

Hierarchical Coverage Path Planning for a Multi-Modal Robot Exploring Disconnected Regions

Sebastian Stelter¹, Daniel Kuan Io U², Sabyasachi Mondal¹, Leonard
Felicetti¹, and Saurabh Upadhyay¹ * **

¹ Centre for Autonomous and Cyber-Physical Systems, Faculty of Engineering and
Applied Sciences, Cranfield University, Cranfield MK43 0AL, United Kingdom.
{Sebastian.Stelter, Saurabh.Upadhyay, Sabyasachi.Mondal,
Leonard.Felicetti}@cranfield.ac.uk

² Department of Engineering Mathematics, University of Bristol, Bristol BS8 1QU,
United Kingdom, United Kingdom. danieluuki@gmail.com

Abstract. We present a near-optimal Coverage Path Planning (CPP) approach for multi-modal robots in complex environments with multiple arbitrarily shaped disconnected regions. The problem arises in ground coverage missions conducted in unstructured terrains, such as planetary exploration or search and rescue missions, where safe regions are disconnected by areas of high slope that a robot with a single locomotion modality cannot traverse. A multi-modal robot can switch between different locomotion modalities (e.g. driving and flying) to safely navigate these challenging environments while ensuring complete coverage. The proposed method identifies both traversable and non-traversable areas based on Digital Elevation Model (DEM) meshes. The problem is then formulated as a hierarchical Traveling Salesman Problem (TSP) and solved using Mixed-Integer Linear Programming (MILP). The proposed approach is evaluated on 500 randomly generated maps and four real-world simulation scenarios constructed from Martian terrain DEM data.

1 Introduction

Coverage Path Planning (CPP) is a fundamental problem in the context of modern robotics and automation, finding use in a variety of areas such as agri-

* Some of the preliminary work was supported by the University of Bristol internal funding “Post-doctoral research price 2021”.

** This version of the contribution has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: <https://doi.org/10.1007/978-3-032-01486-3>. Use of this Accepted Version is subject to the publisher’s Accepted Manuscript terms of use <https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms>

cultural monitoring, factory inspections or surface treatments [1]. Especially in the fields of planetary exploration, and search and rescue applications, unstructured terrains pose an additional challenge [2]. These environments regularly contain many arbitrarily shaped obstacles such as cliffs, boulders or soft sands. In many cases, these obstacles fragment the area into multiple disconnected safe regions. Unmanned Ground Vehicles (UGVs) are therefore incapable of covering such a scenario, as there is no safe way for them to traverse between these disconnected regions. While Unmanned Aerial Vehicles (UAVs) could avoid obstacles by flying over them and thus traverse between regions, they have some drawbacks compared to UGVs in terms of limited battery power and payload restrictions. Further, UAVs are limited to aerial coverage and thus require environments that are suitable for flight. Missions involving sample collection or require lower vantage points are challenging for UAVs.

Multi-modal mobility robot platforms that are capable of switching between different modes of locomotion can be utilized to profit from the advantages of both platforms [3] [4]. While these robots can utilize their primary modality (e.g. driving or walking) for detailed ground coverage, obstacles can be avoided by switching to a secondary modality (e.g. jumping or flying).

These characteristics make multi-modal robots potential candidates for CPP tasks in unstructured terrain with multiple disconnected regions. In this work, we focus on coverage path planning in an area of disconnected safe regions using a multi-modal mobility robot. We call this problem Multi-Modal Multiple Disconnected Region CPP (M3DR-CPP) problem.

Contributions:

- A near-optimal hierarchical Mixed-Integer Linear Programming (MILP) based solution to the M3DR-CPP problem.
- Validation of the proposed approach in realistic high-fidelity 2.5D environments

The remainder of this paper is organized as follows: In Section 2, related work in the field of CPP is presented. Section 3 describes the M3DR-CPP problem this work aims to solve. Section 4 presents our proposed TSP based method to solve the M3DR-CPP problem and Section 5 presents and discusses the results of our work. Section 6 gives a conclusion and an outlook on future work.

2 Related Work

This section gives an overview of the existing approaches for CPP in multi-region scenarios (Section 2.1), and CPP with terrain considerations (Section 2.2).

2.1 Multi-Region CPP

Several approaches have been proposed to solve the CPP problem in scenarios with multiple disconnected regions, primarily in the context of UAVs. One common approach to this problem utilizes the Boustrophedon algorithm [5] to

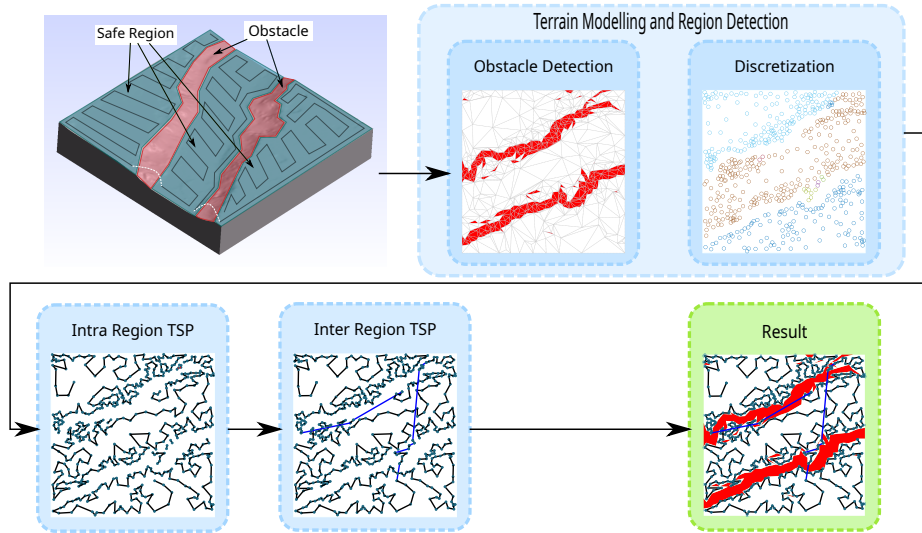


Fig. 1: Illustration of the M3DR-CPP problem and our proposed approach. The robot must cover all safe regions (blue) using its primary modality and cross obstacle regions (red) using its secondary modality to achieve full coverage. An example path is shown in the top left with use of the primary modality in black and use of the secondary modality in white. The proposed approach detects obstacles and discretizes each disconnected region into waypoints. A TSP is solved to generate the path for each region and to connect all regions to generate a full coverage path.

generate a back-and-forth path within each region. The individual region paths are then connected using a variety of different approaches, such as simulated annealing [6], generalized large neighbourhood search [7], genetic algorithms [8] or nearest neighbour heuristics [9]. Some approaches calculate multiple Boustrophedon paths with different orientations in each cell and select a candidate based on the inter-region path, resulting in potentially shorter overall paths. The regions are then connected using methods such as k-nearest neighbour [9] or 2-opt [10].

One method uses Ant Colony Optimization (ACO) to determine the order of regions [11]. Sensor footprint sweeps are then calculated to fill each region and connected using a second ACO formulation. Obstacles are avoided by flying around or over them. The sensor footprint sweeps approach has also been used in a single non-convex region scenario by defining the connecting of the sweeps as a generalized TSP [12]. Another approach uses a hyperheuristic based on a neural network which learns the heuristic for ACO to calculate the coverage path for each robot in a multi-robot scenario [13]. One method also uses ACO to calculate the region visiting order for each robot in a multi-robot scenario, but uses spanning tree coverage to calculate the intra-region paths [14].

Finally, one approach uses self-organizing maps to find the ideal region order [15]. Intra-region paths are then calculated by using concentric shrinking rings. Each ring is discretized into waypoints and then connected to the nearest ring based on waypoint distance. This approach guarantees complete coverage, even in non-convex regions, but does so at the cost of calculation time and coverage path length.

With exception of the last approach, all works in this Section assume a set of convex regions without obstacles, which makes them impractical for use in the M3DR-CPP scenario. Further, these approaches have been developed for UAVs and therefore many of them allow the robot to overshoot the region boundaries, which would lead to collisions and potentially damage to the robot in a ground based scenario. Finally, the approaches in this Section consider a 2D environment and do not take the structure of the terrain into account.

2.2 Terrain based CPP

While multi region CPP approaches generally do not consider terrain information, some single region approaches utilize DEMs to generate safer and more efficient coverage paths. One approach uses terrain data to classify areas as flat, sloped, or unsafe and then generates path segments that naturally follow the terrain slope to avoid steep angles [13]. The path segments are then connected using a genetic algorithm. Another work uses an aerial scout robot to identify the lowest cost path to a target point based on aerial imaging [16]. The approach considers static obstacles and different terrain traversal costs. Another approach considers slopes to detect unsafe areas before calculating a coverage path through the safe areas using spiral spanning tree coverage [17]. This work recognizes that obstacles may separate the area into multiple disconnected regions, but only covers the largest detected region. Finally, one approach detects different terrain types from aerial images using a neural network and uses this information to assign balanced coverage tasks to multiple robots [18].

3 Problem Description

The problem considers a triangulated terrain based on DEM data of the target area. The terrain contains slopes that are too steep for the robot to traverse while using the primary motion modality. We refer to these sloped areas as obstacles. Regions that can be traversed safely in the primary motion modality are called safe regions. The obstacles may entirely separate safe regions from each other, making it impossible to travel between them in the primary modality. The robot can switch to a secondary motion modality which allows it to traverse obstacles, but cannot perform its coverage task while using the secondary modality. It is assumed that the robot sensors can cover the entirety of each triangle in the terrain from the triangle incenter point and that the robot can turn in place. Safe regions that contain only one node are excluded, as CPP in these regions would be trivial.

The M3DR-CPP problem consists of two subproblems. First, given the DEM of the target, safe regions and obstacles need to be identified and discretized into nodes. Second, given a set of regions with nodes, a coverage path that passes through all nodes in all regions needs to be found while minimizing the path length.

4 Proposed M3DR-CPP Method

We propose an algorithm that solves the M3DR-CPP problem in three steps as shown in Fig. 1. In the first step, we model a triangular terrain mesh from the DEM data and detect obstacles and safe regions. In each safe region, the incenter of each triangle is calculated. The resulting incenter points are then used to formulate a hierarchical open-loop TSP problem in step two and three. In the second step, MILP is used to solve a TSP within each disconnected region using all incenter points in that region. As MILP guarantees optimality [19], this step will always generate the optimal path through each disconnected region. In the third step, the endpoints of the paths generated in the second step are used to formulate and solve another TSP in the same manner to connect the intra-region paths. The details of each step are as follows:

4.1 Terrain Modelling and Region Detection

We use the Delaunay triangulation [20] to model the 2.5 D terrain from the given DEM terrain data. The slope of each triangle is calculated as

$$\theta = \arccos n_z. \quad (1)$$

where n_z is the z component of the triangle's normal vector. A user-defined slope threshold $\bar{\theta}$ defines the maximum allowed slope in safe regions. Triangles with a higher slope than the threshold are considered obstacles and cannot be traversed.

We use a clustering approach on all safe triangles to generate disconnected safe regions. The clustering approach considers the terrain mesh as an undirected graph and uses the triangle vertices and edges of the mesh as nodes and edges of the graph, respectively. The MATLAB `conncomp` function is then used to find connected components in the graph. As the graph is based on the terrain mesh, triangles with edges in the same graph component are in the same disconnected region. Next, for each triangle that has been marked in the previous step, the incenter point is calculated as

$$\begin{aligned} x_c &= \frac{ax_1 + bx_2 + cx_3}{a + b + c} \\ y_c &= \frac{ay_1 + by_2 + cy_3}{a + b + c} \\ z_c &= \frac{az_1 + bz_2 + cz_3}{a + b + c} \end{aligned} \quad (2)$$

where $p_1 = (x_1, y_1, z_1), p_2 = (x_2, y_2, z_2), p_3 = (x_3, y_3, z_3)$ are the vertices of the triangle, and a, b, c are the edges. These incenter points serve as nodes for the TSP.

4.2 Intra Region TSP

After defining the regions and nodes that need to be covered, the TSP needs to be solved for each region to generate the path the robot should follow. We frame the TSP as a Mixed-Integer Linear Programming (MILP) problem to leverage the optimality [19] and high scalability [21] of MILP solvers. The problem is formulated as

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} \pi_{ij} \quad (3)$$

$$\sum_{i=1}^n \pi_{ij} = 1 \quad \forall j \quad (4)$$

$$\sum_{j=1}^n \pi_{ij} = 1 \quad \forall i \quad (5)$$

$$\sum_{i \in S} \sum_{j \in \bar{S}} \pi_{ij} \geq 1 \quad \forall S \quad (6)$$

$$\sum_{i \in S} \sum_{j \in S} \pi_{ij} \leq |S| - 1 \quad \forall S \quad (7)$$

where n is the number of nodes, $\pi_{ij} \in \{0, 1\}$ is the decision variable that indicates whether a path between two nodes p_i and p_j is included in the solution and S is any subset of nodes. d_{ij} is the euclidean distance between p_i and p_j defined as

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \quad (8)$$

where $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$ represent the coordinates of the two nodes. Equation (3) describes the cost function of the problem, constrained by Equations (4) – (7). Equations (4) and (5) ensure that each node has only two paths connected to it. This guarantees that each node is only visited once.

Equations (6) and (7) describe the process of sub-tour elimination. This ensures that the result only contains a single path over all nodes. As it would be infeasible to add sub-tour constraints for all possible subsets, we chose an iterative approach. First, the problem is solved without any sub-tour constraints. If the solution contains multiple sub-tours, constraints are added for the nodes in each sub-tour and the problem is solved again. This process is repeated until a single-tour solution is found.

As the robot is expected to cover multiple disconnected regions, it is generally not desirable for the robot to return to its point of origin in each disconnected

region, as this would result in a longer path and make finding a solution more complex. To create this additional constraint, a virtual node p_0 is added to each region. p_0 has an arbitrary position outside of the map and its distance to any other node is defined as

$$d_{ij} = 0 \text{ if } i = 0 \vee j = 0 \quad (9)$$

which implies that the cost to reach p_0 from any node is always 0. This additional node and all edges connected to it will be deleted after the TSP is solved, leaving the two nodes that were connected to it with only a single neighbour. These two nodes will be considered as the endpoints of each region's path, which will be connected in the next step to generate the complete path across all regions. As MILP always converges on an optimal solution [19], it is ensured that the endpoints are placed in a way that reduces the intra-region path length by the largest amount. As no additional cost is generated by connecting to p_0 , the path length for the open loop TSP must be of equal length or shorter than the path length for the closed loop TSP. We solve the aforementioned MILP problem using the MATLAB `solve` function.

4.3 Inter Region TSP

In the final step, the endpoints of all regions need to be connected in sequence to generate a complete path over all regions. To do so, the same TSP approach as described above is used. We again add a node p_0 to generate an open loop solution as our use case does not require a direct return to the origin, but a closed loop can easily be generated by omitting this node.

To generate a valid path, it needs to be ensured that the path segment that enters a region R_k through an endpoint p_i continues to the second endpoint p_j of the same region R_k . We define this constraint by setting the weight between two endpoints of the same region as $d_{ij} = 0$ for both $p_i, p_j \in R_k$.

This means that once one endpoint has been chosen, the optimal next choice will always be either the second node in that region or p_0 . Finally, the edges between the two endpoints of the same region are removed, as they would not be traversed, being replaced with the intra-region solutions from the previous step. p_0 and the corresponding edges are removed as well.

5 Simulation Results

To evaluate the proposed approach, two experiments were conducted. First, to validate the method in a general context, it was tested on 500 randomly generated maps with varying complexity. Second, to demonstrate the versatility of the approach in a realistic real-world scenario, it was evaluated across four scenarios using mars DEM data.

The experiments were conducted on a Lenovo ThinkPad T480s with an Intel i5-8250U @1.6 GHz processor with 4 cores and 8 threads. The computer was equipped with 8 GB of DDR4-2400 RAM and an Intel UHD 620 internal graphics processor.

5.1 General Scenario

The approach was evaluated on 500 randomly generated maps. Each map consisted of 15×15 evenly spaced squares, each consisting of two triangles. The obstacle detection threshold was set to 15° . The slope of each triangle was generated randomly based on a maximum allowed slope. The maximum allowed slope was set randomly between 20° and 40° . Maps with a lower maximum slope generate close to no obstacles, while maps with a higher maximum slope generate almost no safe regions. These scenarios are trivial to solve and were therefore excluded. To simulate more realistic terrain properties, Gaussian smoothing is used with a random smoothing factor between 0 and 0.7. A smoothing factor over 0.7 generates maps void of obstacles in most cases and was therefore excluded. The experiment was run for a total of 500 trials.

These settings resulted in an average of 125.24 safe nodes per map with a maximum of 431, minimum of 9 and standard deviation of 103.84. The average number of disconnected regions was 14.17 with a maximum of 33, a minimum of 1 and a standard deviation of 6.67.

As a baseline for comparison, a greedy algorithm was run on the same set of trials. This algorithm starts at the first node in each region and always chooses the closest unvisited node that can be reached. Paths that would pass through an obstacle are modelled with an additional near-infinite penalty cost to discourage the greedy algorithm from choosing this path. The baseline algorithm managed to find a valid solution for 128 out of the 500 trials. The majority of the successful trials had only a small number of safe nodes with only four trials having more than 100 nodes. Out of these four trials, two had over 400 nodes, but all nodes belonged to the same region. Including trials that returned a path which passes through obstacles, ignoring penalty costs, the average total path length was 140.09 with a maximum of 353.77, minimum of 21.15 and a standard deviation of 85.58.

In contrast to this, our proposed approach returned a valid solution for all 500 trials, independent of node or region count. The average path length while respecting all obstacles is 112.06 with a maximum of 290.86, minimum of 16.62 and standard deviation of 66.59. This is an average reduction in path length of 20.01% whilst conforming to the more strict constraints. Fig. 2 shows that the proposed approach is capable of finding valid solutions in a variety of scenarios. In Fig. 2.a, our approach solves a large obstacle free region with a narrow passage connecting two mostly separated areas. Fig. 2.b shows the solution to a scenario containing multiple large regions that are completely separated by obstacles. Even in highly complex scenarios as shown in Fig. 2.c, which contain multiple narrow passages, dead ends and multiple large regions, the proposed approach is capable of finding a solution. Finally, a solution to a map containing a large number of disconnected regions is shown in Fig. 2. Compared to state-of-the-art approaches like TSP-CPP [10], our proposed approach is capable of generating near-optimal coverage paths in environments with arbitrarily shaped regions with obstacles, whereas TSP-CPP only considers obstacle free environments with convex regions.

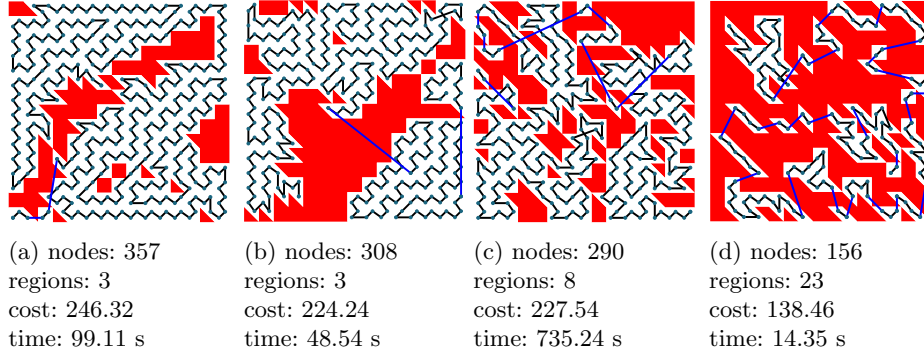


Fig. 2: Samples from the proposed algorithm run on generated maps. Red regions are obstacles, white regions are safe. Black lines indicate the intra-region coverage path, while blue lines indicate the use of the aerial modality.

On average, the proposed approach generated a valid solution within 32.36 seconds with a standard deviation of 72.01 seconds. The fastest solution was found within 7.76 seconds and the slowest within 735.24 seconds. With the exception of one trial, all other trials finished within 10 minutes or less. The trial which required more than 10 minutes to find a near-optimal solution consisted of one large region with several narrow passages and dead ends, as well as several small regions (Fig. 2.c). As nodes in those narrow passages are mostly surrounded by obstacles, most paths that connect to these nodes cross through obstacles and therefore are invalid. Due to the large number of nodes in the region, a large number of paths need to be evaluated before any improvement in path length is found, which explains the overall longer time to find a solution. In the 500 evaluated samples, an exponential relation between the number of nodes and the elapsed time can be observed. There is no strong correlation between the number of regions and the elapsed time.

5.2 Real-World Planetary Scenario

To show the versatility of the proposed approach, the algorithm was run on a selection of scenarios on Martian planetary data. The scenarios were gathered from MarsTrek [22], which provides DEM data as STL files. The MATLAB `reducepatches` function is used to reduce the number of triangles. While this introduces a slight inaccuracy into the solution, it greatly reduces the complexity, making the calculation possible. The resulting maps have a size of $400 * 400$ and generally contain a slightly higher, albeit similar, amount of nodes compared to the previous experiments. Fig. 3.a shows a 3D rendered example of a MarsTrek map that has been solved by our method. The approach can also solve valley scenarios where high slopes separate parts of the map (Fig. 3.b) and hill scenarios where only narrow regions of the map can be covered safely (Fig. 3.c). It can

also solve highly complex scenarios which contain a large number of nodes and regions (Fig. 3.d).

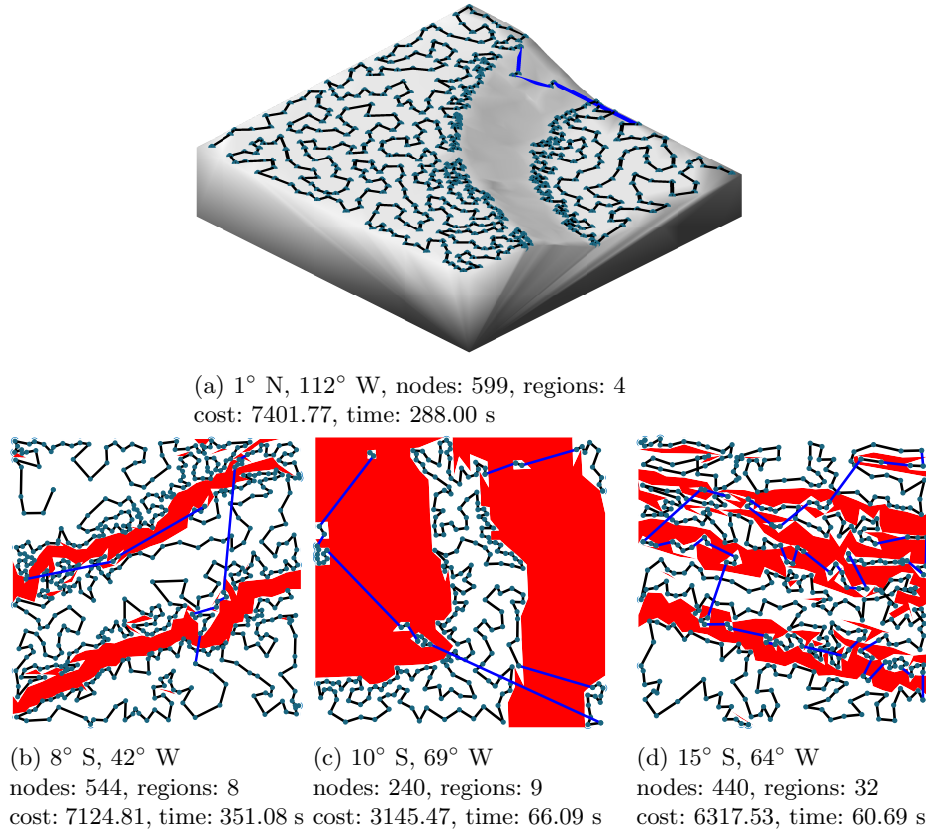


Fig. 3: Samples from the proposed algorithm run on MarsTrek DEM data [22]. (a) shows a 3D rendering of the path taken while (b), (c) and (d) show a top-down view of different scenarios with high slope regions highlighted in red. Black lines indicate the intra-region coverage path, while blue lines indicate the use of the aerial modality.

All trials find a valid near-optimal solution, even with the irregularly spaced nodes that stem from the way MarsTrek data is generated and the patch reduction step. While this is only a qualitative analysis, it still indicates that the proposed algorithm is capable of solving real-world scenarios.

6 Conclusion

We proposed a hierarchical MILP based method for solving the Multi-Modal Multiple Disconnected Region CPP (M3DR-CPP) problem. The proposed method uses terrain slopes to identify obstacles and safe regions on an input mesh based on DEM data. It then generates a coverage path in two steps by first formulating and solving a TSP in each region and then connecting the regions by solving a secondary TSP. The approach generated coverage paths for 500 randomly generated maps with up to 431 nodes in an average of 32.36 seconds. The approach is capable of solving problems with complex obstacle placements and up to 33 regions. It also was able to find solutions in scenarios based on realistic Martian surface data with up to 599 nodes. Generated paths are optimal within each region, but suboptimal between regions, as the placement of region endpoints is not optimized to minimize inter-region paths. Immediate future work considers optimal endpoint node selection to mitigate this problem, and inclusion of traveling and control costs. Future work will also consider path smoothing, dynamic adaptation to changes in the environment, as well as switching to a heuristic TSP solver to improve scalability.

References

- [1] Chee Sheng Tan, Rosmiwati Mohd-Mokhtar, and Mohd Rizal Arshad. “A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms”. In: *IEEE Access* 9 (2021), pp. 119310–119342.
- [2] Benjamin Morrell et al. “NeBula: TEAM CoSTAR’s robotic autonomy solution that won phase II of DARPA subterranean challenge”. In: *Field robotics* 2 (2022), pp. 1432–1506.
- [3] Myungjin Jung, Kai Chuen Tan, and Ran Dai. “Path Planning for a Jumping Rover Team With a Charging Station in Multi-Waypoints Visiting Missions”. In: *Frontiers in Control Engineering* 3 (2022), p. 803468.
- [4] Brandon Araki et al. “Multi-robot path planning for a swarm of robots that can both fly and drive”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5575–5582.
- [5] Howie Choset and Philippe Pignon. “Coverage path planning: The boustrophedon cellular decomposition”. In: *Field and service robotics*. Springer. 1998, pp. 203–209.
- [6] Xuecong Chen et al. “Region coverage path planning of multiple disconnected convex polygons based on simulated annealing algorithm”. In: *2021 IEEE 4th International Conference on Computer and Communication Engineering Technology (CCET)*. IEEE. 2021, pp. 238–242.
- [7] Kevin Yu, Jason M O’Kane, and Pratap Tokekar. “Coverage of an environment using energy-constrained unmanned aerial vehicles”. In: *2019 international conference on robotics and automation (ICRA)*. IEEE. 2019, pp. 3259–3265.

- [8] G Chen et al. “2D multi-area coverage path planning using L-SHADE in simulated ocean survey., 2021, 112”. In: *DOI: <https://doi.org/10.1016/j.asoc>* (2021), p. 107754.
- [9] Lexu Du et al. “A Multi-Regional Path-Planning Method for Rescue UAVs with Priority Constraints”. In: *Drones* 7.12 (2023), p. 692.
- [10] Junfei Xie, Luis Rodolfo Garcia Carrillo, and Lei Jin. “Path planning for UAV to cover multiple separated convex polygonal regions”. In: *IEEE Access* 8 (2020), pp. 51770–51785.
- [11] Abdul Majeed and Seong Oun Hwang. “A multi-objective coverage path planning algorithm for UAVs to cover spatially distributed regions in urban environments”. In: *Aerospace* 8.11 (2021), p. 343.
- [12] Megnath Ramesh et al. “Optimal partitioning of non-convex environments for minimum turn coverage planning”. In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 9731–9738.
- [13] Bocheng Zhao et al. “Clustering-based hyper-heuristic algorithm for multi-region coverage path planning of heterogeneous UAVs”. In: *Neurocomputing* 610 (2024), p. 128528.
- [14] Xiaoxiao Yu et al. “Balanced multi-region coverage path planning for unmanned aerial vehicles”. In: *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2020, pp. 3499–3506.
- [15] Li Wang et al. “Coverage Path Planning for UAVs: An Energy-Efficient Method in Convex and Non-Convex Mixed Regions”. In: *Drones* 8.12 (2024), p. 776.
- [16] Friedrich M Rockenbauer et al. “Traversing Mars: Cooperative Informative Path Planning to Efficiently Navigate Unknown Scenes”. In: *IEEE Robotics and Automation Letters* (2024).
- [17] Jingtao Tang, Chun Sun, and Xinyu Zhang. “MSTC*: Multi-robot coverage path planning under physical constrain”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 2518–2524.
- [18] Xiang Huang et al. “A multi-robot coverage path planning algorithm for the environment with multiple land cover types”. In: *IEEE Access* 8 (2020), pp. 198101–198117.
- [19] Christodoulos A Floudas. *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press, 1995.
- [20] Der-Tsai Lee and Bruce J Schachter. “Two algorithms for constructing a Delaunay triangulation”. In: *International Journal of Computer & Information Sciences* 9.3 (1980), pp. 219–242.
- [21] Thorsten Koch et al. “Progress in mathematical programming solvers from 2001 to 2020”. In: *EURO Journal on Computational Optimization* 10 (2022), p. 100031.
- [22] Emily Law and B Day. “Mars trek: An interactive web portal for current and future missions to Mars”. In: *European Planetary Science Congress*. 2017, EPSC2017–99.



DSPACE

<https://dspace.org/>

Hierarchical coverage path planning for a multi-modal robot exploring disconnected regions

Stelter, Sebastian; Kuan lo U, Daniel; Mondal, Sabyasachi; Felicetti, Leonard; Upadhyay, Saurabh

2026-08-13

Springer

https://doi.org/10.1007/978-3-032-01486-3_33

Stelter S, Kuan lo U D, Mondal S, et al., (2026) Hierarchical coverage path planning for a multi-modal robot exploring disconnected regions. In: 26th Annual Conference, TAROS 2025, 20-22 August 2025, York, UK, Towards Autonomous Robotic Systems, Lecture Notes in Computer Science (LNAI) Volume 16045