Bike Assisted Linear Search and Evacuation

By

Khaled Jawhar

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs in partial fulfillment of the requirements for the degree of

Master

in

Computer Science

Carleton University

Ottawa, Ontario

Abstract

Many linear search and evacuation problems have been studied on robots that move with unit speed seeking to evacuate through an unknown exit located on a line. It is interesting to study the linear search and evacuation problems on robots with variable maximum speed in order to compare the results based on that speed. What makes the problem more interesting is to include a passive entity, such as a bike, that can be used by any of the robots. This will make the problem more generalized and thus it may encompass many previously done studies on linear search and evacuation, as we will demonstrate later. We will revisit the linear search and evacuation problems; however, this time we will use two robots with a bike under the condition that only one robot can use the bike at a time. The exit would be placed at an unknown position on the line. The direction that the robot should follow to reach the exit is unknown. The problem will be divided into two categories: linear search and evacuation. Evacuation in turn will be studied for two communication models: wi-fi and face-to-face.

Regarding the linear search problem, we have shown two different algorithms that are optimal (relative to the upper bound) based on the maximum speed. Additionally, we provided a section related to the lower bound. Regarding the wi-fi evacuation model, we have shown three different algorithms, two of which are optimal (relative to the upper bound) based on the maximum speed. Furthermore, we have also provided the lower bound for the wi-fi model. Regarding the face-to-face model, we have shown three different algorithms (relative to the upper bound), one of which is optimal regardless of the value of the maximum speed. Many graphs have been provided to illustrate how each model performs.

Acknowledgements

To accomplish this thesis, I have received unlimited help, support and patience from my principal supervisor, Prof. Evangelos Kranakis, not to mention his advice and invaluable knowledge of distributed computing. The support and advice of my committee have been invaluable as well on both academic and personal level. I am extremely grateful for that. I would like to acknowledge the support of Carleton University and its technical staff.

I would like to thank my parents, whose love and guidance are with me in whatever I pursue. Most importantly, I would like to thank my wife Sarah Hamdan and my daughter Fatima Jawhar for their love and motivation throughout the writing of the thesis.

Table of Contents

List of Tables

List of Illustrations

Chapter 1

Introduction

1.1 Overview

Linear search and evacuation by autonomous robots are important problems in the area of distributed computing. Linear search is complete when the first robot reaches the exit while evacuation is complete when the last robot reaches the exit.

First search problems for a single robot in the literature were studied on a continuous infinite line by Beck[6] and Bellman [9]. Assuming that the distance and the direction to the exit is unknown, they proposed an optimal algorithm with competitive ratio 9. The term "competitive ratio" is the natural measure of how well an algorithm performs in competition with an adversary that knows the exact location of the exit (i.e. the ratio between the maximum time taken by the robot using a certain strategy to reach an unknown exit and the time taken by the adversary to proceed to the exit directly). The optimal algorithm for the linear search or evacuation problems is the one with minimal linear search time or evacuation time respectively. Efficiency or optimality of an algorithm is sometimes measured using competitive ratio.

There are many factors that affect how linear search and evacuation problems are solved. Let us assume that the exit is located on a line at a distance d from the origin where the robot starts. The orientation represents the direction that the robot must proceed to reach the exit. The simplest case would be if a robot moving with unit speed knows the distance and the orientation and can thus reach the exit in time d. In this case, the competitive ratio will be the time needed by the robot to reach the exit, which is d, divided by the time needed by the adversary to head to the exit directly, which is d as well. Thus the competitive ratio will be 1. A more complicated case would be if the distance is known and the orientation is not known. As a worst case scenario, it may take the robot $3d$ to find the exit since the robot may move d in the wrong direction and thus it will need to switch the direction and move back $2d$ to reach the exit. The competitive ratio in this case is 3. The problem is even more complicated if both the distance and the orientation are not known.

Specifically in the evacuation problem, further complexity can be introduced by increasing the number of robots. This would require that which ever robot finds the exit would need to communicate with the others, and the challenge will be to reduce the time needed for all robots to evacuate.

In the literature, search and evacuation problems have been studied with numerous variations, including starting point (e.g. robots may not know their starting point), shape of the field (e.g. line, circle, square, etc.), information processing abilities of the robot, and tools that aid the robot. An example of a tool would be a bike which is used in the case under study in this thesis.

1.2 Model

We will consider two mobile robots situated initially at some point on a line. The mobile robot can behave like an autonomous vehicle. It can move, stop, turn and switch direction at any time. It can process information and execute algorithms as well. We consider an adversary that can locate the exit at any location which is not known to the robot. The exit is situated at an unknown distance d away from the starting point of the two robots. The robot can walk up to a maximum unit speed. There is a passive entity (i.e. a bike) that is used to aid any of the robots, but which can not be used by more than one robot at once. The robot that is using the bike can move up to a maximum speed $v > 1$. In this environment, we will study linear search and evacuation problems.

Two models can be applied to the robot evacuation problem. The first model is the wireless communication model and the second is the face-to-face model. In the wireless communication model, the robots can communicate with each other at any time. The face-to-face model is more challenging and complex to solve, as the robots need to meet at some point in order to exchange information. For linear search problems, we will be interested in designing algorithms that minimize the time it takes for any of the two robots to discover the exit. Evacuation problems require an additional step, as the algorithm must minimize the time required for the robot which reaches the exit first to bring the other robot to the exit. Both robots should share the bike wisely in order to achieve the minimal evacuation time. Furthermore, the algorithm will specify what trajectory will be followed by each of the robots. The goal is to find the optimal algorithm that guarantees the minimal linear search time or evacuation time.

1.3 Results

In this thesis, we consider linear search and evacuation problems for two robots equipped with a single bike. The linear search problem has two different optimal algorithms depending on the maximum speed v. The first algorithm has both robots moving in opposite directions to search for the exit, while the second algorithm has the faster robot (i.e. the robot with the bike) searching for the exit using a Zig-Zag strategy (explained in the next chapter). If the speed is less than 9, the first algorithm performs better, otherwise the second algorithm performs better. We concluded by providing a lower bound for the linear search problem which is shown to be $\max\{\frac{3d}{v}, d + \frac{d}{v}\}$. The results for the above 2 algorithms are shown in the table below:

Table 1.1: Linear search results (Two robots with a bike)

Algorithm	Title	Evacuation time
5	Opposite direction With Max Speed	d
6	Zig-Zag With Max Speed	$\frac{9d}{v}$

We have studied the evacuation problem using wi-fi and face-to-face models. Pertaining to the wi-fi model, we proposed three algorithms, with two being optimal based on the maximum speed v. Using the first algorithm, both robots move in opposite directions to search for the exit, then as soon as one of the robots finds the exit, it will communicate with the other robot to proceed to the exit. Anyone may think that the first algorithm performs better, since both robots are moving at maximum speed, but this is not the case. The second algorithm is similar to the first one, except that the faster robot won't use its maximum speed while trying to search for the exit. The second algorithm is actually better than the first one for any value of v. This is because when the faster robot reduces its speed while trying to search for the exit, the overall evacuation time for both robots improves. The third algorithm performs best for high speed v. In this algorithm, both robots use a Zig-Zag strategy, but the slower robot will try to stay as close as possible to the faster robot who is using the bike. Clearly the faster robot will find the exit first and will communicate with the slower robot to inform it about the location of the exit. After that, the faster robot will move back to drop off the bike for the slower robot who will grab it and proceed to the exit.

For the wi-fi model, the second and the third algorithms perform better depending on the maximum speed v. We concluded the wi-fi model by providing a proof of the lower bound which is shown to be $\max\{\frac{3d}{v}, \frac{d(v+4)}{2v}, \frac{d(v+3)}{2v}, 2d+\frac{d}{2v}+\frac{d}{2v^2}, d+\frac{d}{v}\}$. The results for the above three algorithms are shown in the table below:

Table 1.2: Wi-fi model results (Two robots with a bike)

Algorithm	Title	Evacuation time
7	Opposite Direction With Max Speed	$d(2+\frac{1}{v})$
8	Opposite Direction With Optimal Speed	$\frac{3d+3dv+d\sqrt{v^2+26v+9}}{4v}$
9	Slower Robot Imitates Faster Robot	$\frac{9d}{v}+\frac{d}{2}-\frac{d}{2v^2}$

Now we will move to the face-to-face model. There are three face-to-face algorithms, with one being optimal regardless of the maximum speed. In the first algorithm, the robot with the bike will use a doubling strategy such that it will use its maximum speed v and move 2^k during each iteration. The slower robot will move with its unit speed and will follow the faster robot until they meet. At the meeting point, the slower robot will reverse its direction to follow the faster robot and this will be repeated during each iteration. The points of intersection of both robots during each iteration will be a sequence that is determined. During the last iteration, when the slower robot doesn't find the faster robot at the specific point of intersection, it will automatically know that the faster robot has reached the exit and it will proceed to the exit directly. In the second algorithm, the faster robot will use a doubling strategy and will move 2^k during each iteration k, but the slower robot will use its own doubling strategy. Both robots will meet at the peak points reached by the slower robot during each iteration. In the third algorithm, we started by proposing an optimal algorithm for the case of two robots without a bike first, since this was useful in finding the optimal algorithm for the case of two robots with a bike. The results are shown in the tables below:

Table 1.3: Face-to-face model results (Two robots without a bike)

Algorithm 12 for two robots without a bike	Evacuation time
Under Assumption ($v \leq$3)	$3d-\frac{6d(v-3)}{3v-1}$
Under Assumption ($v \geq$3)	$3d-\frac{6d(v-3)^2}{(v-1)(3v-1)}$

Table 1.4: Face-to-face model results (Two robots with a bike)

Algorithm	Title	Evacuation time
10	Slower Robot Pursue Faster Robot	$d + \frac{9d}{v} - \frac{5d}{8v^2}$
11	Slower Robot Evacuate Close To Exit Without Aid Algorithm	$\frac{3dv^3+63dv^2+15dv-9d}{2v^2(3v+1)}$
13	Nearest Meeting To Exit ($v \leq \frac{6+\sqrt{41}}{5}$)	$3d - \frac{5v^2-12v-1}{2v(v-1)}d - \frac{5v^2-12v-1}{v(v-1)(3v-1)}d$
13	Nearest Meeting To Exit ($v \geq \frac{6+\sqrt{41}}{5}$)	$3d - \frac{5v^2-12v-1}{2v(v-1)}d + \frac{4(5v^2-12v-1)}{v(v-1)(3v-1)}d$

Results of the thesis will be published in a forthcoming paper [20]

1.4 Outline

In the next chapter we provide some example studies related to linear search and evacuation problems. We started with linear search for a single robot that has unit speed and we showed that when the location and the direction of the exit are not known then the upper bound for the search time is $9d$. We provided some other examples of linear search with two robots. Next was the evacuation problem with two robots, one of which has unit speed while the other has speed between $\frac{1}{3}$ and 1. We showed that the upper bound for the evacuation time is $9d$. We concluded the chapter by highlighting the evacuation problem of two robots with unit speed, but this time on a circle using wi-fi and face-to-face models.

In chapter 3, we started with the main subject of the thesis. We have studied the linear search problem of two robots with a bike. We presented two algorithms that perform better based on the value of the maximum speed v. Additionally, we provided a small section related to the lower bound. In the following two chapters, we switched to the evacuation problem for two robots with a bike using wi-fi and face-to-face models. For the wi-fi model, we have shown three different algorithms under which two algorithms are optimal based on the value of the maximum speed v. We concluded the chapter by showing the lower bound. Furthermore, we studied the face-to-face model and we presented three different algorithms under which one of them is optimal regardless of the value of the maximum speed v. In order to achieve this result, we initially studied the evacuation problem on two robots without a bike and then switched to the two robots with a bike model.

1.5 Related Work

Many researchers have studied the linear search and evacuation problems with one or more robots moving at different speeds. First search problem in the literature focused on infinite line [9] with emphasis on stochastic search models and their analysis. Additional research on stochastic models can be found in the books [1, 23] as well as in [2] which emphasizes on the rendezvous problem. Many variants of the search problem have been considered, such as having static or moving targets, multiple robots with or without communication capabilities, and in environments that may not be known by the robots ahead of time. The search problem has been studied in environments with a pair of distinct speed robots [5], on various communication models such as wi-fi and face-to-face [19] and on many different environments [8, 10, 11]. Evacuation problems, such that the exit must be reached by all robots, have been studied extensively as well. Evacuation was studied on different communication models [17, 22] and in various environments including disks, triangles, and circles [12, 14, 15, 16].

Chapter 2

Background

In this chapter we will highlight various cases of linear search and evacuation. We will start by considering the linear search problem with a single robot and we will show that if the direction and the exit are unknown, then the upper bound will be $9d$. We will also study the linear search problem with two robots. We will show that having two robots will improve the search time. Then we will switch to the evacuation problem with two robots. We will show the case of two robots seeking to evacuate through an unknown exit located on a line. We consider that one of the robots has unit speed and the other has speed between $\frac{1}{3}$ and 1. The surprising result is that the upper bound will be $9d$. Finally, we will show the competitive ratio of the wi-fi and the face-to-face models of two robots starting at the center of the circle and seeking to evacuate through an unknown exit that is located at the circumference of the circle.

2.1 Linear Search with a single robot

We consider a single robot that is located at the origin. The robot can move with maximum speed 1. The search for the exit is done in one dimension and on an infinite line. The search terminates when the robot finds the exit. There are many cases that can be considered.

- Case 1: If the direction and the distance d to the exit are known

  If the exit is situated at distance d from the origin, then the robot will find the exit in time d.

- Case 2: If the distance d to the exit is known but the direction is not known, we can consider the following algorithm:

Algorithm 1: Algorithm for linear search with known distance and unknown direction

```
Start from the origin O;
Choose a direction;
Move d;
if exit is not found) then
    Reverse the direction;
    Move 2d;
end
Quit search;
```

  The robot will choose a direction and move. The worst case scenario would be if the direction that is chosen by the robot is the wrong one. In this case since the robot won't find the exit at distance d in the wrong direction, then it will switch direction and go back $2d$ to find the exit at the other side. Since the robot is moving with unit speed then it will reach the exit in time $3d$. Therefore the competitive ratio of this algorithm will be $\frac{3d}{d} = 3$. If the robot visits one of the two points $(-d, d)$, the adversary will try to put the exit at the other point. Thus it is not possible for the robot to visit both points in time less than $3d$. We conclude that $3d$ is also a lower bound and the competitive ratio of any algorithm would be at least $3d$.

- Case 3: If the direction and the distance d to the exit are not known

  The robot starts at the origin and can move with speed 1. The robot needs to explore both directions in order to find the exit. The best way to achieve this goal is to select a direction and move distance 1. If the exit is not found, the robot will reverse direction and move double the previous distance up until the exit is found. The movement, which is repeated periodically, and which uses a sequence of positive distances that specifies the turning points, is called the Zig-Zag search algorithm. The competitive ratio for the Zig-Zag search algorithm is determined to be 9. A canonical Zig-Zag search algorithm is defined below:

Algorithm 2: Zig-Zag Algorithm

Consider X as infinite sequence of distances 2^0,2^1,...;

for $i \leftarrow 0$ **to** ∞ **do**

  if *i is odd(resp.even)* **then**

    Move right (resp. left) a distance 2^k unless the exit is found;

    if *exit is found* **then**

      Quit search

    end

    Turn; then move left (resp. right), return to origin

  end

end

The competitive ratio for the above algorithm is calculated as follows:

Every time the robot changes direction and moves twice the previous distance. Thus, if the exit is at distance d, then $2^k < d \leq 2^{k+2}$ for some k. Hence the search time T will be calculated as follows:

$$T = 2 \cdot 1 + 2 \cdot 2 + \cdots + 2 \cdot 2^{k+1} + d$$

$$= 2 \cdot (2^{k+2} - 1) + d = 2^3 \cdot 2^k - 2 + d \leq 8d + d = 9d$$

Thus, the competitive ratio of this algorithm is $\frac{9d}{d} = 9$.

The competitive ratio of 9 has also been proven to be the lower bound for the linear search problem with a single robot having maximum unit speed [7].

Assuming, as in the previous case, that the direction and the distance to the exit is unknown, let us have the following variation to the linear search with a single robot model. Assume that the robot moves with speed 1 away from the origin, and speed $v > 1$ towards the origin [18]. In order to find the exit, the robot would use the Zig-Zag search algorithm that is mentioned above. The difference is that during iteration k, the robot needs time 2^k to move distance 2^k away from the origin and $\frac{2^k}{v}$ to come back to the origin. So the time that is needed to complete iteration k would be $2^k + \frac{2^k}{v}$. In order to calculate the upper bound, the adversary can place the exit between 2^k and 2^{k+1}. In this case, the robot needs to complete the $(k+1)$ iteration in order to find the exit during the $(k+2)$ iteration, since the adversary can place the exit at the other side of the origin. After doing the calculation, the time needed will be $(1+\frac{1}{s})(2^{k+2}-1)$ to reach the origin and an additional d is needed to reach the exit, since the robot is moving away from the origin with unit speed. Thus the competitive ratio will be $5+\frac{4}{s}$. Additional work has been done on the linear search with single robot problem e.g. [3, 4, 21]

2.2 Linear search with two robots

After discussing linear search with a single robot, it would be interesting to explore whether more than one robot can do the search faster. An interesting study called "parallel search" was proposed. In parallel search, we assume that there are two robots that have maximum unit speed. We denote the distance to the exit by d . Without loss of generality, we assume that the two robots are initially placed at the same starting point and that the exit is situated at an unknown point on a line. We assume that the direction to the exit is unknown. Since we are studying linear search, it is sufficient for one robot to find the exit. In other words, when one robot finds the exit, it does not care about having the other robot proceed to the exit. There are two cases to consider:

- Case 1: The distance to the exit is known and the direction is not known

  Since the direction to the exit is not known, then both robots will move in opposite directions by distance d. One of the two robots will definitely find the exit, and will take time d to do so. The competitive ratio in this case is 1, in comparison to 3 for the linear search with a single robot. The algorithm is as follows:

Algorithm 3: Algorithm for linear search with known distance and unknown direction

Both robots start from the origin O and move distance d in opposite directions;

The robot who finds the exit shouts "I found the exit";

Quit search;

- Case 2: The distance and the direction to the exit are not known

  Assume that the exit is situated at a distance d which is not known to the robots. Both robots will move in opposite directions until one of them finds the exit, after which it will shout "I found the exit." The time needed to find the exit by one of the robots in this case is d and the competitive ratio is 1, in comparison to 9 for the line search with a single robot. Having more than two robots with maximum unit speed won't help in achieving a better competitive ratio. The algorithm is as follows:

Algorithm 4: Algorithm for linear search with an unknown distance and direction

Both robots start from the origin O and move in opposite directions;

The robot who finds the exit shouts "I found the exit";

Quit search;

As we have shown in the previous examples about linear search, the problem terminates when the first robot succeeds in finding the exit. In the case of two robots, one more constraint can be added which is to have the second robot evacuate through the exit as well. This will introduce the evacuation problem. After highlighting the linear search problem with one and two robots, we would like to provide some examples of the evacuation problem.

2.3 Linear evacuation of two robots, faster with unit speed and slower with speed less than one

An interesting case arises when there are two robots, one with unit speed and the other with speed less than 1 [13]. The surprising result is that the evacuation time is proven to be less than $9d$ if the speed of the slower robot is larger than $\frac{1}{3}$. The proof of this is illustrated by simply having the robot with unit speed follow the doubling strategy until it reaches the exit. The slower robot will use speed $\frac{1}{3}$ and will start after 4 units of time, assuming that both robots are at the origin when they start. The slower robot will follow the doubling strategy as well but it will take more time to complete each iteration. For the slower robot to execute one iteration using its doubling strategy, it needs $2 \cdot 3 \cdot 2^k$ and this is because it is moving with speed $\frac{1}{3}$ which means that the time taken by the slower robot to complete each iteration should be equal to the time needed by the faster robot to complete each iteration multiplied by 3. The meeting points for both robots will be at the extreme points that were reached by the faster robot during each iteration.

In order to calculate the evacuation time, we need to find the time needed by the faster robot to reach the exit. We add to it the time that the faster robot needs to inform the slower robot about the path to the exit and finally we add the time needed by both robots to reach the exit. We assume that the distance d is between 2^{k-2} and 2^k. Let us say that the distance to the exit is $d = 2^{k-2} + \epsilon$, where ϵ is a positive number. When the faster robot reaches the destination, the slower one will be at a distance $\frac{4}{3}\epsilon$ away from the faster robot. This is because the last meeting point between the two robots would be at 2^{k-2}, and the faster robot will move ϵ to reach the exit while the slower robot will move $\frac{\epsilon}{3}$ in the opposite direction, leaving the robots with $\frac{4\epsilon}{3}$ distance between them. All that the faster robot needs to do is to move toward the slower robot and inform it about the location of the exit. The faster robot needs time $\left(\frac{\frac{4\epsilon}{3}}{\frac{2}{3}}\right) = 2\epsilon$ to meet the slower robot and to inform it of the exit. In order for both robots to go back to the exit, it is sufficient to find the time needed by the slower robot to reach the exit, which is $\left(\frac{(2\epsilon)}{\frac{1}{3}} = 6\epsilon\right)$.

The evacuation time needed will be the value of the evacuation time for the doubling strategy used by the faster robot which is $2(2^k - 1) + d$, and we add to it $2\epsilon + 6\epsilon = 8\epsilon$. Hence the result of the evacuation time for the upper bound case will be $2(2^k - 1) + d + 8\epsilon = 2(4(2^{k-2} + \epsilon) - 4\epsilon - 1) + d + 8\epsilon = 8d - 8\epsilon - 2 + d + 8\epsilon = 9d - 2$. We conclude that if the speed of the slower robot is larger than $\frac{1}{3}$, then the evacuation time for the upper bound case will be $9d$. On the other hand, when the speed of the slower robot is less than $\frac{1}{3}$, the evacuation time for the pair will be larger than $9d$. Another related case is having two fast mobile robots and one slow robot. The same procedure that was followed with one fast and one slow mobile robots can be followed here as well. The fast robot needs $4d$ to reach the exit and $5d$ to inform the other fast mobile robot. If the slow mobile robot has speed of at least $\frac{1}{5}$ then the fast and the slow robot will arrive at the same time to the exit. This proves that this case also has an evacuation time of $9d$.

2.4 Evacuation through the circumference of the circle with two robots having unit speed

After providing some examples of robot evacuation on a line, we will provide an example on a circle as well. In such a case, the two robots are located at the center of the circle. Many other factors can be taken into consideration as well, such as the speed of the robots and whether the exit is on the circumference or inside the circle. Many more complications can be added such as having more robots to evacuate. We consider the case when both robots start at the center and the exit is unknown and located at the circumference of the circle. Two communication models are taken into account: wi-fi communication model and face-to-face model.

2.4.1 Wi-fi Model with robots starting at the center

In the wireless communication model case, the message that is broadcast can be acquired by other robots independent of their current position in the circle. We will start with a case study about two robots starting with the same speed at the origin having the exit situated at an unknown location at the circumference of the circle. The objective is to find the optimal algorithm for the two robots to evacuate which can lead to the best evacuation time for both robots to reach the exit.

Let us start with the scenario that both robots walk together until they hit the circle at a point A. Assume that at A the robots explore the perimeter in opposite directions. One of the robots, let us say R_1, finds the exit. Then obviously considering wireless communication model, R_1 will inform the other robot R_2 which in turn will move directly through a chord toward the exit from point B. Definitely the robot R_1 will move on the circumference of the circle with an angular distance which is less than $180°$, since at $180°$ the two robots will meet and this is the only point where the two robots can both meet at the exit without communicating with each other. If the angular distance between A and B is equal to c, then the length of the chord taken by robot R_2 will be equal to $2\sin(\pi - c)$ which is $2\sin c$. In the worst case scenario the evacuation time will be $\max\{1 + c + 2\sin c\}$, given that c is between $[0, \pi]$ as shown in the figure below. It is interesting to find the point on the circle where the time needed is maximized. Consider the function $f(c) = 1 + c + 2\sin c$ in the interval $[0, \pi]$. The function is maximized at the point B with angle $2\frac{\pi}{3}$ and hence $f(c)$ will take the value $1 + 2\frac{\pi}{3} + \sqrt{3}$ which is approximately equals to 4.826.

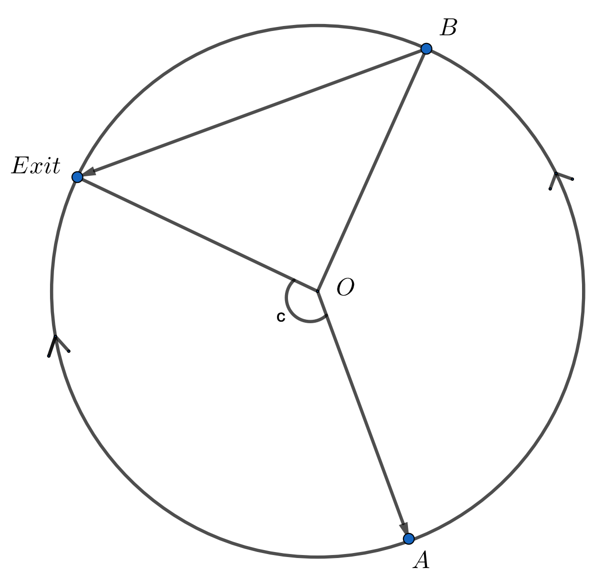

Figure 2.1: The two robots start at the center O of the circle. They hit the circle at A then each of the two robots moves in opposite direction. The robot who will hit the exit first will communicate with the other robot who is at B to proceed toward the exit.

2.4.2 Face-to-face Model with robots starting at the center

Another communication model that can be considered for the case of the circle is the face-to-face model. In this case both robots walk together until they hit the circle at a point A. Assume that at point A, each of the two robots move in the opposite direction. Let us say one of the two robots R_1 finds the exit first at some point B. At this moment the other robot R_2 will be at point C symmetric to B. Then R_1 will move toward a point D in a way that the chord BD will be equal to the arc CD. Both robots will arrive at the point D at the same time since the length of the arc CD is equal to the chord BD. Both robots will head toward the exit situated at B. It is proven in this case that the upper bound is equals to $1 + \frac{\alpha}{2} + 3\sin\frac{\alpha}{2}$ which is approximately equal to 5.74 given that the unit radius is 1 and $\cos\frac{\alpha}{2} = -\frac{1}{3}$.

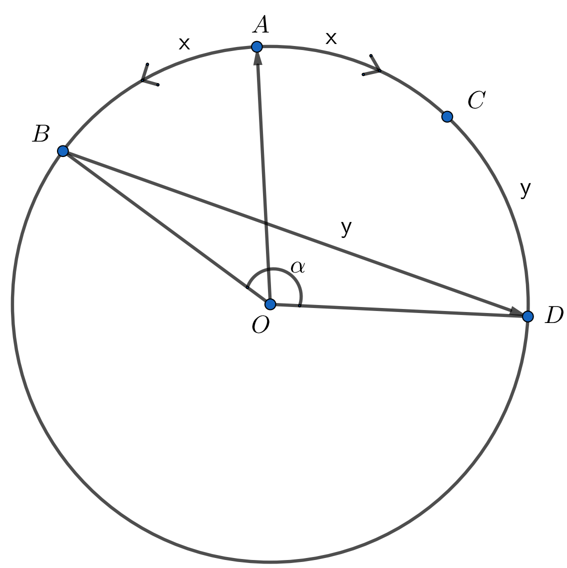

Figure 2.2: The two robots start at the origin. They hit the circle at A. Then one of the two robots reaches the exit at B in which case the other robot will be at C. The robot who reaches the exit moves in a chord BD to meet the other robot at D, then both robots head toward the exit.

Chapter 3

Linear Search

In this chapter we will study the linear search problem for two robots seeking to evacuate through an unknown exit located on a line. Any of the robots can walk with unit speed. There is one bike that can be used by any of the robots. The robot who is using the bike can move with speed v such that $1 \leq v$. Consider two robots R_1 and R_2; assume that the exit is at distance d away from the origin and that the two robots start at the origin. The objective is to find the minimal time needed to find the exit by any of the two robots. Definitely for the linear search problem there is no point in switching the bike between the two robots since the objective is to find the exit by only one of them. We will consider two algorithms to solve this problem and each one will show better results depending on the speed v. For the following two algorithms, let us assume that robot R_1 is using the bike and moving with speed $v \geq 1$ while the other robot R_2 is moving with unit speed. We will start with the first algorithm.

3.1 Opposite Direction With Max Speed Algorithm

The first algorithm is nothing more than having the two robots move in opposite directions. The search is complete when the exit is found by any of the two robots.

Algorithm 5: OppDirectionWithMaxSpeedLS

R_1 moves right with speed v ;

R_2 moves left with unit speed ;

if *The exit is found by any robot* **then**
    Quit search;

end

Theorem 3.1.1. *The search time for Algorithm 5 is upper bounded by* d

Proof. Assume the robots move in opposite directions. Without loss of generality, assume that robot R_1 who is using the bike moves to the right with speed v and that robot R_2 who is walking moves to the left with unit speed. Then there are two cases to consider.

- The exit is found by the robot who is using the bike

  Evacuation time $T_1 = \frac{d}{v}$

- The exit is found by the robot who is walking.

  Evacuation time $T_2 = d$

Therefore Evacuation time $T = \max\{T_1, T_2\} = \max\{d, \frac{d}{v}\} = d$ □

3.2 Zig-Zag With Max Speed Algorithm

The disadvantage of having the two robots move in opposite directions is that the faster robot will use its speed to search in one direction only. Since the objective for linear search is to find the exit, it makes more sense for the faster robot that is using the bike to search for the exit. The robot with the bike will search for the exit using a Zig-Zag strategy with its maximum speed v. The algorithm becomes as follows:

Algorithm 6: ZigZagWithMaxSpeed

for $k \leftarrow 1$ **to** ∞ **do**
  if *k is odd(resp.even)* **then**
    Move right (resp. left) a distance 2^k with maximum speed v unless the exit is found;
    if *exit is found* **then**
      Quit search
    end
    Turn; then move left (resp. right), return to origin
  end
end

Theorem 3.2.1. *The search time for Algorithm6 is upper bounded by* $\frac{9d}{v}$ *and is optimal if* $9 \leq v$

Proof. Assume that the exit is at distance d away from the origin O and that both robots R_1 and R_2 start at O. As the figure below shows, the faster robot will use a doubling strategy. In this case, the faster robot who is using the bike will need time $2 \cdot \frac{2^k}{v}$ to complete any iteration k. Assume that $2^{k-2} < d \leq 2^k$. Based on that, the search time will be as follows:

$$\begin{aligned}
T &= \frac{2 \cdot 2^0 + 2 \cdot 2^1 + \cdots + 2 \cdot 2^{k-1}}{v} + \frac{d}{v} \\
&= \frac{2(2^k - 1)}{v} + \frac{d}{v} \\
&= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} \\
&\leq \frac{9d}{v}
\end{aligned}$$

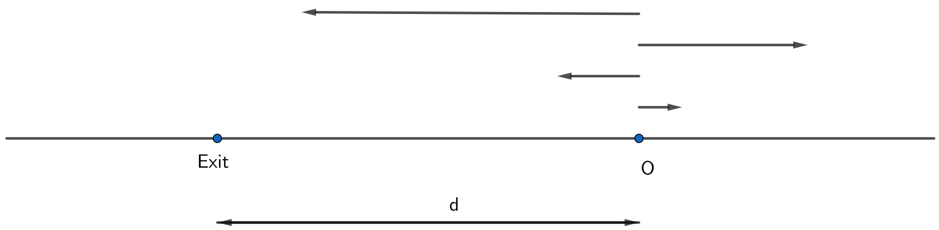

Figure 3.1: The robot with the bike will use doubling strategy and will move with speed v to search for the exit which is at distance d away from the origin.

□

3.3 Comparison of the two algorithms

In order to find the value of v for which the second algorithm is better, we will set the evacuation time for the first algorithm less than the evacuation time for the second algorithm. This leads to the following:

$$d < \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v}$$
$$\Leftrightarrow (v-1)d - 2^{k+1} + 2 < 0$$
$$\Leftrightarrow (v-1)d - 8.2^{k-2} + 2 < 0$$
$$\Leftrightarrow v < \frac{8 \cdot 2^{k-2}}{d} - \frac{2}{d} + 1$$
$$\Leftrightarrow v < \frac{8d}{d} - \frac{2}{d} + 1 < 9$$

Thus Algorithm 5 is better than Algorithm 6 if $v < 9$. This can be shown in the following graph where the intersection point represents that with speed 9.

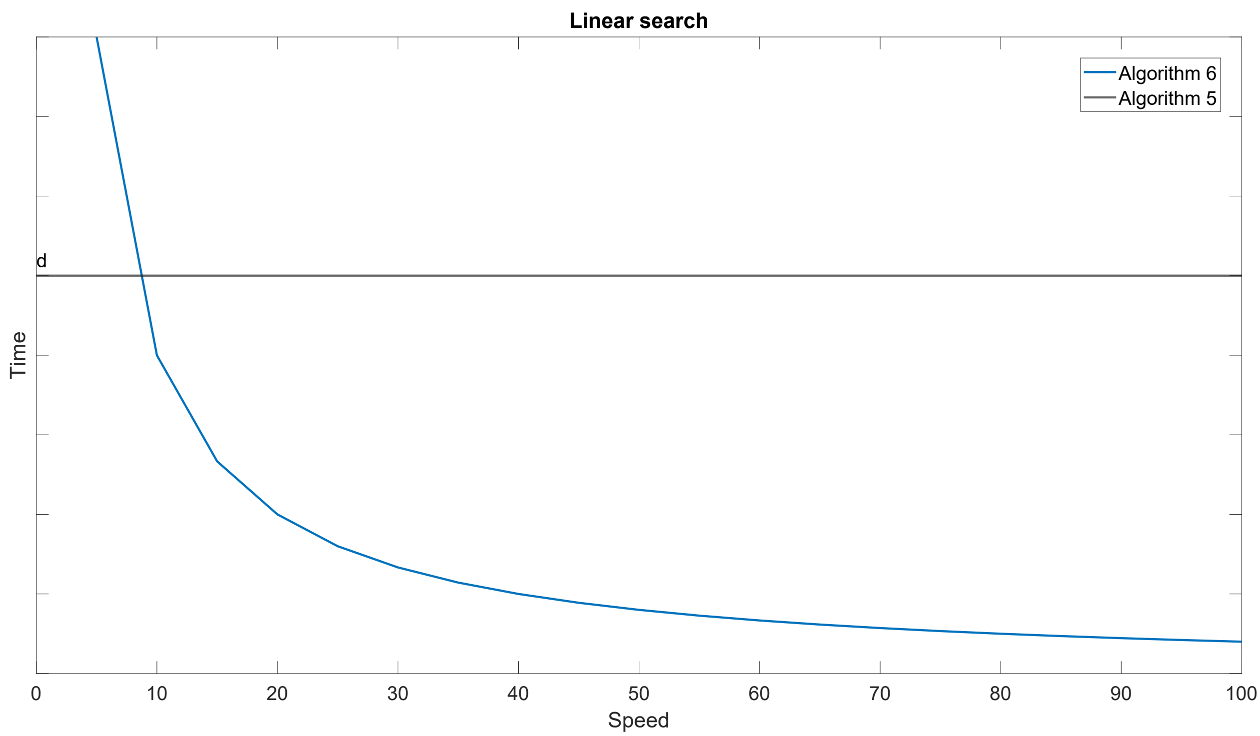

Figure 3.2: As the graph shows, when the speed is higher than 9, then algorithm6 performs better

3.4 Lower bound for two robots with a bike using linear search

Assume that there are two robots, with one walking at unit speed and the other using the bike at speed $v \geq 1$. One of the two robots must visit both points $-d$ and $+d$. Without loss of generality assume that -d is the first point to be visited by one of the two robots, then the adversary will place the exit at $+d$. Consider that t is the time needed by any of the robots to reach the point $-d$ first. We have two cases to consider:

- Case 1: $-d$ is visited by the robot with the bike first

  If the walking robot is between [$-d$,O], then the best case scenario would be if the walking robot is at O. The robot with the bike needs $\frac{2d}{v}$ to reach the exit versus d by the walking robot. Thus evacuation time $T_1 = \min\{t + \frac{2d}{v}, t + d\} \geq \min\{\frac{d}{v} + \frac{2d}{v}, \frac{d}{v} + d\} \geq \min\{\frac{3d}{v}, \frac{d}{v} + d\}$

- Case 2: $-d$ is visited by the walking robot first

If the robot with the bike is between [$-d$,O], then the best case scenario would be if the robot with the bike is at O. The walking robot needs $2d$ to reach the exit versus $\frac{d}{v}$ by the robot with the bike. Thus evacuation time $T_2 = \min\{t + \frac{d}{v}, t + 2d\} \geq \min\{d + \frac{d}{v}, 3d\} \geq d + \frac{d}{v}$

We conclude that the evacuation time $T = \max\{T_1, T_2\} = \max\{\frac{3d}{v}, d + \frac{d}{v}\}$

Chapter 4

Evacuation using Wi-Fi communication Model

As mentioned previously, consider two robots starting from the origin. Both robots can walk with maximum speed 1. The exit is placed at distance d away from the origin. The distance, direction, and location of the exit is unknown to the robots. There is a bike available that can be used by any of the two robots. The robot using the bike can move up to a maximum speed $v \geq 1$. We will discuss in this chapter the wireless communication model for this problem, which means that the two robots can communicate between each other at any time regardless of their distance. There are three algorithms that will be considered in this case. two of which will be optimal depending on the maximum speed v.

4.1 Opposite Direction With Max Speed Algorithm

In this algorithm, the robots move in opposite directions with their maximum speed, assuming that the robot riding the bike moves with speed v. The robot who finds the exit first will communicate with the other robot to proceed to the exit. The algorithm is as follows:

Algorithm 7: OppDirectionWithMaxSpeedWiFi

Each robot goes in opposite directions (assuming one robot is using the bike) ;

The robot that finds the exit communicates with the other robot ;

The other robot heads to the exit

Theorem 4.1.1. *The evacuation time for Algorithm 7 using the Wi-Fi model is upper bounded by*

$$\max\left\{2d + \frac{d}{v}, \frac{2d}{v} + \frac{d}{2} + \frac{d}{2v^2}\right\}$$

Proof. There are two cases to consider here:

- Case 1: The exit is found by the robot riding the bike.

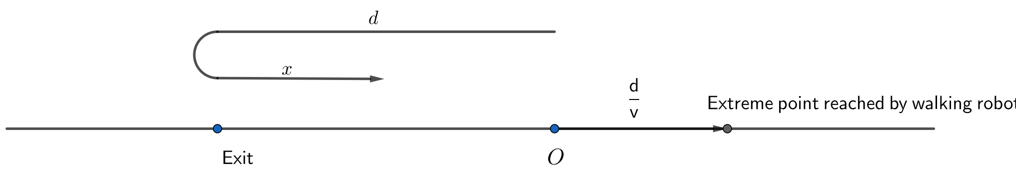

Figure 4.1: The robot with the bike will move with speed v and will reach the exit, which is situated at distance d away from O. Then it will drop off the bike at distance x. The walking robot will come from point $\frac{d}{v}$ to pick up the bike and will arrive to the exit at the same time as the other robot.

When the robot with the bike finds the exit, which is at distance d from the origin, it needs time $\frac{d}{v}$. The robot with the bike will communicate with the other robot, who is walking, to come to the exit. Since the exit is found by the robot who is riding the bike first, then the bike can be dropped off at some distance x away from the exit such that the walking robot can pick it up and arrive to the exit at the same time as the other robot. To find the distance x where the bike is dropped off, we have $\frac{x}{v} + x = d + \frac{d}{v} - x + \frac{x}{v}$. This leads to $x = \frac{d}{2} + \frac{d}{2v}$.

Hence the walking robot who is at distance $\frac{d}{v}$ when the faster robot reaches the exit, will need $\frac{d}{v}$ to reach the origin, in addition to $d - x + \frac{x}{v} = \frac{d}{2} + \frac{d}{2v^2}$ to reach the exit. Therefore, the evacuation time will be $\frac{2d}{v} + \frac{d}{2} + \frac{d}{2v^2}$.

- Case 2: The exit is found by the walking robot.

  When the robot who is walking finds the exit which is at distance d to the left of the origin, the robot with the bike will be at distance dv to the right of the origin. The walking robot will communicate with the other robot who is using the bike to come to the exit. The robot with the bike turns back and goes to the exit with additional time $d + \frac{d}{v}$. The evacuation time in this case will be $d + d + \frac{d}{v} = 2d + \frac{d}{v}$.

Therefore the evacuation time for this algorithm will be $\max\left\{2d + \frac{d}{v}, \frac{2d}{v} + \frac{d}{2} + \frac{d}{2v^2}\right\}$. □

4.2 Opposite Direction With Optimal Speed Algorithm

In the previous algorithm, anyone would assume that it is better for each of the robots to use its maximum speed. However, this is not what the results of this study have shown. We illustrate in the following example that having the robot with the bike moves with unit speed until one of the two robots reaches the exit, will lead to an evacuation time which is as good as that of Algorithm 7.

Example 4.2.1. Assume that robot R_1 is walking and moving in the rightward direction and robot R_2 is using the bike and moving in the leftward direction. Consider a case where robot R_2 who is using the bike is moving with unit speed to search for the exit instead of using its maximum speed. Assuming the exit is at distance d away from the origin, the evacuation time can be calculated as follows:

- Case 1: Robot R_1 reaches the exit first

  Robot R_1 needs time d to reach the exit, and robot R_2 will be at distance d on the other side of the origin. R_1 informs R_2 about the location of the exit and R_2 proceeds to the exit with maximum speed v. The evacuation time will be as follows:

$$T_1 = d + \frac{d}{v} + \frac{d}{v} = d + \frac{2d}{v}$$

- Case 2: Robot R_2 reaches the exit first

  Robot R_2 needs time d to reach the exit. Then in this case R_1 will be at distance d on the other side of the origin. R_2 will inform R_1 about the location of the exit and will move to drop off the bike at distance x away from the exit. The distance x will be determined in a way that when R_2 drops off the bike, R_1 will pick it up and will arrive at the same time with R_2. Based on that, x will be calculated as follows:

$$x + \frac{x}{v} = 2d - x + \frac{x}{v} \implies x = d$$

  Substituting x to calculate the evacuation time T_2 gives:

$$T_2 = d + x + \frac{x}{v} = d + d + \frac{d}{v} = 2d + \frac{d}{v}$$

Therefore $T = \max\{T_1, T_2\} = \max\left\{d + \frac{2d}{v}, 2d + \frac{d}{v}\right\}$.

It is surprising to note that although R_2 used unit speed to search for the exit, the resultant evacuation time is as good as that of Algorithm 7. Interestingly, the example above demonstrates that it is actually more efficient for the robot with the bike to move with optimal speed u less than its maximum speed v till the exit is found by one of the robots. Considering that both robots move in opposite directions, the robot with the bike will move with speed u and the other walking robot with unit speed. The robot who finds the exit first will communicate with the other robot who will move with its maximum speed. Assuming that robot R_1 is walking and robot R_2 is using the bike, the algorithm will be as follows:

Algorithm 8: OppDirectionWithOptimalSpeed

R_1 moves left with unit speed;

R_2 moves right with speed $u = \frac{1}{4}(-v - 1 + \sqrt{v^2 + 26v + 9})$;

if R_1 *reaches the exit* **then**

  Inform R_2 about the location of the exit;

  R_2 moves toward the exit with its maximum speed v;

end

else if R_2 *reaches the exit* **then**

  Inform R_1 about the location of the exit;

  Drop-off the bike at distance $\frac{d}{2} + \frac{d}{2u}$;

  R_2 heads back toward the exit;

  R_1 reverses the direction back to the exit then picks up the bike and moves to the exit with maximum speed v;

end

Theorem 4.2.1. *The evacuation time for Algorithm 8 is upper bounded by*

$$\frac{3d + 3dv + d\sqrt{v^2 + 26v + 9}}{4v}$$

Proof. In order to find the optimal speed u that can be used by robot R_2 to search for the exit. This can be achieved by having robot R_1, who is walking, move in the rightward direction and robot R_2, who is using the bike, move in the leftward direction. There are two cases to consider:

- Case 1: Walking robot reaches the exit first.

  The time needed by R_1 to reach the exit is d. At this point R_2 will be at distance du away from the origin since it is moving with speed u. As the algorithm mentioned, R_2 will use its maximum speed v on the way back. Thus, it needs $\frac{du}{v}$ to reach the origin and it will need another $\frac{d}{v}$ to join R_1 and reach the exit. Therefore, the evacuation time will be

  $$T_1 = d + \frac{du}{v} + \frac{d}{v} = \frac{dv + du + d}{v}$$

- Case 2: Robot with the bike reaches the exit first.

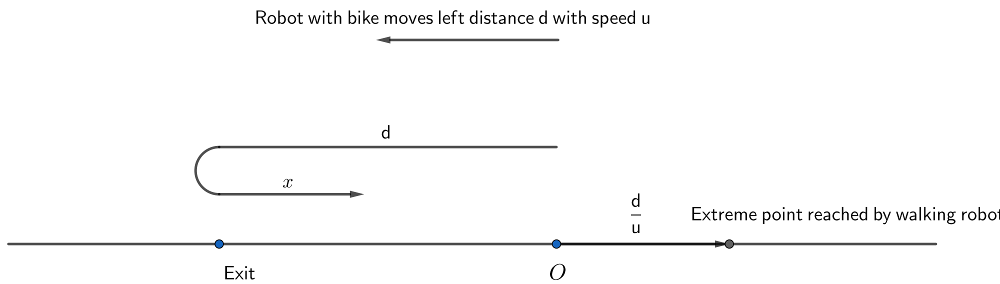

Figure 4.2: The robot with the bike will move to the left with speed u till it reaches the exit. Then it will move back with speed v to drop off the bike at distance x away from exit. The walking robot will come from point $\frac{d}{u}$ to pick up the bike and will arrive to the exit at the same time with the other robot

The time needed by R_2 to reach the exit is $\frac{d}{u}$. As soon as R_2 reaches the exit, it will inform R_1 immediately. At this point in time, R_1 will be at $\frac{d}{u}$ on the other side of the origin since it is moving with unit speed. R_2 will go back distance x to drop off the bike for R_1. The key point to find x is to have R_2 drop off the bike in a way that R_1 can pick it up and arrive at the same time as R_2. We know from Algorithm 8 that R_2 will not use its maximum speed and will move with speed u only until it reaches the exit. The only reason for not using its maximum speed before reaching the exit is to avoid having both robots farther apart from each other since this will increase the overall evacuation time. Thus when R_2 goes back to drop off the bike, it will use its maximum speed v. Based on that we have the following:

$$x + \frac{x}{v} = d + \frac{d}{u} - x + \frac{x}{v} \implies x = \frac{d}{2} + \frac{d}{2u}$$

Substituting x to calculate the evacuation time T_2 gives:

$$\begin{aligned} T_2 &= \frac{d}{u} + x + \frac{x}{v} \\ &= \frac{d}{u} + \frac{d}{2} + \frac{d}{2u} + \frac{d}{2v} + \frac{d}{2uv} \\ &= \frac{2dv + duv + dv + du + d}{2uv} \\ &= \frac{3dv + du + duv + d}{2uv} \end{aligned}$$

In order to find the best evacuation time, we need to find the best value of u which makes the maximum of T_1 and T_2 minimized given that $1 \leq u \leq v$. This means that the objective is to

$$minimize \max \left\{ \frac{dv + du + d}{v}, \frac{3dv + du + duv + d}{2uv} \right\} \tag{4.1}$$

In order to find the solution for (4.1), we will plot the graph for $T_1$and T_2 as follows:

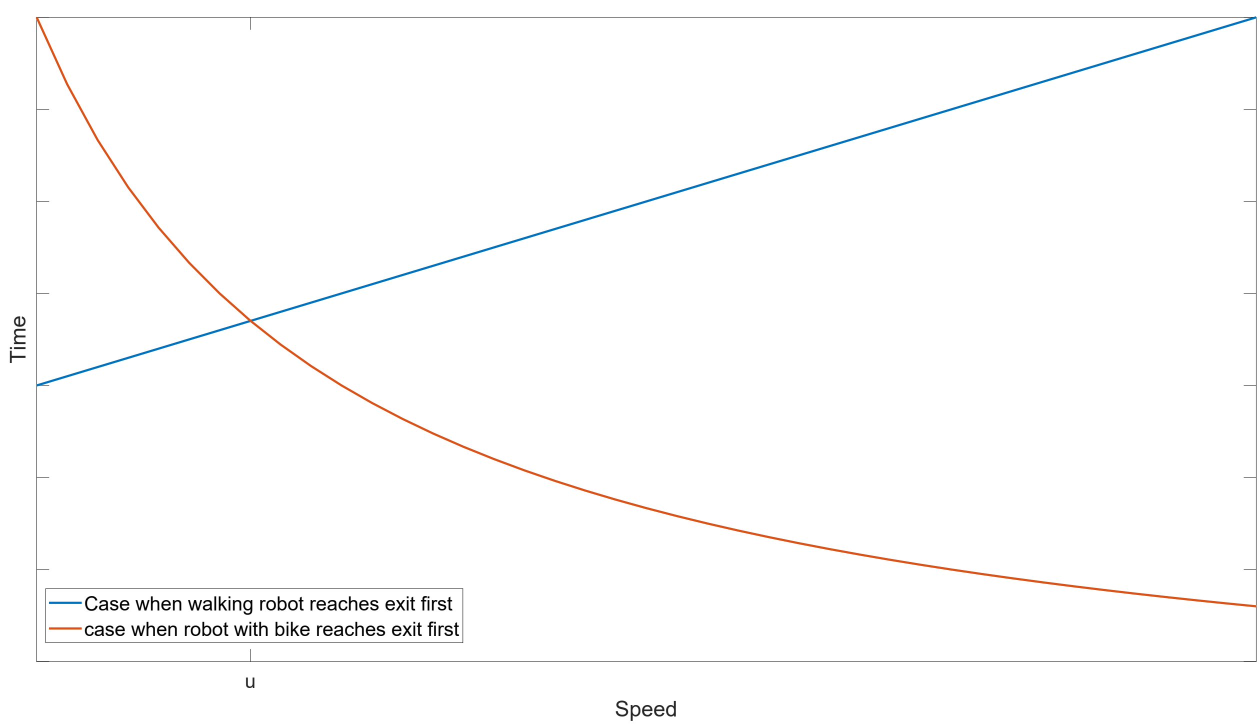

Figure 4.3: Evacuation time plots for Algorithm 8 using wi-fi model. The point of intersection of the two cases refers to the value of the speed u that makes the maximum evacuation time for the two cases above minimized.

We noticed after plotting the graph that (4.1) will be true at the point of intersection between both evacuation time plots T_1 and T_2. This will give the following:

$$\frac{u+v+1}{v} = \frac{3v+u+uv+1}{2uv} \implies 2u^2v + 2uv^2 + 2uv = 3v^2 + uv + uv^2 + v$$
$$\implies 2u^2v + (v^2+v)u - 3v^2 - v = 0$$

The equation will have two roots, choosing the positive one gives the following:

$$\begin{aligned} u &= \frac{1}{4v}(-v^2 - v + \sqrt{v^4 + 2v^3 + v^2 - 8v(-3v^2 - v)})) \\ &= \frac{1}{4v}(-v^2 - v + \sqrt{v^4 + 2v^3 + v^2 + 24v^3 + 8v^2}) \\ &= \frac{1}{4v}(-v^2 - v + \sqrt{v^4 + 26v^3 + 9v^2}) \\ &= \frac{1}{4}(-v - 1 + \sqrt{v^2 + 26v + 9}) \end{aligned}$$

In order to find the Evacuation time T, we can substitute u in T_1 or T_2. Thus if we substitute u in T_1, we get the following:

$$\begin{aligned} T &= \frac{dv + du + d}{v} \\ &= \frac{1}{v}(dv + \frac{-dv - d + d\sqrt{v^2 + 26v + 9}}{4} + d) \\ &= \frac{4dv - dv - d + d\sqrt{v^2 + 26v + 9} + 4d}{4v} \\ &= \frac{3d + 3dv + d\sqrt{v^2 + 26v + 9}}{4v} \end{aligned}$$

□

4.3 Slower Robot Imitates Faster Robot Algorithm

In this algorithm we consider that the faster robot who is using the bike is moving with its maximum speed v using a doubling strategy to search for the exit. The slower robot will eventually use a doubling strategy as well but will move with unit speed and will try to stay as close as possible to the faster robot. This can be achieved by having the slower robot move $\frac{2^k}{v}$ during the k^{th} iteration, since moving any further will cause the slower robot to be farther away from the faster robot during the $(k + 1)$ iteration. Assuming that robot R_1 is using the bike and robot R_2 is walking with unit speed, the algorithm will be as follows:

Algorithm 9: SlowerImitateFaster

---

for $k \leftarrow 1$ **to** ∞ **do**

  if *k is odd(resp.even)* **then**

    R_1 moves right (resp. left) a distance 2^k unless the exit is found;

    R_2 moves right (resp. left) a distance $\frac{2^k}{v}$;

    if *exit is found by* R_1 **then**

      Communicate with R_2;

      R_1 moves back $\frac{d}{2} - \frac{d}{2v}$ to leave the bike for R_2 then returns to exit;

      R_2 continues toward the exit after picking up the bike left by R_1;

      Quit;

    end

    R_1 turns; then moves left (resp. right), return to origin;

    R_2 turns; then moves left (resp. right), return to origin;

  end

end

Theorem 4.3.1. *The evacuation time for Algorithm 9 using wi-fi model is upper bounded by*

$$\frac{9d}{v} + \frac{d}{2} - \frac{d}{2v^2}$$

Proof. In this algorithm the robot with the bike uses a doubling strategy with maximum speed v. The walking robot will follow the faster robot but will move $\frac{2^k}{v}$ in each iteration instead of 2^k. The robot with the bike will reach the exit first then will communicate with the walking robot to proceed to the exit. The robot with the bike will go back distance $\frac{d}{2} - \frac{d}{2v}$ to drop off the bike so that the walking robot can pick it up on its way to the exit. We will justify why robot R_1 needs to move $\frac{d}{2} - \frac{d}{2v}$ after reaching the exit to leave the bike for robot R_2.

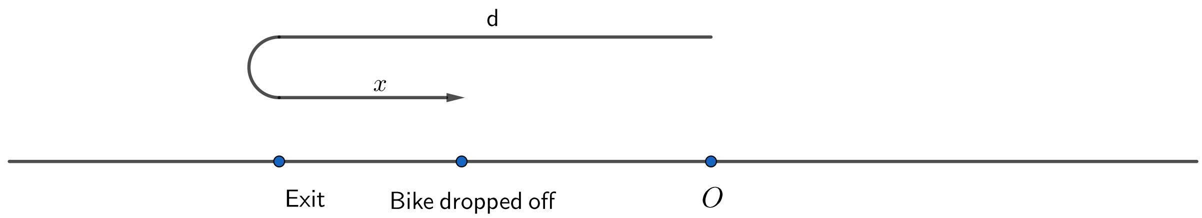

Figure 4.4: During the last iteration both robots will be at O. When the robot with the bike reaches the exit which is situated at distance d, it will go back distance x to drop off the bike so as the walking robot can pick it up and reach the exit at the same time with the other robot

After robot R_1 reaches the exit, there is no benefit of the robot staying at the exit with the bike since the other robot R_2 who is moving with unit speed can benefit from the bike to reach the exit faster. The key to find the distance x which is the distance between the exit and the point where the bike is dropped off is to have robot R_1, who is using the bike, drop it off at a point such that when it goes back to the exit it will reach the exit at the same time as robot R_2, who is walking. If we consider that d is the distance from the origin to the exit and x is the distance from the exit to the point where robot R_1 drops off the bike, then based on the figure above we have $d - x + \frac{x}{v} = \frac{d}{v} + \frac{x}{v} + x$ which leads to $x = \frac{d}{2} - \frac{d}{2v}$. This will guarantee that when the robot who is using the bike drops it off at distance x, then it will reach the exit at the same time as the walking robot. Hence we guarantee that the bike is not kept unnecessarily with the robot who reaches the exit first.

Assume that the exit is found during the k^{th} iteration, then $2^{k-2} < d \leq 2^k$. We can calculate the evacuation time as follows:

$$
\begin{aligned}
T &= \frac{2 \cdot 2^0}{v} + \frac{2 \cdot 2^1}{v} + \cdots + \frac{2 \cdot 2^{k-1}}{v} + d - x + \frac{x}{v} \\
&= \frac{2(2^k - 1)}{v} + d - x + \frac{x}{v} \\
&= \frac{2^{k+1}}{v} - \frac{2}{v} + d - \frac{d}{2} + \frac{d}{2v} + \frac{d}{2v} - \frac{d}{2v^2} \\
&= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{2} + \frac{d}{v} - \frac{d}{2v^2} \\
&\leq 2^3 . \frac{2^{k-2}}{v} - \frac{2}{v} + \frac{d}{2} + \frac{d}{v} - \frac{d}{2v^2} \\
&\leq \frac{8d}{v} - \frac{2}{v} + \frac{d}{2} + \frac{d}{v} - \frac{d}{2v^2} \\
&\leq \frac{9d}{v} + \frac{d}{2} - \frac{d}{2v^2} - \frac{2}{v} \\
&\leq \frac{9d}{v} + \frac{d}{2} - \frac{d}{2v^2}
\end{aligned}
$$

□

4.4 Comparison of the three algorithms

Each one of the above algorithms works better depending on a certain speed v to be determined. In order to find for what value of v, Algorithm 9 performs better than Algorithm 7, we will consider the evacuation time for Algorithm 9 less than evacuation time for Algorithm 7, taking into consideration that any of the two algorithms should work better in all the cases where $2^{k-2} < d \leq 2^k$. Based on what was mentioned, we have the following:

$$
\begin{aligned}
&\frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{2} + \frac{d}{v} - \frac{d}{2v^2} < 2d + \frac{d}{v} \\
&\implies \frac{2^{k+1}}{v} - \frac{2}{v} - \frac{d}{2v^2} < \frac{3}{2} d
\end{aligned}
$$

Multiplying the above equation by $2v^2$ leads to: $3dv^2 + (4 - 2^{k+2})v + d > 0$, which should be true for all $2^{k-2} < d \leq 2^k$.

After further calculation, we can notice that the equation is true if $5.3 \leq v$, which means that

Algorithm 9 is better than Algorithm 7 if $v \geq 5.3$.

Now it is time to find for what value of v, Algorithm 9 performs better than Algorithm8. In order to achieve this, we have the following:

$$\frac{9d}{v} + \frac{d}{2} - \frac{d}{2v^2} - \frac{2}{v} \leq \frac{3d + 3dv + d\sqrt{v^2 + 26v + 9}}{4v}$$

If $v = 11.75$, then the equation will be approximately true. Therefore, we conclude that Algorithm 9 performs better than Algorithm 8 if $v \geq 11.75$. This is expected since Algorithm 8 is better than Algorithm 7 for any value of v.

Example 4.4.1. we can notice that for high speed, the evacuation time for Algorithm 8 converges to d while that for Algorithm 9 converges to $\frac{d}{2}$. This can be noticed by taking the following samples:

- If $v = 15$

  Evacuation time for Algorithm 8=1.216 while that for Algorithm 9=1.0977.

- If $v = 30$

  Evacuation time for Algorithm 8=1.117 while that for Algorithm 9=0.7994.

- If $v = 100$

  Evacuation time for Algorithm 8=1.03822 while that for Algorithm 9=0.5899.

We will show how the three algorithms perform in the below graph.

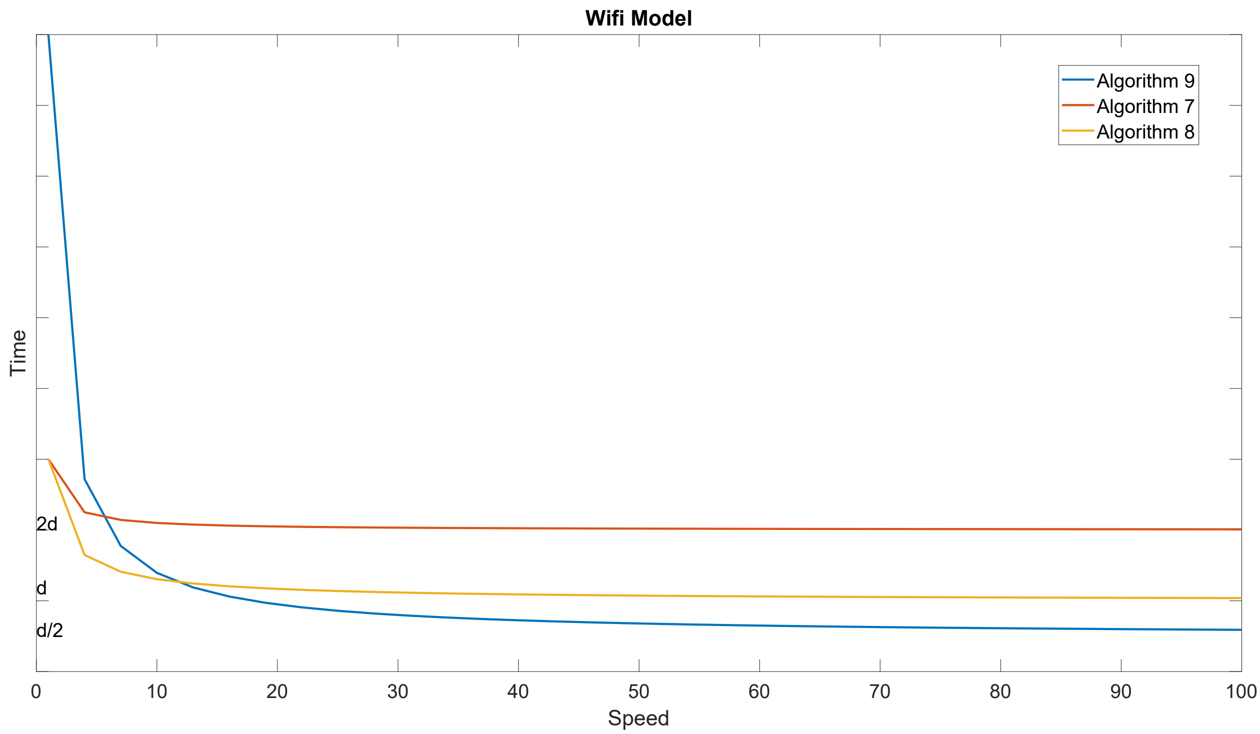

Figure 4.5: Graph for the three algorithms using the wi-fi model. On high speed, the evacuation time for Algorithm 9 converges to $\frac{d}{2}$ versus $2d$ and d for Algorithm 7 and Algorithm 8 respectively

4.5 Lower bound for two robots with a bike evacuating on a line using wi-fi model

Assume that there are two robots R_1 and R_2 under which R_1 is walking with unit speed and R_2 is using the bike and moving with speed v such that $v \geq 1$. In order to find the lower bound consider two cases. In the first case we assume that the point $-d$ is visited by the robot R_2 who is using the bike first. In this case the adversary can place the exit at point d and the walking robot R_1 can be anywhere within $[-d, d]$. In the second case, we assume that the point $-d$ is visited by the walking robot R_1 first and thus the adversary can place the exit at point d. Similarly as in the previous case, the other robot R_2 can be located anywhere within the interval $[-d, d]$. We consider that t is the time needed by any of the robots to reach the point $-d$ first, assuming that they both start at the origin O. Now we can calculate the evacuation time for each of the two cases in order to find the lower bound.

- Case 1: If the robot R_2 who is using the bike reaches the point $-d$ first then we have two cases.
  - Case 1.1: The adversary places the exit at point d. Then there are two cases:
    - Case 1.1.1: Robot R_1 is between $[O, d]$

      The best case scenario will be if R_1 is at point d. Then in this case the evacuation time T_1=$t + \frac{d}{v} + \frac{d}{v} \geq \frac{d}{v} + \frac{d}{v} + \frac{d}{v} = \frac{3d}{v}$.
    - Case 1.1.2: Robot R_1 is between $[-d, O]$

      The best case scenario will be if R_1 is at point O. Then either the walking robot will arrive first if $v < 2$. Otherwise the robot with the bike will drop off the bike at distance x away from the point d such that both robots arrive at the same time. Then in this case the distance x can be calculated as follows:

      $\frac{2d-x}{v} + x = d - x + \frac{x}{v} \implies x = \frac{dv-2d}{2v-2}$

      Hence the evacuation time T_1 will be as follows:

      $T_1 \geq t + \frac{2d-x}{v} + x \geq \frac{d}{v} + \frac{2d-x}{v} + x \geq \frac{d(v+4)}{2v}$.
  - Case 1.2: If the adversary places the exit at point $-d$.

    Definitely the case that will be considered is if the robot R_1 is in the interval $[O, d]$. The best case scenario will be if R_1 is at O. In this case the robot R_2 can move distance $\frac{d}{2}$ to drop off the bike for the walking robot R_1 taking into consideration that both robots will arrive to the exit at the same time. In this case the evacuation time T_1 will be as follows:

    $T_1 = t + \frac{d}{2} + \frac{d}{2v} \geq \frac{d}{v} + \frac{d}{2} + \frac{d}{2v} \geq \frac{2d+d+dv}{2v} \geq \frac{d(v+3)}{2v}$

  We conclude that the total evacuation time T_1 for case 1 will be as follows:

  $T_1 = \max\{\frac{3d}{v}, \frac{d(v+4)}{2v}, \frac{d(v+3)}{2v}\}$

- Case 2: Walking robot R_1 reaches $-d$ first

- Case 2.1: If the robot R_2 is between $[O, d]$ and the adversary places the exit at d.

  Then the best scenario will be if R_2 is at O. In this case R_2 will drop off the bike to the left of O at distance x such that both robots arrive to the exit at the same time.

  $\implies \frac{x}{v} + x + d = d - x + \frac{x}{v} + \frac{d}{v} \implies 2x = \frac{d}{v} \implies x = \frac{d}{2v}$

  $\implies T_2 = t + d - x + \frac{d}{v} + \frac{x}{v} \geq d + d - \frac{d}{2v} + \frac{d}{v} + \frac{d}{2v^2} \geq 2d + \frac{d}{2v} + \frac{d}{2v^2}$

- Case 2.2: If the exit is at $-d$ and R_2 is between $[0, d]$.

  then the best case scenario will be if R_2 is at O. then evacuation time T will be

  $T_2 = t + \frac{d}{v} \geq d + \frac{d}{v}$

We conclude that the total evacuation time T_2 for case 2 will be as follows:

$T_2 = \max\{2d + \frac{d}{2v} + \frac{d}{2v^2}, d + \frac{d}{v}\}$

Hence we conclude that $T = \max\{T_1, T_2\} = \max\{\frac{3d}{v}, \frac{d(v+4)}{2v}, \frac{d(v+3)}{2v}, 2d + \frac{d}{2v} + \frac{d}{2v^2}, d + \frac{d}{v}\}$

Chapter 5

Evacuation using face-to-face Model

In this model we have two robots R_1 and R_2. Both robots have unit speed when they walk. Robot R_1 is using a bike and can move up to a maximum speed $v \geq 1$, while robot R_2 is walking with unit speed. Both robots can communicate only face-to-face (i.e. at a specific point on a line). The objective is to find the optimal evacuation time achieved on a line using face-to-face model. We will explain below three different algorithms that guarantee the correctness of the evacuation problem. The last algorithm is divided into two parts, one with a bike and the other without a bike, and has been shown to be the optimal one.

5.1 Slower Robot Pursue Faster Robot Algorithm

In this algorithm, we assume that the slower robot will follow the faster robot. Since the faster robot is using doubling strategy, then during any iteration, let us say the k^{th} one, the faster robot will reverse the direction after reaching 2^k and will meet the slower robot at some point X_k. At the meeting point the slower robot will reverse its direction. We notice from this that the slower robot is following a deterministic strategy specified through a sequence $\{X_1, X_2 \ldots X_k\}$ where each X_k represents the meeting point for the two robots during the k^{th} iteration. In other words, the faster robot will follow a doubling strategy with factor 2^k while the slower robot will follow the sequence specified above. When the faster robot reaches the exit, it will go back distance x, which will be determined later, to drop off the bike and then will continue back toward the exit. Given that $a = \frac{1-v}{1+v}$, $b = \frac{1}{1+v}$ and considering that robot R_1 is using the bike and robot R_2 is walking with unit speed, we can write down the algorithm as follows:

Algorithm 10: SlowerPursueFaster

---

for $k \leftarrow 1$ **to** ∞ **do**
  if *k is odd(resp.even)* **then**
    R_1 moves right (resp. left) a distance 2^k unless the exit is found;
    R_2 moves right (resp. left) a distance $\frac{2b(2^k - a^k)}{2-a}$;
    if *The exit is found by R_1* **then**
      Move back distance x to drop off the bike for R_2 then switch direction toward the exit ;
      R_2 picks up bike and moves toward the exit;
      Quit;
    end
    R_1 turns; then moves left (resp. right), return to origin;
    R_2 turns; then moves left (resp. right), return to origin;
  end
end

Theorem 5.1.1. *The evacuation time for Algorithm 10 using face-to-face model is upper bounded by*

$$\frac{9d}{v} + d - \frac{5d}{8v^2}$$

Proof. In order to find the sequence $\{X_1, X_2 \ldots X_k\}$, we did the following:

During the 1^{st} iteration, in order to calculate X_1, we know that the faster robot will move 2^0 to reach the peak point and then will come back $2^0 - X_1$ with speed v to reach point X_1 while the slower robot will move X_1 with unit speed during the same time. Given that $a = \frac{1-v}{1+v}$ and $b = \frac{1}{1+v}$, we have the following:

$$X_1 = \frac{2^0}{v} + \frac{2^0}{v} - \frac{X_1}{v} \implies X_1 = \frac{2}{1+v} = 2 \cdot b$$

During the 2^{nd} iteration we have the following:

$$X_1 + X_2 = \frac{1}{v}(X_1 + 2 + 2 - X_2)$$
$$\implies X_2 = \frac{4}{1+v} + \frac{1-v}{1+v} \cdot X_1 = a \cdot X_1 + 2^2 \cdot b$$

Since $X_2 = a \cdot X_1 + 2^2 \cdot b$, then for the k^{th} iteration we have:

$$X_k = a \cdot X_{k-1} + 2^k \cdot b$$

Replacing $X_{k-1} = a \cdot X_{k-2} + b \cdot 2^{k-1}$ in the above equation gives

$$X_k = a(a \cdot X_{k-2} + b \cdot 2^{k-1}) + 2^k \cdot b$$

Similarly replacing $X_{k-2} = a \cdot X_{k-3} + b \cdot 2^{k-2}$ gives

$$X_k = a^3 \cdot X_{k-3} + a^2 \cdot 2^{k-2} \cdot b + a \cdot b \cdot 2^{k-1} + 2^k \cdot b$$

Following the same calculation down till X_1 leads to the following:

$$X_k = b \cdot 2^k((\frac{a}{2})^0 + \cdots + (\frac{a}{2})^{k-1}) = \frac{b \cdot 2^k(1 - (\frac{a}{2})^k)}{1 - \frac{a}{2}} = \frac{2 \cdot b(2^k - a^k)}{2^k(2-a)} \cdot 2^k \cdot b = \frac{2 \cdot b(2^k - a^k)}{2-a}$$

Assuming that the exit is found during the k^{th} iteration. Before writing down the evacuation time, let us find the distance x away from the exit, where the bike will be dropped off by the faster robot. We know that both robots will meet at each entry of the sequence and eventually they will meet at X_{k-1}.

- Define T_1 to be the time needed by the faster robot to go to the exit from the point of intersection between the two robots at X_{k-1} then to return distance x to drop off the bike then go back to the exit. Thus T_1 can be defined as follows:

$T_1 = \frac{1}{v}(X_{k-1} + x + d) + x$

- Define T_2 to be the time needed by the slower robot to go from the point of intersection between the two robots at X_{k-1} to the exit while picking up the bike on its way. Thus T_2 can be defined as follows:

  $T_2 = X_{k-1} + d - x + \frac{x}{v}$

The best thing that the faster robot can do is to drop off the bike and arrive at the same time to the exit with the slower robot who will pick up the bike on its way. This can be achieved by having:

$$\begin{aligned} & T_1 = T_2 \\ \implies & \frac{1}{v}(X_{k-1} + x + d) + x = X_{k-1} + d - x + \frac{x}{v} \\ \implies & x = \frac{1}{2}(X_{k-1} + d) - \frac{1}{2v}(d + X_{k-1}) \end{aligned}$$

The evacuation time T can be written as follows:

$$T = \frac{1}{v}(2 \cdot 2^0 + 2 \cdot 2^1 + \cdots + 2 \cdot 2^{k-1}) + \frac{d}{v} + \frac{x}{v} + x$$

Replacing the value of x which was calculated above gives:

$$\begin{aligned} T &= \frac{2}{v}(2^k - 1) + \frac{d}{v} + \frac{d}{2v} - \frac{d}{2v^2} + \frac{d}{2} - \frac{d}{2v} + \frac{1}{2v}X_{k-1} - \frac{1}{2v}X_{k-1} + \frac{1}{2}X_{k-1} - \frac{1}{2v^2}X_{k-1} \\ &= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} + \frac{d}{2} - \frac{d}{2v^2} + (\frac{1}{2} - \frac{1}{2v^2})X_{k-1} \end{aligned}$$

Back to $X_{k-1} = \frac{2b(2^{k-1} - a^{k-1})}{2-a}$ where $a = \frac{1-v}{1+v}$ and $b = \frac{1}{1+v}$

It is obvious that $-1 < a < 0$ and $0 < b \leq \frac{1}{2} \implies X_{k-1} \leq \frac{1}{2}(2^{k-1} + 1) = 2^{k-2} + \frac{1}{2}$

Knowing that $2^{k-2} < d \leq 2^k$, the evacuation time T can be simplified as follows:

$$\begin{aligned}
\implies T &\leq \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} + \frac{d}{2} - \frac{d}{2v^2} + (\frac{1}{2} - \frac{1}{2v^2})(2^{k-2} + \frac{1}{2}) \\
&\leq \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} + \frac{d}{2} - \frac{d}{2v^2} + 2^{k-3} + \frac{1}{4} - \frac{2^{k-2}}{2v^2} - \frac{1}{4v^2} \\
&\leq \frac{8d}{v} - \frac{2}{v} + \frac{d}{v} + \frac{d}{2} - \frac{d}{2v^2} + \frac{d}{2} + \frac{1}{4} - \frac{d}{8v^2} - \frac{1}{4v^2} \\
&\leq \frac{9d}{v} + d - \frac{5d}{8v^2} - \frac{1}{4v^2} + \frac{1}{4} - \frac{2}{v} \\
&\leq \frac{9d}{v} + d - \frac{5d}{8v^2}
\end{aligned}$$

□

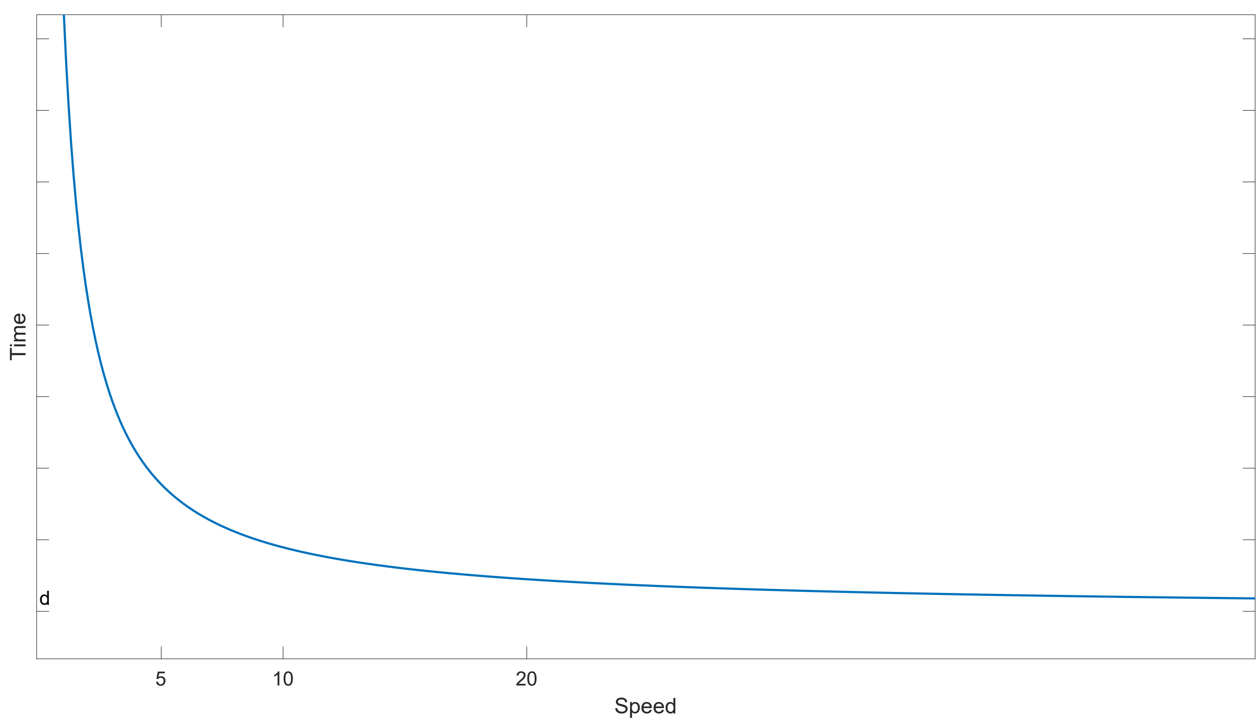

Figure 5.1: Graph for Algorithm 10 using face-to-face model. On high speed the evacuation time converges to d

5.2 Slower Robot Evacuate Close To Exit Without Aid Algorithm

In this algorithm, the faster robot with the bike uses a doubling strategy with its maximum speed v, while the slower robot will try to be as close as possible to the faster robot. In order to achieve that, the slower robot will use a doubling strategy as well but will use its own factor. The factor will be determined based on the fact the both robots should meet at a specific point during each iteration. These meeting points will form a sequence under which its element during each iteration k is proven to be $\frac{2^{k+2}}{3v+1}$. During the last iteration, when the faster robot finds the exit, the slower robot will eventually reach the meeting point and will not find the faster robot there. This will let it know that it should keep going toward the exit. Assuming that robot R_1 is using the bike and robot R_2 is walking with unit speed, the algorithm will be as follows:

Algorithm 11: SlowerEvacuateCloseToExitWithoutAid

```
for k ← 1 to ∞ do
    if k is odd(resp.even) then
        R_1 moves right (resp. left) a distance 2^k unless the exit is found;
        R_2 moves right (resp. left) a distance 2^{k+2}/(3v+1);
        if k=1 then
            R_2 waits for R_1 ;
        end
        if exit is found by R_1 then
            Move back distance d/2 − d/(2v) + 2^k(v−1)/(v(3v+1)) to drop off the bike for R_2 then switch
             direction toward the exit;
            R_2 picks up bike and moves toward exit;
            Quit;
        end
        R_1 turns; then moves left (resp. right), return to origin;
        R_2 turns; then moves left (resp. right), return to origin;
    end
end
```

Theorem 5.2.1. *The evacuation time for Algorithm 11 using face-to-face model is upper bounded by*

$$\frac{3dv^3 + 63dv^2 + 15dv - 9d}{2v^2(3v+1)}$$

Proof. The faster robot is using a doubling strategy and is moving 2^k during each iteration k. The slower robot will use a doubling strategy as well and it will follow the faster robot. In order to keep the slower robot as close as possible to the faster one, we must find the sequence that the slower robot should follow. We assume that both robots meet at a certain point X_k and that they are willing to meet at point X_{k+1} at the same time without waiting for one another, taking into consideration that $X_{k+1} = 2X_k$. The sequence can be calculated as follows:

$$\begin{aligned} X_k + X_{k+1} &= \frac{1}{v}(X_k + 2^{k+1} + 2^{k+1} - X_{k+1}) \\ \implies X_k + 2X_k &= \frac{1}{v}(X_k + 2^{k+1} + 2^{k+1} - 2X_k) \\ \implies \frac{3v+1}{v}X_k &= \frac{2^{k+2}}{v} \\ \implies X_k &= \frac{2^{k+2}}{3v+1} \end{aligned}$$

We have the sequence $\{X_0, X_1 \dots X_k\}$ given that $X_k = \frac{2^{k+2}}{3v+1}$ where $k \geq 1$. Each of the two robots will use its own doubling strategy. During each iteration, they will meet on both sides at specific points which are elements of the above sequence. During the k^{th} iteration, when the faster robot reaches the exit, it will move back distance x to drop off the bike such that the slower robot can pick it up and reach the exit at the same time as itself. The distance x can be calculated as follows:

$$\begin{aligned} &\frac{d}{v} + \frac{1}{v} \cdot X_{k-1} + \frac{x}{v} + x = d + X_{k-1} - x + \frac{x}{v} \\ &\frac{d}{v} + \frac{2^{k+1}}{v(3v+1)} + \frac{x}{v} + x = d + \frac{2^{k+1}}{3v+1} - x + \frac{x}{v} \\ &\implies x = \frac{d}{2} - \frac{d}{2v} + \frac{2^k(v-1)}{v(3v+1)} \end{aligned}$$

Assuming that $2^{k-2} < d \leq 2^k$ and replacing x which was calculated above, the evacuation time T can be stated as follows:

$$
\begin{aligned}
T &= \frac{1}{v}(2 \cdot 2^0 + 2 \cdot 2^1 + \cdots + 2 \cdot 2^{k-1}) + \frac{d}{v} + \frac{x}{v} + x \\
&= \frac{2}{v}(2^k - 1) + \frac{d}{v} + \frac{x}{v} + x \\
&= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} + \frac{x}{v} + x \\
&= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} + \frac{d}{2v} - \frac{d}{2v^2} + \frac{2^k(v-1)}{v^2(3v+1)} + \frac{d}{2} - \frac{d}{2v} + \frac{2^k(v-1)}{v(3v+1)} \\
&\leq \frac{16d}{2v} - \frac{2}{v} + \frac{d}{v} + \frac{d}{2} - \frac{d}{2v^2} + \frac{4d(v-1)}{v^2(3v+1)} + \frac{4d(v-1)}{v(3v+1)} \\
&\leq \frac{18d}{2v} + \frac{d}{2} - \frac{d}{2v^2} + \frac{4d(v-1)}{v^2(3v+1)} + \frac{4d(v-1)}{v(3v+1)} - \frac{2}{v} \\
&\leq \frac{54dv^2 + 18dv + 3dv^3 + dv^2 - 3dv - d + 8vd - 8d + 8dv^2 - 8dv}{2v^2(3v+1)} - \frac{2}{v} \\
&\leq \frac{3dv^3 + 63dv^2 + 15dv - 9d}{2v^2(3v+1)} - \frac{2}{v} \\
&\leq \frac{3dv^3 + 63dv^2 + 15dv - 9d}{2v^2(3v+1)}
\end{aligned}
$$

□

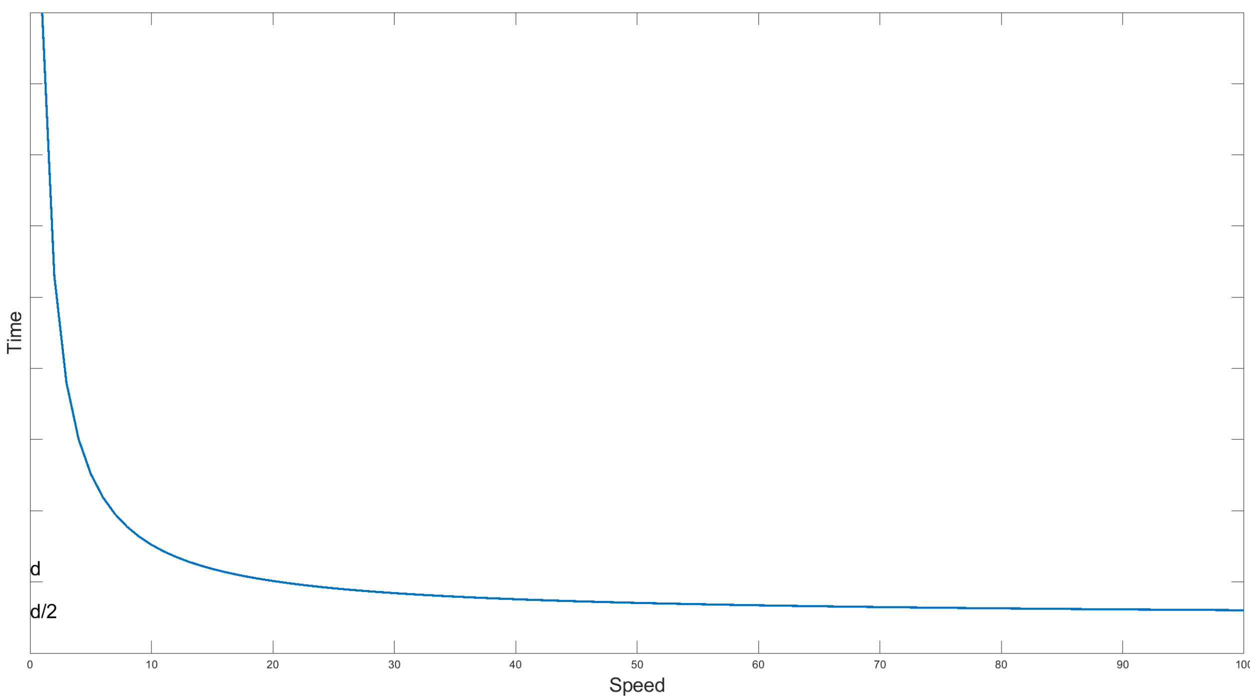

Figure 5.2: Graph for Algorithm 11 using face-to-face model. On high speed the evacuation time converges to $\frac{d}{2}$

5.3 Nearest Meeting To Exit Algorithm

In order to reduce the evacuation time, it is more suitable for the faster robot with the bike to search for the exit while the slower robot follows a doubling strategy that will keep it as close as possible to the faster robot and will expedite its travel to the exit during the last iteration. In order to achieve that, the purpose will be to find this deterministic doubling strategy that the slower robot should follow.

In this section we will explain the optimal algorithm. The evacuation process is divided into two parts. The first part will be to find the exit and the second part will be to inform the other robot to proceed to the exit. It makes more sense for the robot who is using the bike to search for the exit, as it would be less optimal for the robot to drop off the bike before finding the exit. After finding the exit, the faster robot will pursue the slower robot to inform it about the location of the exit and will drop off the bike on its way back so that the slower robot can proceed to the exit faster. The best way to search for the exit is by using a doubling strategy with maximum speed v. If we assume that the faster robot with the bike is using doubling strategy, then it makes sense for the other walking robot to be as close as possible to the faster robot. This can be achieved by finding the best meeting point with the faster robot during each iteration which enables the slower robot to be as close as possible to the exit in the last iteration. Let us consider the following sequence:$X = \{X_1, X_2, X_3, \ldots, X_k\}$, where $X_k = rX_{k-1}$. The purpose is to find the best value of r which is the factor related to the doubling strategy that the slower robot follows. Definitely the best meeting point would be the peak point reached by the slower robot during each iteration, since it will be the closest to the exit. Assume that both robots meet at some point X_{k-1} during the $k-1$ iteration and they are willing to meet during the k^{th} iteration without waiting for one another, then we have the following:

$$X_{k-1} + X_k = \frac{1}{v} \cdot (2^{k-1} - X_{k-1}) + \frac{2^{k-1}}{v} + \frac{X_k}{v}$$
$$X_{k-1} + rX_{k-1} = \frac{1}{v} \cdot (2^{k-1} - X_{k-1}) + \frac{2^{k-1}}{v} + \frac{r}{v} \cdot X_{k-1}$$
$$\implies (r+1) \cdot X_{k-1} = \frac{2^k}{v} + \frac{r-1}{v} \cdot X_{k-1}$$
$$\implies rvX_{k-1} + vX_{k-1} - rX_{k-1} + X_{k-1} = 2^k$$
$$\implies X_{k-1} = \frac{2^k}{rv + v - r + 1}$$

Similarly we have $X_k = \frac{2^{k+1}}{rv+v-r+1}$. Consider $X_k = rX_{k-1}$, then we can deduce that $r = 2$. Substituting $r = 2$ gives $X_k = \frac{2^{k+1}}{3v-1}$. So we conclude that the slower robot will use the doubling strategy and will follow the sequence $X = \{\frac{4}{3v-1}, \frac{8}{3v-1}, \ldots, \frac{2^{k+1}}{3v-1}\}$.

After finding the sequence that each robot will follow, we would like to simplify the problem by studying it on two robots without a bike first, and then we will switch to the two robots with a bike model.

5.3.1 Evacuation of two robots without a bike

Before creating the algorithm for the two robots with a bike using face-to-face model, we will modify the problem by removing the bike from the model and considering two robots, R_1 and R_2, seeking to evacuate through an exit situated on a line, where the first robot has speed $v \geq 1$ and the second robot has unit speed. What differentiates this problem from the one with the bike is that the robot riding the bike needs to drop off the bike after reaching the exit such that the slower robot can pick it up and reach the exit at the same time with the faster robot. We can notice that the upper bound of Algorithm 12 is optimal even on specific values of v. If we consider $v = 1$, it leads to competitive ratio 9 which is proven to be the best competitive ratio that is achieved for two robots moving with unit speed. If we consider $v = 3$ it leads to competitive ratio 3 which is the best competitive ratio since having a scenario of two robots one with unit speed and the other with speed 3 achieves the same competitive ratio as if one robot with speed 3 is searching for the exit which is definitely a surprising result. Assuming that R_1 moves with speed v and R_2 with unit speed, we can write down the algorithm as follows:

Algorithm 12: Algorithm for two robots evacuating using face to face model without a bike

---

for $k \leftarrow 1$ **to** ∞ **do**
- **if** k *is odd(resp.even)* **then**
  - R_1 moves right (resp. left) a distance 2^k unless the exit is found;
  - R_2 moves right (resp. left) a distance $\frac{2^{k+1}}{3v-1}$;
  - **if** *k=1* **then**
    - R_2 waits for R_1 ;
  - **if** *exit is found by* R_1 **then**
    - R_1 switches direction to inform R_2;
    - R_2 continues to exit with R_1;
    - Quit;
  - **if** *exit is found by* R_2 **then**
    - R_2 waits till R_1 comes to exit;
    - Break;
  - R_1 turns; then moves left (resp. right), return to origin;
  - R_2 turns; then moves left (resp. right), return to origin;

Theorem 5.3.1. *The evacuation time for Algorithm 12 is upper bounded by*

$$-\frac{6d(v-3)^2}{(v-1)(3v-1)} + 3d \;\; if \; v > 3$$
$$-\frac{6d(v-3)}{(3v-1)} + 3d \;\; if \; v \leq 3$$

Proof. We are studying this algorithm for all $v \geq 1$. Consider that the exit is found during the k^{th} iteration, then this leads to the fact that $2^{k-2} < d \leq 2^k$. Both robots R_1 and R_2 will meet at a known point, which is at $\frac{2^{k+1}}{3v-1}$. In some cases, depending on the value of v, the slower robot may reach the exit before the faster robot and this happens when the exit is between 2^{k-2} and the meeting point during the k^{th} iteration which may occur for some values of v. There are two cases that should be taken into consideration:

- Case 1: $2^{k-2} < d \leq \frac{2^{k+1}}{3v-1} \leq 2^k$.

  The evacuation time T will be in this case as follows:

$$
\begin{aligned}
T &= \frac{1}{v}(2 \cdot 2^0 + 2 \cdot 2^1 + \cdots + 2 \cdot 2^{k-1}) + \frac{d}{v} \\
&= \frac{1}{v} \cdot 2(2^k - 1) + \frac{d}{v} \\
&= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} \\
&\leq \frac{8d}{v} + \frac{d}{v} - \frac{2}{v} = \frac{9d}{v} - \frac{2}{v}
\end{aligned}
$$

- Case 2: Either $\frac{2^{k+1}}{3v-1} < 2^{k-2} < d \leq 2^k$ or $2^{k-2} < \frac{2^{k+1}}{3v-1} < d \leq 2^k$.

  Assume $d = \frac{2^{k+1}}{3v-1} + e$ where $e \geq 0$.

  R_1 who is moving with speed v will reach the exit before R_2. Since the exit is at distance e from the meeting point, then from that point, R_1 needs time $\frac{e}{v}$ to reach the exit. During this time, R_2 will be at distance e on the other side of the meeting point. Therefore, when R_1 reaches the exit, R_2 will be at distance $e + \frac{e}{v} = e\frac{v+1}{v}$ away from the exit.

  We will find now the distance x that robot R_2 moves from the point R_1 reaches the exit till the point it meets R_1.

$$
\begin{aligned}
x &= \frac{e(v+1)}{v^2} + \frac{x}{v} \\
&\implies \frac{x(v-1)}{v} = \frac{e(v+1)}{v^2} \\
&\implies x = \frac{v+1}{v-1} \cdot \frac{e}{v}
\end{aligned}
$$

  Therefore, when robot R_1 reaches robot R_2 to inform it about the exit, R_2 will be far from the exit by distance $y = \frac{v+1}{v-1} \cdot \frac{e}{v} + e\frac{v+1}{v}$. The evacuation time T can be calculated as follows:

$$\begin{aligned}
T &= \frac{1}{v}(2 \cdot 2^0 + 2 \cdot 2^1 + \cdots + 2 \cdot 2^{k-1}) + \frac{d}{v} + y + \frac{y}{v} \\
&= \frac{1}{v}(2 \cdot 2^0 + 2 \cdot 2^1 + \cdots + 2 \cdot 2^{k-1}) + \frac{d}{v} + \frac{e(v+1)^2}{v^2} + \frac{v+1}{v-1} \cdot \frac{e}{v^2} + \frac{v+1}{v-1} \cdot \frac{e}{v} \\
&= \frac{2(2^k - 1)}{v} + \frac{d}{v} + e + \frac{2e}{v} + \frac{e}{v^2} + \frac{2ve + e + v^2 e}{v^2(v-1)} \\
&= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} + \frac{2e}{v} + \frac{e}{v^2} + e + \frac{2ve + e + v^2 e}{v^2(v-1)} \\
&= 3d - \frac{d}{v} - 3e + \frac{e}{v} - \frac{2}{v} + \frac{d}{v} + \frac{2e}{v} + \frac{e}{v^2} + e + \frac{2ve + e + v^2 e}{v^2(v-1)} \\
&= 3d - \frac{2}{v} - 2e + \frac{3e}{v} + \frac{e}{v^2} + \frac{2ve + e + v^2 e}{v^2(v-1)} \\
&= 3d - \frac{2}{v} + e \cdot \frac{-2(v^3 - v^2) + 3(v^2 - v) + v - 1 + v^2 + 2v + 1}{v^2(v-1)} \\
&= e \cdot \frac{-2v^3 + 2v^2 + 3v^2 - 3v + v - 1 + v^2 + 2v + 1}{(v-1)v^2} + 3d - \frac{2}{v} \\
&= e \cdot \frac{-2v^3 + 6v^2}{(v-1)v^2} + 3d - \frac{2}{v} \\
&= e \cdot \frac{-2v + 6}{v-1} - \frac{2}{v} + 3d \\
&= -e \cdot \frac{2(v-3)}{v-1} + 3d - \frac{2}{v}
\end{aligned}$$

Substituting $e = d - \frac{2^{k+1}}{3v-1}$ gives the following:

$$\begin{aligned}
T &= \frac{2(v-3)}{v-1}(d - \frac{2^{k+1}}{3v-1}) + 3d - \frac{2}{v} \\
&= -\frac{2d(v-3)}{v-1} + 4(v-3) \cdot \frac{2^k}{(v-1)(3v-1)} + 3d - \frac{2}{v}
\end{aligned}$$

There are two cases to consider here, either $v \leq 3$ or $v > 3$

- Case1: If $v \leq 3$

  Since $v - 3$ is negative and $d \leq 2^k$ then we have the following:

$$\begin{aligned}
T &\leq -\frac{2d(v-3)}{v-1} + \frac{4d(v-3)}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq \frac{d}{(v-1)(3v-1)} \cdot ((-2v+6)(3v-1) + (4v-12)) + 3d - \frac{2}{v}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{(-6v^2+2v+18v-6+4v-12)d}{(3v-1)(v-1)} + 3d - \frac{2}{v} \\
&\leq \frac{(-6v^2+24v-18)d}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq -6d\frac{v^2-4v+3}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq -\frac{6d(v-3)(v-1)}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq -\frac{6d(v-3)}{(3v-1)} + 3d - \frac{2}{v} \\
&\leq -\frac{6d(v-3)}{(3v-1)} + 3d
\end{aligned}
$$

- Case2: If $3 \leq v$

  Since $d > 2^{k-2}$ and $v-3$ is positive then we have the following:

$$
\begin{aligned}
T &\leq -\frac{2d(v-3)}{v-1} + \frac{16(v-3)2^{k-2}}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq -\frac{2d(v-3)}{v-1} + \frac{16(v-3)d}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq \frac{-2d(3v^2-v-9v+3)+16dv-48d}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq \frac{-6dv^2+20dv-6d+16dv-48d}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq \frac{-6dv^2+36dv-54d}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq -\frac{6d(v-3)^2}{(v-1)(3v-1)} + 3d - \frac{2}{v} \\
&\leq -\frac{6d(v-3)^2}{(v-1)(3v-1)} + 3d
\end{aligned}
$$

$\square$

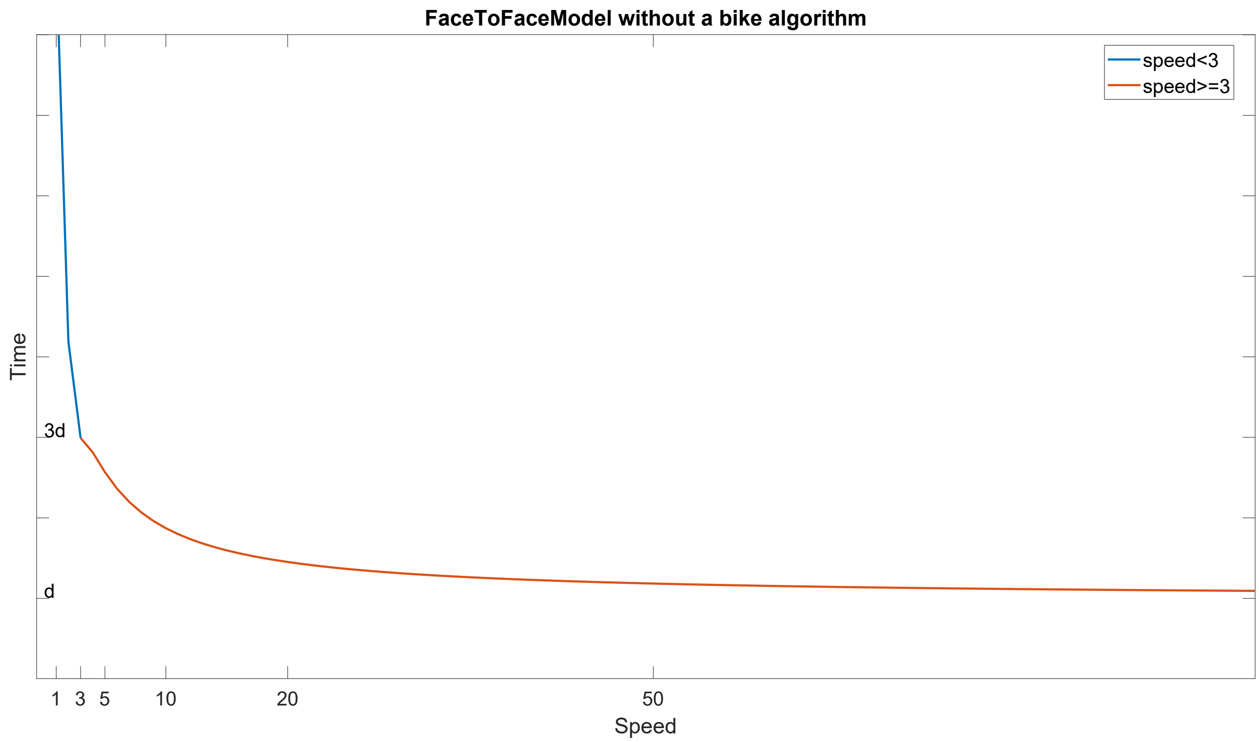

Figure 5.3: Graph for two robots without a bike using face-to-face model. There are two cases considered when the evacuation time was calculated based on whether the speed is above or below 3. The two cases are illustrated in the graph above. If the speed is 3, the evacuation time is $3d$, while if the speed is high, the evacuation time converges to $\frac{d}{2}$.

5.3.2 Evacuation of two robots with a bike

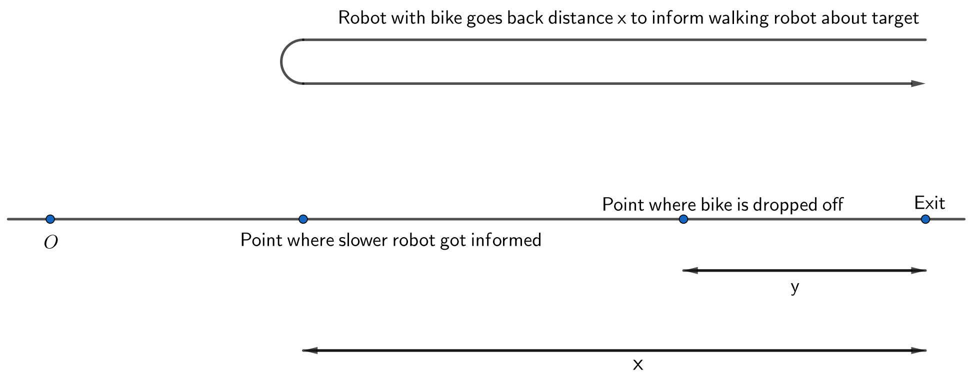

Figure 5.4: During the last iteration, the two robots meet at the extreme point reached by the walking robot. The robot with the bike proceeds to the exit and the walking robot reverses the direction toward the origin. When the robot with the bike reaches the exit, it will go back distance x to inform the walking robot then on the way back it will drop off the bike at distance y away from the exit and proceed to the exit. The walking robot will pick up the bike on its way and head to the exit.

Let us go back to the two robots' model with a bike. We know from the previous section that when R_1 reaches the exit then R_2 will be away from R_1 by $x = \frac{e(v+1)}{v} + \frac{v+1}{v-1} \cdot \frac{e}{v}$. Now it is required to find at what distance y away from the exit should robot R_1 drop off the bike so that robot R_2 can pick it up and proceed to the exit and reach it at the same time as R_1. After we find out the distance y, we will go back to create the algorithm for the two robots with a bike model. In order to find out the distance y, and based on the figure above, we have the following:

$$\frac{x}{v} - \frac{y}{v} + y = x - y + \frac{y}{v}$$
$$\implies \frac{x}{v} - \frac{y}{v} + 2y = x + \frac{y}{v}$$
$$\implies y(2v-2) = x(v-1)$$
$$\implies y = \frac{x(v-1)}{2(v-1)} = \frac{x}{2}$$

First, assume that robot R_1 is using the bike and robot R_2 is walking. Next, consider x as the distance between the exit and the last meeting point between the two robots before they reach the exit. The algorithm becomes as follows:

Algorithm 13: NearestMeetingToExit

```
for k ← 1 to ∞ do
    if k is odd(resp.even) then
        R1 moves right (resp. left) a distance 2^k unless the exit is found;
        R2 moves right (resp. left) a distance 2^(k+1)/(3v−1);
        if k = 1 then
            R2 waits for R1 ;
        if exit is found by R1 then
            R1 switches direction to inform R2;
            R1 drops off the bike at distance x/2;
            R2 picks up the bike and continues to the exit;
            Quit;
        if exit is found by R2 then
            R2 waits till R1 comes to the exit;
            Break;
        R1 turns; then moves left (resp. right), return to origin;
        R2 turns; then moves left (resp. right), return to origin;
```

Theorem 5.3.2. *The evacuation time for Algorithm 13 using face-to-face model is upper bounded by*

$$3d - \frac{5v^2 - 12v - 1}{2v(v-1)}d + \frac{5v^2 - 12v - 1}{v(v-1)(3v-1)}d \ \ if\ 1 < v \leq \frac{6+\sqrt{41}}{5}$$
$$3d - \frac{5v^2 - 12v - 1}{2v(v-1)}d + \frac{4(5v^2 - 12v - 1)}{v(v-1)(3v-1)}d \ \ if\ \frac{6+\sqrt{41}}{5} \leq v$$

Proof. The evacuation time T can be calculated as follows:

$$\begin{aligned}T &= \frac{2\cdot 2^0 + 2\cdot 2^1 + \cdots + 2\cdot 2^{k-1}}{v} + \frac{d}{v} + \frac{x}{v} + \frac{y}{v} + y \\ &= \frac{2(2^k-1)}{v} + \frac{d}{v} + \frac{x}{v} + \frac{x}{2v} + \frac{x}{2} \\ &= \frac{2^{k+1}}{v} - \frac{2}{v} + \frac{d}{v} + \frac{3}{2v}(e + \frac{e}{v} + \frac{v+1}{v-1}\cdot\frac{e}{v}) + \frac{1}{2}(e + \frac{e}{v} + \frac{v+1}{v-1}\cdot\frac{e}{v})\end{aligned}$$

Since $d = e + \frac{2^{k+1}}{3v-1}$, then replacing $\frac{2^{k+1}}{v} = 3d - 3e - \frac{d}{v} + \frac{e}{v}$ in the above equation gives the following:

$$\begin{aligned}T &= 3d - \frac{d}{v} - 3e + \frac{e}{v} - \frac{2}{v} + \frac{d}{v} + \frac{3e}{2v} + \frac{3e}{2v^2} + \frac{3e(v+1)}{2v^2(v-1)} + \frac{e}{2} + \frac{e}{2v} + \frac{e(v+1)}{2v(v-1)} \\ &= 3d - \frac{2}{v} + \frac{3e}{v} - \frac{5e}{2} + \frac{3e}{2v^2} + \frac{3ev + 3e + ev^2 + ev}{2v^2(v-1)} \\ &= 3d - \frac{2}{v} + \frac{3e}{v} - \frac{5e}{2} + \frac{3e}{2v^2} + \frac{ev^2 + 4ev + 3e}{2v^2(v-1)} \\ &= 3d - \frac{2}{v} + e\cdot\frac{6v^2 - 6v - 5v^3 + 5v^2 + 3v - 3 + v^2 + 4v + 3}{2(v-1)v^2} = \\ &= 3d - \frac{2}{v} - e\cdot\frac{5v^2 - 12v - 1}{2v(v-1)} \\ &= 3d - \frac{2}{v} - \frac{5v^2 - 12v - 1}{2v(v-1)}(d - \frac{2^{k+1}}{3v-1}) \\ &= 3d - d\cdot\frac{5v^2 - 12v - 1}{2v(v-1)} + 2^k\frac{5v^2 - 12v - 1}{v(v-1)(3v-1)} - \frac{2}{v}\end{aligned}$$

We would like to interpret the result above. There are two roots for the quadratic equation $5v^2 - 12v - 1$ which are approximately 2.4 and -0.08. If $v = 2.4$ then the evacuation time will approximate $3d$ while in the previous model which is two robots without a bike, we have seen that the evacuation time is approximating $3d$ if $v = 3$. This is what we logically expect since in the presence of the bike, it is expected that the two robots will evacuate faster. There are two cases to consider here:

- Case1: $1 < v \leq \frac{6+\sqrt{41}}{5}$

Since $5v^2 - 12v - 1 \leq 0$ and $d \leq 2^k$

$$\begin{aligned}\implies T &\leq 3d - \frac{5v^2 - 12v - 1}{2v(v-1)}d + \frac{5v^2 - 12v - 1}{v(v-1)(3v-1)}d - \frac{2}{v} \\ &\leq 3d - \frac{5v^2 - 12v - 1}{2v(v-1)}d + \frac{5v^2 - 12v - 1}{v(v-1)(3v-1)}d\end{aligned}$$

- Case2: $\frac{6+\sqrt{41}}{5} \leq v$.

  Since $0 \leq 5v^2 - 12v - 1$ and $2^{k-2} \leq d$

$$\begin{aligned}\implies T &\leq 3d - \frac{5v^2 - 12v - 1}{2v(v-1)}d + \frac{5v^2 - 12v - 1}{v(v-1)(3v-1)} \cdot 2^2 \cdot 2^{k-2} - \frac{2}{v} \\ &\leq 3d - \frac{5v^2 - 12v - 1}{2v(v-1)}d + \frac{4(5v^2 - 12v - 1)}{v(v-1)(3v-1)}d - \frac{2}{v} \\ &\leq 3d - \frac{5v^2 - 12v - 1}{2v(v-1)}d + \frac{4(5v^2 - 12v - 1)}{v(v-1)(3v-1)}d\end{aligned}$$

□

5.4 Comparison of the three algorithms

Example 5.4.1. We will take various samples in order to show how the evacuation time varies between the three algorithms

- If $v = 10$

  Evacuation time will approximately be 1.89 for Algorithm 10 versus 1.522 for Algorithm 11 and $1.4758d$ for Algorithm 13

- If $v = 100$

  Evacuation time will approximately be 1.089 for Algorithm 10 versus 0.6031 for Algorithm 11 and $0.60134d$ for Algorithm 13

We can notice that the evacuation time is approximating $\frac{d}{2}$ when the speed becomes large enough. This can be shown clearly in the graph below.

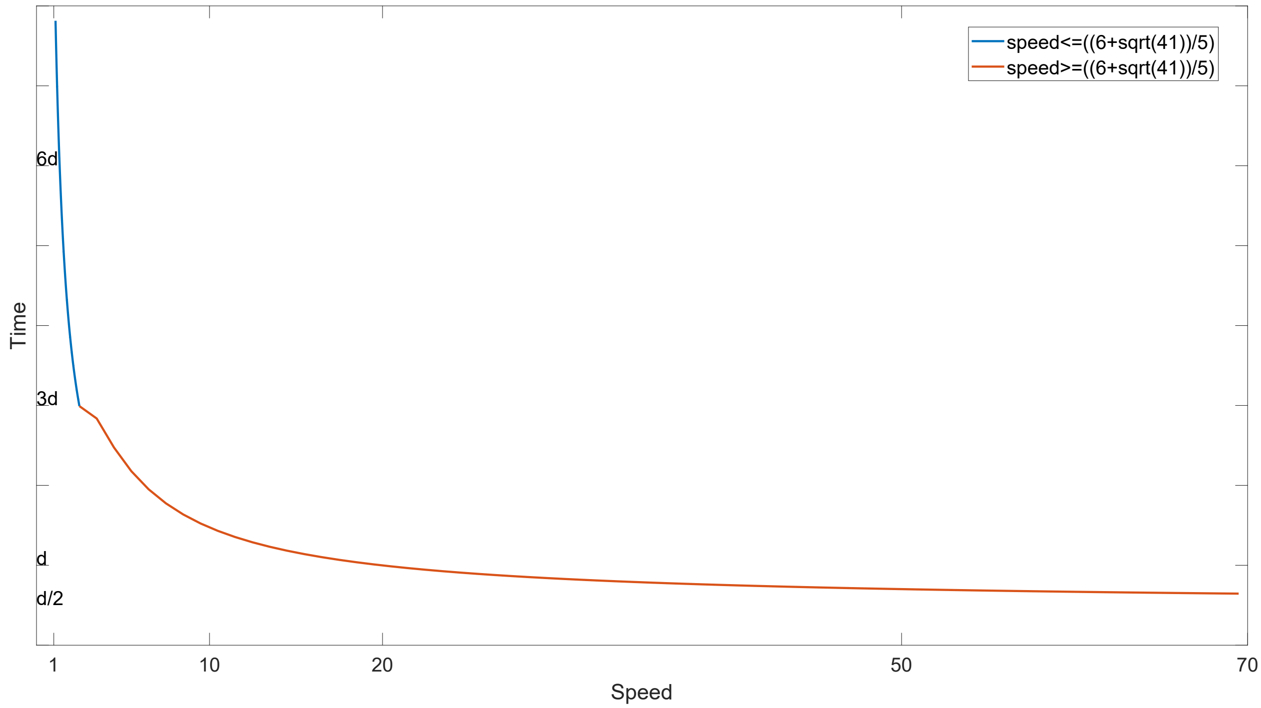

Figure 5.5: Graph for Nearest Meeting To Exit Algorithm using face-to-face model. There are two cases considered when the evacuation time was calculated based on whether the speed is above or below $\frac{6+\sqrt{41}}{5}$. The two cases are illustrated in the graph above. If the speed is $\frac{6+\sqrt{41}}{5}$, the evacuation time is $3d$, while if the speed is high, the evacuation time converges to $\frac{d}{2}$.

Chapter 6

Conclusion and Open Problems

In this thesis we considered linear search and evacuation problems using face-to-face and wi-fi models on a line for two robots including a passive entity (i.e., a bike). Regarding the linear search problem, we proposed two algorithms that are optimal based on the maximum speed. The second algorithm performs better if the speed is greater than 9. We concluded that when the speed is large enough (i.e., above 9) then the walking robot will be a dummy robot of no use since the robot with the bike will be able to find the exit on its own. This is because even if the exit is located on the side where the walking robot is moving, the faster robot will still be able to find the exit first. We concluded the linear search problem by providing a lower bound which was shown to be $\max\{\frac{3d}{v}, d + \frac{d}{v}\}$.

Regarding the evacuation problem, we proposed three different algorithms on each of the two models: wi-fi and face-to-face. Regarding the wi-fi model, there are two algorithms that are optimal based on the maximum speed, while for the face-to-face model there is one algorithm which is optimal regardless of the value of the maximum speed. Finding the optimal algorithm for the model of two robots without a bike allowed us to find the optimal algorithm for the model of two robots with a bike. Logically speaking if both robots start at the origin and they know where the exit is, then the best thing they can do is to share the bike between one another. In this case any of the two robots will walk $\frac{d}{2}$ with unit speed and will use the bike to move with maximum speed v over the other $\frac{d}{2}$ part. Thus, the best evacuation time achieved on high speed is $\frac{d}{2}$. It is not immediately intuitive that on high speed, the evacuation time for the two optimal algorithms related to the wi-fi and face-to-face models converge to $\frac{d}{2}$. It is surprising as well to note that the performance of the optimal algorithm for the face-to-face model is very close to that of the wi-fi model as the graph of both models shows.

This field of study has many unanswered questions. For example, the model for the linear search and evacuation problems with a bike can be extended to include more than two robots. In this case, it is harder to find an optimal algorithm that enables the robots to use the bike efficiently in order to decrease the linear search or evacuation time. The model can include many bikes in addition to having many robots, taking into consideration that the number of robots must exceed the number of bikes. The problem may also be studied on various domains as well such as disc, circle,square, etc. For future work on the problem of two robots with a bike, we are planning to find the optimal algorithm with tight lower bound related to the face-to-face model. The problem can also be extended to include, at most, one faulty robot.

Bibliography

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

[2] S. Alpern and S. Gal. *The theory of search games and rendezvous*. Springer, 2003.

[3] Steve Alpern and Shmuel Gal. Searching for an agent who may or may not want to be found. *Operations Research*, 50(2):311–323, 2002.

[4] Ricardo A Baezayates, Joseph C Culberson, and Gregory JE Rawlins. Searching in the plane. *Information and computation*, 106(2):234–252, 1993.

[5] Evangelos Bampas, Jurek Czyzowicz, Leszek Gąsieniec, David Ilcinkas, Ralf Klasing, Tomasz Kociumaka, and Dominik Pająk. Linear search by a pair of distinct-speed robots. *Algorithmica*, 81(1):317–342, 2019.

[6] Anatole Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964.

[7] Anatole Beck and DJ Newman. Yet more on the linear search problem. *Israel journal of mathematics*, 8(4):419–429, 1970.

[8] Anatole Beck and Peter Warren. The return of the linear search problem. *Israel Journal of Mathematics*, 14(2):169–183, 1973.

[9] R. Bellman. An optimal search. *SIAM Review*, 5(3):274–274, 1963.

[10] Prosenjit Bose and Jean-Lou De Carufel. A general framework for searching on a line. *Theoretical Computer Science*, 703:1–17, 2017.

[11] Prosenjit Bose, Jean-Lou De Carufel, and Stephane Durocher. Revisiting the problem of searching on a line. In *European Symposium on Algorithms*, pages 205–216. Springer, 2013.

[12] Sebastian Brandt, Felix Laufenberg, Yuezhou Lv, David Stolz, and Roger Wattenhofer. Collaboration without communication: Evacuating two robots from a disk. In *International Conference on Algorithms and Complexity*, pages 104–115. Springer, 2017.

[13] Marek Chrobak, Leszek Gąsieniec, Thomas Gorry, and Russell Martin. Group search on the line. In *International Conference on Current Trends in Theory and Practice of Informatics*, pages 164–176. Springer, 2015.

[14] Huda Chuangpishit, Saeed Mehrabi, Lata Narayanan, and Jaroslav Opatrny. Evacuating an equilateral triangle in the face-to-face model. In *21st International Conference on Principles of Distributed Systems (OPODIS 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[15] Jurek Czyzowicz, Stefan Dobrev, Konstantinos Georgiou, Evangelos Kranakis, and Fraser MacQuarrie. Evacuating two robots from multiple unknown exits in a circle. *Theoretical Computer Science*, 709:20–30, 2018.

[16] Jurek Czyzowicz, Leszek Gąsieniec, Thomas Gorry, Evangelos Kranakis, Russell Martin, and Dominik Pajak. Evacuating robots via unknown exit in a disk. In *International Symposium on Distributed Computing*, pages 122–136. Springer, 2014.

[17] Jurek Czyzowicz, Konstantinos Georgiou, Evangelos Kranakis, Lata Narayanan, Jaroslav Opatrny, and Birgit Vogtenhuber. Evacuating robots from a disk using face-to-face communication. In *International Conference on Algorithms and Complexity*, pages 140–152. Springer, 2015.

[18] Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, and Sunil Shende. Linear search with terrain-dependent speeds. In *International Conference on Algorithms and Complexity*, pages 430–441. Springer, 2017.

[19] Konstantinos Georgiou, George Karakostas, and Evangelos Kranakis. Search-and-fetch with 2 robots on a disk: wireless and face-to-face communication models. *arXiv preprint arXiv:1611.10208*, 2016.

[20] Khaled Jawhar and Evangelos Kranakis. Bike assisted linear search and evacuation, to appear. 2020.

[21] Ming-Yang Kao, John H Reif, and Stephen R Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1):63–79, 1996.

[22] Debasish Pattanayak, H Ramesh, Partha Sarathi Mandal, and Stefan Schmid. Evacuating two robots from two unknown exits on the perimeter of a disk with wireless communication. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, page 20. ACM, 2018.

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