ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Αναζήτηση εξόδου σε κύκλο υπό την παρουσία faulty robot

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Αναγνωστόπουλος

Α.Μ. 03116066

Επιβλέπων: Αριστείδης Παγουρτζής, Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2021

Η σελίδα αυτή είναι σκόπιμα λευκή.

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Αναζήτηση εξόδου σε κύκλο υπό την παρουσία faulty robot

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Αναγνωστόπουλος

Α.Μ. 03116066

Επιβλέπων: Αριστείδης Παγουρτζής, Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11/11/2021.

Αριστείδης Παγουρτζής	Δημήτριος Φωτάκης	Ευριπίδης Μάρκου
Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π	Αναπληρωτής Καθηγητής Πανεπιστήμιο Θεσσαλίας

Αθήνα, Νοέμβριος 2021

....

Θεόδωρος Αναγνωστόπουλος

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

Περίληψη

Η αναζήτηση αποτελεί κύριο μέρος της καθημερινότητας των ανθρώπων. Για αυτό το λόγο προβλήματα αναζήτησης αποτελούν σημαντική ερευνητική περιοχή της επιστήμης των υπολογιστών. Στην παρούσα διπλωματική εργασία θα ασχοληθούμε με τη μελέτη ενός τέτοιου προβλήματος. Συγκεκριμένα προτείνουμε λύσεις για το πρόβλημα αναζήτησης εξόδου, η οποία βρίσκεται στην περιφέρεια ενός κύκλου. Την αναζήτηση την πραγματοποιούν σημειακά ρομπότ. Αυτά έχουν ως αρχική θέση το κέντρο του κύκλου και κινούνται με ταχύτητα ίση με ένα. Επικοινωνούν μεταξύ τους ασύρματα και το μήνυμα που στέλνει το κάθε ρομπότ περιέχει την ταυτότητα του ρομπότ, η οποία δεν μπορεί να αλλαχθεί με κανένα τρόπο. Μελετάμε την περίπτωση που έχουμε n ρομπότ από τα οποία τα f είναι faulty. Το πρόβλημα θεωρούμε ότι έχει λυθεί όταν ένα non-faulty ρομπότ έχει βρει την έξοδο και τα υπόλοιπα non-faulty ξέρουν με σιγουριά την θέση της εξόδου. Τα faulty χωρίζονται σε δύο κατηγορίες, στα crash και στα Byzantine. Τα crash είναι ελαττωματικά και μπορεί να σταματήσουν να λειτουργούν οποιαδήποτε στιγμή. Τα Byzantine έχουν ως στόχο να αποπροσανατολίσουν τα υπόλοιπα ρομπότ και μεταφέρουν ψευδείς πληροφορίες για την θέση της εξόδου. Συγκεκριμένα θα μελετηθεί η περίπτωση όπου διαθέτουμε δύο Byzantine. Η κύρια συνεισφορά της εργασίας είναι η ανάπτυξη ενός αλγορίθμου που πετυχαίνει χρόνο $1 + \frac{8\pi}{n}$ για την περίπτωση $(n, 2, 2)$ και $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$ για την περίπτωση $(n, f, 2)$. Πρώτα παρουσιάζουμε μια αναλυτική περιγραφή του αλγορίθμου και αποδεικνύουμε την ορθότητα και τα άνω φράγματα στον χρόνο εκτέλεσης. Στη συνέχεια παρουσιάζονται διάφορα παραδείγματα που δείχνουν την δυσκολία στην ύπαρξη αλγορίθμου που επιλύει το ίδιο πρόβλημα σε λιγότερο χρόνο. Στο τέλος γίνεται μία εισαγωγή στην επίλυση της περίπτωσης (n, f, f), που βρίσκεται υπό ολοκλήρωση αυτή την περίοδο. Ο χρόνος που έχουμε καταλήξει είναι $1 + 2f\frac{2\pi}{n}$ για $f \leq 6$, $1 + 2 + (f + 2)\frac{2\pi}{n}$ για $f \leq 16$ και $1 + 2 + (f + 3)\frac{2\pi}{n}$ για τις υπόλοιπες περιπτώσεις.

Λέξεις κλειδιά : Faulty, Byzantine, crash, ασύρματη επικοινωνία, έξοδος κύκλου, αναζήτηση

Η σελίδα αυτή είναι σκόπιμα λευκή.

Abstract

Searching is a part of our everyday life. Search problems appear everywhere, and for this reason they represent a main field of study in computer science. In the current thesis, we will propose solutions for the search problem presented below. We assume that we have a circle of unit radius, in which there is a hidden exit. We also have robots that we send from the center of the circle, to find that exit. Specifically, we have n robots, that f of them are faulty. Their speed equals one and they communicate with each other wirelessly. They send messages to each other that are tagged by the ID of each robot, and such tags cannot be altered. We stop the search, when at least one non-faulty robot has found the exit and all other non-faulty know where the exit is. Moreover, the faulty robots can be of two different types. The first one is the crash faulty. These robots are defective and they can stop working without notice. The other type is the Byzantine. These robots can transmit messages that contain false data about the position of the exit. In our study we solve the case, in which we have two Byzantines. The timi complexity of our algorithm time for the worst-case scenario is $1 + \frac{8\pi}{n}$ for $(n, 2, 2)$ and $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$ for $(n, f, 2)$. Moreover, we present some examples that show the difficulty of finding an algorithm that solves the same problem faster than our algorithm. In the end we introduce the (n, f, f) problem on which we are currently working. The time that we have achieved so far is $1 + 2f\frac{2\pi}{n}$ for $f \leq 6$, $1 + 2 + (f + 2)\frac{2\pi}{n}$ for $f \leq 16$ and $1 + 2 + (f + 3)\frac{2\pi}{n}$ for the remaining situations.

Keywords: Faulty robots, crash faulty, Byzantine, circle search, hidden exit, wireless communication

Η σελίδα αυτή είναι σκόπιμα λευκή.

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας, θα ήθελα, πρωτίστως, να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Παγουρτζή, τόσο για την ανάθεση του συγκεκριμένου θέματος όσο και για τη συνεχή υποστήριξη και καθοδήγηση του κατά τη συγγραφή της παρούσας διπλωματικής εργασίας.

Επιπλέον θα ήθελα να ευχαριστήσω τον υποψήφιο διδάκτορα Γιάννη Παπαϊωάννου για την συνεργασία του και την βοήθεια που πρόσφερε κατά τη διάρκεια εκπόνησης της διπλωματικής μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη που μου πρόσφεραν.

Η σελίδα αυτή είναι σκόπιμα λευκή.

Περιεχόμενα

Ευρετήριο Πινάκων

Ευρετήριο Σχημάτων

1.Εισαγωγή

1.1 Σημασία προβλημάτων αναζήτησης

Η σημασία της αναζήτησης στην ανθρώπινη ζωή είναι τα πάντα. Αυτό γίνεται καλύτερα αντιληπτό αν αναλογιστούμε ότι καθημερινά ο άνθρωπος κάτι ψάχνει. Αυτός μπορεί να ψάχνει από ένα χαμένο αντικείμενο μέχρι ένα μέρος. Για να το θέσουμε καλύτερα αυτός μπορεί να ψάχνει το οτιδήποτε. Επιπλέον το θέμα της αναζήτησης μπορεί να εμβαθύνει και σε πιο βαθιά ζητήματα όπως η αναζήτηση της σωστής απόφασης, της γνώσης ή της ταυτότητας κάποιου. Από τα παραπάνω κανένας δεν μπορεί να αμφισβητήσει την σημασία της αναζήτησης. Ο καθένας δένεται σε ποικίλους στόχους της αναζήτησης και μεθόδους της. Πέρα όμως από την καθημερινότητα τα προβλήματα αναζήτησης είναι στο επίκεντρο σε σχεδόν όλους τους τομείς της επιστήμης των υπολογιστών. Παραλλαγές προβλημάτων αναζήτησης έρχονται στο φως στην μελέτη των δομών δεδομένων, εφαρμογών βάσεων δεδομένων, υπολογιστικής γεωμετρίας και τεχνητής νοημοσύνης. Για αυτό το λόγο μεγάλος αριθμός προβλημάτων έχουν μελετηθεί, κάποια από τα οποία θα αναλυθούν παρακάτω [1].

1.2 Προϋπάρχουσα Έρευνα

Οι πρώτες μελέτες σε προβλήματα αναζήτησης πραγματοποιήθηκαν κατά τη διάρκεια του δεύτερου παγκόσμιου πολέμου από τον Bernard Koopman και τους συναδέλφους του στην Anti-Submarine Warfare Research Group Operations του πολεμικού ναυτικού της Αμερικής. Σκοπός τους ήταν να μελετήσουν και να βρουν αποτελεσματικούς τρόπους για την ανίχνευση υποβρυχίων. Ακόμα και σήμερα η θεωρία αναζήτησης που ανέπτυξαν χρησιμοποιείται για την εύρεση αντικειμένων ακόμα και σε αποστολές διάσωσης. Για παράδειγμα το Αμερικανικό ναυτικό χρησιμοποίησε τη θεωρία αναζήτησης για την αναζήτηση της Η-βόμβας που χάθηκε στον ωκεανό το 1996. Παρόμοια την χρησιμοποίησε για να βρει το υποβρύχιο με όνομα “Scorpion” το 1968 [2].

1.2.1 Αναζήτηση σε ευθεία γραμμή

Το πρόβλημα αυτό μελετήθηκε πρώτα από τον Beck και Bellman ([3], [4]) και αποτελεί το αρχικό πρόβλημα σε μία σειρά παραλλαγών. Μία τέτοια παραλλαγή θα μελετήσουμε σε αυτή τη διπλωματική. Σε αυτό το πρόβλημα διαθέτουμε μια ευθεία γραμμή πάνω στην οποία βρίσκεται η έξοδος η οποία αναζητάται. Θεωρούμε ένα σημείο στην ευθεία από το οποίο αρχίζει η αναζήτηση και είτε δεξιά είτε αριστερά από αυτό βρίσκεται η έξοδος. Η αναζήτηση πραγματοποιείται από κινητά ρομπότ. Στην αρχική εκδοχή η μελέτη πραγματοποιήθηκε με το να έχουμε στη διάθεσή μας ένα κινητό ρομπότ το οποίο αναζητά την έξοδο. Στη συνέχεια επεκτάθηκε με το να διαθέτουμε n ρομπότ από τα οποία f είναι faulty. Δηλαδή αυτά μπορεί να δώσουν ψευδή έξοδο και να μην ενημερώσουν αν περάσουν από την αληθινή ή απλά σταματάνε να λειτουργούν ([5], [6]). Τα αποτελέσματα που έχουν παραχθεί μέχρι σήμερα για την εκδοχή που έχουμε faulty είναι τα παρακάτω.

Πίνακας 1

Πάνω (upper) και κάτω (lower) φράγμα (bound) για το χρόνο αναζήτησης $S_d(n, f)$ για δοσμένο αριθμό $n \leq 6$ από ρομπότ και $f = 1, 2$ Byz UB και Byz LB δηλώνουν το γνωστό upper και lower bound για Byzantine faults ενώ Crash UB και Crash LB δηλώνουν το γνωστό upper και lower bound για crash faults. Το d συμβολίζει την απόσταση της εξόδου από την αρχική θέση.

n,f	Byz. UB	Byz. LB	Crash-UB	Crash-LB
3,1	9d	3,93d	5,24d	3,76d
4,1	3d	3d	d	d
5,1	2d	2d	d	d
6,1	d	d	d	d
5,2	9d	3,57d	4,43d	3,57d
6,2	4d	3d	d	d

Πίνακας 2

Upper και lower bound για την αναλογία ασυμπτωτικής ανταγωνιστικής αναζήτησης $S(\beta)$ για ποικίλες τιμές της πυκνότητας β. Σημείωση ότι για $\beta \geq \frac{1}{2}$ το πρόβλημα αναζήτησης είναι αδύνατον να λυθεί. Το β ορίζεται ως η πυκνότητα $\frac{f}{n}$

β	$\leq \frac{1}{4}$	$(\frac{1}{4}, \frac{3}{10}$	$(\frac{3}{10}, \frac{1}{3}$	$(\frac{1}{3}, \frac{5}{14}$	$(\frac{5}{14}, \frac{13}{34}]$	$(\frac{13}{34}, \frac{19}{46}]$	$(\frac{19}{46}, \frac{47}{110}]$	$(\frac{47}{110}, \frac{65}{146}]$	$(\frac{65}{146}, \frac{157}{396}]$	$(\frac{157}{396}, \frac{1}{2}]$
UB	1	2	3	3	4	5	6	7	8	9
LB	1	2	2	3	3	3	3	3	3	3

1.2.2 Αναζήτηση σε άγνωστο περιβάλλον

Το πρόβλημα αυτό μελετάει το γνωστό w-lane-cow-path πρόβλημα. Θεωρούμε μία αγελάδα η οποία στέκεται σε ένα σταυροδρόμι με w δρόμους να οδηγούν σε άγνωστη περιοχή. Σε μία από αυτές τις περιοχές βρίσκεται ο στόχος σε απόσταση n, ενώ τα υπόλοιπα μονοπάτια συνεχίζουν για πάντα. Η αγελάδα γνωρίζει ότι έφτασε στον στόχο μόνο εάν βρεθεί πάνω του. Αν αυτή γνωρίζει το μονοπάτι που οδηγεί στο στόχο, χρειάζεται να περπατήσει απόσταση ίση με n. Αν δεν ξέρει πρέπει να υπολογισθεί αλγόριθμος που θα την οδηγεί στο να βρει τον στόχο, κάνοντας την μικρότερη απόσταση. Το παραπάνω πρόβλημα παίζει σημαντικό ρόλο στη ρομποτική, συγκεκριμένα όταν ένα ρομπότ τοποθετείται σε άγνωστη περιοχή. Για παράδειγμα όταν ένα ρομπότ εξερευνά έναν χώρο, κάθε φορά που "πέφτει" πάνω σε ένα εμπόδιο χρειάζεται να βρει το τέλος του εμποδίου για να προχωρήσει. Τα αποτελέσματα του αλγορίθμου που κατέληξαν οι Ming-Yang Kao, John H. Reif, και Stephen R. Tate φαίνονται παρακάτω [7]:

Πίνακας 3

Κατά προσέγγιση τιμές για μικρά w. Το w συμβολίζει τον αριθμό των δρόμων που έχει το σταυροδρόμι. Η τιμή της μεταβλητής r_w^* βρίσκεται από την εξίσωση: $\ln r = \frac{1+r+r^2+...+r^{w-1}}{1+2r^2+...+(w-1)r^w}$

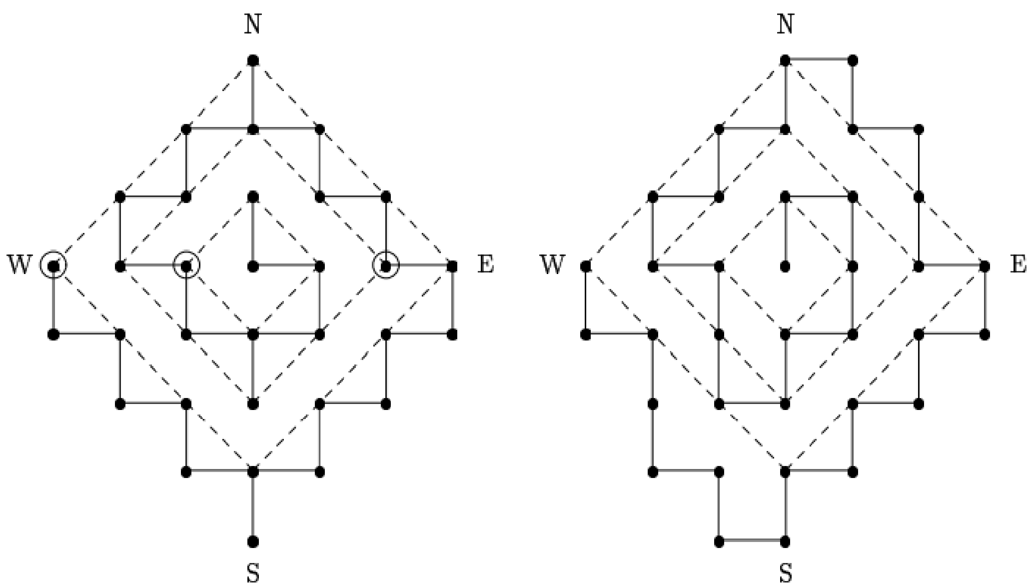
w	r_w^*	Competitive ratio of SmartCow	Optimal deterministic ratio
2	3.59112	4.59112	9.00000
3	2.01092	7.73232	14.5
4	1.62193	10.84181	19.96296
5	1.44827	13.94159	25.41406
6	1.35020	17.03709	30.85984
7	1.28726	20.13033	36.30277

1.2.3. Αναζήτηση στο χώρο

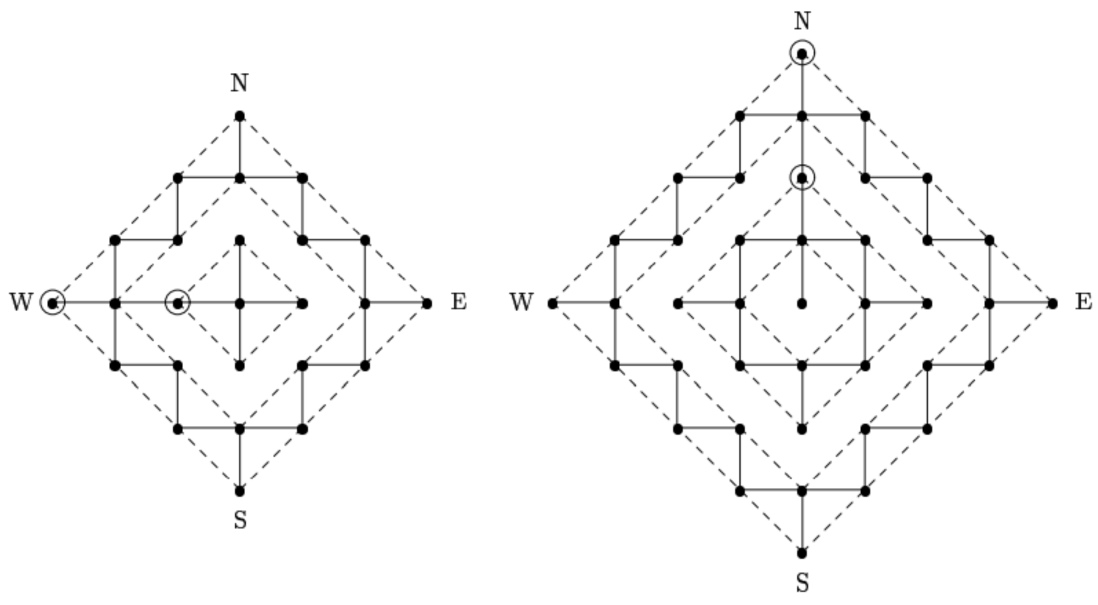
Από τους Ricardo A. Baeza-Yates, Joseph C. Culberson και Gregory J. E. Rawlins μελετήθηκε το ποιος είναι ο καλύτερος τρόπος να βρεθεί ένα αντικείμενο σε έναν άπειρο χώρο [8].

● Αναζήτηση σημείο πάνω σε πλέγμα

Σε αυτή την περίπτωση ένα ρομπότ πρέπει να βρει το στόχο πάνω σε ορθογώνιο πλέγμα. Αυτό μπορεί να κινηθεί πάνω, κάτω δεξιά και αριστερά. Ο αλγόριθμος σχηματικά είναι ο ακόλουθος:



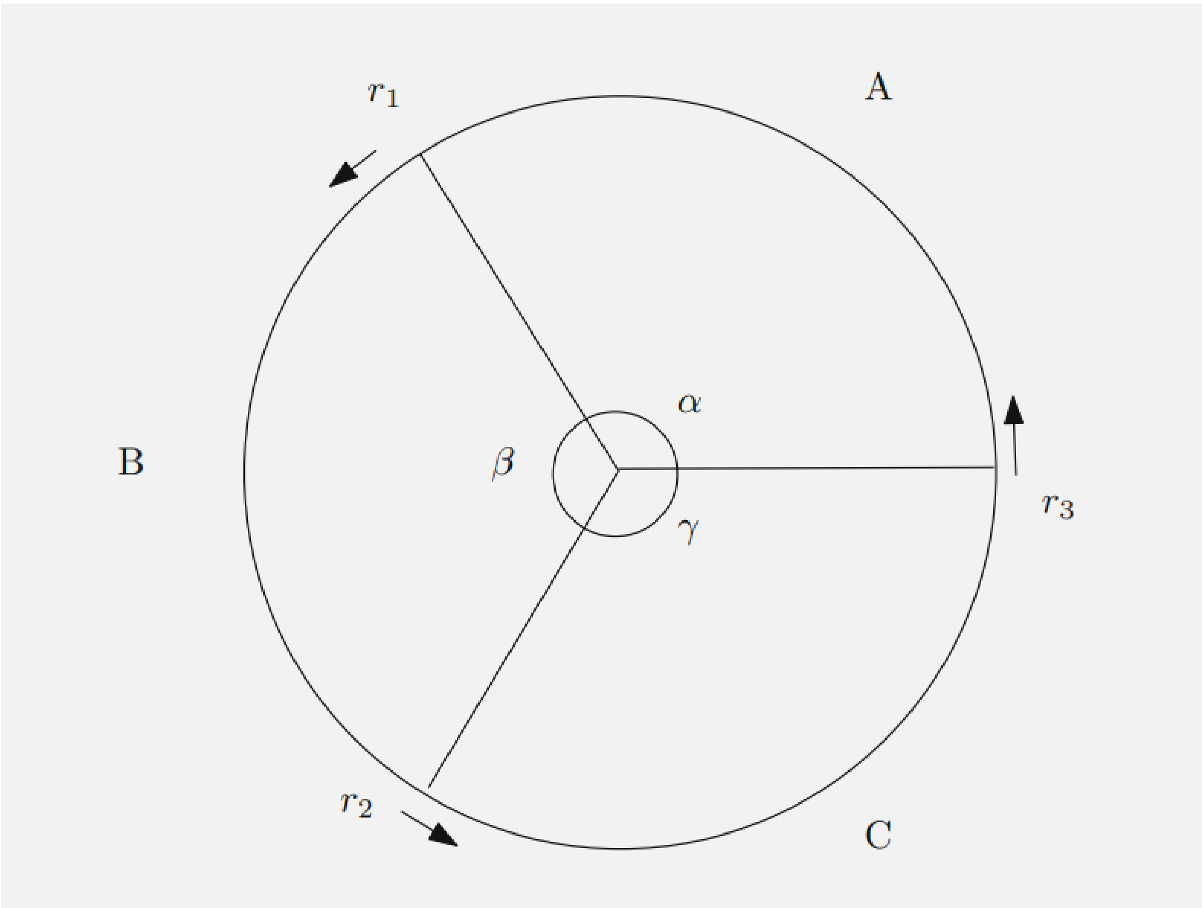
Σχήμα 1: Ισορροπημένοι αλγόριθμοι για καλύτερη συμπεριφορά στην χειρότερη περίπτωση. Η αριστερή εικόνα δείχνει τον ισορροπημένο αλγόριθμο ενώ η δεξιά τον ανάστροφο ισορροπημένο αλγόριθμο.



Σχήμα 2: Περιττός και ζυγός σπειροειδής αλγόριθμος. Η αριστερή εικόνα αναπαριστά τον περιττό, ενώ η δεξιά των ζυγό.

1.2.4 Εκκένωση σε δίσκο παρά την παρουσία faulty ρομπότ

Σε αυτό το πρόβλημα που μελετήθηκε από τους Jurek Czyzowicz, Konstantinos Georgiou, Maxime Godon, Evangelos Kranakis, Danny Krizanc, Wojciech Rytter και Michal Wlodarczyk [9] ο χρόνος υπολογίζεται με βάση το να φεύγει και το τελευταίο ρομπότ από την έξοδο. Διαθέτουμε έναν κύκλο και τρία ρομπότ από τα οποία το πολύ ένα είναι faulty, είτε crash είτε Byzantine. Η έξοδος βρίσκεται σε άγνωστο σημείο στην περιφέρεια του κύκλου. Στόχος είναι να βρεθεί η έξοδος και στη συνέχεια όλα τα ρομπότ να περάσουν από αυτή και να φύγουν από τον κύκλο. Το αποτέλεσμα που προέκυψε όταν έχουμε τρία ρομπότ από τα οποία το ένα είναι Byzantine είναι $1 + \frac{4\pi}{3} + \sqrt{3}$. Το πρόβλημα όπου από τα τρία το ένα είναι crash λύνεται σε χρόνο 6.309. Το lower bound για το πρόβλημα με crash είναι $1 + \frac{4\pi}{3}$. Για το πρόβλημα με Byzantine είναι 5.948.



Σχήμα 3: Όλα τα ρομπότ κινούνται με φορά αντίθετη των δεικτών του ρολογιού. Το τόξο A περιλαμβάνει το r_3 και εξαιρεί το r_1. Το τόξο B περιλαμβάνει το r_1 και εξαιρεί το r_2.Το τόξο C περιλαμβάνει το r_2 και εξαιρεί το r_3.

Πριν από αυτό το πρόβλημα είχε μελετηθεί το πρόβλημα της εκκένωσης σε δίσκο με ρομπότ από τα οποία κανένα δεν είναι faulty για δύο περιπτώσεις επικοινωνίας, την ασύρματη και την μη ασύρματη. Αυτό το πρόβλημα μελετήθηκε από τους J. Czyzowicz, K. Georgiou, E. Kranakis, L. Narayanan , J. Opatrny και B. Vogtenhuber ([10], [11]). Τα αποτελέσματα παρουσιάζονται στον παρακάτω πίνακα.

Πίνακας 4

Αποτελέσματα για $\kappa = 2$ *ή* $\kappa = 3$*, όπου κ ισούται με τον αριθμό των ρομπότ*

Τύπος Επικοινωνίας	Άνω Φράγμα	Κάτω Φράγμα
Μη ασύρματη με $\kappa = 2$	~5.74	~5.199
Ασύρματη με $\kappa = 2$	$1 + \frac{2\pi}{3} + \sqrt{3} \sim 4.83$	$1 + \frac{2\pi}{3} + \sqrt{3} \sim 4.83$
Μη ασύρματη με 3	$3 + \frac{2\pi}{3} \sim 5.09$	~4.519
Ασύρματη με $\kappa = 3$	~4.22	~4.159

Πίνακας 5

Αποτελέσματα *για* μεγάλο κ

Τύπος Επικοινωνίας	Άνω Φράγμα	Κάτω Φράγμα
Μη ασύρματη	$3 + \frac{2\pi}{k}$	$3 + \frac{2\pi}{k} - O(k^{-2})$
Ασύρματη	$3 + \frac{\pi}{k} + O(k^{-4/3})$	$3 + \frac{\pi}{k}$

1.2.5 Αναζήτηση σε κύκλο παρά την παρουσία faulty ρομπότ με ένα Byzantine

Διαθέτουμε έναν κύκλο με ακτίνα ένα στην περιφέρεια του οποίου υπάρχει μία έξοδος. Στόχος είναι η υλοποίηση αλγορίθμου που να βρίσκει αυτή την έξοδο από n ρομπότ. Από αυτά, τα f είναι faulty τα οποία χωρίζονται σε crash και Byzantine. Τα crash είναι ελαττωματικά και η λειτουργία τους μπορεί να σταματήσει οποιαδήποτε στιγμή. Τα Byzantine ενημερώνουν τα υπόλοιπα ρομπότ με ψευδή στοιχεία για τη θέση της εξόδου. Το πρόβλημα αναζήτησης στο οποίο διαθέτουμε μόνο ένα Byzantine μελετήθηκε από τους Konstantinos Georgiou, Evangelos Kranakis, Nikos Leonardos, Aris Pagourtzis και Ioannis Papaioannou [12]. Συγκεκριμένα μελετήθηκαν οι περιπτώσεις όπου από faulty έχουμε μόνο ένα Byzantine και κανένα crash και όπου έχουμε f faulty από τα οποία το ένα είναι Byzantine. Βγήκαν τα ακόλουθα συμπεράσματα:

- Lower Bound: $1 + \frac{(f+1)2\pi}{n}$, όπου n το σύνολο των ρομπότ και $f \geq 1$ crash failures
- Worst Case Search: $1 + \frac{4\pi}{n}$, όπου n το σύνολο των ρομπότ από τα οποία ένα είναι Byzantine και τα υπόλοιπα honest
- Upper Bound: $1 + \frac{2\pi}{n}f + 2sin\frac{2\pi}{n}$, όπου n το σύνολο των ρομπότ και f το σύνολο των faulty από τα οποία το ένα είναι Byzantine και τα υπόλοιπα crash.

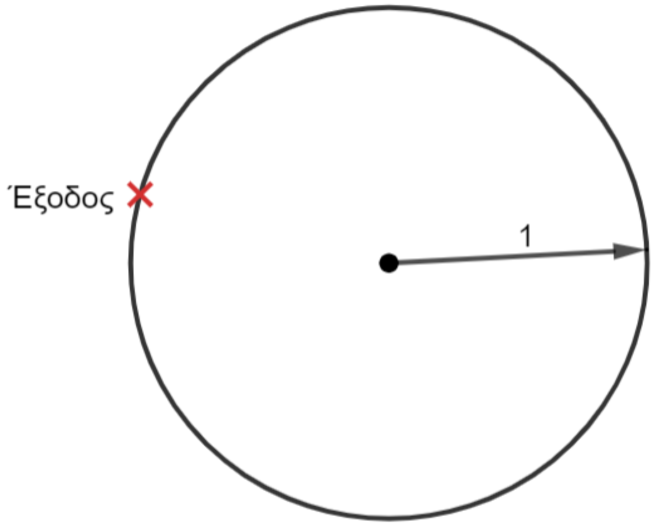
1.2.6 Λοιπά Προβλήματα

Πέρα από τα παραπάνω έχουν μελετηθεί πολλά ακόμα προβλήματα. Αξίζει να γίνει αναφορά στο βιβλίο The Theory Of Search Games And Rendezvous των S. Alpren και S. Gal, στο οποίο λύνεται πληθώρα προβλημάτων αναζήτησης [13]. Ακόμα ενδιαφέρον παρουσιάζει η αλλαγή στο πρόβλημα αναζήτησης στη γραμμή που έγινε από τους Erik D. Demaine, Sándor P. Fekete και Shmuel Gal [14]. Σε αυτό το πρόβλημα κάθε φορά που ένα ρομπότ κάνει αλλαγή στην πορεία του υπάρχει ένα κόστος d το οποίο προσθέτει επιπλέον δυσκολία στο αρχικό πρόβλημα. Επίσης παραλλαγές έχουν γίνει και στο πρόβλημα της εκκένωσης που έχει παρουσιαστεί. Εκτός από κύκλο έχουν μελετηθεί προβλήματα και για άλλα γεωμετρικά σχήματα όπως τρίγωνο και τετράγωνο [15]. Ακόμα ενδιαφέροντα προβλήματα παρουσιάζει και το βιβλίο Distributed Computing by Mobile Entities [16]. Τέλος αξίζει να αναφερθούμε στην πρόβλημα εκκένωσης που έγινε για δύο Byzantine από τους Nikos Leonardos, Aris Pagourtzis και Ioannis Papaioannou [17]. Σε αυτή την έρευνα βρίσκεται το άνω φράγμα για το πρόβλημα εκκένωσης υπό την ύπαρξη δύο Byzantine.

1.3 Περιεχόμενο και Συνεισφορά Παρούσας Διπλωματικής

1.3.1 Διατύπωση Προβλήματος

Στην παρούσα διπλωματική εργασία θα μελετηθεί το παρακάτω πρόβλημα. Θεωρούμε έναν κύκλο ακτίνας ένα (1), στον οποίο υπάρχει μία έξοδος στην περιφέρειά του. Στόχος μας είναι η υλοποίηση αλγορίθμου που να βρίσκει την έξοδο στον λιγότερο δυνατό χρόνο. Για την εύρεση της εξόδου θεωρούμε ότι διαθέτουμε ρομπότ τα οποία δεν έχουν διαστάσεις και τα ονομάζουμε mobile agents. Για συντομία στην παρούσα διπλωματική θα τα αναφέρουμε ως ρομπότ.



Σχήμα 4: Το περιβάλλον του προβλήματός μας.

1.3.2 Κατηγορίες mobile agents

Διαθέτουμε δύο ειδών mobile agents, τα honest και τα faulty.

- Honest Mobile Agents
  Αυτά είναι ειλικρινή και αν περάσουν από την έξοδο θα ενημερώσουν τα υπόλοιπα ρομπότ ότι βρήκαν την έξοδο, μεταδίδοντάς τους τη σωστή θέση.

- Faulty Mobile Agents
  Αυτά χωρίζονται σε δύο υποκατηγορίες στα crash και στα Byzantine.

  - Crash Mobile Agents
    Αυτά είναι ελαττωματικά και μπορούν ανά οποιαδήποτε στιγμή να σταματήσουν να λειτουργούν. Αυτό σημαίνει ότι θα παραμείνουν αδρανή και θα αδυνατούν να επικοινωνήσουν με τα υπόλοιπα ρομπότ.

  - Byzantine Mobile Agents
    Αυτά ενημερώνουν λανθασμένα για την έξοδο. Συγκεκριμένα μπορούν να περάσουν από την έξοδο χωρίς να ενημερώσουν ή και να περάσουν από σημείο που δεν βρίσκεται η έξοδος και να ειδοποιήσουν ότι την βρήκαν. Σκοπός τους είναι να

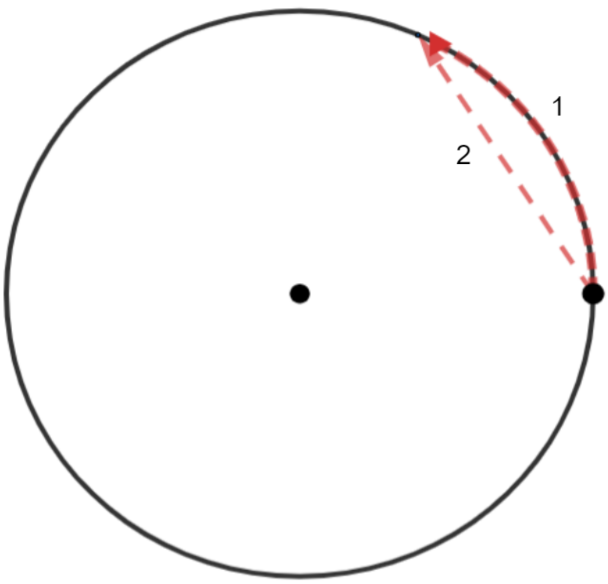
ξεγελάσουν τα υπόλοιπα ρομπότ και να τα παρεμποδίσουν από το να βρουν την αληθινή έξοδο.

1.3.3 Επικοινωνία των mobile agents

Τα ρομπότ επικοινωνούν ασύρματα ανεξάρτητα από την μεταξύ τους απόσταση. Ο χρόνος επικοινωνίας θεωρείται αμελητέος. Τα μηνύματα που ανταλλάσσουν αφορούν πληροφορίες σχετικά με την θέση τους και το αν βρήκαν ή όχι την έξοδο. Τα ρομπότ δεν μπορούν να πουν ψέματα για τη θέση τους, καθώς με βάση το χρόνο που θα στείλουν το μήνυμα, μπορούν τα υπόλοιπα ρομπότ να υπολογίσουν την θέση τους. Επομένως αν ενημερώσουν με ψευδή στοιχεία για την θέση τους, τα υπόλοιπα θα το καταλάβουν και σαν αποτέλεσμα θα καταλήξουν ότι αυτά που ενημέρωσαν ψευδά είναι Byzantine. Επομένως δεν θα λαμβάνουν υπόψη τους τα μηνύματα που θα στέλνουν τα αναγνωρισμένα Byzantine. Τέλος το κάθε μήνυμα περιέχει την ταυτότητα του κάθε ρομπότ, η οποία δεν μπορεί να αλλαχθεί με κανέναν τρόπο.

1.3.4 Κίνηση των mobile agents

Όλα τα ρομπότ ξεκινούν την πορεία τους από το κέντρο κύκλου ακτίνας ένα. Η ταχύτητα τους είναι ένα και είναι κοινή για όλα. Κατά τη διάρκεια της κίνησης τους αυτά κινούνται στην περιφέρεια του κύκλου. Αν χρειαστεί μπορούν να μετακινηθούν και στο εσωτερικό του κύκλου ώστε να φτάσουν σε συγκεκριμένο σημείο γρηγορότερα.



Σχήμα 5: Οι δύο πιθανές κινήσεις των ρομπότ. Στην (1) το ρομπότ κινείται πάνω στην περιφέρεια του κύκλου. Μπορεί να κινείται και προς την αντίθετη κατεύθυνση. Στην (2) το ρομπότ κινείται πάνω σε χορδή.

1.3.5 Βασικά κάτω φράγματα

- Επίλυση όταν διαθέτουμε μόνο honest και Byzantine ρομπότ.

Θεωρούμε ότι διαθέτουμε f Byzantine και έχουμε στο σύνολο n mobile agents. Για να μπορούμε να βρούμε με σιγουριά την έξοδο πρέπει να διαθέτουμε τους honest mobile agents σε πλειοψηφία. Σε διαφορετική περίπτωση οι Byzantine θα μας εμποδίζουν να βρούμε την έξοδο. Θεωρούμε ότι έχουμε f Byzantine και λιγότερους ή ίσους honest. Αν και οι f Byzantine συμφωνήσουν στην ίδια ανακοίνωση, τότε θα καταλήξουμε σε δύο ανακοινώσεις. Η μία θα προέρχεται από τα f Byzantine, ενώ η άλλη ανακοίνωση η οποία είναι και η σωστή θα προέρχεται από όλους τους honest. Εφόσον οι honest είναι ίσοι ή λιγότεροι από τους Byzantine, δεν μπορούμε να ξεχωρίσουμε ποια από τις δύο ανακοινώσεις είναι η σωστή. Στην περίπτωση που έχουμε f honest, θα έχουμε δύο ανακοινώσεις που θα προέρχονται από f ρομπότ η κάθε μία και έτσι δεν μπορούμε να απορρίψουμε μία από τις δύο. Χρειαζόμαστε ένα ακόμα honest το οποίο θα απορρίψει την μία ανακοίνωση και θα επιβεβαιώσει την άλλη. Στην περίπτωση που έχουμε λιγότερα από f honest, η ανακοίνωση που προέρχεται από τα Byzantine θα έχει επιβεβαιωθεί από περισσότερα ρομπότ από ότι την σωστή ανακοίνωση. Από τα παραπάνω προκύπτει ότι πρέπει να διαθέτουμε τα honest σε πλειοψηφία. Επομένως στην περίπτωση που έχουμε f Byzantine θα πρέπει να ισχύει $n \geq 2 \cdot f + 1$, δηλαδή να διαθέτουμε $f + 1$ honest mobile agents.

- Επίλυση όταν διαθέτουμε μόνο honest και crash ρομπότ.

Θεωρούμε ότι διαθέτουμε c crash mobile agents και n mobile agents. Για να μπορούμε με σιγουριά να βρούμε την έξοδο θα πρέπει να περάσει τουλάχιστον ένας honest mobile agent (καθώς δεν διαθέτουμε Byzantine για να μας παραπληροφορήσει). Συνεπώς στην περίπτωση που έχουμε μόνο c crash mobile agents θα πρέπει να ισχύει $n \geq c + 1$.

- Επίλυση όταν διαθέτουμε όλους τους τύπους ρομπότ.

Τέλος διαθέτουμε f faulty από τα οποία b είναι Byzantine mobile agents και n σύνολο mobile agents. Επομένως διαθέτουμε $f - b$ crash mobile agents. Από συνδυασμό των παραπάνω θα πρέπει να ισχύει $n \geq 2 \cdot b + 1 + f - b$.

Σε όλες τις υπόλοιπες περιπτώσεις δεν μπορεί να λυθεί το πρόβλημα.

Ο αλγόριθμος μελετάται στην περίπτωση της χειρότερης περίπτωσης. Δηλαδή τόσο η έξοδος όσο και η επιλογή των Byzantine γίνεται με τέτοιο τρόπο ώστε η εύρεση της εξόδου να πραγματοποιείται στον χειρότερο χρόνο. Θεωρούμε ότι η έξοδος έχει βρεθεί όταν από αυτή έχει περάσει τουλάχιστον ένα μη faulty και όλα τα υπόλοιπα ρομπότ πείθονται ότι αυτή είναι η έξοδος χωρίς αμφιβολία. Ο αλγόριθμος σε αυτό το πρόβλημα σε σχέση με τα προβλήματα εκκένωσης, τελειώνει όταν βρεθεί η έξοδος και όχι όταν όλα τα ρομπότ έχουν φτάσει στην έξοδο.

1.3.6 Λοιπά Χαρακτηριστικά του Μοντέλου

Για να γίνουν πιο εύκολα κατανοητά τα παρακάτω προσθέτουμε κάποια επιπλέον δεδομένα.

Όταν κάποιος mobile agent βρίσκει και ανακοινώνει μία έξοδο, τότε αυτόματα απορρίπτει όλες τις άλλες εξόδους που έχουν βρεθεί ή θα βρεθούν και συνεπώς δεν έχει νόημα να τον μετακινήσουμε από την στιγμή και έπειτα που θα ανακοινώσει έξοδο. Αν έβρισκε και σε άλλο σημείο έξοδο τότε αυτόματα θα ήταν αντιληπτό ότι αυτός θα ήταν Byzantine και συνεπώς αυτόματα θα απορρίπτονταν όλες οι ανακοινώσεις του. Ένας honest mobile agent ανακοινώνει μόνο όταν περάσει από την αληθινή έξοδο και εφόσον από τα δεδομένα του προβλήματος διαθέτουμε μόνο μία έξοδο, δεν μπορεί να βρει έξοδο σε πάνω από ένα σημείο.

Στον αλγόριθμο που θα παρουσιάσουμε οι mobile agents κινούνται αντίθετα της φοράς των δεικτών του ρολογιού (counterclockwise) εκτός και αν διευκρινιστεί κάτι διαφορετικό.

Ο γενικός συμβολισμός του προβλήματος είναι ο $S(n, f, b)$. Το n αναφέρεται στον συνολικό αριθμό των mobile agents, το f στον συνολικό αριθμό των faulty και το b στον αριθμό των faulty mobile agents που είναι Byzantine.

Ο ν mobile agent συμβολίζεται ως $α_{ν \, mod \, n}$ το οποίο για ευκολία το γράφουμε $α_ν$.

Ο κύκλος χωρίζεται σε τόσα ίσα τόξα όσο και ο συνολικός αριθμός των mobile agents. Το κάθε τόξο συμβολίζεται ως εξής: $S_ν$, όπου ν είναι ο mobile agent που θα περάσει πρώτος από αυτό το τόξο. Ισχύει και όπως παραπάνω ότι το ν τόξο συμβολίζεται ως $S_{ν \, mod \, n}$ το οποίο για ευκολία το γράφουμε $S_ν$. Τέλος τα τόξα αλλιώς τα ονομάζουμε Sectors.

1.3.7 Επίλυση Αναζήτησης με Δύο Byzantine Ρομπότ

Στην παρούσα διπλωματική έγινε η μελέτη όταν διαθέτουμε ακριβώς δύο Byzantine. Τα αποτελέσματα που πήραμε είναι τα ακόλουθα:

- Worst Case Search: $1 + \frac{8π}{n}$, όπου n το σύνολο των ρομπότ από τα οποία δύο είναι Byzantine και τα υπόλοιπα honest. Δηλαδή έχουμε την περίπτωση $(n, 2, 2)$.

- Upper Bound: $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$, όπου n το σύνολο των ρομπότ και f το σύνολο των faulty από τα οποία το ένα είναι Byzantine και τα υπόλοιπα crash. Δηλαδή έχουμε την περίπτωση $(n, f, 2)$.

- Lower Bound: $1 + \frac{(f+1)2\pi}{n}$, όπου n το σύνολο των ρομπότ και $f \geq 1$ crash failures

Σε αυτό το σημείο θα παραθέσουμε ένα θεώρημα, το οποίο θα χρησιμοποιηθεί παρακάτω στην ανάλυση του αλγορίθμου που υλοποιήσαμε.

Θεώρημα 1:

Αν έχει περάσει από κάθε σημείο του κύκλου τουλάχιστον ένα honest και έχουμε μόνο μία έξοδο (είτε δεν έχουν βρεθεί άλλες είτε όσες έχουν βρεθεί έχουν απορριφθεί) τότε αυτή είναι η σωστή.

Απόδειξη:

Σε περίπτωση που αυτή η έξοδος είναι ψευδής, τότε δεν έχει βρεθεί η αληθινή έξοδος. Όμως αφού από κάθε σημείο του κύκλου έχει περάσει τουλάχιστον ένα honest ρομπότ σημαίνει ότι κάποιο honest πέρασε από την έξοδο χωρίς να ειδοποιήσει, το οποίο είναι άτοπο.

1.3.8 Επέκταση στην Γενική Περίπτωση

Εφόσον λύθηκε το πρόβλημα για την περίπτωση των δύο Byzantine, είναι επόμενο να επιλυθεί το πρόβλημα όπου έχουμε f Byzantine. Αρχικά χωρίς την ύπαρξη crash ρομπότ. Εφόσον υπάρχουν f Byzantine, για να μπορεί να λυθεί το πρόβλημα απαιτούνται $n \geq 2f + 1$ συνολικά ρομπότ. Το πρόβλημα βρίσκεται υπό επίλυση αυτή την περίοδο και τα αποτελέσματα όσων αφορά το άνω φράγμα για το (n, f, f) πρόβλημα είναι τα εξής:

- $1 + 2f\frac{2\pi}{n}$, για $n \leq 6$
- $1 + 2 + (f + 2)\frac{2\pi}{n}$, για $n \leq 16$
- $1 + 2 + (f + 3)\frac{2\pi}{n}$, για τις υπόλοιπες περιπτώσεις

2. Άνω φράγμα για την περίπτωση των δύο Byzantine

2.1 Περίπτωση (5,2,2): 5 ρομπότ, 2 Byzantine

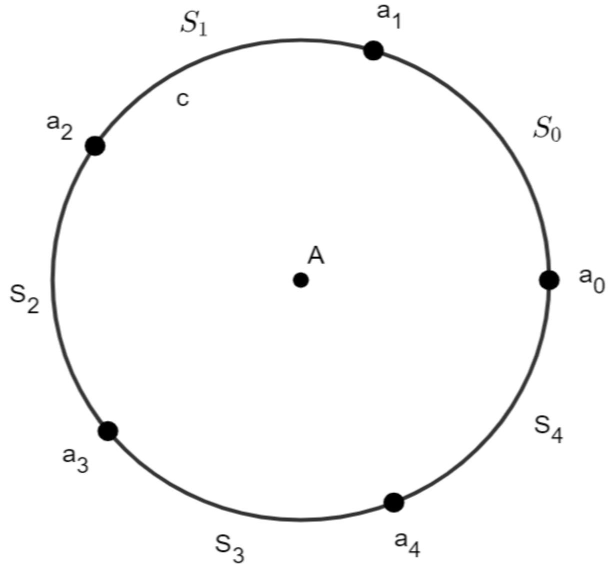
2.1.1 Περιγραφή Αλγορίθμου

Διαθέτουμε πέντε στο σύνολο ρομπότ από τα οποία τα δύο είναι Byzantine. Επομένως τα υπόλοιπα τρία είναι honest. Παρατηρούμε ότι $5 \geq 2 \cdot 2 + 1$ και επομένως μπορεί να λυθεί το πρόβλημα. Όλα τα ρομπότ ξεκινάνε από το κέντρο του κύκλου. Το πρόβλημα χωρίζεται σε τέσσερις φάσεις και η ανάλυση θα γίνεται για κάθε μία φάση. Ο μέγιστος αριθμός διαφορετικών ανακοινώσεων για εξόδους που μπορούμε να έχουμε είναι τρεις. Δύο από τα δύο Byzantine και μία από το/τα honest. Τα honest δεν μπορούν να βρουν διαφορετικές εξόδους μεταξύ τους.

- **Ανάλυση Φάσεων**

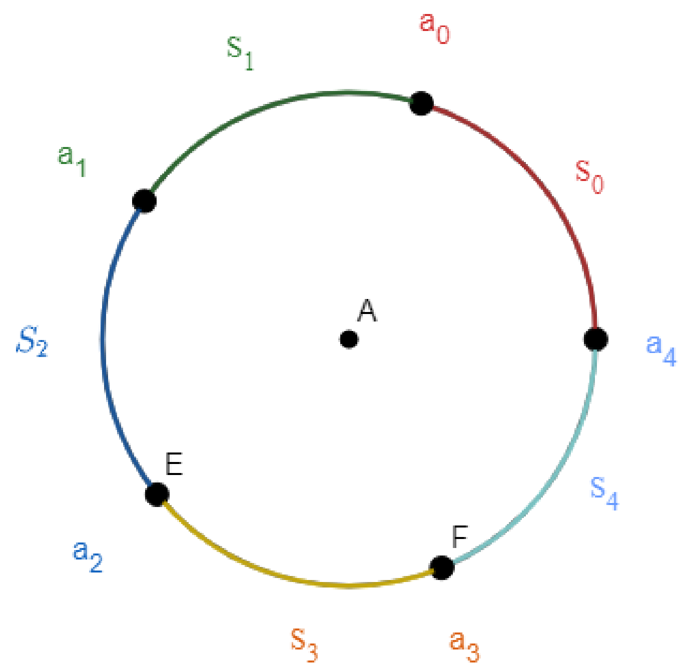
Όπως αναφέραμε οι φάσεις είναι τέσσερις και χωρίζονται ως εξής:

1. **Φάση 0:** Τα ρομπότ από το κέντρο του κύκλου μετακινούνται στη θέση τους στην περιφέρεια του κύκλου. Η θέση για το ρομπότ α_κ είναι η $\kappa \cdot \theta$, όπου $\theta = \frac{2\pi}{5}$. Δηλαδή χωρίζουμε τον κύκλο στα πέντε.



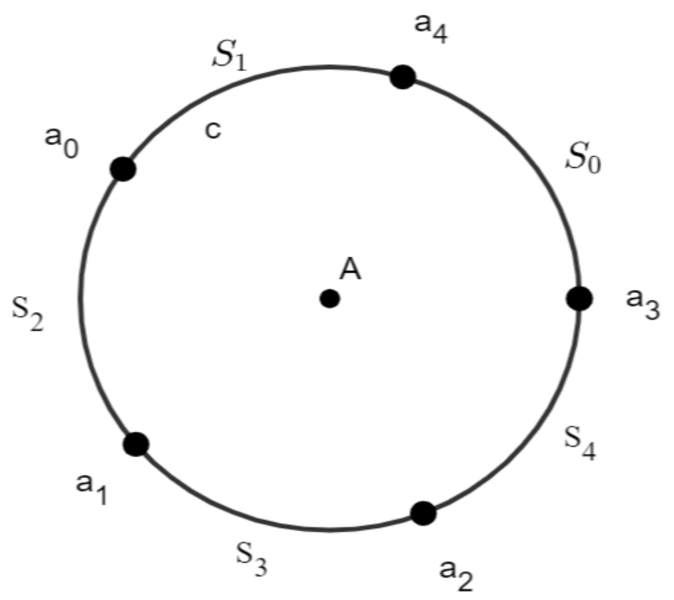
Σχήμα 6: Τα ρομπότ μετά το πέρας της φάσης 0.

2. **Φάση 1:** Μετακίνηση των ρομπότ με φορά αντίθετη των δεικτών του ρολογιού (counterclockwise) από τη θέση κ·θ στη $(\kappa + 1)\cdot\theta$, δηλαδή κίνηση κατά $\frac{2\pi}{5}$.



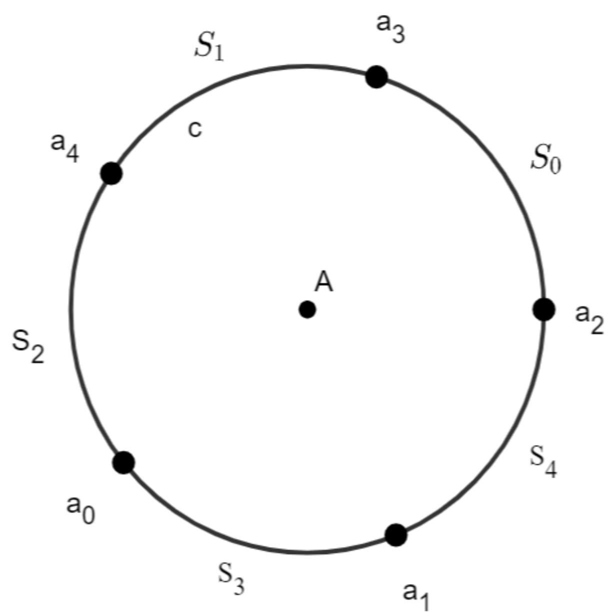
Σχήμα 7: Η θέση των ρομπότ μετά το πέρας της φάσης 1.

3. **Φάση 2:** Μετακίνηση των ρομπότ με φορά αντίθετη των δεικτών του ρολογιού (ccw) από τη θέση $(\kappa + 1)\cdot\theta$ στη $(\kappa + 2)\cdot\theta$.



Σχήμα 8: Η θέση των ρομπότ μετά το πέρας της φάσης 2.

4. **Φάση 3:** Μετακίνηση των ρομπότ με φορά αντίθετη των δεικτών του ρολογιού (ccw) από τη θέση $(\kappa + 2)\cdot\theta$ στη $(\kappa + 3)\cdot\theta$.



Σχήμα 9: Η θέση των ρομπότ μετά το πέρας της φάσης 3.

- **Φάση 1**

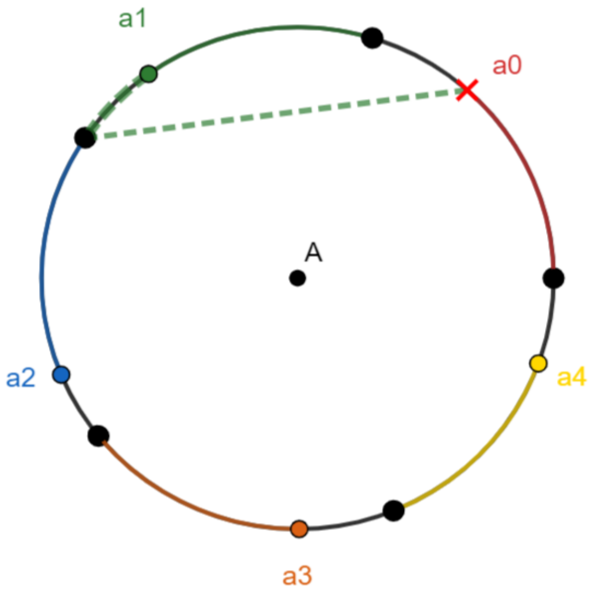
Η ανάλυση θα αρχίσει από την πρώτη φάση και θα μελετηθούν όλες οι περιπτώσεις. Εφόσον διαθέτουμε μία έξοδο και δύο Byzantine, γίνεται εύκολα αντιληπτό ότι μπορούν σε κάθε φάση να ανακοινωθούν μηδέν, μία, δύο και τρεις έξοδοι. Παραπάνω δεν γίνεται καθώς ο μέγιστος αριθμός εξόδων είναι όταν βρει το honest την αληθινή έξοδο και τα δύο Byzantine ανακοινώσουν δύο ξεχωριστές εξόδους.

— Μηδέν ρομπότ ανακοινώνουν την έξοδο

Σε αυτή την περίπτωση προχωράμε στην επόμενη φάση, η οποία είναι η δύο. Πρακτικά μετακινούμε όλα τα ρομπότ κατά θ με φορά αντίθετη των δεικτών του ρολογιού (ccw). Σε αυτή την περίπτωση από το τόξο στο οποίο βρίσκεται η έξοδος πέρασε Byzantine.

— Ένα ρομπότ ανακοινώνει έξοδο

Θεωρούμε ότι την έξοδο την ανακοινώνει το α_ν. Αρχικά όλα τα ρομπότ ολοκληρώνουν την φάση ένα. Στην συνέχεια στέλνουμε το $\alpha_{\nu+1}$ στην έξοδο του α_ν με χορδή. Τα υπόλοιπα συνεχίζουν κανονικά την κίνηση τους για τρεις επιπλέον γύρους. Αν σε έναν γύρο στο Sector $\nu + 3$ έχει ανακοινωθεί έξοδος τότε στέλνουμε το $\alpha_{\nu+1}$ σε αυτή την έξοδο με χορδή. Στο τέλος της κίνησης από την έξοδο που ανακοίνωσε το α_ν θα έχουν περάσει πέντε στο σύνολο ρομπότ (μαζί με το α_ν) και έτσι καθώς έχουμε ακριβώς δύο Byzantine, αυτό σημαίνει ότι τα τρία θα είναι honest και έτσι αν το α_ν είναι honest ακόμα δύο θα επιβεβαιώσουν την έξοδο. Σε περίπτωση που απορριφθεί τότε καταλαβαίνουμε ότι το α_ν είναι Byzantine και επομένως παραμένει μόνο ένα Byzantine. Άρα πλέον χρειάζονται μόνο τρία ρομπότ για να είμαστε σίγουροι για την ορθότητα μίας εξόδου. Συγκεκριμένα αρκεί δύο για να επιβεβαιώσουν ή να απορρίψουν μία έξοδο για να αποφανθούμε με σιγουριά για το αν είναι αληθινή ή λανθασμένη αντίστοιχα. Παρατηρούμε ότι σε όλα τα τόξα εκτός από τα $S_{\nu+3}$ και $S_{\nu+4}$ περνάνε τρία ρομπότ (όπως φαίνεται από τον παρακάτω πίνακα) και επομένως δύο από αυτά είναι honest και αν ανακοινωθεί έξοδος σε κάποιο από αυτά τα τόξα για να είναι η σωστή θα πρέπει να επιβεβαιωθεί από δύο ρομπότ. Για το τόξο $S_{\nu+3}$ αν στον έναν επιπλέον γύρο δεν ανακοινωθεί έξοδος από το $\alpha_{\nu+2}$ τότε δεν βρίσκεται σε αυτό το τόξο η έξοδος καθώς ένα από τα δύο που δεν ανακοίνωσαν έξοδο είναι honest. Σε αντίθετη περίπτωση στέλνουμε το $\alpha_{\nu+1}$ και αν αυτό επιβεβαιώσει την έξοδο τότε αυτή είναι η σωστή. Για το $S_{\nu+2}$ αν το $\alpha_{\nu+4}$ ανακοινώσει έξοδο τότε αυτή είναι η σωστή μόνο αν δεν έχει βρεθεί η έξοδος σε άλλο τόξο. Αυτό ισχύει γιατί από κάθε σημείο έχουν περάσει τουλάχιστον δύο ρομπότ από τα οποία το ένα τουλάχιστον είναι honest και έτσι αν αυτή η έξοδος είναι λανθασμένη, τότε σημαίνει ότι κάποιο honest δεν βρήκε έξοδο το οποίο είναι άτοπο. (Θεώρημα 1)



Σχήμα 10: Φάση 1 - Μόνο το α_0 ανακοινώνει έξοδο.

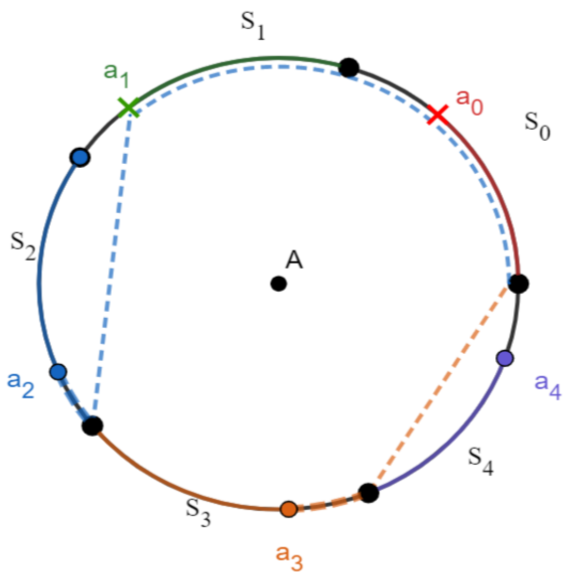
Πίνακας 6

Κινήσεις των ρομπότ για την φάση ένα και όταν ένα ρομπότ ανακοινώνει ότι βρήκε την έξοδο. Τα κενά κελιά δηλώνουν ότι δεν περνάει κανένα ρομπότ από το συγκεκριμένο τόξο τη δεδομένη χρονική στιγμή

Χρόνος / Τόξο	Γύρος 1 ($\frac{2\pi}{5}$)	Γύρος 2 ($\frac{4\pi}{5}$)	Γύρος 3 ($\frac{6\pi}{5}$)	Γύρος 4 ($\frac{8\pi}{5}$)
S_v	α_v	α_{v+4}	α_{v+3}	α_{v+2}
S_{v+1}	α_{v+1}	--	α_{v+4}	α_{v+3}
S_{v+2}	α_{v+2}	--	--	α_{v+4}
S_{v+3}	α_{v+3}	α_{v+2}	--	--
S_{v+4}	α_{v+4}	α_{v+3}	α_{v+2}	--
Έξοδος που βρήκε το α_v	α_v	α_{v+1}, α_{v+4}	α_{v+3}	α_{v+2}

— Δύο συνεχόμενα ρομπότ ανακοινώνουν έξοδο

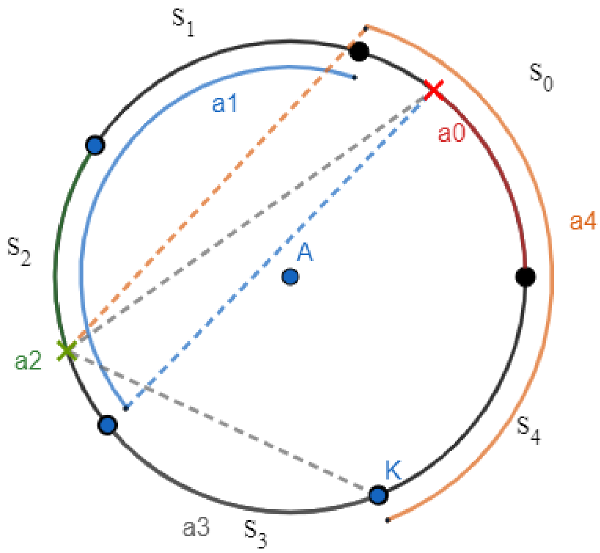
Έστω ότι αυτά είναι της μορφής α_ν και $\alpha_{\nu+1}$. Τότε αν και τα δύο είναι Byzantine σημαίνει ότι τα υπόλοιπα είναι honest και έτσι αφού δεν ανακοινωσαν έξοδο, η έξοδος θα βρίσκεται σε σημείο από όπου έχουν περάσει τα δύο Byzantine. Σε αντίθετη περίπτωση η έξοδος είναι μία από τις δύο που έχουν ανακοινωθεί. Άρα πάντα η λύση βρίσκεται στα τόξα S_ν και $S_{\nu+1}$, δηλαδή σε αυτά που έχουν περάσει και έχουν ανακοινώσει έξοδο τα α_ν και $\alpha_{\nu+1}$. Στέλνουμε το $\alpha_{\nu+2}$ στην έξοδο του $\alpha_{\nu+1}$ και έπειτα εκτελεί αντίστροφη κίνηση. Στέλνουμε το $\alpha_{\nu-2}$ στο $\alpha_{\nu-1}$ και έπειτα να εκτελεί κανονική κίνηση. Το $\alpha_{\nu-1}$ εκτελεί κανονική κίνηση. Άρα σε επιπλέον τρεις γύρους θα έχουν περάσει τέσσερα ρομπότ από κάθε έξοδο και η κάθε έξοδος θα έχει μία απόρριψη από το ρομπότ που ανακοίνωσε αλλού έξοδο . Αν σε μία από τις δύο, τρία επιβεβαιώσουν τότε αυτή θα είναι η έξοδος. Αλλιώς σημαίνει ότι τα δύο ρομπότ που ανακοίνωσαν έξοδο (δηλαδή τα α_ν και $\alpha_{\nu+1}$) είναι Byzantine και όλα τα υπόλοιπα είναι honest. Επομένως όποια έξοδο βρουν τα honest, αυτή θα είναι και η σωστή.



Σχήμα 11: Φάση 1 - Τα α_0 και α_1 βρίσκουν εξόδους.

— Δύο μη συνεχόμενα ρομπότ ανακοινώνουν έξοδο

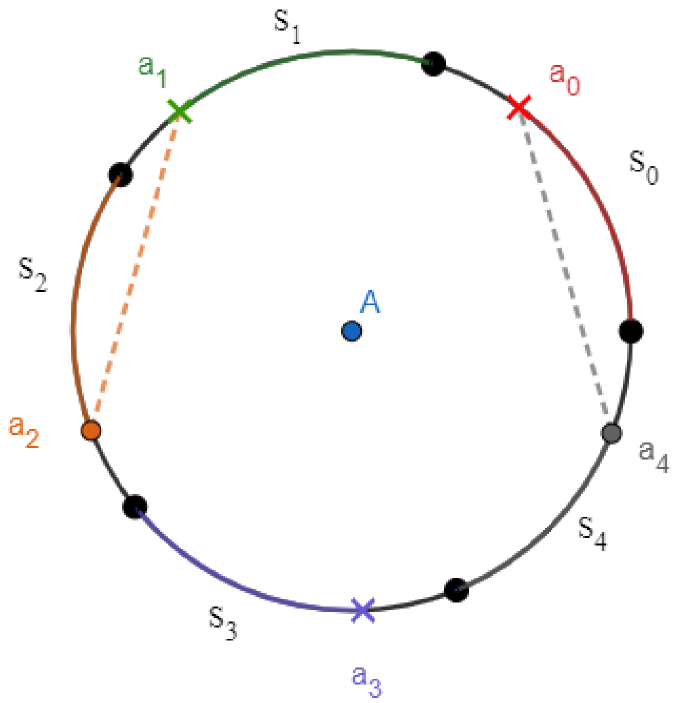
Έστω ότι αυτά είναι της μορφής α_ν και α_κ με $\kappa \neq \nu \pm 1$, δηλαδή δεν είναι συνεχόμενα. Τότε για έναν γύρο τα $\alpha_{\kappa-1}$ και $\alpha_{\nu-1}$ συνεχίζουν κανονικά και στη συνέχεια πάνε απευθείας στην έξοδο του α_ν και α_κ αντίστοιχα. Ακόμα το $\alpha_{\kappa+1}$ πηγαίνει κατευθείαν στην έξοδο του α_κ και στη συνέχεια στην έξοδο του α_ν .Αν τρία στο σύνολο επιβεβαιώσουν μία από τις εξόδους είτε του α_ν είτε του α_κ τότε αυτή είναι η έξοδος. Αλλιώς σημαίνει ότι τα α_ν και α_κ είναι Byzantine καθώς απορρίφθηκε η έξοδος τους. Σαν αποτέλεσμα τα υπόλοιπα ρομπότ είναι honest και έτσι όποια έξοδο βρουν, αυτή θα είναι η σωστή.



Σχήμα 12: Φάση 1 - Τα α_0 και α_2 ανακοινώνουν εξόδους.

— Τρία ρομπότ ανακοινώνουν έξοδο

Τρία ρομπότ ανακοινώνουν έξοδο. Αφού υπάρχουν μόνο δύο Byzantine σημαίνει ότι τουλάχιστον ένα από τα τρία είναι honest. Αν δύο ήταν honest τότε θα έχουν βρεθεί δύο αληθινές έξοδοι το οποίο είναι άτοπο. Άρα ακριβώς ένα είναι honest και τα υπόλοιπα δύο είναι Byzantine. Τα υπόλοιπα δύο που δεν ανακοίνωσαν έξοδο είναι honest (αφου τα δύο Byzantine έχουν ειδοποιήσει για έξοδο) και έτσι τα στέλνουμε στις δύο από τις τρεις εξόδους και συγκεκριμένα στις πιο κοντινές τους. Αν ένα από τα δύο επιβεβαιώσει τότε αυτή είναι η έξοδος, αλλιώς η έξοδος είναι εκείνη που δεν στείλαμε κάποιο ρομπότ.



Σχήμα 13: Φάση 1 - Τα $α_0$, $α_1$ και $α_3$ ανακοινώνουν εξόδους.

- ## Φάση 2

Εφόσον διαθέτουμε μία έξοδο και δύο Byzantine γίνεται εύκολα αντιληπτό ότι μπορούν σε κάθε φάση να βρεθούν μηδέν, μία, δύο και τρεις έξοδοι. Παραπάνω δεν γίνεται καθώς ο μέγιστος αριθμός εξόδων είναι όταν βρει το honest την αληθινή έξοδο και τα δύο Byzantine βρουν δύο ξεχωριστές εξόδους.

— Μηδέν ρομπότ βρίσκουν την έξοδο

Σε αυτή την περίπτωση προχωράμε στην επόμενη φάση, η οποία είναι η τρία. Πρακτικά μετακινούμε όλα τα ρομπότ κατά θ με φορά αντίθετη των δεικτών του ρολογιού (ccw). Εφόσον ούτε στην φάση δύο βρέθηκε η έξοδος καταλαβαίνουμε ότι από το τόξο που βρίσκεται η έξοδος έχουν περάσει δύο Byzantine. Ένα στην φάση ένα και ένα στην φάση δύο. Επομένως καταλαβαίνουμε ότι τα δύο Byzantine θα είναι συνεχόμενα, δηλαδή της μορφής $α_ν$ και $α_{ν+1}$.

— Ένα ρομπότ ανακοινώνει έξοδο

Έστω ότι την ανακοινώνει το $α_ν$. Τότε από την έξοδο έχει περάσει και το $α_{ν+1}$. Άρα τουλάχιστον ένα από τα $α_ν$ και $α_{ν+1}$ είναι Byzantine και η έξοδος βρίσκεται στο διάστημα $\{\frac{(ν)\cdot 2π}{5} - \frac{(ν+2)\cdot 2π}{5}\} \rightarrow \{S_ν - S_{ν+2}\}$.

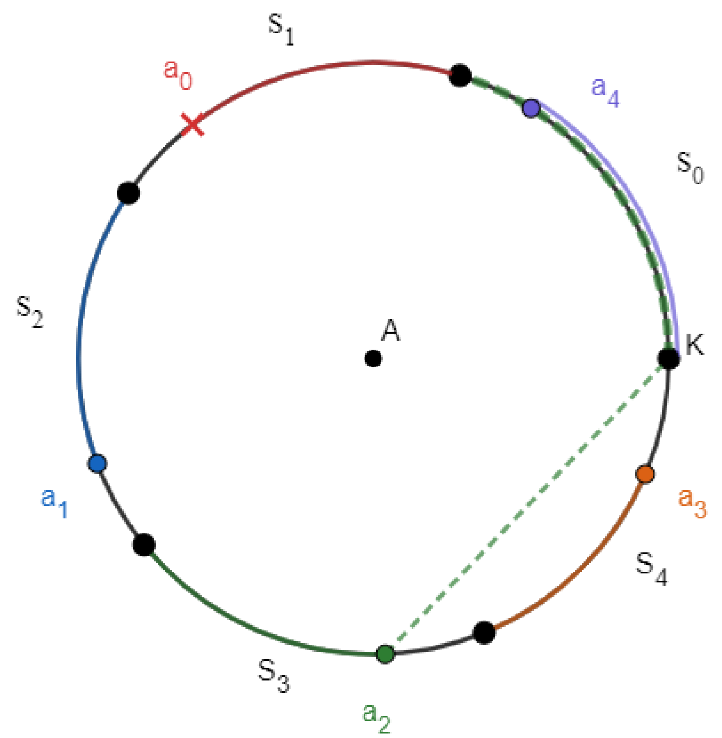
Εξήγηση:

- Έστω ότι είναι και τα δύο Byzantine. Άρα στο τόξο $S_{ν+1}$ έχουν περάσει μόνο Byzantine (στην φάση ένα πέρασε το $α_{ν+1}$ και στην φάση δύο πέρασε το $α_ν$) ενώ από όλα τα υπόλοιπα τόξα έχει περάσει τουλάχιστον ένα honest χωρίς να έχει βρει έξοδο, και άρα η έξοδος βρίσκεται στο τόξο $S_{ν+1}$.

- Έστω ότι το $α_{ν+1}$ είναι honest και συνεπώς το $α_ν$ είναι Byzantine. Το $α_ν$ έχει περάσει από τα τόξα $S_ν$ και $S_{ν+1}$. Από το $S_{ν+1}$ έχει περάσει το $α_{ν+1}$ που είναι honest και δεν βρήκε έξοδο και άρα δεν βρίσκεται η έξοδος σε αυτό το τόξο. Στο τόξο $S_ν$ έχει περάσει εκτός από το $α_ν$ και το $α_{ν-1}$ το οποίο δεν ξέρουμε τι είναι και έτσι μπορεί να βρίσκεται σε αυτό το τόξο η έξοδος. Από τα υπόλοιπα τόξα έχουν περάσει τουλάχιστον ένα honest και δεν έχει βρεθεί έξοδος και άρα δεν βρίσκεται σε αυτά η έξοδος. Αν σε αυτά είχαν περάσει δύο Byzantine τότε μαζί με το $α_ν$ θα είχαμε στο σύνολο τρία Byzantine το οποίο είναι άτοπο. Από τα παραπάνω καταλήγουμε ότι η έξοδος βρίσκεται στο τόξο $S_ν$ και αφού δεν την βρήκε το $α_{ν-1}$ σημαίνει ότι αυτό είναι το δεύτερο Byzantine.

- Αν το $α_ν$ είναι honest έχουμε βρει την πραγματική έξοδο στο τόξο $S_{ν+1}$.

Στέλνουμε το $α_{ν+2}$ στην έξοδο του $α_ν$. Τα υπόλοιπα συνεχίζουν κανονικά την κίνηση τους. Στο τέλος κίνησης θα έχουν περάσει συνολικά από την έξοδο του $α_ν$ πέντε ρομπότ και έτσι αν τρία στο σύνολο συμφωνήσουν την ίδια έξοδο αυτή είναι η σωστή. Σε αντίθετη περίπτωση η έξοδος απορρίπτεται και το $α_ν$ είναι Byzantine και σαν αποτέλεσμα απομένει ένα Byzantine. Με βάση τον παρακάτω πίνακα που δείχνει τα ρομπότ που έχουν περάσει από κάθε τόξο παρατηρούμε ότι από όλα εκτός από το $S_ν$ έχουν περάσει τρία ρομπότ (εκτός του $α_ν$) από τα οποία τουλάχιστον δύο είναι honest (έχει απομείνει μόνο ένα Byzantine) και έτσι αν σε κάποιο από αυτά βρίσκεται η αληθινή έξοδος θα πρέπει να ανακοινωθεί από τουλάχιστον δύο ρομπότ. Αν στο $S_ν$ δεν βρεθεί έξοδος, τότε δεν μπορεί να βρίσκεται σε αυτό η έξοδος καθώς τουλάχιστον ένα από τα δύο είναι honest. Αν βρεθεί έξοδος τότε αυτή είναι η αληθινή, μόνο εάν δεν έχει βρεθεί άλλη έξοδο σε άλλο σημείο.(Θεώρημα 1):



Σχήμα 14: Φάση 2 - Μόνο το α_0 ανακοινώνει έξοδο.

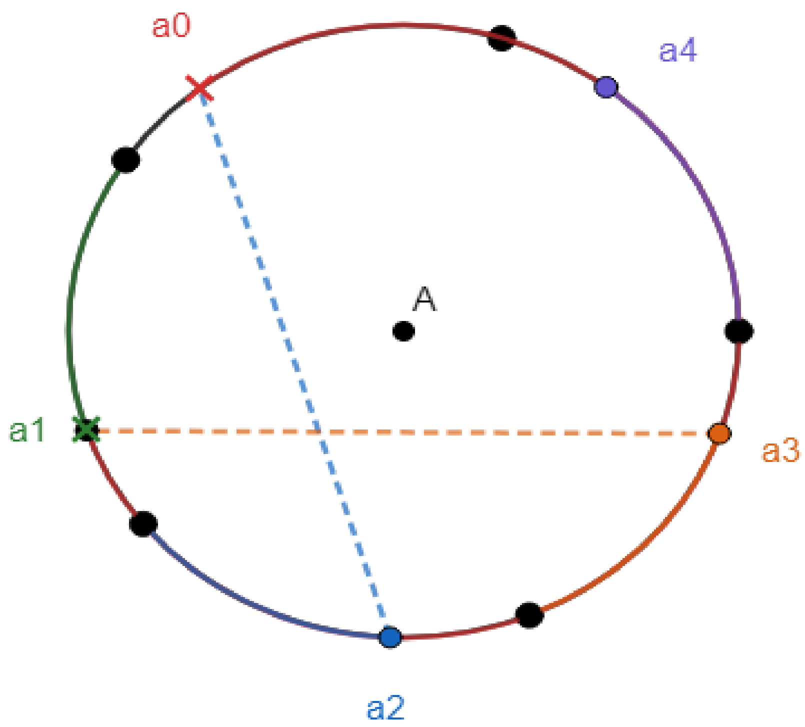
Πίνακας 7

Κινήσεις των ρομπότ για την φάση δύο όταν ένα ρομπότ ανακοινώνει ότι βρήκε την έξοδο.

Χρόνος Τόξο	Γύρος 1($\frac{2\pi}{5}$)	Γύρος 2 ($\frac{4\pi}{5}$)	Γύρος 3 ($\frac{6\pi}{5}$)	Γύρος 4 ($\frac{8\pi}{5}$)
S_v	α_v	α_{v+4}	α_{v+3}	--
S_{v+1}	α_{v+1}	α_v	α_{v+4}	α_{v+3}
S_{v+2}	α_{v+2}	α_{v+1}	--	α_{v+4}
S_{v+3}	α_{v+3}	α_{v+2}	α_{v+1}	--
S_{v+4}	α_{v+4}	α_{v+3}	--	α_{v+11}
Έξοδος που βρήκε το α_v	α_{v+1}	α_v	α_{v+4}	α_{v+2}, α_{v+3}

— Δύο συνεχόμενα ρομπότ ανακοινώνουν έξοδο

Έστω $α_ν$ και $α_{ν+1}$ (δηλαδή συνεχόμενα). Στέλνουμε το $α_{ν+2}$ στην έξοδο του $α_ν$ και το $α_{ν+3}$ στην έξοδο του $α_{ν+1}$. Τα υπόλοιπα συνεχίζουν κανονικά. Στο τέλος της κίνησης θα έχουν περάσει συνολικά τέσσερα ρομπότ από την έξοδο του $α_ν$ και πέντε από την έξοδο του $α_{ν+1}$. Άρα αν ακόμα δύο συμφωνήσουν με το $α_{ν+1}$, τότε η σωστή έξοδος είναι του $α_ν$. Διαφορετικά απορρίπτεται και το $α_{ν+1}$ είναι Byzantine. Επομένως απομένει ένα Byzantine και έτσι αν τουλάχιστον ένα έχει συμφωνήσει με το $α_0$ θα σημαίνει ότι αυτό είναι honest και η έξοδος που ανακοίνωσε είναι η σωστή. Διαφορετικά η έξοδος του απορρίπτεται και το $α_ν$ είναι Byzantine. Επομένως τα δύο Byzantine είναι τα $α_ν$ και $α_{ν+1}$ και τα υπόλοιπα είναι honest και όπου βρουν έξοδο αυτή θα είναι η σωστή.



Σχήμα 15: Φάση 2 - Τα $α_0$ και $α_1$ ανακοινώνουν εξόδους.

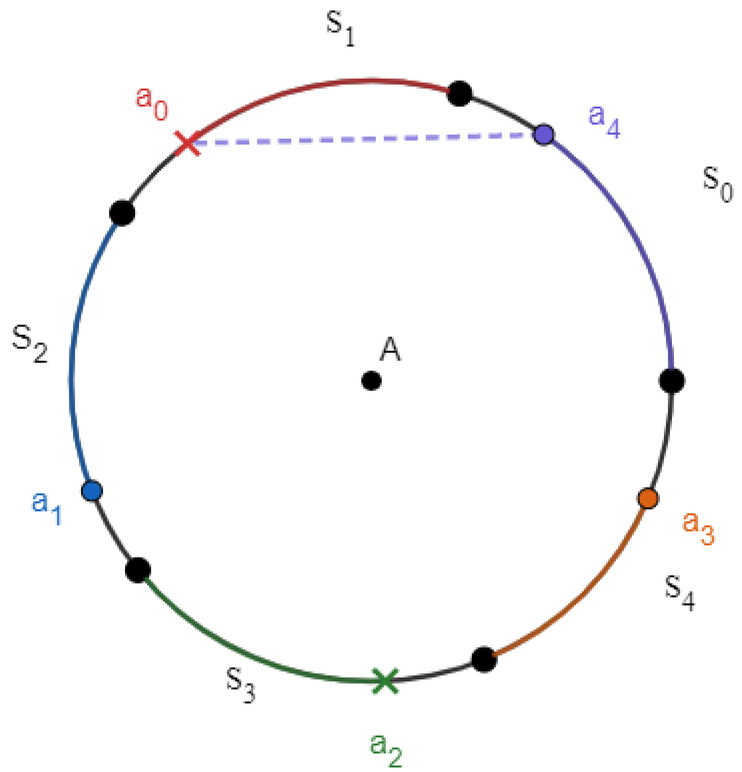
<u>Πίνακας 8</u>

Κινήσεις των ρομπότ για την φάση δύο όταν δύο συνεχόμενα ρομπότ ανακοινώνουν ότι βρήκαν έξοδο.

Τόξο / Χρόνος	Γύρος 1 ($\frac{2\pi}{5}$)	Γύρος 2 ($\frac{4\pi}{5}$)	Γύρος 3 ($\frac{6\pi}{5}$)	Γύρος 4 ($\frac{8\pi}{5}$)
S_v	α_v	α_{v+4}	--	--
S_{v+1}	α_{v+1}	α_v	α_{v+4}	α_{v+3}
S_{v+2}	α_{v+2}	α_{v+1}	--	α_{v+4}
S_{v+3}	α_{v+3}	α_{v+2}	--	--
S_{v+4}	α_{v+4}	α_{v+3}	--	--
Έξοδος που βρήκε το α_v	α_{v+1}	α_v	α_{v+4}	α_{v+2}
Έξοδος που βρήκε το α_{v+1}	α_{v+2}	α_{v+1}, α_v	--	α_{v+4}, α_{v+3}

— Δύο μη συνεχόμενα ρομπότ ανακοινώνουν έξοδο

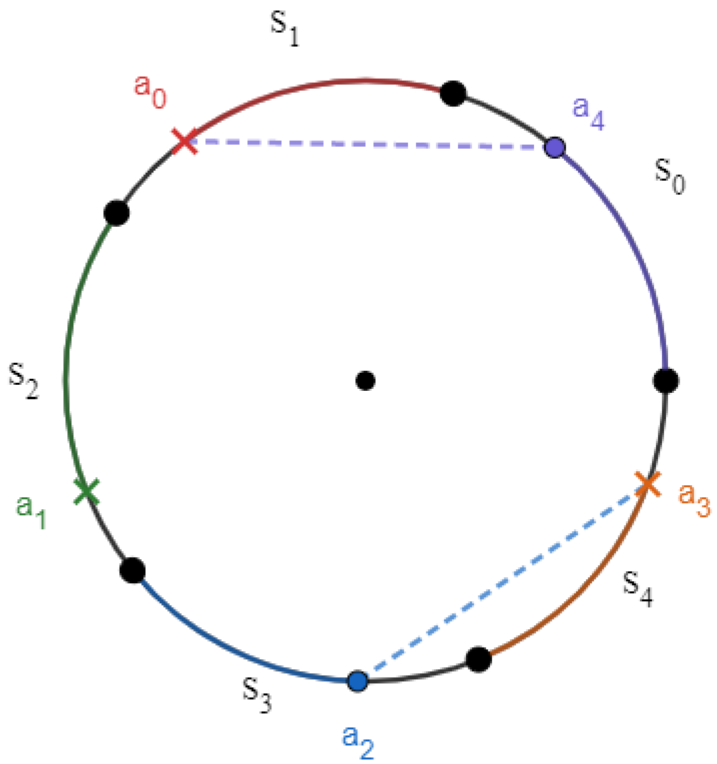
Έστω ότι αυτά είναι της μορφής α_ν και α_κ με $\kappa \neq \nu \pm 1$. Αν ήταν και τα δύο byzanines τότε από όλα τα σημεία του κύκλου έχει περάσει τουλάχιστον ένα honest και δεν έχει βρει κανένα έξοδο, το οποίο είναι άτοπο. Άρα είτε το α_ν είτε το α_κ είναι Byzantine. Ακόμα από τις δύο εξόδους έχουν περάσει και τα $\alpha_{\nu+1}$ και $\alpha_{\kappa+1}$ αντίστοιχα, τα οποία δεν βρήκαν έξοδο και έτσι και ένα από αυτά είναι Byzantine. Το ένα ρομπότ που απομένει είναι honest. Το στέλνουμε σε μία από τις δύο εξόδους αν αυτό επιβεβαιώσει αυτή είναι η έξοδος, αλλιώς είναι η άλλη.



Σχήμα 16: Φάση 2 - Το $α_0$ και το $α_2$ ανακοινώνουν εξόδους.

— Τρία ρομπότ ανακοινώνουν έξοδο

Τρία ρομπότ ανακοινώνουν έξοδο. Αφού υπάρχουν μόνο δύο Byzantine σημαίνει ότι τουλάχιστον ένα από τα τρία είναι honest. Αν δύο ήταν honest τότε θα έχουν βρεθεί δύο αληθινές έξοδοι το οποίο είναι άτοπο. Άρα ακριβώς ένα είναι honest και τα υπόλοιπα δύο είναι Byzantine. Τα υπόλοιπα δύο που δεν ανακοίνωσαν έξοδο είναι honest (αφου τα δύο Byzantine έχουν ειδοποιήσει για έξοδο) και έτσι τα στέλνουμε στις δύο από τις τρεις εξόδους με βάση το ποια βρίσκεται σε μικρότερη απόσταση. Αν ένα από τα δύο επιβεβαιώσει τότε αυτή είναι η έξοδος αλλιώς η έξοδος είναι εκείνη που δεν στείλαμε κάποιο ρομπότ. Σημειώνεται ότι αν κάποιο από αυτά που δεν ανακοίνωσαν έξοδο και είναι με βεβαιότητα honest, έχει περάσει από σημείο εξόδου στον προηγούμενο γύρο, αυτή η έξοδος απορρίπτεται χωρίς να χρειαστεί να στείλουμε κάποιο ρομπότ.



Σχήμα 17: Φάση 2 - Τα $α_0$, $α_1$ και $α_3$ ανακοινώνουν εξόδους.

- ## Φάση 3

Εφόσον διαθέτουμε μία έξοδο και δύο Byzantine γίνεται εύκολα αντιληπτό ότι μπορούν σε κάθε φάση να ανακοινωθούν μηδέν, μία, δύο και τρεις έξοδοι. Παραπάνω δεν γίνεται καθώς ο μέγιστος αριθμός εξόδων είναι όταν βρει το honest την αληθινή έξοδο και τα δύο Byzantine ανακοινώσουν δύο ξεχωριστές εξόδους.

— Μηδέν ρομπότ ανακοινώνουν έξοδο

Στην φάση τρία που βρισκόμαστε τώρα έχουν περάσει τρία ρομπότ από κάθε τόξο και αφού έχουμε δύο Byzantine, σημαίνει ότι από κάθε τόξο θα έχει περάσει τουλάχιστον ένα honest και επομένως δεν γίνεται να μην βρεθεί έξοδος στην φάση τρία.

— Ένα ρομπότ ανακοινώνει έξοδο

Ένα ρομπότ ανακοινώνει έξοδο. Αυτό θα είναι honest καθώς σε άλλη περίπτωση θα έπρεπε να είχε βρεθεί έξοδος και από άλλο ρομπότ που θα είναι honest.

— Δύο ρομπότ ανακοινώνουν έξοδο

Δύο ρομπότ ανακοινώνουν έξοδο. Αφού έχουμε φτάσει στη φάση όπου από κάθε σημείο του κύκλου έχουν περάσει 3 ρομπότ, αυτό θα σημαίνει ότι τα Byzantine θα είναι διαδοχικά. Αν δεν είναι διαδοχικά τότε στην φάση δύο από κάθε σημείου του κύκλου θα είχε περάσει τουλάχιστον ένα honest και επομένως θα έπρεπε να είχε βρεθεί η έξοδος. Άρα οι δύο έξοδοι θα είναι της μορφής $α_κ$ και $α_{κ+1}$ ή $α_κ$ και $α_{κ+2}$, με το honest να είναι το $α_κ$. Αν ήταν το $α_{κ+1}$ ή $α_{κ+2}$ τότε η έξοδος θα έπρεπε να είχε ήδη βρεθεί από το $α_{κ+3}$ το οποίο είναι honest.

— Τρία ρομπότ ανακοινώνουν έξοδο

Τρία ρομπότ ανακοινώνουν έξοδο. Όπως παραπάνω θα είναι της μορφής $α_κ$, $α_{κ+1}$, $α_{κ+2}$ με honest το $α_κ$ με ίδιο τρόπο όπως παραπάνω.

2.1.2 Ορθότητα Αλγορίθμου

Η ορθότητα του αλγορίθμου αναλύθηκε μέσα στην ανάλυση των φάσεων. Συνοπτικά στέλνουμε τόσα ρομπότ στις ανακοινώσεις, ώστε στο τέλος να έχουν απορριφθεί όλες εκτός από μία. Για να απορριφθεί μία έξοδος, χρειάζεται να την απορρίψουν $f + 1$ ρομπότ. Εφόσον διαθέτουμε f Byzantine, σημαίνει ότι τουλάχιστον ένα από τα $f + 1$ θα είναι honest και επομένως αυτή η ανακοίνωση απορρίπτεται εντελώς.

2.2 Περίπτωση (n,2,2): *n* ρομπότ, *2* Byzantine

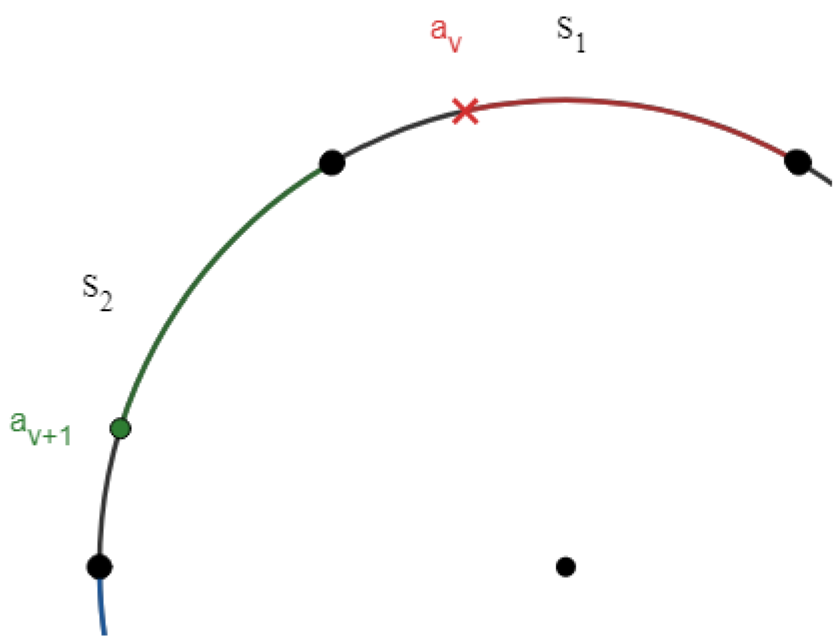
2.2.1 Περιγραφή Αλγορίθμου

Σε αυτό το σημείο θα γενικεύσουμε την λύση για το (5,2,2) στην περίπτωση που έχουμε n ρομπότ στο σύνολο από τα οποία τα δύο είναι Byzantine. Δηλαδή θα γίνει η μελέτη της περίπτωσης (n,2,2). Οι φάσεις παραμένουν όπως και στην περίπτωση των πέντε στο σύνολο ρομπότ. Στην νέα περίπτωση χωρίζουμε τον κύκλο σε n μέρη και η θέση που μετακινείται το $α_κ$ ρομπότ στην φάση 0 είναι η $κ·θ$, όπου $θ = \frac{2π}{n}$.

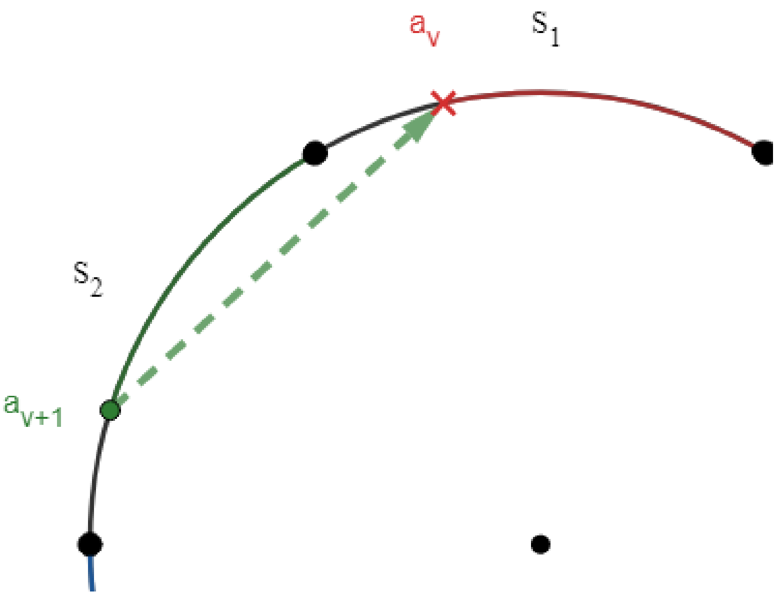
- **Ανάλυση Φάσης 1**

— Ένα ρομπότ ανακοινώνει έξοδο

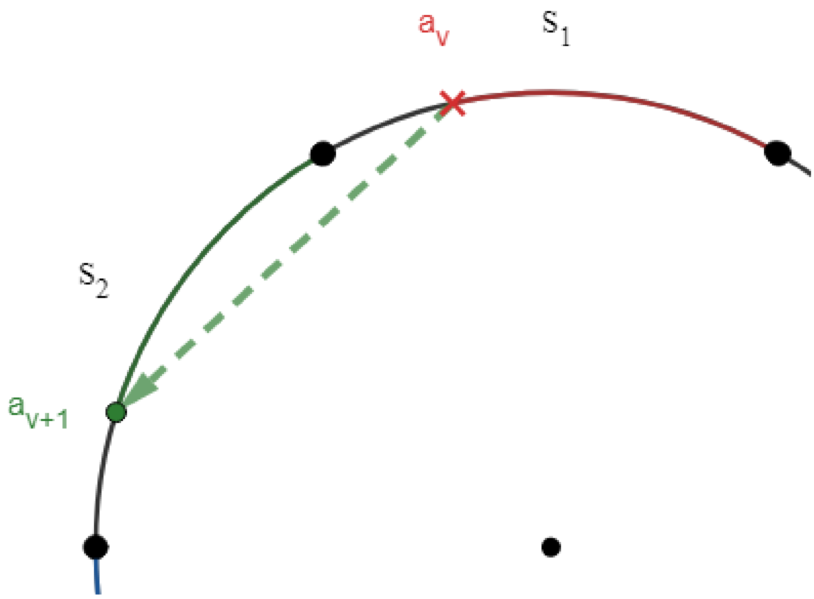
Στέλνουμε το $α_{ν+1}$ στην έξοδο του $α_ν$ με χορδή, χωρίς να ολοκληρώσει την κίνηση που θα εκτελούσε αν δεν ανακοινωνόταν έξοδος στην φάση ένα. Στη συνέχεια το στέλνουμε μέσω χορδής στο σημείο που είχε σταματήσει στον πρώτο γύρο και συνεχίζει κανονικά την πορεία του. Τα υπόλοιπα συνεχίζουν κανονικά για τρεις ακόμα γύρους. Στο τέλος θα έχουν περάσει στο σύνολο τέσσερις γύροι. Από την έξοδο που ανακοίνωσε το $α_ν$ θα έχουν περάσει στο σύνολο πέντε ρομπότ και έτσι με σιγουριά θα γνωρίζουμε αν αυτή είναι η αληθινή έξοδος. Αν αυτή απορριφθεί τότε το $α_ν$ είναι Byzantine και ως αποτέλεσμα απομένει ένα Byzantine. Εφόσον έχουμε ένα Byzantine χρειαζόμαστε πλέον τρία ρομπότ για να επιβεβαιωθεί ή απορριφθεί μία έξοδος. Από όλα τα τόξα περνάνε τουλάχιστον τρία ρομπότ εκτός από το $S_{ν+3}$. Σε αυτό περνάνε μόνο δύο καθώς υπό φυσιολογική κίνηση θα πέρναγαν τα $α_ν$ και $α_{ν+1}$. Το $α_ν$ είναι Byzantine και έτσι δεν έχει νόημα να το στείλουμε σε αυτό το τόξο και το $α_{ν+1}$ με το να σταλθεί στην έξοδο του $α_ν$ προλαβαίνει να αναζητήσει την έξοδο μέχρι το τόξο $S_{ν+2}$. Επομένως αν ανακοινωθεί έξοδος σε οποιαδήποτε τόξο πέρα του $S_{ν+3}$ μπορούμε με βεβαιότητα να γνωρίζουμε αν είναι η σωστή ή όχι. Αν δεν ανακοινωθεί έξοδος στο S_{V+3} τότε δεν μπορεί να βρίσκεται η έξοδος σε αυτό το τόξο καθώς έχουν περάσει δύο ρομπότ από τα οποία το ένα είναι σίγουρα honest. Αν ανακοινωθεί έξοδος, τότε αυτή είναι η αληθινή, μόνο αν δεν έχει βρεθεί η αληθινή έξοδος σε άλλο σημείο. (Θεώρημα 1)



Σχήμα 18 Φάση 1- Περιγραφή της κίνησης του ρομπότ $α_{ν+1}$. Το $α_ν$ ανακοινώνει έξοδο.



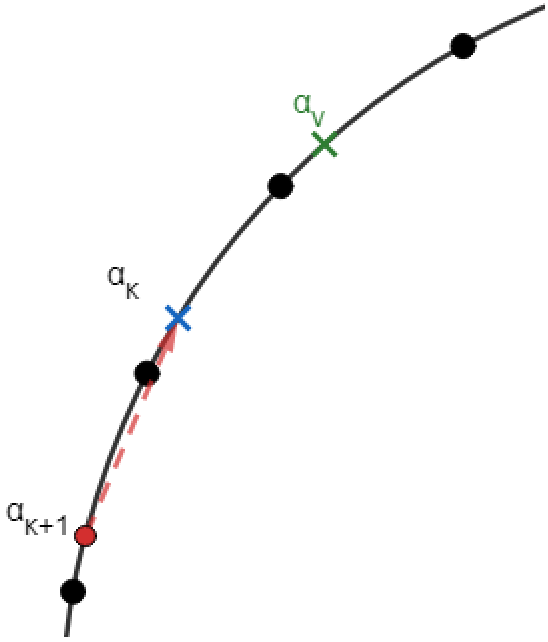
Σχήμα 19 Φάση 1- Περιγραφή της κίνησης του ρομπότ $α_{ν+1}$. Το $α_{ν+1}$ μετακινείται στην έξοδο του $α_ν$.



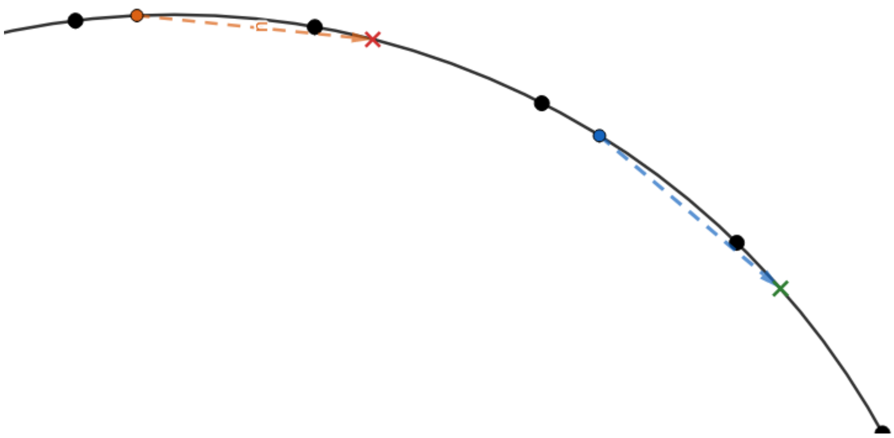
Σχήμα 20 Φάση 1- Περιγραφή της κίνησης του ρομπότ $α_{ν+1}$. Το το $α_{ν+1}$ επιστρέφει στο σημείο που είχε σταματήσει.

— Δύο ρομπότ ανακοινώνουν έξοδο

Έστω ότι ανακοινώνουν έξοδο τα $α_ν$ και $α_κ$. Τότε στέλνουμε τα $α_{ν+1}$ και $α_{κ+1}$ στην έξοδο του $α_ν$ και $α_κ$ αντίστοιχα. Αυτό γίνεται μόνο αν τα $α_{ν+1}$ ή $α_{κ+1}$ δεν ταυτίζονται με κάποιο από τα $α_ν$ ή $α_κ$. Τα υπόλοιπα συνεχίζουν κανονικά την πορεία τους για ακόμα τρεις γύρους. Στο τέλος από κάθε ρομπότ θα έχουν περάσει στο σύνολο πέντε ρομπότ (τέσσερα συν την απόρριψη από το άλλο ρομπότ που έχει βρει έξοδο). Έτσι με βεβαιότητα μπορούμε για κάθε μία να αποφανθούμε αν είναι η αληθινή ή ψευδή. Αν απορριφθούν και οι δύο, τότε τα $α_ν$ και $α_κ$ είναι τα δύο Byzantine και όλα τα υπόλοιπα είναι honest και συνεπώς όποια έξοδο βρουν, αυτή θα είναι και η αληθινή.



Σχήμα 21: Φάση 1 - Περιγραφή της κίνησης όταν δύο ρομπότ ανακοινώνουν έξοδο. Το $α_κ$ ταυτίζεται με το $α_{ν+1}$.



Σχήμα 22: Φάση 1 - Περιγραφή της κίνησης όταν δύο ρομπότ ανακοινώνουν έξοδο. Δεν υπάρχει ταύτιση μεταξύ των ρομπότ.

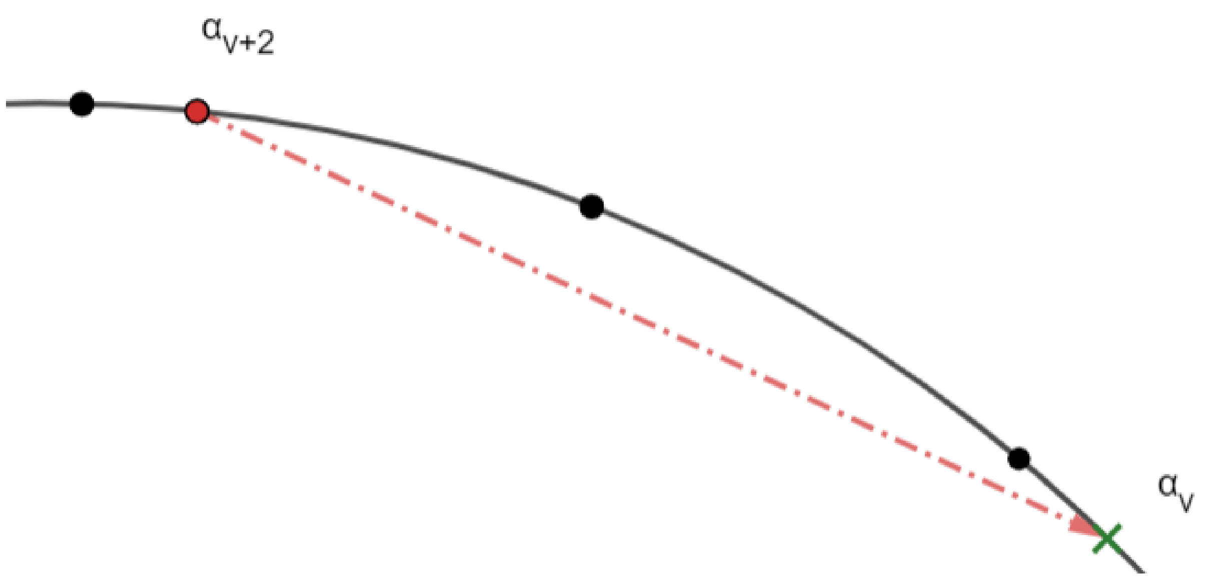
— Τρία ρομπότ ανακοινώνουν έξοδο

Εφόσον έχουμε δύο Byzantine και έχουν ανακοινωθεί τρεις έξοδοι, οι δύο θα προέρχονται από Byzantine και η μία από honest. Να έχουν ανακοινωθεί πάνω από μία από honest δει είναι εφικτό. Όσα δεν βρήκαν έξοδο ως αποτέλεσμα θα είναι honest και τα στέλνουμε σε δύο από τις τρεις εξόδους (στις πιο κοντινές τους). Αν κάποιο από αυτά επιβεβαιώσει την έξοδο, αυτή θα είναι και η αληθινή. Αν κανένα δεν επιβεβαιώσει, τότε σημαίνει ότι οι δύο έξοδοι που έλεγξαν προέρχονται από Byzantine και αυτή που δεν ελέγχθηκε, προέρχεται από honest και άρα αυτή είναι η σωστή.

- **Ανάλυση Φάσης 2**

— Ένα ρομπότ ανακοινώνει έξοδο

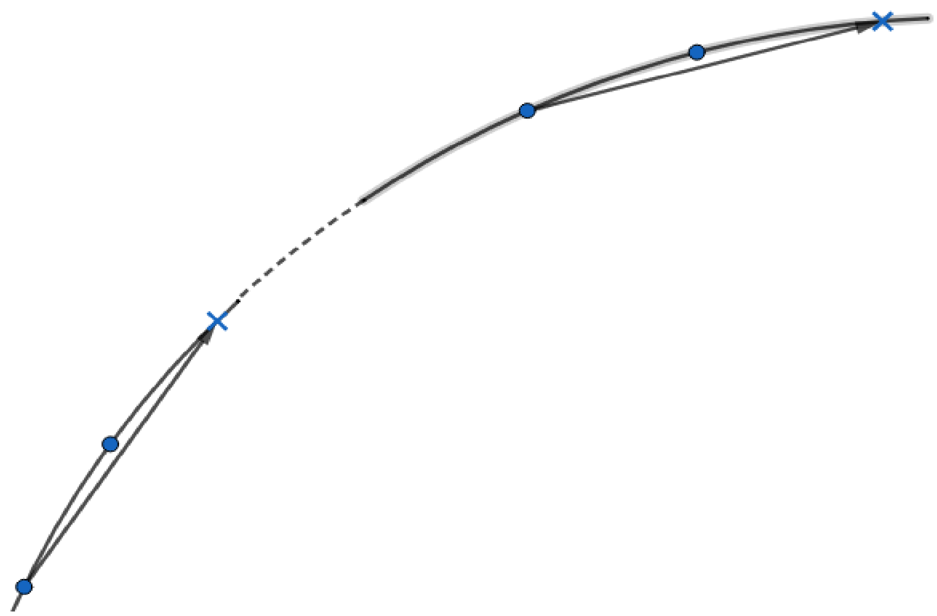
Έστω ότι την ανακοινώνει το $α_ν$. Τότε στέλνουμε το $α_{ν+2}$ στην έξοδο του $α_ν$. Τα υπόλοιπα συνεχίζουν κανονικά για ακόμα δύο γύρους. Στο τέλος από την έξοδο που ανακοίνωσε το $α_ν$ θα έχουν περάσει στο σύνολο πέντε ρομπότ και έτσι με σιγουριά μπορούμε να αποφανθούμε για την έξοδο του. Αν αυτή απορριφθεί, τότε το $α_ν$ είναι Byzantine και απομένει ακόμα ένα Byzantine. Από κάθε σημείο του κύκλου έχουν περάσει τρία ρομπότ και έτσι με σιγουριά μπορούμε να βρούμε την έξοδο.



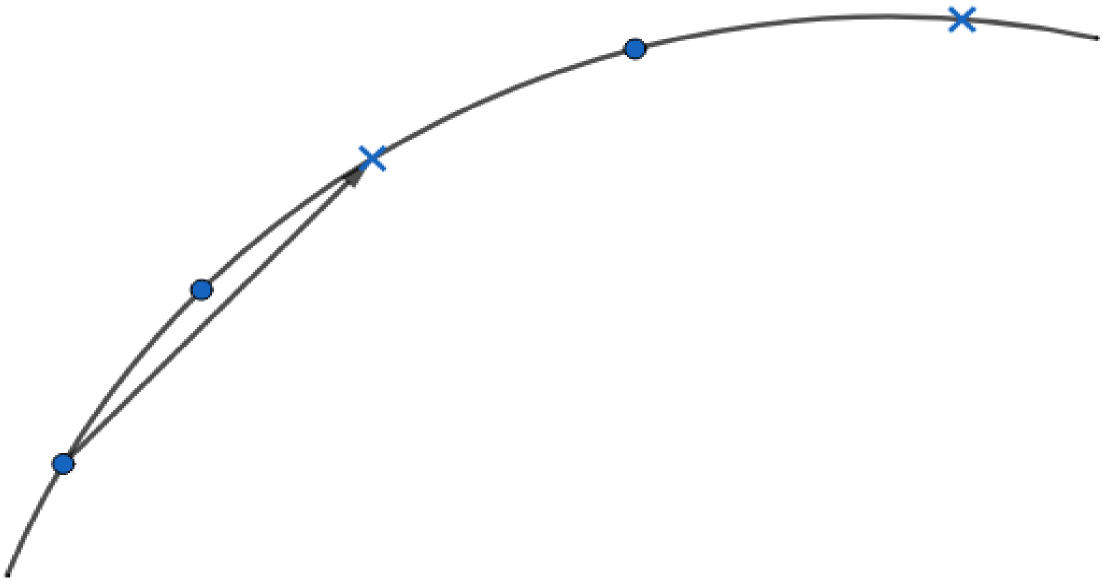
Σχήμα 23: Φάση 1 - Περιγραφή της κίνησης του $α_{ν+2}$ στην έξοδο του $α_ν$.

— Δύο ρομπότ ανακοινώνουν έξοδο

Έστω ότι ανακοινώνουν έξοδο τα $α_ν$ και $α_κ$. Τότε στέλνουμε τα $α_{ν+2}$ και $α_{κ+2}$ στην έξοδο του $α_ν$ και $α_κ$ αντίστοιχα, εκτός αν κάποιο από αυτά ταυτίζεται με το $α_ν$ ή $α_κ$. Τα υπόλοιπα συνεχίζουν για ακόμα δύο γύρους. Στο τέλος για κάθε μία από τις δύο εξόδους θα έχουμε γνώμη από συνολικά πέντε ρομπότ και έτσι με σιγουριά αυτή επιβεβαιώνεται η απορρίπτεται. Αν απορριφθούν και οι δύο, τότε τα $α_ν$ και $α_κ$ είναι Byzantine και έτσι τα υπόλοιπα είναι honest και όποια έξοδο βρουν αυτή θα είναι και η σωστή.



Σχήμα 24: Φάση 2 - Αναπαράσταση χωρίς να ταυτίζεται κάποιο από τα $α_{κ+2}$ ή $α_{ν+2}$με τα $α_κ$ ή $α_ν$.



Σχήμα 25: Φάση 2 - Αναπαράσταση με το $α_{ν+2}$ να ταυτίζεται με το $α_κ$.

— Τρία ρομπότ ανακοινώνουν έξοδο

Εφόσον έχουμε δύο Byzantine και έχουν ανακοινωθεί τρεις έξοδοι, οι δύο θα προέρχονται από Byzantine και η μία από honest. Όσα δεν ανακοίνωσαν έξοδο ως αποτέλεσμα θα είναι honest και τα στέλνουμε σε δύο από τις τρεις εξόδους (στις πιο κοντινές τους). Αν κάποιο από αυτά επιβεβαιώσει την έξοδο, αυτή θα είναι και η αληθινή. Αν κανένα δεν επιβεβαιώσει, τότε σημαίνει ότι οι δύο έξοδοι που έλεγξαν προέρχονται από Byzantine και αυτή που δεν ελέγχθηκε από honest και άρα αυτή είναι η σωστή. Σημειώνεται ότι μπορεί να μην χρειαστεί να στείλουμε κάποιο honest σε κάποια έξοδο, αν αυτό έχει περάσει στον προηγούμενο γύρο από οποιαδήποτε έξοδο που ανακοινώθηκε τώρα, καθώς αυτή η έξοδος θα απορριφθεί αυτόματα.

• Ανάλυση Φάσης 3

— Ένα ρομπότ ανακοινώνει έξοδο

Στην φάση τρία από κάθε σημείο του κύκλου έχουν περάσει τρία ρομπότ από τα οποία τουλάχιστον το ένα είναι honest. Επομένως από κάθε σημείο του κύκλου έχει περάσει τουλάχιστον ένα honest και επομένως η αληθινή έξοδος θα ανακοινωθεί από το honest. Εφόσον υπάρχει μόνο μία ανακοίνωση για έξοδο, τότε αυτόματα η έξοδος που ανακοινώθηκε είναι η αληθινή. Αν δεν ήταν τότε θα σημαίνει ότι κάποιο honest πέρασε από την έξοδο και δεν ειδοποίησε, το οποίο είναι άτοπο.

— Δύο ρομπότ ανακοινώνουν έξοδο

Δύο ρομπότ ανακοινώνουν έξοδο. Αφού έχουμε φτάσει στη φάση όπου από κάθε σημείο του κύκλου έχουν περάσει τρία ρομπότ, αυτό θα σημαίνει ότι τα Byzantine θα είναι διαδοχικά. Αν δεν είναι διαδοχικά τότε στην φάση δύο από κάθε σημείου του κύκλου θα είχε περάσει τουλάχιστον ένα honest και επομένως θα έπρεπε να είχε βρεθεί η έξοδος. Άρα οι δύο έξοδοι θα είναι της μορφής α_κ και $\alpha_{\kappa+1}$ ή α_κ και $\alpha_{\kappa+2}$, με το honest να είναι το α_κ. Αν ήταν το $\alpha_{\kappa+1}$ ή $\alpha_{\kappa+2}$ τότε η έξοδος θα έπρεπε να είχε ήδη βρεθεί από το $\alpha_{\kappa+3}$ το οποίο είναι honest.

— Τρία ρομπότ ανακοινώνουν έξοδο

Τρία ρομπότ ανακοινώνουν έξοδο. Όπως παραπάνω θα είναι της μορφής α_κ, $\alpha_{\kappa+1}$, $\alpha_{\kappa+2}$ με honest το α_κ με ίδιο τρόπο όπως παραπάνω.

2.1.2 Ορθότητα Αλγορίθμου

Η ορθότητα του αλγορίθμου αναλύθηκε μέσα στην ανάλυση των φάσεων. Συνοπτικά στέλνουμε τόσα ρομπότ στις ανακοινώσεις, ώστε στο τέλος να έχουν απορριφθεί όλες εκτός από μία. Για να απορριφθεί μία έξοδος, χρειάζεται να την απορρίψουν $f + 1$ ρομπότ. Εφόσον διαθέτουμε f Byzantine, σημαίνει ότι τουλάχιστον ένα από τα $f + 1$ θα είναι honest και επομένως αυτή η ανακοίνωση απορρίπτεται εντελώς.

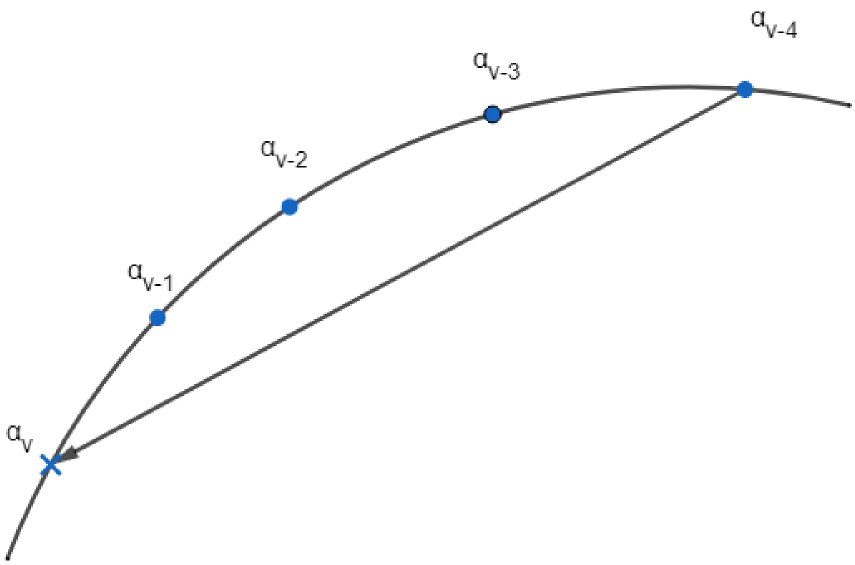
2.3 Περίπτωση (n,f,2): *n* ρομπότ, *f* faulty, *2* Byzantine

2.3.1 Περιγραφή Επίλυσης

Σε αυτή την περίπτωση διαθέτουμε n στο σύνολο ρομπότ. Επιπλέον διαθέτουμε δύο Byzantine. Η διαφορά με την προηγούμενη περίπτωση είναι ότι διαθέτουμε και $f - 2$ crash ρομπότ. Αυτά υπενθυμίζουμε ότι μπορεί να σταματήσουν να λειτουργούν και να παραμείνουν αδρανή οποιαδήποτε στιγμή. Επομένως δεν μπορούμε να στηριχτούμε σε αυτά στο να βρουν την έξοδο, καθώς υπάρχει η πιθανότητα να σταματήσουν την λειτουργία τους πριν περάσουν από αυτή. Παρακάτω παρουσιάζεται ο αλγόριθμος και στη συνέχεια θα αναλυθεί.

2.3.2 Αλγόριθμος

1. Μετακίνηση του $α_κ$ από το κέντρο του κύκλου στο τόξο γωνίας κ·θ, όπου $θ = \frac{2π}{n}$.
2. Μετακίνηση του κάθε ρομπότ με φορά αντίθετη της φοράς των δεικτών του ρολογιού (counterclockwise) για $f - 1$ γύρους.
3. Αν ανακοινωθούν έξοδοι τότε στέλνουμε τα ρομπότ τέσσερα Sectors μακριά από τις εξόδους.
4. Αλλιώς συνεχίζουμε για ένα γύρο.
5. Αν ανακοινωθούν έξοδοι τότε στέλνουμε τα ρομπότ τρία Sectors μακριά από τις εξόδους.
6. Αλλιώς συνεχίζουμε για ένα γύρο ακόμα.



Σχήμα 26: Το ρομπότ τέσσερα τόξα μακριά πάει στην έξοδο.

2.3.3 Θεωρήματα – Αποδείξεις

Θεώρημα 2

Αν ανακοινωθούν τρεις έξοδοι σε διαφορετικά Sectors τότε η λύση βρίσκεται σε χρόνο $1 + \frac{2π}{n}(f + 1)$.

Απόδειξη:

Στον παραπάνω χρόνο από κάθε έξοδο έχουν περάσει $f + 1$ ρομπότ. Και από τις τρεις μαζί έχουν περάσει τουλάχιστον $f + 3$ robots (διαφορετικά οι έξοδοι θα ταυτιζόντουσαν) και επομένως έχουν περάσει τουλάχιστον τρία honest. Δύο από τα οποία πρέπει να είναι διαφορετικά από αυτό που έκανε την σωστή ανακοίνωση. Τα οποία θα έχουν ψάξει τα δύο Sectors από τα τρία, είτε με το να επιβεβαιώσουν την σωστή έξοδο είτε με το να διαψεύσουν τις λανθασμένες. Η κάθε έξοδος έχει δύο απορρίψεις από τα ρομπότ που βρήκαν έξοδο σε άλλο σημείο. Επομένως αν ένα από τα δύο honest απορρίψει έξοδο Byzantine αυτή απορρίπτεται καθώς θα έχει απορριφθεί από τρία ρομπότ. Αν επιβεβαιώσει την έξοδο του honest τότε αυτόματα θα απορρίψει τις άλλες δύο εξόδους και καθώς κάθε μία από αυτές θα έχει τρεις συνολικά απορρίψεις, απορρίπτονται οριστικά.

2.3.4 Σενάριο χειρότερης περίπτωση - Worst case scenario

Περίπτωση 1: Καμία έξοδος μετά το βήμα 5

Τότε στον επόμενο γύρο είτε θα ανακοινωθούν μία είτε δύο είτε τρεις έξοδοι σε διαφορετικά Sectors. Αυτό συμβαίνει επειδή ο αλγόριθμος θα έχει τρέξει για $f + 1$ γύρους και επομένως από κάθε Sector θα έχει περάσει τουλάχιστον ένα honest το οποίο θα ανακοινώσει την σωστή έξοδο. Στην περίπτωση που έχουμε μία έξοδο τότε αυτή αναγκαστικά είναι η σωστή καθώς από κάθε σημείο έχει περάσει τουλάχιστον ένα honest. Επομένως αν δεν είναι η σωστή, τότε θα σημαίνει πως ένα honest ρομπότ πέρασε από την έξοδο χωρίς να ειδοποιήσει το οποίο είναι άτοπο. Στην περίπτωση που έχουμε δύο εξόδους θα σημαίνει ότι από την σωστή έξοδο έχουν περάσει f faulty και ένα honest. Από την άλλη έξοδο έχει περάσει σίγουρα ένα Byzantine. Αυτό το Byzantine έχει περάσει και από την άλλη έξοδο και έτσι η μέγιστη διαφορά μεταξύ των δύο εξόδων σε Sectors είναι $f + 1$, με το τόξο της εξόδου με το Byzantine να βρίσκεται μπροστά από την έξοδο με την αληθινή έξοδο. Ακόμα το honest που ανακοινωσε την σωστή έξοδο, πέρασε τελευταίο από το τόξο και επομένως ξεκίνησε $f + 1$ πίσω από την έξοδο και επομένως δεν έχει περάσει από την έξοδο με την λανθασμένη έξοδο. Γνωρίζουμε επιπλέον ότι και οι δύο έξοδοι έχουν απορριφθεί από δύο ρομπότ και χρειάζεται μόνο μία απόρριψη για απορριφθεί τελείως μία από τις δύο. Όπως αναφέραμε το ένα ρομπότ που βρήκε έξοδο δεν έχει περάσει από το άλλο τόξο και επομένως απορρίπτει αυτόματα την μία έξοδο και σαν αποτέλεσμα η μία έξοδος απορρίπτεται από τρία ρομπότ και απορρίπτεται και απομένει μία έξοδος, η οποία είναι η σωστή. Στην περίπτωση με τρεις εξόδους σε διαφορετικά τόξα λύνεται από το θεώρημα 2. Ο χρόνος επίλυσης είναι $1 + \frac{2\pi}{n}(f + 1)$.

- Περίπτωση 2: Υπάρχουν έξοδοι μετά από f γύρους

  - Μία έξοδος χωρίς επιβεβαιώσεις ή διαψεύσεις

Έστω το $α_κ$ ανακοινώνει έξοδο. Εφόσον δεν έχει διαψεύσεις από αυτό έχουν περάσει $f - 1$ faulty. Όμως τα crash είναι $f - 2$ και επομένως πρέπει να υπάρχει τουλάχιστον μία επιβεβαίωση/απόρριψη. Επομένως δεν γίνεται να υπάρξει αυτή η περίπτωση.

  - Μία έξοδος με μία επιβεβαίωση ή διάψευση

Έστω ότι την έξοδο την ανακοινώνει το $α_κ$. Από αυτή την έξοδο έχουν περάσει $f - 2$ crash ρομπότ, και επομένως από τα επόμενα τρία robot τουλάχιστον το ένα είναι honest και συνολικά θα περάσουν από την έξοδο πέντε ρομπότ που δεν είναι crash και έτσι τουλάχιστον τα τρία θα είναι honest και θα ξέρουμε με σιγουριά αν η έξοδος είναι σωστή ή λανθασμένη. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2π}{n}f + 2sin(\frac{3π}{n})$.

  - Δύο έξοδοι στο ίδιο Τόξο

Στην συγκεκριμένη περίπτωση υπάρχουν δύο υποπεριπτώσεις. Αυτές είναι οι εξής:

- Με μία απόρριψη/επιβεβαίωση από έστω το ρομπότ $α_κ$.
  Τότε θα έχουμε $f - 3$ crash να έχουν περάσει. Ακόμα η μία έξοδος ανακοινώθηκε από honest και η άλλη από Byzantine, διαφορετικά θα έπρεπε να είχε βρεθεί έξοδος σε άλλο Sector. Για τα επόμενα τρία ξέρουμε ότι το πολύ το ένα θα είναι crash. Αν το $α_κ$ απέρριψε την σωστή έξοδο τότε είναι Byzantine και επομένως από τα επόμενα τρία τουλάχιστον τα δύο θα είναι honest και θα επιβεβαιώσουν την σωστή έξοδο. Αλλιώς θα είναι honest και τουλάχιστον ένα από τα επόμενα τρία θα είναι honest και θα επιβεβαιώσει την σωστή έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2π}{n}f + 2sin(\frac{3π}{n})$.

- Με καμία απόρριψη/επιβεβαίωση.
  Τότε τα $f - 2$ είναι crash. Από τα επόμενα τρία τα δύο τουλάχιστον είναι honest και θα επιβεβαιώσουν την σωστή έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2π}{n}f + 2sin(\frac{3π}{n})$.

- Δύο έξοδοι σε διαφορετικά Τόξα

Από κάθε έξοδο έχουν περάσει $f - 1$ ακόμα ρομπότ. Τα crash είναι $f - 2$ οπότε αναγκαστικά το καθένα θα έχει από μία απόρριψη/επιβεβαίωση ακόμα. Γίνεται στη μία από τις δύο εξόδους η απόρριψη να ταυτίζεται με το ρομπότ που βρήκε την άλλη έξοδο. Ακόμα η μέγιστη απόσταση που μπορεί να έχουν οι δύο έξοδοι είναι δύο τόξα. Αν απείχαν παραπάνω τότε από την μία έξοδο θα έχουν περάσει $f - 5$ crush και στο σύνολο πέντε no-crash και έτσι αυτή η έξοδος θα είχε απορριφθεί/επιβεβαιωθεί με σιγουριά. Από τα επόμενα δύο το ένα είναι σίγουρα honest και θα δώσει την σωστή έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}f + 2sin(\frac{2\pi}{n})$.

- Τρεις έξοδοι στο ίδιο Τόξο

Τότε έχουμε ένα honest , δύο Byzantine κα $f - 3$ crash. Επομένως από τα επόμενα δύο τουλάχιστον το ένα είναι honest και θα επιβεβαιώσει την σωστή έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}f + 2sin(\frac{2\pi}{n})$.

- Τρεις έξοδοι, οι δύο στο ίδιο Τόξο

Από τα δύο τόξα έχουν περάσει τουλάχιστον $f - 3$ crash. Ακόμα η μία έξοδος έχει ανακοινωθεί από honest .Η απόσταση των δύο τόξων δεν μπορεί να ξεπερνά τα δύο sectors. Για να μην έχει απορριφθεί ήδη η μία έξοδος σημαίνει ότι έχουν περάσει μόνο crash και ρομπότ που έχουν βρει έξοδο. Εφόσον από το καθένα έχουν περάσει $f - 3$ crash απομένει ένα crash. Αν απείχαν παραπάνω από δύο τόξα τότε θα έχουν περάσει πάνω από δύο διαφορετικά crash, το οποίο είναι άτοπο. Επομένως από τα επόμενα τρία τουλάχιστον τα δύο θα είναι honest και θα επιβεβαιώσουν την σωστή έξοδο. Χρόνος $1 + \frac{2\pi}{n}f + 2sin(\frac{3\pi}{n})$.

- Τρεις έξοδοι σε διαφορετικά Τόξα

Λύνεται με βάση το Θεώρημα 2.

— Περίπτωση 3: Υπάρχουν έξοδοι μετά από f-1 γύρους

- Μία έξοδος χωρίς απορρίψεις/επιβεβαιώσεις

Από αυτή την έξοδο έχουν περάσει $f - 2$ crash. Επομένως από τα επόμενα τέσσερα τα δύο τουλάχιστον είναι honest και θα επιβεβαιώσουν την έξοδο αν είναι η σωστή. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$.

- Μία έξοδος με μία απόρριψη/επιβεβαίωση

Από την έξοδο έχουν περάσει $f - 3$ crash. Επομένως από τα επόμενα τέσσερα το ένα το πολύ είναι crash. Επομένως από την έξοδο θα έχουν περάσει σύνολο πέντε ρομπότ από τα οποία τα τρία είναι honest και θα επιβεβαιώσουν ή απορρίψουν την έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$.

- Μία έξοδος με δύο απορρίψεις/επιβεβαιώσεις

Από την έξοδο έχουν περάσει $f - 4$ crash. Επομένως από τα επόμενα τέσσερα τα δύο το πολύ είναι crash. Επομενως από την έξοδο θα έχουν περάσει σύνολο πέντε ρομπότ από τα οποία τα τρία είναι honest και θα επιβεβαιώσουν ή απορρίψουν την έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$.

- Δύο έξοδοι στο ίδιο τόξο χωρίς απόρριψη/επιβεβαίωση

Από αυτό το τόξο έχουν περάσει $f - 3$ crash. Επομένως από τα επόμενα τέσσερα το ένα το πολύ είναι crash. Επομενως από την έξοδο θα έχουν περάσει σύνολο πέντε ρομπότ από τα οποία τα τρία είναι honest και θα επιβεβαιώσουν ή απορρίψουν την έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$.

- Δύο έξοδοι στο ίδιο τόξο με απόρριψη/επιβεβαίωση

Από την έξοδο έχουν περάσει $f - 4$ crash. Επομένως από τα επόμενα τέσσερα τα δύο το πολύ είναι crash. Επομενως από την έξοδο θα έχουν περάσει σύνολο πέντε ρομπότ από τα οποία τα τρία είναι honest και θα επιβεβαιώσουν ή απορρίψουν την έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$.

- Δύο έξοδοι σε διαφορετικά τόξα

Η μέγιστη απόσταση που μπορεί να έχουν είναι τρία τόξα. Αν είχαν παραπάνω τότε από το ένα θα είχαν περάσει $f - 6$ crash από το ένα και πέντε no-crash. Από τα επόμενα δύο το ένα είναι honest. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{3\pi}{n})$.

- Τρεις έξοδοι σε ίδιο τόξο

Τρεις έξοδοι σε ίδιο τόξο. Τότε θα έχουμε από αυτό το τόξο $f - 4$ crash και τα δύο Byzantines και ένα honest. Από τα επόμενα τρία τουλάχιστον ένα είναι honest και θα επιβεβαιώσει την σωστή έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{3\pi}{n})$.

- Τρεις έξοδοι, οι δύο στο ίδιο τόξο

Από τα δύο τόξα έχουν περάσει τουλάχιστον $f - 4$ crash. Ακόμα η μία έξοδος έχει βρεθεί από honest .Η απόσταση των δύο τόξων δεν μπορεί να ξεπερνα τους τρεις γύρους. Όμοια με προηγούμενη περίπτωση. Επομένως από τα επόμενα τέσσερα τουλάχιστον τα δύο θα είναι honest και θα επιβεβαιώσουν την σωστή έξοδο. Ο χρόνος που απαιτείται για να βρεθεί η έξοδος σε αυτή την περίπτωση είναι $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$.

- Τρεις έξοδο σε διαφορετικά τόξα

Λύνεται με βάση το Θεώρημα 2.

3. Κάτω φράγμα για την περίπτωση των δύο Byzantine

3.1 Περιγραφή προηγούμενης επίλυσης

Έχει βρεθεί ότι ισχύει: $S_c(n,f) \geq 1 + \frac{(f+1)2\pi}{n}$. Αρχικά εφόσον η ταχύτητα των ρομπότ ισούται με ένα χρειάζεται χρόνος ίσος με ένα για να φτάσουν από το κέντρο του κύκλου στη περιφέρειά του. Ακόμα για να είμαστε σίγουροι ότι θα περάσει honest από το σημείο που βρίσκεται η έξοδος θα πρέπει να περάσουν $f + 1$ ρομπότ από κάθε σημείο. Θεωρούμε ότι με l_i συμβολίζουμε τα μήκη περίμετρος από τα οποία έχουν περάσει ακριβώς i ρομπότ. Το i παίρνει τιμές μεταξύ του μηδενός μέχρι το n. Είναι εύκολα κατανοητό από τα παραπάνω ότι $l_0 = l_1 = \ldots = l_f$ και $l_{f+1} + l_{f+2} + \ldots + l_n = 2\pi$. Το άθροισμα των μερών της περιμέτρου που έχει εξερευνηθεί από ρομπότ ισούται με $(f + 1)l_{f+1} + (f + 2)l_{f+2} + \ldots + nl_n$. Αν θεωρήσουμε ότι τα ρομπότ πετυχαίνουν το παραπάνω σε χρόνο t τότε ισχύει ότι:

$$\begin{aligned} nt &\geq (f + 1)l_{f+1} + (f + 2)l_{f+2} + \ldots + nl_n \\ &\geq (f + 1)\left(l_{f+1} + l_{f+2} + \ldots + l_n\right) \\ &\geq (f + 1)2\pi \end{aligned}$$

Σαν αποτέλεσμα θα έχουμε ότι $t \geq \frac{(f+1)2\pi}{n}$. Από αυτό προκύπτει ότι $1 + \frac{(f+1)2\pi}{n}$ [12].

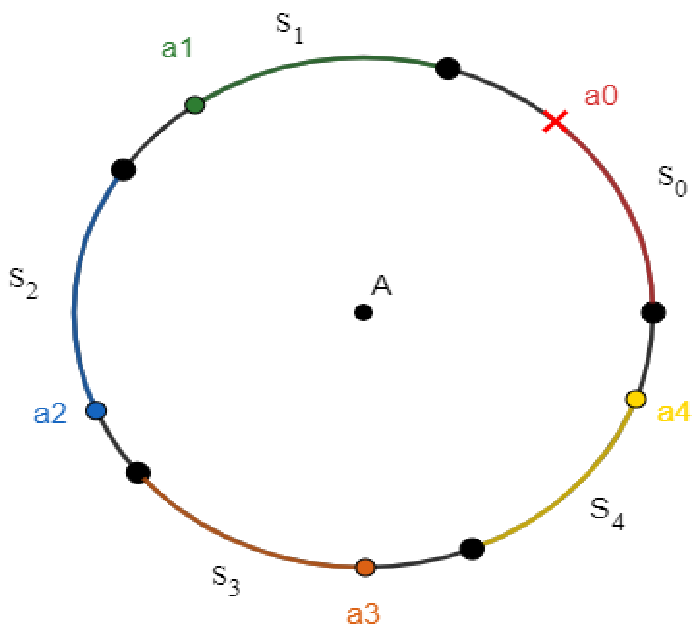
3.2 Νέο κάτω φράγμα

3.2.1 Αντιπαραδείγματα

Παρατηρούμε ότι το κάτω φράγμα που έχει βρεθεί ισούται με $S_c(n,f) \geq 1 + \frac{(f+1)2\pi}{n}$. Αυτό για την περίπτωση που έχουμε δύο Byzantine γίνεται $S_c(n,f) \geq 1 + \frac{6\pi}{n}$. Το άνω φράγμα που βρήκαμε ισούται με $S_c(n,f) \leq 1 + \frac{8\pi}{n}$. Αυτό σημαίνει ότι υπάρχει περίπτωση να υπάρχει πιο γρήγορος αλγόριθμος που να λύνει το πρόβλημα σε τρεις γύρους από ότι σε τέσσερις που το λύνουμε εμείς. Θα περιγράψουμε διάφορα παραδείγματα που δείχνουν την δυσκολία να γίνεται να λυθεί το πρόβλημα πιο γρήγορα από τον αλγόριθμο που παρουσιάσαμε.

— Αντιπαράδειγμα 1

Αρχικά θεωρούμε ότι το πρόβλημα λύνεται σε τρεις γύρους. Είμαστε στην περίπτωση (5,2,2). Θεωρούμε ότι ένα ρομπότ ανακοινώνει έξοδο στην πρώτη φάση. Έστω ότι το α_0 βρίσκει έξοδο. Θα βρισκόμαστε σε τέτοια φάση:



Σχήμα 27: Το α_0 ανακοινώνει έξοδο στην πρώτη φάση.

Για να είμαστε σίγουροι για την έξοδο του α_0 θα πρέπει να ελεγχθεί από ακόμα τέσσερα ρομπότ. Ο μόνος τρόπος να γίνει αυτό μέσα σε σύνολο τριών γύρων είναι τα α_4 και α_3 να συνεχίσουν κανονικά, ενώ τα α_1 και α_2 να κάνουν αντίστροφη κίνηση. Στο παρακάτω πίνακα βλέπουμε τα ρομπότ που θα έχουν περάσει από κάθε τόξο.

Πίνακας 9

Κινήσεις των ρομπότ για το αντιπαράδειγμα ένα.

	Γύρος 1	Γύρος 2	Γύρος 3
S_0	α_0	α_4	α_3 α_1
S_1	α_1	--	α_4
S_2	α_2	--	--

S_3	α_3	--	--
S_4	α_4	α_3	--
έξοδος του α_0	α_0	α_4 α_1	α_3 α_2

Παρατηρούμε ότι από την έξοδο του α_0 περνάνε πέντε ρομπότ και έτσι με σιγουριά ξέρουμε αν είναι σωστή. Θεωρούμε ότι το α_0 και το α_2 είναι Byzantine και ότι η έξοδος βρίσκεται στο τόξο S_2. Το α_0 ειδοποιεί για έξοδο και απορρίπτεται μετά από τρεις γύρους. Όμως από το τόξο S_2 δεν έχει περάσει άλλο ρομπότ πέρα του α_2 και έτσι σε τρεις γύρους δεν έχουμε βρει την έξοδο.

— Αντιπαράδειγμα 2

Θεωρούμε ότι αφήνουμε να τρέξουν και οι τρεις φάσεις και στη συνέχεια να δούμε αν μπορούμε να βρούμε με σιγουριά την αληθινή έξοδο. Παρακάτω παραθέτουμε τον πίνακα με τις κινήσεις των ρομπότ σε αυτούς τους τρεις γύρους.

Πίνακας 10

Κινήσεις των ρομπότ για το αντιπαράδειγμα δύο.

	Γύρος 1	Γύρος 2	Γύρος 3
S_0	α_0	α_4	α_3
S_1	α_1	α_0	α_4
S_2	α_2	α_1	α_0
S_3	α_3	α_2	α_1
S_4	α_4	α_3	α_2

Παίρνουμε την εξής περίπτωση. Έστω ότι το α_0 είναι honest και έχει ανακοινώσει έξοδο στο τόξο S_0. Το α_4 και το α_2 είναι Byzantine. Αν το α_4 και το α_2 ανακοινώσουν ίδια έξοδο στο S_4, θα έχουμε ότι αυτή η έξοδος θα έχει δύο επιβεβαιώσεις (από το α_4 και α_2) και δύο απορρίψεις από το α_0 και α_3. Η έξοδος του α_0 θα έχει δύο επιβεβαιώσεις από το α_0 και α_3 και δύο απορρίψεις από το α_4 και α_2. Παρατηρούμε ότι δεν μπορούμε να ξεχωρίσουμε ποια από τις δύο εξόδους είναι σωστή χωρίς να στείλουμε κάποιο άλλο ρομπότ να ελέγξει σε επόμενο γύρο.

3.2.2 Αποτελέσματα Παραδειγμάτων

Από τα παραπάνω βλέπουμε ότι λογικά το πρόβλημα δεν μπορεί να λυθεί σε τρεις γύρους. Σε τρεις γύρους προλαβαίνουμε είτε από κάθε σημείο να περάσουν τρία ρομπότ είτε να περάσουν από κάποια παραπάνω από τρία και από κάποια άλλα λιγότερα από τρία. Και στις δύο περιπτώσεις είδαμε παραδείγματα που μας δείχνουν ότι το πρόβλημα δεν μπορεί να λυθεί σε τρεις γύρους και ότι χρειαζόμαστε τον επιπλέον γύρο. Επομένως φαίνεται ότι ο αλγόριθμος που έχουμε παρουσιάσει πετυχαίνει τον μικρότερο δυνατό χρόνο.

4. Επέκταση στη Γενική Περίπτωση

4.1 Περιγραφή προβλήματος

Εφόσον έχει λυθεί το πρόβλημα με την ύπαρξη ένα ή δύο Byzantine, το επόμενο βήμα είναι να βρεθεί ένας γενικευμένος αλγόριθμος που θα λύνει το πρόβλημα για f αριθμό Byzantine. Ο νέος συμβολισμός είναι ο (n, f, f). Όπου n είναι ο αριθμός όλων των ρομπότ, f είναι ο αριθμός των faulty και b είναι ο αριθμός των Byzantine. Όλα τα δεδομένα που είχαμε για την περίπτωση με τα δύο Byzantine παραμένουν και σε αυτή την περίπτωση.

4.1 Περιγραφή επίλυσης

Είναι εύκολα αντιληπτό ότι υπάρχει μεγαλύτερη δυσκολία σε σχέση με την προηγούμενη περίπτωση. Συγκεκριμένα με το να αυξάνεται ο αριθμός των Byzantine, αυξάνονται και οι συνδυασμοί διαφορετικών ανακοινώσεων εξόδων με ή χωρίς απορρίψεις/εξόδων καθώς και το που θα βρεθούν αυτές. Η ανάλυση σε κάθε φάση φαίνεται να μην βοηθάει στην εύρεση της γενικής λύσης. Για αυτό το λόγο βρισκόμαστε στην επεξεργασία μίας επίλυσης που διαφέρει από την προηγούμενή μας. Αρχικά στέλνουμε όλα τα ρομπότ από την αρχή του κύκλου στην περιφέρεια του κύκλου. Στη συνέχεια τρέχουμε τον αλγόριθμο για $f + 1$ γύρους. Από το παραπάνω συνεπάγεται ότι από κάθε σημείο του κύκλου θα έχουν περάσει $f + 1$ ρομπότ και αφού διαθέτουμε f Byzantine, τουλάχιστον ένα honest θα έχει περάσει από κάθε σημείο και έτσι η έξοδος θα έχει βρεθεί. Στη συνέχεια με διάφορες κινήσεις θέλουμε με σιγουριά να αποδείξουμε το ποια έξοδο από όσες έχουν ανακοινωθεί είναι η αληθινή. Λύνοντας για αρκετές περιπτώσεις έχουμε καταλήξει σε δύο χρόνους. Στον $1 + 2 + \frac{(f+1)2\pi}{n}$ και στον $1 + 2 + \frac{(f+2)2\pi}{n}$. Στο τωρινό σημείο υλοποιούμε τον κανονικό αλγόριθμο που θα ορίζει με ακρίβεια ποια ρομπότ ελέγχουν ποιες εξόδους.

4.2 Άνω φράγμα

Σαν μέγιστος χρόνος έχει βρεθεί το εξής:

- $S_c(n, f) \leq 1 + \frac{2f2\pi}{n}$, για $f \leq 6$
- $S_c(n, f) \leq 1 + 2 + \frac{(f+2)2\pi}{n}$, για $f \leq 16$

- $S_c(n, f) \leq 1 + 2 + \frac{(f+3)2\pi}{n}$

Για $f \leq 3$ το πρόβλημα λύνεται παρόμοια με αυτό που έχει δύο Byzantine. Δηλαδή λύνεται με το να αναλύουμε την κάθε φάση και το να στέλνουμε ρομπότ πίσω για να ελέγξουν την/τις έξοδο/εξόδους. Επιλέγουμε αυτή την επίλυση γιατί ο χρόνος είναι μικρότερος από αυτόν της δεύτερης περίπτωσης. Για $f \geq 3$ γίνεται όμως μεγαλύτερος και για αυτό, επιλέγουμε να επιλυθεί με διαφορετικό τρόπο το πρόβλημα. Για τους άλλους δύο τύπους θεωρούμε το εξής. Θέλουμε να βρούμε το χρόνο που λύνει το χειρότερης περίπτωσης σενάριο. Αρχικά θα βρούμε ποιο είναι αυτό το σενάριο.

4.2.1 Σενάριο χειρότερης περίπτωσης - Worst case scenario

Για να βρούμε το χειρότερο σενάριο παίρνουμε τα εξής:

- Όσο λιγότεροι έξοδοι έχουν ανακοινωθεί τόσο περισσότερα Byzantine απομένουν. Επομένως πρέπει να στείλουμε περισσότερα σε κάθε έξοδο για να επιβεβαιωθεί/απορριφθεί με σιγουριά.

- Όσο περισσότερα κοινά ρομπότ έχουν περάσει από τα Sectors που ανακοινώθηκαν οι έξοδοι, τόσο πιο δύσκολο καθώς δεν υπάρχουν αυτόματες απορρίψεις και παραμένουν περισσότερα Byzantines. Αν κάποιο μη κοινό ρομπότ ανακοινώσει μία έξοδο τότε αυτόματα απορρίπτει όλες τις άλλες και επομένως οι υπόλοιπες έξοδοι απορρίπτονται από μεγαλύτερο αριθμό ρομπότ και όχι μόνο από αυτά που έχουν περάσει από αυτές..

- Όσο πιο ομοιόμορφα κατανεμημένες οι επιβεβαιώσεις τόσο πιο δύσκολο καθώς σε άλλη περίπτωση απορρίπτουμε/επιβεβαιώνουμε από αυτό με τις λιγότερες απορρίψεις προς τις περισσότερες και επομένως κάθε φορά μειώνεται ο αριθμός των διαθέσιμων Byzantine.

Από όλα τα παραπάνω προκύπτει ότι η πιο δύσκολη περίπτωση είναι αυτή που έχουν ανακοινωθεί τρεις έξοδοι σε τρία συνεχόμενα τόξα, όπου η κάθε έξοδος έχει επιβεβαιωθεί από τον ίδιο αριθμό κοινό ρομπότ. Με πρώτη ματιά είναι αντιληπτό ότι η πιο δύσκολη περίπτωση είναι αυτή των δύο εξόδων σε δύο συνεχόμενα τόξα, όπου η κάθε έξοδος έχει επιβεβαιωθεί από τον ίδιο αριθμό κοινό ρομπότ. Αυτό δεν αληθεύει στην πραγματικότητα καθώς μπορούμε να στείλουμε όσα ρομπότ δεν έχουν περάσει από το ένα Sector σε αυτό και είτε θα το επιβεβαιώσουν είτε θα το απορρίψουν.

4.2.2 Άνω φράγμα για το σενάριο χειρότερης περίπτωσης

Τα δεδομένα που έχουμε για το πρόβλημα μας όταν έχουμε τρέξει τον αλγόριθμό μας για $f + 1$ για τρεις συνεχόμενους γύρους είναι:

- Τρία συνεχόμενα Sectors που έχουν ανακοινωθεί μία έξοδος στο καθένα.

- $f + 3$ ρομπότ έχουν περάσει συνολικά και από τα τρία Sectors. Το κάθε Sector έχει ελεγχθεί από $f + 1$ ρομπότ. Το πρώτο Sector με το δεύτερο έχουν διαφορετικό ένα ρομπότ και επομένως από αυτά τα δύο έχουν περάσει στο σύνολο $f + 2$ ρομπότ. Το δεύτερο με το τρίτο έχουν διαφορετικό ένα ρομπότ, ενώ το πρώτο με το τρίτο έχουν διαφορετικά δύο ρομπότ. Επομένως και από τα τρία ρομπότ έχουν περάσει $f + 3$ ρομπότ.

- Απομένουν $2f + 1 - (f + 3) = 2f + 1 - f - 3 = f - 2$ ρομπότ που δεν έχουν περάσει από κανένα από τα τρία Sectors που έχει βρεθεί έξοδος. Παίρνουμε σαν συνολικό αριθμό των ρομπότ, τον ελάχιστο δυνατό αριθμό που πρέπει να έχουμε για να μπορεί να λυθεί το πρόβλημα. Όσο αυξάνεται ο συνολικός αριθμός, τόσο περισσότερα ρομπότ απομένουν και έτσι μπορούμε να στείλουμε περισσότερα σε κάθε Sector και το πρόβλημα λύνεται πιο εύκολα.

- $f + 1$ ρομπότ έχουν περάσει από κάθε Sector.

- $f + 1 - 2 = f - 1$ κοινά ρομπότ. Αφού το πρώτο με το τρίτο έχουν δύο διαφορετικά ρομπότ από όσα ρομπότ έχουν περάσει αρκεί να αφαιρέσουμε τα δύο κοινά.

- $\frac{f-1}{3}$ ρομπότ επιβεβαιώνουν την κάθε έξοδο. Εφόσον θέλουμε ομοιόμορφη κατανομή στις απορρίψεις διαιρούμε τα κοινά ρομπότ με τον αριθμό των Sector.

- Byzantine που έχουν περάσει από τα τρία Sectors $f + 1 - \frac{f-1}{3} = \frac{3f+3-f+1}{3} = \frac{2f+4}{3}$. Αν θεωρήσουμε ότι η σωστή έξοδος έχει ανακοινωθεί στο πρώτο Sector τότε ο αριθμός των Byzantine που σίγουρα έχουν περάσει είναι ο αριθμός που απέρριψαν την έξοδο του. Το ίδιο ισχύει και για τα άλλα δύο τόξα. Εφόσον έχουν επιβεβαιωθεί από τον ίδιο αριθμό ρομπότ, συνεπάγεται ότι έχουν απορριφθεί και από τον ίδιο αριθμό.

- Byzantine που απομένουν $f - \frac{2f+4}{3} = \frac{3f-2f-4}{3} = \frac{f-4}{3}$. Αφαιρούμε από τον συνολικό αριθμό των Byzantine που έχουμε στο πρόβλημά μας τον αριθμό που υπολογίσαμε παραπάνω.

Εφόσον απομένουν $\frac{f-4}{3}$ Byzantine και εμείς έχουμε $f - 2$ ρομπότ που δεν έχουν περάσει από τα τρία τόξα, πρέπει να βρούμε αν έχουμε αρκετά ρομπότ για να επιβεβαιώσουμε μία ή περισσότερες εξόδους. Υπενθυμίζουμε ότι για να είμαστε σίγουροι θα πρέπει τα honest να είναι σε πλειοψηφία. Έχουμε $\frac{f-4}{3}$ Byzantine και επομένως χρειαζόμαστε $\frac{f-4}{3} + 1$ honest. Σαν αποτέλεσμα είναι αναγκαίο να στείλουμε $2\frac{f-4}{3} + 1$. Παίρνουμε $f - 2 = k[2\frac{f-4}{3} + 1] \Rightarrow$

$$\kappa = \frac{f-2}{2\frac{f-4}{3}+1} = \frac{f-2}{\frac{2f-8+3}{3}} = \frac{3f-6}{2f-5} = \frac{2f-5+f-1}{2f-5} = 1 + \frac{f-1}{2f-5}.$$

Επομένως μπορούμε να στείλουμε στο ένα Sector τον απαραίτητο αριθμό από ρομπότ. Σε αυτό στέλνουμε $2\frac{f-4}{3} + 1 - 2$, όπου αφαιρούμε δύο γιατί θα στείλουμε τα δύο από τα $f + 3$ που δεν έχουν περάσει από αυτό το Sector. Επομένως έχουμε επιπλέον τέσσερα από αυτά που έχουν απομείνει. Δύο από αυτά που τελικά δεν στείλαμε στο ένα Sector και επιπλέον δύο από αυτά που έχουν περάσει μόνο από δύο από τα τρία Sectors. Επομένως έχουμε από ρομπότ $\frac{f-1}{2f-5}[2\frac{f-4}{3} + 1] + 4 = \frac{f-1}{2f-5}(\frac{2f-8+3}{3}) + 4 = \frac{(f-1)(2f-5)+3\cdot 4(2f-5)}{3(2f-5)}$ $= \frac{2f^2-5f-2f+5+24f-60}{3(2f-5)} = \frac{2f^2+17f-55}{3(2f-5)} (1)$.

Βρίσκουμε τη διακρίνουσα $\Delta = 17^2 - 4\cdot 2\cdot(-55) = 729 = 27^2$.

Επομένως παίρνουμε:

$$\chi_{1,2} = \frac{-17\pm 27}{2\cdot 2}$$
$$\Rightarrow \chi_1 = \frac{5}{2}$$
$$\Rightarrow \chi_2 = -11$$

Συνεπώς η (1) γίνεται $= \frac{2(f-\frac{5}{2})(f+11)}{3(2f-5)} = \frac{(2f-5)(f+11)}{3(2f-5)} = \frac{f+11}{3}$.

Σε αυτό το στάδιο θέλουμε να βρούμε αν τα ρομπότ που έχουν απομείνει είναι αρκετά για να επιβεβαιώσουν/απορρίψουν ακόμα μία έξοδο. Επομένως θέλουμε:

$$\frac{f+11}{3} \geq 2\frac{f-4}{3} + 1$$
$$\Rightarrow f + 11 \geq 2f - 8 + 3$$
$$\Rightarrow f + 11 \geq 2f - 5$$
$$\Rightarrow f \leq 16$$

Συμπεραίνουμε ότι για $f \leq 16$ μπορούμε να επιβεβαιώσουμε/απορρίψουμε δύο εξόδους σε δύο τόξα και επομένως αν απορριφθούν και οι δύο, τότε η λύση είναι αυτή που βρίσκεται στο τόξο που δεν στείλαμε κάποιο ρομπότ. Ο χρόνος για αυτό είναι $S_c(n,f) \leq 1 + 2 + \frac{(f+2)2\pi}{n}$. Σε περίπτωση που έχουμε $f \geq 16$ επιβεβαιώνουμε/απορρίπτουμε την έξοδο στο ένα τόξο, όμως δεν έχουμε αρκετά ρομπότ για να επιβεβαιώσουμε/απορρίψουμε την έξοδο σε άλλο τόξο. Χρειαζόμαστε έναν επιπλέον γύρο ώστε να στείλουμε κάποια από αυτά που έχουμε ήδη στείλει στο διπλανό Sector, ώστε να συμπληρωθεί ο απαιτούμενος αριθμός των ρομπότ. Σε αυτή τη περίπτωση ο χρόνος που απαιτείται είναι $S_c(n,f) \leq 1 + 2 + \frac{(f+3)2\pi}{n}$. Το 2 στον τύπο προκύπτει από τη μέγιστη απόσταση που μπορεί να χρειαστεί κάποιο ρομπότ για να ελέγξει κάποια ανακοίνωση. Η μέγιστη αυτή απόσταση ισούται με τη διάμετρο, η οποία είναι ίση με 2.

4.3 Άνω φράγμα σε σχέση με το κάτω φράγμα

Τη δεδομένη χρονική στιγμή το κάτω φράγμα στην γενική περίπτωση (n,f,f) είναι $1 + \frac{2(f+1)\pi}{n}$. Παρατηρούμε ότι υπάρχει μία διαφορά ανάμεσα στο άνω φράγμα που έχουμε βρει και στο κάτω φράγμα που έχει αποδειχθεί. Αυτό επιτρέπει την ύπαρξη αλγορίθμου που επιλύει το γενικευμένο πρόβλημα σε λιγότερο χρόνο από ότι έχουμε υπολογίσει. Παρατηρούμε, όμως ότι το άνω φράγμα που έχουμε βρει δεν απέχει πολύ από το κάτω, το οποίο μας δείχνει ότι η λύση που στοχεύουμε είναι αρκετά καλή. Αυτό γίνεται καλύτερα αντιληπτό από τον παρακάτω πίνακα.

Πίνακας 11

Σύγκριση κάτω με άνω φράγμα για διάφορες περιπτώσεις.

Περίπτωση	Κάτω Φράγμα	Άνω Φράγμα	Διαφορά
(7,3,3)	~4,89	~6,38	~1,49
(19,9,9)	~4,31	~6,64	~2,33
(25,12,12)	~4,27	~6,52	~2,25
(35,17,17)	~4,23	~6,59	~2,36
(41,20,20)	~4,22	~6,52	~2,3

4.4 Σύγκριση των άνω ορίων

Όπως έχουμε αναφέρει το άνω φράγμα που έχουμε καταλήξει είναι το εξής:

- $S_c(n,f) \leq 1 + \frac{2f2\pi}{n}$, για $f \leq 6$

- $S_c(n, f) \le 1 + 2 + \frac{(f+2)2\pi}{n}$, για $f \le 16$

- $S_c(n, f) \le 1 + 2 + \frac{(f+3)2\pi}{n}$

Αξίζει να αναφερθεί ότι με παρόμοιο τρόπο όπως στην περίπτωση με τα δύο Byzantine μπορεί να λυθεί το και το γενικό πρόβλημα. Ο χρόνος σε αυτή την περίπτωση είναι $S_c(n, f) \le 1 + \frac{2f2\pi}{n}$. Για αυτό το λόγο θέλουμε να συγκρίνουμε αυτόν τον χρόνο με το άνω φράγμα χωρίς να λάβουμε την πρώτη περίπτωση.

Πίνακας 12

Σύγκριση των δύο άνω ορίων.

Περίπτωση	$1 + \frac{2f2\pi}{n}$	$1 + 2 + \frac{(f+2)2\pi}{n}, f \le 16$	$1 + 2 + \frac{(f+3)2\pi}{n}, f > 16$
(7,3,3)	~6,38	~7,49	--
(13,6,6)	~6,80	~6,86	--
(15,7,7)	~6,86	~6,77	--
(19,9,9)	~6,95	~6,64	--
(25,12,12)	~7,03	~6,52	--
(35,17,17)	~7,10	--	~6,59
(41,20,20)	~7,13	--	~6,52

Επομένως παρατηρούμε ότι για $f \le 6$ είναι πιο γρήγορο να λύσουμε με τρόπο παρόμοιο με αυτόν που παρουσιάστηκε στην περίπτωση $(n, 2, 2)$. Όσο μεγαλώνει το f παρατηρούμε ότι ο πρώτος χρόνος αυξάνεται ενώ ο δεύτερος μειώνεται.

5. Επίλογος

5.1 Προβλήματα για μελλοντική έρευνα

Το πρώτο πρόβλημα που βρίσκεται ήδη προς επίλυση είναι η γενίκευση του προβλήματος σε (n, f, f), που παρουσιάσαμε παραπάνω. Η μελέτη του προβλήματος έχει ήδη ξεκινήσει και ο πρώτος αρχικός αλγόριθμος έχει υλοποιηθεί και βρίσκεται σε έλεγχο αν επιλύει όλες τις περιπτώσεις. Δυστυχώς δεν προλάβαμε να ολοκληρώσουμε την υλοποίηση και για αυτό δεν βρίσκεται στην παρούσα διπλωματική. Όμως συνεχίζουμε να δουλεύουμε πάνω στην υλοποίηση. Στη συνέχεια το επόμενο βήμα είναι η γενίκευση σε μορφή (n, f, b). Δηλάδη από τα f faulty τα b θα είναι Byzantine και τα υπόλοιπα crash. Ακόμα ενδιαφέρον παρουσιάζει η εύρεση και του κάτω ορίου πέρα από το άνω φράγμα. Με την επίλυση και αυτού του προβλήματος μπορούμε να προχωρήσουμε σε παραλλαγές και περαιτέρω γενικεύσεις.. Η πρώτη παραλλαγή είναι το πρόβλημα της εκκένωσης. Σε αυτή την περίπτωση το πρόβλημα δεν τελειώνει όταν βρεθεί η έξοδος, αλλά όταν όλα τα ρομπότ περάσουν από την έξοδο. Ήδη έχουμε παρουσιάσει προβλήματα εκκένωσης, οπότε θα μπορούσαμε να λύσουμε και αυτό στη γενική του περίπτωση. Το πρόβλημα είτε το αρχικό είτε της εκκένωσης μπορεί να γίνει πιο σύνθετο αν προσθέσουμε παραπάνω εξόδους που θέλουμε να τις βρούμε όλες ή έστω τη μία. Ακόμα μπορούμε να επεκταθούμε και σε άλλους γεωμετρικούς χώρους ή ακόμα και στον κύκλο αλλά η έξοδος να βρίσκεται σε οποιοδήποτε σημείο. Μπορούμε να επεκταθούμε και σε προβλήματα που θέλουμε να αποκλείσουμε έναν χώρο, κλείνοντας όλες τις διαθέσιμες εξόδους. Αυτό είναι και επίκαιρο καθώς μπορεί να διαθέτουμε ένα χώρο που είναι μολυσμένος από COVID-19 και θέλουμε να αποκλείσουμε όλες τις «εξόδους» του ιού από αυτή την περιοχή. Από όλα τα παραπάνω γίνεται αντιληπτό ότι υπάρχει μεγάλος αριθμός προβλημάτων που μπορούν να αποτελέσουν αντικείμενο μελλοντικής έρευνας.

5.2. Συμπεράσματα

Έχουμε καταλήξει σε αλγόριθμο για την επίλυση των περιπτώσεων $(n, 2, 2)$ και $(n, f, 2)$. Οι χρόνοι για το σενάριο χειρότερης περίπτωσης είναι $1 + \frac{8\pi}{n}$ για την πρώτη περίπτωση και $1 + \frac{2\pi}{n}(f - 1) + 2sin(\frac{4\pi}{n})$ για την δεύτερη. Ο αλγόριθμος λειτουργεί με την τεχνική του deviation. Δηλαδή ρομπότ ανάλογα με το αν βρέθηκε έξοδος γυρνάνε πίσω να ελέγξουν ενώ τα υπόλοιπα συνεχίζουν κανονικά. Στην συνέχεια προσπαθήσαμε να δείξουμε ότι αυτοί οι χρόνοι αποτελούν τους καλύτερους δυνατούς. Για να το πετύχουμε αυτό περιγράψαμε δύο παραδείγματα που δείχνουν την δυσκολία στο να βρεθεί αλγόριθμος που πετυχαίνει καλύτερο χρόνο. Στο τέλος έχει αρχίσει η μελέτη του γενικευμένου προβλήματος με τωρινό χρόνο για την χειρότερη περίπτωση να είναι $1 + 2f\frac{2\pi}{n}$ για $f \leq 6$, $1 + 2 + (f + 2)\frac{2\pi}{n}$ για $f \leq 16$ και $1 + 2 + (f + 3)\frac{2\pi}{n}$ για τις υπόλοιπες περιπτώσεις. Καταλήξαμε ότι για $f \leq 6$ μπορούμε να λύσουμε το πρόβλημα με παρόμοιο τρόπο όπως για την περίπτωση των δύο Byzantine. Για $f > 6$ προσπαθούμε να λύσουμε το πρόβλημα χωρίς το deviation καθώς παρατηρήσαμε ότι έτσι πετυχαίνουμε καλύτερο χρόνο. Η σημασία της επίλυσης του γενικευμένου προβλήματος είναι ιδιαίτερη καθώς αντιλαμβανόμαστε ότι λύνονται περιπτώσεις που έχουν μεγάλους αριθμούς και δεν θα ήταν δυνατόν να λυθούν στο χέρι. Ακόμα μία τέτοια γενίκευση θα βοηθήσει στην επίλυση και άλλων παρόμοιων προβλημάτων. Η δυσκολία μίας τέτοιας επίλυσης βρίσκεται στον μεγάλο αριθμό των συνδυασμών των ανακοινώσεων καθώς και στη δυσκολία να ομαδοποιηθούν οι περιπτώσεις.

Βιβλιογραφικές Αναφορές

[1] R. Ahlswede, I. Wegener, Search problems, Wiley-Interscience, 1987.

[2] L. Stone, Theory of optimal search, Academic Press New York, 1975.

[3] A. Beck, On the linear search problem, Israel Journal of Mathematics 2 (4) (1964) 221–228. https://doi.org/10.1007/BF02759737

[4] R. Bellman, An optimal search, Siam Review 5 (3) (1963) 274–274.

[5] J. Czyzowicz, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, Search on a line with faulty robots, Distributed Computing 32 (6) (2019) 493–504.

[6] J. Czyzowicz, K. Georgiou, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, S. Shende, Search on a line by byzantine robots, in: ISAAC, 2016, pp. 27:1–27:12.

[7] M.-Y. Kao, J. H. Reif, S. R. Tate, Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem, Information and Computation 131 (1) (1996) 63–79.

[8] R. Baeza-Yates, J. Culberson, G. Rawlins, Searching in the plane, Inf. Comput. 106 (2) (1993) 234–252 ISSN 0890-5401, https://doi.org/10.1006/inco.1993.1054.

[9] J. Czyzowicz, K. Georgiou, M. Godon, E. Kranakis, D. Krizanc, W. Rytter, M. Włodarczyk, Evacuation from a disc in the presence of a faulty robot, in: International Colloquium on Structural Information and Communication Complexity, Springer, 2017, pp. 158–173.

[10] J. Czyzowicz, L. Gasieniec, T. Gorry, E. Kranakis, R. Martin, D. Pajak, Evacuating robots from an unknown exit located on the perimeter of a disc, in: DISC 2014, Springer, Austin, Texas, 2014.

[11] Czyzowicz J., Georgiou K., Kranakis E., Narayanan L., Opatrny J., Vogtenhuber B. (2015) Evacuating Robots from a Disk Using Face-to-Face Communication (Extended Abstract). In: Paschos V., Widmayer P. (eds) Algorithms and Complexity. CIAC 2015. Lecture Notes in Computer Science, vol 9079. Springer, Cham. https://doi.org/10.1007/978-3-319-18173-8_10

[12] Georgiou K., Kranakis E., Leonardos N., Pagourtzis A., Papaioannou I. (2019) Optimal Circle Search Despite the Presence of Faulty Robots. In: Dressler F., Scheideler C. (eds) Algorithms for Sensor Systems. ALGOSENSORS 2019. Lecture Notes in Computer Science, vol 11931. Springer, Cham. https://doi.org/10.1007/978-3-030-34405-4_11

[13] S. Alpern, S. Gal, The theory of search games and rendezvous, Vol. 55, Springer, 2003.

[14] Demaine, Erik & Fekete, Sándor & Gal, Shmuel. (2006). Online Searching with Turn Cost. Theoretical Computer Science. 361. 342-355. 10.1016/j.tcs.2006.05.018.

[15] Czyzowicz, J., Kranakis, E., Krizanc, D., Narayanan, L., Opatrny, J., & Shende, S. (2015). Wireless autonomous robot evacuation from equilateral triangles and squares. In S. Papavassiliou, & S. Ruehrup (Eds.), Ad-hoc, Mobile, and Wireless Networks - 14th International Conference, ADHOC-NOW 2015, Proceedings (pp. 181-194). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 9143). Springer Verlag. https://doi.org/10.1007/978-3-319-19662-6_13

[16] J. Czyzowicz, K. Georgiou, E. Kranakis, Group search and evacuation, in: P. Flocchini, G. Prencipe, N. Santoro (Eds.), Distributed Computing by Mobile Entities; Current Research in Moving and Computing, Springer, 2019, Ch. 14, pp. 335–370.

[17] Leonardos N., Pagourtzis A., Papaioannou I. (2021) Byzantine Fault Tolerant Symmetric-Persistent Circle Evacuation. In: Gąsieniec L., Klasing R., Radzik T. (eds) Algorithms for Sensor Systems. ALGOSENSORS 2021. Lecture Notes in Computer Science, vol 12961, pp.111-123.Springer, Cham. https://doi.org/10.1007/978-3-030-89240-1_8

[18] Shantanu Das, Riccardo Focardi, Flaminia L. Luccio, Euripides Markou, Marco Squarcina, Gathering of robots in a ring with mobile faults, Theoretical Computer Science, Volume 764, 2019, Pages 42-60, ISSN 0304-3975, https://doi.org/10.1016/j.tcs.2018.05.002.

[19] Das, Shantanu & Giachoudis, Nikos & Luccio, Flaminia & Markou, Euripides. (2019). Gathering of Robots in a Grid with Mobile Faults: 45th International Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Slovakia, January 27-30, 2019, Proceedings. 10.1007/978-3-030-10801-4_14.