



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**  
**ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Web εφαρμογή για την εύρεση βέλτιστων σημείων τοποθέτησης  
μετρητών καταγραφής της ποιότητας του αέρα**

**ΦΟΙΤΗΤΗΣ**

**ΖΟΥΛΟΥΜΗΣ ΧΑΡΑΛΑΜΠΟΣ**

**ΑΜ 2006**

**ΕΠΙΒΛΕΠΟΝ ΚΑΘΗΓΗΤΗΣ**

**Δρ. ΣΑΧΑΡΙΔΗΣ ΓΕΩΡΓΙΟΣ**

**ΚΑΘΗΓΗΤΗΣ ΣΤΟ ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟΥ  
ΘΕΣΣΑΛΙΑΣ**

**ΒΟΛΟΣ , 5 ΟΚΤΩΒΡΙΟΥ 2022**

---

## **Πρόλογος**

Η ρύπανση του περιβάλλοντος είναι ένα μείζον ζήτημα το οποίο επηρεάζει τόσο τον άνθρωπο όσο και την φύση. Η κατηγορία με την μεγαλύτερη σημασία είναι η αέρια ρύπανση η οποία έχει άμεσο αντίκτυπο. Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας (World Health Organisation) το 99% του παγκόσμιου πληθυσμού ζούσε σε περιοχές που δεν πληρούσαν τα αποδεκτά όρια για την ποιότητα του εξωτερικού αέρα, το οποίο συνεπάγεται αύξηση των ασθενειών καθώς και αλλαγή του κλίματος. Επιπλέον, εκτός από τις άμεσες, υπάρχουν και οι έμμεσες συνέπειες όπως η οικονομική επιβάρυνση αλλά και οι ψυχολογικές επιπτώσεις. Για του λόγους αυτούς, η μελέτη της αέριας ρύπανσης έχει μεγάλη σημασία με απώτερο σκοπό την ελαχιστοποίηση της.

Για την αποτελεσματικότητα της μελέτης αέριων ρύπων απαιτείται η ακριβής μελέτη αυτών σε ένα εύρος χρόνου. Η καταμέτρηση αυτή μπορεί να επιτευχθεί με την χρήση ειδικών μετρητών που βρίσκονται διασκορπισμένοι μέσα σε μία περιοχή. Ζήτημα προς διευρέυνηση αποτελεί η επίδραση των σημείων τοποθέτησης ως προς την αξιοπιστία των μετρήσεων.

Για την τοποθέτηση των μετρητών στα βέλτιστα σημεία μεγάλη σημασία έχει συλλογή όλων των δεδομένων που επηρεάζουν την συγκέντρωση των αέριων ρύπων σε κάθε περιοχή. Τα δεδομένα μπορεί να είναι μετεωρολογικά ή να σχετίζονται με τα είδη των ρύπων στην περιοχή (βιομηχανικοί, αστικοί, οικιακοί). Η εύρεση των βέλτιστων σημείων κατασκευής γίνεται συχνά μέσο του γραμμικού προγραμματισμού. Λόγω όμως της πληθώρας των παραμέτρων, της μοναδικότητας καθώς και της μεγάλης έκτασης της περιοχής που χρειάζεται να γίνει η μελέτη δημιουργείται η ανάγκη της διευκόλυνσης της συλλογής δεδομένων.

Η απλοποίηση αυτή μπορεί να επιτευχθεί με την αυτοματοποίηση της συλλογής των δεδομένων μέσω της δημιουργίας προγραμμάτων που έχουν ως στόχο να βοηθήσουν τον χρήστη ελαχιστοποιώντας τον χρόνο που απαιτείται από την έναρξη της συγκέντρωσης των δεδομένων μέχρι και την εύρεση των βέλτιστων σημείων. Στη παρούσα εργασία παρουσιάζεται η ανάπτυξη ενός λογισμικού, με στόχο την επίλυση του προβλήματος αυτού.

---

## Περιεχόμενα

Πρόλογος.....	2
Ενότητα 1: Εισαγωγή.....	5
Ενότητα 2: Βιβλιογραφική ανασκόπηση.....	5
2.1 Ατμοσφαιρική ρύπανση.....	5
2.1.1 Ορισμός.....	5
2.1.2 Οξείδια του αζώτου (NOx).....	6
2.1.3 Οξείδια του θείου (SOx).....	6
2.1.4 Πτητικοί Υδρογονάνθρακες.....	6
2.1.5 Οξείδια του άνθρακα(COx) .....	6
2.1.6 Αιωρούμενα Σωματίδια.....	7
2.2 Παράγοντες ατμοσφαιρικής ρύπανσης.....	7
2.2.1 Εισαγωγή.....	7
2.2.2 Άνεμος.....	8
2.2.3 Πολεοδομική μορφολογία και πηγές ρύπανσης.....	8
2.3 Όρια συγκέντρωσης ρύπων.....	8
2.4 Δίκτυο παρακολούθησης αέριας ρύπανσης.....	10
2.4.1 Μέθοδοι καταγραφής συγκεντρώσεων.....	10
2.4.2 Στόχοι δικτύου παρακολούθησης ποιότητας αέρα.....	10
2.4.3 Χωροθέτησης μετρητών αιωρούμενων σωματιδίων.....	10
Ενότητα 3: Μεθοδολογία.....	11
3.1 Αρχικά στοιχεία.....	11
3.1.1 Εισαγωγή.....	11
3.1.2 Δείκτες Μεταβλητών.....	12
3.1.3 Μεταβλητές της μεθοδολογίας.....	12
3.2 Προσδιορισμός υποψήφιων σημείων.....	12
3.3 Επίδραση ανέμων.....	12
3.4 Κριτήρια ρύπανσης.....	14
3.4.1 Μέγεθος δρόμου.....	14
3.4.2 Ύψος κτιρίων.....	14

---

3.4.3 Όγκος κυκλοφορίας.....	15
3.4.4 Οικιακή θέρμανση.....	15
3.4.5 Πηγές ρύπανσης.....	16
3.5 Κατηγοριοποίηση σημείων.....	17
3.6 Ευρεση βέλτιστης λύσης.....	18
<b>Ενότητα 4: Αποτελέσματα εφαρμογής.....</b>	<b>19</b>
4.1 Τοποθεσία εφαρμογής του προγράμματος.....	19
4.2 Ορισμός σημείων.....	19
4.3 Κατηγοριοποίηση σημείων.....	22
4.4 Αποτελέσματα βελτιστοποίησης.....	24
<b>Συμπεράσματα.....</b>	<b>43</b>
<b>Παράρτημα.....</b>	<b>44</b>
Εγκατάσταση και εκτέλεση εφαρμογής.....	44
Κώδικες προγραμμάτων.....	49
<b>Βιβλιογραφία.....</b>	<b>141</b>

---

## **Ενότητα 1: Εισαγωγή**

Σκοπός της παρούσας διπλωματικής εργασίας είναι η κατασκευή εφαρμογής ιστού (web application) με σκοπό τον προσδιορισμό των βέλτιστων σημείων για την κατασκευή μετρητών αέριων ρύπων μέσω μαθηματικού μοντέλου μεικτού αέριου προγραμματισμού.

Αρχικά θα γίνει αναφορά στα είδη των αέριων ρύπων και στην επίδραση που έχουν στο περιβάλλον και στον άνθρωπο. Επιπλέον, θα γίνει παράθεση των χαρακτηριστικών μιας περιοχής που μπορούν να συμβάλλουν στη συγκέντρωση ή την απομάκρυνση των αέριων ρύπων. Το πρώτο μέρος ολοκληρώνεται με την παρουσίαση των μετρητών αέριων ρύπων, των χαρακτηριστικών καθώς και της σημασίας τους.

Το δεύτερο μέρος αφορά το μαθηματικό μοντέλο το οποίο θα εφαρμοστεί για την εύρεση της βέλτιστης λύσης. Αυτό συνεπάγεται αναφορά στους δείκτες καθώς και τις μεταβλητές του προβλήματος. Στη συνέχεια γίνεται παρουσίαση των χαρακτηριστικών που συμβάλλουν στην αέρια ρύπανση μίας περιοχής. Τέλος, γίνεται ο ορισμός των μεθόδων κατηγοριοποιήσεων των σημείων σε ομάδες βάση των συντελεστών βαρύτητας, της αντικειμενικής συνάρτησης και επιπλέον κριτηρίων μέσω των οποίων θα επιτευχθεί η βέλτιστη λύση του προβλήματος.

Επιπροσθέτως, γίνεται εφαρμογή του προγράμματος για την πόλη του Βόλου, γίνεται η παρουσίαση των πηγών ρύπανσης και οι συνθήκες που επικρατούν. Διατυπώνεται η περιοχή και τα πιθανά σημεία σημεία κατασκευής των μετρητών. Ακολουθεί η κατηγοριοποίηση των σημείων για δύο διαφορετικές μεθόδους, για διαφορετικό αριθμό μετρητών. Στη συνέχεια παρουσιάζονται τα αποτελέσματα που προκύπτουν από την βελτιστοποίηση και τα συμπεράσματα για τις κατηγοριοποιήσεις αυτές. Μέσω των στοιχείων αυτών διακρίνονται οι βέλτιστες λύσεις. Η διπλωματική ολοκληρώνεται με έναν οδηγό εγκατάστασης των προγραμμάτων, εκτέλεσης της εφαρμογής και του κώδικα που αναπτύχθηκε για την δημιουργία την εφαρμογής και εκτέλεσης του μαθηματικού μοντέλου.

Τέλος στο παράρτημα παρουσιάζεται ο οδηγός χρήσης των προγραμμάτων που θα χρησιμοποιηθούν. Πρώτο βήμα είναι η εγκατάσταση των προγραμμάτων (Netbeans , C++) και στη συνέχεια παρουσιάζεται ο οδηγός για την εγκατάσταση και την χρήση του κάθε προγράμματος.

## **Ενότητα 2: Βιβλιογραφική ανασκόπηση**

### **2.1 Ατμοσφαιρική ρύπανση<sup>[1],[28],[30]</sup>**

#### **2.1.1 Ορισμός**

Ατμοσφαιρική ρύπανση είναι η παρουσία ρύπων στην ατμόσφαιρα σε ποσότητα τέτοια ώστε να προκαλέσει αρνητικές συνέπειες στην ανθρώπινη υγεία και στο οικοσύστημα. Οι πηγές ρύπανσης μπορεί είναι να είτε ανθρωπογενείς είτε φυσικές. Σε οποιαδήποτε από τις δύο περιπτώσεις οι αέριοι ρύποι μπορεί να σχηματίζονται απευθείας από την πηγή (πρωτογενείς ρύποι) ή να σχηματίζονται μέσω χημικών αντιδράσεων με την ατμόσφαιρα (δευτερογενείς ρύποι). Στη συνέχεια θα γίνει αναφορά στους κυριότερους ρύπους.

### 2.1.2 Οξείδια του αζώτου (NO<sub>x</sub>)<sup>[2]</sup>

Σύμφωνα με μελέτες οι δύο μεγαλύτερες πηγές εκπομπών οξειδίων του αζώτου είναι η οι βιομηχανίες και τα οχήματα. Το άζωτο περιέχεται στην ατμόσφαιρα σε ποσοστό 78% . Τα οξείδια του σχηματίζονται μέσω χημικής αντίδρασης η οποία συμβαίνει σε υψηλές θερμοκρασίες που παρατηρούνται κατά την διάρκεια της καύσης κυρίως ορυκτών καυσίμων σε κινητήρες οχημάτων ή εργοστάσια.

Η παρουσία των οξειδίων του αζώτου στην ατμόσφαιρα έχει ως αποτέλεσμα την μείωση της φωτεινότητας και την δημιουργία της φωτοχημικής αιθαλομίχλης. Σε συνθήκες βροχής αντιδρώντας με το νερό δημιουργούν νιτρικό οξύ (όξινη βροχή) και μεταβάλλουν το pH του εδάφους εμποδίζοντας την ανάπτυξη των καλλιεργειών. Τέλος, τα οξείδια του αζώτου μπορούν να δημιουργήσουν αναπνευστικά προβλήματα καθώς μέσω της υψηλής υγρασίας σχηματίζονται ατμοί νιτρικού οξέος που εισχωρούν στους πνεύμονες δημιουργώντας ακόμα και μόνιμη βλάβη στους πνεύμονες σε υψηλή περιεκτικότητα που στη συνέχεια μπορούν να επεκταθούν σε καρδιακά προβλήματα.

### 2.1.3 Οξείδια του θείου (SO<sub>x</sub>)<sup>[3],[4],[24]</sup>

Το κυριότερο οξείδιο του θείου είναι το διοξείδιο (SO<sub>2</sub>). Οι κυριότερες πηγές του είναι τα διυλιστήρια πετρελαίου, η κεντρική θέρμανση και τα πλοία, γεγονός που εντάσσει στις πηγές και τα λιμάνια αστικών περιοχών. Στην ατμόσφαιρα το διοξείδιο του αντιδρά με την ατμόσφαιρα και σχηματίζει τριοξείδιο του θείου (SO<sub>3</sub>) που σε συνδιασμό με τους υδρατμούς σχηματίζει ομίχλη θειικού οξέος. Η παρουσία τους στην ατμόσφαιρα μπορεί να δημιουργήσει αναπνευστικά και καρδιαγγειακά προβλήματα καθώς και γαστρικό καρκίνο(ο τρίτος πιο συχνός θάνατος από καρκίνο).

### 2.1.4 Πτητικοί Υδρογονάνθρακες<sup>[5],[25]</sup>

Οι ενώσεις χαρακτηρίζονται ως πτητικές και είναι μία μεγάλη ομάδα το υγρών όσο και αερίων. Οι κύριες πηγές τους είναι οι βαφές και οι επικαλύψεις, το βενζόλιο το οποίο βρίσκεται στο καπνό των τσιγάρων καθώς και σε καύσιμα και στα καυσάερια οχημάτων. Επιπλέον, σε αυτή την κατηγορία εντάσσονται και το πετρέλαιο και το χλωροαιθυλένιο που προκαλούν καρκινικές παθήσεις. Σε αυτή την κατηγορία υπάρχουν και ενώσεις οι οποίες μπορούν να αντιδράσουν με τα οξείδια του αζώτου ή το όζον δημιουργώντας νέα προϊόντα.

Λόγω της μεγάλης ποικιλίας στοιχείων οι επιπτώσεις που έχουν στην ανθρώπινη υγεία περιλαμβάνουν αναπνευστικά προβλήματα, πονοκεφάλους ναυτίες, βλάβες στο συκώτι και στα νεφρά ακόμα και προβλήματα στο νευρικό σύστημα.

### 2.1.5 Οξείδια του άνθρακα (CO<sub>x</sub>)<sup>[6],[7],[26]</sup>

Το μονοξείδιο του άνθρακα περιέχεται στην ατμόσφαιρα σε μικρές συγκεντρώσεις αλλά παράγεται σε μεγάλο βαθμό από την μεταφορά οχημάτων και κάθε μορφής μηχανήματα που λειτουργεί με κινητήρα εσωτερικής και εξωτερικής καύσης. Το συγκεκριμένο αέριο είναι άοσμο και δηλητηριώδες. Είναι σημαντικό να αναφερθεί ότι η παραγωγή του αυξάνεται τον χειμώνα όπου οι καύσεις των μηχανών είναι ατελέστερες. Συμβάλλει στο φαινόμενο εγκλεισμού της ακτινοβολίας και η εκτεταμένη έκθεση του

ανθρώπου στο μονοξείδιο του άνθρακα μπορεί να προκαλέσει προβλήματα στη όραση απώλεια ισορροπίας, κόπωση και προβλήματα στο νευρικό σύστημα. Επίσης το διοξείδιο του άνθρακα το οποίο παράγεται μέσω της καύσης ορυκτών καυσίμων και υπάρχει στην ατμόσφαιρα. Είναι αέριο του θερμοκηπίου και συνπώς συμβάλλει στην θέρμανση του πλανήτη κα σε μεγάλες περιεκτικότητες μπορεί να προκαλέσει κεφαλαλγία, δύσπνοια αλλά και παραισθήσεις.

### **2.1.6 Αιωρούμενα Σωματίδια<sup>[8],[9],[10],[11],[27]</sup>**

Τα αιωρούμενα σωματίδια (Particle Matter ,PM) είναι υγρά ή στερεά σωματίδια τα οποία μπορούν να αιωρούνται στην ατμόσφαιρα λόγω των μικρών τους διαστάσεων. Το εύρος διάστασης των σωματιδίων αυτών είναι από μερικά nm έως κάποιες δεκάδες μm. Τα αιωρούμενα σωματίδια μπορούν να χαρακτηριστούν σε εισπνεύσιμα για PM<100 μm τα οποία μπορούν να εισέρχονται μέχρι τον λάρυγγα, το θωρακικό κλάσμα για PM<10 μm που διασχίζουν το λάρυγγα και τους πνεύμονες και τέλος τα αναπνεύσιμα για PM<4 μm τα οποία φτάνουν μέχρι την περιοχή που γίνεται η ανταλλαγή αερίων. Τα αιωρούμενα σωματίδια χωρίζονται σε άλλες δύο βασικές κατηγορίες. Σωματίδια με διάμετρο μικρότερη απο 2,5 μm ονομάζονται PM2.5 ή λεπτόκοκκα, ενώ για διάμετρο μικρότερη από 10 μm ονομάζονται PM10 ή χοντρόκοκκα. Με βάση τον σχηματισμό τους τα σωματίδια χωρίζονται σε αερολύματα συμπύκνωσης τα οποία σχηματίζονται από την συμπύκνωση των ατμών ή από αντιδράσεις αερίων και στα αερολύματα διασποράς που σχηματίζονται από θραύση στερεών, εκνέφωση υγρών ή διασποράς σκόνης.

Τα αιωρούμενα σωματίδια μπορούν να εμφανιστούν φυσικά ή μέσω ανθρώπινων ενεργειών. Οι φυσικές πηγές των σωματιδίων μπορεί να είναι ηφαίστεια, καταιγίδες σκόνης, η επιφάνεια του ωκεανού αλλά και οι πυρκαγές δασών και γενικότερα της βλάστησης, με πρόσφατο φαινόμενο την πυρκαγιά της Εύβοιας τον Αύγουστο του 2021. Εκτός από τις φυσικές υπάρχουν και οι ανθρωπογενείς πηγές όπως η καύση ορυκτών καυσίμων καθώς και από την φθορά των δρόμων.

Η εισπνοή των αιωρούμενων σωματιδίων μπορεί να έχει επιβλαβείς συνέπειες τόσο στον άνθρωπο όσο και στα ζώα όπως καρκίνο του πνεύμονα ,άσθμα, καρδιακές παθήσεις ,αναπτυξιακές διαταραχές αλλά και ψυχικά προβλήματα. Αποτέλεσμα των οποίων είναι η αύξηση των θανάτων παγκοσμίως και σύμφωνα με έρευνα του Παγκόσμιου Οργανισμού Υγείας εκτιμήθηκε ότι «η ατμοσφαιρική ρύπανση από λεπτά σωματίδια (PM2.5), προκαλεί περίπου το 3% της θνησιμότητας από καρδιοπνευμονική νόσο, περίπου το 5% της θνησιμότητας από καρκίνο της τραχείας, του βρόγχου και πνεύμονα και περίπου το 1% της θνησιμότητας από οξείες αναπνευστικές λοιμώξεις σε παιδιά κάτω των 5 ετών, παγκοσμίως».

## **2.2 Παράγοντες ατμοσφαιρικής ρύπανσης**

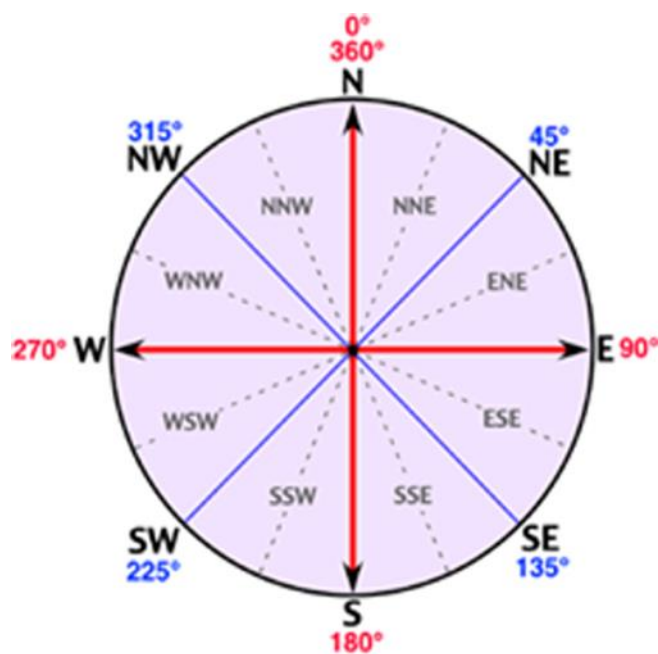
### **2.2.1 Εισαγωγή**

Η ατμοσφαιρική ρύπανση προκαλείται σε μεγάλο βαθμό από τον ανθρώπινο παράγοντα.Το μεγαλύτερο μέρος σχετίζεται με την καύση καυσίμων είτε μέσω των βιομηχανιών είτε μέσω των κινητών πηγών όπως τα αυτοκίνητα, τα πλοία, τα αεροπλάνα ή τα τρένα. Εκτός όμως από της πηγές των εκπομπών των αερίων σημαντική είναι και η επίδραση του κλίματος καθώς και της πολεοδομικής μορφολογίας της περιοχής που μπορούν να συμβάλλουν στην συγκέντρωση ή την αραιώση των ρύπων.

## 2.2.2 Άνεμος<sup>[12]</sup>

Η κατεύθυνση του ανέμου και η ταχύτητα του μπορεί να μεταφέρει τους αέριους ρύπους από την πηγή παραγωγής και να τους διασκορπίσει. Οι μεγαλύτερες ταχύτητες ανέμου έχουν ως αποτέλεσμα την διασπορά των αέριων ρύπων σε μεγαλύτερο εύρος μειώνοντας έτσι την συγκέντρωσή τους στην περιοχή. Τέλος η κατεύθυνση του ανέμου μπορεί να επηρεάσει σε μεγάλο βαθμό την ρύπανση μίας περιοχής καθώς μπορεί να διανέμει τους ρύπους σε μία διαφορετική περιοχή μεταβάλλοντας τα επίπεδα των ρύπων κάνοντας σημαντική την μελέτη των ανέμων σε μια περιοχή. Για τους λόγους αυτούς θα κατασκευαστεί μία κατηγορία για την συσχέτιση των ανέμων με τους άλλους παράγοντες του μαθηματικού μοντέλου.

Σχήμα 1 :Κατευθύνσεις ανέμου



## 2.2.3 Πολεοδομική μορφολογία και πηγές ρύπανσης (πλάτος δρόμων, ύψος κτιρίων , βιομηχανικές περιοχές και λιμάνια)<sup>[13],[14],[29]</sup>

Η αστική μορφολογία επηρεάζει το τοπικό κλίμα μίας περιοχής. Το ύψος των κτηρίων, το πλάτος των δρόμων και κατ'επέκταση η κυκλοφοριακή συμφόρηση και η ύπαρξη ανοιχτού χώρου επιδρούν στο μικροκλίμα της περιοχής. Επιπλέον, η ύπαρξη πηγών αέριων ρύπων όπως βιομηχανίες ή λιμάνια στην περιοχή ή κοντά στην περιοχή είναι ακόμα μια σημαντική παράμετρος για την μελέτη των αέριων ρύπων. Λόγω της άμεσης εξάρτησής τους με την αέρια ρύπανση είναι απαραίτητη η δημιουργία κατηγοριών για τις παραμέτρους αυτές.

## 2.3 Όρια συγκέντρωσης ρύπων<sup>[23]</sup>

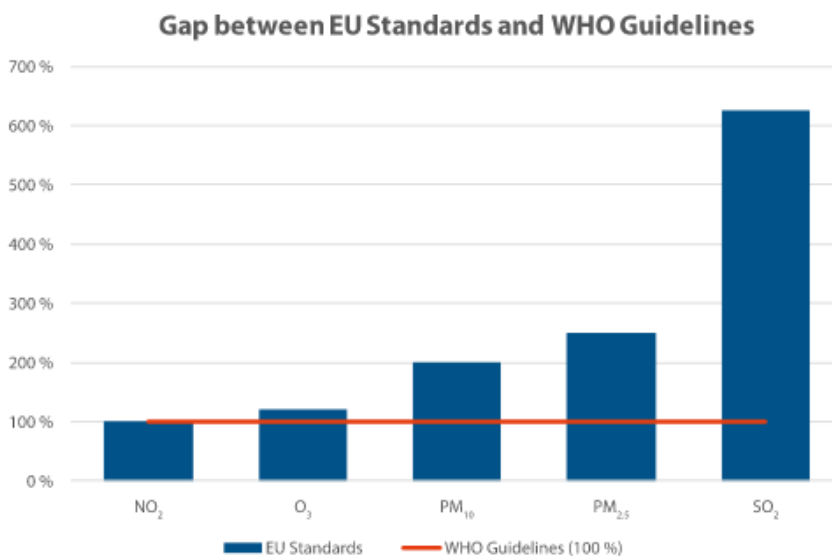
Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας, η ατμοσφαιρική ρύπανση είναι ο μεγαλύτερος περιβαλλοντικός κίνδυνος. Τα αέρια ρύπανσης όπως το όζον, το διοξείδιο του αζώτου καθώς και τα αιωρούμενα σωματίδια προκαλούν χιλιάδες θανάτους κάθε χρόνο. Για τον λόγο αυτό η οδηγία της ποιότητας του αέρα του 2008 είναι ο ακρογωνιαίος λίθος της πολιτικής της ΕΕ για καθαρό αέρα, καθώς



ορίζει τα όρια για την ποιότητα του αέρα και κατ'επέκταση τα όρια της συγκέντρωσης των ρύπων. Παρακάτω παρουσιάζονται τα πρότυπα ποιότητας του αέρα, οι κατευθυντήριες γραμμές του Παγκόσμιου Οργανισμού Υγείας και γίνεται σύγκριση των δύο ορίων.

**Πίνακας 1 : Πρότυπα της ΕΕ για την ποιότητα του αέρα και κατευθυντήριες γραμμές του ΠΟΥ**

Pollutant	Period	WHO Guidelines $\mu\text{g}/\text{m}^3$	EU AAQ Directive limit values $\mu\text{g}/\text{m}^3$	No. of times in a year that the EU standards can be exceeded
NO <sub>2</sub>	1 year	40	40	-
	1 hour	200	200	18
O <sub>3</sub>	8 hours	100	120	25
PM <sub>10</sub>	1 year	20	40	-
	24 hours	50 <sup>(a)</sup>	50	35
PM <sub>2.5</sub>	1 year	10	25	-
	24 hours	25	-	-
SO <sub>2</sub>	24 hours	20	125	3
	1 hour	-	350	24
	10 minutes	500	-	-



Sources: WHO Air quality guidelines (2005) and AAQ Directive 2008/50/EC.

**Γράφημα 1 : Απόκλιση πρότυπων της ΕΕ και κατευθυντήριων γραμμών του ΠΟΥ**

## 2.4 Δίκτυο παρακολούθησης αέριας ρύπανσης

### 2.4.1 Μέθοδοι καταγραφής συγκεντρώσεων<sup>[16]</sup>

Ένα δίκτυο μετρητών αέριων ρύπων έχει ως στόχο την καταγραφή των επιπέδων των ρύπων με σκοπό την αξιολόγηση της ποιότητας του αέρα βάσει των ορίων που έχουν οριστεί. Υπάρχουν δύο κατηγορίες μεθόδων για τον καθορισμό της συγκέντρωσης αιωρούμενων σωματιδίων. Η ενεργητική μέθοδος έχει μεγαλύτερη ακρίβεια στις μετρήσεις της συγκέντρωσης σε πραγματικό χρόνο και αυτό επιτυγχάνεται με την αναρρόφηση ποσότητας αέρα και τον υπολογισμό των αέριων σωματιδίων στην ποσότητα αυτή. Οι ενεργητικές μέθοδοι που έχουν αναγνωρισθεί από επίσημους οργανισμούς European Environmental Agency (Leeuw, 2005) και Environmental Protection Agency (NARSTO, 2006) είναι η μέθοδος απορρόφησης ακτινοβολίας β, η μέθοδος δονούμενου στοιχείου και η μέθοδος σκεδασμού φωτός. Χαρακτηρίζονται από υψηλό κόστος κατασκευής και λειτουργίας λόγω της ανάγκης συνεχούς τους λειτουργίας για την συλλογή δεδομένων σε πραγματικό χρόνο. Σε αντίθεση οι παθητικές μέθοδοι συλλέγουν τα αιωρούμενα σωματίδια σε ειδικές επιφάνειες χωρίς αναρρόφηση και υπολογίζουν τη μέση συγκέντρωση για το διάστημα λειτουργίας τους. Επιπλέον, δεν απαιτείται η συνεχής λειτουργία των μετρητών μειώνοντας το κόστος τους αλλά μειώνοντας και την ακρίβεια των αποτελεσμάτων.

### 2.4.2 Στόχοι δικτύου παρακολούθησης ποιότητας αέρα<sup>[16]</sup>

Ένα δίκτυο παρακολούθησης της ποιότητας του αέρα μπορεί, συγκεντρωτικά, να εξυπηρετήσει έναν ή περισσότερους από τους παρακάτω σκοπούς.

1. Παρακολούθηση για τη συμμόρφωση στην υφιστάμενη νομοθεσία η οποία καθορίζει τα όρια ποιότητας του αέρα,
2. Παρακολούθηση για την αντιπροσωπευτική αποτίμηση της ποιότητας του αέρα σε μία περιοχή,
3. Παρακολούθηση της ποιότητας του αέρα για την εκτίμηση της έκθεσης του πληθυσμού στην ατμοσφαιρική ρύπανση,
4. Παρακολούθηση της ατμοσφαιρικής ρύπανσης κοντά σε μεγάλες πηγές εκπομπών ώστε να αποφεύγεται η μεγάλη επιβάρυνση γειτονικών περιοχών,
5. Παρακολούθηση για την καταγραφή/πρόγνωση επεισοδίων ρύπανσης, ώστε να υπάρξει προειδοποίηση στον πληθυσμό, και την λήψη εκτάκτων, βραχυπρόθεσμων μέτρων για τον περιορισμό των υψηλών συγκεντρώσεων,
6. Παρακολούθηση της ατμοσφαιρικής ρύπανσης για την υποστήριξη ερευνητικών προσπαθειών.

### 2.4.3 Χωροθέτηση μετρητών αιωρούμενων σωματιδίων<sup>[16]</sup>

Τα κριτήρια για την ανάπτυξη του δικτύου όπως έχουν οριστεί από το Τεχνικό Επιμελητήριο Ελλάδας (ΤΕΕ) περιλαμβάνουν τα παρακάτω:

1. Επιλογή των περιοχών στις οποίες θα παρακολουθούνται τα επίπεδα σωματιδιακής ρύπανσης. Ο σχεδιασμός ενός δικτύου παρακολούθησης της ποιότητας του αέρα θα πρέπει να λαμβάνει υπόψιν τις εκπομπές των ρύπων στην περιοχή ενδιαφέροντος καθώς και τα επίπεδα ατμοσφαιρικής ρύπανσης, όπως αυτά καταγράφονται από υφιστάμενους σταθμούς παρακολούθησης της ποιότητας του αέρα και από αποτελέσματα προσομοιώσεων με τη βοήθεια αριθμητικών μοντέλων. Ακόμη, σημαντικό κριτήριο για τη χωροθέτηση σταθμών παρακολούθησης της ποιότητας του αέρα είναι και η έκθεση των ανθρώπων στην ατμοσφαιρική ρύπανση.

2. Κατηγοριοποίηση των περιοχών παρακολούθησης. Προκειμένου να διευκολυνθεί η ανταλλαγή πληροφοριών εφαρμόζεται ένα σύστημα ταξινόμησης των σταθμών ποιότητας του αέρα.
3. Περιοχή αντιπροσωπευτικότητας των σταθμών. Η περιοχή μέσα στην οποία δεν εμφανίζονται σημαντικές διαφορές στα επίπεδα ρύπανσης ονομάζεται περιοχή αντιπροσωπευτικότητας του σταθμού. Ο προσδιορισμός της αντιπροσωπευτικότητας του σταθμού είναι ιδιαίτερα χρήσιμος στην περίπτωση που τα δεδομένα χρησιμοποιούνται για την εκτίμηση της έκθεσης του πληθυσμού καθώς και στην περίπτωση που τα δεδομένα του σταθμού χρησιμοποιούνται για την επαλήθευση μοντέλων διασποράς.
4. Έλεγχος/διασφάλιση ποιότητας των μετρήσεων. Προκειμένου να σχεδιασθεί ένα σύστημα ελέγχου/διασφάλισης της ποιότητας των μετρήσεων θα πρέπει πρώτα να προσδιορισθούν οι αντικειμενικοί στόχοι της παρακολούθησης της ποιότητας του αέρα. Η ακρίβεια των μετρήσεων σχετίζεται άμεσα με τη χρήση για την οποία προορίζονται.

Τα δίκτυα καταγραφής των επιπέδων συγκέντρωσης των αιωρούμενων σωματιδίων που αποτελούνται από διατάξεις που χρησιμοποιούν μία από τις ενεργητικές μεθόδους υπολογισμού. Στόχος τους είναι εκτός της αξιολόγησης της ποιότητας και η χρήση των δεδομένων αυτών για την προφύλαξη της υγείας των ανθρώπων. Λόγω της διακύμανσης της συγκέντρωσης στο εύρος μιας περιοχής, εξαιτίας των παραμέτρων που αναφέραμε προηγουμένως, είναι απαραίτητη η κατασκευή και η εγκατάσταση ενός μεγάλου αριθμού μετρητών με σκοπό την εύρεση ρεαλιστικών αποτελεσμάτων. Το κόστος των μετρητών και ιδιαίτερα των ενεργητικών μας οδηγεί στην ανάγκη για την ελαχιστοποίηση του πλήθους των μετρητών που απαιτούνται σε μία περιοχή διατηρώντας της ακρίβεια των αποτελεσμάτων. Στην συνέχεια θα αναπτύξουμε την μεθοδολογία εύρεσης βέλτιστης διάταξης μετρητών αέριων ρύπων.

## **Ενότητα 3: Μεθοδολογία**

### **3.1 Αρχικά στοιχεία**

#### **3.1.1 Εισαγωγή**

Στην παρούσα ενότητα θα αναλυθεί η μεθοδολογία εύρεσης βέλτιστων σημείων για την κατασκευή μονάδων καταγραφής .

Βήματα της μεθοδολογίας:

- α) Προσδιορισμός υποψήφιων σημείων για μετρητές
- β) Χαρακτηριστικά ανέμων ανά σημείο
- γ) Κριτήρια ρύπανσης ανά σημείο
- δ) Κατηγοριοποίηση σημείων (clustering)
- ε) Εύρεση βέλτιστης διάταξης

### 3.1.2 Δείκτες Μεταβλητών

**i** : Δείκτης για τα πιθανά σημεία κατασκευής μετρητών ρύπανσης

**j** : Δείκτης για τα σημεία υψηλής ρύπανσης

**e** : Δείκτης για τα κριτήρια ρύπανσης

**m** : Δείκτης των μετεωρολογικών φαινομένων

### 3.1.3 Μεταβλητές της μεθοδολογίας

$(x_i, y_i)$  ή  $(x_j, y_j)$  : Συντεταγμένες σημείων (μετρητών ρύπανσης ή υψηλής ρύπανσης).

**We,i** : Συντελεστής βαρύτητας των κριτηρίων ρύπανσης e σε κάθε σημείο i .

**Pm** : Ποσοστό εμφάνισης κάθε τύπου ανέμου .

**Pi,j,m** : Συντελεστής που δηλώνει αν το κριτήριο ρύπανσης j (όσες και οι πηγές ρύπανσης του κριτηρίου 5) επηρεάζει το σημείο i υπό το μετεωρολογικό φαινόμενο m. Μπορεί να πάρει τις τιμές 0 έως και 1 .

**Wtotal,i** : Αθροιστικός συντελεστής που μας δείχνει το πόσο επιβαρυνόμενο είναι το σημείο i σε ρύπανση συνυπολογίζοντας όλες τις περιπτώσεις κριτηρίων ρύπανσης (Συνολική βαρύτητα ρύπανσης σημείου).

**dist<sub>ij</sub>** : απόσταση σημείου i με το σημείο υψηλής ρύπανσης j

**gwnies<sub>ij</sub>** : γωνία του σημείου τοποθέτησης μετρητή i σε σχέση με το σημείο υψηλής ρύπανσης j

**Group\_number** : Αριθμός μετρητών που θα τοποθετηθούν

## 3.2 Προσδιορισμός υποψήφιων σημείων

Αρχικό στάδιο είναι ο ορισμός σημείων στην περιοχή τα οποία επιλέγονται ώστε να απέχουν μεταξύ τους μία σταθερή απόσταση η οποία θα ορίζεται στην αρχή της εφαρμογής. Η ακριβής θέση των σημείων μπορεί να μεταβληθεί σε σημεία κόντα στο αρχικό που θα επιτρέπουν την κατασκευή των μετρητών (π.χ. όταν τα σημεία βρίσκονται πάνω σε ποτάμια ή σε κτήρια). Ο αριθμός των σημείων αποφασίζεται από τον χρήστη.

## 3.3 Επίδραση ανέμων

Όπως αναφέρθηκε στη βιβλιογραφία, η ατμοσφαιρική ρύπανση επηρεάζεται άμεσα από την κατεύθυνση των ανέμων. Επίσης, οι ατμοσφαιρικοί ρύποι μπορούν να συσσωρευτούν σε μια περιοχή. Συνεπώς, η υποβάθμιση του βιοτικού επιπέδου καθιστά απαραίτητη την απομάκρυνση των ρύπων. Στην παρούσα εργασία λαμβάνεται υπόψη η κατεύθυνση του αέρα η οποία θα χρησιμοποιηθεί στο κριτήριο των πηγών ρύπανσης λόγω βιομηχανιών, λιμανιών.

Τα δεδομένα για τις κατευθύνσεις των ανέμων θα ληφθούν από το αρχείο excel που θα τροφοδοτήσει ο χρήστης. Παρακάτω παρουσιάζονται τα στατιστικά για τις κατευθύνσεις των ανέμων του έτους 2021 στη πόλη του Βόλου.

Πίνακας 2: Κατεύθυνση- Ένταση Ανέμων<sup>[17]</sup>

Μποφόρ\Κατεύθυνση	N	NNE	NE	ENE	E	ESE	SE	SSE
0(νηνεμία)	25,89%	0%	0%	0%	0%	0%	0%	0%
1 to 2	2,46%	2,73%	5,46%	17,48%	1,09%	1,09%	1,91%	12,84%
>3	0,54%	1,63%	0%	0,54%	0%	0%	0%	0%

Μποφόρ\Κατεύθυνση	S	SSW	SW	WSW	W	WNW	NW	NNW
0(νηνεμία)	0%	0%	0%	0%	0%	0%	0%	0%
1 to 2	2,73%	1,09%	5,73%	1,09%	1,36%	4,64%	5,46%	3%
>3	0%	0%	0%	0%	0%	0,54%	0,54%	0%

#### Κατεύθυνση ανέμου-δείκτης m

- Τιμές του δείκτη m :
- m = 1 → κατεύθυνση ανέμου = N ( 0° ,360° )
  - m = 2 → κατεύθυνση ανέμου = NNE ( 22.5° )
  - m = 3 → κατεύθυνση ανέμου = NE ( 45° )
  - m = 4 → κατεύθυνση ανέμου = ENE ( 67.5° )
  - m = 5 → κατεύθυνση ανέμου = E ( 90° )
  - m = 6 → κατεύθυνση ανέμου = ESE ( 112.5° )
  - m = 7 → κατεύθυνση ανέμου = SE ( 135° )
  - m = 8 → κατεύθυνση ανέμου = SSE ( 147.5° )
  - m = 9 → κατεύθυνση ανέμου = S ( 180° )
  - m = 10 → κατεύθυνση ανέμου = SSW ( 202.5° )
  - m = 11 → κατεύθυνση ανέμου = SW ( 225° )
  - m = 12 → κατεύθυνση ανέμου = WSW ( 247.5° )
  - m = 13 → κατεύθυνση ανέμου = W ( 270° )
  - m = 14 → κατεύθυνση ανέμου = WNW ( 292.5° )
  - m = 15 → κατεύθυνση ανέμου = NW ( 315° )
  - m = 16 → κατεύθυνση ανέμου = NNW ( 337.5° )

Υπολογισμός  $P_{i,j,m}$  :

**Pm** : Ποσοστό εμφάνισης κάθε τύπου ανέμου

**wind\_impact\_(κατεύθυνση ανέμου)[i][j]** : ποσοστό που επηρεάζεται το σημείο  $i$  από το σημείο ρύπανσης  $j$  για τις κατεύθυνσεις ανέμου μεταξύ των οποίων βρίσκεται το σημείο  $i$  βάση της γωνίας του σε σχέση με το σημείο  $j$  ( $gwnies[i, j]$ )

Το **αριστερό όριο** και **δεξί όριο** παρακάτω αναφέρονται στις κατευθύνσεις ανέμου μεταξύ των οποίων βρίσκεται η μεταβλητή  $gwnies[i][j] \forall i, j$  (π.χ. η γωνία  $13^\circ$  είναι μεταξύ των κατευθύνσεων N αριστερά και NNE δεξιά) .

$$\text{wind\_impact\_}(\text{Αριστερό όριο})[i][j] = \frac{gwnies[i][j] - (\text{Μοίρες αριστερού όριο})}{22.5}$$

$$\text{wind\_impact\_}(\text{Δεξιό όριο})[i][j] = \frac{(\text{Μοίρες δεξιού όριο}) - gwnies[i][j]}{22.5}$$

$$P_{i,j,m} = \sum_1^m \text{wind\_impact\_}[m][i][j] * P_m \quad \forall i, j$$

### 3.4 Κριτήρια ρύπανσης

#### 3.4.1 Μέγεθος δρόμου

Το πρώτο κριτήριο σχετίζεται με το μέγεθος των δρόμων. Το πλάτος των δρόμων μπορεί να ευνοεί ή να εμποδίζει την απομάκρυνση των ρύπων για μεγάλο ή μικρό πλάτος αντίστοιχα. Στον παρακάτω πίνακα παρουσιάζονται οι κατηγορίες του κριτηρίου.

Πίνακας 3: Κριτήριο 1<sup>ο</sup> - Μέγεθος δρόμου

<b>Δ)Κριτήριο 1<sup>ο</sup> - Μέγεθος δρόμου</b>		
<b>Μέγεθος Δρόμου</b>	<b>Πλάτος σε (m)</b>	<b>W1,i</b>
<b>Κατηγορία 1</b>	20-30	1
<b>Κατηγορία 2</b>	15-20	2
<b>Κατηγορία 3</b>	10-15	3
<b>Κατηγορία 4</b>	<10	4

#### 3.4.2 Ύψος κτιρίων

Το δεύτερο κριτήριο αφορά το ύψος των κτιρίων καθώς το ύψος των κτισμάτων επηρεάζει την ευκολία με την οποία μπορούν να απομακρυνθούν οι ρύποι της περιοχής. Για περιοχές με χαμηλό ύψος ευνοείται η απομάκρυνση αυτή σε αντίθεση με τις περιοχές μεγάλου ύψους και στην περίπτωση μηδενικού ύψους

δηλαδή έλλειψη δόμησης υπάρχει φυσικός αερισμός της περιοχής. Στη συνέχεια παρουσιάζονται οι κατηγορίες ανάλογα με το ύψος των κτηρίων σε κάθε σημείο.

Πίνακας 4: Κριτήριο 2<sup>ο</sup> - Ύψος κτιρίων

<b>II)Κριτήριο 2<sup>ο</sup> - Ύψος κτιρίων</b>		
<b>Ύψος Κτηρίων</b>	<b>Ύψος σε (m)</b>	<b>W2,i</b>
<b>Κατηγορία 0</b>	0	0
<b>Κατηγορία 1</b>	<10	1
<b>Κατηγορία 2</b>	10-15	2
<b>Κατηγορία 3</b>	15-20	3
<b>Κατηγορία 4</b>	>20	4

### 3.4.3 Όγκος κυκλοφορίας

Το τρίτο κριτήριο σχετίζεται με τον όγκο κυκλοφορίας σε μία περιοχή. Όσο μεγαλύτερη είναι η κυκλοφορία τόσο μεγαλύτερη είναι η παραγωγή ρύπων στην περιοχή λόγω της κυκλοφορίας. Οι κατηγορίες δημιουργήθηκαν σύμφωνα με την παράγραφο 3.5.1 του άρθρου 3 του ΦΕΚ 88 ΑΑΠ/2016 <sup>[18]</sup> και παρουσιάζονται στον παρακάτω πίνακα.

Πίνακας 5: Κριτήριο 3<sup>ο</sup> – Όγκος Κυκλοφορίας

<b>III)Κριτήριο 3<sup>ο</sup> – Όγκος Κυκλοφορίας</b>		
<b>Όγκος Κυκλοφορίας</b>	<b>Τύπος οδών</b>	<b>W3,i</b>
<b>Κατηγορία 0</b>	Πεζόδρομοι	<b>0</b>
<b>Κατηγορία 1</b>	Τοπικές Οδοί (εξυπηρετούν τις περιοχές κατοικίας)	1
<b>Κατηγορία 2</b>	Συλλεκτήριες Οδοί (Μετακινήσεις προς κύριες αρτηρίες και πολεοδομικές ενότητες)	2
<b>Κατηγορία 3</b>	Δευτερεύουσες Οδοί (Μικρότερος όγκος κυκλοφορίας από τις κύριες)	3
<b>Κατηγορία 4</b>	Κύριες Αρτηρίες (Υψηλές Ταχύτητες, Μεγάλος Όγκος Κυκλοφορίας)	4
<b>Κατηγορία 5</b>	Ελεύθεροι Λεωφόροι	5

### 3.4.4 Οικιακή θέρμανση<sup>[19]</sup>

Το τέταρτο κριτήριο αφορά την οικιακή θέρμανση και συγκεκριμένα την ενέργεια την οποία καταναλώνουν τα σημεία ενδιαφέροντος κατα μέσο όρο. Στον πίνακα Π 3.7.1 του ΦΕΚ 88 ΑΑΠ/2016 <sup>[18]</sup> το πολεοδομικό συγκρότημα της περιοχής χωρίζεται σε ενότητες και δεδομένου ότι ο θεωρητικός

πληθυσμός είναι 1 άτομο/ 18 τ.μ. μπορούμε να υπολογίσουμε τα τ.μ. κάθε ενότητας από το Προεδρικό Διάταγμα υπ' αριθμ. 41/2018, ΦΕΚ 80/Α/7-5-2018 Κανονισμός Πυροπροστασίας Κτιρίων <sup>[19]</sup>. Για κάθε 1τ.μ. καταναλώνονται 200 kcal ενέργειας δίνοντας τα kcal ενέργειας που καταναλώνονται στην περιοχή. Στο παρακάτω πίνακα αναγράφονται οι κατηγορίες του κριτηρίου.

Πίνακας 6: Κριτήριο 4ο - Οικιακή θέρμανση

<b><u>IV)Κριτήριο 4ο - Ποσότητα ενέργειας που καταναλώνεται για οικιακή θέρμανση</u></b>		
<b>Ποσότητα</b>	<b>Ποσότητα σε kcal</b>	<b>W4,i</b>
<b>Κατηγορία 1</b>	<20.000.000	1
<b>Κατηγορία 2</b>	20.000.000-30.000.000	2
<b>Κατηγορία 3</b>	30.000.000-40.000.000	3
<b>Κατηγορία 4</b>	40.000.000-50.000.000	4
<b>Κατηγορία 5</b>	>50.000.000	5

### 3.4.5 Πηγές ρύπανσης(βιομηχανίες , λιμάνια)

Το πέμπτο κριτήριο αφορά τις επιπλέον πηγές ρύπανσης στην περιοχή την οποία εξετάζουμε (βιομηχανικές περιοχές , λιμάνια κτλ.). Στον παρακάτω πίνακα δίνονται οι κατηγορίες για τις πηγές ρύπανσης .Στη συνέχεια με την χρήση των συντελεστών των κατηγοριών ,καθώς και των αποστάσεων των πηγών από τα σημεία θα διατυπωθεί ο τύπος για τον συντελεστή βαρύτητας (W5,i).

Πίνακας 7: Κριτήριο 5ο – Πηγές ρύπανσης

<b><u>V)Κριτήριο 5ο –Πηγές ρύπανσης(ο αριθμός των οποίων εξαρτάται από τον χρήστη)</u></b>		
<b>Πηγές Ρύπανσης</b>	<b>Τύποι πηγών ρύπανσης</b>	<b>Zj</b>
<b>Κατηγορία 1</b>	Μικρή παραγωγική μονάδα	1
<b>Κατηγορία 2</b>	Μεσαία παραγωγική μονάδα	2
<b>Κατηγορία 3</b>	Μεγάλη παραγωγική μονάδα-λιμάνι	3
<b>Κατηγορία 4</b>	Πολύ μεγάλη παραγωγική μονάδα - βιομηχανική ζώνη(με πολλές παραγωγικές μονάδες)	4
<b>Κατηγορία 5</b>	Βιομηχανική ζώνη με πολλές μονάδες ή εξαιρετικά πολύ μεγάλη παραγωγική μονάδα	5



Υπολογισμός  $W_{5,i}$  :

Για το κριτήριο 5 ο συντελεστής βαρύτητας ( $W_{5,i}$ ) λαμβάνει υπόψιν την απόσταση της πηγής ρύπανσης από το κάθε σημείο. Έτσι καταλήγουμε στην παρακάτω σχέση.

$$W_{5ij} = Z_j - Z_j * (0.02/500) * \text{dist}_{ij}$$

Θεωρούμε ότι για κάθε 500 μέτρα από το σημείο παραγωγής το  $W_{5ij}$  έχει μείωση περίπου 2%

### 3.5 Κατηγοριοποίηση σημείων

Για την αύξηση της ποικιλότητας των δεδομένων μετρήσεων της ατμοσφαιρικής ρύπανσης, επιδιώκεται η τοποθέτηση μονάδων καταγραφής της ποιότητας του αέρα σε σύνολα σημείων με διαφορετικά επίπεδα ρύπανσης. Μέσω της κατηγοριοποίησης δημιουργούνται ομάδες σημείων. Στη συνέχεια παρουσιάζονται οι δύο μέθοδοι κατηγοριοποιήσεων. Η κατηγοριοποίηση θα γίνει και στις δύο περιπτώσεις βάση των τιμών  $W_{\text{total},i}$  του κάθε σημείου.

Υπολογισμός του συνολικού συντελεστή βαρύτητας κάθε σημείου  $i$  ( $W_{\text{total},i}$ ) :

$$W_{\text{total}[i]} = \sum_1^4 [W_{ie}] + \sum_5^{j+5} [W_{5ij} * (\sum_1^m P_{i,j,m})] \quad \forall i$$

- Κατηγοριοποίηση 1 (ίσα διαστήματα)

Η μέθοδος αυτή έχει ως στόχο να χωρίσει τα σημεία σε ομάδες βάση του συνολικού συντελεστή βαρύτητας. Το εύρος των τιμών  $W_{\text{total}}$  χωρίζεται σε  $g$  διαστήματα όπου  $g$  αριθμός των μετρητών που θα κατασκευαστούν και στη συνέχεια τα σημεία αναθέτονται στην ομάδα όπου το εύρος της εμπεριέχει την τιμή του  $W_{\text{total}}$  του σημείου.

Τα διαστήματα υπολογίζονται από την ακόλουθη σχέση:

$$\text{Διάστημα} = \frac{\text{MAX}(W_{\text{total}}) - \text{MIN}(W_{\text{total}})}{\text{Group\_number}}$$

- Κατηγοριοποίηση 2 (ίδιος αριθμός σημείων)

Τα σημεία χωρίζονται σε  $g$  ομάδες και κατανέμονται από το μικρότερο  $W_{\text{total}}$  προς το μεγαλύτερο ώστε κάθε ομάδα να έχει τον ίδιο αριθμό σημείων.

$w[g]$  : αριθμός σημείων σε κάθε ομάδα που δημιουργήθηκε

$Group[g][w]$  : κάθε γραμμή περιέχει τα σημεία της κάθε ομάδας

Για την αξιολόγηση των αποτελεσμάτων θα χρησιμοποιηθούν δύο επιπλέον μεγέθη που σε συνδυασμό με την αντικειμενική συνάρτηση θα μας επιδείξουν την βέλτιστη λύση. Τα μεγέθη αυτά είναι το **ποσοστό κάλυψης** το οποίο ορίζεται ως το άθροισμα των σημείων που έχουν επιλεγθεί στην λύση και των σημείων που βρίσκονται εντός της εμβέλειας του μετρητή αέριου ρύπου του κάθε σημείου προς το σύνολο των σημείων ενδιαφέροντος. Το δεύτερο μέγεθος είναι το **ποσοστό της υπερκάλυψης** που ορίζεται ως το άθροισμα των σημείων που καλύπτονται από περισσότερο από έναν μετρητή προς το άθροισμα όλων των σημείων.

Η επιλογή του αριθμού των μετρητών αέριων ρύπων επηρεάζει την κάλυψη και την υπερκάλυψη της περιοχής. Για τον λόγο αυτό απαιτείται η δοκιμή ενός εύρους διαφορετικών τιμών μετρητών με σκοπό την επιλογή του βέλτιστου αριθμού μετρητών .

### 3.6 Ευρεση βέλτιστης λύσης

Τελικός στόχος του προγράμματος είναι η εύρεση των βέλτιστων σημείων για τους μετρητές αέριων ρύπων.

- i. Οι μεταβλητές απόφασης που θα χρησιμοποιηθούν για την επίλυση είναι :
  - **Kig** : δυαδική μεταβλητή που δηλώνει αν στο σημείο  $i$  που ανήκει στην ομάδα  $g$  τοποθετείται μετρητής (0 δεν τοποθετείται ή 1 τοποθετείται )
- ii. Οι περιορισμοί που χρησιμοποιούνται είναι :
  - Ο πρώτος περιορισμός εξασφαλίζει ότι σε κάθε σημείο θα τοποθετηθεί το πολύ ένας μετρητής της ποιότητας του αέρα.

$$\sum_1^{Group\_number} K_{ig} \leq 1 \quad \forall i$$

- Ο δεύτερος περιορισμός εξασφαλίζει ότι σε κάθε ομάδα θα τοποθετηθεί ακριβώς ένας μετρητής της ποιότητας του αέρα.

$$\sum_1^{points\_num} K_{ig} \leq 1 \quad \forall g$$

- Ο τρίτος περιορισμός εξασφαλίζει ότι μεταξύ των σημείων  $i$  της κάθε ομάδας που δημιουργήθηκε με τις κατηγοριοποιήσεις θα τοποθετείται μετρητής σε ακριβώς ένα από αυτά.

$$\sum_1^{w[g]} K[\text{Group}[g][w]]g \leq 1 \quad \forall g$$

- iii. Η αντικειμενική συνάρτηση έχει ως στόχο την τοποθέτηση των μετρητών στα σημεία με τα μεγαλύτερα επίπεδα ρύπανσης για οποιαδήποτε είδος κατηγοριοποίησης μεγιστοποιώντας ταυτόχρονα την μεταξύ τους απόσταση ώστε να επιτευχθεί η καλύτερη δυνατή κάλυψη.

$$\text{Max Obj} = \sum_1^g \sum_1^i \left( \left( \frac{W_{\text{total}[i]}}{\text{max\_Wtotal}} \right) * K_{\text{ig}[i][g]} + \sum_1^l \left( \frac{\text{Dist}[i][j]}{\text{max\_dist}} * K_{\text{ig}[i][g]} * K_{\text{il}[l][g]} \right) \right)$$

$l, i$ : πλήθος των σημείων τοποθέτησης μετρητών

$\text{max\_Wtotal}$ : μέγιστος συνολικός συντελεστής βαρύτητας μεταξύ όλων των σημείων

$\text{max\_dist}$ : μέγιστη απόσταση μεταξύ δύο οποιοδήποτε σημείων

Τελικά, η εκτέλεση του προγράμματος παρέχει ως έξοδο στον χρήστη ένα σύνολο δυαδικών τιμών που εκφράζουν τις βέλτιστες λύσεις των μετρητών, την τιμή της αντικειμενικής συνάρτησης, την κάλυψη καθώς και την υπερκάλυψη .

Η βελτιστοποίηση θα πραγματοποιηθεί με την χρήση της εφαρμογής βελτιστοποίησης Cplex<sup>[20]</sup>

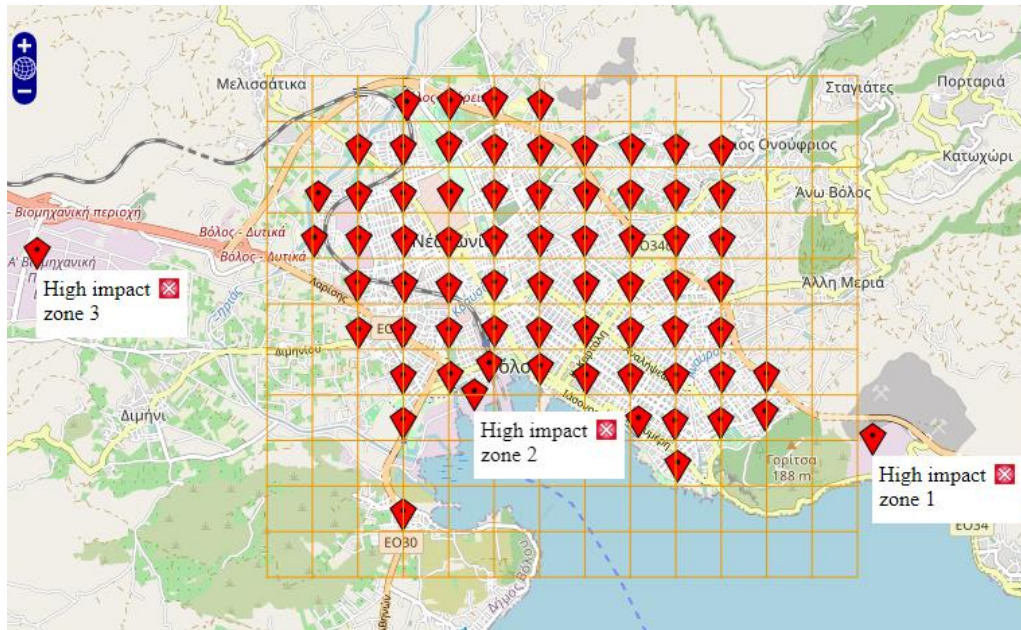
## **Ενότητα 4: Αποτελέσματα εφαρμογής**

### **4.1 Τοποθεσία εφαρμογής του προγράμματος<sup>[21],[22]</sup>**

Το πρόγραμμα θα εφαρμοστεί για την πόλη του Βόλου που είναι η πρωτεύουσα του νομού Μαγνησίας. Η ατμοσφαιρική ρύπανση στην περιοχή είναι αυξημένη ξεπερνώντας πολλές φορές τα όρια που έχουν οριστεί από τον Παγκόσμιο Οργανισμό Υγείας που παρουσιάστηκαν προηγουμένως. Επίσης, οι βιομηχανικές μονάδες που είναι ιδιαίτερα επιβλαβείς για την ποιότητα του αέρα στην περιοχή είναι, το εργοστάσιο της ΑΓΕΤ που βρίσκεται στον παραλιακό δρόμο προς Αγριά και Ανατολικό Πήλιο, η Χαλυβουργία Βόλου με εγκαταστάσεις που αποτελούνται από την κύρια βιομηχανική μονάδα στο Βελεστίνο, το ελασματοουργείο επιμηκών προϊόντων και το εργοστάσιο παραγωγής πλεγμάτων. Τέλος ιδιαίτερα επιβαρυντικό είναι και το εμπορικό λιμάνι που βρίσκεται πολύ κοντά στο κέντρο του Βόλου. Για τους λόγους αυτούς το λιμάνι και οι δύο βιομηχανικές ομάδες θα ενταχθούν στο κριτήριο 5 των πηγών υψηλής ρύπανσης.

### **4.2 Ορισμός σημείων**

Αρχικά ορίζουμε τα σημεία καθώς και τις πηγές υψηλής ρύπανσης. Επιλέχθηκαν 67 σημεία για την κατασκευή των μετρητών αέριων ρύπων. Στην παρακάτω εικόνα βλέπουμε τα σημεία καθώς και τις πηγές ρύπανσης που αναφέραμε προηγουμένως.



### Σημεία και συντεταγμένες

- 1 ) 39.34622370993237 , 22.926152268324696
- 2 ) 39.35518380883058 , 22.926152268324696
- 3 ) 39.35956388345646 , 22.926152268324696
- 4 ) 39.35972978997558 , 22.93207458582987
- 5 ) 39.36410957956806 , 22.92048744288493
- 6 ) 39.36868815685862 , 22.920358696852226
- 7 ) 39.373133734506425 , 22.914779702101523
- 8 ) 39.37754585632023 , 22.915337601575963
- 9 ) 39.382156723228945 , 22.92040161219676
- 10 ) 39.38656827477192 , 22.92668871012732
- 11 ) 39.382156723228945 , 22.926216641341043
- 12 ) 39.38661802752239 , 22.932096043500785
- 13 ) 39.386783869767534 , 22.93780378428509
- 14 ) 39.386518521985984 , 22.943597355756662
- 15 ) 39.38207379665818 , 22.94364027110119
- 16 ) 39.38200745533049 , 22.94939092722913
- 17 ) 39.382140137924225 , 22.955313244734302

- 18 ) 39.382140137924225 , 22.960935154829535
- 19 ) 39.38204062600325 , 22.966814556990176
- 20 ) 39.377595615503814 , 22.966857472333807
- 21 ) 39.3729844473134 , 22.966814556990176
- 22 ) 39.36863839132853 , 22.966728726301106
- 23 ) 39.36405981077587 , 22.966728726301106
- 24 ) 39.35971319934229 , 22.972522297773576
- 25 ) 39.35516721711766 , 22.966728726301106
- 26 ) 39.35101916502876 , 22.96119264689494
- 27 ) 39.35539950075258 , 22.956171551618677
- 28 ) 39.35513403367791 , 22.960892239485
- 29 ) 39.3597463806063 , 22.966771641645643
- 30 ) 39.359613655454346 , 22.96097807017317
- 31 ) 39.36429206484204 , 22.960892239485
- 32 ) 39.36870474536147 , 22.960978070174068
- 33 ) 39.37318349683412 , 22.960978070174068
- 34 ) 39.37766196102468 , 22.960935154829535
- 35 ) 39.37762878827178 , 22.95522741404613
- 36 ) 39.37315032195318 , 22.955270329389766
- 37 ) 39.36860521428807 , 22.955184498701602
- 38 ) 39.3641261691592 , 22.95522741404613
- 39 ) 39.35964683676561 , 22.955184498701602
- 40 ) 39.35964683676561 , 22.949433842573658
- 41 ) 39.3641925274795 , 22.949476757918195
- 42 ) 39.36863839132853 , 22.949433842573658
- 43 ) 39.37315032195318 , 22.94939092722913
- 44 ) 39.37766196102468 , 22.949433842573658
- 45 ) 39.37766196102468 , 22.94351152506849
- 46 ) 39.37315032195318 , 22.94364027110119
- 47 ) 39.36860521428807 , 22.94364027110119

- 48 ) 39.36415934832687 , 22.943554440413024  
49 ) 39.36054272622488 , 22.94372610179026  
50 ) 39.36084135349126 , 22.937202969466114  
51 ) 39.36422570661634 , 22.93784669962962  
52 ) 39.36873792235461 , 22.937760868940554  
53 ) 39.37318349683412 , 22.937760868940554  
54 ) 39.37766196102468 , 22.937760868940554  
55 ) 39.38210696729942 , 22.93780378428509  
56 ) 39.382438672836486 , 22.93190292445353  
57 ) 39.373017622273835 , 22.931988755141695  
58 ) 39.37756244272008 , 22.92610935298106  
59 ) 39.37311714705786 , 22.926238099013762  
60 ) 39.37756244272008 , 22.920444527541292  
61 ) 39.37311714705786 , 22.920444527541292  
62 ) 39.36857203723325 , 22.926238099013762  
63 ) 39.36853886016128 , 22.931945839797166  
64 ) 39.36412616915991 , 22.931988755141695  
65 ) 39.36412616915991 , 22.926195183669233  
66 ) 39.37762878827318 , 22.932074585830765  
67 ) 39.3559370433529 , 22.97223557480371

### 4.3 Κατηγοριοποίηση σημείων

Στη συνέχεια θα παρουσιαστούν τα αποτελέσματα για το κάθε είδος κατηγοριοποίησης που αναπτύχθηκαν στο προηγούμενο κεφάλαιο. Ο αριθμός μετρητών(Group\_number) θα είναι για το εύρος 16 έως 26 και θα επιλέξουμε την καλύτερη δυνατή λύση.

Αρχικά θα παρουσιάσουμε τις ομάδες που θα σχηματιστούν για κάθε αριθμό μετρητών για τα δύο είδη κατηγοριοποιήσεων που παρουσιάστηκαν προηγουμένως.

#### Κατηγοριοποίηση 1(ίσα διαστήματα)

##### *Δεκαέξι ομάδες σημείων*

Ομάδα 1 – Σημεία : 7

Ομάδα 2 – Σημεία : 4 , 64

Ομάδα 3 – Σημεία : 44

Ομάδα 4 – Σημεία : 9 , 15 , 17 , 19 , 21 , 23  
Ομάδα 5 – Σημεία : 11 , 45 , 46 , 56 , 66  
Ομάδα 6 – Σημεία : 5 , 13  
Ομάδα 7 – Σημεία : 1 , 2 , 3 , 6 , 8 , 10 , 12 , 16 22 , 24 , 25 , 32 , 34 , 50 , 61 , 62  
Ομάδα 8 – Σημεία : 14 , 18 , 20 , 26 , 30 , 33 , 35 , 49 , 54 , 55 , 60 , 63  
Ομάδα 9 – Σημεία : 58  
Ομάδα 10 – Σημεία : 29 , 31 , 43 , 51 , 57  
Ομάδα 11 – Σημεία : 36 , 42 , 53 , 59  
Ομάδα 12 – Σημεία : 37 , 38 , 40 , 47 , 52 , 65 , 67  
Ομάδα 13 – Σημεία : 27 , 28  
Ομάδα 14 – Σημεία : 39  
Ομάδα 15 – Σημεία : 41  
Ομάδα 16 – Σημεία : 48

#### *Δεκαεπτά ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4 , 64  
Ομάδα 3 – Σημεία : 44  
Ομάδα 4 – Σημεία : 9,15,17,19,21  
Ομάδα 5 – Σημεία : 11,23  
Ομάδα 6 – Σημεία : 5,45,46,56 ,66  
Ομάδα 7 – Σημεία : 1,2,3,6,8,10,12,13,16,22,24,25,32,34,50,61,62  
Ομάδα 8 – Σημεία : 14,18,20,26,30,33,35,49,54,55,60,63  
Ομάδα 9 – Σημεία : 58  
Ομάδα 10 – Σημεία : 29,43,51,57  
Ομάδα 11 – Σημεία : 31,36,42,53  
Ομάδα 12 – Σημεία : 37,38,40,47,52,59,65,67  
Ομάδα 13 – Σημεία : 27  
Ομάδα 14 – Σημεία : 28  
Ομάδα 15 – Σημεία : 39  
Ομάδα 16 – Σημεία : 41  
Ομάδα 17 – Σημεία : 48

#### *Δεκαοχτώ ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4 , 64  
Ομάδα 3 – Σημεία : 44  
Ομάδα 4 – Σημεία : 9  
Ομάδα 5 – Σημεία : 15 , 17 , 19 , 21 , 23  
Ομάδα 6 – Σημεία : 5 , 11 , 45 , 46 , 56 , 66  
Ομάδα 7 – Σημεία : 8 , 10 , 13 , 16 , 34 , 61  
Ομάδα 8 – Σημεία : 1 , 2 , 3 , 6 , 12 , 22 , 24 , 25 , 32 , 50 , 62  
Ομάδα 9 – Σημεία : 14 , 18 , 20 , 26 , 30 , 33 , 35 , 49 , 54 , 55 , 60 , 63  
Ομάδα 10 – Σημεία : 58  
Ομάδα 11 – Σημεία : 43 , 57

Ομάδα 12 – Σημεία : 29 , 31 , 42 , 51  
Ομάδα 13 – Σημεία : 36 , 37 , 38 , 40 , 47 , 52 , 53 , 59 , 65 , 67  
Ομάδα 14 – Σημεία : 27  
Ομάδα 15 – Σημεία : 28  
Ομάδα 16 – Σημεία : 39  
Ομάδα 17 – Σημεία : 41  
Ομάδα 18 – Σημεία : 48

#### *Δεκαεννιά ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4 , 64  
Ομάδα 3 – Σημεία : 44  
Ομάδα 4 – Σημεία : 9  
Ομάδα 5 – Σημεία : 15 , 17 , 19 , 21 , 23  
Ομάδα 6 – Σημεία : 11 , 45 , 46 , 56 , 66  
Ομάδα 7 – Σημεία : 5  
Ομάδα 8 – Σημεία : 1 , 2 , 3 , 6 , 8 , 10 , 12 , 13 , 16 , 22 , 24 , 25 , 32 , 34 , 50 , 61 , 62  
Ομάδα 9 – Σημεία : 14 , 18 , 20 , 26 , 30 , 33 , 35 , 49 54 , 55 , 60  
Ομάδα 10 – Σημεία : 63  
Ομάδα 11 – Σημεία : 58  
Ομάδα 12 – Σημεία : 29 , 31 , 42 , 43 , 51 , 57  
Ομάδα 13 – Σημεία : 36 , 53 , 59  
Ομάδα 14 – Σημεία : 37 , 38 , 40 , 47 , 52 , 65 , 67  
Ομάδα 15 – Σημεία : 27  
Ομάδα 16 – Σημεία : 28  
Ομάδα 17 – Σημεία : 39  
Ομάδα 18 – Σημεία : 41  
Ομάδα 19 – Σημεία : 48

#### *Είκοσι ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4  
Ομάδα 3 – Σημεία : 64  
Ομάδα 4 – Σημεία : 44  
Ομάδα 5 – Σημεία : 9 15 17 19 21  
Ομάδα 6 – Σημεία : 23  
Ομάδα 7 – Σημεία : 5 11 45 46 56 66  
Ομάδα 8 – Σημεία : 2 3 8 10 12 13 16 34 61  
Ομάδα 9 – Σημεία : 1 6 22 24 25 32 50 62  
Ομάδα 10 – Σημεία : 14 18 20 26 30 33 35 49 54 55 60 63  
Ομάδα 11 – Σημεία : 58  
Ομάδα 12 – Σημεία : 29  
Ομάδα 13 – Σημεία : 31 42 43 51 57  
Ομάδα 14 – Σημεία : 36 38 40 53 59  
Ομάδα 15 – Σημεία : 37 47 52 65 67



Ομάδα 16 – Σημεία : 27  
Ομάδα 17 – Σημεία : 28  
Ομάδα 18 – Σημεία : 39  
Ομάδα 19 – Σημεία : 41  
Ομάδα 20 – Σημεία : 48

### *Εικοσιένα ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4  
Ομάδα 3 – Σημεία : 64  
Ομάδα 4 – Σημεία : 44  
Ομάδα 5 – Σημεία : 15 , 17  
Ομάδα 6 – Σημεία : 9 , 19 , 21 , 23  
Ομάδα 7 – Σημεία : 5 , 11 , 45 , 46 , 56 , 66  
Ομάδα 8 – Σημεία : 1  
Ομάδα 9 – Σημεία : 2 , 3 , 6 , 8 , 10 , 12 , 13 , 16 , 22 , 24 , 25 , 32 , 34 , 50 , 61 , 62  
Ομάδα 10 – Σημεία : 14 , 18 , 20 , 26 , 30 , 33 , 35 , 49 , 54 , 55 , 60  
Ομάδα 11 – Σημεία : 63  
Ομάδα 12 – Σημεία : 58  
Ομάδα 13 – Σημεία : 43 , 57  
Ομάδα 14 – Σημεία : 29 , 31 , 42 , 51  
Ομάδα 15 – Σημεία : 36 , 38 , 40 , 47 , 52 , 53 , 59 , 65 , 67  
Ομάδα 16 – Σημεία : 37  
Ομάδα 17 – Σημεία : 27  
Ομάδα 18 – Σημεία : 28  
Ομάδα 19 – Σημεία : 39  
Ομάδα 20 – Σημεία : 41  
Ομάδα 21 – Σημεία : 48

### *Εικοσιδύο ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4  
Ομάδα 3 – Σημεία : 64  
Ομάδα 4 – Σημεία : 44  
Ομάδα 5 – Σημεία : 15 , 17  
Ομάδα 6 – Σημεία : 9 , 19 , 21 , 23  
Ομάδα 7 – Σημεία : 5 , 11 , 45 , 46 , 56 , 66  
Ομάδα 8 – Σημεία : 2  
Ομάδα 9 – Σημεία : 3 , 6 , 8 , 10 , 12 , 13 , 16 , 22 , 24 , 25 , 32 , 34 , 50 , 61 , 62  
Ομάδα 10 – Σημεία : 1 , 14 , 18 , 35 , 54 , 55  
Ομάδα 11 – Σημεία : 20 , 26 , 30 , 33 , 49 , 60 , 63  
Ομάδα 12 – Σημεία : 58  
Ομάδα 13 – Σημεία : 29  
Ομάδα 14 – Σημεία : 31 , 42 , 43 , 51 , 57

Ομάδα 15 – Σημεία : 36 , 53  
Ομάδα 16 – Σημεία : 37 , 38 , 40 , 47 , 52 , 59 , 65  
Ομάδα 17 – Σημεία : 67  
Ομάδα 18 – Σημεία : 27  
Ομάδα 19 – Σημεία : 28  
Ομάδα 20 – Σημεία : 39  
Ομάδα 21 – Σημεία : 41  
Ομάδα 22 – Σημεία : 48

### *Εικοσιτρείς ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4  
Ομάδα 3 – Σημεία : 64  
Ομάδα 4 – Σημεία : 44  
Ομάδα 5 – Σημεία : 9  
Ομάδα 6 – Σημεία : 15 , 17 , 19 , 21 , 23  
Ομάδα 7 – Σημεία : 5  
Ομάδα 8 – Σημεία : 11 , 45 , 46 , 56 , 66  
Ομάδα 9 – Σημεία : 2 , 8 , 10 , 12 , 13 , 16 , 34 , 61  
Ομάδα 10 – Σημεία : 1 , 3 , 6 , 22 , 24 , 25 , 32 , 50 , 62  
Ομάδα 11 – Σημεία : 14 , 18 , 20 , 26 , 30 , 33 , 35 , 49 , 54 , 55 , 60  
Ομάδα 12 – Σημεία : 63  
Ομάδα 13 – Σημεία : 58  
Ομάδα 14 – Σημεία : 57  
Ομάδα 15 – Σημεία : 29 , 31 , 42 , 43 , 51  
Ομάδα 16 – Σημεία : 36 , 40 , 53 , 59  
Ομάδα 17 – Σημεία : 37 , 38 , 47 , 52 , 65  
Ομάδα 18 – Σημεία : 67  
Ομάδα 19 – Σημεία : 27  
Ομάδα 20 – Σημεία : 28  
Ομάδα 21 – Σημεία : 39  
Ομάδα 22 – Σημεία : 41  
Ομάδα 23 – Σημεία : 48

### *Εικοσιτέσσερις ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4  
Ομάδα 3 – Σημεία : 64  
Ομάδα 4 – Σημεία : 44  
Ομάδα 5 – Σημεία : 9  
Ομάδα 6 – Σημεία : 15 , 17 , 19 , 21  
Ομάδα 7 – Σημεία : 23  
Ομάδα 8 – Σημεία : 5 , 11 , 45 , 46 , 56 , 66  
Ομάδα 9 – Σημεία : 2  
Ομάδα 10 – Σημεία : 3 , 6 , 8 , 10 , 12 , 13 , 16 , 22 , 24 , 25 , 34 , 50 , 61 , 62  
Ομάδα 11 – Σημεία : 1 , 14 , 32 , 35  
Ομάδα 12 – Σημεία : 18 , 20 , 26 , 30 , 33 , 49 , 54 , 55 , 60 , 63

Ομάδα 13 – Σημεία : 58  
Ομάδα 14 – Σημεία : 29  
Ομάδα 15 – Σημεία : 43 , 51 , 57  
Ομάδα 16 – Σημεία : 31 , 42  
Ομάδα 17 – Σημεία : 36 , 38 , 40 , 47 , 52 , 53 , 59  
Ομάδα 18 – Σημεία : 65  
Ομάδα 19 – Σημεία : 37  
Ομάδα 20 – Σημεία : 67  
Ομάδα 21 – Σημεία : 27  
Ομάδα 22 – Σημεία : 28  
Ομάδα 23 – Σημεία : 39  
Ομάδα 24 – Σημεία : 41 , 48

### *Εικοσιπέντε ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4  
Ομάδα 3 – Σημεία : 64  
Ομάδα 4 – Σημεία : 44  
Ομάδα 5 – Σημεία : 15  
Ομάδα 6 – Σημεία : 17 , 19  
Ομάδα 7 – Σημεία : 9 , 21 , 23  
Ομάδα 8 – Σημεία : 11 , 45 , 46 , 56 , 66  
Ομάδα 9 – Σημεία : 5  
Ομάδα 10 – Σημεία : 2 , 3 , 8 , 10 , 12 , 13 , 16 , 34 , 61  
Ομάδα 11 – Σημεία : 1 , 6 , 22 , 24 , 25 , 32 , 50 , 62  
Ομάδα 12 – Σημεία : 14 , 18 , 20 , 26 , 30 , 33 , 35 , 49 , 54 , 55 , 60  
Ομάδα 13 – Σημεία : 63  
Ομάδα 14 – Σημεία : 58  
Ομάδα 15 – Σημεία : 29  
Ομάδα 16 – Σημεία : 31 , 42 , 43 , 51 , 57  
Ομάδα 17 – Σημεία : 53  
Ομάδα 18 – Σημεία : 36 , 37 , 38 , 40 , 47 , 52 , 59  
Ομάδα 19 – Σημεία : 65  
Ομάδα 20 – Σημεία : 67  
Ομάδα 21 – Σημεία : 27  
Ομάδα 22 – Σημεία : 28  
Ομάδα 23 – Σημεία : 39  
Ομάδα 24 – Σημεία : 41  
Ομάδα 25 – Σημεία : 48

### *Εικοσιέξι ομάδες σημείων*

Ομάδα 1 – Σημεία : 7  
Ομάδα 2 – Σημεία : 4  
Ομάδα 3 – Σημεία : 64  
Ομάδα 4 – Σημεία : 44

Ομάδα 5 – Σημεία : 9  
Ομάδα 6 – Σημεία : 15  
Ομάδα 7 – Σημεία : 17 , 19 , 21 , 23  
Ομάδα 8 – Σημεία : 5  
Ομάδα 9 – Σημεία : 11 , 45 , 46 , 56 , 66  
Ομάδα 10 – Σημεία : 8 , 13 , 34  
Ομάδα 11 – Σημεία : 1 , 2 , 3 , 6 , 10 , 12 , 16 , 22 , 24 , 25 , 32 , 50 , 61 , 62  
Ομάδα 12 – Σημεία : 14 , 18 , 35 , 54 , 55  
Ομάδα 13 – Σημεία : 20 , 26 , 30 , 33 , 49 , 60 , 63  
Ομάδα 14 – Σημεία : 58  
Ομάδα 15 – Σημεία : 43  
Ομάδα 16 – Σημεία : 57  
Ομάδα 17 – Σημεία : 29 , 31 , 42 , 51  
Ομάδα 18 – Σημεία : 36 , 53 , 59  
Ομάδα 19 – Σημεία : 37 , 38 , 40 , 47 , 52  
Ομάδα 20 – Σημεία : 65  
Ομάδα 21 – Σημεία : 67  
Ομάδα 22 – Σημεία : 27  
Ομάδα 23 – Σημεία : 28  
Ομάδα 24 – Σημεία : 39  
Ομάδα 25 – Σημεία : 41  
Ομάδα 26 – Σημεία : 48

#### Κατηγοριοποίηση 2(ίδιος αριθμό σημείων)

##### *Δεκαέξι ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64 , 44 , 17  
Ομάδα 2 – Σημεία : 15 , 19 , 21 , 9 , 23  
Ομάδα 3 – Σημεία : 11 , 56 , 66 , 46 , 45  
Ομάδα 4 – Σημεία : 5 , 13 , 34 , 8  
Ομάδα 5 – Σημεία : 10 , 16 , 61 , 2  
Ομάδα 6 – Σημεία : 12 , 3 , 50 , 24  
Ομάδα 7 – Σημεία : 22 , 6 , 25 , 62  
Ομάδα 8 – Σημεία : 32 , 1 , 14 , 35  
Ομάδα 9 – Σημεία : 55 , 54 , 18 , 33  
Ομάδα 10 – Σημεία : 60 , 20 , 49 , 30  
Ομάδα 11 – Σημεία : 26 , 63 , 58 , 57  
Ομάδα 12 – Σημεία : 43 , 29 , 51 , 31  
Ομάδα 13 – Σημεία : 42 , 53 , 36 , 59  
Ομάδα 14 – Σημεία : 40 , 38 , 65 , 52  
Ομάδα 15 – Σημεία : 47 , 67 , 37 , 27  
Ομάδα 16 – Σημεία : 28 , 39 , 41 , 48

##### *Δεκαεπτά ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64 , 44  
Ομάδα 2 – Σημεία : 17 , 15 , 19 , 21  
Ομάδα 3 – Σημεία : 9 , 23 , 11 , 56  
Ομάδα 4 – Σημεία : 66 , 46 , 45 , 5  
Ομάδα 5 – Σημεία : 13 , 34 8 , 10  
Ομάδα 6 – Σημεία : 16 , 61 , 2 , 12  
Ομάδα 7 – Σημεία : 3 , 50 , 24 , 22  
Ομάδα 8 – Σημεία : 6 , 25 62 , 32  
Ομάδα 9 – Σημεία : 1 , 14 , 35 , 55  
Ομάδα 10 – Σημεία : 54 , 18 , 33 , 60  
Ομάδα 11 – Σημεία : 20 , 49 , 30 , 26  
Ομάδα 12 – Σημεία : 63 , 58 , 57 , 43  
Ομάδα 13 – Σημεία : 29 , 51 , 31 , 42  
Ομάδα 14 – Σημεία : 53 , 36 , 59 , 40  
Ομάδα 15 – Σημεία : 38 , 65 , 52 , 47  
Ομάδα 16 – Σημεία : 67 , 37 , 27 , 28  
Ομάδα 17 – Σημεία : 39 , 41 , 48

#### *Δεκαοχτώ ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64 , 44  
Ομάδα 2 – Σημεία : 17 , 15 , 19 , 21  
Ομάδα 3 – Σημεία : 9 , 23 , 11 , 56  
Ομάδα 4 – Σημεία : 66 , 46 , 45 , 5  
Ομάδα 5 – Σημεία : 13 , 34 , 8 , 10  
Ομάδα 6 – Σημεία : 16 , 61 , 2 , 12  
Ομάδα 7 – Σημεία : 3 , 50 , 24 , 22  
Ομάδα 8 – Σημεία : 6 , 25 , 62 , 32  
Ομάδα 9 – Σημεία : 1 , 14 , 35 , 55  
Ομάδα 10 – Σημεία : 54 , 18 , 33 , 60  
Ομάδα 11 – Σημεία : 20 , 49 , 30 , 26  
Ομάδα 12 – Σημεία : 63 , 58 , 57 , 43  
Ομάδα 13 – Σημεία : 29 , 51 , 31 , 42  
Ομάδα 14 – Σημεία : 53 , 36 , 59  
Ομάδα 15 – Σημεία : 40 , 38 , 65  
Ομάδα 16 – Σημεία : 52 , 47 , 67  
Ομάδα 17 – Σημεία : 37 , 27 , 28  
Ομάδα 18 – Σημεία : 39 , 41 , 48

#### *Δεκαεννιά ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64 , 44  
Ομάδα 2 – Σημεία : 17 , 15 , 19 , 21  
Ομάδα 3 – Σημεία : 9 , 23 , 11 , 56  
Ομάδα 4 – Σημεία : 66 , 46 , 45 , 5  
Ομάδα 5 – Σημεία : 13 , 34 , 8 , 10  
Ομάδα 6 – Σημεία : 16 , 61 , 2 , 12  
Ομάδα 7 – Σημεία : 3 , 50 , 24 , 22  
Ομάδα 8 – Σημεία : 6 , 25 , 62 , 32

Ομάδα 9 – Σημεία : 1 , 14 , 35 , 55  
Ομάδα 10 – Σημεία : 54 , 18 , 33 , 60  
Ομάδα 11 – Σημεία : 20 , 49 , 30  
Ομάδα 12 – Σημεία : 26 , 63 , 58  
Ομάδα 13 – Σημεία : 57 , 43 , 29  
Ομάδα 14 – Σημεία : 51 , 31 , 42  
Ομάδα 15 – Σημεία : 53 , 36 , 59  
Ομάδα 16 – Σημεία : 40 , 38 , 65  
Ομάδα 17 – Σημεία : 52 , 47 , 67  
Ομάδα 18 – Σημεία : 37 , 27 , 28  
Ομάδα 19 – Σημεία : 39 , 41 , 48

### *Είκοσι ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64 , 44  
Ομάδα 2 – Σημεία : 17 , 15 , 19 , 21  
Ομάδα 3 – Σημεία : 9 , 23 , 11 , 56  
Ομάδα 4 – Σημεία : 66 , 46 , 45 , 5  
Ομάδα 5 – Σημεία : 13 , 34 , 8 , 10  
Ομάδα 6 – Σημεία : 16 , 61 , 2 , 12  
Ομάδα 7 – Σημεία : 3 , 50 , 24 , 22  
Ομάδα 8 – Σημεία : 6 , 25 , 62  
Ομάδα 9 – Σημεία : 32 , 1 , 14  
Ομάδα 10 – Σημεία : 35 , 55 , 54  
Ομάδα 11 – Σημεία : 18 , 33 , 60  
Ομάδα 12 – Σημεία : 20 , 49 , 30  
Ομάδα 13 – Σημεία : 26 , 63 , 58  
Ομάδα 14 – Σημεία : 57 , 43 , 29  
Ομάδα 15 – Σημεία : 51 , 31 , 42  
Ομάδα 16 – Σημεία : 53 , 36 , 59  
Ομάδα 17 – Σημεία : 40 , 38 , 65  
Ομάδα 18 – Σημεία : 52 , 47 , 67  
Ομάδα 19 – Σημεία : 37 , 27 , 28  
Ομάδα 20 – Σημεία : 39 , 41 , 48

### *Εικοσιένα ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64 , 44  
Ομάδα 2 – Σημεία : 17 , 15 , 19 , 21  
Ομάδα 3 – Σημεία : 9 , 23 , 11 56  
Ομάδα 4 – Σημεία : 66 , 46 , 45 , 5  
Ομάδα 5 – Σημεία : 13 , 34 , 8  
Ομάδα 6 – Σημεία : 10 , 16 , 61  
Ομάδα 7 – Σημεία : 2 , 12 , 3  
Ομάδα 8 – Σημεία : 50 , 24 , 22  
Ομάδα 9 – Σημεία : 6 , 25 , 62

Ομάδα 10 – Σημεία : 32 , 1 , 14  
Ομάδα 11 – Σημεία : 35 , 55 , 54  
Ομάδα 12 – Σημεία : 18 , 33 , 60  
Ομάδα 13 – Σημεία : 20 , 49 , 30  
Ομάδα 14 – Σημεία : 26 , 63 , 58  
Ομάδα 15 – Σημεία : 57 , 43 , 29  
Ομάδα 16 – Σημεία : 51 , 31 , 42  
Ομάδα 17 – Σημεία : 53 , 36 , 59  
Ομάδα 18 – Σημεία : 40 , 38 , 65  
Ομάδα 19 – Σημεία : 52 , 47 , 67  
Ομάδα 20 – Σημεία : 37 , 27 , 28  
Ομάδα 21 – Σημεία : 39 , 41 , 48

### *Εικοσιδύο ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64 , 44  
Ομάδα 2 – Σημεία : 17 , 15 , 19  
Ομάδα 3 – Σημεία : 21 , 9 , 23  
Ομάδα 4 – Σημεία : 11 , 56 , 66  
Ομάδα 5 – Σημεία : 46 , 45 , 5  
Ομάδα 6 – Σημεία : 13 , 34 , 8  
Ομάδα 7 – Σημεία : 10 , 16 , 61  
Ομάδα 8 – Σημεία : 2 , 12 , 3  
Ομάδα 9 – Σημεία : 50 , 24 , 22  
Ομάδα 10 – Σημεία : 6 , 25 , 62  
Ομάδα 11 – Σημεία : 32 , 1 , 14  
Ομάδα 12 – Σημεία : 35 , 55 , 54  
Ομάδα 13 – Σημεία : 18 , 33 , 60  
Ομάδα 14 – Σημεία : 20 , 49 , 30  
Ομάδα 15 – Σημεία : 26 , 63 , 58  
Ομάδα 16 – Σημεία : 57 , 43 , 29  
Ομάδα 17 – Σημεία : 51 , 31 , 42  
Ομάδα 18 – Σημεία : 53 , 36 , 59  
Ομάδα 19 – Σημεία : 40 , 38 , 65  
Ομάδα 20 – Σημεία : 52 , 47 , 67  
Ομάδα 21 – Σημεία : 37 , 27 , 28  
Ομάδα 22 – Σημεία : 39 , 41 , 48

### *Εικοσιτρείς ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64  
Ομάδα 2 – Σημεία : 44 , 17 , 15  
Ομάδα 3 – Σημεία : 19 , 21 , 9  
Ομάδα 4 – Σημεία : 23 , 11 , 56  
Ομάδα 5 – Σημεία : 66 , 46 , 45  
Ομάδα 6 – Σημεία : 5 , 13 , 34

Ομάδα 7 – Σημεία : 8 , 10 , 16  
Ομάδα 8 – Σημεία : 61 , 2 , 12  
Ομάδα 9 – Σημεία : 3 , 50 , 24  
Ομάδα 10 – Σημεία : 22 , 6 , 25  
Ομάδα 11 – Σημεία : 62 , 32 , 1  
Ομάδα 12 – Σημεία : 14 , 35 , 55  
Ομάδα 13 – Σημεία : 54 , 18 , 33  
Ομάδα 14 – Σημεία : 60 , 20 , 49  
Ομάδα 15 – Σημεία : 30 , 26 , 63  
Ομάδα 16 – Σημεία : 58 , 57 , 43  
Ομάδα 17 – Σημεία : 29 , 51 , 31  
Ομάδα 18 – Σημεία : 42 , 53 , 36  
Ομάδα 19 – Σημεία : 59 , 40 , 38  
Ομάδα 20 – Σημεία : 65 , 52 , 47  
Ομάδα 21 – Σημεία : 67 , 37 , 27  
Ομάδα 22 – Σημεία : 28 , 39  
Ομάδα 23 – Σημεία : 41 , 48

#### *Εικοσιτέσσερις ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64  
Ομάδα 2 – Σημεία : 44 , 17 , 15  
Ομάδα 3 – Σημεία : 19 , 21 , 9  
Ομάδα 4 – Σημεία : 23 , 11 , 56  
Ομάδα 5 – Σημεία : 66 , 46 , 45  
Ομάδα 6 – Σημεία : 5 , 13 , 34  
Ομάδα 7 – Σημεία : 8 , 10 , 16  
Ομάδα 8 – Σημεία : 61 , 2 , 12  
Ομάδα 9 – Σημεία : 3 , 50 , 24  
Ομάδα 10 – Σημεία : 22 , 6 , 25  
Ομάδα 11 – Σημεία : 62 , 32 , 1  
Ομάδα 12 – Σημεία : 14 , 35 , 55  
Ομάδα 13 – Σημεία : 54 , 18 , 33  
Ομάδα 14 – Σημεία : 60 , 20 , 49  
Ομάδα 15 – Σημεία : 30 , 26 , 63  
Ομάδα 16 – Σημεία : 58 , 57 , 43  
Ομάδα 17 – Σημεία : 29 , 51 , 31  
Ομάδα 18 – Σημεία : 42 , 53 , 36  
Ομάδα 19 – Σημεία : 59 , 40 , 38  
Ομάδα 20 – Σημεία : 65 , 52  
Ομάδα 21 – Σημεία : 47 , 67  
Ομάδα 22 – Σημεία : 37 , 27  
Ομάδα 23 – Σημεία : 28 , 39  
Ομάδα 24 – Σημεία : 41 , 48

#### *Εικοσιπέντε ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64  
Ομάδα 2 – Σημεία : 44 , 17 , 15



Ομάδα 3 – Σημεία : 19 , 21 , 9  
Ομάδα 4 – Σημεία : 23 , 11 , 56  
Ομάδα 5 – Σημεία : 66 , 46 , 45  
Ομάδα 6 – Σημεία : 5 , 13 , 34  
Ομάδα 7 – Σημεία : 8 , 10 , 16  
Ομάδα 8 – Σημεία : 61 , 2 , 12  
Ομάδα 9 – Σημεία : 3 , 50 , 24  
Ομάδα 10 – Σημεία : 22 , 6 , 25  
Ομάδα 11 – Σημεία : 62 , 32 , 1  
Ομάδα 12 – Σημεία : 14 , 35 , 55  
Ομάδα 13 – Σημεία : 54 , 18 , 33  
Ομάδα 14 – Σημεία : 60 , 20 , 49  
Ομάδα 15 – Σημεία : 30 , 26 , 63  
Ομάδα 16 – Σημεία : 58 , 57 , 43  
Ομάδα 17 – Σημεία : 29 , 51 , 31  
Ομάδα 18 – Σημεία : 42 , 53  
Ομάδα 19 – Σημεία : 36 , 59  
Ομάδα 20 – Σημεία : 40 , 38  
Ομάδα 21 – Σημεία : 65 , 52  
Ομάδα 22 – Σημεία : 47 , 67  
Ομάδα 23 – Σημεία : 37 , 27  
Ομάδα 24 – Σημεία : 28 , 39  
Ομάδα 25 – Σημεία : 41 , 48

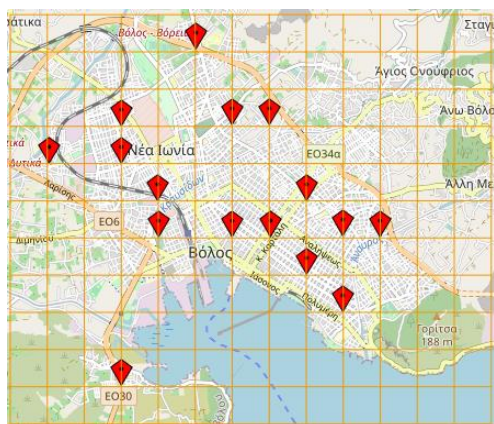
### *Εικοσιέξι ομάδες σημείων*

Ομάδα 1 – Σημεία : 7 , 4 , 64  
Ομάδα 2 – Σημεία : 44 , 17 , 15  
Ομάδα 3 – Σημεία : 19 , 21 , 9  
Ομάδα 4 – Σημεία : 23 , 11 , 56  
Ομάδα 5 – Σημεία : 66 , 46 , 45  
Ομάδα 6 – Σημεία : 5 , 13 , 34  
Ομάδα 7 – Σημεία : 8 , 10 , 16  
Ομάδα 8 – Σημεία : 61 , 2 , 12  
Ομάδα 9 – Σημεία : 3 , 50 , 24  
Ομάδα 10 – Σημεία : 22 , 6 , 25  
Ομάδα 11 – Σημεία : 62 , 32 , 1  
Ομάδα 12 – Σημεία : 14 , 35 , 55  
Ομάδα 13 – Σημεία : 54 , 18 , 33  
Ομάδα 14 – Σημεία : 60 , 20 , 49  
Ομάδα 15 – Σημεία : 30 , 26 , 63  
Ομάδα 16 – Σημεία : 58 , 57  
Ομάδα 17 – Σημεία : 43 , 29  
Ομάδα 18 – Σημεία : 51 , 31  
Ομάδα 19 – Σημεία : 42 , 53  
Ομάδα 20 – Σημεία : 36 , 59

- Ομάδα 21 – Σημεία : 40 , 38
- Ομάδα 22 – Σημεία : 65 , 52
- Ομάδα 23 – Σημεία : 47 , 67
- Ομάδα 24 – Σημεία : 37 , 27
- Ομάδα 25 – Σημεία : 28 , 39
- Ομάδα 26 – Σημεία : 41 , 48

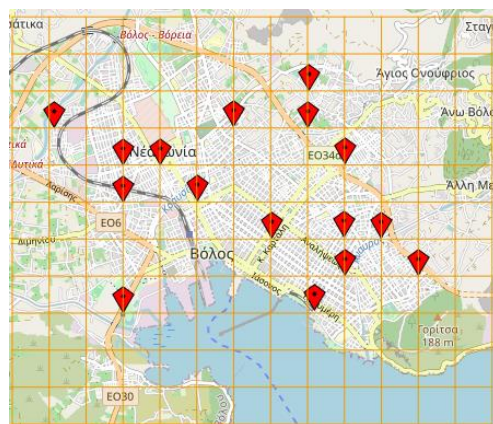
#### 4.4 Αποτελέσματα βελτιστοποίησης

Δεκαέξι μετρητές - Κατηγοριοποίηση 1



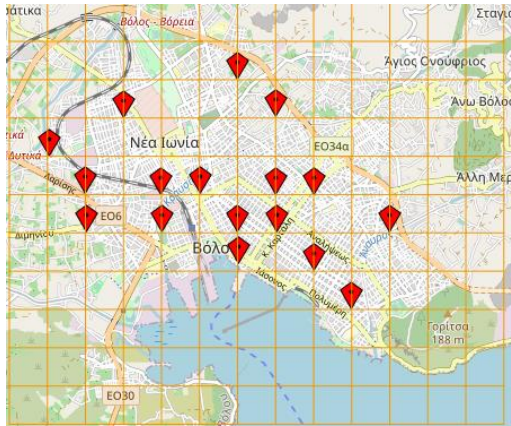
Βέλτιστα σημεία	1,7,13,23,28,31,37,39, 41,44,45,48,58,59,63, 64
Αντικειμενική συνάρτηση	622,784
Κάλυψη	58,209%
Υπερκάλυψη	14,9254%

Δεκαέξι μετρητές - Κατηγοριοποίηση 2



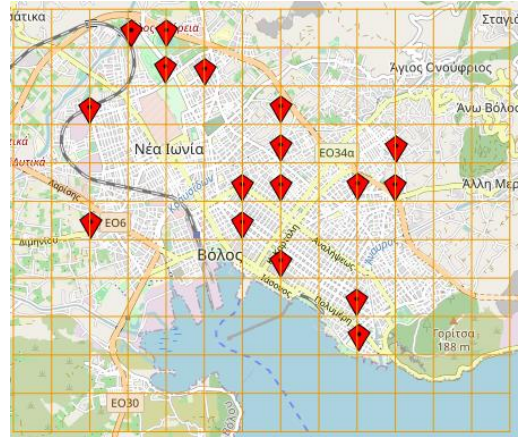
Βέλτιστα σημεία	2,8,17,23,24,27,30,31, 33,35,41,45,52,57,59, 62
Αντικειμενική συνάρτηση	653,327
Κάλυψη	55,2239%
Υπερκάλυψη	16,4179%

### Δεκαεπτά μετρητές - Κατηγοριοποίηση 1



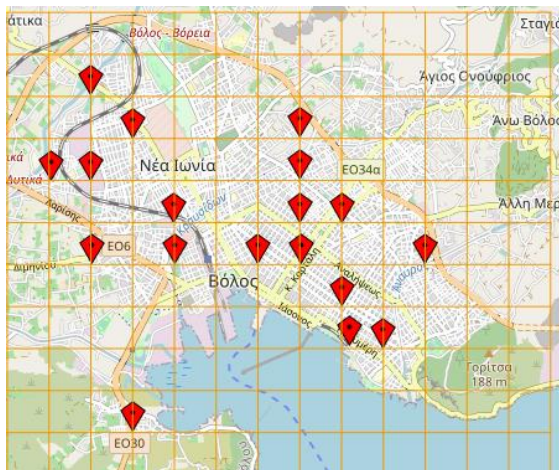
Βέλτιστα σημεία	5,6,7,15,23,28,37,39,41,42,44,48,49,52,58,63,64
Αντικειμενική συνάρτηση	651,269
Κάλυψη	56,7164%
Υπερκάλυψη	20,8955%

### Δεκαεπτά μετρητές -Κατηγοριοποίηση 2

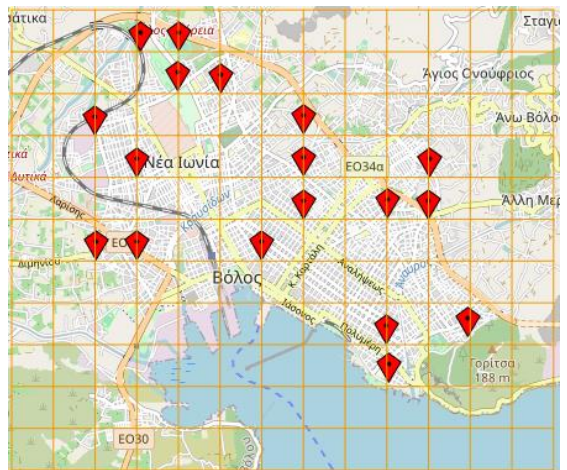


Βέλτιστα σημεία	5,10,12,21,22,26,28,32,40,42,43,44,47,48,55,56,60
Αντικειμενική συνάρτηση	642,644
Κάλυψη	53,7313%
Υπερκάλυψη	17,9104%

### Δεκαοχτώ μετρητές - Κατηγοριοποίηση 1



### Δεκαοχτώ μετρητές - Κατηγοριοποίηση 2

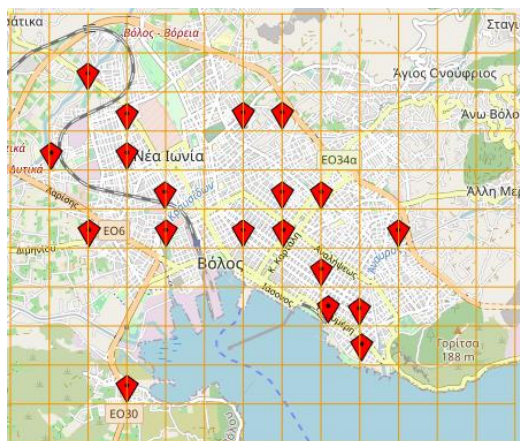




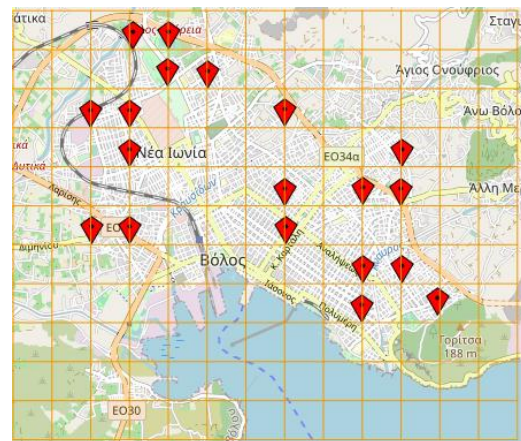
Βέλτιστα σημεία	1,5,7,9,23,27,28,37,39,41,42,43,44,48,58,61,63,64
Αντικειμενική συνάρτηση	696,463
Κάλυψη	55,2239%
Υπερκάλυψη	23,8806%

Βέλτιστα σημεία	5,10,12,21,22,26,28,32,42,43,44,48,55,56,59,60,65,67
Αντικειμενική συνάρτηση	692,448
Κάλυψη	61,194%
Υπερκάλυψη	20,8955%

### Δεκαεννία μετρητές - Κατηγοριοποίηση 1



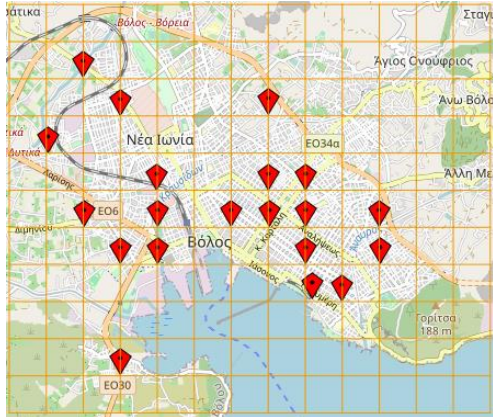
### Δεκαεννία μετρητές - Κατηγοριοποίηση 2



Βέλτιστα σημεία	1,5,7,9,23,26,27,28,37,39,41,42,44,45,48,58,59,63,64
Αντικειμενική συνάρτηση	733,431
Κάλυψη	55,2239%
Υπερκάλυψη	25,3731%

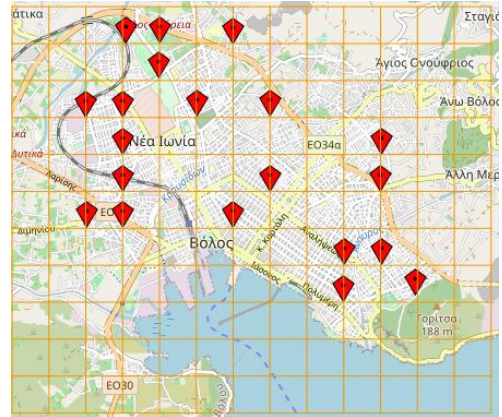
Βέλτιστα σημεία	5,10,12,21,22,28,29,30,32,41,42,44,55,56,58,59,60,65,67
Αντικειμενική συνάρτηση	775,239
Κάλυψη	59,7015%
Υπερκάλυψη	26,8657%

### Είκοσι μετρητές - Κατηγοριοποίηση 1



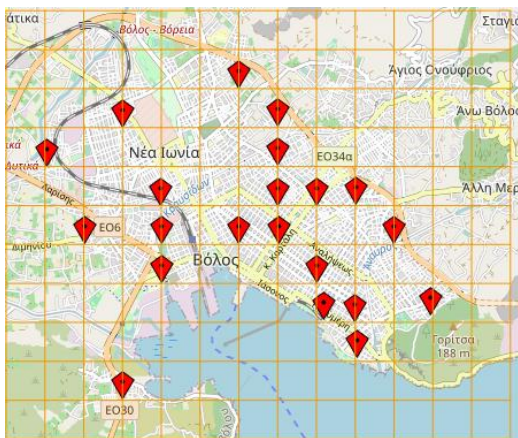
Βέλτιστα σημεία	1,3,4,5,7,9,23,27,28,29,37,38,39,41,42,44,48,58,63,64
Αντικειμενική συνάρτηση	761,769
Κάλυψη	58,209%
Υπερκάλυψη	22,3881%

### Είκοσι μετρητές - Κατηγοριοποίηση 2

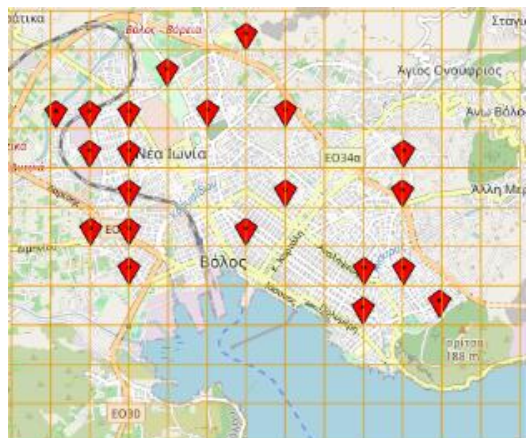


Βέλτιστα σημεία	5,10,12,14,21,22,28,29,30,42,44,48,54,56,58,59,60,62,65,67
Αντικειμενική συνάρτηση	768,395
Κάλυψη	68,6567%
Υπερκάλυψη	25,3731%

### Εικοσιένιας μετρητές - Κατηγοριοποίηση 1



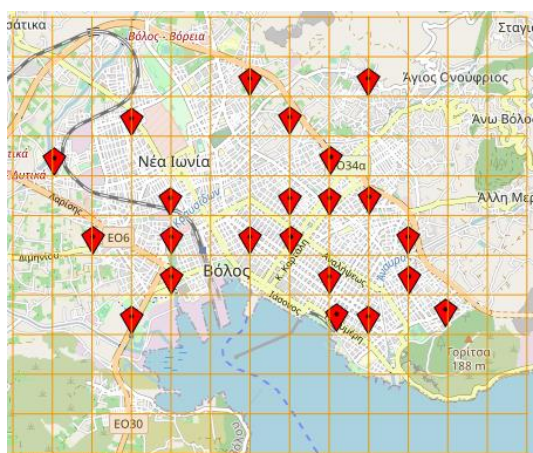
### Εικοσιένιας μετρητές - Κατηγοριοποίηση 2



Βέλτιστα σημεία	1,4,5,7,15,23,26,27,28,32,37,39,41,42,43,44,48,58,63,64,67
Αντικειμενική συνάρτηση	799.067
Κάλυψη	68.6567%
Υπερκάλυψη	22.3881%

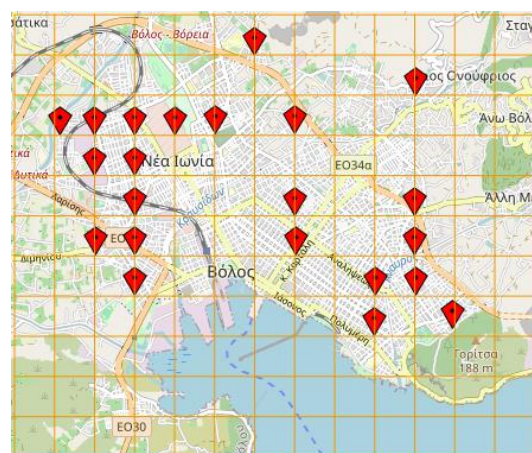
Βέλτιστα σημεία	3,5,8,14,21,22,28,29,30,42,44,48,54,56,58,59,60,61,62,65,67
Αντικειμενική συνάρτηση	800,058
Κάλυψη	71,6418%
Υπερκάλυψη	22,3881%

### Εικοσιδύο μετρητές - Κατηγοριοποίηση 1



Βέλτιστα σημεία	2,4,5,7,15,18,23,27,28,29,32,36,37,39,41,42,44,48,58,63,64,67
Αντικειμενική συνάρτηση	845,809
Κάλυψη	73,1343%
Υπερκάλυψη	28,3582%

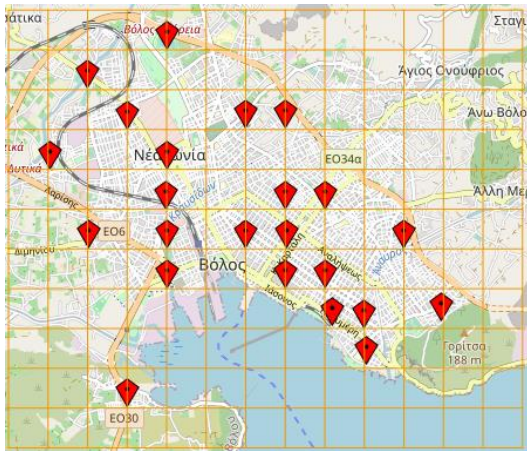
### Εικοσιδύο μετρητές - Κατηγοριοποίηση 2



Βέλτιστα σημεία	3,5,8,14,19,22,23,28,29,30,41,42,44,54,58,59,60,61,62,65,66,67
Αντικειμενική συνάρτηση	871,395
Κάλυψη	67,1642%
Υπερκάλυψη	26,8657%

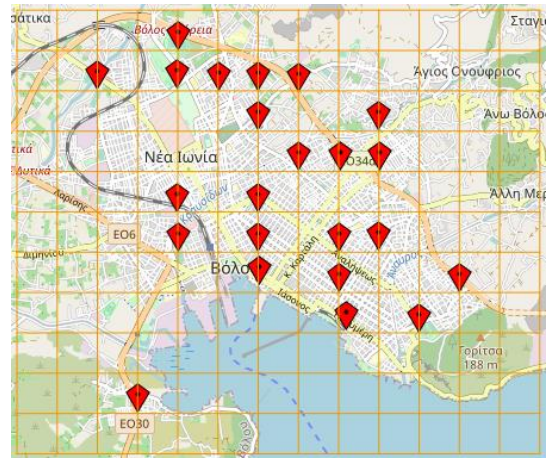


### Εικοσιτρείς μετρητές - Κατηγοριοποίηση 1



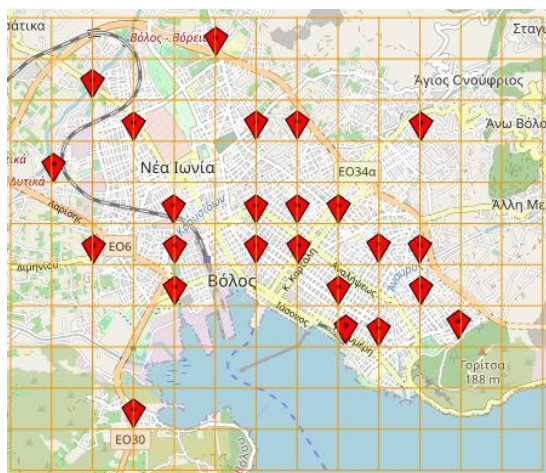
Βέλτιστα σημεία	1,4,5,9,12,23,26,27,28,37,39,40,41,42,44,45,48,57,58,63,64,67,77
Αντικειμενική συνάρτηση	874,602
Κάλυψη	68,6567%
Υπερκάλυψη	25,3731%

### Εικοσιτρείς μετρητές - Κατηγοριοποίηση 2

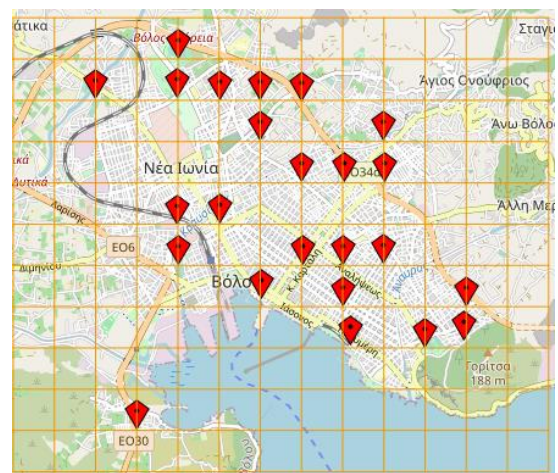


Βέλτιστα σημεία	1,9,12,15,16,24,25,27,31,33,34,36,38,39,43,45,47,48,49,55,56,63,64
Αντικειμενική συνάρτηση	869,104
Κάλυψη	68,6567%
Υπερκάλυψη	31,3433%

### Εικοσιτέσσερις μετρητές - Κατηγοριοποίηση 1



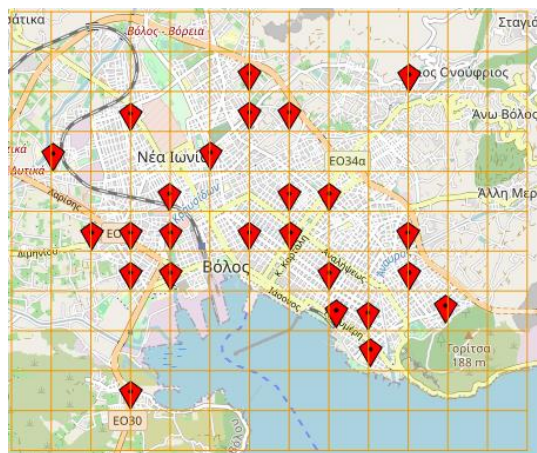
### Εικοσιτέσσερις μετρητές - Κατηγοριοποίηση 2



Βέλτιστα σημεία	1,4,5,7,9,13,20,23,27,28,29,31,37,39,41,42,44,45,47,48,58,63,64,67
954,978	920,555
Κάλυψη	70,1493%
Υπερκάλυψη	25,3731%

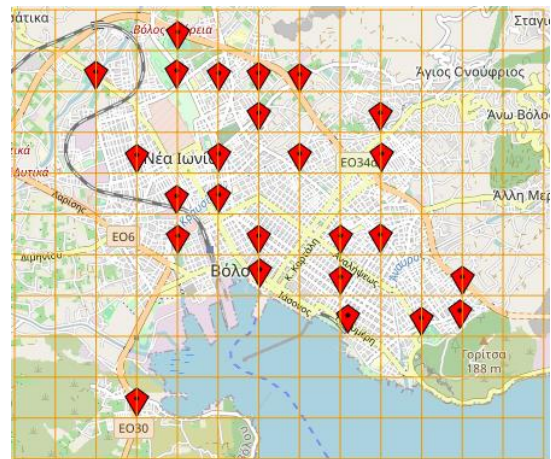
Βέλτιστα σημεία	1,9,12,15,16,24,25,27,31,33,34,36,38,39,41,43,45,49,52,55,56,63,64,67
Αντικειμενική συνάρτηση	973,181
Κάλυψη	68,6567%
Υπερκάλυψη	31,3433%

### Εικοσιπέντε μετρητές - Κατηγοριοποίηση 1



Βέλτιστα σημεία	7,4,64,44,15,19,23,45,5,3,1,26,63,58,29,42,53,37,65,67,27,28,39,41,48
Αντικειμενική συνάρτηση	946,635
Κάλυψη	70,1493%
Υπερκάλυψη	35,8209%

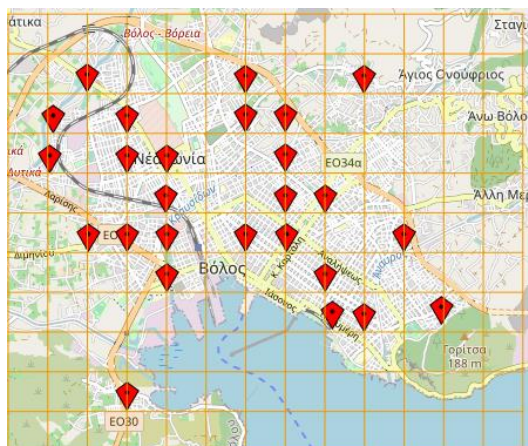
### Εικοσιπέντε μετρητές - Κατηγοριοποίηση 2



Βέλτιστα σημεία	1,9,12,15,16,24,25,27,31,33,34,38,39,43,45,48,49,52,53,55,56,59,63,64,67
Αντικειμενική συνάρτηση	967,748
Κάλυψη	74,6269%
Υπερκάλυψη	34,3284%

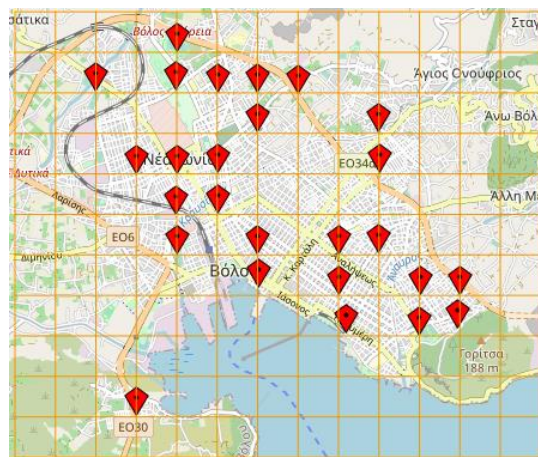


## Εικοσιέξι μετρητές - Κατηγοριοποίηση 1



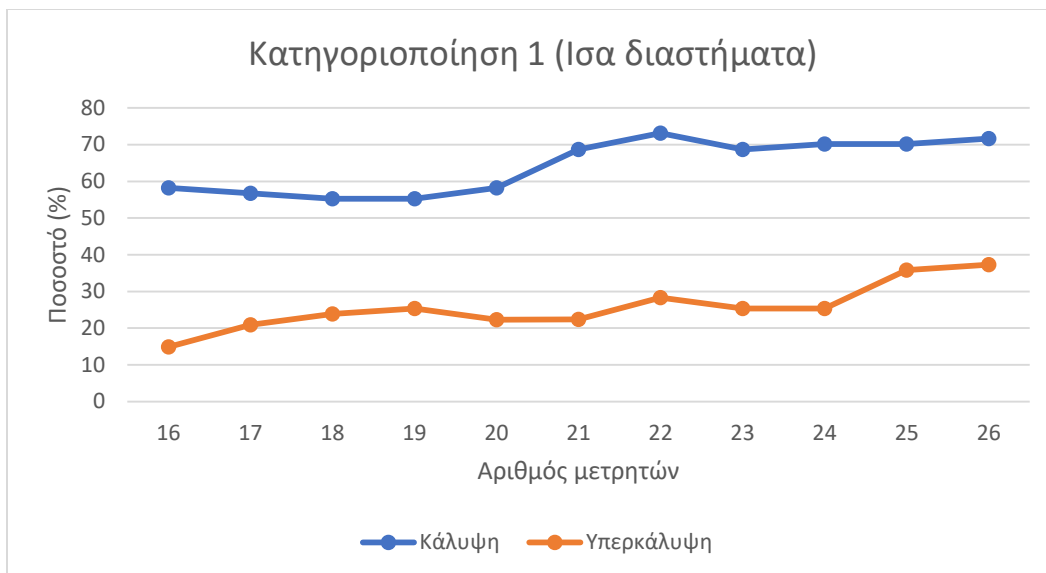
Βέλτιστα σημεία	1,4,5,7,8,9,15,18,27,28,37,39,41,43,44,45,48,52,53,57,58,59,63,64,65,67
Αντικειμενική συνάρτηση	989,628
Κάλυψη	71,6418%
Υπερκάλυψη	37,3134%

## Εικοσιέξι μετρητές - Κατηγοριοποίηση 2

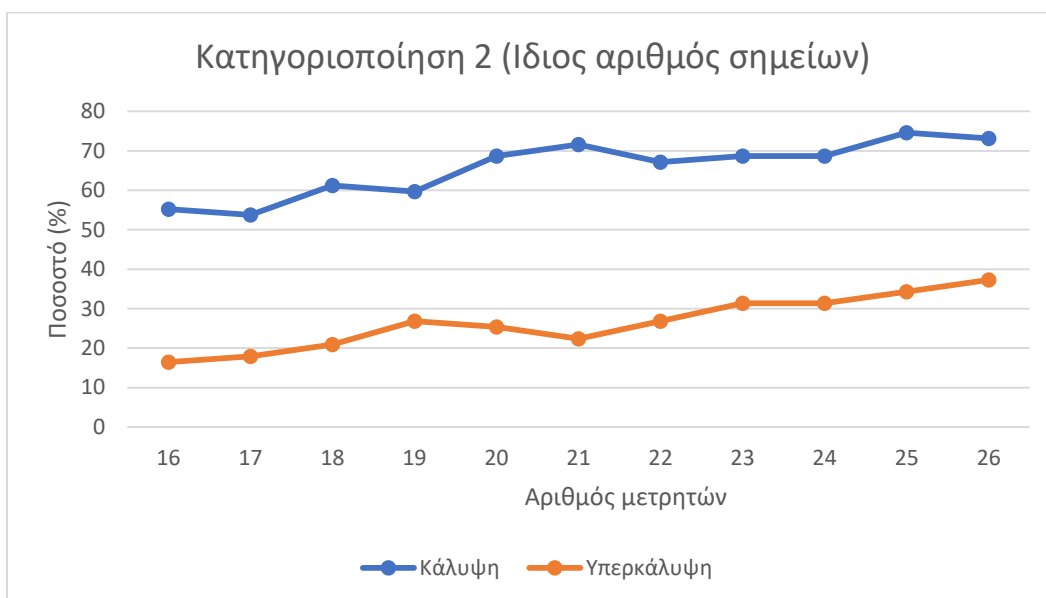


Βέλτιστα σημεία	1,9,12,15,16,24,25,27,29,31,33,34,38,39,45,48,49,52,53,55,56,57,59,63,64,67
Αντικειμενική συνάρτηση	1012,61
Κάλυψη	73,1343%
Υπερκάλυψη	37,3134%

Παρατηρώντας τα σημεία κατασκευής των μετρητών για τις δύο κατηγοριοποιήσεις στον ίδιο αριθμό σημείων είναι εμφανές ότι για την κατηγοριοποίηση 1 τα σημεία τοποθετούνται κατά μήκος κεντρικών δρόμων και κοντά στο κέντρο. Ενώ για την κατηγοριοποίηση 2 τα σημεία κατανέμονται στα άκρα της περιοχής του Βόλου. Η κατανομή αυτή στα άκρα είναι πιθανόν να οφείλεται στην βαρύτητα που έχουν τα σημεία υψηλής ρύπανσης του κριτηρίου 5 καθώς και του κριτηρίου 3 του όγκου κυκλοφορίας για τις κύριες αρτηρίες και τις ελεύθερες λεωφόρους και σε συνδυασμό με την κατανομή των σημείων σε ομάδες με τον ίδιο αριθμό σημείων παρατηρείται αυτό το μοτίβο. Η βέλτιστη τιμή παρατηρείται για 21 μετρητές για την κατηγοριοποίηση 2 με κάλυψη 71,6418 % και υπερκάλυψη 22,3881 %. Ενώ για την κατηγοριοποίηση 1 ο βέλτιστος αριθμός μετρητών είναι στα 22 σημεία με κάλυψη 73,1343 % και υπερκάλυψη 28,3582 %. Ελέγχοντας τις τιμές για μεγαλύτερο αριθμό μετρητών για τις δύο κατηγοριοποιήσεις και μέσω των παρακάτω διαγραμμάτων παρατηρείται ότι το ποσοστό κάλυψης δεν παρουσιάζει μεγάλες αυξήσεις σε αντίθεση με την υπερκάλυψη. Αυτό οδηγεί στην μεγάλη αύξηση του κόστους για μικρή αύξηση στην υπερκάλυψη.



Γράφημα 2:Αποτελέσματα κατηγοριοποίησης 1



Γράφημα 3:Αποτελέσματα κατηγοριοποίησης 2

## Συμπεράσματα

Η ποιότητα του αέρα μεταβάλλεται λόγω μιας πληθώρας παραγόντων. Τόσο ο πληθυσμός της περιοχής όσο και οι πηγές ρύπανσης της μπορούν να έχουν μεγάλο αντίκτυπο στο κλίμα μίας περιοχής. Στην εμφάνιση αυτών των ρύπων συμβάλλει η συνεχής ανάπτυξη της τεχνολογίας, η εισροή του πληθυσμού στα αστικά κέντρα αλλά και η εκβιομηχάνιση. Μέσω της αλλαγής αυτής προκαλούνται αρνητικές επιπτώσεις τόσο στο περιβάλλον όσο και στην υγεία των ανθρώπων οδηγώντας στην υποβάθμιση της ποιότητας της καθημερινής ζωής. Είναι εμφανές ότι η συνεχής μελέτη των αέριων ρύπων είναι απαραίτητη. Για τον λόγο αυτό η κατασκευή μετρητών και η τοποθέτηση τους στα βέλτιστα σημεία έχει μεγάλη βαρύτητα. Έτσι θα επιτευχθεί ο καλύτερος δυνατός έλεγχος των αέριων ρύπων με τελικό στόχο την μείωση αυτών.

Το πρόγραμμα αναπτύχθηκε σε περιβάλλον Java και το μαθηματικό μοντέλο σε C++. Στόχος είναι η διευκόλυνση της συλλογής των δεδομένων που απαιτούνται για την εύρεση σημείων κατασκευής μετρητών αέριων ρύπων και η εύρεση της βέλτιστης λύσης. Για την εύρεση των βέλτιστων λύσεων έγινε η σύγκριση δύο διαφορετικών κατηγοριοποιήσεων με σκοπό την τοποθέτηση των μετρητών στις πιο αντιπροσωπευτικές περιοχές, διατηρώντας την μέγιστη δυνατή κάλυψη και ελαχιστοποιώντας την υπερκάλυψη καθώς και των αριθμό των μετρητών. Οι κατηγοριοποιήσεις σε ομάδες έγιναν βάση των συντελεστών βαρύτητας της ρύπανσης κάθε σημείου. Η εφαρμογή κατασκευάστηκε ώστε να μπορεί να εφαρμοστεί σε οποιαδήποτε περιοχή παγκοσμίως. Σε αυτό συμβάλλει η δημιουργία του κριτηρίου των πηγών ρύπων, καθώς και η ομοιομορφία κλιμάκων των τιμών των κριτηρίων για την καλύτερη σύγκριση των σημείων. Το πλήθος των πηγών ρύπων ορίζεται από τον χρήστη ώστε να προσαρμόζεται στην περιοχή ελέγχου.

Από την εφαρμογή της βελτιστοποίησης για τις δύο μεθόδους κατηγοριοποίησης παρατηρήθηκε ότι η κατηγοριοποίηση ίσων διαστημάτων (κατηγοριοποίηση 1) τοποθετεί τα σημεία κοντά στο κέντρο. Αντιθέτως, η κατηγοριοποίηση ίδιου αριθμού σημείων σε κάθε ομάδα (κατηγοριοποίηση 2) τείνει να τοποθετεί τα σημεία στα άκρα της περιοχής. Αυτό ενδεχομένως συμβαίνει εξαιτίας της υψηλής ρύπανσης που προκαλείται από τις κύριες αρτηρίες και λεωφόρους αλλά και τις βιομηχανίες που βρίσκονται στα άκρα της περιοχής.

Σε αυτό το σημείο αναφέρονται οι περιορισμοί της παρούσας εργασίας. Στα πλαίσια της επίδρασης του ανέμου πρέπει να σημειωθεί ότι λήφθηκε υπόψιν στους υπολογισμούς μόνο η επίδραση της κατεύθυνσης του ανέμου αγνοώντας την ένταση του. Η ένταση του αέρα έχει άμεσο αντίκτυπο στην ταχύτητα διασποράς των αέριων ρύπων και η μελέτη του ακριβή μηχανισμού λειτουργίας θα μπορέσει να οδηγήσει σε ακόμα πιο ακριβή αποτελέσματα. Ένας ακόμα περιορισμός της εφαρμογής είναι η έλλειψη κριτηρίων για την παρουσία ακραίων καιρικών φαινομένων (ανεμοθύελλες, όξινη βροχή) ή άλλων αστάθμητων παραγόντων (πόλεμοι, πυρκαγιές, καταστροφή εργοστασίων). Τέτοιου είδους φαινόμενα επηρεάζουν σε μεγάλο βαθμό τους αέριους ρύπους προκαλώντας μεγάλες αποκλίσεις στις μετρήσεις. Επιπλέον, η εφαρμογή μπορεί να βελτιωθεί με την δημιουργία αναλυτικότερων κριτηρίων. Τα κριτήρια αυτά μπορούν να μεταβάλλονται σε βάθος χρόνου ή να έχουν κάποια εποχικότητα. Τέλος, η μελέτη και η δοκιμή διαφορετικών μοντέλων βελτιστοποίησης και κατηγοριοποιήσεων μπορεί επίσης να συμβάλλει στην καλύτερη αξιολόγηση των αποτελεσμάτων και πιθανώς σε καλύτερες λύσεις.

## ΠΑΡΑΡΤΗΜΑ

### Εγκατάσταση και εκτέλεση εφαρμογής

#### **Εγκατάσταση προγραμμάτων**

Αρχικά απαιτείται η εγκατάσταση της java έκδοσης 11 από τον παρακάτω σύνδεσμο([techspot.com/downloads/5553-java-jdk.html](http://techspot.com/downloads/5553-java-jdk.html)).

Επόμενο βήμα είναι η εγκατάσταση του προγράμματος Apache Netbeans (η εφαρμογή δημιουργήθηκε με την έκδοση 12.0) από το παρακάτω σύνδεσμο ( <https://netbeans.apache.org/download>) ή αναζητώντας στο διαδίκτυο το πρόγραμμα. Μετά την λήψη του αρχείου εγκατάστασης για την συσκευή (windows ,linux ,macos) ο χρήστης πρέπει να εκτελέσει το πρόγραμμα.

Το δεύτερο πρόγραμμα που απαιτείται να εγκατασταθεί είναι το Visual Studio ( η εφαρμογή δημιουργήθηκε με την έκδοση Visual Studio 2017) από το παρακάτω σύνδεσμο (<https://visualstudio.microsoft.com/vs/older-downloads/>) γίνεται η λήψη και η εκτέλεση του αρχείου (απαιτείται λογαριασμός Microsoft ή μέσω του email της σχολής).

Επόμενο βήμα είναι η λήψη του προγράμματος βελτιστοποίησης Cplex από τον παρακάτω σύνδεσμο(<http://ibm.biz/CPLEXonAI>) και μόλις συνδεθεί ο χρήστης με τον λογαριασμό του κατεβαίνει στη σελίδα και στο Software βρίσκουμε την καρτέλα ILOG Cplex Optimization Studio. Πατώντας στην καρτέλα ανοίγει ένα παράθυρο και πατώντας το Download μεταβαίνει στη νέα σελίδα. Στα Search options πατώντας το text και γράφοντας «IBM ILOG CPLEX Optimization Studio Multiplatform Multilingual 12.9 eAssembly» εμφανίζονται τα αρχεία ανάλογα το λογισμικό του υπολογιστή(windows,linux). Για την εγκατάσταση του στο visual studio στο παρακάτω σύνδεσμο παρουσιάζεται ένας οδηγός(<https://www.leandro-coelho.com/how-to-download-install-ibm-ilog-cplex-optimization-studio-windows/>).

Τέλος για την δημιουργία της εφαρμογής(web application) στο netbeans απαιτείται η χρήση διακομιστή(server) μέσω του apache tomcat από τον παρακάτω σύνδεσμο ([tomcat.apache.org/download-90.cgi](http://tomcat.apache.org/download-90.cgi)). Μετά την λήψη του αρχείου zip ακολουθεί η αποσυμπίεση του.

#### **Apache Netbeans**

Πρώτο βήμα είναι η εκτέλεση του προγράμματος Apache Netbeans. Στη συνέχεια γίνεται η δημιουργία της εφαρμογής. Πατώντας File > New Project > Java with Ant > Java Web > Web application> Next. Σε αυτό το σημείο ο χρήστης πρέπει να κατεβάσει και να κάνει λήψη και εγκατάσταση της javaFX πατώντας στο κουμπί «Download and activate». Σε περίπτωση που η εγκατάσταση έχει γίνει ήδη αυτό το βήμα παραβλέπεται. Επόμενο βήμα είναι η ονομασία της εφαρμογής καθώς και ο ορισμός της τοποθεσίας(Project Location) και του φακέλου του(Project Folder). Μέσω του πλήκτρου «Next» γίνεται η μετάβαση στην επομενη σελίδα όπου πρέπει να οριστεί ο διακομιστής που θα χρησιμοποιήσουμε. Πατώντας «Add» εμφανίζεται ένα παράθυρο όπου επιλέγουμε το είδος του διακομιστή Apache Tomcat or TomEE και στη συνέχεια «Next». Τέλος ο χρήστης με το κουμπί «Browse» αναζητά τον φάκελο του διακομιστή που

κατέβασε προηγουμένως (apache-tomcat-9.0.65) , ορίζει όνομα χρήστη(username) και κωδικό(password), κάνει ‘τικ’ το κουτάκι (Create user if it does not exist) και παταεί το κουμπί «Finish».

Αφού δημιουργηθεί η εφαρμογή ο χρήστης εισάγει όλα τα αρχεία html που αναφέρονται παρακάτω χωρίς να αλλάξει την ονομασία των αρχείων αντιγράφοντας τον κώδικα.

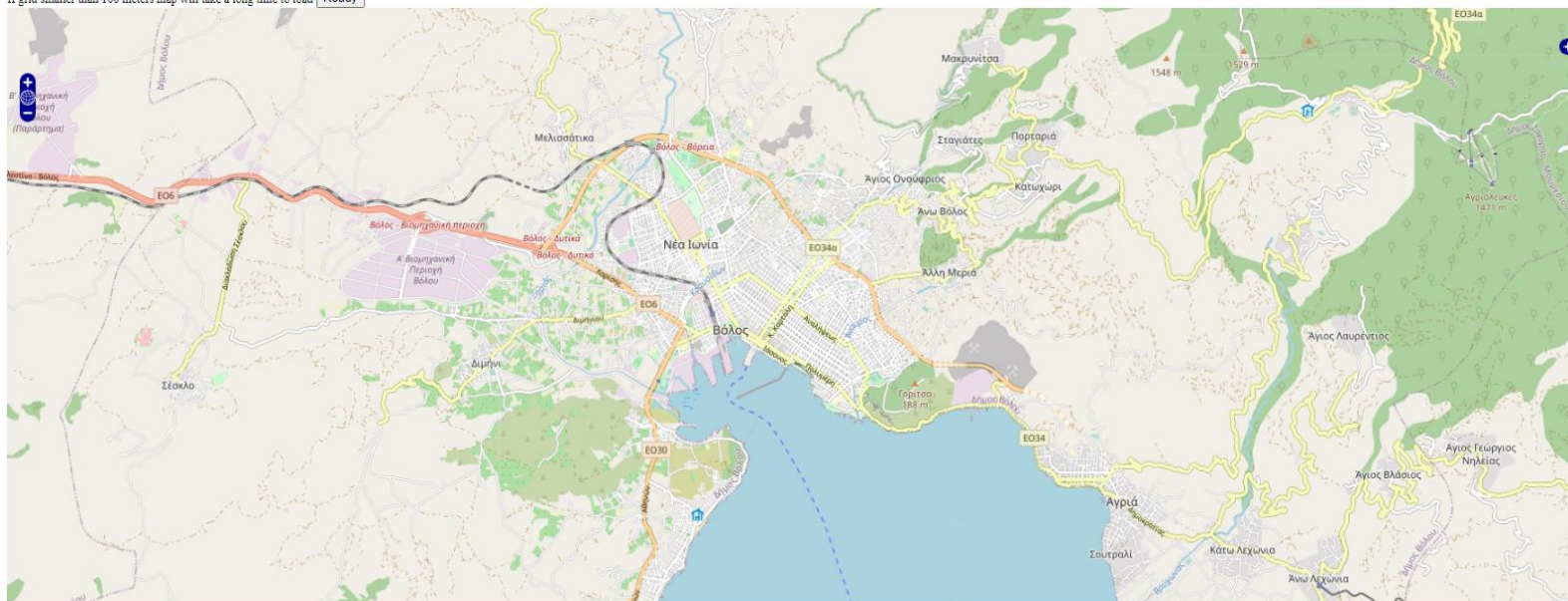
Στην πρώτη καρτέλα τοποθετούνται οι γεωγραφικές συντεταγμένες του κέντρου όπου θα αναζητηθούν τα βέλτιστα σημεία κατασκευής μετρητών ποιότητας των αέριων ρύπων και στη συνέχεια πατώντας το κουμπί «Save» και στη συνέχεια το κουμπί «Continue» γίνεται η μετάβαση στην επόμενη σελίδα.

## Optimal points for pollution recorders

Latitude:   
Longitude:   
   
Created by Charalampos Zouloumis

Στην δεύτερη σελίδα θα οριστεί η περιοχή μέσα στην οποία θα κατασκευαστούν οι μετρητές. Στο Grid distance ορίζεται η απόσταση μεταξύ των διπλανών σημείων του πλέγματος και στη συνέχεια κάνοντας κλικ σε δύο σημεία στον χάρτη τα οποία θα είναι οι δύο αντίθετες γωνίες ενός ορθογωνίου όπου θα σχηματιστεί το πλέγμα και στη συνέχεια πατώντας το κουμπί «Ready» ο χρήστης οδηγείται στο επόμενο βήμα.

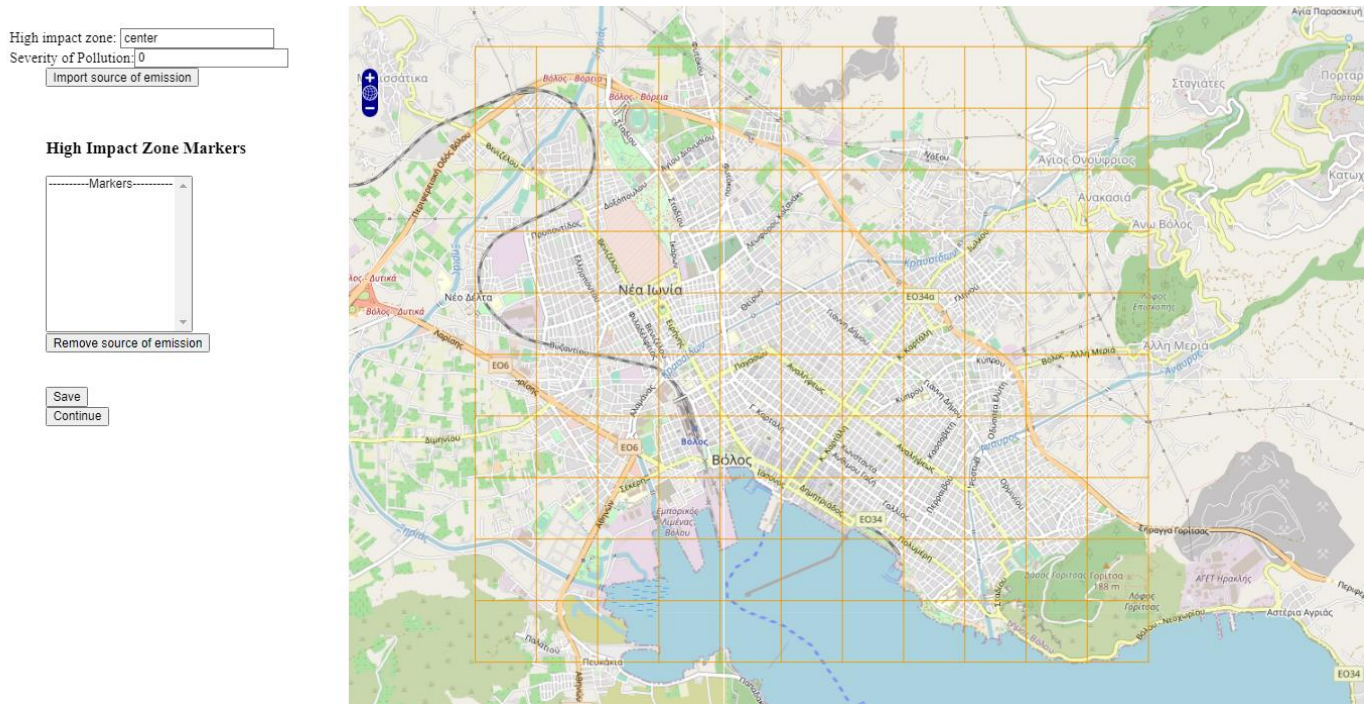
Grid distance(meters):   
If grid smaller than 100 meters map will take a long time to load



Στην τρίτη σελίδα τοποθετούνται τα σημεία πηγών ρύπανσης. Στο “High impact zone” δίνεται ένα όνομα για την πηγή ρύπανσης και στο “Severity of pollution”, ορίζεται η τιμή του σημείου αυτού βάση του κριτηρίου 5 που αναφέρθηκε προηγουμένως ,μόλις ο χρήστης επιλέξει το σημείο στο χάρτη και πατάει το κουμπί «Import source of emission» για να εισάγει το σημείο και για την αφαίρεση ενός σημείου «Remove source of emission» αφού επιλέξουμε το σημείο που θέλει στην λίστα.



Μόλις εισαχθούν όλα τα σημεία, πατώντας το κουμπί «Save» γίνεται η αποθήκευση των δεδομένων και στη συνέχεια «Continue».



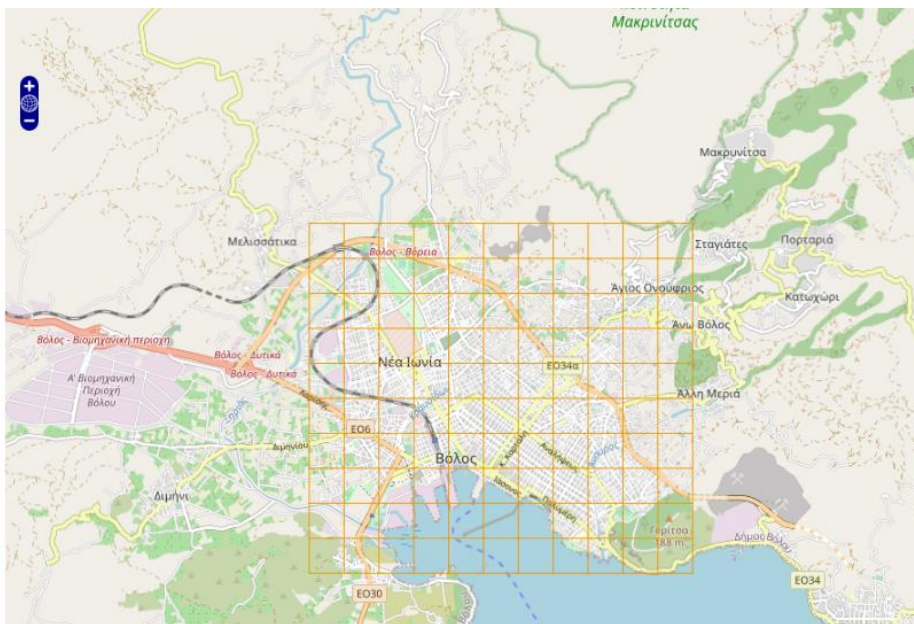
Ακολουθεί η εισαγωγή των πιθανών σημείων κατασκευής των μετρητών. Στο “Point of interest” ορίζεται η διεύθυνση του σημείου (Προσοχή στη διεύθυνση δεν πρέπει να υπάρχουν κενά μεταξύ λέξεων ή αριθμών, μπορεί να γίνει η χρήση της κατω παύλας “\_” και στο “Point coverage” ορίζεται η εμβέλεια που θέλουμε να έχουν οι μετρητές(σε μέτρα). Στη συνέχεια επιλέγοντας το σημείο στο χάρτη και πατώντας το κουμπί «Add point of interest» εισάγεται το σημείο και «Remove point of interest» για την αφαίρεση του.

Μόλις εισαχθούν όλα τα σημεία πατώντας το κουμπί «Save» γίνεται αποθήκευση και λήψη των αρχείων txt με τον αριθμό των σημείων και τις μεταξύ τους αποστάσεις. Τέλος πατώντας «Continue» γίνεται η μετάβαση στο επόμενο βήμα.

Point of interest:   
 Point coverage:

**Markers**

-----Markers-----



Στην τέταρτη σελίδα ορίζονται οι τιμές των τεσσάρων πρώτων κριτηρίων ρύπανσης για κάθε σημείο. Επόμενο βήμα είναι η εισαγωγή της κατεύθυνσης των ανέμων από το αρχείο excel (μορφής xlsm,xls) .

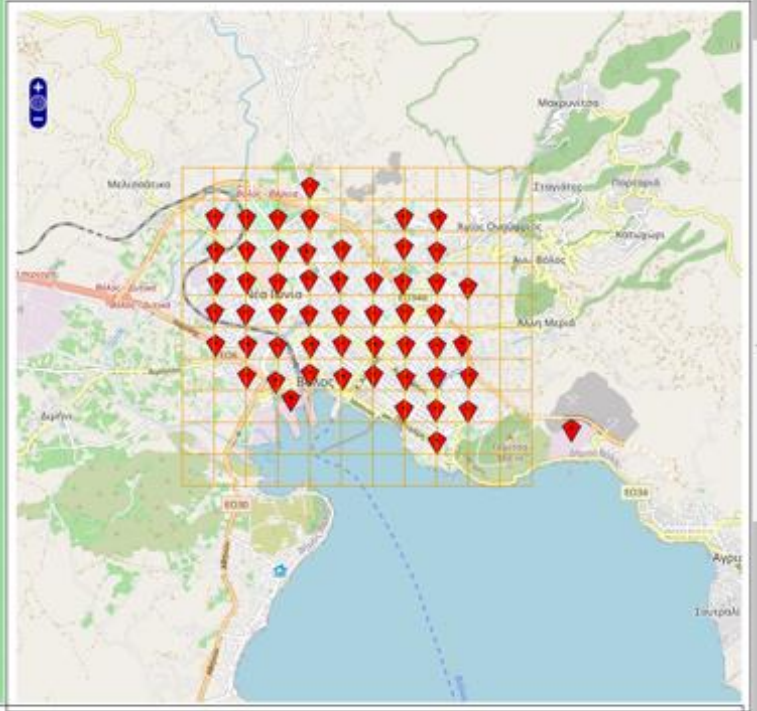
Οδηγίες excel

Το αρχείο excel πρέπει να περιέχει έναν πίνακα όπως και η εικόνα του excel του κεφαλαίου 2. Το όνομα το φύλλου εργασίας (worksheet) του αρχείου πρέπει να είναι “FINAL” και ο πίνακας πρέπει να έχει διαστάσεις 4 x 17 (από το 3-6 και από το A-Q)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1						FINAL TABLE											
2																	
3	μποφόρ	N	NNE	NE	ENE	E	ESE	SE	SSE	S	SSW	SW	WSW	W	WNW	NW	NNW
4	0(νηνεμία)	25.89%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
5	1 to 2	2.46%	2.73%	5.46%	17.48%	1.09%	1.09%	1.91%	12.84%	2.73%	1.09%	5.73%	1.09%	1.36%	4.64%	5.46%	3%
6	>3	0.54%	1.63%	0%	0.54%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0.54%	0.54%	0%

Μετά το ‘ ανέβασμα’ του αρχείου κάνουμε λήψη “Download” του txt αρχείου. Τέλος ορίζουμε τον αριθμό των μετρητών που θέλουμε να κατασκευάσουμε “Number of devices” γίνεται αποθήκευση πατώντας το «Save» και στη συνέχεια «Continue».

POSITIONS	ROAD WIDTH	BUILDING	TRAFFIC VOLUME	HOME HEATING
Position 1				
Position 2				
Position 3				
Position 4				
Position 5				
Position 6				
Position 7				
Position 8				
Position 9				
Position 10				
Position 11				
Position 12				
Position 13				
Position 14				
Position 15				
Position 16				
Position 17				
Position 18				
Position 19				
Position 20				
Position 21				
Position 22				
Position 23				
Position 24				
Position 25				
Position 26				
Position 27				
Position 28				
Position 29				
Position 30				
Position 31				
Position 32				
Position 33				
Position 34				
Position 35				
Position 36				
Position 37				
Position 38				
Position 39				
Position 40				
Position 41				
Position 42				
Position 43				
Position 44				
Position 45				
Position 46				
Position 47				
Position 48				
Position 49				
Position 50				
Position 51				
Position 52				



Choose Excel Wind Data File

Sheet name

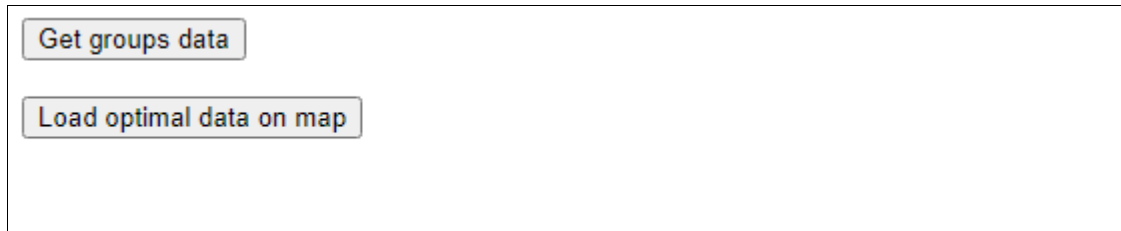
Download

Number of devices

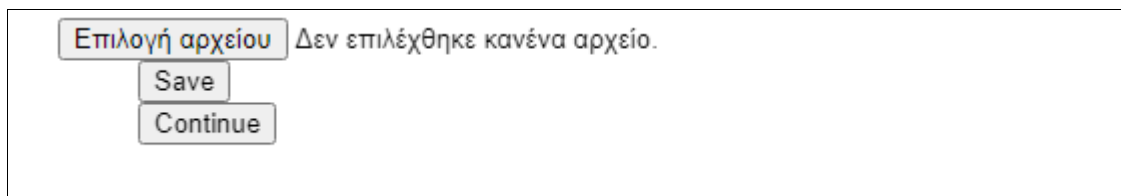
Save  Continue



Πατώντας το «Get groups data» γίνεται λήψη των ομάδων βάση των κατηγοριοποιήσεων που αναφέρθηκαν στην παράγραφο 2.4 και στη συνέχεια «Load optimal data on map» για να τροφοδοτήσουμε τα αρχεία που θα πάρουμε από την επίλυση τις βελτιστοποίησης με τη χρήση της Cplex στον χάρτη.



Τελευταίο βήμα είναι η εμφάνιση των σημείων στον χάρτη αφού εκτελέσουμε το πρόγραμμα βελτιστοποίησης. Αρχικά επιλέγουμε το αρχείο της βελτιστοποίησης και στη συνέχεια αποθηκεύουμε (Save) και συνεχίζουμε (Continue) στην εμφάνιση των σημείων στο χάρτη.



## Visual Studio C++

Αρχικά ο χρήστης ανοίγει το αρχείο `iloadmipex1.vcxproj` που βρίσκεται στον φάκελο εγκατάστασης της Cplex(IBM) όπως στο παρακάτω μονοπάτι (π.χ. `C:\Program Files\IBM\ILOG\CPLEX_Studio129\cplex\examples\x64_windows_vs2015\stat_mda`). Στη συνέχεια ανοίγει το αρχείο και αντικαθιστά τον κώδικα με τον κώδικα που παρουσιάζεται παρακάτω. Τέλος το αποθηκεύει ως ένα καινούργιο αρχείο και για την εκτέλεση του απαιτείται να ανοίγει το αρχείο με δικαιώματα διαχειριστή.

Εκτελώντας το πρόγραμμα αρχικά ζητείται το μονοπάτι (path) των αρχείων που έχουν εξαχθεί από την εφαρμογή Apache Netbeans (π.χ. D:\Downloads\ Distance.txt). Αφού δώθει για κάθε αρχείο το μονοπάτι το πρόγραμμα θα δώσει τα αποτελέσματα της βετιστοποίησης για κάθε μέθοδο κατηγοριοποίησης( ανάλογα με το πλήθος των σημείων μπορεί να καθυστερήσει η εύρεση της βέλτιστης λύσης).

Μόλις βρεθεί η βέτιστη λύση για κάθε κατηγοριοποίηση ζητείται το μονοπάτι στο οποίο θα γίνει η λήψη του αρχείου των αποτελεσμάτων και ολοκληρώνεται η εκτέλεση του προγράμματος.

## **Κώδικες προγραμμάτων**

### **Αρχείο “ index.html”**

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Document</title>
  </head>
  <body>
    <h1 style="text-align:center">Optimal points for pollution recorders</h1>
    <form action="1.Intro.html">
      Latitude: <input type="number" id="lat" value="39.365951"/><br />
      Longitude: <input type="number" id="lon" value="22.951762" /><br />
      <input type="button" id="bt" value="Save" onclick="saveFile()" />
      <input type="submit" id="bt" value="Continue" />
    </form>
    <p>Created by Charalampos Zouloumis</p>
  </body>
  <script>
    let saveFile = () => {
      const center_lat = document.getElementById("lat").value;
      const center_lon = document.getElementById("lon").value;
      localStorage.setItem("lat_coord",center_lat);
```

```
        localStorage.setItem("lon_coord",center_lon);
    };
</script>
</html>
```

## Αρχείο “1.Intro.html”

```
<!DOCTYPE html>
<html>
  <head>
    <title>Grid length</title>
    <form action="2a.Main_impact_zones.html">
      Grid distance(meters):<input type="number" id="Grid" name="Grid"/> <br/>
      <small> If grid smaller than 100 meters map will take a long time to load </small>
      <input type="submit" value="Ready" onclick="myFunction()" />
    </form>
    <script src="http://www.openlayers.org/api/OpenLayers.js"></script>
    <script>
      var map,vectorLayer,selectMarkerControl,selectedFeature;
      var lat = localStorage.getItem("lat_coord");
      var lon = localStorage.getItem("lon_coord");
      var zoom = 13;
      var position;
      var arr1 = [];
      var arr2 = [];
      var fromProjection = new OpenLayers.Projection("EPSG:4326");// Transform from WGS 1984
      var toProjection = new OpenLayers.Projection("EPSG:900913");// to Spherical Mercator Projection
      var cntrposition = new OpenLayers.LonLat(lon, lat).transform( fromProjection, toProjection);
      k=0;
      const glength = localStorage.getItem("lgrid");
```

```

function init(){
    map = new OpenLayers.Map("Map",{
        controls : [ new OpenLayers.Control.LayerSwitcher({})]);
    var mapnik = new OpenLayers.Layer.OSM("MAP");
    map.addLayer(mapnik);

    map.addControl(new OpenLayers.Control.Navigation());
    map.addControl(new OpenLayers.Control.ZoomPanel());
    map.addControl(new OpenLayers.Control.KeyboardDefaults());
    var click = new OpenLayers.Control.Click();
    map.addControl(click);
    click.activate();
    map.setCenter(cnrposition, zoom);};
OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control, {
defaultHandlerOptions: {
'single': true,
'double': false,
'pixelTolerance': 0,
'stopSingle': false,
'stopDouble': false
},
initialize: function(options) {
    this.handlerOptions = OpenLayers.Util.extend(
        {}, this.defaultHandlerOptions
    );
    OpenLayers.Control.prototype.initialize.apply(
        this, arguments
    );
    this.handler = new OpenLayers.Handler.Click(
        this, {

```

```

        'click': this.trigger
    }, this.handlerOptions
);
},
trigger: function(geo) {
var lonlat = map.getLonLatFromPixel(geo.xy);
lonlat1= new OpenLayers.LonLat(lonlat.lon,lonlat.lat).transform(toProjection,fromProjection);
alert(lonlat1.lat + "," + lonlat1.lon);
    x = lonlat1.lat ;
    y = lonlat1.lon ;
k=k+1 ;
if (k<=2){
    var mark = new OpenLayers.Layer.Markers("Position"+k);
    map.addLayer(mark);
    mark.addMarker(new OpenLayers.Marker(lonlat));
    arr1.push(x);
    arr2.push(y); } } });
function myFunction(){
const input = document.getElementById("Grid");
localStorage.setItem("lgrid",input.value);
localStorage.setItem("xgrid1",arr1[0]);
localStorage.setItem("xgrid2",arr1[1]);
localStorage.setItem("ygrid1",arr2[0]);
localStorage.setItem("ygrid2",arr2[1]);
alert("GRID LENGTH : " + input.value); }
</script>
<body onload='init();'>
    <div id="Map" style="height: 650px" ></div>
</body>
</html>

```

## Αρχείο “2a.Main\_impact zones.html”

```
<!DOCTYPE html>

<title>Problem Marker - Impact Zones</title>

<style type="text/css">
  html, body, #demo {
    position: absolute;
    left: 2%;}

  html, body, #activeMarkers {
    position: absolute;
    left: 2%;}

  html, body, #Map {
    position: fixed;
    top: 0;
    right: 0;
    width: 75%;
    height: 100%; }
</style>

  <form>
  </form>

<script src="http://www.openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript">
  var lonlat;

  var map,vectorLayer,selectMarkerControl,selectedFeature,markers, marker,x,y;

  var lat = 39.365951;
  var lon = 22.951762;
  var zoom = 13;
  var position;
  var arr1 = [];
  var arr2 = [];
  var coordx = [];
```

```

var coordy = [];
var arrMarkers = [];
var coordarr = [];
var dist = [];
var pollution_per = [];
var pollution_imp = [];
var sum ;
var per1,per2 ,per3 ,per4
var fromProjection = new OpenLayers.Projection("EPSG:4326");// Transform from WGS 1984
var toProjection = new OpenLayers.Projection("EPSG:900913");// to Spherical Mercator Projection
var cntrposition = new OpenLayers.LonLat(lon, lat).transform( fromProjection, toProjection);
k=0;
l=0;
const glength = localStorage.getItem("lgrid");
const xlen1 = localStorage.getItem("xgrid1");
const xlen2 = localStorage.getItem("xgrid2");
const ylen1 = localStorage.getItem("ygrid1");
const ylen2 = localStorage.getItem("ygrid2");

function init(){
    map = new OpenLayers.Map("Map",{
        controls : [ ] } );
    var mapnik = new OpenLayers.Layer.OSM("MAP");
    markers = new OpenLayers.Layer.Markers( "Markers" );
    map.addLayers([mapnik,markers]);
    map.addControl(new OpenLayers.Control.Navigation());
    map.addControl(new OpenLayers.Control.ZoomPanel());
    map.addControl(new OpenLayers.Control.KeyboardDefaults());
    var click = new OpenLayers.Control.Click();
    map.addControl(click);

```

```

click.activate();

// grammes gia thn dhmiourgia grid
// Add Vector
var vector = new OpenLayers.Layer.Vector("Grid");
map.addLayers([vector]);
var xlength1 = xlen1;
var xlength2 = xlen2;
var ylength1 = ylen1;
var ylength2 = ylen2;
var swapx ;
var swapy ;

if(xlength2 < xlength1 ){
    swapx = xlength1 ;
    xlength1 = xlength2 ;
    xlength2 = swapx ;}

if(ylength2 < ylength1 ){
    swapy = ylength1 ;
    ylength1 = ylength2 ;
    ylength2 = swapy ;}

var longitude ;
var latitude ;

for (longitude = +ylength1;longitude <= +ylength2;) {
    for (latitude = +xlength1; latitude <= +xlength2;) {

```



```

        var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        latitude = +latitude + 0.000008984*glength ;

        var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        var line1 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point1, end_point1]));

        vector.addFeatures([line1]);
    }

    var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

    longitude = +longitude + 0.000011588*glength ;

    var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

    var line2 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point2,
end_point2]));

    vector.addFeatures([line2]); }

for (latitude = +xlength1; latitude <= +xlength2;) {
    for (longitude = +ylength1; longitude <= +ylength2;) {
        var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        longitude = +longitude + 0.000011588*glength ;

        var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        var line2 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point2, end_point2]));

        vector.addFeatures([line2]);}

        var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        latitude = +latitude + 0.000008984*glength ;

        var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

```

```

    var line1 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point1,
end_point1]));
    vector.addFeatures([line1]); }

    map.setCenter(cnrposition, zoom); };
OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control, {
defaultHandlerOptions: {
'single': true,
'double': false,
'pixelTolerance': 0,
'stopSingle': false,
'stopDouble': false
},
initialize: function(options) {
    this.handlerOptions = OpenLayers.Util.extend(
    {}, this.defaultHandlerOptions
    );
    OpenLayers.Control.prototype.initialize.apply(
    this, arguments
    );
    this.handler = new OpenLayers.Handler.Click(
    this, {
        'click': this.trigger
    }, this.handlerOptions ); },
trigger: function(geo) {
lonlat = map.getLonLatFromPixel(geo.xy);
lonlat1= new OpenLayers.LonLat(lonlat.lon,lonlat.lat).transform(toProjection,fromProjection);
    x = lonlat.lat ;
    y = lonlat.lon ;
w1 = lonlat1.lat ;

```

```
w2 = lonlat1.lon ; } });
```

```
function AddMarker(markerID,severity,lonlat) {  
    marker = new OpenLayers.Marker(lonlat);  
    marker.MarkerID = markerID;  
    marker.severity = severity;  
    markers.addMarker(marker);  
    arrMarkers.push(marker); }
```

```
function displayMarker() {  
    var i = document.getElementById('MarkerList');  
    var newMarker = document.createElement('option');  
    newMarker.text = document.getElementById("marker_ID").value ;  
    arr1.push(w1);  
    arr2.push(w2);  
    coordx.push(x);  
    coordy.push(y);  
    coordarr.push(lonlat);  
    if (newMarker.text == null || newMarker.text == "") {  
        alert("Please specify an Marker ID");}  
    else { try {  
        i.add(newMarker, i.newMarker[null]); // standards compliant; doesn't work in IE  
    }  
    catch (e) {  
        i.add(newMarker, null); // IE only }  
    var markerID = document.getElementById("marker_ID").value;  
    var severity = document.getElementById("pollution_severity").value;  
    k=k+1;  
    pollution_imp.push(severity);  
    AddMarker(markerID,severity,lonlat); } }
```

```

function RemoveMarker(markerID) {
    for (var i in arrMarkers) {
        if (arrMarkers[i].MarkerID == markerID) {
            markers.removeMarker(arrMarkers[i]);
            arrMarkers.splice(i, 1);
            arr1.splice(i, 1);
            arr2.splice(i, 1);
            coordarr.splice(i, 1);
            coordx.splice(i, 1);
            coordy.splice(i, 1);
            pollution_imp.splice(i, 1);
            k=k-1 ;
            return; } } }

function removeMarker() {
    var i = document.getElementById("MarkerList");
    var id = i.options[i.selectedIndex].value;
    i.remove(i.selectedIndex);
    RemoveMarker(id); }

////////////////////////////////////

let saveFile = () => {
    localStorage.myArray1_impact = JSON.stringify(coordx);
    localStorage.myArray2_impact = JSON.stringify(coordy);
    localStorage.markcoord_impact = JSON.stringify(coordarr);
    localStorage.x_impactcoord = JSON.stringify(arr1);
    localStorage.y_impactcoord = JSON.stringify(arr2);
    localStorage.pollution_severity = JSON.stringify(pollution_imp); };

</script>
<body onload="init();">
<div id="Map"></div>

```

```

<br /><div id="demo">
High impact zone: <input type="text" id="marker_ID" value="center"/><br />
Severity of Pollution:<input type="number" id="pollution_severity" value="0"/><br />
<dd><button value="Add Marker" OnClick="displayMarker()">Import source of
emission</button></dd>
<br /><br />
<h3><dd> High Impact Zone Markers </dd></h3>
<dd><select id="MarkerList" size="10" multiple="multiple" style="text-align:center" ></dd>
<option>-----Markers-----</option>
</select><br />
<dd><button value="Remove Marker" OnClick="removeMarker()">Remove source of
emission</button></dd>
<br /><br />
<dd><input type="button" id="save" value="Save" onclick="saveFile()" /></dd>
<dd><input type="button" id="bt" value="Continue" onclick="
window.open('2b.Main_points.html','_blank')" /></dd>
</div>
</body>

```

## Αρχείο “2b.Main\_points.html”

```

<!DOCTYPE html>
<title>Problem Marker</title>
<style type="text/css">
html, body, #demo {
position: absolute;
left: 2%; }
html, body, #activeMarkers {
position: absolute;
left: 2%; }
html, body, #Map {
position: fixed;

```

```

top: 0;
right: 0;
width: 75%;
height: 100%;
}
</style>
<form>
</form>
<script src="http://www.openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript">
var lonlat;
var map,vectorLayer,selectMarkerControl,selectedFeature,markers, marker,x,y;
var lat = 39.365951;
var lon = 22.951762;
var zoom = 13;
var position;
var arr1 = [];
var arr2 = [];
var coordx = [];
var coordy = [];
var arrMarkers = [];
var coordarr = [];
var cover=[];
var dist = [];
var fromProjection = new OpenLayers.Projection("EPSG:4326");// Transform from WGS 1984
var toProjection = new OpenLayers.Projection("EPSG:900913");// to Spherical Mercator Projection
var cntposition = new OpenLayers.LonLat(lon, lat).transform( fromProjection, toProjection);
k=0;
l=0;
const glength = localStorage.getItem("lgrid");

```

```

const xlen1 = localStorage.getItem("xgrid1");
const xlen2 = localStorage.getItem("xgrid2");
const ylen1 = localStorage.getItem("ygrid1");
const ylen2 = localStorage.getItem("ygrid2");

function init(){
    map = new OpenLayers.Map("Map",{
        controls :
        [ // new OpenLayers.Control.PanZoomBar(),
          //new OpenLayers.Control.LayerSwitcher({})
          // new OpenLayers.Control.MousePosition({})//,
          // new OpenLayers.Control.ScaleLine(),
          // new OpenLayers.Control.OverviewMap() ] } );
    var mapnik = new OpenLayers.Layer.OSM("MAP");
    markers = new OpenLayers.Layer.Markers( "Markers" );
    map.addLayers([mapnik,markers]);
    map.addControl(new OpenLayers.Control.Navigation());
    map.addControl(new OpenLayers.Control.ZoomPanel());
    map.addControl(new OpenLayers.Control.KeyboardDefaults());
    var click = new OpenLayers.Control.Click();
    map.addControl(click);
    click.activate();

    // grammes gia thn dhmiourgia grid
    // Add Vector
    var vector = new OpenLayers.Layer.Vector("Grid");
    map.addLayers([vector]);
    var xlength1 = xlen1;
    var xlength2 = xlen2;
    var ylength1 = ylen1;

```

```

var ylength2 = ylen2;
var swapx ;
var swapy ;
if(xlength2 < xlength1 ){
    swapx = xlength1 ;
    xlength1 = xlength2 ;
    xlength2 = swapx ; }
if(ylength2 < ylength1 ){
    swapy = ylength1 ;
    ylength1 = ylength2 ;
    ylength2 = swapy ; }
var longitude ;
var latitude ;

for (longitude = +ylength1;longitude <= +ylength2;) {
    for (latitude = +xlength1; latitude <= +xlength2;) {
        var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        latitude = +latitude + 0.000008984*glength ;
        var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        var line1 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point1, end_point1]));
        vector.addFeatures([line1]); }
        var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        longitude = +longitude + 0.000011588*glength ;
        var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        var line2 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point2,
end_point2]));
        vector.addFeatures([line2]); }

```



```

for (latitude = +xlength1; latitude <= +xlength2;) {
    for (longitude = +ylength1; longitude <= +ylength2;) {
        var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        longitude = +longitude + 0.000011588*glength ;
        var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        var line2 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point2, end_point2]));
        vector.addFeatures([line2]); }
        var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        latitude = +latitude + 0.000008984*glength ;
        var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        var line1 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point1,
end_point1]));
        vector.addFeatures([line1]); }
        map.setCenter(cntrposition, zoom);
    };

```

```

OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control, {
defaultHandlerOptions: {
'single': true,
'double': false,
'pixelTolerance': 0,
'stopSingle': false,
'stopDouble': false },
initialize: function(options) {
    this.handlerOptions = OpenLayers.Util.extend(

```

```

    }, this.defaultHandlerOptions
  );
  OpenLayers.Control.prototype.initialize.apply(
    this, arguments
  );
  this.handler = new OpenLayers.Handler.Click(
    this, {'click': this.trigger
      }, this.handlerOptions); },
trigger: function(geo) {
lonlat = map.getLonLatFromPixel(geo.xy);
lonlat1= new OpenLayers.LonLat(lonlat.lon,lonlat.lat).transform(toProjection,fromProjection);
//alert(lonlat1.lat + "," + lonlat1.lon);
    x = lonlat.lat ;
    y = lonlat.lon ;
w1 = lonlat1.lat ;
w2 = lonlat1.lon ;  } });

function AddMarker(markerID,point_cover, lonlat) {
    marker = new OpenLayers.Marker(lonlat);
    marker.MarkerID = markerID;
    markers.addMarker(marker);
    arrMarkers.push(marker);
    cover.push(point_cover); }

function displayMarker() {
    var i = document.getElementById('MarkerList');
    var newMarker = document.createElement('option');
    newMarker.text = document.getElementById("marker_ID").value;
    arr1.push(w1);
    arr2.push(w2);
    coordx.push(x);

```

```

coordy.push(y);
coordarr.push(lonlat);
if (newMarker.text == null || newMarker.text == "") {
    alert("Please specify an Marker ID"); }
else { try {
    i.add(newMarker, i.newMarker[null]); // standards compliant; doesn't work in IE
    }
catch (e) {
    i.add(newMarker, null); // IE only }
var point_cover = document.getElementById("coverage").value;
var markerID = document.getElementById("marker_ID").value;
k=k+1;
AddMarker(markerID,point_cover ,lonlat); } }

```

```

function RemoveMarker(markerID) {
    for (var i in arrMarkers) {
        if (arrMarkers[i].MarkerID == markerID) {
            markers.removeMarker(arrMarkers[i]);
            arrMarkers.splice(i, 1);
            arr1.splice(i, 1);
            arr2.splice(i, 1);
            coordarr.splice(i, 1);
            coordx.splice(i, 1);
            coordy.splice(i, 1);
            cover.splice(i, 1);
            k=k-1 ;
            return; } } }

```

```

function removeMarker() {
    var i = document.getElementById("MarkerList");

```

```

    var id = i.options[i.selectedIndex].value;

    i.remove(i.selectedIndex);

    RemoveMarker(id); }

////////////////////////////////////

let saveFile = () => {

// Get the data from each element on the form.
localStorage.myArray1 = JSON.stringify(coordx);
localStorage.myArray2 = JSON.stringify(coordy);
localStorage.markcoord = JSON.stringify(coordarr);
localStorage.xcoord = JSON.stringify(arr1);
localStorage.ycoord = JSON.stringify(arr2);
localStorage.points_coverage = JSON.stringify(cover);

let data =

    'Grid Length: ' + glength +

    '\r' + 'Coordinates: ';

for (i = 0; i <arr1.length; i++){

    b= i + 1 ;

    data = data + b + ' ) ' + arr1[i] + ' ' + arr2[i] + ' |Coverage: ' + cover[i]+' |Adress:
'+arrMarkers[i].MarkerID+'\r' + '          ';

}

var distance ;

let data2 = 'Marker Distance'+ '\r' ;

for(i = 0; i <arr1.length; i++){

    for(j=i+1          ;          j          <arr1.length;          j++){Math.sqrt(((+arr1[i]/0.000008984)-
(+arr1[j]/0.000008984))*((+arr1[i]/0.000008984)-(+arr1[j]/0.000008984))+((+arr2[i]/0.000011588)-
(+arr2[j]/0.000011588))*((+arr2[i]/0.000011588)-(+arr2[j]/0.000011588)))

```

```

        distance= Math.sqrt(((+arr1[i]/0.000008984)-(+arr1[j]/0.000008984))*((+arr1[i]/0.000008984)-
(+arr1[j]/0.000008984))+((+arr2[i]/0.000011588)-(+arr2[j]/0.000011588))*((+arr2[i]/0.000011588)-
(+arr2[j]/0.000011588)));
        dist.push(distance);
        data2 = data2 +'distance -> '+ i +','+ j+ ' = '+ dist[l]+'r';
        l=l+1; } }

// Convert the text to BLOB.
const textToBLOB = new Blob([data], { type: 'text/plain' } );
const sFileName = 'formData.txt'; // The file to save the data.
let newLink = document.createElement("a");
newLink.download = sFileName;

if (window.webkitURL !== null) {
    newLink.href = window.webkitURL.createObjectURL(textToBLOB); }
else { newLink.href = window.URL.createObjectURL(textToBLOB);
    newLink.style.display = "none";
    document.body.appendChild(newLink); }
const textToBLOB2 = new Blob([data2], { type: 'text/plain' } );
const sFileName2 = 'Distance.txt'; // The file to save the data.
let newLink2 = document.createElement("a");
newLink2.download = sFileName2;

if (window.webkitURL !== null) {
    newLink2.href = window.webkitURL.createObjectURL(textToBLOB2); }
else { newLink2.href = window.URL.createObjectURL(textToBLOB2);
    newLink2.style.display = "none";
    document.body.appendChild(newLink2); }
newLink.click();
newLink2.click();
localStorage.setItem("posnum",k); };

```

```

</script>
<body onload="init();">
<div id="Map"></div>

<br /><div id="demo">
Point of interest: <input type="text" id="marker_ID" value="Address"/><br />
Point coverage: <input type="number" id="coverage" value="500"/><br />

<dd><button value="Add Marker" OnClick="displayMarker()">Add point of interest</button></dd>
<br /><br />
<h3><dd> Markers </dd></h3>
<dd><select id="MarkerList" size="10" multiple="multiple" style="text-align:center" ></dd>
<option>-----Markers-----</option>
</select><br />
<dd><button value="Remove Marker" OnClick="removeMarker()">Remove point of
interest</button></dd>
<br /><br />
<br /><br />

<dd><input type="button" id="bt" value="Save" onclick="saveFile()" /></dd>
<dd><input type="button" id="bt" value="Continue" onclick=" window.open('3a.data
collection.html','_blank')" /></dd>
</div>
</body>

```

## Αρχείο “3a.data collection.html”

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
```

```
    <script>
```

```
      var roadwidth = [];
```

```
      var building = [];
```

```
      var trafvol = [];
```

```
      var homeheat = [];
```

```
      var indust = [];
```

```
      const numbpos = localStorage.getItem("posnum");
```

```
      const c = +numbpos ;
```

```
      var criteria = [];
```

```
      $(function(){
```

```
        $("#mydiv").html("");
```

```
        for(i=1;i<c+1 ;i++){
```

```
          $("#mydiv").append('<input type="text" value="Position ' + i+' " style="width: 90%"  
><br/>');
```

```
        }
```

```
        for(i=1;i<c+1 ;i++){
```

```
          $("#mydiv1").append('<input type="number" id="data1'+i+' " name="data1'+i+' "  
style="width: 90%"><br/>');
```

```
        }
```

```
        for(i=1;i<c+1;i++){
```

```
          $("#mydiv2").append('<input type="number" id="data2'+i+' " name="data2'+i+' "  
style="width: 90%"><br/>');
```

```

    }

    for(i=1;i<c+1;i++){
        $("#mydiv3").append('<input    type="number"    id="data3'+i+'"    name="data3'+i+'"
style="width: 90%"/><br/>');
    }

    for(i=1;i<c+1;i++){
        $("#mydiv4").append('<input    type="number"    id="data4'+i+'"    name="data4'+i+'"
style="width: 90%"/><br/>');
    }
});

let saveFile = () => {
    localStorage.setItem('range', document.getElementById("number_range").value);
    var hold_data;
    roadwidth = [];
    building = [];
    trafvol = [];
    homeheat = [];
    indust = [];
    // Get the data from each element on the form.
    j=0;
    // KATHGOPIA 1
    criteria[j]= new Array();
    for(i=1;i<c+1;i++){
        l=i-1;
        hold_data = document.getElementById('data1'+i)
        roadwidth.push(hold_data.value);
        criteria[j][l] = hold_data.value;
    }
}

```



```
j=j+1;

// KATHΓΟΡΙΑ 2
criteria[j]= new Array();
for(i=1;i<c+1;i++){
    l=i-1;
    hold_data = document.getElementById('data2'+i)
    building.push(hold_data.value);
    criteria[j][l] = hold_data.value;
}
j=j+1;
```

```
// KATHΓΟΡΙΑ 3
criteria[j]= new Array();
for(i=1;i<c+1;i++){
    l=i-1;
    hold_data = document.getElementById('data3'+i)
    trafvol.push(hold_data.value);
    criteria[j][l] = hold_data.value;
}
j=j+1;
```

```
// KATHΓΟΡΙΑ 4
criteria[j]= new Array();
for(i=1;i<c+1;i++){
    l=i-1;
    hold_data = document.getElementById('data4'+i)
    homeheat.push(hold_data.value);
    criteria[j][l] = hold_data.value;
}
```

```

//-----//
// This variable stores all the data.
let data1 = 'Category 1 - Road width(0.12)+'\r' ;
for (i = 0; i <roadwidth.length; i++){
    b= i + 1 ;
    data1 = data1 + 'Position ' + b +') ' + roadwidth[i] + '\r';
}
let data2 = 'Category 2 - Building(0.12)+'\r' ;
for (i = 0; i <building.length; i++){
    b= i + 1 ;
    data2 = data2 + 'Position ' + b +') ' + building[i] + '\r';
}

let data3 = 'Category 3 - Traffic Volume - Speeds(0.12)+'\r' ;
for (i = 0; i <trafvol.length; i++){
    b= i + 1 ;
    data3 = data3 + 'Position ' + b +') ' + trafvol[i] + '\r';
}

let data4 = 'Category 4 - Home Heating(0.35)+'\r' ;
for (i = 0; i <homeheat.length; i++){
    b= i + 1 ;
    data4 = data4 + 'Position ' + b +') ' + homeheat[i] + '\r';
}

localStorage.categories = JSON.stringify(criteria);
// Convert the text to BLOB.
const textToBLOB1 = new Blob([data1], { type: 'text/plain' } );
const sFileName1 = 'group1.txt'; // The file to save the data.
let newLink1 = document.createElement("a");

```

```

newLink1.download = sFileName1;
if (window.webkitURL !== null) {
    newLink1.href = window.webkitURL.createObjectURL(textToBLOB1);
}
else {
    newLink1.href = window.URL.createObjectURL(textToBLOB1);
    newLink1.style.display = "none";
    document.body.appendChild(newLink1);
}
////////////////////////////////////
const textToBLOB2 = new Blob([data2], { type: 'text/plain' });
const sFileName2 = 'group2.txt';      // The file to save the data.

let newLink2 = document.createElement("a");
newLink2.download = sFileName2;
if (window.webkitURL !== null) {
    newLink2.href = window.webkitURL.createObjectURL(textToBLOB2);
}
else {
    newLink2.href = window.URL.createObjectURL(textToBLOB2);
    newLink2.style.display = "none";
    document.body.appendChild(newLink2);
}
////////////////////////////////////
const textToBLOB3 = new Blob([data3], { type: 'text/plain' });
const sFileName3 = 'group3.txt';      // The file to save the data.

let newLink3 = document.createElement("a");
newLink3.download = sFileName3;

if (window.webkitURL !== null) {

```

```

    newLink3.href = window.webkitURL.createObjectURL(textToBLOB3);
}
else {
    newLink3.href = window.URL.createObjectURL(textToBLOB3);
    newLink3.style.display = "none";
    document.body.appendChild(newLink3);
}

```

```

////////////////////////////////////

```

```

const textToBLOB4 = new Blob([data4], { type: 'text/plain' } );
const sFileName4 = 'group4.txt';      // The file to save the data.

```

```

let newLink4 = document.createElement("a");
newLink4.download = sFileName4;

```

```

if (window.webkitURL !== null) {
    newLink4.href = window.webkitURL.createObjectURL(textToBLOB4);
}
else {
    newLink4.href = window.URL.createObjectURL(textToBLOB4);
    newLink4.style.display = "none";
    document.body.appendChild(newLink4);
}

```

```

////////////////////////////////////

```

```

newLink1.click();
newLink2.click();
newLink3.click();
newLink4.click();

```

```

    };
</script>
<style>
    iframe {
box-sizing: border-box;
width: 100%;
margin: 0;
padding: 10px;
border: none;
height: 100%;
}

#contentframe {
    top: 50%;
    right: 0px;
    width: 50% ;
    height: 100% ;
    position:absolute;
}

table {
    table-layout: fixed;
    width:50%;
}
</style>
</head>
<body>

<form id="form1">
<table>
<colgroup>
    <col style="background-color: #DCC48E">

```

```

    <col style="background-color: #97DB9A">
    <col style="background-color: #97DB9A">
    <col style="background-color: #97DB9A">
    <col style="background-color: #97DB9A">
    <col style="background-color: #97DB9A">
</colgroup>
<tr>
    <td> POSITIONS </td>
    <td> ROAD WIDTH </td>
    <td> BUILDING </td>
    <td> TRAFFIC VOLUME</td>
    <td> HOME HEATING</td>
</tr>

<tr>
    <td><div id="mydiv"></div></td>
    <td><div id="mydiv1"></div></td>
    <td><div id="mydiv2"></div></td>
    <td><div id="mydiv3"></div></td>
    <td><div id="mydiv4"></div></td>
</tr>
</table>

<iframe width="258" height="62" frameborder="0" scrolling="no" marginheight="0"
marginwidth="0" src="excel.html" style="border: 1px solid black"></iframe><br/>

Number of devices:<input type="number" id="number_range" />
<br />
<input type="button" id="bt" value="Save" onclick="saveFile()" />
<input type="button" id="bt" value="Continue" onclick="
window.open('4.Calculations.html','_blank')" />
</form>

```

```

    <div id="contentframe" >
        <iframe id ="#kallender" width="425" height="350" frameborder="0" scrolling="no"
marginheight="0" marginwidth="0" src="map for data collection.html" style="border: 1px solid
black"></iframe><br/>
    </div>
</body>
</html>

```

### Αρχείο “4.Calculations.html”

```

<!DOCTYPE html>
<html>
  <head>

    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
    <script>
function myFunction(){
  const wind = JSON.parse(localStorage.windstats);
  const category = JSON.parse(localStorage.categories);

  const xcoord = JSON.parse(localStorage.xcoord);
  const ycoord = JSON.parse(localStorage.ycoord);
  const ximpact_array = JSON.parse(localStorage.x_impactcoord);
  const yimpact_array = JSON.parse(localStorage.y_impactcoord);
  const pollution_impact = JSON.parse(localStorage.pollution_severity);

  const numbpos = localStorage.getItem("posnum");
  const group_number = localStorage.getItem('range'); //oi arithmoi twn omadwn gia thn
dimioyrgei groups

  var T5=[];
  var gwnies = [];
  var distance_table = [];
  var windtable = [];
  var W_impactsource=[];
  var Pei = [];
  var groups1 = [];
  var groups1_points = [];
  var groups2 = [];

```

```

var groups2_points = [];

var wind_impact_N = [];
var wind_impact_NNE = [];
var wind_impact_NE = [];
var wind_impact_ENE = [];
var wind_impact_E = [];
var wind_impact_ESE = [];
var wind_impact_SE = [];
var wind_impact_SSE = [];
var wind_impact_S = [];
var wind_impact_SSW = [];
var wind_impact_SW = [];
var wind_impact_WSW = [];
var wind_impact_W = [];
var wind_impact_WNW = [];
var wind_impact_NW = [];
var wind_impact_NNW = [];

var Wtotal= [];
var Wtotal_sorted = [];
var points_sorted = [];

//dimoyrgia windtable ola ta m se mia seira
windtable[0] = wind[1][1] ;

// ποσοστα κριτηρίων 1 - 4

for(i=2;i<wind.length;i++){
    for(j=1;j<wind[1].length;j++){
        windtable.push(wind[i][j]);
    }
}

function getAngleDeg(ax,ay,bx,by) {
    var angleRad = Math.atan((ay-by)/(ax-bx));
    var angleDeg = angleRad * 180 / Math.PI;

```



```

    return(angleDeg);
}

for (i=0; i<xcoord.length ; i++){
    gwnies[i]= new Array();
    for (j=0; j<ximpact_array.length ; j++){
        var moires = getAngleDeg(yimpact_array[j],ximpact_array[j],ycoord[i],xcoord[i]);

        if(ximpact_array[j]<xcoord[i] & yimpact_array[j]<ycoord[i]){ // 1o tetarthmorio
            gwnies[i][j]= moires;

            }else if(ximpact_array[j]<xcoord[i] & yimpact_array[j]>ycoord[i]){// 2o
tetarthmorio
            gwnies[i][j]= moires + 180;

            }else if(ximpact_array[j]>xcoord[i] & yimpact_array[j]<ycoord[i]){// 4o
tetarthmorio
            gwnies[i][j]= moires +360;

            }else if(ximpact_array[j]>xcoord[i] & yimpact_array[j]>ycoord[i]){// 3o
tetarthmorio
            gwnies[i][j]= moires +180;

            }
            else if(ximpact_array[j]=xcoord[i] & yimpact_array[j]>ycoord[i]){// katw
            gwnies[i][j]= 180;
            }else if(ximpact_array[j]=xcoord[i] & yimpact_array[j]<ycoord[i]){// panw
            gwnies[i][j]= 0;
            }else if(yimpact_array[j]=ycoord[i] & ximpact_array[j]>xcoord[i]){// aristera
            gwnies[i][j]= 270;
            }else if(yimpact_array[j]=ycoord[i] & ximpact_array[j]<xcoord[i]){// dekcia
            gwnies[i][j]= 90;
            }
        }
    }
}

// Υπολογισμος αποστασεων απο τα σημεια high impact

```

```

for (i=0; i<xcoord.length ; i++){
    distance_table[i]= new Array();
    W_impactsource[i]= new Array();
    for (j=0; j<ximpact_array.length ; j++){
        distance_table[i][j] =Math.sqrt(((ximpact_array[j]-
xcoord[i])/0.000008984)*((ximpact_array[j]-xcoord[i])/0.000008984) + ((yimpact_array[j]-
ycoord[i])/0.000011588)*((yimpact_array[j]-ycoord[i])/0.000011588));
        dist = distance_table[i][j] ;
        W_impactsource[i][j] = pollution_impact[j] - pollution_impact[j]* (0.02/500)*dist ;
        // alert(ximpact_array[j]);
        //alert(distance_table[i][j]);
    }
}

```

*// Δημιουργεια πινακων ανεμων*

```

for (i=0; i<xcoord.length ; i++){
    wind_impact_N[i]= new Array();
    wind_impact_NNE[i]= new Array();
    wind_impact_NE[i]= new Array();
    wind_impact_ENE[i]= new Array();
    wind_impact_E[i]= new Array();
    wind_impact_ESE[i]= new Array();
    wind_impact_SE[i]= new Array();
    wind_impact_SSE[i]= new Array();
    wind_impact_S[i]= new Array();
    wind_impact_SSW[i]= new Array();
    wind_impact_SW[i]= new Array();
    wind_impact_WSW[i]= new Array();
    wind_impact_W[i]= new Array();
    wind_impact_WNW[i]= new Array();
    wind_impact_NW[i]= new Array();
    wind_impact_NNW[i]= new Array();

    for (j=0; j<ximpact_array.length ; j++){

        wind_impact_N[i][j]=0;
        wind_impact_NNE[i][j]=0;
        wind_impact_NE[i][j]=0;
        wind_impact_ENE[i][j]=0;
        wind_impact_E[i][j]=0;
        wind_impact_ESE[i][j]=0;
        wind_impact_SE[i][j]=0;
        wind_impact_SSE[i][j]=0;
        wind_impact_S[i][j]=0;
    }
}

```

```

wind_impact_SSW[i][j]=0;
wind_impact_SW[i][j]=0;
wind_impact_WSW[i][j]=0;
wind_impact_W[i][j]=0;
wind_impact_WNW[i][j]=0;
wind_impact_NW[i][j]=0;
wind_impact_NNW[i][j]=0;
}
}

```

// Υπολογισμος Tm,e,i ----ποιο ανεμοι επιρεαζουν καθε σημειο για τα εργοστασια βιομηχανιες

```

for (i=0; i<xcoord.length ; i++){

    for (j=0; j<ximpact_array.length ; j++){

        if(gwnies[i][j]> 0 & gwnies[i][j]< 22.5){
            wind_impact_S[i][j]= (gwnies[i][j]-0)/22.5 ;
            wind_impact_SSW[i][j]= (22.5-gwnies[i][j])/22.5 ;

        }else if(gwnies[i][j]> 22.5 & gwnies[i][j]< 45){
            wind_impact_SSW[i][j]= (gwnies[i][j]-22.5)/22.5 ;
            wind_impact_SW[i][j]= (45-gwnies[i][j])/22.5 ;

        }else if(gwnies[i][j]> 45 & gwnies[i][j]< 67.5){
            wind_impact_SW[i][j]= (gwnies[i][j]-45)/22.5 ;
            wind_impact_WSW[i][j]= (67.5-gwnies[i][j])/22.5 ;

        }else if(gwnies[i][j]> 67.5 & gwnies[i][j]< 90){
            wind_impact_WSW[i][j]= (gwnies[i][j]-67.5)/22.5 ;
            wind_impact_W[i][j]=(90-gwnies[i][j])/22.5 ;

        }else if(gwnies[i][j]> 90 & gwnies[i][j]< 112.5){
            wind_impact_W[i][j]=(gwnies[i][j]-90)/22.5 ;
            wind_impact_WNW[i][j]=(112.5-gwnies[i][j])/22.5 ;

        }else if(gwnies[i][j]> 112.5 & gwnies[i][j]< 135){
            wind_impact_WNW[i][j]=(gwnies[i][j]-112.5)/22.5 ; ;
            wind_impact_NW[i][j]=(135-gwnies[i][j])/22.5 ;

        }else if(gwnies[i][j]> 135 & gwnies[i][j]< 157.5){
            wind_impact_NW[i][j]=(gwnies[i][j]-135)/22.5 ;
            wind_impact_NNW[i][j]=(157.5-gwnies[i][j])/22.5 ;
        }
    }
}

```

```

}else if(gwnies[i][j]> 157.5 & gwnies[i][j]< 180){
    wind_impact_NNW[i][j]=(gwnies[i][j]-157.5)/22.5 ;
    wind_impact_N[i][j]=(180-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 180 & gwnies[i][j]< 202.5){
    wind_impact_N[i][j]=(gwnies[i][j]-180)/22.5 ;
    wind_impact_NNE[i][j]=(202.5-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 202.5 & gwnies[i][j]< 225){
    wind_impact_NNE[i][j]=(gwnies[i][j]-202.5)/22.5 ;
    wind_impact_NE[i][j]=(225-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 225 & gwnies[i][j]< 247.5){
    wind_impact_NE[i][j]=(gwnies[i][j]-225)/22.5 ;
    wind_impact_ENE[i][j]=(247.5-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 247.5 & gwnies[i][j]< 270){
    wind_impact_ENE[i][j]=(gwnies[i][j]-247.5)/22.5 ;
    wind_impact_E[i][j]=(270-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 270 & gwnies[i][j]< 292.5){
    wind_impact_E[i][j]=(gwnies[i][j]-270)/22.5 ;
    wind_impact_ESE[i][j]=(292.5-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 292.5 & gwnies[i][j]< 315){
    wind_impact_ESE[i][j]=(gwnies[i][j]-292.5)/22.5 ;
    wind_impact_SE[i][j]=(315-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 315 & gwnies[i][j]< 337.5){
    wind_impact_SE[i][j]=(gwnies[i][j]-315)/22.5 ;
    wind_impact_SSE[i][j]=(337.5-gwnies[i][j])/22.5 ;

}else if(gwnies[i][j]> 337.5 ){
    wind_impact_SSE[i][j]=(gwnies[i][j]-337.5)/22.5 ;
    wind_impact_S[i][j]= (360-gwnies[i][j])/22.5 ;
}

else if(gwnies[i][j]= 0 ){
    wind_impact_S[i][j]= 1 ;

}else if(gwnies[i][j]= 22.5 ){
    wind_impact_SSW[i][j]= 1 ;

}else if(gwnies[i][j]= 45 ){
    wind_impact_SW[i][j]= 1 ;

```

```

}else if(gwnies[i][j]= 67.5 ){
    wind_impact_WSW[i][j]= 1 ;

}else if(gwnies[i][j]= 90 ){
    wind_impact_W[i][j]= 1 ;

}else if(gwnies[i][j]= 112.5 ){
    wind_impact_WNW[i][j]= 1 ;

}else if(gwnies[i][j]= 135 ){
    wind_impact_NW[i][j]= 1 ;

}else if(gwnies[i][j]= 157.5 ){
    wind_impact_NNW[i][j]= 1 ;

}else if(gwnies[i][j]= 180 ){
    wind_impact_N[i][j]= 1 ;

}else if(gwnies[i][j]= 202.5 ){
    wind_impact_NNE[i][j]= 1 ;

}else if(gwnies[i][j]= 225 ){
    wind_impact_NE[i][j]= 1 ;

}else if(gwnies[i][j]= 247.5 ){
    wind_impact_ENE[i][j]= 1 ;

}else if(gwnies[i][j]= 270 ){
    wind_impact_E[i][j]= 1 ;

}else if(gwnies[i][j]= 292.5 ){
    wind_impact_ESE[i][j]= 1 ;

}else if(gwnies[i][j]= 315 ){
    wind_impact_SE[i][j]= 1 ;

}else if(gwnies[i][j]= 337.5 ){
    wind_impact_SSE[i][j]= 1 ;

}
}
}

```

// Υπολογισμος Ρει για τις 4 πρωτες κατηγοριες

```

for(i=0; i<xcoord.length; i++){

    Pei[i] = new Array();
    for(j=0; j<4; j++){
        Pei[i][j] =0;
        for(m=0; m<windtable.length; m++){
            Pei[i][j] = Pei[i][j] + windtable[m] ;

        }

    }

}

//Υπολογισμος Pei για high impact zones
for(i=0; i<xcoord.length; i++){
    k=0;

    for(j=4; j<ximpact_array.length + 4 ; j++){
        Pei[i][j] =0;

        for(m=0; m<windtable.length; m++){

            if(m==0){
                Pei[i][j]=Pei[i][j] + windtable[m] * 0 ; //οταν έχουμε ημεμια δεν επηρεάζονται
                αλλα σημεια παρα μονο το σημειο οπου παραγονται οι ρυποι
            }else if(m==1 || m==17){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_N[i][k] ;
            }else if(m==2 || m==18){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_NNE[i][k] ;
            }else if(m==3 || m==19){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_NE[i][k] ;
            }else if(m==4 || m==20){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_ENE[i][k] ;
            }else if(m==5 || m==21){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_E[i][k] ;
            }else if(m==6 || m==22){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_ESE[i][k] ;
            }else if(m==7 || m==23){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_SE[i][k] ;
            }else if(m==8 || m==24){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_SSE[i][k] ;
            }else if(m==9 || m==25){
                Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_S[i][k] ;
            }
        }
    }
}

```

```

    }else if(m==10 || m==26){
        Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_SSW[i][k] ;
    }else if(m==11 || m==27){
        Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_SW[i][k] ;
    }else if(m==12 || m==28){
        Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_WSW[i][k] ;
    }else if(m==13 || m==29){
        Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_W[i][k] ;
    }else if(m==14 || m==30){
        Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_WNW[i][k] ;
    }else if(m==15 || m==31){
        Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_NW[i][k] ;
    }else if(m==16 || m==32){
        Pei[i][j]=Pei[i][j] + windtable[m] * wind_impact_NNW[i][k] ;
    }

}
k=k+1;
}

}

// Υπολογισμος Wtotal

for(i=0; i<xcoord.length; i++){
    Wtotal[i]=0;

    k=0;

    for (j=0; j<ximpact_array.length +category.length ; j++){

        if(j<category.length){

            Wtotal[i] = Wtotal[i] + category[j][i]* Pei[i][j] ;

        }

        if(j>=category.length){

            Wtotal[i]= Wtotal[i] + W_impactsource[i][k]* Pei[i][j];

            k=k+1;
        }
    }
}

```

```

    }

}

// Νέο array για να ταξινομησω τα Wtotal\

for (i=0; i < Wtotal.length; i++){
    Wtotal_sorted[i] = Wtotal[i];
    points_sorted[i] = i+1;
}

temp=0;
temp_number = 0;
for (i = 0; i < Wtotal_sorted.length; i++) {
    for (var j = i; j < Wtotal_sorted.length; j++) {
        if (Wtotal_sorted[j] < Wtotal_sorted[i]) {
            temp = Wtotal_sorted[j];
            temp_number = points_sorted[j]
            Wtotal_sorted[j] = Wtotal_sorted[i];
            points_sorted[j] = points_sorted[i] ;
            Wtotal_sorted[i] = temp;
            points_sorted[i] = temp_number;
        }
    }
}
max = Wtotal_sorted[Wtotal_sorted.length-1];

let data1 = 'Wtotal_1 '+ '\r' ;
let data2 = 'Group points_1 '+ '\r' ;
// Κατηγοριοποίηση 1 (ισα διαστήματα)
min = Wtotal_sorted[0];

l=0;
k=0;
var counter=0;
for(i=min;i<max; ){

```



```

groups1[k]= new Array();
groups1_points[k] = new Array();
counter = counter +1;
for(j=0; j< Wtotal.length ; j++){

    if ( Wtotal[j]>= i & Wtotal[j]<=i+((max-min)/(group_number)) ) {

        groups1[k][l]= Wtotal[j];
        groups1_points[k][l] = j+1;

        l=l+1;
    }else if(counter == group_number & Wtotal[j]>= i+((max-
min)/(group_number))){
        groups1[k][l]= Wtotal[j];
        groups1_points[k][l] = j+1;

        l=l+1;
        i=i+ ((max-min)/(group_number));
    }

}

i=i+ ((max-min)/(group_number));

k=k+1;

l=0;
}

// Ελεγχος ωστε όλα τα γρουπ να έχουν ενα τουλαχιστον σημειο
for(i=0; i<groups1.length; i++){

    if ( groups1[i].length == 0 ) {

        for(k=i; k<groups1.length-1; k++){

            if(groups1[k+1].length !=0){

                groups1[i].push(groups1[k+1][0]);
                groups1[k+1].shift();
            }
        }
    }
}

```

```

        groups1_points[i].push(groups1_points[k+1][0]);
        groups1_points[k+1].shift();

        break;
    }
}

}

}

for(i=groups1.length-1; i>1; i--){

    if ( groups1[i].length == 0 ) {

        for(k=i; k>0; k--){

            if(groups1[k-1].length!=0){

                groups1[i].push(groups1[k-1][groups1[k-1].length-1]);
                groups1[k-1].pop();

                groups1_points[i].push(groups1_points[k-1][groups1_points[k-1].length-1]);
                groups1_points[k-1].pop();

                break;
            }
        }

    }

}

// Pass array data to a file for txt conversion
for(k=0 ; k<groups1.length; k++){

    for(l=0; l<groups1[k].length; l++ ){

        data1 = data1 + groups1[k][l]+ ' ';
        data2 = data2 + groups1_points[k][l]+ ' ';

    }

}

```

```
data1 = data1 + '\r';  
data2 = data2 + '\r';  
}
```

```
// Κατηγοριοποίηση 2 (γρουπ με ιδιο αριθμο σημειων)
```

```
let data3 = 'Wtotal_2 '+ '\r' ;  
let data4 = 'Group points_2 '+ '\r' ;
```

```
for(i=0; i< group_number ; i++ ){  
  
    groups2[i] = new Array();  
    groups2_points[i] = new Array();  
  
}  
  
l=0;  
  
for(i=0; i< numbpos; i++){  
  
    groups2[l].push(0);  
    groups2_points[l].push(0);  
  
    l=l+1;  
    if(((i+1) % group_number) == 0 & i!=0){  
  
        l=0;  
    }  
}  
  
k=0;
```

```

for(i=0;i<groups2.length; i++){

    for(j=0; j<groups2[i].length; j++){

        groups2_points[i][j] = points_sorted[k];
        groups2[i][j] = Wtotal_sorted[k] ;
        data3 = data3 + groups2[i][j]+ ' ';
        data4 = data4 + groups2_points[i][j]+ ' ';
        k=k+1;
    }
    data3 = data3 + '\r';
    data4 = data4 + '\r';
}

console.log(Wtotal);

console.log(groups1);
console.log(groups1_points);

console.log(groups2);
console.log(groups2_points);

// CGroups txt file.
const textToBLOB1 = new Blob([data1], { type: 'text/plain' });
const sFileName1 = 'Wtotal 1.txt'; // The file to save the data.

let newLink1 = document.createElement("a");
newLink1.download = sFileName1;

if (window.webkitURL !== null) {
    newLink1.href = window.webkitURL.createObjectURL(textToBLOB1);
}
else {
    newLink1.href = window.URL.createObjectURL(textToBLOB1);
    newLink1.style.display = "none";
    document.body.appendChild(newLink1);
}

// Wtotal txt file .
const textToBLOB2 = new Blob([data2], { type: 'text/plain' });
const sFileName2 = 'Groups point 1.txt'; // The file to save the data.

```

```

let newLink2 = document.createElement("a");
newLink2.download = sFileName2;

if (window.webkitURL !== null) {
    newLink2.href = window.webkitURL.createObjectURL(textToBLOB2);
}
else {
    newLink2.href = window.URL.createObjectURL(textToBLOB2);
    newLink2.style.display = "none";
    document.body.appendChild(newLink2);
}

const textToBLOB3 = new Blob([data3], { type: 'text/plain' });
const sFileName3 = 'Wtotal 2.txt'; // The file to save the data.

```

```

let newLink3 = document.createElement("a");
newLink3.download = sFileName3;

if (window.webkitURL !== null) {
    newLink3.href = window.webkitURL.createObjectURL(textToBLOB3);
}
else {
    newLink3.href = window.URL.createObjectURL(textToBLOB3);
    newLink3.style.display = "none";
    document.body.appendChild(newLink3);
}

```

```

////////////////////////////////////
const textToBLOB4 = new Blob([data4], { type: 'text/plain' });
const sFileName4 = 'Groups point 2.txt'; // The file to save the data.

```

```

let newLink4 = document.createElement("a");
newLink4.download = sFileName4;

if (window.webkitURL !== null) {
    newLink4.href = window.webkitURL.createObjectURL(textToBLOB4);
}
else {
    newLink4.href = window.URL.createObjectURL(textToBLOB4);
    newLink4.style.display = "none";
    document.body.appendChild(newLink4);
}

```

```

        newLink1.click();
        newLink2.click();
        newLink3.click();
        newLink4.click();
    }
</script>
</head>
<input type="submit" value="Get groups data" onclick="myFunction()" />
<br /><br />
<input type="button" id="bt" value="Load optimal data on map" onclick="
window.open('Read txt.html','_blank')" />
</html>

```

### Αρχείο “Optimal points map.html”

```

<!DOCTYPE html>
<title>Problem Marker</title>
<style type="text/css">
    html, body, #Map {
        position: fixed;
        top: 0;
        right: 0;
        width: 100%;
        height: 100%;
    }
</style>
<form>
</form>
<script src="http://www.openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript">
    const pointString1 = JSON.parse(localStorage.myArray1);
    const pointString2 = JSON.parse(localStorage.myArray2);
    const impactString1 = JSON.parse(localStorage.myArray1_impact);

```

```

const impactString2 = JSON.parse(localStorage.myArray2_impact);
const markerlonlat = JSON.parse(localStorage.markcoord);
const markerlonlat_impact = JSON.parse(localStorage.markcoord_impact);
const Optimal_x = JSON.parse(localStorage.getItem("optx"));
const Optimal_y = JSON.parse(localStorage.getItem("opty"));
const Optimal_points = JSON.parse(localStorage.getItem("points"));
const obj_value = JSON.parse(localStorage.getItem("objective_value"));
const coverage = JSON.parse(localStorage.getItem("points_coverage"));
const overcoverage = JSON.parse(localStorage.getItem("points_overcoverage"));
var lonlat;
var map,vectorLayer,selectMarkerControl,selectedFeature,markers, marker,x,y;
var lat = 39.365951;
var lon = 22.951762;
var zoom = 13;
var position;
var coord= [];
var dist = [];
var fromProjection = new OpenLayers.Projection("EPSG:4326");// Transform from WGS 1984
var toProjection = new OpenLayers.Projection("EPSG:900913");// to Spherical Mercator Projection
var cntrposition = new OpenLayers.LonLat(lon, lat).transform( fromProjection, toProjection);
k=0;
l=0;
const glength = localStorage.getItem("lgrid");
const xlen1 = localStorage.getItem("xgrid1");
const xlen2 = localStorage.getItem("xgrid2");
const ylen1 = localStorage.getItem("ygrid1");
const ylen2 = localStorage.getItem("ygrid2");

function init(){
    map = new OpenLayers.Map("Map",{

```

```

        controls :
        [ ] });

var mapnik = new OpenLayers.Layer.OSM("MAP");
markers = new OpenLayers.Layer.Markers( "Markers" );
map.addLayers([mapnik,markers]);
map.addControl(new OpenLayers.Control.Navigation());
map.addControl(new OpenLayers.Control.ZoomPanel());
map.addControl(new OpenLayers.Control.KeyboardDefaults());
var click = new OpenLayers.Control.Click();
map.addControl(click);
click.activate();

// grammes gia thn dhmiourgia grid
// Add Vector
var vector = new OpenLayers.Layer.Vector("Grid");
map.addLayers([vector]);
var xlength1 = xlen1;
var xlength2 = xlen2;
var ylength1 = ylen1;
var ylength2 = ylen2;
var swapx ;
var swapy ;

if(xlength2 < xlength1 ){
    swapx = xlength1 ;
    xlength1 = xlength2 ;
    xlength2 = swapx ;
}

if(ylength2 < ylength1 ){
    swapy = ylength1 ;

```



```

    ylength1 = ylength2 ;
    ylength2 = swapy ; }
var longitude ;
var latitude ;

for (longitude = +ylength1;longitude <= +ylength2;) {
    for (latitude = +xlength1; latitude <= +xlength2;) {
        var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        latitude = +latitude + 0.000008984*glength ;
        var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        var line1 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point1, end_point1]));
        vector.addFeatures([line1]);
    }
    var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
    longitude = +longitude + 0.000011588*glength ;
    var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
    var line2 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point2,
end_point2]));
    vector.addFeatures([line2]);
}

for (latitude = +xlength1; latitude <= +xlength2;) {
    for (longitude = +ylength1;longitude <= +ylength2;) {
        var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        longitude = +longitude + 0.000011588*glength ;
        var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

```

```

        var line2 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point2, end_point2]));
        vector.addFeatures([line2]);
    }

    var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
    latitude = +latitude + 0.000008984*glength ;
    var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

    var line1 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point1,
end_point1]));
    vector.addFeatures([line1]);
}

// ADD MARKERS ON TO THE MAP
for (i = 0; i<Optimal_points.length; i++){
    k= i+1;

    var coord = new OpenLayers.LonLat(Optimal_y[i],Optimal_x[i]).transform( fromProjection,
toProjection);

    marker = new OpenLayers.Marker(coord);
    markers.addMarker(marker);
}

// CENTER MAP
map.setCenter(cnrtrposition, zoom);
};
OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control, {
defaultHandlerOptions: {
'single': true,
'double': false,
'pixelTolerance': 0,
'stopSingle': false,

```

```

'stopDouble': false
},

initialize: function(options) {
    this.handlerOptions = OpenLayers.Util.extend(
        {}, this.defaultHandlerOptions
    );
    OpenLayers.Control.prototype.initialize.apply(
        this, arguments
    );
    this.handler = new OpenLayers.Handler.Click(
        this, {
            'click': this.trigger
        }, this.handlerOptions
    );
},

trigger: function(geo) {
    lonlat = map.getLonLatFromPixel(geo.xy);
    lonlat1 = new OpenLayers.LonLat(lonlat.lon,lonlat.lat).transform(toProjection,fromProjection);

    for (i = 0; i<Optimal_points.length; i++){
        k=i+1;
        var popup = new OpenLayers.Popup("chicken",
            new OpenLayers.LonLat(Optimal_y[i],Optimal_x[i]).transform(
                toProjection,
                fromProjection),
            new OpenLayers.Size(73,20),
            "Point "+Optimal_points[i],
            true);
        map.addPopup(popup);
    }
}

```

```
    } } });  
</script>  
<body onload="init();">  
<div id="Map"></div>  
</div>  
</body>
```

## Αρχείο “Read txt.html”

```
<!DOCTYPE html>  
<title>Problem Marker</title>  
<style type="text/css">  
    html, body, #demo {  
        position: absolute;  
        left: 2%;  
    }  
    html, body, #activeMarkers {  
        position: absolute;  
        left: 2%;  
  
    }  
    html, body, #Map {  
        position: fixed;  
        top: 0;  
        right: 0;  
        width: 75%;  
        height: 100%;  
    }  
</style>  
<form>
```

```

    </form>
<script src="http://www.openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript">
    var Data_array= [] ;
    var Cordx_array= [] ;
    var Cordy_array= [] ;

    function readFile(input) {
        let file = input.files[0];
        let reader = new FileReader();
        reader.readAsText(file);
        reader.onload = function() {
            Array_data =reader.result.split(" ");
            obj_value =Array_data[2];
            coverage=Array_data[4];
            overcoverage =Array_data[7];
            var k=14;
            var l=16;
            var m=17;

            for(i=0; i<Array_data.length ; i++){
                if(k==i){
                    Data_array.push(Array_data[i]);
                    k +=6 ;
                }
                if(l==i){
                    Cordx_array.push(Array_data[i]);
                    l +=6 ;
                }
                if(m==i){

```

```

        Cordy_array.push(Array_data[i]);
        m +=6 ;
    }
}
console.log(Data_array);
console.log(Cordx_array);
console.log(Cordy_array);
};
reader.onerror = function() {
    console.log(reader.error);
};
};

////////////////////////////////////
let saveFile = () => {
    localStorage.setItem("objective_value", JSON.stringify(obj_value));
    localStorage.setItem("points_coverage", JSON.stringify(coverage));
    localStorage.setItem("points_overcoverage", JSON.stringify(overcoverage));
    localStorage.setItem("points", JSON.stringify(Data_array));
    localStorage.setItem("optx", JSON.stringify(Cordx_array));
    localStorage.setItem("opty", JSON.stringify(Cordy_array));
};

</script>
<body onload="init();">
<input type="file" onchange="readFile(this)">
<dd><input type="button" id="bt" value="Save" onclick="saveFile()" /></dd>
<dd><input type="button" id="bt" value="Continue" onclick=" window.open('Optimal points
map.html','_blank')" /></dd>

</div>
</body>

```

## Αρχείο “excel.html”

```
<!DOCTYPE html>
```

```
<!--
```

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

```
-->
```

```
<html>
```

```
  <head>
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.13.5/xlsx.full.min.js"></script> type="text/javascript"
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/xlsx/0.13.5/jszip.js"></script> type="text/javascript"
```

```
    <script type="text/javascript">
```

```
      var sheet_name ;
```

```
      var windtable = [];
```

```
      var windtable1 = [];
```

```
      var windtable2 = [];
```

```
      var windtable3 = [];
```

```
      var windtable4 = [];
```

```
function Download() {
```

```
  //Reference the FileUpload element.
```

```
  var fileUpload = document.getElementById("file");
```

```
  //Validate whether File is valid Excel file.
```

```

var regex = /^[a-zA-Z0-9\s_\|\.\-:]+(.xls|.xlsx)$/;
if (regex.test(fileUpload.value.toLowerCase())) {
    if (typeof (FileReader) != "undefined") {
        var reader = new FileReader();

        //For Browsers other than IE.
        if (reader.readAsBinaryString) {
            reader.onload = function (e) {
                ProcessExcel(e.target.result);
            };
            reader.readAsBinaryString(fileUpload.files[0]);
        } else {
            //For IE Browser.
            reader.onload = function (e) {
                var data = "";
                var bytes = new Uint8Array(e.target.result);
                for (var i = 0; i < bytes.byteLength; i++) {
                    data += String.fromCharCode(bytes[i]);
                }
                ProcessExcel(data);
            };
            reader.readAsArrayBuffer(fileUpload.files[0]);
        }
    } else {
        alert("This browser does not support HTML5.");
    }
} else {
    alert("Please upload a valid Excel file.");
}
};

```



```

function ProcessExcel(data) {
    //Read the Excel File data.
    var workbook = XLSX.read(data, {
        type: 'binary'
    });

    //Fetch the name of First Sheet.
    var sheet_name = document.getElementById("Sheet");
    var sheet_name = sheet_name.value ;
    //sheet_name = workbook.SheetNames[13];

    var address_of_cell=[];

    for(i=12; i<15; ){
        var address_of_cell_1 = 'A'+ i ;
        var address_of_cell_2 = 'B'+ i ;
        var address_of_cell_3 = 'C'+ i ;
        var address_of_cell_4 = 'D'+ i ;
        var address_of_cell_5 = 'E'+ i ;
        var address_of_cell_6 = 'F'+ i ;
        var address_of_cell_7 = 'G'+ i ;
        var address_of_cell_8 = 'H'+ i ;
        var address_of_cell_9 = 'I'+ i ;
        var address_of_cell_10= 'G'+ i ;
        var address_of_cell_11= 'K'+ i ;
        var address_of_cell_12= 'L'+ i ;
        var address_of_cell_13= 'M'+ i ;
        var address_of_cell_14= 'N'+ i ;
        var address_of_cell_15= 'O'+ i ;
    }
}

```

```

var address_of_cell_16= 'P'+ i ;
var address_of_cell_17= 'Q'+ i ;

address_of_cell.push(address_of_cell_1);
address_of_cell.push(address_of_cell_2);
address_of_cell.push(address_of_cell_3);
address_of_cell.push(address_of_cell_4);
address_of_cell.push(address_of_cell_5);
address_of_cell.push(address_of_cell_6);
address_of_cell.push(address_of_cell_7);
address_of_cell.push(address_of_cell_8);
address_of_cell.push(address_of_cell_9);
address_of_cell.push(address_of_cell_10);
address_of_cell.push(address_of_cell_11);
address_of_cell.push(address_of_cell_12);
address_of_cell.push(address_of_cell_13);
address_of_cell.push(address_of_cell_14);
address_of_cell.push(address_of_cell_15);
address_of_cell.push(address_of_cell_16);
address_of_cell.push(address_of_cell_17);
i=i+1 ;

}

/* Get worksheet */
var worksheet = workbook.Sheets[sheet_name];

/* Find desired cell */

let winddata =" ;

```

```

j=0;
k=0;
windtable[j] = new Array();
for(i=0; i<address_of_cell.length; i++){
  if ((Math.floor(i/17))!=0 & (i%17)==0 ) {
    winddata = winddata + '\r' ;
    j= j+1 ;
    k=0;
    windtable[j] = new Array();
  }

  var desired_cell = worksheet[address_of_cell[i]];

  /* Get the value */

  var desired_value =desired_cell.v ;

  windtable[j][k] = desired_value;
  winddata = winddata + desired_value + ' ';
  k=k+1;
}

console.log(windtable);

localStorage.windstats = JSON.stringify(windtable);

const textToBLOB3 = new Blob([winddata], { type: 'text/plain' } );

```

```

const sFileName3 = 'Wind_data.txt';      // The file to save the data.

let newLink3 = document.createElement("a");
newLink3.download = sFileName3;

if (window.webkitURL !== null) {
    newLink3.href = window.webkitURL.createObjectURL(textToBLOB3);
}
else {
    newLink3.href = window.URL.createObjectURL(textToBLOB3);
    newLink3.style.display = "none";
    document.body.appendChild(newLink3);
}
newLink3.click();

};
</script>
</head>


```

```
<div id="dvExcel"></div>
</html>
```

## Αρχείο “map for data collection.html”

```
<!DOCTYPE html>
<title>Problem Marker</title>
<style type="text/css">
  html, body, #Map {
    position: fixed;
    top: 0;
    right: 0;
    width: 100%;
    height: 100%;
  }
</style>
  <form>
  </form>
<script src="http://www.openlayers.org/api/OpenLayers.js"></script>
<script type="text/javascript">
  const pointString1 = JSON.parse(localStorage.myArray1);
  const pointString2 = JSON.parse(localStorage.myArray2);
  const impactString1 = JSON.parse(localStorage.myArray1_impact);
  const impactString2 = JSON.parse(localStorage.myArray2_impact);
  const markerlonlat = JSON.parse(localStorage.markcoord);
  const markerlonlat_impact = JSON.parse(localStorage.markcoord_impact);
  var lonlat;
  var map,vectorLayer,selectMarkerControl,selectedFeature,markers, marker,x,y;
  var lat = 39.365951;
```

```

var lon = 22.951762;
var zoom = 13;
var position;
var coord= [];
var dist = [];
var fromProjection = new OpenLayers.Projection("EPSG:4326");// Transform from WGS 1984
var toProjection = new OpenLayers.Projection("EPSG:900913");// to Spherical Mercator Projection
var cntrposition = new OpenLayers.LonLat(lon, lat).transform( fromProjection, toProjection);
k=0;
l=0;
const glength = localStorage.getItem("lgrid");
const xlen1 = localStorage.getItem("xgrid1");
const xlen2 = localStorage.getItem("xgrid2");
const ylen1 = localStorage.getItem("ygrid1");
const ylen2 = localStorage.getItem("ygrid2");

function init(){
    map = new OpenLayers.Map("Map",{
        controls :
        [ ] });

    var mapnik = new OpenLayers.Layer.OSM("MAP");
    markers = new OpenLayers.Layer.Markers( "Markers" );
    map.addLayers([mapnik,markers]);
    map.addControl(new OpenLayers.Control.Navigation());
    map.addControl(new OpenLayers.Control.ZoomPanel());
    map.addControl(new OpenLayers.Control.KeyboardDefaults());
    var click = new OpenLayers.Control.Click();
    map.addControl(click);
    click.activate();

```

```

// grammes gia thn dhmiourgia grid
// Add Vector
var vector = new OpenLayers.Layer.Vector("Grid");
map.addLayers([vector]);
var xlength1 = xlen1;
var xlength2 = xlen2;
var ylength1 = ylen1;
var ylength2 = ylen2;
var swapx ;
var swapy ;

if(xlength2 < xlength1 ){
    swapx = xlength1 ;
    xlength1 = xlength2 ;
    xlength2 = swapx ;
}
if(ylength2 < ylength1 ){
    swapy = ylength1 ;
    ylength1 = ylength2 ;
    ylength2 = swapy ;
}
var longitude ;
var latitude ;

for (longitude = +ylength1;longitude <= +ylength2;) {
    for (latitude = +xlength1; latitude <= +xlength2;) {
        var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);
        latitude = +latitude + 0.000008984*glength ;

```

```

        var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        var line1 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point1, end_point1]));

        vector.addFeatures([line1]);
    }

    var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

    longitude = +longitude + 0.000011588*glength ;

    var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

    var line2 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point2,
end_point2]));

    vector.addFeatures([line2]);
}

for (latitude = +xlength1; latitude <= +xlength2;) {
    for (longitude = +ylength1; longitude <= +ylength2;) {
        var start_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        longitude = +longitude + 0.000011588*glength ;

        var end_point2 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

        var line2 = new OpenLayers.Feature.Vector(new
OpenLayers.Geometry.LineString([start_point2, end_point2]));

        vector.addFeatures([line2]);
    }

    var start_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

    latitude = +latitude + 0.000008984*glength ;

    var end_point1 = new OpenLayers.Geometry.Point(longitude, latitude).transform(
fromProjection, toProjection);

    var line1 = new OpenLayers.Feature.Vector(new OpenLayers.Geometry.LineString([start_point1,
end_point1]));

```



```

    vector.addFeatures([line1]);
}

// ADD MARKERS ON TO THE MAP
for (i = 0; i<markerlonlat.length; i++){
    k= i+1;
    var coord = markerlonlat[i];
    marker = new OpenLayers.Marker(coord);
    markers.addMarker(marker);
}

for (i = 0; i<markerlonlat_impact.length; i++){
    k= i+1;
    var coord = markerlonlat_impact[i];
    marker = new OpenLayers.Marker(coord);
    markers.addMarker(marker);
}

// CENTER MAP
map.setCenter(cnrposition, zoom);
};

OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control, {
defaultHandlerOptions: {
'single': true,
'double': false,
'pixelTolerance': 0,
'stopSingle': false,
'stopDouble': false
},

```

```

initialize: function(options) {
    this.handlerOptions = OpenLayers.Util.extend(
        {}, this.defaultHandlerOptions
    );
    OpenLayers.Control.prototype.initialize.apply(
        this, arguments
    );
    this.handler = new OpenLayers.Handler.Click(
        this, {
            'click': this.trigger
        }, this.handlerOptions
    );
},
trigger: function(geo) {
    lonlat = map.getLonLatFromPixel(geo.xy);
    lonlat1= new OpenLayers.LonLat(lonlat.lon,lonlat.lat).transform(toProjection,fromProjection);

    for (i = 0; i<markerlonlat.length; i++){
        k=i+1;
        var popup = new OpenLayers.Popup("chicken",
            new OpenLayers.LonLat(pointString2[i],pointString1[i]),
            new OpenLayers.Size(73,20),
            "Position "+k,
            true);
        map.addPopup(popup);
    }

    for (i = 0; i<markerlonlat_impact.length; i++){
        k=i+1;
        var popup = new OpenLayers.Popup("chicken",

```

```

        new OpenLayers.LonLat(impactString2[i],impactString1[i]),
        new OpenLayers.Size(80,40),
        "High impact zone "+k,
        true);
    map.addPopup(popup);
}
}
});
</script>
<body onload="init();">
<div id="Map"></div>
</div>
</body>

```

### Αρχείο C++ Visual studio “Source.cpp”

```

#include <iostream>
#include <string>
#include <cmath>
#include <iomanip>
#include <vector>
// Add cplex
#include <ilcplex/ilocplex.h>

using namespace std;
int points_num;

std::vector<std::string> pointsArray{};
std::vector<std::string> distArray{};

```

```

std::vector<std::string> group1Array{};
std::vector<std::string> group2Array{};
std::vector<std::string> Wtot1Array{};
std::vector<std::string> Wtot2Array{};

std::vector<int> grouping1_column{};
std::vector<int> grouping2_column{};

typedef IloArray<IloNumVarArray> NumVar2d;
int main()
{

    ifstream distFile;
    ifstream pointsFile;
    ifstream Groups_1;
    ifstream Groups_2;
    ifstream Wtotal_1;
    ifstream Wtotal_2;

    string Dist_path;
    string Points_path;
    string Group1_path;
    string Wtotal1_path;
    string Group2_path;
    string Wtotal2_path;
    string Opt1_path;
    string Opt2_path;

```

```
cout << "Give path of Distance.txt file: ";  
getline(cin, Dist_path);  
distFile.open(Dist_path);  
cout << "\n";
```

```
cout << "Give path of formData.txt file: ";  
getline(cin, Points_path);  
pointsFile.open(Points_path);  
cout << "\n";
```

```
cout << "Give path of Groups point 1.txt file:";  
getline(cin, Group1_path);  
Groups_1.open(Group1_path);  
cout << "\n";
```

```
cout << "Give path of Wtotal 1.txt file:";  
getline(cin, Wtotal1_path);  
Wtotal_1.open(Wtotal1_path);  
cout << "\n";
```

```
cout << "Give path of Groups point 2.txt file:";  
getline(cin, Group2_path);  
Groups_2.open(Group2_path);  
cout << "\n";
```

```
cout << "Give path of Wtotal 2.txt file:";  
getline(cin, Wtotal2_path);  
Wtotal_2.open(Wtotal2_path);  
cout << "\n";
```

```
int count_lines = 0;
int max_column1 = 0;
int max_column2 = 0;

if (!pointsFile) {
    cout << "Unable to open file formData.txt";
    exit(1); // call system to stop
}
else {

    int Words_number = 0;

    string x;
    string y;

    while (!pointsFile.eof()) {

        pointsFile >> x;

        pointsArray.push_back(x);
        Words_number = Words_number + 1;
    }
}
```

```
int number_of_words = Words_number - 1; // -1 giati kanei mia extra epanalhpsk kai gaiti  
arxizoyme thn arthmish apo to 0 kai oxi apo to 1
```

```
points_num = stoi(pointsArray[pointsArray.size() - 9]);
```

```
}
```

```
if (!distFile) {
```

```
    cout << "Unable to open file Distance.txt";
```

```
    exit(1); // call system to stop
```

```
}
```

```
else {
```

```
    int Words_number = 0;
```

```
    string x;
```

```
    while (!distFile.eof()) {
```

```
        distFile >> x;
```

```
        distArray.push_back(x);
```

```
        Words_number = Words_number + 1;
```

```
    }
```

```
    int number_of_words = Words_number - 1;
```

```

}

// Create 2d array for the distance between points
std::vector< std::vector<double> > Dist(points_num, std::vector<double>(points_num, 0));

// Create arrays for the coordinates of all the points
std::vector<double> Cordx{ };
std::vector<double> Cordy{ };
std::vector<double> Coverage{ };

for (int i = 0; i < points_num; i++) {

    Cordx.push_back(0);
    Cordy.push_back(0);
    Coverage.push_back(0);
}

int k = 6;
for (int i = 0; i < points_num; i++) {

    for (int j = i; j < points_num; j++) {

        if (i == j) {
            Dist[i][j] = 0;
        }
    }
}

```



```

        else {
            Dist[i][j] = std::stof(distArray[k]);
            Dist[j][i] = std::stof(distArray[k]);
            k = k + 5;
        }
    }
}

```

```

// find max dist

```

```

double max_dist = 0;

```

```

for (int i = 0; i < points_num; i++) {

```

```

    for (int j = i; j < points_num; j++) {

```

```

        if (Dist[i][j] >= max_dist) {

```

```

            max_dist = Dist[i][j];

```

```

        }

```

```

    }

```

```

}

```

```

int l = 1;

```

```

for (int i = 0; i < points_num; i++) {

```

```

    l = l + 5;

```

```

    Cordx[i] = std::stof(pointsArray[l]);

```

```

    l = l + 1;

```

```

    Cordy[i] = std::stof(pointsArray[l]);

```

```

    l = l + 2;

```

```

    Coverage[i] = std::stof(pointsArray[l]);

```

```

}

```

```

char c = '|';
std::string s(1, c);

// Arxeia Groups
if (!Groups_1) {
    cout << "Unable to open file Group 1.txt";
    exit(1); // call system to stop
}
else {

    int Words_number = 0;

    string x;

    while (!Groups_1.eof()) {
        Groups_1 >> x;

        if (x == s) {
            count_lines = count_lines + 1;
            grouping1_column.push_back(0);
        }

        group1Array.push_back(x);
        Words_number = Words_number + 1;
    }
    count_lines = count_lines - 1;

```

```

int number_of_words = Words_number - 1;
int array_length = count_lines;

int column = 0;
int l = 0;
for (int i = 2; i < number_of_words; i++) {

    column = column + 1;
    if (group1Array[i] == s) {
        column = column - 1;
        grouping1_column[l] = column;
        l = l + 1;
        if (column > max_column1) {
            max_column1 = column;
        }
        column = 0;
    }
}
grouping1_column.pop_back();
count_lines = 0;

l = 0;
if (!Groups_2) {
    cout << "Unable to open file Group 2.txt";
    exit(1); // call system to stop
}
else {

```

```
int Words_number = 0;

string x;
int column = 0;

while (!Groups_2.eof()) {

    Groups_2 >> x;

    if (x == s) {
        count_lines = count_lines + 1;

        grouping2_column.push_back(0);
        l = l + 1;
    }

    group2Array.push_back(x);
    Words_number = Words_number + 1;

}

count_lines = count_lines - 1;
```

```

int number_of_words = Words_number - 1;
int array_length = count_lines;

column = 0;
l = 0;
for (int i = 2; i < number_of_words; i++) {

    column = column + 1;
    if (group2Array[i] == s) {
        column = column - 1;
        grouping2_column[l] = column;
        l = l + 1;
        if (column > max_column2) {
            max_column2 = column;
        }
        column = 0;
    }
}
grouping2_column.pop_back();

// Arxeia Wtotals
count_lines = 0;
if (!Wtotal_1) {
    cout << "Unable to open file Distance (1).txt";
    exit(1); // call system to stop
}
else {

```

```

int Words_number = 0;
string x;
while (!Wtotal_1.eof()) {
    Wtotal_1 >> x;
    if (x == s) {
        count_lines = count_lines + 1;
    }

    Wtot1Array.push_back(x);
    Words_number = Words_number + 1;
}

count_lines = count_lines - 1;

int number_of_words = Words_number - 1;
int array_length = count_lines;
}
count_lines = 0;
if (!Wtotal_2) {
    cout << "Unable to open file Distance (1).txt";
    exit(1); // call system to stop
}
else {

int Words_number = 0;
string x;
while (!Wtotal_2.eof()) {

```

```

        Wtotal_2 >> x;
        if (x == s) {
            count_lines = count_lines + 1;
        }

        Wtot2Array.push_back(x);
        Words_number = Words_number + 1;
    }

    count_lines = count_lines - 1;
    int number_of_words = Words_number - 1;

}

int array_length = count_lines;

int column1 = max_column1;
int column2 = max_column2;

std::vector< std::vector<int> > Groups1(array_length, std::vector<int>(column1, 0));
std::vector< std::vector<double> > Wtot1(array_length, std::vector<double>(column1, 0));
std::vector< std::vector<int> > Groups2(array_length, std::vector<int>(column2, 0));
std::vector< std::vector<double> > Wtot2(array_length, std::vector<double>(column2, 0));

for (int i = 0; i < array_length; i++) {

    for (int j = 0; j < column1 - grouping1_column[i]; j++) {

        Groups1[i].pop_back();
        Wtot1[i].pop_back();
    }
}

```

```

    }

}

for (int i = 0; i < array_length; i++) {

    for (int j = 0; j < column2 - grouping2_column[i]; j++) {

        Groups2[i].pop_back();
        Wtot2[i].pop_back();
    }
}

```

// pernaw sta arxeia Groups1,2 kai Wtot1,2 tis times tous

```

l = 1;
k = 2;
for (int i = 0; i < array_length; i++) {

    int arrSize = Groups1[i].size();

    for (int j = 0; j < arrSize; j++) {
        Groups1[i][j] = std::stoi(group1Array[k]);
        Wtot1[i][j] = std::stod(Wtot1Array[l]);

        l = l + 1;
        k = k + 1;
    }
}

```



```

        k = k + 1;
        l = l + 1;
    }

    l = 1;
    k = 2;

    for (int i = 0; i < array_length; i++) {

        int arrSize = Groups2[i].size();

        for (int j = 0; j < arrSize; j++) {
            Groups2[i][j] = std::stoi(group2Array[k]);
            Wtot2[i][j] = std::stod(Wtot2Array[l]);

            l = l + 1;
            k = k + 1;
        }

        k = k + 1;
        l = l + 1;
    }

    std::vector<double> Wtotal_final(points_num, 0);

    int hold_value_int = 0;

```

```

double hold_value_double = 0;

for (int i = 0; i < array_length; i++) {

    for (int j = 0; j < Groups1[i].size(); j++) {

        Wtotal_final[Groups1[i][j] - 1] = Wtot1[i][j];

        //cout << Wtotal_final[Groups1[i][j] - 1] << endl;

    }

}

float max_Wtot = Wtot1[array_length - 1][Groups1[array_length - 1].size() - 1];

// CPLEX
double obj1;
double obj2;

//groups kathgoriopoishshs 1
IloEnv env1;
try {

    IloModel Model1(env1);
    IloCplex cplex1(env1);

    NumVar2d Kig1(env1, points_num);

```

```

for (int i = 0; i < points_num; i++) {

    Kig1[i] = IloNumVarArray(env1, array_length, 0, 1, ILOINT);
}

for (int l = 0; l < points_num; l++) {

    IloExpr expr1(env1);
    for (int g = 0; g < array_length; g++) {

        expr1 += Kig1[l][g];

    }

    Modell.add(expr1 <= 1);
}

for (int g = 0; g < array_length; g++) {

    IloExpr expr2(env1);
    for (int l = 0; l < points_num; l++) {

        expr2 += Kig1[l][g];

    }

    Modell.add(expr2 == 1);
}

```

```

for (int g = 0; g < array_length; g++) {

    IloExpr expr3(env1);
    for (int i = 0; i < Groups1[g].size(); i++) {

        expr3 += Kig1[Groups1[g][i] - 1][g];

    }

    Model1.add(expr3 == 1);
}

// Antikeimenikh
IloExpr Exp_Final1(env1);

for (int g = 0; g < array_length; g++) {

    for (int i = 0; i < points_num; i++) {

        for (int j = 0; j < points_num; j++) {

            Exp_Final1 += (Wtotal_final[i] / max_Wtot)*Kig1[i][g] +
(Dist[i][j] / max_dist)*Kig1[i][g] * Kig1[j][g];

        }

    }

}

Model1.add(IloMaximize(env1, Exp_Final1));

//Solve
cplex1.extract(Model1);

```

```

cplex1.solve();
obj1 = cplex1.getObjValue();
cout << obj1 << endl;
////////////////////////////////////
// Pass to txt file
std::vector<int> Opt_points1 {};
std::vector<          std::vector<int>          >          Coverage_array1(points_num,
std::vector<int>(points_num, 0)); //pinkas gia thn katagrafh coverage overcoverage
cout << "Give path to download optimal results for the first method ";
getline(cin, Opt1_path);
string hold_string;
hold_string = "/Optimal points1.txt";
Opt1_path = Opt1_path + hold_string;
std::ofstream Optimal_points1_file(Opt1_path);

for (int g = 0; g < array_length; g++) {

    for (int i = 0; i < points_num; i++) {
        int x = cplex1.getValue(Kig1[i][g]);
        if ( x == 1) {
            Opt_points1.push_back(i + 1);
        }
    }
}

int i;
for (int k = 0; k < Opt_points1.size(); k++) {
    i = Opt_points1[k]-1;
    for (int j = 0; j < points_num; j++) {

        if (Coverage[i]>= Dist[i][j]) {

```

```

        Coverage_array1[i][j] = 1;
    }
}
int coverage = 0;
int overcoverage = 0;

for (int j = 0; j < points_num; j++) {
    int sum = 0;
    for (int i = 0; i < points_num; i++) {

        sum = sum + Coverage_array1[i][j];

    }
    if (sum >= 1) {
        coverage = coverage + 1;

    }
    if (sum >= 2) {
        overcoverage = overcoverage + 1;

    }
}

//////////
//////////Pass data to file
//////////

double overcoverage_percent1 = double(overcoverage) / double(points_num) * 100;
double coverage_percent1 = double(coverage) / double(points_num) *100;
cout << overcoverage_percent1;

```

```
Optimal_points1_file << "Objective value: " << obj1 << " Coverage: " <<
coverage_percent1 << " % Overcoverage: " << overcoverage_percent1 << " % " << "\n";
```

```
for (int g = 0; g < array_length; g++) {

    for (int i = 0; i < points_num; i++) {

        int x = cplex1.getValue(Kig1[i][g]);

        if (x == 1) {

            Optimal_points1_file << "Group_" << g + 1 << " Point: " << i +
1 << " Coord: " << Cordx[i] << " " << Cordy[i] << " ";

            Opt_points1.push_back(i + 1);

        }

    }

    Optimal_points1_file << "\n";

}
```

```
Optimal_points1_file.close();

}

catch (IloException& e) {

    cerr << "concert exception caught:" << e << endl;

}

catch (...) {

    cerr << "Unknown exception caught" << endl;

}
```

```
////////////////////////////////////
```

```
IloEnv env2;

try {

    IloModel Model2(env2);

    IloCplex cplex2(env2);

    NumVar2d Kig2(env2, points_num);

    for (int i = 0; i < points_num; i++) {
```

```

        Kig2[i] = IloNumVarArray(env2, array_length, 0, 1, ILOINT);
    }

    for (int l = 0; l < points_num; l++) {

        IloExpr expr1(env2);
        for (int g = 0; g < array_length; g++) {

            expr1 += Kig2[l][g];
        }
        Model2.add(expr1 <= 1);
    }
    for (int g = 0; g < array_length; g++) {

        IloExpr expr2(env2);
        for (int l = 0; l < points_num; l++) {

            expr2 += Kig2[l][g];

        }

        Model2.add(expr2 == 1);
    }

    for (int g = 0; g < array_length; g++) {

        IloExpr expr3(env2);
        for (int i = 0; i < Groups2[g].size(); i++) {

```



```

        expr3 += Kig2[Groups2[g][i] - 1][g];

    }

    Model2.add(expr3 == 1);
}

// Antikeimenikh
IloExpr Exp_Final2(env2);

for (int g = 0; g < array_length; g++) {

    for (int i = 0; i < points_num; i++) {

        for (int j = 0; j < points_num; j++) {

            Exp_Final2 += (Wtotal_final[i] / max_Wtot)*Kig2[i][g] +
(Dist[i][j] / max_dist)*Kig2[i][g] * Kig2[j][g];
        }
    }
}

Model2.add(IloMaximize(env2, Exp_Final2));

//Solve
cplex2.extract(Model2);
cplex2.solve();
obj2 = cplex2.getObjValue();
cout << obj2 << endl;

// Pass to txt file
std::ofstream Optimal_points2_file;
std::vector<int> Opt_points2{};

```

```

std::vector<std::vector<int>> Coverage_array2(points_num,
std::vector<int>(points_num, 0)); //pinkas gia thn katagrafh coverage overcoverage

cout << "Give path to download optimal results for the second method ";

getline(cin, Opt2_path);

string hold_string;

hold_string = "/Optimal points2.txt";

Opt2_path = Opt2_path + hold_string;

Optimal_points2_file.open(Opt2_path);

for (int g = 0; g < array_length; g++) {

    for (int i = 0; i < points_num; i++) {

        int x = cplex2.getValue(Kig2[i][g]);

        if (x == 1) {

            Opt_points2.push_back(i + 1);

        }

    }

}

int i;

for (int k = 0; k < Opt_points2.size(); k++) {

    i = Opt_points2[k]-1;

    for (int j = 0; j < points_num; j++) {

        if (Coverage[i] >= Dist[i][j]) {

            Coverage_array2[i][j] = 1;

        }

    }

}

int coverage = 0;

int overcoverage = 0;

```

```

for (int j = 0; j < points_num; j++) {
    int sum = 0;
    for (int i = 0; i < points_num; i++) {
        sum = sum + Coverage_array2[i][j];
    }
    if (sum >= 1) {
        coverage = coverage + 1;
    }
    if (sum >= 2) {
        overcoverage = overcoverage + 1;
    }
}

////////////////////////////////////
////////////////////////////////////Pass data to file
////////////////////////////////////

double overcoverage_percent2 = double(overcoverage) / double(points_num) * 100;
double coverage_percent2 = double(coverage) / double(points_num) * 100;
cout << overcoverage_percent2;

Optimal_points2_file << "Objective value: " << obj2 << " Coverage: " <<
coverage_percent2 << " % Overcoverage: " << overcoverage_percent2 << " % " << "\n";

for (int g = 0; g < array_length; g++) {

    for (int i = 0; i < points_num; i++) {
        int x = cplex2.getValue(Kig2[i][g]);
        if (x == 1) {
            Optimal_points2_file << "Group_" << g + 1 << " Point: " << i +
1 << " Coord: " << Cordx[i] << " " << Cordy[i] << " ";
            Opt_points2.push_back(i + 1);
        }
    }
}

```

```
        Optimal_points2_file << "\n";
    }
}
catch (IloException& e) {
    cerr << "concert exception caught:" << e << endl;
}
catch (...) {
    cerr << "Unknown exception caught" << endl;
}
}
```

## **Βιβλιογραφία**

1. R.S. Brodkey & H.C. Hershey , «Φαινόμενα Μεταφοράς». Εκδ Τζιόλας, 2017
2. Omidvarborna; et al. (2015). "Recent studies on soot modeling for diesel combustion". *Renewable and Sustainable Energy Reviews*. 48: 635–647
3. Jong Shik Chung, Sang Chul Paik, Hee Sung Kim, Deok Sung Lee, In Sik Nam . «Removal of H<sub>2</sub>S and/or SO<sub>2</sub> by catalytic conversion technologies» . *Catalysis Today* Volume 35, Issues 1–2, 14 March 1997, Pages 37-43
4. Jalal Abadi, Ali Zarrabi ,Farid Hashemi ,Amirhossein Zabolian, Masoud Najafi, Maliheh Entezari, Kiavash Hushmandi, Amir Reza Aref, Haroon Khan, Pooyan Makvandi, Saeed Ashrafizaveh, Tahereh Farkhondeh, Milad Ashrafizadeh, Saeed Samarghandian, Michael R. Hamblin. «The role of SOX family transcription factors in gastric cancer» *International Journal of Biological Macromolecules* Volume 180, 1 June 2021, Pages 608-624
5. Yu, Chuck; Crump, Derrick (1998). «A review of the emission of VOCs from polymeric materials used in buildings». *Building and Environment* **33** (6): 357–74
6. Sigel, Astrid and Sigel, Roland K. O. (2009). *Metal-Carbon Bonds in Enzymes and Cofactors*. Royal Society of Chemistry. σελ. 243
7. Meislich H., Nechamkin H., Sharefkin J.: *Οργανική Χημεία, μετάφρ. Α. Βάρβογλης*, Αθήνα 1983
8. Omidvarborna (December 2015). «NO<sub>x</sub> emissions from low-temperature combustion of biodiesel made of various feedstocks and blends». *Fuel Processing Technology* 140: 113–118
9. Non-exhaust Particulate Emissions from Road Transport. OECD. 2020
10. Lu, Jackson G (April 2020). "Air pollution: A systematic review of its psychological, economic, and social effects". *Current Opinion in Psychology*. **32**: 52–65.
11. Cohen AJ, Ross Anderson H, Ostro B, Pandey KD, Krzyzanowski M, Künzli N, et al. (2005). "The global burden of disease due to outdoor air pollution". *Journal of Toxicology and Environmental Health. Part A*. **68** (13–14): 1301–7.
12. Nicolas Reiminger ,Xavier Jurado ,José Vazquez ,Cédric Wemmert ,Nadège Blond ,Matthieu Dufresne, Jonathan Wertel. «Effects of wind speed and atmospheric stability on the air pollution reduction rate induced by noise barriers». *Journal of Wind Engineering and Industrial Aerodynamics* Volume 200, May 2020, 104160
13. Priyantha Edussuriya, Andrew Chan, Aldo Malvin . «Urban morphology and air quality in dense residential environments: Correlations between morphological parameters and air pollution at street-level» .*Journal of Engineering Science and Technology* Vol. 9, No. 1 (2014) 64 – 80 School of Engineering, Taylor’s University.
14. Γεώργιος Κ.Δ Σαχαρίδης , Κωνσταντίνος Γουργουλιάνης. Δελτίο Τύπου Green Your Air, Πανεπιστήμιο Θεσσαλίας, Ιούνιος 2020
15. <https://op.europa.eu/webpub/eca/special-reports/air-quality-23-2018/en/>

16. Μάρκος Ασσαέλ, Δημήτρης Μελάς, Κωνσταντίνος Καρατζάς, Χαρίσιος Αχιλλιάς, Κωνσταντίνος Κακοσίμος. «Καταγραφή και Μελέτη αιωρούμενων σωματιδίων στο Πολεοδομικό Συγκρότημα της Θεσσαλονίκης», Τεχνικό Επιμελητήριο Ελλάδος Τμήμα Κεντρικής Μακεδονίας.
17. [https://www.meteoblue.com/en/weather/historyclimate/climatemodelled/volos\\_greece\\_251833](https://www.meteoblue.com/en/weather/historyclimate/climatemodelled/volos_greece_251833)
18. Εφημερίδα της Κυβερνήσεως Τεύχος Α.Α.Π, Αριθμός Φύλλου 237, 4/11/2016
19. Προεδρικό Διάταγμα υπ' αριθμ. 41/2018, ΦΕΚ 80/Α/7-5-2018 (Κανονισμός Πυροπροστασίας Κτιρίων)
20. IBM ILOG CPLEX Optimization Studio CPLEX User's Manual Version 12 Release 8
21. «Green Your Air: Η ατμοσφαιρική ρύπανση στην πόλη του Βόλου». Πανεπιστήμιο Θεσσαλίας. Αρχαιοθετήθηκε από το πρωτότυπο στις 11 Ιουλίου 2020. Ανακτήθηκε στις 11 Ιουλίου 2020.
22. Drew Fryer, Chloe Chan and Tom Crocker ,(2016).Nerves of steel.CDB driving sustainable technologies
23. [https://www.who.int/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
24. International Journal of Biological Macromolecules Volume 180, 1 June 2021, Pages 608-624. «The role of SOX family transcription factors in gastric cancer»
25. «Spatiotemporal distribution characteristics and potential sources of VOCs at an industrial harbor city in southern Taiwan: Three-year VOCs monitoring data analysis» , Journal of Environmental Management Volume 303, 1 February 2022, 114259
26. Weinstock, B.; Niki, H. (1972). "Carbon Monoxide Balance in Nature". Science 176 (4032): 290–2. Bibcode:1972Sci...176..290W. doi:10.1126/science.176.4032.290
27. «Αιωρούμενα σωματίδια PM2.5 και PM10 στην ατμόσφαιρα της Θεσσαλονίκης: Επίπεδα συγκεντρώσεων-Χημική σύσταση». Χρυσικού Λ., Αργυρόπουλος Γ., Φλαρούντζου Α. , Τερζή Ε., Κούρας Α., Σοφωνίου Μ., Σαμαρά, Κ. , Νικολάου Κ. και Βαβατζανίδης Α.
28. Carrington, Damian; McMullan, Lydia; Blight, Garry; Roberts, Simon; Hulley-Jones, Frank (17 May 2019). "Revealed: air pollution may be damaging every organ in the body"
29. Edussuriya, P., Chan, A. and Ye, A. (2009). Urban morphology and air quality in dense residential environments of Hong Kong. Part I: District level analysis. Atmospheric Environment, 45(27), 4789-4803
30. Εμμανουέλλα Ρεμουντάκη . Αέρας και ατμοσφαιρική ρύπανση ,2010.