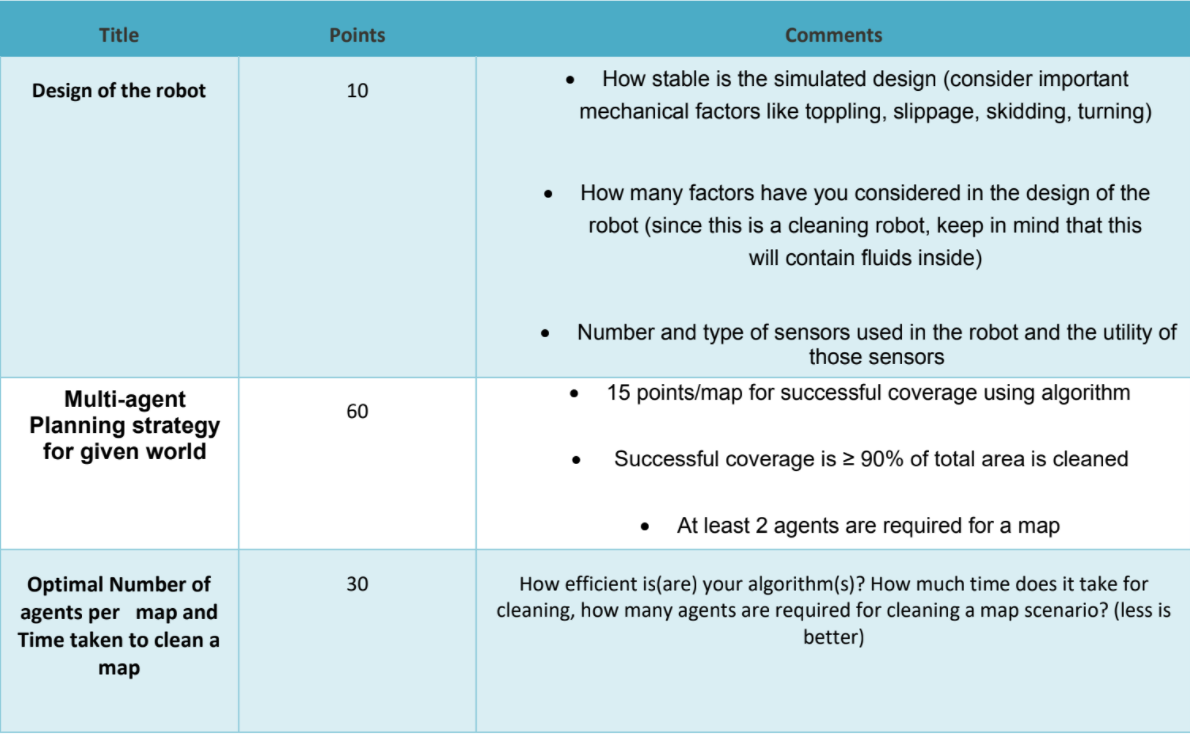
Complete Coverage Path Planning Module

Team TechKnights’ Alliance - AIITRA ‘21

# Scoring criteria :



# 

# Literature Review : 9/21 Reviewed ;

| 1 | Complete coverage path planning in an agricultural environment | [Link](https://drive.google.com/file/d/18PU7k_L8nqVIm02WK_cdjNtY9iwJlXQc/view) |  |
| --- | --- | --- | --- |
| 2 | An algorithm for multi-robot collision-free navigation based on shortest distance | [Link](https://drive.google.com/file/d/1rmC-xbOLRTyGVMFbZ9ya3ObkDAYv8D5E/view?usp=sharing) | - - - |
| 3 | On Complete Coverage Path Planning Algorithms for Non-holonomic Mobile Robots: Survey and Challenges | [Link](https://drive.google.com/file/d/12Odn_FzMQghhtRyJxlFhQL-bFioIfZHO/view?usp=sharing) |  |
| 4 | A Bezier curve based path planning in a multi-agent robot soccer system without  violating the acceleration limits | [Link](https://drive.google.com/file/d/19HBgZRoZGjzGBRth0Gc5R6GiVSq_Eq5w/view?usp=sharing) | - - - |
| 5 | Efficient approaches for multi-agent planning | [Link](https://drive.google.com/file/d/17QfFyFGGtlL_7qMAeNMUS2oSrjgwA9BX/view?usp=sharing) |  |
| 6 | Multi-Agent Collaborative Path Planning Based on Staying Alive Policy | [Link](https://drive.google.com/file/d/1f1kXRLiy4xE6whoLHEati9hk_DsVjs77/view?usp=sharing) |  |
| 7 | Multi-agent energy-efficient coverage path planning | [Link](https://drive.google.com/file/d/12ib2FDB1gB64AHzwvlBdq58anYvhlAgg/view?usp=sharing) | - - - |
| 8 | Multi-objective multi-robot path planning in continuous environment  using an enhanced genetic algorithm | [Link](https://drive.google.com/file/d/1MuhMZ2ckIWzj18TvPmENb6mQbkY7nZs2/view?usp=sharing) | Done |
| 9 | Multi-Robot Path Planning | [Link](https://drive.google.com/file/d/1UZK5-OEEYzZGnyz3U_gf87ZNAxcDcB4W/view?usp=sharing) |  |
| 10 | Multi-Robot-Path-Planning and Motion Coordination | [Link](https://drive.google.com/file/d/1uTTeCylCZ2Wi6EUE1mkKV7fqIRXb8Da3/view?usp=sharing) |  |
| 11 | Towards Designing Multi-Agent Coverage Systems Capable of Anticipation and Tight Coordination with Detailed Environmental and Perception Models | [Link](https://drive.google.com/file/d/1xJzsOXB4Qe0zYw33zR6dTm_A6ZdHgLqg/view?usp=sharing) |  |
| 12 | More Planning Problems and Methods | [Link](https://drive.google.com/file/d/1HCv4f8SwMGF4S_I9K-vLgduvZAERCuQP/view?usp=sharing) | Done |
| 13 | A Comparison of Path Planning Algorithms for Robotic Vacuum Cleaners | [Link](https://drive.google.com/file/d/1G9tXORchwSnqFYlHrYBzRIBvB_pjbvlH/view?usp=sharing) |  |
| 14 | A Survey on Coverage Path Planning for Robotics | [Link](https://drive.google.com/file/d/1GjCsfilyv0Nffm1swnMnh15KT7OeZJTv/view?usp=sharing) |  |
| 15 | Autonomous Positioning and Full Traversal Path Planning Technology | [Link](https://drive.google.com/file/d/1-sh7tfWpWqL2xpUnXC1NCRxwkCVw39JZ/view?usp=sharing) |  |
| 16 | Bug Algorithms and Path Planning | [Link](https://drive.google.com/file/d/1HDQIM5wMfhtt2OfRxYxRODG9B1-juWgc/view?usp=sharing) | Done |
| 17 | e\_star An Online Coverage Path Planning Algorithm | [Link](https://drive.google.com/file/d/1xzgPeLmTtJpaLKvAbvmhdrYSWjRBZ0qW/view?usp=sharing) |  |
| 18 | Path\_planning\_algorithm\_development\_for\_autonomous | [Link](https://drive.google.com/file/d/16mnkZs_RjvikZo9lUe4DIK-QnbqZAe01/view?usp=sharing) |  |
| 19 | Robovac (Autonomous Robotic Vacuum Cleaner) | [Link](https://drive.google.com/file/d/1nMYL5QKW1JFZpQgfeSH1pCRoOzewE_RS/view?usp=sharing) | - - - |
| 20 | Roomba Algorithms and Visualization | [Link](https://drive.google.com/file/d/1sZC6T_-5LczIHROiSYhebqCMWBrA9Kk8/view?usp=sharing) | - - - |
| 21 | Scheduling and Motion Planning of iRobot Roomba | [Link](https://drive.google.com/file/d/1PPs16rJ5dIlB2sp8lSmh4zBwg-y_KJXc/view?usp=sharing) | - - - |

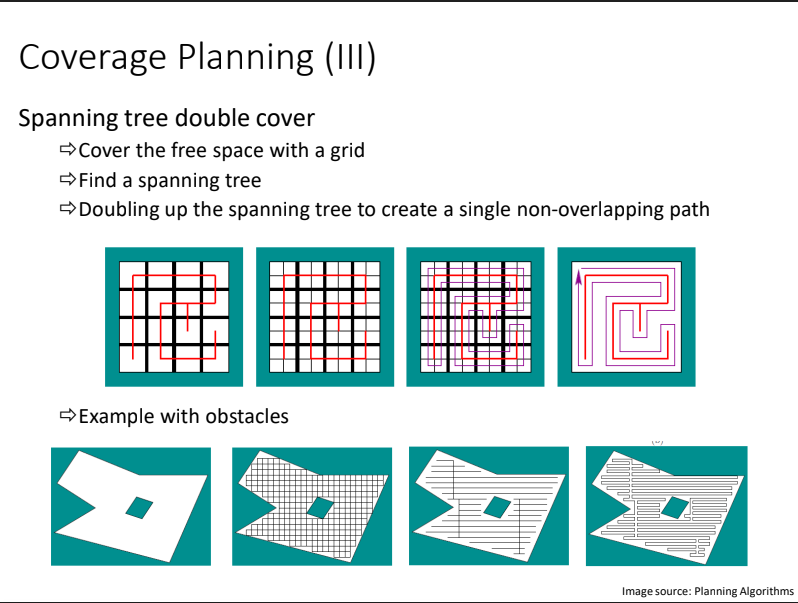
**#12 Coverage Planning**

An important problem in practice is coverage planning. What is the main goal here?

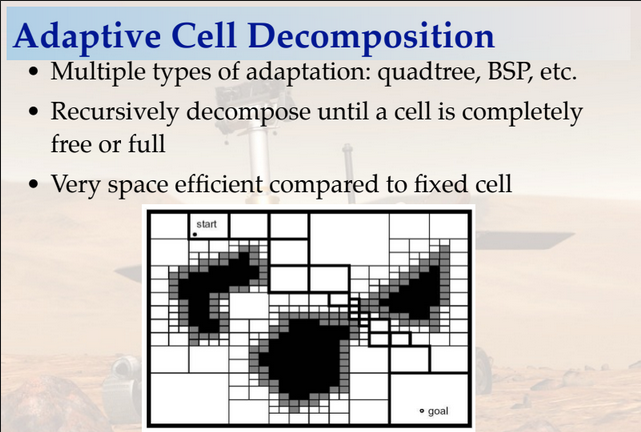
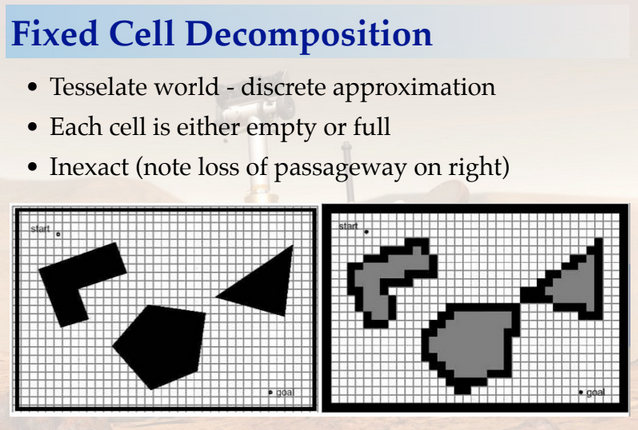
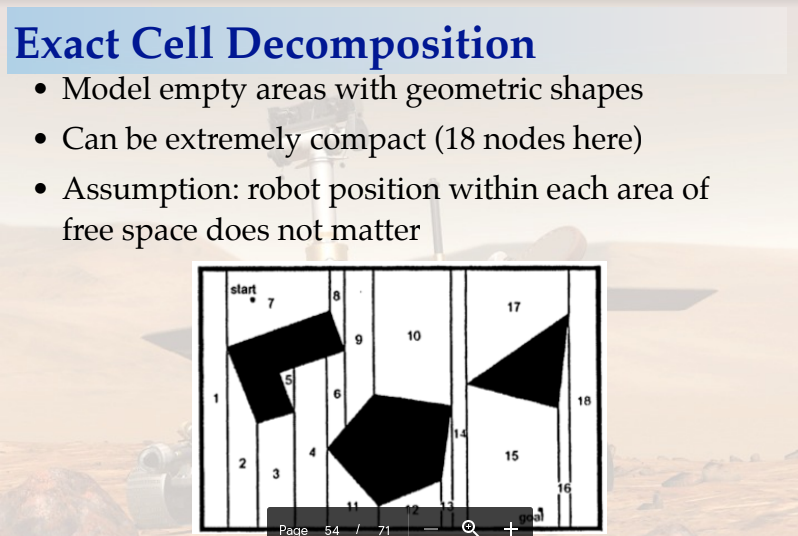
* Cover (using limited footprint) the space with least amount of total travel
* A second objective can be minimizing turns, which takes time

Boustrophedon decomposition

* One way to do coverage planning is intuitive: move back and forth!
* Basically, do a (special) vertical cell decomposition
* Then, do boustrophedon paths in each cell
* Issue: when cells split, may need to travel back and waste time
* Can be resolved using spanning tree double covering



**#16 Decomposition Pg 53 - 56**



Features:

* Obstacle resolution: 1x1
* Cell resolution: 0.5x0.5
* Path resolution : 1x1 (since the path obtained is a closed loop!)
* Resolution can be refined if needed. Higher computation power is needed for the path optimization
* Simulation CSV Headers added in Templates

Workflow:

* Downloaded DARP-Python
* Tweaked code to print the path
* Tweaked code to print number of straights along with number of corners
* Developed [Map Launch Commands](https://drive.google.com/file/d/1quld5yHNhJd_VEJCXEZPjvi7RCRvD8V9/view?usp=sharing) for Path Optimization
* Run simulations for different Bot positions. Store number of straights and turns. Use them to compute time
* For a given number of bots, find the position configuration for minimum time taken (store other values to display mean and variance of time taken)
* Find for number of robots ranging from 2 to 10-15 (such that cost-time intersection is near middle of graph)
* Script to automate simulation.

# Points taken into consideration / to be mentioned :

* **Size of bot vs cell resolution :** cell resolution should be equal to the size of robot for the optimal cleaning calculation mentioned in the proposal.

For a 50m\*50m map, taking the resolution to be 0.5\*0.5 gives us a 100\*100 grid. Running the algorithm multiple times for multiple configurations on a 100\*100 grid is a task in itself.

Taking a 0.35\*0.35 resolution, we would have obtained a 145\*145 grid.which would not be possible on the limited computational power that we have. Hence we have used a resolution of 0.5\*0.5m for the path planning purposes while limiting the robot size to 0.35\*0.35m as per the mentioned specifications

* **Initial robot configuration setup :**  For robots placed close by, it takes a much larger number of iterations to get to the solution. Due to limited computational power, we have only considered cases where the robots are ‘atleast’ 20 meters apart.

The Data for the number of combinations taken into considerations and the number of left out due to

this constraint has also been recorded to give a clear idea of the amount of data being used to conclude our result.

With higher computational power, we can find optimal paths for all possible combinations of initial positions of the robots.