σειρά. Ακολουθούν τα αποτελέσματα που τυπώνονται μετά την ταξινόμηση αυτή καθώς και το αναγνωριστικό (id) της ημι-ακμής που αποτελεί dent line για την εσοχή.

Dent types:

1=north 2=south 3=west 4=east

Dent = 1 and its origin is 16

Dent = 1 and its origin is 24

Dent = 1 and its origin is 32

Dent = 2 and its origin is 6

Dent = 2 and its origin is 10

Dent = 2 and its origin is 38

Dent = 3 and its origin is 29

Dent = 4 and its origin is 3

Dent = 4 and its origin is 19

Έπειτα, επεξεργαζόμαστε κάθε εσοχή σύμφωνα με την ανωτέρω προκαθορισμένη σειρά και επεκτείνουμε κάθε γραμμή εσοχής (dent line) προς τις δύο κατευθύνσεις. Αναλυτικότερα, επεκτείνουμε πρώτα τις Βόρειες Εσοχές (N-dents) προς τα αριστερά και έπειτα προς τα δεξιά έως ότου συναντήσουμε το σύνορο του πολυγώνου. Για παράδειγμα, το dent line της πρώτης Βόρειας Εσοχής είναι η ημι-ακμή ((570, 520), (550, 520)). Επεκτείνοντας την ημι-ακμή αυτή προς τα αριστερά συναντάμε την ημι-ακμή ((480, 540), (480, 450)) του πολυγώνου. Επομένως, στο σημείο τομής δημιουργείται μία νέα κορυφή με συντεταγμένες (480, 520). Η νέα

ημι-ακμή που δημιουργείται, με κορυφή αφετηρίας το ένα άκρο του dent line και τελικό άκρο το νέο σημείο, είναι η ((550, 520), (480, 520)). Ομοίως, επεκτείνοντας το dent line προς τα δεξιά, συναντάμε την ημι-ακμή ((580, 530), (580, 450)). Στο σημείο που τέμνει το σύνορο του πολυγώνου, δημιουργείται η κορυφή (580, 520). Η νέα ημι-ακμή που δημιουργείται είναι η ((580, 520), (570, 520)). Σημειώνεται ότι η δίδυμη (twin) ημι-ακμή της είναι η ((570, 520), (580, 520)). Ομοίως, επεκτείνεται κάθε dent line των Νότιων Εσοχών (S-dents).

Dent No1

N-dent:

$$(x1, y1) = (570, 520)$$

$$(x2, y2) = (550, 520)$$

Edge found on the left: (480,540) - (480,450)

The new pair of coordinates on the left is: (480,520)

The new extended half-edge is: (550,520) - (480,520)

Edge found on the right: (580,530) - (580,450)

The new pair of coordinates on the right is: (580,520)

The new extended half-edge is: (580,520) - (570,520)

Dent No2

N-dent:

$$(x1, y1) = (180, 480)$$

$$(x2, y2) = (110, 480)$$

Edge found on the left: (80,500) - (80,400)

The new pair of coordinates on the left is: (80,480)

The new extended half-edge is: (110,480) - (80,480)

Edge found on the right: (450,350) - (450,540)

The new pair of coordinates on the right is: (450,480)

The new extended half-edge is: (450,480) - (180,480)

Dent No3

N-dent:

$(x1, y1) = (130, 130)$

$(x2, y2) = (120, 130)$

Edge found on the left: $(100,240) - (100,100)$

The new pair of coordinates on the left is: $(100,130)$

The new extended half-edge is: $(120,130) - (100,130)$

Edge found on the right: $(150,150) - (150,100)$

The new pair of coordinates on the right is: $(150,130)$

The new extended half-edge is: $(150,130) - (130,130)$

Dent No4

S-dent:

$(x1, y1) = (450, 540)$

$(x2, y2) = (480, 540)$

Edge found on the right: $(550,520) - (550,610)$

The new pair of coordinates on the right is: $(550,540)$

The new extended half-edge is: $(480,540) - (550,540)$

Edge found on the left: $(180,670) - (180,480)$

The new pair of coordinates on the left is: $(180,540)$

The new extended half-edge is: $(180,540) - (450,540)$

Dent No5

S-dent:

$(x1, y1) = (580, 530)$

$(x2, y2) = (590, 530)$

Edge found on the right: $(600,410) - (600,570)$

The new pair of coordinates on the right is: $(600,530)$

The new extended half-edge is: $(590,530) - (600,530)$

Edge found on the left: $(570,570) - (570,520)$

The new pair of coordinates on the left is: $(570,530)$

The new extended half-edge is: $(570,530) - (580,530)$

Dent No6

S-dent:

$$(x1, y1) = (150, 150)$$

$$(x2, y2) = (200, 150)$$

Edge found on the right: (500,110) - (500,300)

The new pair of coordinates on the right is: (500,150)

The new extended half-edge is: (200,150) - (500,150)

Edge found on the left: (130,280) - (130,130)

The new pair of coordinates on the left is: (130,150)

The new extended half-edge is: (130,150) - (150,150)

Προκειμένου να επεκτείνουμε τη Δυτική Εσοχή (W-dent) προς τα Νότια, βρίσκουμε το πλήθος των ενδιάμεσων ημι-ακμών που συναντάμε μέχρι να φτάσουμε στο σύνορο του πολυγώνου. Στον αριθμό αυτόν δεν περιλαμβάνεται το σύνορο του πολυγώνου. Στο συγκεκριμένο παράδειγμα, η ημι-ακμή αυτή είναι μία, η ((200, 150), (500, 150)), και προέρχεται από την επέκταση του S-dent ((150, 150), (200, 150)). Η νέα κορυφή που δημιουργείται είναι η (380, 150) και χωρίζει την ημι-ακμή που βρήκαμε σε δύο επιμέρους ημι-ακμές, καθεμία από αυτές έχει και μία δίδυμη (twin) ημι-ακμή με αντίθετη φορά. Η διαδικασία αυτή επαναλαμβάνεται για το σύνορο του πολυγώνου και, στη συνέχεια, επεκτείνουμε τη Δυτική Εσοχή προς τον Βορρά.

Ομοίως, επεκτείνουμε και τις Ανατολικές Εσοχές (E-dents) του πολυγώνου προς τις δύο κατευθύνσεις. Μία βασική διαφορά κατά τη δημιουργία και ενημέρωση των ημι-ακμών είναι ότι η δίδυμη ημι-ακμή μιας ημι-ακμής που βρίσκεται στο σύνορο του πολυγώνου έχει τιμή null. Αντιθέτως, όταν τέμνεται μία ενδιάμεση ημι-ακμή καθώς επεκτείνουμε το dent line, δημιουργούμε νέα εγγραφή για τη δίδυμη ημι-ακμή της και ενημερώνουμε καταλλήλως της ημι-ακμές που επηρεάζονται.

Dent No7

W-dent:

$$(x1, y1) = (380, 400)$$

$$(x2, y2) = (380, 280)$$

The number of horizontal half-edges we meet (except the border) by extending
on the down side is: 1

Half-edge found on the right: (200,150) - (500,150)

The new pair of coordinates is: (380,150)

Half-edge found on the right: (200,110) - (500,110)

The new pair of coordinates is: (380,110)

The number of horizontal half-edges we meet (except the border) by extending
on the upper side is: 2

Half-edge found on the left: (450,480) - (180,480)

The new pair of coordinates is: (380,480)

Half-edge found on the left: (450,540) - (180,540)

The new pair of coordinates is: (380,540)

Half-edge found on the left: (550,610) - (300,610)

The new pair of coordinates is: (380,610)

Dent No8

E-dent:

(x1, y1) = (420, 300)

(x2, y2) = (420, 350)

Half-edge found on the right: (450,480) - (380,480)

The new pair of coordinates is: (420,480)

Half-edge found on the right: (450,540) - (380,540)

The new pair of coordinates is: (420,540)

Half-edge found on the right: (550,610) - (380,610)

The new pair of coordinates is: (420,610)

Half-edge found on the left: (380,150) - (500,150)

The new pair of coordinates is: (420,150)

Half-edge found on the left: (380,110) - (500,110)

The new pair of coordinates is: (420,110)

Dent No9

E-dent:

(x1, y1) = (300, 610)

(x2, y2) = (300, 650)

Half-edge found on the right: (370,670) - (180,670)

The new pair of coordinates is: (300,670)

Half-edge found on the left: (180,540) - (380,540)

The new pair of coordinates is: (300,540)

Half-edge found on the left: (180,480) - (380,480)

The new pair of coordinates is: (300,480)

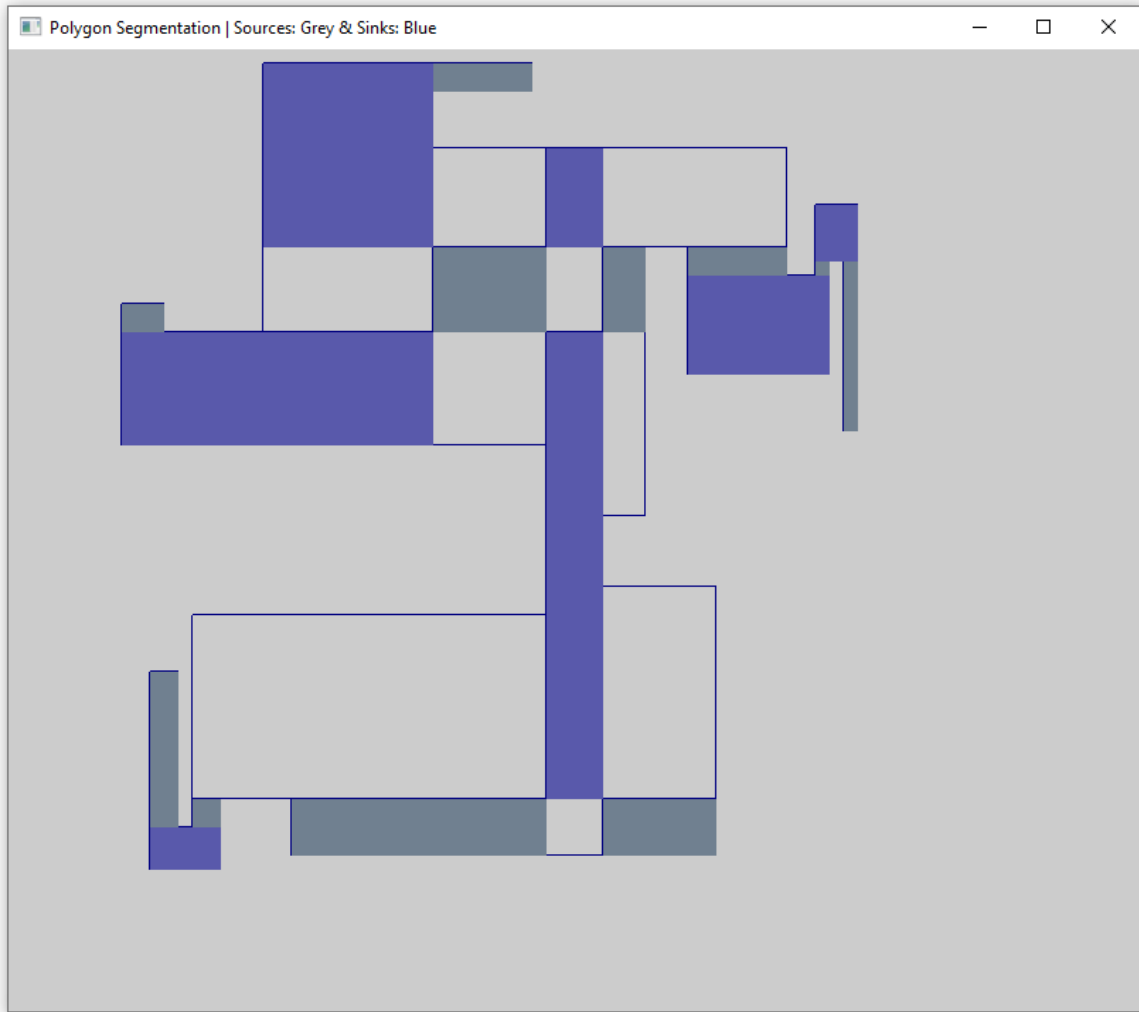
Half-edge found on the left: (80,400) - (380,400)

The new pair of coordinates is: (300,400)

Εφόσον έχει ολοκληρωθεί η επέκταση όλων των εσοχών του πολυγώνου, έχουν δημιουργηθεί επιμέρους περιοχές. Κάθε ημι-ακμή του πολυγώνου που δε βρίσκεται στο σύνορό του έχει μία κατεύθυνση που της έχει δοθεί προκειμένου να σχηματιστεί το διάγραμμα εσοχών (dent diagram), όπως περιγράφεται στην Ενότητα 2.1. Έτσι, υπολογίζονται και τυπώνονται οι πηγές (sources) και οι καταβόθρες (sinks) του πολυγώνου, σύμφωνα με τον ορισμό της Ενότητας 1.1. Ακολουθεί η οπτικοποίηση των πηγών και των καταβόθρων του πολυγώνου, με γκρι και μπλε χρώμα αντιστοίχως (Σχήμα 2.14).

Sources: 3 5 7 9 10 12 14 19 22 23 25

Sinks: 0 1 2 4 6 15 17



Σχήμα 2.14: Οπτικοποίηση πηγών και καταβοθρών, με γκρι και μπλε χρώμα αντίστοιχως.

Έπειτα, εκτελούμε μονότονο DFS για κάθε περιοχή του πολυγώνου. Παρακάτω τυπώνονται οι περιοχές που επισκεπτόμαστε από κάθε περιοχή.

Regions visited after DFS from region 0:

0 10 2 7 20 4 24 17

Regions visited after DFS from region 1:

1 12 13 6 18 19 20 8 4 21 5

Regions visited after DFS from region 2:

2 10 0 9

Regions visited after DFS from region 3:

3 15 26 6 18 11 22 21

Regions visited after DFS from region 4:

4 20 19 18 7 0 24 17 16 15 25 26 8 6 13 12 1 14 11 22 21

Regions visited after DFS from region 5:

5 1

Regions visited after DFS from region 6:

6 18 19 20 8 4 24 17 25 16 26 15 3 13 12 1 14 11 22 21

Regions visited after DFS from region 7:

7 20 4 24 17 0

Regions visited after DFS from region 8:

8 19 18 20 4 24 17 25 16 15 26 6 13 12 1 14 11 22 21

Regions visited after DFS from region 9:

9 2

Regions visited after DFS from region 10:

10 2 0

Regions visited after DFS from region 11:

11 22 21 6 18 19 20 8 4 24 17 25 16 26 15 3 13 14

Regions visited after DFS from region 12:

12 13 6 18 19 20 8 4 21 1

Regions visited after DFS from region 13:

13 6 18 19 20 8 4 11 22 21 12 1 14

Regions visited after DFS from region 14:

14 11 22 21 6 18 19 20 8 4 13

Regions visited after DFS from region 15:

15 26 6 18 19 20 8 4 11 22 21 25 24 16 17 23 3

Regions visited after DFS from region 16:

16 25 26 6 18 11 22 21 8 19 20 4 24 17 23 15

Regions visited after DFS from region 17:

17 23 16 25 26 6 18 11 22 21 8 19 15 24 4 20 7 0

Regions visited after DFS from region 18:

18 6 8 4 24 17 25 16 26 15 3 13 12 1 14 11 19 20

Regions visited after DFS from region 19:

19 8 4 24 17 25 16 15 26 6 13 12 1 14 11 18 20

Regions visited after DFS from region 20:

20 4 24 17 16 15 25 26 8 6 13 12 1 14 11 19 18 7 0

Regions visited after DFS from region 21:

21 6 8 4 24 17 25 16 26 15 3 13 12 1 14 11 22

Regions visited after DFS from region 22:

22 11 6 8 4 24 17 25 16 26 15 3 13 14 21

Regions visited after DFS from region 23:

23 17 16 15

Regions visited after DFS from region 24:

24 17 16 15 25 26 6 18 11 22 21 8 19 4 20 7 0

Regions visited after DFS from region 25:

25 16 17 15 26 6 18 11 22 21 8 19 20 4 24

Regions visited after DFS from region 26:

26 15 16 17 3 6 18 19 20 8 4 11 22 21 25 24

Για τη δημιουργία του γραφήματος H_s ορίζουμε ως κόμβους του γραφήματος όλες τις περιοχές στις οποίες έχει χωριστεί το πολύγωνο. Δύο τέτοιοι κόμβοι είναι γειτονικοί αν κάθε σημείο της μίας περιοχής s -βλέπει, δηλαδή βλέπει με ένα staircase path, κάθε σημείο της άλλης περιοχής. Για τη δημιουργία του πίνακα γειτνίασης χρησιμοποιούνται τα αποτελέσματα της εκτέλεσης DFS, δηλαδή τις λίστες με τις περιοχές που επισκεπτόμαστε από κάθε περιοχή. Στον πίνακα που ακολουθεί (Πίνακας 2.3) σημειώνουμε με “1” το κελί του πίνακα που αντιστοιχεί σε δύο γειτονικούς κόμβους του γραφήματος H_s . Η αρίθμηση ξεκινάει από πάνω προς τα κάτω και από τα αριστερά προς τα δεξιά για τις περιοχές 0 έως και 26 που αποτελούν τους κόμβους του γραφήματος.

Πίνακας 2.3: Πίνακας γειτνίασης για το γράφημα H_5 .

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
0	1	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	
1	0	1	0	0	1	1	1	0	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	
2	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	
4	1	1	0	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	
5	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	1	0	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	
7	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	
8	0	1	0	0	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	
9	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	0	0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	
12	0	1	0	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	
13	0	1	0	0	1	0	1	0	1	0	0	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	
14	0	0	0	0	1	0	1	0	1	0	0	1	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	
15	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
16	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
17	1	0	0	0	1	0	1	1	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
18	0	1	0	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	
19	0	1	0	0	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
20	1	1	0	0	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
21	0	1	0	1	1	0	1	0	1	0	0	1	1	1	1	1	1	1	0	0	0	1	1	0	1	1	1	
22	0	0	0	1	1	0	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	1	1	0	1	1	1	
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0	
24	1	0	0	0	1	0	1	1	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	
25	0	0	0	0	1	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	
26	0	0	0	1	1	0	1	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	

Από το γράφημα H_s κατασκευάζουμε το γράφημα H ορίζοντας ως κόμβους του γραφήματος όλες τις περιοχές του πολυγώνου. Δύο τέτοιοι κόμβοι u, v συνδέονται με ακμή αν υπάρχει κόμβος στο H_s που είναι γειτονικός, στο γράφημα H_s , και στον κόμβο u και στον v . Ακολουθεί ο πίνακας γειτνίασης για το γράφημα H (Πίνακας 2.4).

Πίνακας 2.4: Πίνακας γειτνίασης για το γράφημα H .

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
2	1	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0
3	0	1	0	0	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0
6	1	1	0	1	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0
11	1	1	0	1	1	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	0	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1
13	1	1	0	1	1	1	1	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1
14	1	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1
15	1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
16	1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
18	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
19	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
21	1	1	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1

22	1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	
23	1	0	0	1	1	0	1	1	1	0	0	1	0	0	0	1	1	1	1	1	1	0	1	1	1
24	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
25	1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1
26	1	1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Έχοντας κατασκευάσει το γράφημα H , ελέγχουμε για κάθε ζεύγος μη γειτονικών κόμβων εάν είναι 2-ζεύγη (2-pair). Παρακάτω παρουσιάζονται τα αποτελέσματα των 2-pair που εντοπίστηκαν στο συγκεκριμένο παράδειγμα.

- Two-pair: 26 - 25
- Two-pair: 26 - 24
- Two-pair: 26 - 23
- Two-pair: 26 - 22
- Two-pair: 26 - 21
- Two-pair: 26 - 20
- Two-pair: 26 - 19
- Two-pair: 26 - 18
- Two-pair: 26 - 17
- Two-pair: 26 - 16
- Two-pair: 26 - 15
- Two-pair: 26 - 11
- Two-pair: 26 - 8
- Two-pair: 26 - 6
- Two-pair: 26 - 4
- Two-pair: 26 - 3
- Two-pair: 14 - 13
- Two-pair: 14 - 12
- Two-pair: 14 - 7
- Two-pair: 14 - 1
- Two-pair: 14 - 0
- Two-pair: 10 - 9
- Two-pair: 10 - 2

Τέλος, τυπώνονται οι τέσσερις κλίκες του γραφήματος και ο κόμβος-αντιπρόσωπος κάθε κλίκας. Οι περιοχές που αντιστοιχούν στους κόμβους-αντιπροσώπους αποτελούν καταβόθρες του πολυγώνου εισόδου. Για παράδειγμα, ο κόμβος 2 είναι αντιπρόσωπος της κλίκας 2, 9, 10.

Cliques and Guard Locations

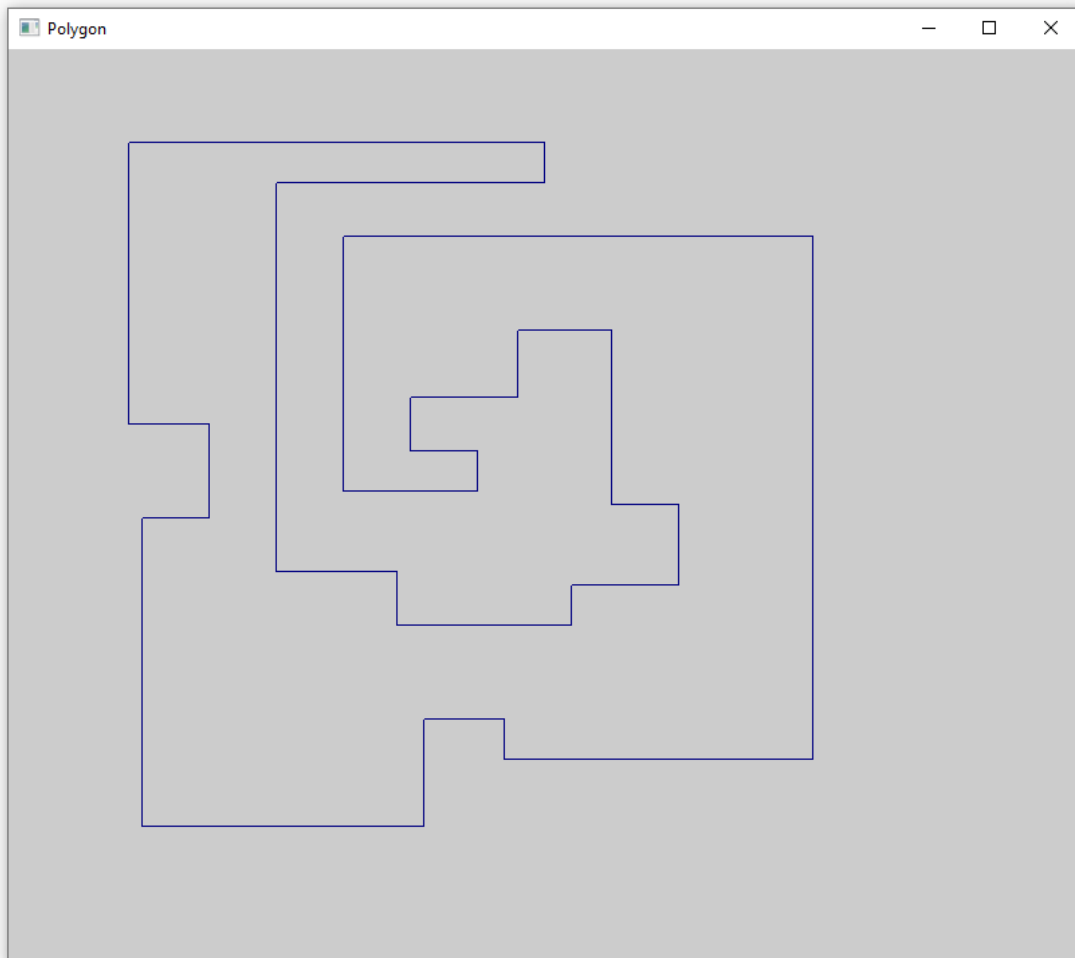
- 1) 5 --> 1
- 2) 2 9 10 --> 2
- 3) 0 1 7 12 13 14 --> 4
- 4) 3 4 6 8 11 15 16 17 18 19 20 21 22 23 24 25 26 --> 15

Ακολουθεί ένα επιπλέον παράδειγμα εκτέλεσης του προγράμματος για το ακόλουθο αρχείο εισόδου:

```
32
100 100
310 100
310 180
370 180
370 150
600 150
600 540
250 540
250 350
350 350
350 380
300 380
300 420
380 420
380 470
450 470
450 340
```

500 340
500 280
420 280
420 250
290 250
290 290
200 290
200 580
400 580
400 610
90 610
90 400
150 400
150 330
100 330

Στο Σχήμα 2.15 απεικονίζεται το πολύγωνο εισόδου, όπως ακριβώς εμφανίζεται κατά την εκτέλεση του προγράμματος.



Σχήμα 2.15: Οπτικοποίηση πολυγώνου εισόδου.

Στη συνέχεια, τυπώνονται οι εσοχές και τα αντίστοιχα dent line κάθε εσοχής. Σύμφωνα με τα ακόλουθα αποτελέσματα, στο συγκεκριμένο πολύγωνο υπάρχει μία Βόρεια Εσοχή (N-dent), δύο Νότιες Εσοχές (S-dents), δύο Ανατολικές Εσοχές (E-dents) και δύο Δυτικές Εσοχές (W-dents). Γίνεται ταξινόμηση των εσοχών έτσι ώστε να επεξεργαστούν, στη συνέχεια, με προκαθορισμένη σειρά. Ακολουθούν τα αποτελέσματα που τυπώνονται μετά την ταξινόμηση αυτή καθώς και το αναγνωριστικό (id) της ημι-ακμής που αποτελεί dent line για την εσοχή.

S-dent found!

The S-dent consists of:

1: (310,180)

2: (370,180)

E-dent found!

The E-dent consists of:

1: (300,380)

2: (300,420)

S-dent found!

The S-dent consists of:

1: (380,470)

2: (450,470)

W-dent found!

The W-dent consists of:

1: (500,340)

2: (500,280)

N-dent found!

The N-dent consists of:

1: (420,250)

2: (290,250)

E-dent found!

The E-dent consists of:

1: (200,290)

2: (200,580)

W-dent found!

The W-dent consists of:

1: (150,400)

2: (150,330)

Dent types:

1=north 2=south 3=west 4=east

Dent = 1 and its origin is 20

Dent = 2 and its origin is 2

Dent = 2 and its origin is 14

Dent = 3 and its origin is 17

Dent = 3 and its origin is 29

Dent = 4 and its origin is 11

Dent = 4 and its origin is 23

Έπειτα, επεξεργαζόμαστε κάθε εσοχή σύμφωνα με την ανωτέρω προκαθορισμένη σειρά και επεκτείνουμε κάθε γραμμή εσοχής (dent line) προς τις δύο κατευθύνσεις. Αναλυτικότερα, επεκτείνουμε πρώτα τις Βόρειες Εσοχές (N-dents) προς τα αριστερά και έπειτα προς τα δεξιά έως ότου συναντήσουμε το σύνορο του πολυγώνου. Για παράδειγμα, το dent line της Βόρειας Εσοχής είναι η ημι-ακμή ((420, 250), (290, 250)). Επεκτείνοντας την ημι-ακμή αυτή προς τα αριστερά συναντάμε την ημι-ακμή ((100, 330), (100, 100)) του πολυγώνου. Επομένως, στο σημείο τομής δημιουργείται μία νέα κορυφή με συντεταγμένες (100, 250). Η νέα ημι-ακμή που δημιουργείται, με κορυφή αφετηρίας το ένα άκρο του dent line και τελικό άκρο το νέο σημείο, είναι η ((290, 250), (100, 250)). Ομοίως, επεκτείνοντας το dent line προς τα δεξιά, συναντάμε την ημι-ακμή ((600, 150), (600, 540)). Στο σημείο που τέμνει το σύνορο του πολυγώνου, δημιουργείται η κορυφή (600, 250). Η νέα ημι-ακμή που δημιουργείται είναι η ((600, 250), (420, 250)). Σημειώνεται ότι η δίδυμη (twin) ημι-ακμή της είναι η ((420, 250), (600, 250)). Ομοίως, επεκτείνεται κάθε dent line των Νότιων Εσοχών (S-dents).

Dent No1

N-dent:

(x1, y1) = (420, 250)

(x2, y2) = (290, 250)

Edge found on the left: (100,330) - (100,100)

The new pair of coordinates on the left is: (100,250)

The new extended half-edge is: (290,250) - (100,250)

Edge found on the right: (600,150) - (600,540)

The new pair of coordinates on the right is: (600,250)

The new extended half-edge is: (600,250) - (420,250)

Dent No2

S-dent:

(x1, y1) = (310, 180)

(x2, y2) = (370, 180)

Edge found on the right: (600,150) - (600,250)

The new pair of coordinates on the right is: (600,180)

The new extended half-edge is: (370,180) - (600,180)

Edge found on the left: (100,250) - (100,100)

The new pair of coordinates on the left is: (100,180)

The new extended half-edge is: (100,180) - (310,180)

Dent No3

S-dent:

(x1, y1) = (380, 470)

(x2, y2) = (450, 470)

Edge found on the right: (600,250) - (600,540)

The new pair of coordinates on the right is: (600,470)

The new extended half-edge is: (450,470) - (600,470)

Edge found on the left: (250,540) - (250,350)

The new pair of coordinates on the left is: (250,470)

The new extended half-edge is: (250,470) - (380,470)

Dent No4

W-dent:

(x1, y1) = (500, 340)

(x2, y2) = (500, 280)

The number of horizontal half-edges we meet (except the border) by extending on the down side is: 2

Half-edge found on the right: (420,250) - (600,250)

The new pair of coordinates is: (500,250)

Half-edge found on the right: (370,180) - (600,180)

The new pair of coordinates is: (500,180)

Half-edge found on the right: (370,150) - (600,150)

The new pair of coordinates is: (500,150)

The number of horizontal half-edges we meet (except the border) by extending on the upper side is: 1

Half-edge found on the left: (600,470) - (450,470)

The new pair of coordinates is: (500,470)

Half-edge found on the left: (600,540) - (250,540)

The new pair of coordinates is: (500,540)

Dent No5

W-dent:

(x1, y1) = (150, 400)

(x2, y2) = (150, 330)

The number of horizontal half-edges we meet (except the border) by extending on the down side is: 2

Half-edge found on the right: (100,250) - (290,250)

The new pair of coordinates is: (150,250)

Half-edge found on the right: (100,180) - (310,180)

The new pair of coordinates is: (150,180)

Half-edge found on the right: (100,100) - (310,100)

The new pair of coordinates is: (150,100)

The number of horizontal half-edges we meet (except the border) by extending on the upper side is: 0

Half-edge found on the left: (400,610) - (90,610)

The new pair of coordinates is: (150,610)

Dent No6

E-dent:

(x1, y1) = (300, 380)

(x2, y2) = (300, 420)

The number of horizontal half-edges we meet (except the border) by extending on the upper side is: 1

Half-edge found on the right: (380,470) - (250,470)

The new pair of coordinates is: (300,470)

Half-edge found on the right: (500,540) - (250,540)

The new pair of coordinates is: (300,540)

Half-edge found on the left: (250,350) - (350,350)

The new pair of coordinates is: (300,350)

Dent No7

E-dent:

(x1, y1) = (200, 290)

(x2, y2) = (200, 580)

The number of horizontal half-edges we meet (except the border) by extending on the upper side is: 0

Half-edge found on the right: (400,610) - (150,610)

The new pair of coordinates is: (200,610)

Half-edge found on the left: (150,250) - (290,250)

The new pair of coordinates is: (200,250)

Half-edge found on the left: (150,180) - (310,180)

The new pair of coordinates is: (200,180)

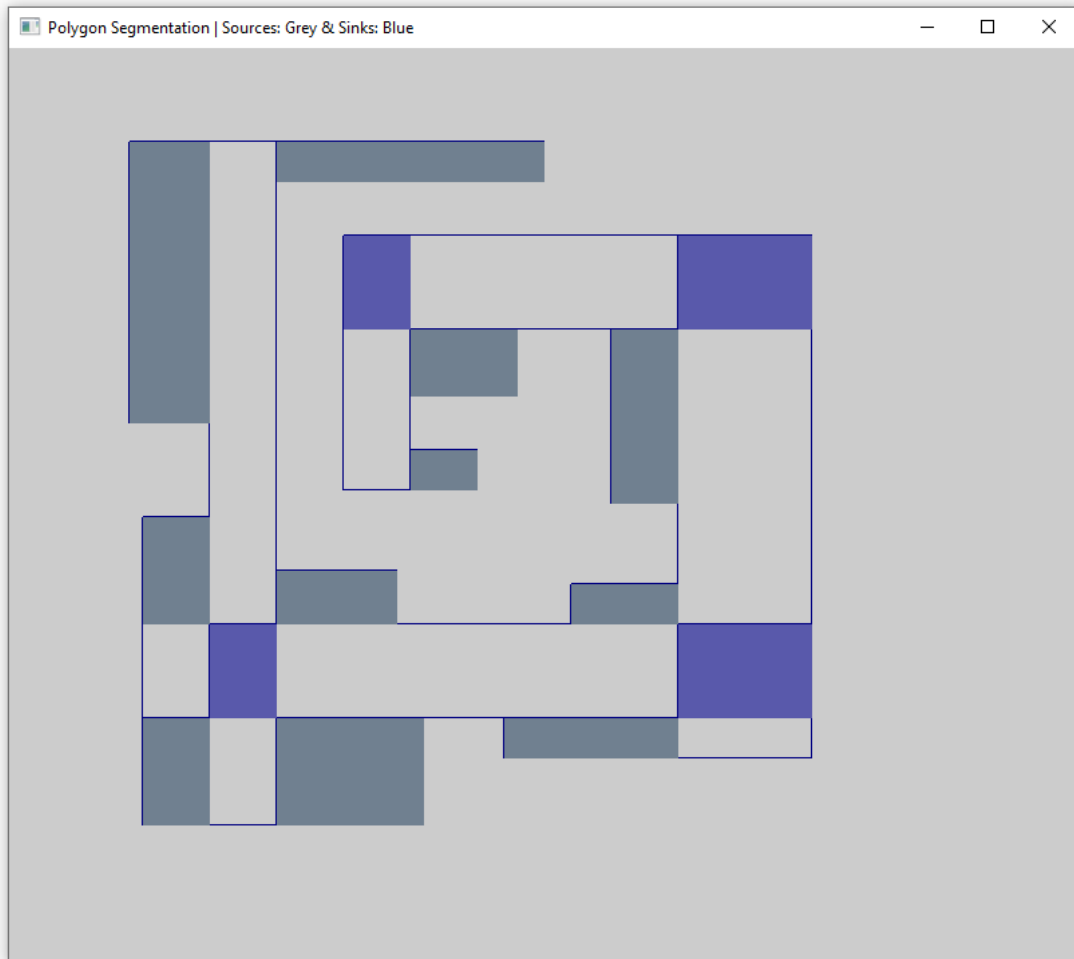
Half-edge found on the left: (150,100) - (310,100)

The new pair of coordinates is: (200,100)

Εφόσον έχει ολοκληρωθεί η επέκταση όλων των εσοχών του πολυγώνου, έχουν δημιουργηθεί επιμέρους περιοχές. Κάθε ημι-ακμή του πολυγώνου που δε βρίσκεται στο σύνορό του έχει μία κατεύθυνση που της έχει δοθεί προκειμένου να σχηματιστεί το διάγραμμα εσοχών (dent diagram), όπως περιγράφεται στην Ενότητα 2.1. Έτσι, υπολογίζονται και τυπώνονται οι πηγές (sources) και οι καταβόθρες (sinks) του πολυγώνου, σύμφωνα με τον ορισμό της Ενότητας 1.1. Ακολουθεί η οπτικοποίηση των πηγών και των καταβοθρών του πολυγώνου, με γκρι και μπλε χρώμα αντιστοίχως (Σχήμα 2.16).

Sources: 7 9 10 12 14 15 16 18 19 20 22

Sinks: 0 2 8 11



Σχήμα 2.16: Οπτικοποίηση πηγών και καταβροθρών, με γκρι και μπλε χρώμα αντίστοιχως.

Έπειτα, εκτελούμε μονότονο DFS για κάθε περιοχή του πολυγώνου. Παρακάτω τυπώνονται οι περιοχές που επισκεπτόμαστε από κάθε περιοχή.

Regions visited after DFS from region 0:

0 21 8 1 15 12 13 14 4 22 9 7 20 3 5 10 17 11 2

Regions visited after DFS from region 1:

1 19 15 12 13 14 8 21 22 9 3 0 4 20

Regions visited after DFS from region 2:

2 17 11 6 16 10 5 7 21 8 13 14 4 22 9 0 3

Regions visited after DFS from region 3:

3 0 21 8 1 15 12 13 7 20 5 10 17 11 2 9

Regions visited after DFS from region 4:

4 22 21 0 5 2 7 20 8 1 19 15 12 13 14

Regions visited after DFS from region 5:

5 10 17 11 7 21 8 13 14 4 22 9 0 3 2

Regions visited after DFS from region 6:

6 18 16 17 2 11

Regions visited after DFS from region 7:

7 5 0 3 2 21 8 13 14 4 22 9

Regions visited after DFS from region 8:

8 21 22 9 3 0 5 2 7 20 1 19 15 12 13 14 4

Regions visited after DFS from region 9:

9 3 0 5 2 21 8 1 15 12 13 7 20

Regions visited after DFS from region 10:

10 5 0 3 2 17 11

Regions visited after DFS from region 11:

11 17 16 10 5 0 3 2 6 18

Regions visited after DFS from region 12:

12 1 19 8 21 22 9 3 0 4 20 13 14

Regions visited after DFS from region 13:

13 8 21 22 9 3 0 5 2 7 20 1 19 4 12 14

Regions visited after DFS from region 14:

14 4 22 21 0 5 2 7 20 8 1 19 13 12

Regions visited after DFS from region 15:

15 1 19 8 21 22 9 3 0 4 20

Regions visited after DFS from region 16:

16 6 11 17 2

Regions visited after DFS from region 17:

17 11 6 16 10 5 0 3 2

Regions visited after DFS from region 18:

18 6 11

Regions visited after DFS from region 19:

19 1 15 12 13 14 8 4

Regions visited after DFS from region 20:

20 1 15 12 13 14 8 4 21 22 9 3 0

Regions visited after DFS from region 21:

21 8 1 15 12 13 14 4 22 9 3 0 5 2 7 20

Regions visited after DFS from region 22:

22 4 8 1 15 12 13 14 21 0 5 2 7 20

Για τη δημιουργία του γραφήματος H_s ορίζουμε ως κόμβους του γραφήματος όλες τις περιοχές στις οποίες έχει χωριστεί το πολύγωνο. Δύο τέτοιοι κόμβοι είναι γειτονικοί αν κάθε σημείο της μίας περιοχής s -βλέπει, δηλαδή βλέπει με ένα staircase path, κάθε σημείο της άλλης περιοχής. Για τη δημιουργία του πίνακα γειτνίασης χρησιμοποιούνται τα αποτελέσματα της εκτέλεσης DFS, δηλαδή οι λίστες με

τις περιοχές που επισκεπτόμαστε από κάθε περιοχή. Στον πίνακα που ακολουθεί (Πίνακας 2.5) σημειώνουμε με “1” το κελί του πίνακα που αντιστοιχεί σε δύο γειτονικούς κόμβους του γραφήματος H_S . Η αρίθμηση ξεκινάει από πάνω προς τα κάτω και από τα αριστερά προς τα δεξιά για τις περιοχές 0 έως και 22 που αποτελούν τους κόμβους του γραφήματος.

Πίνακας 2.5: Πίνακας γειτνίασης για το γράφημα H_S .

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1
1	1	1	0	1	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1
2	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	1	1
3	1	1	1	1	0	1	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	0
4	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1
5	1	0	1	1	1	1	0	1	1	1	1	1	0	1	1	0	0	1	0	0	0	1	1
6	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	0	0
7	1	0	1	1	1	1	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	1
8	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	1	1	1	1
9	1	1	1	1	0	1	0	1	1	1	0	0	1	1	0	1	0	0	0	0	1	1	0
10	1	0	1	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0
11	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	1	0	0	0	0
12	1	1	0	1	1	0	0	0	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1
13	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1
14	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	0	0	0	0	1	1	1	1
15	1	1	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	1	1	1
16	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
17	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0
18	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
19	0	1	0	0	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0
20	1	1	0	1	1	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1
21	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1

22 | 1 1 1 0 1 1 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1 1 1

Από το γράφημα H_S κατασκευάζουμε το γράφημα H ορίζοντας ως κόμβους του γραφήματος όλες τις περιοχές του πολυγώνου. Δύο τέτοιοι κόμβοι u, v συνδέονται με ακμή αν υπάρχει κόμβος στο H_S που είναι γειτονικός, στο γράφημα H_S , και στον κόμβο u και στον v . Ακολουθεί ο πίνακας γειτνίασης για το γράφημα H (Πίνακας 2.6).

Πίνακας 2.6: Πίνακας γειτνίασης για το γράφημα H .

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1
2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
5	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	0	1	1	1	1	0	1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	1
7	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
8	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
9	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1
12	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1
15	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1
16	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	0	0	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
18	1	0	1	1	0	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0
19	1	1	1	1	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	0	0	1	1

20	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0

Έχοντας κατασκευάσει το γράφημα H , ελέγχουμε για κάθε ζεύγος μη γειτονικών κόμβων εάν είναι 2-ζεύγη (2-pair). Παρακάτω παρουσιάζονται τα αποτελέσματα των 2-pair που εντοπίστηκαν στο συγκεκριμένο παράδειγμα.

- Two-pair: 22 - 21
- Two-pair: 22 - 20
- Two-pair: 22 - 19
- Two-pair: 22 - 15
- Two-pair: 22 - 14
- Two-pair: 22 - 13
- Two-pair: 22 - 12
- Two-pair: 22 - 9
- Two-pair: 22 - 8
- Two-pair: 22 - 7
- Two-pair: 22 - 5
- Two-pair: 22 - 4
- Two-pair: 22 - 3
- Two-pair: 22 - 2
- Two-pair: 22 - 1
- Two-pair: 22 - 0
- Two-pair: 18 - 17
- Two-pair: 18 - 16
- Two-pair: 18 - 11
- Two-pair: 18 - 10
- Two-pair: 18 - 6

Τέλος, τυπώνονται οι δύο κλίκες του γραφήματος και ο κόμβος-αντιπρόσωπος κάθε κλίκας. Οι περιοχές που αντιστοιχούν στους κόμβους-αντιπροσώπους

αποτελούν καταβόθρες του πολυγώνου εισόδου. Για παράδειγμα, ο κόμβος 11 είναι αντιπρόσωπος της κλίμακας 6, 10, 11, 16, 17, 18.

Cliques and Guard Locations

1) 6 10 11 16 17 18 --> 11

2) 0 1 2 3 4 5 7 8 9 12 13 14 15 19 20 21 22 --> 8

ΚΕΦΑΛΑΙΟ 3

ΑΛΓΟΡΙΘΜΟΣ ΚΑΤΑΣΚΕΥΗΣ ΟΡΘΟΓΩΝΙΩΝ

ΠΟΛΥΓΩΝΩΝ

-
- 3.1 Ο Αλγόριθμος Κατασκευής Ορθογωνίων Πολυγώνων
 - 3.2 Παράδειγμα Εφαρμογής του Αλγορίθμου
 - 3.3 Η Υλοποίηση
-

3.1 Ο Αλγόριθμος Κατασκευής Ορθογωνίων Πολυγώνων

Η δυνατότητα κατασκευής “τυχαίων” πολυγώνων, συνήθως με δοθέν σύνολο ή πλήθος κορυφών, αποτελεί ένα πολύ ενδιαφέρον ερευνητικό πρόβλημα καθώς είναι ιδιαίτερα χρήσιμη στην πειραματική μελέτη αλγορίθμων για πολύγωνα και την ανάλυση της συμπεριφοράς τους στην πράξη. Κάποιες πρόσφατες δημοσιεύσεις εξετάζουν την τυχαία παραγωγή απλών πολυγώνων με δοθέν σύνολο κορυφών [13, 14]. Καθώς δεν είναι γνωστός κανείς πολυωνυμικός αλγόριθμος για αυτό το πρόβλημα, το ενδιαφέρον των ερευνητών εστιάστηκε είτε σε ευρετικούς αλγορίθμους [13] είτε σε ειδικές κατηγορίες πολυγώνων όπως μονότονα [14] ή αστερόσχημα (star-shaped) [15]. Για το πρόβλημα της παραγωγής απλών ορθογωνίων πολυγώνων, ο O'Rourke ανέπτυξε έναν αλγόριθμο στο πλαίσιο της πειραματικής μελέτης των αλγορίθμων που περιγράφηκαν στην εργασία [16] και παρατήρησε ότι δεν είναι εύκολο να εξασφαλισθεί το τελικό πλήθος κορυφών του παραχθέντος

ορθογωνίου πολυγώνου. Το 2004, οι Tomas και Bajuelos [17] περιέγραψαν δύο αλγορίθμους για το εν λόγω πρόβλημα. Ο ένας είναι πολυωνυμικού χρόνου και επιτρέπει την κατασκευή ορθογωνίων πολυγώνων με αυξανόμενο πλήθος κορυφών ξεκινώντας από ένα τετράγωνο μοναδιαίας πλευράς. Ο δεύτερος βασίζεται σε προγραμματισμό με περιορισμούς και επιτρέπει τον υπολογισμό όλων των ορθογωνίων πολυγώνων με n κορυφές (χωρίς συννευθιακές ακμές) που μπορούν να σχεδιασθούν σε ένα $(n/2) \times (n/2)$ -πλέγμα για μικρές τιμές του n .

Ο αλγόριθμος κατασκευής ορθογωνίων πολυγώνων που θα παρουσιάσουμε βασίζεται στη χρήση ενός πίνακα $M[0..V-1, 0..H-1]$ με στοιχεία που έχουν τιμή μεγαλύτερη ή ίση του μηδενός, όπου τα H, V δίδονται από τον χρήστη και ορίζουν ένα περικλείον ορθογώνιο παραλληλόγραμμο (bounding box) για το τελικό ορθογώνιο πολύγωνο. Κάθε στοιχείο του πίνακα αντιστοιχεί σε ένα μικρό τετράγωνο πλευράς L . Συνολικά, ο πίνακας αντιστοιχεί στην ορθογώνια περιοχή του επιπέδου με κάτω αριστερή κορυφή $(0, 0)$ και άνω δεξιά κορυφή την $(V*L, H*L)$. Ειδικά, το στοιχείο $M[i, j]$ αντιστοιχεί στο τετράγωνο με κάτω αριστερή κορυφή $(i*L, j*L)$ και άνω δεξιά κορυφή $((i+1)*L, (j+1)*L)$ και είναι εκτός του πολυγώνου αν $M[i, j] = 0$ αλλιώς είναι εντός του πολυγώνου.

Προκειμένου να μην έχει εκφυλισμένα σημεία το πολύγωνο, ορίζουμε μία παράμετρο E (expansion) και ουσιαστικά κάθε τετράγωνο πλευράς L που ανήκει στο πολύγωνο, επεκτείνεται κατά E προς κάθε πλευρά (Σχήμα 3.1). Έτσι, αν δύο τετράγωνα πλευράς L είναι, όπως φαίνεται στο Σχήμα 3.1, το πολύγωνο δεν έχει εκφυλισμένα σημεία. Σημειώνεται ότι στο Σχήμα 3.1 τα μαύρα τετράγωνα αντιστοιχούν σε τετράγωνα εντός του πολυγώνου, ενώ τα λευκά στο εξωτερικό του πολυγώνου.



Σχήμα 3.1: Επέκταση πλευράς κατά E .

Η κατασκευή βασίζεται σε μία παράμετρο πυκνότητας D , που δίδεται από τον χρήστη, και η ιδέα είναι σε κάθε γραμμή το αναμενόμενο πλήθος των τετραγώνων που ανήκουν στο πολύγωνο να είναι $H \cdot D$.

Παράλληλα, θα πρέπει να έχουμε ένα τελικό ορθογώνιο πολύγωνο (και όχι διάφορα τμήματα πολυγώνου) και αυτό το πολύγωνο να είναι απλό. Επομένως, θα πρέπει να εγγυηθούμε ότι δεν θα δημιουργηθεί κάποια οπή στο πολύγωνο.

Ο αλγόριθμος περιλαμβάνει 2 βήματα: στο πρώτο, γεμίζουμε τον πίνακα $M[0..V-1, 0..H-1]$ με βάση την παράμετρο πυκνότητας $D \in (0,1)$, ενώ στο δεύτερο, μετατρέπουμε το περιεχόμενο του πίνακα M σε ένα ορθογώνιο πολύγωνο.

3.1.1 Ο Αλγόριθμος Υπολογισμού του Πίνακα M

Υπενθυμίζουμε ότι θετικές τιμές στα στοιχεία του πίνακα M αντιστοιχούν σε τετράγωνα εντός του πολυγώνου, ενώ μηδενικές τιμές σε τετράγωνα στο εξωτερικό του πολυγώνου και ότι θεωρούμε ότι οι δείκτες γραμμών του πίνακα M αυξάνουν από κάτω προς τα επάνω και οι δείκτες στηλών του πίνακα από αριστερά προς τα δεξιά. Γεμίζουμε τον πίνακα γραμμή-γραμμή από επάνω προς τα κάτω, δηλαδή από τη γραμμή $V-1$ προς τη γραμμή 0 .

Ξεκινώντας από την $(V-1)$ -οστή γραμμή, γεμίζουμε τις αντίστοιχες στήλες επιλέγοντας έναν αριθμό τυχαία και ομοιόμορφα στο διάστημα $[0,1)$. Εάν ο αριθμός αυτός είναι μικρότερος από την τιμή της παραμέτρου πυκνότητας D που έχουμε ορίσει, τότε το συγκεκριμένο κελί του πίνακα M βρίσκεται εντός του πολυγώνου ενώ, σε αντίθετη περίπτωση, το κελί βρίσκεται εκτός του πολυγώνου και του αναθέτουμε την τιμή 0 . Για τα κελιά εντός του πολυγώνου φροντίζουμε ώστε τα κελιά σε κάθε ομάδα συνεχόμενων κελιών να φέρουν την ίδια τιμή (ανήκουν στο ίδιο τμήμα του πολυγώνου) ενώ διαφορετικές ομάδες να χαρακτηρίζονται από διαφορετικές τιμές στον πίνακα M .

Στη συνέχεια, επεξεργαζόμαστε τις γραμμές $V-2, V-3, \dots, 1$ του πίνακα M . Για κάθε μία από αυτές τις γραμμές, έστω i , κάνουμε τα εξής:

1. Διασχίζουμε τη γραμμή $i+1$ (δηλαδή τη γραμμή αμέσως από πάνω) του πίνακα M και αποθηκεύουμε σε έναν βοηθητικό πίνακα (έστω A) τα διαστήματα δεικτών στηλών της γραμμής για τα τμήματα εντός του πολυγώνου και την

αντίστοιχη τιμή των στοιχείων. Αξίζει να σημειωθεί ότι τα στοιχεία $M[i+1,*]$ που αντιστοιχούν σε κάθε διάστημα έχουν την ίδια θετική τιμή η οποία και αντιστοιχεί σε ένα διαφορετικό τμήμα του πολυγώνου.

2. Εξασφαλίζουμε ότι για κάθε ζεύγος διαστημάτων του ίδιου τμήματος του πολυγώνου στη γραμμή $i+1$, θα υπάρχει κενό κελί στη γραμμή i μεταξύ αυτών των διαστημάτων ώστε να μη σχηματισθεί οπή με τη θέση του κενού κελιού να επιλέγεται τυχαία και ομοιόμορφα. Κάθε τέτοια τιμή κενού κελιού για την τρέχουσα γραμμή, εισάγεται σε ένα σύνολο $OutPolygon$. Εξασφαλίζουμε, επίσης, ότι κάθε τμήμα του πολυγώνου (σύνολο ξένων μεταξύ τους διαστημάτων στηλών με την ίδια τιμή στη γραμμή $i+1$) θα συνεχίσει να υφίσταται στη γραμμή i , έτσι ώστε όλα αυτά τα τμήματα να συνδεθούν σε ένα πολύγωνο. Έτσι, διασχίζοντας τον πίνακα A , χωρίζουμε τα αποθηκευμένα διαστήματα στηλών από τη γραμμή $i+1$ σε υποσύνολα, ανάλογα με την τιμή των στοιχείων του πίνακα M σε αυτά. Για κάθε τέτοιο υποσύνολο διαστημάτων $\{ [s_0, t_0], [s_1, t_1], \dots, [s_k, t_k] \}$, επιλέγουμε τυχαία και ομοιόμορφα έναν δείκτη στήλης c μεταξύ των τιμών στο σύνολο $\{s_0-1, s_0, \dots, t_0, t_0+1\} \cup \{s_1-1, s_1, \dots, t_1+1\} \cup \dots \cup \{s_k-1, s_k, \dots, t_k+1\}$ και επιβάλλουμε ότι το στοιχείο $M[i, c]$ θα έχει θετική τιμή. Κάθε τέτοια τιμή κελιού εντός του πολυγώνου για την τρέχουσα γραμμή, εισάγεται σε ένα σύνολο $InPolygon$.

3. Αν ισχύει ότι $H > |InPolygon| + |OutPolygon|$ και $H*D > |InPolygon|$ (όπου $|A|$ δηλώνει το μέγεθος του συνόλου A), γεμίζουμε τις υπόλοιπες θέσεις της τρέχουσας γραμμής του πίνακα M με πιθανότητα

$$D' = (H*D - |InPolygon|) / (H - |InPolygon| - |OutPolygon|).$$

Κατόπιν, αναθέτουμε τις τελικές τιμές στα στοιχεία της γραμμής i εντός του πολυγώνου, ελέγχοντας αν ένα διάστημα στη γραμμή $i+1$ συνδέει δύο ή περισσότερα τμήματα του πολυγώνου στη γραμμή i και, αντίστοιχα, αν κάποιο διάστημα στη γραμμή i συνδέει δύο ή περισσότερα τμήματα του πολυγώνου στη γραμμή $i+1$.

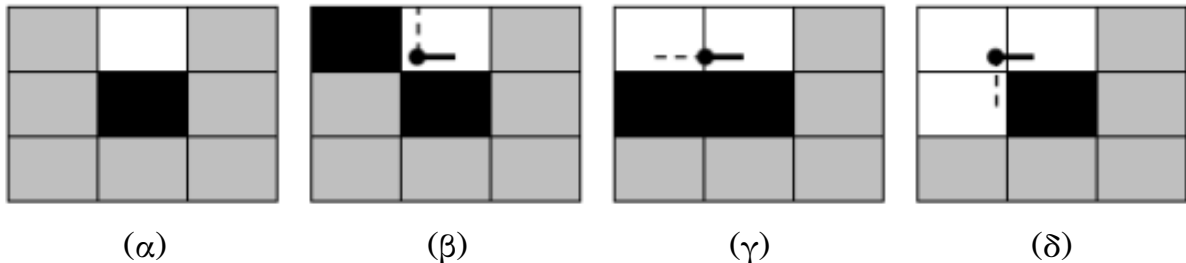
Τέλος, συμπληρώνουμε τα στοιχεία της γραμμής 0 του πίνακα M , όπου εκτός από το να φροντίσουμε να μην σχηματισθούν οπές και καθένα από τα διάφορα τμήματα του πολυγώνου να συνεχισθεί στη γραμμή 0 , εξασφαλίζουμε ότι όλα τα τμήματα συνδέονται σε ένα. Εφόσον, μετά από τα παραπάνω, έχουμε στη γραμμή

Ο λιγότερα από $H \cdot D$ κελιά εντός του πολυγώνου, γεμίζουμε τις υπόλοιπες θέσεις της γραμμής 0 με πιθανότητα D' όπως παραπάνω. Καθώς αυτή η διαδικασία ενδέχεται να δημιουργήσει απομονωμένα τμήματα του πολυγώνου, με ένα επιπλέον πέρασμα τα συνδέουμε στο υπάρχον πολύγωνο.

Η πολυπλοκότητα χρόνου αυτού του βήματος του αλγορίθμου είναι $O(VH^2)$ και η πολυπλοκότητα χώρου είναι $O(VH)$.

3.1.2 Ο Αλγόριθμος Εξαγωγής Ορθογωνίου Πολυγώνου

Με βάση τις τιμές του πίνακα M , στο βήμα αυτό εξάγουμε το αντίστοιχο ορθογώνιο πολύγωνο. Η ιδέα είναι ότι όποτε έχουμε ένα κελί του πίνακα M που ανήκει στο πολύγωνο δίπλα σε ένα άλλο που δεν ανήκει στο πολύγωνο, υπολογίζουμε ένα κατάλληλο ευθύγραμμο τμήμα το οποίο θα ανήκει στο σύνορο του υπολογιζόμενου ορθογωνίου πολυγώνου λαμβάνοντας υπόψιν, όπως αναφέραμε, και την παράμετρο E που χρησιμοποιείται προκειμένου να μην εμφανισθούν εκφυλισμένα σημεία στο πολύγωνο (Σχήμα 3.1).



Σχήμα 3.2: Τα τετράγωνα με μαύρο χρώμα είναι εντός του πολυγώνου, με λευκό χρώμα είναι εκτός του πολυγώνου και με γκριζο χρώμα μπορεί να είναι οτιδήποτε από τα δύο. Η συνέχεια του συνόρου του πολυγώνου απεικονίζεται με τη διακεκομμένη γραμμή.

Αναλυτικότερα, για κάθε στοιχείο $M[i, j]$ του πίνακα M με τιμή διάφορη του μηδενός, ελέγχουμε εάν τα γειτονικά στοιχεία (πάνω, κάτω, αριστερά, δεξιά) είναι κενά, δηλαδή έχουν τιμή ίση με το μηδέν, και εφόσον είναι, δημιουργούμε ένα ευθύγραμμο τμήμα. Ενδεικτικά αν η τρέχουσα γραμμή την οποία εξετάζουμε είναι η τελευταία γραμμή ($V-1$) του πίνακα M ή $M[i+1, j] = 0$ (Σχήμα 3.2.α), τότε

δημιουργούμε ένα ευθύγραμμο τμήμα πάνω από το στοιχείο $M[i, j]$. Το αριστερό άκρο (p) αυτού του ευθύγραμμου τμήματος ορίζεται ως εξής:

Αν ($j == 0$ OR ($M[i, j-1] == 0$ AND $M[i+1, j+1] == 0$))

$p \leftarrow ((i+1+E)*L, (j-E)*L);$ /* Σχήμα 3.2.δ */

Αλλιώς Αν ($M[i+1, j-1] > 0$)

$p \leftarrow ((i+1+E)*L, (j+E)*L);$ /* Σχήμα 3.2.β */

Αλλιώς /* $M[i+1, j-1] == 0$ AND $M[i, j-1] > 0$ */

$p \leftarrow ((i+1+E)*L, j*L);$ /* Σχήμα 3.2.γ */

Ομοίως, ορίζουμε το δεξί άκρο (q) του ευθύγραμμου τμήματος.

Στη συνέχεια, συνδέουμε σε μια αλυσίδα τα ευθύγραμμα τμήματα που υπολογίσαμε στο προηγούμενο βήμα, η οποία είναι το σύνορο του πολυγώνου. Τέλος, σε περίπτωση διαδοχικών συνευθειακών ευθυγράμμων τμημάτων, αφαιρούμε τις ενδιάμεσες κορυφές και έχουμε την τελική ακολουθία κορυφών του παραγόμενου ορθογωνίου πολυγώνου.

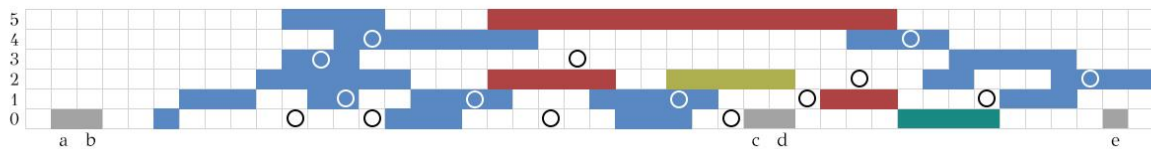
Η πολυπλοκότητα χρόνου και χώρου αυτού του βήματος του αλγορίθμου είναι $O(VH)$.

3.2 Παράδειγμα Εφαρμογής του Αλγορίθμου

Ακολουθεί ένα παράδειγμα για το γέμισμα του πίνακα M με “τυχαίες” τιμές. Θεωρούμε ότι οι δείκτες γραμμών του πίνακα αυξάνουν από κάτω προς τα πάνω και οι δείκτες στηλών του πίνακα από αριστερά προς τα δεξιά. Το γέμισμα του πίνακα M ξεκινάει από τη γραμμή $V-1$, δηλαδή τη γραμμή 5, και οι τιμές αυτές είναι ίδιες για τα κελιά του πίνακα M που ανήκουν στο ίδιο τμήμα του πολυγώνου. Στο Σχήμα 3.3 τα πρώτα τέσσερα κελιά του πίνακα στη γραμμή 5 έχουν την ίδια τιμή, έστω την τιμή 1, και απεικονίζονται με μπλε χρώμα, ενώ τα υπόλοιπα κελιά απεικονίζονται με κόκκινο χρώμα και έχουν διαφορετική τιμή, έστω την τιμή 2. Εξασφαλίζουμε ότι σε κάθε γραμμή θα υπάρχει κενό κελί ώστε να μη σχηματισθεί οπή. Στο Σχήμα 3.3 σημειώνουμε τα κελιά του πίνακα που πρέπει να μείνουν εκτός του

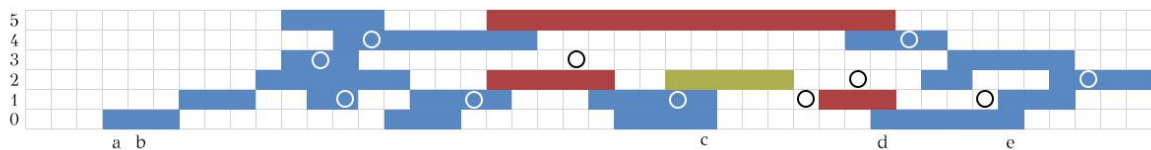
πολυγώνου με έναν μαύρο κύκλο με λευκό εσωτερικό. Αντίστοιχα, σημειώνουμε με λευκό κύκλο και έγχρωμο εσωτερικό, τα κελιά του πίνακα που πρέπει να είναι εντός του τελικού πολυγώνου.

Στη γραμμή 1 του πίνακα M (Σχήμα 3.3) τα κελιά με ίδιο χρώμα αντιστοιχούν στο ίδιο τμήμα του πολυγώνου. Συγκεκριμένα, με την ολοκλήρωση της επεξεργασίας της γραμμής 1, τα κελιά με μπλε χρώμα έχουν την ίδια τιμή και αυτό σημαίνει ότι ενώνονται σε ένα τμήμα του πολυγώνου στις ανώτερες γραμμές του πίνακα.



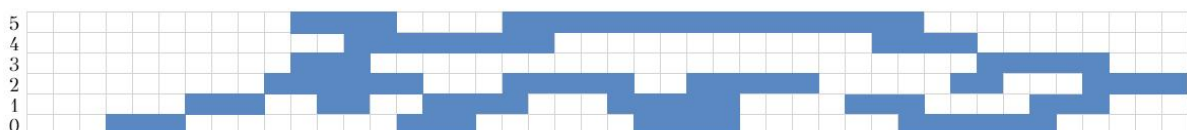
Σχήμα 3.3: Παράδειγμα εφαρμογής του αλγορίθμου.

Στη γραμμή 0, τα κελιά με πράσινο χρώμα προστίθενται ώστε να συνδεθεί το μπλε με το κόκκινο τμήμα της γραμμής 1 που επίσης εξασφαλίζουν ότι όλο αυτό το τμήμα συνεχίζεται στη γραμμή 0 (Σχήμα 3.4). Τα κελιά με μπλε χρώμα στη γραμμή 0 αντιστοιχούν σε κελιά που συνδέονται με το υπόλοιπο πολύγωνο ενώ αυτά με γκρι χρώμα είναι απομονωμένα. Έτσι, στο τελευταίο πέρασμα, τα κελιά με γκρι χρώμα μεταφέρονται και συνδέονται στο πολύγωνο (Σχήμα 3.4).



Σχήμα 3.4: Παράδειγμα εφαρμογής του αλγορίθμου.

Στο Σχήμα 3.5 απεικονίζεται το τελικό πολύγωνο που προκύπτει.



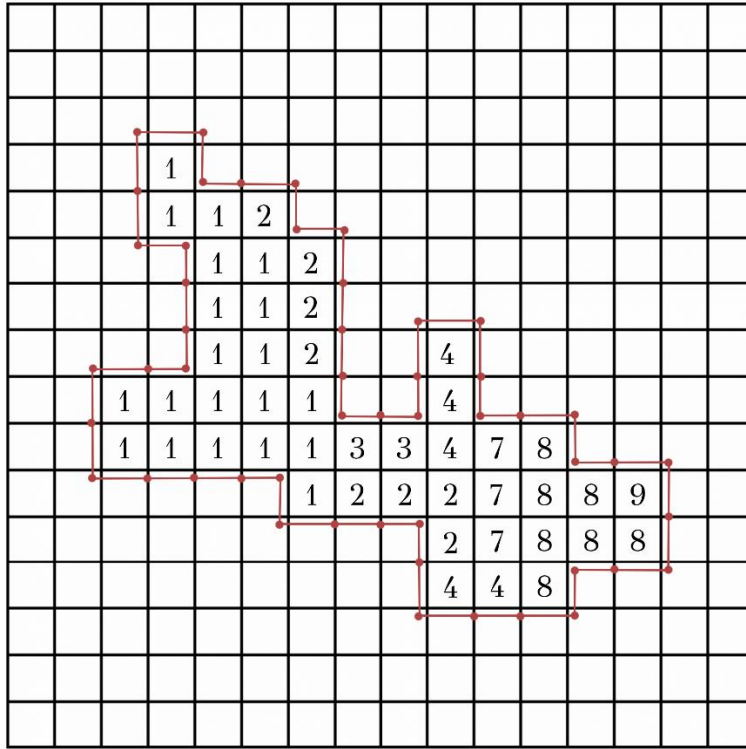
Σχήμα 3.5: Απεικόνιση τελικού πολυγώνου.

Ας θεωρήσουμε τώρα ένα παράδειγμα εφαρμογής του δευτέρου βήματος του αλγορίθμου και έστω ο πίνακας M που απεικονίζεται στο Σχήμα 3.6 και στον οποίο τα κενά κελιά έχουν τιμή 0.

Σε αυτήν την περίπτωση, ο αλγόριθμος παράγει 50 ευθύγραμμα τμήματα (Σχήμα 3.7) για παράμετρο E (expansion) $E=0.05*L$ και πλευρά τετραγώνου $L=1$. Τα ευθύγραμμα τμήματα δίνονται αναλυτικά παρακάτω όπου για κάθε ευθύγραμμο τμήμα, ο πρώτος και ο δεύτερος αριθμός αντιστοιχούν στις συντεταγμένες (P_x, P_y) του ενός άκρου του και ο τρίτος και τέταρτος αριθμός στις συντεταγμένες του δεύτερου άκρου του (Q_x, Q_y) . Οι δύο επόμενοι αριθμοί αντιστοιχούν στα δύο γειτονικά ευθύγραμμα τμήματα, incident1 και incident2, και τα οποία μας βοηθούν να συνδέσουμε τα ευθύγραμμα τμήματα σε μια αλυσίδα. Π.χ., το ευθύγραμμο τμήμα 0, έχει ως άκρα τις συντεταγμένες $(2.5, 4.5)$ και $(2.5, 2.5)$ και ως γείτονες τα ευθύγραμμα τμήματα 2 και 1.

				1															
				1	1	2													
					1	1	2												
					1	1	2												
						1	1	2			4								
				1	1	1	1	1			4								
				1	1	1	1	1	3	3	4	7	8						
								1	2	2	2	7	8	8	9				
											2	7	8	8	8				
											4	4	8						

Σχήμα 3.6: Παράδειγμα του πίνακα M.



Σχήμα 3.7: Σχηματισμός πολυγώνου.

Παραγόμενα ευθύγραμμα τμήματα:

0: 2.5 4.5 2.5 2.5 2 1

1: 3.5 4.5 2.5 4.5 0 5

2: 2.5 2.5 4 2.5 0 4

3: 5.5 2.5 5.5 3.5 4 8

4: 4 2.5 5.5 2.5 2 3

5: 3.5 5 3.5 4.5 1 6

6: 3.5 6.5 3.5 5 5 7

7: 4.5 6.5 3.5 6.5 6 9

8: 5.5 3.5 6 3.5 3 11

9: 4.5 7.5 4.5 6.5 7 10

10: 6 7.5 4.5 7.5 9 12

11: 6 3.5 7 3.5 8 13

12: 7 7.5 6 7.5 10 14

13: 7 3.5 7.5 3.5 11 20

14: 8 7.5 7 7.5 12 21

15: 6.5 10.5 6.5 8.5 17 16
16: 8 10.5 6.5 10.5 15 22
17: 6.5 8.5 8 8.5 15 23
18: 7.5 3 7.5 1.5 19 20
19: 7.5 1.5 9 1.5 18 25
20: 7.5 3.5 7.5 3 18 13
21: 8.5 7.5 8 7.5 14 29
22: 8.5 10.5 8 10.5 16 31
23: 8 8.5 8.5 8.5 17 30
24: 10.5 1.5 10.5 3 25 26
25: 9 1.5 10.5 1.5 19 24
26: 10.5 3 10.5 4 24 27
27: 10.5 4 10.5 5 26 28
28: 10.5 5 10.5 5.5 27 35
29: 8.5 8 8.5 7.5 21 30
30: 8.5 8.5 8.5 8 29 23
31: 8.5 11 8.5 10.5 22 32
32: 8.5 12.5 8.5 11 31 33
33: 9.5 12.5 8.5 12.5 32 38
34: 11.5 5.5 11.5 7 35 36
35: 10.5 5.5 11.5 5.5 28 34
36: 11.5 7 11.5 8 34 37
37: 11.5 8 11.5 8.5 36 41
38: 9.5 13 9.5 12.5 33 39
39: 9.5 14.5 9.5 13 38 40
40: 11 14.5 9.5 14.5 39 44
41: 11.5 8.5 12 8.5 37 46
42: 12.5 12.5 12.5 13 49 43
43: 12.5 13 12.5 14.5 42 44
44: 12.5 14.5 11 14.5 40 43
45: 13.5 8.5 13.5 10 46 47
46: 12 8.5 13.5 8.5 41 45

47: 13.5 10 13.5 11 45 48

48: 13.5 11 13.5 12.5 47 49

49: 13.5 12.5 12.5 12.5 42 48

Με χρήση των αναφορών στα γειτονικά ευθύγραμμα τμήματα, συνδέουμε όλα τα ευθύγραμμα τμήματα σε μια αλυσίδα. Στη συνέχεια, αφαιρούμε τις ενδιάμεσες συνευθειακές κορυφές που είναι περιττές και καταλήγουμε στα εξής 26 τελικά σημεία για το προκύπτον πολύγωνο:

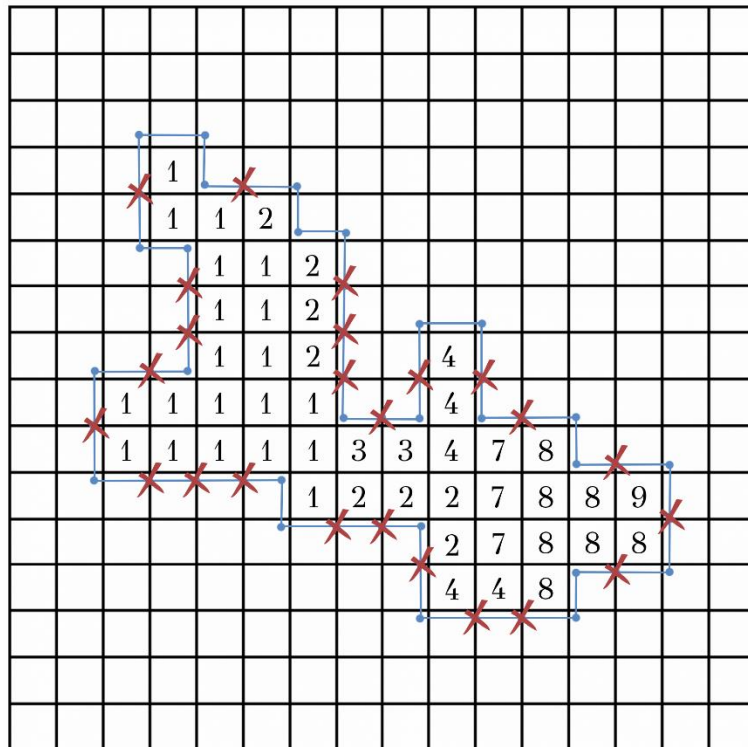
2.5 - 2.5
5.5 - 2.5
5.5 - 3.5
7.5 - 3.5
7.5 - 1.5
10.5 - 1.5
10.5 - 5.5
11.5 - 5.5
11.5 - 8.5
13.5 - 8.5
13.5 - 12.5
12.5 - 12.5
12.5 - 14.5
9.5 - 14.5
9.5 - 12.5
8.5 - 12.5
8.5 - 10.5
6.5 - 10.5
6.5 - 8.5
8.5 - 8.5
8.5 - 7.5
4.5 - 7.5
4.5 - 6.5

3.5 - 6.5

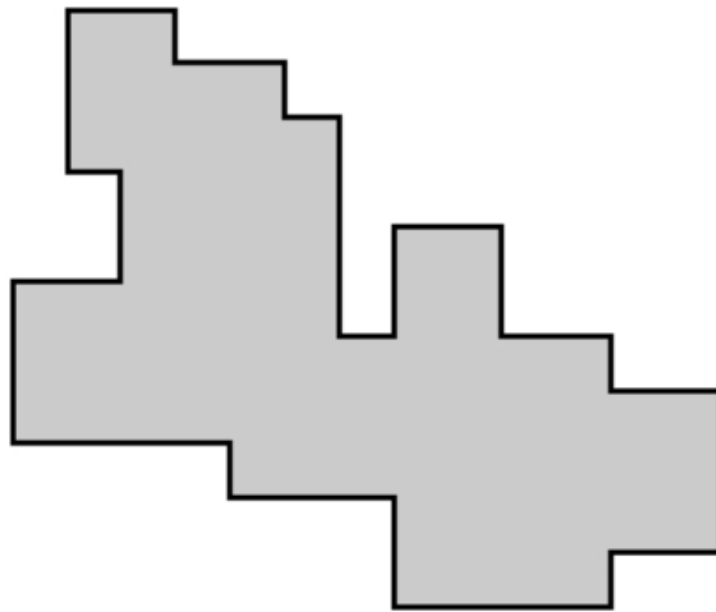
3.5 - 4.5

2.5 - 4.5

Στο Σχήμα 3.8 απεικονίζονται τα σημεία που αφαιρέθηκαν, ενώ στο Σχήμα 3.9 οπτικοποιείται το τελικό πολύγωνο που προκύπτει.



Σχήμα 3.8: Αφαίρεση περιττών σημείων.



Σχήμα 3.9: Οπτικοποίηση τελικού πολυγώνου.

3.3 Η Υλοποίηση

Για την υλοποίηση του αλγορίθμου κατασκευής ορθογωνίων παραλληλογράμμων χρησιμοποιήσαμε έναν δισδιάστατο πίνακα M που περιέχει ακέραιες τιμές. Οι τιμές αυτές είναι μεγαλύτερες του μηδενός αν το συγκεκριμένο κελί του πίνακα M βρίσκεται εντός του πολυγώνου και 0 σε αντίθετη περίπτωση.

```
int **M;
```

Η συνάρτηση `fill_array_M()` είναι υπεύθυνη για τη δημιουργία του πίνακα M ο οποίος και θα χρησιμοποιηθεί από τη συνάρτηση `construct_polygon()`. Στη συνάρτηση αυτή, ένας από τους ελέγχους που πραγματοποιείται είναι και ο εξής. Για κάθε τμήμα του πολυγώνου στη γραμμή $i+1$, εξασφαλίζουμε ότι θα συνεχίσει να υφίσταται στη γραμμή i ώστε όλα να συνδεθούν σε ένα πολύγωνο. Για τον σκοπό αυτό, χωρίζουμε τα αποθηκευμένα διαστήματα στηλών από τη γραμμή $i+1$ σε σύνολα C_1, C_2, \dots ανάλογα με την τιμή των στοιχείων του πίνακα M σε αυτά. Υπολογίζοντας το πλήθος των διαφορετικών τιμών, δημιουργούμε έναν πίνακα από `struct`

τέτοιου μεγέθους. Κάθε struct αποθηκεύει μία ακέραια τιμή και μία λίστα με τις θέσεις των διαστημάτων με την τιμή αυτή σε έναν βοηθητικό πίνακα.

```
struct intervalsInfo intervalPairs[count_different_values];
struct intervalsInfo{
    int value;
    list<int> interval_pairs;
};
```

Επιπλέον, για κάθε ζεύγος διαστημάτων του ίδιου τμήματος του πολυγώνου στη γραμμή $i+1$, εξασφαλίζουμε ότι στη γραμμή i θα υπάρχει κενό κελί ώστε να μη σχηματισθεί οπή. Στη λίστα outPolygon αποθηκεύουμε τους δείκτες στήλης για κάθε τρέχουσα γραμμή του πίνακα M που πρέπει να είναι εκτός πολυγώνου έτσι ώστε να μη σχηματιστεί οπή. Αντίστοιχα, στη λίστα inPolygon αποθηκεύουμε τους δείκτες στήλης για κάθε τρέχουσα γραμμή του πίνακα M που πρέπει να είναι εντός του πολυγώνου. Μέσω της λίστας αυτής εξασφαλίζουμε ότι τουλάχιστον ένα διάστημα με διαφορετική τιμή θα συνεχίσει να υφίσταται και στην επόμενη γραμμή.

```
list<int> outPolygon;
list<int> inPolygon;
```

Η συνάρτηση construct_polygon() υλοποιεί την κατασκευή του πολυγώνου από την αναπαράσταση του πολυγώνου, όπως αυτή προκύπτει από τη δημιουργία του πίνακα M . Στη συνάρτηση αυτή δημιουργούνται τα σημεία που θα αποτελέσουν τις κορυφές του τελικού ορθογωνίου πολυγώνου που θα σχηματιστεί.

Κατά το τελευταίο στάδιο του αλγορίθμου κρίνεται απαραίτητη η αποθήκευση όλων των ζευγών (x, y) σε μία δομή δεδομένων που θα επιτρέπει την άμεση προσθήκη ή διαγραφή ενός ζεύγους. Η διαγραφή αυτή γίνεται όταν υπάρχουν πάνω από δύο συνευθειακά σημεία στον ίδιο άξονα συντεταγμένων. Πιο συγκεκριμένα, αν το προηγούμενο και το επόμενο σημείο ενός σημείου p έχουν ίδια x - ή ίδια y -συντεταγμένη, τότε αφαιρούμε το σημείο p από τη δομή. Τα σημεία, που αποτελούν και τις κορυφές του ορθογωνίου πολυγώνου που προκύπτει, αποθηκεύονται σε vector από αντικείμενα (Points) που περιέχουν τις συντεταγμένες (x, y) κάθε σημείου του τελικού πολυγώνου.

```
vector<Points> p;  
Points::Points(float new_x_point, float new_y_point){  
    x_point = new_x_point;  
    y_point = new_y_point;  
}
```

ΚΕΦΑΛΑΙΟ 4

ΣΥΜΠΕΡΑΣΜΑΤΑ-ΕΠΕΚΤΑΣΕΙΣ

Σε αυτή τη μεταπτυχιακή εργασία πραγματευθήκαμε το πρόβλημα της ελάχιστης κάλυψης με s -αστέρες γενικών ορθογωνίων πολυγώνων χωρίς οπές. Ειδικότερα, μελετήσαμε και υλοποιήσαμε στη γλώσσα προγραμματισμού C++ τον αλγόριθμο των Motwani, Raghunathan και Saran για το παραπάνω πρόβλημα [3]. Ο εν λόγω αλγόριθμος έχει πολυπλοκότητα χρόνου $O(n^8)$ όπου n είναι το πλήθος κορυφών του δοθέντος ορθογωνίου πολυγώνου (χάρης στην πιο πρόσφατη εργασία των Eschen et al. [12], η πολυπλοκότητα χρόνου μειώνεται σε $O(n^6)$). Θα ήταν λοιπόν ενδιαφέρον να βρεθεί ταχύτερος αλγόριθμος. Στο πλαίσιο της παρούσας μεταπτυχιακής εργασίας δοκιμάσαμε κάποιες ιδέες χωρίς όμως να επιτύχουμε κάποιο από αποτέλεσμα. Έτσι, το πρόβλημα της επινόησης ταχύτερου αλγορίθμου για το παραπάνω πρόβλημα παραμένει ανοιχτό, ενώ επίσης ενδιαφέρον θα ήταν να βρεθεί ένας αμιγώς γεωμετρικός αλγόριθμος.

Επιπλέον, παρουσιάσαμε έναν αλγόριθμο για την κατασκευή “τυχαίων” ορθογωνίων πολυγώνων χωρίς οπές με βάση έναν συντελεστή “πυκνότητας” και θα ήταν ενδιαφέρουσα η εκτενέστερη πειραματική μελέτη των αποτελεσμάτων του.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] J. Culberson and R. A. Reckhow, ‘Orthogonally Convex Coverings of Orthogonal Polygons without Holes’, *Journal of Computer and System Sciences*, vol. 39, pp. 166–204, Department of Computer Science, University of Alberta, 1989.
- [2] E. J. van Leeuwen and J. van Leeuwen, ‘Convex Polygon Intersection Graphs’, Technical Report, Department of Information and Computing Sciences, Utrecht University, 2010.
- [3] R. Motwani, A. Raghunathan and H. Saran, ‘Covering Orthogonal Polygons with Star Polygons: The Perfect Graph Approach’, in *Journal of Computer and System Sciences*, 1990, pp. 19-48.
- [4] W. J. Masek, ‘Some NP-complete set covering problems, unpublished manuscript’, MIT, Aug. 1979 (referenced in M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, p. 232, 1979).
- [5] D. S. Franzblau and D. J. Kleitman, ‘An algorithm for constructing regions with rectangles: Independence and minimum generating sets for collections of intervals’, in ‘Proceedings, Sixteenth Annual ACM Symposium on Theory of Computing, April 1984’, pp. 167-174.
- [6] J. M. Keil, ‘Minimally covering a horizontally convex orthogonal polygon’, in ‘Proceedings, Second Annual Symposium on Computational Geometry, June 1986’, pp. 43-51.

- [7] J. Culberson and R. Reckhow, ‘Dent Digrams: A Unified Approach to Polygon Covering Problems.’ Tech. Rep. TR 87-14, Dept. of Computing Science, University of Alberta, July 1987.
- [8] J. Spinrad and R. Sritharan, ‘Algorithms for weakly triangulated graphs’ Dept. of Computing Science, Vanderbilt University, Nashville, TN 37235, USA, 1995, pp. 181-191.
- [9] R. Motwani, A. Raghunathan and H. Saran, ‘Perfect graphs and orthogonally convex covers’, *SIAM J. Discrete Math.* 14, 1989.
- [10] R. B. Hayward, ‘Weakly triangulated graphs’, *J. Combin. Theory Ser. B* 39, 1985, pp. 200-209.
- [11] R. Hayward, C. Hoang and F. Maffray, ‘Optimizing weakly triangulated graphs’, *Graphs Combin.* 5 (1989) 339-349; 6 (1990) 33-35.
- [12] E.M. Eschen, C.T. Hoang, M.D.T. Petrick, and R. Sritharan, ‘Disjoint clique cutsets in graphs without long holes’, *J. Graph Theory* 48(4) (2005), pp. 277-298.
- [13] T. Auer and M. Held, ‘Heuristics for the generation of random polygons’, in F. Fiala, E. Kranakis, J.-R. Sack (eds), *Proc. 8th Canadian Conference on Computational Geometry (CCCG’96)*, Carleton University Press (1996), pp. 38-44.
- [14] G. Zhu, G. Sundaram, J. Snoeyink, and J. Mitchell, ‘Generating random polygons with given vertices’, *Comp. Geom. Theory and Appl.* 6(5) (1996), 277-290.
- [15] C. Sohler, ‘Generating random star-shaped polygons’, in *Proc. 11th Canadian Conference on Computational Geometry (CCCG’99)* (1999), pp. 174-177.
- [16] J. O’Rourke, I. Pashchenko, and G. Tewari, ‘Partitioning orthogonal polygons into fat rectangles’, in *Proc. 13th Canadian Conference on Computational Geometry (CCCG’01)* (2001), pp. 133-136.
- [17] A.P. Tomas and A.L. Bajuelos, ‘Generating random Orthogonal polygons’, in R. Cornejo et al. (eds) *CAEPIA-TTIA 2003*, LNAI 3040, pp. 364-373, 2004.

ΣΥΝΤΟΜΟ ΒΙΟΓΡΑΦΙΚΟ

Όνοματεπώνυμο: Ειρήνη Γορδίου

Τηλέφωνο επικοινωνίας: 6976481382

Ημερομηνία γέννησης: 10/01/1993

Email: eirinigord@gmail.com

ΕΚΠΑΙΔΕΥΣΗ

- Μεταπτυχιακό δίπλωμα στην Πληροφορική με εξειδίκευση στη Θεωρία Επιστήμης των Υπολογιστών του τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Ιωαννίνων (2021)
- Απόφοιτη Πληροφορικής του τμήματος Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Ιωαννίνων (2017)
- Απολυτήριο λυκείου από το 1^ο Ενιαίο Λύκειο Άνδρου (2011)

ΞΕΝΕΣ ΓΛΩΣΣΕΣ

- Γαλλικά - Diplome d' etudes en langue francaise, institute francais d' Athenes, Delf B2 (2010)
- Κρατικό Πιστοποιητικό Γλωσσομάθειας στην αγγλική γλώσσα επιπέδου Γ1 (2009)
- University of Cambridge ESOL Examinations, Entry 3, Level B2 (2006)

ΕΡΓΑΣΙΑΚΗ ΠΡΟΫΠΗΡΕΣΙΑ

- Full Stack Web Designer/Developer και Digital Marketing Expert στην εταιρεία Wapp Development House στα Ιωάννινα (2018-σήμερα)