



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

# Μελέτη και Ανάθεση Πόρων σε Χωρικά Δίκτυα Παρακολούθησης Παρουσία Εμποδίων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
Χρήστος Ν. Τσανικίδης

Επιβλέπων: Συμεών Παπαβασιλείου  
Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2018





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

# Μελέτη και Ανάθεση Πόρων σε Χωρικά Δίκτυα Παρακολούθησης Παρουσία Εμποδίων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χρήστος Ν. Τσανικίδης

**Επιβλέπων:** Συμεών Παπαβασιλείου  
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Μαρτίου 2018.

.....  
Συμεών Παπαβασιλείου  
Καθηγητής Ε.Μ.Π.

.....  
Θεοδώρα Βαρβαρίγου  
Καθηγήτρια Ε.Μ.Π.

.....  
Ιωάννα Ρουσσάκη  
Επίκουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάρτιος 2018

.....  
Τσανικίδης Χρήστος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© Τσανικίδης Χρήστος, 2018

Με επιφύλαξη παντός δικαιώματος - All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς το συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Στην εργασία αυτή μελετάται ένα πρόβλημα κάλυψης ή παρακολούθησης του χώρου από αισθητήρες τοποθετημένους τυχαία σε ένα περιβάλλον με κυρτά αδιαφανή εμπόδια κατανεμημένα τυχαία. Οι αισθητήρες μπορούν να καλύπτουν τα σημεία ενός δίσκου με κέντρο αυτούς, στα οποία διαθέτουν ορατότητα γραμμής (line of sight). Οι αισθητήρες έχουν την δυνατότητα να μεταβάλλουν την ακτίνα κάλυψής τους, και με την αύξηση αυτής καταναλώνουν περισσότερη ενέργεια. Ο στόχος μας είναι να μπορούμε να παρακολουθούμε ένα μεγάλο ποσοστό του χώρου χωρίς να καταναλώνουμε πολλή ενέργεια. Για να πετύχουμε το ζητούμενο μοντελοποιήσαμε το πρόβλημα ως ένα πρόβλημα μεγιστοποίησης του λόγου του συνεργατικά καλυπτόμενου εμβαδού δια τη συνολική καταναλισκόμενη ενέργεια στο υποσύνολο των έγκυρων αναθέσεων των ακτίνων, που έχουν πάνω από ένα προκαθορισμένο ποσοστό κάλυψης. Έπειτα διατυπώσαμε ένα απλούστερο εναλλακτικό κριτήριο, όπου θέλαμε να ελαχιστοποιήσουμε την καταναλισκόμενη ενέργεια στο χώρο των λύσεων που έχουν τουλάχιστον το όριο κάλυψης που απαιτούμε. Δείξαμε ότι αυτό το πρόβλημα έχει λύση σε ένα μικρότερο σύνολο λύσεων στο οποίο η απλούστερη διατύπωση ταυτίζεται με την αρχική του λόγου εμβαδού δια ενέργεια. Επιπλέον είδαμε ότι ο μικρότερος αυτός χώρος λύσεων είναι υποσύνολο του συνόρου της αρχικής εφικτής περιοχής. Έπειτα αναπτύξαμε τρεις μεθόδους για την αντιμετώπιση των προβλημάτων αυτών, γύρω από την κεντρική ιδέα προτίμησης κόμβων οι οποίοι με μικρές διαφοροποιήσεις της ακτίνας έχουν μεγάλες διαφοροποιήσεις στο λόγο Εμβαδού δια Ενέργεια. 1) Η πρώτη μέθοδος, του Άπληστου Κεντρικοποιημένου Αλγόριθμου, ξεκινάει από μια μηδενική κατάσταση για όλες τις ακτίνες των αισθητήρων και κάθε φορά επιλέγει να αυξήσει τον αισθητήρα με τον καλύτερο λόγο εμβαδού δια ενέργειας. Ο αλγόριθμος τερματίζει όταν περάσει το όριο κάλυψης που θέλουμε. 2) Έπειτα αναπτύξαμε έναν Τυχαίοποιημένο Άπληστο Αλγόριθμο ο οποίος επιλέγει από ένα τυχαίο υποσύνολο των κόμβων, τον κόμβο που θα αυξηθεί. 3) Τέλος ορίσαμε έναν Κατανεμημένο Τοπικό Αλγόριθμο, ο οποίος ορίζει έναν γράφο που εκφράζει τις εξαρτήσεις μεταξύ των κόμβων, και βασισμένοι σε αυτόν οι κόμβοι αποφασίζουν εάν θα μεταβάλλουν ή όχι την ακτίνα τους ανεξάρτητα. Ορίσαμε επίσης ένα κατανεμημένο κριτήριο απόφασης διακοπής αύξησης της ακτίνας ανεξάρτητα για κάθε κόμβο. Ολοκληρώσαμε την εργασία αυτή με εκτεταμένες προσομοιώσεις, όπου και χρησιμοποιήθηκαν μετρικές για την αξιολόγηση των αποτελεσμάτων.

**Λέξεις-Κλειδιά:** Ευρετικές Μέθοδοι, Βελτιστοποίηση, Χωρικά Δίκτυα Παρακολούθησης, Ανάθεση Πόρων, Προβλήμα Κάλυψης



# Abstract

In this work, we are studying a problem of coverage or monitoring of the space using sensors deployed randomly in an environment, where randomly positioned opaque convex obstacles exist. Sensors can cover the points of a disk which are in a line of sight from the sensor. Sensors can increase or decrease their sensing radius, and an increase (decrease) of the radius results in increased (decreased) energy consumption. Our objective is to monitor a high percentage of the space without spending much energy. To satisfy those requirements, we have modeled this problem as an optimization problem of the ratio of the total area covered and total energy consumed, in the subset of valid radii assignments for which we have at least a certain predecided coverage percentage. We have continued by defining a second, simpler optimization problem, where only the minimization of energy consumed is required, in the same feasible set. We have shown that this problem can only have its solution in a subset of the boundary of the feasible set. Furthermore, we have shown that the two problems are equivalent in a subset of the feasible solution, and have also shown that this subset is equal to the previously mentioned subset of the boundary in the case we require less than full coverage. We have then developed three methods to attempt to solve those problems, revolving around the main idea of preferring changing nodes which have a good changing area per changing energy ratio. 1) The first method, of the Greedy Centralized Algorithm, starts by setting radii of sensors equal to zero, and then, in each iteration it increases the radius of the sensor with the largest changing area per changing energy ratio. The algorithm terminates when the coverage threshold is surpassed. 2) Then, we continued by providing a Randomized Greedy Algorithm, which selects the node with the best ratio among a random subset of nodes. 3) Lastly, we have provided a Distributed Local Algorithm, where nodes, decide independently to either increase or not their radius based on a graph which expresses their interdependencies. A distributed criterion for the determination of the per-node termination of the algorithm is described. We have concluded this work with extensive simulations, and have reviewed the results using several metrics.

**Keywords:** Heuristics, Optimization, Spatial Monitoring Networks, Resource Allocation, Coverage Problem





# Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Συμεών Παπαβασιλείου, ο οποίος μου προσέφερε τη δυνατότητα να κάνω έρευνα σε ένα ρεαλιστικό πλαίσιο, προσομοιώνοντας σε ένα μέρος την καθημερινότητα ενός ερευνητή, όπου και απέκτησα την αυτοπεποίθηση ότι θα είναι κατάλληλες για εμένα οι διδακτορικές σπουδές. Πέρα από την ώθηση για καλή δουλειά, με ρεαλιστικές προσδοκίες, από έναν καθηγητή που αποτελεί για εμένα πρότυπο ερευνητή, ο κ. Παπαβασιλείου με αλτρουισμό με συμβούλευε σε πολλά επίπεδα σχετικά με την ακαδημαϊκή ζωή εύστοχα και με απόλυτη ειλικρίνεια και ευγένεια.

Θα ήθελα να ευχαριστήσω επίσης θερμά τον Μεταδιδακτορικό ερευνητή Βασίλη Καρυώτη και την Υποψήφια διδάκτορα Μαργαρίτα Βιτοροπούλου για τη συνεργασία μας. Τους δύο ερευνητές τους γνώρισα στα πλαίσια του μαθήματος των Κοινωνικών Δικτύων, όπου και ήδη είχα σχηματίσει θετική εικόνα ως προς τις γνώσεις τους, τον ενθουσιασμό τους για το αντικείμενο τους και τη στάση τους ως προς το μάθημα και το φοιτητή. Στα πλαίσια της συνεργασίας μου, με τον Βασίλη να είναι δύο βήματα, και τη Μαργαρίτα ένα βήμα μπροστά από εμένα, οι οπτικές τους και τα σχόλια τους είχαν αξία σε δύο άξονες για εμένα. Η συμμετοχή τους στο έργο αυτό ήταν διαρκής και καθοριστική για τη θετική πορεία του. Έμαθα πολλά τόσο από τον Βασίλη όσο και από τη Μαργαρίτα και εύχομαι η συνέχεια μου να γίνει σε ένα αντίστοιχο κλίμα και επίπεδο με αυτό που δημιούργησαν οι δυο ερευνητές όπου και είχα την ευκαιρία να μυηθώ στη συνεργατική έρευνα.

Με αφορμή την ολοκλήρωση των σπουδών μου με αυτή την εργασία, θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές μου. Από κάθε έναν είχα κάτι να μάθω. Πάρα πολλοί ήταν εκείνοι από τους οποίους έλαβα όχι μόνο πληροφορίες, αλλά γνώση και τρόπο σκέψης. Ειδικά όμως θα ήθελα να κάνω αναφορά και σε έναν καθηγητή που με ενέπνευσε με τις διαλέξεις του λόγω της φιλοσοφίας της σκέψης του, την εκφραστικότητα του και την ανθρώπινη πλευρά του, τον κ. Δημήτρη Φωτάκη.

Θα ήθελα να ευχαριστήσω όλους εκείνους τους ανθρώπους που επηρέαζαν την πορεία μου στο Μετσόβιο, φυσικά ξεκινώντας από τους γονείς μου οι οποίοι με στήριξαν ανιδιοτελώς και είχα την ευκαιρία να αφιερωθώ στις σπουδές μου χωρίς την ανάγκη να εργαστώ. Έπειτα τους συμφοιτητές μου, οι οποίοι με παρακινούσαν να γίνομαι καλύτερος μαζί με εκείνους. Σε αυτή τη σχολή το συμπέρασμα μου ήταν πως το άτομο που κάθεται δίπλα μου είναι ένας αφανής ήρωας. Κάθε φορά που οι συνθήκες επέτρεπαν να γνωρίσω περισσότερο τον άνθρωπο αυτό, πάντα έβρισκα κάποιον με βάθος, με όποιον τρόπο και εάν αυτό εκφραζόταν.

Τέλος, ευχαριστώ το Εθνικό Μετσόβιο Πολυτεχνείο. Τα όσα μου προσφέρθηκαν μέσα από αυτό το ίδρυμα είναι ανεκτίμητα και θα είναι πάντα κομμάτι του ποιός είμαι πως σκέφτομαι και που βαδίζω. Ήταν χαρά μου να είμαι φοιτητής της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.



# Περιεχόμενα

<b>1</b>	<b>ΕΙΣΑΓΩΓΗ</b>	<b>4</b>
1.1	Οργάνωση . . . . .	5
<b>2</b>	<b>ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ</b>	<b>7</b>
2.1	Χωρικά Δίκτυα - Γεωμετρικοί Γράφοι . . . . .	7
2.1.1	Γραφοθεωρητικοί Ορισμοί . . . . .	7
2.1.2	Γεωμετρικοί Γράφοι . . . . .	7
2.1.3	Γράφοι Τομής . . . . .	8
2.2	Σημειακές Διαδικασίες . . . . .	8
2.3	Μαθηματική Βελτιστοποίηση . . . . .	10
2.3.1	Εισαγωγικές Έννοιες . . . . .	10
2.3.2	Βελτιστοποίηση Πολλαπλών Κριτηρίων . . . . .	10
2.3.3	Κυρτή και Μη-Κυρτή Βελτιστοποίηση . . . . .	12
2.4	Ασύρματα Δίκτυα Αισθητήρων . . . . .	13
2.4.1	Μεταβίβαση Πληροφορίας . . . . .	13
2.4.2	Είδη Αίσθησης . . . . .	14
2.4.3	Αρχιτεκτονική Αισθητήρων . . . . .	14
2.4.4	Πηγή, Κατανάλωση και Διαχείριση Ενέργειας . . . . .	15
2.4.5	Οπτικά Δίκτυα Αισθητήρων . . . . .	16
<b>3</b>	<b>ΣΧΕΤΙΚΟ ΕΡΓΟ</b>	<b>17</b>
3.1	Έλεγχος Τοπολογίας . . . . .	17
3.2	Προβλήματα Κάλυψης . . . . .	17
3.3	Χρονοπρογραμματισμός Αισθητήρων . . . . .	18
3.4	Μοντέλα Κάλυψης . . . . .	18
3.5	Ανίχνευση Κάλυψης . . . . .	18
3.6	Σχέση Κάλυψης Αίσθησης και Κάλυψης Επικοινωνίας . . . . .	19
3.7	Τοποθέτηση Αισθητήρων . . . . .	20
3.8	Κατανεμημένη ή Κεντρική Σχεδίαση . . . . .	20
<b>4</b>	<b>ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ</b>	<b>21</b>
4.1	Μοντελοποίηση του Χώρου . . . . .	21
4.2	Μοντελοποίηση των Εμποδίων . . . . .	21
4.3	Μοντελοποίηση των Αισθητήρων . . . . .	24
4.4	Δυνατές Επεκτάσεις του Μοντέλου . . . . .	25

4.5	Τυποποίηση του Προβλήματος . . . . .	25
4.5.1	Κλασματική Προσέγγιση . . . . .	25
4.5.2	Απλοποιημένη Τυποποίηση του Προβλήματος . . . . .	26
4.5.3	Λύση της Απλοποιημένης Εκδοχής στο Περιορισμένο Σύνολο .	28
4.5.4	Μορφή της Συνάρτησης Ενέργειας . . . . .	30
4.5.5	Μορφή της Συνάρτησης Κάλυψης . . . . .	31
<b>5</b>	<b>ΕΥΡΕΤΙΚΕΣ ΛΥΣΕΙΣ</b>	<b>34</b>
5.1	Προκαταρκτικές Έννοιες . . . . .	34
5.2	Άπληστος Κεντρικοποιημένος Αλγόριθμος . . . . .	38
5.2.1	Παρουσίαση και Εξήγηση Πρώτου Αλγόριθμου . . . . .	39
5.2.2	Προβληματικές Περιπτώσεις Πρώτου Αλγόριθμου . . . . .	42
5.2.3	Θεμελίωση της Επιλογής . . . . .	43
5.2.4	Θέματα Αποδοτικότητας Υλοποίησης . . . . .	45
5.3	Τυχαιοποιημένος Κεντρικοποιημένος Αλγόριθμος . . . . .	47
5.4	Κατανεμημένος Τοπικός Αλγόριθμος . . . . .	48
5.4.1	Επαγόμενος Γράφος Εξάρτησης . . . . .	48
5.4.2	Απόφαση Αύξησης Ακτίνας . . . . .	49
5.4.3	Κριτήριο Τερματισμού Αύξησης Ακτίνας . . . . .	50
5.4.4	Περιγραφή Αλγόριθμου . . . . .	51
5.5	Δυνατές Επεκτάσεις με Προσεγγίσεις Ολικής Βελτιστοποίησης . . . . .	53
<b>6</b>	<b>ΑΞΙΟΛΟΓΗΣΗ</b>	<b>55</b>
6.1	Μετρικές . . . . .	55
6.2	Αποτελέσματα Κεντρικοποιημένων Αλγόριθμων . . . . .	56
6.3	Αποτελέσματα Κατανεμημένου Αλγόριθμου . . . . .	57
6.4	Σύγκριση Κεντρικού και Τυχαιοποιημένου σε Κοινές Τοπολογίες . . .	58
6.5	Πειραματική Αξιολόγηση των Επιλογών του Βήματος $\Delta r$ . . . . .	59
<b>7</b>	<b>ΕΠΙΛΟΓΟΣ</b>	<b>63</b>
7.1	Σύνοψη και Συμπεράσματα . . . . .	63
7.2	Δυνατές Μελλοντικές Επεκτάσεις . . . . .	64
<b>8</b>	<b>ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ</b>	<b>67</b>

## Κατάλογος Σχημάτων

1	Μέτωπο Pareto . . . . .	12
2	Block διάγραμμα ενός WSN κόμβου . . . . .	15
3	Παραδείγματα περιπτώσεων κόμβου και ύπαρξη κενού στην κάλυψη . . . . .	19
4	Παράδειγμα περίπτωσης πλήρους κάλυψης και σχέσης $\tilde{S}$ με $\partial S$ . . . . .	29
5	Παράδειγμα χώρου καταστάσεων για δύο αισθητήρες . . . . .	31
6	Διάταξη περικυκλωμένου από εμπόδια αισθητήρα . . . . .	33
7	Εξαρτήσεις Διάφορων Μεγεθών . . . . .	34
8	Παράδειγμα πολύγωνου ορατότητας . . . . .	35
9	Παράδειγμα πεδίου ορατότητας . . . . .	36
10	Παράδειγμα μιας Ανάθεσης του Αλγόριθμου . . . . .	38
11	Πολυγωνικές συνιστώσες σχετικές με την κάλυψη ενός αισθητήρα . . . . .	39
12	Οπτικοποίηση του λόγου εμβαδού δια ενέργεια . . . . .	42
13	Οπτικοποίηση του λόγου εμβαδού δια ενέργεια με και χωρίς τη θεωρηση των άλλων αισθητήρων . . . . .	43
14	Προβληματική περίπτωση του κεντρικού αλγόριθμου . . . . .	44
15	Παράδειγμα Επαγόμενου Γράφου εξάρτησης. . . . .	49
16	Επαγόμενος Γράφος Εξάρτησης για διαφορετικές μέγιστες ακτίνες . . . . .	50
17	Ιστόγραμμα αποτελεσμάτων Κεντρικού Αλγόριθμου . . . . .	57
18	Ιστόγραμμα αποτελεσμάτων Τυχαιοποιημένου Αλγόριθμου . . . . .	57
19	Αποτελέσματα του Τοπικού Αλγόριθμου, με χρήση Scatterplot, για κάλυψη περιφέρειας 78% . . . . .	58
20	Σύγκριση της μεθόδου CHECK-ALL και CHECK-10-R σε 100 τοπολογίες στο λόγο κατανάλωσης . . . . .	59
21	Σύγκριση της μεθόδου CHECK-ALL και CHECK-10-R σε 100 τοπολογίες στο λόγο εμβαδού δια ενέργεια . . . . .	60
22	Σύγκριση της μεθόδου CHECK-ALL και CHECK-10-R σε 20 τοπολογίες στο λόγο κατανάλωσης, για 300 κόμβους. . . . .	61
23	Σύγκριση του λόγου κατανάλωσης για διαφορετικό βήμα αύξησης των αισθητήρων για 20 τοπολογίες . . . . .	61
24	Σύγκριση του πλήθους βημάτων πριν τη σύγκλιση για διαφορετικό βήμα αύξησης των αισθητήρων για 20 τοπολογίες . . . . .	62

# 1. ΕΙΣΑΓΩΓΗ

Στον κόσμο μας σήμερα παρατηρείται αύξηση των ασύρματων δικτύων αισθητήρων και παρακολούθησης. Τα δίκτυα αυτά αλληλεπιδρούν με το φυσικό κόσμο με πολλούς διαφορετικούς τρόπους. Το πρόβλημα της *Ανάθεσης Πόρων (Resource Allocation)* γενικότερα στα δίκτυα αυτά εμφανίζεται όλο και περισσότερο μιας και η ραγδαία εξέλιξη της τεχνολογίας επιτρέπει νέες εφαρμογές που έχουν ως αποτέλεσμα νέες προκλήσεις [23, 28], ιδιαίτερα στον αναπτυσσόμενο κλάδο του *Internet of Things (IoT)*, αλλά και στα *Wireless Sensor Networks (WSN)* [18]. Η επιλογή κατάλληλων αναθέσεων των πόρων μπορεί να έχει σημαντική επίδραση στην επέκταση της διάρκειας ζωής ενός δικτύου, πρόβλημα που είναι κεντρικό στα WSN και έχει άμεση σύνδεση με το κόστος αλλά και τη χρησιμότητα του δικτύου. Ταυτόχρονα η διάρκεια ζωής εξασφαλίζει συνεχόμενη λειτουργία (uptime) και λιγότερη παρέμβαση από τον ανθρώπινο παράγοντα.

Στα WSN, δυο πολύ βασικά ζητήματα, τα οποία και αλληλεπιδρούν με τους πόρους του συστήματος, και συγκεκριμένα τη διαθέσιμη ενέργεια, είναι η *Κάλυψη (Coverage)*, και η *Συνδεσιμότητα (Connectivity)* στο δίκτυο μας. Η Κάλυψη είναι ένα μέτρο του πόσο καλυμένος είναι ο χώρος μας από την αίσθηση ή παρακολούθηση που επιτελεί το δίκτυο, και είναι ένα κριτήριο της ποιότητας του δικτύου ως προς το σκοπό του. Από την άλλη, η έννοια της Συνδεσιμότητας εκφράζει τη συνεκτικότητα των κόμβων του δικτύου, δηλαδή τη δυνατότητα τους να επικοινωνούν μεταξύ τους, και να μεταβιβάζουν χρήσιμη πληροφορία με κάποιο σκοπό. Η ύπαρξη της Συνδεσιμότητας σε κάποιες εφαρμογές είναι βασική προϋπόθεση για κάθε κόμβο ώστε να έχει αυτός συνεισφορά στο δίκτυο - για παράδειγμα όταν συλλέγεται από αισθητήρες πληροφορία θερμοκρασίας σε διάφορα σημεία, κάθε κόμβος θα πρέπει να μεταβιβάσει σε έναν κεντρικό κόμβο (Base Station) την πληροφορία αυτή, πράγμα που απαιτεί συνδεσιμότητα. Τα προβλήματα της Συνδεσιμότητας και της Κάλυψης σχετίζονται μεταξύ τους όταν η ακτίνα αίσθησης  $R_s$  είναι μεταβλητή: πολλές φορές οι ερευνητές αντιμετωπίζουν παράλληλα την επιλογή των ακτίνων αίσθησης  $R_s$  και επικοινωνίας  $R_c$ , και σε ειδικές περιπτώσεις, όπως θα δούμε στη συνέχεια, μπορούμε λύνοντας το πρόβλημα της επιλογής ακτίνων αίσθησης, να επιλέξουμε ακτίνες επικοινωνίας που εξασφαλίζουν συνδεσιμότητα.

Στην εργασία αυτή θα ασχοληθούμε ειδικά με το παρακάτω ερώτημα: πως μπορούμε με ένα σύνολο κόμβων που “παρατηρούν” το χώρο στον οποίο έχουν οπτική ορατότητα γραμμής (line of sight) έως μια ακτίνα αίσθησης (sensing radius), να ορίσουμε τις ακτίνες αυτές ώστε να ελαχιστοποιήσουμε την κατανάλωση ενέργειας, εάν θεωρήσουμε ότι η μεγαλύτερη ακτίνα συνεπάγεται μεγαλύτερη κατανάλωση, ενώ μεγιστοποιούμε την κάλυψη. Το ότι μεγαλύτερη ακτίνα μπορεί να συνεπάγεται μεγαλύτερη κατανάλωση ενέργειας είναι μια αναμενόμενη περίπτωση όπου υπάρχει ένα δίλημμα (trade-off) για

το πόσο πρέπει να αυξήσουμε την ακτίνα πριν αυτή έχει ως αποτέλεσμα μια ενεργειακά ασύμφορη κατάσταση. Ως κίνητρο για αυτή τη μελέτη, κάποιος μπορεί να φανταστεί ένα δίκτυο όπου ένας κόμβος που “αισθάνεται περισσότερα”, λόγω μεγαλύτερης ακτίνας αίσθησης, θα επεξεργάζεται μεγαλύτερο όγκο πληροφορίας με περισσότερο θόρυβο και αβεβαιότητα, και ως εκ τούτου απαιτητικότερες τεχνικές επεξεργασίας σήματος που βελτιώνουν το σηματοθορυβικό λόγο (SNR) είναι αναγκαίες [25, 29]. Αυτές οι τεχνικές οδηγούν σε μεγαλύτερη κατανάλωση ενέργειας. Βέβαια αυτό δεν είναι το μοναδικό σενάριο - οπουδήποτε ισχύουν οι κεντρικοί περιορισμοί που περιγράφηκαν παραπάνω, οι λύσεις μας μπορούν να υιοθετηθούν δίχως πρόβλημα. Εδώ αξίζει να γίνει σύγκριση με τη δεύτερη κάλυψη, αυτή της επικοινωνίας, όπου κάθε κόμβος καλύπτει μια ακτίνα επικοινωνίας  $R_c$ , και όπου για να λάβει ένας γειτονικός κόμβος ένα μήνυμα που προσπαθούμε να μεταφέρουμε πρέπει να εκπέμπουμε με μια απαιτούμενη ισχύ, και συνεπώς εάν θέλουμε να αυξήσουμε τον δίσκο κάλυψης, πρέπει να καταναλώνουμε περισσότερη ενέργεια. Σε συγκεκριμένες εφαρμογές, όπως αυτή που περιγράφηκε ως κίνητρο, η ακτίνα αίσθησης μπορεί να λειτουργεί με κόστος έως και δυο τάξεις μεγέθους λιγότερες στην ενέργεια, όμως λόγω του ότι είναι μια συνεχής λειτουργία σε σχέση με τη συχνά σποραδική επικοινωνία μεταξύ κόμβων, είναι εξίσου σημαντική η βελτιστοποίηση της [20, 25].

Στη βιβλιογραφία, κατά την έως τώρα γνώση μας, δεν έχουν δοθεί προσεγγίσεις για την περίπτωση όπου λογίζονται εμπόδια - χαρακτηριστικό που αποτυπώνει και τη συνεισφορά μας αφού η απλή ύπαρξη των εμποδίων οδηγεί σε ένα πρόβλημα τελείως διαφορετικό από το αρχικό. Τα εμπόδια μας θεωρούμε ότι είναι αδιαφανή (opaque): ο χώρος τους δεν καλύπτεται και δεν υπάρχει η ανάγκη να καλυφθεί, και ένας αισθητήρας δεν έχει τη δυνατότητα να καλύψει χώρο πίσω από ένα εμπόδιο.

## 1.1. Οργάνωση

Η εργασία αυτή διαρθρώνεται ως εξής: Αρχικά θα παρουσιάσουμε το θεωρητικό υπόβαθρο που θα αξιοποιηθεί και το οποίο θεωρείται σημαντικό να συμπληρωθεί ως μη βασική γνώση. Στη συνέχεια θα προχωρήσουμε παρουσιάζοντας το σχετικό έργο. Αμέσως μετά θα μιλήσουμε για το μοντέλο του προβλήματος - θα κάνουμε αυστηρή την έννοια του εμποδίου, του χώρου ενδιαφέροντος, και του αισθητήρα. Θα ορίσουμε με τον απαραίτητο φερελισμό το πρόβλημα που θα προσπαθήσουμε να λύσουμε, και θα εξετάσουμε τα διάφορα λεπτά σημεία που εμφανίζονται στην τυποποίηση, και τις συνέπειες αυτών. Έπειτα θα συνεχίσουμε με το να δώσουμε συγκεκριμένες ευρετικές λύσεις για το πρόβλημα με διάφορες παραλλαγές για την εκάστοτε περίπτωση, με τους τρεις κεντρικούς άξονες να είναι ένας Κεντρικοποιημένος Άπληστος Αλγόριθμος, ένας Τυχαιοποιημένος Κεντρι-

κοινημένος Αλγόριθμος, και ένας Κατανεμημένος Τοπικός Αλγόριθμος. Τέλος, θα αξιολογήσουμε την εργασία μας με διάφορες μετρικές επίδοσης που θα οριστούν στη συνέχεια και στον Επίλογο θα γίνει μια σύνοψη όπου θα δοθούν κατευθύνσεις για την περαιτέρω επέκταση της παρούσας εργασίας.



## 2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Σε αυτήν την ενότητα θα ξεκινήσουμε με την εισαγωγή σε πεδία και έννοιες που σχετίζονται με το υπόψη πρόβλημα και μπορούν να αξιοποιηθούν, είτε για την περιγραφή, είτε για την επίλυση του προβλήματος μας.

### 2.1. Χωρικά Δίκτυα - Γεωμετρικοί Γράφοι

#### 2.1.1. Γραφοθεωρητικοί Ορισμοί

Οι γράφοι μας επιτρέπουν να αναπαραστήσουμε κάποιες οντότητες, ως κορυφές (vertices) και κάποιες συσχετίσεις αυτών ως ακμές (edges). Συμβολίζουμε με  $\mathcal{V}$  ένα σύνολο με στοιχεία τις κορυφές, και  $\mathcal{E}$  το σύνολο των ακμών, όπου κάθε ακμή είναι ένα σύνολο δύο στοιχείων  $\{v_1, v_2\}$ ,  $v_1, v_2 \in \mathcal{V}$  (περίπτωση μη-κατευθυνόμενων γράφων). Κάθε ακμή που ανήκει στο σύνολο  $\mathcal{E}$  κωδικοποιεί την ύπαρξη της συσχέτισης που μας ενδιαφέρει. Δεδομένης της συνολοθεωρητικής αυτής περιγραφής των γράφων, όλοι οι αντίστοιχοι συμβολισμοί των συνόλων έχουν εφαρμογή, όπως για παράδειγμα το πλήθος των κορυφών που συμβολίζεται ως  $|\mathcal{V}|$ . Συνολικά ο γράφος θα συμβολίζεται ως ένα ζεύγος των δύο αυτών χαρακτηριστικών  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . Για την διευκόλυνση της περιγραφής, το σύνολο  $\mathcal{V}$  θα αποτελείται εν γένει από πιο σύνθετα αντικείμενα, όπως σύνολα, αντί να ορίζεται ένα ενδιάμεσο σύμβολο που λειτουργεί ως ταυτότητα η οποία δεν περιέχει επιπλέον δομή. Έτσι για παράδειγμα μπορούμε να έχουμε ένα σύνολο τριών κορυφών  $\{[0, 1], [2, 3], [4, 5]\}$  αντί να ορίζουμε ταυτότητες που έχουν το ρόλο των κορυφών  $v_1, v_2, v_3$  και αντιστοιχούν στα  $[0, 1], [2, 3], [4, 5]$ .

#### 2.1.2. Γεωμετρικοί Γράφοι

Ειδικότερα στην εργασία μας θα ασχοληθούμε με Χωρικά Δίκτυα (Spatial Networks) [14] τα οποία αναπαριστώνται με γεωμετρικούς γράφους - δηλαδή γράφους στους οποίους αποδίδεται γεωμετρική σημασία σε κορυφές και ακμές. Οι Γεωμετρικοί Γράφοι διακρίνονται σε ντετερμινιστικούς ή διαφορετικά κανονικούς γράφους, δηλαδή εκείνοι όπου υπάρχει ένας σαφής κανόνας για το πως κατανέμονται ακμές και κορυφές και τυχαίους γράφους όπου τόσο οι θέσεις των κορυφών όσο και οι ακμές περιλαμβάνουν τυχαιότητα. Ένας ενδιαφέρων γεωμετρικός γράφος είναι ο γράφος μοναδιαίων δίσκων (Unit disk Graph) (ή ο αντίστοιχος Random Geometric Graph). Αυτός ο γράφος προκύπτει από ένα σύνολο σημείων στο επίπεδο, όπου δύο σημεία συνδέονται μεταξύ τους εάν η απόστασή τους είναι κάτω από κάποιο συγκεκριμένο όριο. Τα χωρικά δίκτυα απαντώνται σε πολλές εφαρμογές, όπως ad-hoc δίκτυα κινητών συσκευών, δίκτυα ανθρώπων σε μια κοινωνική συγκέντρωση και άλλα. Σημειώνεται πως ο γεωμετρικός χώρος δεν

χρειάζεται πάντοτε να είναι το Ευκλείδειο επίπεδο, αλλά θα μπορούσε να είναι κάποια πιο σύνθετη επιφάνεια, όπως για παράδειγμα η επιφάνεια της Γης - αρκεί να μπορεί να οριστεί καλώς η έννοια της απόστασης. Στην παρούσα εργασία, οι γράφοι που μας ενδιαφέρουν έχουν κορυφές οι οποίες βρίσκονται στο  $\mathbb{R}^2$ , ενώ οι ακμές ενώνουν τα δύο σημεία αυτά, οπότε εύκολα μπορεί κανείς να ορίσει ένα μέτρο απόστασης ή βάρους - αυτό του μήκους του ευθύγραμμου τμήματος που τα ενώνει.

### 2.1.3. Γράφοι Τομής

Οι γράφοι τομής (intersection graphs) είναι γράφοι που προκύπτουν από μια οικογένεια συνόλων  $\mathcal{V} = \{S_1, S_2, \dots, S_n\}$ . Εάν η οικογένεια αυτή εκφράζει τις κορυφές, τότε το σύνολο των ακμών ορίζεται ως

$$\mathcal{E} = \{\{S_i, S_j\} \mid (i \neq j) \wedge (S_i \cap S_j \neq \emptyset)\}$$

Με απλά λόγια, δύο σύνολα συνδέονται με ακμή εάν η τομή τους είναι μη κενή, ενώ δεν επιτρέπουμε τις ανακυκλώσεις. Ο ορισμός αυτός είναι κάπως γενικός, και πράγματι μέσω ενός γράφου τομής μπορεί κανείς να ορίσει οποιοδήποτε γράφο, όμως έχει αξία να αναφερόμαστε σε αυτούς σε πιο ειδικές περιπτώσεις στις οποίες η συνολοθεωρητική περιγραφή είναι η απλούστερη. Ο Unit Disk Graph που αναφέρθηκε νωρίτερα έχει μια πολύ φυσική περιγραφή μέσα από γράφους τομής βασιζόμενο σε δίσκους - η ίδια ιδέα θα χρησιμοποιηθεί αργότερα για υποσύνολα δίσκων που εκφράζουν την ορατότητα των αισθητήρων.

## 2.2. Σημειακές Διαδικασίες

Οι σημειακές διαδικασίες (Point Processes) περιγράφουν διαδικασίες ή διαφορετικά γεννιέστερες τυχαίων μοτίβων σημειακών αντικειμένων σε ένα χώρο [1]. Στην παρούσα εργασία θα μας απασχολήσουν σημειακές διαδικασίες στον  $\mathbb{R}^2$ . Θεωρούμε ότι δεν γίνεται πολλαπλά αντικείμενα να καταλήγουν στο ίδιο σημείο και αναφερόμαστε σε απλές σημειακές διαδικασίες (Simple Point Processes). Δύο βασικές σημειακές διαδικασίες, οι οποίες και σχετίζονται μεταξύ τους, είναι οι ομοιογενείς περιπτώσεις των Poisson Point Process (PPP) και Binomial Point Process (BPP). Η ομοιογενής PPP στις δύο διαστάσεις έχει μια πυκνότητα σημείων ανά μονάδα χώρου  $m^2$ , έστω  $\lambda \left[\frac{\text{points}}{m^2}\right]$ , που εκφράζει τον μέσο αριθμό σημείων που εμφανίζονται σε κάθε μονάδα χώρου. Έτσι σε ένα χωρίο εμβαδού  $A \ [m^2]$ , ο αναμενόμενος αριθμός σημείων θα είναι  $A \cdot \lambda \ [points]$ . Στην BPP είναι δεδομένο το πλήθος των σημείων,  $n$ , και αυτά κατανέμονται με ομοιόμορφο τρόπο στο χώρο ενδιαφέροντος, σε αντιπαράθεση με την περίπτωση του PPP που το πλήθος των σημείων είναι τυχαία μεταβλητή. Δεσμευμένου του πλήθους των σημείων

στην περίπτωση του PPP, η διαδικασία ταυτίζεται με την BPP. Ο χώρος στον οποίο αναπτύσσονται οι διαδικασίες αυτές μπορεί να είναι αρκετά σύνθετος. Για την παρούσα εργασία θα είναι εν γένει μη-κυρτός.

Πέρα από τις δύο κεντρικές διαδικασίες που περιγράφηκαν, μπορεί κανείς να καταλήξει σε νέες διαδικασίες με πράξεις όπως μετασχηματισμούς (transformation), δηλαδή μέσα από απεικονίσεις με τη χρήση συναρτήσεων, εκλεπτύνσεις (thinning), όπου με βάση κάποιο κριτήριο επιλέγουμε ένα υποσύνολο των σημείων της διαδικασίας (διαφορετικά, φιλτράρουμε κάποια σημεία), υπερθέσεις (superposition) όπου με τη χρήση πολλών διαδικασιών παίρνουμε μια συνολική διαδικασία η οποία περιλαμβάνει τα σημεία κάθε διαδικασίας, ή ομαδοποίηση (clustering) όπου μέσα από μια αρχική διαδικασία επιλέγουμε κάποια κέντρα γύρω από τα οποία θα δημιουργηθούν ομάδες σημείων μέσα από έναν δεύτερο κανόνα γέννησης.

Τέλος υπάρχουν και οι marked point processes, όπου αποτελούν σημειακές διαδικασίες οι οποίες έχουν επιπλέον ένα μαρκάρισμα ή ταυτότητα για κάθε σημείο. Αυτό το μαρκάρισμα συνήθως εκφράζει κάποιο χαρακτηριστικό του σημείου το οποίο εξαρτάται από την εφαρμογή. Το μαρκάρισμα ανήκει σε χώρους οι οποίοι έχουν καλές ιδιότητες που εξασφαλίζουν ότι τα μαρκάρια μπορούν να αγνοηθούν χωρίς απειρισμούς πυκνότητας στον αμαρκάριστο χώρο. Για παράδειγμα, ένα μαρκάρισμα μπορεί να ανήκει σε χώρους όπως το  $[a, b]$ , αλλά όχι σε χώρους με θετική πυκνότητα από το διάστημα  $(-\infty, \infty)$  διότι η αφαίρεση του μαρκάριατος θα συγκέντρωνε την ισοδύναμη άπειρη πυκνότητα στον αμαρκάριστο χώρο, κάτι που δεν είναι επιθυμητό.

Συνολικά, με συνδυασμό των παραπάνω διαδικασιών μας δίνεται η δυνατότητα να μοντελοποιήσουμε ένα μεγάλο εύρος περιπτώσεων.

**Προσομοίωση.** Οι προσομοιώσεις των διαδικασιών BPP και PPP μπορεί να γίνει με παρόμοιο τρόπο. Συγκεκριμένα για την PPP, προσομοιώνοντας μια τυχαία μεταβλητή από την κατανομή Poisson, με κατάλληλο  $\lambda = \text{area}(B)\lambda_1$ , όπου  $\lambda_1$  η ένταση της PPP, και  $B$  το μέτρο του χώρου όπου εμφανίζονται τα σημεία της διαδικασίας, έχουμε ως αποτέλεσμα έναν αριθμό σημείων  $k$ . Για την προσομοίωση μιας τιμής από την Poisson κατανομή, μπορεί να χρησιμοποιηθεί η μέθοδος Inverse transform sampling. Τότε δεδομένου αυτού του πλήθους σημείων η PPP και η BPP ταυτίζονται. Για την προσομοίωση μιας BPP σε ένα επίπεδο, παράγουμε ένα τυχαίο σημείο εντός του μικρότερου ορθογώνιου παραλληλόγραμμου που περιέχει το χώρο που μας ενδιαφέρει. Έπειτα εξετάζουμε εάν το σημείο αυτό βρίσκεται στην περιοχή ενδιαφέροντος. Εάν ναι διατηρούμε το σημείο ως έγκυρο, διαφορετικά το απορρίπτουμε και επαναλαμβάνουμε τη διαδικασία (rejection method).

## 2.3. Μαθηματική Βελτιστοποίηση

### 2.3.1. Εισαγωγικές Έννοιες

Η μαθηματική βελτιστοποίηση (*mathematical optimization*) είναι ένας κλάδος των μαθηματικών και της επιχειρησιακής έρευνας που ασχολείται με την επιλογή ή εύρεση των καταλληλότερων παραμέτρων-μεταβλητών, ώστε να μεγιστοποιηθεί ή να ελαχιστοποιηθεί μια ζητούμενη ποσότητα-κριτήριο. Έτσι μπορεί να μιλάμε για  $n$  παραμέτρους οι οποίες παίρνουν τιμές από ένα υποσύνολο του  $\mathbb{R}^n$ , το οποίο αποκαλούμε *εφικτό σύνολο* (*Feasible Set*). Ο υπόλοιπος χώρος καλείται *μη-εφικτό σύνολο* (*Infeasible Set*) και εκφράζει τους συνδυασμούς των παραμέτρων που δεν έχουν ενδιαφέρον ή νόημα να μελετηθούν. Το κριτήριο αποκαλείται και βαθμωτή *αντικειμενική συνάρτηση* (*objective function*), έστω  $f$ , και το πρόβλημα μπορεί σε περιπτώσεις να είναι η εύρεση της καλύτερης τιμής που λαμβάνει η  $f$ , δηλαδή το  $\max f(\mathbf{x})$  ή των παραμέτρων που οδηγούν στην καλύτερη τιμή  $\arg \max f(\mathbf{x})$ .

Ένας ορισμός ενός προβλήματος μαθηματικού προγραμματισμού ορίζεται ως εξής:

$$\begin{array}{ll} \text{minimize/maximize} & f(\mathbf{x}) \\ \text{subject to} & C_i(\mathbf{x}) = 0, i \in E \\ & C_i(\mathbf{x}) \leq 0, i \in I. \end{array}$$

Όπου τα σύνολα δεικτών  $E, I$  αντιστοιχούν σε εκείνους τους δείκτες των περιορισμών ισότητας και ανισότητας, με τα  $\mathbf{x}$  που τους ικανοποιούν να ανήκουν στο εφικτό σύνολο.

### 2.3.2. Βελτιστοποίηση Πολλαπλών Κριτηρίων

Στη μαθηματική βελτιστοποίηση πέρα από τη βελτιστοποίηση μιας μονοδιάστατης ποσότητας, όπου η ολική διάταξη διευκολύνει την απόφαση, υπάρχουν σε πολλά προβλήματα περισσότερες ποσότητες που επιθυμούμε να μεγιστοποιήσουμε (ή να ελαχιστοποιήσουμε). Τότε το πρόβλημα της απουσίας εγγενούς διάταξης δεν μας επιτρέπει να αποφασίσουμε με προφανή τρόπο μεταξύ δύο λύσεων. Σε αυτές τις περιπτώσεις μπορεί να οριστεί μια μερική διάταξη. Επιλέγοντας την *κατα συνιστώσες* (*component-wise*) μερική διάταξη<sup>1</sup>, αυτό έχει ως συνέπεια την δημιουργία του *μετώπου Pareto* [19].

Ας υποθέσουμε ότι έχουμε δύο κριτήρια που θέλουμε να μεγιστοποιήσουμε. Εάν έχουμε μια λύση με τα κριτήρια να λαμβάνουν τις ποσότητες  $(a_1, b_1)$ , και μπορούμε να βελτιώσουμε την μια ποσότητα, έστω την  $a_1$  σε μια καλύτερη, την  $a_2 (> a_1)$  χωρίς να

<sup>1</sup>Η κατά συνιστώσες μερική διάταξη (*component-wise ordering*), ορίζεται ως η διάταξη όπου το  $[y_1, y_2, \dots, y_n]^T \preceq [x_1, x_2, \dots, x_n]^T$ , συνεπάγεται το  $\{y_1 \leq x_1, y_2 \leq x_2, \dots, y_n \leq x_n\}$

μεταβληθεί το δεύτερο κριτήριο, τότε μπορούμε χωρίς αμφιβολία να προτιμήσουμε την  $(a_2, b_1)$ , τότε λέμε ότι το τελευταίο ζεύγος επικρατεί έναντι του πρώτου (dominate). Έστω ότι με αυτή τη διαδικασία καταλήγουμε σε ένα ζεύγος  $(a, b)$  στο οποίο δεν μπορεί να βελτιωθεί καμία ποσότητα χωρίς να χειροτερέψει η άλλη, δηλαδή δεν υπάρχει κατά συνιστώσες καλύτερη επιλογή - τότε λέμε ότι έχουμε μια *Pareto βέλτιστη λύση*.

Ένας ορισμός του μετώπου Pareto εάν θεωρήσουμε  $m$  κριτήρια, με εφικτές λύσεις από ένα συμπαγές σύνολο  $X \subseteq \mathbb{R}^n$  και τιμές της αντικειμενικής συνάρτησης στο εφικτό σύνολο  $Y \subseteq \mathbb{R}^m$ , τότε έχουμε:

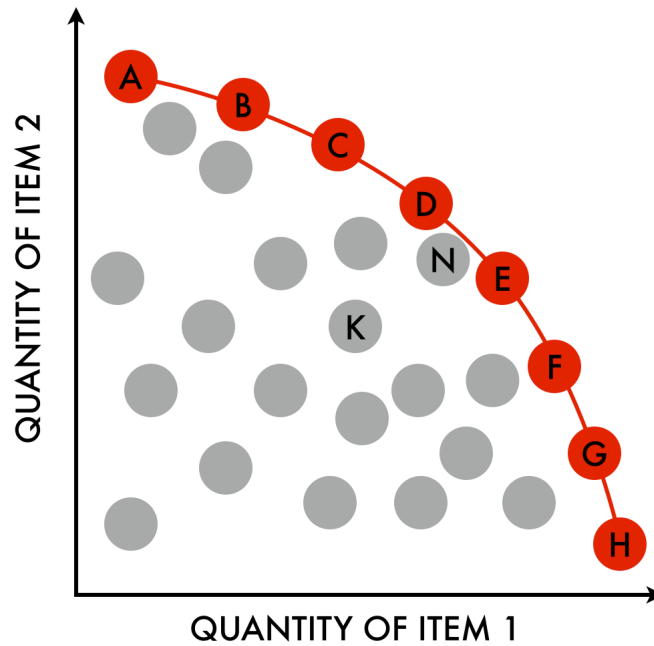
$$P(Y) = \{y' \in Y : \{y'' \in Y : y'' \succ y', y'' \neq y'\} = \emptyset\}$$

Δηλαδή το σύνολο των τιμών των λύσεων, που δεν γίνονται strictly-dominated από κάποια άλλη λύση.

Τόσο το μέτωπο Pareto όσο και όλες οι δυνατές λύσεις του προβλήματος απεικονίζονται στο σχήμα 1. Οι λύσεις πάνω στο κόκκινο σύνορο (οι οποίες εκτείνονται με συνεχή τρόπο) δεν γίνονται strictly-dominated από κάποια άλλη λύση, και συνεπώς είναι Pareto βέλτιστες. Αντίθετα μπορεί κανείς να παρατηρήσει τις λύσεις  $K$  και  $N$ : αυτές είναι χειρότερες σε κάθε συνιστώσα τους από κάποια λύση πάνω στο κόκκινο σύνορο και συνεπώς δεν είναι Pareto βέλτιστες.

**Εκ των υστέρων μέθοδοι.** Ένας τρόπος για να απαντήσουμε στο δίλημμα της επιλογής μεταξύ των λύσεων του μετώπου Pareto είναι με μια *εκ των υστέρων (a posteriori)* επιλογή, δηλαδή μια επιλογή μετά τον υπολογισμό του μετώπου Pareto. Σε πολλές περιπτώσεις, στις μεθόδους αυτές, επειδή είναι δύσκολος ο υπολογισμός ολόκληρου του μετώπου, γίνεται υπολογισμός ενός υποσύνολου του μετώπου ή μιας προσέγγισης του. Η τελική επιλογή γίνεται είτε από έναν υπεύθυνο για την απόφαση (decision maker) που χρησιμοποιεί υποκειμενικά κριτήρια, είτε με τη χρήση κάποιου συμπληρωματικού κριτηρίου που ορίζει ολική διάταξη πάνω στις λύσεις του μετώπου [7, 9].

**Εκ των προτέρων μέθοδοι.** Μια διαφορετική προσέγγιση, την οποία και θα υιοθετήσουμε στα πλαίσια αυτής της εργασίας, είναι η μετάβαση σε μια ολική διάταξη μέσα από μια απεικόνιση  $g$  στο  $\mathbb{R}$  πρώτου υπολογίσουμε το μέτωπο Pareto. Αυτό αναφέρεται στην βιβλιογραφία και ως *βαθμωτοποιημένο πρόβλημα (scalarized problem)*, όπου θα θέλαμε μια “καλή” απεικόνιση  $g$  να έχει βέλτιστες τιμές που αντιστοιχούν σε Pareto optimal λύσεις. Θεωρήστε ένα πρόβλημα όπου θέλουμε να μεγιστοποιήσουμε τα κριτήρια



Σχήμα 1: Στο διάγραμμα αυτό παρουσιάζεται η βελτιστοποίηση δύο κριτηρίων όπου και οπτικοποιείται το μέτωπο Pareto. Μεταξύ των κόκκινων χρωματισμένων λύσεων δεν υπάρχει προφανής επιλογή αφού το ένα από τα δύο χαρακτηριστικά είναι προτιμότερο. Σημειώνεται ότι όλες οι τιμές στην κόκκινη καμπύλη είναι επιτρεπτές. Πηγή: Wikipedia

$f_1(\cdot)$  και  $f_2(\cdot)$ . Αντί να βελτιστοποιήσουμε το κριτήριο  $f_1$  μαζί με το κριτήριο  $f_2$  θα μπορούσαμε να βελτιστοποιήσουμε το κριτήριο  $g(f_1(x), f_2(x)) = f_1(x) + f_2(x)$ . Η δυσκολία σε αυτές τις μεθόδους είναι να εκφραστεί με έναν μαθηματικό τρόπο η προτίμηση, πριν ακόμα σχηματιστεί εικόνα για το ποιο θα είναι το μέτωπο Pareto. Πολλές φορές οι συναρτήσεις που χρησιμοποιούνται για την μετάβαση σε μεθόδους εκ των υστέρων και εκ των προτέρων ταυτίζονται [7, 9].

### 2.3.3. Κυρτή και Μη-Κυρτή Βελτιστοποίηση

Μέχρι τώρα μιλήσαμε για την ταξινόμηση των προβλημάτων και την μοντελοποίηση, και όχι για το πως προσεγγίζουμε την επίλυση σε τέτοια προβλήματα. Όταν μιλάμε για προβλήματα βελτιστοποίησης, όπως έχει τονιστεί και στο [2], η βασική ταξινόμηση δεν είναι σε Γραμμικά και Μη Γραμμικά προβλήματα, αλλά περισσότερο σε *κυρτά* και *μη κυρτά* προβλήματα. Τα κυρτά προβλήματα είναι προβλήματα όπου το κριτήριο είναι κυρτή συνάρτηση και ο εφικτός χώρος είναι κυρτός επίσης. Μια κυρτή συνάρτηση  $f$  σε ένα σύνολο  $D$  είναι μια συνάρτηση για την οποία η σχέση  $\forall \mathbf{x}_1, \forall \mathbf{x}_2 \in D : f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq$

$tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2)$ , με  $t \in [0, 1]$ . Επιπλέον, ένα κυρτό σύνολο  $S$  είναι ένα σύνολο για το οποίο ισχύει  $\mathbf{x}_1, \mathbf{x}_2 \in S \Rightarrow t\mathbf{x}_1 + (1-t)\mathbf{x}_2 \in S$ , με  $t \in [0, 1]$ . Από την άλλη πλευρά, τα μη κυρτά προβλήματα είναι αυτά που δεν ισχύει κάτι από τα παραπάνω. Όταν αντιμετωπίζουμε ένα κυρτό πρόβλημα μπορούμε ακολουθώντας την τοπικά βέλτιστη λύση να καταλήξουμε στην ολικά βέλτιστη λύση, και ειδικότερα όταν έχουμε μια αναλυτική μορφή, μπορούμε απλά να μεταβαίνουμε προς την κατεύθυνση της κλίσης (gradient) για προβλήματα μεγιστοποίησης, με αποτέλεσμα να καταλήγουμε σε ολικό ακρότατο. Σε μη κυρτά προβλήματα αυτό δεν ισχύει, και η δυσκολία εύρεσης καλών ακρότατων ή καλών λύσεων μπορεί να γίνει πολύ πιο σύνθετη, ακόμα και εκθετικής δυσκολίας σε πολλές περιπτώσεις. Τα μη-κυρτά προβλήματα, εκ των οποίων ένα είναι και το πρόβλημα μας με τον φορμαλισμό που θα παραθέσουμε, πολλές φορές απαιτούν μια ευρετική λύση ή έναν προσεγγιστικό αλγόριθμο. Ένας προσεγγιστικός αλγόριθμος θα μας δώσει κάποια αποδεδειγμένα καλή λύση ακόμα και στη χειρότερη περίπτωση, σε σύγκριση με τη βέλτιστη λύση που έχει το πρόβλημα. Τότε θα μιλάμε για  $\alpha$ -approximation λύσεις. Σε πολλές άλλες περιπτώσεις μπορούμε απλά να δοκιμάσουμε ότι ο αλγόριθμος μας δουλεύει ικανοποιητικά σε πολλαπλά σενάρια.

## 2.4. Ασύρματα Δίκτυα Αισθητήρων

Με τον όρο Ασύρματο Δίκτυο Αισθητήρων (Wireless Sensor Network) εννοούμε ένα δίκτυο αισθητήρων στο οποίο οι κόμβοι είναι κατανεμημένοι σε έναν χώρο, και συλλέγουν κάποια τοπική πληροφορία, όπως για παράδειγμα θερμοκρασία, εικόνα ή φωτεινότητα. Παράλληλα στο δίκτυο αυτό, θέλουμε με κάποιο τρόπο αυτή η πληροφορία εάν είναι αναγκαίο, ή εν δυνάμει, να μεταβιβαστεί σε κάποιον κεντρικό κόμβο (Base Station) και στη συνέχεια να αναλυθεί και να αξιοποιηθεί ανάλογα με την περίπτωση.

### 2.4.1. Μεταβίβαση Πληροφορίας

Η μεταβίβαση της πληροφορίας προς τον κεντρικό κόμβο συχνά δεν είναι εφικτό ή δεν είναι προτιμητέο ενεργειακά να γίνει άμεσα, αλλά χρειάζεται να εμπλακούν γειτονικοί κόμβοι, γνωστοί ως μεταγωγείς (relays), οι οποίοι θα μεταβιβάσουν την πληροφορία σε έναν πλησιέστερο προς το στόχο κόμβο, και αναδρομικά η πληροφορία θα φτάσει στον τελικό της προορισμό. Μια ανάλυση του γιατί μπορεί να συμφέρει η μεταφορά της πληροφορίας σε ενδιάμεσους κόμβους, ακόμα και σε περιπτώσεις που ένας κόμβος μπορεί να εκπέμψει με τρόπο τέτοιο ώστε να επικοινωνήσει άμεσα με τον κεντρικό κόμβο, μπορεί να βρεί κανείς στο [21]: ο λόγος οφείλεται στο ότι για να επιτευχθεί η επικοινωνία σε απόσταση  $r$  απαιτείται ενέργεια ανάλογη του  $r^a$ ,  $a \geq 2$  και συνεπώς οι μεγάλες ακτίνες είναι πολύ χειρότερες από πολλές μικρές ακτίνες ακόμα και εάν κάποιος λάβει υπόψη το

κόστος για τη μεταβίβαση (relaying) και την δρομολόγηση (routing).

#### 2.4.2. Είδη Αίσθησης

**Σημειακή Αίσθηση.** Το είδος της τοπικής πληροφορίας που συλλέγεται μπορεί να είναι σημειακή, δηλαδή σε μια θέση  $r$  η πληροφορία που συλλέγεται αφορά το ίδιο σημείο  $r$  - στις περισσότερες εφαρμογές όμως κανείς θα ήθελε με ένα σύνολο αισθητήρων να εξάγει συμπεράσματα όχι για τα πεπερασμένα αυτά σημεία αλλά για το γενικότερο χώρο ενδιαφέροντος (στον  $\mathbb{R}^3$  ή  $\mathbb{R}^2$ ). Αυτό πετυχαίνεται με την υπόθεση ύπαρξης κάποιας χωρικής ή χρονικής συσχέτισης (spatial - time correlation) με τα γειτονικά σημεία και πολλές φορές με κάποιου είδους παρεμβολής - για παράδειγμα Bilinear ή Bicubic παρεμβολή. Σε πολλές περιπτώσεις αυτή η συσχέτιση προκύπτει από την ίδια την φυσική της μέτρησης - ως παράδειγμα μπορούμε να σκεφτούμε έναν αισθητήρα ήχου, όπου η μέτρηση που κατέγραψε είναι η ίδια που θα κατέγραφε ένας αισθητήρας τοποθετημένος σε πολύ κοντινή απόσταση με την καθυστέρηση του να ταξιδέψει το ηχητικό κύμα από το ένα σημείο στο άλλο.

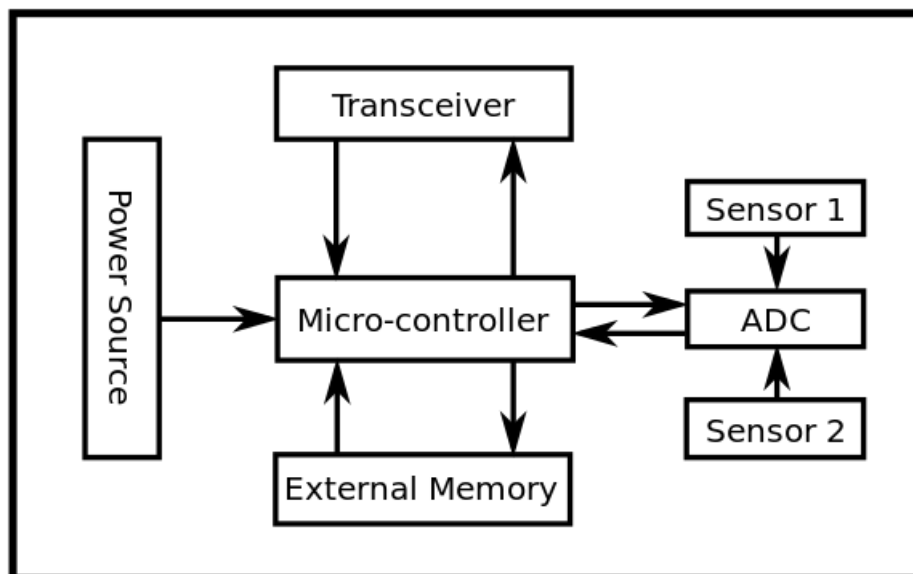
**Τοπική Αίσθηση.** Μπορούμε να θεωρήσουμε μια διαφορετική κλάση αισθητήρων από αυτή των σημειακών, η οποία λαμβάνει πληροφορία σε μια ολόκληρη περιοχή τοπικά στο χώρο περί του σημείου  $r$ . Αυτή η πληροφορία δεν είναι αναγκαίο να προκύπτει άμεσα, αλλά μπορεί να *συμπεραίνεται* (*inferred*) από τη συγκεκριμένη μέτρηση. Για παράδειγμα με κατάλληλη επεξεργασία μπορούμε να θεωρήσουμε τον αισθητήρα ήχου σαν έναν αισθητήρα άμεσης χωρικής μέτρησης, η οποία ενδεχομένως περιλαμβάνει μια συνιστώσα θορύβου μεταβλητής τυπικής απόκλισης σαν συνάρτηση της απόστασης. Ένα άλλο παράδειγμα θα ήταν ένας αισθητήρας εικόνας (image sensor) ο οποίος λαμβάνει πληροφορία σχετική για μια ολόκληρη περιοχή στην οποία διαθέτει οπτική ορατότητα. Στο κείμενο αυτό οι μέθοδοι που θα αναπτυχθούν θα μπορούσαν να μοντελοποιήσουν τέτοιου είδους μετρήσεις αρκεί να χαρακτηρίζονται από τους περιορισμούς που θα συζητήσουμε στη συνέχεια.

#### 2.4.3. Αρχιτεκτονική Αισθητήρων

Ένα WSN μπορεί να έχει από λίγους κόμβους έως και χιλιάδες, ανάλογα με την εφαρμογή. Βασικό στοιχείο για έναν κόμβο είναι ο transceiver που επιτρέπει την ασύρματη επικοινωνία του. Επιπλέον περιλαμβάνεται συνήθως ένας μικροελεγκτής (όπως AVR, ARM) που έχει τη δυνατότητα να διαχειριστεί όλη την πληροφορία και να εκτελέσει εν γένει αυθαίρετες πράξεις. Ο σχεδιαστής πρέπει ωστόσο να έχει κατανόηση των περιορισμών κατανάλωσης που αποτρέπουν εντατικές πράξεις συστηματικά. Τέλος, απαι-



τείται μια πηγή ενέργειας, όπως για παράδειγμα ηλιακή ενέργεια ή μια μπαταρία, η οποία είναι αναγκαία ώστε να λειτουργεί ο αισθητήρας. Στο σχήμα 2 φαίνονται τα στοιχεία ενός τέτοιου κόμβου.



Σχήμα 2: Block διάγραμμα ενός WSN κόμβου ο οποίος διαθέτει δύο αισθήσεις και εξωτερική μνήμη. Ενδιάμεσα βρίσκεται ο μικροελεγκτής που διαχειρίζεται όλα τα σήματα και τη διασύνδεση των διάφορων μερών. Πηγή: Wikipedia

#### 2.4.4. Πηγή, Κατανάλωση και Διαχείριση Ενέργειας

Κεντρικό θέμα στα WSN είναι η διαχείριση της ενέργειας. Τόσο η προέλευση όσο και η ανάθεση των ενεργειακών πόρων στις διάφορες λειτουργίες του αισθητήρα, είναι μείζονος σημασίας στην αποτελεσματικότητα και λειτουργικότητα του δικτύου. Η πιο συνηθισμένη πηγή ενέργειας είναι ο ηλεκτρικός συσσωρευτής (μπαταρία), αλλά είναι εφικτό ένας αισθητήρας να χρησιμοποιεί και ως πηγές ενέργειας την ηλιακή, την ενέργεια των δονήσεων ή άλλες εναλλακτικές πηγές που εξαρτώνται από την εφαρμογή, αλλά και το περιβάλλον στο οποίο είναι τοποθετημένοι οι αισθητήρες (σεισμικό, υποθαλάσσιο, κλπ).

Η ενέργεια καταναλώνεται γενικά σε τρεις λειτουργίες: στη συλλογή, τη διαχείριση και τη μετάδοση της πληροφορίας της μέτρησης. Το πιο σημαντικό κόστος, το οποίο και κινείται σε ανώτερες τάξεις μεγέθους ανα *bit* μεταξύ αυτών των λειτουργιών, είναι η μετάδοση της πληροφορίας. Ως αποτέλεσμα ένας σχεδιαστής πρέπει να λάβει σοβαρά υπόψη το πως θα ελαχιστοποιήσει την μεταβίβαση πληροφορίας. Συγκεκριμένα τόσο

για την μετάδοση πληροφορίας όσο και για τη λήψη της, καταναλώνεται ενέργεια, από εκατοντάδες έως χιλιάδες  $nJ$  για κάθε *bit* πληροφορίας.

Η λειτουργία της διαχείρισης της πληροφορίας στον κόμβο, συχνά εμπεριέχει την επεξεργασία της με στόχο να μειωθεί η ανάγκη επικοινωνίας με κεντρικό κόμβο ή ο όγκος της πληροφορίας που μεταβιβάζεται να περιοριστεί - για παράδειγμα ένας κόμβος θα μπορούσε να συμπιέζει τα δεδομένα πριν τα μεταβιβάσει.

Ο μικροελεγκτής του κόμβου καταναλώνει για κάθε εντολή από το σύνολο εντολών της (instruction set), ενέργεια στη τάξη των  $nJ$ , γεγονός που δίνει κίνητρο για την ανάπτυξη λογισμικού με γνώμονα την ελαχιστοποίηση των εντολών ανά μέτρηση. Επιπλέον, για την περαιτέρω μείωση της κατανάλωσης ο αισθητήρας μπορεί να εισέρχεται σε διαφορετικές καταστάσεις, οι οποίες διαφέρουν από επεξεργαστή σε επεξεργαστή ως προς το πλήθος τους και την κατανάλωση τους. Δύο συνήθεις λειτουργίες του επεξεργαστή για τον περιορισμό της κατανάλωσης ενέργειας είναι οι καταστάσεις αδράνειας (idle), και ύπνου (sleeping).

#### 2.4.5. Οπτικά Δίκτυα Αισθητήρων

Ως μια ειδική περίπτωση των WSN μπορούν να θεωρηθούν τα Οπτικά Δίκτυα Αισθητήρων (Visual Sensor Networks) [22]. Σε αυτά τα δίκτυα, τα οποία είναι αντικείμενο έρευνας για τη βελτίωση τους [8], διαθέτουμε ένα δίκτυο μικροσκοπικών κόμβων οι οποίοι μπορούν να παρατηρήσουν το περιβάλλον σύμφωνα με μια ορατότητα γραμμής (line of sight), δηλαδή ακτινική ορατότητα σε όλα τα σημεία με τα οποία ενδιάμεσα του παρατηρούμενου σημείου και του σημείου του αισθητήρα δεν περιέχεται εμπόδιο. Αυτό μπορούμε να το πετύχουμε για παράδειγμα με χρήση αισθητήρων εικόνας (image sensors). Οι εξελίξεις στην τεχνολογία των αισθητήρων εικόνας δίνουν το έδαφος για περισσότερες εφαρμογές. Σε αυτήν την περίπτωση δεν έχουμε πια σημειακή μέτρηση, αλλά ακτινική γύρω από ένα σημείο. Σε αυτή τη μέτρηση θα μπορούσε κανείς να ορίσει ένα μέτρο ποιότητας για κάθε σημείο της, το οποίο εξαρτάται από την απόσταση από το κέντρο του αισθητήρα.

Το ενδιαφέρον για αυτά τα δίκτυα προκύπτει από μικρές κάμερες οι οποίες είναι αρκετά οικονομικές αλλά με περιορισμένη ποιότητα εικόνας - αποτέλεσμα αυτού είναι η ανάγκη για εκτεταμένη επεξεργασία, συγχώνευση πληροφορίας, ή πιθανοτική αντιμετώπιση με την απόσταση. Συνολικά όσο περισσότερη πληροφορία προσπαθούμε να συλλέξουμε από έναν τέτοιο αισθητήρα, αναμένουμε μεγαλύτερες απαιτήσεις ενέργειας.

## 3. ΣΧΕΤΙΚΟ ΕΡΓΟ

### 3.1. Έλεγχος Τοπολογίας

Με τον όρο *Έλεγχος Τοπολογίας (Topology Control)* για WSN [21, 26] εννοούμε τεχνικές που χρησιμοποιούνται ώστε να μεταβαίνουμε από το μέγιστο δυνατό γράφο επικοινωνίας (communication graph) ενός δικτύου (για μέγιστη ισχύ μετάδοσης και συνεπώς μέγιστη ακτίνα επικοινωνίας από κάθε κόμβο) σε έναν αρκετά μικρότερο αλλά παρόμοια συνδεδεμένο γράφο. Ο αρχικός γράφος έχει κόμβους με μεγάλο βαθμό όταν η μέγιστη ακτίνα επικοινωνίας είναι αρκετά μεγάλη σε σχέση με την πυκνότητα των τοποθετημένων κόμβων στο χώρο ενδιαφέροντος. Σε αυτές τις περιπτώσεις, είναι χρήσιμο να κάνουμε έλεγχο τοπολογίας, καταλήγοντας σε ένα νέο γράφο με πολύ λιγότερες ακμές ή και κορυφές (εάν για παράδειγμα απενεργοποιηθεί ένας κόμβος). Η επίτευξη του Έλεγχου Τοπολογίας γίνεται κατά κύριο λόγο με μεθόδους που θέτουν την ακτίνα επικοινωνίας  $R_c$  σε τιμές αρκετά μικρότερες από τη μέγιστη δυνατή ακτίνα επικοινωνίας  $R_{c,max}$  ενός κόμβου. Αυτό αρχικά έχει πλεονεκτήματα ενεργειακά για τους ίδιους λόγους που προτιμάμε ένα multihop δίκτυο σε μεγάλες κλίμακες έναντι ενός one-hop δικτύου. Παράλληλα οι συγχρούσεις στο κανάλι επικοινωνίας που προκύπτουν από έναν πολύ συνδεδεμένο γράφο επικοινωνίας έχουν ως αποτέλεσμα σημαντικό ποσοστό των πακέτων να χάνονται και να πρέπει να μεταδοθούν και πάλι, γεγονός ανεπιθύμητο ιδιαίτερα λαμβάνοντας υπόψη ότι η επικοινωνία είναι αρκετά ακριβή λειτουργία.

### 3.2. Προβλήματα Κάλυψης

Γενικότερα στη βιβλιογραφία μελετούνται τρία είδη κάλυψης. Η σημειακή κάλυψη, όπου προσπαθούμε να καλύψουμε ένα σύνολο από σημεία-στόχους. Η κάλυψη εμβαδού όπου προσπαθούμε να καλύψουμε μια γενικότερη περιοχή ενδιαφέροντος και η κάλυψη φραγμού (barrier) ή κάλυψη μονοπατιού (path coverage), όπου θέλουμε να ελαχιστοποιήσουμε την πιθανότητα να μην ανιχνευτεί κάποιου είδους εισβολή στο παρατηρούμενο περιβάλλον [10]. Το πρόβλημα της κάλυψης εμβαδού σχετίζεται με το Art Gallery Problem, όπου θέλουμε να τοποθετήσουμε κάμερες σε έναν εκθεσιακό χώρο ώστε κάθε σημείο του να παρακολουθείται από μια κάμερα· η ειδοποιός διαφορά εδώ είναι πως οι κάμερες μπορούν να δουν οσοδήποτε μακριά αρκεί να μην παρεμβάλλονται εμπόδια (έχουν όραση σε όλο το πολύγωνο ορατότητας που θα δούμε αργότερα). Στο πρόβλημα αυτό τα ερωτήματα είναι δυο: που θα τοποθετηθούν οι κάμερες και πόσες κάμερες θα τοποθετηθούν [10].

### 3.3. Χρονοπρογραμματισμός Αισθητήρων

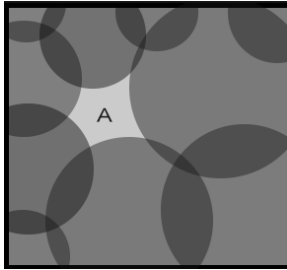
Μια πολύ συχνή προσέγγιση για τη μεγιστοποίηση της διάρκειας ζωής ενός δικτύου είναι η χρήση χρονοπρογραμματισμού, ώστε κάποιοι κόμβοι να τοποθετηθούν σε κατάσταση ύπνου (sleep mode), εφόσον δεν είναι σημαντικοί στη λειτουργία του συστήματος σε κάποια φάση του [5, 27]. Σε διαφορετικές περιόδους, διαφορετικά σύνολα αισθητήρων είναι ενεργά και εξασφαλίζουν την κάλυψη του χώρου, επεκτείνοντας με αυτόν τον τρόπο τη διάρκεια ζωής. Στο [27] το πρόβλημα προσεγγίζεται με τη θεώρηση μεταβλητών αλλά διακριτών ακτίνων αίσθησης, όπου ο στόχος είναι να ελαχιστοποιηθεί η επικάλυψη μεταξύ των περιοχών που καλύπτουν οι αισθητήρες. Αντίθετα στο [6] θεωρείται η περίπτωση της αλλαγής στην ακτίνα με συνεχή τρόπο, αλλά στην περίπτωση της κάλυψης στόχων. Συγκεκριμένα ο στόχος είναι να βρεθεί ένα μέγιστος αριθμός από καλύψεις του συνόλου των σημείων ενδιαφέροντος (set cover) μαζί με τις αντίστοιχες ακτίνες των αισθητήρων. Κάθε διαφορετικό σύνολο κόμβων αντιστοιχεί σε μια περίοδο, οπότε περισσότερες καλύψεις αντιστοιχούν σε περισσότερες περιόδους και άρα μεγαλύτερη διάρκεια ζωής του δικτύου.

### 3.4. Μοντέλα Κάλυψης

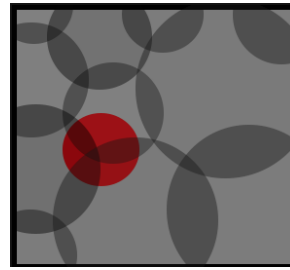
Τα βασικά μοντέλα κάλυψης που απαντώνται στη βιβλιογραφία είναι ο δυαδικός δίσκος (boolean disk) με ιστροπική κάλυψη, ο δυαδικός τομέας ενός δίσκου (sector) όπου υπάρχει κατευθυντική κάλυψη, καθώς και οι πιθανοτικές περιπτώσεις αυτών, δηλαδή περιπτώσεις όπου η κάλυψη γίνεται με κάποια πιθανότητα η οποία συνήθως φθίνει με την απομάκρυνση από τον αισθητήρα. Οι πιθανοτικές περιπτώσεις μοντελοποιούν με περισσότερη ακρίβεια αυτό που συμβαίνει τις περισσότερες φορές, όμως αυξάνουν την πολυπλοκότητα των μεθόδων που πρέπει να αναπτυχθούν. Στην περίπτωση που έχουμε πιθανοτική κάλυψη, εάν ένα σημείο καλύπτεται με πιθανότητα  $p_0$  από έναν αισθητήρα, και με πιθανότητα  $p_1$  από έναν άλλον, η πιθανότητα αυτό το σημείο να μην καλύπτεται είναι  $(1 - p_0)(1 - p_1)$  ενώ η πιθανότητα να καλύπτεται είναι  $1 - (1 - p_0)(1 - p_1)$ . Η ανάλογη σχέση ισχύει για περισσότερους αισθητήρες [24]. Εμείς θα ασχοληθούμε κυρίως με τη δυαδική (καλύπτει-δεν καλύπτει) περίπτωση στα πλαίσια της μελέτης μας.

### 3.5. Ανίχνευση Κάλυψης

Μια συνηθισμένη προσέγγιση για να ανιχνεύσει κανείς εάν υπάρχει πλήρης κάλυψη σε έναν χώρο είναι με τον έλεγχο της κάλυψης της περιφέρειας. Με τη μέθοδο αυτή μπορεί να εξετάσει κανείς την κάλυψη εξετάζοντας εάν οι γείτονες - συμπεριλαμβανομένων των εμποδίων καθώς και τα όρια της περιοχής ενδιαφέροντος - κάθε κόμβου καλύπτουν την



(α') Ύπαρξη κενού κάλυψης.



(β') Περιττός κόμβος.

Σχήμα 3: Παραδείγματα περιττού κόμβου και ύπαρξη κενού στην κάλυψη. Στην πρώτη εικόνα βλέπουμε ότι λόγω της μη κάλυψης της περιφέρειας κάποιων κόμβων, εμφανίζεται ένα κενό κάλυψης. Στη δεύτερη εικόνα βλέπουμε ότι για τον κόκκινο κόμβο, η ενεργοποίησή του είναι περιττή καθώς όλοι οι γείτονες έχουν καλυμμένη την περιφέρεια τους ήδη.

περιφέρεια του. Οι ιδέες αυτές γενικεύονται για  $k$ -coverage [12], όπου κάθε σημείο απαιτείται να καλύπτεται από  $k$  κόμβους αντί για έναν. Ένα επιπλέον σχετικό αποτέλεσμα, μας δίνει ένα τοπικό κριτήριο για την εύρεση των περιττών κόμβων σε μια ανάθεση: αρκεί κανείς να εξετάσει εάν όλοι οι γειτονικοί κόμβοι ενός συγκεκριμένου κόμβου έχουν την περιφέρεια τους καλυμμένη. Τότε δεν υπάρχει κάποιο κενό στο χώρο που να καλύπτεται από τον υπό εξέταση κόμβο [12]. Στο παράδειγμα στο σχήμα 3α' παρατηρούμε ότι οι κοντινοί κόμβοι στο σημείο  $A$  δεν έχουν την περιφέρεια τους καλυμμένη, το οποίο μεταφράζεται σε ύπαρξη κενού και συνεπώς όχι πλήρη κάλυψη. Αντίθετα στο σχήμα 3β' μπορεί κανείς αρχικά να παρατηρήσει ότι όλοι οι κόμβοι έχουν καλυμμένη την περιφέρεια τους, και έχουμε πλήρη κάλυψη. Επιπρόσθετα, ο κόκκινος κόμβος, εάν εξετάσει τους γειτονικούς του κόμβους, εκείνοι έχουν καλυμμένη την περιφέρεια τους χωρίς την παρουσία του κόκκινου κόμβου. Συνεπώς είναι περιττή η παρουσία του κόμβου στην ανάθεση. Τα δύο αυτά αποτελέσματα είναι πολύ χρήσιμα σε κατανημεμένες και τοπικές μεθόδους, και θα γίνει χρήση αυτών από τις μεθόδους μας.

### 3.6. Σχέση Κάλυψης Αίσθησης και Κάλυψης Επικοινωνίας

Εφόσον έχουμε πλήρη κάλυψη αίσθησης (Sensing Coverage) με ένα σύνολο κόμβων εντός πεπερασμένου κυρτού χώρου, με διάλυμα ακτίνων αίσθησης (Sensing Radius)  $R_s$ , τότε ορίζοντας τις ακτίνες επικοινωνίας (Communication Radius)  $R_c$  ως διπλάσιες από τις μέγιστες ακτίνες αίσθησης, μπορούμε να εξασφαλίσουμε ότι θα έχουμε πλήρη κάλυψη επικοινωνίας, αρκεί ο χώρος ενδιαφέροντος να είναι κυρτός [30]. Παρότι το παραπάνω είναι ένα πολύ χρήσιμο κριτήριο, δεν μπορεί να αξιοποιηθεί από εμάς ώστε να

αγνοήσουμε τη μελέτη του προβλήματος της κάλυψης επικοινωνίας αφού λόγω θεώρησης των εμποδίων ο χώρος μας είναι μη κυρτός.

### 3.7. Τοποθέτηση Αισθητήρων

Ένα άλλο σχετικό ζήτημα είναι η τοποθέτηση των αισθητήρων στις περιπτώσεις που είτε οι αισθητήρες μπορούν να μετακινήθούν, είτε είναι εφικτό να ορίσουμε την αρχική τους θέση. Εκεί τα ερωτήματα που προκύπτουν πολλές φορές εμπλέκουν και την επιλογή του πλήθους των κόμβων, πέρα από το ερώτημα της τοποθέτησης τους. Αν μπορούμε να τοποθετήσουμε τους κόμβους ντετερμινιστικά, τότε πολλές φορές επιλέγεται το εξαγωνικό μοτίβο κάλυψης, που μπορεί να καλύψει βέλτιστα ή σχεδόν βέλτιστα το χώρο.

### 3.8. Κατανεμημένη ή Κεντρική Σχεδίαση

Τα θέματα σχεδίασης που προκύπτουν είναι το κατά πόσο ένας αλγόριθμος είναι τοπικός και κατανεμημένος ή όχι, δηλαδή το κατά πόσο οι αποφάσεις λαμβάνονται τοπικά, και η πληροφορία που χρησιμοποιείται για την απόφαση είναι επίσης τοπική, ή χρειάζεται η επικοινωνία με μια κεντρική οντότητα που λαμβάνει τις αποφάσεις ή μεταβιβάζει την πληροφορία [5]. Δεδομένου ότι το πλήθος των κόμβων μπορεί να φτάσει σε αρκετά μεγάλα μεγέθη, είναι εμφανής η προτίμηση ενός μη-κεντρικοποιημένου αλγόριθμου. Ανάλογα, υπάρχει αρκετό ενδιαφέρον για κατανεμημένους αλγόριθμους ή αλγόριθμους που μπορούν να μεταφραστούν σε τέτοιους [12, 25]. Ωστόσο, στα ασύρματα δίκτυα αισθητήρων υπάρχει μια μεγάλη οικογένεια από προβλήματα διαφορετικών περιορισμών ή δυνατοτήτων, και είναι χρήσιμο να συνεχίζουμε να θεωρούμε και την κεντρική περίπτωση.

## 4. ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Σε αυτήν την ενότητα θα μοντελοποιήσουμε το πρόβλημα ώστε να συνεχίσουμε με την αντιμετώπιση του. Θα δώσουμε μια περιγραφή που είναι ικανή να μας δώσει αποτελεσματικές λύσεις αλλά παράλληλα θα συζητήσουμε τις διάφορες επιλογές που υπάρχουν και ποια σημασία και επίδραση κάθε μιας από αυτών.

### 4.1. Μοντελοποίηση του Χώρου

Ο χώρος στον οποίο αναπτύσσεται το θεωρούμενο δίκτυο, μοντελοποιείται ως ένα απλό τετραγωνικό πεδίο πλευράς  $L_0$  το οποίο ξεκινάει από την αρχή των αξόνων και βρίσκεται στο πρώτο τεταρτημόριο:

$$\mathcal{F} = SQUARE_{L_0} = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^2, \mathbf{0} \preceq \mathbf{x} \preceq [L_0, L_0]^T\} \quad (1)$$

Ισοδύναμα θα μπορούσαμε να μιλήσουμε για όλο το χώρο  $\mathbb{R}^2$  με τη θεώρηση ενός άπειρου εμποδίου το οποίο καλύπτει το χώρο  $\mathcal{F}^c$ . Για τον ίδιο λόγο το σχήμα του χώρου δε μας περιορίζει, παρά μόνο σε συνδυασμό με τις διαστάσεις αυτού.

### 4.2. Μοντελοποίηση των Εμποδίων

Γενικότερα δεδομένου του  $\mathcal{F}$ , υπάρχει ένας χώρος πιθανότητας  $(\Omega, S, P)$  όπου  $\Omega = \mathcal{F}$ . Το  $S$  εκφράζει μια σ-άλγεβρα με όλα τα μετρήσιμα υποσύνολα του  $\Omega$ , και κάθε ένα εκφράζει τον ελεύθερο χώρο (δηλαδή αυτόν που προκύπτει αν αφαιρεθεί ο χώρος που καλύπτουν τα εμπόδια) στο χώρο ενδιαφέροντος. Τέλος το  $P$  είναι ένα μέτρο πιθανότητας που αποδίδει πιθανότητες σε όλες τις μετρήσιμες τοπολογίες με εμπόδια στο χώρο αυτό, δηλαδή στα σύνολα του  $S$ . Θα θέλαμε να δημιουργήσουμε μια γεννήτρια τοπολογίων που αντιστοιχίζει μια ομοιόμορφα μη μηδενική πιθανότητα σε όλους τους χώρους με εμπόδια που μπορούν να πραγματωθούν στο πλαίσιο που εξετάζουμε. Λόγου χάρη, δεν είναι ιδιαίτερα χρήσιμο να αποδώσουμε μη μηδενική πιθανότητα στο χώρο που περιλαμβάνει παντού εμπόδια, αφού είναι ένα τετριμμένο και άνευ ενδιαφέροντος σενάριο. Διαφορετικά, θεωρείστε ως εμπόδια τους τοίχους κτιρίων· σε αυτήν την περίπτωση θα θέλαμε μια κατανομή πιθανότητας που αντιστοιχεί θετική πιθανότητα σε όλους τους ρεαλιστικούς χώρους κτιρίων, με όλες τις καλές ιδιότητες που έχουν οι τοίχοι ενός κτίριου ως προς τη σχετική τους θέση. Τότε θα θέλαμε να μπορούμε να παράγουμε όλα τα δυνατά ρεαλιστικά κτίρια, και να μην αφήνουμε για παράδειγμα κλάσεις αυτών εκτός (όπως για παράδειγμα, κτίρια με ένα δωμάτιο) με το να τους αποδίδουμε μηδενική πιθανότητα.

Τα εμπόδια στη γενικότερη περίπτωση, εντός ενός δισδιάστατου κόσμου, μπορούν να είναι συνεχείς κλειστές καμπύλες με πεπερασμένο πλήθος οπών. Στο πλαίσιο του προβλήματος μας όμως, απλά πολυγωνικά εμπόδια χωρίς οπές είναι ικανά να μοντελοποιήσουν ικανοποιητικά την επίδραση των εμποδίων στο δίκτυο ενώ διευκολύνουν την περιγραφή τους (αν θεωρούσαμε οπές τότε θα διαμερίζαμε το χώρο σε περιοχές που δεν επικοινωνούν μεταξύ τους - διαφορετικές συνιστώσες, και άρα το πρόβλημα θα μπορούσε να αναχθεί σε δύο ή περισσότερα ισοδύναμα προβλήματα εμποδίων δίχως οπές). Μια απλή λύση και αρκετά γενική ώστε να παράγουμε τέτοια εμπόδια είναι η εξής: Τοποθετούνται τυχαία σε ένα τετράγωνο  $SQUARE_{L_1}$ ,  $K$  το πλήθος σημεία. Έπειτα λαμβάνουμε το Convex Hull αυτών, δηλαδή το μικρότερο κυρτό σύνολο που περιλαμβάνει τα σημεία αυτά. Στη συνέχεια υπολογίζουμε το κέντρο μάζας του Convex Hull, έστω  $[x_0, y_0]^T \in SQUARE_{L_1}$ . Μετατοπίζουμε το εμπόδιο κατά  $[-x_0, -y_0]^T$  ώστε να τοποθετηθεί το κέντρο μάζας στην αρχή των αξόνων. Τέλος παράγουμε ένα τυχαίο διάνυσμα  $[x_1, y_1]^T \in \mathcal{F}$  και μετατοπίζουμε το εμπόδιο κατά αυτό. Έτσι καταλήγουμε με ένα εμπόδιο με κέντρο μάζας στο  $[x_1, y_1]^T$ . Εάν στο σημείο που θα μετατοπιστεί το εμπόδιο τέμνεται με κάποιο άλλο, τότε αναιρούμε την τελευταία μετατόπιση και δοκιμάζουμε μια νέα· έτσι έχουμε καλύτερο έλεγχο στο αναμενόμενο εμβαδό το οποίο θα καλύπτεται από τα εμπόδια.

Συγκεκριμένα υπάρχει γενικός τύπος (Buchta, Zufallspolygone in konvexen Vielecken, Crelle, 1984) που δίνεται για το αναμενόμενο εμβαδό ενός τυχαίου Convex Hull σε ένα τετράγωνο πλευράς  $L_1$ ,

$$\mathbb{E}[\text{area}(O_i)] = L_1^2 - \frac{8L_1^2}{3(K+1)} \left( \sum_{k=1}^{K+1} \frac{1}{k} \left(1 - \frac{1}{2^k}\right) - \frac{1}{(K+1)2^{K+1}} \right) \quad (2)$$

Θεωρώντας  $N_o$  εμπόδια, το συνολικό αναμενόμενο εμβαδό των εμποδίων θα είναι

$$\mathbb{E}[\text{area}(\mathbf{O})] = N_o \mathbb{E}[\text{area}(O_i)] \quad (3)$$

Με αυτόν τον τρόπο δημιουργούμε κυρτά εμπόδια. Αυτό δε μας μειώνει σημαντικά την εκφραστικότητα περιπτώσεων αφού μπορεί να καταλήξουμε σε κυρτά εμπόδια τοποθετημένα το ένα δίπλα στο άλλο, δημιουργώντας ένα λειτουργικά κοίλο εμπόδιο. Φυσικά η διαδικασία αυτή περιγράφει μια συγκεκριμένη κατανομή ενδιαφέροντων εμποδίων. Σημειώνεται ότι τα κυρτά σύνολα στο  $\mathbb{R}^2$  είναι μετρήσιμα οπότε και η πεπερασμένη και αριθμήσιμη ένωση  $N_o$  τέτοιων ξένων κυρτών συνόλων είναι εξίσου μετρήσιμη [15] - έτσι έχει νόημα να μιλάμε για την απόδοση μέτρου πιθανότητας σε μια συγκεκριμένη διάταξη.

Επιστρέφοντας σε πιο πρακτικό επίπεδο, ο λόγος των παραμέτρων  $L_1, L_0$  καθορίζει το σχετικό μέγεθος των εμποδίων. Το  $K$  είναι μια παράμετρος που δεν πρέπει να είναι



πολύ μεγάλη, καθώς θα προκύπτουν σχήματα συνήθως όμοια με τετράγωνα (εκτός εάν μας ενδιαφέρει ένα τέτοιο περιορισμένο πλαίσιο) [3]. Με τιμές  $K \in \{5, \dots, 10\}$  έχουμε ικανοποιητικά αποτελέσματα ως προς τη διαφορετικότητα των σχημάτων των εμποδίων.

Λόγω του ότι δεν επιτρέπουμε τομή μεταξύ των εμποδίων είναι προφανές ότι πρέπει να περιορίσουμε το πλήθος αυτών, δε θα μπορούσαμε άλλωστε να έχουμε εμπόδια εμβαδού  $10A$  σε ένα χώρο  $A$ .

Θεωρούμε ότι το  $i$ -οστό εμπόδιο που προέκυψε από την παραπάνω διαδικασία περιγράφεται από ένα σύνολο διατεταγμένων σημείων που δηλώνουν τις κορυφές του, τότε για την πλήρη περιγραφή του αρκούν τα σημεία αυτά:

$$O_i = Polygon(o_1, o_2, \dots, o_i)$$

Τελικά το σύνολο των εμποδίων είναι

$$\mathbf{O} = \{O_i \mid i \in \{1, \dots, N_o\}\}$$

Τα δύο χαρακτηριστικά αυτών των εμποδίων είναι τα παρακάτω

1. Δε χρειάζεται να καλύψουμε το χώρο που καλύπτει το εμπόδιο.
2. Ένας αισθητήρας δεν μπορεί να καλύψει το χώρο πίσω από το εμπόδιο, επειδή η ορατότητα γραμμής (line of sight) διακόπτεται από το εμπόδιο.

Συνεπώς οποιοσδήποτε χώρος χαρακτηρίζεται από αυτές τις δύο ιδιότητες μπορεί σε ένα πρόβλημα να μοντελοποιηθεί ως ένα αδιαφανές εμπόδιο.

Η διαδικασία με την οποία προκύπτουν τα εμπόδια αποτελεί μια marked point process, όπου βρισκόμαστε στο χώρο  $\mathcal{F} \times T \times C$ , όπου το  $\mathcal{F}$  εκφράζει το χώρο ενδιαφέροντος μέσα στον οποίον καταλήγει το κέντρο μάζας του εμποδίου, και  $T \times C$  είναι ο χώρος των μαρκάριασματων (marks), όπου συγκεκριμένα το  $T$  είναι ένας δείκτης του βήματος κατα το οποίο εισήχθη το εμπόδιο, και  $C$  είναι ο χώρος των κυρτών εμποδίων που μπορούν να προκύψουν με την διαδικασία που περιγράφηκε. Έπειτα σε αυτή την Marked Point Process κάνουμε εκλέπτυνση διαγράφοντας τα εμπόδια που έχουν επικάλυψη με άλλα εμπόδια τα οποία έχουν μικρότερο αριθμητικά δείκτη (δηλαδή προηγούνται χρονικά), και δεν έχουν διαγραφεί νωρίτερα. Επειδή η πληροφορία του δείκτη-χρόνου  $i \in T$  είναι περιττή μπορούμε να την αφαιρέσουμε από το γενικό μαρκάρισμα.

---

### Συνάρτηση 1: Δημιουργία εμποδίων

---

```
1 function CreateObstacles( $n_0$ ,  $K$ )
2  $\mathbf{O} = \emptyset$ 
3 for  $i$  from 1 to  $n_o$  do
4   randomPoints =  $L_1$  rand( $K,2$ )
5   obs = convexHull(randomPoints)
6   center = centerOfMass(obs)
7   centeredObstacle = translate(obs, -center)
8   while True do
9     newPoint =  $L_1$  rand(1,2)
10    newObs = translate(centeredObstacle, newPoint)
11    if NOT intersectAny( $\mathbf{O}$ , newObs) then
12      Break
13    end
14  end
15   $\mathbf{O} = \mathbf{O} \cup$  newObs
16 end
```

---

### 4.3. Μοντελοποίηση των Αισθητήρων

Οι  $N$  αισθητήρες περιγράφονται πλήρως από το σύνολο με τα σημεία τους:

$$\mathbf{P} = \{s_i = [x_i, y_i]^T \in \mathcal{F} \mid i \in \{1, \dots, N\}\}$$

Άλλοι παράμετροι για έναν αισθητήρα είναι το εύρος των πιθανών ακτίνων του δίσκου. Για απλότητα θα θεωρήσουμε ότι όλες οι ακτίνες έχουν εύρος  $[0, R_{max}]$ , όμως οι μέθοδοι που θα αναπτυχθούν γενικεύονται εξίσου εύκολα σε μη-ομοιογενή δίκτυα. Στη βιβλιογραφία έχουν μελετηθεί είτε συνεχείς αλλαγές στην ακτίνα του δίσκου είτε διακριτές όπως είδαμε στην ενότητα 3.3. Εμείς θεωρούμε για μεγαλύτερη ευελιξία τη συνεχή περίπτωση, κατανοώντας ότι δίχως σημαντικό κόστος στις περισσότερες περιπτώσεις μπορούμε να μεταβούμε στη διακριτή περίπτωση.

Υποθέτουμε ότι μια διαδικασία Poisson Point Process είναι η γεννήτρια αυτών των σημείων των αισθητήρων, στο χώρο που δεν καταλαμβάνεται από εμπόδια. Δεσμευμένου του πλήθους των αισθητήρων, αυτοί θα τοποθετηθούν ομοιόμορφα στον ελεύθερο χώρο ως μια Binomial Point Process [11, 1]. Αυτό είναι ένα ρεαλιστικό σενάριο, σε περιπτώσεις που γίνεται τοποθέτηση αισθητήρων σε αντίξοο περιβάλλον. Σε ένα τέτοιο σενάριο είναι εύλογο να θεωρήσουμε ότι γνωρίζουμε το πλήθος των αισθητήρων.

Μια εναλλακτική, και εύλογη υπόθεση, σε σενάρια όπου χρησιμοποιούνται αεροσκάφη τα οποία ρίχνουν από τον εναέριο χώρο τους αισθητήρες σε εχθρικό περιβάλλον, είναι η θεώρηση μιας Cluster Process ανάλογη της Matern Cluster Process. Αυτό μπορεί να αποδοθεί σε μαζική εκτόξευση αισθητήρων σε διαφορετικά σημεία από το αεροσκάφος, με αποτέλεσμα ομάδες αισθητήρων να έχουν στενή χωρική σχέση.

#### 4.4. Δυνατές Επεκτάσεις του Μοντέλου

Γενικότερα δε θεωρούμε εμπόδια τα οποία είναι διαφανή (transparent), δηλαδή αυτά τα οποία βρίσκονται σε χώρο που δε χρειάζεται να καλύψουμε, αλλά παράλληλα δεν εμποδίζουν την ορατότητα γραμμής της κάλυψης ενός αισθητήρα. Αυτό σημαίνει ότι από τα δύο χαρακτηριστικά που αναφέραμε νωρίτερα για το τι μπορεί να μοντελοποιηθεί ως αδιαφανές (opaque) εμπόδιο, αφαιρούμε τον δεύτερο περιορισμό. Είναι εφικτό να αντιμετωπίσουμε αυτή την περίπτωση αν θεωρήσουμε τα εμπόδια αυτά ως ψευδοαισθητήρες που έχουν σταθερή κάλυψη στο χώρο, ίδια με εκείνη που καταλαμβάνεται από το εμπόδιο. Έτσι για την γενίκευση αρκεί να χρησιμοποιήσουμε σε όποια σημεία έχει νόημα τους ψευδοαισθητήρες, όπου βέβαια, αυτοί δεν λαμβάνονται υπόψη στην αλλαγή ακτίνων.

#### 4.5. Τυποποίηση του Προβλήματος

##### 4.5.1. Κλασματική Προσέγγιση

Μπορούμε να ορίσουμε το πρόβλημα ως ένα πρόβλημα βελτιστοποίησης:

$$\arg \max_{\mathbf{R} \in \mathcal{S} \subseteq \mathcal{D}} \left\{ \frac{Covered(\mathbf{R}; \mathbf{O}, \mathbf{P})}{Energy(\mathbf{R})} \right\} \quad (4)$$

$$\mathcal{S} = \{\mathbf{r} \in \mathcal{D} \mid Covered(\mathbf{r}; \mathbf{O}, \mathbf{P}) \geq T \cdot Covered(\mathbf{R}_{max}; \mathbf{O}, \mathbf{P})\} \quad (5)$$

όπου  $\mathbf{R} = [R_1, R_2, \dots, R_n]^T$  το διάνυσμα με τις ακτίνες των αισθητήρων,  $\mathcal{D} \subset \mathbb{R}^n$ , ο χώρος των επιτρεπτών ακτίνων, που γενικότερα είναι ένα  $n$ -ορθότοπο (γενίκευση του ορθογώνιου παραλληλόγραμμου σε  $n$  διαστάσεις),  $\mathbf{R}_{max}$  είναι το διάνυσμα με όλες τις μέγιστες ακτίνες (σημείο του ορθότοπου με μέγιστη ευκλείδεια νόρμα). Επιπλέον η συνάρτηση  $Covered : \mathcal{D} \rightarrow \mathbb{R}$ , με παράμετρο τα εμπόδια  $\mathbf{O}$ , και τα σημεία  $\mathbf{P}$ , μας δίνει το εμβαδόν της επιφάνειας που καλύπτουν συνεργατικά οι κόμβοι. Από εδώ και στο εξής για απλότητα οι παράμετροι δε θα σημειώνονται παρα μόνον αν είναι αναγκαίο.

Επιπλέον, για να λάβουμε μη τετριμμένα αποτελέσματα, περιορίζουμε το χώρο των λύσεων σε εκείνες τις λύσεις που ανήκουν στο σύνολο  $\mathcal{S}$ , δηλαδή εκείνες που έχουν

λόγο εμβαδού κάλυψης με την επιλεγμένη ανάθεση ακτίνων προς εμβαδού κάλυψης με τις μέγιστες ακτίνες, πάνω από ένα προαποφασισμένο όριο  $T$  (ένα “threshold” που επιλέγει ο σχεδιαστής του συστήματος, με την υπόθεση ότι αρκεί για την εφαρμογή).

Για την *Covered*, ο ορισμός χρησιμοποιεί την έννοια του Total Coverage,  $TCOV(\mathbf{R})$ , που περιγράφεται στην ενότητα 5.1, και εκφράζει το μαθηματικό σύνολο της ολικής κάλυψης για μια συγκεκριμένη ανάθεση. Ορίζεται η κάλυψη:

$$Covered(\mathbf{R}) := \text{area}(TCOV(\mathbf{R})) \quad (6)$$

Τέλος, ορίζεται και η συνάρτηση  $Energy : \mathcal{D} \rightarrow \mathbb{R}$ , που μας δίνει για την ανάθεση ακτίνων  $\mathbf{r} \in \mathcal{D}$ , το κόστος ενέργειας. Θεωρούμε πως η συνάρτηση αυτή, στην γενικότερη μορφή της, μπορεί να γραφεί ως

$$Energy([R_1, R_2, \dots, R_n]^T) = \sum_{i=1}^n f_i(R_i)$$

Δηλαδή μπορεί να διαχωριστεί σε άθροισμα συναρτήσεων μιας μεταβλητής ανεξάρτητων μεταξύ τους. Στην μελέτη μας αρκεί να θεωρήσουμε ότι οι συναρτήσεις  $\{f_i\}$  ταυτίζονται και είναι αύξουσες (συμβολίζουμε  $f^\dagger$ ):

$$Energy([R_1, R_2, \dots, R_n]^T) = \sum_{i=1}^n f^\dagger(R_i) \quad (7)$$

Τέλος πολλές φορές θα θεωρούμε την ειδική περίπτωση όπου η συνάρτηση είναι σε Τετραγωνική Μορφή (Quadratic Form), δηλαδή

$$Energy(\mathbf{r}) = \mathbf{r}^T \mathbf{A} \mathbf{r} \quad (8)$$

όπου συνδυασμένο με τους περιορισμούς του μοντέλου της σχέσης 7 έχει ως αποτέλεσμα  $\mathbf{A} = c\mathbf{I}$ , με  $c > 0$ . Στην εργασία αυτή θα εξετάσουμε την περίπτωση που  $c = 2$ .

#### 4.5.2. Απλοποιημένη Τυποποίηση του Προβλήματος

Μια απλοποίηση στην έκφραση βελτιστοποίησης του προβλήματος θα μπορούσε να οδηγήσει στην παρακάτω τυποποίηση:

$$\arg \min_{\mathbf{R} \in \mathcal{S} \subseteq \mathcal{D}} \{Energy(\mathbf{R})\} \quad (9)$$

Συμβολίζουμε  $\partial\mathcal{S}$ , και θα αποκαλούμε ως σύνορο, από εδώ και στο εξής το πραγματικό σύνορο του  $\mathcal{S}$ ,  $\partial_0\mathcal{S}$ , εάν από αυτό αφαιρέσουμε το σύνορο του  $\mathcal{D}$  και πάρουμε την κλειστότητα του αποτελέσματος. Δηλαδή  $\partial\mathcal{S} := cl(\partial_0\mathcal{S} \setminus \partial\mathcal{D})$ , όπου  $cl(\cdot)$  η κλειστότητα ενός συνόλου. Τότε θεωρώντας ένα διάνυσμα  $\mathbf{r} \in \mathcal{S} \setminus \partial\mathcal{S}$ , πάντα μπορούμε να μειώσουμε κάποιες από τις ακτίνες, καταλήγοντας στο διάνυσμα  $\mathbf{r}' = \mathbf{r} - \boldsymbol{\epsilon}$ ,  $\boldsymbol{\epsilon} \in \mathbb{R}^{n+}$ , και παραμένοντας στην εφικτή περιοχή (feasible region). Επιπλέον η συνάρτηση  $Energy(\cdot)$  διατηρεί την κατά συνιστώσες διάταξη. Δηλαδή:

$$(\mathbf{r}' \preceq \mathbf{r}) \wedge (\mathbf{r}' \neq \mathbf{r}) \Rightarrow Energy(\mathbf{r}') < Energy(\mathbf{r}) \quad (10)$$

Συνεπώς δεν είναι εφικτό να έχω ελάχιστο ενέργειας στο  $\mathcal{S} \setminus \partial\mathcal{S}$ . Άρα η αναζήτηση αρκεί να περιοριστεί στο σύνορο  $\partial\mathcal{S}$ , αφού μόνο εκεί μπορεί να ελαχιστοποιηθεί η ενέργεια, διαφορετικά, στο εσωτερικό της εφικτής περιοχής πάντα μπορώ να μειώσω κάποιες ακτίνες και να βελτιώσω την αντικειμενική συνάρτηση (δηλαδή να μειώσω την ενέργεια), και άρα ισοδύναμα:

$$\arg \min_{\mathbf{R} \in \partial\mathcal{S} \subseteq \mathcal{D}} \{Energy(\mathbf{R})\} \quad (11)$$

Το σύνορο  $\partial\mathcal{S}$  γενικά δεν ταυτίζεται με το σύνολο:

$$\tilde{\mathcal{S}} = \{\mathbf{r} \in \mathcal{D} \mid Covered(\mathbf{r}) = T \cdot Covered(\mathbf{R}_{max})\} \quad (12)$$

Μπορούμε να ισχυριστούμε ότι το  $\partial\mathcal{S} \subseteq \tilde{\mathcal{S}}$  διότι η συνάρτηση  $Covered$  είναι συνεχής ως προς τις ακτίνες και συνεπώς δεν μπορούμε να μεταβούμε από ένα χώρο  $Covered(\mathbf{r}) < C$  σε έναν χώρο  $Covered(\mathbf{r}) > C$  χωρίς να περάσουμε από το  $Covered(\mathbf{r}) = C$ , άρα όποιο σημείο ανήκει στο  $\partial\mathcal{S}$  πρέπει να έχει  $Covered(\mathbf{r}) = T \cdot Covered(\mathbf{R}_{max})$ . Το  $\tilde{\mathcal{S}}$  όμως γενικότερα θα μπορούσε να είναι ένα ευρύτερο σύνολο από το  $\partial\mathcal{S}$ . Για παράδειγμα εάν περιλαμβάνει  $\mathbf{r} \in \mathcal{D}$  τα οποία να μην έχουν συνεργατική κάλυψη που οδηγεί στο όριο  $T$ , αλλά οι ακτίνες μπορεί να είναι τέτοιες ώστε να υπάρχει πλεονασμός. Ως πλεονασμό εννοούμε ότι όποια ακτίνα και να μειωθεί στο διάνυσμα  $\mathbf{r}$ , θα έχουμε ίδια κάλυψη και συνεπώς θα παραμένουμε στην εφικτή περιοχή. Πράγματι, αρχικά μπορεί κανείς να φανταστεί δύο πολύ απλά σενάρια όπου τα δύο σύνολα διαφέρουν: ένα σενάριο είναι όταν δύο ή περισσότεροι αισθητήρες βρίσκονται στο ίδιο ακριβώς σημείο, και έχουν ίδια ακτίνα, κατάλληλη ώστε να έχουμε ακριβώς την κάλυψη που επιθυμούμε: τότε όποιου αισθητήρα και να μειώσουμε την ακτίνα, διατηρούμε ίδια κάλυψη  $T$  αφού οι άλλοι αισθητήρες φροντίζουν για την κάλυψη του ίδιου χώρου, και συνεπώς παραμένουμε στην εφικτή περιοχή  $\mathcal{S}$ , άρα δεν γίνεται να βρισκόμασταν στο σύνορο  $\partial\mathcal{S}$ , όμως από την άλλη βρισκόμασταν σίγουρα στο  $\tilde{\mathcal{S}}$  (αφού ίσχυε η ισότητα). Εάν θεωρήσουμε ότι

η παραγωγή των σημειών όμως προέρχεται από μια απλή σημειακή διαδικασία, τότε δεν θα μπορούσαμε να έχουμε αισθητήρες στο ίδιο σημείο (σχεδόν σίγουρα) - αλλά σε κάθε περίπτωση το σενάριο αυτό είναι αρκετά τετριμμένο ώστε να μη μας απασχολεί. Μια άλλη περίπτωση στην οποία αυτό μπορεί να συμβεί είναι όταν θέλουμε πλήρη κάλυψη, δηλαδή όταν  $T = 1$ . τότε για παράδειγμα όλες οι ακτίνες στο μέγιστο τους δίνουν μια ανάθεση η οποία ανήκει στο  $\tilde{\mathcal{S}}$ , αφού έχει 100% κάλυψη, όμως αυτό μπορεί να μην ανήκει στο σύνορο  $\partial\mathcal{S}$  γιατί εάν μειώσουμε οποιαδήποτε ακτίνα δεν είναι αναγκαίο να βγαίνουμε από την εφικτή περιοχή. Αυτό φαίνεται και στο σχήμα 4. Στην περίπτωση που  $T < 1$ , η κατάσταση διαφοροποιείται σημαντικά. Πρέπει κανείς να εξετάσει εάν είναι εφικτό να έχουμε μια ανάθεση με μερική κάλυψη στο όριο που επιθυμούμε  $T (< 1)$ , και να έχουμε πλεονασμό στις ακτίνες, δηλαδή αν μειωθεί οποιαδήποτε ακτίνα να μην διαφοροποιείται η κάλυψη. Αυτό θα ήταν ισοδύναμο με το να ήταν εφικτό να υπάρχει μια ανάθεση στην οποία έχουμε την επιθυμητή κάλυψη, και όποιον κόμβο και να αυξήσουμε αυτή η κάλυψη να μην αλλάζει έως ένα σημείο, μετά το οποίο η κάλυψη πάλι αυξάνει (διότι πρέπει να μπορεί να αυξάνει ώστε να φτάσει στο 100% της κάλυψης). Από την στιγμή που δεν αναφερόμαστε σε ανάθεση πλήρης κάλυψης, τουλάχιστον ένας κόμβος δε θα έχει πλήρως καλυμμένη την περιφέρεια του από γειτονικούς κόμβους σύμφωνα με τα αποτελέσματα που αναφέρθηκαν στην ενότητα με το Σχετικό Έργο. Στο σημείο που δεν είναι καλυμμένη η περιφέρεια θα υπάρχει κενός χώρος προς κάλυψη, και με την αλλαγή της ακτίνας η τιμή της κάλυψης θα αλλάξει - ως εκ τούτου για  $T < 1$  δεν γίνεται να υπάρχει σημείο του  $\tilde{\mathcal{S}}$  που να μην είναι στο σύνορο. Συνεπώς  $\tilde{\mathcal{S}} = \partial\mathcal{S}$ .

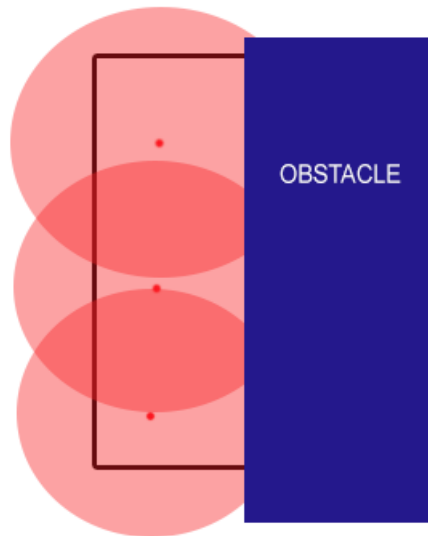
**Σημείωση:** Περιορισμένοι στο  $\tilde{\mathcal{S}}$  για την κλασματική έκδοση βελτιστοποίησης, το  $Covered(\mathbf{R})$  είναι σταθερό και ισούται με  $T \cdot Covered(\mathbf{R}_{max}) = (Constant)$ , άρα

$$\arg \max_{\mathbf{R} \in \tilde{\mathcal{S}}} \left\{ \frac{Covered(\mathbf{R})}{Energy(\mathbf{R})} \right\} = \arg \max_{\mathbf{R} \in \tilde{\mathcal{S}}} \left\{ \frac{(Constant)}{Energy(\mathbf{R})} \right\} = \arg \min_{\mathbf{R} \in \tilde{\mathcal{S}}} \{Energy(\mathbf{R})\}$$

Δηλαδή τα δύο προβλήματα είναι ισοδύναμα σε ένα περιορισμένο σύνολο εφικτών λύσεων.

#### 4.5.3. Λύση της Απλοποιημένης Εκδοχής στο Περιορισμένο Σύνολο

Όπως αναφέρθηκε και στη σημείωση ωρίτερα, η απλοποιημένη εκδοχή με την αρχική κλασματική εκδοχή είναι το ίδιο πρόβλημα στο περιορισμένο σύνολο  $\tilde{\mathcal{S}}$ . Για να κατανοήσουμε καλύτερα αυτήν την σχέση και το πως μπορούμε να την εκμεταλλευτούμε, μπορούμε να χρησιμοποιήσουμε το σχήμα 5. Σε αυτό βλέπουμε το χώρο καταστάσεων



Σχήμα 4: Στο σχήμα αυτό βλέπουμε ένα εμπόδιο και τρεις αισθητήρες με κέντρα τις κόκκινες κουκκίδες, και με μια τρέχουσα κάλυψη - δίσκο. Συνολικά οι δίσκοι κάλυψης καλύπτουν όλη την ενδιαφέρουσα επιφάνεια (εντός της περιοχής μεταξύ του μαύρου συνόρου και του εμπόδιου). Εάν κάποιος δίσκος μειώσει την ακτίνα του η κάλυψη παραμένει στο 100% της μέγιστης και άρα βρισκόμαστε στο  $\tilde{S}$  αλλά όχι στο  $\partial S$ .

των ακτίνων σε ένα πρόβλημα με μόνο δύο αισθητήρες, που οι ακτίνες τους είναι οι  $R_1$  και  $R_2$ . Έτσι στο γράφημα βλέπουμε τον συνδυασμό των  $R_1, R_2$  στην οριζόντια και κάθετη διάσταση αντίστοιχα. Για παράδειγμα η κάτω αριστερά κουκκίδα αντιστοιχεί στην περίπτωση που και οι δύο αισθητήρες έχουν ακτίνα 0, δηλαδή το ζεύγος  $(R_1, R_2) = (0, 0)$ . Η πάνω δεξιά κουκκίδα αντιστοιχεί στην περίπτωση που οι δύο αισθητήρες έχουν μέγιστη ακτίνα  $(R_1, R_2) = (R_{max}, R_{max})$ . Ο χρωματισμένος χώρος με πράσινο, είναι όλο το εσωτερικό της εφικτής περιοχής, δηλαδή ζεύγη ακτίνων τα οποία μας δίνουν μια έγκυρη, αλλά όχι αναγκαία βέλτιστη λύση. Η κόκκινη γραμμή εκφράζει το σύνορο (όπως ορίστηκε) της εφικτής περιοχής, δηλαδή όλοι οι συνδυασμοί που είναι οριακά εντός της εφικτής περιοχής - δεν υπάρχει σφαίρα με κέντρο κάποιο σημείο πάνω στη γραμμή, με οποδήποτε μικρή ακτίνα, η οποία να ανήκει ολοκληρωτικά στην εφικτή περιοχή. Οι μαύρες κουκκίδες εκφράζουν μια διακριτοποίηση του χώρου καταστάσεων, όπου χρησιμοποιείται ένα ομοιόμορφο πλέγμα, και οι ακτίνες μπορούν να αυξηθούν κατά μια ποσότητα, έστω  $\Delta r$ . Η κλασματική έκδοση έχει λύση που μπορεί να βρισχεται πάνω στην κόκκινη γραμμή ή στον γενικότερο πράσινο χώρο (Πρόβλημα 1) ενώ η απλοποιημένη έκδοση όπως είδαμε αρκεί να περιοριστεί στην κόκκινη γραμμή (Πρόβλημα 2). Το ότι τα δύο προβλήματα ταυτίζονται στο σύνορο μας δίνει την δυνατότητα να κινούμαστε με βάση

το κριτήριο το οποίο έχει νόημα για το Κλασματικό Πρόβλημα 1, όσο βρισκόμαστε σε σημεία εκτός από το σύνορο της εφικτής περιοχής, και να σταματάμε όταν βρισκόμαστε οριακά πάνω ή κάτω από το όριο κάλυψης  $T$ . Συγκεκριμένα στην παρούσα εργασία θα σταματάμε όταν θα βρισκόμαστε οριακά πάνω από το όριο. Τότε θα βρισκόμαστε κοντά στο σύνορο της εφικτής περιοχής, με την εκτίμηση ότι είμαστε σε μια καλή λύση για το Κλασματικό Πρόβλημα 2 και το Κλασματικό Πρόβλημα 1 πάνω στο σύνορο αυτό. Έτσι παρατηρώντας το σχήμα και πάλι, βλέπουμε ότι υπάρχει ένα μονοπάτι που εκφράζει την μεταβολή των καταστάσεων, και το οποίο ξεκινάει από το  $(0, 0)$ , δηλώνοντας ότι οι ακτίνες αρχικοποιούνται με μηδενική ακτίνα, και έπειτα κάθε φορά μια ακτίνα αυξάνεται (κίνηση πάνω ή δεξιά) ή μειώνεται (κίνηση κάτω ή αριστερά) κατά  $\Delta r$  σύμφωνα με το κριτήριο του αλγορίθμου μας. Βλέπουμε ότι το μονοπάτι σταματάει με το που εισέλθει στην εφικτή περιοχή, σε ένα σημείο,  $a$ , που θα θέλαμε να αποτελεί ελάχιστο μεταξύ των σημείων της διακριτοποίησης. Μια τέτοια προσέγγιση είναι ένας Άπληστος Προσεγγιστικός αλγόριθμος που ξεκινάει από μη εφικτή λύση, και χτίζει μια λύση εφικτή, σε αντιδιαστολή με έναν αλγόριθμο τοπικής αναζήτησης που ξεκινάει από μια εφικτή λύση και παραμένει σε τέτοιες λύσεις.

Στο ίδιο σχήμα φαίνεται και εκφράζεται ότι σε ολόκληρη την εφικτή περιοχή μπορεί να υπάρχει ένα διαφορετικό από το  $b$  ολικό μέγιστο,  $c$ , ενώ από όλα τα σημεία της διακριτοποίησης μπορεί να υπάρχει ένα διαφορετικό από το  $a$  μέγιστο,  $d$ .

Τότε θα ισχύει βέβαια, για το αντίστοιχο κριτήριο  $F$

$$F(a) \leq \{F(d), F(b)\} \leq F(c)$$

Ο συμβολισμός του συνόλου εκφράζει ότι η ανισότητα ισχύει με οποιοδήποτε στοιχείο του συνόλου. Σημειώνεται ότι δεν γνωρίζουμε ποιο από τα δύο,  $F(b), F(d)$  είναι μεγαλύτερο.

**Μαθηματικός και Κλασματικός Προγραμματισμός.** Με την παραπάνω αρχική τυποποίηση, το πρόβλημα μας είναι ένα πρόβλημα κλασματικού προγραμματισμού (Fractional Programming), δηλαδή ένα πρόβλημα βελτιστοποίησης που ορίζεται από ένα κλάσμα δύο οχι αναγκαία γραμμικών συναρτήσεων.

#### 4.5.4. Μορφή της Συνάρτησης Ενέργειας

Η συνάρτηση ενέργειας εξαρτάται από τη συνάρτηση  $f^\dagger$ . Εάν υποθέσουμε ότι η συνάρτηση αυτή είναι κυρτή (convex) και δύο φορές διαφορίσιμη τότε μπορούμε εύκολα να υπολογίσουμε το Hessian πίνακα ως εξής:





Σχήμα 5: Στο σχήμα αυτό παρουσιάζεται ο χώρος καταστάσεων για ένα πρόβλημα με δύο αισθητήρες. Παράλληλα βλέπουμε μια εξέλιξη ενός βηματικού αλγόριθμου που ξεκινάει από την μηδενική κατάσταση  $(R_1, R_2) = (0, 0)$  και καταλήγει στην κατάσταση  $a$ . Η κατάσταση  $a$  θεωρείται ελάχιστο από τις διακριτές τιμές του πλέγματος, αμέσως εντός της εφικτής περιοχής. Το  $b$  εκφράζει ένα σημείο που είναι ελάχιστο πάνω στο σύνορο της εφικτής περιοχής. Το  $c$  εκφράζει το ολικό μέγιστο σε όλη την εφικτή περιοχή. Το  $d$  εκφράζει το ολικό μέγιστο εντός της εφικτής περιοχής αλλά πάνω στην διακριτοποίηση.

$$H = \begin{bmatrix} f''(x_1) & 0 & 0 & \dots & 0 \\ 0 & f''(x_2) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & f''(x_1) \end{bmatrix} \quad (13)$$

Ο πίνακας είναι θετικά ημιορισμένος (positive semidefinite). Το ίδιο ισχύει και για την περίπτωση της σχέσης 8. Συνεπώς υπο τη συνθήκη αυτή η συνάρτηση *Energy* είναι κυρτή.

#### 4.5.5. Μορφή της Συνάρτησης Κάλυψης

Η συνάρτηση κάλυψης είναι το εμβαδόν μιας ένωσης απο πολυγωνικές περιοχές και ορίστηκε στη σχέση 6.

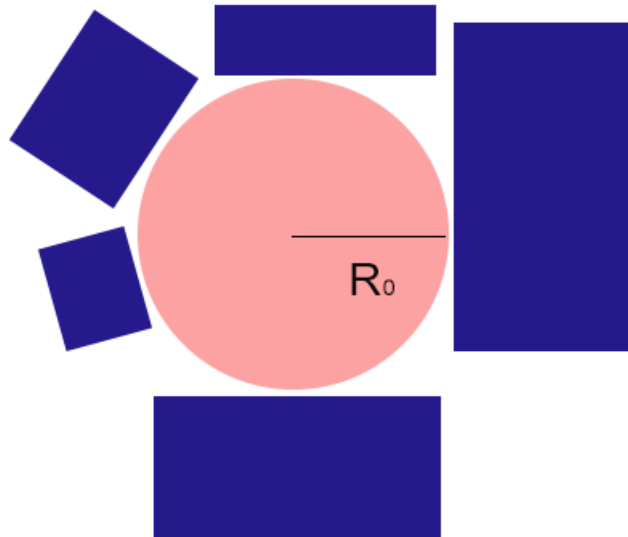
**Συνέχεια και Παραγωγισιμότητα.** Θα θέλαμε να γνωρίζουμε κάποια πράγματα για την ομαλότητα της συνάρτησης κάλυψης, όπως για παράδειγμα το κατά πόσο είναι συνεχής. Ως πρώτο βήμα ας μελετήσουμε την περίπτωση ενός σημείου και ενός εμπόδιου. Όσο ο αισθητήρας έχει ακτίνα  $r$  τέτοια ώστε να μην εμποδίζεται καθόλου από το εμπόδιο, η κάλυψη θα είναι  $\pi r^2$ . Όταν φτάσουμε σε μια ακτίνα  $r_0$ , μετά την οποία η περιοχή του εμπόδιου και ο δίσκος κάλυψης τέμνονται, η κάλυψη θα γίνει  $\pi r^2 - E(r)$ , όπου  $E(r)$  είναι το εμβαδό της τομής του δίσκου με ακτίνα  $r$  και κέντρο τον αισθητήρα, και του ψευδοεμπόδιου, το οποίο ορίζεται ως το ισοδύναμο εμπόδιο που αντιλαμβάνεται ο αισθητήρας λόγω του ότι δεν μπορεί να καλύψει χώρο στον οποίο δεν διαθέτει Line of Sight. Η συνάρτηση  $E$  μπορεί να γραφτεί ως

$$E(r) = \iint_S dx dy = \iint_{\Omega} r' dr' d\theta = \int_{r_0}^r f(r') 2\pi r' dr'$$

Όπου  $f(r)$  είναι μια τμηματικά συνεχής συνάρτηση με πεπερασμένα σημεία ασυνέχειας (σχεδόν παντού συνεχής, και πεπερασμένη, άρα Riemann ολοκληρώσιμη), που περιγράφει το ποσοστό του τόξου του κύκλου με κέντρο τον αισθητήρα και ακτίνα  $r$  που καταλαμβάνεται από το ψευδοεμπόδιο. Έτσι σαν τμηματικά συνεχής συνάρτηση είναι ολοκληρώσιμη. Συνολικά η  $E(r)$  είναι συνεχής αφού η απεικόνιση  $f \mapsto \int_a^x f$  στέλνει το σύνολο των ολοκληρώσιμων συναρτήσεων στο σύνολο των συνεχών συναρτήσεων. Άρα και η έκφραση  $\pi r^2 - E(r)$  θα είναι συνεχής ως διαφορά συνεχών συναρτήσεων. Ανάλογα επιχειρήματα μπορούν να χρησιμοποιηθούν για περισσότερους αισθητήρες όπου μπορεί να χρησιμοποιηθεί η μέθοδος εγκλεισμού-αποκλεισμού για να λογαριάσει κανείς την αλληλεπίδραση των εμβαδών μεταξύ των καλύψεων των αισθητήρων. Ως προς την παραγωγισιμότητα, στην περίπτωση γενικών εμποδίων μπορεί κανείς να κατασκευάσει εμπόδια ώστε να κάνει τη συνάρτηση κάλυψης μη παραγωγίσιμη.

**Κυρτότητα.** Η συνάρτηση της κάλυψης μπορεί επίσης να δειχθεί εύκολα ότι δεν είναι ούτε κυρτή ούτε κοίλη. Πράγματι, παίρνοντας κανείς μια απλή περίπτωση ενός κόμβου, και ενός εμπόδιου, τότε έχουμε ένα μονοδιάστατο χώρο καταστάσεων, και η συνάρτηση *Coverage* είναι επίσης μια απλή συνάρτηση μιας μεταβλητής. Σε αυτήν την περίπτωση ενώ το εμβαδό που καλύπτει ένας αισθητήρας μέχρι ένα σημείο είναι  $\pi r^2$ , μετά, με ένα σύνολο εμποδίων που περικυκλώνουν τον αισθητήρα, μπορούμε να κάνουμε την αύξηση του εμβαδού από εκεί και πέρα τέτοια ώστε η τελική συνάρτηση *Coverage* να μην είναι κυρτή. Ένα παράδειγμα δίνεται στο σχήμα 6. Πράγματι βλέπουμε ότι μέχρι το  $R_0$  το *Coverage* θα είναι κυρτή συνάρτηση και ανάλογη του  $\pi r^2$ , όμως από εκεί και πέρα η αύξηση δε θα οδηγήσει σε σημαντική συνεισφορά εμβαδού, και δε θα ισχύει η βασική σχέση κυρτότητας: παίρνοντας δύο σημεία, έστω  $x_1 = 0$  και  $x_2 \gg R_0$ , η γραμμή που τα ενώνει

θα βρίσκεται κάτω από την καμπύλη της συνάρτησης *Coverage*.



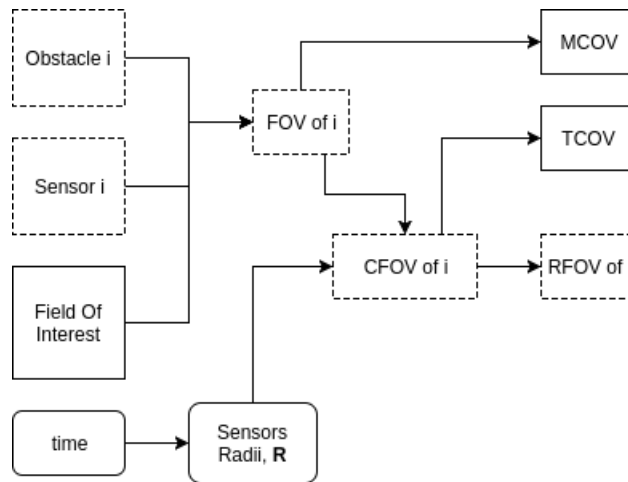
Σχήμα 6: Ένας αισθητήρας ο οποίος είναι περικυκλωμένος από εμπόδια. Περισσότερη αύξηση της ακτίνας δε θα έχει ως αποτέλεσμα σημαντικό χρήσιμο εμβαδό

## 5. ΕΥΡΕΤΙΚΕΣ ΛΥΣΕΙΣ

Σε αυτή την ενότητα θα παρουσιάσουμε κάποιους ευρετικούς αλγόριθμους για την προσέγγιση του προβλήματος που περιγράφηκε στο κεφάλαιο 4. Αρχικά θα δοθούν κάποιες προκαταρκτικές έννοιες, και έπειτα θα δοθούν οι τρεις διαφορετικές ευρετικές.

### 5.1. Προκαταρκτικές Έννοιες

Πριν προχωρήσουμε θα παρουσιάσουμε κάποιες προκαταρκτικές έννοιες που θα χρησιμοποιηθούν εκτενώς αργότερα. Στο σχήμα 7 περιλαμβάνονται οι πιο σημαντικές, και οι μεταξύ τους εξαρτήσεις. Για παράδειγμα το  $FOV_i$ , το οποίο αναφέρεται σε έναν κόμβο και άρα φαίνεται με διακεκομμένες γραμμές, εξαρτάται από τις παραμέτρους στο σύστημα, και καθορίζει με τη σειρά του το  $MCOV$  αλλά και το  $CFOV_i$ . Στο συμβολισμό αργότερα η εξάρτηση από τα τετραγωνισμένα μεγέθη θα είναι συνήθως μόνο σιωπηρή, ενώ η εξάρτηση από την μεταβλητή  $\mathbf{R}$  θα χρησιμοποιείται μόνον όταν κρίνεται αναγκαίο - λόγω χάρη όταν θέλουμε να δώσουμε έμφαση σε συγκεκριμένη ανάθεση. Για παράδειγμα μερικές φορές θα γράφουμε  $CFOV_i$  ενώ άλλες, όπως όταν συγκρίνουμε δύο αναθέσεις  $\mathbf{R}_k$  θα γράφουμε  $CFOV_i(\mathbf{R}_k)$ . Τέλος σημειώνεται ότι εργαζόμαστε με κλειστά σύνολα. Οπότε, παρότι δε θα λέγεται ρητά, μετά από κάθε πράξη το τελικό αποτέλεσμα θα είναι η ένωση του αποτελέσματος με το σύνολο του αποτελέσματος.



Σχήμα 7: Εξαρτήσεις Διάφορων Μεγεθών

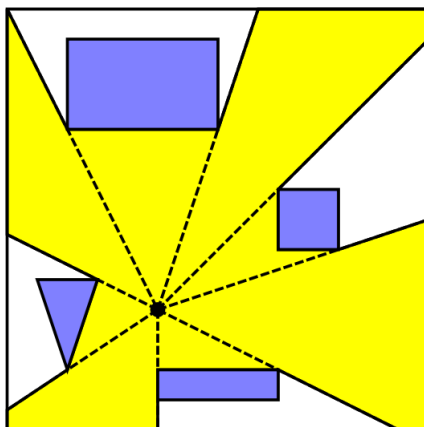
**Δίσκος  $DISK$  και Δαχτυλίδι  $RING$ :** Ορίζουμε τη συνάρτηση δίσκου  $DISK(s, r)$ , που δέχεται είσοδο ένα σημείο και μια ακτίνα, και επιστρέφει το σύνολο  $\{x | x \in \mathbb{R}^2, |s_i - x| \leq r\}$ .

Επίσης ορίζουμε τη συνάρτηση δαχτυλιδιού, εκείνη που εξαρτάται από ένα σημείο και δύο ακτίνες ως

$$RING(s, r_{small}, r_{big}) := DISK(s, r_{big}) \setminus DISK(s, r_{small})$$

Στις υλοποιήσεις μας θα χρησιμοποιηθεί η προσέγγιση αυτών με πολύγωνα αρκετών πλευρών αντί για συμβολικούς υπολογισμούς.

**Πολύγωνο Ορατότητας:** Το πολύγωνο ορατότητας (Visibility Polygon ή VP) ενός κόμβου είναι το σύνολο από όλα τα σημεία με τα οποία υπάρχει *ορατότητα γραμμής (line of sight)* με τον κόμβο [4]. Ένα παράδειγμα VP δίνεται στο σχήμα 8 για έναν κόμβο. Σαν μαθηματικό σύνολο θα συμβολίζεται ως  $\mathcal{V}_i$  για τον  $i$  κόμβο, όπου υπάρχει σιωπηρή εξάρτηση από τα εμπόδια στο χώρο. Ένας βέλτιστος αλγόριθμος  $O(n)$  έχει δοθεί στη βιβλιογραφία [16, 13] καθώς και αποδοτικές υλοποιήσεις παρέχονται από τη βιβλιοθήκη CGAL [4] για τον υπολογισμό του. Σημειώνεται επίσης ότι για τον υπολογισμό του VP θεωρούμε το χώρο  $\mathcal{F}^c$  ως ένα εμπόδιο, και άρα ο κόμβος  $i$  δεν παρατηρεί το χώρο αυτό.

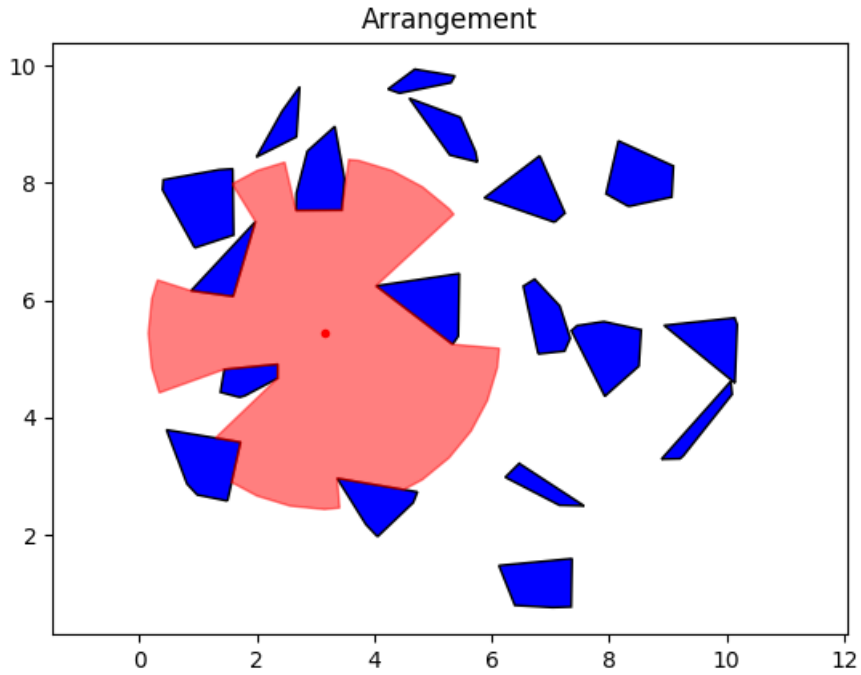


Σχήμα 8: Παράδειγμα πολύγωνου ορατότητας (Visibility Polygon) ενός κόμβου με πολλά εμπόδια - Πηγή: Claudio Rocchini/Wikipedia

**Πεδίο Ορατότητας:** Το πεδίο ορατότητας (FOV) είναι μια ειδική περίπτωση του πολύγωνου ορατότητας, όπου υπάρχει ένα νοητό εμπόδιο το οποίο εμποδίζει την ορατότητα μας, μετά από την μέγιστη ακτίνα που μπορούμε να επεξεργαστούμε. Ένα παράδειγμα φαίνεται στο σχήμα 9 με μια απλή διάταξη με έναν κόμβο. Βλέπουμε λοιπόν πως ο συγκεκριμένος κόμβος δεν μπορεί να καλύψει το χώρο πίσω από τα εμπόδια, και πέρα από

την μέγιστη κάλυψη του. Μαθηματικά, για κοινή μέγιστη ακτίνα των κόμβων  $R_{max}$ , το FOV ορίζεται ως:

$$FOV_i := \mathcal{V}_i \cap DISK(s_i, R_{max})$$



Σχήμα 9: Παράδειγμα πεδίου ορατότητας  $FOV_i$  ενός κόμβου με πολλά εμπόδια.

**Τρέχον Πεδίο Ορατότητας:** Το Current Field of Vision (current-FOV) εκφράζει την κάλυψη του κόμβου  $i$  μια χρονική στιγμή  $t$  στην οποία έχει ακτίνα κάλυψης  $R_i = R_i(t)$ . Το current-FOV προφανώς είναι μεταβλητό αφού εξαρτάται από την τιμή της ακτίνας που μπορεί να αλλάζει ως συνάρτηση του χρόνου:

$$CFOV_i := FOV_i \cap DISK(s_i, R_i)$$

όπου όταν θέλουμε να συγκρίνουμε τα  $CFOV_i$  για διαφορετικές αναθέσεις  $\mathbf{R}_1, \mathbf{R}_2$  θα γράφουμε  $CFOV_i(\mathbf{R}_1), CFOV_i(\mathbf{R}_2)$

**Μειωμένο Πεδίο Ορατότητας:** Το Μειωμένο Πεδίο Ορατότητας, (Reduced Field of Vision) το οποίο θα συμβολίζεται και ως reduced-FOV είναι το FOV του αισθητήρα αν αφαιρεθεί από αυτό το τρέχον FOV των γειτονικών αισθητήρων, και ορίζεται ως:

$$\mathcal{RFOV}_i := \mathcal{FOV}_i \setminus \bigcup_{j \neq i} \mathcal{CFOV}_j$$

Το reduced-FOV έχει εξέλιξη στο χρόνο που επάγεται απο την εξέλιξη των ακτίνων των γειτονικών κόμβων. Παράδειγμα του reduced-FOV φαίνεται στις ζυγές στήλες στο σχήμα 13 σε σύγκριση με το απλό FOV στις μονές στήλες.

**Συνολική Τρέχουσα Κάλυψη:** Η Συνολική κάλυψη (total-coverage) ορίζεται ως η ένωση όλων των καλύψεων.

$$\mathcal{TCOV} := \bigcup_i \mathcal{CFOV}_i \quad (14)$$

ενώ το  $\mathcal{MCOV} := \bigcup_i \mathcal{FOV}_i$  εκφράζει τη μέγιστη κάλυψη. Έτσι μπορούμε αργότερα να θέσουμε ως μετρική κάλυψης το λόγο των εμβαδών των δύο. Η σύγκριση των δύο για μια ανάθεση φαίνεται στο σχήμα 10.

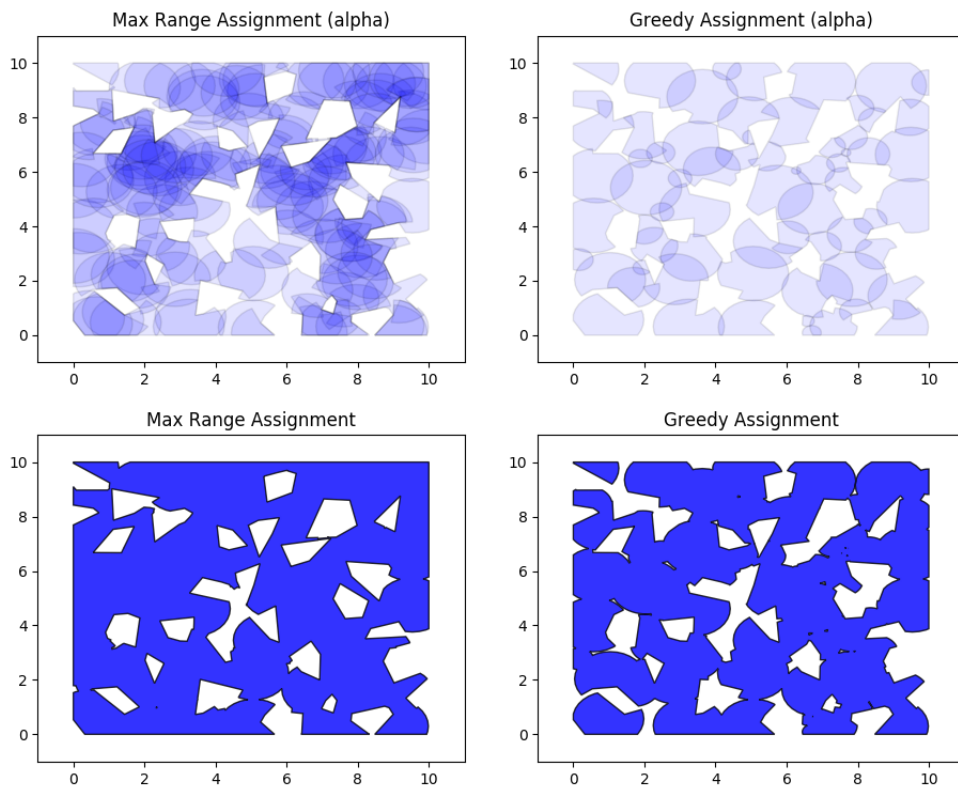
**Καλυμμένη Περιφέρεια:** Πολλές φορές είναι χρήσιμο να γνωρίζουμε το τμήμα της περιφέρειας ενός κόμβου,  $\partial(\mathcal{CFOV}_i)$ , που είναι καλυμένο απο άλλους αισθητήρες, εμπόδια, ή τα όρια της περιοχής επικάλυψης. Έτσι ορίζουμε καλυμμένη περιφέρεια ως το σύνολο  $\mathcal{CCIRC}_i$  ως

$$\mathcal{CCIRC}_i = \partial(\mathcal{CFOV}_i) \cap \left( \left( \bigcup_{j \neq i} \mathcal{CFOV}_j \right) \cup \left( \bigcup_k \mathcal{O}_k \right) \cup \mathcal{F}^c \right) \quad (15)$$

**Ποσοστό Κάλυψης Περιφέρειας:** Ορίζουμε το ποσοστό κάλυψης της περιφέρειας ενός αισθητήρα ως το ποσοστό του μήκους το οποίο τέμνεται είτε με κάποιον άλλο αισθητήρα, είτε με κάποιο εμπόδιο, ή με τα όρια της περιοχής ενδιαφέροντος  $\mathcal{F}$  της συνολικής περιφέρειας.

$$\mathcal{PCOV}_i := \frac{\text{length}(\mathcal{CCIRC}_i)}{\text{length}(\partial(\mathcal{CFOV}_i))} \quad (16)$$

όπου υποθέσαμε το  $\text{length}$  ως μια πράξη που βρίσκει το μήκος μιας καμπύλης.

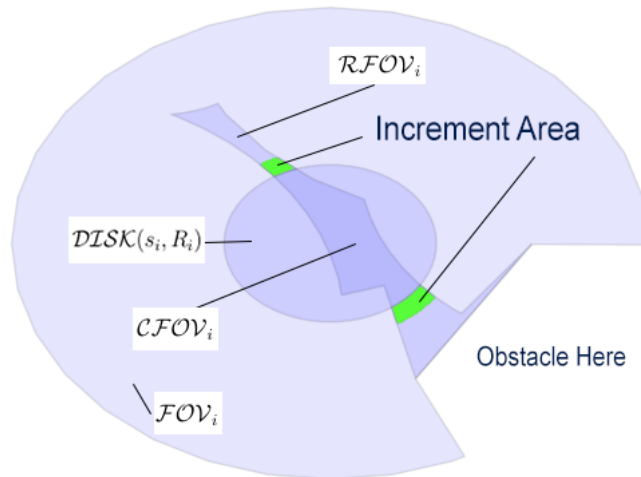


Σχήμα 10: Παράδειγμα του ΜCΟΝ αριστερά πάνω και κάτω, και του ΤCΟΝ δεξιά πάνω και κάτω. Στις πάνω εικόνες φαίνεται με κάθε συνιστώσα της ένωσης με μια διαφάνεια ώστε να δίνεται έμφαση στις μεταξύ τους επικαλύψεις (overlaps), κάτι ανεπιθύμητο. Στις κάτω εικόνες φαίνεται τα τελικά σύνολα.

## 5.2. Άπληστος Κεντριοποιημένος Αλγόριθμος

Αρχικά θα παρουσιάσουμε τον πρώτο Άπληστο Κεντριοποιημένο Αλγόριθμο (Greedy Centralized Algorithm) που υλοποιήσαμε ο οποίος βασίζεται στην κεντρική άπληστη αρχή της επιλογής της καλύτερης φαινομενικά επιλογής. Για την υλοποίηση αυτού του αλγόριθμου υποθέτουμε ότι υπάρχει όλη η διαθέσιμη πληροφορία σχετικά με το δίκτυο (συντεταγμένες εμποδίων, αισθητήρων κλπ) και ταυτόχρονα ότι μπορεί κάποια πληροφορία να μεταβιβαστεί από κόμβο σε κόμβο ανά πάσα στιγμή. Έτσι το πρόβλημα είναι απλά η εύρεση των κατάλληλων ακτίνων. Σε ένα πραγματικό δίκτυο αυτό θα μπορούσε να αντιστοιχεί σε ένα σενάριο που ένας κεντρικός κόμβος μπορεί να κάνει τους αναγκαίους υπολογισμούς και να αναθέσει τις κατάλληλες ακτίνες στους κόμβους.





Σχήμα 11: Στο σχήμα αυτό παρουσιάζονται σχηματικά οι έννοιες που ορίστηκαν και αξιοποιούνται από τις μεθόδους μας. Επιπλέον φαίνεται ως Increment Area το επιπλέον χρήσιμο εμβαδό για μικρή αύξηση της ακτίνας  $\Delta r$ . Ως Obstacle Here σημειώνεται ο χώρος ο οποίος καταλαμβάνεται από εμπόδιο, το οποίο όμως δεν παρουσιάζεται επιπλέον στο σχήμα.

### 5.2.1. Παρουσίαση και Εξήγηση Πρώτου Αλγόριθμου

Προηγουμένως δώσαμε μια γενικότερη μαθηματική περιγραφή των εννοιών που θα αξιοποιηθούν στη συνέχεια. Στην πραγματική υλοποίηση δεν θα υπάρχουν συνεχείς καμπύλες παρα μόνον πολυγωνικές προσεγγίσεις - έτσι ακόμα και εάν δεν είναι ρητό μέσα από τις μαθηματικούς ορισμούς, χρησιμοποιούμε μια κατάλληλη προσέγγιση σε κάθε περίπτωση. Επιπλέον θα υποθέσουμε τις παρακάτω συναρτήσεις:

- **area**: Λαμβάνει ως είσοδο ένα σύνολο (στην πράξη σύνολο από πολυγωνικές περιοχές) και υπολογίζει το εμβαδόν του.
- **CoverageFraction**: Αποτελεί συνάρτηση που δέχεται τις τρέχουσες ακτίνες  $\mathbf{R}$  (και σιωπηρά τα εμπόδια και τις συντεταγμένες) και υπολογίζει τον λόγο  $\frac{Coverage(\mathbf{R})}{Coverage(\mathbf{R}_{max})}$
- **argmax, argmin**: Υπολογίζει το σύνολο των τιμών που μεγιστοποιούν (ή ελαχιστοποιούν) την παράσταση του argmax (ή argmin).
- **select**: Δέχεται ένα σύνολο  $S$  και έναν ακέραιο  $k$  και επιστρέφει με ίση πιθανότητα  $k$  στοιχεία από το σύνολο αυτό. Για παράδειγμα θα μπορούσε σε μια εκτέλεση να έχει αποτέλεσμα:  $select(\{1, \dots, 100\}, 3) = \{19, 38, 65\}$ .

Ο πρώτος αλγόριθμος δέχεται ως είσοδο τα εμπόδια, τα σημεία των αισθητήρων, το όριο κάλυψης που θέλουμε να πετύχουμε, και επιστρέφει την ανάθεση  $\mathbf{R}$ .

---

## Αλγόριθμος-Συνάρτηση 2: Κεντριοποιημένος Greedy Αλγόριθμος

---

```
1 function GreedyAssignment ( $\mathbf{O}, \mathbf{P}, Threshold$ )
2  $\mathbf{R} = \mathbf{0}$ 
3 while CoverageFraction( $\mathbf{R}$ ) < Threshold do
4    $S = \arg \max_{i \in \{1, \dots, N\}} \{incrementRatio(i)\}$ 
5    $i = select(S, 1)$ 
6    $R_i = \min \{R_i + \Delta r, R_{max}\}$ 
7   if BernulliTrial( $p$ ) is succesful then
8      $S = \arg \min_{i \in \{1, \dots, N\}} \{decrementRatio(i)\}$ 
9      $i = select(S, 1)$ 
10     $R_i = \max \{R_i - \Delta r, 0\}$ 
11  end
12 end
13 return  $\mathbf{R}$ 
```

---

Στη γραμμή 2 αρχικοποιούμε τους κόμβους να έχουν ακτίνα κάλυψης 0. Στη συνέχεια, γραμμές 3... 12, μέσα στον βρόχο επανάληψης, όσο δεν έχουμε καλύψει το *Threshold* που θέλουμε (γραμμή 3), αυξάνουμε κάποια ακτίνα (γραμμή 6) ώστε σιγά σιγά να το φτάσουμε. Αυτός ο τρόπος αντιμετώπισης είναι ανάλογος των επαναληπτικών αριθμητικών μεθόδων, και θυμίζει Stochastic Gradient Descent. Στη γραμμή 4 υπολογίζεται ο δείκτης του κόμβου του οποίου αν αυξήσουμε την ακτίνα κατά  $\Delta r$  θα έχει ως αποτέλεσμα τον καλύτερο λόγο  $\frac{\Delta(Area)}{\Delta(Energy)}$ , δηλαδή τον λόγο ΝΕΟΥ καλυπτόμενου Εμβαδου  $\Delta(Area)$  (που δεν καλυπτόταν από κάποιον άλλο αισθητήρα ναωρίτερα) δια της πρόσθετης ενέργειας  $\Delta(Energy)$  που θα καταναλώσουμε, δηλαδή το επιπλέον κόστος που απαιτείται για την αύξηση της ακτίνας κάλυψης. Η μέθοδος *incrementRatio* (η οποία δίνεται αναλυτικά στη συνέχεια) φροντίζει για τον υπολογισμό αυτού του λόγου για έναν κόμβο, και με τη χρήση του *argmax* λαμβάνουμε το δείκτη για την καλύτερη επιλογή από όλους τους κόμβους.

Το πρόβλημα είναι πως αν μονάχα αυξήσουμε τις ακτίνες των κόμβων, κάποιοι κόμβοι που ξεκίνησαν με καλό λόγο και στην πορεία δεν έχουν προοπτική να αυξηθούν ξανά, ίσως καλυφθούν ολοκληρωτικά, από γειτονικούς κόμβους, κάνοντας την λειτουργία τους περιττή. Πιο συχνά, θα καταλήγουμε με κόμβους που ο λόγος τους άλλαξε δραματικά από την εξέλιξη των αναθέσεων των γειτονικών κόμβων. Για αυτό είναι χρήσιμο πέρα από τις αυξήσεις, να έχουμε βήματα μείωσης επιλεγμένων ακτίνων. Σε αυτήν την περίπτωση όμως, από όλους τους κόμβους θα θέλαμε να μειώσουμε την ακτίνα του κόμβου με τον μικρότερο λόγο  $\frac{\Delta(Area)}{\Delta(Energy)}$ , διότι τώρα το  $\Delta(Area)$  εκφράζει το εμβαδόν που δεν θα

καλύπτουμε πια, λόγω του ότι μειώνουμε την ακτίνα, ενώ το  $\Delta(Energy)$  εκφράζει την ενέργεια που εξοικονομούμε.

Αυτό γίνεται στις γραμμές 7... 11, όπου με μια πιθανότητα  $p < 1.0$  επιτελούμε την καλύτερη δυνατή μείωση ακτίνας. Όσο μεγαλύτερη είναι η πιθανότητα αυτή τόσο περισσότερο χρόνο χρειάζεται για να ολοκληρωθεί ο αλγόριθμος. Επιπλέον όσο δεν συμβαίνει η αλλαγή του λόγου του χρήσιμου εμβαδού διά την καταναλισκόμενη ενέργεια σε κάποιους κόμβους, όπως περιγράφηκε νωρίτερα, τότε απλώς θα ακυρώνουμε το τελευταίο βήμα. Με το να πηγαίνουμε στον πιθανοτικό κόσμο για την μείωση της ακτίνας κυρίως απλοποιούμε την υλοποίηση.

---

### Αλγόριθμος-Συνάρτηση 3: Υπολογισμός incrementRatio

---

```

1 function incrementRatio (i)
2 newPotentialRadius = min { $R_i + \Delta r$ ,  $R_{max}$ }
3 ringPoly =  $\mathcal{RLNG}(s_i, R_i, newPotentialRadius)$ 
4 reducedRingPoly =  $ringPoly \cap \mathcal{RFOV}_i$ 
5 energyDiff =  $f^\uparrow(newPotentialRadius) - f^\uparrow(R_i)$ 
6 if energyDiff == 0 then
7 | return 0
8 end
9 else
10 | return  $area(reducedRingPoly)/energyDiff$ 
11 end

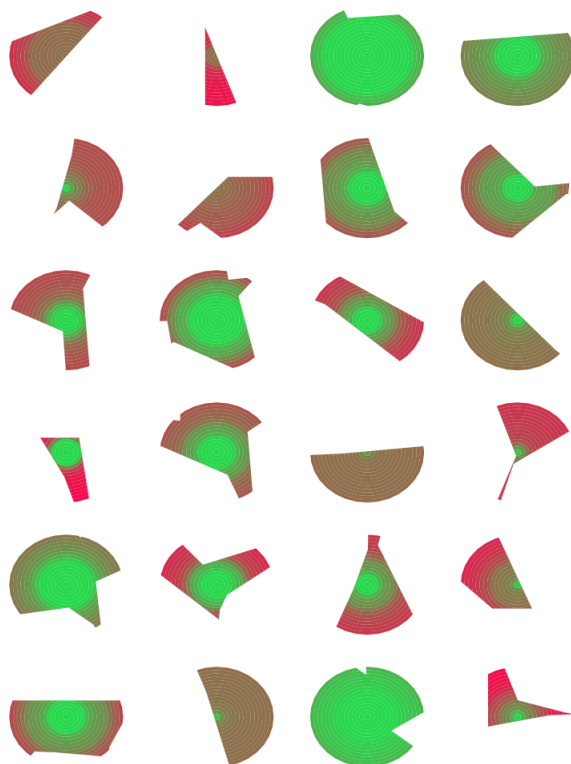
```

---

Δίνεται η υλοποίηση του *incrementRatio*. Στις γραμμές 2-3 υπολογίζουμε ένα δαχτυλίδι που εκτείνεται απο την παλιά ακτίνα κάλυψης εως την καινούρια/μεγαλύτερη ακτίνα κάλυψης. Στη γραμμή 4 βρίσκουμε την τομή του δαχτυλιδιού με το reduced-FOV αφού αυτό έχει ως αποτέλεσμα τον ουσιώδες χώρο που θα καλύψουμε με την αλλαγή της ακτίνας, δηλαδή το χώρο που περιγράφεται απο το reducedRingPoly. Στη γραμμή 5 βρίσκουμε την αύξηση της κατανάλωσης απο το να μεταβούμε στην νέα πιθανή ακτίνα. Εάν αυτή είναι 0 αυτό απλά σημαίνει ότι ο κόμβος είναι ήδη στην μέγιστη ακτίνα του και  $newPotentialRadius == R_i$ , οπότε επιστρέφουμε 0 σε αυτήν την περίπτωση. Διαφορετικά επιστρέφουμε το εμβαδόν του χρήσιμου χώρου δια την πρόσθετη ενέργεια.

Στο σχήμα 12 οπτικοποιούνται οι λόγοι  $\frac{\Delta(Area)}{\Delta(Energy)}$  σε κάποια δαχτυλίδια όμως θεωρώντας μονάχα το FOV, δηλαδή μια a priori είκονα για το πόσο καλός είναι ο λόγος σε αυτά. Στο σχήμα 13 παρουσιάζεται η οπτικοποίηση των λόγων σε δαχτυλίδια για το FOV και το reduced-FOV το οποίο εν τέλει και χρησιμοποιείται. Αυτό που είναι ορατό στις εικόνες αυτές είναι το γεγονός ότι η κάλυψη των άλλων κόμβων μπορεί να κάνει ένα

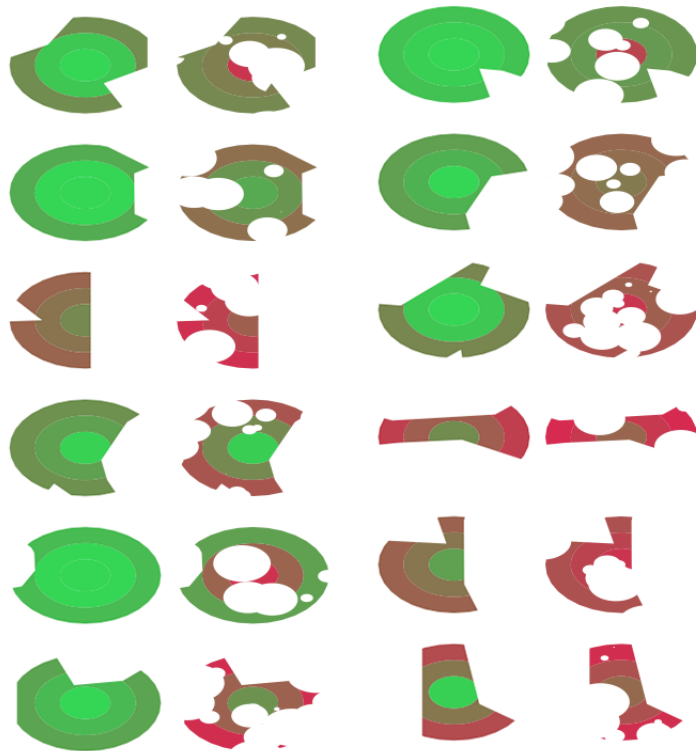
δαχτυλίδι a posteriori ασύμφορο ενώ αρχικά είχε καλό λόγο (απο πράσινο σε κόκκινο).



Σχήμα 12: Ένα δέγμα απο 24 κόμβους, με το FOV τους, διαμερισμένο σε 15 δαχτυλίδια ίδιου πάχους  $\Delta r$ , όπου η συνιστώσα πράσινου στο RGB χρωματισμό, αυξάνεται (ενώ παράλληλα μειώνεται η συνιστώσα του κόκκινου) για μικρότερες τιμές του λόγου  $\frac{Area}{Energy}$  στο δαχτυλίδι.

### 5.2.2. Προβληματικές Περιπτώσεις Πρώτου Αλγόριθμου

Το κριτήριο που χρησιμοποιείται στον αλγόριθμο αυτό αποφασίζει με αρκετά τοπικό τρόπο, βλέποντας τι συμβαίνει στο αμέσως γειτονικό  $\Delta r$  κάθε αισθητήρα, το οποίο έχει και ως αποτέλεσμα ανεπαρκή αντιμετώπιση σε ορισμένες περιπτώσεις. Η μια απο αυτές τις περιπτώσεις φαίνεται στο σχήμα 14. Σε αυτήν την περίπτωση έχουμε τον κάτω αισθητήρα να έχει ανάπτυξη της ακτίνας του πριν απο τον πάνω αισθητήρα, για διάφορους λόγους (τυχαία επιλογή, ύπαρξη ασήμαντου εμπόδιου). Αργότερα όμως ο πάνω αισθητήρας έχει προτιμότερες βελτιώσεις οπότε αρχίζει και μεγαλώνει σημαντικά, καλύπτοντας ένα σημαντικό κομμάτι του χώρου που κάλυπτε ο κάτω αισθητήρας. Θα θέλαμε λοιπόν ο αλγόριθμος να μπορεί και να αποφασίζει ότι ο κάτω αισθητήρας δεν είναι πια χρήσιμος. Ο αλγόριθμος ως κριτήριο για μείωση χρησιμοποιεί όμως το διαφορικό



Σχήμα 13: Παράδειγμα FOV 8 κόμβων χρωματισμένων όπως και στο προηγούμενο σχήμα, διαμερισμένων σε 4 δαχτυλίδια, με την διαφορά ότι τώρα συγκρίνονται με τα σχήματα που προκύπτουν αφαιρώντας τις καλύψεις των γειτονικών κόμβων.

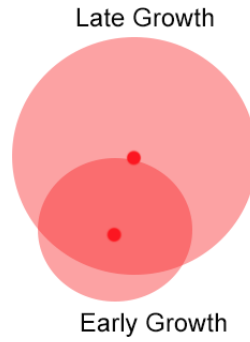
του  $\frac{Area}{Energy}$  για  $\Delta r$  διαφοροποιήσεις της ακτίνας, και λόγω του ότι ο κάτω αισθητήρας έχει το μοναδικό προσφερόμενο εμβαδόν γύρω από την ακτίνα του, ο αλγόριθμος μπορεί να κρίνει ότι δεν δικαιολογείται η μείωση αυτή, αγνοώντας τη συνολικά κακή εικόνα του αισθητήρα.

### 5.2.3. Θεμελίωση της Επιλογής

Ο Αλγόριθμος μας επιλέγει να αυξήσει κάθε φορά τον κόμβο με τον καλύτερο λόγο. Μπορεί διαπιστωτικά αυτή η επιλογή να φαίνεται η καλύτερη δυνατή, αλλά θα θέλαμε να το επιβεβαιώσουμε.

**Λήμμα 5.1 :** Άπειρες αυξήσεις κόμβων με λόγο  $\frac{\Delta(Covered)}{\Delta(Energy)} = \lambda = \frac{x_1}{y_1}$ , δικτύου που αρχικά είχε  $\frac{Covered(\mathbf{R})}{Energy(\mathbf{R})} = \frac{x_0}{y_0}$  έχει ως αποτέλεσμα δίκτυο με λόγο  $\lambda$ .

**Απόδειξη:** Σε ένα βήμα έχουμε



Σχήμα 14: Προβληματική περίπτωση του κεντρικού αλγόριθμου

$$\frac{x_0}{y_0} \rightarrow \frac{x_0 + x_1}{y_0 + y_1} = \frac{x_0 + y_1 \lambda}{y_0 + y_1}$$

Στο βήμα  $n$  έχουμε

$$\frac{x_0}{y_0} \xrightarrow{n \text{ steps}} \frac{x_0 + n\lambda y_1}{y_0 + n y_1}$$

και για άπειρα βήματα έχουμε:

$$\lim_{n \rightarrow +\infty} \frac{\lambda n y_1 \left( \frac{x_0}{y_1 n \lambda} + 1 \right)}{n y_1 \left( \frac{y_0}{y_1 n} + 1 \right)} = \lim_{n \rightarrow +\infty} \frac{\lambda \left( \frac{x_0}{y_1 n \lambda} + 1 \right)}{\frac{y_0}{y_1 n} + 1} = \lambda \quad (17)$$

□

Έτσι, σύμφωνα με αυτό το Λήμμα, εάν είχαμε συστηματικά επιλογές λόγου  $\lambda_1$  και επιλογές λόγου  $\lambda_2$ , με  $\lambda_1 > \lambda_2$ , τότε θα ήταν προτιμότερο να επιλέγουμε τις επιλογές του καλύτερου λόγου.

Αν έχουμε αρχικά λόγο  $\frac{x_0}{y_0}$ , και έχουμε δύο επιλογές αύξησης αισθητήρων με λόγους  $\frac{x_1}{y_1}, \frac{x_2}{y_2}$  όπου  $\frac{x_1}{y_1} > \frac{x_2}{y_2}$ , τότε εάν επιλέξουμε το δεύτερο αισθητήρα, μπορεί να καταλήξουμε σε καλύτερο τελικό λόγο. Συνεπώς δεν είναι τετριμμένα καλύτερη η επιλογή του καλύτερου λόγου για την ανά-ακτίνα διατήρηση του καλύτερου λόγου.

Έστω τώρα ότι οι πρόσθετες ενέργειες  $y_1$  και  $y_2$  των αισθητήρων είναι ίδιες  $y_1 = y_2$ . Τότε

$$\frac{x_0 + x_1}{y_0 + y_1} > \frac{x_0 + x_2}{y_0 + y_1} = \frac{x_0 + x_2}{y_0 + y_2}$$

Συνεπώς εάν δύο αισθητήρες έχουν ίδια αλλαγή ενέργειας, το οποίο συμβαίνει εάν έχουν ίδιο βήμα αύξησης και έχουν την ίδια ακτίνα, τότε πράγματι η καλύτερη επιλογή είναι ο αισθητήρας με τον καλύτερο λόγο Εμβαδού δια Ενέργειας.

Ας υποθέσουμε τώρα ότι υπάρχουν δύο αλγόριθμοι, ο ένας είναι ο δικός μας που επιλέγει πάντα τον καλύτερο λόγο εμβαδού και ενέργειας, και υπάρχει ένας δεύτερος που επιλέγει έναν άλλον αισθητήρα κάθε φορά. Εάν θεωρήσουμε ότι η κατανομή των ενεργειών που προστίθενται για τους δύο αλγόριθμους είναι ίδιες, δηλαδή ότι και οι δύο αλγόριθμοι κάνουν τις επιλογές τους απο κόμβους που έχουν στατιστικά τα ίδια επίπεδα ακτίνας πριν επιλεγούν, τότε έχουμε:

$$\frac{x_i^1}{y_i^1} \geq \frac{x_i^2}{y_i^2}$$

και

$$\sum y_i^1 = \sum y_i^2$$

Τότε

$$S_1 = \frac{x_0 + \sum x_i^1}{y_0 + \sum y_i^1} = \frac{x_0 + \sum x_i^1}{y_0 + \sum y_i^2} \geq \frac{x_0 + \sum x_i^2}{y_0 + \sum y_i^2} = S_2$$

Άρα  $S_1 \geq S_2$

Έτσι λοιπόν βλέπουμε ότι, υπό την υπόθεση ότι η επιπλέον ενέργεια των επιλογών έχει την ίδια κατανομή, για τους δύο αλγόριθμους, τότε μετά απο πολλές αυξήσεις, είναι προτιμότερη η επιλογή του κόμβου με τον καλύτερο λόγο, δηλαδή του άπληστου αλγόριθμου.

#### 5.2.4. Θέματα Αποδοτικότητας Υλοποίησης

Αρχικά πριν μιλήσουμε για αποδοτικότητα, ανατρέχοντας στον αλγόριθμο που παρουσιάστηκε λίγο νωρίτερα, βλέπουμε ότι οι λειτουργίες που είναι εν δυνάμει απαιτητικές, και θα πρέπει να λάβουμε υπόψιν, είναι οι συναρτήσεις *CoverageFraction*, *incrementRatio*, *decrementRatio*.

Η συνάρτηση *CoverageFraction*, όπως αναφέρθηκε αποτελεί ένα κλάσμα, του οποίου ο παρανομαστής είναι πάντα σταθερός, ενώ ο αριθμητής μεταβάλλεται και είναι ίσος με  $Coverage(\mathbf{R})$ , δηλαδή το εμβαδόν της τρέχουσας συνολικής κάλυψης *TCOV*.

Ο προγραμματιστής που θα υλοποιήσει τον αλγόριθμο μπορεί κάθε φορά που αλλάζει ένας αισθητήρας ακτίνα και προστίθεται χρήσιμο εμβαδόν ή αφαιρείται χρήσιμο εμβαδόν, να φροντίζει να ανανεώνει κατα την απαιτούμενη διαφορά μια μεταβλητή που εκφράζει την εικόνα του συστήματος για το  $Coverage(\mathbf{R})$  - ειδική μέριμνα όμως πρέπει να ληφθεί για σφάλματα αριθμών κινητής υποδιαστολής αφού με αυτόν τον τρόπο θα προστίθονται πολλές φορές μικροί αριθμοί σε ένα συγκριτικά μεγαλύτερο άθροισμα, οπότε υπάρχει κίνδυνος τα γνωστά σφάλματα του πρότυπου floating-point αριθμών να αλλοιώσουν σημαντικά το αποτέλεσμα.

Ειδικότερα θα θέλαμε να δούμε εάν μπορούμε να υπολογίσουμε η κάλυψη στην εξέλιξη του χρόνου με κάποιον πιο κομψό τρόπο που βασίζεται στις ιδιότητες των πράξεων των συνόλων. Η τρέχουσα συνολική κάλυψη αποτελείται απο την ένωση των  $CFON_i$ , κάθε ένα απο τα οποία είναι σταθερό εφόσον δεν αλλάζει η ακτίνα του αντίστοιχου αισθητήρα. Έστω ότι σε ένα στάδιο διαθέτουμε αποθηκευμένη την ένωση  $TCOV$ . Εάν κάποια ακτίνα ενός αισθητήρα  $i$  αυξηθεί τότε το νέο  $TCOV'$  θα μπορεί να υπολογιστεί σαν  $TCOV' = TCOV \cup CFON'_i$  αφού  $CFON_i \subseteq CFON'_i$ . Έτσι έχουμε ανανεώσει την τιμή του  $TCOV$  στην νέα τονούμενη τιμή. Τώρα ας υποθέσουμε ότι έχουμε κάποια μείωση ακτίνας του αισθητήρα  $i$ : σε αυτήν την περίπτωση δεν είναι εφικτό να επαναλάβουμε το ίδιο τέχνασμα, όμως μπορούμε να κάνουμε το εξής:

$$TCOV' = TCOV \setminus (CFON_i \setminus CFON'_i) \cup \bigcup_{j \in Nei(i)} CFON_j$$

Οπότε οι συνολοπράξεις αντί για  $n$ , όσες το πλήθος των αισθητήρων, γίνονται ίσες με  $S+3$  για μειώσεις και 1 για αυξήσεις, όπου  $S$  ο μέσος βαθμός του κάθε κόμβου. Άρα έχουμε σε κάθε επανάληψη, στον προηγούμενο αλγόριθμο, κατα μέσο όρο  $1+p(S+3)$  συνολοπράξεις, όπου  $p$  η πιθανότητα μείωσης της ακτίνας όπως την είδαμε νωρίτερα. Σημειώνεται ότι πάντα γίνεται αύξηση οπότε η πιθανότητα της αύξησης είναι 1.

Έπειτα στη συνάρτηση *incrementRatio* η λειτουργία που είναι πιο απαιτητική και πρέπει να εξετάσουμε είναι ο υπολογισμός του  $RFON_i$ . Πράγματι αυτό υπολογίζεται ως η διαφορά απο το  $FOV_i$  της ένωσης των υπόλοιπων κόμβων, αλλα πρακτικότερα αρκεί να πάρουμε την ένωση μόνο των γειτόνων, δηλαδή  $S+1$  συνολοπράξεις απαιτούνται για ένα  $RFON_i$ , και αυτό μεταβάλλεται όταν αλλάζει η ακτίνα ενός γείτονα, δηλαδή  $S$ -φορές πιο συχνά απο το να αλλάξει ένας αισθητήρας.

Ένα πρόβλημα του αλγόριθμου αυτού είναι ότι δεν είναι πολύ αποδοτικός αφού για κάθε αύξηση πρέπει να βρούμε όλους τους αισθητήρες που έχουν τον βέλτιστο λόγο, και να αυξήσουμε εκείνον που έχει τον καλύτερο λόγο.

Σε κάθε εύρεση του βέλτιστου, γίνονται περίπου  $n \cdot S$  ( $n$  το πλήθος των κόμβων) συνολοπράξεις. Αυτό πρέπει να συμβεί αρκετές φορές, έστω  $N$  φορές, πριν φτάσουμε



σε μια ικανοποιητική κάλυψη (που ξεπερνάει το όριο  $T$ ). Άρα συνολικά έχουμε  $n \cdot S \cdot N$  ενώσεις. Για να μειώσουμε το κόστος θα θέλαμε να μειώσουμε αρχικά το πλήθος των ενώσεων. Η πρώτη βελτιστοποίηση που μπορεί να κάνει κανείς είναι να μην υπολογίζει πολλές φορές την ένωση του καλυπτόμενου χώρου απο τους γείτονες ενός κόμβου όταν αυτοί δεν έχουν αλλάξει. Δηλαδή μια κρυφή μνήμη (cache), η οποία διατηρεί μια έγκυρη ένωση των γειτόνων, και να τη χρησιμοποιεί εφόσον δεν έχει αλλάξει κάποιος απο τους γείτονες. Όταν ένας απο τους γείτονες αλλάζει, τότε μη γνωρίζοντας ακριβώς τι αλλαγές έχουν γίνει στην ένωση, θα πρέπει να υπολογίσουμε ξανά την ένωση των γειτόνων. Αυτό θα έχει ως συνέπεια  $\frac{n \cdot S \cdot N \cdot S}{n} = S^2 N$  υπολογισμούς ένωσης (δηλαδή μόνο ένα ποσοστό  $\frac{S}{n}$  του προηγούμενου αποτελέσματος χρειάζεται να υπολογιστεί ξανά - το οποίο αντιστοιχεί στην πιθανότητα ένας γειτονικός κόμβος να έχει τον καλύτερο λόγο και συνεπώς να μεταβληθεί η ακτίνα του).

### 5.3. Τυχαιοποιημένος Κεντρικοποιημένος Αλγόριθμος

Μια παραλλαγή η οποία μπορεί να θεωρηθεί γενίκευση του προηγούμενου αλγόριθμου είναι η εξής: αντί να βρίσκουμε το μέγιστο (ή ελάχιστο) απο όλους τους κόμβους πριν αυξήσουμε (ή μειώσουμε) κάποιον, μπορούμε απλά να βρίσκουμε το αντίστοιχο μέγιστο (ή ελάχιστο) σε ένα τυχαίο υποσύνολο των κόμβων σταθερής πληθικότητας  $n$ . Θα αποκαλούμε τη μέθοδο αυτή **CHECK-n-R** προσέγγιση. Στην ακραία περίπτωση όπου  $n = N$  οι δύο αλγόριθμοι ταυτίζονται λειτουργικά (αφού τυχαίο υποσύνολο με όλα τα στοιχεία του συνόλου είναι το ίδιο με το αρχικό σύνολο). Το σκεπτικό αυτής της προσέγγισης είναι το εξής: το πλήθος των κόμβων που ελέγχουμε κάθε φορά είναι σταθερό και ίσο με  $n$  επιτρέποντας στον αλγόριθμο να έχει ικανοποιητική κλιμάκωση (scaling). Παράλληλα μέσα σε ένα μικρότερο σύνολο  $n$  κόμβων εικάζεται πως θα υπάρχει ένας ικανοποιητικός κόμβος να αυξήσει την ακτίνα του - φυσικά κάτι τέτοιο εξαρτάται απο την τρέχουσα ανάθεση και αναμένει κανείς όσο πλησιάζουμε σε μεγάλη κάλυψη, τόσο μικρότερη να είναι η πιθανότητα να βρίσκουμε καλές αυξήσεις σε μικρά υποσύνολα.

Ένα παράδειγμα στην αξία μιας τέτοιας παραλλαγής είναι αν έχουμε ένα δίκτυο με  $N_1 = 10^6$  αισθητήρες και ένα δίκτυο με  $N_2 = 10^9$ , σε χώρους κατάλληλους ώστε η πυκνότητα των αισθητήρων να είναι ίδια στα δύο δίκτυα. Τότε αν χρησιμοποιήσουμε μια **CHECK-100-R** και στα δύο δίκτυα, το πλήθος των αισθητήρων που πρέπει να ελέγξουμε πριν αυξήσουμε κάποιον μένει ίδιο, και τα αποτελέσματα αναμένουμε να μην αποκλίνουν σημαντικά.

## 5.4. Κατανεμημένος Τοπικός Αλγόριθμος

Το βασικό ζήτημα του προηγούμενου αλγόριθμου είναι ότι η απόφαση για την αλλαγή ακτίνας πρέπει να λαμβάνεται κεντρικά, και μάλιστα σε πολλά βήματα. Θα ήταν προτιμότερο αν μπορούσε κάθε κόμβος, κατόπιν ενδεχομένως συνεννόησης με τους γειτονικούς του κόμβους, να αυξάνει ή να μειώνει την ακτίνα του ώστε μέσω τοπικών αλλαγών να βελτιστοποιείται συνολικά. Έτσι οδηγηθήκαμε σε μια παραλλαγή μέσα από έναν κατανεμημένο, με την έννοια ότι οι αποφάσεις λαμβάνονται στους κόμβους, και τοπικό, με την έννοια ότι χρησιμοποιείται πληροφορία μόνο από γειτονικούς κόμβους, αλγόριθμο.

### 5.4.1. Επαγόμενος Γράφος Εξάρτησης

Το πρώτο πρόβλημα είναι πως θα αυξάνει ένας κόμβος την ακτίνα του σε μια τοπική εκδοχή. Η ιδέα είναι η συνεννόηση για την αύξηση να γίνεται μεταξύ κόμβων που έχουν επικαλυπτόμενα FOVs αφού αυτοί οι κόμβοι επηρεάζουν τη συνεργατική κάλυψη του χώρου που αφορά έναν κόμβο. Έτσι δημιουργείται ένας γράφος με τους κόμβους των οποίων τα FOV έχουν τομή να συνδέονται με ακμή - δηλαδή μιλάμε για έναν γράφο τομής, όπου τα σύνολα αντιστοιχούν στα FOVs. Στην περίπτωση που δεν υπάρχει κανένα εμπόδιο, αυτός ο γράφος είναι ένας Random Geometric Graph (RGG) με παράμετρο  $r$  διπλάσια της μέγιστης ακτίνας αίσθησης  $R_{max}$  των κόμβων, αφού εμείς θέλουμε απλά να υπάρχει τομή των δύο δίσκων και όχι να περιέχουν τον γειτονικό κόμβο. Στην περίπτωση που μας ενδιαφέρει όμως, με την ύπαρξη εμποδίων, έχουμε να κάνουμε με έναν υπογράφο του RGG με μια κατανομή πυκνότητας πιθανότητας για την τοποθέτηση των κόμβων ίση με 0 στα σημεία που υπάρχει εμπόδιο.

Ορίζουμε τον γράφο ως κορυφές να είναι το ήδη ορισμένο σύνολο  $\mathbf{P}$ , και το σύνολο των ακμών  $\mathcal{E}$  όπου:

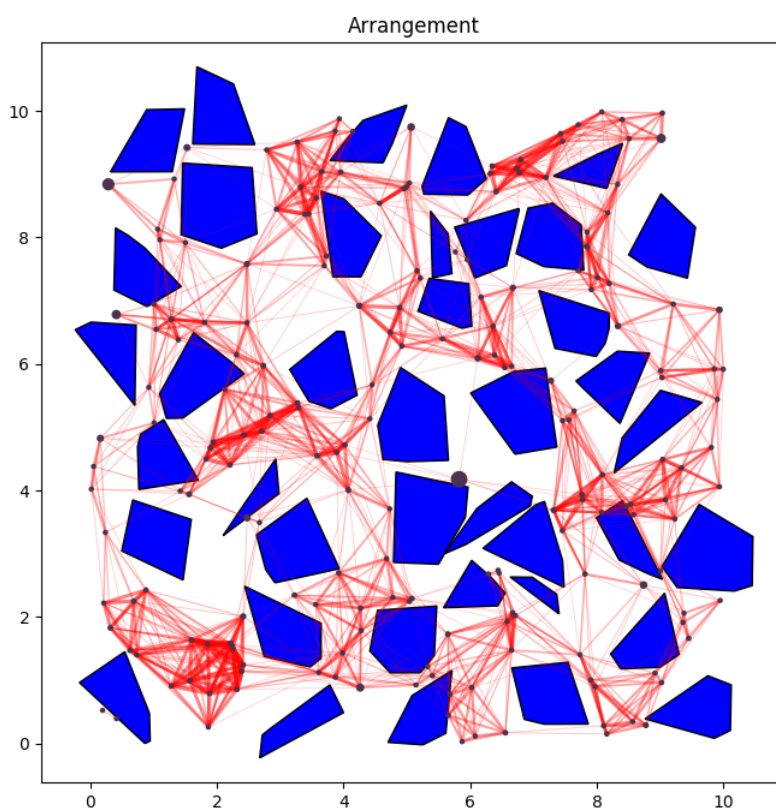
$$\mathcal{G} = (\mathbf{P}, \mathcal{E})$$

όπου  $(s_i, s_j) \in \mathcal{E} \Leftrightarrow (FOV_i \cap FOV_j) \neq \emptyset$ .

Ο γράφος αυτός οπτικοποιείται στο Σχήμα 5, όπου το πάχος κάθε ακμής καθώς και η διαφάνεια της (opacity) προκύπτει από το πόσο σημαντική είναι η τομή των δύο FOVs, δηλαδή το μέτρο/εμβαδό (measure) της τομής των FOV-πολυγώνων. Μεγαλύτερη τομή συνεπάγεται μεγαλύτερη εξάρτηση μεταξύ των δύο κόμβων - ή διαφορετικά έχει μεγάλη σημασία το που θα έχουμε θέσει την ακτίνα του κάθε κόμβου για το τελικό assignment. Λόγου χάρη φανταστείτε δύο κόμβους που έχουν συμπέσει στο ίδιο σημείο - σε αυτήν την περίπτωση η τομή των δύο FOV είναι μέγιστη, και ποσοστιαία 100% του συνολικού εμβαδού κάθε FOV, το οποίο στην πράξη μεταφράζεται στο ότι είναι περιττό να είναι παραπάνω από ένας κόμβος ενεργός ταυτόχρονα. Αυτή η μεγαλύτερη ή μικρότερη εξάρτηση μεταξύ των κόμβων, θα μπορούσε να μεταφραστεί σε έναν γράφο με βάρη για

αλγόριθμους που χρησιμοποιούν την πληροφορία αυτή για επιπλέον βελτιστοποιήσεις. Επιπλέον στην οπτικοποίηση δείχνονται με μεγαλύτερου μεγέθους κουκκίδες, οι κόμβοι που καλύπτουν περισσότερο χώρο που δεν καλύπτεται από άλλους κόμβους, ως ένα μέτρο της σημαντικότητας των κόμβων αυτών. Αυτό προκύπτει ως μια κανονικοποίηση της ποσότητας:

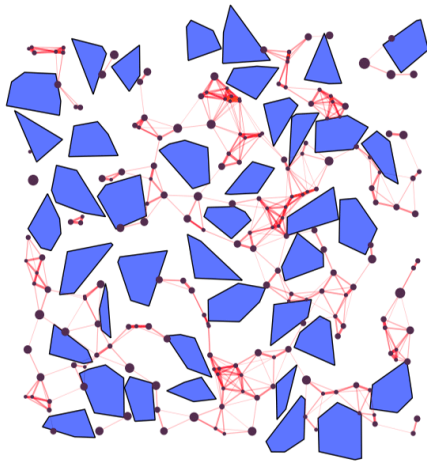
$$\text{area} \left( \mathcal{FOV}_i \setminus \bigcup_{i \neq j} \mathcal{FOV}_j \right)$$



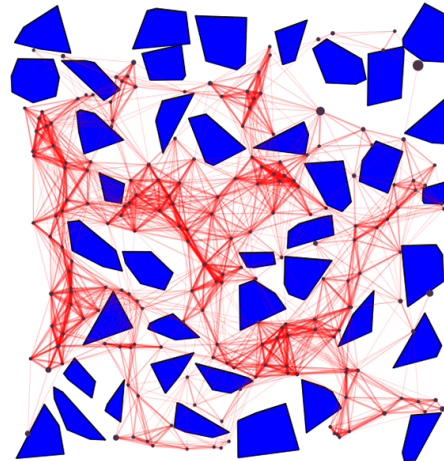
Σχήμα 15: Παράδειγμα Επαγόμενου Γράφου εξάρτησης.

#### 5.4.2. Απόφαση Αύξησης Ακτίνας

Βασισμένοι στον προηγούμενο γράφο, μπορούμε πια να χρησιμοποιήσουμε τον παρακάτω ευρετικό τρόπο για την αύξηση ή μείωση ακτίνας: Εξετάζουμε όλους τους γειτονικούς



(α') Για μέγιστη ακτίνα  $r = 0.5$



(β') Για μέγιστη ακτίνα  $r = 2.0$

Σχήμα 16: Επαγόμενος Γράφος Εξάρτησης για διαφορετικές μέγιστες ακτίνες. Είναι εμφανές ότι ο γράφος είναι πιο πυκνός για μεγαλύτερη μέγιστη ακτίνα.

κόμβους - που ορίζονται με βάση τον προηγούμενο γράφο. Εάν ο κόμβος μας έχει την καλύτερη εφικτή αύξηση, δηλαδή τον καλύτερο λόγο εμβαδού - ενέργειας από όλους τους γείτονές του, τότε αυξάνουμε την ακτίνα του. Επειδή πάντα θα υπάρχει ένας κόμβος με τον καλύτερο λόγο, σε κάθε βήμα θα αυξάνει τουλάχιστον ένας κόμβος την ακτίνα του με κατανομημένο τρόπο. Βέβαια συνήθως περισσότεροι κόμβοι θα έχουν τον καλύτερο λόγο από όλους τους γείτονές τους και συνεπώς θα αυξάνουν την ακτίνα τους.

Ως αποτέλεσμα αυτής της πρότασης, εγείρεται το εξής ερώτημα: πότε ένας κόμβος θα σταματάει να αυξάνει την ακτίνα του; Λόγου χάρι ένας κόμβος απομονωμένος σε αυτόν τον γράφο θα αυξάνει την ακτίνα του συνέχεια - θα πρέπει να εισάγουμε έναν τρόπο ώστε κάποια στιγμή να σταματήσει, διαφορετικά μπορεί να έχουμε πολύ κακές αναθέσεις. Το κριτήριο τερματισμού περιγράφεται στην επόμενη υποπαράγραφο.

Σημαντική λεπτομέρεια είναι πως παρότι αυτοί οι κόμβοι είναι γειτονικοί στα πλαίσια του επαγόμενου γράφου εξάρτησης, εν γένει μπορεί να μην επικοινωνούν καθόλου, ή να επικοινωνούν με περισσότερα από 1-Hop και συνεπώς να μην είναι εφικτή ή να είναι δύσκολη και η τελική απόφαση. Μπορούμε να δουλέψουμε με την προσεγγιστική υπόθεση ότι υπάρχει εύκολη επικοινωνία μεταξύ τους.

### 5.4.3. Κριτήριο Τερματισμού Αύξησης Ακτίνας

Στην απόφαση που αναφέρθηκε νωρίτερα πρέπει με κάποιον τρόπο ένας κόμβος να σταματάει την αύξηση της ακτίνας του. Αν ενδιαφερόμασταν για 100% Coverage θα μπορούσαμε να ελέγχουμε ότι κάθε κόμβος έχει καλυμμένη από γειτονικά FOVs και εμπόδια την

περιφέρεια του, αξιοποιώντας το αποτέλεσμα που δίνεται από [12]. Όμως από τη στιγμή που ενδιαφερόμαστε για μερική κάλυψη θα μπορούσαμε να απαιτήσουμε να καλύπτεται ένα ποσοστό της περιφέρειας κάθε κόμβου με την υπόθεση ότι αυτό θα μεταφράζεται σε γενικότερα ικανοποιητικό ποσοστό κάλυψης.

Πειραματικά φάνηκε ότι αυτή η υπόθεση επαληθεύεται. Δηλαδή ζητώντας ποσοστιαία κάλυψη της περιφέρειας πήραμε ικανοποιητική συνολική κάλυψη του χώρου, όμως υπάρχει κάποια μεταβλητότητα στα αποτελέσματα, η οποία για μεγαλύτερα δίκτυα θα αναμέναμε να είναι ακόμα μικρότερη λόγω του νόμου των μεγάλων αριθμών.

Έτσι λοιπόν, στον αλγόριθμο μας, το κριτήριο για κάθε κόμβο, ώστε αν σταματήσει να αυξάνει την ακτίνα του, είναι

$$PCOV_i \geq l$$

Όπου παραπάνω το  $l$  είναι το ποσοστό της κάλυψης της περιφέρειας που ζητάμε.

#### 5.4.4. Περιγραφή Αλγόριθμου

Θεωρούμε τις συναρτήσεις

- **Neighbourhood**: λαμβάνει ως είσοδο έναν γράφο, και έναν δείκτη ενός κόμβου, και υπολογίζει τους δείκτες των γειτονικών κόμβων στον γράφο αυτό.
- **IncreaseEachInSet**: παίρνει ένα σύνολο δεικτών  $I$  και μια βαθμωτή ποσότητα, και αυξάνει τις ακτίνες  $R_i$  για κάθε  $i \in I$  κατά τη βαθμωτή αυτή ποσότητα.
- **FindRedundantNodes**: είναι μια συνάρτηση που υπολογίζει το σύνολο με τους δείκτες των κόμβων που είναι περιττοί στην τρέχουσα κάλυψη.
- **SetZeroEachInSet**: μια συνάρτηση που παίρνει ένα σύνολο δεικτών, και μηδενίζει τις ακτίνες  $R_i$  αυτών
- **argmax**: εδώ θεωρούμε ότι επιστρέφεται το σύνολο των δεικτών που έχουν το ίδιο καλή αύξηση, και όχι μόνο ένας από αυτούς.

Σε αυτόν τον αλγόριθμο όλοι μαζί οι κόμβοι (γραμμή 4-12) κοιτάνε την γειτονιά τους (γραμμή 8-9) πριν γίνει μια ή περισσότερες αλλαγές “ταυτόχρονα” (γραμμή 14).

Στη γραμμή 5 γίνεται έλεγχος για την κάλυψη της περιφέρειας κάθε κόμβου. Εάν την έχουμε καλύψει αρκετά, τότε συνεχίζουμε με τον επόμενο κόμβο. Στη γραμμή 9 ένας

---

**Αλγόριθμος-Συνάρτηση 4:** Κατανεμημένος Greedy Αλγόριθμος

---

```
1 function GreedyAssignment ( $\mathbf{O}, \mathbf{P}, l$ )
2  $\mathbf{R} = \mathbf{0}$ 
3 while Change of  $\mathbf{R}$  in the last iteration do
4   for  $i$  from 1 to  $N$  do
5     if  $\mathcal{PCOV}_i \geq l$  then
6       Continue
7     end
8     nei = Neighbourhood( $\mathcal{G}, i$ )
9     if  $i \in \arg \max_{j \in (nei) \cup \{i\}} \{incrementRatio(j)\}$  then
10      |  $SetToIncrease = SetToIncrease \cup \{i\}$ 
11    end
12  end
13  SetToZero = FindRedundantNodes(Sensors)
14  IncreaseEachInSetBy(SetToIncrease,  $\Delta r$ )
15  SetZeroEachInSet(SetToZero)
16 end
17 return  $\mathbf{R}$ 
```

---

κόμβος  $i$  εξετάζει εάν ανήκει στο σύνολο με τους καλύτερους κόμβους, εξετάζοντας εάν μεταξύ των κόμβων που έχουν δείκτη απο το σύνολο  $(nei) \cup \{i\}$  έχει τον καλύτερο λόγο αύξησης. Αν είναι ο καλύτερος στο σύνολο αυτό τότε τοποθετείται σε ένα σύνολο απο δείκτες κόμβων `SetToIncrease` που θα αυξηθούν στην επόμενη βηματική αύξηση `IncreaseEachInSetBy`.

Με την εξέλιξη του αλγόριθμου κάποιοι κόμβοι θα γίνουν περιττοί για την κάλυψη, στη γραμμή 13 βρίσκουμε ποιοί είναι αυτοί, σε κάθε επανάληψη, και στη γραμμή 15 μηδενίζουμε την ακτίνα τους.

## 5.5. Δυνατές Επεκτάσεις με Προσεγγίσεις Ολικής Βελτιστοποίησης

Ένα πρόβλημα είναι πως ο αλγόριθμος μας μπορεί να συγκλίνει κοντά σε κάποιο τοπικό μέγιστο, όμως γενικά δεν προσεγγίζει το ολικό μέγιστο. Ένας τρόπος να περιορίσεις λίγο το πρόβλημα είναι με το να κινείσαι με κάποια τυχαίότητα, καθοδηγούμενη μερικώς απο την πληροφορία της γειτονιάς σου. Όσο περισσότερο αξιοποιείς όμως την πληροφορία της γειτονιάς σου, τόσο περισσότερο κινδυνεύεις να παραμείνεις σε ένα τοπικό μέγιστο γύρω απο το οποίο περιπλανιέσαι. Αυτό ακριβώς το χαρακτηριστικό προσπαθούν να λύσουν μέθοδοι όπως `Simulated Annealing`. Το εμπόδιο στην τυχαίότητα μπαίνει απο το εύρος μιας περιοχής μεγίστου ή ελαχίστου. Ένα μεγάλο εύρος μιας περιοχής μεγίστου ή ελαχίστου πιο δύσκολα μπορεί να το ξεπεράσει μια μέθοδος προκειμένου να βρεθεί σε μια άλλη περιοχή με ενδεχομένως προτιμότερο ακρότατο.

Ένα δίλημμα (trade-off) που αναφέρεται και στο [17] είναι το `Exploration versus Exploitation`, δηλαδή το κατα πόσο αξιοποιείται η τοπική πληροφορία έναντι της αναζήτησης, μια ακόμα έκφανση του κόστους ευκαιρίας (`Opportunity Cost`). Γενικά στο πρόβλημα μας είναι εύκολο να φανταστούμε ότι ένας κόμβος δεν παράγει σημαντικές αλλαγές σε καμιά απο τις εκδοχές βελτιστοποίησης που ορίσαμε. Αυτό το χαρακτηριστικό της μερικής ομαλότητας για μεταβολές δείχνει ότι πρέπει να δοθεί περισσότερη έμφαση στο `Exploitation`, κάτι που ισχύει τις περισσότερες φορές σε προβλήματα βελτιστοποίησης [17].

Ένα άλλο χαρακτηριστικό του προβλήματος μας όπως αυτό περιγράφηκε, είναι της χωρικής εξάρτησης των κόμβων που έχει ως αποτέλεσμα να έχουμε σε ένα χωρίο μια καλή μερική λύση, ενώ σε ένα άλλο να έχουμε μια μη ικανοποιητική μερική λύση. Αυτό το χαρακτηριστικό θα μπορούσε κανείς να το εκμεταλλευτεί χρησιμοποιώντας μια ευρετική όπως αυτή του γενετικού προγραμματισμού, που κάνει μεταλλάξεις σε περιοχές γειτονιών ώστε να φροντίζει να διατηρεί τις καλές καταστάσεις σε γειτονιές που έχει βρει ο αλγόριθμος, και να βελτιώνει τις κακές σε άλλες γειτονιές.

Οπτικά και πειραματικά οδηγηθήκαμε στο ότι στο συγκεκριμένο πρόβλημα υπάρχουν πολλά τοπικά ελάχιστα που μας δίνουν λύσεις αρκετά κοντινές στη βέλτιστη - παρατήρηση που στηρίζεται στο ότι στα αποτελέσματα δεν υπάρχουν σημαντικές επικαλύψεις που είναι εφικτό να αποφύγει κανείς.



## 6. ΑΞΙΟΛΟΓΗΣΗ

Για την αξιολόγηση των μεθόδων μας χρησιμοποιήσαμε κάποιες μετρικές ως προς την κατανάλωση του δικτύου, το λόγο εμβαδού και ενέργειας, το πλήθος των βημάτων πριν τη σύγκλιση του αλγόριθμου, τις τυπικές αποκλίσεις των αποτελεσμάτων ώστε να δούμε πόσο συνεπείς και αξιόπιστοι είναι οι αλγόριθμοι μας, καθώς και τη μέση τιμή της ακτίνας των αισθητήρων. Είδαμε επίσης τη διαφοροποίηση των αποτελεσμάτων για διάφορα βήματα αύξησης  $\Delta r$ . Συγκρίναμε τους διάφορους αλγόριθμους μεταξύ τους σε διαφορετικές τοπολογίες αλλά και σε ίδιες.

### 6.1. Μετρικές

**Λόγος Κατανάλωσης.** Μια μετρική που χρησιμοποιήθηκε για την αξιολόγηση είναι ο λόγος κατανάλωσης για την ανάθεση που έχουμε ορίσει μέσα από τον αλγόριθμο, κανονικοποιημένη ως προς τη μέγιστη κατανάλωση, δηλαδή εκείνη που θα είχαμε αν είχαμε θέσει όλες τις ακτίνες αίσθησης μέγιστες.

$$ConsumRatio(\mathbf{R}) := \frac{Energy(\mathbf{R})}{Energy(\mathbf{R}_{max})} \quad (18)$$

Όσο μικρότερος ο λόγος τόσο καλύτερη η ανάθεση.

**Λόγος Εμβαδού δια Ενέργειας.** Από τη στιγμή που έχουμε τυποποιήσει το πρόβλημα και σαν βελτιστοποίηση του λόγου του Εμβαδού διά την Ενέργεια, είναι φυσικό να διαθέτουμε ως μετρική αυτόν το λόγο.

$$ApeRatio'(\mathbf{R}) := \frac{Covered(\mathbf{R})}{Energy(\mathbf{R})} \quad (19)$$

Αρχικά πρέπει να παρατηρήσουμε ότι σε αυτό το κριτήριο μεγαλύτερος λόγος είναι προτιμότερος, σε αντίθεση με το λόγο κατανάλωσης. Επιπλέον θα θέλαμε να κανονικοποιήσουμε αυτόν το λόγο όπως κάναμε στην προηγούμενη μετρική, ώστε να έχουμε μια απόλυτη αξιολόγηση ενός αποτελέσματος, και όχι μια συγκριτική τιμή σε αντιπαραβολή με κάποια άλλη τοπολογία.

Για την κανονικοποίηση μπορούμε και πάλι να χρησιμοποιήσουμε το αποτέλεσμα της ίδιας μετρικής εάν είχαμε τη μέγιστη ανάθεση. Έτσι καταλήγουμε στην παρακάτω μετρική:

$$ApeRatio(\mathbf{R}) := \frac{\frac{Covered(\mathbf{R})}{Energy(\mathbf{R})}}{\frac{Covered(\mathbf{R}_{max})}{Energy(\mathbf{R}_{max})}} \quad (20)$$

Πίνακας 1: Αποτελέσματα για 500 τοπολογίες, 200 αισθητήρες, 40 εμπόδια

Τύπος	Λόγος Κατανάλωσης	Βήματα	Εμβαδό/Ενέργεια
CHECK-ALL	m=0.187, std=0.00678	m=698.6, std=27.4	m=1.80, std=0.0664
CHECK-10-R	m=0.205, std=0.00880	m=732.5, std=27.0	m=1.64, std=0.0729

Συνεπώς αν  $ApeRatio(\mathbf{R}) > 1.0$  τότε έχουμε μια βελτιωτική ανάθεση από την τετριμμένη ανάθεση μέγιστης ακτίνας.

**Σημείωση για τις κανονικοποιημένες μετρικές.** Είναι χρήσιμο να σημειώσουμε την ευαισθησία των επιλεγμένων κανονικοποιημένων μετρικών ως προς το πλήθος των κόμβων (αισθητήρων). Πράγματι υπάρχει μια προφανής τάση με την αύξηση των κόμβων να βελτιώνονται οι μετρικές μας. Αυτό είναι κάτι που αναμένουμε και θέλουμε καθώς με λιγότερους αισθητήρες ο κάθε ένας είναι πιο πιθανό να καλύπτει χρήσιμο χώρο μόνος του - δηλαδή υπάρχει μικρότερη επικάλυψη των μεταξύ τους FOV, και άρα ο αλγόριθμος δεν μπορεί να κάνει κάποια σημαντική βελτίωση αφού δεν έχει πολλές επιλογές για να καλύψει κάποιον χώρο.

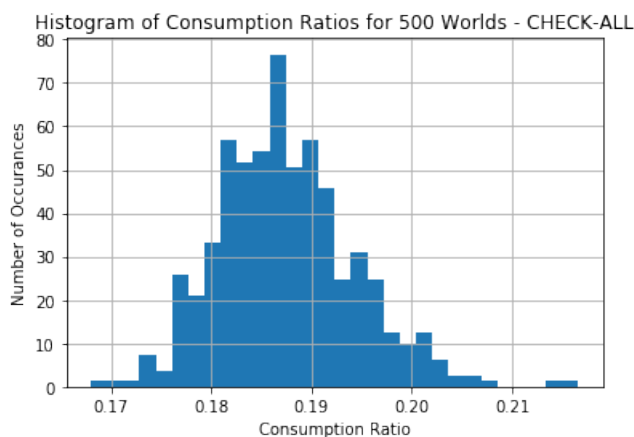
Όπως αναφέρθηκε η βελτίωση είναι αναμενόμενη και θεμιτή και έχει ως συνέπεια να μην έχει νόημα να συγκρίνονται επι της ίδιας βάσης αναθέσεις με διαφορετικό πλήθος κόμβων. Εάν θέλαμε να συγκρίνουμε διαφορετικό πλήθος κόμβων θα μπορούσαμε να επιβεβαιώσουμε ότι υπάρχει μια ικανοποιητική κλίση στην καμπύλη των αποτελεσμάτων μεταξύ πλήθους κόμβων και της μετρικής μας.

## 6.2. Αποτελέσματα Κεντριοποιημένων Αλγορίθμων

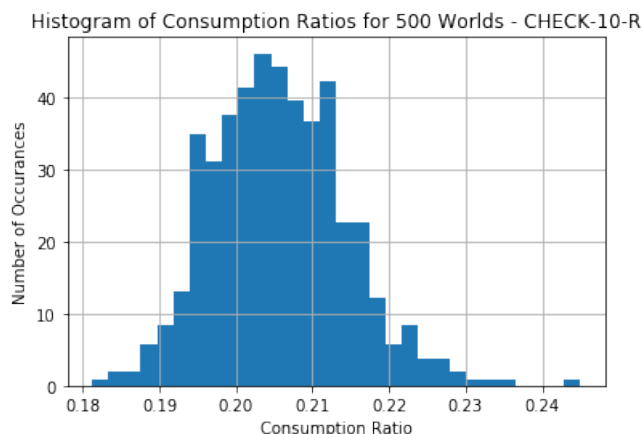
Προσομοιώσαμε 500 τοπολογίες με αλλαγή ακτίνας το  $1/5$  του μέγιστου εύρους σε κάθε βήμα, δηλαδή  $\delta r = \frac{R_{max}-R_{min}}{5}$ , κάτι που εξασφαλίζει ότι από μηδενική ακτίνα φτάνουμε σε μέγιστη με πέντε αυξήσεις. Συγκεκριμένα για μέγιστη ακτίνα ενός μέτρου, ελάχιστη 0 μέτρα, το επίπεδο αύξησης ήταν  $\frac{1m-0m}{5} = 20\text{ cm}$ . Υπενθυμίζεται ότι όλοι οι κόμβοι έχουν κοινό  $\Delta r$ . Στον πίνακα 1 ακολουθούν τα αποτελέσματα.

Ενώ η κατανομή των αποτελεσμάτων φαίνεται στα σχήματα 17 και 18, για τις δύο προσεγγίσεις, όπου έχουμε μια κατανομή που θυμίζει κανονική - κάτι αναμενόμενο δε-

δομένου ότι υπάρχουν πολλές τυχαίες μεταβλητές που έχουν επίδραση στο τελικό αποτέλεσμα και αθροιστικά δημιουργούν μια προσέγγιση της κατανομής, όπως περιμένει κανείς από το κεντρικό οριακό θεώρημα. Για μεγαλύτερο δίκτυο περιμένει κανείς ακόμα μικρότερη μεταβλητότητα, και πάλι ως αποτέλεσμα του κεντρικού οριακού θεωρήματος.



Σχήμα 17: Ιστόγραμμα αποτελεσμάτων μετά από προσομοίωση 500 τοπολογιών για τον Άπληστο Κεντρικό Αλγόριθμο (CHECK-ALL).



Σχήμα 18: Ιστόγραμμα αποτελεσμάτων μετά από προσομοίωση 500 τοπολογιών της παραλλαγής CHECK-10R

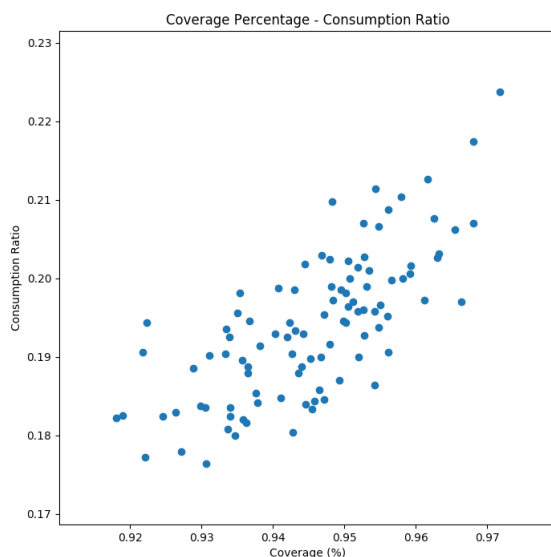
### 6.3. Αποτελέσματα Κατανεμημένου Αλγόριθμου

Γενικότερα στον κατανεμημένο αλγόριθμο η ποσοστιαία κάλυψη για δεδομένο όριο κάλυψης της περιμέτρου αποτελεί τυχαία μεταβλητή με σημαντική μεταβλητότητα. Δοκιμάσαμε διάφορες τιμές για την παράμετρο του ποσοστού κάλυψης της περιφέρειας, και είδαμε

ότι για περίπου 78% κάλυψη περιφέρειας, πήραμε κάλυψη του χώρου κοντά στο επιθυμητό 95%.

Αυτό φαίνεται στο Σχήμα 19 που αποτελεί scatter plot 100 τοπολογιών, όπου ζητάμε η περίμετρος να καλύπτεται τουλάχιστον στο 78% πριν σταματήσουμε να αυξάνουμε τον κόμβο με τον τρόπο που περιγράψαμε νωρίτερα.

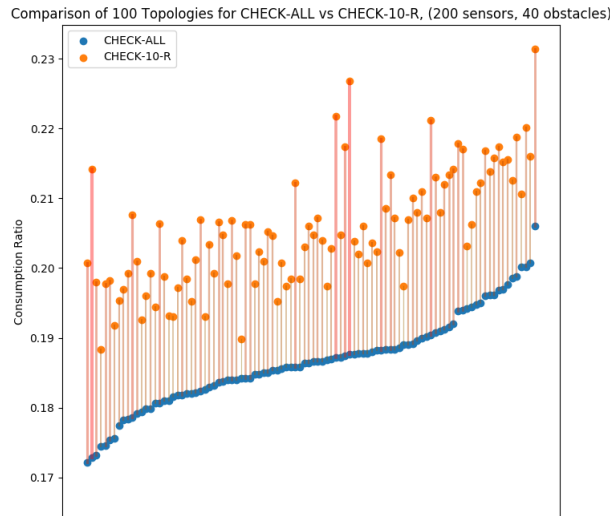
Στις 100 τοπολογίες η μέση τιμή για την κάλυψη ήταν **0.9454**, και η μέση τιμή για την κατανάλωση **0.194**. Ο λόγος εμβαδού - ενέργειας ισούταν με **1.739  $m^2/J$** . Τέλος η μέση τιμή στις ακτίνες των κόμβων ήταν ίση με **0.293**.



Σχήμα 19: Αποτελέσματα του Τοπικού Αλγόριθμου, με χρήση Scatterplot, για κάλυψη περιφέρειας 78%

#### 6.4. Σύγκριση Κεντρικού και Τυχαιοποιημένου σε Κοινές Τοπολογίες

Συγκρίναμε σε 100 τοπολογίες τα αποτελέσματα για τη μετρική Consumption Ratio στις ίδιες τοπολογίες, και παρουσιάζονται στο σχήμα 20. Είναι εμφανές ότι ο τυχαιοποιημένος CHECK-10-R αναμενόμενα υστερεί σε σύγκριση με τον CHECK-ALL, όμως υπο συνθήκες αυτή η διαφορά μπορεί να δικαιολογείται από το χρονικό κέρδος για τον υπολογισμό. Όπως αναμένει κανείς όσο μεγαλώνει η παράμετρος του CHECK-n-R τόσο τα αποτελέσματα τείνουν στα αποτελέσματα του CHECK-ALL .



Σχήμα 20: Σύγκριση της μεθόδου *CHECK-ALL* και *CHECK-10-R* σε 100 τοπολογίες στο λόγο κατανάλωσης

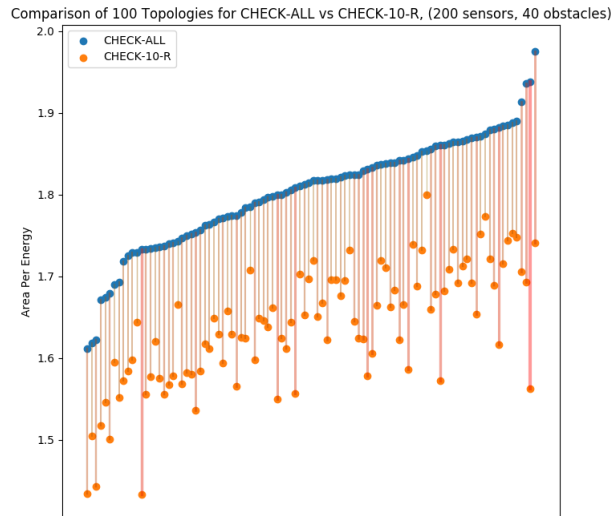
Για τη μετρική του Εμβαδού ανα Ενέργεια που όπως θέσαμε και το πρόβλημα είναι το κριτήριο που θέλουμε να βελτιστοποιήσουμε γενικά έχουμε τα αποτελέσματα που παρουσιάζονται στο σχήμα 21. Οι απόλυτες αριθμητικές ποσότητες δεν έχουν σημασία στα πλαίσια του πειράματός μας αφού δεν έχουμε χρησιμοποιήσει ρεαλιστικές τιμές - όμως η πληροφορία βρίσκεται στη σύγκριση των μεταξύ μεγεθών.

Όπως φαίνεται και στην εικόνα στην *CHECK-ALL* γίνεται καλύτερη αξιοποίηση των αισθητήρων.

Ενδεικτικά, δοκιμάσαμε επίσης τον κεντρικό αλγόριθμο και τον τυχαιοποιημένο αλγόριθμο σε 20 κοινές τοπολογίες για 300 κόμβους, για τη μετρική του λόγου κατανάλωσης. Τα αποτελέσματα παρουσιάζονται στο σχήμα 22. Παρατηρούμε την ευαισθησία του κριτηρίου του λόγου κατανάλωσης ως προς το πλήθος των κόμβων, αφού ο αλγόριθμος στην τοπολογία με 300 κόμβους βελτιώνει σημαντικά το κριτήριο. Συνεπώς ο αλγόριθμος μπορεί να αξιοποιήσει επιτυχώς τις περισσότερες επιλογές που του δίνονται.

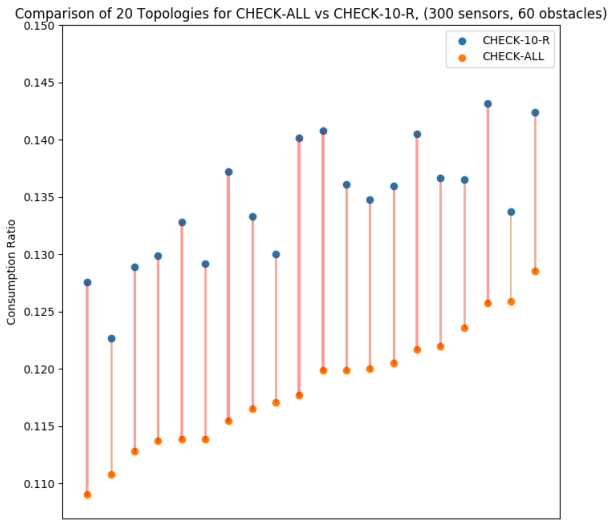
## 6.5. Πειραματική Αξιολόγηση των Επιλογών του Βήματος $\Delta r$

Σε μια άλλη σειρά πειραμάτων, δοκιμάσαμε διάφορες τιμές του  $\Delta r$ , για τιμές  $1/8$  του μέγιστου εύρους,  $\frac{1}{4}$ ,  $\frac{1}{2}$  καθώς και για απλή ενεργοποίηση-απενεργοποίηση του αισθητήρα,

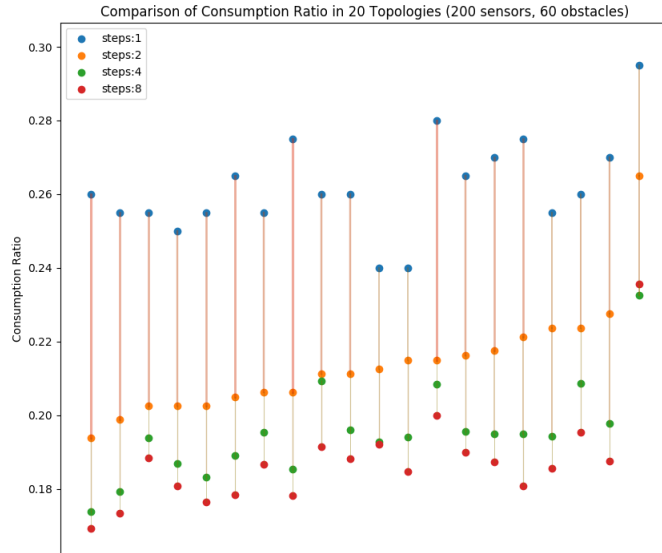


Σχήμα 21: Σύγκριση της μεθόδου *CHECK-ALL* και *CHECK-10-R* σε 100 τοπολογίες στο λόγο εμβαδού δια ενέργεια

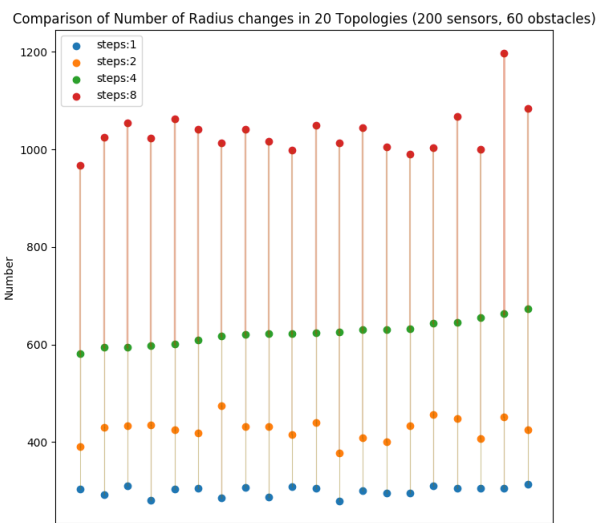
όπου δηλαδή το  $\Delta r$  είναι ίσο με το εύρος των ακτίνων. Έτσι στο σχήμα 23 φαίνονται τα διαφορετικά Consumption Ratios για διάφορες τιμές του βήματος με διαφορετικό χρώμα, σε 20 τοπολογίες. Αρχικά βλέπουμε πως με το ON-OFF (μπλέ χρώμα) έχουμε με διαφορά τη χειρότερη επίδοση, αλλά μόλις μεταβούμε σε δύο στάθμες έχουμε μια σημαντική βελτίωση στην επίδοση. από εκεί και πέρα, στις 4 και 8 στάθμες βλέπουμε ότι η διαφορά μειώνεται σημαντικά. Στο επόμενο σχήμα 24 βλέπουμε το πλήθος των αλλαγών στις ακτίνες αισθητήρων που πραγματοποίησε ο αλγόριθμος μέχρι να τερματίσει. Σε αυτά τα γραφήματα φαίνεται η σημαντική αύξηση του πλήθους των αλλαγών κάποιας ακτίνας, με τη μείωση του βήματος  $\Delta r$ , έχοντας ως συνέπεια μια αντίστοιχη καθυστέρηση στον τερματισμό του αλγόριθμου.



Σχήμα 22: Σύγκριση της μεθόδου CHECK-ALL και CHECK-10-R σε 20 τοπολογίες στο λόγο κατανάλωσης, για 300 κόμβους.



Σχήμα 23: Σύγκριση του λόγου κατανάλωσης για διαφορετικό βήμα αύξησης των αισθητήρων για 20 τοπολογίες



Σχήμα 24: Σύγκριση του πλήθους βημάτων πριν τη σύγκλιση για διαφορετικό βήμα αύξησης των αισθητήρων για 20 τοπολογίες



## 7. ΕΠΙΛΟΓΟΣ

### 7.1. Σύνοψη και Συμπεράσματα

Στην εργασία αυτή μελετήσαμε το πρόβλημα της κάλυψης του χώρου από ένα σύνολο αισθητήρων που καλύπτουν το πεδίο ορατότητας τους, παρουσία εμποδίων, και το μοντελοποιήσαμε ως πρόβλημα μαθηματικής βελτιστοποίησης σε ένα δισδιάστατο χώρο όπου εμπόδια και αισθητήρες είναι τοποθετημένα τυχαία. Είδαμε δύο βασικές εκφράσεις του προβλήματος, μια του λόγου του εμβαδού δια της ενέργειας την οποία και θέλαμε να μεγιστοποιήσουμε, και μια της ελαχιστοποίησης της συνολικής καταναλισκώμενης ενέργειας από τους αισθητήρες. Είδαμε πως αυτές οι δύο διαφορετικές εκφράσεις οδηγούν στο ίδιο πρόβλημα σε έναν περιορισμένο χώρο, και αντιμετωπίσαμε το πρόβλημα με βάση ένα κριτήριο της πρώτης έκδοσης για να λύσουμε και τα δύο προβλήματα κοντά στον περιορισμένο χώρο. Τα αποτελέσματα που λάβαμε ήταν ικανοποιητικά όμως υπήρχε σημαντικό κόστος από άποψη πολυπλοκότητας. Έτσι μεταβήκαμε σε έναν τυχαιοποιημένο αλγόριθμο ο οποίος επιλέγει από  $n$ -τυχαίους αισθητήρες εκείνον που θα επιλέξει να αυξήσει ή να μειώσει. Οι μετρικές χειρότερες ελαφρώς όμως βελτιώθηκε η πολυπλοκότητα του αλγόριθμου. Οι δύο προηγούμενοι αλγόριθμοι όμως ήταν κεντρικοποιημένοι και ως εκ τούτου δεν είναι εφικτό να κλιμακωθούν ικανοποιητικά, οπότε εισάγαμε έναν τελικό τοπικό καταναμημένο αλγόριθμο ο οποίος χρησιμοποιούσε μόνο γνώση της κατάστασης του και εκείνης των γειτόνων του. Αυτή η εκδοχή αξιοποιούσε έναν νέο γράφο που κωδικοποιούσε την εξάρτηση των κόμβων, και με βάση αυτόν αποφασίζαμε ποιος κόμβος και γιατί θα αυξήσει την ακτίνα. Εισάγαμε έναν τρόπο αποκεντρωμένης ανεξάρτητης διακοπής ο οποίος έδωσε μια κάλυψη με κάποια μεταβλητότητα η οποία σε ορισμένες εφαρμογές μπορεί να είναι αποδεκτή. Τα αποτελέσματα ήταν και αυτά ελαφρώς χειρότερα από την περίπτωση του κεντρικοποιημένου αλγόριθμου το οποίο ήταν εξίσου αναμενόμενο.

Το βασικό χαρακτηριστικό των ευρετικών μας είναι ότι δε διαθέτουν κάποια εγγύηση (guarantee), δηλαδή κάποιο λόγο προσέγγισης (approximation ratio) που ξέρουμε ότι τηρείται, οπότε δεν μπορούμε να είμαστε σίγουροι ότι σε συγκεκριμένες περιπτώσεις ο αλγόριθμος θα αντιμετωπίσει το πρόβλημα ικανοποιητικά, συγκριτικά με την καλύτερη λύση - η βασική δυσκολία είναι στην ύπαρξη των εμποδίων, η οποία κάνει δύσκολη την παροχή κάποιου φράγματος σε σχέση με την καλύτερη δυνατή λύση, και η οποία δεν είναι γνωστή.

## 7.2. Δυνατές Μελλοντικές Επεκτάσεις

Οι μέθοδοι που αναπτύχθηκαν μπορούν να οδηγήσουν σε πολλές διαφορετικές παραλλαγές οι οποίες μπορούν να βελτιώσουν την επίδοση επιπρόσθετα, και ανάλογα με τις απαιτήσεις του κάθε προβλήματος να δοθεί έμφαση σε μια προσέγγιση και στο πως θα βελτιωθεί. Εδώ αναφέρουμε κάποιες υποσχόμενες επεκτάσεις προς τις οποίες έχει ενδιαφέρον να κινηθεί κανείς.

**Τρισδιάστατη Επέκταση.** Το ίδιο σκεπτικό μπορεί να επεκταθεί σε κάλυψη σε τρεις διαστάσεις. Τότε θα μιλάμε αντι για δίσκους, για συμπαγείς σφαίρες. Η πολυπλοκότητα της υλοποίησης της προσομοίωσης θα μεγαλώσει αισθητά και κανείς θα πρέπει να απευθυνθεί σε ειδικές βιβλιοθήκες που χειρίζονται συνολοπράξεις σε τρεις διαστάσεις (εφόσον θα ήθελε να μεταφράσει πιστά τους αλγόριθμους), κάτι που δεν υποστηριζόταν από την βιβλιοθήκη που χρησιμοποιήσαμε σε αυτή την εργασία. Φυσικά στις τρεις διαστάσεις κανείς αντι να αναφέρεται για Εμβαδό ανά Ενέργεια θα αναφέρεται σε Όγκο ανά Ενέργεια.

**Προβληματικές Περιπτώσεις.** Στις προβληματικές περιπτώσεις που αναφέραμε νωρίτερα στην εργασία αυτή, ο αλγόριθμος μας υστερεί κατά εμφανή τρόπο. Μια τεχνική για την αντιμετώπιση του προβλήματος της τοπικής καλής κάλυψης αλλά της συνολικής κακής κάλυψης ενός αισθητήρα είναι με κάποια πιθανότητα, ή ανα κάποιες περιόδους να ελέγχεται το εάν είναι συνετό να μηδενιστεί τελείως η ακτίνα του αισθητήρα. Έτσι παρότι μπορεί ο αισθητήρας να ξεκίνησε με καλό λόγο και να κατέληξε σε κακό λόγω των γειτονικών αισθητήρων, θα έχει τη δυνατότητα να απενεργοποιηθεί ώστε να μειώσει την συνολική κατανάλωση του συστήματος.

**Επιθετική στρατηγική μείωσης στην Τοπική Εκδοχή.** Η πολιτική μείωσης στην τοπική εκδοχή που χρησιμοποιήθηκε ήταν κάπως φτωχή αφού μόνο σε περιπτώσεις όπου ένας αισθητήρας ήταν τελείως περιττός του μειώναμε την ακτίνα. Θα θέλαμε να μειώνουμε τις ακτίνες αισθητήρων και σε περιπτώσεις όπου απλά ο λόγος εμβαδού δια ενέργειας για την βηματική μείωση να είναι σχετικά κακός.

**Αξιοποίηση του γράφου με βάρη στην τοπική εκδοχή.** Για την μοντελοποίηση των εξαρτήσεων χρησιμοποιήθηκε ο επαγόμενος γράφος εξάρτησης. Αυτός ο γράφος όπως περιγράψαμε μπορεί να μας δώσει σημαντική πληροφορία για το μέγεθος της εξάρτησης δύο αισθητήρων - πληροφορία που δεν αξιοποιήσαμε καθώς δεν θεωρήσαμε κάποιον προφανή τρόπο να το εισάγουμε. Θα είχε ενδιαφέρον κανείς να μελετήσει το κατα πόσο μπορεί να αξιοποιήσει αυτόν τον γράφο με βάρη.

**Προσαρμοστική Αλλαγή του  $\Delta r$ .** Δοκιμάσαμε με κάποιες σύντομες προσομοιώσεις την ιδέα της προσαρμοστικής αλλαγής του διακριτού βήματος  $\Delta r$ . Αυτό είχε ως αποτέλεσμα να μην κινούμαστε σε αυστηρά καθορισμένες ακτίνες, και να χάνουμε κάποια πλεονεκτήματα όπως για παράδειγμα το caching. Αυτό προκύπτει από το ότι υπάρχουν περισσότερα επίπεδα ακτίνας αφού δεν είναι σαφώς καθορισμένα και η πιθανότητα να μεταβούμε σε ένα κοινό επίπεδο ακτίνας μειώνεται σημαντικά. Έτσι δεν μπορούμε να αξιοποιήσουμε ποσότητες που μένουν ίδιες σε μια συγκεκριμένη ακτίνα.

Συγκεκριμένα η ιδέα είναι να χρησιμοποιήσει κανείς ως βήμα αλλαγής ακτίνας το  $\Delta r_k = \max\{R_{max}e^{-\lambda k}, \Delta r_{min}\}$ , ή με κάποια άλλη φθίνουσα συνάρτηση αντί της αντίστροφης εκθετικής, το οποίο μας δίνει σαν αποτέλεσμα μεγάλες αλλαγές ακτίνας στην αρχή, οι οποίες μειώνονται με κάποιο ρυθμό έως ότου φτάσουν σε έναν κορεσμό για να εξασφαλίσουμε ότι γίνεται κάποια πρόοδος. Η διαίσθηση πίσω από αυτήν την επιλογή είναι ότι στην αρχή υπάρχει χώρος για καλές επιλογές, και το να κοιτάς σε όλο το δίσκο ποιος είναι ο λόγος εμβαδού ενέργειας μας δίνει τη δυνατότητα να κάνουμε ακόμα καλύτερες επιλογές. Τα αποτελέσματα που λάβαμε με κάποιες πρόχειρες δοκιμές ήταν περίπου τα ίδια με τον κεντρικοποιημένο αλγόριθμο, αλλά με αρκετά ταχύτερη σύγκλιση- όμως αναδύονται προβλήματα επιλογής του  $\lambda$  ή της συνάρτησης μείωσης τα οποία πρέπει να μελετηθούν πιο συστηματικά πριν βγουν ασφαλή συμπεράσματα.

**Πιθανότητα μείωσης ανάλογη με την πυκνότητα κόμβων.** Για την μείωση κάποιας ακτίνας χρησιμοποιήθηκε μια σταθερή εμπειρική τιμή, η οποία θεωρούμε ότι πρέπει να είναι ανάλογη με την πυκνότητα των κόμβων. Αυτό στηρίζεται στο ότι οι μειώσεις χρειάζονται λόγω συνωστισμών αισθητήρων και ανατροπών ως προς το λόγο εμβαδού με ενέργεια που είχε κατά την αύξηση σε σύγκριση με την ίδια ποσότητα σε ένα μεταγενέστερο στάδιο όπου ένας γειτονικός κόμβος μεγάλωσε. Έτσι διαισθητικά, με μεγαλύτερη πυκνότητα αναμένουμε περισσότερες ανατροπές.

**Χαλάρωση των περιορισμών για απλοποίηση του προβλήματος.** Η απλοποιημένη έκδοση για ένα μοντέλο τετραγωνικού κόστους της ενέργειας (εξίσωση 8), θα μπορούσε να αναχθεί σε Τετραγωνικό Προγραμματισμό (Quadratic Programming), αν η εφικτή περιοχή μπορούσε να περιγραφεί από ένα πολύτοπο. Παρόλα αυτά αυτό δεν ισχύει ακόμα και στις πολύ απλές περιπτώσεις, όμως ίσως θα είχε ενδιαφέρον να εξετάσει κανείς αν μπορεί να προσεγγιστεί η εφικτή περιοχή από ένα πολύτοπο. Εξάλλου η σταθερά  $T$  ορίζεται αυθαίρετα, οπότε η μερική παραβίαση της μπορεί να αγνοηθεί ή να αντιμετωπιστεί. Η δυσκολία στο πρόβλημα μας βέβαια έχει σχέση με την ίδια την άγνοια μας ως προς την μορφή της εφικτής περιοχής. Ένας σχετικός τρόπος για να γίνει το παραπάνω είναι με τη χρήση Lagrangian Χαλάρωσης, όπου δηλαδή εισάγονται πολλα-

πλασιαστές Lagrange που δίνουν μια προτίμηση να μένουμε εντός της εφικτής περιοχής δίνοντας ένα προβάδισμα σε αυτές τις λύσεις, ενώ τιμωρούν τις λύσεις που βγαίνουν εκτός της εφικτής περιοχής.

## 8. ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ

Η προσομοίωση έγινε σε γλώσσα προγραμματισμού Python. Χρησιμοποιήθηκαν οι βιβλιοθήκες Shapely για πράξεις υπολογιστικής γεωμετρίας όπως για παράδειγμα τομή συνόλων, περιγραφή συνόλων, ένωση συνόλων. Επίσης χρησιμοποιήθηκε κυρίως για την εύρεση του Field of Vision η βιβλιοθήκη Py2D. Πέρα από αυτές τις δυο βιβλιοθήκες δεν έγινε ιδιαίτερη άμεση χρήση κώδικα από τρίτους.

Για την υλοποίηση χρησιμοποιήθηκαν μεταξύ άλλων, δυο κεντρικές κλάσεις. Η μια αναπαριστούσε εμπόδια και ονομαζόταν Obstacle και ακολουθεί η υλοποίηση της:

```
1 from Constants import *
2 from Helpers import *
3 from numpy.random import rand
4 from shapely.geometry import MultiPoint, Point, Polygon, LineString,
   MultiPolygon
5 from shapely.affinity import translate
6
7
8
9 class Obstacle:
10     obs = []
11
12     def __init__(self, polygon):
13         """
14         Vertices should be given in the format:
15         vertices = [ (x1,y1), (x2,y2) ... , (xn,yn), (x1,y1) ]
16         """
17         self.poly = polygon
18         Obstacle.add_polygon(self)
19
20     @staticmethod
21     def add_polygon(obs):
22         Obstacle.obs.append( obs )
23
24     @staticmethod
25     def kill_obs():
26         Obstacle.obs = []
27
28     @staticmethod
29     def obs_polys():
30         return map(lambda o: o.poly, Obstacle.obs)
31
32     @staticmethod
33     def gen_rand_convex():
```

```

32     points = SIZE_PERCENT*ENTITIES_RANGE*rand(POINTS_BASED_POLYS,2)
33     # 2 = dimensions.
34     return MultiPoint(points).convex_hull
35
36     @staticmethod
37     def gen_rand_concave():
38         points = []
39         seed = SIZE_PERCENT*ENTITIES_RANGE*rand(1,2)
40         for i in range(5):
41             seed = seed + 0.4*SIZE_PERCENT*ENTITIES_RANGE*(rand(1,2)
42             -0.5)
43             points.append(list(seed[0]))
44             line = LineString(points)
45             return line.buffer(0.01+rand()*0.08)
46
47     @staticmethod
48     def srand_polys(obstacles):
49         ch = Obstacle.gen_rand_convex()
50         x_c = ch.centroid.x
51         y_c = ch.centroid.y
52         ch = translate(ch, xoff=-x_c, yoff=-y_c)
53
54         while True:
55             rep = ENTITIES_RANGE*rand(1,2)
56             rep_x = rep[0][0]
57             rep_y = rep[0][1]
58             ch = translate(ch, xoff=rep_x, yoff=rep_y)
59             if all(not obs.poly.intersects(ch) for obs in obstacles):
60                 break
61             else:
62                 rep_x = -1 * rep_x
63                 rep_y = -1 * rep_y
64                 ch = translate(ch, xoff=rep_x, yoff=rep_y)
65         return ch
66
67     @staticmethod
68     def gen_rand_topo(obs_num):
69         Obstacle.kill_obs()
70         for i in range(obs_num):
71             Obstacle(Obstacle.srand_polys(Obstacle.obs))

```

Η άλλη κλάση αναπαριστούσε έναν αισθητήρα, ονομαζόταν Sensor και δίνεται παρακάτω:

```

1 from __future__ import print_function
2 import numpy as np

```

```

3 from numpy.random import rand
4 from shapely.geometry import MultiPoint, Point, Polygon, LineString,
  MultiPolygon
5
6 from py2d.FOV import Vision
7 from py2d.Math import Polygon as FPoly
8 from py2d.Math import Vector as FVector
9
10 from Constants import *
11 from Helpers import *
12 from TestHelpers import *
13 from Obstacle import Obstacle
14
15 class Sensor:
16     sens = []
17     range_changes = 0
18
19     def __init__(self, position, max_range=MAX_RANGE, min_range=MIN_RANGE)
20     :
21         """
22         Inputs:
23             position of the sensor.
24             max_range of the sensor.
25             min_range of the sensor.
26         """
27
28         self.point = Point(position)
29         self.maxrange = max_range
30         self.minrange = min_range
31         self.currange = min_range
32         self.inrange_obs = []
33         self.inrange_sens = []
34         self.consume_const = DEFAULT_CONSUME_CONSTANT
35
36         # Metric Relating to the Importance of a sensor. It is the
37         average
38         # clear area, under random arrangements.
39         self.stochastic_imprtance = None
40
41         self._incr_cache = None
42         self._dcr_cache = None
43
44         Sensor.add_sensor(self)

```

```

44 #-----
45 #----- CHANGE STATE OF SENSOR
46 #-----
47
48 def set_range(self, rng):
49     self.currange = bounds(self.minrange, rng, self.maxrange)
50     self.invalidate_caches() # TODO Check again
51     Sensor.range_changes += 1
52     for s in self.inrange_sens:
53         s.invalidate_caches()
54
55 def invalidate_caches(self):
56     self._dcr_cache = None
57     self._incr_cache = None
58
59 #-----
60 #----- METRICS / CHARACTERISTICS
61 #-----
62
63 @memoize
64 def circumference_length_at_r(self, r):
65     vispol = self.actual_vis_polygon_at(r)
66
67     return safe_poly_boundary(vispol).length
68
69 def circumference_length(self):
70     return self.circumference_length_at_r(self.currange)
71
72 def circumference_covered(self):
73     """
74     This Computes the Covered Circumference of the Field of
75     Vision as a percentage of it's total length.
76     """
77     clen = self.circumference_length()
78
79     if clen==0: return 0
80
81     boundary = safe_poly_boundary(self.actual_vis_polygon())
82     boundary_rdc_d_sensors = boundary.difference(self.
neighbourhood_cur_vis_pols())
83     boundary_rdc_d_all = boundary_rdc_d_sensors.difference(self.
neighbourhood_obs_pols())
84
85     return (clen-boundary_rdc_d_all.length)/clen

```



```

86
87
88
89 #-----
90 #----- REDUCED POLYGONS
91 #-----
92
93 def neighbourhood_obs_pols(self):
94     """
95     helper function. Computes the neighbourhood of functions
96     """
97     return union_polygons(map(lambda o: o.buffer(EPSILON), self.
inrange_obs))
98
99 def neighbourhood_max_vis_pols(self):
100     """
101     helper function. Computes the neighbourhood of functions
102     """
103     return union_polygons(map(lambda s: s.visibility_polygon(), self.
inrange_sens))
104
105 def neighbourhood_cur_vis_pols(self):
106     """
107     helper function. Computes the neighbourhood of functions
108     """
109     return union_polygons(map(lambda s: s.actual_vis_polygon(), self.
inrange_sens))
110
111
112 @memoize
113 @pipe_result_dec(assert_valid)
114 def visibility_polygon_alone(self):
115     return self.visibility_polygon().difference(self.
neighbourhood_max_vis_pols())
116
117 @pipe_result_dec(assert_valid)
118 @pipe_result_dec(heal_polygon_with_interiors)
119 def max_reduced_vis_pol(self):
120     """
121     This is the maximum reduced visibility polygon. That is,
122     the visibility polygon if you subtract the regions covered
123     by the nearby sensors.
124     """
125     pol = self.visibility_polygon()

```

```

126     others = assert_valid(reduce(lambda a,s: a.union(s.
actual_vis_polygon()),self.inrange_sens, Polygon()))
127     return pol.difference(others)
128
129     @pipe_result_dec(assert_valid)
130     @pipe_result_dec(heal_polygon_with_interiors)
131     def reduced_ring_polygon(self,rad,dr=DISCR_STEP):
132         """
133         This is like the ring polygon, but having subtracted the covered
area by nearby sensors.
134         """
135         mxpol = self.max_reduced_vis_pol()
136         return mxpol.intersection(self.ring_polygon(rad,dr))
137
138     @pipe_result_dec(assert_valid)
139     def increment_rdcdd_vis_pol(self,dr=DISCR_STEP):
140         if self._incr_cache is not None:
141             return self._incr_cache
142         mxpol = self.max_reduced_vis_pol()
143
144         try:
145             self._incr_cache = mxpol.intersection(self.ring_polygon(self
.currange,dr))
146         except:
147             from IPython.display import display
148             from shapely.validation import explain_validity
149             print(explain_validity(mxpol))
150             print('BOOM!')
151             display(mxpol)
152             display(self.ring_polygon(self.currange,dr))
153             try:
154                 self._incr_cache = mxpol.buffer(EPSILON).intersection(
self.ring_polygon(self.currange,dr))
155                 display(self._incr_cache)
156             except:
157                 raise Exception('Universe Exploded')
158
159         return self._incr_cache
160
161     @pipe_result_dec(assert_valid)
162     def dcr_rdcdd_vis_pol(self,dr=DISCR_STEP):
163         if self._dcr_cache is not None:
164             return self._dcr_cache
165

```

```

166     mxpol = self.max_reduced_vis_pol()
167     self._dcr_cache = mxpol.intersection(self.ring_polygon(self.
currange-dr, dr))
168     return self._dcr_cache
169
170     #-----
171     #----- VISIBILITY POLYGONS
172     #-----
173
174
175     @pipe_result_dec(assert_valid)
176     def actual_vis_polygon(self):
177         """
178         This is the visibility polygon that is covered by the sensor, at
the range it is set.
179         """
180         return self.actual_vis_polygon_at(self.currange)
181
182     @memoize
183     def actual_vis_polygon_at(self, r):
184         """
185         This is the visibility polygon that is covered by the sensor, at
a specified range by
186         the input argument 'r'.
187         """
188         if r <= 0: return Polygon()
189
190         poly = self.visibility_polygon()
191         return poly.intersection(self.point.buffer(r))
192
193     @memoize
194     @pipe_result_dec(assert_valid)
195     #@simplify_poly_dec
196     def visibility_polygon(self):
197         """
198         Compute the maximal visibility polygon for the sensor.
199         """
200         ppoint = FVector(self.point.x, self.point.y)
201         obstructors = [ map(lambda n: FVector(*n), obs.exterior.coords)
for obs in self.inrange_obs]
202         if len(obstructors) == 0:
203             return self.point.buffer(self.maxrange).intersection(
UNIVERSE_POLYGON)
204         vision = Vision(obstructors)

```

```

205     radius = self.maxrange
206     sides = VIS_POL_CIRC_SIDES
207     boundary = FPoly.regular(ppoint, radius, sides)
208     vision = vision.get_vision(ppoint, radius, boundary)
209     return heal_polygon(Polygon(map(lambda n:(n.x,n.y), list(vision))
)) \
210         .intersection(UNIVERSE_POLYGON)
211
212         #.intersection(circle(self.point,
radius)) \
213
214     #-----
215     #----- HELPFUL POLYGONS
216     #-----
217
218     @memoize
219     @pipe_result_dec(assert_valid)
220     def ring_polygon(self, rad, dr):
221         if rad < 0:
222             raise ValueError('rad should be positive or zero')
223
224         dr = dr if rad+dr<=self.maxrange else self.maxrange - rad
225
226         return ring_poly(self.point, rad, dr=dr).intersection(self.
visibility_polygon())
227
228     #-----
229     #----- CONSUMPTION AT REGIONS
230     #-----
231
232     @memoize
233     #@assert_valid_radius(1) TODO
234     def ring_consumption(self, rad, dr):
235         """
236         This is not constant for all since we allow consumption models
to differ.
237
238         This can take as an input a positive dr, for increasing, or
negative dr
239         for decreasing a radius. As result the returned value is signed
240         """
241
242         if dr > 0:
243             dr = dr if rad+dr<=self.maxrange else self.maxrange - rad

```

```

244     elif dr<0:
245         dr = dr if rad+dr>=self.minrange else self.minrange - rad
246     else:
247         print( 'WARNING! YOU ARE CALLING ring_consumption with dr=0.'
)
248
249     return self.consume_at_r(rad+dr) - self.consume_at_r(rad)
250
251     #-----
252     #----- CONSUMPTION AT POINT
253     #-----
254
255     def consume_at_r(self , rad):
256         return self.consume_const*(rad**CONSUMPTION_MODEL) if rad>=0
else 0
257
258     def curr_consum(self):
259         return self.consume_at_r(self.currange)
260
261     def max_consum(self):
262         return self.consume_at_r(self.maxrange)
263
264
265     #####
266
267     #-----
268     #----- DATA
269     #-----
270
271     @staticmethod
272     def add_sensor(sensor):
273         Sensor.sens.append( sensor )
274
275     #-----
276     #----- INITIALIZE
277     #-----
278
279     @staticmethod
280     def kill_sens():
281         Sensor.sens = []
282         Sensor.range_changes = 0
283
284     @staticmethod
285     def zero_ranges():

```

```

286     Sensor.range_changes = 0
287     for s in Sensor.sens:
288         s.set_range(s.minrange)
289
290
291     @staticmethod
292     def max_ranges():
293         Sensor.range_changes = 0
294         for s in Sensor.sens:
295             s.set_range(s.maxrange)
296
297     @staticmethod
298     def rand_ranges(domain=1.0):
299         Sensor.range_changes = 0
300         for s in Sensor.sens:
301             s.set_range(domain*(s.maxrange-s.minrange)*rand()+s.minrange
302 )
303
304 #----- HELPING POLYGONS
305 #-----
306
307
308     @staticmethod
309     def curr_vis_polys():
310         return map(lambda s: s.actual_vis_polygon(), Sensor.sens)
311
312     @staticmethod
313     def visibility_polys():
314         return map(lambda s: s.visibility_polygon(), Sensor.sens)
315
316     @staticmethod
317     def covered_polygon():
318         return reduce(lambda uni, current: uni.union(current), Sensor.
319 curr_vis_polys()) \
320             .intersection(UNIVERSE_POLYGON)
321
322     @staticmethod
323     def max_cov_polygon():
324         return reduce(lambda uni, current: uni.union(current), Sensor.
325 visibility_polys()) \
326             .intersection(UNIVERSE_POLYGON)
327
328 #----- UNDERSTAND ENVIRONMENT

```

```

327 #-----
328
329
330 @staticmethod
331 def sense_inrange_entities():
332     """
333     This method evaluates the in-range obstacles, based on the
334     distance between obstacle and the sensor
335     """
336     for s in Sensor.sens:
337         s.inrange_obs = filter(lambda o: s.point.distance(o) <= s.
338 maxrange, Obstacle.obs_polys())
339
340     max_intersection = 0
341     max_importance = 0
342     for s in Sensor.sens:
343         s.inrange_sens = filter(lambda s2:
344 s2 is not s and s.visibility_polygon().
345 intersects(s2.visibility_polygon()), Sensor.sens)
346         s.inrange_sens_weights = []
347
348         if s.inrange_sens:
349             s.inrange_sens_weights = map(
350 lambda s2: s.visibility_polygon().intersection(s2.
351 visibility_polygon()).area
352 ,s.inrange_sens)
353
354             max_intersection = max(max(s.inrange_sens_weights),
355 max_intersection )
356
357             s.importance_score = s.visibility_polygon_alone().area
358             max_importance = max(max_importance, s.importance_score)
359
360         for s in Sensor.sens:
361             s.inrange_sens_weights = map(lambda w: w/max_intersection ,s
362 .inrange_sens_weights)
363             s.importance_score = s.importance_score/max_importance
364
365 #-----
366 #----- METRICS / STATISTICS
367 #-----
368
369 @staticmethod
370 def max_consumption():

```

```

365         return sum(map(lambda sensor: sensor.max_consumption(), Sensor.sens))
366
367     @staticmethod
368     def current_consumption():
369         return sum(map(lambda sensor: sensor.curr_consumption(), Sensor.sens))
370
371     @staticmethod
372     def consumption_ratio():
373         return Sensor.current_consumption()/Sensor.max_consumption()
374
375     @staticmethod
376     def coverage():
377         """
378         Returns (Covered Area) / (Coverable Area) which is one of the
379         most important metrics we will use.
380         calculated by finding the union of all the covered visibility
381         polygons, evaluating the associated
382         area. Then we divide with the non-blocked area.
383         """
384         return Sensor.covered_polygon().area / Sensor.max_cov_polygon().
385         area
386
387     @staticmethod
388     def area_per_energy():
389         return Sensor.covered_polygon().area/Sensor.current_consumption
390         ()
391
392     @staticmethod
393     def average_rad():
394         avg = np.average(map(lambda s:s.currangle, Sensor.sens))
395         return avg
396
397     #-----
398     #----- TOPOLOGY CREATION
399     #-----
400
401     @staticmethod
402     def random_sensor_position():
403         while True:
404             point = ENTITIES_RANGE*rand(1,2)
405             point = Point(*point)
406             if all(not obs.intersects(point) for obs in Obstacle.
407             obs_polys()):
408                 break

```



```

404     return point
405
406     @staticmethod
407     def get_rand_topo(total_sens):
408         Sensor.kill_sens()
409         for i in range(total_sens):
410             Sensor(Sensor.random_sensor_position())
411             Sensor.sense_inrange_entities()
412
413     #-----
414     #----- PRINTERS / DEBUGGERS
415     #-----
416
417     @staticmethod
418     def stats_printer():
419         print(" Coverage: ", Sensor.coverage(),
420               "\n Consumption Ratio: ", Sensor.consumption_ratio(),
421               "\n")

```

Πέρα από αυτές τις κλάσεις υπήρχαν διάφορα επίπεδα κώδικα που αναλαμβάναν τη δημιουργία της τοπολογίας, την οπτικοποίηση αυτής, και το τρέξιμο των διάφορων στρατηγικών. Για παράδειγμα κάθε στρατηγική ήταν κάποιου είδους Schema,

```

1 from numpy.random import rand
2
3 def create_sequential_acts_schema(p, base, fixer=None):
4     """
5     This returns a schema that has the following properties:
6     calls a base function that will apply a step. Then it correct
7     potential problems with a probability.
8     """
9     if p==0:
10        def schema():
11            return (base(),0)
12    elif fixer:
13        if p==1:
14            def schema():
15                res1 = base()
16                res2 = fixer()
17                return (res1, res2)
18        else:
19            def schema():
20                res1 = base()
21                if rand()<p:
22                    res2 = fixer()

```

```

23         else:
24             res2 = 0
25         return (res1, res2)
26     else:
27         raise Exception('Should provide fixer function')
28     return schema
29
30
31 def create_batch_deterministic_schema(k, m, base, fixer):
32     """
33     This returns a schema that has the following properties:
34     calls a base function that will apply a step. Then it correct
35     potential problems with a probability.
36     """
37     def schema():
38         base_acc = 0
39         fixer_acc = 0
40
41         for i in xrange(k):
42             base_acc += base()
43         for i in xrange(m):
44             fixer_acc += fixer()
45
46         return (base_acc, fixer_acc)
47     return schema
48
49
50 def create_check_and_choose_schema(chooser, fitness_a, act_a, fitness_b,
51 act_b):
52     """
53     This type of schema evaluates the fitness of an
54     action 'A', then the fitness of an action 'B' and applies the action
55     that has a better fitness.
56     """
57     def schema():
58         if fitness_a() > fitness_b():
59             return (act_a(), 0)
60         else:
61             return (0, act_b())
62     return schema
63
64 def create_filter_and_run_schema(filter_what, # The set from which we
65 filter
66                                     filterer, # Apply an initial action

```

```

65         filter_rest , # Apply a different
action for the rest
66         changer):
67     universe = set(filter_what)
68     def schema():
69         to_be_changed = [s for s in filter_what if filterer(s)]
70
71         not_be_changed = universe - set(to_be_changed)
72         to_be_reduced = [s for s in not_be_changed if filter_rest(s) ]
73
74         for sensor in to_be_changed: changer(sensor,1)
75         for sensor in to_be_reduced: changer(sensor,-1)
76
77     return schema

```

Έπειτα σε ένα δεύτερο επίπεδο, υπάρχουν οι εκτελεστές Schemas οι οποίοι λαμβάνανε ως είσοδο ένα Schema και το εκτελούσαν μέχρι να τερματιστεί ο αλγόριθμος ή να ισχύει κάποιο άλλο κριτήριο, ανάλογα με το είδος του εκτελεστή. Ο κώδικας φαίνεται παρακάτω:

```

1 from Constants import *
2 from Sensor import Sensor
3
4 def create_schema_runner(schema,
5                             after_scheme_action=None,
6                             initializer=None,
7                             finisher=None):
8     """
9     This is a schema runner that takes as an input a schema that
10    radii-manipulation operations. It monitors area changes and stops
11    when %-coverage is above a threshold
12    """
13    def run_until_treshold(Threshold):
14        cumulative_area = 0
15        stop_when = Threshold * Sensor.max_cov_polygon().area
16        if initializer: initializer()
17        for i in range(2):
18            while cumulative_area < stop_when+EPSILON:
19                incr_affected, decr_affected = schema()
20                cumulative_area += (incr_affected - decr_affected)
21                if after_scheme_action: after_scheme_action()
22            if i==1: break
23        stop_when -= EPSILON
24        cumulative_area = Sensor.covered_polygon().area

```

```

25
26     if finisher: finisher()
27     return run_until_treshold
28
29 def create_schema_iter_runner(schema,
30                               after_scheme_action=None,
31                               initializer=None,
32                               finisher=None):
33     """
34     This is a schema runner that takes as an input a
35     schema and a number of iterations and applies the
36     schema 'Times' number of times.
37
38     For example this can be passed a local schema that
39     determines when to stop based on other criteria.
40     """
41     def run_until_iters(Times):
42         if initializer: initializer()
43         for i in xrange(Times):
44             schema()
45             if after_scheme_action: after_scheme_action()
46         if finisher: finisher()
47     return run_until_iters

```

## Αναφορές

- [1] Adrian Baddeley, Imre Bárány, and Rolf Schneider. *Stochastic Geometry: Lectures Given at the CIME Summer School Held in Martina Franca, Italy, September 13-18, 2004*. Springer, 2006.
- [2] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [3] Christian Buchta. On the boundary structure of the convex hull of random points. *Adv. Geom*, 12(1):79–190, 2012.
- [4] Francisc Bungiu, Michael Hemmer, John Hershberger, Kan Huang, and Alexander Kröller. Efficient computation of visibility polygons. *arXiv preprint arXiv:1403.3905*, 2014.
- [5] Mihaela Cardei and Jie Wu. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Computer communications*, 29(4):413–420, 2006.
- [6] Mihaela Cardei, Jie Wu, Mingming Lu, and Mohammad O Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *Wireless and Mobile Computing, Networking and Communications, 2005.(Wi-Mob'2005), IEEE International Conference on*, volume 3, pages 438–445. IEEE, 2005.
- [7] Vira Changkong and Yacov Y Haimes. Multiobjective decision making: Theory and methodology. In *North-Holland Series in System Science and Engineering*, volume 8. Elsevier Science Publishing Co New York NY, 1983.
- [8] Youssef Charfi, Naoki Wakamiya, and Masayuki Murata. Challenging issues in visual sensor networks. *IEEE Wireless Communications*, 16(2), 2009.
- [9] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005.
- [10] GaoJun Fan and ShiYao Jin. Coverage problem in wireless sensor network: A survey. *JNW*, 5(9):1033–1040, 2010.
- [11] Martin Haenggi. *Stochastic geometry for wireless networks*. Cambridge University Press, 2012.

- [12] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4):519–528, 2005.
- [13] Barry Joe and Richard B Simpson. Corrections to lee’s visibility polygon algorithm. *BIT Numerical Mathematics*, 27(4):458–473, 1987.
- [14] Vasileios Karyotis, Eleni Stai, and Symeon Papavassiliou. *Evolutionary Dynamics of Complex Communications Networks*. CRC Press, 2013.
- [15] Robert Lang. A note on the measurability of convex sets. *Archiv der Mathematik*, 47(1):90–92, 1986.
- [16] Der-Tsai Lee. Visibility of a simple polygon. *Computer Vision, Graphics, and Image Processing*, 22(2):207–221, 1983.
- [17] Sean Luke. *Essentials of metaheuristics*, volume 113. Lulu Raleigh, 2009.
- [18] Ritesh Kumar Madan. *Resource allocation algorithms for energy efficient wireless networks*. Stanford University, 2006.
- [19] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*, volume 1. Oxford university press New York, 1995.
- [20] Vijay Raghunathan, Curt Schurgers, Sung Park, and Mani B Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal processing magazine*, 19(2):40–50, 2002.
- [21] Paolo Santi. Topology control in wireless ad hoc and sensor networks. *ACM computing surveys (CSUR)*, 37(2):164–194, 2005.
- [22] Stanislava Soro and Wendi Heinzelman. A survey of visual sensor networks. *Advances in multimedia*, 2009, 2009.
- [23] Jun Sun, Eytan Modiano, and Lizhong Zheng. Wireless channel allocation using an auction algorithm. *IEEE Journal on Selected Areas in Communications*, 24(5):1085–1096, 2006.
- [24] Bang Wang. *Coverage control in sensor networks*. Springer Science & Business Media, 2010.
- [25] Jiong Wang and Sirisha Medidi. Energy efficient coverage with variable sensing radii in wireless sensor networks. In *Wireless and Mobile Computing, Networking*

- and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, pages 61–61. IEEE, 2007.
- [26] Yu Wang. Topology control for wireless sensor networks. In *Wireless sensor networks and applications*, pages 113–147. Springer, 2008.
- [27] Jie Wu and Shuhui Yang. Coverage issue in sensor networks with adjustable ranges. In *Parallel Processing Workshops, 2004. ICPP 2004 Workshops. Proceedings. 2004 International Conference on*, pages 61–68. IEEE, 2004.
- [28] Lin Xiao, Mikael Johansson, and Stephen Boyd. Simultaneous routing and resource allocation for wireless networks. In *Proceedings of the Annual Allerton Conference on Communication Control And Computing*, volume 39, pages 513–522. The University; 1998, 2001.
- [29] Zongheng Zhou, Samir R Das, and Himanshu Gupta. Variable radii connected sensor cover in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(1):8, 2009.
- [30] Chuan Zhu, Chunlin Zheng, Lei Shu, and Guangjie Han. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2):619–632, 2012.