

**POPULAR DEMOCRATIC REPUBLIC OF ALGERIA**  
**MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH**  
**MOHAMED BOUDIAF UNIVERSITY - M'SILA**

**Mathematics and Computer Science Faculty**  
**Computer Science Department**  
**N° : 191535110286**



**DOMAIN : Mathematics and Computer  
Science**  
**FIELD: Computer Science**  
**OPTION : Software Ingeneering**

**Thesis presented for obtaining  
Academic Master's degree**

**By: Benmerzoug Rima**

**Surface covering Robot simulator**  
  
using  
**Random walk strategy**

**Submitted to the jury:**

Dr. Adel MOUSSAOUI

M'sila University

Supervisor

Dr. Amel MELIOUH

M'sila University

Dr.Seddik BOUGHERARA

M'sila University

**Academic year : 2021 / 2022**



# Contents

<b>1</b>	<b>GENERAL INTRODUCTION</b>	<b>1</b>
1.1	REAL WORLD USES OF COVERING ROBOTS . . . . .	3
<b>2</b>	<b>SIMULATOR STRUCTURE</b>	<b>7</b>
2.1	Past achievements and most successful coverage methods . . . . .	8
2.1.1	Exact cellular decomposition . . . . .	8
2.1.2	Landmark-based slice decomposition . . . . .	10
2.2	Physical features . . . . .	11
2.3	The simulation model . . . . .	11
2.4	Simulation algorithms . . . . .	13
2.4.1	Classical random walk . . . . .	13
2.4.2	Random walk optimization . . . . .	15
<b>3</b>	<b>TESTS AND RESULTS</b>	<b>20</b>
3.1	Test parameters . . . . .	21
3.2	Testing basic random walk . . . . .	21
3.3	Testing distributed random walk . . . . .	24
3.4	Statistics . . . . .	26

# List of Figures

1.1	Agricultural robotics team cooperating in a single field operation. [7]	3
1.2	Landmine detection robot. [12]	4
1.3	This is the skull bone surgery robot RoBoSculpt, developed by Jordan Bos of Eindhoven University of Technology. On the right there is a phantom piece of skull base; the remainder of the skull was added in virtually to show where this piece of bone is located[17].	5
1.4	Robot vacuum cleaner. [13]	6
2.1	Trapezoidal decomposition as vertexes of an adjacency graph. [14]	9
2.2	A zigzag path covering for each cell. [15]	9
2.3	(a) The arrangement of segments in slices made by the sweep line changes as it sweeps through the environment. (b) The number of segments present in a slice changes as the sweep line enters an obstacle.[16]	10
2.4	Basic design of robot with 4 collision sensors.	11
2.5	Surface simulation with 2 dimensional grid.	12
2.6	Surface simulation with 2 dimensional array of integers.	12
2.7	Surface visualization with robot and obstacles.	13
2.8	Different results of 4 executions of the simple random walk algorithm.	14
2.9	Robot transportation path through 16 parts.	16
2.10	1600 steps of distributed random walk algorithm.	16
3.1	coverage result of 1600 steps of simple random walk.	21
3.2	Big empty spots on a result of a simple random walk covering.	22
3.3	Covered cells and Uncovered cells after 1600 random steps..	23
3.4	Covered cells and recovered cells after 1600 random steps.	23
3.5	coverage result of 1600 steps of distributed random walk.	24
3.6	1600 steps spots that show big spots and small spots difference.	25
3.7	Covered cells and Uncovered cells after 1600 distributed random steps.	25
3.8	Covered cells and recovered cells after 1600 distributed random steps.	26

## LIST OF FIGURES

---

3.9	Coverage difference between 10 executions of both Classical random walk and Distributed random walk, green represents the optimal covered cells (800). . . . .	28
3.10	Covered surface average of 10 Classical random walk processes. . . . .	28
3.11	Covered surface average of 10 Distributed random walk processes. . . . .	29
3.12	Covered surface average and optimization ratio. . . . .	29

# List of Tables

3.1	10 Simple Random walk executions on simulator. . . . .	26
3.2	10 Distributed Random walk executions on simulator. . . . .	27

# CHAPTER 1

## GENERAL INTRODUCTION

Covering spaces is an issue that is present in almost all aspects of life, and this includes for example cleaning the land, cultivating fields, painting walls, excavation, sculpting and painting, medical surgeries, and so on.. In general, the process of covering space is trying to pass through all the points inside it, whenever the repetition is Passing through these points is less, and overcoming obstacles is more accurate, the coverage is better. Researching and improving coverage methods contributes to making it easier for humans and robots to do many jobs better and faster. This uncomplicated work will be directed to students or people who have a simple and basic idea about the random walking algorithm. Things that you might want to know separately are only basic general ideas about surface, robots, simulators, and random walk. I aim in this work to observe the random walking behavior in covering a two-dimensional surface and to discover weaknesses, improve the algorithm by solving problems, and then monitor the behavior of the modified algorithm to evaluate that improvement.

### **Surface**

Surface, In geometry, a two-dimensional collection of points (flat surface), a three-dimensional collection of points whose cross-section is a curve (curved surface), or the boundary of any three-dimensional solid. In general, a surface is a continuous boundary dividing a three-dimensional space into two regions. For example, the surface of a sphere separates the interior from the exterior. [6]

### **Robot**

A robot, is any automatically operated machine that replaces human effort, though it may not resemble human beings in appearance or perform functions in a human-like manner[5].

## **Blind robot**

It is a robot that is not equipped with cameras that allow it to identify the environment around it, usually relying on sensors to help it recognize obstacles and take the right steps.

## **Robot simulator**

A robotics simulator is an application that emulates the robot motion, simulators are used to create embedded applications for a robot without depending physically on the actual machine, thus saving cost and time. In some cases, these applications can be transferred on the real robot (or rebuilt) without modifications.

## **Surface coverage**

Surface coverage or coverage path planning is the task of determining a path that passes over all points of an area or volume while avoiding obstacles. In one of the earliest works on coverage path planning found in the literature [1], Cao, Huang, and Hall defined the requirements a robot must meet to perform a coverage operation. The target application in the aforementioned paper is a mobile robot moving in a flat 2-dimensional environment, the same criteria are applicable to other coverage scenarios. The requirements are as follows:

- Robot must move through all the points in the target area covering it completely.
- Robot must fill the region without overlapping paths.
- Continuous and sequential operation without any repetition of paths is required.
- Robot must avoid all obstacles.
- Simple motion trajectories (e.g. straight lines or circles) should be used (for simplicity in control).
- An “optimal” path is desired under available conditions.

## 1.1 REAL WORLD USES OF COVERING ROBOTS

### Agriculture

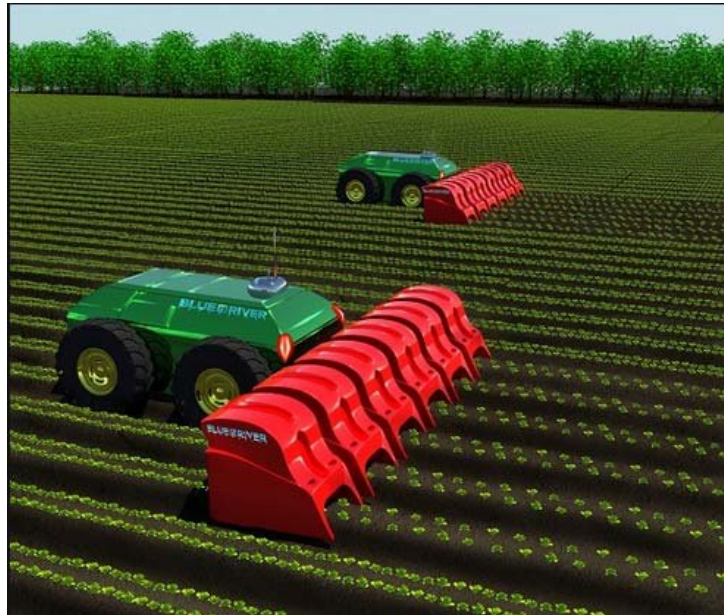


Figure 1.1: Agricultural robotics team cooperating in a single field operation. [7]

Experts predict we will reach 9 billion people by 2050 and as population growth continues, food shortage production will get worse . Today robot farming systems are one of the most effective solutions to increase productivity, wherein very wide agricultural areas and under difficult climatic conditions, these robots perform repetitive and exhausting tasks, saving effort and time for farmers and letting them tend to more strategic matters. Agricultural robots cover the field area with different strategies according to their different tasks:

- Plowing the ground (robot tractor)
- Planting (e.g. rice planting robot)
- Seeding (seeding robot)
- Irrigating
- weed control
- harvesting crops (robot harvester)
- picking fruits

Robot farming systems are more economical for farmers, and they have usually a low-cost system.

## Military

Substitution of personnel in hazardous environments and during fatigue activities is an important military requirement. Many soldiers are injured or killed during the handling and manipulation of dangerous material, especially if the work has to be repeated over and over for a longer period, e.g. de-mining or bomb disposal [3].



Figure 1.2: Landmine detection robot. [12]

Today, robots in wars and battles or while guarding the borders have an important and effective role, and these are the most important purposes for which they are used:

- Reconnaissance, surveillance, and target acquisition.
- Ordnance disposal, mine-clearing operation, and NBC decontamination.
- Security, defense, and sniper discovery.
- Tactical considerations like deception and precision strike.
- Transport, convoying and rescue.
- Sensor and communication networks.[4]
- Fighting forest fires.

## Medicine

Medicine is a field where advances in technology can directly help increase the quality of life for humans. There are several systems currently available to aid in surgical procedures. With the assistance of computer-controlled robotic systems, procedures can be performed more accurately and less invasively [8], [9]. Orthopedic surgery is one such area where the field of robotics is highly applicable. Procedures that require the precise reshaping of bone lend themselves to benefiting from highly accurate robots. In 2010 a research paper was published by



Figure 1.3: This is the skull bone surgery robot RoBoSculpt, developed by Jordan Bos of Eindhoven University of Technology. On the right there is a phantom piece of skull base; the remainder of the skull was added in virtually to show where this piece of bone is located[17].

Nathan Abraham, Alon Wolf, and Howie Choset, which considers some implementation issues in a path planner for achieving uniform coverage of a bony surface, to be used and applied as bone removal in orthopedic surgery. Nathan, Howie, and Alon represented a new system that combines a medical robotic platform with techniques usually associated with mobile robot motion planning, by treating the tip of the cutting tool as a mobile robot and the area of surgery as free space. Their algorithm combined 2 surface covering algorithms (cellular decomposition, sweep lines) and potential functions to generate a path that uniformly covers the bone surface. We can have a clear idea, about the importance of developing coverage algorithms and robotics if we know that the total knee arthroplasties alone would represent an annual cost upwards of 11.2 billion dollars. Any modification that increases a surgeon's accuracy will in turn save a significant amount of money.

## **Household robots**

Household robots help humans complete their daily tasks and reduce work time and effort. They are also called domestic robots. These robots are autonomous to a high degree and can be connected to a Wi-Fi network for greater control and management of tasks. Floor cleaning, lawn mowing, carrying, and moving items, are among the most important jobs a home robot can perform. This type of robot is useful for humans to organize their daily tasks, ease effort, and preserve their time and health, especially old people.



Figure 1.4: Robot vacuum cleaner. [13]

## **Random walk**

A random walk robot motion in general is an uncertain non-organized covering path, in each step the robot chooses randomly one of the available directions, in our case of 2 dimensional surface; up, down, right, or left.

## **Why Random-walk?**

Guided and organized walking techniques rely on calculations and analyzes of space dimensions, robot steps, and other elements, while random walking is a simpler and less complex technique. Random Walk handles less pressure and faster program execution, which is useful in cases where space coverage is not needed for accuracy and perfection.

In this research, we aim to discover how far the random walk is going to be useful for an optimal surface covering, starting with the most naive random walk, passing by some modifications, restrictions to end with an optimal coverage algorithm.

**CHAPTER 2**

**SIMULATOR STRUCTURE**

Before talking about the structure of the simulator and the method of working with random walk, it is possible to take an initial idea about the algorithms of areas covering by identifying the most successful and famous methods that were used to cover areas previously.

## 2.1 Past achievements and most successful coverage methods

A review of the most relevant works in the coverage problem, was given by Enric Galceran and Marc Carreras [10], that highlights the following works:

- Classical exact cellular decomposition methods
- Morse-based cellular decomposition
- Landmark-based topological coverage
- Contact sensor-based coverage of rectilinear environments
- Grid-based methods
- Graph-based coverage
- 3D coverage
- Optimal coverage
- Coverage under uncertainty
- Multi-robot methods

I represent more explanation of two of them; the exact cellular decomposition, and landmark-based slice decomposition, according to Enric Galceran and Marc Carreras [10] explaining.

### 2.1.1 Exact cellular decomposition

A planner based on exact cellular decomposition generates a coverage path in two steps. First, it decomposes the free space into cells and stores the decomposition as an adjacency graph:

1. Decomposition of cells can be generated by sweeping a line through the space.
2. The cell boundaries are formed when some event is encountered by the sweep line.
3. A change on the number of times the sweep line intersects with obstacle boundaries can be used as an event.
4. Two cells are said to be adjacent if they share a common boundary.

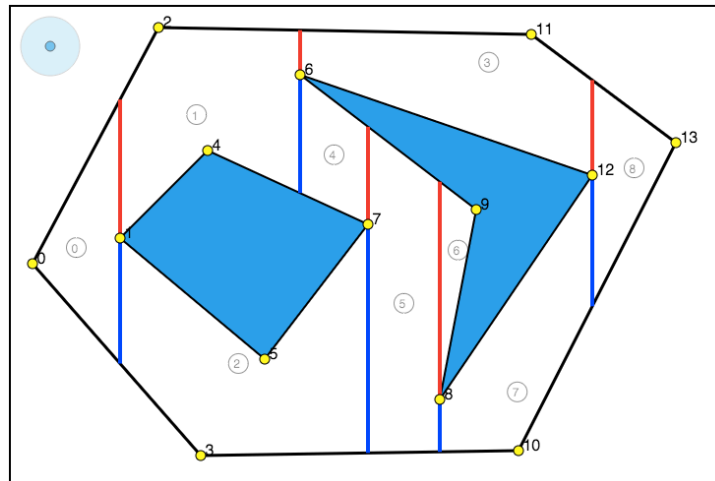


Figure 2.1: Trapezoidal decomposition as vertexes of an adjacency graph. [14]

5. In the adjacency graph, a node represents a cell and an edge represents an adjacency relationship between two cells.

Second step:

1. Computing an exhaustive walk through the adjacency graph (i.e., a sequence that visits each node in the graph exactly once).
2. Finally, a specific zigzag path to cover each cell is generated.

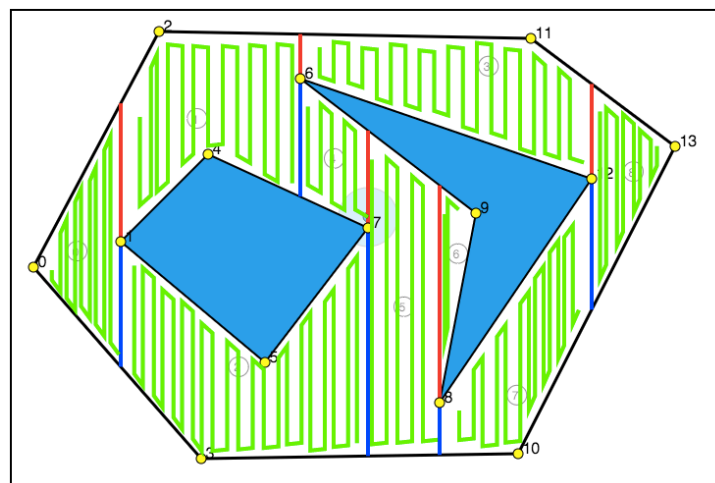


Figure 2.2: A zigzag path covering for each cell. [15]

Notes:

- Trapezoidal decomposition is one of the simplest exact cellular decomposition techniques which can yield a complete coverage path, which handles only planar, polygonal spaces. In the trapezoidal decomposition, each cell is a trapezoid, as shown in Fig 2.1

- The exhaustive walk determines the order in which the cells are visited to achieve complete coverage.

### 2.1.2 Landmark-based slice decomposition

A slice decomposition is created by sweeping a line from the top of an environment to the bottom. There are two types of a region; obstacle, and free space. At any time, the sweep line intersects a number of free space and obstacle regions.

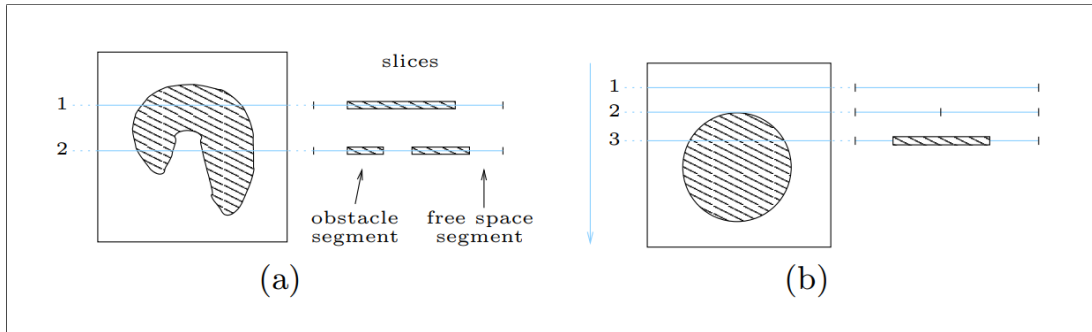


Figure 2.3: (a) The arrangement of segments in slices made by the sweep line changes as it sweeps through the environment. (b) The number of segments present in a slice changes as the sweep line enters an obstacle.[16]

## 2.2 Physical features

It is important to talk first about what kind of bots are suitable for this emulator. The reason of clarifying this in the beginning is that understanding the simulator characteristics and features give a previous obvious view for the covering operation algorithm. It is more practical to talk about hardware features before software, and that is to make the reader of this paper able to have an initial idea about the algorithm before diving into a further explanation.

The robot to be simulated doesn't have cameras, it is equipped with four sensors in the four sides, and therefore its movement will only be horizontal and vertical without rotation. On each of the four sides of the robot there is a collision sensor.



Figure 2.4: Basic design of robot with 4 collision sensors.

The way the robot walks is through equal spaced steps (units), because the space to cover is being treated as a two-dimensional grid of cells, therefore moving one step is covering one cell in the space, see figure 2.5 These are the most important characteristics that affect the emulator's algorithm and its data structures.

## 2.3 The simulation model

The data to be simulated is about three important things:

- Surface
- Robot
- Obstacle

Simulation model is meant to be a description part of the way I simulated the covering process. As this covering process is considering the surface to be covered, its obstacles, and the robot,

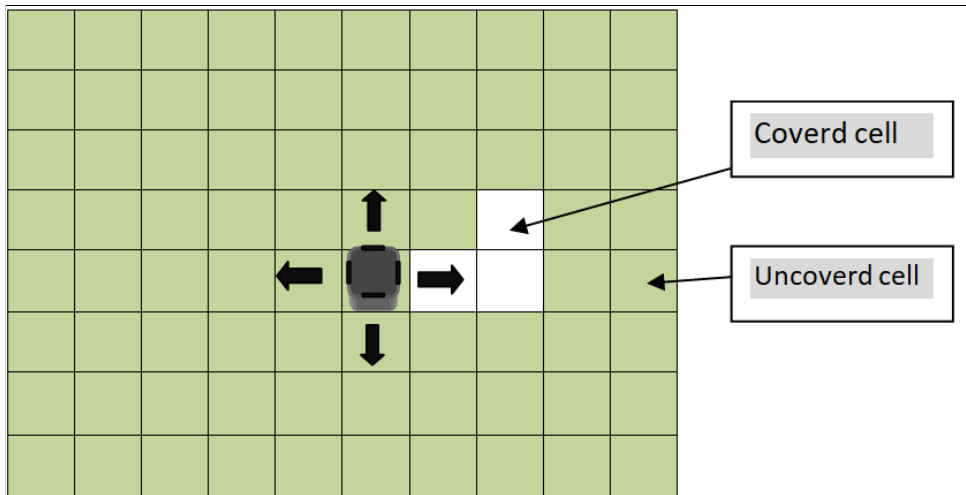


Figure 2.5: Surface simulation with 2 dimensional grid.

explaining this covering simulation includes the explanation of each one of these three objects simulation.

The surface is considered a 2 dimensional grid, it is simulated by a 2 dimensional array of integers.

The cells that are occupied with obstacles are represented by the value 2.

The cell that is occupied by the robot is represented by the value 3.

Covered cells hold the value 1.

Uncovered cells hold the value 0.

See Figures 2.6 and 2.7

0	0	0	2	2	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	2	0	0	0
2	0	0	0	0	3	2	0	0	0
0	2	0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0

Figure 2.6: Surface simulation with 2 dimensional array of integers.

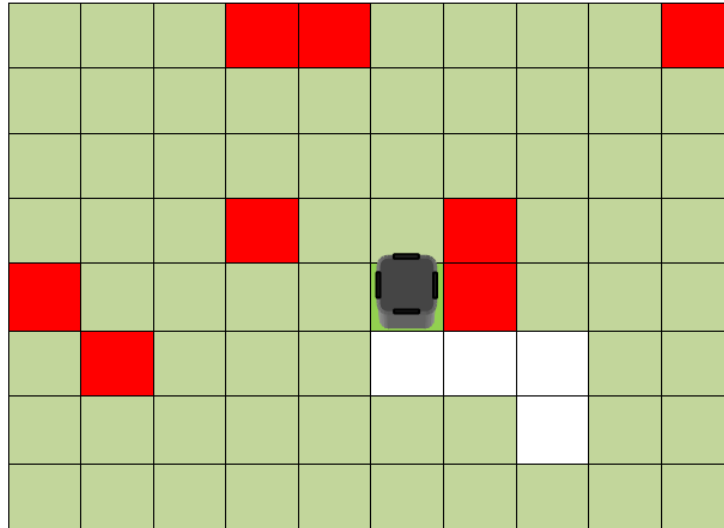


Figure 2.7: Surface visualization with robot and obstacles.

## 2.4 Simulation algorithms

In this work I represent the behavior of random walk strategy on covering surfaces, starting with the naive random walk algorithm, ending at distributed random walk as an optimization.

### 2.4.1 Classical random walk

Classical random walk algorithm is the basic one, where the steps are completely random generated.

From the beginning of the covering process to the end, the steps are not redirected with any organizing or computing, which means that the covered surface can be in any predictable or completely unexpected shape.

#### Algorithm

1. Chose a random direction from (up, down, left and right).
2. Move the robot one step towards the chosen direction.
3. Repeat (1)

```

steps= [(0,1), (0,-1), (1,0), (-1,0)]
READ stepNumber
FOR i=0 to stepNumber
  s= random(steps)
  robot.location+= s
END FOR

```

1. The algorithm reads the number of steps the robot will makes (the input could be turned into time later).
2. For each iteration a random direction is chosen from the list (steps) of four directions, the coordinates of that random step are added to the robot location coordinates.

### Random walk covering examples

In the figure 2.8, a visualization of what the results of the classic random walk might look like.

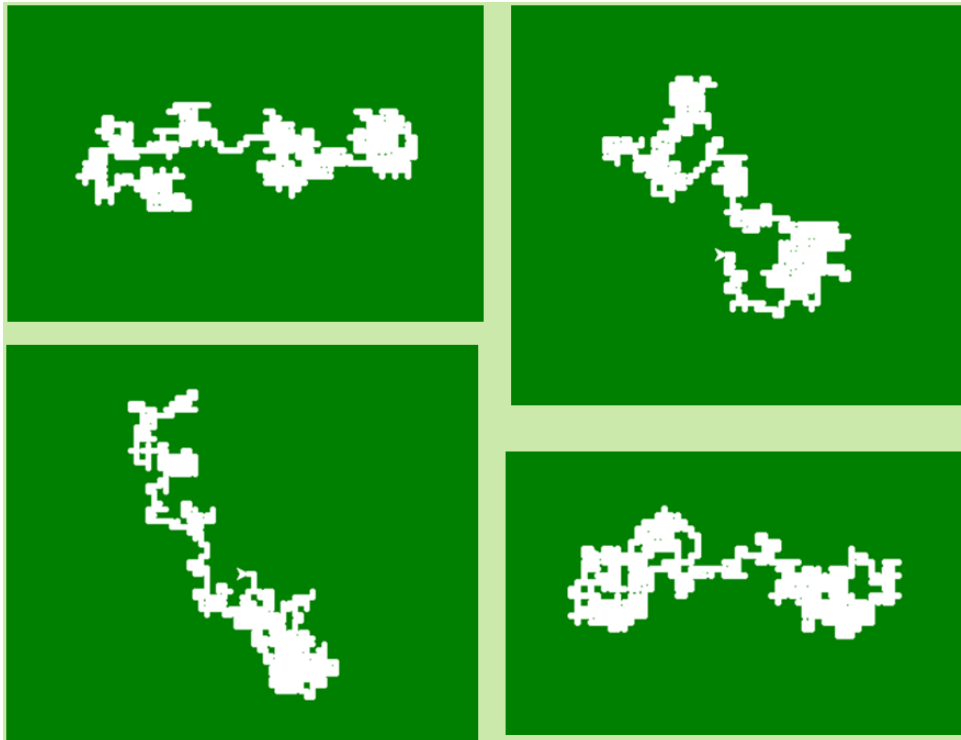


Figure 2.8: Different results of 4 executions of the simple random walk algorithm.

### Advantages

Completely random walk is a quite simple algorithm that has no restrictions on its randomization, that will makes it less complex and more economic.

1. First advantage is complexity, the algorithm is done only in few lines, which makes it too easy to code, modify, or to implement into a simulator or a machine.
2. Time is an important matter in such operations like covering, in the classic random walk, the low complexity of the algorithm affects the time of execution. As this simple random walk has a very low complexity, its execution will be very quick, therefore the robot will perform faster and continue the covering process in a short time.
3. In this covering process the robot could face obstacles at any step, when it uses the naive

random walk the robot changes its direction randomly and continue in the random selection of other steps. In this case we might have a good reaching to small spaces between obstacles that are lowly expected to be covered, thanks to the non oriented walk that is completely random and unpredictable.

4. Thanks to the short time this algorithm takes, the energy is less consumed by a robot.

### **Disadvantages**

Non oriented random walk strategy has several disadvantages on a surface covering operation;

1. Having large non covered spots as a result is a common problem, the random walk with no orientation usually leaves big empty spaces and repeats the covering of other spots, that is will be clear with test and result in chapter N°03 later.
2. After a non complete coverage, the surface may need another execution, which takes more time and energy.
3. Being not aware of the random result, makes the user of the robot insure about the execution the time he must set in.

### **2.4.2 Random walk optimization**

The optimal covering process result is a full covered surface with no repetition on covered cells, an optimization for this simulator random walk algorithm requires the maximization of covered surface, to minimize recovered cells, and to minimize time of process.

To valuate the algorithm efficiency I set a calculator of steps that repeat covering the cells that are already covered, for example at each step the robot choses to take a random step, it will test if the cell the robot will take to is aready covered, if it is the recoverd steps number increases.

I tried to put a small modification on this simple algorithm to move the robot through regions instead of covering intensely one area, the surface will be divided into parts, the robot should move from a part to another after certain number of steps, that is to solve the problem of covering around only some areas, this modification is the distributed random walk algorithm.

### Distributed random walk

It is a modified random walk algorithm that I've called "distributed" because of the instructions that manage the random steps election to be equally distributed on the whole surface.

This distribution is by dividing the surface into parts, see figure n, I chosen in this experience the surface to be divided into 16 parts, the main idea of this algorithm is to distribute the amount of steps on on all these parts equally which solve the big empty spots problem.

Each part is going to be covered separately as an independent surface under no orientation using the classic random walk algorithm, once the number of steps reaches the part steps limit the robot moves to cover the next part, see figure 2.9

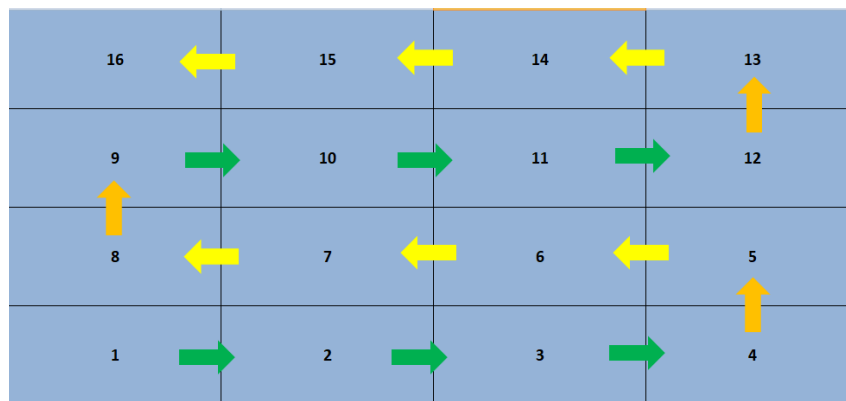


Figure 2.9: Robot transportation path through 16 parts.

In figure 4.7, a visualization of what the results of the distributed random walk might look like.

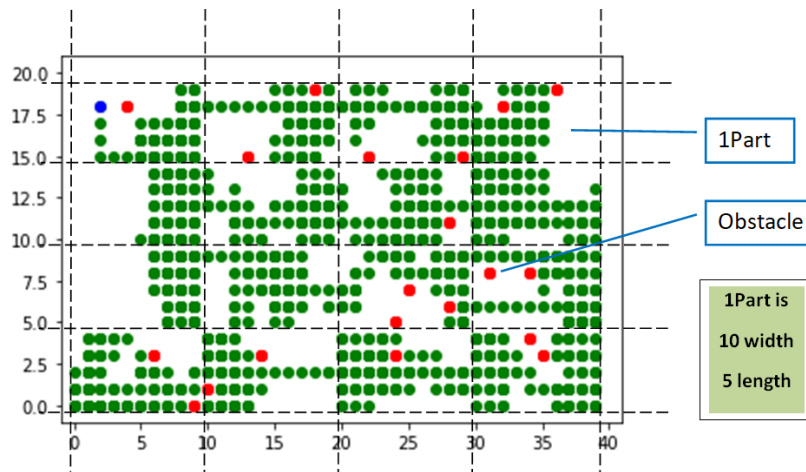


Figure 2.10: 1600 steps of distributed random walk algorithm.

### Algorithm

1. Transport the robot to another part if robot steps reached 100.
2. Chose a random direction from (up, down, left and right).
3. Move the robot one step towards the chosen direction only inside the part borders.
4. Repeat (1)

```

steps= [(0,1), (0,-1), (1,0), (-1,0)]
READ stepNumber

FOR i=0 to stepNumber
  IF robot.walkedSteps IN [limits]
    robot.transportMode <- True
    robot.transport()
  END IF

  s= random(steps)
  robot.location <- robot.location + s
END FOR

```

### Transport Algorithm

Transporting the robot from a part to the next part has 3 cases:

Transport to the right

Transport to the top

Transport to the left figure

Each case process has an algorithm with different parameters, because sometimes the robot moves up, sometimes it moves to right, and sometimes to the left, and the limits where the robot should stop differ from a part to another and from a side transporting to another, for example the robot in part 1 should move to the right successfully until its location coordinate becomes (x=10) which indicates that the transportation to the other part is done.

Here is an explanation of the transportation algorithm to the right side, it is the same structure as transport to left and transport to top, only some parameters and values that differ.

```

transportRobot()
BEGIN
  IF robot.currentPart IN [[1, 2, 3, 9, 10, 11] THEN
    stps <- [(0,1), (0,-1)]
    stp <- random.choice(stps)
    WHILE robot.location[0] != transportUntil[robot.currentPart] THEN
      WHILE mapp[robot.location[0]+1][robot.location[1]] = 2 THEN
        WHILE isRob_atObstacle(robot, stp) THEN
          stps.remove(stp)
          stp <- random.choice(stps)
        END WHILE

        increaseRecoveredcells(robot, stp)
        robot.location[1] <- robot.location[1]+stp[1]
        robot.time <- robot.time+3
        robot.stepsTotal <- robot.stepsTotal+1
        stps <- [(0,1), (0,-1)]
        map[oldLocation[0]][oldLocation[1]] <- 1
        map[robot.location[0]][robot.location[1]] <- 3
        plot.show()
      END WHILE

      increaseRecoveredcells(robot, (1,0))
      robot.location[0] <- robot.location[0]+1
      robot.time <- robot.time+3
      robot.stepsTotal <- robot.stepsTotal+1

      map[oldLocation[0]][oldLocation[1]] <- 1
      map[robot.location[0]][robot.location[1]] <- 3
      plot.show()
    END WHILE

    robot.transportMode <- False
  END IF
END

```

The same algorithm of the Simple random walk, but with an additional verification which is checking the walked steps if they reach the limit (100 steps inside each part), if they do the

robot has to be transported first to the next part before continuing to take random steps inside the new part;

1. Read the a number of steps that are aimed to be walked by the robot (user input).
2. Check if the actual walked steps equal any item in the list [99, 199, 299,.....] which means the end of current part covering, if steps attribute equals an element of the list the robot calls transport method.
3. After a transportation process, the robot takes a random step inside the new part.
4. End process if the user input = walked steps, if not go to (2).

The distributed random walk considers the part borders as obstacles to be avoided randomly, the method `isObstacle()` that plays a role of a sensor returns true if the next random step takes to an obstacle and false if it doesn't, this method is called in both simple random walk and distributed random walk algorithms, but the difference is `robot.transportMode` attribute that is turned to true only when a robot needs a transportation, inside `isObstacle()` method there is a test that checks if the `robot.transportMode` is active, if it is the border of `robot.currentPart` are considered as obstacles and the robot won't be able to move outside the part borders (in distributed Random walk), if the `robot.transportMode` is false the robot will moves freely considering only the surface borders as obstacles (in Simple Random walk).

### **Advantages**

1. Approximate full coverage by moving through the surface parts, and decreasing the big empty spots.
2. Similar result on the surface parts.

### **Disadvantages**

1. After modifying a simple random walk, each orientation added to the robot algorithm often creates other problems that need to be separately managed.
2. The solution of big empty spots creates the problem of small empty spots.
3. Modifications that occur to the simple random walk increase its complexity especially if each modification is optimized itself, this diversity turn the cover operation into a complex one, and consume more time and energy.

# **CHAPTER 3**

## **TESTS AND RESULTS**

### 3.1 Test parameters

The simulation test is based on a surface of  $20 \times 40$  dimensions in cells ( $6.6 \times 13.3$  in meters), I set average dimensions that can be modified on the simulator.

Time in the simulator program is counted in seconds since the robot in this test spends 3 seconds to cover 1 cell, time values in results are recorded in minutes for an easier conception.

I set these parameters (surface dimensions, cell covering time) emulating a cleaning coverage, these parameters values depend on the covering case, they still changeable in the simulator.

### 3.2 Testing basic random walk

After 1600 steps using the classic random walk algorithm, figure 3.1 represent the final result of covering process.

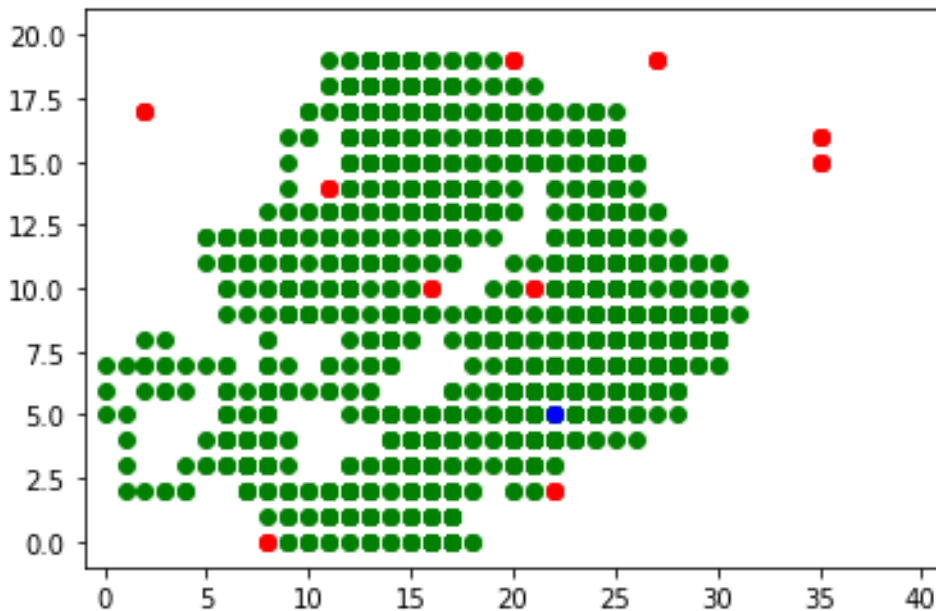


Figure 3.1: coverage result of 1600 steps of simple random walk.

Green dots represent covered cells, red are obstacles, and the white space is the cells that are not covered.

## Result analyzing

It is clear in figure 3.2 that there are a big number of adjacent cells that are not covered, which is more dense on the wright side of the plot and on the left side.

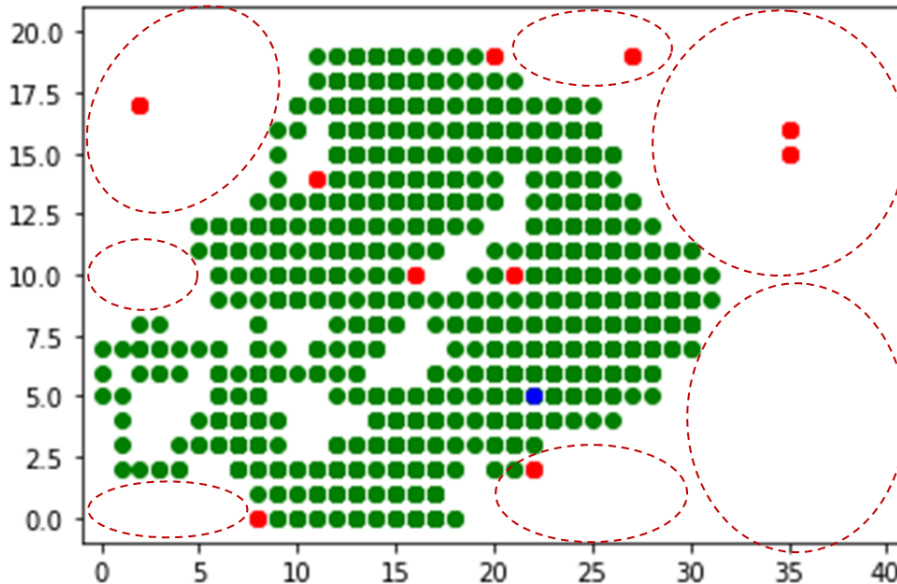


Figure 3.2: Big empty spots on a result of a simple random walk covering.

## Problems

The problem of big empty spots looks like to be often happening in a classic random walk (which some statistics show later), the reason is that the robot don't change region, which means simply it returns to move around covered cells more than discovering other places.

These large empty spaces were solved by the distributed random walk, in the following tests and results a clarification of how the distribution technique solved the isolation non covered cells.

The behaviour of this algorithm that covers covered cells and concentrate around an area decreases covered space and increases uncovered cells, see figure 3.3, which increase recovered cells as well, see figure 3.4.

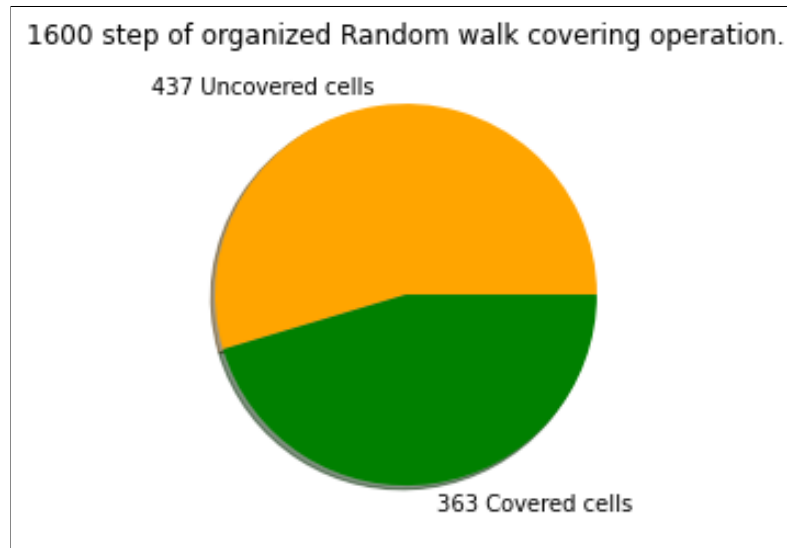


Figure 3.3: Covered cells and Uncovered cells after 1600 random steps..

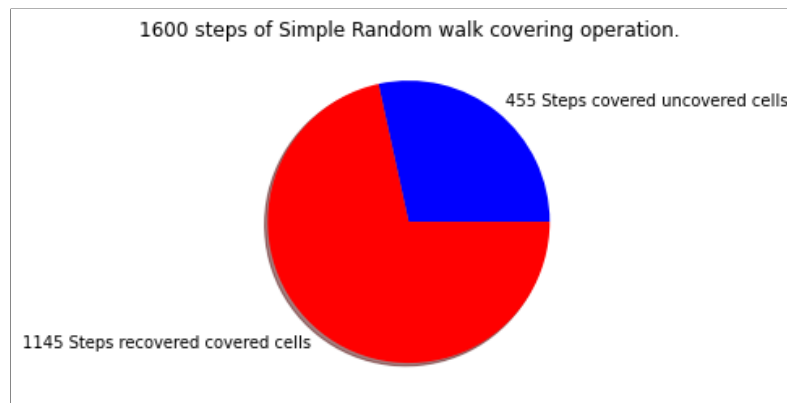


Figure 3.4: Covered cells and recovered cells after 1600 random steps.

### 3.3 Testing distributed random walk

After 1600 step of this robot simulator using the distributed random walk algorithm, figure 3.5 represent the final result of covering process.

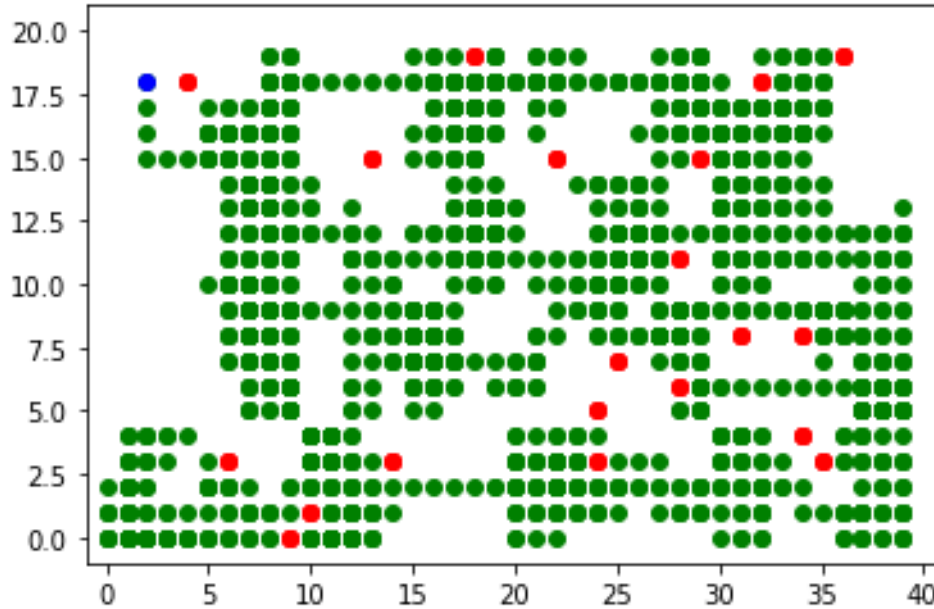


Figure 3.5: coverage result of 1600 steps of distributed random walk.

#### Result analyzing

The following figure shows the difference that a distributed random walk has made.

The distributed random walk result in the right side of figure 3.6 expose a clear decreasing of big empty spots.

The robot barely reached all areas.

the uncovered cells in the distributed random walk result are more adjacent to covered cells than they are in the simple random walk.

## Problems

In this optimization the distribution creates a problem of small distributed spots around the surface, see figure 3.5 and 3.6

It is caused by the explicit robot transportation from a part to another, even if the part isn't covered completely.

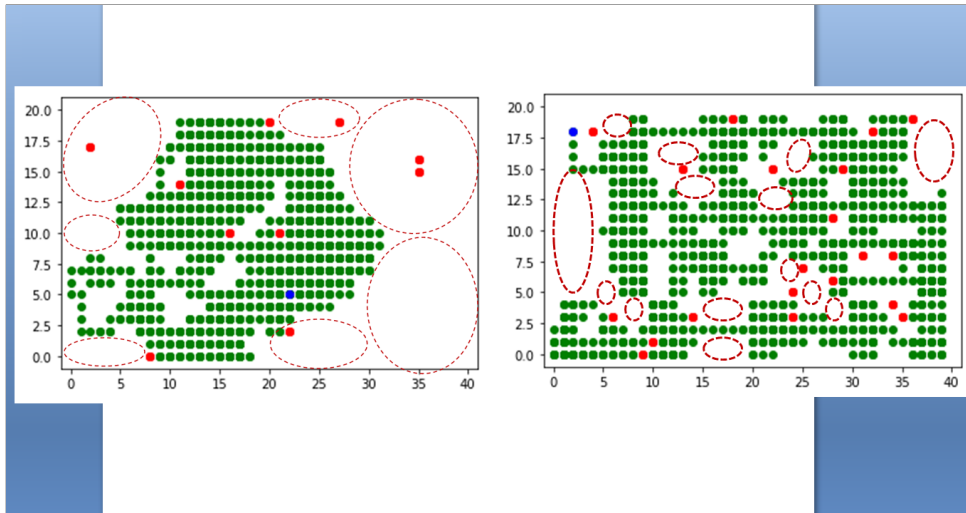


Figure 3.6: 1600 steps spots that show big spots and small spots difference.

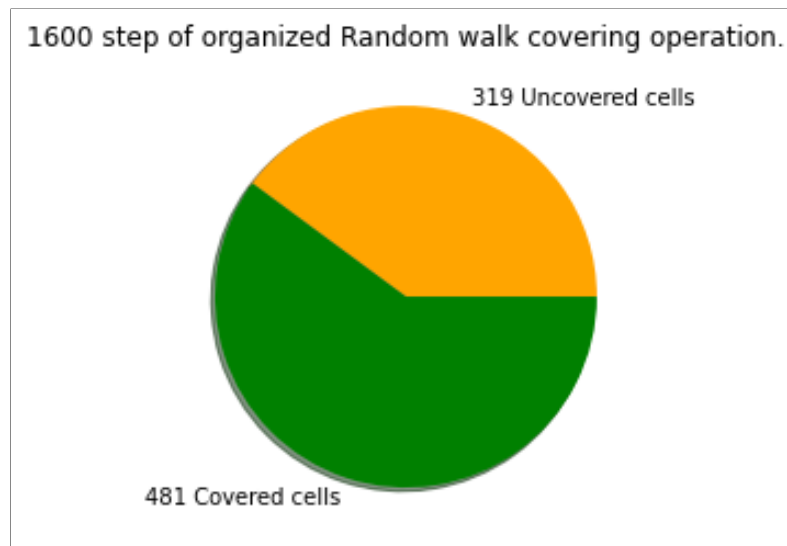


Figure 3.7: Covered cells and Uncovered cells after 1600 distributed random steps.



Figure 3.8: Covered cells and recovered cells after 1600 distributed random steps.

### 3.4 Statistics

In order to measure the effectiveness of both algorithms and for not being based on analyzing the results of one experiment only, I registered 20 executions, 10 executions using the basic random walk coverage, and 10 using the distributed random walk, then I calculated the average results to make the analysis more accurate.

#### Classic random walk

The following results represent the output of 10 executions of basic random walk on a surface of  $6.6 \times 13.3$  meters, using one robot and 20 obstacles that were randomly located at each covering process.

Each row of the table represent 1 process output.

Table 3.1: 10 Simple Random walk executions on simulator.

Covered cells	Walked steps	Recovered cells	Operation time(min)	Covered surface
265	1600	1335	80	33.13%
407	1600	1193	80	50.88%
425	1600	1175	80	53.13%
451	1600	1149	80	56.38%
399	1600	1201	80	49.88%
377	1600	1223	80	47.13%
370	1600	1230	80	46.25%
409	1600	1191	80	51.13%
373	1600	1227	80	46.63%
370	1600	1230	80	46.25%

Average of covered cells: 384.6 cells; 128.2 meters<sup>2</sup>

Average of covered surface: 48%

### Distributed random walk

The following results represent the output of 10 executions of distributed random walk on surface of 6.6×13.3meters, using one robot and 20 obstacles that were randomly located at each covering process.

Each row of the table represent 1 process output.

Table 3.2: 10 Distributed Random walk executions on simulator.

Covered cells	Walked steps	Recovered cells	Operation time(min)	Covered surface
489	1682	1193	84	61.12%
515	1673	1158	83	64.38%
523	1680	1157	84	65.38%
533	1680	1147	84	66.63%
497	1674	1177	83	62.13%
519	1674	1155	83	64.88%
481	1681	1200	84	60.12%
527	1680	1153	84	65.88%
535	1661	1126	83	66.88%
498	1687	1189	84	62.25%

Average of covered cells: 488.4 cells; 162.8 meters<sup>2</sup>

Average of covered surface: 64%

### Note

Time was not counted because the statistics were done with 1600 steps as parameter, which means the same time (1600\*3 seconds) is spent in each one of these processes.

**Evaluation of optimization**

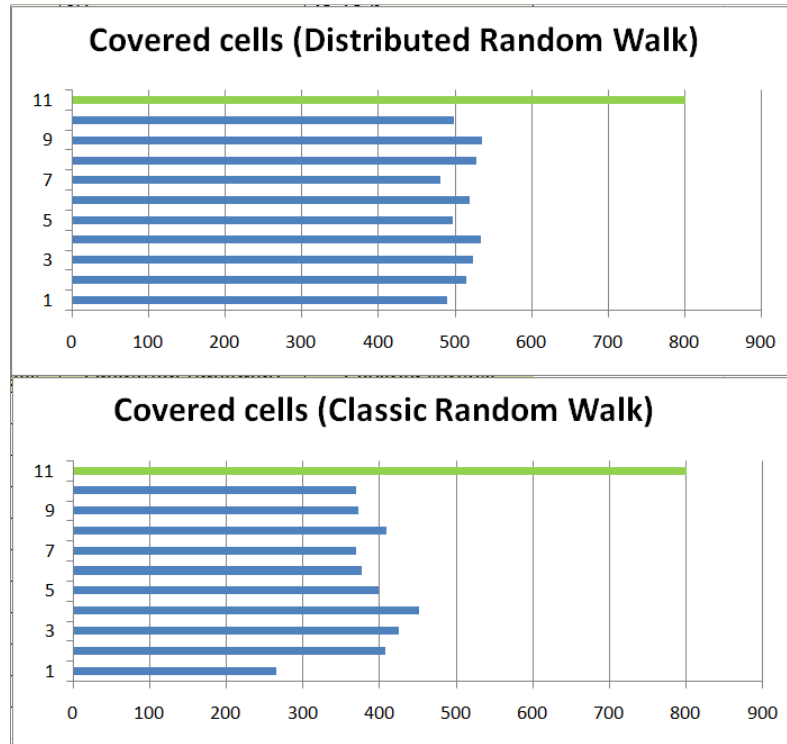


Figure 3.9: Coverage difference between 10 executions of both Classical random walk and Distributed random walk, green represents the optimal covered cells (800).

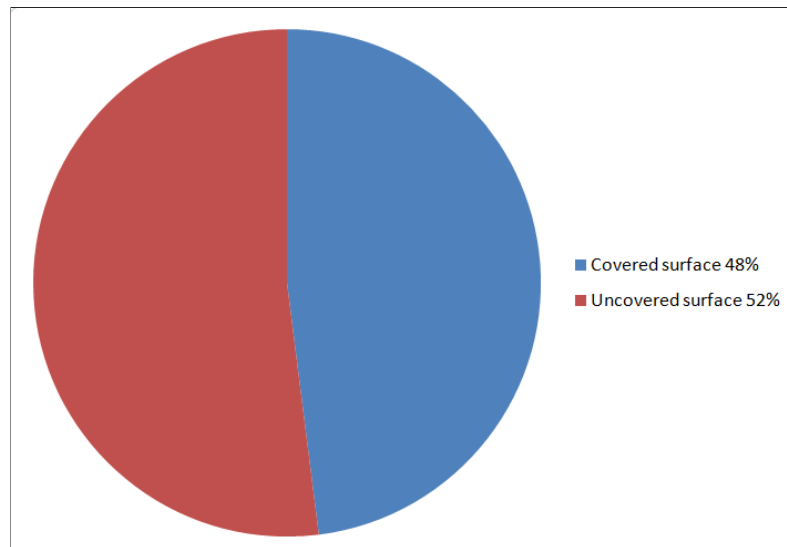


Figure 3.10: Covered surface average of 10 Classical random walk processes.

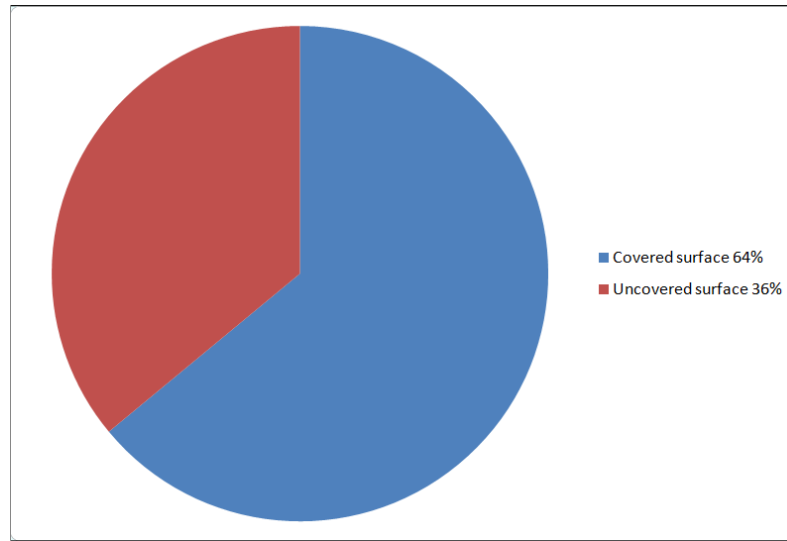


Figure 3.11: Covered surface average of 10 Distributed random walk processes.

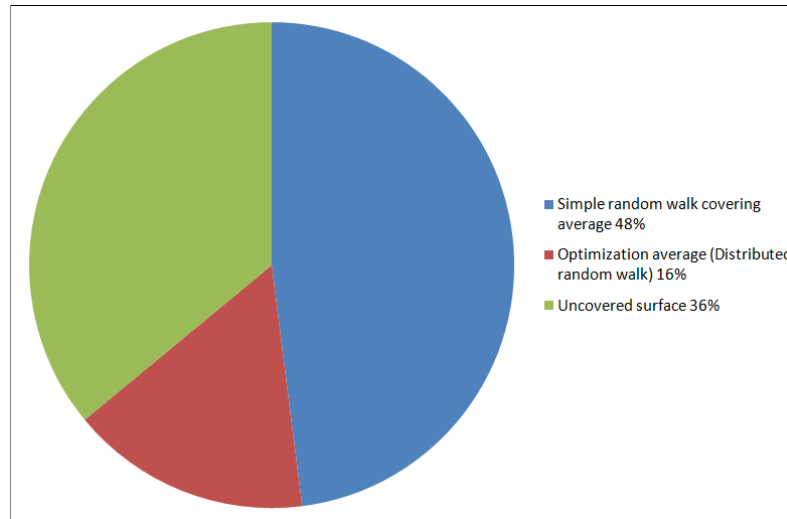


Figure 3.12: Covered surface average and optimization ratio.

# GENERAL CONCLUSION

After I reached the end of this work, it became clearer to me that the problem of area covering using the random walk strategy is a field with broad horizons in which creativity can be used in a variety of ways according to the kind of coverage use. It also became clear to me how much a random walk can be useful in covering a two-dimensional space using a blind robot, and this is after studying the algorithm's behavior in both cases, the first case is the normal random walk and the second is the distributed random walk (an optimization of the first algorithm) where the steps are taken evenly over the sections of the surface.

As for the tests and results of both methods in the last, I showed how useful and effective the optimization was, and highlighted the difference between normal and distributed random walks, a difference that I don't see as big since I didn't go very far in thinking about developing the modification to make the mixture between randomness and organization more intelligent and effective.

It is important to mention that I think that the results I have reached are a good starting point for those who want to improve the random walk algorithm in surface covering issues, and even if I believe that the coverage percentage of the modified algorithm that I reached which is 64% does not reach the required level, I see that this work indicates clearly the weaknesses of the random walk in coverage process, its strengths, and how an optimization affects it, so these points that take time from any researcher consider them a helpful and important resource as a start to looking to improve a more intelligent and accurate random coverage algorithm. After many successive optimizations in this type of coverage to organize it more, the random walk strategy may lose its original content and random nature, which is why if you as a researcher want to develop in it, the question may be more important to ask before; "is using modified works with important results a better idea than starting from scratch with a more effective modification?"

# Bibliography

- [1] Cao, Z. L., Huang, Y. and Hall, E. L.: 1988, Region filling operations with random obstacle avoidance for mobile robotics, *Journal of Robotic Systems* 5(2), 87–102.
- [2] Nathan Abraham, Alon Wolf, and Howie Choset. (2006), A potential function approach to surface coverage for a surgical robot, *Computer Aided Surgery*, 11:1, 1-9, DOI: 10.3109/10929080500432173
- [3] NORTH ATLANTIC TREATY ORGANISATION, RESEARCH AND TECHNOLOGY ORGANISATION. 2008, Multi-Robot systems in military domains, <https://apps.dtic.mil/sti/pdfs/ADA499503.pdf>
- [4] NORTH ATLANTIC TREATY ORGANISATION, RESEARCH AND TECHNOLOGY ORGANISATION. 2008, Multi-Robot systems in military domains, <https://apps.dtic.mil/sti/pdfs/ADA499503.pdf>
- [5] Moravec, H. Peter (2021, February 4). robot. *Encyclopedia Britannica*. <https://www.britannica.com/technology/robot-technology>
- [6] Britannica, T. Editors of Encyclopaedia (2016, October 28). surface. *Encyclopedia Britannica*. <https://www.britannica.com/science/surface-geometry>
- [7] Hameed, Ibrahim. (2018). A Coverage Planner for Multi-Robot Systems in Agriculture. 10.1109/RCAR.2018.8621801.
- [8] Kavoussi L. R., Moore R. G., Adams J. B., Parin A. W. Comparison of robotic versus human laparoscopic camera control. *Proceedings of 2nd Annual International Symposium on Medical Robotics and Computer-Assisted Surgery (MRCAS '95)*, Baltimore, MD, November 1995; 284–287.
- [9] Kazanzides P., Mittelstands B. D., Musits B. L., Bargar W. L., Zuhar J. F., Williamson B., Cain P. W., Cabon E. J. An integrated system for cementless hip replacement. *IEEE Eng Med Biol* 1995; 14: 307–312.

## BIBLIOGRAPHY

---

- [10] Enric Galceran and Marc Carreras, A Survey on Coverage Path Planning for Robotics, Underwater Robotics Research Center (CIRS), University of Girona, 2013.
- [11] Modern rice planting in Taiwan. (2020). [Image]. Retrieved from <https://media.istockphoto.com/photos/the-rice-planter-picture-id147031493?s=612x612>
- [12] U.S. Army CCDC Army Research Laboratory Public Affairs. (2020). New landmine detection method to reduce false alarm rates [Image]. Retrieved from <https://api.army.mil/e2/c/images/2020/08/18/951108c7/max1200.jpg>
- [13] NEATO VS ROOMBA – WHICH IS THE BEST ROBOT VACUUM?. [Image]. Retrieved from <https://homevacuumzone.com/wp-content/uploads/2016/12/iadapt-2.0-navigation-1280x640.jpg>
- [14] Trapezoidal Cell Decomposition and Coverage. [Image]. Retrieved from <https://user.ceng.metu.edu.tr/akifakkus/courses/ceng786/images/hw3/boustrophedonCellDecomposition.png>
- [15] Trapezoidal Cell Decomposition. [Image]. Retrieved from <https://user.ceng.metu.edu.tr/akifakkus/courses/ceng786/images/hw3/cellCoverage.png>
- [16] Sylvia C., Wong Bruce A. MacDonald. Department of Electrical and Computer Engineering, University of Auckland. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136.1187rep=rep1type=pdf>
- [17] Bart van Overbeeke/Eindhoven University of Technology. (2018). New robot for skull base surgery is very accurate and alleviates surgeon's workload [Image]. Retrieved from <https://www.eurekalert.org/multimedia/703904>

## ملخص

هذا العمل عبارة عن دراسة مبسطة حول استخدام استراتيجية المشي العشوائي في تغطية المساحات ثنائية الأبعاد على محاكي روبوت أعمى. الفصل الأول مقدمة عامة لموضوع تغطية الأسطح، متبوعة ببعض التعاريف الأولية وأهمية التغطية في مجالات الحياة المختلفة. ثم البعض من أهم طرق التغطية، الاستراتيجيات والخوارزميات التي تم التوصل إليها في الفصل الثاني مع شرح بسيط من اثنين منهم، ويحتوي الفصل الثاني بشكل أساسي على بنية المحاكى وخوارزمياته. استراتيجية المشي العشوائي مقابل المشي العشوائي الموزع كتحسين سيتم التطرق لهما أيضا، التصورات، الميزات، ونقاط الضعف. محتوى الفصل 3 والأخير هو التجارب والنتائج على تطبيق المحاكى حيث سيكشف القارئ عن مستوى فعالية هذه الخوارزمية المحسنة.

كلمات مفتاحية : تغطية المساحات، روبوت تغطية، المشي العشوائي، محاكي روبوت.

## Abstract

This work is a simplified study on the use of random walk strategy in covering 2D spaces on a blind robot simulator. The first chapter is a general introduction to the topic of surface covering, followed by some initial definitions and the importance of surface covering in different areas of life. Then some of the most important methods of coverage, Achieved strategies and algorithms in the second chapter with a simple explanation of two of them, and the second chapter mainly contains the structure of the simulator and its algorithms. Random walking versus distributed random walking strategy as an improvement will also be addressed, perceptions, advantages, and weaknesses. The content of the 3rd and final chapter is the test and results on the emulator application where the reader will reveal the level of effectiveness of this improved algorithm.

Keywords : Surface covering, covering robot, Random walk, robot simulator.

## Résumé

Ce travail est une étude simplifiée sur l'utilisation de la stratégie de marche aléatoire dans la couverture d'espaces 2D sur un simulateur de robot aveugle. Le premier chapitre est une introduction générale au sujet du revêtement de surface, suivi de quelques définitions initiales et de l'importance du revêtement de surface dans différents domaines de la vie. Ensuite, certaines des méthodes de couverture les plus importantes, les stratégies et algorithmes obtenus dans le deuxième chapitre avec une explication simple de deux d'entre eux, et le deuxième chapitre contient principalement la structure du simulateur et ses algorithmes. La marche aléatoire par rapport à la stratégie de marche aléatoire distribuée en tant qu'amélioration sera également abordée, les perceptions, les avantages et les faiblesses. Le contenu du 3ème et dernier chapitre est le test et les résultats sur l'application émulateur où le lecteur révélera le niveau d'efficacité de cet algorithme amélioré.

Mots clés : Revêtement de surfaces , marche aléatoire, simulateur de robot.