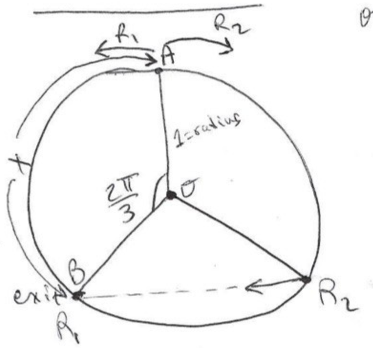
COMP 4001: Distributed Computing Robot Evacuation Project

By:
Khaled Jawhar 100978802
Damjan Markovic 100823871

Scenario 1:

Scenario 1: Both Robots start at center O and they move on opposite direction from point A.



If the angular distance between A an B equals x then the length of the chord taken by robot R_2 equals to $C(x) = 2\sin(\pi - x)$.

Worst case Scenario:

The worst case Evacuation time will become as follows:

$$T \leq \max_{0 < x \leq \pi} \{1 + x + 2\sin(\pi - x)\}$$

$$= \max_{0 < x \leq \pi} \{1 + x + 2\sin x\}$$

Average Case Scenario:

So Evacuation time needed is,

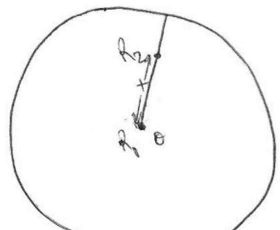
$$1 + x + 2\sin(\pi - x)$$

So the function $f(x) = 1 + x + 2\sin x$ in the interval $[0, \pi]$ is maximized at the point $x^* = \frac{2\pi}{3}$ and $f(x^*) = 1 + 2\pi/3 + \sqrt{3}$.

Scenario 2:

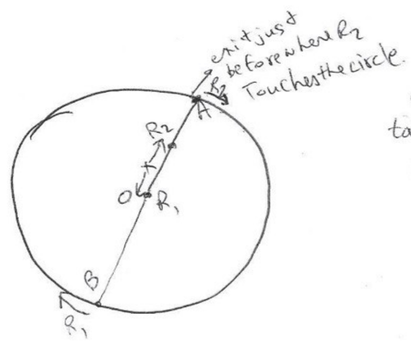
Scenario 2:

figure 1



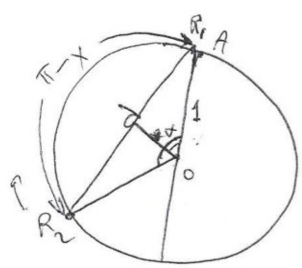
Initially Robot R_1 starts at the center O and R_2 is choosen to be anywhere inside the circle.
If I link R_1 & R_2 by a line and I assume the distance between them is x.

figure 2



take exit just before point A where R_2 touches the circle, when R_1 touches the circle, it will be at $\pi - x$ away from R_2

figure 3



In worst case:
So Evacuation time is

$1 + \pi + d$

when R_1 is at the exit, then R_2 is at distance $\pi - x$ from R_1.

I have $\alpha = \frac{\pi - x}{2}$.

$\sin\alpha = \sin\left(\frac{\pi - x}{2}\right) = \frac{d}{2}$.

So $d = 2\sin\left(\frac{\pi}{2} - \frac{x}{2}\right) = 2\cos\left(\frac{x}{2}\right)$.

Therefore for worst case scenario Evacuation time $= 1 + \pi + 2\cos\left(\frac{x}{2}\right)$

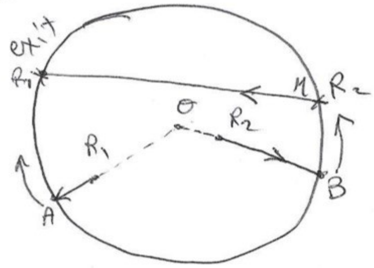
Let us say exit on left side:

In average it will take $1 + 2\cos\left(\frac{x}{2}\right)$ + distance between Robot 1 & exit

Scenario 3:

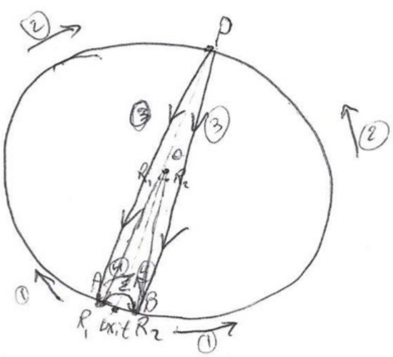
Scenario 3 part 1:

example 1 for average case:



Average Case:

Robots R_1 and R_2 are picked up randomly anywhere inside the circle.
each of the 2 Robots walk in a line defined by connecting the origin to the startpoint and moving the longer Arc length toward each other. when robot R_1 reaches the exit then it communicates with R_2 and infor him to come a straight line toward the exit.

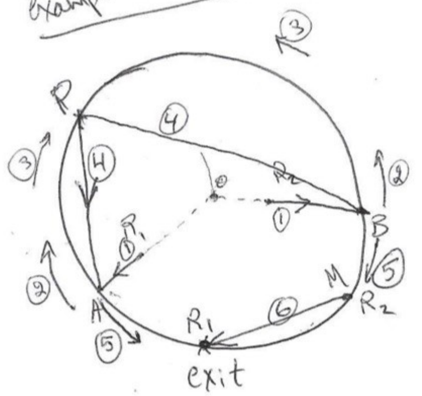


Worst case:

The worst case will be when the 2 Robots are ε (small distance) away from the center and angle between them is ε also. Each robot will move toward points A & B. After that both robots move toward each other and meet at Point P. then each robot move in a line toward A and B back. After that they move toward the exit which is located in the mid point between A & B.

Scenario 3 part 2:

example 2 for average case:



Other Average case:

The other average case will be as follows:

Robot R_1 moves toward point A which is linked by a line from the origin. Robot R_2 moves toward point B which is linked by line from origin. Each of these Robots move toward the other picking up the largest area. Both Robots meet at Point P. After that robot R_1 moves toward A and Robot R_2 moves toward B. They move toward each other. R_1 reach the exit and by that time R_2 will be at point M. then R_1 communicates with R_2 which moves in a line from point M toward the exit.

Detailed overview of scenario 3:

Select a location at random for the exit. This location is to be found exactly on the perimeter of the circle representing the room from which the robots are escaping from.

Place two robots at random anywhere inside this room.

Robot A moves in a line defined by the x and y co-ordinates of its current location and the co-ordinates of the center of the room, away from the center.

Robot B moves in a line defined by the x and y co-ordinates of its current location and the co-ordinates of the center of the room concurrently with Robot A. It also moves away from the center.

As soon as Robot A reaches the perimeter of the room it checks to see whether or not it has reached the exit by any chance. If it has, it sends its current location as x and y co-ordinates to Robot B. As soon as Robot B receives a message from Robot A, it travels in a line defined by the point of its x and y co-ordinates at the time of reception and the point defined by the x and y co-ordinates received from Robot A. Once Robot B reaches the exit the algorithm terminates; both Robots have reached the exit. (Define this situation, where one robot finds the exit and sends a message to the other robot to come over, as the "Jackpot" situation) If Robot A does not find the exit immediately it begins to travel in a circular motion in the direction defined as the direction initially pointing towards the center of the circle.

As soon as Robot B reaches the perimeter of the room it checks to see whether or not it has reached the exit by any chance. It executes the "Jackpot" situation with Robot A as the recipient of the message. If Robot B does not find the exit immediately it begins to travel in a circular motion in the direction defined as the direction initially pointing towards the center of the circle.

The robots should now be travelling in opposite directions on the greater bisection of the circle (defined by the two previously established intersection points) towards each other. If at any point a robot reaches the exit, it executes the “Jackpot” situation with the other robot as the recipient of the x and y co-ordinates of the recently found exit and the algorithm terminates. Otherwise the robots keep on travelling the perimeter of the room until they meet each other (ie their x and y co-ordinates become equal).

Robot A then changes directions and travels in a line defined by its current x and y co-ordinate location (where the two robots have met) and the x and y co-ordinate of the robots initial intersection with the perimeter of the room/circle.

Robot B, at the exact same time, changes directions and travels in a line defined by its current x and y co-ordinate location (where the two robots have met) and the x and y co-ordinate of the robots initial intersection with the perimeter of the room/circle.

As soon as Robot A reaches its original point of intersection with the room, it begins to travel across the perimeter of the room in the direction facing away from the center of the room/circle.

As soon as Robot B reaches its original point of intersection with the room, it begins to travel across the perimeter of the room in the direction facing away from the center of the room/circle.

Both robots should now be travelling in opposite directions across the remaining unexplored section of the perimeter of the room. The exit must be placed somewhere in this remaining section of the perimeter. As soon as one of the robots reaches the exit, it executes the

“Jackpot” situation in order to inform the other robot of the location of the exit. This will cause that robot to move towards and reach the exit. The algorithm then terminates.