Ατομική Διπλωματική Εργασία

Αλγόριθμος για τη Συσσώρευση Ρομπότ Σχήματος Δίσκων στο Επίπεδο

Χρυσοβαλάντης Αγαθαγγέλου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αλγόριθμος για τη Συσσώρευση Ρομπότ Σχήματος Δίσκου στο Επίπεδο

Χρυσοβαλάντης Αγαθαγγέλου

Επιβλέπων Καθηγητής
Μάριος Μαυρονικόλας

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2010

Περίληψη

Στο πρόβλημα *συσσώρευσης* των ρομπότ στο επίπεδο, μια συλλογή από n αυτόνομα ρομπότ επιχειρεί να τερματίσει φτάνοντας σε ένα συνεκτικό σχηματισμό: ένα σχηματισμό όπου υπάρχει μονοπάτι από ένα ρομπότ σε οποιοδήποτε άλλο ρομπότ το οποίο περιέχει μόνο σημεία που ανήκουν σε κάποιο από τα ρομπότ. Υποθέτουμε ότι το κάθε ρομπότ είναι ένας επίπεδος αδιαφανής δίσκος με ακτίνα 1. Ο υπολογισμός του ρομπότ εναλλάσσει φάσεις αποτύπωσης της σκηνής, υπολογισμού και κίνησης μέχρι τον τερματισμό.

Παρουσιάζουμε ένα κατανεμημένο αλγόριθμο για το πρόβλημα της συσσώρευσης των ρομπότ. Ο αλγόριθμος βασίζεται στην ιδέα της μετακίνησης ενός ρομπότ προς το σύνορο του κυρτού περιβλήματος του σχηματισμού που αυτό αντιλαμβάνεται όποτε αποτυπώνει την σκηνή. Έτσι, ο αλγόριθμος επιτυγχάνει επιπρόσθετα *πλήρη ορατότητα* : το κάθε ρομπότ “βλέπει” τουλάχιστον ένα σημείο από κάθε άλλο ρομπότ.

Το αποτέλεσμα αυτό βελτιώνει σημαντικά δύο αλγόριθμους που ήταν προηγουμένως γνωστοί για τις ειδικές περιπτώσεις όπου n=3 και n=4.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Πλαίσιο Εργασίας

Χρησιμοποιώντας ομάδες απλών, χαμηλού κόστους ρομπότ μπορούν να επιτευχθούν δύσκολες διεργασίες σε επικίνδυνα και αφιλόξενα περιβάλλοντα. Στόχος είναι η χρησιμοποίηση απλών, με μικρή υπολογιστική ισχύ συσκευών, οι οποίες μπορούν να παραχθούν σε μεγάλους αριθμούς με αρκετά μικρό κόστος. Τέτοια ρομπότ πρέπει να είναι όσο το δυνατό πιο απλά γίνεται. Τα ρομπότ που χρησιμοποιούμε έχουν σχήμα δίσκου και είναι όλα πανομοιότυπα. Δεν έχουν καθόλου μνήμη από προηγούμενες χρονικές στιγμές και δεν έχουν καμία συσκευή η οποία να τα συγχρονίζει (όπως για παράδειγμα κάποιο κοινό ρολόι). Ακόμα δεν έχουν κανένα σύστημα για επικοινωνία μεταξύ τους, παρά μόνο μία κάμερα σε κάθε ρομπότ. Η κάμερα αυτή βρίσκεται στην περιφέρεια του ρομπότ και φωτογραφίζει προς όλες τις κατευθύνσεις. Ανάλογα με την εικόνα που πήρε η κάμερα, το ρομπότ υπολογίζει το νέο σημείο στο οποίο θα κινηθεί και τέλος, κινείται ευθεία προς αυτό το σημείο. Αυτό επαναλαμβάνεται αρκετές φορές μέχρι τα ρομπότ να καταφέρουν να επιτύχουν κάποιο κοινό στόχο που έχουν.

Τέτοια ρομπότ έχουν αρκετές εφαρμογές που περιλαμβάνουν από αποστολές στο διάστημα μέχρι και τη χρήση τους από το στρατό σε διάφορες αποστολές. Ακόμα παρουσιάζεται μεγάλο ενδιαφέρον από διάφορα επιστημονικά πεδία, όπως η ρομποτική, η τεχνητή νοημοσύνη και ποιό πρόσφατα οι κατανεμημένοι αλγόριθμοι.

1.2 Το Πρόβλημα

Δεδομένου ρομπότ όπως αυτών που περιγράφηκαν στο υποκεφάλαιο 1.1, σκοπός μας είναι να παρουσιάσουμε κάποιο αλγόριθμο, ο οποίος να καταφέρνει να συγκεντρώσει όλα τα ρομπότ μαζί. Έχουμε σαν βασικό περιορισμό ότι τα ρομπότ έχουν όγκο, δηλαδή μπορούν να συγκρουστούν αλλά και να καλύψουν οποιοδήποτε ρομπότ από το να το βλέπει κάποιο άλλο ρομπότ αν βρίσκονται μπροστά του. Ακόμα υπάρχει και ο *εχθρός* ο οποίος έχει σαν δυνατότητα να καθορίζει την ταχύτητα που κινούνται τα ρομπότ, και να σβήνει οποιοδήποτε ρομπότ όποτε θέλει ακόμα και όταν κινείται, για όσο χρόνο θέλει. Μοναδικός περιορισμός του εχθρού είναι ότι κάθε φορά που σβήνει κάποιο ρομπότ, πρέπει σε κάποια χρονική στιγμή στο μέλλον να αφήσει το ρομπότ να κινηθεί μια ελάχιστη απόσταση. Ο αλγόριθμος δεδομένου του αριθμού των ρομπότ, του όγκου των ρομπότ αλλά και του *εχθρού*, πρέπει να καταφέρει να συγκεντρώσει όλα τα ρομπότ μαζί, δημιουργώντας κάποιο ενιαίο σχηματισμό σε κάποια περιοχή που δεν έχει καθοριστεί από πριν.

1.3 Τα Αποτελέσματα μας

<u>Κυρτό Περίβλημα:</u>
Ώς κυρτό περίβλημα (convex hull) ενός συνόλου σημείων ορίζεται το μικρότερο δυνατό κυρτό πολύγωνο που εσωκλείει το σύνολο των σημείων. Ποιο απλά αν έχω ένα σύνολο από σημεία και εφαρμόσω ένα λάστιχο γύρω από αυτά ώστε να περικλείει όλα τα σημεία, το σχήμα που θα έχει το λάστιχο είναι το κυρτό περίβλημα.

Ο αλγόριθμος που προτείνουμε σε αυτή την εργασία επιτυγχάνει τον σκοπό του, αρχικά φέρνοντας όλα τα ρομπότ πάνω στο κυρτό περίβλημα που τα περικλείει και έτσι όλα τα ρομπότ μπορούν να δουν όλα τα υπόλοιπα. Αργότερα τα ρομπότ πλησιάζουν μεταξύ τους, χωρίς να φύγουν από το κυρτό περίβλημα μέχρι όλα μαζί να αποτελούν κάποιο ενιαίο σχηματισμό και όλα να μπορούν να δουν ότι βρίσκονται σε αυτό τον ενιαίο σχηματισμό.

Για να επιτευχθεί ο σκοπός μας, ο αλγόριθμος χρησιμοποιεί κάποιες βασικές γεωμετρικές υπορουτίνες, όπως για παράδειγμα αλγόριθμο που ελέγχει αν ένα σημείο είναι πάνω στο κυρτό περίβλημα, ή αλγόριθμους που αποφασίζουν πόσα ρομπότ είναι συνδεδεμένα μαζί δημιουργώντας κάποιο σχηματισμό, πόσους σχηματισμούς από ρομπότ έχουμε, και πόση απόσταση έχει ο κάθε σχηματισμός με τον γειτονικό του.

1.4 Προηγούμενη Εργασία

Για το πρόβλημα που μελετούμε έχουν μελετηθεί στο παρελθόν και άλλα μοντέλα πέραν από αυτό που επεκτείνεται [3] στην παρούσα εργασία. Το κοινό τους σημείο είναι ότι όλα ακολουθούν ένα κοινό κύκλο λειτουργίας για τα ρομπότ, δηλαδή πρώτα το ρομπότ φωτογραφίζει την περιοχή γύρω από αυτό, μετά με βάση την εικόνα που πήρε στο προηγούμενο βήμα υπολογίζει που πρέπει να κινηθεί και τέλος κινείται. Όμως υπάρχουν πολλές διαφορές, κυρίως ως προς το στόχο του αλγόριθμου και ως προς τις παραχωρήσεις που απαιτεί το κάθε μοντέλο.

Ως προς τον στόχο του αλγορίθμου υπάρχουν οι παραλλαγές που έχουν σαν στόχο τη συγκέντρωση των ρομπότ [1, 2, 3], Δημιουργία διαφόρων γεωμετρικών σχημάτων από τα ρομπότ [4], ο αλγόριθμος να ανέχεται σφάλματα των ρομπότ [1], τα ρομπότ να χωρίζονται σε διάφορα γκρουπ [9], να υπάρχει κάποιο όριο κύκλων μέχρι να επιτευχθεί η συγκέντρωση των ρομπότ [1], τα ρομπότ να ακολουθήσουν κάποιο άλλο ρομπότ που είναι ορισμένο σαν αρχηγός [7] και τέλος να έχουμε ομαλή κατανομή των ρομπότ ώστε να δημιουργηθεί ένα σχήμα [9].

Ως προς τις παραχωρήσεις υπάρχουν οι παραλλαγές με το ασύγχρονο μοντέλο [3], σύγχρονο μοντέλο, ημι-σύγχρονο μοντέλο [10], μοντέλο όπου τα ρομπότ δεν έχουν όγκο [3], έχουν πυξίδα [8], έχουν πυξίδα που θέλει κάποιο χρόνο να σταθεροποιηθεί [8], τα ρομπότ έχουν μνήμη [5], μπορεί να γίνει πρόσβαση στην μνήμη των ρομπότ [5], τα ρομπότ έχουν απεριόριστη απόσταση ορατότητας [3], τα ρομπότ έχουν περιορισμένη απόσταση ορατότητας [2] και τέλος το πρόβλημα λύνεται μόνο για κάποιο συγκεκριμένο αριθμό ρομπότ [3].

1.5 Οργάνωση

Στο Κεφάλαιο 2 θα παρουσιαστεί το μοντέλο και οι ορισμοί. Στο Κεφάλαιο 3 θα παρουσιαστούν κάποιες χρήσιμες γεωμετρικές υπορουτίνες. Στο Κεφάλαιο 4 παρουσιάζεται ο αλγόριθμος και αποδεικνύεται η ορθότητα του. Στο Κεφάλαιο 5 περιέχονται τα συμπεράσματα μας.

Κεφάλαιο 2

Μοντέλο Και Ορισμοί

Στο κεφάλαιο αυτό θα παρουσιάσουμε αναλυτικά το μοντέλο που θα χρησιμοποιήσουμε (μαζί με τους απαραίτητους ορισμούς) και θα ορίσουμε τα προβλήματα που θα επιλύσουμε.

2.1 Γενικά

Θεωρούμε μια συλλογή από n *ρομπότ* στο (άπειρο) επίπεδο. Κάθε ρομπότ έχει σχήμα κυκλικού δίσκου με ακτίνα 1, ο οποίος είναι αδιαφανής. Θα ταυτίζουμε συχνά ένα ρομπότ με το σύνολο των σημείων του κυκλικού δίσκου του. Κάθε ρομπότ διαθέτει μια κάμερα η οποία έχει εμβέλεια ολόκληρο το επίπεδο, και η οποία είναι τοποθετημένη στην περιφέρεια του ρομπότ. Ακόμα τα *ρομπότ* δεν έχουν μνήμη από προηγούμενες καταστάσεις, ούτε κάποιο άλλο σύστημα επικοινωνίας πέρα από την κάμερα.

Μια *διάταξη* α προσδιορίζεται από τις θέσεις $\boldsymbol{C_1, C_2,...,C_n}$ των κέντρων των n ρομπότ στο επίπεδο. Επιπλέον θεωρούμε ότι τα ρομπότ έχουν όγκο, το οποίο δημιουργεί 3 σημαντικούς περιορισμούς: Πρώτον τα ρομπότ μπορεί να συγκρουστούν, δεύτερο τα ρομπότ δεν μπορούν να συγκεντρωθούν σε ένα σημείο (κάτι που μπορούσε να γίνει σε άλλα μοντέλα) και τρίτο και ποιο σημαντικό, κάποιο ρομπότ αν είναι ανάμεσα σε 2 άλλα τα εμποδίζει από το να δουν το ένα το άλλο. Το σημείο $\boldsymbol{S}$ είναι *ορατό* από το ρομπότ i στην διάταξη α αν το ευθύγραμμο τμήμα ($\boldsymbol{C_i,S}$) δεν περιέχει κανένα σημείο που ανήκει σε κάποιο άλλο ρομπότ j ≠ i. Το *στιγμιότυπο* του ρομπότ i, όπου $1 \leq i \leq n$, στη διάταξη α είναι το σύνολο όλων των σημείων που είναι ορατά από το ρομπότ i στη διάταξη α. Διαισθητικά, το στιγμιότυπο του ρομπότ i στη διάταξη α είναι το μέρος της διάταξης που το ρομπότ i μπορεί να “δει” με την κάμερα του. Δυνατόν να υπάρχουν σημεία του επιπέδου τα οποία είναι κρυμμένα πίσω από άλλα ρομπότ και επομένως το ρομπότ i δεν μπορεί να δει.

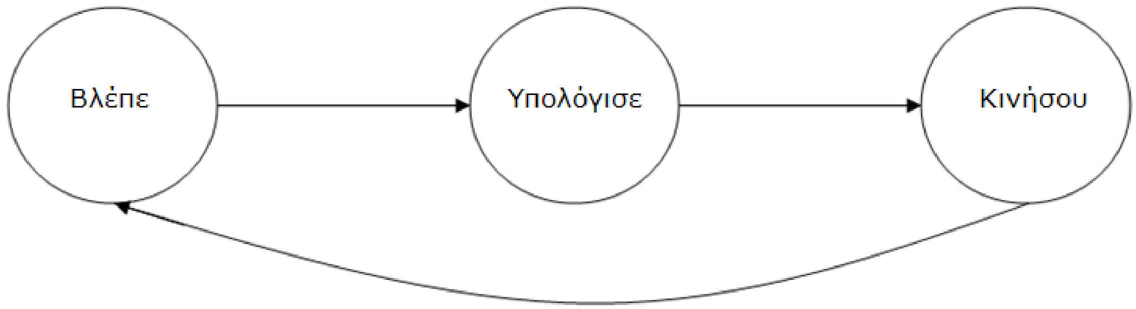
2.2 Τοπικοί Αλγόριθμοι και Κατανεμημένος Αλγόριθμος

Η αρχική διάταξη $α_0$ είναι η διάταξη στην οποία τα n ρομπότ *βρίσκονται αρχικά*. Κάθε ρομπότ όπου $1 \leq i \leq n$, υλοποιεί ένα τοπικό αλγόριθμο $A_ι$. Η εκτέλεση του αλγόριθμου $A_ι$ υποδιαιρείται σε τρείς *φάσεις,* τις Βλέπε, Υπολόγισε και Κινήσου. Το ρομπότ i διέρχεται διαδοχικά από αυτές τις φάσεις κατά κυκλικό τρόπο, όπως φαίνεται στο Σχήμα 2.1

Η εκτέλεση των φάσεων Βλέπε και Υπολόγισε γίνεται ταυτόχρονα και ακαριαία από κάποιο ρομπότ i όπου $1 \leq i \leq n$, ως ένα αδιαίρετο υπολογιστικό βήμα.

- Η είσοδος της φάσης Βλέπε είναι μια διάταξη α, και η έξοδος της είναι το στιγμιότυπο του ρομπότ i στη διάταξη α.
- Η είσοδος της φάσης Υπολόγισε είναι η έξοδος της φάσης Βλέπε που προηγείται στο παρόν υπολογιστικό βήμα. (Δηλαδή, είναι ένα στιγμιότυπο σ του ρομπότ i.) Η έξοδος της φάσης Υπολόγισε είναι η έξοδος του τοπικού αλγόριθμου A_i πάνω στην είσοδο της φάσης. (Έτσι ο αλγόριθμος A_i είναι στην ουσία ένας υπολογιστικός αλγόριθμος.) Η έξοδος του αλγόριθμου A_i είναι πάντοτε ένα σημείο **t** του επιπέδου, το οποίο προσδιορίζεται από το στιγμιότυπο σ· έτσι, θα γράφουμε $\mathbf{t} = A_i(σ)$. Αφού το στιγμιότυπο σ αναπαριστάται από τις θέσεις των κέντρων $C_1,...,C_k$ των ρομπότ 1,…,k όπου $k \leq n$, το σημείο **t** θα προσδιορίζεται ουσιαστικά από τις θέσεις αυτές των κέντρων

  Ένας κατανεμημένος αλγόριθμος είναι μια συλλογή από τοπικούς αλγόριθμους $A_1,…,A_n$.



Σχήμα 2.1: Οι φάσεις Βλέπε, Υπολόγισε και Κινήσου, και η κυκλική εναλλαγή τους

2.3 Βασικοί Ορισμοί και Περιορισμοί

Το μοντέλο που χρησιμοποιούμε είναι ασύγχρονο. Δηλαδή δεν υπάρχει κάποια συσκευή η οποία να συγχρονίζει τα ρομπότ, έτσι ώστε να ξέρουν πότε να πρέπει να αρχίζουν να βλέπουν ή πότε να αρχίζουν να κινούνται. Στο βήμα βλέπε ένα ρομπότ μπορεί να είδε ένα άλλο ρομπότ ενώ εκτελούσε το κινήσου, οπότε η θέση του να άλλαξε. Επιπρόσθετα ένα ρομπότ αν δει μέρος κάποιου άλλου ρομπότ, τότε μπορεί να υπολογίσει το κέντρο του

Στο μοντέλο αυτό έχουμε ένα εχθρό (adversary) ο οποίος λειτουργεί αυθαίρετα και έχει την δυνατότητα να επιλέξει μια από τις 3 πιο κάτω ενέργειες σαν βήμα του: α) Να ξεκινήσει την κίνηση ενός ρομπότ που είναι ακίνητο β) Να σταματήσει ένα ρομπότ που βρίσκεται εν κινήσει
γ) Να συγκρούσει 2 από τα ρομπότ που βρίσκονται σε κίνηση και μπορεί να τα συγκρούσει.

Έτσι έχουμε 2 σύνολα ρομπότ: Τα εν κινήσει (αυτά που συνεχίζουν να εκτελούν κύκλους βλέπε-υπολόγισε-κινήσου) και τα ακίνητα (αυτά που επέλεξε να σβήσει ο εχθρός). Ο εχθρός μπορεί να προκαλέσει συγκρούσεις μεταξύ των ρομπότ, αν τα ευθύγραμμα τμήματα που αντιπροσωπεύουν την πορεία των 2 ρομπότ τέμνονται, είτε σταματώντας το ένα στην πορεία του άλλου, είτε ξεκινώντας τα την κατάλληλη στιγμή ώστε να συναντηθούν πάνω στο σημείο τομής των ευθύγραμμων τμημάτων που αντιπροσωπεύουν τις πορείες τους. Έτσι όταν τα ρομπότ συγκρουστούν ξαναρχίζουν ένα νέο κύκλο βλέπε-υπολόγισε-κινήσου.

Σκοπός του εχθρού είναι να προσθέσει επιπλέον δυσκολία στο πρόβλημα, αναπαριστώντας τυχών δυσλειτουργίες των ρομπότ στα δύσκολα περιβάλλοντα όπου εργάζονται. Ο εχθρός όμως έχει ένα περιορισμό, ότι πρέπει σε κάθε βήμα κινήσου του κάθε ρομπότ που είναι στα εν κινήσει να το αφήνει να κινείται τουλάχιστο μια προκαθορισμένη απόσταση $\chi>0$ (τα ρομπότ δεν γνωρίζουν το χ)

Συνθήκη ζωτικότητας: Σε κάθε άπειρη εκτέλεση, σε κάθε επίθεμα μιας εκτέλεσης, υπάρχει τουλάχιστον ένα ξεκίνα για κάθε υπορουτίνα πριν τερματίσει ο αλγόριθμος.
Έτσι καταλήγουμε ότι: Υπολογισμός = Μια ακολουθία από υπολογιστικά βήματα του εχθρού.

2.4 Πλήρης Ορατότητα και Συσσωρευσιμότητα

Θα ορίσουμε τώρα δυο βασικές ιδιότητες που μπορεί να έχει μια διάταξη. Μια διάταξη α ικανοποιεί την πλήρη ορατότητα αν κάθε ρομπότ i, όπου $1 \leq i \leq n$, το στιγμιότυπο του ρομπότ i στη διάταξη α περιλαμβάνει τουλάχιστον ένα σημείο από κάθε άλλο ρομπότ $j \neq i$. Διαισθητικά, μια διάταξη ικανοποιεί την πλήρη ορατότητα όταν κάθε ρομπότ μπορεί να "δει" κάθε άλλο ρομπότ, και δεν υπάρχει "εμπόδιο" το οποίο αποκρύπτει κάποιο ρομπότ από κάποιο άλλο ρομπότ.
Μια διάταξη α ικανοποιεί την *συσσωρευσιμότητα* αν για οποιαδήποτε 2 ρομπότ i και j, όπου $i \neq j$ και $1 \leq i, j \leq n$, υπάρχουν 2 σημεία $\boldsymbol{S_i}$ και $\boldsymbol{S_j}$ που ανήκουν στα ρομπότ i και j αντίστοιχα, για τα οποία υπάρχει μία ακολουθία από ευθύγραμμα τμήματα με αρχή και πέρας $\boldsymbol{S_i}$ και $\boldsymbol{S_j}$, αντίστοιχα, έτσι ώστε κάθε σημείο στα ευθύγραμμα τμήματα να ανήκει σε κάποιο ρομπότ. Διαισθητικά, μια διάταξη ικανοποιεί την συσσωρευσιμότητα αν κάθε ρομπότ ακουμπάει σε κάποιο άλλο ρομπότ και όλα μαζί σχηματίζουν ένα συνεκτικό σχηματισμό.

Κεφάλαιο 3

Χρήσιμες Ακολουθίες για το Υπολόγισε

Στο κεφάλαιο αυτό παρουσιάζουμε μια συλλογή από ακολουθιακές διαδικασίες για την εκτέλεση γεωμετρικών υπολογισμών. Οι υπορουτίνες αυτές καλούνται από τον τοπικό αλγόριθμο ενός ρομπότ (Έτσι, η κάθε υπορουτίνα εκτελείται ακαριαία).

3.1 ΕίμαιΠάνωΣτοΚυρτόΠερίβλημα

Η διαδικασία ΕίμαιΠάνωΣτοΚυρτόΠερίβλημα επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΕΙΜΑΙ ΠΑΝΩ ΣΤΟ ΚΥΡΤΟ ΠΕΡΙΒΛΗΜΑ

<u>Είσοδος</u>: Μια ακολουθία από n σημεία $\mathbf{C_1, C_2, ..., C_n}$ και ένα ειδικό σημείο **C**.

<u>Απάντηση</u>: ΝΑΙ αν το **C** ανήκει στο Κυρτό Περίβλημα ($\{\mathbf{C_1, C_2, ..., C_n}\}$), αλλιώς ΟΧΙ.

Η διαδικασία ΕίμαιΠανωΣτοΚυρτοΠεριβλημα συνίσταται στον υπολογισμό του κυρτού περιβλήματος Κυρτό Περίβλημα ({$\mathbf{C_1, C_2, ..., C_n}$}) μέσω του αλγόριθμου του Graham[6] και τον έλεγχο αν το σημείο **C** είναι ένα από τα σημεία στην έξοδο του αλγόριθμου του Graham

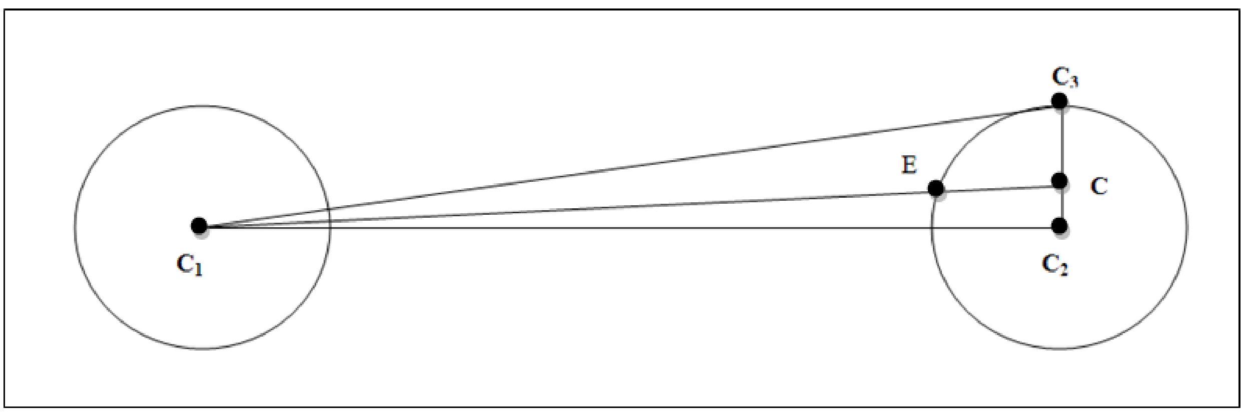
3.2 ΜετακινήσουΣτοΣημείο

Η διαδικασία ΜετακινήσουΣτοΣημείο επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΜΕΤΑΚΙΝΗΣΟΥ ΣΤΟ ΣΗΜΕΙΟ

Είσοδος: Δύο σημεία $\mathbf{C_1}$ και $\mathbf{C_2}$.

Απάντηση: Το σημείο Ε στο Σχήμα 3.1 .



Σχήμα 3.1: Το ευθύγραμμο τμήμα $\mathbf{C_2C_3}$ είναι κάθετο στο ευθύγραμμο τμήμα $\mathbf{C_1C_2}$ **.** Το σημείο **C** απέχει απόσταση $\frac{1}{n}$ από το $\mathbf{C_2}$ και το Ε είναι η τομή του ευθύγραμμου τμήματος $\mathbf{C_1C}$ και της μοναδιαίας περιφέρειας με κέντρο $\mathbf{C_2}$.

Τα σημεία $\mathbf{C_1}$ και $\mathbf{C_2}$ είναι οι τρέχουσες θέσεις του ρομπότ 1 που εκτελεί την διαδικασία ΜετακινήσουΣτοΣημείο και ενός άλλου ρομπότ 2 με το οποίο το ρομπότ 1 θέλει να επιτύχει επαφή. Το σημείο Ε είναι το σημείο προς το οποίο θα κινηθεί το ρομπότ 1. Διαισθητικά η απόσταση $\frac{1}{n}$ χρησιμοποιείται για να υποβοηθήσει ώστε το ρομπότ 1 να παραμείνει ορατό από άλλα ρομπότ και να μην "υποκρυφτεί" από το ρομπότ 2.

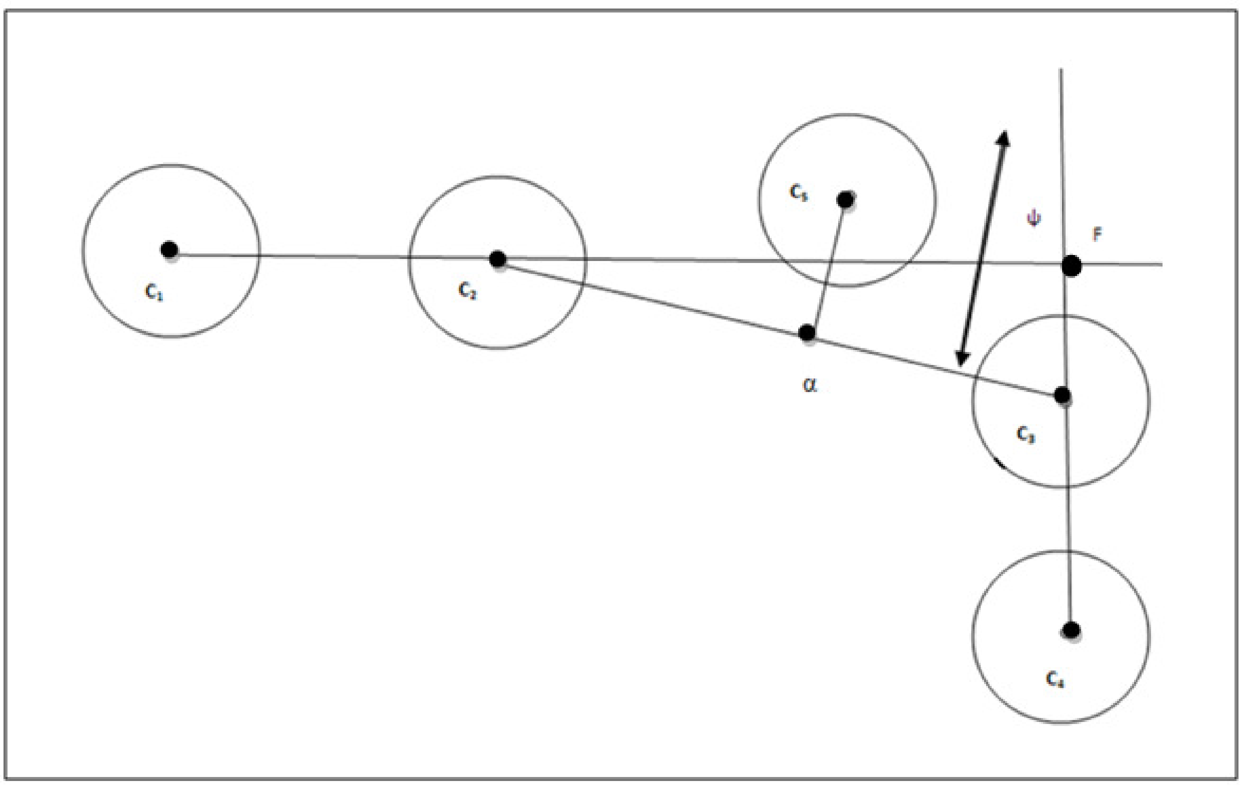
3.3 ΒρέςΤαΣημεία

Η διαδικασία ΒρέςΤαΣημεία επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΒΡΕΣ ΤΑ ΣΗΜΕΙΑ

<u>Είσοδος:</u> Μια ακολουθία από i ($i \leq n$) σημεία $\mathbf{C_1, C_2, ..., C_i}$ του κυρτού περιβλήματος.

<u>Απάντηση:</u> Μια ακολουθία από j ($j< i-1$) σημεία $\mathbf{C_1, C_2, ..., C_j}$ κάθε ένα από τα οποία ο κύκλος που το περιβάλλει με ακτίνα 1 είναι κάτω από την γραμμή $\mathbf{C_1}$F και την $\mathbf{C_4}$F .



Σχήμα 3.2: Τα $\mathbf{C_1}$, $\mathbf{C_2}$, $\mathbf{C_3}$, και $\mathbf{C_4}$ είναι ρομπότ τα οποία βρίσκονται πάνω στο κυρτό περίβλημα. Το $\mathbf{C_5}$ είναι το σημείο το οποίο ελέγχουμε αν μπορεί να μπει ρομπότ πάνω στο κυρτό περίβλημα ανάμεσα στα $\mathbf{C_2}$ και $\mathbf{C_3}$ χωρίς να προκαλέσει αλλαγή. Το α είναι το μέσο της ευθείας $\mathbf{C_2C_3}$. Η α$\mathbf{C_5}$ είναι κάθετη στη $\mathbf{C_2C_3}$ και έχει μήκος 2. Η ψ είναι η απόσταση από το α μέχρι το ποιο μακρινό από αυτή σημείο του κύκλου με ακτίνα 1 από το $\mathbf{C_5}$.

Η διαδικασία ΒρέςΤαΣημεία συνίσταται από τον έλεγχο ανάμεσα σε ποια σημεία του κυρτού περιβλήματος μπορεί να μπει ένα ρομπότ χωρίς να προκαλέσει αλλαγή. Για κάθε ένα απο τα ρομπότ τα οποία βρίσκονται πάνω στο νοητό κυρτό περίβλημα, έλεγξε τις περιπτώσεις που είναι στη θέση $\mathbf{C_2}$ και στη θέση $\mathbf{C_3}$ και το γειτονικό του αριστερά είναι το $\mathbf{C_1}$ και το γειτονικό του δεξιά είναι το $\mathbf{C_4}$ (Σύμφωνα με το σχήμα 3.2)
Αν οποιοδήποτε σημείο του κύκλου σε ακτίνα 1 απο το σημείο $\mathbf{C_5}$, τέμνεται ή βρίσκεται πιο πάνω απο τις $\mathbf{C_1}$F και $\mathbf{C_4}$F, τότε το σημείο απορρίπτεται. Διαφορετικά πρόσθεσε το σημείο αυτό ($\mathbf{C_5}$) στην ακολουθία με τα σημεία που θα επιστρέψει πίσω ο αλγόριθμος.

3.4 ΕπέστρεψεΤιςΣυγκεντρώσεις

Η διαδικασία ΕπέστρεψεΤιςΣυγκεντρώσεις επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΕΠΕΣΤΡΕΨΕ ΤΙΣ ΣΥΓΚΕΝΤΡΩΣΕΙΣ
Είσοδος: Μια ακολουθία από n σημεία $\mathbf{C_1, C_2, ..., C_n}$ που παριστάνουν τα n ρομπότ και ένα ειδικό σημείο **C**.
Απάντηση: Μια ακολουθία από i (i ≤ n) ζεύγη από σημεία**.** Για κάθε ένα από τα ζεύγη το ένα σημείο είναι το αριστερότερο ρομπότ της συγκέντρωσης και το άλλο είναι το δεξιότερο ρομπότ της συγκέντρωσης.

Μια συγκέντρωση αποτελείται από όσα ρομπότ είναι **συνδεδεμένα** (ίδιος ορισμός με το κεφάλαιο 2.2 για τη συσσωρευσιμότητα αλλά εδώ δεν είναι απαραίτητο να είναι και τα n ρομπότ στο σχηματισμό) μεταξύ τους, αλλά μπορεί να υπάρχουν μέχρι και 2 κενά(μικρότερα ίσα με απόσταση $\frac{1}{2n}$) σε μία συγκέντρωση. Αν υπάρχουν περισσότερα από 2 τέτοια κενά τότε τα θεωρούμε σαν διαφορετικές συγκεντρώσεις. Διαισθητικά μέχρι και 2 κενά μεγέθους $\frac{1}{2n}$

μπορούν να συμπεριληφθούν σε μια συγκέντρωση αφού αν ήμαστε σε κατάσταση συμμετρίας τα ρομπότ θα κινούνται με βήματα $\frac{1}{2n}$ μέχρι να συναντηθούν.

<u>Διαδικασία ΕπέστρεψεΤιςΣυγκεντρώσεις:</u>

Θέσε Αρχικό = **C**

Ξεκινώντας από το Αρχικό προχώρα δεξιά όσο υπάρχουν συνδεδεμένα ρομπότ , μέχρι να βρεις κενό(0).

A. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα δεξιά μέχρι να βρεις κενό(1).

1. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα δεξιά μέχρι να βρεις κενό(2).

I. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα δεξιά μέχρι να βρεις κενό(3). Πρόσθεσε στη ακολουθία με τις συγκεντρώσεις που θα επιστρέψεις την τελευταία συγκέντρωση (ρομπότ αριστερά στο κενό 3 και ρομπότ δεξιά στο κενό 2 και τον αριθμό ρομπότ που είναι συνδεδεμένα με αυτά τα ρομπότ συμπεριλαμβανομένου τα ακραία) Πρόσθεσε στην ακολουθία με τις συγκεντρώσεις τη συγκέντρωση(ρομπότ δεξιά στο κενό 1 και ρομπότ αριστερά στο κενό 2 καθώς και τον αριθμό ρομπότ που είναι συνδεδεμένα μαζί τους). Από το αρχικό προχώρα αριστερά μέχρι να βρεις κενό(4). Πρόσθεσε στην ακολουθία με τις συγκεντρώσεις τη συγκέντρωση (ρομπότ δεξιά στο κενό 4 και ρομπότ αριστερά στο κενό 0 καθώς και τον αριθμό των ρομπότ που είναι συνδεδεμένα μαζί τους). Πρόσθεσε στην ακολουθία με τις συγκεντρώσεις τη συγκέντρωση (ρομπότ δεξιά στο κενό 0 και ρομπότ αριστερά στο κενό 1 καθώς και τον αριθμό των ρομπότ που είναι συνδεδεμένα μαζί τους). Θέσε Αρχικό = Ρομπότ δεξιά στο κενό 3. Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα απο την ακολουθία και τερμάτισε.

II. Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε από το Αρχικό προχώρα αριστερά μέχρι να βρεις κενό(3). Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε πρόσθεσε στην ακολουθία τη συγκέντρωση (δεξιά ρομπότ του κενού 3 και αριστερά ρομπότ του κενού 2 και τον αριθμό των ρομπότ που είναι ανάμεσα σε αυτά τα ρομπότ ξεκινώντας από το δεξιά ρομπότ του κενού 3 και πηγαίνοντας δεξιόστροφα). Θέσε Αρχικό = Ρομπότ δεξιά στο κενό 2 . Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από την ακολουθία και τερμάτισε.

2. Αν το κενό μεγαλύτερο από $\frac{1}{2n}$ τότε από το Αρχικό προχώρα αριστερά μέχρι να βρεις κενό(2).

I. Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε πρόσθεσε στην ακολουθία τη συγκέντρωση (δεξιά ρομπότ του κενού 2 και αριστερά ρομπότ του κενού 1 και τον αριθμό των ρομπότ που είναι ανάμεσα σε αυτά τα ρομπότ). Θέσε Αρχικό = Ρομπότ δεξιά στο κενό 1. Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από την ακολουθία και τερμάτισε.

II. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα αριστερά μέχρι να βρεις κενό(3)

α. Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε πρόσθεσε στην ακολουθία τη συγκέντρωση (δεξιά ρομπότ του κενού 3 και αριστερά ρομπότ του κενού 1 και τον αριθμό των ρομπότ που είναι ανάμεσα σε αυτά τα ρομπότ) Θέσε Αρχικό = Ρομπότ δεξιά στο κενό 1. Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από την ακολουθία και τερμάτισε.

β. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα αριστερά μέχρι να βρεις κενό(4). πρόσθεσε στην ακολουθία τη συγκέντρωση (δεξιά ρομπότ του κενού 0 και αριστερά ρομπότ του κενού 1 και όλα τα ρομπότ που είναι ανάμεσα σε αυτά τα ρομπότ) πρόσθεσε στην ακολουθία τη συγκέντρωση (δεξιά ρομπότ του κενού 2 και αριστερά ρομπότ του κενού 0 και τον αριθμό των ρομπότ που είναι ανάμεσα σε αυτά τα ρομπότ) πρόσθεσε στην ακολουθία τη συγκέντρωση (δεξιά ρομπότ του κενού 3 και αριστερά ρομπότ του κενού 2 και όλα τα ρομπότ που είναι ανάμεσα σε αυτά τα ρομπότ) πρόσθεσε στη λίστα τη συγκέντρωση (δεξιά ρομπότ του κενού 4 και αριστερά ρομπότ του κενού 3 και τον αριθμό των ρομπότ που είναι ανάμεσα σε αυτά τα ρομπότ) Θέσε Αρχικό = Ρομπότ δεξιά στο κενό 1. Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από την ακολουθία και τερμάτισε.

Β. Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε από το αρχικό προχώρα αριστερά μέχρι να βρεις κενό(1)

1. Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε πρόσθεσε στην ακολουθία τη συγκέντρωση (αριστερά ρομπότ του κενού 0 και δεξιά ρομπότ του κενού 1 και τον αριθμό των ρομπότ που είναι ανάμεσα τους) Θέσε Αρχικό = δεξιά ρομπότ του κενού 0. Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από την ακολουθία και τερμάτισε.

2. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα αριστερά μέχρι να βρεις κενό (2).

Ι. Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε πρόσθεσε στην ακολουθία τη συγκέντρωση (αριστερά ρομπότ του κενού 0 και δεξιά ρομπότ του κενού 2 και τον αριθμό των ρομπότ που είναι ανάμεσα τους) Θέσε Αρχικό = δεξιά ρομπότ του κενού 0.

Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από τη ακολουθία και τερμάτισε.

II. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα αριστερά μέχρι να βρεις κενό(3).

α. Αν το κενό είναι μεγαλύτερο από $\frac{1}{2n}$ τότε πρόσθεσε στην ακολουθία τη συγκέντρωση (αριστερά ρομπότ του κενού 0 και δεξιά ρομπότ του κενού 3 και τον αριθμό των ρομπότ που είναι ανάμεσα τους) Θέσε Αρχικό = δεξιά ρομπότ του κενού 0. Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία (εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από την ακολουθία και τερμάτισε.

β. Αν το κενό είναι μικρότερο ή ίσο με $\frac{1}{2n}$ τότε προχώρα αριστερά μέχρι να βρεις κενό(4) πρόσθεσε στη λίστα τη συγκέντρωση (αριστερά ρομπότ του κενού 0 και δεξιά ρομπότ του κενού 1 και τον αριθμό των ρομπότ που είναι ανάμεσα τους) πρόσθεσε στην ακολουθία τη συγκέντρωση (αριστερά ρομπότ του κενού 1 και δεξιά ρομπότ του κενού 2 και τον αριθμό των ρομπότ που είναι ανάμεσα τους) πρόσθεσε στην ακολουθία τη συγκέντρωση (αριστερά ρομπότ του κενού 2 και δεξιά ρομπότ του κενού 3 και τον αριθμό των ρομπότ που είναι ανάμεσα τους) πρόσθεσε στην ακολουθία τη συγκέντρωση (αριστερά ρομπότ του κενού 3 και δεξιά ρομπότ του κενού 4 και τον αριθμό των ρομπότ που είναι ανάμεσα τους) Θέσε Αρχικό = δεξιά ρομπότ του κενού 0. Έλεγξε αν σε κάποια από τις συγκεντρώσεις που μόλις πρόσθεσες υπάρχει το ρομπότ που δόθηκε σαν input στη διαδικασία(εκτός και αν είναι η πρώτη εκτέλεση). Αν ναι αφαίρεσε τα διπλότυπα από την ακολουθία και τερμάτισε

3.5 ΠόσηΑπόστασηΈχεις

Η διαδικασία ΠόσηΑπόστασηΈχεις επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΠΟΣΗ ΑΠΟΣΤΑΣΗ ΕΧΕΙΣ

Είσοδος: Μια ακολουθία από n σημεία $\mathbf{C_1}$, $\mathbf{C_2}$, ..., $\mathbf{C_n}$ που παριστάνουν τα n ρομπότ και ένα ειδικό σημείο **C**.

Απάντηση: Ένας αριθμός μεταξύ 1,2 και 3. Αν το ειδικό σημείο είναι το δεξιότερο της συγκέντρωσης που έχει την μικρότερη απόσταση για κενά μεταξύ συγκεντρώσεων, τότε επιστρέφει 1. Αν όλες οι αποστάσεις είναι ίσες τότε επιστρέφει 2. Διαφορετικά επιστρέφει 3.

Η διαδικασία ΠόσηΑπόστασηΈχεις χρησιμοποιεί την διαδικασία ΕπέστρεψεΤιςΣυγκεντρώσεις δίνοντας σαν είσοδο τα n σημεία των ρομπότ και το σημείο του ρομπότ που τρέχει τον αλγόριθμο. Ελέγχει τις αποστάσεις μεταξύ των συγκεντρώσεων που επιστράφηκαν και ανάλογα επιστρέφει 1, 2 ή 3.

3.6 ΕίσαιΣτηΜεγαλύτερηΣυγκέντρωση

Η διαδικασία ΕίσαιΣτηΜεγαλύτερηΣυγκέντρωση επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΕΙΣΑΙ ΣΤΗ ΜΕΓΑΛΥΤΕΡΗ ΣΥΓΚΕΝΤΡΩΣΗ

Είσοδος: Μια ακολουθία από n σημεία $\mathbf{C_1}$, $\mathbf{C_2}$, ..., $\mathbf{C_n}$ που παριστάνουν τα n ρομπότ και ένα ειδικό σημείο **C**.

Απάντηση: Ένας αριθμός μεταξύ 1,2 και 3. Αν το ειδικό σημείο βρίσκεται στην μεγαλύτερη συγκέντρωση, τότε επιστρέφει 1, Αν όλες οι συγκεντρώσεις είναι μεγαλύτερες, τότε επιστρέφει 2. Αν δεν βρίσκεται στη μεγαλύτερη συγκέντρωση επιστρέφει 3.

Η διαδικασία ΕίσαιΣτηΜεγαλύτερηΣυγκέντρωση χρησιμοποιεί την διαδικασία ΕπέστρεψεΤιςΣυγκεντρώσεις δίνοντας σαν είσοδο τα n σημεία των ρομπότ και το σημείο του ρομπότ που τρέχει τον αλγόριθμο. Ελέγχει πόσα ρομπότ αποτελούν κάθε μια από τις συγκεντρώσεις που επιστράφηκαν.

3.7 ΕίσαιΣτηΜικρότερηΣυγκέντρωση

Η διαδικασία ΕίσαιΣτηΜικρότερηΣυγκέντρωση επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΕΙΣΑΙ ΣΤΗ ΜΙΚΡΟΤΕΡΗ ΣΥΓΚΕΝΤΡΩΣΗ

Είσοδος: Μια ακολουθία από n σημεία $\mathbf{C_1}$**,** $\mathbf{C_2}$**, ...,** $\mathbf{C_n}$ που παριστάνουν τα n ρομπότ και ένα ειδικό σημείο **C**.

Απάντηση: Ένας αριθμός μεταξύ 1,2 και 3. Αν το ειδικό σημείο βρίσκεται στην μικρότερη συγκέντρωση, τότε επιστρέφει 1. Αν το ειδικό σημείο δεν βρίσκεται στη μικρότερη τότε επιστρέφει 2. Αν όλες οι συγκεντρώσεις είναι μικρότερες τότε επιστρέφει 3.

Η διαδικασία ΕίσαιΣτηΜικρότερηΣυγκέντρωση χρησιμοποιεί την διαδικασία ΕπέστρεψεΤιςΣυγκεντρώσεις δίνοντας σαν είσοδο τα n σημεία των ρομπότ και το σημείο του ρομπότ που τρέχει τον αλγόριθμο. Ελέγχει πόσα ρομπότ αποτελούν κάθε μια από τις συγκεντρώσεις που επιστράφηκαν.

Κεφάλαιο 4

Αλγόριθμος Και Απόδειξη

4.1 Περίληψη Διαδικασιών

Δηλώνουμε τις πιο κάτω 17 καταστάσεις οι οποίες φαίνονται και στο σχήμα 4.1:

0.Αρχή:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος ξεκινά να τρέχει τον αλγόριθμο αυτό.

1.Πάνω Στο Κυρτό περίβλημα:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος είναι πάνω στο νοητό Κυρτό περίβλημα.

2. Όχι πάνω στο Κυρτό περίβλημα:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος είναι μέσα στο νοητό Κυρτό περίβλημα.

3. Πλήρης Ορατότητα:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος είναι πάνω στο νοητό Κυρτό περίβλημα.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορεί να δει όλα τα υπόλοιπα n-1 ρομπότ.
- Όλα τα υπόλοιπα n-1 ρομπότ βρίσκονται πάνω στο κυρτό περίβλημα.

4.Συνδεδεμένα:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 3.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορεί να δεί ότι όλα τα ρομπότ είναι συνδεδεμένα μεταξύ τους.

5.Όχι Συνδεδεμένα:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 3
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορεί να δεί ότι όλα τα ρομπότ δεν είναι συνδεδεμένα μεταξύ τους.

6.Όχι Πλήρης Ορατότητα:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος είναι πάνω στο νοητό Κυρτό περίβλημα.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος δεν μπορεί να δει όλα τα υπόλοιπα n-1 ρομπότ, ή δεν είναι όλα τα υπόλοιπα n-1 ρομπότ πάνω στο κυρτό περίβλημα.

7.Δεν υπάρχει χώρος για άλλους:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 6.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορεί να δει ότι δεν υπάρχει χώρος πάνω στο νοητό Κυρτό περίβλημα(Η μέγιστη απόσταση μεταξύ 2 γειτονικών ρομπότ που βρίσκονται πάνω στο Κυρτό περίβλημα είναι μικρότερη απο 2, όπου 1 η ακτίνα του ρομπότ δίσκου).

8.Υπάρχει Χώρος για άλλους:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 6.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορεί να δει ότι υπάρχει χώρος πάνω στο νοητό Κυρτό περίβλημα(Υπάρχουν 2 γειτονικά ρομπότ πάνω στο κυρτό περίβλημα των οποίων η απόσταση τους είναι μεγαλύτερη απο 2).

9.Δεν είναι σε Ευθεία :

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 8.
- Κανένα άλλο ρομπότ δεν είναι στην ίδια ευθεία με το ρομπότ στο οποίο τρέχει ο αλγόριθμος.

10. Σε ευθεία:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 8.
- Κάποιο άλλο ρομπότ είναι στην ίδια ευθεία με το ρομπότ στο οποίο τρέχει ο αλγόριθμος.

11.Βλέπει 1 ρομπότ:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 10.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορεί να δει μόνο 1 ρομπότ στην ευθεία του.

12.Βλέπει 2 ρομπότ:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση10.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορεί να δει 2 ρομπότ στην ίδια ευθεία με αυτό έτσι βρίσκεται ανάμεσα στα 2 αυτά ρομπότ.

13.Κολλημένο:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 2.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος αγγίζει κάποιο άλλο ρομπότ.

14.Όχι κολλημένο:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 2.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος δεν αγγίζει κανένα άλλο ρομπότ.

15.Θα προκαλέσει αλλαγή:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 14.
- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος αν κινηθεί σύμφωνα με τον αλγόριθμο θα προκαλέσει αλλαγή στο νοητό Κυρτό περίβλημα και δέν υπάρχει τρόπος να κινηθεί σύμφωνα με τον αλγόριθμο και να μην προκληθεί αλλαγή.

16.Δεν θα προκαλέσει αλλαγή:

- Το ρομπότ στο οποίο τρέχει ο αλγόριθμος βρίσκεται στην κατάσταση 14.
- Υπάρχουν τρόποι ώστε Το ρομπότ στο οποίο τρέχει ο αλγόριθμος να κινηθεί όπως ορίζει ο αλγόριθμος και να μην προκληθούν αλλαγές στο Κυρτό περίβλημα.

<u>Λήμμα 4.1:</u>

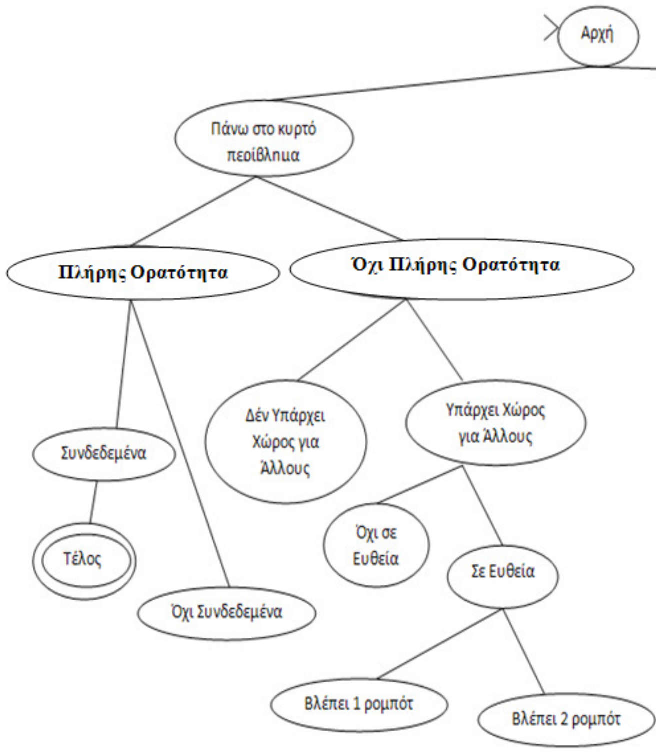
Οι καταστάσεις 0-16 είναι όλες οι πιθανές καταστάσεις(με βάση την εικόνα από το βλέπε) στις οποίες μπορεί να περάσει ένα ρομπότ.

<u>Απόδειξη:</u> Είναι εύκολο να δούμε ότι για κάθε κατάσταση μπορούμε να μεταβούμε σε 2 άλλες καταστάσεις οι οποίες η μια είναι το συμπλήρωμα τις άλλης, και ξεκινώντας απο την αρχική κατάσταση βλέπουμε ότι καλύπτονται όλες οι πιθανές καταστάσεις.

Ο αλγόριθμος αποτελείται απο 17 διαδικασίες, οι οποίες, η κάθε μια απο αυτές αντιμετωπίζει μία απο τις πιθανές καταστάσεις του αλγορίθμου και μπορεί να δηλωθεί ως ακολούθως:

Αλγόριθμος για n ρομπότ:

If Κατάσταση== i **do** TREAT κατάσταση i



Σχήμα 4.1: Στο σχήμα αυτό φαίνονται όλες οι πιθανές καταστάσεις του αλγορίθμου και οι μεταβάσεις τους. Όσες καταστάσεις δεν έχουν μετάβαση σε κάποια άλλη κατάσταση είναι τελικές, δηλαδή αποφασίζουν που θα κινηθεί το ρομπότ.

4.2 Περιγραφή Υποδιαδικασιών

Το υποκεφάλαιο αυτό έχει 17 μέρη, κάθε ένα από τα οποία περιγράφει μια διαδικασία

4.2.1 Αρχή

Διαδικασία Αρχή:

Αν βρίσκεσαι στην κατάσταση 0, κάλεσε την διαδικασία ΕίμαιΠάνωΣτοΚυρτόΠερίβλημα με είσοδο το σημείο στο οποίο βρίσκεσαι και τα σημεία στα οποία βρίσκονται όλα τα ρομπότ. Αν επιστρέψει ναι μετακινήσου στην κατάσταση ΠάνωΣτοΚυρτόΠερίβλημα, διαφορετικά μετακινήσου στην κατάσταση ΌχιΠάνωΣτοΚυρτόΠερίβλημα .

Λήμμα 4.2:

Ο αλγόριθμος Αρχή είναι ένας σωστός αλγόριθμος ο οποίος μπορεί να ξεχωρίσει αν κάποιο ρομπότ είναι πάνω στο κυρτό περίβλημα η όχι.

Απόδειξη: Απο τον αλγόριθμο του Graham μπορούμε να διαχωρίσουμε αν το ρομπότ ανήκει στο νοητό κυρτό περίβλημα ή όχι.

4.2.2 ΠάνωΣτοΚυρτόΠερίβλημα

Η διαδικασία ΠάνωΣτοΚυρτόΠερίβλημα επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΠΑΝΩ ΣΤΟ ΚΥΡΤΟ ΠΕΡΙΒΛΗΜΑ

Είσοδος: ΜΙΑ ακολουθία από n σημεία $\mathbf{C_1, C_2, ..., C_n}$.

Απάντηση: ΝΑΙ αν όλα τα ρομπότ βρίσκονται πάνω στο κυρτό περίβλημα, αλλιώς ΟΧΙ.

Διαδικασία ΠάνωΣτοΚυρτόΠερίβλημα:

- Εάν μπορείς να δεις όλα τα υπόλοιπα n-1 ρομπότ (να σχεδιάσεις μία νοητή γραμμή απο εσένα προς κάθε ένα απο τα υπόλοιπα ρομπότ χωρίς να υπάρχει κάποιο άλλο ρομπότ ενδιάμεσα) για κάθε ένα από τα υπόλοιπα n-1 ρομπότ κάλεσε την διαδικασία ΕίμαιΠάνωΣτοΚυρτόΠερίβλημα αν η διαδικασία αυτή απαντήσει ναι για όλα τότε μετακινήσου στην κατάσταση ΠλήρηςΟρατότητα.
- Διαφορετικά μετακινήσου στην κατάσταση ΌχιΠλήρηςΟρατότητα .

Λήμμα 4.3:

Ο αλγόριθμος ΠάνωΣτοΚυρτόΠερίβλημα, είναι ένας σωστός αλγόριθμος ο οποίος μπορεί να καταλάβει αν ένα ρομπότ έχει *Πλήρη Ορατότητα* και όλα τα υπόλοιπα έχουν *Πλήρη Ορατότητα* ή όχι.

Απόδειξη:

Δεδομένου ότι ο μπορούμε να σχεδιάσουμε μια νοητή γραμμή απο το ρομπότ στο οποίο τρέχει ο αλγόριθμος μπορούμε να αποφανθούμε με ακρίβεια αν μπορούμε να δούμε όλα τα υπόλοιπα ρομπότ. Δεδομένου ότι μπορούμε να δούμε όλα τα υπόλοιπα ρομπότ μπορούμε να τρέξουμε την διαδικασία ΕίμαιΣτοΚυρτόΠερίβλημα για κάθε ένα απο τα υπόλοιπα ρομπότ και να μας δώσει σωστά αποτελέσματα αν τα υπόλοιπα ρομπότ είναι πάνω στο νοητό κυρτό περίβλημα ή όχι. Αν όλα τα ρομπότ είναι πάνω στο νοητό κυρτό περίβλημα τότε αυτόματα σημαίνει πως όλα μπορούν να δουν όλα τα υπόλοιπα ρομπότ (απο τον ορισμό του κυρτού περιβλήματος για μη ύπαρξη εσοχών).

4.2.3 ΠλήρηςΟρατότητα

Η διαδικασία ΠλήρηςΟρατότητα επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΠΛΗΡΗΣ ΟΡΑΤΟΤΗΤΑ

Είσοδος: ΜΙΑ ακολουθία από n σημεία $C_1, C_2, ..., C_n$.

Απάντηση: ΝΑΙ αν όλα τα ρομπότ είναι *συσσωρευμένα*, αλλιώς ΟΧΙ.

Διαδικασία ΠλήρηςΟρατότητα:

- Πιάσε τυχαία ένα ρομπότ. Βάλε αυτό το ρομπότ στη λίστα με το όνομα σχηματισμός
- Για κάθε ένα απο τα ρομπότ που είναι στη λίστα σχηματισμός όσα ρομπότ εφάπτονται με αυτό και δεν είναι στη λίστα σχηματισμός, πρόσθεσε το ρομπότ στη λίστα σχηματισμός
- Επανέλαβε την πιο πάνω διαδικασία μέχρι να μην γίνουν αλλαγές στη λίστα σχηματισμός
- Μέτρησε τα ρομπότ που είναι μέσα στη λίστα σχηματισμός
- Αν τα ρομπότ που βρίσκονται μέσα στη λίστα σχηματισμός είναι n τότε μετακινήσου στην κατάσταση Συνδεδεμένα, διαφορετικά μετακινήσου στην κατάσταση ΌχιΣυνδεδεμένα.

Λήμμα 4.4:

Η Διαδικασία ΠλήρηςΟρατότητα είναι ένας σωστός αλγόριθμος με τον οποίο μπορούμε να διαχωρίσουμε αν όλα τα ρομπότ είναι συνδεδεμένα ή όχι

Απόδειξη:

Οι καταστάσεις στις οποίες μεταβαίνει ο αλγόριθμος είναι 2 και η μία είναι η συμπληρωματική της άλλης επομένως καλύπτονται όλες οι πιθανές περιπτώσεις σε αυτή την κατάσταση. Με τα 3 πρώτα βήματα της διαδικασίας αυτής πετυχαίνουμε για κάποιο τυχαίο ρομπότ να βρούμε όλα τα ρομπότ τα οποία είναι συνδεδεμένα με αυτό σχηματίζοντας κάποιο σχηματισμό. Αν τα ρομπότ που υπάρχουν στη λίστα σχηματισμός είναι ίσα με τον αριθμό όλων των ρομπότ τότε όλα είναι συνδεδεμένα, διαφορετικά θα σημαίνει ότι έχουμε τουλάχιστο 2 διαφορετικούς σχηματισμούς, οπότε δεν είναι συνδεδεμένα.

4.2.4 ΌχιΠλήρηςΟρατότητα

Η διαδικασία ΟχιΠλήρηςΟρατότητα επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΟΧΙ ΠΛΗΡΗΣ ΟΡΑΤΟΤΗΤΑ

Είσοδος: MIA ακολουθία από n σημεία $\mathbf{C_1, C_2, ..., C_n}$ που βρίσκονται πάνω στο κυρτό περίβλημα.

Απάντηση: ΝΑΙ αν υπάρχει κάποιο κενό πάνω στο κυρτό περίβλημα ≥ 2, αλλιώς ΟΧΙ.

Διαδικασία ΌχιΠλήρηςΟρατότητα:

- Για κάθε μια απο τις πλευρές του νοητού κυρτού περιβλήματος έλεγξε αν υπάρχει κάποια πλευρά η οποία να έχει μήκος τουλάχιστο ίσο με 2 .
- Αν υπάρχει μια τέτοια πλευρά, τότε μετακινήσου στην κατάσταση ΥπάρχειΧώροςΓιαΆλλους, διαφορετικά μετακινήσου στην κατάσταση ΔενΥπάρχειΧώροςΓιαΆλλους.

Λήμμα 4.5:

Η Διαδικασία ΌχιΠλήρηςΟρατότητα είναι ένας σωστός αλγόριθμος με τον οποίο μπορούμε να διαχωρίσουμε αν υπάρχει διαθέσιμος χώρος για κάποιο ρομπότ για να μπει πάνω στο νοητό κυρτό περίβλημα ή όχι.

Απόδειξη:

Οι καταστάσεις στις οποίες μεταβαίνει ο αλγόριθμος είναι 2 και η μία είναι η συμπληρωματική της άλλης επομένως καλύπτονται όλες οι πιθανές περιπτώσεις σε αυτή την κατάσταση. Ό αλγόριθμος ελέγχει όλες τις πλευρές του νοητού κυρτού περιβλήματος, επομένως αν υπάρχει κάποια μεγαλύτερη απο 2 τότε θα την βρει, αν δέν υπάρχει τέτοια πλευρά, τότε όταν θα τελειώσουν οι πλευρές του νοητού κυρτού περιβλήματος, ο αλγόριθμος θα αναφέρει ότι δεν υπάρχει χώρος.

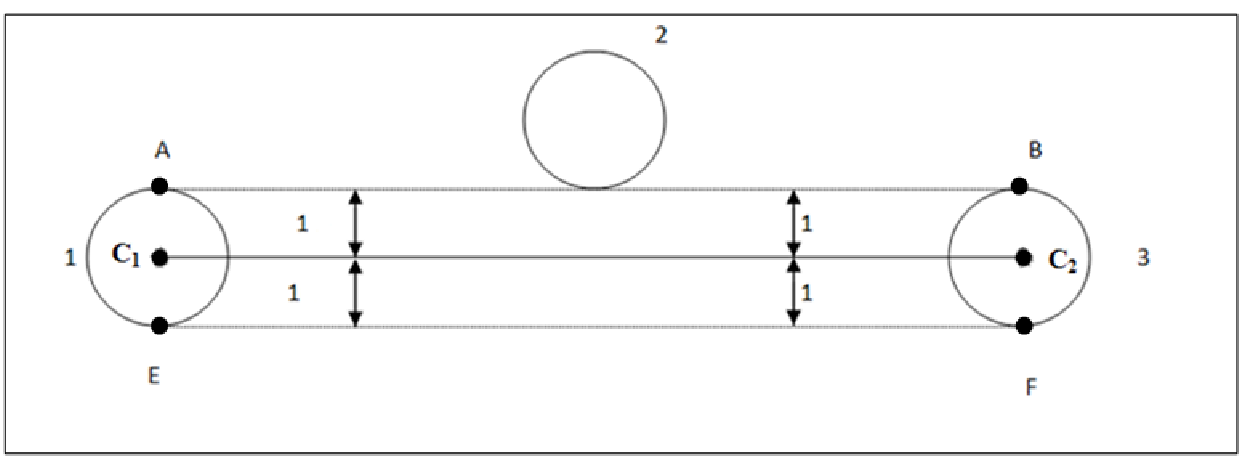
4.2.5 ΥπάρχειΧώροςΓιαΑλλους

Η διαδικασία ΥπάρχειΧώροςΓιαΆλλους επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΥΠΑΡΧΕΙ ΧΩΡΟΣ ΓΙΑ ΑΛΛΟΥΣ

Είσοδος: Μια ακολουθία από n σημεία $\mathbf{C_1, C_2, ..., C_n}$ που βρίσκονται πάνω στο κυρτό περίβλημα και ένα ειδικό σημείο **C**.

Απάντηση: ΝΑΙ αν το σημείο **C** είναι σε ευθεία, αλλιώς ΟΧΙ.



Σχήμα 4.2: Στο σχήμα αυτό τα ρομπότ 1, 2 και 3 είναι γειτονικά πάνω στο κυρτό περίβλημα, τα $\mathbf{C_1}$, $\mathbf{C_2}$ τα κέντρα τον ρομπότ 1 και 3 αντίστοιχα και η AE είναι κάθετη πάνω στην $\mathbf{C_1C_2}$ και η BF κάθετη πάνω στην $\mathbf{C_1C_2}$

Διαδικασία ΥπάρχειΧώροςΓιαΑλλους:

- Έλεγξε τις διάφορες περιπτώσεις με τα γειτονικά σου ρομπότ πάνω στο νοητό κυρτό περίβλημα για το Σχήμα 4.2, ποιές απο αυτές μπορεί να ισχύουν (δηλαδή π.χ. εσύ να είσαι το ρομπότ 1, ο γείτονας σου στα δεξιά το ρομπότ 2 και ο άλλος γείτονας του γείτονα σου το ρομπότ 3, παρόμοια να είσαι το ρομπότ 2 και παρόμοια το ρομπότ 3)
- Αν για κάποια απο τις ποιό πάνω πιθανές περιπτώσεις υπάρχει κάποιο σημείο του ρομπότ 2 που να είναι μέσα στο τετράγωνο A,B,E,F τότε μετακινήσου στην κατάσταση ΕίσαιΣεΕυθεία, διαφορετικά μετακινήσου στην κατάσταση ΔενΕίσαιΣεΕυθεία.

4.2.6 ΕίσαιΣεΕυθεία

Η διαδικασία ΕίσαιΣεΕυθεία επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΕΙΣΑΙ ΣΕ ΕΥΘΕΙΑ

Είσοδος: Μια ακολουθία από n σημεία $\mathbf{C_1, C_2, ..., C_n}$ που βρίσκονται πάνω στο κυρτό περίβλημα και ένα ειδικό σημείο **C**.

Απάντηση: ΝΑΙ αν το σημείο **C** είναι ανάμεσα σε 2 σημεία στην ίδια ευθεία, αλλιώς ΟΧΙ.

Διαδικασία ΕίσαιΣεΕυθεία:

- Έλεγξε τις διάφορες περιπτώσεις με τα γειτονικά σου ρομπότ πάνω στο νοητό κυρτό περίβλημα σύμφωνα με το Σχήμα 4.2, ποιές απο αυτές μπορεί να ισχύουν (δηλαδή π.χ. εσύ να είσαι το ρομπότ 1, ο γείτονας σου στα δεξιά το ρομπότ 2 και ο άλλος γείτονας του γείτονα σου το ρομπότ 3.
- Για τις περιπτώσεις στις οποίες το ρομπότ 2 βρίσκεται μέσα στο τετράγωνο A,B,E,F και αν σε τουλάχιστο μία απο αυτές τις περιπτώσεις είσαι το ρομπότ 2, τότε μετακινήσου στην κατάσταση ΒλέπειςΔύοΡομπότ, διαφορετικά μετακινήσου στην κατάσταση ΒλέπειςΈναΡομπότ.

4.2.7 ΌχιΠάνωΣτοΚυρτόΠερίβλημα

Η διαδικασία ΌχιΠάνωΣτοΚυρτόΠερίβλημα επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΟΧΙ ΠΑΝΩ ΣΤΟ ΚΥΡΤΟ ΠΕΡΙΒΛΗΜΑ

Είσοδος: ΜΙΑ ακολουθία από n σημεία $\mathbf{C_1, C_2, ..., C_n}$ που βρίσκονται πάνω στο κυρτό περίβλημα και ένα ειδικό σημείο **C**.

Απάντηση: ΝΑΙ αν το σημείο **C** εφάπτεται με κάποιο άλλο σημείο, αλλιώς ΟΧΙ.

Αν κάποιο απο τα σημεία της περιφέρειας σου εφάπτεται με κάποιο σημείο κάποιου άλλου ρομπότ, τότε μετακινήσου στην κατάσταση Κολλημένο, διαφορετικά μετακινήσου στην κατάσταση ΌχιΚολλημένο.

4.2.8 ΌχιΚολλημένο

Η διαδικασία ΌχιΚολλημένο επιλύει το εξής (ακολουθιακό) πρόβλημα απόφασης:

ΟΧΙ ΚΟΛΛΗΜΕΝΟ

Είσοδος: ΜΙΑ ακολουθία από n σημεία $\mathbf{C_1}$, $\mathbf{C_2}$, ..., $\mathbf{C_n}$ που βρίσκονται πάνω στο κυρτό περίβλημα και ένα ειδικό σημείο **C**.

Απάντηση: ΝΑΙ αν πρέπει να προκληθεί αλλαγή στο κυρτό περίβλημα για να μπει πάνω το ρομπότ που βρίσκεται στο **C** , αλλιώς ΟΧΙ.

Διαδικασία ΌχιΚολλημένο:

Έλεγξε το κυρτό περίβλημα αν υπάρχει κάποιο σημείο στο οποίο μπορείς να μεταβείς σύμφωνα με την ακολουθία ΒρεςΤαΣημεία χωρίς να προκαλέσεις αλλαγές σε κάποιο απο τα ρομπότ τα οποία ήδη βρίσκονται πάνω στο κυρτό περίβλημα. Άν υπάρχει τουλάχιστο ένα τέτοιο σημείο, τότε μετακινήσου στην κατάσταση ΔενΘαΠροκαλέσειΑλλαγή, διαφορετικά μετακινήσου στην κατάσταση ΘαΠροκαλέσειΑλλαγή.

4.2.9 Συνδεδεμένα

Διαδικασία Συνδεδεμένα:

Αν είσαι στην διαδικασία συνδεδεμένα τότε το πρόβλημα έχει λυθεί, το νέο σημείο στο οποίο θα κινηθείς στον κύκλο κινήσου είναι το ίδιο με το σημείο στο οποίο βρίσκεσαι τώρα.

4.2.10 ΌχιΣυνδεδεμένα

Διαδικασία ΌχιΣυνδεδεμένα:

Α. Αν έχεις δεξιά σου και αριστερά σου ρομπότ τα οποία εφάπτονται σε εσένα, τότε το νέο σημείο στο οποίο θα κινηθείς είναι το ίδιο με το σημείο στο οποίο βρίσκεσαι τώρα.

Β. Διαφορετικά Κάλεσε την διαδικασία ΕίσαιΣτηΜεγαλύτερηΣυγκέντρωση.

1. Αν επιστρέψει 1 τότε το νέο σημείο στο οποίο θα κινηθείς είναι το ίδιο με το σημείο στο οποίο βρίσκεσαι τώρα.
2. Αν επιστρέψει 2 τότε κάλεσε την διαδικασία Πόση απόσταση απέχεις.

Ι. Αν επιστέψει 1 τότε κάλεσε την διαδικασία ΜετακινήθουΣτοΣημείο στην οποία θα δώσεις τις συντεταγμένες του σημείου στο οποίο είσαι τώρα και το σημείο που είναι ο γείτονας σου πάνω στο κυρτό περίβλημα στα δεξιά.

ΙΙ. Αν επιστρέψει 2, τότε Κάλεσε την διαδικασία ΕπέστρεψεΤιςΣυγκεντρώσεις, φέρε μία ευθεία γραμμή από τα κέντρα τον ρομπότ που είναι στα άκρα της συγκέντρωσης σου(ΑΒ). Φέρε παράλληλη την ΓΔ προς αυτή την γραμμή(ΑΒ) να περνά από το κέντρο σου (Γ), φέρε κάθετη στη ΓΔ από το σημείο Γ με κατεύθυνση μέσα στο νοητό κυρτό περίβλημα με απόσταση $\frac{1}{2n}$. Θα κινηθείς μόνο εάν δεν θα συγκρουστείς με άλλο της συγκέντρωσης σου ή αν είσαι στα άκρα της συγκέντρωσης νοουμένου ότι στη συγκέντρωση σου δεν υπάρχει κενό. Το νέο σημείο στο οποίο θα κινηθείς είναι η άκρη της κάθετης(το Δ) Διαφορετικά το νέο σημείο είναι το ίδιο με αυτό που είσαι τώρα.

ΙΙΙ. Διαφορετικά το νέο σημείο στο οποίο θα κινηθείς είναι το ίδιο με το σημείο στο οποίο βρίσκεσαι τώρα.

3. Αν επιστρέψει 3 Κάλεσε την διαδικασία ΕίσαιΣτηΜικρότερηΣυγκέντρωση

I. Αν επιστρέψει 1 τότε κάλεσε την διαδικασία ΜετακινήθουΣτοΣημείο στην οποία θα δώσεις τις συντεταγμένες του σημείου στο οποίο είσαι τώρα και το σημείο που είναι ο γείτονας σου πάνω στο κυρτό περίβλημα στα δεξιά.

II. Αν επιστρέψει 2 τότε το νέο σημείο στο οποίο θα κινηθείς είναι το ίδιο με το σημείο στο οποίο βρίσκεσαι τώρα.

III. Αν επιστρέψει 3 τότε τότε κάλεσε την διαδικασία ΠόσηΑπόστασηΑπέχεις.

α. Αν επιστέψει 1 τότε κάλεσε την διαδικασία ΜετακινήθουΣτοΣημείο στην οποία θα δώσεις τις συντεταγμένες του σημείου στο οποίο είσαι τώρα και το σημείο που είναι ο γείτονας σου πάνω στο κυρτόπερίβλημα στα δεξιά

β. Αν επιστρέψει 2 τότε φέρε μία γραμμή από τα κέντρα τον ρομπότ που είναι στα άκρα της συγκέντρωσης σου(ΑΒ). Φέρε παράλληλη ΓΔ προς αυτή την γραμμή(ΑΒ) να περνά από το κέντρο σου(Γ), Φέρε κάθετη στη ΓΔ από το σημείο Γ με κατεύθυνση μέσα στο νοητό κυρτό περίβλημα με απόσταση $\frac{1}{2n}$. Το νέο σημείο στο οποίο θα κινηθείς είναι η άκρη της κάθετης(Το Δ) Αν είσαι ένα ρομπότ μόνο σου τότε τράβα γραμμή από τα κέντρα τον γειτονικών σου ρομπότ (ΑΒ) φέρε γραμμή ΓΔ από το κέντρο σου στη μέση της ΑΒ (Γ το κέντρο σου) Το νέο σημείο στο οποίο θα κινηθείς είναι σε απόσταση $\frac{1}{2n}$ από το Γ πάνω στην ΓΔ.

γ. Διαφορετικά το νέο σημείο στο οποίο θα κινηθείς είναι το ίδιο με το σημείο στο οποίο βρίσκεσαι τώρα.

Λήμμα 4.6:

Η διαδικασία ΌχιΣυνδεδεμένα δεν θα μας μεταφέρει στην κατάσταση ΌχιΠλήρηςΟρατότητα ποτέ και θα οδηγήσει στη συγκέντρωση όλων των ρομπότ.

Απόδειξη:

Στην περίπτωση που υπάρχουν *συγκεντρώσεις* διαφορετικού μεγέθους, πάντα θα κινείται ρομπότ το οποίο βρίσκεται στην μικρότερη *συγκέντρωση* οπότε τελικά θα οδηγηθούμε στο να υπάρχει μόνο μία συγκέντρωση ρομπότ, το *Πλήρης Ορατότητα* διατηρείται αφού τα ρομπότ κινούνται πάνω στο νοητό κυρτό περίβλημα του οποίου οι ιδιότητες μας διασφαλίζουν ότι όλα τα σημεία του έχουν ορατότητα προς όλα τα υπόλοιπα σημεία του. Αν υπάρχουν μόνο συγκεντρώσεις με το ίδιο μέγεθος τότε θα κινούνται ρομπότ μόνο από την συγκέντρωση η οποία το δεξιότερο της ρομπότ έχει την κοντινότερη απόσταση από το αριστερότερο ρομπότ της γειτονικής του συγκέντρωσης(στα δεξιά) Αυτό μας διασφαλίζει ότι θα υπάρξει κάποια συγκέντρωση μεγαλύτερη από τις άλλες, έτσι θα ξεφύγουμε από αυτή την κατάσταση και θα ισχύσει η πρώτη περίπτωση. Στην τελευταία περίπτωση, όπου δηλαδή έχουμε ίσες συγκεντρώσεις με ίσες αποστάσεις, αυτό σημαίνει ότι οι συγκεντρώσεις μας είναι συμμετρικές. Αφού είναι συμμετρικές τις βάζουμε να πλησιάσουν με μικρά βήματα ώστε να μην χαλάσει το νοητό κυρτό περίβλημα. Αυτό μας διασφαλίζει είτε ότι θα έχουμε διαφορετικές αποστάσεις οπότε θα ισχύσει η δεύτερη περίπτωση, είτε στο ότι τελικά όλοι οι σχηματισμοί θα ακουμπήσουν δημιουργώντας ένα *ενιαίο σχηματισμό*.

4.2.11 ΔενΥπάρχειΧώροςΓιαΆλλους

Διαδικασία ΔενΥπάρχειΧώροςΓιαΆλλους:

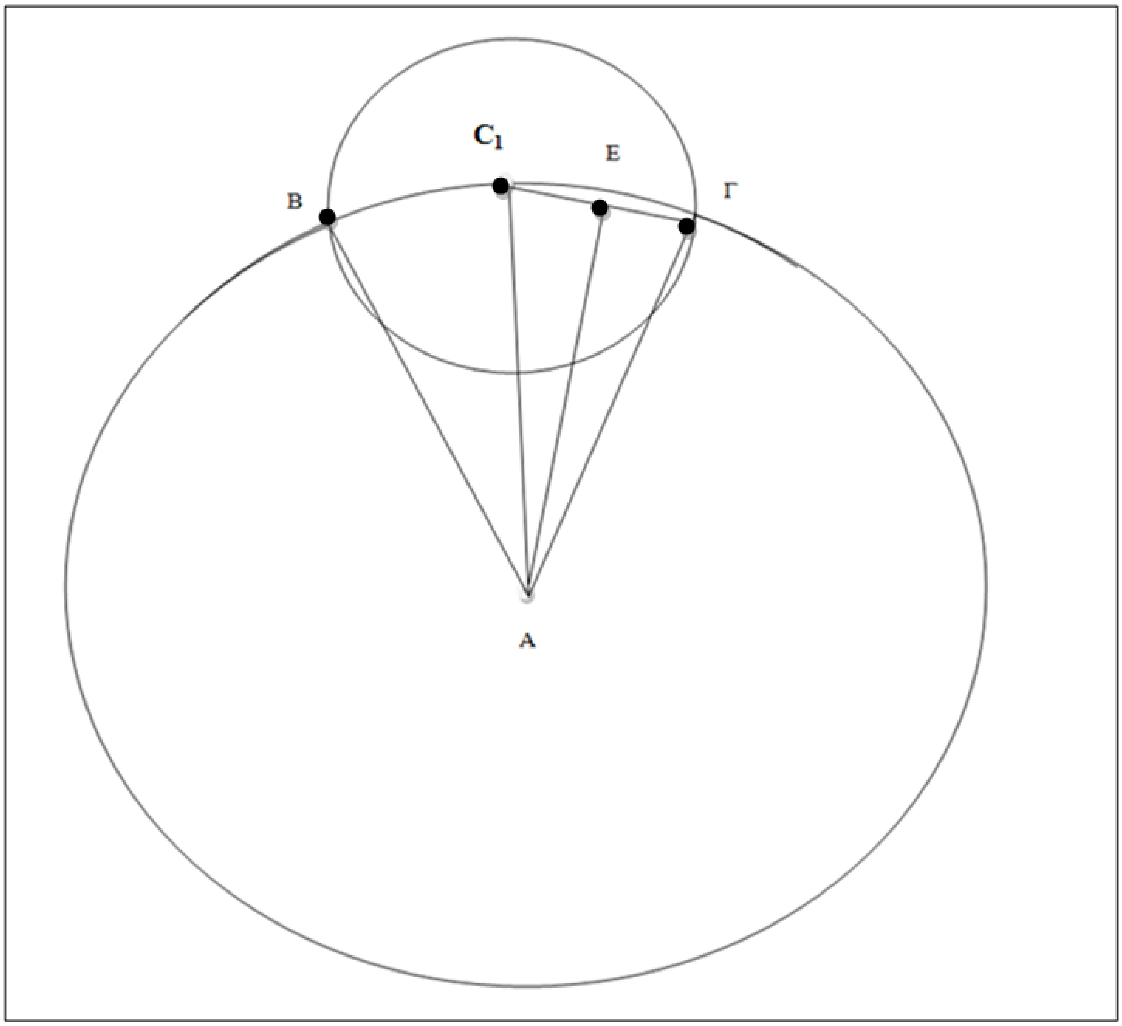
- Υπολόγισε την γραμμή από το κέντρο του αριστερά γειτονικού σου ρομπότ μέχρι το κέντρο του δεξιά γειτονικού σου ρομπότ πάνω στο νοητό κυρτό περίβλημα.
- Υπολόγισε το μέσο της γραμμής που υπολόγισες στο προηγούμενο βήμα
- Φέρε κάθετη γραμμή πάνω στην ευθεία που ενώνει τα κέντρα των γειτονικών σου ρομπότ που ξεκινά από το μέσο της ευθείας των κέντρων και τελειώνει σε απόσταση $\frac{1}{\sin(\frac{90}{n})} + 1$ όπου n ο αριθμός των ρομπότ με κατεύθυνση έξω από το κυρτό περίβλημα.

Λήμμα 4.7:

Η διαδικασία ΔενΥπάρχειΧώροςΓιαΆλλους θα δημιουργήσει αρκετό χώρο ώστε να χωρέσουν τα ρομπότ τα οποία είναι μέσα στο νοητό κυρτό περίβλημα να μπούν πάνω στο νοητό περίβλημα.

Απόδειξη:

Το μικρότερο πιθανό νοητό κυρτό περίβλημα που μπορεί να υπάρξει είναι ένας κύκλος πάνω στον οποίο θα είναι όλα τα ρομπότ και το κάθε ρομπότ θα εφάπτεται με 2 άλλα ρομπότ και όλα τα ρομπότ μαζί θα σχηματίζουν κύκλο.



Σχήμα 4.3: Το σχήμα αυτό παριστάνει ένα ρομπότ (ο μικρός κύκλος) και ο μεγάλος κύκλος παριστάνει τον χώρο που βρίσκονται n ρομπότ συνδεδεμένα σχηματίζοντας κύκλο. Α το κέντρο του μεγάλου κύκλου, Β, Γ σημεία τομής μικρού κύκλου και μεγάλου, $\mathbf{C_1}$ το κέντρο του μικρού κύκλου και Ε το μέσο της $\mathbf{C_1}$Γ

Έστω ο κύκλος του σχήματος 4.3, Στον κύκλο αυτό μπορούν να μπουν n ρομπότ, οπότε η γωνία ΒΑΓ($<180^o$) = $\frac{360}{n}$. Η γωνία $\mathbf{C_1}$ΑΓ($<180^o$) = $\frac{180}{n}$. Η γωνία ΕΑΓ($<180^o$) = $\frac{90}{n}$ αφού το Ε είναι το μέσο της $\mathbf{C_1}$Γ.

Οπότε $\sin(\frac{90}{n}) = \frac{\frac{C_1\Gamma}{2}}{A\Gamma}$ ξέρουμε ότι $\mathbf{C_1}$Γ = 1 αφού είναι η ακτίνα του μικρού κύκλου,

Οπότε $\sin(\frac{90}{n}) = \frac{\frac{1}{2}}{A\Gamma} = \frac{1}{2A\Gamma}$

Οπότε ΑΓ= $\frac{1}{2\sin(\frac{90}{n})}$. Η ΑΓ είναι η ακτίνα του μεγάλου κύκλου

Επομένως αν από την θέση στην οποία βρισκόμαστε μετακινηθούμε 2 ακτίνες αυτού του κύκλου προς τα έξω από το νοητό κυρτό περίβλημα, σύμφωνα με τον ορισμό του κύκλου (διάμετρος = 2 ακτίνες), τότε θα υπάρχει αρκετός χώρος για όλα τα ρομπότ.

4.2.12 ΌχιΣεΕυθεία

<u>Διαδικασία ΌχιΣεΕυθεία:</u>

Η συντεταγμένες στις οποίες θα κινηθείς στο βήμα κινήσου είναι οι συντεταγμένες της θέσης στην οποία βρίσκεσαι τώρα

4.2.13 ΒλέπειςΈναΡομπότ

<u>Διαδικασία Βλέπεις1Ρομπότ:</u>

Η συντεταγμένες στις οποίες θα κινηθείς στο βήμα κινήσου είναι οι συντεταγμένες της θέσης στην οποία βρίσκεσαι τώρα

4.2.14 ΒλέπειςΔύοΡομπότ

<u>Διαδικασία ΒλέπειςΔυοΡομπότ:</u>

> Με βάση το σχήμα του σχήματος 4.2, κινήσου κάθετα στη γραμμή $\mathbf{C_1C_2}$ με κατεύθυνση έξω απο το νοητό κυρτό περίβλημα, έτσι ώστε το κέντρο σου να είναι σε απόσταση 2 απο την γραμμή $\mathbf{C_1C_2}$

4.2.15 ΕίσαιΚολλημένο

<u>Διαδικασία ΕίσαιΚολλημένο:</u>

- Τρέξε την ακολουθία ΒρεςΤαΣημεία
- Αν δεν επιστραφεί κανένα σημείο τότε επέλεξε τα πιο κοντινά σου 2 γειτονικά σημεία τα οποία βρίσκονται πάνω στο νοητό κυρτό περίβλημα. Αν κάποιο από τα ρομπότ που είναι κολλημένο μαζί σου είναι πιο κοντά στα σημεία αυτά από εσένα, τότε μείνε στο ίδιο σημείο που είσαι τώρα, διαφορετικά φέρε μια ευθεία απο το ένα κέντρο στο άλλο, υπολόγισε το μέσο της ευθείας, φέρε κάθετη πάνω στην ευθεία στο σημείο αυτό με μήκος 2 μέτρα και κατεύθυνση προς τα έξω απο το νοητό κυρτό περίβλημα, και η άκρη της κάθετης θα είναι το σημείο στο οποίο θα κινηθείς στο βήμα κινήσου.
- Απο τα σημεία που σου επέστεψε επέλεξε το ποιό κοντινό σε εσένα, αν κάποιο απο τα ρομπότ με τα οποία είσαι κολλημένος είναι πιο κοντά στο σημείο αυτό απο εσένα επέλεξε το επόμενο πιο κοντινό σε εσένα σημείο απο τα σημεία που επιστράφηκαν απο την διαδικασία ΒρεςΤαΣημεία με βάση τα ρομπότ που είναι κολλημένα μαζί σου και επανέλαβε μέχρι να βρεις κάποιο που είναι πιο κοντινό προς εσένα απο τα υπόλοιπα, ή έχουν εξαντληθεί τα σημεία που επέστρεψε η ακολουθία ΒρεςΤαΣημεια.
- Αν έχουν εξαντληθεί τα σημεία που επέστρεψε η ακολουθία ΒρεςΤαΣημεια, τότε το σημείο στο οποίο θα μετακινηθείς είναι το ίδιο με το σημείο που είσαι τώρα
- Αν έχεις βρει κάποιο το οποίο είναι πιο κοντινό ως προς εσένα , τότε στο βήμα κινήσου θα κινηθείς σε αυτό το σημείο.

4.2.16 ΘαΠροκαλέσειςΑλλαγή

Διαδικασία ΘαΠροκαλέσειςΑλλαγή

- Επέλεξε τα πιο κοντινά σου 2 γειτονικά σημεία (που απέχουν περισσότερο από 2) τα οποία βρίσκονται πάνω στο νοητό κυρτό περίβλημα φέρε μια ευθεία απο το ένα κέντρο στο άλλο, υπολόγισε το μέσο της ευθείας, φέρε κάθετη πάνω στην ευθεία στο σημείο αυτό με μήκος 2 μέτρα και κατεύθυνση προς τα έξω απο το νοητό κυρτό περίβλημα, και η άκρη της κάθετης θα είναι το σημείο στο οποίο θα κινηθείς στο βήμα κινήσου

4.2.17 ΔενΘαΠροκαλέσειςΑλλαγή

Διαδικασία ΔενΘαΠροκαλέσειςΑλλαγή

- Τρέξε την διαδικασία ΒρεςΤαΣημεια
- Βρες το πιο κοντινό σε εσένα απο αυτά τα σημεία
- Το σημείο αυτό θα είναι το σημείο στο οποίο θα κινηθείς στο βήμα κινήσου

4.3 Απόδειξη

Λήμμα 4.8:

Απο το Σχήμα 3.2, υπάρχει μια απόσταση για την οποία όταν το $\mathbf{C_2}$ και το $\mathbf{C_3}$ βρίσκονται σε απόσταση ίση με αυτή ή μεγαλύτερη, τότε αν κάποιο ρομπότ θέλει να μπει ανάμεσα στο $\mathbf{C_2}$ και το $\mathbf{C_3}$ Χρησιμοποιώντας την διαδικασία ΒρέςΤαΣημεία, ΕίσαιΚολλημένο ή ΔενΘαΠροκαλέσειΑλλαγή, δεν θα έχουμε καμία αλλαγή πάνω στο νοητό κυρτό περίβλημα (δηλαδή Κάποια σημεία τα οποία είναι τώρα πάνω το νοητό κυρτό περίβλημα, να βρεθούν στην κατάσταση ΕίσαιΣεΕυθεία, είτε να βρεθούν μέσα στο νοητό κυρτό περίβλημα).

Απόδειξη:

Έστω η γωνία F $\mathbf{C_2}$ $\mathbf{C_3}$=φ=1(η ελάχιστη επιτρεπτή κλίση) στο σχήμα 3.2. Σύμφωνα με αυτή την εικόνα χρειαζόμαστε απόσταση ψ= 4 (η απόσταση ψ είναι η απόσταση που απέχει το μέσο της

C_2 C_3=χ κάθετα απο την προέκταση της γραμμής C_1C_2) για να μην προκληθούν αλλαγές στο νοητό κυρτό περίβλημα.

Επομένως : tanφ = $\frac{\psi}{\frac{x}{2}}$ => ψ = $\frac{\tan\varphi \times \chi}{2}$

Θέλουμε ψ=4 επομένως: 4 = $\frac{\tan\varphi \times \chi}{2}$ => χ = $\frac{8}{\tan\varphi}$

Η ελάχιστη γωνία φ είναι 1 επομένως: χ = $\frac{8}{\tan 1}$

Αυτή η απόσταση χ είναι η ελάχιστη απόσταση που απαιτείται μεταξύ 2 σημείων για να μήν έχουμε αλλαγή.

<u>Λήμμα 4.9</u>:

Με τις τελικές καταστάσεις του αλγορίθμου αυτού για την περίπτωση στην οποία δέν έχουμε *Πλήρη Ορατότητα*, το μέγεθος του νοητού κυρτού περιβλήματος θα αυξάνεται συνεχώς όσο χρειάζεται μέχρι να έχουμε *Πλήρη Ορατότητα* ή θα παραμένει σταθερό (δηλαδή δεν θα μικραίνει, μέγεθος = εμβαδό/διαστάσεις)

<u>Απόδειξη:</u>

Απο τις καταστάσεις ΔενΥπάρχειΧώροςΓιαΆλλους, ΒλέπειςΔυοΡομπότ, Κολλημένο, ΘαΠροκαλέσειΑλλαγή, ΔενΘαΠροκαλέσειΑλλαγή παρατηρούμε ότι τα ρομπότ έχουν την τάση να κινούνται προς τα έξω απο το κυρτό περίβλημα, οπότε κάθε φορά μεγαλώνει το κυρτό περίβλημα. Στις περιπτώσεις όπου απο τις μετακινήσεις αυτές κάποια ρομπότ ενώ ήταν πάνω στο κυρτό περίβλημα, βρεθούν μέσα στο κυρτό περίβλημα, πάλι το μέγεθος του κυρτού περιβλήματος έχει αυξηθεί, και σε μετέπειτα κινήσεις τον ρομπότ που βρέθηκαν μέσα στο κυρτό περίβλημα, πάλι θα έχουν σαν αποτέλεσμα το νοητό κυρτό περίβλημα να μεγαλώσει. Στις καταστάσεις ΌχιΣεΕυθεία και ΒλέπειΕναΡομπότ τα ρομπότ παραμένουν στάσιμα οπότε το μέγεθος του νοητού κυρτού περιβλήματος παραμένει σταθερό. Οι πιο πάνω καταστάσεις είναι όλες οι πιθανές καταστάσεις σε περίπτωση που δεν έχουμε *Πλήρη Ορατότητα* και έτσι αποδεικνύεται ότι το μέγεθος του νοητού κυρτού περιβλήματος είτε αυξάνεται είτε παραμένει σταθερό.

<u>Λήμμα 4.10:</u>

Ο αλγόριθμος αυτός θα καταφέρει να φέρει τα ρομπότ σε κατάσταση *Πλήρης Ορατότητας*.

<u>Απόδειξη:</u>

Απο το λήμμα 4.8 βλέπουμε ότι υπάρχει μια απόσταση χ για 2 ρομπότ τέτοια ώστε να μήν μπορεί να γίνει άλλη αλλαγή πάνω στο νοητό κυρτό περίβλημα. Έτσι έχουμε 2 περιπτώσεις:

Α) Την περίπτωση στην οποία θα γίνονται συνεχείς αλλαγές πάνω στο νοητό κυρτό περίβλημα, οπότε με τις αλλαγές σημαίνει ότι το νοητό κυρτό περίβλημα μεγαλώνει, οπότε θα φτάσουμε σε κάποια κατάσταση όπου όλα τα ρομπότ θα έχουν απόσταση χ μεταξύ τους, έτσι δεν θα γίνει άλλη αλλαγή και όλα τα ρομπότ θα είναι πάνω στο νοητό κυρτό περίβλημα.

Β) Την περίπτωση στην οποία θα σταματήσουν σε κάποια χρονική στιγμή να γίνονται αλλαγές πάνω στο νοητό κυρτό περίβλημα οπότε αυτό θα σημαίνει ότι όλα τα ρομπότ είναι πάνω στο νοητό κυρτό περίβλημα, αφού άν κάποιο απο αυτά ήταν στο εσωτερικό του κυρτού περιβλήματος, θα είχε προσπαθήσει να βγει προς τα έξω σύμφωνα με τον αλγόριθμο.

Απο τις περιπτώσεις Α και Β καταλήγουμε στο κοινό συμπέρασμα ότι όλα τα ρομπότ τελικά θα καταλήξουν να είναι πάνω στο κυρτό περίβλημα.

Απο τον ορισμό του κυρτού περιβλήματος και δεδομένου ότι δεν υπάρχει κανένα ρομπότ μέσα στο κυρτό περίβλημα, τότε μπορούμε να καταλήξουμε στο συμπέρασμα ότι όλα τα ρομπότ θα μπορούν να δουν όλα τα υπόλοιπα ρομπότ, έτσι φτάσαμε σε κατάσταση Πλήρης Ορατότητας.

<u>Θεώρημα:</u>

Ο αλγόριθμος αυτός θα καταφέρει να συγκεντρώσει όλα τα ρομπότ μαζί και θα τερματίσει

<u>Απόδειξη:</u>

Απο το λήμμα 4.6 και το λήμμα 4.10 μπορούμε να καταλήξουμε στο συμπέρασμα ότι ο αλγόριθμος αυτός θα καταφέρει να συγκεντρώσει όλα τα ρομπότ μαζί και θα τερματίσει.

Κεφάλαιο 5

Συμπεράσματα

5.1 Γενικά Συμπεράσματα

Στα προηγούμενα κεφάλαια είδαμε την περιγραφή του μοντέλου που χρησιμοποιούμε για αυτό το πρόβλημα. Είδαμε τις δυσκολίες που οφείλονται στο ότι τα ρομπότ σε αυτό το μοντέλο θεωρούνται ότι έχουν όγκο. Ο αλγόριθμος λύει το πρόβλημα με 2 βασικά βήματα. Αρχικά φέρνει όλα τα ρομπότ σε κατάσταση *πλήρης ορατότητας* και μετά συγκεντρώνονται όλα μαζί. Η *πλήρης ορατότητα* επιτυγχάνεται από το γεγονός ότι όλα τα ρομπότ προσπαθούν να πάνε στο κυρτό περίβλημα. Αυτό με τη σειρά του ωθεί το κυρτό περίβλημα να μεγαλώσει και έτσι να φτάσει σε κάποιο μέγεθος στο οποίο θα χωρούν όλα τα ρομπότ πάνω. Εκεί χάρη στις ιδιότητες του κυρτού περιβλήματος όλα τα ρομπότ μπορούν να δουν όλα τα υπόλοιπα. Το δεύτερο βήμα, δηλαδή η συγκέντρωση των ρομπότ επιτυγχάνεται δίνοντας προτεραιότητα σε ποιο ρομπότ να κινηθεί. Πρωτεύον παράγοντα για την προτεραιότητα χρησιμοποιούμε το μέγεθος των συγκεντρώσεων και δευτερεύων την απόσταση των συγκεντρώσεων (μέγεθος κενών). Τα ρομπότ πηγαίνουν προς την μεγαλύτερη συγκέντρωση, ή προς την μεγαλύτερη απόσταση, κινούμενα πάνω στο κυρτό περίβλημα, ώστε να μην χαλάσει η κατάσταση πλήρης ορατότητας στην οποία βρισκόμαστε. Αν υπάρχει απόλυτη συμμετρία τα ρομπότ πλησιάζουν με μικρά βήματα μέχρι να ακουμπήσουν όλες οι συγκεντρώσεις δημιουργώντας μια.

Βιβλιογραφία

[1] N. Agmon και D. Peleg, “Fault Tolerant Gathering Algorithms for Autonomous Mobile Robots”, *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, Ιανουάριος 2003.

[2] H. Ando, Y. Oasa, I. Suzuki και M. Yamashita, “Distributed Memoryless Point Convergence Algorithm for Mobile Robots with Limited Visibility”, *IEEE Transactions on Robotics and Automation*, Τόμος 15, Τεύχος 5, σελ. 818-828, 1999.

[3] J. Czyzowicz, L. Gąsieniec και A. Pelc, “Gathering Few Fat Mobile Robots in the Plane”, *Theoretical Computer Science*, Τόμος 410, Τεύχη 6-7, σελ. 481-499, 2009.

[4] P. Flocchini, G. Prencipe, N. Santoro και P. Widmayer, ”Gathering of Asynchronous Robots With Limited Visibility”, *Theoretical Computer Science*, Τόμος 337, Τεύχη 1-3, σελ. 147-168, 2005.

[5] P. Fraigniaud και A. Pelec, “Deterministic Rendezvous in Trees with Little Memory”, *Proceedings of the 22nd International Symposium on Distributed Computing*, Lecture Notes in Computer Science, Τόμος 5218 σελ. 242–256, 2008.

[6] R. L. Graham, “An Efficient Algorithm for Determining the Convex Hull of a Finite Plenar Set”, *Information Processing Letters*, Τόμος 1, Τεύχος 4, σελ. 132-133, 1972.

[7] G. Prencipe, *Distributed Coordination of a Set of Autonomous Robots*, Διδακτορική Διατριβή, Universita Degli Studi di Pisa, 2002.

[8] S. Souissi, X. Défago και M. Yamashita, “Gathering Asynchronous Mobile Robots with Inaccurate Compasses”, *Proceding of the 10th International Conference on Principles of Distributed Systems*, Lecture Notes in Computer Science, Τόμος 4305, σελ. 333-349, Δεκέμβριος 2006.

[9] K. Sugihara και I. Suzuki, “Distributed Algorithms for Formation of Geometric Patterns with Many Mobile Robots”, *Journal of Robotic Systems*, Τόμος 13, Τεύχος 3, σελ. 127-139, 1996.

[10] I. Suzuki και M. Yamashita, “Characterizing Geometric Patterns Formable by Oblivious Anonymous Mobile Robots”, *Theoretical Computer Science*, Τόμος 411, Τεύχη 26-28, σελ. 2433-2453, 2010.