Comprehensive Disinfection and Coverage Optimization System with Multi-Robot Coordination

Siva Surya Venkat Busi, Jhati Seelapureddy, Vidit Hirve, Anney Romo Herrera

1. Abstract

There has been extensive research on multi-robot systems achieving coverage tasks such as comprehensive disinfection. Based on the research, in this project, we propose two control approaches, one is based on generic logic and the other with a combination of two existing models. The fundamental model is centralized, which means all the robots get information about areas covered by other robots and their positions so that previously covered areas will be exempted from future coverage planning to increase overall efficiency. The second model, the Market-Potential model is a combination of the market-based model and the Potential field Model. The mathematical models of the fundamental model and market-potential model are described below. All the models were validated using simulations in MATLAB. Also, we added a Dynamic Penalty Controller to the Market-Potential model to increase its speed efficiency.

2. Mathematical Model

2.1 Goal

In this project, we tried to achieve the collective behavior of a swarm of ‘n’ number of ground robots to perform a coverage task of comprehensive disinfection. There are many similar coverage problems like disinfection such as cleaning robots, fruit-picker robots that use graph-based approach, potential-field based approach, ACO approach, etc. The availability of the data on this topic is so vast, that our goal of comparing our controller with existing methods in a 2-D environment was possible. Our goal was to create a ‘Public Library’ environment, run our comprehensive disinfection and coverage simulation successfully, and verify with other models.

2.2 Assumptions and Constraints

- The robots are assumed to be equipped with an unlimited supply of disinfecting agents.
- The map of a public library is provided to the robots to eliminate the need for real-time mapping. Mapping is not required .
- The robots can move only in 4 cardinal directions(Up, Down, Left, Right) to simplify programming and simulations.
- The robots can only move 1 node per second to its immediately adjacent nodes in its cardinal directions.
- There are no dynamic obstacles on the map that the robots have to avoid.

2.3 Parameters

- Our model works for any number of robots, but for this project, we decided to use 5 robots.
- The robots were initialized randomly to test the robustness of our algorithms.
- We tested our model for Comprehensive Disinfection on a 14x12 grid map of a 'Public Library', it can also work for bigger maps.
- We also included a semi-closed room to test our models for the robot's ability to navigate and disinfect in slightly more complex environments.
- We tested the models on several maps having random obstacles and the model was successfully able to perform the task.

2.4 Models

Considering the goal, assumptions, constraints, and parameters, a fundamental model was developed using unique techniques not covered in class. This fundamental model is described in section 2.4.1. Based on this fundamental model and general concepts taken from market-based bidding models and potential field models a second model was created. This second model is described in section 2.4.4, while the concepts it is based on are described in sections 2.4.2 and 2.4.3. In general, both models operate based on a given 2-D map. This map is represented as a series of 1's and 0's. In this case, 1's represent the areas of a room that are blocked off either by a wall or an obstacle. These are the unreachable areas of a room. On the other hand, 0's represent the reachable areas of a room that need to be disinfected.

2.4.1 Fundamental Model

The fundamental model created followed all the assumptions mentioned above. This is a centralized model, so all robots have information on the areas that have been covered and the position other robots are in. Although this model could be used for any amount of robots, a total of 5 robots were used. Let's define the robot number as $i \in 1,2,..,5$. The position of each robot i is represented as $[x_i, y_i]$. This position is associated with a node or a 0 on the given 2-D map. When moving, each robot i only has four possible positions it can move to assuming there are no obstacles or walls. The direction of these possible positions is described as m in the equation below.

$$m = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Eqn. 2.4.1.1

It can move from its position to the node directly above, below, left, or right. Furthermore, the possible positions that the robots can move to are described in the equation below as M.

$$M = \begin{bmatrix} x_i & y_i \\ x_i & y_i \\ x_i & y_i \\ x_i & y_i \end{bmatrix} - m = \begin{bmatrix} x_i & y_i - 1 \\ x_i - 1 & y_i \\ x_i & y_i + 1 \\ x_i + 1 & y_i \end{bmatrix}$$

Eqn. 2.4.1.2

This can also be visually represented in the figure below, where the blue dots are the nodes and the orange lines are the possible moves.

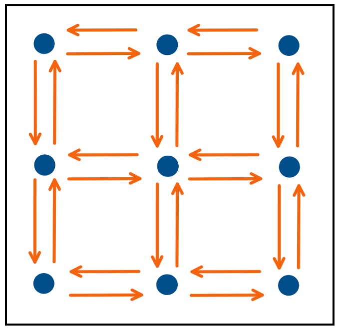

Fig. 1

If there is an obstacle, wall, or other robot in the next possible node position, then that node is no longer considered a possible option. Using the possible moves, the robots then adhere to the logic described in the figure below. This decision-making process is repeated until the entire map is covered. Additionally, the distance to the nearest uncovered cell is calculated using the Euclidean distance formula, which will later be described in section 3 of the report.

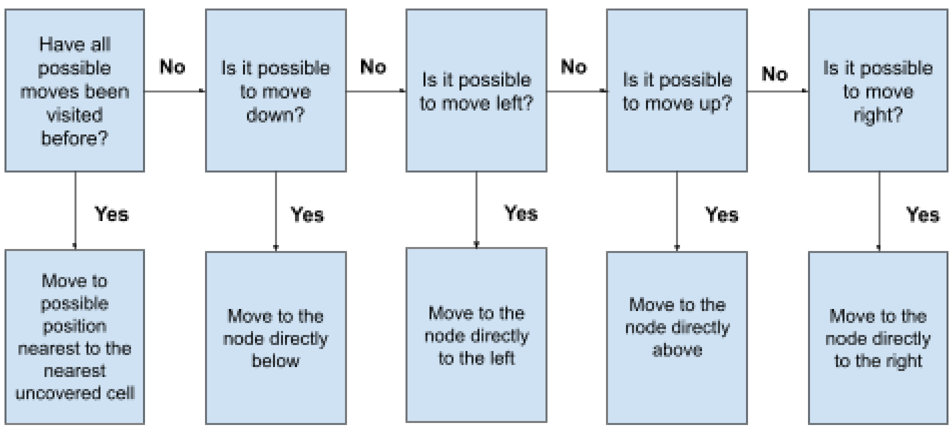

Fig. 2

2.4.2 Market-Based Model

Market-based models mimic a market economy in different ways and are usually used for robots to bid on tasks that they can perform. There will be a set of tasks that require specific skills and a set of robots with various or specific skills. Robots will always try to maximize their profit by reducing costs. A task which they do not have the best skillset to perform will cost them more. A market-based auction model allows robots to bid on tasks based on their interest, measured by their ability to perform a task[2]. The bid per robot i is shown in the following equation [8], where v_i and w_i calculate the difference between the robots' first and second tasks of interest. ϵ is a "complementary slackness," a variable to offset the profit and adjust for what each robot is willing to sacrifice.

$$\gamma_i = v_i - w_i + \epsilon \quad \textbf{Eq 2.4.2.1}$$

Because the robots enter a bidding contest, their bid will increase and the actual price they pay, p_{ij}, will increase as they compete. We can define j as a specific task that is part of a set of tasks S. For $\forall\, j \in S$, the continuous increase of the price the robots pay is described in the equation below[8].

$$p_i = p_i + \delta \qquad \textbf{Eq 2.4.2.2}$$

This will result in a discrepancy between their original budget for the task, u_{ij}, and the actual price they pay if they win the task. This difference is described as m_{ij} in the equations below[8].

$$m_{ij} = u_{ij} - p_j \qquad \textbf{Eq 2.4.2.3}$$

$$m_{ij} \geq max\{u_{ij} - p_{ij}\} - \epsilon \qquad \textbf{Eq 2.4.2.4}$$

Robots take information like this price discrepancy or profit margin to decide when they stop or continue bidding. If the cost of a task is greater than the reward from completing the task, then robots stop bidding and a winner of the auction is chosen. Most of the time, two robots cannot complete a single task at once for only one winner is chosen. Depending on the task, more robots can win the auction, but more robots do not mean more efficiency [2]. In general, the robot(s) with the biggest profit will win the auction and this is represented by the equation below[8].

$$i^* = argmax_{i \epsilon P(j)} \gamma_i \qquad \textbf{Eq 2.4.2.5}$$

When a robot loses an auction, the equations above are then adjusted so that the second choice task becomes the first choice. Similarly, their previous third choice becomes their second choice, and so on. This process continues to repeat until they win an auction. This change in the order of choices is represented in the equation below[8].

$$\left(j^{'}\right) = \{j |\, m_{ij} = m_{ij*}\},\ \forall j \neq j^* \qquad \textbf{Eq 2.4.2.6}$$

2.4.3 Potential-Field Model

Generally, in a potential-field model, a magnetic field is created by the environment where a robot is considered to be a particle navigating throughout the field. This magnetic field has both attractive and repulsive forces. In the case of the model described in this paper, nodes on the occupancy map that do not have an obstacle or a wall exert an attractive force. Similarly, nodes that do contain an obstacle or wall exert a repulsive force. As robots move closer to obstacles, the repulsive force from said obstacle increases. The result of these attractive and repulsive forces guide robots.

Robots can have position coordinates q(x, y, z) and an associated potential field U(q). The potential force F(q) is computed as[1][3][5][9]

$$F(q) = -\nabla U(q) \qquad \textbf{Eq 2.4.3.1}$$

All robots share the same shape and altitude (z = 0). Our focus is on N-wheeled mobile robots navigating within an environment, which contains M static obstacles.

Attractive Forces:

The attractive force[1][3][5][9] produced by the goal point g_i and acting on robot i at position q_i, where i=1...N, is:

$$F_a(q_i, g_i) = -\xi p(q_i, g_i)$$ **Eq 2.4.3.2**

where ξ is the positive scale factor

$p(q_i, g_i)$ is the distance between robot i and the target point g_i

$p(q_i, g_i) = \left|q_i - g_i\right|$ where |.|is the Euclidean distance.

Also, Each robot i has its own goal point g_i for i=1…….N.

Now the combination of Attractive Forces Vector is:

$$\begin{bmatrix} Fa(q1, g1) \\ Fa(q2, g1) \\ . \\ . \\ . \\ Fa(qN, gN) \end{bmatrix} = \begin{bmatrix} -\zeta p(q1, g1) \\ -\zeta p(q2g2) \\ . \\ . \\ . \\ -\zeta p(qN,\ gN) \end{bmatrix}$$ **Eq 2.4.3.3**

Repulsive Forces

Obstacles generate repulsive forces[1][3][5][9] when the robot is near, pushing it away from the danger zone. q_{oj} is the position of the obstacle j (j = 1, 2, ..., M) and $p(q_i, qo_j)$ is the distance between the robot i and obstacle j.

The equation of repulsive force generated by an obstacle j and applied to the robot i is given by:

$$F_{ro}(q_i, qo_j) = \begin{cases} \zeta\left(\frac{1}{p(q_i, qo_j)} - \frac{1}{po_j}\right)^2 \frac{1}{p^2(q_i, qo_j} \cdot \\ \nabla p(q_i, qo_j)\ if\ p(q_i, qo_j) \le po_j \\ 0\ if\ p(q_i, qo_j) > po_j \end{cases}$$ **Eq 2.4.3.4**

where po_j is the threshold radius of obstacle j.

If M obstacles,

$$F_{ro}(q_i) = \sum_{j=1}^{M} F_{ro}(q_i, qo_j) \quad \textbf{Eq 2.4.3.5}$$

Also, the repulsive force exerted on robot k by a neighboring robot i is

$$F_{rr}(q_i, q_k) = \begin{cases} \zeta\left(\frac{1}{p(q_i, q_k)} - \frac{1}{pr_k}\right)^2 \frac{1}{p^2(q_i, q_k)} \cdot \\ \nabla p(q_i, q_k)\ if\ p(q_i, q_k) \le pr_k \\ 0\ if\ p(q_i, q_k) > pr_k \end{cases} \quad \textbf{Eq 2.4.3.6}$$

$p(q_i, q_k)$ is the distance between robot i and robot k

pr_k is the threshold radius of robot k

Considering (N-1) robots modifies the above equation as :

$$F_{rr}(q_i) = \sum_{k=1, k\neq i}^{N} F_{rr}(q_i, q_k) \quad \textbf{Eq 2.4.3.7}$$

Now, Generally repulsive forces can be written as:

$$F_r(q_i) = F_{ro}(q_i) + F_{rr}(q_i) \quad \textbf{Eq 2.4.3.8}$$

Resultant forces

The robot particle experiences attractive forces from the target and repulsive forces from obstacles to prevent collisions. The resulting forces guide the robot towards its goal, based on the distances to the goal, obstacles, and other robots in the workspace.

Resultant forces applied to robot i is given by[1][3][5][9],

$$F(q_i) = F_a(q_i) + F_r(q_i) = F_a(q_i) + F_{ro}(q_i) + F_{rr}(q_i) \quad \textbf{Eq 2.4.3.9}$$

Now, gathering the resultant force of each robot in one system gives the matrix as follows:

$$\begin{bmatrix} F(q_1) \\ F(q_2) \\ . \\ . \\ . \\ F(q_N) \end{bmatrix} = \begin{bmatrix} F_a(q_1, g_1) + \sum_{j=1}^{M} F_{ro}(q_1, qo_j) + \sum_{k=1,k\neq 1}^{N} F_{rr}(q_1, qk) \\ F_a(q_2, g_2) + \sum_{j=1}^{M} F_{ro}(q_2, qo_j) + \sum_{k=1,k\neq 2}^{N} F_{rr}(q_2, qk) \\ . \\ . \\ . \\ F_a(q_N, g_N) + \sum_{j=1}^{M} F_{ro}(q_N, qo_j) + \sum_{k=1,k\neq N}^{N} F_{rr}(q_N, qk) \end{bmatrix} \quad \textbf{Eq 2.4.3.10}$$

The Matrix shown above helps in planning the path for a group of robots working together in an environment that has fixed obstacles while ensuring that the robots avoid collisions among themselves and with the static obstacles present in their workspace.

2.4.4 Market-Potential Model

Using the underlying concepts of a market-based model, a potential-field model, and the fundamental model previously explained, a second model was created in MATLAB. Similar to the fundamental model, this model is bounded by a choice of four moves, described in **Eq. 2.4.3.1**. Because of this constraint, in a solely market-based model, this would constrain the set of tasks S that each robot could perform or reach and not each robot would bid on the same tasks. They would, possibly repeatedly, be entering auctions without any competitors. Instead, this model has tasks place bids for robots to pick them. The "tasks," in this case, are the four nodes around the robot that the robot can move to. The bid that the task can make depends on its distance to a node that has not been disinfected/covered. The minimal the distance, the higher the bid. Taking equation M described in **Eq. 2.4.3.1** and the nearest uncovered node as n, the bids each possible move can offer based on distance, B_d^{-1}, is represented in the equation below.

$$B_d = \sqrt[2]{(M_x - n_x)^2 + (M_y - n_y)^2} = \left(\begin{bmatrix} x_i - n_x \\ x_i - 1 - n_x \\ x_i - n_x \\ x_i + 1 - n_x \end{bmatrix}^2 + \begin{bmatrix} y_i - 1 - n_y \\ y_i - n_y \\ y_i + 1 - n_y \\ y_i - n_y \end{bmatrix}^2 \right)^{\frac{1}{2}} \qquad \textbf{Eq 2.4.3.1}$$

This distance-based bid alone is not enough to prevent robots from colliding with each other and into walls or obstacles. Therefore, the bid is substantially lowered if that move contains an obstacle. This is accounted for with the variable B_o. Additionally, it is also lowered depending on the amount of time that possible move has been visited by introducing a dynamic penalty. This dynamic penalty is represented as B_p in the following equation and later explained in section 3.1 of the report. Overall, the total bid, B, of each move is represented as follows.

$$B = B_d^{-1} - B_p - B_o \qquad \textbf{Eq 2.4.3.2}$$

The highest bidding possible position wins and the robot moves to that position. Although at a surface level this might seem solely a market-based model, this strategy also incorporates the ideas of a potential-field model. Because of the nature of the possible positions bidding for the robot, the components of what makes a bid higher or lower can be seen as attractive and repulsive forces, respectively. For example, B_d^{-1} is an attractive force, while B_p and B_o are repulsive forces. Having these attractive and repulsive forces creates a field navigating each robot.

3. Theoretical Analysis

3.1 Dynamic Penalty Controller

When there are no unvisited nodes in the immediate neighbors of the robots, we observe that the robots are inclined to move in a loop or revisit an already covered node which reduces the coverage efficiency and takes more time to convergence/ cover all the nodes. To prevent this we introduce a penalty in our models to increase the coverage efficiency by adding a penalty when the robot revisits a particular node in the map. Due to the penalty, the robot does not prefer revisiting the nodes when all the nearby nodes are already visited and searching for unvisited nodes.

Let P be the Penalty function.

- Static/Constant Penalty: The penalty remains the same regardless of the count of revisits

$$P_{static} = P \qquad \textbf{Eqn. 3.1.1}$$

After adding this penalty, we came across a situation where the robot was still stuck in a loop when there was a semi-closed area. It is due to the static penalty and the penalty is the same, no matter how many times the robot revisits the same node. So we introduced a dynamic penalty to overcome the problem.

Dynamic Penalty:

$$P_{dynamic}(t) = \int_0^t P\, dt^{|} \qquad \textbf{Eqn. 3.1.2}$$

represents the cumulative dynamic penalty over time

Where $P_{dynamic}(t)$ is accumulated dynamic penalty up to time t

P is increased in penalty rate per unit time

The penalty is not predetermined, instead, it increases in a stepwise manner depending on the number of times the robot reaches a certain node in this case. In particular, we keep the number of visits the robot holds the node within this instance and then multiply that by the baseline penalty to raise the penalty in proportion to the number of holds the robot has on the node.

Now to express $P_{dynamic}(t)$ in terms of no. of revisits (N),

Using the relationship between N & t:

$$N = \frac{1}{T}\int_0^t dt^{|} \qquad \textbf{Eqn. 3.1.3}$$

where T is the time interval between revisits

By rearranging,

$$dt = T . dN$$ **Eqn. 3.1.4**

Substituting in ( )

$$P_{dynamic}(N) = \int_0^N P . T dN^|$$ **Eqn. 3.1.5**

The above equation shows a cumulative dynamic penalty against the number of visitors (N), time between visits (T), and rate of penalty increase (P).

3.2 Coverage Efficiency

Coverage Efficiency ($E_{total}(t)$) is a measure of the aggregate efficiency with which a multi-robotic architecture covers and maps an area. As the unoccupied cells are being marked as Occupied by robots as soon as they visit them, this stored info can be retrieved and used to calculate the Coverage Efficiency[6].
It is expressed as a ratio between the total occupied area and the free space on the map at any given point in time[7].
Let ($A_i(t)$) be the areas covered by an individual robot which is product of number of visited cells ($V_i(t)$) and the size of a cell
As the occupancy map is a 2D grid, the size of a cell is C^2.Therefore,

$$(A_i(t)) = (V_i(t)) . C^2$$ **Eqn. 3.2.1**

Now combining all the individual coverage areas gives the total coverage Area A_{cov}

$$A_{cov} = \sum_{i=1}^{N} A_i(t)$$ **Eqn. 3.2.2**

Also, Total Free area is the summation of all the unoccupied cells which is given by the product of the number of Free cells $F_i(t)$ and their size i.e.,

$$A_{free} = (F_i(t)) . C^2$$ **Eqn. 3.2.3**

3.2.1 Overall Efficiency for all the Robots

As,

$$Coverage\ efficiency = \frac{Total\ Coverage\ Area}{Total\ Free\ Area}$$ **Eqn. 3.2.4**

From eqn (**3.2.3**) & (**3.2.4**)

$$= \frac{A_{cov}}{A_{free}} = \frac{\sum_{i=1}^{N} A_i(t)}{(F_i(t)) \, . \, C^2}$$ **Eqn. 3.2.5**

$$\text{So, } Overall\ Coverage\ efficiency = \frac{\sum_{i=1}^{N} A_i(t)}{(F_i(t)) \, . \, C^2}$$ **Eqn. 3.2.6**

This ratio is an indicator of how efficiently the multi-robot system traverses and disinfects the environment[6][7].

3.2.2 Individual Robot Coverage Efficiency

Individual Robot Coverage Efficiency is the percentage contribution of an individual robot to the total coverage achieved by the robotic team. It is given by the ratio of the area of coverage of that robot to the entire free space[6][7].

$$Individual\ Robot\ Coverage\ efficiency = \frac{Individual\ Robot\ Coverage\ Area}{Total\ Free\ Area}$$ **Eqn. 3.2.2.1**

From eqn (**3.2.3**) & (**3.2.2.1**)

$$\frac{A_i(t)}{A_{free}} = \frac{(V_i(t)) \, . \, C^2}{(F_i(t)) \, . \, C^2} = \frac{(V_i(t))}{(F_i(t))}$$ **Eqn. 3.2.2.2**

$$Individual\ Robot\ Coverage\ efficiency = \frac{(V_i(t))}{(F_i(t))}$$ **Eqn. 3.2.2.3**

It is more directed towards how much of an impact one particular robot makes for the entire coverage.

3.3 Collision Avoidance

As explained in section 2 of the report, robots are aware of the areas on the map that correspond to unreachable areas and areas that need to be covered/disinfected. They are also aware of the position of other robots. Because they are aware of this information, they will never move toward a node where another robot is positioned. Therefore, there is collision avoidance. This collision avoidance can be proved via a distance analysis, explained below.

3.3.1 Distance Analysis

The distance between one robot i to another robot j at each point in time can be calculated via the Euclidean distance formula[4]:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$ **Eqn. 3.3.1.1**

This formula is derived from the Pythagoras theorem formula to calculate the hypotenuse of a triangle when you know the lengths of the other two sides of the triangle, as described in the equation below.

$$A^2 + B^2 = C^2$$ **Eqn. 3.3.1.2**

Where C represents the hypotenuse, B represents one shorter side, and A represents another. In this case, A can represent the x-directional distance between two robots, B can represent the y-directional distance, and C can represent the total distance. If we solve for C and plug in our variables, we end up with the Euclidean distance formula.

Theoretically, if the Euclidean distance between two robots, robot i and robot j, is greater than zero at all points in time, $d_{ij} > 0$, then those two robots are never at the same location at the same time. They may be in the same x-coordinate or the same y-coordinate, but not both at once. Meaning, they do not collide.

Now, if we had a multi-robot system of 5 robots and take all possible combinations of robots {1,2}, {1,3},{1,4},{1,5},{2,3},{2,4},{2,5},{3,4},{3,5}, and {4,5} we can form the equation below. If that equation is true at all times, then none of the 5 robots ever collide with each other at any time.

$$min(d_{12}, d_{13}, d_{23}) > 0$$ **Eqn. 3.3.1.3**

4. Validation

For the following sections of the report, two models were simulated in MATLAB. These are the models that were previously described in sections 2.4.1 and 2.4.4 of the report. The following graphs describe the paths taken by each robot in each simulation to achieve the complete coverage indicated in the second subplot. These graphs are the end snippets of animations, submitted separately. It is worth noting that although mostly our own, the code for these simulations was aided by chatGPT. This code is also submitted separately.

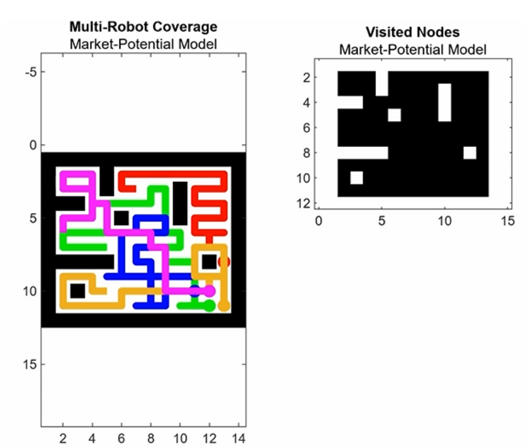

Fig. 3

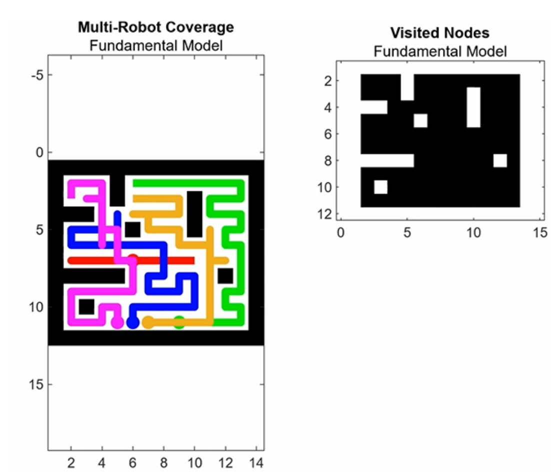

Fig. 4

4.1 Coverage Efficiency

4.1.1 Overall Efficiency for all the Robots

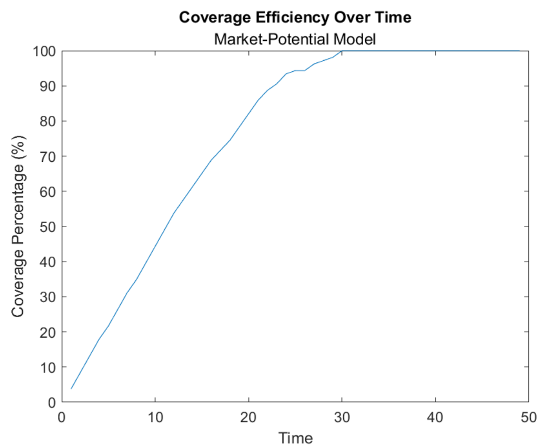

Fig. 5

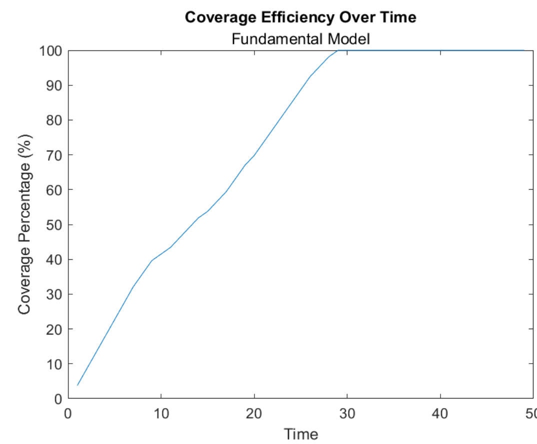

Fig. 6

After the simulation was completed, we found the coverage percentage to be 100 percent in both of our simulations, which means that our robots moved to all the free nodes. Now, we plotted a graph between coverage percentage and time to understand how the coverage percentage reacts as time passes. From [Fig. 5](#) we can see that the coverage percentage in the Market-Potential Model is increasing rapidly at the start but as it nears complete coverage, it slows down, so we can say that as the free nodes decrease, the robots are revisiting the nodes to find an unvisited node which makes the change in coverage efficiency slower. From [Fig. 6](#)(Fundamental Model) the decrease in the rate of change of coverage percentage doesn't change that drastically. In this model, the coverage efficiency is increasing at the same rate which shows that the robots are revisiting the nodes more frequently throughout the coverage unlike in the Market-Potential model where the revisiting of nodes increases in the end due to insufficient free nodes.

4.1.2 Individual Robot Coverage

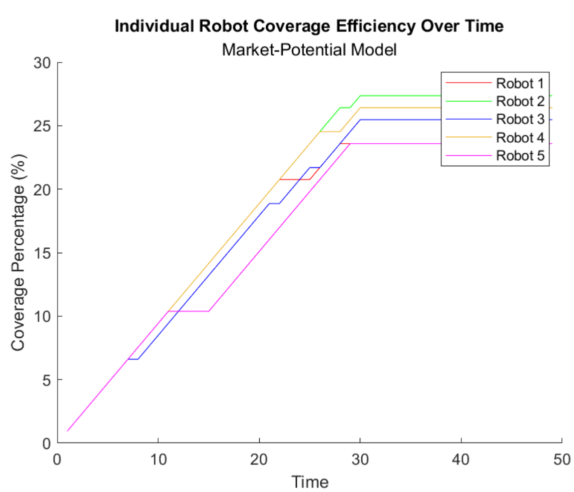

Fig. 7

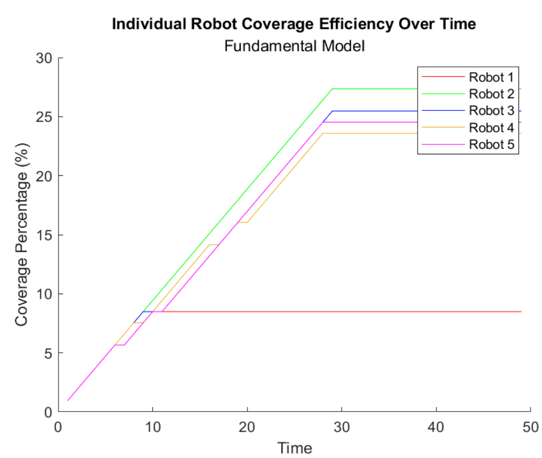

Fig. 8

As we plotted the overall efficiencies of the models in section 4.1.1, we now try to understand the coverage efficiencies of each robot. To do this we store the nodes covered by each robot in each node and calculate its percentage from the total free nodes and plot it with respect to time. From this we get **Fig. 7** and **Fig. 8**. When we add the coverage percentage of each robot to find the total coverage percentage, we get more than 100 percent for both of the models. This is due to the robots revisiting the nodes. When the robot has no free nodes in its four cardinal adjacent nodes, it revisits a node to reach unvisited nodes in the map according to the controller model. As both the models allow revisits, the robots revisit a visited node to reach an unvisited node. As we compare the total coverage percentage of all the robots, we can see that the fundamental model has a lower coverage percentage when compared to the Market-Potential model. This shows that in the fundamental model, there will be fewer revisits which makes the coverage percentage of the robots less. So from this, we can say that the Market-Potential model has more revisits and increased distances traveled compared to the fundamental model.

4.2 Collision Avoidance

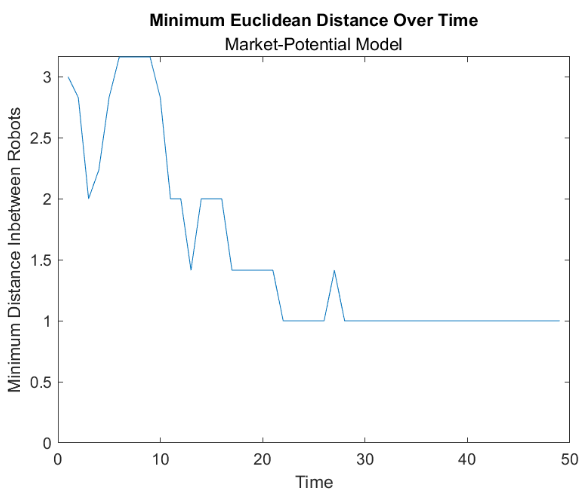

Fig. 9

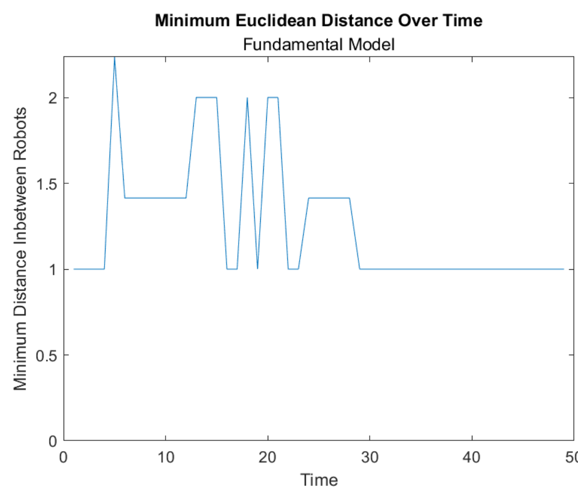

Fig. 10

Based on the graphs above, the smallest distance between robots for both models never reaches a value below 1. In other words, it is never a value of 0. As explained in section 3.3 of the report, if it never reaches 0 then the robots are never in the same position and never collide. Both of the simulated models avoid collision.

Notable from **Fig. 9** and **Fig. 10** is how often robots are 1 node away from each other. The distance between robots for the Market-Potential model seems to be gradually decreasing as the simulation approaches full coverage. On the other hand, robots are generally closer to each other for the Fundamental model.

5. Conclusion

We tested the ‘Market-Potential Model’ and ‘Fundamental Model’ for coverage efficiency over time, individual robot coverage, and collision avoidance. After comparing the results achieved, we concluded that both the models achieved 100% coverage after the 30-second mark and there was not much of a difference. But the robot revisits the nodes more in the fundamental model throughout the coverage and the revisits increase at the end of the simulation for the ‘Market-Potential Model’. While testing the individual robot coverage, we can see from graphs **Fig. 7** and **Fig. 8** that the time taken to achieve maximum individual coverage was also very similar in both models. And the total distance covered by each robot will be higher for the ‘Market-Potential Model’. From the graphs **Fig. 9** and **Fig. 10**, the Collision Avoidance graph with minimum Euclidean distance also shows similarities between both models. Also, the robots are generally closer to each other for the Fundamental model. After analyzing all the results, we conclude that the ‘Fundamental Model’ is as efficient as the ‘Market-Potential Model’ and can give similar results and each has its own advantages and disadvantages when we are using it for Disinfection with Multi-Robot Coordination.

6. Contributions

Section	Contributions
Section 2: Mathematical Models 2.1 Goal 2.2 Assumptions and Constraints 2.3 Parameters 2.4 Models 2.4.1 Fundamental Model 2.4.2 Market-Based Model 2.4.3 Potential-Field Model 2.4.4 Market-Potential Model	Overall everyone contributed to this section in different ways. For the most part: Sections 2.1-2.3 by Vidit and edited by Siva. Sections 2.4-2.4.1 by Anney and edited by Vidit. Section 2.4.2 by Jhati and edited by Anney. Section 2.4.3 by Siva and edited by Jhati. Section 2.4.3 was a combination of everyone due to its nature and was edited by Anney.
Section 3: Theoretical Analysis 3.1 Dynamic Penalty Controller 3.2 Coverage Efficiency 3.2.1 Overall Efficiency for all the Robots 3.2.2 Individual Robot Coverage Efficiency 3.3 Collision Avoidance	Again everyone contributed. It could more generally be broken down: Section 3.1 by Siva and Jhati Section 3.2- 3.2.1 by Jhati and Siva Section 3.2.2 by Vidit Section 3.3 by Anney Everyone edited sections.
Section 4: Validation 4.1 Coverage Efficiency 4.1.1 Overall Efficiency for all the Robots 4.1.2 Individual Robot Coverage 4.2 Collision Avoidance	Similarly, everyone contributed. For the most part: Section 4.1 by Anney and edited by Vidit Section 4.1.1 by Siva and edited by Jhati Section 4.1.2 by Jhati and edited by Siva Section 4.2 by Anney and edited by Vidit

7. References

[1] A.Ahmed, Alaa & Abdalla, Turki & Abed, Ali. (2015). Path Planning of Mobile Robot by using Modified Optimized Potential Field Method. International Journal of Computer Applications. 113. 6-10. 10.5120/19812-1614.

[2] Dias, M. & Zlot, Robert & Kalra, Nidhi & Stentz, Anthony. (2006). Market-Based Multirobot Coordination: A Survey and Analysis. Proceedings of the IEEE. 94. 1257 - 1270. 10.1109/JPROC.2006.876939.

[3] F. Matoui, B. Boussaid and M. N. Abdelkrim, "Local minimum solution for the potential field method in multiple robot motion planning task," *2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, Monastir, Tunisia, 2015, pp. 452-457, doi: 10.1109/STA.2015.7505223.

[4] *Euclidean distance formula - derivation, examples*. Cuemath. (n.d.). https://www.cuemath.com/euclidean-distance-formula/

[5] Jiubo, Sun & Liu, Guoliang & Tian, Guohui & Zhang, Jianhua. (2019). Smart Obstacle Avoidance Using a Danger Index for a Dynamic Environment. Applied Sciences. 9. 1589. 10.3390/app9081589.

[6] Karapetyan, Nare & Benson, Kelly & McKinney, Chris & Taslakian, Perouz & Rekleitis, Ioannis. (2017). Efficient Multi-Robot Coverage of a Known Environment. 1846-1852. 10.1109/IROS.2017.8206000.

[7] Kapoutsis, A.C., Chatzichristofis, S.A. & Kosmatopoulos, E.B. DARP: Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning. J Intell Robot Syst 86, 663–680 (2017). https://doi.org/10.1007/s10846-016-0461-x

[8] Liu, Lantao & Shell, Dylan. (2013). Optimal Market-based Multi-Robot Task Allocation via Strategic Pricing. 10.15607/RSS.2013.IX.033.

[9] Matoui, Fethi & Boussaid, B. & Metoui, Brahim & Bel Haj Frej, Ghazi & Mohamed Naceur, Abdelkrim. (2017). Path planning of a group of robots with potential field approach: decentralized architecture. IFAC-PapersOnLine. 50. 11473-11478. 10.1016/j.ifacol.2017.08.1822.