

## RESEARCH ARTICLE

WILEY

# 3D hybrid path planning for optimized coverage of agricultural fields: A novel approach for wheeled robots

Danial Pour Arab<sup>1,2</sup>  | Matthias Spisser<sup>2</sup> | Caroline Essert<sup>1</sup>

<sup>1</sup>ICube, Université de Strasbourg, Illkirch, France

<sup>2</sup>T&S - Technology and Strategy Strasbourg, Schiltigheim, France

## Correspondence

Danial Pour Arab, ICube, Université de Strasbourg, CNRS (UMR 7357), 300 Bd Sébastien Brant, 67400 Illkirch, France.  
Email: [dpourarab@unistra.fr](mailto:dpourarab@unistra.fr) and [danialpourarab@gmail.com](mailto:danialpourarab@gmail.com)

## Funding information

Technology and Strategy Group (T&S), Grant/Award Number: CIFRE 2019/1084; French National Association for Research and Technology

## Abstract

Over the last few decades, the agricultural industry has made significant advances in autonomous systems, such as wheeled robots, with the primary objective of improving efficiency while reducing the impact on the environment. In this context, determining a path for the robot that optimizes coverage while taking into account topography, robot characteristics, and operational requirements, is critical. In this paper, we present H-CCPP, a novel hybrid method that combines the comprehensive coverage benefits of our previous approach O-CCPP with the computational efficiency of the Fields2Cover algorithm. Besides optimizing coverage area, overlaps, and overall travel time, it significantly improves the computation process, and enhances the flexibility of trajectory generation. H-CCPP also considers terrain inclination to address soil erosion and energy consumption. In an effort to support this innovative approach, we have also created and made available a public data set that includes both 2D and 3D representations of 30 agricultural fields. This resource not only allows us to illustrate the effectiveness of our approach but also provides invaluable data for future research in complete coverage path planning (CCPP) for modern agriculture.

## KEYWORDS

3D field, autonomous agriculture, complete coverage path planning, path planning, precision agriculture, vehicle routing problem, wheeled robots

## 1 | INTRODUCTION

Robotics is permeating everyday life, offering greater efficiency and precision in many sectors, including agriculture. With the growing global population necessitating increased food and agricultural production (Marques et al., 2019), addressing the environmental challenges linked to agriculture, such as air pollution, mortality rates (Giannakis et al., 2019), and the negative effects of pesticides (Meftaul et al., 2020), becomes critical. Consequently, the need for

precise autonomous systems to optimize cost-efficient field operations has never been more apparent.

Developing an autonomous system in agriculture demands the integration of multiple specialized and interconnected subsystems, including path planning, path following, positioning, and perception systems. This article focuses on path planning; for other subsystems, readers can refer to Tang et al. (2023) and Vieira et al. (2021, 2022). Path planning can be classified into two categories: Point-to-Point and Complete Coverage. Point-to-point path planning focuses on

Danial Pour Arab contributed equally to this article.

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2024 The Author(s). *Journal of Field Robotics* published by Wiley Periodicals LLC.

determining the most effective path from a designated start to a specific endpoint. In contrast, *complete coverage path planning* (CCPP) ensures every point within a defined workspace is covered efficiently. The significance of CCPP extends to a variety of applications, such as demining (Đakulovic & Petrovic, 2012), cleaning (Viet et al., 2013), and mapping and monitoring (Becker et al., 2022), highlighting its versatility across numerous sectors.

In this paper, we focus on CCPP for autonomous agricultural robots, and on the constraints specific to agriculture. Especially, agricultural CCPP involves the creation of a path that effectively covers an entire field, one of the main considerations being the need to avoid repeated passes over the same location. For instance, a second pass over an area already sown would not only destroy the initial seeds, but also compact the soil, increase seed costs, and ultimately diminish the overall field productivity. Let us note however that, while CCPP is applicable in both arable and orchard farming, each presents unique challenges and requirements. Arable farming, involving the cultivation of crops on large fields, requires generating a path that covers the entire area for tasks, like, seeding, fertilizing, or harvesting. On the other hand, orchard farming includes the cultivation of fruit trees and nuts, where the path planning mainly revolves around navigating through predefined rows between trees to perform tasks, such as pruning and spraying. Our study is primarily focused on CCPP for arable farming. This focus is due to the distinct challenge it presents in generating paths for vast open spaces without predefined trajectories, necessitating innovative CCPP strategies to enhance productivity.

Constraints and rules specific to agriculture are diverse. Generally, the path of an agricultural wheeled robot is composed of a series of back-and-forth trajectories, with continuous half-turns at areas along the field edges, known as *headlands*, determined by the implement's width. Their spacing is equal to the width of the implement typically referred to as the *working width*. Planning such a path involves taking into account the topographical and geometrical features of the field, that sometimes have complex shapes, the capabilities of the robot, and the characteristics of its implements, to optimize productivity, cost, and environmental impact. In addition, operational constraints vary based on the machinery. For instance, in seeding, the implement must be lowered to the ground during activation, and it can stay activated only during straight or slightly curved paths. During half-turns, it must be deactivated and lifted to avoid damage. This is not a concern in pulverization where the implement does not come into contact with the ground. Cross-constraints may occur, where the efficiency of certain operations are linked to others. For example, the pulverization path depends on the tramline farming system established during tillage, which creates parallel tracks in the field to prevent soil compaction and damage to crops (Bochtis et al., 2010). Finally, the three-dimensional (3D) aspect of a field, in particular its inclinations and slope directions, has a significant impact on soil erosion and energy consumption. Soil erosion has long been and continues to be a major concern. As highlighted by Dogru and Marques (2015) and Van Doren et al. (1951), fields planted perpendicular to the slope reduce soil erosion and surface runoff, and

climbing slopes can drastically increase energy consumption. Therefore, ignoring these topographic variations can lead to imprecise energy cost estimates and ineffective optimization on nonflat fields. Although this problem has been well documented, it is still rarely taken into account in trajectory planning algorithms.

In this study, we introduce a new hybrid approach for CCPP, with the aim of addressing a maximum of agricultural constraints with the best efficiency. This method draws from the strengths of our previous work (Pour Arab et al., 2023a) that effectively covers headlands, handles intricate field geometries, considers field entry points, and accounts for the distance required for lowering and raising the implement into the ground. It also incorporates the benefits of the Fields2Cover (F2C) method (Mier et al., 2023) that efficiently generates parallel trajectories for a variety of driving directions. This innovative approach surpasses the limitations of both methods and additionally takes into account trajectory inclinations, which indirectly impact soil erosion and the energy consumption of the robot. In addition, we introduce a data set consisting of 30 diverse agricultural fields in France, varying from 1.83 to 13.21 ha in size. The data set, made available to the community on Zenodo (Pour Arab & Essert, 2024), contains the necessary information for each field and covers a broad spectrum from simple to complex field shapes.

The structure of this paper is as follows: Section 2 provides a survey of related works. Section 3 outlines the objectives and key contributions of this paper. Our innovative methodology is detailed in Section 4. The results and comparative discussion of our new approach versus our previous one, as applied to our data set, are presented in Section 5. Lastly, Section 7 concludes the study and proposes directions for future research.

## 2 | RELATED WORKS

The field of agricultural robotics has seen significant advancements, with a broad spectrum of robotic systems being developed to address the multifaceted challenges of modern agriculture. These systems range from ground-based wheeled and tracked robots to aerial drones and sophisticated robotic arms, each designed to perform specific tasks, such as monitoring crop health (Chin et al., 2023), precision spraying (Mahajan et al., 2023), and automated harvesting (Mail et al., 2023).

This paper, however, concentrates on CCPP for wheeled robots. Wheeled robots have been the subject of extensive research, focusing on optimizing their path planning to ensure thorough coverage and efficient task execution. Notable studies that have explored different dimensions of wheeled robotic solutions include Fasiolo et al. (2023), which shed light on applications of autonomous ground mobile robots in agriculture, highlighting advancements in hardware, localization, navigation, path-planning strategies, and AI integration.

Existing literature often breaks down the complex CCPP problem into two tasks. First, *Coverage Path Planning* (CPP) generates parallel trajectories. Second, the *Vehicle Routing Problem* (VRP) or *Agricultural Vehicle Routing Problem* (AVRP) focuses on optimally sequencing

these trajectories using half-turns and assigning them to one or multiple robots (Pour Arab et al., 2023a).

In this context, CPP primarily applies to arable farming operations, while AVRPs approaches could be adapted for both arable farming and orchard farming. In arable farming, parallel tracks could be generated using CPP methods and interconnected through AVRPs methods. Conversely, in orchard farming, paths are mainly predefined due to the structured rows of trees or vines, making AVRPs methods alone potentially adequate for determining the traversal order. However, substantial efforts are directed towards orchard-specific solutions. Our focus, however, remains on CCPP for arable farming, directing readers interested in orchard farming methodologies to Zhang et al. (2024) for path planning for clustered lawnmower robots in orchards with an improved A\* algorithm, to Xu and Rai (2024) for autonomous navigation and motion planning in peach orchards using vision-based technologies, to Wang et al. (2022) for autonomous navigation spray systems utilizing 3D laser light detection and ranging (LiDAR) and millimeter wave radar, to Wang et al. (2023) for deep reinforcement learning-based path planning in kiwifruit orchards, and to Santos et al. (2022) for path planning with hybrid maps in vineyards.

Within the context of arable farming, common CPP strategies include creating back-and-forth trajectories parallel to the field's longest edge or accepting a specific driving direction to generate them (Jeon et al., 2021; Nilsson & Zhou, 2020; Nørremark et al., 2022). An enhancement to this approach was to introduce the concept of generating parallel trajectories along a curved reference line (Jensen et al., 2012; Zhou et al., 2020).

The assumption that reducing half-turns could lower energy consumption and soil compaction, has led to various strategies. Edwards et al. (2017) proposed aligning trajectories with the boundary requiring the fewest half-turns. Cao et al. (2019) applied the rotating calipers algorithm. Furthermore, Mier et al. (2023) employed a brute force algorithm to determine the optimal driving direction using a discretized angles testing method.

For concave fields, advanced strategies are proposed. Jin and Tang (2010) introduced subdivision of the field into smaller convex subfields, generating trajectories parallel to a given reference line for each. An optimal sequence of subfields was then determined. Oksanen and Visala (2009) used trapezoidal field decomposition and a heuristic to select the optimal driving direction from defined angles (0°, 30°, 60°, 90°, 120°, and 150°), optimizing metrics, including the operated area per total time, subfield area per remaining area, and operated distance in subfields.

A few researchers have gone beyond the 2D field structure, incorporating 3D data of the field into their CPP algorithms. Hameed (2014) applied a genetic algorithm to identify an optimal reference line that minimizes tracks number and turning cost, considering trajectory inclinations to optimize fuel consumption. Dogru and Marques (2015) and Shen et al. (2020) applied a field decomposition considering inclinations, followed by a genetic algorithm to determine the optimal driving direction for each subfield, and an optimal subfield sequence to minimize energy consumption. Hameed et al.

(2016) also factored in the inclination to find an optimal driving direction that minimizes skips/overlaps. Jin and Tang (2011) suggested a decomposition method that separates the field into flat and slope areas, selecting a reference direction that minimizes coverage cost (weighted average of headland turning cost, soil erosion cost, and curved trajectory cost) from field edge segments and slope contours.

Significant research efforts have been dedicated to exploring the AVRPs using single or multiple robots. In these scenarios, the parallel lines are either provided as an input or generated through a proposed CPP method. The focus of several of these studies rests on a single-robot system. Bochtis et al. (2013) proposed a Clarke–Wright algorithm-based solution. Mier et al. (2023) and Plessen (2019) developed a pattern-based routing algorithm. When considering a single stationary service unit, Jensen et al. (2015) suggested an approach rooted in the state–space search technique. Nilsson and Zhou (2020) presented a strategy built on the artificial bee colony algorithm. On the other hand, Vahdanjoo et al. (2020) proposed a simulated annealing-based approach, adapted multiple stationary service units. Evans et al. (2020) incorporated genetic algorithms into his methodology, considering a mobile service unit.

Transitioning to multirobot AVRPs, various strategies have been proposed. Burger et al. (2013) presented an approach anchored in mixed-integer linear programming. Seyyedhasani et al. (2019) developed a heuristic approach utilizing the Clarke–Wright algorithm in conjunction with the Tabu search. Notably, Cariou et al. (2020) presented a methodology to address AVRPs for a homogeneous robot convoy. Techniques using simulated annealing have been proposed by Conesa-Muñoz et al. (2016). Utamima et al. (2019) presented an evolutionary approach enhanced through a neighborhood search, while Khajepour et al. (2020) employed an adaptive large neighborhood search in their methodology. The presence of a single stationary service unit was considered in the approaches presented by Conesa-Muñoz et al. (2016), Khajepour et al. (2020), and Utamima et al. (2019). Jensen et al. (2012) offered an approach based on the Dijkstra algorithm, considering a mobile service unit cooperating with a primary service unit.

Furthermore, Bochtis and Sørensen (2010) proposed a solution based on a breadth-first search algorithm modified by additional heuristics, factoring in either one or two stationary service units or a mobile service unit. Nørremark et al. (2022) expanded the approach proposed by Nilsson and Zhou (2020) to incorporate a mobile service unit for on-the-go unloading in both the headland and main field, to establish unloading timings irrespective of the harvester's full bin level, and to consider the transport unit's operational time outside the field. Their main objective was to minimize time and distance costs for all vehicles engaged in harvest operations.

Contrary to the common approach of addressing CCP and AVRPs as distinct issues, our prior work introduced a unified one-step CCPP algorithm (Pour Arab et al., 2023a). This algorithm, founded on the principles of tree construction and exploration from each specified entry point, allowed for comprehensive coverage of both convex and concave fields, including the headlands. As a result, it not only

ensured full coverage of the field and headlands but also identified suitable exits along accessible field edges. Building on this, an extension of the approach was developed to generate solutions with a row-skip pattern (Pour Arab et al., 2023b). However, this extended algorithm is currently best suited to simple fields that do not require a field decomposition.

Despite the considerable research efforts, coverage of headlands has largely been overlooked in most studies. A few notable exceptions include the methodologies proposed by Edwards et al. (2017), Jeon et al. (2021), Nilsson and Zhou (2020), Nørremark et al. (2022), and Pour Arab et al. (2023a, 2023b), which have all incorporated automatic headland coverage into their strategies. Edwards et al. (2017) and Jeon et al. (2021) additionally contemplated reverse moves for trajectories within the headlands, thereby ensuring full coverage of all field corners. However, when considering complex field scenarios that require both headland coverage and field decomposition, only the approach introduced in our previous work (Pour Arab et al., 2023a) proves efficient and effective.

### 3 | MOTIVATIONS AND CONTRIBUTIONS

In our previous work (Pour Arab et al., 2023a), we developed a tree-based CCPP method for an autonomous agricultural robot. Our original approach *O*-CCPP allows one to partition complex field shapes into subfields. It incrementally constructs paths of back-and-forth trajectories for each subfield with multiple turn options at each end accounting for implement transitions, connects the paths with the adjacent subfields, and adds a final route to optimally cover the headlands and reach a possible exit. The algorithm stores in a solution space all feasible and effective paths from an entry to an exit, then evaluates each solution based on several metrics. The solutions are clustered according to their major driving direction, and the best solution within each cluster is identified using a multiobjective optimization function and presented to the user, allowing to provide alternatives with varied patterns. This method has shown a good versatility and ability to plan optimal trajectories for various field geometries, but has some limitations. For instance, it can only generate solutions aligned with one of the field edges, create row-skip patterns only for simpler fields, and it is relatively slow, with an average computation time of 624.6s in our previous study.

In contrast, the *F2C* algorithm described in Mier et al. (2023) is more cost-effective, with a reported average computation time of 1.52s. *F2C* is an open-source library also developed for agricultural CCPP. Its primary functions include the generation of headlands to host turns, the creation of parallel tracks in the central area in varying driving directions from 0° to 180°, and the systematic ordering of these tracks. The tracks are optimally connected with half-turns in the headlands, according to a single objective optimization function. Despite its undeniable qualities, *F2C* also has certain limitations. It does not generate headland coverage or routes from entry or to exit points, cannot partition complex field shapes to cover them in different directions, does not account for implement transitions, and optimizes only one single objective at a time.

To benefit from the best features of both methods, we introduce in this article a novel hybrid approach *H*-CCPP that merges the robust capabilities and advanced features of *O*-CCPP with the efficiency and flexibility of the *F2C* method. In the hybrid approach, the overall architecture of *O*-CCPP, that is, field decomposition—generation of solutions (back-and-forth paths, headlands, and routes to exit)—selection of optimal solutions is preserved, as well as the multiple objective optimizations. The most time-consuming part of the *O*-CCPP algorithm, the incremental generation of the back-and-forth trajectories with multiple turn options, is replaced with the more efficient creation of parallel tracks at various orientations of *F2C*, to create a smaller but more flexible solution space with a wide range of driving directions. Compared to *F2C*, *H*-CCPP adds the field decomposition, accounts for multiple turn types with implement transitions, and adds headland coverage and routes to exits. In addition, *H*-CCPP incorporates a new metric to account for field inclination, a pivotal factor influencing soil erosion and the robot's energy consumption.

Incorporating inclination data necessitates a more thorough representation of the agricultural field. Despite a few studies considering 3D field data and inclines, there is a noticeable gap in the comprehensive data set containing both 2D and 3D surfaces of fields. To address this, we created a unique data set of 30 diverse agricultural fields, handpicked from various regions in France, aiming to bridge this knowledge gap.

The next sections of this article detail the proposed approach, the data set built for experimental validation, and discuss the experimental results.

### 4 | METHODOLOGY

This section presents the new hybrid solution designed to tackle the complexities of CCPP. Our proposed method has the ability to examine driving directions from 0° to 180°, with increments set by a fixed step size. For each direction, it generates parallel trajectories within the field, identifies an optimal entry point, ensures full headland coverage, and initiates a search for an exit path towards the nearest accessible field edge. To boost coverage efficiency, it evaluates two different coverage patterns, sequential and row-skip, for each driving direction assessed.

When a dividing line is provided, it decomposes the field polygon into distinct subfields. The method evaluates every combination of driving directions and coverage patterns for these subfields. It begins with each accessible subfield, and after fully covering the initial subfield, it seeks out the closest uncovered subfield, repeating the process. Once full coverage of all subfields is confirmed, it initiates a search for the closest potential exit point.

Once a collection of viable solutions is produced, an optimal solution is determined using a selection process. This solution is chosen to maximize the coverage rate while minimizing overlap, nonworking travel distance, and operation time. In addition, considering trajectory inclinations indirectly optimizes soil erosion and energy consumption.

For the successful application of this method, an initial step of gathering accurate field data and building both 2D and 3D field models is required. Hence, in this section, we further discuss the data collection process and the creation of a detailed data set of agricultural fields. Following this, we proceed to elaborate on the intricacies of our proposed method.

#### 4.1 | Terrain modeling: Data acquisition and data set construction

In this section, we present a comprehensive overview of the data needed to generate 2D and 3D models of fields. We also provide a data set of 30 fields in France, enabling the evaluation of various applications, like, path planning, soil erosion analysis, and energy consumption estimation.

##### 4.1.1 | Data acquisition

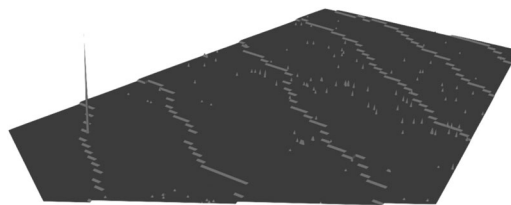
The necessary data for constructing both 3D and 2D field models in this study involves several elements: counterclockwise points defining the field's border, one or more pairs of points signifying access segments, field elevation data, and optionally, sets of clockwise points indicating static obstacles.

For acquiring access segments, field borders, and obstacle borders, we utilized the Géoportail (2023) annotation tool. This tool facilitates drawing various geometric shapes directly on satellite images. The annotated shapes can be exported in Keyhole Markup Language, an Extensible Markup Language (XML)-based format for geographic annotations.

Elevation data were obtained using IGN's elevation calculation services (IGN, 2023), implemented as a representational state transfer application programming interface (REST API). In addition to these services, IGN also provides *Digital Elevation Models (DEM)* of the entire French territory with various resolutions. However, downloading the full DEM data set might demand substantial disk space.

An elevation grid within the field's bounding box was created to acquire field elevation data, using the REST API to determine each point's elevation. Notably, the method's execution time increases with the number of grid points. For instance, an elevation grid with a spacing of  $\ell_e = 0.5\text{m}$  takes over 13 h to determine the elevation of 136, 859 points for a 3.34-ha field. Additionally, high-resolution elevation grids might introduce noise into the data, as illustrated in Figure 1.

To overcome the challenges, we used a lower-resolution elevation grid with a point spacing of  $\ell_e = 5\text{m}$ . Each grid point's altitude was acquired using the REST API, and a normalization process was applied to set the minimum altitude to zero while preserving the original elevation variance. This normalized, lower-resolution elevation grid can serve as the basis for constructing a high-resolution 3D surface of the field using elevation interpolation techniques, which we detail in Section 4.1.2.



**FIGURE 1** High-definition elevation data sourced via the REST API from IGN (2023). REST API, representational state transfer application programming interface.

##### 4.1.2 | Terrain modeling

This section describes the process of creating a high-resolution 2D field surface, starting with input data conversion from geographic to Cartesian coordinates. It then proceeds with the generation of a 3D field surface, integrating the 2D surface, elevation grid, and a popular interpolation technique.

**2D surface construction:** For the 2D surface, we first create a 2D polygon of the field, excluding obstacles. We then generate a grid within the field's bounding box with a  $\ell_g$  spacing, retaining only the points within the 2D polygon. Following this, points are interpolated on the field and obstacle borders with a maximum  $\ell_g$  spacing. Finally, the remaining points are triangulated to generate the 2D surface. This process is illustrated in Figure 2.

**3D surface construction:** A high-resolution 3D surface is generated from a 2D field surface using a process that relies on the *Inverse Distance Weighting (IDW)* interpolation technique, originally proposed by Shepard (1968). In this process, the elevation of a point  $P$  of the 2D surface is calculated according to its  $N$  nearest neighbor points on the elevation grid ( $\{P_i | P_i \in \mathbb{R}^2, i \in \mathbb{N}, 1 \leq i \leq N\}$ ) as follows:

$$f(P) = \begin{cases} \frac{\sum_{i=1}^N w_i(P_i) f(P_i)}{\sum_{i=1}^N w_i(P_i)} & \text{if } d(P, P_i) \neq 0 \text{ for all } i, \\ f(P_i) & \text{if } d(P, P_i) = 0 \text{ for some } i, \end{cases} \quad (1)$$

$$w(P_i) = \frac{1}{d(P, P_i)^p}, \quad (2)$$

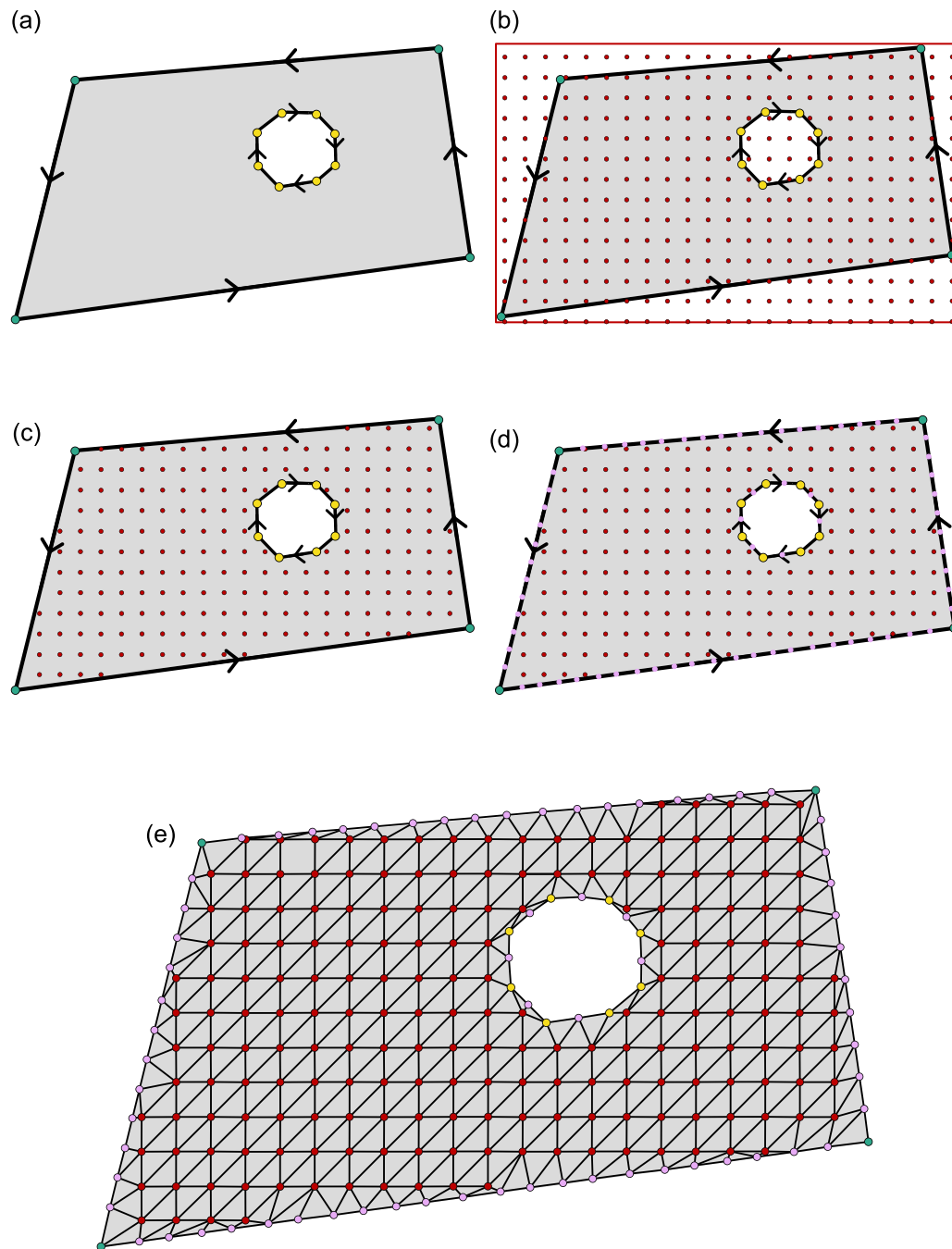
where  $d$  represents the Euclidean distance, while  $p$  is a positive real number referred to as the power parameter.

The absence of a ground truth complicates the accuracy assessment of our terrain modeling approach. A solution could be to use LiDAR-equipped drones to create a precise ground truth data set. This would allow a thorough evaluation of IDW and other interpolation methods for 3D field reconstruction.

##### 4.1.3 | A data set of real fields

We created a data set comprising 30 agricultural fields situated in France. Chosen manually, these fields have a wide variety of





**FIGURE 2** Sequential phases of 2D field modeling for a convex field featuring one obstacle. (a) 2D field layout with an obstacle, (b) grid overlay within the field's bounding box, (c) keeping only grid points within the field, (d) interpolating points on the borders, and (e) remaining points are triangulated to generate the 2D surface.

shapes and sizes, ranging from 1.83 to 13.21 ha. This includes simple geometric configurations not necessitating field decomposition, as well as more complex shapes that require it. The data set is designed to provide an extensive set of real-world fields, ideal for evaluating and validating path-planning approaches. This data set is publicly available on Zenodo (Pour Arab & Essert, 2024).

The data set compiles the following data for each field into distinct files:

- an overhead image (PNG),
- a 2D boundary representation (XML),
- a 2D mesh surface (Polygonal File Format [PLY]) with  $\ell_g = 0.25$  m,

- an elevation grid (PLY) with  $\ell_e = 5$  m,
- a 3D mesh surface (PLY) with IDW parameters  $N = 20$  and  $p = 2$ ,
- a collection of 2D line segments depicting access segments (XML).

The PLY file format (Turk, 1994) is used for both 2D and 3D data storage, with 2D surfaces with zero  $z$  value.

## 4.2 | Hybrid CCPP approach

Our method requires field data (2D polygon, 3D surface, and access segments) and robot data. This includes minimum turning radius when the robot's implement is engaged, that is, lowered into the soil, ( $\gamma_{on}$ ) or disengaged, that is, lifted from the ground, ( $\gamma_{off}$ ), the distance needed for the proper lowering or raising of the implement ( $\ell_t$ ), offset between robot and implement ( $\ell_o$ ), working width ( $w$ ), robot speed when the implement is off ( $V_{off}$ ), on ( $V_{on}$ ), in transitioning ( $V_{gap}$ ), and the minimum travel distance before disengaging the implement ( $\Delta_{mwd}$ ).

Following the acquisition of necessary field and robot data, our proposed method progresses through three key phases: preprocessing, coverage path generation, and selection of the optimal solution.

### 4.2.1 | Preprocessing

The preprocessing phase aligns closely with our original approach (Pour Arab et al., 2023a), distinguished by a single variation: the quantity of gap-covering trajectories. In this version, this number is user-determined rather than being fixed at one.

Preprocessing encompasses the creation of headlands, inclusive of  $p$  inner trajectories and  $g$  gap-covering trajectories, along with generating turning spaces based on the working width and robot-specific parameters, like,  $\gamma_{off}$  and  $\ell_o$ . Moreover, given a dividing line, the field undergoes subdivision into distinct subfields. For an in-depth understanding of this process, the readers are referred to Pour Arab et al. (2023a).

In the original approach, entry points selection occurred during preprocessing, necessitating manual decision-making by an operator to determine which points to retain or discard. In this novel approach, rather than predefining entry points during preprocessing, a specific entry point is identified for each driving direction after generating parallel tracks and ordering them.

### 4.2.2 | Coverage path generation

Upon completion of the preprocessing phase, we proceed with a comprehensive process to create and store an array of potential solutions. This process is outlined in the following algorithm:

#### Algorithm 1. Coverage path generation algorithm

```

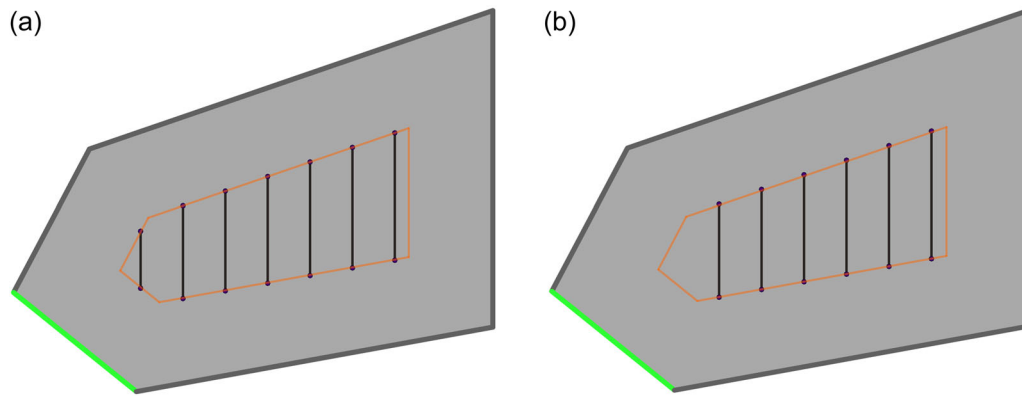
for each subfield do
  for each driving direction from 0° to 180° do
    for each subfield do
      if subfield has access segment then
        while there are uncovered subfields do
          Generate sequential coverage path using Dubins (1957) method
          Generate sequential coverage path using Reeds and Shepp (1990) method
          Generate row-skip coverage path using Dubins (1957) method
          Generate row-skip coverage path using Reeds and Shepp (1990) method
        for each coverage path do
          Find nearest uncovered subfield
        end for
      end while
      Find the shortest exit path
      Add all solutions to the solution space
    end if
  end for
end for

```

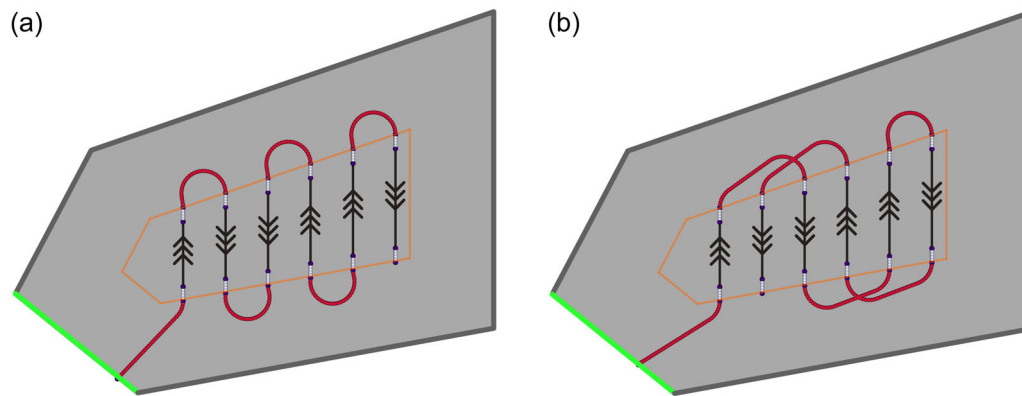
To generate coverage paths, for driving directions from 0° to 180° with a constant step size denoted as  $\ell_s$ , first parallel tracks are generated in the main part of the field using F2C. Then tracks that do not satisfy the minimum working distance are eliminated. Figure 3 provides a visual representation of this procedure for a driving direction of  $\frac{\pi}{2}$ .

Subsequently, two ordering possibilities, sequential and row-skip, are considered. The connection between these ordered tracks mainly involves half-turns that can be generated using either Dubins method (Dubins, 1957) or Reeds–Shepp method (Reeds & Shepp, 1990). For simplicity, we describe our approach employing the Dubins method for turn generations. However the same process is also applicable for generating solution with turns generated via Reeds–Shepp method. For each of these scenarios, the nearest entry on an access segment is identified to enter the field and begin traveling along the ordered parallel tracks. To do so, the access segment is discretized into a set of points, spaced at intervals of  $\ell_a$ . Each of these points is then associated a direction perpendicular to the access segment, oriented towards the field's interior. Afterwards, a Dubins trajectory is formulated to reach ordered tracks. Figure 4 illustrates the path starting from the access segment and covering all parallel tracks for both patterns. In addition, transition trajectories are appended to both ends of each track for lowering and raising the implement.

Once all parallel tracks are incorporated into the path, the coverage of the headlands begins. This process starts by identifying the nearest headlands to the path's endpoint, considering only endpoints of each headland's first-level inner trajectory, that is, those nearest to the field edge. The shortest Dubins trajectory then links this to the path's endpoint. After covering all first-level inner trajectories, the method continues to cover the second-level and all remaining inner and gap-covering trajectories.



**FIGURE 3** Generation of parallel tracks aligned with a driving direction of  $\frac{\pi}{2}$ . The headland borders are shown in orange, and access segments are illustrated in green. (a) Generated parallel tracks and (b) following the removal of short tracks.



**FIGURE 4** Depiction of a path initiating at an access segment, covering all tracks. The headland borders are shown in orange, and access segments are illustrated in green. Turns are highlighted in red, while transition trajectories are marked in white. (a) Sequential pattern and (b) row-skip pattern.

This circular coverage pattern used for headlands can adopt either a clockwise or counterclockwise orientation, depending on which endpoint of a first-level inner trajectory lies nearest to the termination of the last track. The main motivation behind employing this circular pattern for covering headlands serves to minimize computational time while facilitating further enhancement of the method allowing complete coverage of all corners of the field.

The process of covering the headlands, applicable to both the sequential and row-skip patterns, is illustrated in Figure 5, with the gap-covering trajectories being excluded for clarity. Figure 5a,c depicts the path extending from the access segment up to the end of the last covered inner trajectory. In contrast, Figure 5b,d exclusively illustrates part of the path, from the end of the last track up to the final point of the last traversed inner trajectory. It is important to emphasize that, in specific cases like the one shown in the figure, some inner and gap-covering trajectories might remain uncovered due to the constraints imposed by the minimum working distance.

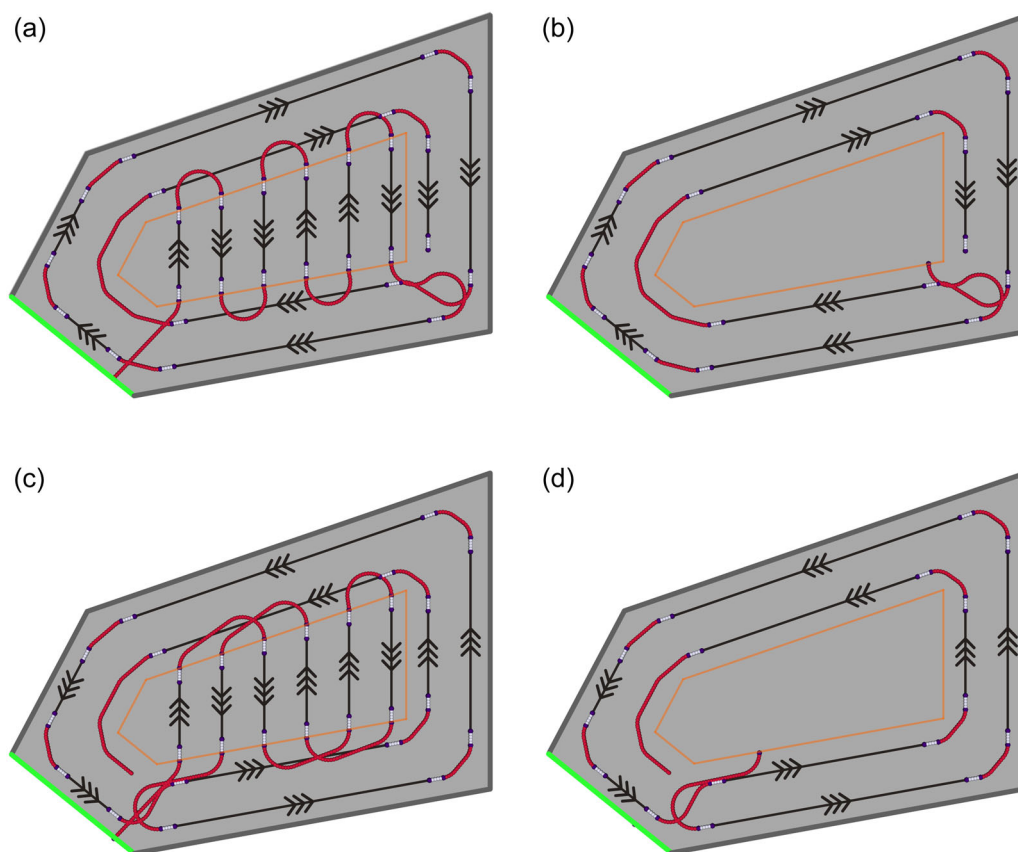
After completing headland coverage, the method identifies the nearest access segment to exit the field using a *circular search*. This

search consists of navigating both clockwise and counterclockwise through the first-level inner trajectories of the headlands to locate the nearest access segment. It results in two potential exit routes, with the shorter one being selected as the preferred exit path. The results of this exit search for both sequential and row-skip patterns are shown in Figure 6.

As illustrated in Figure 6b, if the final inner trajectory allows continuation in the same direction for exiting, a transition is initiated just before the access segment to deactivate the implement, enabling a safe exit without a directional change. If this is not viable, the access segment is discretized into points spaced by  $\ell_a$ . Each point is linked with a direction oriented towards the field's interior, and a Dubins trajectory is computed from the path's end to the closest point, finalizing the exit path, as depicted in Figure 6d.

This exiting process may occasionally lead to overlaps in the headlands, especially when an access segment is nearby. In certain situations, it might be necessary to traverse the first-level trajectory of some headlands more than once to reach the nearest segment. For instance, Figure 6b demonstrates a scenario where the





**FIGURE 5** Illustration of the headland coverage process for both sequential and row-skip patterns. Headland borders are in orange, access segments in green, turns in red, and transition trajectories in white. (a) Postheadland coverage (sequential), (b) headland path (sequential), (c) postheadland coverage (row-skip), and (d) headland path (row-skip).

first-level trajectory of the lower headland is re-traversed to reach an access segment situated on the right side of the field.

Upon completion of these processes, for each driving direction, four types of coverage paths are formulated: two using a sequential pattern and two employing a row-skip pattern. Specifically, within these patterns, turns are generated using either the Dubins (1957) method or the Reeds and Shepp (1990) method. These paths are subsequently added into the solution space for further evaluation and comparison.

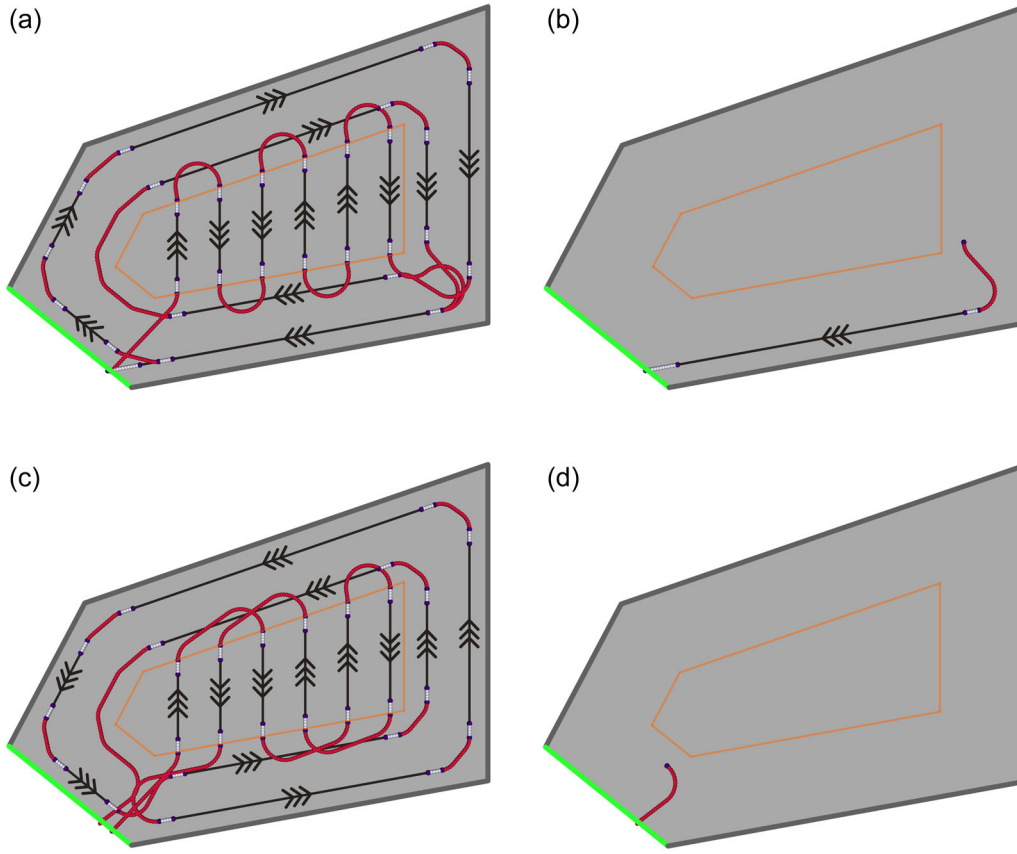
For complex fields partitioned into subfields, the algorithm systematically explores all combinations of driving directions and coverage patterns. This includes generating all potential driving directions from  $0^\circ$  to  $180^\circ$ , which may result in a large number of combinations ( $180^2$  potential combinations for two subfields with a step size of  $\ell_s = 1$ ).

For each combination of driving directions, the algorithm exhaustively explores all possible visitation orders of the subfields, considering both sequential and row-skip coverage patterns. For instance, in a scenario with two subfields, each with an access segment, all order combinations are generated, including the sequences of visiting subfield one before two, and vice versa. For each order, the applicability of both sequential and row-skip coverage patterns for each respective subfield would be further evaluated.

For a given combination of driving directions, the process is initiated while focusing on each subfield that possesses at least one access segment. All the previously outlined steps for initiating coverage in the first subfield, generating and traversing all its tracks using both the sequential and row-skip patterns, as well as ensuring complete coverage of its headlands are executed. Consequently, two coverage paths are obtained for the first subfield: one with a sequential pattern and another with a row-skip pattern. Subsequently, for each of these paths, a circular search is conducted, similar to the search employed for finding the nearest access segment. However, in this case, the objective is to identify the nearest uncovered subfield. Thus, the path is further extended to reach the closest unvisited subfield, completing the coverage of that particular subfield. This process is repeated until all subfields are fully covered.

Subsequently, if the last covered subfield has at least an access segment, a circular search is conducted to identify the shortest exit path. Otherwise, a circular search is conducted to locate the nearest subfield that does possess at least one access segment. Upon reaching this subfield, another circular search is executed to generate the shortest exit path.

Following the acquisition of all feasible solutions for a given combination of driving directions, these solutions are added into the solution space. Subsequently, the method advances to create



**FIGURE 6** Full coverage and exit trajectories for both sequential and row-skip patterns. Headland borders are in orange, access segments in green, turns in red, and transition trajectories in white. (a) Complete coverage (sequential), (b) exit path (sequential), (c) complete coverage (row-skip), and (d) exit path (row-skip).

solutions for different combinations of driving directions, continuously iterating this process until all possible combinations have been considered.

#### 4.2.3 | Selection of the optimal solution

Once potential solutions are generated and stored in a solution space, the performance indicators, or metrics, for each solution, including the coverage rate ( $S_{cov}$ ), overlap rate ( $S_{ovl}$ ), nonworking traveled distance ( $S_{nwd}$ ), and a slightly modified version of operation time ( $S_{otm}$ ) are computed similar to the method in Pour Arab et al. (2023a). Moreover, an innovative metric, the slope of working trajectories ( $S_{slp}$ ), is also considered to enhance the precision of the solution evaluation process.

In this approach, operation time calculation for a solution includes the robot's average speed during backward moves with the implement off ( $V_{rev}$ ), alongside the previously accounted average speeds: when the implement is active ( $V_{on}$ ), during implement transitions from on to off or vice-versa ( $V_{gap}$ ), and with the implement off during a forward move ( $V_{off}$ ). This refinement allows us to distinguish forward and backward moves during a half-turn for the computation of operation time by dividing the total length of each trajectory type

by its respective average speed, thereby providing a more accurate evaluation.

To calculate the slope of working trajectories within a solution, each trajectory is discretized into points, spaced at  $\ell_{slp}$ . The elevation for each point is computed using a ray casting technique inspired by Roth (1982), forming a vertical line at each point to identify its altitude through the intersection with the field's 3D surface. The slope of segments between consecutive points is calculated by determining the angle with the x-y plane using:

$$SLP = \arctan\left(\frac{\text{rise}}{\text{run}}\right) \times \frac{180.0}{\pi}, \quad (3)$$

where *rise* represents the height difference between two points, and *run* is the x-y distance between them. The length of each individual segment is cumulatively accumulated into the categories below based on their slope:

- *Flat*( $\ell_{s0}$ ): if  $0 \leq SLP < 2$  or  $-2 < SLP \leq 0$ .
- *Slightly sloped*( $\ell_{s1}$ ): if  $2 \leq SLP < 5$  or  $-5 < SLP \leq -2$ .
- *Sloped*( $\ell_{s2}$ ): if  $5 \leq SLP < 8$  or  $-8 < SLP \leq -5$ .
- *Steep*( $\ell_{s3}$ ): if  $8 \leq SLP < 12$  or  $-12 < SLP \leq -8$ .
- *Very steep*( $\ell_{s4}$ ): if  $12 \leq SLP < 16$  or  $-16 < SLP \leq -12$ .
- *Extremely steep*( $\ell_{s5}$ ): if  $SLP \geq 16$  or  $SLP \leq -16$ .

For any given solution, the terms  $\ell_{s0}$ – $\ell_{s5}$  represent the total lengths of segments within each slope category. Each of these lengths is then normalized using the following equation:

$$S = \frac{S - S_{\min}}{S_{\max} - S_{\min}}, \quad (4)$$

where  $S_{\min}$  and  $S_{\max}$  represent the lowest and highest values, respectively, for the associated slope category or metric across the solution space.

A solution is then associated with two vectors:  $\mathbf{L}_s = (\ell_{s0} \ell_{s1} \ell_{s2} \ell_{s3} \ell_{s4} \ell_{s5})$  for segment lengths and  $\mathbf{W}_s = (W_{s0} W_{s1} W_{s2} W_{s3} W_{s4} W_{s5})$  for the weights assigned to each slope category. These weights are provided as input data. Finally, for a solution, the slope metric  $S_{slp}$  is computed as follows:

$$S_{slp} = \frac{\mathbf{L}_s \mathbf{W}_s^T}{W_{s0} + W_{s1} + W_{s2} + W_{s3} + W_{s4} + W_{s5}}. \quad (5)$$

Using this metric, along with previously defined metrics, we establish a set of soft constraints or cost functions. These are denoted by  $\mathbf{C} = (1 - S_{cov} S_{ovl} S_{nwd} S_{otm} S_{slp})$ . Corresponding weights, provided as input, are represented as  $\mathbf{W} = (W_{cov} W_{ovl} W_{nwd} W_{otm} W_{slp})$ . The aggregate cost for each solution is then calculated using

$$f = \frac{\mathbf{C} \mathbf{W}^T}{W_{cov} + W_{ovl} + W_{otm} + W_{nwd} + W_{slp}}. \quad (6)$$

Through this set of diverse soft constraints and their weights, the solution with the lowest cost is selected. The customizability of the algorithm is facilitated by the potential adjustments of these weights, corresponding to each soft constraint as well as each slope category. Such flexibility permits prioritization of certain aspects of the optimization, such as limiting overlaps or nonworking traveled distance. This consequently allows users to customize the algorithm according to their specific needs and criteria.

It is noteworthy to mention that the delineation of slope categories was formulated in collaboration with an expert. Nonetheless, they can be fine-tuned to align with the particular demands and criteria of a given operation.

#### 4.2.4 | Complexity analysis

Algorithm 1 shows that the number  $S$  of solutions generated by H-CCPP depends on the following parameters: the number  $N_{sf}$  of subfields into which the field is subdivided, the number  $N_{as}$  of subfields with at least one access segment, the number  $C$  of considered combinations of patterns (sequential and/or row-skip) and turn types (Dubins and/or Reeds–Shepp), and the number  $D$  of explored driving directions, that corresponds to 180 divided by the chosen angle precision  $\ell_s$ .  $S$  is given by Equation (7).

$$S = (D \times C)^{N_{sf}} \times N_{as}. \quad (7)$$

For instance, with a field of simple shape not subdivided ( $N_{sf} = 1$  and  $N_{as} = 1$ ), and a precision of  $\ell_s = 3^\circ$  corresponding to  $D = 60$

explored driving directions, H-CCPP generates between 60 and 240 solutions, depending whether a single or up to four combinations of patterns and turn types are considered. For a field divided into two subfields, with at least an access segment in each subfield, the number of solutions will vary between 7200 and 115,200 depending on the combinations of patterns and turn types.

It can be easily deduced from that formula and the given examples that even if H-CCPP is adapted to various field sizes and configurations, due to this polynomial complexity it is recommended to limit field partitioning to the strict minimum to avoid combinatorial explosion. In the case of subdivided fields, limiting the number of combinations of patterns and turn types may also ensure to maintain a reasonable efficiency.

## 5 | RESULTS AND DISCUSSION

### 5.1 | Experimental setup

As in our original work, we used C++ to develop a program with a graphical user interface for parameter configuration and visual representation of solutions. Computations are sped up using parallel processing via an OpenMP (2021) implementation. The program was tested on an Intel Xeon W-2135 central processing unit @ 3.70 GHz 12 with 32 GB RAM setup.

To evaluate the efficacy of the presented approach and facilitate a comparative analysis with our previous methodology, the proposed data set outlined in Section 4.1.3 was used. The data set encompasses a total of 30 fields, with the initial 20 fields (Fields #1–#20) deemed as simple, featuring no field decomposition. The remaining 10 fields (Fields #21–#30) were designated as complex, incorporating a minimum of two dividing lines to enable exploration of various field decompositions.

The parameters for the robot remained unchanged from the original approach, ensuring consistency in our analysis. However, in the presented approach, certain parameters from the original method, like, global overlap, local loop, switch thresholds, and coverage, were not applicable. Instead, we introduced novel parameters specifically tailored for the new approach, including the driving direction step size ( $\ell_s$ ), the spacing of access segment discretization ( $\ell_a$ ), and the interval between points on the working trajectory used to calculate slope ( $\ell_{slp}$ ). These parameters were configured with values of  $3^\circ$ , 0.5m, and 0.5m, respectively. Detailed information regarding the remaining parameters employed to evaluate the efficacy of our new approach is presented in Table 1.

Section 5.2 includes a detailed and comparative analysis of the approach introduced compared to the original approach. Furthermore, an illustrative result is provided to emphasize noteworthy features of the presented approach.

### 5.2 | Analysis of the results

To facilitate a comprehensive and equitable comparison between our original method (O-CCPP) and the newly introduced method (H-CCPP)

that incorporates an additional metric based on the field inclination, we devised a strategy that merges the solution spaces generated by both methods under a unified solution space before submitting them to the same set of metrics. However, O-CCPP generates a

**TABLE 1** Parameters for proposed approach—Set 1.

Parameter	Description	Value
$\ell_s$	Driving direction step size	3°
$\gamma_{on}$	Minimum turning radius—implement on	15m
$\gamma_{off}$	Minimum turning radius—implement off	1.5m
$V_{on}$	Average speed—implement on	3.5m/s
$V_{gap}$	Average speed—implement transition	2.5m/s
$V_{off}$	Average speed—implement off	1.5m/s
$V_{rev}$	Average speed—implement off—backward	1.0m/s
$\ell_t$	Transition trajectory length	2m
$\ell_o$	Robot-implement offset	2m
$\Delta_{mwd}$	Minimum working distance threshold	8m
$p$	Number of inner trajectories	2
$g$	Number of gap-covering trajectories	1
$w$	Working width	3m
$W_{cov}$	Weight of $S_{cov}$	0.25
$W_{ovl}$	Weight of $S_{ovl}$	0.10
$W_{nwd}$	Weight of $S_{nwd}$	0.05
$W_{otm}$	Weight of $S_{otm}$	0.05
$W_{slp}$	Weight of $S_{slp}$	0.55
$W_{s0}$	Weight of $\ell_{s0}$	0.00
$W_{s1}$	Weight of $\ell_{s1}$	0.10
$W_{s2}$	Weight of $\ell_{s2}$	0.15
$W_{s3}$	Weight of $\ell_{s3}$	0.20
$W_{s4}$	Weight of $\ell_{s4}$	0.25
$W_{s5}$	Weight of $\ell_{s5}$	0.30

significantly larger amount of solutions, that can reach up to more than 1 million in the most complex cases. Incorporating all O-CCPP solutions into the merged solution space would have been too computationally expensive. Importing only the most optimal solution of O-CCPP would not be a good option either, as this solution may not remain the most optimal with the additional metric. Therefore, an intermediate solution was chosen. As in O-CCPP, the solutions are categorized into clusters based on the major direction of working trajectories, and only the most optimal solution of each cluster is presented to the user, we chose to import from O-CCPP only these best-performing solutions of each cluster, to challenge the H-CCPP solutions. Specifically, these best-performing solutions of each cluster are the ones that meet the criteria outlined in our previous work (Pour Arab et al., 2023a), that is, the costs associated with coverage rate, overlap rate, nonworking traveled distance, and operation time, weighted at 0.6, 0.1, 0.2, and 0.1, respectively. This selective approach allows us to manage the complexity and scale of the data effectively, facilitating a comprehensive analysis within practical computational limits.

The average outcomes attained by O-CCPP and H-CCPP are presented in Table 2. Findings for simple fields demonstrate that O-CCPP attained a marginally better coverage rate, while H-CCPP exhibited a slightly lower overlap rate. Conversely, in the case of complex fields, O-CCPP exhibited marginally superior performance in terms of coverage and overlap rates. Notably, H-CCPP demonstrated significantly faster computational time compared to O-CCPP across both simple and complex fields. This was the case even though H-CCPP explored a wider range of driving directions and included two different coverage patterns (sequential and row-skip), while O-CCPP only employed the sequential coverage pattern.

This evaluation revealed a remarkable reduction in computational time achieved by H-CCPP in comparison to O-CCPP. For simple fields, the average computational time was reduced from 170.48 to 6.23s, resulting in an impressive reduction of approximately 96.35%. Similarly, for complex fields, the average computational time decreased from 5382.03 to 3202.89s, representing a substantial reduction of approximately 40.49%. This substantial decrease in computation time offers notable advantages, including faster decision-making, more efficient utilization of resources, and the

**TABLE 2** Comparative numerical analysis of H-CCPP versus O-CCPP.

Approach	Area (ha)		Coverage (%)		Overlap (%)		Computational time (s)	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD
<i>(a) Simple fields</i>								
H-CCPP	4.87	2.82	98.23	0.91	2.62	1.12	6.32	3.85
O-CCPP	4.87	2.82	98.54	0.69	2.63	1.34	170.48	222.45
<i>(b) Complex fields</i>								
H-CCPP	4.69	2.41	97.54	0.80	3.51	1.21	3202.89	1684.70
O-CCPP	4.69	2.41	98.04	0.72	2.55	1.16	5382.03	5077.03

Abbreviation: CCPP, complete coverage path planning.

ability to adapt to changing field conditions. It is also observed that, with the chosen parameters, none of the solutions selected by either method contained Reeds–Shepp curves with backward movements. The limited use of these curves by both methods can be attributed to the operational parameters, more particularly the working width and the robot's minimum turning radius. In such scenarios where the working width equals or exceeds twice the robot's minimum turning radius, short and continuous half-turns may be feasible without necessitating directional switches.

Figure 7 illustrates radar charts providing a comprehensive visualization of the performance of H-CCPP and O-CCPP, employing the criteria outlined in Section 4.2.3. These charts illustrate the number of fields wherein each approach outperformed the other in terms of individual criteria. Higher values for each criterion signify superior performance of the corresponding approach in minimizing costs while considering their respective weights.

Upon examining the radar charts, it becomes apparent that H-CCPP surpassed O-CCPP with respect to slope cost in 18 out of 20 simple fields and in all complex fields. Conversely, O-CCPP exhibited stronger performance in terms of overlaps, operation time, non-working traveled distance, and worked area, securing higher scores in 11, 19, 18, and 14 of the 20 simple fields, respectively, and in 6, 10, 10, and 9 of the 10 complex fields, respectively. These findings suggest that while H-CCPP exhibits greater effectiveness in minimizing slope cost, O-CCPP demonstrates superior performance across other criteria, including worked area, overlaps, nonworking distance, and operation time.

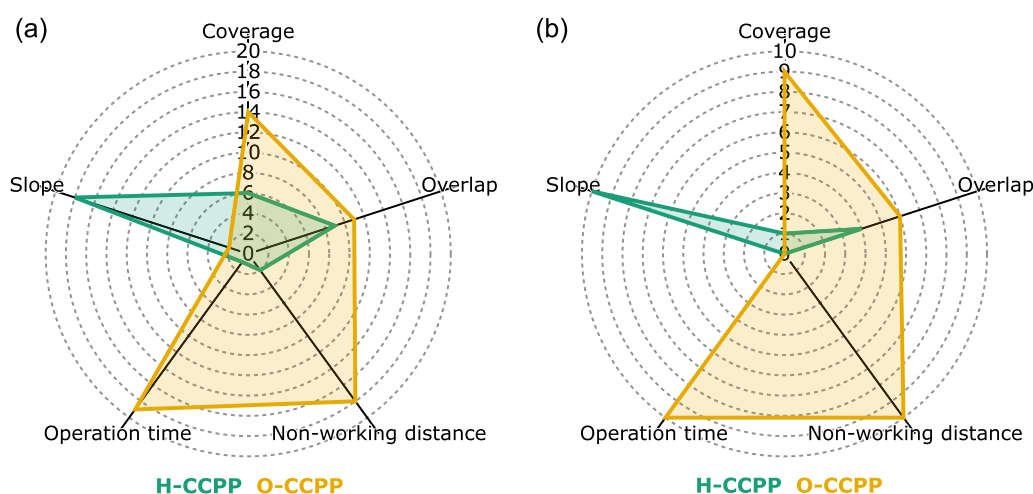
While O-CCPP may exhibit superior performance on specific individual criteria, the computation of the final cost by averaging all costs and employing Equation (6) unveiled that H-CCPP outperformed O-CCPP for 70% of the simple fields. In the case of complex fields, each approach demonstrated efficiency for 50% of the cases. These findings indicate that, on average, H-CCPP attains superior overall performance.

Figure 8 illustrates the results obtained from the application of H-CCPP and O-CCPP on Field #21, which features exclusive accessibility through its two left-side edges. To provide a more intricate view of the results, close-up 3D views are presented in Figure 9, enabling a clearer observation of trajectory inclinations and slopes.

The chosen solution by H-CCPP adopts a row-skip pattern and does not incorporate any field decomposition, as it effectively proposes trajectories perpendicular to the slope of the field. This particular solution exhibits potential efficiency in minimizing energy consumption and mitigating soil runoff induced by irrigation or rainfall. However, it entails a greater number of half-turns, thereby resulting in longer nonworking traveled distances. In contrast, the solution selected by O-CCPP incorporates a field decomposition strategy, yielding shorter nonworking traveled distances.

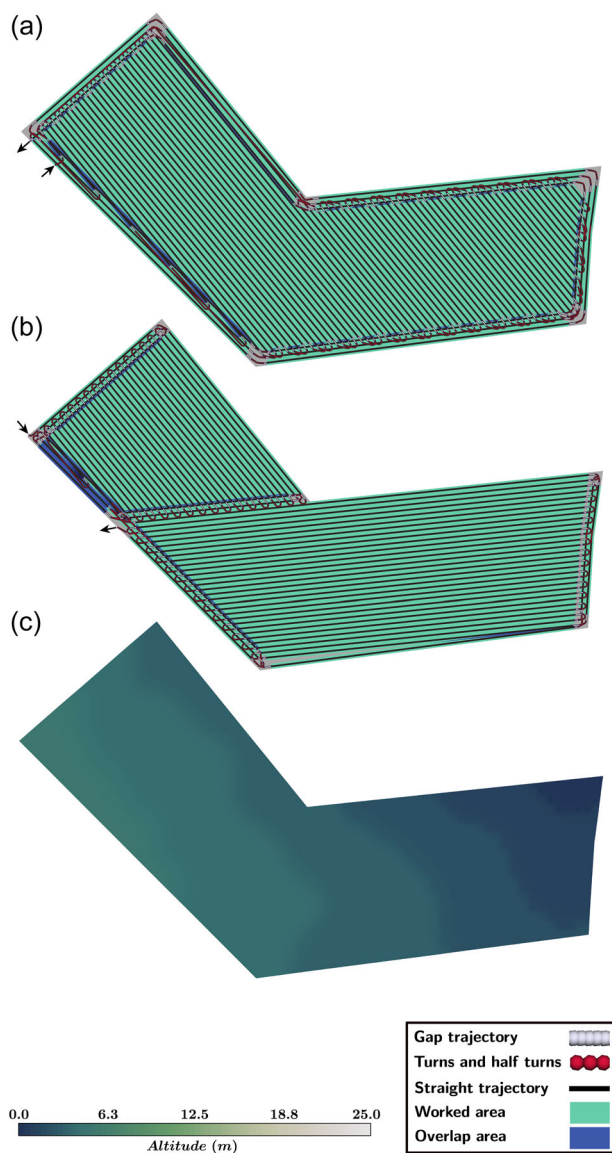
The divergence in the solution selection between O-CCPP and H-CCPP for Field #21 can be attributed to the limited capacity of O-CCPP to execute a row-skip pattern. As a result, there might be an increase in overlaps when the robot tries to find an access segment to exit the field, reducing the overall effectiveness of the solution.

It is noteworthy to highlight the substantial impact that adjusting the weights assigned to the soft constraints can have on the outcomes of both approaches. For instance, assigning greater weight to the cost associated with nonworking traveled distance may prompt H-CCPP to select a solution that incorporates field decomposition. Conversely, for O-CCPP, improving its score in terms of slope cost is not feasible. This is due to the inherent limitation imposed by the number of driving directions, namely, the general directions, which are constrained to align with the field's edges within the results obtained by O-CCPP. Furthermore, as the best solution from all families, encompassing solutions with all conceivable driving directions, has already been included in the solution space, and a significant weight has been assigned to slope cost, the solutions produced by O-CCPP were not optimal with respect to slope cost. Consequently, its performance could not be improved beyond what has been achieved in the current evaluation.



**FIGURE 7** Radar charts illustrating the number of fields wherein each approach outperformed the other in terms of individual criteria. Higher values for each criterion indicate better performance. (a) Simple fields and (b) complex fields. CCPP, complete coverage path planning.





**FIGURE 8** Visual representation of results for Field #21 using H-CCPP and O-CCPP. (a) H-CCPP, (b) O-CCPP, and (c) 3D data. CCPP, complete coverage path planning.

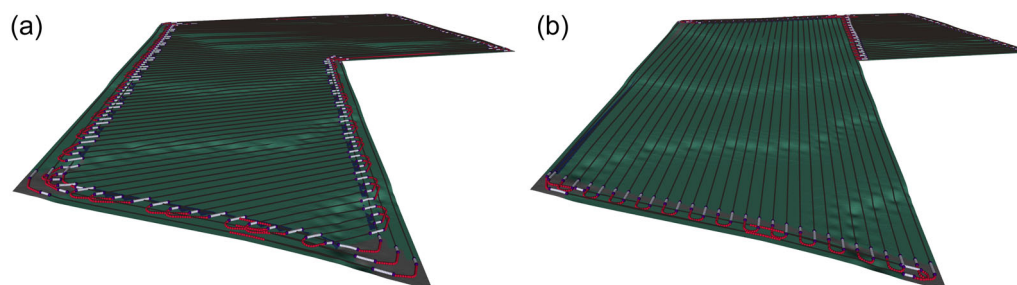
In subsequent sections, we highlight H-CCPP's adaptability and its ability to meet diverse operational requirements, showcasing its flexibility across different scenarios.

### 5.3 | H-CCPP under different optimization settings

The purpose of this section is to exemplify the adaptability of the proposed approach in generating solutions that effectively satisfy diverse criteria. To achieve this, several tests were conducted on Field #22, wherein the emphasis was placed on the weight of a specific soft constraint within the selection method. The different settings of the selection method are presented in Table 3, while the remaining parameters remain unchanged as specified in Table 1. To facilitate a clear visual differentiation between the solutions derived under each optimization setting, the weight of one particular soft constraint was adjusted to one, while the weights of the remaining constraints were set to zero for each respective setting, as given in Table 3.

During the conducted tests on Field #22, the preexisting solution space generated for this specific field was used. As a result, the component of the approach tasked with constructing the solution space and calculating all associated costs was executed only once. In contrast, the computation of the final cost, determined by Equation (6), was performed three times, corresponding to each optimization setting. Thus, the computation time remained consistent across all tests.

Figure 10 depicts the results achieved for each optimization setting. The solution chosen for Setting #1 (top image), where the entire weight was allocated to the coverage cost, includes no field decomposition. This choice is primarily influenced by the fact that decomposing the field into subfields would result in uncovered corners and subsequently lead to a reduced coverage area in comparison to solutions that do not involve field decomposition. However, it is worth mentioning that enhancing this approach to encompass the coverage of corners during headland coverage through reverse moves might alter this result.



**FIGURE 9** Close-up views of results for Field #21 using (a) H-CCPP and (b) O-CCPP. CCPP, complete coverage path planning.



**TABLE 3** Different settings for the selection method of H-CCPP.

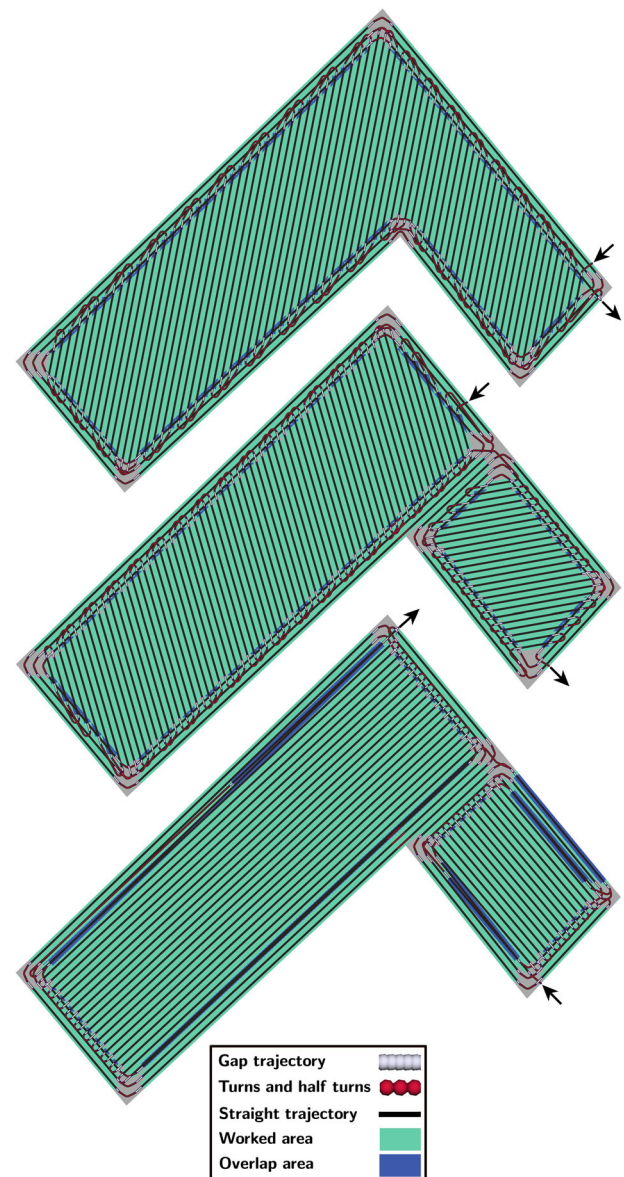
Parameter	Description	Value
(a) Optimization setting #1		
$W_{cov}$	Weight of $S_{cov}$	1
$W_{ovl}$	Weight of $S_{ovl}$	0
$W_{nwd}$	Weight of $S_{nwd}$	0
$W_{otm}$	Weight of $S_{otm}$	0
$W_{slp}$	Weight of $S_{slp}$	0
(b) Optimization setting #2#1		
$W_{cov}$	Weight of $S_{cov}$	0
$W_{ovl}$	Weight of $S_{ovl}$	1
$W_{nwd}$	Weight of $S_{nwd}$	0
$W_{otm}$	Weight of $S_{otm}$	0
$W_{slp}$	Weight of $S_{slp}$	0
(c) Optimization setting #3#2		
$W_{cov}$	Weight of $S_{cov}$	0
$W_{ovl}$	Weight of $S_{ovl}$	0
$W_{nwd}$	Weight of $S_{nwd}$	1
$W_{otm}$	Weight of $S_{otm}$	0
$W_{slp}$	Weight of $S_{slp}$	0

Abbreviation: CCPP, complete coverage path planning.

The solution derived for Setting #2 (Figure 10 middle image), where exclusive weight was allocated to the overlap cost, is noteworthy. The selected solution encompasses a field decomposition strategy, where a sequential pattern was employed for one subfield, while a row-skip pattern was utilized for another. This particular arrangement arises from the adoption of a row-skip pattern for the first subfield, thereby facilitating reduced overlap when traveling to the second subfield.

Under Setting #3, where the entire weight is dedicated to the cost of nonworking traveled distance, the selected solution (as seen in the bottom image in Figure 10) features a field decomposition. Within this decomposition, the driving direction of each subfield aligns with its lengthiest edge. This outcome is expected, as selecting the longest edge as the driving direction would inherently reduce the number of half-turns and subsequently minimize the nonworking traveled distance.

Undoubtedly, these conducted tests exemplified the adaptability and versatility of the proposed approach in generating solutions capable of fulfilling various combinations of constraints and criteria. Through the modification of each soft constraint's weight, the selection method successfully identified distinct solutions that prioritize specific criteria over others. In real-world scenarios, where the simultaneous consideration and optimization of multiple factors are imperative, this remarkable ability to adapt and strike a favorable balance between conflicting criteria holds significant value.

**FIGURE 10** Visual representation of outcomes for Field #22 across various optimization settings: Settings #1–#3 (from top to bottom).

## 5.4 | H-CCPP under different operational parameters

This subsection illustrates the flexibility of our approach when applied to different operational parameters, primarily those associated with the characteristics of the robot and its implementation. A comprehensive test was conducted across our entire data set with the distinct set of parameters presented in Table 4. The selection of these parameters was intentionally aimed at challenging the operational limits where continuous half-turns are usually infeasible, to enforce conditions under which the use of Reeds–Shepp curves with backward movements becomes necessary. Specifically, we adjusted the minimum

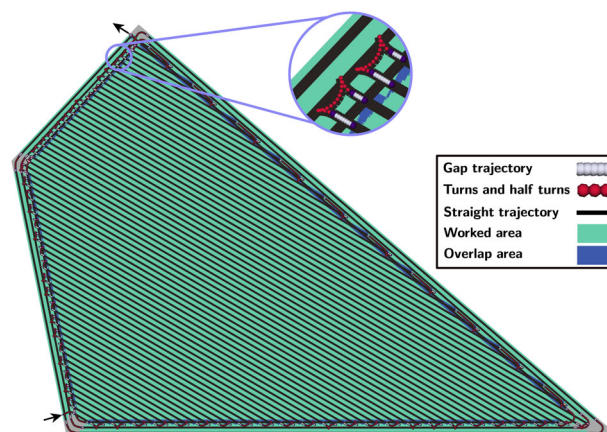
turning radius to be greater than half of the working width while keeping an identical headland width.

The results reported in Table 5 demonstrate a significant presence of *Backward Nonworking Travel Distances*, with a mean of 27.07% for simple fields and 28.68% for complex fields, indicating the incorporation of turns involving backward movements. This adaptation resulted in an average coverage of 98.90% for simple fields and 98.38% for complex fields, with overlaps at 4.35% and

5.10%, respectively. It was particularly advantageous in decreasing total nonworking traveled distances and required less space in headlands to perform half-turns, as illustrated in Figure 11. Such turns are a practical solution when the allocated headland zone is limited. They also offer a potential advantage in reducing soil compaction by decreasing the size of headland zones.

It is important to note that all solutions where turns were generated using the Dubins method were discarded due to their infeasibility near field corners, where some portions of turns would have lied outside the field boundaries. This early exclusion led to a notable decrease in computation time compared to the results shown in Table 2, as the calculation of metrics for these unaccepted solutions was terminated prematurely.

Our study reveals that the applicability of Dubins and Reeds–Shepp methods depends on the robot's operational parameters. When the minimum turning radius is at most half the working width, both methods yield similar paths, without the need for reverse moves. Conversely, scenarios with larger turning radius demand the Reeds–Shepp method's reverse capabilities for effective turn generation, especially in limited headland areas. Not all robots can execute reverse moves, influencing method selection. When this characteristic is known, preselecting one method or the other depending on the robot capabilities and headland width can significantly reduce the number of generated solutions and consequently the computational time, as discussed in Section 4.2.4.



**FIGURE 11** Visual representation of results for Field #20 using H-CCPP. CCPP, complete coverage path planning.

**TABLE 4** Parameters for proposed approach—Set 2.

Parameter	Description	Value
$\ell_s$	Driving direction step size	3°
$\gamma_{on}$	Minimum turning radius—implement on	10m
$\gamma_{off}$	Minimum turning radius—implement off	2.8m
$V_{on}$	Average speed—implement on	4.5m/s
$V_{gap}$	Average speed—implement transition	1.5m/s
$V_{off}$	Average speed—implement off	3.5m/s
$V_{rev}$	Average speed—implement off—backward	3m/s
$\ell_t$	Transition trajectory length	1.5m
$\ell_o$	Robot-implement offset	1m
$\Delta_{mwd}$	Minimum working distance threshold	3m
$p$	Number of inner trajectories	2
$g$	Number of gap-covering trajectories	1
$w$	Working width	3m
$W_{cov}$	Weight of $S_{cov}$	0.30
$W_{ovl}$	Weight of $S_{ovl}$	0.15
$W_{nwd}$	Weight of $S_{nwd}$	0.10
$W_{otm}$	Weight of $S_{otm}$	0.10
$W_{slp}$	Weight of $S_{slp}$	0.35
$W_{s0}$	Weight of $\ell_{s0}$	0.00
$W_{s1}$	Weight of $\ell_{s1}$	0.10
$W_{s2}$	Weight of $\ell_{s2}$	0.15
$W_{s3}$	Weight of $\ell_{s3}$	0.20
$W_{s4}$	Weight of $\ell_{s4}$	0.25
$W_{s5}$	Weight of $\ell_{s5}$	0.30

**TABLE 5** Numerical results of H-CCPP under varied operational parameters.

Fields	Area (ha)		Coverage (%)		Overlap (%)		BNTD (%)		Computational time (s)	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Simple	4.87	2.82	98.90	0.66	4.35	2.05	27.07	8.17	4.48	2.45
Complex	4.87	2.82	98.38	0.62	5.10	1.09	28.68	4.37	1391.37	607.46

Note: The BNTD column indicates the ratio of backward nonworking travel distance to total nonworking travel distance.

Abbreviation: CCPP, complete coverage path planning.

In this study, we explored the capabilities of our path-planning system under a range of operational parameters and settings, showcasing its adaptability and versatility. While no real-world experiments were performed at this point, the range of parameters tested was designed to reflect a variety of potential agricultural scenarios and robot characteristics. Such flexibility also suggests the high potential of our method in conducting exploratory studies to identify optimal operational parameters that lead to more efficient solutions and facilitate the design of robotic systems to fit optimal settings.

## 6 | DATA SHARING AND DATA ACCESSIBILITY

Data supporting the findings of this study are openly available as part of the *Agricultural Fields 2D and 3D Models Data set* on Zenodo (Pour Arab & Essert, 2024). Additionally, the results of the test discussed in Section 5.4 are published on this data set in the form of way-points. In future work, we plan to provide more results using a broader range of parameters and setups.

## 7 | CONCLUSION

In this article, we introduced a novel hybrid approach (H-CCPP) for generating highly efficient coverage paths for agricultural robots. This novel approach offers several notable advantages over our previous CCPP method (O-CCPP), including enhanced processing speed, the exploration of diverse driving directions for parallel track generation within the field, surpassing the limitation of solely relying on trajectories aligned with one field edge. Moreover, H-CCPP embraced the adoption of row-skip patterns for both simple and complex fields, comprehensively investigating all conceivable combinations of coverage patterns for all subfields. Furthermore, H-CCPP demonstrated the capability to automatically determine both entry and exit points, contrasting with the limited determination of the exit point based on a given entry point in O-CCPP. To showcase the effectiveness of H-CCPP, we introduced a comprehensive public data set of 30 fields located in France and we conducted a rigorous evaluation, comparing its performance to O-CCPP.

While H-CCPP demonstrated significantly superior performance in minimizing the slope cost, O-CCPP exhibited slightly better results in terms of overlaps, worked area, operation time, and nonworking traveled distance. However, upon computing the final cost, which takes into account all criteria, H-CCPP surpassed O-CCPP in 70% of simple fields and 50% of complex fields.

Furthermore, we demonstrated the remarkable versatility of H-CCPP in generating solutions capable of meeting diverse constraints and criteria. This was exemplified through a series of tests conducted on Field #22, wherein the weight assigned to each soft constraint was adjusted within the selection method. As a result of this adaptability, the selection method successfully identified varied solutions that prioritized specific criteria over others.

The findings of the study demonstrated that H-CCPP is a highly effective approach for generating coverage paths for agricultural robots, particularly in relation to minimizing slope cost. Its notable flexibility and adaptability make it invaluable in real-world applications where the simultaneous consideration and optimization of multiple factors are crucial.

Despite the progress made, we also acknowledge several limitations and potential areas for future research. A potential area for improvement in H-CCPP lies in the incorporation of reverse moves to cover corners during headland coverage. This enhancement has the potential to enhance the overall coverage rate performance of H-CCPP. Other limitations are the need for conducting real-world experiments for a more comprehensive evaluation, handling static and dynamic obstacles, consideration of robot capacity in terms of energy and agricultural materials, and the inclusion of multirobot systems for cooperative task completion. Addressing these areas of improvement will further enhance the performance, adaptability, and efficiency of autonomous agricultural robots, providing even more compelling solutions for the agricultural industry.

## ACKNOWLEDGMENTS

This work was funded by research grant CIFRE 2019/1084 from Technology and Strategy Group (T&S) and the French National Association for Research and Technology (ANRT).

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in *Agricultural Fields 2D and 3D Models Data set* at <https://doi.org/10.5281/zenodo.10949632>.

## ORCID

Danial Pour Arab  <https://orcid.org/0000-0002-2640-7496>

## REFERENCES

- Becker, K., Oehler, M. & Von Stryk, O. (2022) 3D coverage path planning for efficient construction progress monitoring. In: *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* IEEE, pp. 174–179.
- Bochtis, D. & Sørensen, C. (2010) The vehicle routing problem in field logistics: part II. *Biosystems Engineering*, 105, 180–188.
- Bochtis, D.D., Sørensen, C.G., Busato, P. & Berruto, R. (2013) Benefits from optimal route planning based on b-patterns. *Biosystems Engineering*, 115, 389–395.
- Bochtis, D., Sørensen, C., Busato, P., Hameed, I., Rodias, E., Green, O. et al. (2010) Tramline establishment in controlled traffic farming based on operational machinery cost. *Biosystems Engineering*, 107, 221–231.
- Burger, M., Huiskamp, M. & Keviczky, T. (2013) Complete field coverage as a multi-vehicle routing problem. *IFAC Proceedings Volumes*, 46, 97–102. 4th IFAC Conference on Modelling and Control in Agriculture, Horticulture and Post Harvest Industry.
- Cao, Y., Han, Y., Chen, J., Liu, X., Zhang, Z. & Zhang, K. (2019) Optimal coverage path planning algorithm of the tractor-formation based on probabilistic roadmaps. In: *2019 IEEE International Conference on Unmanned Systems and Artificial Intelligence (ICUSAI)*. IEEE, pp. 27–32.
- Cariou, C., Laneur, J., Roux, J.-C. & Lenain, R. (2020) Multi-robots trajectory planning for farm field coverage. In: *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, pp. 351–356.



- Chin, R., Catal, C. & Kassahun, A. (2023) Plant disease detection using drones in precision agriculture. *Precision Agriculture*, 24, 1663–1682.
- Conesa-Muñoz, J., Pajares, G. & Ribeiro, A. (2016) Mix-opt: a new route operator for optimal coverage path planning for a fleet in an agricultural environment. *Expert Systems with Applications*, 54, 364–378.
- Dakulovic, M. & Petrovic, I. (2012) Complete coverage path planning of mobile robots for humanitarian demining. *Industrial Robot: An International Journal*, 39, 484–493.
- Dogru, S. & Marques, L. (2015) Energy efficient coverage path planning for autonomous mobile robots on 3D terrain. In: *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*. IEEE, pp. 118–123.
- Dubins, L.E. (1957) On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79, 497–516.
- Edwards, G.T., Hinge, J., Skou-Nielsen, N., Villa-Henriksen, A., Sørensen, C.A.G. & Green, O. (2017) Route planning evaluation of a prototype optimised infield route planner for neutral material flow agricultural operations. *Biosystems Engineering*, 153, 149–157.
- Evans, J.T., Pitla, S.K., Luck, J.D. & Kocher, M. (2020) Row crop grain harvester path optimization in headland patterns. *Computers and Electronics in Agriculture*, 171, 105295.
- Fasiolo, D.T., Scalera, L., Maset, E. & Gasparetto, A. (2023) Towards autonomous mapping in agriculture: a review of supportive technologies for ground robotics. *Robotics and Autonomous Systems*, 169, 104514.
- Giannakis, E., Kushta, J., Giannadaki, D., Georgiou, G.K., Bruggeman, A. & Lelieveld, J. (2019) Exploring the economy-wide effects of agriculture on air quality and health: evidence from Europe. *Science of the Total Environment*, 663, 889–900.
- Géoportail. (2023) *The national portal for territorial knowledge implemented by IGN*. <https://www.geoportail.gouv.fr/>
- Hameed, I.A. (2014) Intelligent coverage path planning for agricultural robots and autonomous machines on three-dimensional terrain. *Journal of Intelligent & Robotic Systems*, 74, 965–983.
- Hameed, I.A., la Cour-Harbo, A. & Osen, O.L. (2016) Side-to-side 3D coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. *Robotics and Autonomous Systems*, 76, 36–45.
- IGN. (2023) REST API of IGN for determining the elevation of one or more points. <https://geoservices.ign.fr/documentation/services/API-et-services-ogc/calcul-altimetric-rest>
- Jensen, M.A.F., Bochtis, D., Sørensen, C.G., Blas, M.R. & Lykkegaard, K.L. (2012) In-field and inter-field path planning for agricultural transport units. *Computers & Industrial Engineering*, 63, 1054–1061.
- Jensen, M.F., Bochtis, D. & Sørensen, C.G. (2015) Coverage planning for capacitated field operations, part II: optimisation. *Biosystems Engineering*, 139, 149–164.
- Jeon, C.-W., Kim, H.-J., Yun, C., Han, X. & Kim, J.H. (2021) Design and validation testing of a complete paddy field-coverage path planner for a fully autonomous tillage tractor. *Biosystems Engineering*, 208, 79–97.
- Jin, J. & Tang, L. (2010) Optimal coverage path planning for arable farming on 2D surfaces. *Transactions of the ASABE*, 53, 283–295.
- Jin, J. & Tang, L. (2011) Coverage path planning on three-dimensional terrain for arable farming. *Journal of Field Robotics*, 28, 424–440.
- Khajepour, A., Sheikhmohammady, M. & Nikbakhsh, E. (2020) Field path planning using capacitated arc routing problem. *Computers and Electronics in Agriculture*, 173, 105401.
- Mahajan, S., Darodkar, A., Dahake, P. & Bahl, L. (2023) Review on pesticides spraying agricultural drone. *AIP Conference Proceedings*, 2753, 020017. <https://doi.org/10.1063/5.0127796>
- Mail, M.F., Maja, J.M., Marshall, M., Cutulle, M., Miller, G. & Barnes, E. (2023) Agricultural harvesting robot concept design and system components: a review. *AgriEngineering*, 5, 777–800.
- Marques, A., Martins, I.S., Kastner, T., Plutzer, C., Theurl, M.C., Eisenmenger, N. et al. (2019) Increasing impacts of land use on biodiversity and carbon sequestration driven by population and economic growth. *Nature Ecology & Evolution*, 3, 628–637.
- Meftaul, I.M., Venkateswarlu, K., Dharmarajan, R., Annamalai, P. & Megharaj, M. (2020) Pesticides in the urban environment: a potential threat that knocks at the door. *Science of the Total Environment*, 711, 134612.
- Mier, G., Valente, J. & de Bruin, S. (2023) Fields2Cover: an open-source coverage path planning library for unmanned agricultural vehicles. *IEEE Robotics and Automation Letters*, 8, 2166–2172.
- Nilsson, R.S. & Zhou, K. (2020) Method and bench-marking framework for coverage path planning in arable farming. *Biosystems Engineering*, 198, 248–265.
- Nørremark, M., Nilsson, R.S. & Sørensen, C.A.G. (2022) In-field route planning optimisation and performance indicators of grain harvest operations. *Agronomy*, 12, 1151.
- Oksanen, T. & Visala, A. (2009) Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26, 651–668.
- OpenMP. (2021) *The OpenMP API supports multi-platform shared-memory parallel programming in C/C++ and Fortran*. <https://www.openmp.org/>
- Plessen, M.G. (2019) Optimal in-field routing for full and partial field coverage with arbitrary non-convex fields and multiple obstacle areas. *Biosystems Engineering*, 186, 234–245.
- Pour Arab, D. & Essert, C. (2024) *Agricultural fields 2D and 3D models dataset*. Zenodo. <https://doi.org/10.5281/zenodo.10949632>
- Pour Arab, D., Spisser, M. & Essert, C. (2023a) Complete coverage path planning for wheeled agricultural robots. *Journal of Field Robotics*, 40, 1460–1503.
- Pour Arab, D., Spisser, M. & Essert, C. (2023b) Introduction of a row-skip pattern in complete coverage path planning for agricultural fields. In: *2023 9th International Conference on Automation, Robotics and Applications (ICARA)*. IEEE, pp. 107–111.
- Reeds, J. & Shepp, L. (1990) Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145, 367–393.
- Roth, S.D. (1982) Ray casting for modeling solids. *Computer Graphics and Image Processing*, 18, 109–144.
- Santos, L.C., Santos, F.N., Aguiar, A.S., Valente, A. & Costa, P. (2022) Path planning with hybrid maps for processing and memory usage optimisation. In: *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, pp. 27–33. <https://doi.org/10.1109/ICARSC55462.2022.9784767>
- Seyyedhasani, H., Dvorak, J.S. & Roemmele, E. (2019) Routing algorithm selection for field coverage planning based on field shape and fleet size. *Computers and Electronics in Agriculture*, 156, 523–529.
- Shen, M., Wang, S., Wang, S. & Su, Y. (2020) Simulation study on coverage path planning of autonomous tasks in hilly farmland based on energy consumption model. *Mathematical Problems in Engineering*, 2020, 4535734.
- Shepard, D. (1968) A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM National Conference*. ACM '68. Association for Computing Machinery, New York, NY, USA. pp. 517–524. <https://doi.org/10.1145/800186.810616>
- Tang, Y., Zhou, H., Wang, H. & Zhang, Y. (2023) Fruit detection and positioning technology for a Camellia Oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. *Expert Systems with Applications*, 211, 118573.
- Turk, G. (1994) *The PLY polygon file format*. <http://gamma.cs.unc.edu/POWERPLANT/papers/ply.pdf>
- Utamima, A., Reiners, T. & Ansariipoor, A.H. (2019) Evolutionary estimation of distribution algorithm for agricultural routing planning in field logistics. *Procedia Computer Science*, 161, 560–567.
- Vahdanjoo, M., Zhou, K. & Sørensen, C.A.G. (2020) Route planning for agricultural machines with multiple depots: manure application case study. *Agronomy*, 10, 1608.

- Van Doren, C. A., Stauffer, R.S. & Kidder, E.H. (1951) Effect of contour farming on soil loss and runoff. *Soil Science Society of America Journal*, 15, 413–417.
- Vieira, D., Orjuela, R., Spisser, M. & Basset, M. (2021) Longitudinal vehicle control based on off-road tire model for soft soil applications. *IFAC-PapersOnLine*, 54, 304–309.
- Vieira, D., Orjuela, R., Spisser, M. & Basset, M. (2022) Positioning and attitude determination for precision agriculture robots based on IMU and two RTK GPSs sensor fusion. *IFAC-PapersOnLine*, 55, 60–65.
- Viet, H.H., Dang, V.-H., Laskar, M.N.U. & Chung, T. (2013) BA\*: an online complete coverage algorithm for cleaning robots. *Applied Intelligence*, 39, 217–235.
- Wang, Y., He, Z., Cao, D., Ma, L., Li, K., Jia, L. et al. (2023) Coverage path planning for kiwifruit picking robots based on deep reinforcement learning. *Computers and Electronics in Agriculture*, 205, 107593.
- Wang, S., Song, J., Qi, P., Yuan, C., Wu, H., Zhang, L. et al. (2022) Design and development of orchard autonomous navigation spray system. *Frontiers in Plant Science*, 13, 960686.
- Xu, S. & Rai, R. (2024) Vision-based autonomous navigation stack for tractors operating in peach orchards. *Computers and Electronics in Agriculture*, 217, 108558.
- Zhang, M., Li, X., Wang, L., Jin, L. & Wang, S. (2024) A path planning system for orchard mower based on improved A\* algorithm. *Agronomy*, 14, 391.
- Zhou, K., Jensen, A.L., Bochtis, D., Nørremark, M., Kateris, D. & Sørensen, C.G. (2020) Metric map generation for autonomous field operations. *Agronomy*, 10, 83.

**How to cite this article:** Pour Arab, D., Spisser, M. & Essert, C. (2025) 3D hybrid path planning for optimized coverage of agricultural fields: a novel approach for wheeled robots. *Journal of Field Robotics*, 42, 455–473. <https://doi.org/10.1002/rob.22422>