This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3044987, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Coverage Path Planning Techniques for Inspection of Disjoint Regions with Precedence Provision

ZEBA KHANAM [1], SANGEET SAHA [1], SHOAIB EHSAN [1], (SENIOR MEMBER, IEEE), RUSTAM STOLKIN [2], (MEMBER, IEEE) AND KLAUS MCDONALD-MAIER [1], (SENIOR MEMBER, IEEE)

[1]School of Computer Science and Electronic Engineering (CSEE), University of Essex,U.K (e-mail: {zeba.khanam, sangeet.saha, sehsan, kdm}@essex.ac.uk)
[2]University of Birmingham, UK (e-mail: R.Stolkin@cs.bham.ac.uk)

Corresponding author: Zeba Khanam (e-mail: zeba.khanam@ essex.ac.uk).

This work is supported by the UK Engineering and Physical Sciences Research Council through grants EP/R02572X/1 and EP/P017487/1.

ABSTRACT Recent times are witnessing an emergence of sites that are hazardous for human access. This has created a global demand to equip agents with the ability to autonomously inspect such environments by computing a coverage path effectively and efficiently. However, inspection of such sites requires agents to consider the correlation of work, providing precedence provision in visiting regions. The current approaches to compute coverage path in the hazardous sites, however, do not consider precedence provision. To this end, coverage path planning strategies are proposed, which provide precedence provision. To meet the challenges, the problem is divided into two phases: inter-region and intra-region path planning. In the 'inter-region' path planning of the approach, the site comprising of multiple disjoint regions is modelled as connectivity graph. Two novel approaches, Mixed Integer Linear Programming (MILP) solution and heuristic based techniques, are proposed to generate the ordered sequence of regions to be traversed. In the 'intra-region' path planning of the approach, each region is decomposed into a grid and Boustrophedon Motion is planned over each region. The ability of combined approach to provide complete coverage is proved under minor assumption. An investigative study has been conducted to elucidate the efficiency of the proposed approach in different scenarios using simulation experiments. The proposed approach is evaluated against baseline approaches. The results manifest a significant reduction in cost and execution time, which caters to inspection of target sites comprising of multiple disjoint regions with precedence provision.

INDEX TERMS Autonomous Systems, Coverage Path Planning, Precedence Provision, Inspection, Optimization Algorithms .

I. INTRODUCTION

AGENT based inspection of any site can be characterized as a coverage path planning (CPP) problem, where a path is derived, such that every single point of the environment is traversed. Even though CPP is a well researched problem in literature [1]–[3], the research has typically focussed on coverage of single region or multiple connected regions. However, the advancement in science and technology has resulted in significant increase in the sites which can be characterized by multiple disjoint regions and require inclusion of precedence provision, due to the prevailing environmental conditions like high pressure, extreme temperature and presence of radiation.

Recent accidents at such sites like the Fukushima Daiichi nuclear meltdown, Gulf of Mexico oil spill etc. have resulted in environmental catastrophe. This has generated a global demand of periodic inspection of sites with extreme environment. Human inspection of these sites is hazardous and requires an agent to perform this daunting task. The inspection of an extreme environment site has the following salient characteristics which need to be considered while computing coverage path:

1) Disjoint Regions: The precarious sites consist of regions which are spatially distributed and an agent is entrusted to cover each region. Some applications of coverage of disjoint regions are the survey of post-

disaster scenarios [4], precision agriculture [5] to name a few.

2) Correlated Nature of Work: Inspection of the site requires agent to perform sensing and actuating actions like corrosion check, radiation monitoring [6] [7]. There exists a certain degree of correlation between these actions, which impose requirement of visiting regions with a precedence provision. In the experimental study conducted by Khanam et al. [8], it was observed that annealing process starts as soon as radiation exposure is removed. This allows us to conclude, that after covering regions which have a presence of radiation, the next region must be a non-critical area.

The development of agents which can survey sites has revolutionized the field of surveillance and inspection. The recent [9] agent based fixed altitude CPP problem mapped the environment monitoring problem as a multiple target visiting problem. The multiple targets visiting problem was then formulated as Travelling Salesman Problem (TSP) [10]. TSP generated an optimal path, where each target was visited once taking into consideration disjoint regions, however ignoring the constraint of correlated nature of work. Recently, agent *'CARMA'* (Continuous Autonomous Radiation Monitoring Assistance) [11] used greedy online CPP to inspect Sellafield nuclear facility. However, it failed to consider distributed nature of regions. In other cases, where the agent was assigned to inspect any region [3], [12], the CPP problem was formulated as the derivation of the optimal path, so that one region was covered considering the precedence provision. Another recent agent based post-disaster survey had proposed a variant of grid based TSP solution for systematic investigation of site with distributed regions [13]. However, it failed to eliminate the sub tours [14], which is an integral problem of TSP.

Although, agent based inspection has been studied extensively, to the best of the authors knowledge, none of the works focuses on agent based coverage path planning techniques considering both the salient characteristics. This paper seeks to bridge this gap, by proposing a coverage path planning technique which inspects any site with disjoint regions providing precedence provision.

In this paper, we adopt a scheme which provides a solution to the problem of inspection of multiple non-overlapping regions with given precedence (see Figure 1). The coverage path planning problem for multiple disjoint regions is first presented as a Mixed Integer Linear Programming (MILP) formulation . The problem being very complex, use of the MILP formulation makes the solution approach highly compute intensive, and restricts its scalability to problems containing only a small number of regions. However, the intended application of inspection requires agent to cover vast geographical areas, where both the number of regions and their average sizes may be pretty large. This demands a heuristic solution where the coverage path planning problem of disjoint regions has been decomposed into two parts: 1) Inter-region path planning and 2) Intra-region path planning.

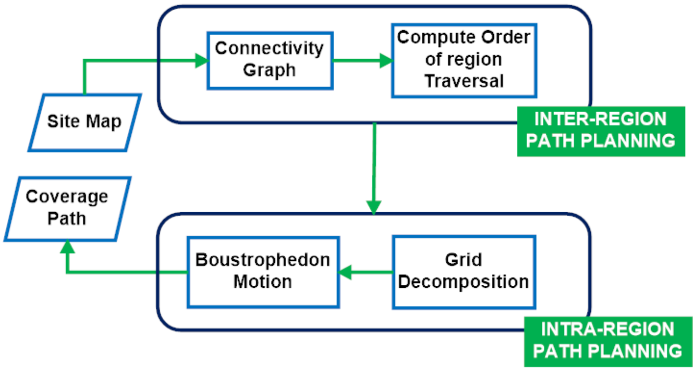

Site Map
Connectivity Graph
Compute Order of region Traversal
INTER-REGION PATH PLANNING
Coverage Path
Boustrophedon Motion
Grid Decomposition
INTRA-REGION PATH PLANNING

FIGURE 1: Flowchart illustrating an overview of all the steps required to compute the coverage path

We consider a scenario where the regions are far apart, therefore the distance between two regions can be approximated as distance between their centroid. Our scheme treads a route by initially dividing the problem into two phase: Inter-region path planning and Intra-region path planning. In the first phase, an attempt is made to answer a pertinent question: How can an agent inspect the entire site by visiting each region while considering the specified order of correlated nature of region visit. A MILP solution and a construction heuristic, Inter-Region Path Planning (*'IRPP'*) are proposed, which generate the order of inter-region traversal. This order serves as an input to the second phase of *'Intra-region traversal'*, where entry and exit point for each region are calculated on the basis of preceding and succeeding region, in the order of generated inter-region traversal. Deriving this information, the second phase first decomposes the region into a grid and then calculates the Boustrophedon Motion between entry and exit grid cell for each region.

A. TARGETED APPLICATION

CPP is integral to many robotics applications, such as painter robots [15], vacuum cleaning robots [16], autonomous underwater vehicles [17], demining robots [18] and precision agriculture [19]. However, most applications of CPP have considered computation of the coverage path for a single region in the past. However, new applications have emerged, where agents needs to cover multiple regions that are spatially distributed. Some of the applications targeting computation of a coverage path over multiple disjoint region are: post-disaster management scenario where a UAV is assigned to access the damage for multiple-disaster areas [13]. Another important application is deployment of UAVs to perform search and rescue mission over multiple disjoint regions [5].

However, recent accidents like Fukushima Daiichi nuclear meltdown, Gulf of Mexico oil spill have generated a global demand of regular inspection of such sites. Such sites, like nuclear power plants, oil and gas pipelines, chemical industry comprise extreme environments and require an aerial agent to inspect these precarious sites at regular intervals. Our recent experimental study has found out the sensors deployed in the agents to degrade [7], [8]. The degradation model observed that the sensor starts annealing from damages after recorded time. These observations indicate that, while inspecting these precarious regions, the agent must adhere to a predefined precedence so that the deployed sensor could benefit the annealing effect. Therefore, the proposed work of coverage path planning with precedence provision finds application in inspection of regions with extreme environment.

B. CONTRIBUTION

The main technical contributions of this paper can be summarised as:

1) **Precedence Provision:** Our previous studies [7], [8] have shown that agents require to visit regions in specific order. Thus, this work provides precedence provision, while covering multiple disjoint regions with extreme environments.
2) **Inter-Region Path Planning:** The given site of multiple non-overlapping regions is transformed into a connectivity graph where each vertex represents a region and edge represents the cost incurred traversing from one region to another. Using the connectivity graph, a mixed integer linear programming solution considering disjoint regions and precedence provision is formulated to generate an optimal inter-region traversal. A heuristic based technique, known as 'Inter-Region Path Planning'(*IRPP*), is proposed which is capable to produce close to optimal inter-region order with less overheads.
3) **Intra-Region Path Planning:** An intra-region traversal technique is proposed to derive the path for agent to traverse each region using Boustrophedon Motion.
4) **Experimental studies:** We evaluate the performance of our proposed strategies using simulation studies for wide range of distributed regions based sites. The results have demonstrated the efficiency and efficacy of the proposed approaches and have compared with the state-of-the-art techniques.

C. PAPER ORGANISATION

The remainder of the paper is organised as follows. Section II provides an overview of the related works in this area. This is followed by problem formulation in Section III. Section IV encapsulates a discussion on the proposed inter-region path planning technique. The intra-region path traversal technique is presented in Section V. The experimental evaluation of the proposed method is given in Section VI. Finally, the conclusions are presented in Section VII.

II. RELATED WORKS

There are many sites, where human inspection is impossible, as the human access to such sites is restricted due to the harsh prevailing conditions. To address this challenge, many agents had been previously deployed at these sites [20]. Due to the degradation of the localization and imaging sensors [7], [8], [21] in the environment, had led to the failure of agents [22]. This directed the research into the direction of development of radiation hardened and pressure sustainable sensors [23]–[25], which elongated the life of the agent inside such sites. Due to the heavy weight of radiation hardened sensors, they could typically be deployed in ground based agents. Recently, commercial-off-the shelf sensors based agents operating at fixed altitude [13] have been used to perform the inspection of disaster hit areas. Taking cue, we solve a inspection problem by imposing characteristics of disjoint regions with precedence provision.

Another constraint specific to the inspection of extreme environment sites like power plants is equipping the agent with an ability to compute a set of achievable routes where sensing and perception of each region is dependent on the information of one of the preceding regions [26]. For instance, to inspect high rise building structures like radiation storage centres, cooling towers or turbines, the building is divided horizontally into each region, like floors, such that information of each region is dependent upon the information of its predecessor region [27]. Therefore, we consider computation of the coverage path for distributed regions having precedence provision.

The advent of commercial off-the shelf sensor based agents to survey sites has been a ground breaking development. One of the recent development was agent CARMA (Continuous Autonomous Radiation Monitoring Assistance) [11], which used sensor based CPP to inspect Sellafield, the largest nuclear facility in Europe. However, it did not consider the degradation of sensors due to presence of radiation. In an another work [9], agent tried to monitor an environment with unknown intensity of extreme environmental conditions. It translated the problem to a multiple target visiting problem. The multiple targets visiting problem was transformed to a TSP [10]. The work considered the limited energy and time budget, however it failed to take into consideration the correlated nature of work.

In few other works [3], [12], the agent surveyed a site by considering CPP as a problem of optimal path computation so that the agent fully covered the region without imposing any constraints. Another relevant work aimed to survey a post-disaster site by proposing a solution which used a grid based TSP solution [13]. However, this did not take in consideration the presence of cyclic sub tours [14], which rendered the solution ineffective. In few of the recent works [28]–[31], authors investigated inspection using multiple agents by computing cooperative path planning to visit a target. However, due to degradation of agents' sensor in extreme environment, the deployment of multiple agents will not be judicious decision.

Diverse research has envisioned the problem of the single agent based inspection as a CPP problem. Key relevant works in this field have been discussed in detail in the following surveys: Choset [1], Galceran and Carreras [3] and Bormann et. al. [2]. Choset [1] related the problem to variant of TSP, *covering salesman problem*, where instead of visiting all the points individually in a region, an agent must visit a single point in neighbourhood of the region from where the agent's sensor can cover the entire neighbourhood of the point. We take inspiration from this relation, while solving the inter region path planning, we represent the centre of the region as a vertex of connectivity graph.

Choset also proposed classification of CPP as off-line or on-line problem. The off-line coverage algorithm assumes the environment to be stationary and its map to be available. The on-line algorithms, on the other hand, do not assume any a priori knowledge of the environment and use real time sensor data for run-time path computation. Since the target site in the problem at hand does not have any human presence and is periodically inspected, we can assume that the offline map is a correct representation of the site. Apart from that, since environmental condition degrade the sensors, relying on sensor data for coverage decision will be cost and time inefficient. Therefore, we propose an off-line CPP algorithm for inspection of extreme environment sites. Bormann et al. [2] presented a survey of the off-line coverage algorithms and observed that the CPP problems due to their complexity are typically computed not on the entire map, but individual units of map, known as 'region'.

The problem of coverage path planning by agents has been formulated as an optimization problem in recent times [32]. The optimization problem can be divided into following two sub-categories: 1) Classical Approaches, and 2) Heuristic Approaches. Popular classical approaches focusing on optimality guarantees for coverage path planning are: Potential Field [33], Grid-Based [34], Probabilistic Roadmap Methods (PRM) [35] and Rapidly exploring Random Trees (RRT) [36]. The heuristic approaches which search the solution space for near optimal solution can be sub-categorized as: Neural Network [37], Fuzzy logic techniques [38], Gravitational Search Algorithm [39], Genetic Algorithm [40] and Ant-Colony Algorithm [41]. The optimization problem of computing the coverage path has been classified as NP-Hard, the problem of coverage of disjoint regions adds to its complexity. Therefore, we have used the technique of decomposition of original problem into sub-problems which are highly correlated, this has been widely used in some of the optimization algorithms [42].

This has allowed to divide our inspection problem into two parts: 1) inter-region and 2) intra-region path planning. Using the research finding that the inspection problem of sites has two salient characteristics, we seek to solve the problem as inter-region and intra-region path planning. In the next section, we present the problem formulation.

III. PROBLEM FORMULATION

In this section, we describe the problem of computation of CPP of disjoint regions with precedence provision and formally state the optimization problem as constant satisfaction problem via mixed integer linear programming formulation.

A. PROBLEM DESCRIPTION AND ASSUMPTION

Let us consider a scenario where an agent equipped with a sensor and is initially stationed at depot v_0. Then, the agent is dispatched to inspect a set of disjoint and spatially distributed rectangular regions, $\mathcal{M} = \{1, \dots, m\}$. Each region $i \in \mathcal{M}$ is described by its four vertices $(v_{i1}, v_{i2}, v_{i3}, v_{i4}) = V_i$. We suppose that the target site does not consist of any exclusion or forbidden zone. After inspecting the entire site, the agent is required to return back to the depot. We assume that the agent operates at the fixed altitude and constant speed. If $\mathbb{R}^+ = \{a \in \mathbb{R} : a \geq 0\}$, the configuration space Ω for agent $\mathcal{A}$ can be formulated as $\Omega \subseteq \mathbb{R}^2$. The sensing range of the sensor mounted on the agent is $r \times r$. We assume that the agent has sufficient power to complete the task. The geographical area of the site has been modelled using a discrete 2D grid structure such that size of each grid cell is designed to be $r \times r$, so that if the agent visits the centre of grid cell, the entire grid cell can been covered. *The objective of the problem is to find the optimal tour for the agent that start and ends at the depot, such that agent's sensor footprint along the traversed path covers each target region completely and the total travel cost is minimized. Apart from covering each target location, the agent also has to achieve the goal of traversing the target regions in an ordered sequence.*

The type of agent used for inspection also places an additional constraint on the shape of the path. For instance, if the agent is a fixed-wing UAV, its mechanical structure determines minimum safe turn radius restricting the agent to make abrupt changes in their direction. In this problem, we have assumed that our agent is a multirotor UAV that is allowed to make sharp turns with arbitrary turn radius, allowing significant increase in the flexibility, while designing the path to be a linear problem. A similar approach was proposed for a control method which was based on linearization of the oxygenator's dynamical model [43].

B. NOTATIONS

Addressing the site inspection problem more formally, we introduce the basic notations. The problem of CPP of disjoint regions can be sub-divided into: 1) inter-region path planning 2) intra-region path planning. The intra-region coverage path for each region $i \in \mathcal{M}$ can be described as P_i. $P_i = p_{ik}$ is the set of Cartesian coordinate of centre of each grid cells in region i, such that by visiting all the locations in P_i entire region i has been covered, where $k \in [g_i]$, g_i is the set of grid cells located in the region i. The visiting order of the locations in P_i are captured using a decision variable z^i_{kl}. The decision variable $z^i_{kl} = 1$, if agent traverses from location p_{ik} to p_{il}, otherwise, $z^i_{kl} = 0$.

The entrance location in region i is described by decision variable en^i_k, such that $en^i_k = 1$, if the agent enters region from location p_{ik}, otherwise $en^i_k = 0$. Furthermore, we introduce ex^i_k as decision variable to describe the exit location in region i, such that $ex^i_k = 1$, if the agent exits from region from location p_{ik}, otherwise $ex^i_k = 0$. Finally, to capture the order of visit of the target regions, we introduce decision variable x_{ij}, where $i, j \in \mathcal{M}$, such that $x_{ij} = 1$, if the agent traverses from region i to region j, otherwise $x_{ij} = 0$. The specified order in which location must be traversed is given by $\Theta = (i,j)|i,j \in \mathcal{M}$. To capture this order, we introduce a decision variable y_{ij} such that $y_{ij} = 1$, if the agent traverses from region i to j for specified order $(i,j) \in \Theta$, otherwise $y_{ij} = 0$.

C. MIXED INTEGER PROGRAMMING FORMULATION

With the notations described in above section, we can now describe the coverage path that covers the entire target site with precedence provision. The total travel cost C can be calculated as:

$$C = \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} \sum_{k=1}^{g_i} \sum_{l=1}^{g_j} x_{ij} ex_k^i en_l^j D(p_{ik}, p_{jl})$$
$$+ \sum_{i=1}^{m} \sum_{k=1}^{g_i} \sum_{l=1, l\neq k}^{g_i} z_{kl}^i D(p_{ik}, p_{il}) + \sum_{(i,j)\in\Theta} \hat{P}(1 - y_{ij})$$

where function $D(x,y)$ calculates the travel cost from location x to y and $\hat{P}$ is the penalty imposed, if the agent does not traverse regions in the specified order Θ. To ensure the validity of the tour, the following constraints are imposed:

$$\sum_{j=1, j\neq i}^{m} x_{ij} = 1, \; \forall i \in \mathcal{M} \tag{1}$$

$$\sum_{i=1, i\neq j}^{m} x_{ij} = 1, \; \forall j \in \mathcal{M} \tag{2}$$

$$x_{ij} \geq y_{ij} \qquad (i,j) \in \Theta \tag{3}$$

$$\sum_{l=1, l\neq k}^{g_i} z_{kl}^i = 1 - ex_k^i, \; \forall i \in \mathcal{M}, l \in g_i \tag{4}$$

$$\sum_{k=1, k\neq l}^{g_i} z_{kl}^i = 1 - en_k^i, \; \forall i \in \mathcal{M}, k \in g_i \tag{5}$$

$$\sum_{k=1}^{g_i} en_k^i = 1, \; \sum_{k=1}^{g_i} ex_k^i = 1, \; \forall i \in \mathcal{M} \tag{6}$$

$$x_{ij}, y_{ij} \in \{0,1\}, \; \forall i,j \in \mathcal{M} \tag{7}$$

$$z_{kl}^i, en_k^i, ex_k^i \in \{0,1\}, \; \forall i \in \mathcal{M}, k,l \in g_i \tag{8}$$

$$en_k^i + ex_k^i \leq 1, \; \forall i \in \mathcal{M}, k \in g_i \tag{9}$$

In the above mentioned constraints, constraints (1) - (2) ensure that each region is inspected by the agent exactly once. The constraint (3) ensures that the specified order edge in Θ is visited. The constraints (4) - (5) ensure that each location $p_{ik} \in P_i$ in region i is traversed only once. The constraint (6) ensures that for each region there is only one entrance and exit. The constraints (7) - (9) ensure that the decision variables take valid values.

The inspection problem can thus be formulated as:

$$\begin{aligned} &minimize \quad C \\ &subject\ to: \ constraints(1) - (9) \end{aligned} \tag{10}$$

D. DISCUSSION

The problem of coverage of disjoint region with precedence provision can be modelled as an optimization problem of integrated Chinese Postman Problem-Coverage Path Problem. Analysis shows that the both problems are NP-Hard in the strong sense [3], [44] and an optimal solution strategy to it will be prohibitively expensive in terms of solution generation times and required storage space. Further, it is also difficult to design a deterministic/greedy heuristic strategy which avoids solution enumeration, but can deliver satisfactory outputs under all realistic scenarios. In past, such complex optimization problem were solved by decomposing into co-related problems [42]. Hence, we have resorted to solve the this problem by decomposing into two problems: 1) Inter-region path planning 2) Intra-region path planning.

1) *Inter-region* path planning: It aims to find the order of sequence $S(\mathcal{M}) = \{i, j, \ldots, k \mid \forall i,j,k \in \mathcal{M}\}$ in which the agent visits all the regions traversing all the specified edges denoted by $(i,j) \in \Theta$.
2) *Intra-region* path planning: It can be stated as path computation such that each and every point in any given region i is covered by the agent $\mathcal{A}$.

Note: If the size of each region $i \leq r \times r$ (sensor's range), the inspection problem is reduced to only inter-region path planning, where each region $i \in \mathcal{M}$ can be described by its centre $\hat{v}_i$. If there is only one region i whose size is larger than sensor's range $r \times r$, the problem is reduced to intra-region path planning.

IV. INTER-REGION PATH PLANNING

In this section, we aim to solve the problem of inter-region path planning. To compute the inter-region path, we represent the site as a connectivity graph $\mathcal{G}$. The connectivity graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is an Eulerian graph, where $\mathcal{V}$ $=\{i, j, ..k| \; \forall i,j,k \in \mathcal{M}\}$ and $\mathcal{E}$ = $\{e_i| \; e_i$ is cost of traversal between region $i-1$ and region $i\}$. The graph is computed in following subsection.

A. CONNECTIVITY GRAPH

Any region $i \in \mathcal{M}$ is defined by four vertices $\{v_{i1}, v_{i2}, v_{i3}, v_{i4}\}$. Using these four vertices, the coordinates of the centre of a region i, $\hat{v}_i$ can be derived using:

$$\hat{v_i}^x = v_{i4}^x + \frac{|v_{i4}^x - v_{i3}^x|}{2} \tag{11}$$

$$\hat{v_i}^y = v_{i4}^y + \frac{|v_{i4}^y - v_{i1}^y|}{2} \tag{12}$$

The centre $\hat{v}_i$ is used to describe the region i, such that $\mathcal{V} = \{\hat{v}_i \mid \hat{v}_i \; is\ the\ centre\ of\ the\ region\ i \; \in \; \mathcal{M}\}$. To address the problem of inter-region path traversal, we first eliminate the degenerated edges of the graph, which connect a vertex to itself also known as a self-loop. The judicious elimination of self-loops helps in realization of the goal trajectory minimization. After the elimination of self-loops, we determine the cost which agent $\mathcal{A}$ will incur in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3044987, IEEE Access

traversing between two regions. The cost of traversal for the agent $\mathcal{A}$ for traversing from region i to region j is defined as $c_{i,j}$ and is computed as follows:

$$c_{i,j} = A^{*}\ distance\ between\ \hat{v}_i\ and\ \hat{v}_j \tag{13}$$

The agent $\mathcal{A}$ can traverse the next region j from current region i iff region i is the neighbour of region j. We assume that generated graph $\mathcal{G}$ contains an Eulerian trail, i.e. there exists a cycle where each edge can be traversed only once. This is illustrated in Algorithm 1, where we calculate the cost of all the neighbours of region i. The calculated inter-region traversal cost (c) and specified possible set of traversal (Θ') will be used as the input to the MILP based inter-region traversal strategy discussed in the next section.

Algorithm 1 CONNECTIVITY GRAPH

```
Input: 1. $\mathcal{M} = \{1, .., m\}$ : set of $m$ regions
       2. $\Theta = \{(i, j) \mid i, j \in \mathcal{M}\}$ : order of sites to be inspected
       3. $N$: a large positive number
Output: $\Theta', c$ : Set of specified possible region traversal,
        cost of inter-region traversal

INITIALIZATION
$\Theta' = \Theta$
/*$\Theta'$ denotes modified ordered site traversal set*/
SELF-LOOP ELIMINATION
for each region $i \in \mathcal{M}$ do
    Centre of region $i$ calculated using Equation 11, 12
    $c_{i,i} = N$ /* Assign the cost of self-loop $(i, i)$ as $N$ to
    eliminate self loop traversal.*/
    if exists a self loop $(i, i) \in \Theta'$ then
        $\Theta' = \Theta' \cap (i, i)$
        /*Remove the self loop $(i, i)$ from $\Theta'$.*/
    end if
end for
COST MATRIX CALCULATION
for each region $i \in \mathcal{M}$ do
    for each neighbour region $j$ of $i$ do
        Assign $c_{i,j}$ as A* distance between centre $\hat{v}_i$ and
        $\hat{v}_j$)
        /* Determine the cost of traversal from region $i$ to
        region $j$*/
    end for
end for
```

B. MIXED INTEGER LINEAR PROGRAMMING BASED INTER-REGION TRAVERSAL STRATEGY

In this section, we propose a *MILP* solution to inter-region path traversal problem. To address this problem we introduce two binary decision variable x_{ij} and y_{ij}. The decision variables are defined as:

1) $x_{ij} = 1$ if the agent $\mathcal{A}$ traverses region i to region j, otherwise $x_{ij} = 0$.
2) $y_{ij} = 1$ if the agent $\mathcal{A}$ traverses region i to j for the specified edge $(i, j) \in \Theta'$, otherwise $y_{ij} = 0$.

Objective: The inspection problem can be formulated as finding values of x_{ij} and y_{ij}, $\forall i, j \in \mathcal{M}$, such that the path starts and ends at the start region (also called depot)), each region is covered exactly once by the agent, and the total cost, $Cost$, given below is minimized.

$$Cost = \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{(i,j)\in\Theta'} \hat{c}_{ij}(1 - y_{ij}) \tag{14}$$

where c_{ij} is the travelling cost incurred by the agent $\mathcal{A}$, if it travels from region i to j, obtained using Algorithm 1 and $\hat{c}_{ij}$ is the penalty imposed on the agent $\mathcal{A}$ if it does not traverse region $(i, j) \in \Theta'$. The objective of the formulation can be written as:

$$minimize\ Cost \tag{15}$$

The first term in Equation 14 calculates the total travel costs for inter-regional movements. The last term calculates the penalty incurred if the order of sequence defined in Θ' is not followed by the agent $\mathcal{A}$. The validity of the required path traversal is ensured by imposing following constraints:

1) **Departure Constraint**: The constraint in Equation 16 ensures that the agent $\mathcal{A}$ leaves each region only once.

$$\sum_{i\in\mathcal{M}\setminus i\neq j} x_{ij} = 1, \quad \forall j \in \mathcal{M} \tag{16}$$

2) **Arrival Constraint**: The constraint in Equation 17 ensures that the agent $\mathcal{A}$ arrives only once in each region.

$$\sum_{j\in\mathcal{M}\setminus i\neq j} x_{ij} = 1, \quad \forall i \in \mathcal{M} \tag{17}$$

3) **Continuity Constraint**: The constraint in Equation 18 ensures that the inter-region path traversed by the agent $\mathcal{A}$ is continuous.

$$\sum_{j\in\mathcal{M}, i\neq j} x_{ij} = \sum_{j\in\mathcal{M}, i\neq j} x_{ji} \tag{18}$$

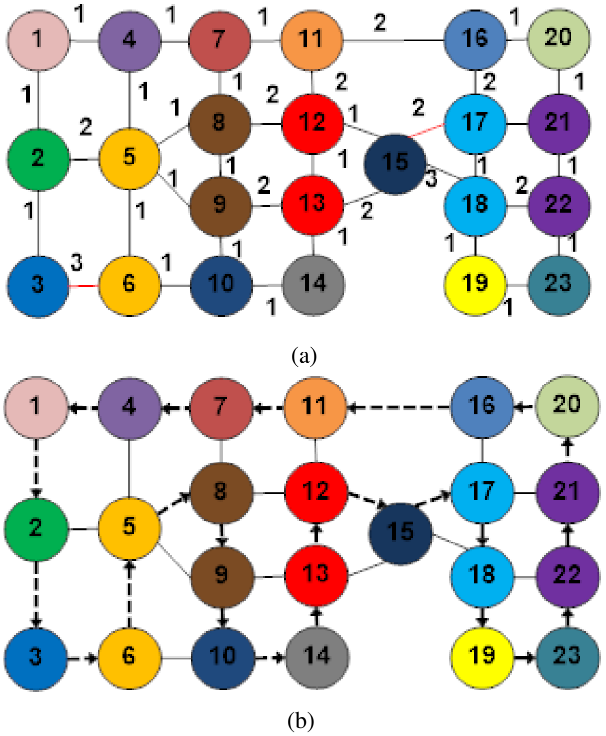
4) **Direct Path Constraint**: The constraint in Equation 19 ensures that the next region j to be traversed from region i is connected.

$$x_{ij} = 0, \quad \forall i, j \in \mathcal{M} \mid \nexists! \{\text{direct path between i and j}\} \tag{19}$$

5) **Specified Order Constraint**: The constraint in Equation 20 forces variable y_{ij} to take value 0, if not traversed by the agent, i.e. $x_{ij} = 0$. Conversely, if the agent takes the route (i, j), i.e, $x_{ij} > 0$, the minimization of the objective function forces variable y_{ij} to take value 1.

$$x_{ij} \geq y_{ij} \qquad (i, j) \in \Theta \tag{20}$$

6) **Sub-Tour Elimination:** Another major issue which path planning problems suffer from, is the ability of the agent $\mathcal{A}$ to get stuck in a sub-tour cycle in a graph $\mathcal{G}$, which has not been considered in previous coverage works [13], [11]. An additional decision variables u_i



(a)

(b)

FIGURE 2: Site depicted using (a) Inter-region connectivity graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where red edge $\{(3,6),(15,17)\} \in \Theta$ denote specified inspection order (b) CPLEX generated inter-region path denoted using dotted lines

is introduced in this constraint to order all regions excluding the depot to prevent illegal cycle formations. This is ensured by constraining $u_j \geq u_{i+1}$ when $x_{ij} = 1$

$$u_i + 1 \leq u_j + N * (1 - x_{ij}) \tag{21}$$

where, N is a large positive number.

C. EXAMPLE 1: ILLUSTRATING MIXED INTEGER LINEAR PROGRAMMING STRATEGY

We elucidate the working of our proposed MILP strategy using an example site of a spaceport. The given site resembles a site used in [45] comprising of 23 regions. Each unit region is a vertex v_i of the connectivity graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, shown in Figure 2a.

Let us assume that agent $\mathcal{A}$ operates at a fixed altitude and each region is inter connected by a unit distance such that $\forall j, k \; c_{j,k} = 1$. For now, we assume that the traversal from one region to another generates a cost which is shown on each edge (see Figure 2b). This assumption will be relaxed in Section VI. As per MILP formulation discussed in above subsection IV-B, the CPLEX [46] solver generates the following optimal tour as illustrated in Figure 2b. The specified order route for inspection is $\Theta = \{(3,6),(15,17)\}$. The generate path is: $\{1- > 2- > 3- > 6- > 5- > 8- > 9- > 10- > 14- > 13- > 12- > 15- > 17- > 18- > 19- > 23- > 22- > 21- > 20- > 16- > 11- > 7- > 4- > 1\}$. By traversing the generated path, the agent incurs a cost of 27 units.

From Figure 2b, it is evident that path generated for agent $\mathcal{A}$ has following salient features to ensure its validity:

1) Depot: The agent $\mathcal{A}$ starts and ends at the same region (depot, region = 1).
2) Sub-Tour: Apart from the cycle which starts and ends at depot, there are no other cycle present in the path.
3) Specified Route: The agent traverse all the edges in the specified route $\Theta = \{(3,6),(15,17)\}$.
4) Trajectory Minimization: The constraint of time and energy is met, as no region is traversed more than once in the generated path.

Therefore, it is evident from the given example that the MILP strategy is able to generate a valid path which meets all the specified constraints of the inspection problem.

D. IRPP: INTER REGION PATH PLANNING TECHNIQUE

The MILP based inter-region path planning solution described in the previous section generates optimal solution. However, its' exponential computational complexity makes it expensive, as the number of region increases. Therefore, in this section, we propose an efficient heuristic algorithm, *'IRPP'*, which generates acceptable solution to the problem even when the number of regions are fairly large. The pseudocode of IRPP is presented in Algorithm 2.

The algorithm is a heuristic approach, which attempts to inspect the specified order route $(i,j) \in \Theta'$ first. This in turn minimizes the traversal cost, as the penalties for not traversing arcs $(i,j) \in \Theta'$ are not imposed. It also takes in consideration that self loop and sub-tours are eliminated. An example at the end of this section is illustrated to elucidate our proposed approach.

1) IRPP Strategy

The main motivation of the inter-region path planning heuristic is to determine the order of region traversal such that:

1) Minimum cost of traversal is incurred.
2) All the regions are traversed once and agent $\mathcal{A}$ starts and ends at depot.
3) Specified order of traversal is the route under taken by the agent $\mathcal{A}$.

The algorithm comprises of four blocks: 1) Initialization, 2) Order Generation, 3) Feasible Solution, and 4) Cost Calculation. Each block is discussed in details as follows:

Initialization (Line 1-4): In this block, we initialize four variables: m, χ, n, Z. Variable m and n denote number of regions and number of specified edges, respectively. χ and Z are the arrays containing the order of region traversal which are initialized as $NULL$.

Order Generation (Line 5-7): To meet the three requirements, IRPP heuristic attempts to start the route by traversing the arcs $(i,j) \in \Theta'$ in parallel. As shown in line 5-7, in the order generation block from Algorithm 2, an order of region traversal Z_q for each arc $q \in \Theta'$ is generated by using function *NEXT_REGN ()*. Since, there is no correlation between paths generated for each arc $q \in \Theta'$, each order generation is computed in parallel.

Feasible Solution (Line 8-20): After obtaining a list of order of inter-region traversal Z from last block, we search for all feasible solutions among the generated solutions of inter-region, Z. The solution is classified as feasible if, and only if, it meets the second motivation, i.e. each region is traversed once and start and ending region are same. To determine the feasible solution, we check in line 8-20, whether the generated order contains all the m distinct regions and the starting and ending region are identical. Those solutions which are unfeasible are discarded in line 18. This is helpful in saving the computation time in the next block, the *cost calculation* block.

Cost Calculation (Line 21-25): In the cost calculation block, we determine the cost of traversal which the agent $\mathcal{A}$ will incur if it takes the feasible route. The cost of traversal is calculated in line 22, Algorithm 2 by summing the individual cost of arcs in the feasible order of traversal. In line 24, we select a feasible solution from Z which has minimum cost. In the selected minimum cost feasible solution, the order of traversal begins with the region i for arc $(i, j) \in \Theta'$. However, for our scenario, order of traversal must begin from the depot. Thus, we re-order the order of traversal such that traversal of route begins from depot.

2) Next Region Function: NEXT_REGN ()

IRPP calls function NEXT_REGN () to generate the order of traversal for all the arc $q = (i, j) \in \Theta'$. NEXT_REGN () returns the feasible solution Z_i. The function is composed of three components: *VSRF, NNV and TERMINATION*. The pseudocode for NEXT_REGN () is presented in Algorithm 3.

VSRF: Visit Specific Route First (Line 1-3): In the VSRF phase, the agent $\mathcal{A}$ visits the specified route $(i, j) \in \Theta'$ first. After reaching point j, it must take a decision to visit any one of the neighbours of j which is taken in the next phase, NNV. However, a decision point variable $next_{region}$ is initialized to j in this phase. Another variable list ρ_{cost} which assists in taking the decision of which next region to traverse after j is initialized to NULL. ρ_{cost} maintains the cost incurred for traversing each neighbour k after region j.

NNV: Next Neighbour Visit (Line 4-18): The phase next neighbour visit is responsible for taking the decision of traversing which neighbour of the current region j. First, we compute all those neighbours of j, which have not been previously visited. From the unvisited neighbours, the heuristic can either execute the **look ahead plan** or the **cost calculative plan**. The agent $\mathcal{A}$ finds out whether one of following two criteria is met:

1) **CRITERION I:** Neighbouring region k has an arc $(j, k) \in \Theta'$, i.e. by traversing route (j, k) it meets the specific inspection order in Θ'.

Algorithm 2 IRPP

```
Input: 1. $\mathcal{M} = \{1, .., m\}$ : set of $m$ regions
       2. $\Theta' = \{(i, j) \mid i, j \in \mathcal{M}\}$ : order of sites to be inspected
       3. $c_{m,m}$ : cost of $\mathcal{M}$ inter-region traversal
Output: $Cost, \chi$: cost and order of inter-region traversal
                    INITIALIZATION
 1:    $m = |M|$
 2:    $\chi = NULL$
 3:    $n = |\Theta'|$
 4:    $Z = NULL$ /*$\chi$, $Z$ denotes order of inter-region traversal and sequence starting from inspection edge, respectively.*/
                    ORDER GENERATION
 5: for each arc $q \in \Theta'$ in parallel do
 6:    $Z_q$ = NEXT_REGN($q$); /*Order of traversal $Z_q$ starting from arc $q$*/
 7: end for
                    FEASIBLE SOLUTION
 8: for each solution $Z_q \in Z$ do
 9:    if Order of traversal in $Z_q$ containts all $m$ distinct regions then
10:       if $Z_q$ finishes at the start region then
11:          $Z_q$ solution is Feasible
12:       else
13:          $Z_q$ is not feasible solution;
14:          Discard $Z_q$;
15:       end if
16:    else
17:       $Z_q$ is not feasible solution;
18:       Discard $Z_q$;
19:    end if
20: end for
                    COST CALCULATION
21: for each feasible solution $Z_q \in Z$ do
22:    $cost_{Z_q} = \sum_{(j,k) \in Z_q} c_{j,k}$;
       /*Calculate the cost $cost_{Z_q}$ as sum of the cost of all the arcs $j$ in the feasible solution $Z_q$*/
23: end for
24: $Cost$ = $min\{cost_{Z_1}, cost_{Z_2}, \ldots, cost_{Z_q}\}$;
    /*Assign $Cost$ with minimum cost solution in $Z$*/
25: $\chi$ = re-order the minimum cost traversal route such that inter-region tour begins from depot;
```

2) **CRITERION II:** Neighbour l of neighbouring region k has an arc $(k, l) \in \Theta'$, i.e. by traversing route (k, l) it meets the specific inspection order in Θ'.

Even if one of the criteria is met, a look-ahead plan is executed, where the agent traverses region k irrespective of cost to meet the stringent criteria of specified route travel. In case none of criteria of route travel are met, we go ahead with cost calculative plan where we select the next neighbour to traverse based on the cost of traversal from region j to k, $c_{j,k}$. The cost for each neighbour k of current region j is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3044987, IEEE Access

stored in the previously defined list ρ_{cost} as follows:

$$\rho_{cost} = \rho_{cost} \cup c_{j,k} \tag{22}$$

The main challenge stems from the equal cost of traversal of all the neighbours. For instance, if the agent traverses the edge $(3, 6)$. If the agent executes the policy of selecting vertex based on its occurrence first which is generally employed in many heuristics like next neighbour first [47], the path generated contains cycle or sub tours. To overcome this challenge, we deploy *'Away From Sink'* strategy. Since, the agent needs to traverse the entire tour and return to the starting region, we label the start region as the *'sink'*. The *'Away from the sink'* strategy allows the agent to traverse the neighbour which is farthest from the sink, so that all the regions are traversed once, without the presence of any subtours.

TERMINATION (Line 19-26): In this phase, we find out whether in the last phase the agent $\mathcal{A}$ finds that all the neighbouring regions have been previously visited. This is indicative of the fact that the tour is complete. However, if the agent visits unvisited neighbouring regions and computes ρ_{cost} list, then we select the neighbour k to be visited which has minimum cost $c_{j,k}$ in the ρ_{cost} list. Based on the selected region k, a new arc (j, k) is added to the solution Z_q. To further elucidate the working of our heuristic, we present an example in the following subsection.

E. EXAMPLE 2: ILLUSTRATING WORKING OF INTER-REGION PATH PLANNING HEURISTIC

The construction heuristic *'IRPP'* is illustrated through a working example in this subsection. Let us consider the same extreme environment site as previously discussed in Section IV-C, i.e. number of regions $m = 23$ and specified order $\Theta = \{(3, 6), (15, 17)\}$. As per the *'VSRF: Visit Specific Route First'* block of Algorithm 3, the agent will first traverse edges in the specified order, i.e. $(3, 6)$ and $(15, 17)$.

The cost of traversal from one vertex to another is equal. For instance, if the agent traverses the edge $(3, 6)$. As per Figure 2a, by being stationed at vertex 6, the agent $\mathcal{A}$ has an option to traverse either vertex 5 or 10. Since the cost of traversal between neighbouring vertex is a unit cost, it takes the policy of *'away from sink'*. The path generated for agent $\mathcal{A}$ is $\{1- > 2- > 3- > 6- > 10- > 14- > 13- > 12- > 15- > 17- > 18- > 19- > 23- > 22- > 21- > 20- > 16- > 11- > 7- > 8- > 9- > 5- > 4- > 1\}$, as shown in Figure 3.

The *'away from sink'* is executed when the agent traverses the path $(3- > 6- > 10)$ and is stationed at vertex *'10'*. As illustrated in Figure 2a, vertex *'10'* has two equal cost neighbour vertices *'9' and '14'*. If the agent uses the traditional next neighbour methodology, it would have selected region *'9'*, which will result in generation of a cycle in the path. Since, the agent starts from the region *'3'* and the tour will end by traversing all the regions once and end at the region *'3'*. We classify the region *'3'* as sink vertex.

Algorithm 3 NEXT_REGN

```
Input: 1. $\mathcal{M} = \{1, .., m\}$ : set of m regions
       2. $\Theta' = \{(i, j) \mid i, j \in \mathcal{M}\}$ : order of sites to be inspected
       3. $c_{m,m}$ : cost of $M$ inter-region traversal
Output: $Z_q$: order of inter-region traversal
    Function NEXT_REGN:
              VSRF: Visit Specific Route First
 1: Visit the arc $q$ using VISIT_ARC(q)
 2: Initialize decision point $next_{region}$ as $j$;
    /* $next_{region}$ denotes the next region to visited after j*/
 3: Initialize the neighbour cost list $\rho_{cost}$ as $NULL$;
              NNV: Next Neighbour Visit
 4: for each neighbor $k$ of $j$ which are unvisited do
 5:   $l$ : neighbours of $k$;
 6:   if CRITERION I OR CRITERION II satisfied then
        /*****LOOK-AHEAD PLAN****/
 7:     Assign $k$ as $next_{region}$;
 8:   else
 9:     /*****COST-CONSTRUCTIVE PLAN*****/
10:     if (traversal of $(j, k)$ is the minimum cost among all
        the neighbours) then
11:       Add $c_{j,k}$ to the $\rho_{cost}$ list
12:       /****AWAY FROM SINK***/
13:       if $k$ is farthest from sink vertex then
14:         Add $c_{j,k}$ to the $\rho_{cost}$ list
15:       end if
16:     end if
17:   end if
18: end for
                    TERMINATION
19: if no new neighbour visited then
      /**AGENT HAS COMPLETED THE TOUR**/
20:   return $Z_q$;
      /*Return generated solution $Z_q$*/
21:   STOP;
22: else
23:   if $\rho_{cost} \neq NULL$ then
24:     Assign $next_{region}$ = $k$ such that $k$ has minimum
        cost in $\rho_{cost}$;
25:   end if
26: end if
27: Visit the arc $q = (j, next_{region})$ using VISIT_ARC($q$)
28: NEXT_REGN($q_{i+1}$;)
    Function VISIT_ARC:
29: ADD arc $q = (i, j)$ to $Z_q$
```

The main aim of the inter-region path planning technique is to generate a path where the agent $\mathcal{A}$ traverses all the regions at least once with out getting stuck in any sub-tours. If the agent moves from region *'10'* to region *'9'*, it moves closer to sink signifying that it is cutting a path short, i.e. not traversing all regions. Therefore, we select the next region to be traversed, which is farthest from the sink vertex *'3'*, i.e. vertex *'14'*. While being stationed at vertex *'12'*, the agent

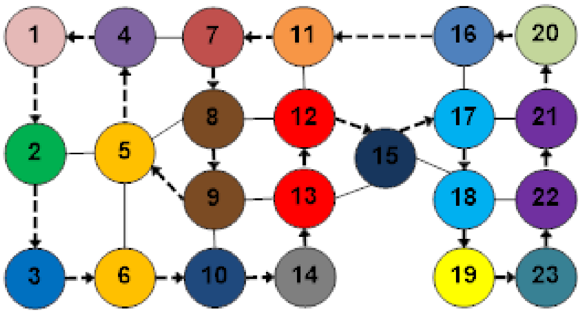


FIGURE 3: Inter-region path planning generated using our proposed heuristic *'IRPP'* for the site depicted in figure 2a

executes the *'look-ahead'* plan and move to region *'15'*, which allows it to travel the specified edge $(15-17) \in \Theta$.

Using the 'IRPP' heuristic, we illustrate the generated inter-region path in Figure 3. It is evident from the figure that the generated path differs from the solution proposed by the MILP Strategy, however, it is able to achieve the same, $Cost = 27$.

V. INTRA-REGION PATH PLANNING

After computing the inter-regional path for agent $\mathcal{A}$ in previous Section IV, we tackle the problem of computation of coverage path for each region $i \in \mathcal{M}$ in this section. The order of the inter-region path χ plays a pivotal role in determining the intra-region paths. The intra-regional path is computed first by decomposing the regions into the grids and then computing Boustrophedon Motion for each region as described in the following subsections.

A. GRID DECOMPOSITION

For computation of any path, the starting and ending locations are the necessary inputs. Similarly, for computing a coverage path for any region i, we first and foremost determine the entrance and exit for all the regions i. Thus, to find an optimal intra-region path for any region $i \in \mathcal{M}$, we take into consideration the preceding and succeeding regions to determine the start and end location, respectively.

As known, a rectangular region i is characterized by four vertices: $\{v_{i1}, v_{i2}, v_{i3}, v_{i4}\}$. If $i-1$ is the preceding region for region i, the entrance for the region i is the vertex connecting both the regions i and $i-1$. For instance, let us consider there are four regions as shown in Figure 4 characterized by their four vertices. The optimal inter-region path generated is $\{1->2->3->4->1\}$. For region 2, the preceding region is 1 so the entrance point can either be v_{24} or v_{21}. Similarly, if $i+1$ is the succeeding region, the exit for the region i is the vertex connecting the regions i and $i+1$. Now, in Figure 4, the succeeding region is region 3. Therefore, the exit point from region 2 is either v_{24} or region v_{23}.

After determining the entrance and exist, we decompose the region into a collection of grid cells. Let each region i be decomposed into a grid Ψ_i with a set of uniform grid cells

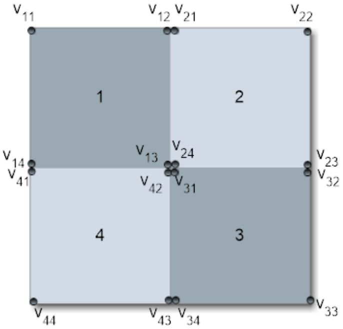


FIGURE 4: An example site with four regions with their corner vertices

such that

$$\Psi_i = \{\psi_{jk} | 1 \leq j \leq \frac{l_i}{\lceil \frac{l_i}{r} \rceil}, 1 \leq k \leq \frac{b_i}{\lceil \frac{b_i}{r} \rceil}\} \qquad (23)$$

where, $l_i = ||v_{i4} - v_{i3}||$ and $b_i = ||v_{i2} - v_{i3}||$ is the length and breadth of each region i, respectively, $||.||$ is a L_2 norm operator and $r \times r$ is the sensing radius of agent $\mathcal{A}$. The number of grid cells, in region i is $|\Psi_i| = \lceil \frac{l_i}{r} \rceil \lceil \frac{b_i}{r} \rceil$. Since $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} \leq r$ and $\frac{b_i}{\lceil \frac{b_i}{r} \rceil} \leq r$, i.e., size of each grid cell $\psi_{jk} \in \Psi$ is either smaller or equal to the sensing range of the agent. This implies that if an agent visits a grid cell, it is able to successfully map or cover the entire area of grid cell in a single visit. If we consider each cell as a region and the problem of intra-region traversal to tackle translates to how can an agent cover all the cells once, with different start and end location.

B. BOUSTROPHEDON MOTION

After decomposing each region into a grid and determining the starting and end point for each region, we aim to compute the coverage path for the intra region traversal. According to the recent survey [2], two important categories of CPP techniques are *'Boustrphedon CPP'* and *'Grid based Travelling Salesman CPP'*. *Boustrophedon CPP* technique is an exact cellular decomposition technique which decomposes the site into smaller cells using a sweep line. After each cell is obtained, the orientation of each cell is analysed and back-forth motion (also known as *'Boustrophedon Motion'*) is setup.

In *'Grid based TSP'*, unlike *'Boustrphedon Path Planning'*, the site is decomposed into a regularly spaced grid with a uniform grid cell size. The CPP problem now translates into generating a TSP tour, i.e. traversing each cell once and starting and ending at same cell location. In our case, the starting and ending location for the intra-region traversal are not same, therefore we combine both the techniques and tread a different route.

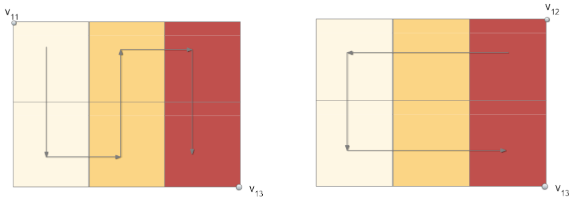
After decomposing each region into a grid and determining the start and end vertices as described in last subsection, we compute a Boustrophedon Motion between the start and end location based on the orientation using Algorithm 4.

We first analyse the orientation of start and end vertices. If the start and end vertices are located diagonally opposite, the coverage path is a vertical Boustrophedon Motion as shown in Figure 5a. The start and ending vertices v_{11} and v_{13}, respectively, are diagonally opposite, thus Algorithm 4 generates a horizontal back and forth motion. Similarly, if the start and end vertices are located on the same edge, the coverage path is a horizontal Boustrophedon Motion. As illustrated in Figure 5b, the start and end vertices v_{12} and v_{13}, respectively, share a common edge, therefore a horizontal back and forth coverage path is generated. Now, we elucidate that the grid-decomposition based Boustrophedon intra-regional path planning approach is able to provide full coverage using Theorem 1.

Algorithm 4 BOUSTROPHEDON MOTION

```
Input:  1. $\Psi_i = \{\psi_{jk}\}$ : Grid $\Psi_i$ representing region $i$
        2. $v_{ss'}$ : Starting vertex
        3. $v_{ee'}$ : Ending vertex
Output: $\mathcal{P}$: coverage path

ORIENTATION ANALYSIS
if $|s' - e'| > 1$ then
    /*Starting and Ending Vertices are Diagonally Opposite*/
    $\mathcal{P}$: Generate a Back and Forth Vertical Motion
else
    /*Starting and Ending vertices lie on same edge*/
    $\mathcal{P}$: Generate a Back and Forth Horizontal Motion
end if
```



(a) (b)

FIGURE 5: Boustrphedon Motion for (a) diagonally opposite vertices (b) vertices with same edge

Theorem 1 (*Full Coverage of Grid based Decomposition with Inter-Region Path Planning*:)
Consider the intra-regional path planning problem described above. The intra-region path formulated by grid decomposition approach provides full coverage if $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} = \frac{b_i}{\lceil \frac{b_i}{r} \rceil} = r, \forall i \in \mathcal{M}$ and each grid cell is regarded as region.
Proof: If $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} = \frac{b_i}{\lceil \frac{b_i}{r} \rceil} = r$, the size of each grid cell equals the sensing rage of the agent $\mathcal{A}$, this implies that the shortest paths to plan through the centres of all the cells in any region $i \in \mathcal{M}$ is equivalent to the shortest path to cover the entire region i. Therefore, when applying grid decomposition along with the inter-region path planning approach, the grid decomposition based path planning allows full coverage of the target site.

■

VI. EXPERIMENTAL EVALUATION

In this section, we perform simulation studies to evaluate the performance of our proposed approach in terms of *'full coverage', 'efficiency' and 'comparison with existing work'*. All the algorithms are implemented using MATLAB and IBM-CPLEX, and the experiments are performed on a DELL XPS IDV8QVO with Intel Core i5, 8GB memory and 225GB storage [1].

A. EXPERIMENTAL SETUP

The input data for our experiments have been generated using the following framework.

Target site generation: The data corresponding to the regions for a site which needs to be covered has been generated randomly. The number of regions in the sites is given by the user with an upper bound of 50 regions. This assumption is based on the limit of an aerial agent to perform coverage with given energy budget [13]. The regions in the site are assumed to be distributed around an origin marked by the coordinates (0,0). A new region is created by first tentatively selecting a centroid for it. The coordinates of this centroid are obtained by selecting its distance from the origin through a random distribution U1 $[0, 200]$ and also determining the angle that it subtends on the x axis, from another uniform random distribution U2 $[0, 360]$.

A new region is created using function $rectangle()$ [48]. This function takes as input the coordinates of the centroid (obtained as above), width and height of each region to be generated. $rectangle()$ returns a rectangle based on the region parameters provided as input. The width and height of a region are generated from normal distribution having standard deviation as $\sigma_{side} = 2.5$ and the mean as $\mu_{side} = 7.5$. The new region is actually accepted if it does not overlap with any other region already generated; otherwise, it is discarded. We repeat these steps unless and until desired number of regions are created. In case of a rejection, the steps to create a new region are repeated. The above steps are continued until the required number of regions have been generated. The size of a region is measured in terms of number of grid cells that it contains. The size of each grid cell is assumed to be $1 \times 1\ m^2$.

Specific Order generation: As discussed above, the overall operation has an associated order which provides precedence provision for coverage. The specified order Θ is obtained where first we obtain the number of specified edges which the agent must traverse using normal distribution having standard deviation as $\sigma = \frac{m}{8}$ and the mean as $\mu = \frac{m}{4}$. The specified pair of edge in Θ is selected by uniform distribution U3 $[1, m]$. If the edge pair already exists in Θ, this edge pair is discarded.

[1] The code for all algorithms can be found at: https://github.com/ZebaKhanam91/CoveragePathPlanning

Agent Characteristics: The agent is assumed to have square footprints whose areas are expressed as an integral number of grid cells. The sensing range of agent r can take different values ranging from $\{1, 2, 3, 4\}$.

B. FULL COVERAGE

Theorem 1 in Section V-B shows that our proposed technique can compute full coverage path for each region, if grid-based decomposition is coupled with inter-region traversal. The inter-region traversal generates a path under the minor assumption that the connectivity graph generated in Section IV-A contains at least one Eulerian trail, i.e. there exits a cycle where each edge is traversed only once. The minor assumption for intra-region traversal is that $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} = \frac{b_i}{\lceil \frac{b_i}{r} \rceil} = r$. We study the optimality of the proposed inspection approach using the following experiment designed to elucidate the ability of our approach to find the solution when the above mentioned assumption is either satisfied or violated.

1) Experiment I

The Experiment I is designed to investigate the assumption of the inter-region traversal, i.e. *'there exists at least one Eulerian trail'*. The first experiment is carried out under a two site setting depicted using the connectivity graphs in Figure 6. In the first and second setting, we determine the path when the assumption is satisfied and relaxed, respectively. The first experiment contains three regions ($\mathcal{M} = 1, 2, 3$), which have equal length and width, such that $l_i = b_i = 4$, $\forall i \in \mathcal{M}$. The sensor range of the agent $\mathcal{A}$ is $r = 2$. Thus, $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} = \frac{b_i}{\lceil \frac{b_i}{r} \rceil} = r = 2$, holds for both cases.

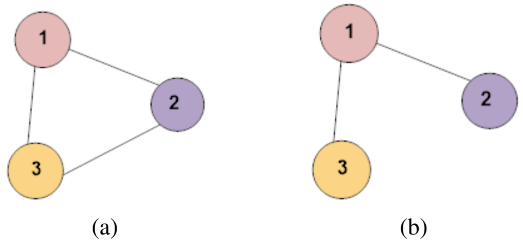


FIGURE 6: Experiment I site (a) containing eulerian trail (b) without eulerian trail

The order of inspection for the site is $\Theta = (1, 2)$. The distance between each region is unit cost. CPLEX and IRPP heuristic both generate the path $\{1- > 2- > 3- > 1\}$ for the first setting with Eulerian trail with $cost = 3\ units$. The optimal coverage path is depicted in Figure 7a. To illustrate the performance of CPP algorithm when the underlying assumption of inter-region traversal is violated, the region *'2'* and *'3'* are not connected. The remaining settings like inspection order and cost traversal remain same. The path generated by the CPLEX and IRPP heuristics $\{1- > 2- > 1\}$, which is a sub-optimal solution as region *'3'* is not traversed. The provision of correlation between work (precedence provision) forces the agent to traverse the edge $1- > 2$. Since, the region *'2'* is only connected to region *'1'*, the agent traverses back to the depot 1. Now, it can traverse to region 3, however it is unable minimize coverage trajectory. This allows us to conclude that if the underlying assumption of inter-region path planning, connectivity graph $\mathcal{G}$ derived using Algorithm 1 contains Eulerian trail, is violated then the computed coverage path does not cover the entire site.

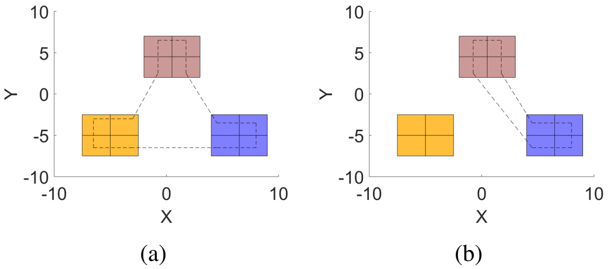


FIGURE 7: Full coverage path found when site (a) containing Eulerian trail (b) without Eulerian trail

2) Experiment II

The Experiment II is designed to investigate the assumption of the intra-region traversal, i.e. $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} = \frac{b_i}{\lceil \frac{b_i}{r} \rceil} = r$. The second experiment is carried out under two site setting depicted using the connectivity graphs in Figure 6 which was considered in Experiment I. However, the sensor range is increased $r = 3$, thus violating the constraint $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} \neq \frac{b_i}{\lceil \frac{b_i}{r} \rceil} \neq r$. Under this setting, the optimal path generated is similar to the path generated in Experiment I illustrated in Figure 7. This allows us to conclude that even if the minor assumption is violated, intra-region path traversal generates a full coverage path solution, this can be attributed to the combination of grid decomposition with Boustrophedon Motion.

3) Experiment III

The experiment III is designed to investigate the performance of inspection algorithm, when unlike previous experiments, the regions of traversal are not similar. The connectivity graph of the site is shown in Figure 8a. The region *'1'* is the largest region of size 6×4 and the other two regions are of size 2×2. The range of sensor is set to $r = 2$. The inter-region path generated using CPLEX and IRPP is $\{1- > 2- > 3- > 1\}$. In intra-region traversal, the region *'2'* and region *'3'* have a single grid cell and since $\frac{l_i}{\lceil \frac{l_i}{r} \rceil} = \frac{b_i}{\lceil \frac{b_i}{r} \rceil} = r$, the assumption holds true, thus visiting the centre of the grid cell is sufficient to cover the entire grid cell. Similarly, region *'1'* is decomposed into 3×2 grid cells using Algorithm 4 , the agent $\mathcal{A}$ executes a vertical Boustrophedon Motion.

C. EFFICIENCY STUDY

In this subsection, we investigate the efficiency of the proposed approach. As the approach consists of two phases: inter region and intra-region path planning. We design the

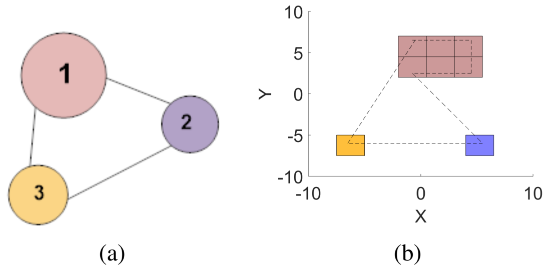


FIGURE 8: Full coverage path found when site (a) containing eulerian trail (b) without eulerian trail

following experiments to study the properties of each phase with respect to region size and number of regions.

1) Experiment IV

In Experiment IV, we study the properties of two proposed techniques for inter-region traversal: the MILP based technique (proposed in Section IV-B) and IRPP heuristic (proposed in Section IV-D) under varying number of regions. We consider a set of uniform regions of size $l_i = 2$, $b_i = 4$, the sensor range is set to $r = 2$. The order of execution $\Theta = (2, 1)$. As shown in Figure 9, the execution time of both the MILP based technique and the *'IRPP'* heuristic based approach increases with the increment in number of regions indicating that both the proposed techniques suffer from the curse of dimensionality with respect to number of regions. However, if we compare the performance of both the techniques from Figure 10, we can conclude that the MILP based technique yields a path with less cost as compared to 'IRPP' based heuristic with increase in the number of regions. In order to achieve this more computation time is required as the number of regions increase (see Figure 10).

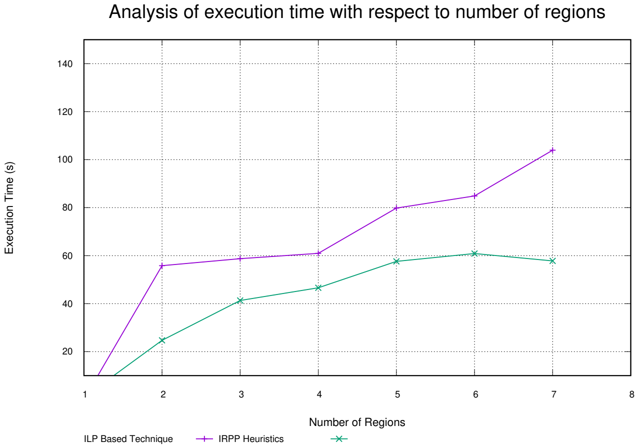


FIGURE 9: Analysis of execution time for Inter-region traversal with respect to number of regions

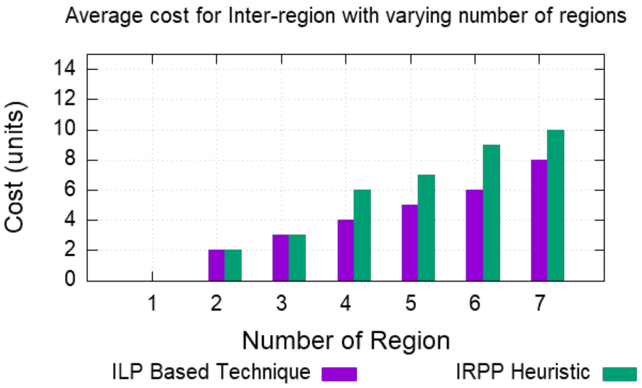


FIGURE 10: Analysis of cost for Inter-region traversal with respect to number of regions

2) Experiment V

Experiment V is designed to investigate the efficiency of the proposed intra-region traversal technique under varying region sizes. In this experiment, two regions are considered with length of each region set to be $l_i = 2, i \in \{1, 2\}$. The sensor range is set to $r = 2$. The breadth of each region is then increased simultaneously. The inter-region path generated using both the inter-region traversal techniques is $\{1- > 2- > 1\}$ with $cost = 2\, units$.

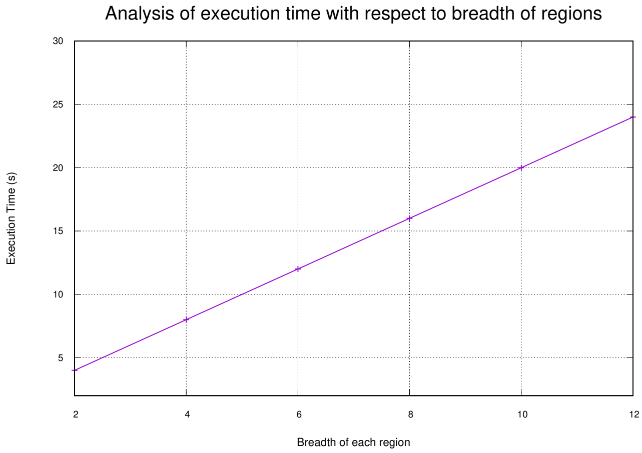


FIGURE 11: Analysis of execution time for Intra-region traversal with respect to breadth of regions

It is observed from Figure 11 that the execution time varies linearly with the increment in breadth of region. This trend can be attributed to the fact that as breadth of each region increases, the number of grid cells increase. Therefore, this leads to overall increase in execution time required to traverse each region indicating scalability to region size.

3) Experiment VI

Experiment VI is designed to show the capability of the proposed approach in carrying out inspection of relatively large scale sites. It was suggested in [13] that agents moving at a fixed altitude can traverse up to 17 regions in a single trip. In Figure 11, we had deduced that the execution time for intra-region traversal is directly proportional to number of grid cells. Figure 12 observes a trend that execution time for the inter-region traversal using IRPP heuristics increases with the increment in the number of regions. Any large scale site can have large number of regions. However, in a recent survey [2], it was stated that coverage problem is computed on a part of map rather a complete map. We assume the upper limit of number of regions in a partial map for computation of coverage to be 50 such that $M = \{1, 2, \ldots, 50\}$. We calculated the execution time for the MILP technique to be 889.58s. This illustrates that MILP is capable of generating a feasible solution for even the large scale sites. The IRPP heuristic based technique takes up to 140.7s to generate a feasible solution.

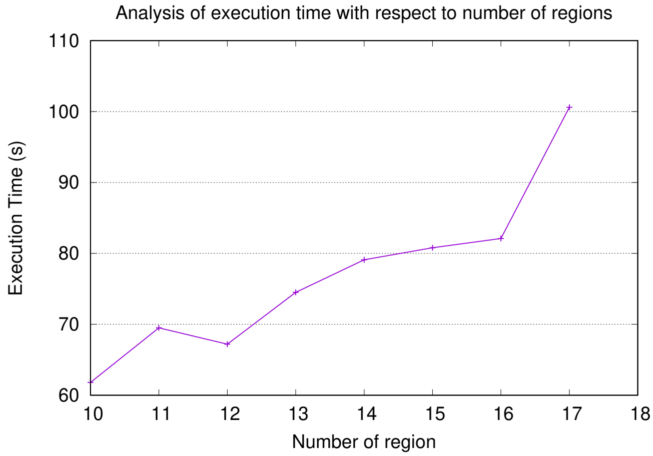


FIGURE 12: Analysis of execution time for inter-region traversal with respect to number of regions for large scale sites

D. COMPARISON WITH EXISTING WORKS

In this section we evaluate the performance of our proposed approach with respect to following approaches of CPP of disjoint regions:

1) TSPP [5]: This method first derives inter-region path planning using genetic algorithm [49] and then determines intra-regional path.
2) Dynamic Programming [13]: The coverage problem is solved using integrated approach of dynamic programming.
3) Extended Genetic Algorithm [40]: The coverage problem is used as an integrated problem using genetic algorithm [49].

In this comparison study, genetic algorithm is implement based on the code in [50] where the parameter setting were tuned to be: 1) Number of iterations: 10^4, 2) Population size: 100, and 3) termination when fitness score does not change for 10 iterations. The dynamic algorithm was implemented based on code in [51].

We compare the performance of our work with the above mentioned works in terms of coverage cost. Figure 13 shows a trend that when the number of regions are small (less than or equal to 3) the performance of all the approaches are similar. However, as the number of regions increase, the coverage cost of existing works increases. The main reason that they incur a large cost is because they fail to follow the order as observed in Figure 13. Keeping up with the trend in respect to coverage cost, we can conclude that our approach with precedence provision outperforms all the set of works [5], [13], [40].

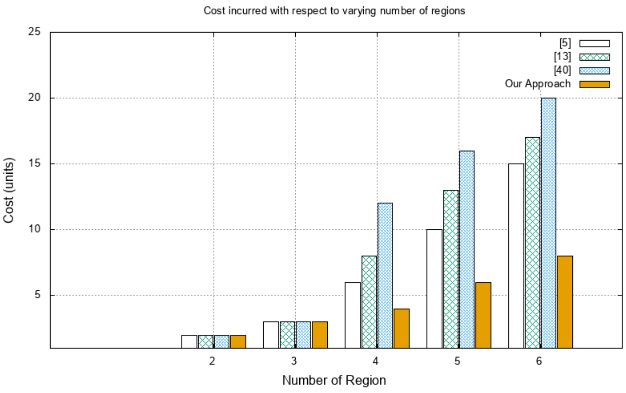


FIGURE 13: Cost incurred with respect to number of regions

VII. CONCLUSION

In this paper, new CPP techniques for inspection of sites with disjoint regions and precedence provision were mathematically formulated. The problem was first decoupled as inter-region path planning and intra-region path planning. MILP and heuristic based technique are proposed to solve the inter-region path planning. For intra-region path planning, each region is decomposed into a grid and Boustrophedon Motion is planned over each region. The combination of both the approaches generates a full coverage path is proved under minor assumption. The simulation studies are carried out to study the full coverage and efficiency of the proposed approach in different scenarios. A comparison study with the state-of-the-art techniques revealed that our approach outperformed them in terms of coverage cost incurred.

ACKNOWLEDGEMENT

This work is supported by the UK EPSRC through grants EP/R02572X/1 and EP/P017487/1.

REFERENCES

[1] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[2] R. Bormann, F. Jordan, J. Hampp, and M. Hägele, "Indoor coverage path planning: survey, implementation, analysis," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1718–1725.

[3] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[4] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, "Multi-uas path-planning for a large-scale disjoint disaster management," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 799–807.

[5] J. I. Vasquez-Gomez, J.-C. Herrera-Lozada, and M. Olguin-Carbajal, "Coverage path planning for surveying disjoint areas," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 899–904.

[6] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *12th Conference on Field and Service Robotics (FSR 2019)*, 2019.

[7] B. Aslam, S. Saha, Z. Khanam, X. Zhai, S. Ehsan, R. Stolkin, and K. McDonald-Maier, "Gamma-induced degradation analysis of commercial off-the-shelf camera sensors," in *2019 IEEE SENSORS*. IEEE, 2019, pp. 1–4.

[8] Z. Khanam, S. Saha, B. Aslam, X. Zhai, S. Ehsan, C. Cazzaniga, C. Frost, R. Stolkin, and K. McDonald-Maier, "Degradation measurement of kinect sensor under fast neutron beamline," in *2019 IEEE Radiation Effects Data Workshop*. IEEE, 2019, pp. 1–5.

[9] N. Pinkam, A. A. R. Newaz, S. Jeong, and N. Y. Chong, "Rapid coverage of regions of interest for environmental monitoring," *Intelligent Service Robotics*, vol. 12, no. 4, pp. 393–406, 2019.

[10] C. Pan, M. Zhou, Y. Qiao, and N. Wu, "Scheduling cluster tools in semiconductor manufacturing: Recent advances and challenges," *IEEE transactions on automation science and engineering*, vol. 15, no. 2, pp. 586–601, 2017.

[11] B. Bird, A. Griffiths, H. Martin, E. Codres, J. Jones, A. Stancu, B. Lennox, S. Watson, and X. Poteau, "A robot to monitor nuclear facilities: Using autonomous radiation-monitoring assistance to reduce risk and cost," *IEEE Robotics & Automation Magazine*, vol. 26, no. 1, pp. 35–43, 2018.

[12] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. Polycarpou, "Jointly-optimized searching and tracking with random finite sets," *IEEE Transactions on Mobile Computing*, 2019.

[13] J. Xie, L. R. G. Carrillo, and L. Jin, "An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems," *IEEE control systems letters*, vol. 3, no. 1, pp. 67–72, 2018.

[14] Y. Yuan, D. Cattaruzza, M. Ogier, and F. Semet, "A note on the lifted miller–tucker–zemlin subtour elimination constraints for routing problems with time windows," *Operations Research Letters*, 2020.

[15] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset, and A. A. Rizzi, "Uniform coverage of automotive surface patches," *The International Journal of Robotics Research*, vol. 24, no. 11, pp. 883–898, 2005.

[16] F. Yasutomi, M. Yamada, and K. Tsukamoto, "Cleaning robot control," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 1839–1841.

[17] D. Zhu, C. Tian, B. Sun, and C. Luo, "Complete coverage path planning of autonomous underwater vehicle based on gbnn algorithm," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 1, pp. 237–249, 2019.

[18] M. Đakulovic and I. Petrovic, "Complete coverage path planning of mobile robots for humanitarian demining," *Industrial Robot: An International Journal*, 2012.

[19] T. H. Pham, Y. Bestaoui, and S. Mammar, "Aerial robot coverage path planning approach with concave obstacles in precision agriculture," in *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2017, pp. 43–48.

[20] I. Tsitsimpelis, C. J. Taylor, B. Lennox, and M. J. Joyce, "A review of ground-based robotic systems for the characterization of nuclear environments," *Progress in Nuclear Energy*, vol. 111, pp. 109–124, 2019.

[21] T. Kelley, K. Hutchings, M. Jenkinson, I. Dawson, K. Krawec, P. Miyagawa, J. Kern, and R. Edge, "In situ monitoring of pvdf ultrasound transducers under gamma irradiation," *Nondestructive Testing and Evaluation*, pp. 1–15, 2019.

[22] S. Kawatsuma, M. Fukushima, and T. Okada, "Emergency response by robots to fukushima-daiichi accident: summary and lessons learned," *Industrial Robot: An International Journal*, 2012.

[23] S. Rizzolo, V. Goiffon, F. Corbière, R. Molina, A. Chabane, S. Girard, P. Paillet, P. Magnan, A. Boukenter, T. Allanche *et al.*, "Radiation hardness comparison of cmos image sensor technologies at high total ionizing dose levels," *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 111–119, 2018.

[24] F. Datchi, A. Dewaele, P. Loubeyre, R. Letoullec, Y. Le Godec, and B. Canny, "Optical pressure sensors for high-pressure–high-temperature studies in a diamond anvil cell," *High Pressure Research*, vol. 27, no. 4, pp. 447–463, 2007.

[25] V. Goiffon, F. Corbière, S. Rolando, M. Estribeau, P. Magnan, B. Avon, J. Baer, M. Gaillardin, R. Molina, P. Paillet *et al.*, "Multi-mgy radiation hard cmos image sensor: Design, characterization and x/gamma rays total ionizing dose tests," *IEEE Transactions on nuclear science*, vol. 62, no. 6, pp. 2956–2964, 2015.

[26] R. Almadhoun, T. Taha, L. Seneviratne, J. Dias, and G. Cai, "A survey on inspecting structures using robotic systems," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 1729881416663664, 2016.

[27] S. Jung, S. Song, P. Youn, and H. Myung, "Multi-layer coverage path planner for autonomous structural inspection of high-rise structures," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[28] L. Jin, S. Li, X. Luo, Y. Li, and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3812–3821, 2018.

[29] X. Yin, Z. Li, L. Zhang, and M. Han, "Distributed state estimation of sensor-network systems subject to markovian channel switching with application to a chemical process," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 6, pp. 864–874, 2018.

[30] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6648–6653.

[31] T. Nguyen, H. M. La, T. D. Le, and M. Jafari, "Formation control and obstacle avoidance of multiple rectangular agents with limited communication ranges," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 4, pp. 680–691, 2016.

[32] M. N. Zafar and J. Mohanta, "Methodology for path planning and optimization of mobile robots: A review," *Procedia computer science*, vol. 133, pp. 141–152, 2018.

[33] J. Song and S. Gupta, "e: An online coverage path planning algorithm," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 526–533, 2018.

[34] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation," *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 801–812, 2011.

[35] B. Park, J. Choi, and W. K. Chung, "Incremental hierarchical roadmap construction for efficient path planning," *ETRI Journal*, vol. 40, no. 4, pp. 458–470, 2018.

[36] Y. Bouzid, Y. Bestaoui, and H. Siguerdidjane, "Quadrotor-uav optimal coverage path planning in cluttered environment with a limited onboard energy," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 979–984.

[37] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 718–724, 2004.

[38] R.-E. Precup and R.-C. David, *Nature-inspired optimization algorithms for fuzzy controlled servo systems*. Butterworth-Heinemann, 2019.

[39] C. Purcaru, R.-E. Precup, D. Iercan, L.-O. Fedorovici, R.-C. David, and F. Dragan, "Optimal robot path planning using gravitational search algorithm," *International Journal of Artificial Intelligence*, vol. 10, no. S13, pp. 1–20, 2013.

[40] J. Xie, L. R. G. Carrillo, and L. Jin, "Path planning for uav to cover multiple separated convex polygonal regions," *IEEE Access*, vol. 8, pp. 51 770–51 785, 2020.

[41] G. Han, Z. Zhou, T. Zhang, H. Wang, L. Liu, Y. Peng, and M. Guizani, "Ant-colony-based complete-coverage path-planning algorithm for underwater gliders in ocean areas with thermoclines," *IEEE Transactions on Vehicular Technology*, 2020.

[42] A. Soares, R. Râbelo, and A. Delbem, "Optimization based on phylogram analysis," *Expert Systems with Applications*, vol. 78, pp. 32–50, 2017.

[43] G. Rigatos, P. Siano, D. Selisteanu, and R. Precup, "Nonlinear optimal control of oxygen and carbon dioxide levels in blood," *Intelligent Industrial Systems*, vol. 3, no. 2, pp. 61–75, 2017.

[44] Y. Nobert and J.-C. Picard, "An optimal algorithm for the mixed chinese postman problem," *Networks: An International Journal*, vol. 27, no. 2, pp. 95–108, 1996.

[45] J. He, Y. Geng, F. Liu, and C. Xu, "Cc-kf: Enhanced toa performance in multipath and nlos indoor extreme environment," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 3766–3774, 2014.

[46] C. Bliek1ú, P. Bonami, and A. Lodi, "Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report," in *Proceedings of the twenty-sixth RAMP symposium*, 2014, pp. 16–17.

[47] Z.-W. Lin, S.-Y. Fang, Y.-W. Chang, W.-C. Rao, and C.-H. Kuan, "Provably good max–min-m-neighbor-tsp-based subfield scheduling for electron-beam photomask fabrication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 2, pp. 378–391, 2017.

[48] R. Properties, "Create rectangle with sharp or curved corners - matlab rectangle- mathworks united kingdom," 2020. [Online]. Available: https://uk.mathworks.com/help/matlab/ref/rectangle.html

[49] C. Moon, J. Kim, G. Choi, and Y. Seo, "An efficient genetic algorithm for the traveling salesman problem with precedence constraints," *European Journal of Operational Research*, vol. 140, no. 3, pp. 606–617, 2002.

[50] J. Kirk, "Traveling salesman problem-genetic algorithm," *Retrieved from the MATLAB File Exchange website: www. mathworks. com/matlabcentral/fileexchange/13680-travelingsalesman-problem-genetic-algorithm*, 2007.

[51] O. Sundstrom and L. Guzzella, "A generic dynamic programming matlab function," in *2009 IEEE control applications,(CCA) & intelligent control,(ISIC)*. IEEE, 2009, pp. 1625–1630.

ZEBA KHANAM is a Research Officer in the Embedded and Intelligent Systems Laboratory at the University of Essex where she is currently also pursuing a PhD degree. She has received her B.Tech degree in Computer Engineering from Aligarh Muslim University, Aligarh, India in 2013, were she was awarded a gold medal for securing first rank, and a M.Tech degree in Advanced Electronic Systems from Academy of Scientific and Innovative Research, India in 2015. Zeba has worked as a Scientist in Machine Vision Laboratory, CSIR-CEERI, Pilani, India which allowed her to gain industrial and academic experience in computer vision and embedded systems application in Minimally Invasive Surgery and Endoscopy. Her current research focuses on robot vision and path planning in nuclear environment.

SANGEET SAHA received the B.Tech. degree in information technology, the M.Tech. degree in computer science and engineering, and the Ph.D. degree from the University of Calcutta, Kolkata, in 2011, 2013, and 2018, respectively. In 2018, he was a Research Fellow with Tata Consultancy Services (TCS). He has been appointed as a Senior Research Officer with the Embedded and Intelligent Systems (EIS) Research Group, University of Essex, Colchester, Since May 2018. His current research interests include real-time scheduling, scheduling for reconfigurable computers, real-time and fault-tolerant embedded systems, and cloud computing.

SHOAIB EHSAN (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2003, and the Ph.D. degree in computing and electronic systems (with specialization in computer vision) from the University of Essex, Colchester, U.K., in 2012. He has an extensive industrial and academic experience in the areas of embedded systems, embedded software design, computer vision, and image processing. His current research interests are in intrusion detection for embedded systems, local feature detection and description techniques, and image feature matching and performance analysis of vision systems. He was a recipient of the University of Essex Post Graduate Research Scholarship, the Overseas Research Student Scholarship, and the prestigious Sullivan Doctoral Thesis Prize awarded annually by the British Machine Vision Association.

RUSTAM STOLKIN (Member, IEEE) received the M.Eng. degree in engineering science from the University of Oxford, Oxford, U.K., in 1998, and the Ph.D. degree in computer vision from University College London, London, U.K., in 2004. He is currently the Director and the Royal Society Industry Fellow of the National Centre for Nuclear Robotics, Birmingham, U.K., and a Professor of robotics with the University of Birmingham, Birmingham, where he is the Founder and the Director of the Extreme Robotics Lab. He is also the Director of Spinout Company A.R.M Robotics Ltd., Barakaldo, Spain. He is highly interdisciplinary, with research interests spanning computer vision and image processing, machine learning and AI, robotic grasping and manipulation, and human–robot interaction.

KLAUS MCDONALD-MAIER (Senior Member, IEEE) is currently the Head of the Embedded and Intelligent Systems Laboratory, University of Essex, Colchester, U.K. He is also the Chief Scientist with UltraSoC Technologies Ltd., the CEO of Metrarc Ltd., and a Visiting Professor with the University of Kent. His current research interests include embedded systems and system-on-chip design, security, development support and technology, parallel and energy-efficient architectures, computer vision, data analytics, and the application of soft computing and image processing techniques for real-world problems. He is a member of VDE and a Fellow of the IET.