Received: 7 March 2023 | Revised: 27 October 2023 | Accepted: 22 December 2023

DOI: 10.1002/rob.22286

SURVEY ARTICLE

WILEY

Optimal guidance track generation for precision agriculture: A review of coverage path planning techniques

Maria Höffmann | Shruti Patel | Christof Büskens

Optimization and Optimal Control, Center for Industrial Mathematics, University of Bremen, Bremen, Germany

Correspondence
Maria Höffmann, Optimization and Optimal Control, Center for Industrial Mathematics, University of Bremen, Bremen, Germany.
Email: mhffmann@uni-bremen.de

Funding information
German Government's Special Purpose Fund held at Landwirtschaftliche Rentenbank, Grant/Award Number: 909052

Abstract

The Complete Coverage Path Planning (CCPP) problem is a subfield of industrial motion planning that has applications in various domains, ranging from mobile robotics to treatment applications. Especially in precision agriculture with a high level of automation, the use of CCPP techniques is essential for efficient resource utilization, reduced soil compaction, and increased yields. This paper reviews the CCPP problem in the context of machines operating in agricultural fields and proposes a methodological approach consisting of three steps: Generating the *Guidance Tracks* (i.e., the track system along which the path should be oriented), determining the traversing sequence through these tracks, and planning a smooth and drivable path. This paper provides an in-depth review of optimization-based approaches that deal with the first step, the generation of the guidance track system. Thereby, a comprehensive and pedagogical approach for the generation of guidance tracks for arbitrarily shaped two-dimensional regions of interest is provided, along with an overview and detailed elaboration on different exact cellular decomposition techniques found in literature. Furthermore, cost functions are outlined for the different approaches presented in this work, which are utilized to generate optimal guidance tracks. Finally, this survey serves as an introductory guide for research and practitioners to solve the CCPP problem effectively and efficiently.

KEYWORDS

agriculture, complete coverage path planning, exact cellular decomposition, field robotics, guidance track system, optimization

1 | INTRODUCTION

The Complete Coverage Path Planning (CCPP) problem arises when dealing with mobile agents engaged in operations that require an efficient route traversing every single point in a given area of interest. Classified as a subfield of industrial motion planning, solving the CCPP problem in an optimal way is highly relevant for a large number of domains, ranging from mobile robot applications such as agriculture, autonomous cleaning (Ntawumenyikizaba et al., 2012), autonomous lawn mowing (Höffmann, Clemens, et al., 2022), structural inspection (Galceran et al., 2015), and surveillance (Basilico & Carpin, 2015), to industrial or treatment applications such as milling (Kalburgi et al., 2020), laser cleaning (Ye et al., 2020) or spray painting (Kiemel et al., 2019). The use of CCPP techniques is especially

The project is supported by funds of the German Government's Special Purpose Fund held at Landwirtschaftliche Rentenbank.

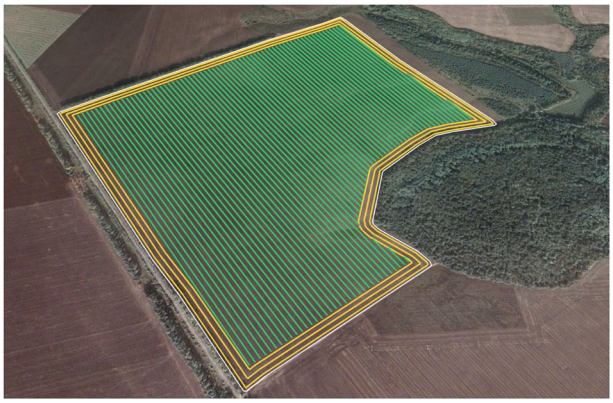

FIGURE 1 Complete coverage of an agricultural field with headland (yellow) and in-field (green) guidance tracks (Höffmann, Patel, et al., 2022).

essential in precision agriculture using unmanned ground vehicles or the so-called auto-guidance devices, as they facilitate efficient use of resources such as fuel, fertilizers, and land, reduce soil compaction, and ultimately increase yields. In this work, the CCPP problem is reviewed in the context of machines operating in agricultural fields such as in Figure 1.

There are various classifications of CCPP algorithms to be found in the literature (Oksanen, 2007). Depending on whether or not the boundary of the area to be traversed is a priori mapped out, CCPP algorithms can be identified as *offline* or *online*, respectively, with the latter approach also termed as *sensor-based coverage* as it relies on real-time sensor data from the robot to deduce information about the environment. Furthermore, according to the dimension of the space to be covered, CCPP approaches may be classified into two-dimensional (2D), when the robot motion is constrained to a plane, or three-dimensional (3D), with 3D coverage algorithms finding relevance in operations with unmanned aerial vehicles (UAVs) or autonomous underwater vehicles (AUVs), for instance for applications involving structural inspections (Bircher et al., 2016). Finally, in contrast to the classical *single-robot* coverage algorithms, much attention has been given to *multirobot* CCPP solutions recently, useful in case of broader area coverage to optimize operation time and increase robustness (Fevgas et al., 2022). The focus of this work is on complete coverage of agricultural fields, therefore the methods surveyed in this paper are offline and 2D. Moreover, although the unique challenges that arise in multirobot coverage path planning are not considered here specifically, the reviewed methods are adaptable and can be extended to multiagent systems, for instance by dividing the coverage task into several subtasks and allocating them to multirobots in advance (Cai et al., 2023; Lin & Huang, 2021).

1.1 | CCPP problem formulation

The objective of CCPP is to determine an efficient path for a robot with a fixed working width $w \in \mathbb{R}_+$, ensuring maximum coverage of the *region of interest* (ROI) while staying within the designated *workspace*. In certain cases, these two areas may differ. For example, UAVs may slightly extend beyond the ROI boundary to achieve improved coverage or smoother turns (Coombes et al., 2020). However, this paper specifically concentrates on ground-based agricultural robots, assuming that the ROI and workspace are equivalent, thereby restricting the robot's movement within the ROI.

In this paper, we define the ROI, denoted as $\mathcal{R}$, as a 2D polygonal area. The ROI is represented by a single polygonal outer boundary and contains several polygonal obstacles that are entirely located within this boundary. The obstacles within the ROI are nonself-intersecting and do not intersect with each other.

The aim of CCPP is to determine an efficient path, such that the swath of a machine covers as much as possible of the region $\mathcal{R}$. Thereby, different objectives and constraints can be considered, including but not limited to (Galceran & Carreras, 2013):

- Maximize the coverage of $\mathcal{R}$;
- Path should be feasible w.r.t. workspace (here $\mathcal{R}$);
- Minimize the overlap of covered areas;
- Minimize the overall path length;
- Minimize the overall execution time;
- Minimize the overall energy consumption.

Requirement (1) is the key element in a CCPP problem. A critical point, especially when considering agricultural ground machines, is requirement (2), to ensure that the boundaries of $\mathcal{R}$ are not violated. For time and energy-efficient processing, requirements (3)–(6) are commonly used examples, especially in this domain to save resources and ensure environmental sustainability. Note that for homogeneous motion on a flat surface, minimal path length automatically accounts for minimal execution time, but considerations such as sloped terrain for ground robots or wind direction for UAVs might make optimal path length and optimal time into competing objectives. To ensure that the resulting path is actually drivable by a nonholonomic machine with a minimum turning radius $r_{\min} \in \mathbb{R}_+$, the path has to fulfill additional requirements:

- Smooth, that is, at least $\mathcal{C}^3$ continuity has proven to be desirable to ensure smooth curvature (Wei & Liu, 2010);
- Curvature constrained by $\kappa_{\max} = r_{\min}^{-1}$.

Optimizing coverage of an agricultural field involves balancing the trade-off between maximizing area coverage and minimizing energy consumption.

1.2 | Methodological approach

In the context of operating small-dimension robots within confined spaces like vacuum cleaners and lawn mowers, the utilization of randomized motion is a prevalent approach. This method involves the robot moving along straight lines and executing turns with random

FIGURE 2 Methodological approach for solving a CCPP problem: After the *Guidance Track Generation*, the traversing order of these tracks is determined in the *Route Planning*. Based on this traversing sequence, a smooth coverage path is determined during *Path Smoothing*. This paper focuses on the first step (Guidance Track Generation).

angles upon reaching the boundary of the ROI. Despite its suboptimality, it has been discussed as a cost-effective solution for such scenarios. As stated in Choset (2001), the main benefit of such a random approach is that localization sensors and complex path planning algorithms are not necessary. However, this method is wholly unsuitable for (precision) agricultural tasks as it does not provide the level of accuracy and the operation costs of the platform as well as damage to arable soil are significantly higher (Santos et al., 2020).

A common method for processing an agricultural field is to divide it into two areas: the headland and the interior field. The headland is the area along the boundaries of the field (Hameed et al., 2010). The width of this area can be selected to match the machine specifics, such as the maximum curvature κ_{max} and the working width w. The interior field is covered using parallel tracks. This approach allows for complete coverage of the entire field and is shown in Figure 1. Based on this initial idea for classifying the field is the further procedure for determining a coverage path, see Figure 2. The open-source CCPP library for agricultural vehicles *Fields2Cover* (Mier et al., 2023), uses the same methodology in its algorithms.

- Generate *Guidance Tracks*, so to speak the track system along which the path should be oriented. The headland guidance tracks are closed polygons, the interior guidance tracks are open polygons.
- Determine the order in which the guidance tracks should be traversed. This is referred to in the literature as *Route Planning*, and deals mostly with the interior tracks.
- Determine a smooth and drivable coverage path based on the guidance tracks and the sequence.

In this paper, we give an in-depth review of optimization-based approaches that deal with the first step, the generation of the guidance track system. This typically starts with a decomposition method used to divide the area of interest into sub-areas or cells, and tracks are oriented optimally to cover each cell. Route planning and path smoothing, although essential to the motion planning pipeline of a robot, are auxiliary to the first step—guidance track generation—when it comes to determining the completeness of the coverage. Therefore, optimal placement of guidance tracks plays a major role in reducing the overall costs of processing a field. A number of possible optimization criteria can be considered when formulating the cost function at this step. For large agricultural vehicles, is it pertinent to reduce the number of turns along with the total path length, as machine constraints make turning maneuvers costly in terms of time and energy spent, and contribute to machine wear and tear. This is often done indirectly by choosing the optimal decomposition technique and computing the optimal track orientation for each decomposed cell. In general, taking into account various optimality criteria when creating guidance tracks can lead to vastly different end results compared to using rule-of-thumb guidelines or heuristics alone.

1.3 | Existing surveys

In the course of the last decades, several literature reviews dealing with the solution to CCPP problems were published. One of the early ones from 2001 is Choset (2001), where the main focus is on the description of exact and approximate decomposition techniques and multirobot coverage. Some of the mentioned basic algorithms, for example, the Boustrophedon decomposition are still being utilized in current works. The review Galceran and Carreras (2013) contains a collection of many different approaches connected to coverage path planning. It covers besides the basic decomposition algorithms also the topics of online coverage, grid-based approaches using wavefront of spanning trees, and coverage in 3D terrain. Although the review is extensive in its scope of methods presented, only one small portion is dedicated to an overview of literature dealing with optimization in CCPP. The work Khan et al. (2017) is split up into the three main parts of decomposition, optimization of the subregion sequence and path smoothing, which is similar to the methodology that is utilized in this paper. Nevertheless, all three topics are covered very briefly without algorithmic details. An in-depth review of CCPP for UAVs is provided in Cabreira et al. (2019), with methods for both exact and approximate cellular decomposition. A classification of the methods is provided according to the shape of the area of interest and different performance metrics applicable specifically to UAVs. A very thorough and recent review on route planning in CCPP is given in Tan et al. (2021), classifying the discussed methods in classical and heuristic approaches, and comparing their advantages and disadvantages. In Chakraborty et al. (2022) a review focusing explicitly on agricultural ground robots is given, where both point-to-point and coverage path planning is being tackled. Santos et al. (2020) also focuses exclusively on coverage path planning for agricultural ground robots, but for both Chakraborty et al. (2022) and Santos et al. (2020) the focus is mainly on a meta-analysis and comparison of existing literature, rather than providing details on the content of the literature itself.

1.4 | Main contributions

When dealing with a vast array of methods and evaluations for solving complete coverage path planning, it can be daunting to identify an appropriate initial approach. Therefore, we believe there is the need for a systematic, step-by-step prescription of a methodology for CCPP that can guide researchers and practitioners in their efforts to solve this problem effectively and efficiently, which

is what this manuscript aims to provide. Moreover, a structured review with emphasis on optimization-based techniques in CCPP has been missing in literature so far. Here, the focus will be primarily on optimization in *guidance track generation*, wherein the most suitable algorithms for agricultural applications are described in detail, with accompanying mathematical definitions and illustrations demonstrating the concepts explained. Although we concentrate on guidance track generation in this review, a complete picture of the three-step process is nevertheless necessary when the reader is faced with her/his first CCPP assignment. Therefore we direct the reader to references (Bochtis & Vougioukas, 2008; Hameed et al., 2011; Utamima & Djunaidy, 2021; Utamima et al., 2022, 2018) to get an overview on route planning techniques, and to (Backman et al., 2015; Fraichard & Scheuer, 2004; Höffmann, Patel, et al., 2022; Rapoport et al., 2021; Ravankar et al., 2018) to tackle the path smoothing step.

The key contributions of this work can be summarized as follows:

- Comprehensive and pedagogical approach for generating guidance tracks for arbitrarily-shaped 2D ROIs.
- Introductory guide for individuals new to the field, providing them with a comprehensive understanding of the key concepts and approaches involved.
- Overview and detailed elaboration on different exact cellular decomposition techniques in literature.
- Selection of cost functions for the approaches presented in this work to evaluate the optimality.
- Presentation of advantages and limitations of the presented decomposition techniques and cost functions.

The paper is organized as follows: In Section 2, methods for generating guidance track for headlands are briefly elaborated on. Next, the more complex interior guidance tracks are considered, for which a combination of ROI decomposition and optimal track alignment is the most common approach. For this purpose, Section 3 initially considers only convex polygons. Optimal alignment of the tracks is the crucial factor in this case, for which several optimization criteria are reviewed in Section 4. In Section 5, the focus is on optimal decomposition of a complex ROI into simple subpolygons, such that findings from the previous sections are directly applicable to the decomposed regions. Up to this point, only straight interior guidance tracks have been considered. In contrast, Section 6 discusses the use of curved tracks. In Section 7 the different decomposition techniques as well as different optimization criteria are compared and an introductory guide for individuals new to the field is provided. Finally, in Section 8 conclusions and future directions are stated.

2 | HEADLAND GUIDANCE TRACKS

The headland area is mainly used for turning maneuvers going from one interior track to another. One or more headland tracks are generated inside and along the polygonal boundaries of $\mathcal{R}$, as shown in red in Figure 3. There is widespread agreement in the literature on the determination of these tracks, with the majority of attention directed towards the more complex interior guidance tracks as opposed to headland tracks. Examples where the latter are explicitly adressed are Hameed et al. (2010) and Parsons et al. (2023). The number of headland tracks is determined by the ratio of the machine's minimum turning radius $r_{\min}$ and its working width w. To ensure enough space for turning maneuvers and transitioning between two interior tracks, the number of headland tracks must be higher than the ratio $r_{\min}/w$ (Parsons et al., 2023).

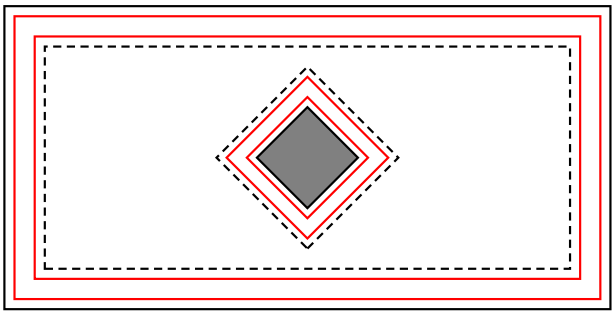

FIGURE 3 ROI with rectangular outer boundary and one obstacle (gray), two headland tracks (red) along the boundaries of the ROI, and the resulting inner boundaries (dashed). ROI, region of interest.

By determining parallels to the individual segments of $\mathcal{R}$ and computing the intersection points of adjacent segments, headland tracks are generated (Hameed et al., 2010). Thereby, the outermost track shall have a distance of $w/2$ from the boundary of $\mathcal{R}$, and between two neighboring headland tracks there is a distance of w. At a distance of $w/2$ to the innermost headland track, an internal boundary can be generated, the *inner field boundary*. This defines the area where the interior tracks will need to be placed. This can be seen in Figure 3 as the dashed line. As a note, for example, in Parsons et al. (2023) the headland area runs not only along the field and obstacle boundaries, but also along the boundaries of subpolygons that are created during the decomposition of complex fields, see Section 5. In this case, the generation of the headland tracks follows the same approach as described above, but results in a significantly enlarged overall headland area.

This straightforward approach to compute headland tracks can result in singularities, which is visualized in Figure 4. To counteract this problem, a singularity filter was described in Hameed et al. (2010). For each new track, it is checked whether two segments of this track intersect. If this is the case, the vertices of all in-between segments are replaced by the intersection point, visualized in Figure 5a. Furthermore, to avoid irregularities, for example minimal concavities in the path that cannot be tracked with the machine, an irregularity filter is described in Hameed et al. (2010). It is checked if the (virtual) intersection of two segments lies on one of these two segments, and thereby if the distance to the endpoint of the other segment is below a minimum threshold ϑ, all vertices of the in-between segments are replaced by this intersection, see Figure 5b. In

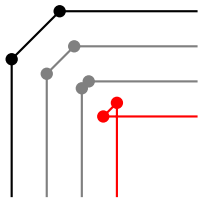

FIGURE 4 Boundary (black) and resulting parallel headland tracks (gray and red) using the straightforward way of drawing parallel line segments and connecting the corresponding intersection points. This results in the "swallowtail" singularity (red), where the resulting track is self-intersecting.

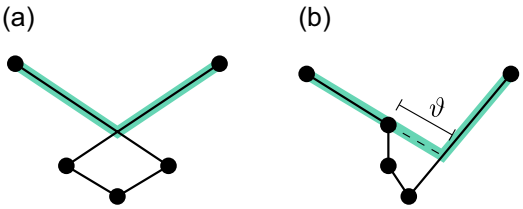

FIGURE 5 Filters applied in Hameed et al. (2010) to dissolve irregularities in headland guidance tracks, with initial tracks (black) and filtered tracks (green). (a) Singularity filter and (b) irregularity filter with threshold ϑ.

Hameed et al. (2010), they select a threshold of $\vartheta = w/2$, to secure the coverage of the area corresponding to the deleted points.

It must be noted that the nonholonomic vehicle cannot traverse the sharp corners of these headland tracks. In a subsequent step, the already mentioned path smoothing, the tracks need to be smoothed, see for example, Höffmann, Patel, et al. (2022), where an approach utilizing NURBS curves with optimal weights has been developed.

3 | GUIDANCE TRACKS FOR CONVEX AREA

Different strategies for covering a simple convex polygon using interior tracks based on geometric patterns can be found in literature (Cabreira et al., 2019). The two most commonly used patterns are back-and-forth and spiral, see Figure 6. The back-and-forth pattern covers the polygon with parallel tracks, while the spiral pattern starts along the polygon boundary and spirals inward. The spiral pattern is a widely accepted method within the field of CCPP for UAVs, as demonstrated in Cabreira et al. (2019, 2018). Two different spiral covering approaches for a car-like underactuated vehicle in unstructured environment are presented in Bosse et al. (2007), where also curvature constraints are considered.

In the context of agriculture, particularly when utilizing large nonholonomic vehicles, the back-and-forth pattern has been extensively tested and proven to be a reliable standard approach. The free parameter for this pattern is the angle θ of the parallel guidance

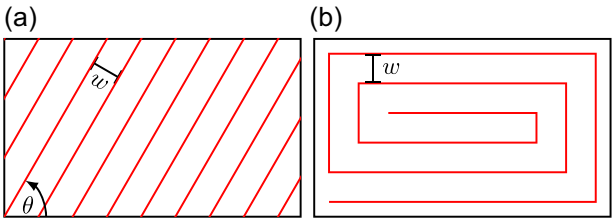

FIGURE 6 The two most common geometric patterns (red) to fill a convex area, with an effective working width w. (a) Back-and-forth pattern with angle θ and (b) spiral pattern.

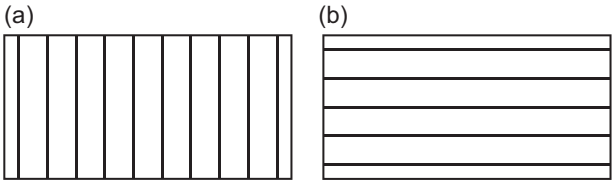

FIGURE 7 Covering a polygon with equidistant straight parallel guidance tracks with different angles θ. (a) $\theta = \frac{\pi}{2}$ and (b) $\theta = 0$.

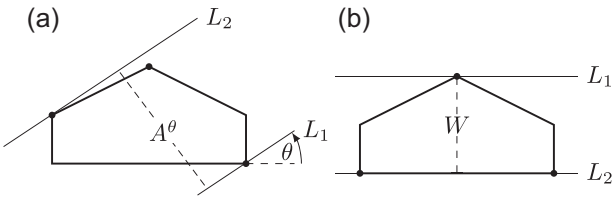

FIGURE 8 Different parallel support lines L_1, L_2 and the altitude and width of a convex polygon. (a) Altitude A^θ with $\theta = \frac{\pi}{6}$ and (b) width W with $\theta_W = 0$.

tracks, see Figure 7. To determine the distance between the parallel tracks and the number of tracks, we have to introduce the altitude of a polygon (Li et al., 2011).

Definition 1 (Polygon altitude). Given a convex polygon P and an angle θ, a *line of support* L^θ is aligned by an angle of θ and intersects P such that the interior of P lies to one side of L. The *altitude* A_P^θ is the distance between a pair of parallel support lines of P with an angle of θ.

See Figure 8a for a visualization of these entities. Usually, the altitude of a polygon is not an integer multiple of the vehicle's working width w, such that the effective working width w_P^θ for a given convex polygon P and equidistant guidance tracks aligned by an angle of θ is determined by

$$w_P^\theta = \frac{A_P^\theta}{n_P^\theta} \quad \text{with} \quad n_P^\theta = \left\lceil \frac{A_P^\theta}{w} \right\rceil, \tag{1}$$

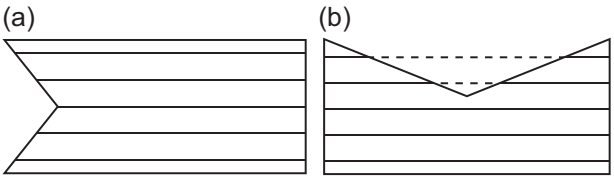

FIGURE 9 Covering different concave polygons with guidance tracks analogously to the convex approach results in uninterrupted or interrupted tracks, dependent on the polygons shape, adapted from Cabreira et al. (2019). (a) Uninterrupted and (b) interrupted.

where n_P^θ denotes the number of equidistant tracks with an angle of θ to cover a convex polygon P. In particular, this ensures that the maximum distance between two adjacent tracks is w and, as a consequence, that there will be no gaps in processing the field. Small overlaps of the swath induced by $w_P^\theta < w$ provide a more reliable result in terms of coverage in the real implementation of the coverage path.

Optimizing the angle θ of these tracks makes the processing of the field more efficient (Li et al., 2011). The efficiency can be measured in different ways, and is expressed by a cost function C. This results in the optimization problem

$$\begin{aligned} \min_{\theta\in[0,\pi)} \quad & C(\theta; P) \\ \text{s.t.} \quad & \mathcal{T}(\theta) \text{ covers } P \end{aligned} \tag{2}$$

where P is the convex polygon to be covered, and $\mathcal{T}(\theta)$ is the set of parallel equidistant guidance tracks aligned by θ.

This procedure is applicable to nonconvex polygons as well. However, it must be noted that the set of feasible directions is restricted or even empty. A direction is called feasible in this context, when the resulting tracks are not interrupted, as visualized in Figure 9. For a special class of non-convex polygons, the monotone polygons, there exists at least one feasible direction.

Definition 2 (Monotone polygon). A polygon P is called *monotone w.r.t. direction* α, if every line orthogonal to α intersections the boundary of P at most twice. Furthermore, P is called *monotone*, if there exists a direction α, such that P is monotone w.r.t. α.

Before delving into the discussion on covering arbitrarily shaped ROI, the subsequent section will present a concise overview of various optimization criteria employed to enhance the direction of the guidance tracks.

4 | COST FUNCTIONS

In the literature, different ways to measure the efficiency of the guidance tracks are used. In the following section, a selection of the most commonly used ones is introduced.

4.1 | Minimal number of turning maneuvers

One widely used performance criterion for coverage paths is the number of turning maneuvers performed by agricultural ground vehicles. These maneuvers require a reduction in speed and can result in additional forces on the tracks or tires, as well as increased stress on the soil. Consequently, time and energy costs increase with the number of turning maneuvers. As such, research often seeks to minimize the number of maneuvers as a means of indirectly optimizing for operation time, conserving energy and extending fuel use (Cabreira et al., 2019). A large part of the literature uses the criterion of minimizing the number of parallel tracks, and thus the number of turning maneuvers (Cabreira et al., 2019; Huang, 2001; Li et al., 2011; Torres et al., 2016; Yu & Hung, 2015). The costs to cover a convex polygon P with guidance tracks with an angle θ are then defined by

$$C(\theta; P) = n_P^\theta. \tag{3}$$

For a convex polygon, it is possible to determine the direction of the tracks so that these costs are minimized. To do this, the width of the convex polygon has to be determined (Li et al., 2011).

Definition 3 (Polygon width). The *width* W_P of a convex polygon P is the minimum altitude

$$W_P = A_P^{\theta_W} = \min_\theta A_P^\theta. \tag{4}$$

with the corresponding support line angle θ_W.

Once the optimal angle θ_W for a convex polygon P has been determined, aligning the guidance tracks to this angle results in $(n_P^{\theta_W} - 1)$ minimum required turning maneuvers.

In Huang (2001) it is shown that the associated support lines for the width of a convex polygon are parallel to a polygon edge, which is also visualized in Figure 8b. Due to this restriction of the search space for the optimal angle θ_W, the calculation of the width can be implemented more efficiently. In Houle and Toussaint (1985) an algorithm is presented, which determines the width of a polygon and the corresponding angle θ_W in linear time. Likewise, Li et al. (2011) shows a new approach to effectively calculate the width in linear time complexity. The concept of altitude and width can further be extended to general non-convex polygons with holes, see Figure 10. In Bochkarev and Smith (2016) an algorithm for determining the width of a general polygon with a runtime of $O(n^2 \log n)$ is described, where n is the number of polygon vertices.

4.2 | Minimal path length

A different approach for parameterizing guidance track efficiency is used by Jin and Tang (2010). To keep the total costs of the final path

as low as possible, particularly complex maneuvers should be avoided in addition to minimizing the number of turning maneuvers. This results in a cost function

$$C(\theta; P) = \sum_{i=1}^{n_p^\theta - 1} l_i(\theta), \tag{5}$$

where $l_i(\theta)$ is the path length of the ith turning maneuver considering the guidance track angle θ. Especially in situations where the angle ψ between the interior and headland tracks deviates considerably from being orthogonal, the required path length increases significantly, which is visualized in Figure 11. In addition to that, dependent on this angle, the minimum turning radius of the machine, the working width, and the available headland area, different turn types are preferred. Especially in the agricultural context with the huge machines and big turning radii, different types of maneuvers are being performed. In Jin and Tang (2010) detailed description of the five most common turning maneuvers in the agricultural context are presented and analytical formulas for determining their path lengths are derived. In Figure 12, the considered turning maneuvers are sketched. Based on the path lengths, a decision tree is developed, which outputs the optimal turn type for each turning situation from one track to the next. With this decision tree, the overall path length can be calculated.

This approach assumes that the tracks are run sequentially, such that the field is processed from one side to the other. This approach may not be the optimal solution for fields of complex shapes with internal or even substantially curved boundaries (Rapoport et al., 2021). The problem of the optimal order of guidance tracks is another topic, the aforementioned route planning, that is not covered in the scope of this paper. In Utamima and Djunaidy (2021) an extensive review about different approaches to solve the route planning problem is given.

4.3 | Minimal overlap area with headland

A similar approach to Section 4.2, but with a different motivation, is taken in Hameed et al. (2011). Here, it is not the path length that is to be minimized, but overlapping areas, that is, areas that have been traversed several times. If areas in agriculture are processed too often, this can lead to overfertilization, increased use of chemicals, or overwatering, which has both ecological and economic disadvantages. The goal is to align the interior tracks such that the overlapping area with the headland tracks is minimized. If interior and headland tracks meet perpendicularly, this overlapping area is zero, see Figure 11a. If this is not the case, areas are created which are processed more than once, see Figure 11b,c. The following cost function is thus considered

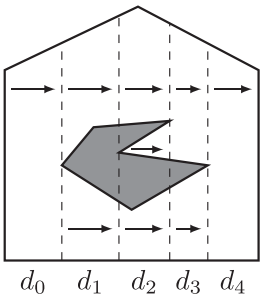

FIGURE 10 Visualization of the computation of the altitude of a general polygon by adding up the altitudes of the individual cells: $A_2^\pi = d_0 + 2d_1 + 3d_2 + 2d_3 + d_4$, adapted from (Bochkarev & Smith, 2016).

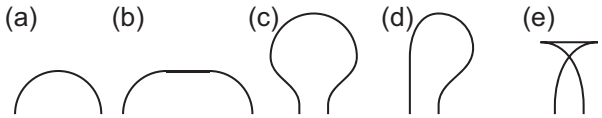

FIGURE 12 Selection of the most common agricultural turn types, traversing in the headland area from one interior guidance track to the other. (a) U, (b) Flat, (c) Omega, (d) Hook, and (e) Fishtail.

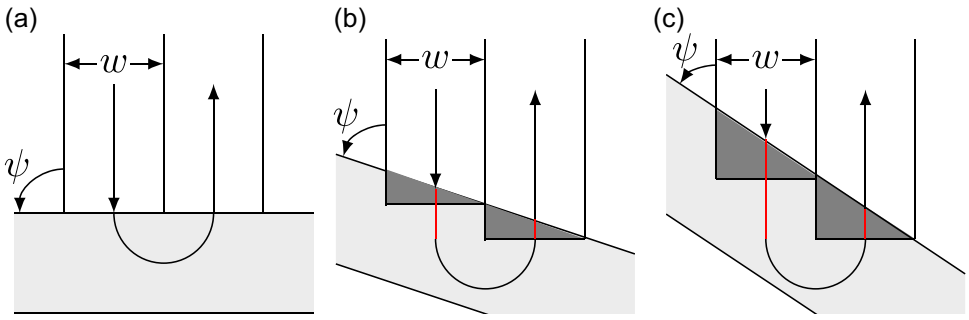

FIGURE 11 Comparison of turning maneuvers at different angles ψ between headland and interior tracks, resulting in increased path length (red) and overlap areas (dark gray), adapted from Jin and Tang (2010). (a) $\psi = \frac{\pi}{2}$, (b) $\psi = \frac{\pi}{2}$, (c) $\psi = \frac{\pi}{2}$.

$$C(\theta; P) = \sum_{i=1}^{n_P^\theta - 1} O_i(\theta), \tag{6}$$

where $O_i(\theta)$ denotes the overlapping area of the ith interior track with the upper and lower headland.

5 | GUIDANCE TRACKS FOR NONCONVEX AREA

When dealing with concave polygons without holes (Torres et al., 2016) gives a first step to check if the concavity is a constraint at all and a decomposition is needed. Similar to the convex polygon, the optimal direction of the guidance tracks is determined, as described in Section 3. Subsequently, it is checked whether the resulting guidance tracks are interrupted, as visualized in Figure 9. If this is the case, a decomposition is necessary. The ROI is decomposed into a set of subpolygons with a simple structure, for example convex or monotone. In a subsequent step, the interior guidance tracks for each of these subpolygons are generated, following the procedure from Section 3.

When utilizing a single machine to execute a task on a field, it is necessary to subsequently examine the order of the subfields in the route planning phase (Khan et al., 2017). Furthermore, when multiple machines are taken into consideration, the division of the area can for example be utilized to assign designated areas to each machine.

5.1 | Cellular decomposition

Cellular decomposition algorithms are widely used and have been in application for solving CPP problems for a long time (Latombe, 1991). The approach is based on the divison of the region into nonoverlapping *cells*. In Choset (2001), these algorithms are classified into *approximate*, *semiapproximate* and *exact* decompositions.

Approximate decompositions have been introduced initially in Elfes (1987) and Moravec and Elfes (1985) and are based on a discrete grid with fixed resolution, which is overlaid on the area. Each cell of the grid that does not belong to the area (e.g., outside the outer boundaries or inside an obstacle) is marked as blocked. The non-blocked cells then form an approximated decomposition of the area. This is visualized in Figure 13a, where the white cells depict the decomposition. In addition, approaches exist to further improve the accuracy of this decomposition. Here, the cells that only partially belong to the ROI are recursively reduced in size, for example in quadtree decomposition (Yijun et al., 2021). A rough approximation of the ROI does not require much computational effort. The higher the accuracy of the grid, the higher the accuracy of the decomposition, but also the higher the computational cost. For ROIs defined by a high number of vertices, this approach does not scale well. Especially in CCPP problems involving small robots, for example, UAVs or autonomous vacuum cleaner, this decomposition approach is often followed (Nam et al., 2016). Thereby, the size of the grid cells corresponds to the working width of the robot. The goal of the CCPP problem is then to determine a path so that each cell has been visited, for example, using a graph-based search. To name one example, in Ghaddar et al. (2020) a CCPP problem for UAVs with nonflying zones is considered, where the area is divided using a grid-cell approach into cells of equal size, that matches the resolution of the UAV footprint.

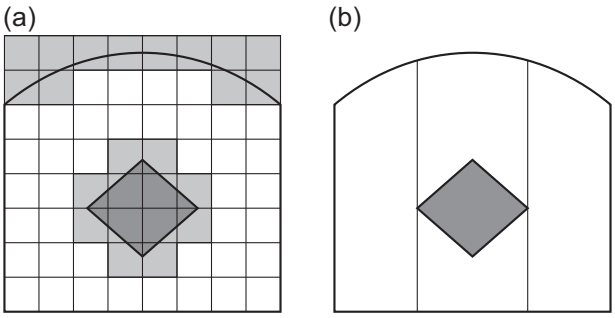

FIGURE 13 The two prominent classes of cellular decomposition approaches, introduced by Choset (2001). (a) Approximatea and (b) exact.

In contrast, exact cellular decomposition yields a set of disjoint cells $\mathcal{D} \subset \mathbb{P}$, such that

$$\mathcal{R} = \bigcup_{p \in \mathcal{D}} p, \tag{7}$$

so the decomposition gives a precise and complete representation of $\mathcal{R}$. Typically, the decomposition technique is chosen such that each of these cells can be covered using a simple back-and-forth motion. Most-used examples are convex or monotone cells. In Figure 13b, the exact decomposition results in four monotone cells. This decomposition class has been followed in many works (Bochkarev & Smith, 2016; Nielsen et al., 2019; Parsons et al., 2023), especially in the context of large robots and machines that are less agile and flexible to control than UAVs or vacuum cleaners. In particular, with respect to the methodology we will follow in this work, exact cell decomposition is found to be more appropriate compared to the approximate decomposition. Therefore, we will explain this class of decomposition in more detail in the following sections.

5.2 | Minimum cell decomposition

From a coverage perspective, it can be beneficial to decompose the ROI into as few cells as possible. This thought leads to the technique of minimum cell decomposition, to decompose a polygonal area $\mathcal{R}$ into a minimum number of cells

$$\begin{aligned} &\min \ |\mathcal{D}| \\ &\text{s.t.} \ \ \mathcal{R} = \bigcup_{p \in \mathcal{D}} p, \end{aligned} \tag{8}$$

where the cells in $\mathcal{D}$ are usually simple convex polygons. The contruction of such a minimum cell decomposition is a problem that is well-known and examined in history and is not particularly designed for coverage path planning purposes, but for example, for point-to-point path planning, to generate a feasible path in a complex region (Jan et al., 2012).

In Chazelle and Dobkin (1985) an algorithm for decomposing a nonconvex simple polygon into a minimum number of convex polygons is proposed. This algorithm utilizes Steiner points to resolve concave vertices. To accomplish this resolution, patterns are introduced to connect concave vertices without creating new concave angles. This allowed for the polygon to be decomposed into a minimal number of convex polygons. In contrast, Keil (1985) presents a method for decomposing a polygon into simpler components without the usage of Steiner points. The considered simpler components can be convex, spiral, star-shaped or monotone. The proposed method was demonstrated to be more efficient than algorithms based on Steiner points.

Another prominent example for a well-known convex decomposition algorithm is the Delauny triangulation (Chazelle, 1985; Chen & Xu, 2004; Jan et al., 2012). It is useful for creating a mesh of triangles that can be used for many applications, such as creating a finite element mesh, interpolating scattered data (Rippa, 1990) and creating Voronoi diagrams (Fortune, 1995). The triangulation is created by connecting each point to its nearest neighbors, thus forming a triangle. This techniqe maximizes the smallest angle of all resulting triangles.

In Schachter (1978), a convex decomposition technique based on a Delaunay triangulation of the polygon is proposed and implemented as a divide-and-conquer approach. Specifically, the algorithm only considers the Delaunay edges starting from a concave vertex of the polygon, choosing only the minimum number of edges necessary to obtain a convex decomposition.

However, in cases where the ROI consists of a boundary polygon together with several obstacle polygons, these algorithms are not applicable. It is possible that obstacle boundaries will be crossed during the decomposition, such that a resulting cell will be partly in the ROI and partly on the obstacle. There exist several extensions, for example, the constrained Delauny triangulation, where polygonal areas with obstacles are taken into account (Chew, 1987; Fernández et al., 2008; Fortune, 1995). Despite this advancement, the problem of decomposing a polygonal area with holes into the minimum number of convex cells is known to be NP-hard (Keil, 1985). Furthermore, a broad overview about polygon decomposition algorithms is given by Keil (2000).

Even if these approaches are standard ways to determine a decomposition of the ROI, the objective of minimum number of simple cells may not always lead to a desired and optimal result. This will be discussed in the next section, for example, visualized in Figure 18b. Optimization criteria to measure efficiency like the ones discussed in Section 4 should be taken into account to achieve an optimal result. In the following sections, such optimal decomposition techniques are examined.

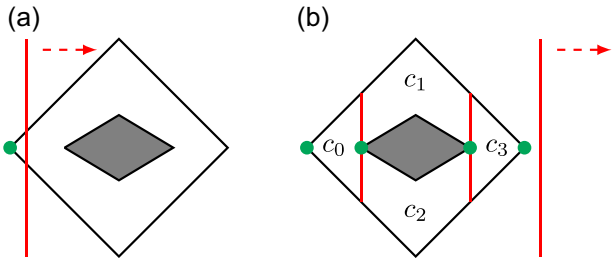

FIGURE 14 Sweepline based decomposition: Sweepline (red) sweeps (here from left to right) over region with one obstacle (gray), detects events (green) and decomposes the region with lines (red) parallel to sweepline. (a) Initial region and (b) decomposition into four cells.

5.3 | Sweepline based decompositions

A wide-spread approach for exact cellular decomposition, especially in the context of coverage path planning, is based on a line sweep algorithm, where a *sweepline* sweeps over $\mathcal{R}$. During this, *events* are detected, and lines parallel to the sweepline are inserted at these events to decompose the region, as visualized in Figure 14.

5.3.1 | Trapezoidal decomposition

The classic and most simple sweepline-based decomposition technique is trapezoidal decomposition, where the resulting cells are trapezoids, as shown in Figure 15a. There may also be triangles that can be considered as degenerate trapezoids, where one of the parallel sides has a length of zero (Choset et al., 2005).

During the linesweep, five different event types *Open*, *Split*, *Merge*, *Close*, and *Inflection* are detected, which are visualized in Figure 16. The Open and Close events open and close the corresponding cell, respectively. The Split event divides a cell into two new cells, while the Merge event combines two cells into one new cell. The Inflection event closes one cell and opens a new one. The resulting decomposition significantly depends on the polygonal representation of the region, as all vertices of the polygonal ROI are treated as events. A detailed description and discussion of this algorithm can be found in Choset et al. (2005) and Seidel (1991) gives a simple and fast way to implement the algorithm. In Kim et al. (2014), a CCPP algorithm for mining robots based on trapezoidal decomposition is applied. In Jimenez et al. (2007) a genetic algorithm is utilized to achieve optimal coverage, which is being combined with the decomposition of the ROI into subregions using the trapezoidal decomposition.

This algorithm has one main drawback: It may result in numerous and narrow cells which is inefficient from a coverage perspective (Choset et al., 2005). In Figure 15a this drawback can be observed. To counteract this, there exist subsequent merging approaches that can be applied. Two cells can be merged if they are entirely adjacent on

one of their two parallel sides (Li et al., 2011; Oksanen & Visala, 2009).

Definition 4 (Entire adjacency). Two polygons P_1 and P_2 are called *adjacent*, if one edge of P_1 coincides with an edge of P_2. The polygons are called *entirely adjacent*, if an edge of P_1 is equal to an edge of P_2, see Figure 17.

In Figure 18 two different scenarios are visualized, to show that merging all entirely adjacent cells is not always beneficial. In Li et al. (2011) another merge condition is introduced, for which the desired track directions for each convex cell are independently calculated, following Section 3. If the guiding tracks for entirely adjacent cells are parallel, these cells are merged together.

After this merging step, the cells are not convex anymore, but monotone polygons w.r.t. the sweep direction. This limits the choice of available directions for the guidance tracks. Due to the monotony, guidance tracks parallel to the sweepline are always feasible.

5.3.2 | Boustrophedon decomposition

Another prominent sweepline-based decomposition technique is the Boustrophedon (literally means "turning like oxen in plowing") Decomposition (Choset, 2000), see Figure 15b. The main idea is to generate cells, that can be covered with a back-and-forth motion. This results in a decomposition consisting of monotone cells w.r.t. the sweep direction, and is equal to the approach of trapezoidal decomposition combined with a subsequent merging of entirely adjacent cells.

In contrast to the trapezoidal decomposition, vertices of the ROI that represent an Inflection event are not considered in this type of decomposition. This results in a rather small number of cells, which is clearly visible in Figure 15b, and each cell is a monotone polygon w.r.t. the sweep direction. The monotony ensures that each cell can individually be covered by a simple back-and-forth motion, with the tracks parallel to the sweepline. This decomposition technique was first introduced in Choset (2000), followed by an extension described

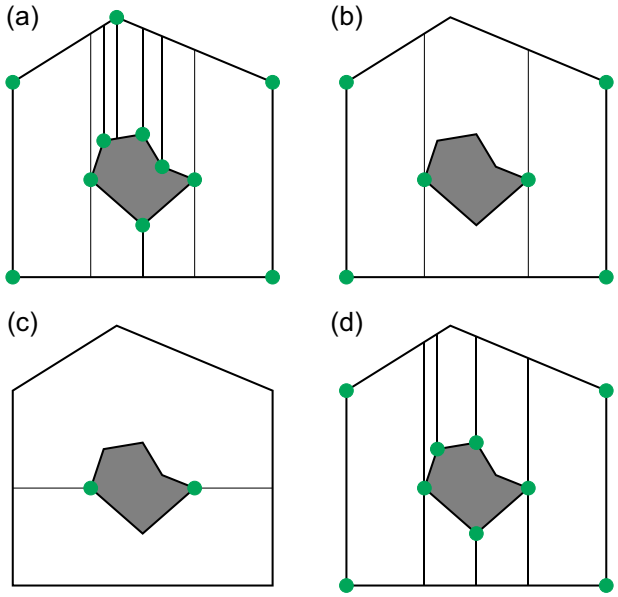

FIGURE 15 Different sweepline based decomposition techniques with the considered events (green), assuming a vertical sweepline. (a) Trapezoidal (Choset et al., 2005), (b) Boustrophedon (Choset et al., 2000), (c) Alternative Boustrophedon decomposition perpendicular to sweepline direction (Pham et al., 2017), (d) Boustrophedon including concave inflections (Yu & Hung, 2015).

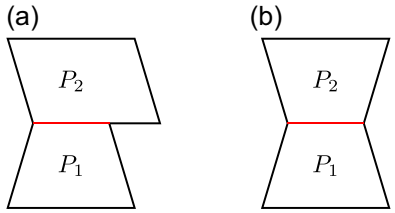

FIGURE 17 (Entire) adjacency of two polygons P_1 and P_2. (a) Adjacent: Edge of P_1 (red) coincides with an edge of P_2. (b) Entirely adjacent: Edge of P_1 (red) equals an edge of P_2.

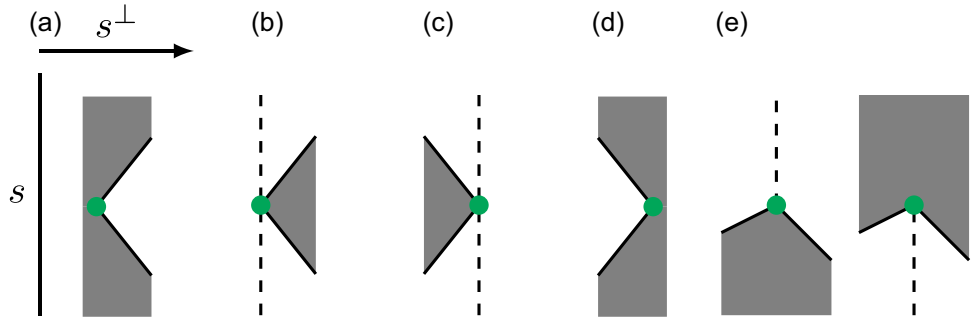

FIGURE 16 Excerpts of an ROI (white: included in ROI, gray: excluded in ROI), different event types (green) and the resulting decomposition line (dashed). (a) Open (b) split (c) merge (d) close (e) inflection. ROI, region of interest.

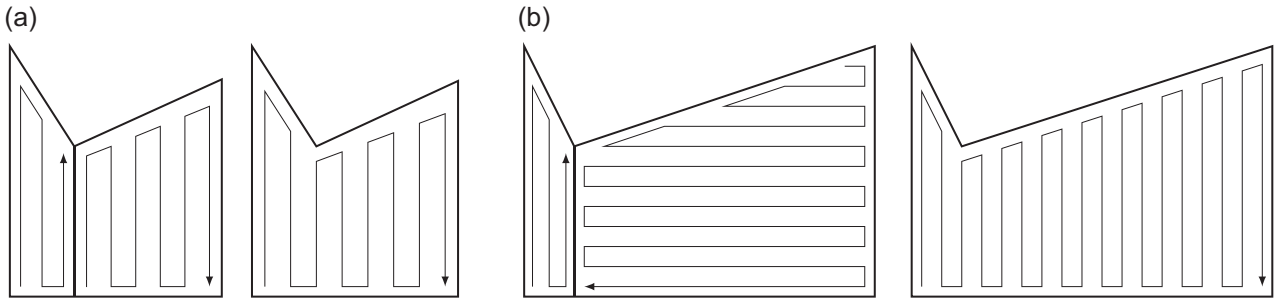

FIGURE 18 Examplary impacts of merging entirely adjacent cells on the number of tracks (left: initial, right: after merge). (a) Cells with the same track direction: Decrease in number of tracks (9 → 8). (b) Cells with different track directions: Increase in number of tracks (15 → 18).

in Huang (2001), where multiple events can occur simultaneously and an event can also consist of a sequence of consecutive vertices.

This decomposition approach is a very basic and frequently used technique in different application fields. In Rekleitis et al. (2008) it is applied in the context of multirobot CCPP, whereas (Liu et al., 2019) utilizes a Boustrophedon decomposition to perform efficient point-to-point path planning. Further examples are the application in the field of CCPP for inspection with UAVs (Pérez-González et al., 2021), and for harbor seabed surveys with AUVs (Fang & Anstee, 2010). In Mannadiar and Rekleitis (2010) a CCPP algorithm is proposed to achieve minimal path length based on the Boustrophedon decomposition.

The authors in Pham et al. (2017) propose an alternative Boustrophedon approach, where you decompose the area perpendicular to the sweepline, instead of parallel, which is visualized in Figure 15c. For this approach, only the event types Merge and Split are considered, the events Open and Close do not trigger a decomposition in that point. It is shown that this adaption leads to a decrease of the number of cells after decomposition, and that it works well also with several concave obstacles in the area.

In Yu and Hung (2015) a sweepline decomposition that results in convex cells is presented. This technique combines the trapezoidal and Boustrophedon decomposition to balance out the disadvantages of both. It is based on the idea of differentiating concave and convex Inflection events. The decomposition then only occurs at concave vertices, as visualized in Figure 15d. This results on the one hand in a reduced number of cells compared to trapezoidal decomposition, but all cells are convex, in contrast to the Boustrophedon decomposition.

5.3.3 | Morse decomposition

The two previous approaches are both restricted to polygonal regions. The *Morse* decomposition can be seen as a generalization of the Boustrophedon decompositon to smooth boundaries. This approach uses ideas from Canny (1988), where a so-called *slicing method* was first applied to motion planning (Choset et al., 2005). To introduce this approach, we will first define the concept of line sweeping in more mathematical terms.

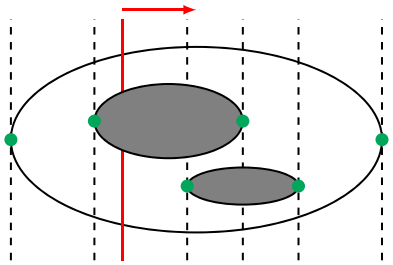

FIGURE 19 Critical points (green) and connectivity intervals (dashed lines) of sweepline with $h(x_1, x_2) = x_1$. Connectivity of current sweepline (red) is 2.

Definition 5 (Sweepline). A *sweepline* is the preimage of a real-valued function $h : \mathcal{R} \to \mathbb{R}$, that is,

$$\mathcal{R}_\lambda = \{x \in \mathcal{R} | h(x) = \lambda\}, \lambda \in \mathbb{R}. \tag{9}$$

The variation of λ will sweep the line through the region.

Since the smooth region is not defined by vertices, another approach to identify events is needed. The general idea of Morse decomposition is to discover connectivity changes of the sweepline, for example, splitting it up into smaller pieces when first encountering an obstacle. An increase in connectivity is related to an Open or Split event, such that new cells are created. A decrease is related to Close or Merge events, such that multiple cells are completed and a single cell is created or no action is taken. In Figure 19 an example for these connectivity changes can be seen.

It is well known from analysis that a function takes its extreme values where the derivatives vanish, or at the boundary of the domain.

Definition 6 (Critical point). Let $h : \mathbb{R}^2 \to \mathbb{R}$ be a real-valued differentiable function. A point $p \in \mathbb{R}^2$ is called a *critical point* of h, if it holds

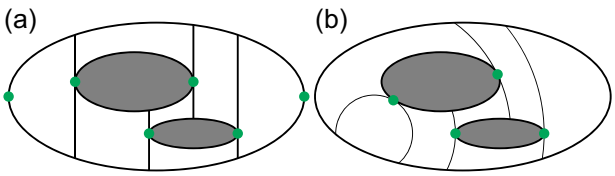

FIGURE 20 Critical points (green) and Morse decomposition for different sweep functions h. (a) $h(x_1, x_2) = (x_1)$, (b) $h(x_1, x_2) = (x_1^2 + x_2^2)$.

$$\frac{\partial h}{\partial x_i}(p) = 0, \text{ for all } i \in \{1, 2\}. \tag{10}$$

A critical point p is *nondegenerate*, if and only if its Hessian is nonsingular.

To decompose $\mathcal{R}$, the critical points of $h|_{\partial\mathcal{R}}$, that is, the function h restricted to the boundary $\partial\mathcal{R}$, are sought. To determine these, Acar et al. (2002) shows that a critical point is located where the sweepline runs tangentially to $\partial\mathcal{R}$. This fact is also evident in Figure 19.

Definition 7 (Morse function). If all critical points of a function are nondegenerate, the function is a *Morse function* (Milnor, 1963).

In other terms, this means that the critical points are isolated. Under the assumption that $\partial\mathcal{R}$ has a structure that ensures $h|_{\partial\mathcal{R}}$ to be a Morse function, in Acar et al. (2002) it is shown that the connectivity of the sweepline does change if and only if it passes through a critical point of $h|_{\partial\mathcal{R}}$. Due to the absence of obstacles between critical points, they can be used to decompose the region into cells, such that each one can be covered with simple back-and-forth motion, see Figure 20a. As a remark, the authors (Fomenko & Kunii, 1997) prove that it is always possible to find such a Morse function, by just perturbing $\mathcal{R}$ by a small amount.

Choosing different functions h yields different sweepline shapes and therefore different decompositions. In the above examples we used a vertical straight line as a sweepline, which also emits the aforementioned Boustrophedon decomposition. The result can be seen in Figure 20a. This is realized by choosing the sweep function

$$h(x_1, x_2) = x_1.$$

Another example is

$$h(x_1, x_2) = x_1^2 + x_2^2,$$

with which the sweepline is a circle and the sweeping is realized by increasing the radius, visualized in Figure 20b. With this, a spiral coverage pattern can be generated (Galceran & Carreras, 2013).

In addition to the definition over smooth regions, Acar et al. (2002) extends the Morse decomposition to nonsmooth boundaries using generalized gradients, which is briefly visualized in Figure 21. This extension is applicable if the function h is lipschitz continuous

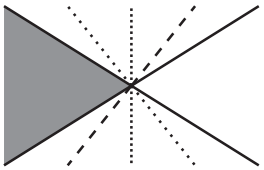

FIGURE 21 The generalized gradient is the set of all vectors enclosed by the adjacent surface gradients (solid lines). Three elements of the generalized gradient (dashed/dotted) in the nonsmooth point.

(Degiovanni, 1997). Furthermore, according to the standard definition of Morse decomposition, the critical points must be nondegenerate (Milnor, 1963). However, in Butler et al. (1999) a coverage algorithm is presented which includes the decomposition of rectilinear obstacles with degenerate critical points.

5.3.4 | Optimal sweepline direction

The presented algorithms for decomposing a polygon using a sweepline-based approach assume a fixed sweepline direction, which can be optimized. A heuristic and naive, but often sufficiently good and fast to determine approach is to choose the sweepline parallel to the longest polygon side (Hameed et al., 2010; Li et al., 2011).

Optimizing the angle of the sweepline to minimize the cost of the decomposition is more time-consuming, but can result in significantly more efficient guidance tracks. To measure the efficiency of the resulting guidance tracks, an approach based on the costs shown in Section 3 can be used. The overall costs of the decomposition are calculated by the sum of the costs for the individual cells.

There exist two main approaches for this cost calculation: The guidance tracks in each cell of the decomposition are aligned

(1) parallel to the sweepline, or
(2) optimal w.r.t. the cost function.

The first approach is fast and efficient to implement, but can result in rather nonoptimal tracks, visualized in Figure 22a. Aligning the guidance tracks parallel to the sweepline is by definition of the sweepline approach feasible for all resulting cells, such that no additional computational effort is required to check for feasibility in each cell. Especially in agriculture, this approach can be beneficial to obtain a particularly uniform processing of the entire field. The optimal angle θ^* of the sweepline is therefore the solution of the following optimization problem

$$\min_{\theta \in [0,\pi)} \sum_{p \in \mathcal{D}(\theta)} C(\theta; p), \tag{11}$$

with $\mathcal{D}(\theta)$ denoting the set of resulting cells of the sweepline decomposition with sweepline angle θ.

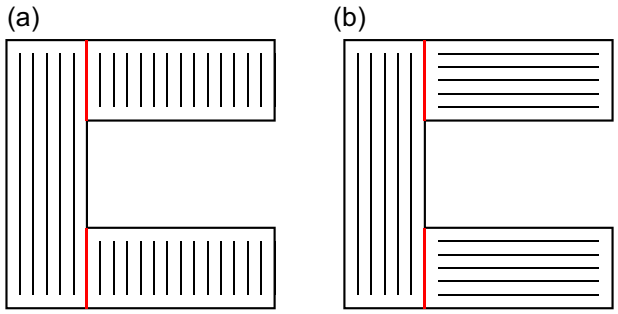

FIGURE 22 Decomposition of C-shaped region by vertical sweepline decomposition (red), and assigned track directions for each cell. (a) Track directions in all cells parallel to the sweepline and (b) track directions in each cell optimal w.r.t. the number of turns.

In Huang (2001), convex regions with convex holes are considered first. Starting from a sweepline-based decomposition algorithm, the optimal sweepline direction is determined by considering the above optimization problem, using the subregion altitudes as cost function. To do so, an analytical formula for the altitude is given and evaluated. Furthermore, it is justified that this approach can be extended to general concave regions without restrictions. For example, in Hameed, Bochtis, Sørensen, Jensen, et al. (2013) such a minimization problem is solved by an exhaustive search.

However, depending on the ROI, the efficiency could be significantly increased if the tracks in the subfields are aligned differently. In Figure 22, it can be seen that chosing the optimal track direction regarding a cost function, here the number of turns, results in an improved overall result. According to this, for each cell $p \in \mathcal{D}(\theta)$, an individual optimal driving angle θ_p^* can be determined, whereby the optimal angle is a solution of the optimization problem (2). Thus, the optimal sweepline angle is a solution of the bilevel optimization problem

$$\begin{aligned} \min_{\theta \in [0,\pi)} \quad & \textstyle\sum_{p \in \mathcal{D}(\theta)} C(\theta_p^*; p) \\ \text{s.t.} \quad & \theta_p^* \text{ solves } \begin{cases} \min\limits_{\theta_p \in \Theta_p} C(\theta_p; p), \\ \text{s.t. } \mathcal{T}(\theta_p) \text{ covers } p, \end{cases} \end{aligned} \tag{12}$$

where $\Theta_p \subset [0, \pi]$ denotes the set of feasible track direction for cell p. This bilevel approach is utilized for example in Yu and Hung (2015), where the cost function represents the width of the cell. In the case that the optimal track directions of two entirely adjacent cells are equal, these cells are merged in a subsequent step.

In Coombes et al. (2020), a greedy top-down approach is used to obtain a convex decomposition of the ROI. In this approach, trapezoidal decomposition is used, but only one cell division at an event occurs in each step, resulting in two (or more) cells. Since these cells may not be convex, the algorithm works recursively until a convex decomposition is obtained. Due to the greedyness, the algorithm generally does not result in the optimal solution, but an acceptable solution is found fast and efficiently. Thus, even for large and very concave ROIs, this approach is feasible. To determine the optimal sweep line direction, decompositions for several directions are determined in parallel and afterwards the one with the lowest costs is selected, whereby each cell gets assigned an individual track direction.

A heuristic approach for finding the optimal sweepline direction is described in Oksanen and Visala (2009). The costs are initially calculated in six equally spaced directions within the range $[0, \pi]$. The three directions with the lowest costs are selected and the others are discarded. The step size in the search direction angle is halved (in the first step, it was $\frac{\pi}{6}$). New search directions are added to either side of the three best directions, and the costs for these directions are calculated if they have not yet been evaluated. If the goal resolution is achieved, the process is terminated; otherwise, it restarts by selecting the three best directions. After five iterations, the resolution is below one degree, which has been found to be sufficient for the purpose of this approach. It is shown that the final result is comparable with a brute-force solution.

5.4 | Optimized merge options

Another common approach in the literature to determine an optimal cell decomposition is to first perform an arbitrary convex decomposition, and then merge the resulting cells optimally so that the cost of the decomposition, for example, the sum of the widths, is minimized.

5.4.1 | Initial convex decomposition

Multiple methodologies exist for achieving a proper convex decomposition. Conventional sweepline-based decompositions, such as the Trapezoidal method (refer to Section 5.3.1), do not yield the desired output as they consistently generate cells with two parallel sides. Consequently, this section outlines two distinct approaches for accomplishing a convex decomposition.

The algorithm in Nielsen et al. (2019) is based on the observation that convex decomposition tends to occur at points where the interior angle between two edges of an ROI is greater than 180°. To implement this idea, the adjacent edges of the ROI are extended until they intersect the boundary of the ROI.

Definition 8 (Edge extension). Let P be a nonconvex polygon with a concave vertex v. Denote by e_l and e_r the two adjacent edges of v. The *edge extension* in v is defined by the two line segments $c_l = \{v, w_l\}$ and $c_r = \{v, w_r\}$, that are parallel to e_r and e_l, respectively, with $c_l, c_r \subset P$ and $w_l, w_r \in \partial P \backslash \{v\}$.

As a result of this extension process, all of the created cells are convex, see Figure 23. In Figure 23a the edge extension of a single concave vertex is visualized. In Figure 23b this is applied to an exemplary region.

In Huang (2001) an algorithm is presented, which is based on executing several sweepline decompositions independently and overlaying all decompositions, combined with the idea of the

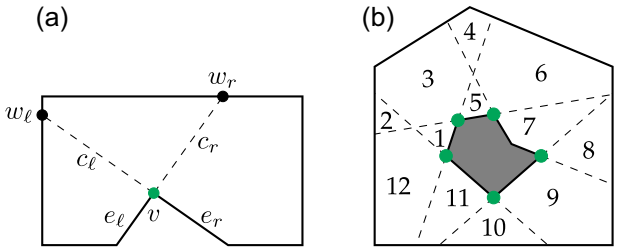

FIGURE 23 Interior edge extension (dashed) in the concave vertices (green). (a) Adjacent edges $e_{l,r}$ of v are extended by $c_{r,l}$, respectively. (b) Resulting decomposition.

previously mentioned edge extension. Thereby, the selected sweepline directions run parallel to all edges of the ROI. In addition to that, all nonconvex vertices have to be extended until a boundary is hit. The resulting cells are convex polygons.

5.4.2 | Finding optimal merge options

In a subsequent step, the individual cells should be merged together such that the sum of the costs of the resulting cells is minimized. In the course of minimization, the direction of the guidance tracks in the merged cells is also determined. There exist different approaches to tackle this optimization problem.

In Nielsen et al. (2019), all possible combinations of the initial cells that can be merged into a convex polygon are indentified, called convex merge options. Examples for convex merge options in the situation visualized in Figure 23b are (2, 3), (2, 3, 4, 5, 6), (8, 9, 10), and so on. The goal is to find the optimal set of convex merge options, such that the ROI is completely covered and the cost function, for example, the sum of cell widths, is minimized.

This problem can be framed as a *set partitioning problem*. The set of initial cells can be represented as $(\mathcal{S}_i)_{i\in\mathcal{I}}$, the set of convex merge options as $(\Omega_j)_{j\in\mathcal{J}}$. The convex merge option selection is then formulated as the following Integer Programming model

$$\begin{aligned} \min_{\lambda\in\{0,1\}^{|\Omega|}} \quad & \sum_{j\in\mathcal{J}} C(\Omega_j)\lambda_j \\ \text{s.t.} \quad & \sum_{j\in\mathcal{J}} a_{ij}\lambda_j = 1, \text{ for all } i\in\mathcal{I} \\ & \text{with } a_{ij} = \begin{cases} 1 & \text{if } \mathcal{S}_i \subseteq \Omega_j \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{13}$$

with the integer decision variables $\lambda_j \in \{0, 1\}$ for all $j \in \mathcal{J}$, that equal one if the merge option Ω_j is selected and zero otherwise. Furthermore, $C(\Omega_j)$ denotes the costs for merge option Ω_j, such that the objective function is the sum of the costs of the selected merge options. The constraint ensures that each initial cell is in exactly one selected convex merge option, to yield an actual partition.

In case of a large number of initial cells, it can take a significant amount of time to find all possible convex merge options. To address this issue, an addition has been shown in Nielsen et al. (2019) to approximate this set, which narrows down the search space. The algorithm has been tested against benchmark algorithms (Huang, 2001; Li et al., 2011) and found to perform comparably in terms of minimizing the total width of the polygons and in terms of computational efficiency in finding a solution. This decomposition technique is applied, for example, in Akshya et al. (2022) in the context of CCPP for UAVs.

Another solving technique is presented in Huang (2001), where a dynamic programming approach is used to find the optimal merge options of the individual cells. The initial decomposition is performed as described above. From this, an adjacency graph $\mathcal{G}$ can be generated, where each node symbolizes a cell, and the nodes are connected if they share a common edge. The dynamic programming approach is now based on two different options:

(1) Split the current graph into two subgraphs $\mathcal{G}_{1,2}$, resulting in two smaller subproblems.
(2) Try to cover all cells as one large cell.

Thus the minimum sum of the widths is recursively defined by

$$S_W(\mathcal{G}) = \min\left\{C(\mathcal{G}), \min_i\left\{S_W\left(\overset{i}{\mathcal{G}}_1\right) + S_W\left(\overset{i}{\mathcal{G}}_2\right)\right\}\right\}, \tag{14}$$

where i iterates over all splitting possibilities to the graph $\mathcal{G}$. Furthermore, $C(\mathcal{G})$ describes the cost of coverage when all cells belonging to nodes of $\mathcal{G}$ are considered as a single subregion. Figure 24 shows the first step and partially the second step of this decomposition. Compared to single-sweepline decompositions, this can improve the efficiency of the tracks even more, as shown in Figure 25. The results of single and multiple-sweepline decomposition together with the direction of the assigned tracks in the cells are shown. The sum of cell widths is chosen as measure for efficiency (Huang, 2001), and it is clear that the result of the multiple-sweepline approach (Figure 25e) is superior compared to the others.

This approach has exponential runtime, but is nevertheless widely-used in the literature as a benchmark solution (Coombes et al., 2020; Nielsen et al., 2019; Yu & Hung, 2015).

5.5 | Optimal decomposing cuts

A decomposition algorithm is described in Bochkarev and Smith (2016) that involves first an arbitrary convex decomposition, then using optimization techniques to further improve this decomposition. The mathematical foundation of this approach is utilizing the interior edge extension, that was discussed in Section 5.4.

Definition 9 (Cut space). Let P be a nonconvex polygon with a concave vertex v. Denote by w_l and w_r the endpoints of the edge extension of v. The *cut space* $\mathcal{CS} \subset \partial P$ is the set of all points on ∂P between w_l and w_r, that are visible from v.

The cut space typically consists of the combination of line segments, as visualized in Figure 26 (blue).

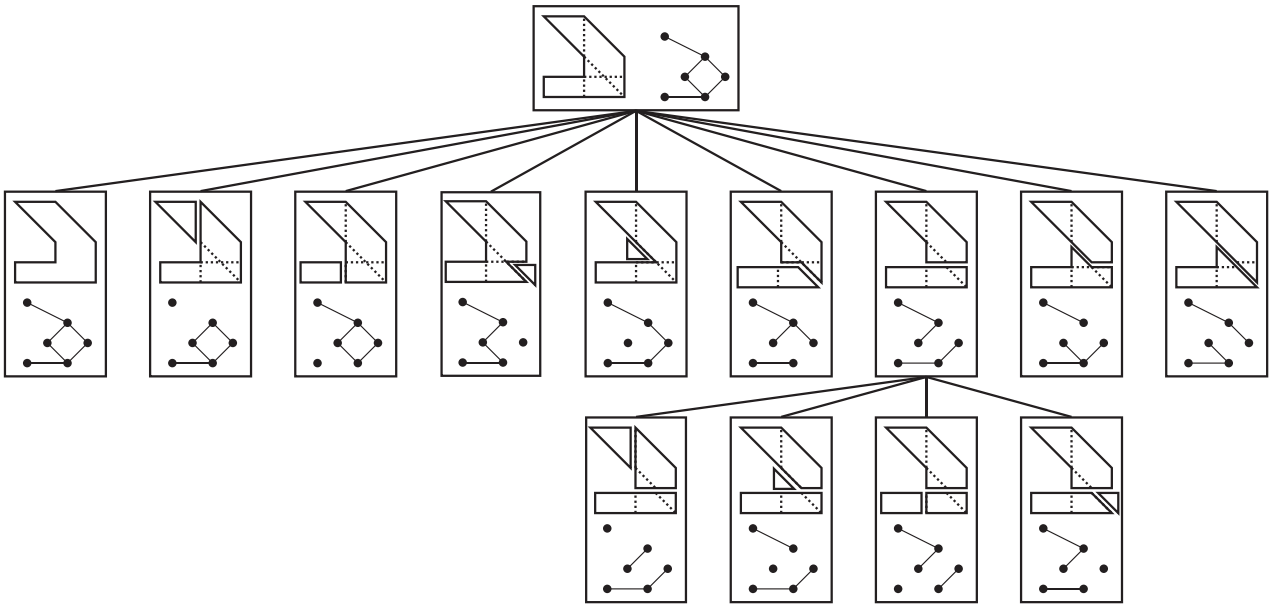

FIGURE 24 First and partly second step of dynamic programming approach of minimum sum of widths decomposition, following the approach in Huang (2001), adapted from Huang (2001).

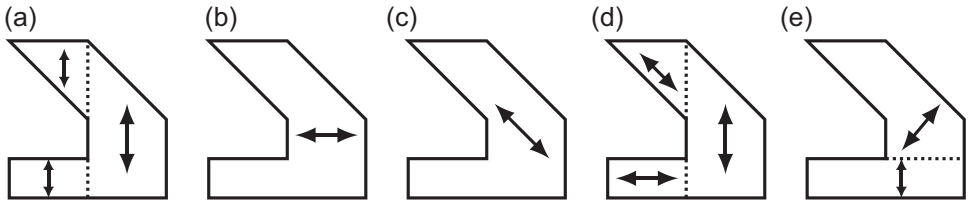

FIGURE 25 Comparison of single-sweepline and multiple-sweepline approaches, evaluated based on the sum of cell altitudes, adapted from Huang (2001). The arrows indicate the assigned track direction in the corresponding cell. (a)–(c) Single-sweepline decomposition with equal track directions parallel to the sweepline, (d) single-sweepline decomposition with optimal track directions, and (e) multiple-sweepline decomposition with optimal track directions. (a) 3, (b) 2, (c) 2.12, (d) 2.21, and (e) 1.56.

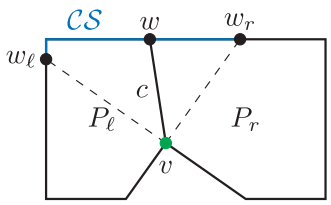

FIGURE 26 Cut space $\mathcal{CS}$ (blue) and decomposing cut c in the concave vertex v. Resulting in two polygons P_l and P_r, adapted from Bochkarev and Smith (2016).

Definition 10 (Decomposing cut). Let P be a nonconvex polygon with a concave vertex v. A *decomposing cut* is a straight line segment $c = \{v, w\}$ with $w \in \mathcal{CS}$.

A decomposing cut divides a polygon with a single concave vertex into two new convex polygons, P_l and P_r, as shown in Figure 26. The optimal decomposing cut is the cut that minimizes the sum of the costs of P_l and P_r. When using the polygon width as the cost function, it has been demonstrated in Bochkarev and Smith (2016) that an optimal decomposing cut is parallel to an edge of P_l or P_r.

The first step of the algorithm described in Bochkarev and Smith (2016) is to create an initial decomposition using any convex decomposition technique. The remainder of the algorithm seeks to improve this decomposition by decreasing the overall costs. As part of the reoptimization process, the algorithm removes previously made cuts of the convex decomposition by keeping track of concave vertices from the original region. For each vertex on this list, the algorithm searches for edges that were not originally present in the polygon. These edges are eliminated one after the other, resulting in a nonconvex cell. The algorithm then attempts to find an optimal decomposing cut from the concave vertex. This reoptimization continues, until the costs can no longer be reduced.

5.6 | Dividing lines

In Jin and Tang (2010) another algorithm with the goal of finding the optimal solution for dividing a given region and assigning guidance

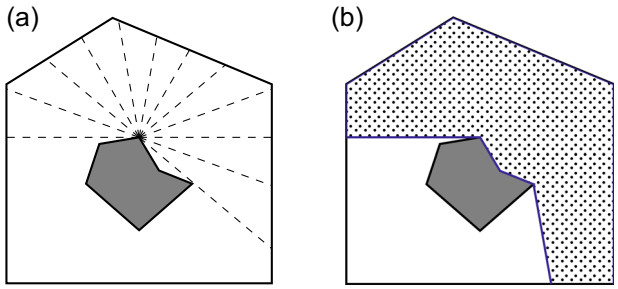

FIGURE 27 Examplary visualization of finding dividing lines for decomposition. (a) Rays from one vertex to build the undirected searching graph. (b) Example of a dividing line (blue) resulting in two cells (blank and dotted).

track directions within each subregion is presented. The search process is guided by a cost function that takes into account the cost of different types of angled turns in the headland, as shown in (5).

The base of this algorithm is the creation of an undirected graph, which is used as a tool for the subsequent search process. The graph construction is realized by creating rays from each vertex of the region, until they intersect with the boundary, visualized in Figure 27a. The precision of the resulting decomposition scheme depends on the angle between these rays. The polygon vertices of the ROI and the intersection points of the rays with the boundary represent the nodes of the graph. The initial edges of the ROI and the newly created line segments (rays) represent the graph edges. The algorithm then conducts depth-first searches in the graph to find all possible lines that divide the field into two subregions. Each search results in a dividing line made up of a sequence of edges in the graph, which in turn divides the field into two subregions, exemplary shown in Figure 27b. The algorithm uses a divide-and-conquer strategy to find the optimal decomposition for the whole region. First the optimal track direction for the ROI without any decomposition is searched and the costs are calculated. Then all possible ways of dividing the ROI into two subregions are examined and their costs calculated, by recursively applying the algorithm to each subregion. This process is repeated until all possible solutions have been thoroughly investigated. The decomposition with the lowest total cost is then chosen as the optimal solution.

5.7 | Clustering of guidance tracks

A rather different approach is followed in Hameed, Bochtis, & Sørensen (2013) and Hameed et al. (2010), which is not based on a field decomposition. Instead, it directly generates straight and parallel guidance tracks for any nonconvex ROI. If a guidance track intersects with the boundaries of the ROI (e.g., due to an obstacle), the track is split into smaller segments and the segments that are outside of the region are discarded. These segments are then organized into blocks, that can be covered together. This sorting is based on the number of splits per track. The resulting coverage of the ROI is similar to what is

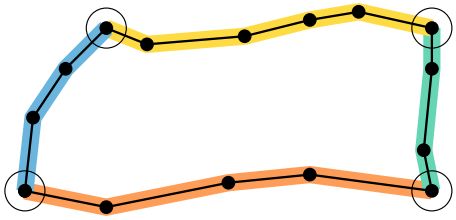

FIGURE 28 Example for selection of the longest curved edge, with four splitting points (circle) and four curved edges inbetween, adapted from Hameed et al. (2010).

achieved using the Boustrophedon decomposition, combined with guidance tracks that are parallel to the sweepline direction. The same author shows in Hameed (2017) a slightly different approach, where the sorting of the splitted tracks is being done with a *k*-means clustering algorithm.

6 | CURVED GUIDANCE TRACKS

Straight guidance tracks are often used in literature due to their simplicity and effectiveness for polygonal ROIs with long straight edges. However, ROIs with curved edges can be more challenging to cover using straight tracks because they cannot accurately follow the curvature of the ROI (Oksanen & Visala, 2009). As a result, some literature also addresses the use of curved guidance tracks for these types of ROIs.

The approach described in Hameed et al. (2010) for the generation of curved guidance tracks is split up into two parts: The determination of the longest curved edge of the ROI, and the generation of curved guidance tracks parallel to this edge.

Definition 11 (Curved edge). A *curved edge* is a sequence of straight edges of the ROI that meet the criteria of having an angle between two successive edges that is less than or equal to a specified threshold angle γ_{max}.

Note that if it holds $\gamma_{max} = 0$, this results in a straight edge.

To determine the longest curved edge of the ROI, the angles between two successive edges of the main boundary of the field are compared to a specified threshold value. The points between the edges where this threshold is exceeded are referred to as *splitting points*. The curved edges of the ROI are composed of the sequences of edges between these points, as visualized in Figure 28. The length of each curved edge is calculated by summing the lengths of its constituent straight segments, and the longest curve is identified by comparing these lengths.

The aim is to generate parallel guidance tracks along the longest curved edge of the ROI to cover the entire region (Hameed et al., 2010). This is done analogously to the generation of headland tracks from section Section 2. This approach can be used to process individual cells of decomposed ROIs, as well as any concave ROI. In

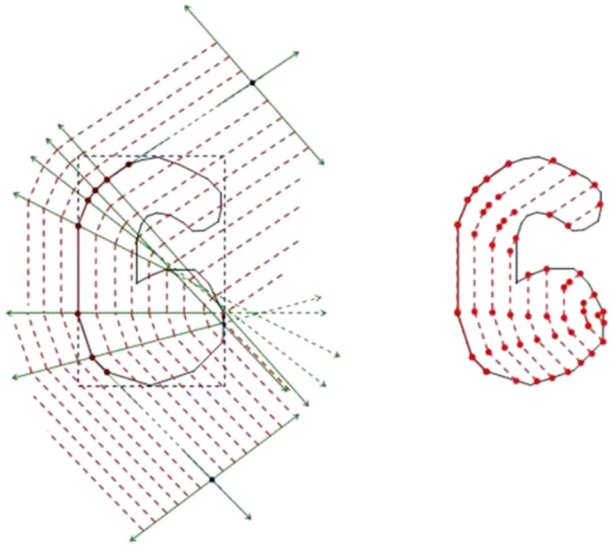

FIGURE 29 Generation of curved guidance tracks (Hameed et al., 2010).

the case of concave region, possibly with obstacles, the grouping of the tracks described in Section 5.7 can be utilized. An example of curved guidance tracks is shown in Figure 29. The same author gives an extension of this approach applicable to 3D terrain in Hameed, Bochtis, Sørensen, Jensen, et al. (2013).

6.1 | Consideration of curvature constraints

Despite its simplicity, this approach has some limitations, which are discussed in Rapoport et al. (2021). The curvature of the guidance tracks produced by this method is not constant, which can lead to singularities in the tracks, as already mentioned in the case for the headland tracks in Section 2. To eliminate these singularities, it may be necessary to further refine the tracks. Additionally, in practice, such nonsmooth paths can only be implemented through the use of machines with differential control of the rear wheel rotation speed, a kinematic scheme that is not widely used in agriculture. In reality, the steering angle of agricultural machines is constrained, which leads to an upper constraint of the curvature κ_{max}. Therefore a smooth path has to be created in a subsequent path smoothing step (see Section 1.2), which is traversing the guidance tracks as close as possible, while considering the curvature constraint κ_{max}. Already taking this restriction into account in the guidance track generation stage can improve the overall performance of the path in the end. If this constraint is not taken into account, it can lead to uncovered areas due to the constraint, as visualized in Figure 30. To comply with the constraint, either uncovered or overlapping areas are produced by the machine while traversing the path. The approach presented in Rapoport et al. (2021) allows but aims to minimize the total overlap of adjacent swaths while ensuring complete coverage and maintaining the maximum curvature constraint on the tracks. This is achieved by solving an optimization problem, formulated as an Second

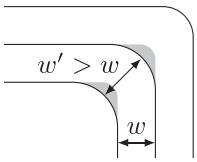

FIGURE 30 Appearence of uncovered areas (gray) in case of a maximum curvature constraint, adapted from Rapoport et al. (2021).

Order Cone program (SOCP), for which efficient computational solving methods exist. The aim of the optimization is to minimize the average curvature in the points of the guidance tracks and the average width of the overlap between adjacent swaths. This is done by using B-Splines to smoothen the discrete points of the guidance tracks to ensure the curvature constraint. As a result, the subsequent paths are straightened and the curvature constraint is satisfied, albeit with some degree of overlap between adjacent swaths. The basic steps of the algorithm are sketched below:

(1) Construct a B-Spline based on the initial points of the curved path and select equidistant points $p_0, \ldots, p_N$ on this smooth curve
(2) Determine the normals to the B-Spline n_i with $\|n_i\| = 1$ for each p_i.
(3) Solve the optimization problem below (15) for the optimal scalar valued step sizes a_i to determine new points $p_i^{new} = p_i + a_i n_i$.
(4) Use the points p_i^{new} as a set of control points for the next track and restart at 1.

The optimization problem in step 3 can be formulated as

$$\begin{aligned} \min_{a_1 \ldots, a_N} \quad & \sum_{i=1}^{N} \left(\beta \kappa_i^2 - a_i \right) \\ \text{s.t.} \quad & \alpha w \le a_i \le w, i = 0, \ldots, N \\ & \kappa_i \le \kappa_{max}, i = 0, \ldots, N, \end{aligned} \tag{15}$$

where the scalar α denotes the feasible measure of overlap of the adjacent swath with swath width w, and the scalar $\beta > 0$ is a weighting factor. Furthermore, κ_i denotes the curvature of the B-spline generated by using the points $p_i + a_i n_i$ for $i = 0, \ldots, N$ as control points, and evaluated in $p_i + a_i n_i$.

The same authors extend this spline approach idea (Tormagov & Rapoport, 2021) to 3D terrain by including 3D curvature in the computations.

7 | DISCUSSION

Coverage path planning in agricultural applications is a crucial task for optimizing efficiency and maximizing productivity. This survey serves as an introductory guide for individuals new to the field, providing them with a comprehensive understanding of the key concepts and approaches involved.

One of the primary considerations when embarking on coverage path planning is to identify the specific objectives that need to be

TABLE 1 Comparison of different cost functions considered for generating optimal interior guidance tracks.

Cost function	References	Advantages	Limitations
Number of turning maneuvers	Cabreira et al. (2019), Huang (2001), Li et al. (2011), Torres et al. (2016), Yu & Hung, 2015)	• Straightforward computation with linear complexity by polygon altitude and width • Independent of machine characteristics • Most common approach in literature	• Track traversing sequence and final smooth path (i.e. turning maneuvers) are not taken into account • Land topography is not considered • Kinematic or dynamic motion model of the machine is not considered
Path length	Jin & Tang (2010)	• Taking into account the actual turning maneuvers leads to a more accurate and meaningful result	• Assumption of sequential track traversing order • Less generalizable to other machines due to the determined turn types • Costful calculation of path length
Overlap area with headland	Hameed et al. (2011)	• Considering the different costs of turns with different angles to the headland	• Costful calculation of overlap area • Not commonly used • Measuring the efficiency of the path only using the overlap area can lead to an insufficient result

optimized in their particular application. By determining these objectives, farmers and researchers can focus their efforts on achieving the desired outcomes effectively.

The combination of headland tracks and interior tracks is a widely adopted procedure, especially for nonholonomic agricultural machines with large turning radii. The headland area plays a vital role in ensuring complete coverage in such cases. Therefore, it is recommended to follow this general procedure, at least as an initial starting point.

The generation of headland tracks is a well-discussed topic in the literature, making it relatively straightforward to implement based on established methodologies discussed in Section 2. By adhering to the suggested guidelines, researchers and practitioners can generate headland tracks efficiently and consistently.

Moving on to interior guidance tracks, there are various approaches available. For simpler ROIs, a fast and straightforward solution involves generating parallel tracks and clustering them afterwards, as shortly presented in Section 5.7. This approach eliminates the need for complex decomposition techniques and can yield quick results. It is particularly suitable for relatively uncomplicated ROIs.

However, a more comprehensive and common approach involves a combination of decomposition techniques and filling the resulting simple cells with parallel tracks. To determine the optimal track direction in each simple cell, a cost function must be defined. The discussed cost functions in Section 4 are summarized and compared in Table 1. Minimizing the number of turning maneuvers, as discussed in Section 4.1 is a commonly used and easy-to-implement objective, delivering meaningful results. Depending on the specific application, other objectives such as path length or overlap area, see Sections 4.2 and 4.3, can also be considered, necessitating the use of an appropriate motion model or other machine characteristics. Importantly, the calculation of the actual smooth path has to be performed simultaneously to obtain accurate objective values, which significantly increases the computational complexity of the optimization process.

When deciding between generating all tracks in a parallel manner or determining the optimal direction for each cell, as discussed in Section 5.3.4, farmers' preferences for a uniform result may influence the choice. However, opting for an optimal direction for each cell can lead to further improvements in the solution, particularly for irregularly shaped ROIs like C-shaped fields, visualized in Figure 22. Nonetheless, this approach requires turning maneuvers within the interior of the field.

Choosing a suitable decomposition method is the next step in the process. A summary of the discussed decomposition techniques in this paper is presented in Table 2, along with their advantages and limitations. The Boustrophedon method or its variations, discussed in Section 5.3.2, are recommended as a fast and straightforward starting point. By selecting an optimal sweep line direction following Section 5.3.4, this decomposition technique can produce satisfactory results that are both practical and applicable in real-world scenarios due to the regularity of the resulting cells.

For those seeking more sophisticated decomposition techniques and for research purposes, algorithms such as optimal merge options, optimal decomposing cuts, and dividing lines, referring to Sections 5.4–5.6, can be explored. However, it is important to note that these algorithms come with exponential computational complexity and lack readily available implementations. Nevertheless, they are suitable for providing benchmark solutions for different performance evaluations.

TABLE 2 Summary of different decomposition techniques used for generating optimal interior guidance tracks.

Decomposition method	References	Advantages	Limitations
Minimum cell	Chazelle (1985), Chazelle and Dobkin (1985), Fortune (1995), Keil (2000, 1985), and Schachter (1978)	• Standard algorithms, well-tested and established • Convex cells	• Number of cells is the only optimization criteria • (Some) do not cover the case of obstacles
Trapezoidal	Choset et al. (2005), Jimenez et al. (2007), Kim et al. (2014), Li et al. (2011), Oksanen & Visala (2009), Seidel (1991)	• Straightforward implementation • Convex cells • $O(n \log n)$ computation time, n number of vertices	• Highly dependent on the polygonal representation of the area • Often results in numerous and narrow cells
Boustrophedon	Choset (2000), Fang and Anstee (2010), Liu et al., (2019), Mannadiar and Rekleitis (2010), Pérez-González et al. (2021), Rekleitis et al. (2008)	• Straightforward implementation • Independent of polygonal representation of ROI • Usually significantly fewer cells than Trapezoidal • $O(n \log n)$ computation time, n number of vertices	• Non-convex (but monotone) cells
Alt. Boustrophedon	Pham et al. (2017)	• Straightforward implementation • Fewer cells than Boustrophedon	• Non-convex (but monotone) cells
Boustrophedon incl. concave inflections	Yu & Hung (2015)	• Straightforward implementation • Convex cells • Fewer cells than Trapezoidal	• More cells than Boustrophedon
Morse	Acar et al. (2002), Choset et al. (2005), Galceran & Carreras (2013), Milnor, 1963)	• Adjustable to variety of different field shapes by using different sweep functions h	• Smooth field boundaries required • Suitability of sweep function h highly dependent on the shape of the ROI
Optimal Merge Options	Huang (2001), Nielsen et al., 2019)	• Delivers optimal results • Result in (Huang, 2001) superior to single sweepline approach	• Eventually results in a large number of convex cells • Exponential complexity • Computation time blows up in case of a large number of concave vertices
Optimal Decomposing Cuts	Bochkarev and Smith (2016)	• Proof of correctness given	• Result highly depends on the initial convex decomposition
Dividing Lines	Jin and Tang (2010)	• Costs of different types of angled turns are included in the optimization process • Adjustable precision by choosing the angle between the rays	• Dependent on the resolution for the angle • Only sequential turns are considered

(Continues)

TABLE 2 (Continued)

Decomposition method	References	Advantages	Limitations
		• $O(n^3 \log n)$ computation time, n number of vertices	• Performance decreases significantly with the number of obstacles
		• Several improvements to the optimization are given to lower the computation time	
Grouping of Guidance Tracks	Hameed et al. (2010), Hameed (2017), Hameed, Bochtis, and Sørensen (2013)	• No decomposition technique needed	• Only parallel guidance tracks possible
		• Easy to implement	

Although the adoption of curved interior tracks is not as common, it can be beneficial when dealing with curved ROIs. Researchers are encouraged to experiment with this approach to evaluate its efficacy and assess the resulting coverage path.

After generating the overall track system, the calculation of intra- and intertraversing sequences, that is, the cell traversing order and the track traversing order inside each cell, becomes essential, and is followed by determining a smooth path. These steps complete the process of coverage path planning, providing farmers and researchers with a comprehensive path for efficient and effective agricultural operations. For example the software library *Fields2Cover* presented in Mier et al. (2023) can serve as a starting point for developing CCPP methods.

In summary, coverage path planning in agricultural applications involves a careful consideration of objectives, the adoption of well-established procedures, and the utilization of appropriate decomposition techniques. Researchers and practitioners should strive to balance simplicity, computational efficiency, and the specific requirements of their ROIs to achieve optimal coverage paths. By implementing the suggested methodologies and exploring alternative approaches, the agricultural community can enhance productivity and contribute to sustainable farming practices.

8 | CONCLUSION AND FUTURE WORK

In this paper, we provided a framework for generating optimal guidance tracks for complete coverage path planning in agricultural fields, and gave a detailed survey of methods existing in literature to approach the CCPP problem. These methods are crucial in path planning in present-day agricultural practices, not just to make autonomous processes more efficient, but also as a way to conserve arable soil from damage due to compaction and over-processing. While there is general consensus in literature on how to plan guidance tracks for headlands, several strategies exist for interior tracks depending on the shape of the ROI. For convex ROIs, we identified optimal ways for parallel track placement and various minimization criteria that can be included in the cost function. For non-convex regions, we gave an in-depth review of techniques for optimal ROI decomposition, including optimal sweepline directions and optimal decomposing cuts. These optimization criteria, when accounted for in guidance track generation, can improve the processing of a field significantly. Consequently, a thorough comparison of these techniques is necessary before choosing a specific method ad-hoc. For fields with curved boundaries, we surveyed existing methods for generating curved guidance tracks, including consideration of curvature constraints. Furthermore, this survey serves as an introductory guide for individuals entering the research field, providing them with a comprehensive overview of the presented methods. Within this context, the advantages and disadvantages of each approach are thoroughly examined, leading to practical recommendations based on these evaluations.

Based on the surveyed literature, an important future direction for CCPP in agricultural fields is the implementation and comparison of different approaches, using a benchmark data set of test and real fields, and using different performance metrics to evaluate the best approach for a given ROI. Furthermore, the optimization approaches outlined for guidance track generation can be used in conjunction with optimization in route planning to determine the least costly way for processing a field.

ACKNOWLEDGMENTS

Open Access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Maria Höffmann http://orcid.org/0000-0002-6172-0760

REFERENCES

Acar, E.U., Choset, H., Rizzi, A.A., Atkar, P.N. & Hull, D. (2002) Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21, 331–344.

Akshya, J., Priyadarsini, P., Gadupudi, P. & Paladugula, K. (2022) Path planning of intelligent UAVs using computational geometric techniques. In: *International Conference on Electronic Systems and Intelligent Computing (ICESIC)*. pp. 190–193.

Backman, J., Piirainen, P. & Oksanen, T. (2015) Smooth turning path generation for agricultural vehicles in headlands. *Biosystems Engineering*, 139, 76–86. Available from: https://www.sciencedirect.com/science/article/pii/S1537511015001397

Basilico, N. & Carpin, S. (2015) Deploying teams of heterogeneous UAVs in cooperative two-level surveillance missions. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 610–615.

Bircher, A., Kamel, M.S., Alexis, K., Burri, M., Oettershagen, P., Omari, S., et al. (2016) Three-dimensional coverage path planning via viewpoint resampling and tour optimization for aerial robots. *Autonomous Robots*, 40, 1059–1078.

Bochkarev, S. & Smith, S.L. (2016) On minimizing turns in robot coverage path planning. In: *International Conference on Automation Science and Engineering*. IEEE Computer Society, pp. 1237–1242.

Bochtis, D.D. & Vougioukas, S.G. (2008) Minimising the non-working distance travelled by machines operating in a headland field pattern. *Biosystems Engineering*, 101, 1–12.

Bosse, M., Nourani-Vatani, N. & Roberts, J. (2007) Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 698–703.

Butler, Z.J., Rizzi, A.A. & Hollis, R.L. (1999) Contact sensor-based coverage of rectilinear environments. In: *IEEE International Symposium on Intelligent Control*.

Cabreira, T.M., Brisolara, L.B. & Paulo, R.F. (2019) Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3, 1–38.

Cabreira, T.M., Franco, C.D., Ferreira, P.R. & Buttazzo, G.C. (2018) Energy-aware spiral coverage path planning for UAV photogrammetric applications. *IEEE Robotics and Automation Letters*, 3(4), 3662–3668.

Cai, C., Chen, J., Yan, Q. & Liu, F. (2023) A multi-robot coverage path planning method for maritime search and rescue using multiple AUVs. *Remote Sensing*, 15(1), 93.

Canny, J. (1988) Constructing roadmaps of semi-algebraic sets, I: completeness. *Artificial Intelligence*, 37(1), 203–222.

Chakraborty, S., Elangovan, D., Govindarajan, P.L., Elnaggar, M.F., Alrashed, M.M. & Kamel, S. (2022) A comprehensive review of path planning for agricultural ground robots. *Sustainability*, 14, 9156.

Chazelle, B. (1985) Approximation and decomposition of shapes. In: Schwartz, J. T., & Yap, C. K. (Eds.), *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics*. Lawrence Erlbaum Associates, pp. 145–185.

Chazelle, B. & Dobkin, D.P. (1985) Optimal convex decompositions. In: *Computational Geometry*, Ser. Machine Intelligence and Pattern Recognition, vol. 2. pp. 63–133.

Chen, L. & Xu, J.-c. (2004) Optimal delaunay triangulations. *Journal of Computational Mathematics*, 22(2), 299–308.

Chew, L. (1987) Constrained delaunay triangulations. In: *3rd Symposium on Computational Geometry*. pp. 215–222.

Choset, H. (2000) Coverage of known spaces: the boustrophedon cellular decomposition. *Autonomous Robots*, 9(3), 247–253.

Choset, H. (2001) Coverage for robotics: a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31, 113–126.

Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., et al. (2005) *Principles of robot motion: Theory, algorithms, and implementation*. MIT Press.

Coombes, M., Fletcher, T., Chen, W.H. & Liu, C. (2020) Decomposition-based mission planning for fixed-wing UAVs surveying in wind. *Journal of Field Robotics*, 37, 440–465.

Degiovanni, M. (1997) Nonsmooth critical point theory and applications. *Nonlinear Analysis: Theory, Methods & Applications*, 30(1), 89–99.

Elfes, A. (1987) Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, 3(3), 249–265.

Fang, C. & Anstee, S. (2010) Coverage path planning for harbour seabed surveys using an autonomous underwater vehicle. In: *Oceans, IEEE Sydney*. pp. 1–8.

Fernández, J., Tóth, B.G., Cánovas, L. & Pelegrín, B. (2008) A practical algorithm for decomposing polygonal domains into convex polygons by diagonals. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 16, 367–387.

Fevgas, G., Lagkas, T., Argyriou, V. & Sarigiannidis, P. (2022) Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles. *Sensors*, 22(3), 1235.

Fomenko, A. & Kunii, T. (1997) *Topological modeling for visualization*. Tokyo: Springer-Verlag.

Fortune, S. (1995) Voronoi diagrams and delaunay triangulations. In: *Lecture notes series on computing: Computing in euclidean geometry*, vol. 4, pp. 225–265.

Fraichard, T. & Scheuer, A. (2004) From reeds and shepp's to continuous-curvature paths. *Transactions on Robotics*, 20, 1025–1035.

Galceran, E., Campos, R., Palomeras, N., Ribas, D., Carreras, M. & Ridao, P. (2015) Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles. *Journal of Field Robotics*, 32(7), 952–983.

Galceran, E. & Carreras, M. (2013) A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61, 1258–1276.

Ghaddar, A., Merei, A. & Natalizio, E. (2020) PPS: energy-aware grid-based coverage path planning for UAVs using area partitioning in the presence of NFZs. *Sensors*, 20(13), 3742.

Hameed, I., Bochtis, D., Sørensen, C. & Nøremark, M. (2010) Automated generation of guidance lines for operational field planning. *Biosystems Engineering*, 107, 294–306.

Hameed, I.A. (2017) Coverage path planning software for autonomous robotic lawn mower using dubins' curve. In: *IEEE International Conference on Real-Time Computing and Robotics (RCAR)*. Institute of Electrical and Electronics Engineers Inc., pp. 517–522.

Hameed, I.A., Bochtis, D. & Sørensen, C.A. (2013) An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas. *International Journal of Advanced Robotic Systems*, 10, 1–9.

Hameed, I.A., Bochtis, D.D. & Sorensen, C.G. (2011) Driving angle and track sequence optimization for operational path planning using genetic algorithms. *Applied Engineering in Agriculture*, 27, 1077–1086.

Hameed, I.A., Bochtis, D.D., Sørensen, C.G., Jensen, A.L. & Larsen, R. (2013) Optimized driving direction based on a three-dimensional field representation. *Computers and Electronics in Agriculture*, 91, 145–153.

Höffmann, M., Clemens, J., Stronzek-Pfeifer, D., Simonelli, R., Serov, A., Schettino, S., et al. (2022) Coverage path planning and precise localization for autonomous lawn mowers. In: *6th IEEE International Conference on Robotic Computing (IRC)*. pp. 238–242.

Höffmann, M., Patel, S. & Büskens, C. (2022) Weight-optimized NURBS curves: headland paths for nonholonomic field robots. In: *8th International Conference on Automation, Robotics and Applications (ICARA)*. pp. 81–85.

Houle, M.E. & Toussaint, G.T. (1985) Computing the width of a set. In: *1st Symposium on Computational Geometry*. Association for Computing Machinery, pp. 1–7.

Huang, W.H. (2001) Optimal line-sweep-based decompositions for coverage algorithms. In: *IEEE International Conference on Robotics and Automation (ICRA)*. vol. 1. pp. 27–32.

Jan, G.E., Tsai, W.C., Sun, C.-C. & Lin, B.-S. (2012) A delaunay triangulation-based shortest path algorithm with O(nlogn) time in the Euclidean plane. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. pp. 186–189.

Jimenez, P., Shirinzadeh, B., Nicholson, A. & Alici, G. (2007) Optimal area covering using genetic algorithms. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*. pp. 1–5.

Jin, J. & Tang, L. (2010) Optimal coverage path planning for arable farming on 2D surfaces. *Transactions of the ASABE*, 53, 283–295.

Kalburgi, S., Nair, V.G. & Guruprasad, K. (2020) *Application of coverage path planning algorithm for milling operations*. Springer, Singapore, pp. 213–220.

Keil, J. (2000) Polygon decomposition. *Handbook of Computational Geometry*, 2, 491–518.

Keil, J.M. (1985) Decomposing a polygon into simpler components. *SIAM Journal on Computing*, 14(4), 799–817.

Khan, A., Noreen, I. & Habib, Z. (2017) On complete coverage path planning algorithms for non-holonomic mobile robots: survey and challenges. *Journal of Information Science and Engineering*, 33, 101–121.

Kiemel, J., Yang, P., Meißner, P. & Kröger, T. (2019) PaintRL: coverage path planning for industrial spray painting with reinforcement learning. In: *RSS Workshop on Closing the Reality Gap in Sim2real Transfer for Robotic Manipulation*.

Kim, D.H., Hoang, G., Bae, M.-J., Kim, J.W., Yoon, S.M., Yeo, T.-K., et al. (2014) Path tracking control coverage of a mining robot based on exhaustive path planning with exact cell decomposition. In: *14th International Conference on Control, Automation and Systems (ICCAS)*. pp. 730–735.

Latombe, J.-C. (1991) *Robot motion planning*. Springer, New York, NY.

Li, Y., Chen, H., Er, M.J. & Wang, X. (2011) Coverage path planning for UAVs based on enhanced exact cellular decomposition method. *Mechatronics*, 21, 876–885.

Lin, H.-Y. & Huang, Y.-C. (2021) Collaborative complete coverage and path planning for multi-robot exploration. *Sensors*, 21(11), 3709.

Liu, Y., Tian, M., Wang, X. & Lv, J. (2019) Study on path planning of intelligent mower based on UWB location. In: *7th International Conference on Robot Intelligence Technology and Applications*. pp. 248–253.

Mannadiar, R. & Rekleitis, I. (2010) Optimal coverage of a known arbitrary environment. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 5525–5530.

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(4), 2166–2172.

Milnor, J. (1963) *Morse theory*. Princeton, NJ: Princeton University Press.

Moravec, H. & Elfes, A. (1985) High resolution maps from wide angle sonar. In: *IEEE International Conference on Robotics and Automation (ICRA)*. vol. 2. pp. 116–121.

Nam, L.H., Huang, L., Li, X.J. & Xu, J.F. (2016) An approach for coverage path planning for UAVs. In: *14th IEEE International Workshop on Advanced Motion Control (AMC)*. pp. 411–416.

Nielsen, L.D., Sung, I. & Nielsen, P. (2019) Convex decomposition for a coverage path planning for autonomous vehicles: interior extension of edges. *Sensors*, 19, 4165.

Ntawumenyikizaba, A., Viet, H.H. & Chung, T. (2012) An online complete coverage algorithm for cleaning robots based on boustrophedon motions and A* search. In: *8th International Conference on Information Science and Digital Content Technology (ICIDT)*. vol. 2. pp. 401–405.

Oksanen, T. (2007) *Path planning algorithms for agricultural field machines*. Ph.D. dissertation, Helsinki University of Technology, Automation Technology Laboratory.

Oksanen, T. & Visala, A. (2009) Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26, 651–668.

Parsons, T., Hanafi Sheikhha, F., Ahmadi Khiyavi, O., Seo, J., Kim, W. & Lee, S. (2023) Optimal path generation with obstacle avoidance and subfield connection for an autonomous tractor. *Agriculture*, 13(1), 56.

Pérez-González, A., Benítez-Montoya, N., Jaramillo-Duque, A. & Cano-Quintero, J.B. (2021) Coverage path planning with semantic segmentation for UAV in PV plants. *Applied Sciences*, 11(24), 12093.

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

Rapoport, L., Generalov, A., Shavin, M. & Tormagov, T. (2021) Navigation and control problems in precision farming. In: *28th International Conference on Integrated Navigation Systems (ICINS)*.

Ravankar, A., Ravankar, A., Kobayashi, Y., Hoshino, Y. & Peng, C.-C. (2018) Path smoothing techniques in robot navigation: state-of-the-art, current and future challenges. *Sensors*, 18, 3170.

Rekleitis, I., New, A.P., Rankin, E.S. & Choset, H. (2008) Efficient boustrophedon multi-robot coverage: an algorithmic approach. *Annals of Mathematics and Artificial Intelligence*, 52(2–4), 109–142.

Rippa, S. (1990) Minimal roughness property of the delaunay triangulation. *Computer Aided Geometric Design*, 7(6), 489–497.

Santos, L.C., Santos, F.N., Pires, E.J.S., Valente, A., Costa, P. & Magalhaes, S. (2020) Path planning for ground robots in agriculture: a short review. In: *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. Institute of Electrical and Electronics Engineers Inc. pp. 61–66.

Schachter, B. (1978) Decomposition of polygons into convex sets. *IEEE Transactions on Computers*, C-27(11), 1078–1082.

Seidel, R. (1991) A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry: Theory and Applications*, 1, 51–64.

Tan, C.S., Mohd-Mokhtar, R. & Arshad, M.R. (2021) A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access*, 9, 119310–119342.

Tormagov, T. & Rapoport, L. (2021) Coverage path planning for 3D terrain with constraints on trajectory curvature based on second-order cone programming. *Communications in Computer and Information Science*, 1514, 258–272.

Torres, M., Pelta, D.A., Verdegay, J.L. & Torres, J.C. (2016) Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Systems with Applications*, 55, 441–451.

Utamima, A. & Djunaidy, A. (2021) Agricultural routing planning: A narrative review of literature. In: *Procedia Computer Science*. vol. 197. pp. 693–700.

Utamima, A., Reiners, T. & Ansaripoor, A. (2022) Evolutionary neighborhood discovery algorithm for agricultural routing planning in multiple fields. *Annals of Operations Research*, 316, 955–977.

Utamima, A., Reiners, T., Ansaripoor, A. & Seyyedhasani, H. (2018) *Contemporary approaches and strategies for applied logistics*. The Agricultural Routing Planning in Field Logistics, pp. 261–283.

Wei, J.-H. & Liu, J.-S. (2010) Generating minimax-curvature and shorter η 3-spline path using multi-objective variable-length genetic algorithm. In: *International Conference on Networking, Sensing and Control (ICNSC)*, pp. 319–324.

Ye, X., Luo, L., Hou, L., Duan, Y. & Wu, Y. (2020) Laser ablation manipulator coverage path planning method based on an improved ant colony algorithm. *Applied Sciences*, 10(23), 8641.

Yijun, Z., Jiadong, X. & Chen, L. (2021) A fast bi-directional A* algorithm based on quad-tree decomposition and hierarchical map. *IEEE Access*, 9, 102877–102885.

Yu, X. & Hung, J. (2015) Coverage path planning based on a multiple sweep line decomposition. In: *41st Conference of the IEEE Industrial Electronics Society (IECON)*. pp. 4052–4058.

How to cite this article: Höffmann, M., Patel, S. & Büskens, C. (2024) Optimal guidance track generation for precision agriculture: A review of coverage path planning techniques. *Journal of Field Robotics*, 41, 823–844. https://doi.org/10.1002/rob.22286