OPTIMAL CIRCLE SEARCH DESPITE THE PRESENCE OF FAULTY ROBOTS

”Optimized Exit Pathfinding amidst Faulty Robot Navigation in Circular Environments”

A Project Report submitted by

Vanshika Kumawat

in partial fulfillment of the requirements for the award of the degree of

M.Sc.-M.Tech.

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Indian Institute of Technology Jodhpur
Department of Mathematics

May 2024

Declaration

I hereby declare that the work presented in this Project Report titled **Optical circle search despite of faulty robots**. submitted to the **Indian Institute of Technology Jodhpur** in partial fulfillment of the requirements for the award of the degree of **M.Sc.-M.Tech.**, is a bonafide record of the research work carried out under the supervision of **Dr. Subhash Bhagat**. The contents of this Project Report in full or in parts, have not been submitted to, and will not be submitted by me to, any other Institute or University in India or abroad for the award of any degree or diploma.

Vanshika Kumawat
M22MA207

Certificate

This is to certify that the Project Report titled Optimal circle search despite of faulty robots, submitted by **Vanshika Kumawat (M22MA207)** to the **Indian Institute of Technology Jodhpur** for the award of the degree of **M.Sc.-M.Tech.**, is a bonafide record of the research work done by him under my supervision. To the best of my knowledge, the contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Subhash Bhagat

Abstract

Let n robots that have f malfunctioning robots among them. Observe that since n and f represent the number of robots, they are whole numbers here. Additionally, there are more than one robot ($n > 1$). The problem is therefore designated as *(n, f)*-search on a circle; for $n > 1$ robots, f of which are defective, it is a search problem of a hidden exit on a circle of unit radius. Every robot begins at the circle's center and has a maximum speed of one to travel any place. Robots mayelectronically communicate while searching.

Since each robot has an individual identity, none of the communications sent by any robot can be tampered with. When an exit is located by a non-faulty robot (which needs to visit its location) and the other non-faulty robots are aware of the exit's precise location, the search is deemed to be over.

Two types of malfunctioning robots are being considered: Robots with crash defects: These robots might follow instructions to stop working and then stop working altogether. Robots with byzantine flaws: These robots are more cunning and can send out misleading signals at any time while the search is on. Their goal is to add another deceptive element to the search process by fooling non-faulty robots by, for example, giving false information about the position of the exit.

Further enhancing the research now consider the same cases for multiple exits, taking (n,f,k)-Search where k represents the number of exits on the unit circle. The task is to calculate the optimal time in finding the nearest hidden exit by the first non- faulty robot and inform location to rest of the robots. In this case of multiple exits taking Crash Faults here for less complexity of algorithms and in future it can be solved for byzantine faults also. Derived the optimal algorithm specially for the Two Unknown exit case and similarly generalized for the three four and so on up to n-1 exit cases with the Crash Faults.

Acknowledgments

I extend my heartfelt gratitude to **Dr.Subhash Bhagat**, my supervisor, for affording me the opportunity to delve into the realm of research. His unwavering support and encouragement throughout my thesis journey have not only facilitated a profound learning experience but also instilled in me the confidence to navigate the complexities of this field. Without his guidance, my intellectual horizons would have remained confined.
Special thanks are due to my colleagues under the guidance of **Dr.Subhash Bhagat**, whose collaboration and assistance have contributed significantly to the progress of my research. I am also indebted to my batchmates and seniors at IIT Jodhpur for the delightful moments we have shared, adding warmth and positivity to my two-year tenure on campus.
My deepest appreciation goes to my family for their constant motivation, urging me to pursue and realize my dreams. Their unwavering support has been a pillar of strength throughout this academic journey.
Lastly, I express my gratitude to the **Department of Mathematics, Indian Institute of Technology, Jodhpur**, for fostering a welcoming environment that facilitated my research. Their timely assistance and support were indispensable, making it possible for me to successfully complete my thesis.

Vanshika Kumawat

M.Sc.- M.Tech. (Mathematics -DCS)
M22MA207

Contents

1 Introduction And Background.

Since search issues are fundamental to many branches of mathematics and theoretical computer science, they are extremely important. Imagine dimensionless robots, sometimes referred to as mobile agents, scouring the perimeter of a unit radius disc for an exit that has been put at an unknown location. Some of these wirelessly interactive robots are either Byzantine or crash defective.

Byzantine-faulty robots are malevolent; they can trick their peers by sending misleading data, leading to delays that impact the system's worst-case search time. Crash-faulty robots can abruptly stop working, in which case they are unable to transmit any information.

This methodology departs from traditional search models using mobile agents, in which errors were limited to the search domain (a graph, continuous infinite line, or plane). With this new method, if one of the mobile agents is malfunctioning, it becomes harder to finish the search and imposes more stress on the entire group of mobile agents.

The computation model's and the mobile agents' capacities:

The goal is to generate search algorithms that detect the exit and have the worst-case optimal running time, as determined by how long it takes the first robot to locate the exit and persuade the other robots. The model's key components—communication, robot movement, crashes, byzantine defects, and the influence of the adversary—all play crucial roles in determining how well the algorithms are created.

Communication:

Regardless of exactly how far distant the robots are from one another, they are able to instantly and wirelessly communicate. They can communicate with one another by exchanging information on where they are, how far they've come from the beginning, and whether they've found the exit. By deciphering information from each other's transmissions, the robots are able to figure out their relative positions.

The robots don't need the GPS due to the fact that they are outfitted with pedestrian meters for measuring distance. To further ensure message validity and avoid any manipulation by other robots, each message is additionally uniquely tagged according to the robot's identity.

Robot movement:

The robotic journey begins at the center of a unit radius disc, with a consistent speed limit of one for every robot. Robots are able to recognize the exit if it is within their range when they move over the area and sense the disk's edge. They may navigate around the disk's edge and take "shortcuts" into its interior because of their flexible movement.

Fault types:

Robots communicate with each other and cooperate to follow predetermined itineraries. They also know each other's routes. This implies that depending on when and what kind of communication a robot transmits, individuals may be able to determine where it is.

A robot that has a crash defect may cease functioning at any moment, becoming motionless and incapable of sending messages. On the contrary, a Byzantine robot may alter its course and manipulate data in an attempt to trick other robots into thinking it's where the exit is. Notably, a

Byzantine robot can also imitate the actions of a robot with a devastating problem.

Adversary:
In worst-case analysis, suppose a scenario in which an adversary might impact the location of the exit as well as the malicious robot's actions. The malevolent robot's path and the messages it transmits can also be tracked by this control. The enemy wants to extend the time it takes to finish the search as much as possible. When a functioning robot arrives at the exit and any surviving agents can be confidently informed of the proper exit position, the search has been deemed successful.

2 Literature Survey

Initially proposed as an evacuation problem, the circle search model for n non-faulty robots is based on the completion time of the final robot to find the hidden exit. This approach has been thoroughly investigated in wireless and in-person communication contexts. Many studies have now been published on the subject, with a special emphasis on evacuation in many scenarios, including the face-to-face model and equilateral triangles. It is interesting to note that, in contrast, to search, evacuation measures the algorithm's completion by the amount of time it takes the final working robot to find the exit; in other words, the algorithm cannot be deemed complete until all robots have reached the exit.

3 Problem Definition and Objective

Finding the best algorithms for the (n, f)-search problem, also known as the circle search problem, is the goal when $n > 1$ and f of them are either crash faulty or byzantine faulty. This means that the optimal worst-case search completion time is the amount of time it takes for the first non-faulty robot to locate the exit, ensuring that all other non-faulty robots are aware of its location. Taking into account both scenarios for the $(n, 1)$-search, determine its lower and upper boundaries, and then expand its application by adding more defective robots.The main contribution is related to the $(n, 1)$-search issue with a robot that has a Byzantine fault.

4 Preliminaries and notation

The dimensionless robots start out at the center of a unit radius disc, and assume the existence of n robots, f of which are defective. The exit is situated at the disk's perimeter on the unit circle. The fastest speed that robots can move is 1.
Every honest agent in the algorithm travels at this maximum pace, guaranteeing that every agent is always aware of the whereabouts of other agents following the rules. Exactly f of the n mobile agents, symbolized as a_0, a_1, ..., a_{n1}, are defective. Index addition and subtraction are carried out modulo n. Indices are regarded as elements of Z_n. *(n, f)*-search looks for n robots, where f is a broken robot. Robots are able to search in both clockwise and anticlockwise directions around the unit circle (CCW). Presuming that the exit is always announced when an honest agent discovers it. Furthermore, an honest agent notifies everyone else if it discovers a mistaken announcement made by another agent.
The *(n,* 1)-search Problems with a Byzantine-faulty robot is the major emphasis. S(n) is a representation of the infimum of the time, taking into account all algorithms needed for the first robot to arrive at the exit without any issues. This ensures that every robot can accurately determine the location of the exit. Likewise, let $S_c(n, f)$ represent the optimal search completion time for the *(n, f*)-search problem involving crash-faulty robots.

Determining the lower and upper bounds for the *(n, f)* search and then generalizing it by increasing the number of malfunctioning robots yields the optimal search.

Lower Bound

Due to the fact that crash-faulty robots have the potential to cease functioning altogether, this means that in a series of n robots, of which f are crash-faulty, if one of the non-faulty robots finds the exit and transmits information to the other two robots, the faulty robots will not reach the exit or accept the information as true without doing their own due diligence. Hence, it is the amount of time needed for at least $f + 1$ robots to search the perimeter for an exit.

Lower bounds for the crash faulty robots:
The worst-case search time $S_c(n, f)$ for $n \geq f + 1$ exactly f of which are crash-faulty robots:

$$S_c(n, f) \geq 1 + (f + 1)\frac{2\pi}{n}$$

Algorithm:

- A robot requires at least one time to reach the perimeter since all n of the robots are positioned in the center of the unit circle and have a maximum speed of 1. Currently, there are n robots with crash defects, which could cause the robots to stop moving.

- Additionally, each location on the perimeter needs to be visited by a minimum of f + 1 robots. Failure to do so will allow the adversary to cause the at most f robots that visit this point to become faulty, meaning they won't communicate, causing the non-faulty robots to miss theexit.

- So, the time taken by the first non faulty robot to search for exit is greater than or equal to the sum of time taken to reach perimeter and search on each point on the perimeter till exit found.

Now note that as the Byzantine robots misguide the other robots so for cross checking for the accuracy robots have to traverse more distance on the perimeter as compared to the case of Crash faulty robots .Therefore, the worst case search time for the Byzantine faulty robots is more than the worst case search time for the Crash faulty robots means $(n) \geq S_c(n, 1)$.

Using this result, find another result that:

Lower Bound for Byzantine $(n, 1)$-Search

The worst-case search time S(n) for $n \geq 2$ robots, exactly one of which is Byzantine-faulty, satisfies

$$S(n) \geq 1 + \frac{4\pi}{n}.$$

Briefly it can be said that when there are two or more than two robots placed at the center of the circle and exactly one of them is Byzantine faulty then the lower bound for the worst case search time is greater than the sum of the time taken to reach the perimeter at its sector and the time in searching exit at two sectors out of n sectors.

Upper Bound

As in the case of Lower Bounds, now consider the cases for the upper Bound in (n, f)-Search with Crash Faulty and the Byzantine Faulty Robots:

Upper Bound for (n, f)-Search with Crash Faults. The worst-case search time $S_c(n, f)$ for $n \geq 2$ robots exactly f of which are prone to crash failures satisfies.

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

Algorithm

- Each n agent starts from the center of the unit circle and reaches as the corresponding sector. If it is assumed that i=0,1,...,n-1 agents move to point $i\beta$ of the unit circle where $\beta := \frac{2\pi}{n}$is a sector part of the perimeter.

- Now it searches for counterclockwise direction until exit is found, it is reported instantaneously. Clearly, every sector Sj of the circle would be visited by $f + 1$ robots if they all followed the protocol.

- Since there are at most f faulty robots, there must be at least one honest robot that will visit Sj and announce the correct location. As there can only be crash failures there will not be any contradicting announcements

- So the time taken by the first non faulty robot to search for exit is less than or equal to the sum of time taken to reach perimeter and search on each point on the perimeter in its sector till exit found and inform other agents.

Upper Bound for Search with one Byzantine Fault

Analyze upper bounds for search problems with a Byzantine agent. Main result is the following as it gives the worst-case search time and we need to reduce it to get the optimal result.

Upper Bound for $(n, 1)$**-Search**
The worst-case search time S(n) for $n \geq 2$ robots exactly one of which is faulty satisfies

$$S(n) \leq 1 + \frac{4\pi}{n}$$

Thus, combining the above results, conclude that the worst-case search completion time for (n, 1)-searchsatisfies.

$$S(n) = 1 + \frac{4\pi}{n}$$

Algorithm:

- For the (2, 1)-search, note that it is easy to verify S(2) = 1 + 2 since one of the two robots is defective and the other is not; as a result, the non-faulty robot must search the whole perimeter.

- The total time required to reach the perimeter—which is one—and to walk around the full perimeter—will therefore be the worst-case search completion time in this scenario. So, it is simple to grasp the S(2) example.

- Now the target is to show the upper bound for the cases (3,1)-search and (4,1)-search.And the research can be continued by raising the number of robots.

(3, 1)-search with a Byzantine-faulty robot

(3,1)-Search with a Byzantine - faulty robot means that there are three robots placed at the center of a circle and one out of them is Byzantine faulty.
So to calculate the worst-case search time there are some possibilities that first one may be faulty and transmits incorrect information to its two non-faulty peers which leads to wastage of time or

The second possibility is that the first of these robots is non faulty, it finds the exit point and informsthe rest one faulty and other non faulty robot.
Now again there are the possibilities that second robot is faulty and doesn't accepts the information of first robot and instead of this it gives the information that exit is in its sector.

- The worst-case search time for 3 robots exactly one of which is faulty satisfies.

$$S(3) \geq 1 + \frac{4\pi}{3}$$

.

- In this algorithm Initially, all the robots are placed at the center of the unit circle. As the search starts all robots move to the perimeter in such a way that they divide into equal sectors, and as known it takes time 1 to reach the perimeter, this becomes easy as they can communicate easily and every robot knows the location of other robots.

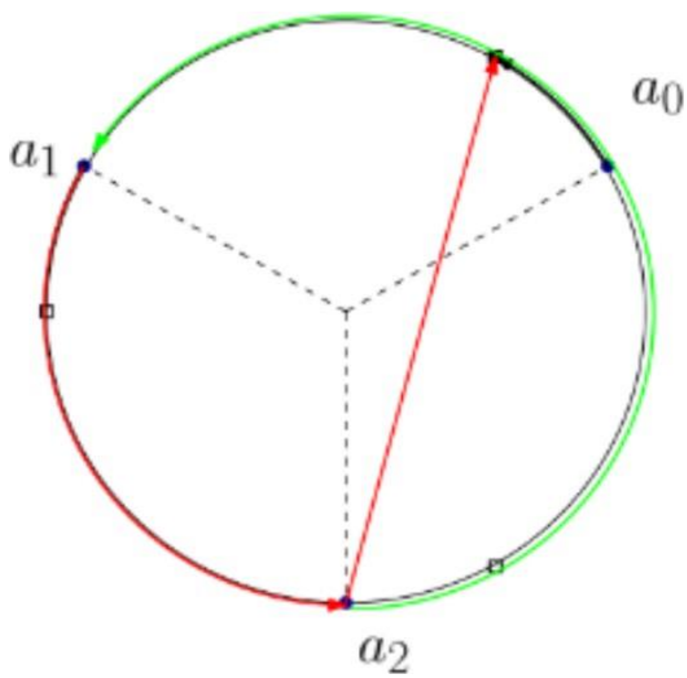


Figure 1:

- Here is the case of 3 robots in which 1 of them is Byzantine faulty. Let's rename the robots as a_0, a_1, a_2. Then first these robots start searching in the CCW direction.

- Now consider the case that a0 is non-faulty and it finds the exit and informs to rest then either a1 or a2 is byzantine faulty. So again, there arise two possible cases, if a1 is faulty then it misguides the other robots saying it found the exit in its sector. So a2 has the two locations for the exit given by a0 and a1. Now it is the task of a2 robots to find the true location of exit by identifying the faulty robots. So a2 goes searching in the green region in the diagram and finds a0 true. So, it is concluded that the true location is given by a0, now again this message transmits, and a1 reaches to exit through the location given by a0 following the red path.

- So, the time taken in traversing the perimeter becomes $\frac{4\pi}{3}$ and 1 to reach the perimeter. So proved.

(4, 1)-search with a Byzantine-faulty robot

The search time for 4 robots exactly one of which is faulty satisfies

$$S(4) \leq 1 + \pi$$

.

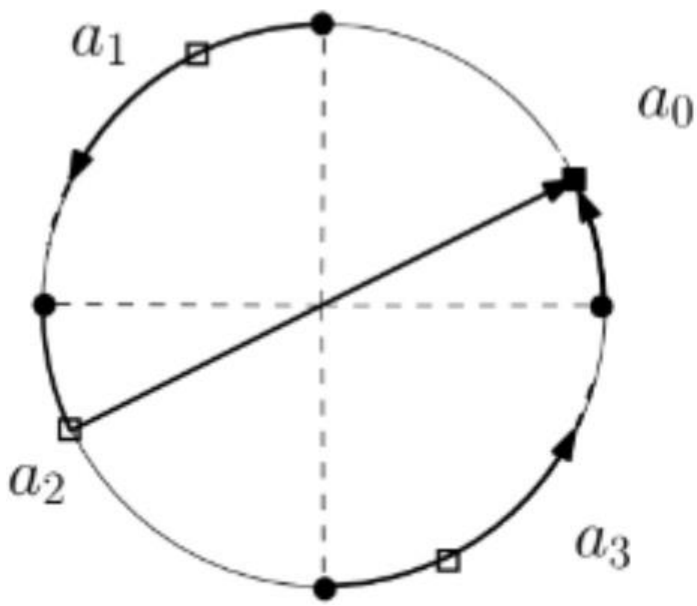


Figure 2:

- In this algorithm Initially, all the robots are placed at the center of the unit circle. As the search starts all robots move to the perimeter in such a way that they divide into equal sectors, and as known it takes time 1 to reach the perimeter, this becomes easy as they can communicate easily and every robot knows the location of other robots.

- Here is the case of 4 robots in which 1 of them is Byzantine faulty. Let's rename the robots as a0, a1, a2,and a3. Then first these robots start searching in the CCW direction.

- Now consider the case that a_0 is non-faulty and it finds the exit and informs to rest then either a_1 ,a_2 or a_3 is byzantine faulty. So again there arise three possible cases, any one of a_1 ,a_2 or a_3 is faulty then it misguides the other robots saying it found the exit in its sector. Suppose a1 is faulty then it makes an announcement that it finds the exit , So there will be two locations of exit for a_2 anda_3.Then they move and check the true location, a_3 finds the true location by a0 then informs rest so they reach location.

- So, the time taken in traversing the perimeter becomes $\frac{4\pi}{4}$ = π and 1 to reach the perimeter.
  So proved.

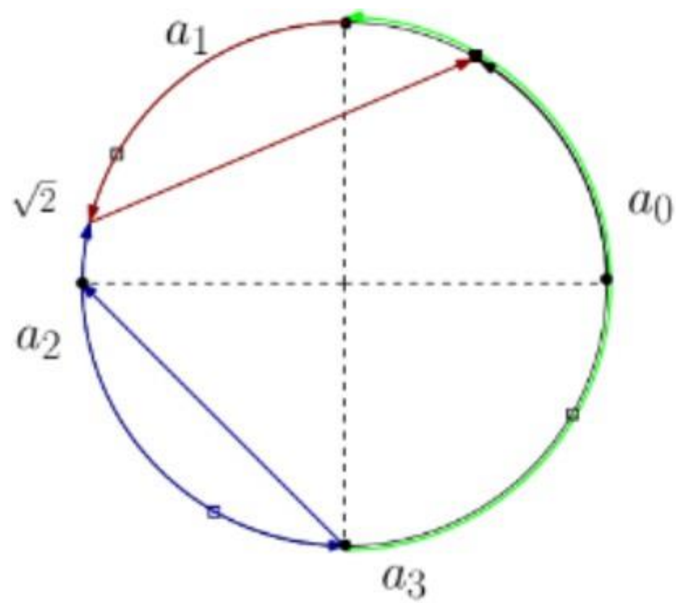


Figure 3:

Consider another Algorithm for (4,1)-Search that may be more Optimal, in which the announcement made by a_0 is correct but the faulty robot a_1 is also giving the exit in its region of search. So for shorted distance traversing a_2 goes to a1 leaving a_0 giving more distance traversing after it knows that it is the incorrect location so it moves back again to the exit shown by a_0 as shown in blue line, after a confirmation made by a_3 in green line.

So the time taken in traversing the perimeter

becomes $\frac{4\pi}{4}$= π and 1 to reach the perimeter. So proved.

Determining the lower and upper bounds for the (n, f, k) search and then generalizing it by increasing the number of Crash faults robots yields the optimal search and number of exits.

Lower and Upper Bounds for (n,f,2)-search

Due to the fact that crash-faulty robots have the potential to cease functioning altogether, this means that in a series of n robots, of which f are crash-faulty and k unknown exits , if one of the non-faulty robots finds the nearest exit and transmits information to the other robots, the faulty robots will not reach the exit or accept the information as true without doing their own due diligence. Hence, it is the amount of time needed for at least f robots to search the perimeter for an exit.

The worst-case search time $S_c(n, f, k)$ for $n \geq f + 1$ exactly f of which are crash-faulty robots (taking k=2 unknown exits) have lower and upper bounds respectively:

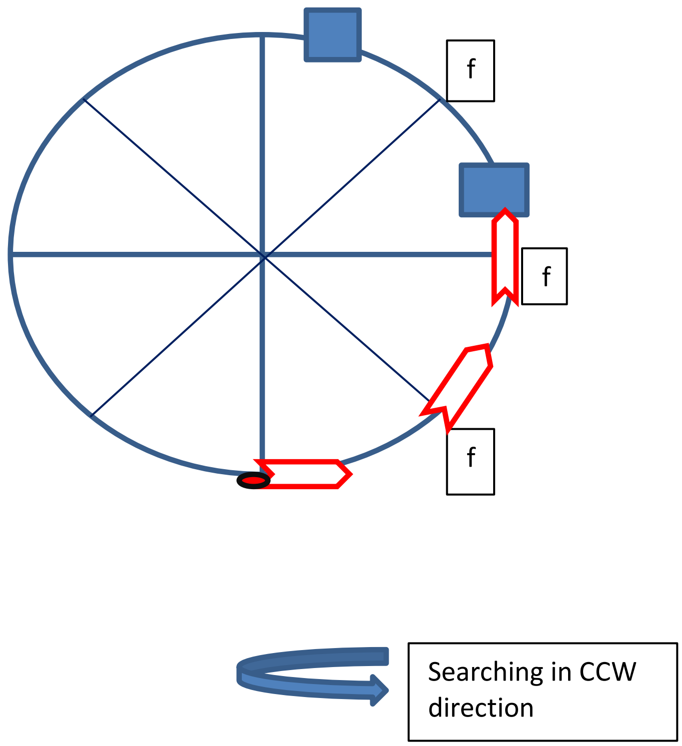
$$S_c(n, f, 2) \geq 1 + \frac{2\pi}{n}$$
$$\textit{and}$$
$$S_c(n, f, 2) \leq 1 + \frac{2\pi f}{n}$$

Algorithm:

- A robot requires at least one time to reach the perimeter since all n of the robots are positioned in the center of the unit circle and have a maximum speed of 1. Currently, there are n robots with crash defects, which could cause the robots to stop moving.
- Additionally, each location on the perimeter needs to be visited by a minimum of f + 1 robots. Failure to do so will allow the adversary to cause the at most f robots that visit this point to become faulty, meaning they won't communicate, causing the non-faulty robots to miss the exit.
- So the time taken by the first non faulty robot to search for exit is greater than or equal to the sum of time taken to reach perimeter and search on each point on the perimeter till exit found.
- Each n agent starts from the center of the unit circle and reaches as the corresponding sector. If it is assumed that i=0,1,...,n-1 agents move to point $i\beta$ of the unit circle where $\beta := 2\pi/n$ is a sector part of the perimeter.
- Now it searches for counterclockwise direction until exit is found, it is reported instantaneously. Clearly, every sector Sj of the circle would be visited by $f + 1$ robots if they all followed the protocol.
- Since there are at most f faulty robots, there must be at least one honest robot that will visit Sj and announce the correct location. As there can only be crash failures there will not be any contradicting announcements
- So the time taken by the first non faulty robot to search for exit is less than or equal to the sum

of time taken to reach perimeter and search on each point on the perimeter in its sector till exit found and inform other agents.

•



•

- Now for the best search case means getting the lower bound for the algorithm, the easiest path is that any one of the exits has no faulty robot nearest to it and it finds the exit easily on moving in CCW with affecting with the number of faulty robots or the number of exits. So in this case it doesn't matters whether the other exit is closed with the faulty robots or not because the search is complete in finding the first nearest exit by any of the honest agent. Therefore, it takes at most *2π/n time for the robot to search in its sector. Thus, the lower bound satisfies independent of f and k for all cases of Crash Faults:*

$$T = S_c(n, f, 2) \geq 1 + \frac{2\pi}{n}$$

- *On the other hand, for the worst case search this time adversary will make both the complexities in the positions of unknown exits and the positions of faults. The worst possible case out of those is that both exits are close to each other separating with a single faulty robot between them and rest of the crash faulty robots are consecutively arranged behind the other exit and at last the series of non-faulty robots comes. In this case the first honest agent must traverse through distance equals to f sectors and finds the first exit, then informs the location to another exit. Thus the upper bound or the worst-case search time for the algorithm with 2 unknown exits satisfies*

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

Here note that when the first nearest exit is found by the non-faulty agent and informs the location to others, the search completes. Therefore, it doesn't needs to check for the next exit in this algorithm. But if for the extension of research if the algorithm is made for the evacuation problem then it measures the time upto the last finder of exit means there may be the possibility that the next exit may be nearest to other non-faulty robots after making some changes in the direction.So as a future problem these cases can be considered for further research.

Crash faulty Robots with two unknown exits :

In (n,f,2) -Search the worst-case search time is

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

Where n>f+1 robots Considering all the possible cases in this search

1).Non Faulty Robots(f=0):

When no faulty robots are present in the search it means all are non-faulty agents and moves in CCW direction in their sector to find nearest exit in that sector, then any of those non faulty agent finds the exit first and it informs the location to others. Therefore, it only takes time to reach the perimeter and then search in one sector. So, the optimal search time is T <1+ $\frac{2\pi}{n}$.

2). One Faulty Robots(f=1):

When one faulty robot present in the search it means all are honest agents except one crash fault which becomes motionless and rest non faulty agents moves in CCW direction in their sector to find nearest exit in that sector, then any of those honest agent finds the exit first and it informs the location to others. Here arises the two possibilities:

(a)Both the unknown exits lie in the fault free sector and the faulty robot is at some other position except in these sectors, then the honest agent which is nearest to the exit first finds the exit succeeds and informs the location to others. It is independent of the faults, same as done in case of non-faulty robots.

(b) the faulty robot is just before one of the exits. Since there is only one faulty robot, it means the sector of the other exit is free of faults. Therefore, it only takes time to reach the perimeter and then search in this fault free sector independent of the exit with fault.

So, the optimal search time is T =1+ $\frac{2\pi}{n}$

3). Two Faulty Robots(f=2):

When two faulty robots present in the search it means all are honest agents except two crash fault which becomes motionless and rest non faulty agents moves in CCW direction in there sector to find nearest exit in that sector, then any of those honest agent finds the exit first and it informs the location to others.Here arises the three possibilities :

(a)Both the unknown exits lie in the fault free sector and the faulty robots are at some other position except in these sectors, then the honest agent which is nearest to the exit first finds the exit succeeds and informs the location to others. It is independent of faults same as done in case of non-faulty robots.

(b) One of the faulty robot is just before one of the exit and the other faulty robot is at some other place except before another exit, it means the sector of the other exit is free of faults. Therefore, it only takes time to reach the perimeter and then search in this fault free sector independent of the exit with fault. So, the optimal search time is T =1+ $\frac{2\pi}{n}$.

(c) When there is faults present before both the exits then the honest agent is placed in the next sector from that, so the honest agent must traverse its sector along with the sector of faulty robot to reach the nearest exit. Therefore, it must traverse approximately the distance of both fault sectors. So, the optimal search time is T =1+$\frac{4\pi}{n}$.

4). Three Faulty Robots(f=3):

When three faulty robots present in the search it means all are honest agents except three crash fault which becomes motionless and rest non faulty agents moves in CCW direction in there sector to find nearest exit in that sector, then any of those honest agent finds the exit first and it informs the location to others.Here arises the three possibilities :

(a)Both the unknown exits lie in the fault free sector and the faulty robots are at some other position except in these sectors, then the honest agent which is nearest to the exit first finds the exit succeeds and informs the location to others. It is independent of faults, same as done in case of non-faulty robots.
(b) One of the faulty robots is just before one of the exits and the other two faulty robots at some other place except before another exit, it means the sector of the other exit is free of faults. Therefore, it only takes time to reach the perimeter and then search in this fault free sector independent of the exit with fault.
So, the optimal search time is T =1+ $\frac{2\pi}{n}$.
(c) When both exits are at distant position separated by two faulty robot on the sectors between them and rest one faults is present before the next exit when moving in CCW ,then the honest agent are placed in the next sector from that so the honest agent has to traverse only its sector along with the sector of one faulty robot to reach the nearest exit, thus it remains unaffected of the rest two faults after the exit .Therefore, it has to traverse approximately the distance of 2 fault sectors. So, the optimal search time is T =1+ $\frac{4\pi}{n}$.
(d) When both exits are at consecutive position separated by a faulty robot on that sector and rest two faults are consecutively present after that before the next exit ,then the honest agent are placed in the next sector from that so the honest agent has to traverse its sector along with the sector of two faulty robot to reach the nearest exit. Therefore, it must traverse approximately the distance of 3 fault sectors. So, the optimal search time is T =1+ $\frac{6\pi}{n}$.

Similarly, the cases can be checked further by increasing the number of faults. Thus, we get the optimal search time for the two-exit case in the group of n number of robots out of them f is Crash Faults satisfies the algorithm:

$$\boldsymbol{T \leq 1 + f\frac{2\pi}{n}}$$

Note that similarly, it can be verified for the algorithms for increased number of exits for the Crash Faults. But it is difficult to find for the Byzantine Faults as they misguide the robots, so it may lead to an increase in the time taken to reach the exit.

5 Results

Considering the Optimal algorithms for the problem (n, 1)-Search when n 2.. Methods for (n, 1)-search on a circle admits a worst-case optimal solution with search completion time $1 + \frac{4\pi}{n}$. T hen provide a lower bound for robots displaying crash defects and, thus for Byzantine robots.

These results demonstrate the optimal solutions and time bounds for the (n, 1)-search problem on a circle, and provide insights into handling faulty robots, including Byzantine robots.

The worst-case search time $S_c(n, f, k)$ for $n \geq f + 1$ exactly f of which are crash-faulty robots (taking k=2 unknown exits) satisfies T= $S_c(n, f, 2) \leq 1 + f\frac{2\pi}{n}$.

The worst-case search time $S_c(n, f, k)$ for $n \geq f + 1$ exactly f of which are crash-faulty robots (taking k=3 unknown exits) satisfies T= $S_c(n, f, 3) \leq 1 + (f-1)\frac{2\pi}{n}$.

The worst-case search time $S_c(n, f, k)$ for $n \geq f + 1$ exactly f of which are crash-faulty robots (taking k=4,5,6...,n-1 unknown exits) satisfies T= $S_c(n, f, k) \leq 1 + (f+1-k).\frac{2\pi}{n}$

6 Conclusion

The optimality for the Byzantine case is surprising, given the scarcity of tight bounds for search on a circle even in the wireless model.Extending the results to multiple Byzantine-faulty robots or to the evacuation problem are two challenging open problems in the context of circle search.

These results demonstrate the optimal solutions and time bounds for the (n, f, k)-search problem on a circle, providing insights into handling faulty robots, including Byzantine robots. By addressing the open problems, we can further advance the field of optimal circle search in the presence of faulty robots.

7 Future Problem:

Looking to the future, there are still some problems that require further research:

- Investigating the impact of varying robot speeds on search efficiency.
- Exploring strategies to handle Byzantine-faulty robots in the search process.
- Examining the scalability of the optimal search time bounds for larger robot networks.
- Cases of increasing Byzantine robots instead of being focused on only one case of robot.
- Multiple exits with byzantine robots should be examined.
- Two severe open challenges in the context of circle search are extending the results either to

numerous robots with Byzantine faults or to the evacuation problem. By addressing these future problems, we can continue to enhance the field of optimal circle search in the presence of faulty robots.

References

Ahlswede, R. and Wegener, I. (1987). *Search problems*. John Wiley & Sons, Inc.

Alpern, S. and Gal, S. (2006). *The theory of search games and rendezvous*, volume 55. Springer Science & Business Media.

Baeza-Yates, R. (1993). J. culberson and g. rawlins searching in the plane. *Journal of Information Computing*.

Bellman, R. (1963). An optimal search. *Siam Review*, 5(3):274–274.

Czyzowicz, J., Gasieniec, L., Gorry, T., Kranakis, E., Martin, R., and Pajak, D. (2014). Evacuating robots via unknown exit in a disk. In *International Symposium on Distributed Computing*, pages 122–136. Springer.

Czyzowicz, J., Georgiou, K., Godon, M., Kranakis, E., Krizanc, D., Rytter, W., and Wl-odarczyk, M. (2017). Evacuation from a disc in the presence of a faulty robot. In *International Colloquium on Structural Information and Communication Complexity*, pages 158–173. Springer.

Czyzowicz, J., Georgiou, K., and Kranakis, E. (2019). Group search and evacuation. *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, pages 335–370.

Czyzowicz, J., Georgiou, K., Kranakis, E., Krizanc, D., Narayanan, L., Opatrny, J., and Shende, S. (2016). Search on a line by byzantine robots. *arXiv preprint arXiv:1611.08209*.
Ahlswede and Wegener (1987) Alpern and Gal (2006) Baeza-Yates (1993) Bellman (1963) Czyzowicz et al. (2014) Czyzowicz et al. (2016) Czyzowicz et al. (2017) Czyzowicz et al. (2019)

PAPER NAME

Vanshika_report.pdf

AUTHOR

Vanshika Kumawat

WORD COUNT

6076 Words

CHARACTER COUNT

28453 Characters

PAGE COUNT

22 Pages

FILE SIZE

484.5KB

SUBMISSION DATE

May 15, 2024 9:41 PM GMT+5:30

REPORT DATE

May 15, 2024 9:42 PM GMT+5:30

20% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 15% Internet database
- 16% Publications database
- Crossref database
- Crossref Posted Content database
- 4% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Cited material