FIELDS2COVER

Transparent and efficient coverage paths for autonomous agricultural vehicles

by Gonzalo Mier Muñoz

Propositions

1. Farm income is directly impacted by the routes taken by agricultural vehicles. (this thesis)
2. Automation in farming will replace labor, not farmers. (this thesis)
3. Unpaid paper reviewing impoverishes science. (science)
4. Sexual-affective relationships between PhD candidates and their supervisors generate intolerable power dynamics in academia. (science)
5. Universal basic income is key to solving mental health crises. (society)
6. Open-source communities run the world thanklessly. (society)

Propositions belonging to the thesis, entitled
Fields2Cover: Transparent and efficient coverage paths for autonomous agricultural vehicles

Gonzalo Mier
Wageningen, January 30, 2026

Fields2Cover: Transparent and efficient coverage paths for autonomous agricultural vehicles

Gonzalo Mier

Thesis committee

Promotor
Prof. Dr *ir.* S. de Bruin
Professor of Geo-information Science and Remote Sensing
Wageningen University & Research

Co-promotor
Dr J. Valente
Scientist, Centre for Automation and Robotics
Spanish National Research Council, Madrid, Spain

Other members:
Prof. Dr A. Fensel, Wageningen University & Research
Prof. Dr M. Kreveld, Utrecht University
Prof. Dr T. Oksanen, Technical University of Munich, Germany
Prof. Dr C. Grøn Sørensen, Aarhus University, Denmark

This research was conducted under the auspices of the C.T. de Wit Graduate School of Production Ecology & Resource Conservation (PE&RC)

Fields2Cover:

Transparent and efficient coverage paths for autonomous agricultural vehicles

Gonzalo Mier

Thesis
submitted in fulfillment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus
Prof. Dr C. Kroeze,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on Friday 30 January 2026
at 3:30 p.m. in the Omnia Auditorium.

Gonzalo Mier

Fields2Cover: Transparent and efficient coverage paths for autonomous agricultural vehicles

143 pages.
PhD thesis, Wageningen University, Wageningen, the Netherlands (2026)
With references, with summary in English

DOI https://doi.org/10.18174/681452

Summary

By 2050 the world must feed about 9.4 billion people while many farms already struggle to hire enough workers. Farmers need to raise yields with fewer hands and without degrading soil health. For decades, they have searched to increase their yields using ever heavier machines. These machines compact the soil, which blocks water flow and hinders root growth. Such damage can be mitigated by adopting lighter robots and by rethinking how machines move across the fields.

Controlled Traffic Farming keeps machinery on fixed lanes and limits compaction to narrow strips. To use those lanes, a path has to be planned that follows them, and still covers the whole field. This task defines the Agricultural Coverage Path Planning problem (ACPP).

Most work on ACPP reduces path length or energy and covers only the inner field. The headland coverage is usually overlooked, leaving that area uncovered. Furthermore, soil compaction is typically approximated by the path length, without considering the soil properties themselves. Moreover, researchers usually do not share their code or data, slowing down progress and blocking fair comparison of methods.

The main objective of this study is to improve the transparency and efficiency of coverage path planners by releasing an open-source library with the developed methods, and by proposing new algorithms that include the soil compaction as a objective function and achieving headland coverage. The thesis formulates and addresses four research objectives: (1) Integrating coverage path planning algorithms into a unified framework; (2) Enhancing the scientific transparency of coverage path planning research; (3) Improving headland coverage and headland turning feasibility; (4) Reducing soil compaction through coverage path planning optimization. After introducing the topic, the thesis contains four main chapters, each dedicated to exploring one of the research objectives outlined above. These chapters are followed by a synthesis focusing on the advances of the research addressed in the thesis and their potential implications for addressing the agricultural coverage path planning problem.

Chapter 2 introduces Fields2Cover, an open source library for the agricultural coverage path planning problem. The first version of the library aimed to cover convex fields with

straight swaths. The library groups the coverage path planning problem into four modules that generate the headland, create the swaths, sort those swaths into a route, and smooth the route into a path. This modularity allows researchers to swap methods and test ideas without re-implementing the complete workflow. The library is available in C++, Python, and as a ROS module, and it was tested in experiments with real robots. The Fields2Cover library is provided with an open license in Github.

Chapter 3 develops and demonstrates a benchmark facilitating the comparison of new ACPP algorithms. The chapter extends the modular structure from Chapter 2, dividing methods into five modules and allowing flexibility in their execution order. It also enhances the library to support non-convex fields with obstacles. The benchmark tool is released with a dataset of fields geometries. Additionally, the library was extended to include a configuration file that allows adapting to user needs. It is accompanied by three elaborate use-case examples.

Chapter 4 presents a headland coverage path planner with three corner planners, designed to cover both convex and concave corners on field boundaries. These planners consider the dimensions of both the robot and a fixed implement, as well as maximum curvature and curvature change rate constraints. The algorithm allows selecting the appropriate corner planner based on whether the objective is to minimize path length or to maximize area coverage. Finally, this chapter explains how to concatenate curves in corners to effectively cover the headlands of a field.

Chapter 5 focuses on optimizing the coverage route to reduce soil compaction by integrating the SoilFlex model into the objective function. To illustrate the difference between this objective and that of simply minimizing path length, planning was conducted under varying operational frequencies, from a single pass up to three passes. The model was also tested in intercropping scenarios, where two crops share the same field and two vehicles are used to cover the swaths while minimizing the overall soil compaction.

By releasing the Fields2Cover library and its benchmark data, this thesis opens coverage path planning to all and makes its steps transparent. The library uses a modular workflow. ACPP splits into many subproblems, and this modular approach allows third parties implement a new method for one subproblem without rewriting the rest. Using the Fields2Cover library as a baseline, the work refines ACPP by adding a soil compaction objective function and a headland coverage planner. These advancements improve the soil health by optimizing the paths followed by the machinery. This thesis contributes to increasing trust in ACPP methods, helps other researchers conduct their research faster, and reduce soil damage caused by machinery. Its open design also invites other teams to integrate ACPP into larger systems, opening the door to new applications.

Contents

Chapter 1

Introduction

1.1 Labor shortage and soil degradation

In thirty years the global population will have grown substantially. Demographers expect about 9.4 billion people by 2050 (Economic and Affairs, 2021). At the same time many farms lack enough hands to collect harvest and to operate machines (Bousmah and Grenier, 2022). Farmers must produce more food with fewer workers while also keeping their soils healthy (Ajibade et al., 2023).

Soil degradation is one of the main threats to that goal (Hossain et al., 2020). One third of the world's cropland shows signs of erosion, salt build-up, high acidity, loss of organic matter, or compaction (Smith et al., 2024). Soil compaction deserves special attention because tight pores hold little water and air, and they block the roots from reaching nutrients (Nawaz et al., 2013). Heavy farm machines cause most of this soil compaction (Calleja-Huerta et al., 2023).

Scientists see two chief ways to ease soil stress. First they suggest lighter field machines (Keller et al., 2019). Second they turn to Controlled Traffic Farming (CTF) which keeps wheels on fixed lanes and leaves the rest of the field loose (Tamirat et al., 2022). Both approaches need precise guidance that steers every pass along the planned lanes while still treating the whole crop area. To guide those vehicles, a path that covers the complete field is needed. Designing such guidance while ensuring full coverage forms the core of the Coverage Path Planning problem (CPP) (Oksanen and Visala, 2009).

1.2 The agricultural coverage path planning problem

Coverage Path Planning, or CPP, has its origin in robotics applications that are quite distinct from those in agriculture (Galceran and Carreras, 2013). As a first example, vacuum cleaning robots use CPP to cover an area without prior knowledge of its geometry, including any inside obstacles (Yasutomi et al., 1988). Secondly, demining robots move within a predefined area, but in contrast to typical agricultural operations, they may be allowed to revisit a previously covered zone if that shortens the path (Acar et al., 2003). Thirdly, flying drones that need to cover an area with imagery may be allowed to navigate outside the region-of-interest, if there are no obstacles that block the path of the drone (Barrientos et al., 2011). In every domain, the planner must cover the target area while minimizing the distance traveled (Galceran and Carreras, 2013).

In Agricultural Coverage Path Planning (ACPP) the problem is more complex. Zones outside of the region of interest (beyond the field boundaries) cannot be trespassed due to risk of collision (Plessen, 2025). Typical obstacles are trees, fences, and ditches. Moreover, the coverage path must avoid double coverage because repeat passes may damage the crop (Hameed et al., 2016). On the other hand, leaving an area uncovered implies less field

yield or the possibility of having weeds inside the field (Jin and Tang, 2011). Lastly, the field borders are known, or they can be measured, which allows us to plan the coverage path before the operation (Versleijen and de Bruin, 2019).

Agricultural fields are usually divided into an area where most crop productivity is achieved, called the inner field, and an outer area where turns are made, called the headlands (Pour Arab et al., 2022). The inner field is typically covered by adjacent swaths. These swaths can be arranged according to the farmer's preferences, crop operational characteristics, or by using an optimization algorithm (de Bruin et al., 2009). The agricultural CPP problem has been primarily studied for non-holonomic vehicles, which are vehicles that cannot rotate in place (He et al., 2023). Those vehicles are physically constrained in their maximum curvature and maximum steering rate. Hence, they can only follow smooth paths that abide the robot's steering constraints. Turning maneuvers between the swaths are done on the headlands, which are adjacent to the field boundaries and any in-field obstacles (Guevara et al., 2020). As headlands are less productive, their size is minimized without compromising the minimum space required for the turns (Spekken and de Bruin, 2013).

Comparison of coverage path alternatives requires an objective function, which captures the goals and costs of field operations (Bostelmann-Arp et al., 2023). Common objective functions include total path length, operation time, number of turns, a measure of soil compaction, coverage completeness, overlapped area, and energy consumption (Höffmann et al., 2024). There are many objective functions, but some of them are proxies of others. For example, the number of turns is an indicator of total operational time, since turns imply idle operation whilst speed is slower than that for swath coverage (Jin and Tang, 2010).

In convex, obstacle-free fields, swaths may be covered sequentially by alternating direction until the entire area is treated (Bochtis and Vougioukas, 2008). In concave fields, such as fields with obstacles, this sequence is typically inapplicable, necessitating route optimization algorithms to determine the optimal order for covering the swaths (Hameed et al., 2011). The agricultural operation itself imposes additional constraints to coverage planning. For instance, when harvesting, it is crucial to cover the headlands first so that the standing crop is not damaged prior to harvesting by the turning on the headlands (Wang et al., 2025). Conversely, during seeding, the inner field is prioritized, leaving headland coverage for later if required (Bochtis and Oksanen, 2009).

Other important aspects to be considered in coverage planning are the vehicle's weight and load capacity. Some operations require the robot to transport a load that changes during the task (Spekken and de Bruin, 2013). For example, during fertilization the robot's tank gradually empties as it traverses the swaths, necessitating periodic stops for refilling (Bochtis et al., 2010b). On the contrary, a combine harvester's tank fills during operation and must be emptied before its capacity is surpassed (Lu et al., 2020). The

route planner must thus manage both the order of swath coverage and the logistics of recharge points, which are essential for operational optimization (Hameed et al., 2013a). This challenge is compounded when multiple robots are working simultaneously, as work must be efficiently distributed to minimize the overall objective function (Bochtis et al., 2010b).

An additional challenge in ACPP is the scarcity of publicly available information (Chakraborty et al., 2022). Unlike other robotics fields, datasets and open-source code in ACPP are limited, hindering reproducibility and the comparison of existing methods (Utamima and Djunaidy, 2022). This lack of accessible resources poses a significant barrier for researchers, who often have to repeatedly collect data and implement algorithms, which leads to a waste of resources.

1.3 Open science in the context of coverage path planning

The Open Science movement seeks to open scientific progress to every researcher and citizen, making work both accessible and reproducible (Collaboration, 2015). Three of the movement pillars are open access, open data, and open-source software. Together they bring clarity, invite collaboration, and let others confirm or refine earlier findings (Crüwell et al., 2019). By easing the flow of information, open science speeds advances across many fields (Munafò et al., 2017). In precision agriculture, these principles have the potential to transform how we address the ACPP challenges, raising both efficiency and sustainability in agricultural management.

Open access publishing shifts the cost of availability away from readers and removes paywalls that hinder knowledge transfer (Sitek and Bertelmann, 2014). Open-source software reveals the exact methodology used so that others can adopt, adjust, or improve it (Fecher and Friesike, 2014). This is particularly important in ACPP, where advanced solutions must be adaptable and scalable across diverse agricultural environments. Open data further supports the replication of research by providing the original datasets (Grossman et al., 2010), thus ensuring result veracity (Ioannidis, 2005). These three ideas would give ACPP projects the flexibility they need to suit multiple crops, fields, and machine fleets.

Several active projects related to agriculture and robotics have already demonstrated the scientific and social impact of open-source. For example, GODAN (Global Open Data for Agriculture and Nutrition) shares agricultural and nutrition data on a global scale (Musker and Schaap, 2018). Farmhack is a community dedicated to developing open-source tools that allow farmers to share and refine agricultural technologies (*Farm hack website* n.d.). In robotics, ROS (Robot Operating System) streamlines communication between robots and fosters a broad developer base (Macenski et al., 2022). Nav2, built on ROS, provides

open navigation tools for complex settings (Macenski et al., 2020). These projects not only provide open code, but also publicly available documentation for anyone to use and extend.

Nevertheless, resistance to open science in ACPP is not accidental. The high costs of field trials and dataset curation encourage proprietary control (Mukherjee and Stern, 2009). Furthermore, the diversity of fields, crops, and machines complicates standard protocols for ACPP (Chakraborty et al., 2022). Path planners also optimize distinct goals such as path length, soil compaction, or energy use, which makes fair comparison difficult and slows down the adoption of shared benchmarks (Utamima and Djunaidy, 2022).

To overcome these obstacles, we need incentives that reward openness, standards that reflect farm diversity, and testbeds that measure a variety of objectives. If these conditions are met, open science can turn ACPP into a fast-moving, community driven field that delivers practical tools for sustainable agriculture.

1.4 Problem statement and research objectives

Over recent decades, various individual efforts have been undertaken to solve the ACPP problem, primarily focusing on finding the shortest path for covering the inner field of monocultures (Höffmann et al., 2024). Although automation reduces the need for manual labor, these approaches often overlook fundamental issues such as headland coverage (Pour Arab et al., 2022) and minimizing soil compaction (Jin and Tang, 2011). Studies addressing soil compaction often rely on overly simplistic linear models (Santos et al., 2018). Similarly, when headlands are treated merely as transition zones between swaths, they are not adequately covered (Nilsson and Zhou, 2020). Pour Arab et al. (2022) proposed several methods for headland coverage, but the work fails to specify which criteria to use to select a particular method. Moreover, the navigation of concave corners in headlands remains largely unexplored in the literature.

In addition, knowledge on ACPP is fragmented across the literature, much of which is behind paywalls. As previously mentioned, the software and datasets associated with these studies are rarely shared, underscoring the need for a comprehensive solution that compiles the state of the art with detailed documentation (Utamima and Djunaidy, 2022). Such a solution must be generic enough to address any ACPP use case while remaining flexible enough to adapt to specific situations.

The primary objectives of this thesis are to improve the open access in ACPP and to solve the challenges associated with optimizing field coverage while minimizing soil compaction. These objectives are explored through the following research objectives:

1. **Integrating coverage path planning algorithms into a unified framework**
2. **Enhancing the scientific transparency of coverage path planning research**

3. **Improving headland coverage and headland turning feasibility**
4. **Reducing soil compaction through coverage path planning optimization**

1.5 Thesis outline

This thesis is organized in six chapters, beginning with this introduction. Chapters 2 through 5 address the research questions outlined above. The following is an overview of the upcoming chapters:

- Chapter 2 introduces the open-source library Fields2Cover, which offers a modular structure for ACPP focused on convex fields and straight swaths. The library divides algorithms into four modules. Fields2Cover aims to provide a generic, publicly accessible solution to stimulate interest and attract developers and researchers. The library is available in C++, Python, and as a ROS module, and its generated paths have been tested on a real field robot.
- Chapter 3 develops and demonstrates a benchmark facilitating the comparison of new ACPP algorithms. The chapter extends the modular structure from Chapter 2, dividing methods into five modules and allowing flexibility in their execution order. It also enhances the library to support non-convex fields with obstacles. The benchmark tool is released with a dataset of fields geometries. Additionally, the library was extended to include a configuration file that allows adapting to user needs. It is accompanied by three elaborated use-case examples.
- Chapter 4 presents a headland coverage path planner with three corner planners, designed to cover both convex and concave corners on field boundaries. These planners consider the dimensions of both the robot and a fixed implement, as well as maximum curvature and curvature change rate constraints. The algorithm allows selecting the appropriate corner planner based on whether the objective is to minimize path length or to maximize area coverage. Finally, this chapter explains how to concatenate curves in corners to effectively cover the headlands of a field.
- Chapter 5 focuses on optimizing the coverage route to reduce soil compaction by integrating the SoilFlex model (Keller et al., 2007) into the objective function. To illustrate the difference between this objective and that of simply minimizing path length, planning was conducted under varying operational frequencies, from a single pass up to three passes. The model was also tested in intercropping scenarios, where two crops share the same field and two vehicles are used to cover the swaths while minimizing the overall soil compaction.
- Chapter 6 concludes the thesis by summarizing the main findings and reflecting on future challenges in ACPP.

Chapter 2

Fields2Cover: An open-source coverage path planning library for unmanned agricultural vehicles

This chapter is based on:

G. Mier, J. Valente, and S. de Bruin (2023b). "Fields2Cover: An open-source coverage path planning library for unmanned agricultural vehicles". *IEEE Robotics and Automation Letters* 8.4, 2166–2172. DOI: 10.1109/LRA.2023.3248439

Abstract

This chapter describes Fields2Cover[1], a novel open source library for coverage path planning (CPP) for agricultural vehicles. While there are several CPP solutions nowadays, there have been limited efforts to unify them into an open source library and provide benchmarking tools to compare their performance. Fields2Cover provides a framework for planning coverage paths, developing novel techniques, and benchmarking state-of-the-art algorithms. The library features a modular and extensible architecture that supports various vehicles and can be used for a variety of applications, including farms. Its core modules are: a headland generator, a swath generator, a route planner and a path planner. An interface to the Robot Operating System (ROS) is also supplied as an add-on. In this chapter, the functionalities of the library for planning a coverage path in agriculture are demonstrated using 8 state-of-the-art methods and 7 objective functions in simulation and field experiments.

[1] https://github.com/Fields2Cover/Fields2Cover

2.1 Introduction

In developed countries, there is a shortage of skilled workers to operate agricultural machinery (Christiaensen et al., 2020). This shortage can be alleviated with the development of autonomous machinery. Unlike manually operated machinery, autonomous vehicle operations need meticulous planning beforehand. The problem of determining a path to cover a field is known as coverage path planning (CPP). CPP is of high importance for cleaning (Bormann et al., 2018), surveillance robots (Jensen-Nau et al., 2020), lawn mowers (Hameed, 2017), and agricultural vehicles (Oksanen and Visala, 2009), where it has been addressed in several works.

Whilst there have been many efforts, most of the (partial) CPP solutions have not been released as open-source software, thus hindering more rapid advances in CPP by the scientific community. The packages shown in Table 2.1 are the only open-source software to the best of our knowledge. Note that the software packages listed in Table 2.1 solve the CPP problem partially, but require several modifications in order to be customized to different unmanned vehicles and applications.

This chapter aims to fill the above mentioned gap by proposing and releasing to the community an open-source CPP library for field coverage. The library was designed focusing in four modules that are the core of CPP solutions: a headland generator, a swath generator, a route planner, and a path planner. Each module includes at least one state-of-the-art method and one objective function. The library currently only supports convex fields on arable farmland. Regardless, there is an urgent need for an open source software solution to fill the existing gap in the CPP problem in agriculture. The ultimate goal of the library is to ease the state of-the-art algorithm benchmark and to accelerate CPP research and application.

2.1.1 Related work

Owing to the non-holonomous nature of agricultural vehicles, a region of the field known as headlands must be reserved for turning the vehicle. The most basic approach is to allocate a constant width area around the field. This strategy allocates a large amount of space to a poor yield area. Depending on how the swaths are arranged, some headland areas are parallel to the swaths and hence they are not needed for turning. By only constructing headlands along the field edges where turns are made, the area reserved for them can be minimized (Jin, 2009; Oksanen and Visala, 2009).

Swaths are generated in the inner field, which is the remaining region after subtracting the headlands. In two-dimensional planar fields, a reference line can be applied as a guide for the generation of swaths, where each parallel creates a swath (de Bruin et al., 2014; Jin, 2009; Oksanen and Visala, 2009). This line can be chosen for convenience or by an algorithm such as brute force or a meta-heuristic. Oksanen and Visala (2009)

Table 2.1: Comparison between coverage path planning open-source software solutions. Repositories (rows) are compared in terms of (1) available documentation (Docs); (2) Computation of exact solutions rather than using a discretizing grids (No grid); (3) Support for non-holonomous vehicles in turns; (4) The option to reserve maneuvering space at the field boundaries (Headlands support); (5) The possibility to modify the objective function; (6) Applicability for agricultural ground robots.

Package name	*Docs*	*No grid used*	*Non-holonomous*	*Headlands support*	*Customizable objective function*	*Terrestrial agricultural vehicles*
RJJxp/CoveragePlanning	✗	✔	✗	✗	✗	✗
Nobleo/full_coverage_path_planner	✔	✗	✔	✗	✗	✗
Ipa320/ipa_coverage_planning	✔	✔	✔	✗	✗	✗
Ethz-asl/polygon_coverage_planning	✔	✔	✗	✗	✗	✗
Irvingvasquez/ocpp	✗	✔	✗	✗	✗	✗
Greenzie/boustrophedon_planner	✗	✔	✔	✔	✗	✗
Ipiano/coverage-planning	✔	✔	✗	✗	✔	✗
Fields2Cover	✔	✔	✔	✔	✔	✔

describes a driving angle search strategy that requires fewer iterations than brute force search but it does not guarantee finding the global minimum. Objective functions such as the number of turns or the sum of swath lengths are used to determine optimality in swath generation (Jin, 2009).

The distance (Jin, 2009) and time (Meuth and Wunsch, 2008) required to cover the field are affected by the order of the swaths. A route is the sequence of the swaths to cover. The Boustrophedon order, which travels the swaths sequentially from one side of the field to the other, and the snake order, which skips one swath at each turn and returns through the uncovered swaths, are popular preset routing patterns (Zhou et al., 2015). Objective functions such as distance, number of rotations, or time necessary to traverse the field (Jin, 2009; Meuth and Wunsch, 2008) are minimized by finding the optimal route through meta-heuristics (Spekken et al., 2016).

A path is composed of the swaths of a route connected by turns, forming a continuous line along which the vehicle will drive. Dubins' or Reeds-Shepp's curves are turns that minimize the path length of the turns (Dubins, 1957; Reeds and Shepp, 1990). These curves are made by either curve segments or straight lines. The main problem is that there is an instantaneous change of curvature at the transition point between two segments. Techniques such as numerical integrators (Backman et al., 2015) or clothoids (Sabelhaus et al., 2013) are employed to smooth the turn to avoid the curvature discontinuity. Furthermore, to navigate from a swath to the headlands, turns such as non-uniform rational B-spline (NURBS) curves can be adopted (Höffmann et al., 2022).

CPP problems are composed of numerous sub-problems, several of which have received special attention in the literature. For example, Spekken et al. (2016) presents an approach for calculating the coverage path in undulating terrain that however does not consider turns between rows or headland creation. Nilsson and Zhou (2020) and Nørremark et al. (2022) divide the CPP problem into two major modules: Field Partitioning/Representation, where the distribution of headlands and swaths in the field is set up, and Route Planning, which determines the optimal order of travelling the swaths within sub-fields. In the latter framework, each module has more than one function, increasing the complexity of comparing multiple variations of the module.

2.1.2 Existing open-source software

There have been web applications, such as GAOS (de Bruin et al., 2014), that allowed farmers to design or adapt coverage paths with a user-friendly interface. Many of such web applications, despite being a great help to the farming community, have been developed in collaboration with companies, restricting the possibility to release the code to the public domain.

The currently existing open source CPP repositories are listed in Table 2.1. Although seven other projects were found, none of them can be adopted for farming purposes with ground robots. As mentioned above, ground robots in agriculture are generally non-holonomous, so turning maneuvers must be planned to move from one swath to another. Unfortunately, some packages (Baehnemann et al., n.d.; Jiaping, n.d.; Stelter, n.d.; Vasquez, n.d.) only compute the route to cover a region. These packages are designed for quadrotors (Baehnemann et al., n.d.; Vasquez, n.d.) or for indoor robots (Jiaping, n.d.). However, the code needs to be modified to support path generation for non-holonomous robots. A special case of CPP is the Nobleo package (Clephas et al., n.d.) which, although the vehicle used is non-holonomous, uses a grid to define the nodes that should be covered at least once. In agriculture, it is important to reduce the damage caused by the wheels of the vehicle, so it is not recommended to cover the same swath several times (Clephas et al., n.d.) or to cross through the main field (Baehnemann et al., n.d.; IPA, n.d.; Jiaping, n.d.). On the other hand, Greenzie (n.d.), which was developed for lawn mowers, is the only package that supports headlands, along with Fields2Cover. Unlike arable farming, mowers are constrained to avoid repeated tracks for field traffic, thus the coverage path is created with random sweep angles. For this reason, Greenzie does neither provide an optimizer nor an objective function for planning the swaths. In contrast, Ipiano (Stelter, n.d.) provides an interface to change the objective function used by its optimizer, but here no headland support is offered. Fields2Cover is the only software solution that provides algorithms to create a coverage path for terrestrial agricultural robots, including optimizers and objective functions to generate the best path, headland support and turn planning.

2.1.3 Contributions

The main contributions of this chapter and the Fields2Cover library are:

1. A publicly-available library (Fields2cover) providing connectable modules to address CPP problems with unmanned agricultural vehicles. Those modules can be effortlessly customized for other CPP problems.
2. Benchmark tools for quantitative comparison between the CPP algorithms and approaches.
3. A quantitative comparison using 38 convex fields between eight state-of-the-art CPP approaches/methods and seven objective functions.
4. Experiments with a commercial unmanned agricultural vehicle, demonstrating Fields2Cover's capability to provide real-world solutions.
5. Contribute to building a research community by providing an accessible platform for discussing novel ideas, solving issues and sharing new variants of the problem.

2.2 Fields2Cover

Fields2Cover is designed in four modules (Fig. 2.1): 1) Headland Generator, 2) Swath Generator, 3) Route Planner and 4) Path Planner. The inputs of the CPP problem are the shape of the field and the vehicle specifications, while the output is the coverage path of the field. Methods from the same module can be used interchangeably to compare their solutions independently of the rest of the CPP problem.

2.2.1 Headland Generator module

The Headland Generator module currently implements a single method that buffers the border of the field in the inward direction by a custom constant width (see Module 1 in Fig. 2.1). The objective function of this module is the area of the remaining field after removing the headlands.

$$A_r = \frac{A_{\bar{hl}}}{A_f} \tag{2.1}$$

where $A_{\bar{hl}}$ is the area of the field without headlands (mainland), A_f the area of the original field, and A_r is the ratio of mainland to the original field.

2.2.2 Swath Generator module

The inner field (i.e., excluding the headlands) is the input of the Swath Generator module (see Module 2 in Fig. 2.1). This region is divided into parallel swaths matching the

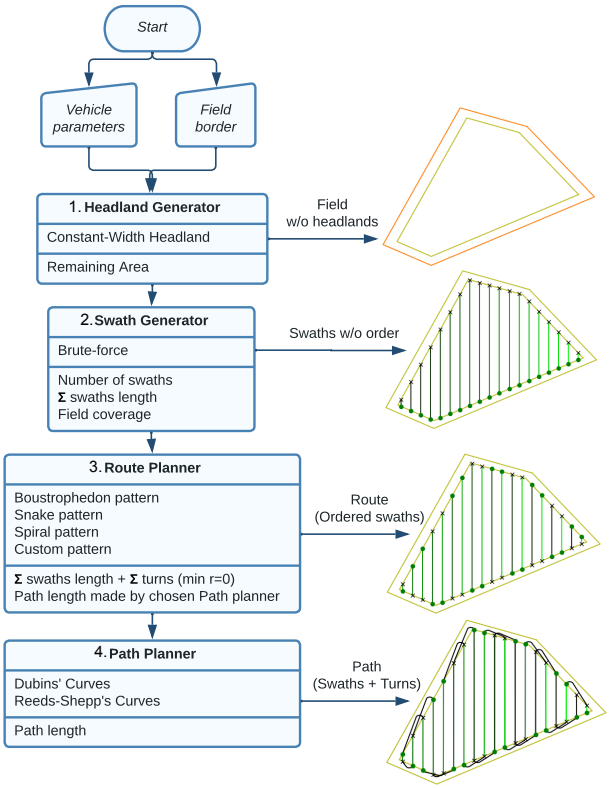


Figure 2.1: Diagram of the Fields2Cover library. The library contains four core modules: 1) Headland Generator, 2) Swath Generator, 3) Route Planner and 4) Path Planner. Each of the modules is represented as a box with 3 slots: the name of the module, the methods implemented, and the objective functions available. The output of a module is the input of the next module.

operating width. In the current version, the library only supports parallel non-overlapping swaths. Fields2Cover has a brute force algorithm to find the optimal sweep angle by trying discretized angles using a given step size. If the computer running the library supports multiple threads, several sweep angles are tried in parallel (Intel, n.d.).

This module currently implements 3 objective functions:

- Minimize the Number of Swaths. This objective function depends on the shape and the area of the field, and the width of the robot. The number of swaths is limited by the equation:

$$0 \leq \#S_\alpha \leq \frac{A_{\bar{hl}}}{R_w}, \tag{2.2}$$

  where $\#S_\alpha$ is the number of swaths for a given sweep angle α, $A_{\bar{hl}}$ is the area of the field without headland, and R_w is the operational width of the robot. The shape of the field that maximizes the minimum number of swaths is the square field, which

results in:

$$min_\alpha \ \#S_\alpha^\diamondsuit \simeq \frac{\sqrt{A_{\bar{hl}}}}{R_w}, \tag{2.3}$$

where $\#S_\alpha^\diamondsuit$ is the number of swaths in a square field with a given sweep angle α. Therefore, the optimal value of this objective function is less than the square root of the area of the field.

- Maximize the Field Coverage:

$$A_{cov} = \frac{A_{\bar{hl}} \cap \{\cup_i \ S^i\})}{A_{\bar{hl}}}, \tag{2.4}$$

where A_{cov} is the fraction of area covered, $A_{\bar{hl}}$ is the field without headlands, S^i is the i^{th} swath, $\cap$ is the intersection operator, and $(\cup_i \ S^i)$ is the union of all the swaths.

- Minimize the Swaths Length:

$$\sum_i^N length(S^i) = \sum_i^N \sum_j^{S_p^i - 1} ||S_{p=j+1}^i - S_{p=j}^i||_2, \tag{2.5}$$

where $\sum_i length(S^i)$ is the sum of the length of the swaths, N is the number of swaths, S_p^i is the number of points that the i^{th} swath has, $S_{p=j}^i$ is the j^{th} point of the i^{th} swath, and $||x||_2$ is the Euclidean norm.

2.2.3 Route Planner module

The Route Planner module uses the swaths created earlier to produce the route (see Module 3 in Fig. 2.1). Fields2Cover contains several predefined route patterns, which include the boustrophedon pattern, the snake pattern, the spiral pattern and a custom pattern. The Boustrophedon pattern covers the swaths sequentially, and the Snake pattern skips one swath each time to traverse the field in one direction and returns through, covering the uncovered swaths. The Spiral pattern is a variation of the Snake pattern, that sorts the swaths in clusters of a fixed size with the snake pattern. The custom pattern requires specification of the swath order by the user. To compare different routes, the library provides as objective function the length of the path generated by the Path Planning module. It also computes the path length with in-place turns, which correspond to zero turning radius. The path length of in-place turns is computed as :

$$L_0 = \sum_{i=1}^N length(S^i) + \sum_{i=2}^N ||S_{p=1}^i - S_{p=M}^{i-1}||_2 \tag{2.6}$$

where L_0 is the path length with in-place turns, $\sum_{i=1}^N length(S^i)$ is the sum of the lengths of the swaths, N is the number of swaths, $S_{p=1}^i$ is the first point of the i^{th} swath, $S_{p=M}^i$ is the last point of the i^{th} swath, and $||x||_2$ is the Euclidean norm.

2.2.4 Path Planner module

The inputs of the Path Planner module (see Module 4 in Fig. 2.1) are the route (sorted swaths) and the vehicle parameters. Once the route is known, the turns to complete the path are computed. In the current version of the library, the path planner applies the same type of curves for all the headland turns. Fields2Cover currently supports straight curves, the Dubins' curves (Dubins, 1957) and the Reeds-Shepp's (Reeds and Shepp, 1990) curves, using the path length as the single objective function.

2.2.5 ROS wrapper

Although the Fields2Cover library does not depend on ROS, an interface with ROS1 and ROS2 is provided as an add-on. The fields2cover_ros[2] package provides functions that convert Fields2Cover data types into ROS messages. Services are created to execute modules directly from ROS topics. Launch files are used to script examples of the package. RVIZ-support is also provided to visualize the results of the modules. Methods, objective functions and parameters can be modified in real time thanks to *rqt_reconfigure*[3].

2.2.6 Design & Implementation

Fields2Cover is implemented using C++17, with a Python interface using Swig (Beazley et al., 1996), and released under BSD-3 license. The design of Fields2Cover aims to serve both scientists and service providers, and is intended to be easily used.

The reason for making Fields2Cover an open-source library is that doing so encourages the development of additional functionality by providing the code to the community. Likewise, Fields2Cover widely employs open-source libraries from third parties to streamline the development process of state-of-the-art algorithms. For scientists, priority is given to a flexible design, which allows to extend or modify existing algorithms. Additionally, a benchmark against which to compare new solutions is added. For service providers, utility concerns the ability to plan the best coverage path for a given objective function in a straightforward manner. The modularity of Fields2Cover is key to ensure its usefulness for both cases. In addition, the library provides tests, tutorials, and extended documentation[4] to reduce the learning curve.

2.3 Results

Several experiments were conducted to demonstrate the functionalities of Fields2Cover. Firstly, coverage paths were created for convex fields from the Nilsson's benchmark (Nilsson and Zhou, 2020). In these simulations, the experiments focus on the optimization of the

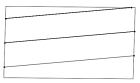
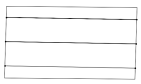
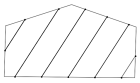
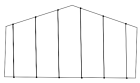
[2]https://github.com/Fields2Cover/fields2cover_ros
[3]http://wiki.ros.org/rqt_reconfigure
[4]https://fields2cover.github.io/

Figure 2.2: The AgBot 5.115T2, from the company Agreed B.V (The Netherlands), is a differential robot with continuous treads. The weight of the robot is 7.8t, the total width of the robot is 2.5 m, the minimum turning radius avoiding excessive soil damage is 2.1 m. For the experiments, the operational width of the robot (width of the coupled tool) was assigned the same value as the width of the robot. The AgBot 5.115T2 has 4-cylinder Deutz Diesel Engine, stage 5 with 156hp, and an electric drive train with a maximum speed of 13.5 km/h. Some onboard sensors are 2 cameras, a RTK-GNSS receiver and an IMU.

objective functions and the computation time of those methods. Secondly, real field experiments were conducted in an agricultural field with a commercial robot (Fig. 2.2) of the company AgXeed B.V (The Netherlands). The aim of the experiment was to program the coverage trajectory of the robot using the Fields2Cover library and assess whether a designed coverage path is efficiently traversed by the robot. The planned path is previously transferred to the robot with Protobuf (Google, n.d.). The protobuf message defines the path as timestamps, positions, velocities, and orientations. It also contains the geometry of the field boundary to prevent the vehicle from leaving the field. The sensor data collected during the coverage path, such as the GNSS position and the velocity, is returned from the AgBot as a rosbag (Field et al., n.d.).

Experiments were done with a laptop MSI GF627RE with Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz (4 cores, 8 threads) with Ubuntu 20.04.5.

Table 2.2: Comparison of swaths generated using brute force optimizing one of the three objective functions: sum of swath lengths (minimization problem), number of swaths (minimization problem) and field coverage (maximization problem). The parallel lines inside the field are the centers of the generated swaths.

Field Name	Swath length	Number Swaths	Field Coverage
REC_A			
CIR_B			
SAL_B			

2.3.1 Simulation results

Three simulation experiments were performed. Firstly, the optimal route was computed for three different fields to visually inspect the effects of the objective function. Secondly, the coverage path was computed for 38 convex fields with every possible combination of the algorithms provided by the library. The combination of algorithms for creating a coverage path were compared using the path length as the objective function. Thirdly, the time for computing coverage paths was recorded using several objective functions of the Swath Generator module. The relationship between the area of the field and the computation time was found.

The first decision for coverage path planning of a field is the objective function to be optimized by the swath generator (Brute force algorithm). The optimal pitch angle of the swaths may vary with the chosen objective function. Therefore, the first experiment provides examples of optimal swaths for the fields *REC_A*, *CIR_B* and *SAL_B* from the Nilsson and Zhou (2020)'s benchmark, which are shown in table 2.2. The fields were re-scaled to an area of $100m^2$. If the number of swaths is minimized, the number of turns is also reduced. For instance, fields *CIR_B* and *SAL_B* are covered using a single turn. If maximum field coverage is to be achieved, *CIR_B* needs seven turns, while *SAL_B* needs five. Field coverage is typically achieved when swaths are parallel or perpendicular to one of the edges. In contrast, the swath-length objective function may produce many short swaths (bottom-left of *CIR_B* with swath length), that reduce the total length of the swaths.

The second experiment was conducted using 38 convex fields of the benchmark of Nilsson and Zhou (2020), re-scalated to an area of 1 ha (Fig. 2.3). For each field, a headland of $7.5m$ (three times the operational width of the robot) was generated with the constant width generator. Next, the brute force algorithm was used to generate the optimal swaths

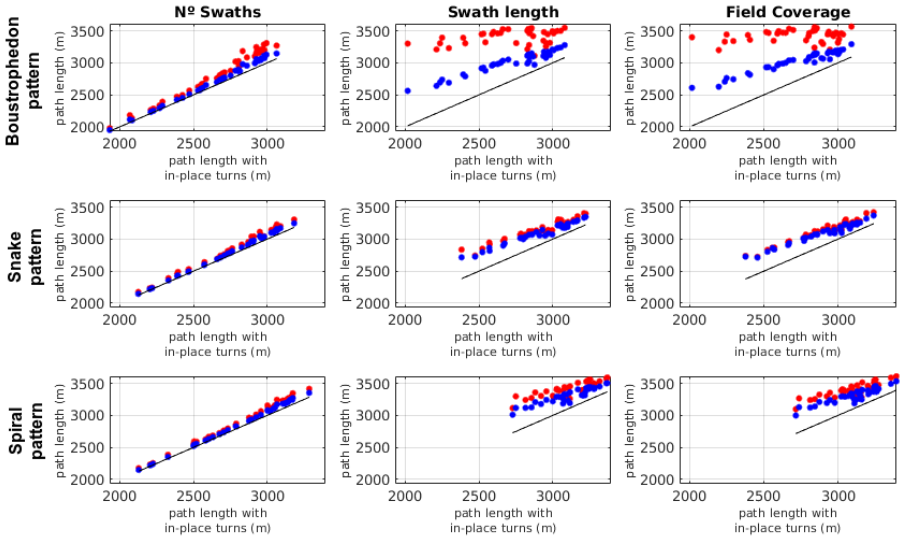


Figure 2.3: Coverage path length comparison. Columns refer to the objective function that was optimized using the brute force algorithm, while each row refers to a particular route planner pattern. Each subplot represents the optimized path length (y-axis) with respect to the optimized path length for zero-radius turns (x-axis). A single dot represents a coverage path length for a field with a chosen size from the benchmark of Nilsson and Zhou (2020). Red dots correspond to paths using Dubins' curves, blue dots are for Reeds-Shepp's curves. The black lines show the 1:1 relations.

for each objective function criterion shown in Table 2.2. The route planners sorted the swaths with the boustrophedon, snake, or spiral (bulk of 6 swaths) pattern. Lastly, the computed path length (L_R) was used for comparing the coverage paths computed with Dubins' and Reeds-Shepp's curves against the length of paths with in-place turns (L_0), which have the least possible path length for a holonomic vehicle. Each column of Fig. 2.3 refers to the objective function criterion that was optimized by the swath generator, and each row denotes a particular route planner pattern. A subplot represents the optimized path length (y-axis) with respect to the optimized path length for in-place turns (x-axis). A single dot represents a computed coverage path, with position (L_0^i, L_R^i) for the i^th coverage path. The color of the dots denotes the type of curve (Dubins or Reed-Shepp), while black lines represent the 1:1 relation. Greater values for L_0 imply that the route generated is longer and distance between swaths is larger. The difference between L_R^i and L_0^i manifests the length of the turns. This difference relates to the time that the machinery is non-productive. Therefore, a substantial difference between the black line and the colored dots denotes a path for which turning takes more time. As shown in Fig. 2.3, a percentage between 0.5% and 50% of the coverage path was spent on turns. When the number of turns is reduced, the distance traveled is reduced accordingly. The distance used for turning increases when the boustrophedon pattern is applied, since a shorter width between swaths requires a larger turn to comply with the minimum turning radius requirement. For instance, in the first column of the figure 2.3, the difference between the

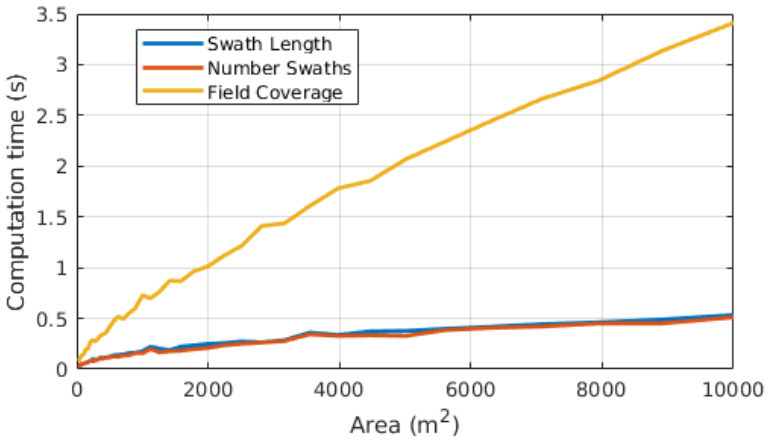


Figure 2.4: Time required to compute a path according to the objective function used. Algorithms used are constant width headland generator, parallel brute force for swath generation, Boustrophedon route order and Dubins' curves.

path length using Dubins' curves and in-place turns is smaller than in the other columns. Field coverage and swath length behaved equally in terms of coverage path length. With any of the objective functions presented, the boustrophedon pattern produced the shortest pattern with in-place turns, the snake pattern was the second and the spiral pattern the longest. The length of the boustrophedon pattern increases when the minimum turning radius is required.

In the last simulated experiment, the computation time of planning a coverage path was measured in relation to the area of the field and the objective function of the swath generator (Fig. 2.4). The constant headland width was set to three times the width of the robot. Next, the parallel brute force algorithm optimized the pitch angle of the swaths, which were sorted using a boustrophedon pattern. Finally, the path planner used Dubins' curves to create the coverage path. This experiment measures the computation time for coverage paths according to the three swath generator's objective functions in relation to the area of the field. Fields2Cover computed a coverage path for a field of 1 ha in less than 3.5 seconds using Field coverage as the objective function, while only 0.5 seconds were needed using the number of swaths or the swath length as the objective functions. Since the computation of the number of swath and swath length is proportional to the number of swaths and the number of swaths is proportional to the width of the field perpendicular to the driving direction, the computational time grows proportional to the square root of the area of the field. The computation time using the latter two objective functions can be approximated by:

$$T_c = C_0 * \frac{\sqrt{A_{\bar{hl}}}}{R_w} + C_1 \tag{2.7}$$

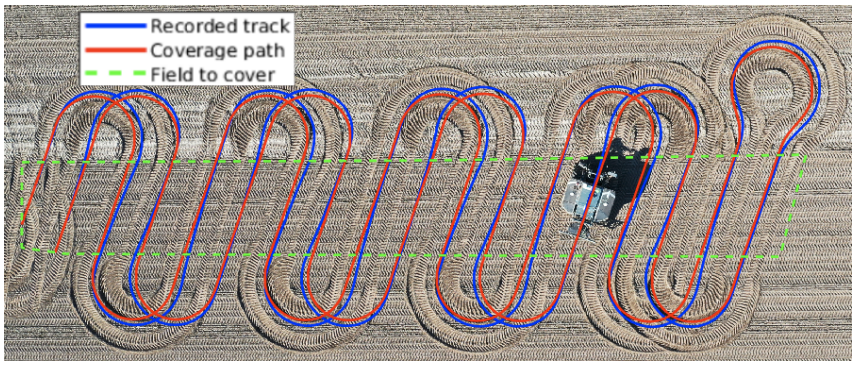


Figure 2.5: AgBot covering a narrow area (shape on green). The coverage path plan in red and the position of the AgBot in blue. The AgBot is halfway the coverage task. The starting point is near the left edge of the area.

where T_c is the computation time, C_0 and C_1 are constants, $A_{\bar{hl}}$ is the area of the field, and R_w is the operational width of the robot. This relationship is only true when the field is convex, so it can be covered with a single pattern.

The field coverage is computationally the most demanding objective function because it computes the difference between the field and the union of the areas of each swath. The computation time of this objective function grows linearly with the area of the field. Geometrical operations such as *'difference'* and *'union'* are more expensive than returning the number of swaths, which is the size of the vector of swaths.

Computational time analysis focused on the objective function of the brute force algorithm, which consumes more than 80% of the total time of the coverage path planning.

2.3.2 Field experiment

A field experiment was conducted using the AgBot shown in (Fig. 2.2). In the extreme case shown in Figure 2.5, the AgBot covered an elongated narrow area. Objective functions like the minimum swath length or the number of turns would produce swaths parallel to the longest edge of the field. However, here we show a coverage pattern given a custom angle that allows observing the turns in the field. The produced swaths were sorted using the Snake pattern and connected by Dubins' curves. The difference between the planned path and the recorded track in Figure 2.5 can be attributed to the planned minimum turning radius being shorter than permissible for the AgBot. Therefore, the recorded GNSS data show slightly wider turns than the planned path. Turns made with the snake pattern always skip one swath, except for the turn at the rightmost part of the field where the coverage direction changed. This turn is sharper, causing wider tracks on the ground, greater soil slippage, and thus more soil damage (Janulevičius and Giedra, 2009). Despite the slippage, the AgBot was capable of covering the field with the path designed by the library routines.

2.4 Conclusions & Future work

In this work, we introduced Fields2Cover, a Coverage Path Planning open-source library for agricultural vehicles. Fields2Cover was implemented to bundle the research knowledge on this topic and to help other developers to accelerate their projects. Currently, it supports the creation of coverage paths for convex fields, with a flexible and simple structure thanks to its modular design. The library has four modules, which are: the headlands generator, with a constant width headlands generator; the swath generator, with a brute force optimizer; the route planner, with three types of patterns; and the path planner, with Dubins' and Reed-Shepp's curves. The last three modules have their own objective functions specific to their domains. Fields2Cover was tested using simulation with a public benchmark and in a real field.

Fields2Cover is an ongoing project, which means the functionality of the library will be expanded in the coming years. Future developments are supported and maintained by the first author of this thesis, with the collaboration of the open-source community. Assumptions, such as flat topography, convex fields, enough maneuvering space in the headlands, absence of capacity limits and planning for arable crops, were made to reduce complexity for the first release. Non-convex fields with obstacles, sloping land, capacitated vehicles and permanent crops such as orchards provide challenges for further research and development, part of which are considered within the Fields2Cover project.

Since the release of this software library, a community has formed around the coverage path planning problem. In less than 6 months since its release, many developers have shown their support for this project through assigning 150 github stars, code contributions and by providing suggestions for future versions. This community support shows the value of the project, which emphasizes the relevance of addressing the coverage path planning problem.

Chapter 3

Fields2Benchmark: An open-source benchmark for coverage path planning methods in agriculture

This chapter is based on:

G. Mier, A. M. Casado Faulí, J. Valente, and S. de Bruin (2025a). "Fields2Benchmark: An open-source benchmark for coverage path planning methods in agriculture". *Smart Agricultural Technology*, 101156. DOI: 10.1016/j.atech.2025.101156

Abstract

The agricultural coverage path planning problem focuses on optimizing coverage paths for agricultural operations. Despite its importance, existing agricultural coverage path planning solutions are highly application-specific, limiting their generalizability and reproducibility. This chapter introduces Fields2Benchmark, an open-source, modular benchmark designed to standardize the evaluation of agricultural coverage path planning algorithms. Fields2Benchmark includes a dataset with 350 real-world fields, featuring non-convex fields and in-field obstacles. The benchmark decomposes the agricultural coverage path planning problem into five modules, i.e., headland generation, swath generation, route planning, and path planning— allowing researchers to evaluate and compare algorithms modularly. Each module supports interchangeable algorithms and objective functions, enabling customization for diverse use cases. Fields2Benchmark extends the existing Fields2Cover library by supporting capacitated operations, non-convex fields, and additional functionalities like headland width options and use case-specific swath adjustments. Outputs are recorded as structured numerical data and visual representations to facilitate detailed analysis. The capabilities of the benchmark were validated across three use cases concerning field arrangement and route and path planning with and without capacity constraints. Results demonstrate its ability to handle complex field geometries, compare algorithms effectively, and evaluate computational performance. Fields2Benchmark is computationally efficient, with planning times suitable for real-time applications. It is supported by publicly available datasets and code. By standardizing agricultural coverage path planning evaluation, Fields2Benchmark aims to improve the reproducibility in this field, accelerating the research in agricultural robotics and field operations.

3.1 Introduction

The Coverage Path Planning problem for agricultural environments (ACPP) aims to find an optimal path that completely covers a field. Coverage is crucial for tasks such as seeding, mowing, and crop protection, where efficiency and completeness determine crop yield and cost. The ACPP problem is generally solved using application-specific solutions, making them difficult to generalize (Chakraborty et al., 2022). Despite the need for comparability and reproducibility between different methods, there is no standardized benchmark for ACPP (Utamima and Djunaidy, 2022). This lack of benchmarking makes it difficult to compare the performance of different ACPP approaches (Li et al., 2022).

A standardized benchmark requires open-source code for execution and a public dataset for input (Kistowski et al., 2015). To our knowledge, there is no publicly available code to run an ACPP benchmark. The only open-source software library available for ACPP is Fields2Cover (Mier et al., 2023b), but its initial release lacked tools for solving non-convex field geometries and did not consider a robot with capacity constraints. Additionally, Fields2Cover did not include benchmark functionalities or provide a public data set for comparative analysis. Today's published datasets fail to capture the complexities of many agricultural fields. For instance, Khosravani (Khosravani Moghadam et al., 2020) provides a single convex field to compare route planners. Nilsson's dataset (Nilsson and Zhou, 2020) offers 54 manually-created geometries without in-field obstacles. Recently, Pour Arab and Essert (2024) introduced a dataset of 30 French fields, but these also lack obstacles, limiting their utility for evaluating ACPP algorithms on more complex problems.

To address these gaps, we developed Fields2Benchmark, an open-source benchmark for ACPP. Fields2Benchmark standardizes ACPP evaluation by dividing the planning process into five separate, replaceable modules: Field Decomposition, Headland Generation, Swath Generation, Route Planning, and Path Planning. This modular framework ensures reproducible comparisons and accelerates algorithm innovation in agricultural robotics. Furthermore, the benchmark is integrated in Fields2Cover[1] to encourage community adoption.

3.2 Background agricultural coverage path planning

The aim of the Agricultural Coverage Path Planning problem is to generate a path to cover an agricultural field. Those fields typically consist of headlands and the inner field. Headlands provide space for machinery to turn (Guevara et al., 2020), and they are located along the external boundaries and around large in-field obstacles (Fig. 3.1, arrow 5). The inner field is covered using swaths (Bochtis et al., 2010a), which are parallel strips of

[1] `https://github.com/Fields2Cover/Fields2Cover`

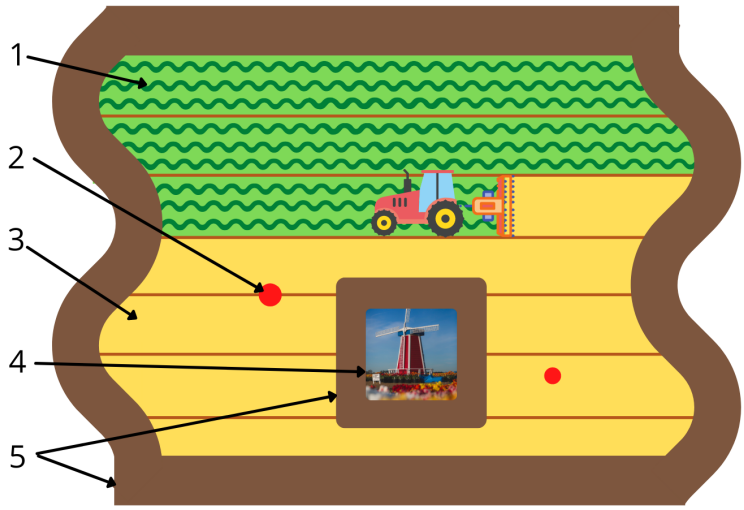


Figure 3.1: Scheme of a field being harvested. 1- Uncovered swaths. 2- Small obstacles (red dots). 3- Already covered swath. 4- Large obstacle. 5- Headlands.

land that must be covered (Fig. 3.1, arrows 1 & 3). Similarly to swaths in the inner field, headlands may also have swaths, which are called headland tracks. The goal of ACPP is to plan an efficient route –the sequence in which swaths are traversed– and then connect these swaths into a continuous path that the agricultural vehicle, or robot, can follow (Linker and Blass, 2008). When the robot has a finite operational capacity, for instance, in pesticide application or seeding, reload points become necessary to manage its resource limits (Jensen et al., 2015).

The ACPP problem is commonly divided into distinct steps (Filip et al., 2020): Field Decomposition, Headland Generation, Swath Generation, Route Planning, and Path Planning. Field Decomposition divides fields into sub-cells, which is commonly achieved by boustrophedon cellular decomposition, involving two stages: first splitting fields into smaller cells, then merging these cells into larger units. The splitting usually follows a predefined coverage direction (Bochtis and Oksanen, 2009) or extends edges from the field boundary (Nielsen et al., 2019). Optimization algorithms merge the resulting sub-cells to reduce their number while preserving navigability (Jin and Tang, 2010).

The Headland Generation step defines turning areas for agricultural machinery, either using a constant width (de Bruin et al., 2009) or minimizing headland width individually for each side to reduce wasted space (Spekken et al., 2015). Following this, Swath Generation covers the internal field area with either straight or curved paths. Straight swath generation involves selecting an optimal coverage angle, which can be identified through brute-force search (Hameed et al., 2013a), optimization methods (Oksanen et al., 2007), or constrained angle searches (Oksanen and Visala, 2009). In fields with elevation changes, curved swaths may outperform straight swaths by saving energy and avoiding challenging coverage angles.

Curved swaths can follow the longest curved field edge (Hameed et al., 2010) or they are a compromise between the curves of opposing field edges (Spekken et al., 2016).

Next, Route Planning sorts generated swaths efficiently, often employing metaheuristic optimization methods reviewed comprehensively by Filip et al. (2020). The final step, Path Planning, connects sorted swaths into navigable routes, either swath-to-swath directly (Guevara et al., 2020) or via headlands (Höffmann et al., 2022), ensuring paths have smooth, drivable turns suitable for robotic vehicles.

The above steps are executed sequentially. However, some methods (Höffmann et al., 2024; Mier et al., 2025c) aim to simultaneously solve more than one step of the ACPP problem, without following the sequential order. For example, to generate headlands (step 2) with the minimum width on each border (Spekken and de Bruin, 2013), the coverage angle of the swaths has to be calculated (step 3) before handling step 2.

3.3 The Fields2Benchmark open-source benchmark

Fields2Benchmark organizes the evaluation of ACPP in two interconnected blocks: Field Arrangement and Route and Path Planning. The Field Arrangement block involves the Field Decomposition, the Headland Generator, and the Swath Generator modules. The Route and Path Planning block comprises the Route Planning and Path Planning modules. As shown in Fig. 3.2, the framework operates as follows. First, the benchmark requires the specification of input parameters, being physical parameters and use case decisions. Next, for each module within the framework, a set of selected algorithms is initialized. Simultaneously, objective functions are defined to assess the outputs of each module. The benchmark then executes the modules sequentially, where the outputs of a preceding module serve as the inputs for the subsequent module. The benchmark outputs are recorded in a database and saved as images to ease further processing.

3.3.1 Input parameters

A wide range of applications are supported by means of a configuration step to define the use case. The settings of this configuration are classified into physical parameters and use case decisions.

Physical parameters

The physical parameters concern physical measures, such as the robot's dimensions and the geometry of the field to cover.

The robot is defined by its physical width, its coverage width –also called operational width–, and its minimum turning radius. For some operations, such as fertilization and

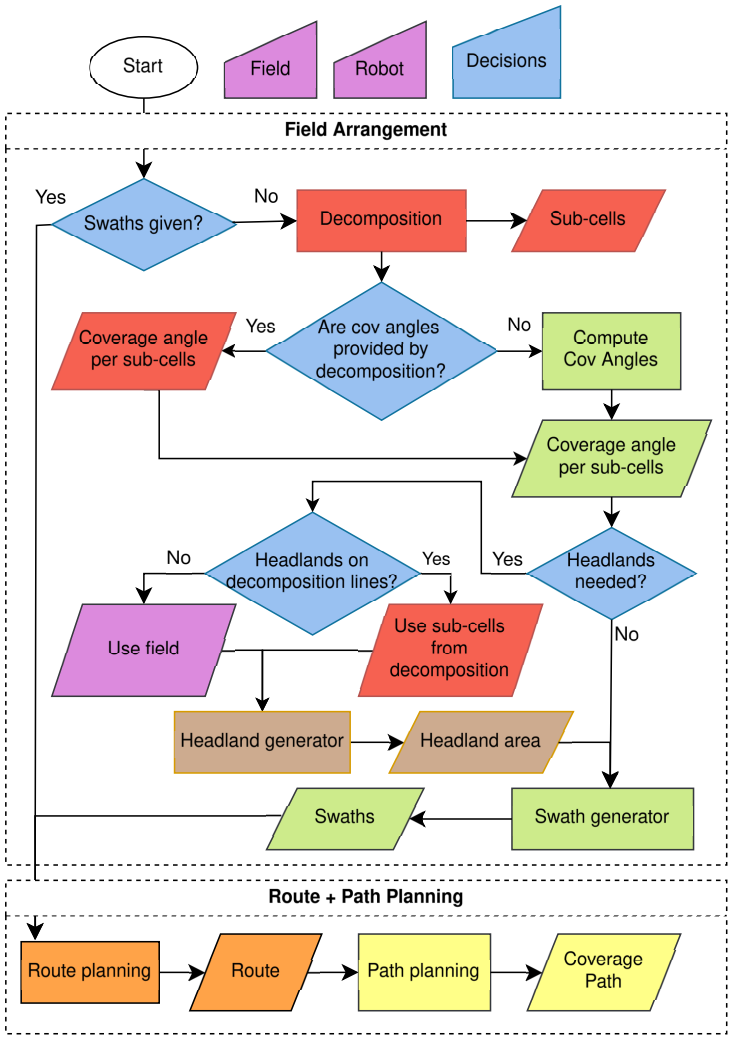


Figure 3.2: Fields2Benchmark workflow covering all processing steps from input configuration to the final coverage path output. Colors denote different inputs and modules, where purple represents the physical inputs of the robot and the field; blue indicates use case decisions; red, green, and brown correspond to the Field Decomposition, Swath Generator and Headland Generator modules in the Field Arrangement block; orange and yellow denote the Route Planning and Path Planning modules, respectively.

harvesting, the robot has a capacity that is emptied or filled during operation. The capacity is characterized by a rate, a maximum capacity, and a switch indicating whether it is used continuously (e.g., fuel consumption), or only during swath coverage (e.g., fertilization). Whenever planning a capacitated operation, the field representation also requires reloading points, which are sites used to refill or empty the robot during the operation.

The field boundaries are defined as polygons; the outer ring delineates the field's external boundary while the inner rings indicate obstacles (Kottman, 2002). The field geometry can be specified in any coordinate reference system, as Fields2Benchmark has functionality

to transform global points to a local reference system. If the points are in geographic (lat. long.) coordinates, they are first converted to UTM and then shifted to a local system whose origin is at the first point of the field's outer border. Once a coverage path is computed, the results can be reverted to the original reference system. The field definition further allows specification of the start and end locations of field operations.

Use case decisions

Apart from the specifications of machinery and field geometry, agricultural operations have specific needs, which add constraints to a general ACPP problem. For example, a lawnmower can arbitrarily cross the inner field to traverse between any two points, but a harvester would damage the standing crop when doing so. The constraints are expressed as use case decisions, which consist of:

- Are the swaths pre-established? In plantations such as orchards (Vélez et al., 2024), the swaths have already been established so that the ACPP only concerns routing and path planning. In scenarios such as orchards where swaths are pre-established, predetermined swaths are utilized, thereby omitting the Field Arrangement block from the evaluation.
- Coverage angle given or computed? When swaths are not provided but a preferred direction is known, the coverage angle may be specified to generate the swaths accordingly. Alternatively, if no coverage angle is specified, the Swath Generator module employs optimization algorithms based on several objective functions to determine an optimal angle.
- Headland width options. There are four possibilities to choose this width: 1) half of the physical width of the robot, e.g. for holonomous vehicles; 2) the minimum width required for turns; 3) the product of the robot's coverage width and an integer, e.g. to have non-overlapping headland tracks; 4) no headlands, for cases like Unmanned Aerial Vehicles (UAV).
- Headland at decomposition lines. In order to mitigate in-field turning maneuvers, headlands are generated along the decomposition lines that segment a non-convex field into sub-cells; however, their generation is not always mandatory.

3.3.2 Field Arrangement block

The algorithms and objective functions implemented in the Field Arrangement block are summarized in Table 3.1.

Decomposition module

Fields with concavities and obstacles may benefit from splitting into sub-cells, which are covered separately. Although the decomposition of fields into sub-cells can enhance

Table 3.1: List of algorithms and cost functions available in the Field Arrangement block of Fields2Benchmark, for each of the different modules.

		Algorithms	Objective Functions
Field Arrangement	Decompositions	No decomposition (ND) Trapezoidal (TD) Boustrophedon (BD)	Min-Sum-Altitude (MSA)
	Headland Generators	Constant Width (CW) Required Width (RW)	Remaining mainland area (RMA)
	Swath Generators	Longest Edge (LE) Curved Swaths (CS) Brute Force (BF) Oksanen search (OS)	Number of swaths (NS) Sum of Swath Lengths (SL) Field coverage (FC)

coverage performance, it remains an optional step in the benchmark framework. The Decomposition module offers two algorithms: Trapezoidal cellular decomposition (Latombe, 1991) and boustrophedon cellular decomposition (Choset, 2000). The trapezoidal cellular decomposition uses each vertex point of the field to generate split lines. The boustrophedon cellular decomposition is similar to the trapezoidal decomposition, but only splits in specific vertex points. Both algorithms rely on a split angle to make the decomposition. The optimal split line is searched by evaluating all angles between 1 and 360 degrees with a step of 1 degree, which can be adapted.

The objective function provided in this module is the Minimum Sum of Altitude (MSA) (Huang, 2001). The altitude is defined as the widest width of a polygon, which is proportional to the minimum number of swaths that covers it. The MSA returns the sum of the altitude of each sub-cell.

Headland generator module

The Headland Generator module provides two algorithms: Constant width and required width. The constant width algorithm (Hameed et al., 2010) inwardly buffers the field borders by a value that is common for all the borders. In contrast, the required width algorithm computes the offset by equation (Spekken and de Bruin, 2013):

$$H_i = r_{rob} * (sin(\theta - \gamma_i) + 1) + w_{rob}/2 \tag{3.1}$$

where H_i is the headland width required to turn, θ is the coverage angle, γ_i is the angle of the segment i^{th} of the border, and w_{rob} and r_{rob} are the width and the minimum turning radius of the robot, respectively. The required width algorithm requires that the coverage angle be determined prior to the generation of the inner field; therefore, the angle must be either specified directly or computed via the Swath Generator module.

The headland width for the constant width algorithm is typically provided by the farmer. If not entered, an automatic criterion is used. As a means of precaution, this implementation picks the worst case scenario from Eq. 3.1:

$$H_i = r_{rob} * 2 + w_{rob}/2 \tag{3.2}$$

The distance between the first headland pass and the field borders is at least $w_{rob}/2$, whilst adjacent headland tracks are separated by the robot coverage width.

The objective function of the Headland Generator module is the relative inner-field area, which is the ratio between the inner-field area and the total area. The optimization function maximizes the relative inner-field area.

Swath generator module

Swaths are generated by offsetting a seed line at a distance equal to the coverage width of the robot until the field is completely covered. For a straight seed line, its angle — the coverage angle — determines the direction of all the swaths. As a rule of thumb, farmers commonly select the direction of the longest edge of the field (Bochtis and Vougioukas, 2008). The coverage angle can also be searched using optimization methods. Two of those optimizers are the brute force search over a discrete set of angles (de Bruin et al., 2009; Hameed et al., 2013a; Spekken and de Bruin, 2013), and the Oksanen method (Oksanen et al., 2007; Oksanen and Visala, 2009), which first identifies a set of angles, to then proceed with a greedy search around the best angles of the first step. Both optimizers require an objective function to evaluate the results. The objective functions provided by Fields2Benchmark are:

- Number of swaths. In this work the Jin approximation (Jin and Tang, 2010) is used: $N_i = L_i \frac{|sin(\theta - \alpha_i)|}{2 * w_{rob}}$, where N_i is the number of turns, θ is the coverage angle, w_{rob} is the robot width, and L_i and α_i are the length and the direction of the edge of the border, respectively. This function should be minimized.
- Sum of swath lengths. This function should be minimized.
- Field coverage, i.e., the ratio between the area covered by the swaths and the total area of the field. This is a maximization objective function.

When the borders of the field are curved segments, the swaths can be generated following a curved seed line (Hameed et al., 2010). Fields2Benchmark provides an algorithm to generate swaths following the longest curved edge. A curved edge is defined as a sequence of several straight segments in which the angle difference between the direction of two consecutive segments is smaller than a threshold. A parallel swath is generated at a distance s from a seed curve by first constructing bisectors for each pair of consecutive segments. Subsequently, a point located at distance s along each bisector is selected; the concatenation of these points yields the resulting curved swath.

Table 3.2: List of algorithms and cost functions available in the Route and Path Planning block of Fields2Benchmark, for each of the different modules.

		Algorithms	Objective Functions
Route and Path Planning	Route Planners	Convex Fields:	Route Length Path Length
		Boustrophedon	
		Snake	
		Spiral	
		Any Fields:	
		B-Patterns w/o capacity	
		B-Patterns with capacity	
	Path Planners	Dubins Reeds-Shepp	Path Length

3.3.3 Route and Path Planning block

The Fields2Benchmark algorithms and objective function of the Route and Path Planning block are listed in Table 3.2.

Route planning module

Generating a route given a set of swaths involves tasks: 1) sorting the coverage order of the swaths, and 2) connecting the swaths with turns.

In a convex field, the swaths can be sorted following a predefined pattern (Zhou et al., 2015). The patterns provided by Fields2Benchmark are: Boustrophedon, in which swaths are covered sequentially; Snake pattern, in which at each end point one swath is skipped, covering the field using the odd swaths first, followed by the even swaths in reversed order; Spiral pattern, which covers blocks of n swaths each time from opposite sides of the field towards the center. Another possibility is optimizing the coverage by minimizing an objective function, which is referred to as B-patterns (Bochtis et al., 2013).

The patterns assume that adjacent swaths in the coverage order have opposite coverage directions, so that the end of one swath and the start of the next swath are at the same headland border. However, fields with concavities or obstacles do not follow that rule, requiring optimization of the order of the swaths and their final direction. To achieve this, Fields2Benchmark uses the or-tools library (Perron, 2011) to implement the B-patterns method. This method receives the costs of driving between swaths extremes and to the start and end point, and returns the coverage order of the swaths and their direction. Note that the B-patterns algorithm relies on automated hyperparameter tuning as implemented in the or-tools library.

Minimization of the driving costs requires all possible connections between pairs of swaths in the field. A graph is used to find the shortest path between the extremes of the swaths. The headlands tracks are also included in the graph. As the extremes of the inner field swaths are not placed directly on the headland tracks, the extremes of these swaths are connected by snapping to the closest point in the headland tracks. The shortest path to traverse the headlands from one point to the other is found by the Floyd-Warshall algorithm (Floyd, 1962).

The B-patterns method has been extended to support capacitated robots. A robot capacity is considered full at the start of the route if it empties along the path, or empty if it fills. The capacity rate depends on its type; capacity may only concern operations on swaths, e.g. fertilizing, or continuous operation, e.g. fuel consumption. For each capacity, at least one reload point is required, which is a point at which the capacity returns to the state it was at the start of the route. More than one reload point can be set at the same location so as to serve multiple visits. Reload points are incorporated into the B-patterns method as optional nodes, and are connected to the nearest points on the headland tracks within the shortest path graph. Fields2Benchmark supports route planning for a robot with multiple capacities, and with one or more reload points for each capacity.

The route planning module provides two objective functions: the route length and the path length. The route length refers to the total distance of the planned route that connects all swaths within the field. The path length, on the other hand, is the total distance of the path generated by applying a path planning algorithm to the route. This includes detailed movements, such as turns and transitions between swaths, as computed by the path planning module. While the route length provides a higher-level measure, the path length captures the movement details. Both objective functions are minimized.

Path planning module

Transforming a route into a complete path involves inserting smooth connections between swaths, which take the holonomic constraints of the robot into account. When the endpoints of two swaths are on the same headland border, the turn between them can be computed using either Dubins' curves (Dubins, 1957) or Reeds-Shepp's curves (Reeds and Shepp, 1990). Both algorithms generate the shortest path between two positions using straight and circular segments with the robot's minimum turning radius. The key difference is that Dubins' curves only allow forward movements, whereas Reeds-Shepp's curves also permit reverse movements.

For connections involving transitions between different headland borders, the problem becomes more complex. Unlike turns between nearby swath endpoints, navigating along headland tracks requires determining the start and end points of each turn along the sequence. Here, the center point in each set of three consecutive points along the sequence is considered a corner. The distance from the start and end points of the turn to the

corner, d_c, is approximated as the minimum turning radius of the robot. Points at a distance d_c from the corner are generated along the intersecting segments to serve as the start and end points of the turn. Dubins or Reeds-Shepp curves are then used to compute the shortest path between these points, ensuring that the turns respect the robot's motion constraints. If the required turning distance exceeds the length of a segment on either side of the corner, the corresponding point is omitted. In cases where the turning distance for two contiguous corners exceeds the shared segment length, the non-shared segments are extended until they intersect, thereby producing a new corner. This process is iteratively repeated to ensure feasible turns.

The objective function provided by Fields2Benchmark is the total path length, which quantifies the overall distance traveled by the robot, including all turning maneuvers.

3.3.4 Benchmark outputs

The results generated by Fields2Benchmark are recorded in a structured database. Since Fields2Benchmark supports multiple fields simultaneously, the outputs for each field are saved in separate folders. These folders contain all relevant data, including both numerical metrics and visual outputs.

The metrics, which are calculated by the previously defined objective functions are logged in a file named after the experiment. Each entry includes the following information:

- Algorithm identifiers for each module in the workflow.
- Values of the objective functions.
- Processing times for each step in the workflow and for evaluating each objective function (Tables 3.1 and 3.2).
- Metadata such as the field area, the number of points defining the outer border, and the decisions applied.

Moreover, images are generated for each module, providing a graphical representation of the module outputs. These images enable a visual comparison of results alongside numerical data.

3.3.5 Benchmark implementation

The benchmark is implemented in C++17 with an object-oriented approach. Each algorithm (see Tables 3.1 and 3.2) is encapsulated as a class that inherits from a module-specific base class. This base class defines methods such as initialization, execution, and result logging. This structure ensures consistency across the Field Arrangement and Route and Path Planning blocks.

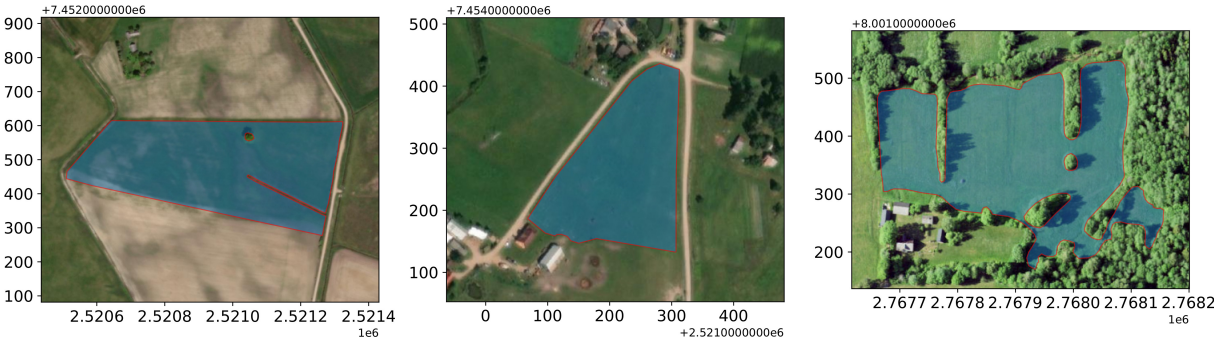


Figure 3.3: Satellite images of the fields used in the use cases 1-3, from left to right respectively. The first example features a non-convex border with an obstacle, the second shows a triangular field with a curved border to demonstrate curved swath generation, and the third presents an extremely challenging case with multiple obstacles and narrow corridors to challenge the complete workflow, including capacity route planning. Imagery source: were created using ESRI imagery (ESRI, 2025) available through the OpenStreetMap project (OpenStreetMap contributors, 2025).

Objective functions follow a similar design. A general base class captures the core concept of an objective function, and module-specific base classes extend this functionality. Each module pairs with objective functions to evaluate its performance. For every module, we maintain a collection of algorithm instances—each stored as a pointer to its base class—along with its associated objective functions.

This design, implemented from scratch based on the literature, enhances modularity and facilitates the integration of new methods.

3.4 Results

The results are organized along three use cases: 1) field arrangement; 2) route and path planning without capacity constraints; 3) route and path planning with capacity constraints. These use cases allow the analysis of each block separately, showing the capabilities of the benchmark. Each use case is demonstrated using a different field to showcase Fields2Benchmark's ability to handle complex geometries while keeping the number of figures in this work reasonable. The used fields are depicted in Fig. 3.3. These examples serve as illustrative cases rather than statistically representative samples of the 350 fields.

To further evaluate the performance of the ACPP algorithms provided with the benchmark, an experiment was carried out focusing on the processing time required for each method. These results are shown per module to provide some insights in their relationship with the inputs. This experiment was done on 350 fields, including the fields of the use cases.

Table 3.3: Selection of results from the benchmark of the Field Arrangement block of use case 1. Algorithms and objective functions complete names can be found in Table 3.1 [3].

Algorithm	MSA (m)	RMA	NS	SL (m)	FC
1) ND + CW + LE	**202.8**	0.867	79	1.76e4	0.867
2) ND + CW + CS	**202.8**	0.867	79	1.76e4	0.867
3) ND + RW + LE	**202.8**	**0.900**	79	1.83e4	0.900
4) ND + RW + (BF+NS)	**202.8**	**0.900**	79	1.81e4	0.890
5) ND + RW + (OS+NS)	**202.8**	**0.900**	79	1.83e4	**0.901**
6) BD + CW + (BF+NS)	238.5	0.764	64	1.50e4	0.741
7) TD + CW + (BF+SL)	241.3	0.480	**36**	**8.23e3**	0.406

Fields used for the experiments in this chapter were compiled into a publicly-available dataset[2]. These fields were extracted and processed from the EuroCrops dataset (Schneider et al., 2023), manually selecting fields from The Netherlands, Estonia, and Lithuania.

The robot employed was the same for all the experiments, except for the limited capacity in the use case 3. This robot has a $3m$ coverage width, a $3m$ physical width, and a minimum turning radius of $2m$. The following use case decisions apply: 1) the swaths were generated by the Swath Generator module; 2) the headland width is the minimum required, and 3) headlands are created around the decomposition lines.

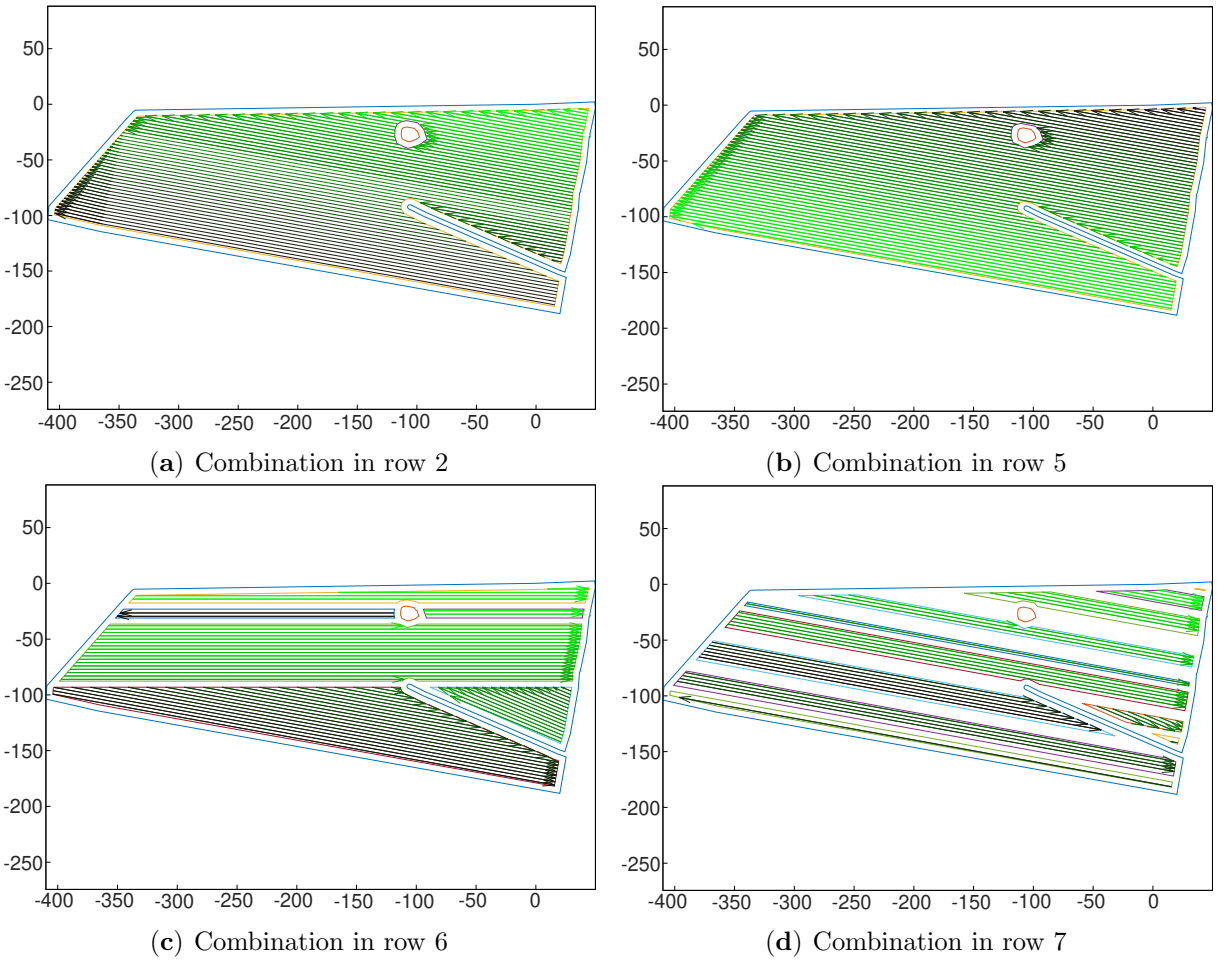
3.4.1 Use case 1: field arrangement

The first use case employed the benchmark to compare the modules of the Field Arrangement block. All algorithms in Table 3.1 were provided to the benchmark, taking into account that both brute force and Oksanen search were optimized against the number of swaths and the sum of swath lengths. Accordingly, 36 combinations were tested in this use case. In Table 3.3, seven of these combinations are shown, which were selected because they provide the best values for the objective functions for this specific use case, or because they used an algorithm not used for the better ranked combinations.

Fig. 3.4 portrays the combinations 2, 5, 6 and 7 from the Table 3.3. For this particular field, curved swaths (Fig. 3.4a) do not provide any benefit in any of the objective functions used (Table 3.1, row 2). The headlands in Fig. 3.4a and Fig. 3.4b can be visually compared, as Fig. 3.4b uses the required width algorithm, requiring less headland area. The southern border of the field benefits the most from the selection of the Required width algorithm in Fig. 3.4b. Fig. 3.4c and Fig. 3.4d show the effects of the boustrophedon and trapezoidal

[2]Dataset can be found at: `https://doi.org/10.5281/zenodo.14524735`

[3]ND - No decomposition; BD - Boustrophedon decomposition; TD - Trapezoidal decomposition; CW - Constant width; RW - Required width; LE - Longest edge; CS - Curved swaths; BF - Brute force; OS - Oksanen search; MSA - Minimum sum altitude; RMA - Remaining mainland area; NS - Number of swaths; SL - Sum of swath lengths; FC - Field coverage



(**a**) Combination in row 2

(**b**) Combination in row 5

(**c**) Combination in row 6

(**d**) Combination in row 7

Figure 3.4: Headland and swaths generated by a selection of the combinations of algorithms from Table 3.3. The XY-axis are plotted in meters, using as origin a point in the field border. The outer field border is shown in blue.

decompositions, respectively. Combination 7 in Table 3.3 (Fig. 3.4d) optimizes the number of swaths and sum of swath lengths, but it provides the worst Remaining Mainland Area value. This is because the trapezoidal decomposition generates a greater number of sub-cells, each requiring a headland around it.

3.4.2 Use case 2: route and path planning without capacity constraints

The second use case plans the route and the path of a robot without capacities to test the Route and Path Planning block. In this case, the swaths are given as generated using the constant headland and the curved swath algorithm while skipping the decomposition step. The field selected for this use case is nearly convex and without obstacles, which allows using patterns for sorting the swaths. The compared algorithms for the route planning module are the B-patterns, the boustrophedon pattern, the snake pattern and the spiral pattern with $n = 6$ as block size, and for the path planning module, the Dubins' and

Table 3.4: Benchmark of the Route and Path Planning block for the use case 2.

Algorithm	Route Length (m)	Path Length (m)
1) B-Patterns + Dubins	4390.4	4336.2
2) B-Patterns + Reeds-Shepp	4390.4	4211.2
3) Boustrophedon + Dubins	**4383.8**	4336.7
4) Boustrophedon + Reeds-Shepp	**4383.8**	**4205.4**
5) Snake + Dubins	4548.3	4374.5
6) Snake + Reeds-Shepp	4548.3	4327.5
7) Spiral + Dubins	4660.4	4522.2
8) Spiral + Reeds-Shepp	4660.4	4471.0

the Reeds-Shepp's curves. All the combinations concerning this use case are shown in Table 3.4. Paths generated by the combinations 1, 4, 5 and 8 are illustrated in Fig. 3.5 to provide further insights.

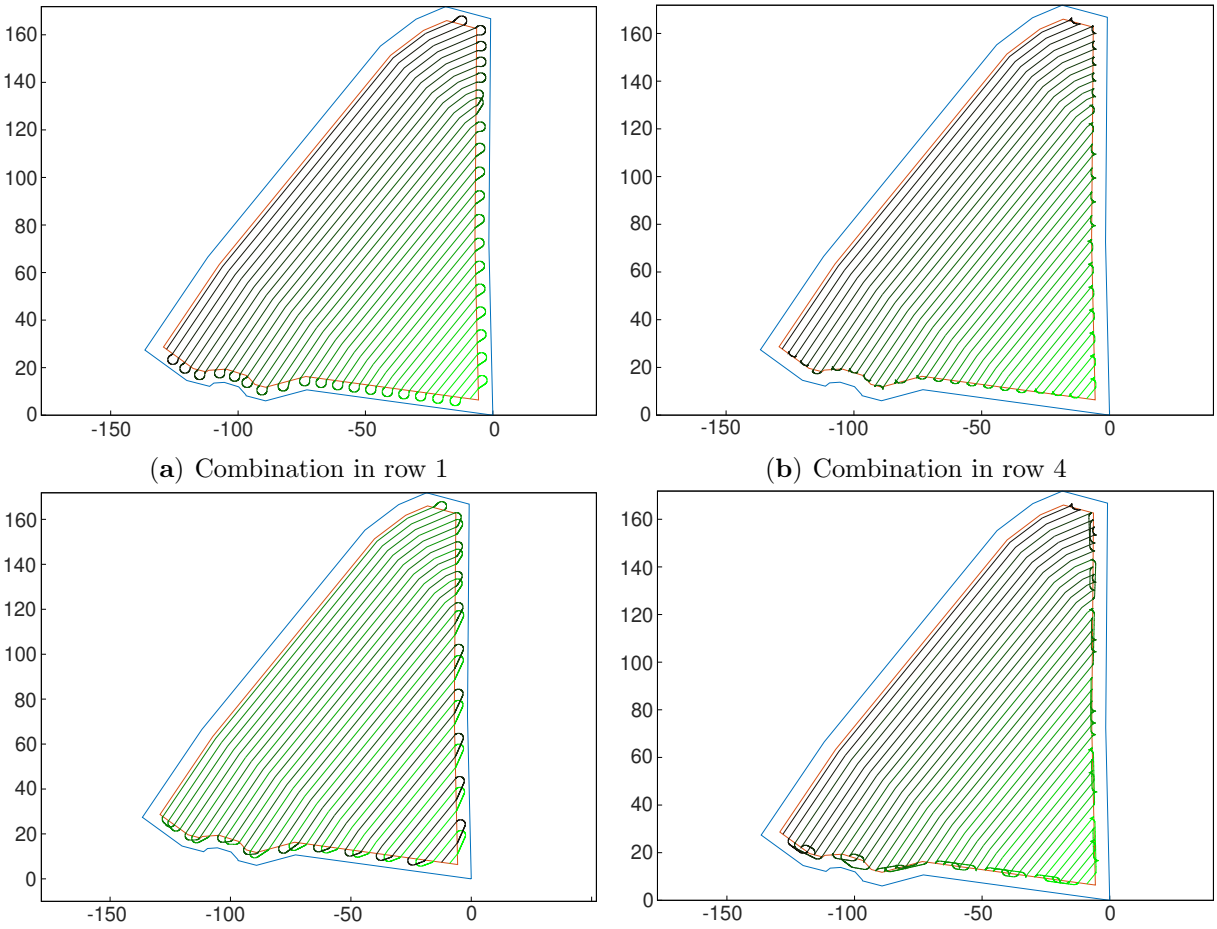
The boustrophedon pattern had the shortest route length for this specific field (Table 3.4). This result makes sense as the parallel swaths are covered sequentially, which minimizes the distance travelled through the headlands. The B-patterns algorithm should also have discovered this pattern, but it encountered a local minimum during the optimization, which prevented finding the optimal solution. The routes of the snake and spiral patterns were longer than the boustrophedon and B-patterns.

In all the combinations, Reed-Shepp's curves outperformed Dubins', but this is because the turns are partly inside the inner field. Turning in the inner field may damage the crop, making it unsuitable for many agricultural operations. This finding reveals the relevance of providing visual results by the benchmark. Simply adapting the algorithm by offsetting the swaths by a fixed amount would avoid turns inside the inner field.

3.4.3 Use case 3: route and path planning with capacity constraints

The last use case aims to plan a coverage path for a capacitated robot in a complex field. The swaths were generated using the same algorithms as for the previous use case, and then connected using the B-patterns algorithm and Dubins' curves with and without capacity constraints.

The field used for this use case and the planned coverage paths are shown in Fig. 3.6. For this experiment, the start and end points of the route and the reload point are located at different points on the outer border of the field. With capacity constraints (Fig. 3.6b), the robot has a single capacity, which is only deployed on the swaths and allows covering $3000m$. Knowing that the total swath length is around $6302.2m$, the robot has to reload twice to complete the operation.



(**a**) Combination in row 1

(**b**) Combination in row 4

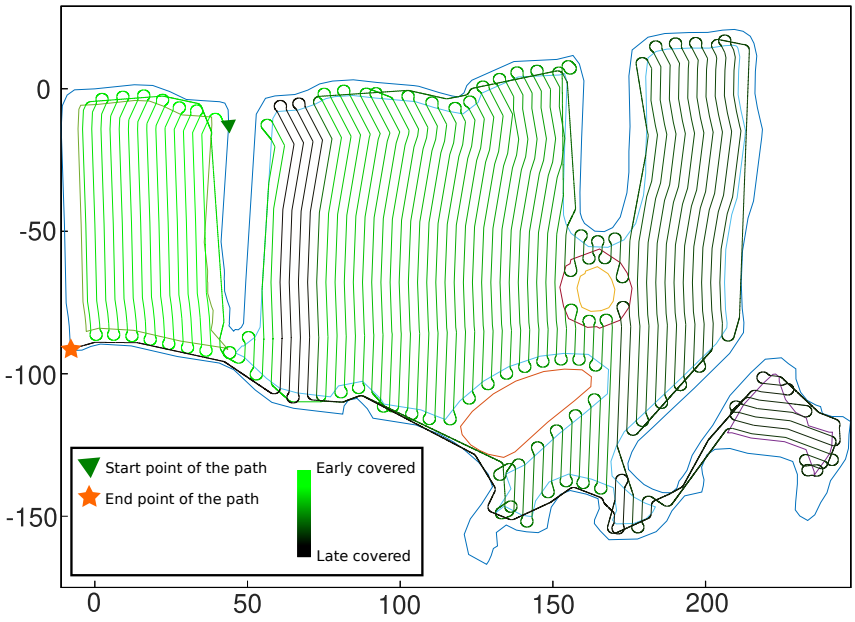
(**c**) Combination in row 5

(**d**) Combination in row 8

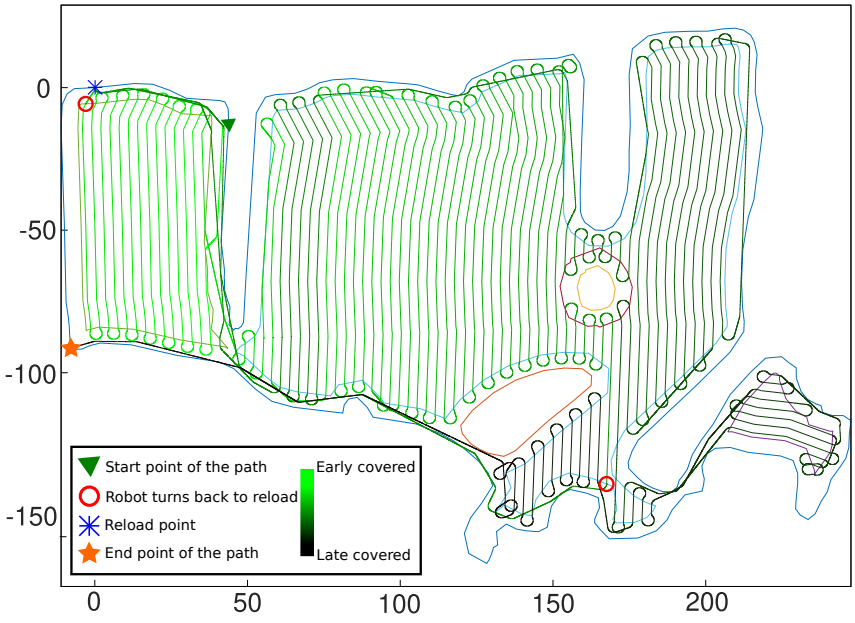
Figure 3.5: Paths generated by a selection of the combinations of algorithms from Table 3.4. The XY-axis are plotted in meters, using as origin a point in the field border. The outer field border is shown in blue.

The headland paths shown in Fig. 3.6a and Fig. 3.6b approach the field boundary closer than half the width of the robot. This problem is caused by the smoother used to simplify the headland paths, which is applied without checking whether the final path is contained within the field.

The optimized path lengths are $8359m$ and $8655m$ for the non-capacitated and capacitated problem, respectively. The difference between both path lengths is due to the cost of travelling to the reload points. The coverage order of the swaths is similar in both cases, differing from each other in the reloading paths when needed, and the coverage order of a sequence of four swaths. With capacity constraints, optimized route length is $8621m$. Contrary to the previous use case, the route length is shorter than the path length, because of the complexity of the headland passes. These results demonstrate the benchmark's capability of optimizing coverage paths for robots with limited capacity.



(**a**) Route without capacity constraints



(**b**) Route with capacity constraints

Figure 3.6: Coverage paths obtained in use case 3, using B-patterns Dubins' curves. These paths are (a) without and (b) with capacity constraints. In (b), the robot has to reload twice during the operation. The reload point is located at (0,0) and the red circles indicate where the robot interrupts the swath coverage to reload. In both cases, the start point is represented by a green triangle, and the end point with an orange star. The path is traversed from green to black.

3.4.4 Processing time

In this last experiment, the computational performance of the benchmark was analyzed. For this purpose, 350 fields were tested across all five modules. Running the benchmark experiment with all fields took 12 hours, on an Intel(R) Xeon(R) E-2276M CPU.

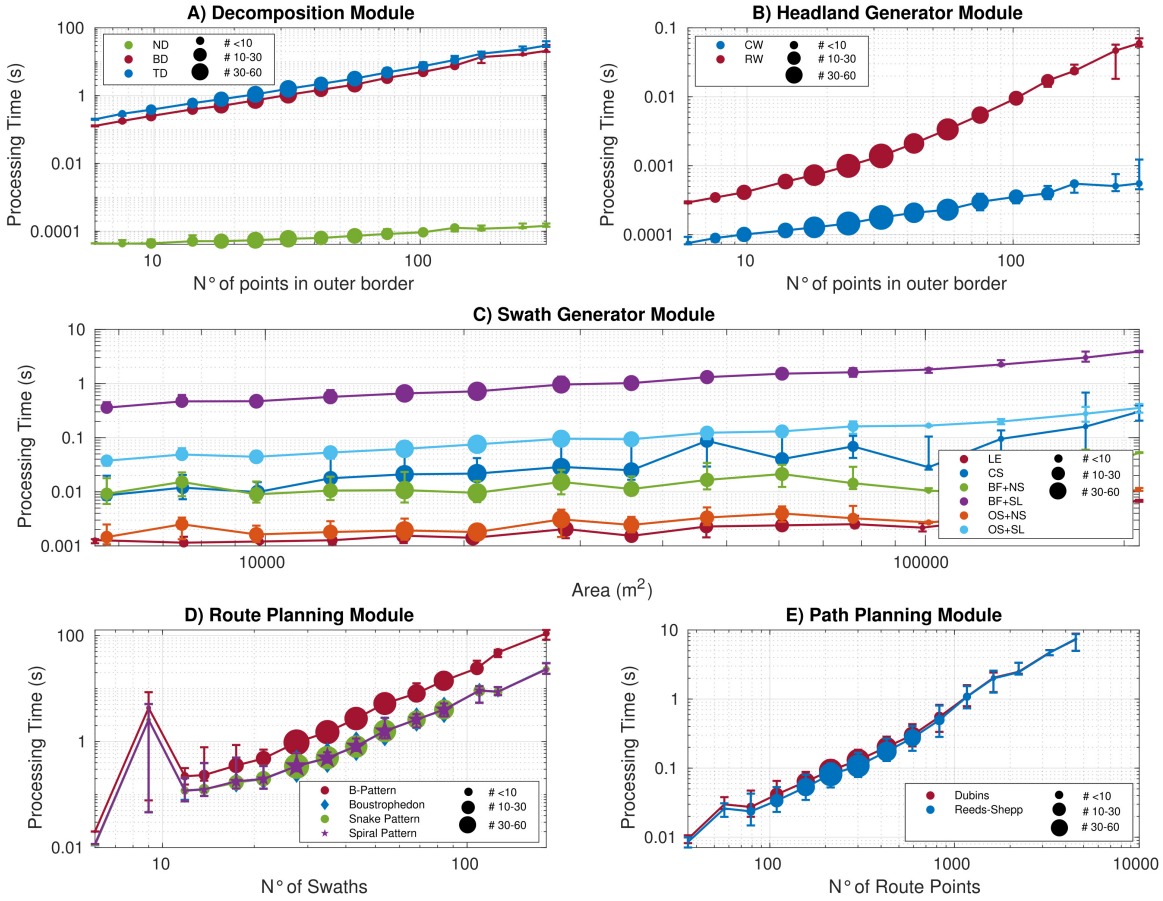


Figure 3.7: Time analysis experiments. Data bins along the x-axis represent input sizes, while whiskers indicate the upper and lower quartile of processing times. Panels A - E correspond to the Decomposition, Headland Generation, Swath Generator, Route Planning, and Path Planning modules, respectively. Abbreviations in the panels A, B and C correspond to the notation used in Table 3.1.

The observed relationships between the processing time and the inputs for each module are shown in Fig. 3.7. Only the plots showing a clear trend are presented here:

- Decomposition module (Fig. 3.7.A): the processing time of the algorithm was linearly related with the number of points in the outer border of the field. The trapezoidal decomposition was slower than the boustrophedon decomposition because more sub-cells were generated. Both decompositions also allowed skipping the Decomposition module, which is shown as "No decomposition". Interestingly, the processing time of this module was not related to the area of the field. The maximum processing time of this module for a field was 49.5 seconds by the trapezoidal decomposition, and the mean processing time was of 2.24 seconds.
- Headland Generator module (Fig. 3.7.B): For this analysis, the Decomposition module was skipped. Similarly to the Decomposition module, the processing time

of this Headland generator related to the number of points in the outer border of the field. The constant width headland generator grew linearly, while the required width algorithm increased faster, owing to the need to consider each segment of the border independently. The maximum processing time of this module for a field was 0.1 seconds by the required width algorithm, and the mean processing time was 2.5 milliseconds.

- Swath Generator module (Fig. 3.7.C): For this analysis, the decomposition was skipped, and the headland was generated with constant width. The processing time of this module increased with the inner field area. The longest edge algorithm was the fastest method, as the coverage angle was computed using only the field border. Moreover, Oksanen's search was faster than brute force owing to fewer evaluations of the objective function and the number of swaths objective function being faster than summing swath lengths. The maximum processing time of this module for a field was 5.77 seconds by the brute force optimizing the sum of swath lengths, and the mean processing time was 0.20 seconds.
- Route Planning module (Fig. 3.7.D): For this analysis, the swaths were generated with the brute force optimizer, minimizing the number of swaths from the previous module. The route planning processing time increased with the number of swaths. All algorithms, except that of the B-patterns, had similar processing times. This similarity was because the algorithms spent most time on generating the shortest path connections between swaths. In contrast, the B-patterns took more time because it relied on an optimizer. The maximum processing time of this module was 2.28 minutes using the B-patterns and the mean processing time was 4.52 seconds.
- Path Planning module (Fig. 3.7.E): This analysis used all the routes from the previous test. A trend was found between the number of route points and the processing time of the Path Planning module. There was a small difference between the processing times for Dubins and Reeds-Shepp, being Dubins a bit slower. Although theoretically Dubins should compute faster than Reed–Shepp, the timing includes the full execution of the module's algorithms. Since the benchmark times complete implementations, not just that of the core routines, Dubins' extra discretization step makes it slower than Reed–Shepp. The maximum processing time of this module for a field was 9.64 seconds using Dubins, and the mean processing time was 0.28 seconds.

3.5 Discussion

The Fields2Benchmark provides a structured framework to assess ACPP methods over diverse agricultural operations, complementing existing datasets such as those by Nilsson and Zhou (2020) and Pour Arab and Essert (2024). It offers fields of varying complexity, includ-

ing obstacles and non-convex shapes, overcoming issues of comparability, generalizability, and reproducibility identified in prior works (Utamima and Djunaidy, 2022).

Several unusual situations arose in our experiments that merit attention. In the first use case (Table 3.3, Fig. 3.4), the trapezoidal decomposition led to notably poor results, creating excessively small mainland areas despite optimizing for fewer swaths and shorter swath lengths (Fig. 3.4d). This outcome highlights an inherent trade-off in decomposition strategies previously noted but not explored deeply (Filip et al., 2020; Spekken and de Bruin, 2013).

Another unexpected result emerged in the second use case (Table 3.4, Fig. 3.5), where simpler heuristic approaches like boustrophedon surpassed more complex optimization algorithms under nearly convex conditions. Particularly, the B-patterns algorithm failed to achieve the expected global optimum. Sorting the swaths to find a coverage route is a NP-hard problem, which means that the complexity time of the solvers grow exponentially with its size. Even with a small convex field, finding the global optimum using brute-force becomes a vastly time-consuming task (Khosravani Moghadam et al., 2020). Thus, when the B-patterns optimizer stalls in a local minimum, it reflects inherent complexity of the problem rather than a flaw in the implementation. This aligns with findings from structured environment studies (Zhou et al., 2015).

In the third use case (Fig. 3.6), capacity constraints were introduced to compare how resource limitations affect coverage routes. When the robot must reload, the total path length increases. This behavior correlates with field complexity and resource layout. In Fields2Benchmark, the capability to handle curved swaths (Figs. 3.4a, 3.5 and 3.6) is shown for all the steps of the ACPP workflow, including the route planning with capacity constraints.

Processing time analyses across 350 fields revealed that route planning, especially when optimizing B-patterns, represents the primary computational bottleneck (Fig. 3.7.D). Although the planning remains fast enough for practical use in autonomous workflows (Slaughter et al., 2008), exhaustive angle searches and detailed connection computations currently may limit scalability to larger or more intricate fields.

Nonetheless, challenges and limitations remain. The current benchmark does not handle all ACPP variants, such as headland coverage or fields with undulating terrain. The assumption that soil compaction is proportional to path length may simplify reality (Mier et al., 2023a), and future comparisons could incorporate soil properties to further explore methods that minimize the impact of machinery on soil. Moreover, the path planning algorithms presented in this work –Dubins and Reeds-Shepp curves– ignore steering-rate bounds. This was done for conciseness purposes and to focus on the workflow. Nevertheless, it is noted that Fields2Cover includes functionality for the continuous versions of the here demonstrated algorithms.

The benchmark currently employs a limited set of evaluation metrics, primarily focused on path length, number of swaths, and processing time. While these metrics provide valuable insights, they overlook important practical considerations such as execution safety, machine wear, and fuel consumption, which can have significant operational impacts. The modular design of Fields2Benchmark supports future expansion, with the potential to incorporate additional metrics and features, including dynamic re-planning, multi-robot coordination, and energy consumption models, to further enhance its scope.

The proposed framework omits headland coverage. Although researchers commonly skip it in ACPP (Höffmann et al., 2024; Oksanen and Visala, 2009), headland coverage has gained recent interest. Pour Arab et al. (2022) handles headlands by performing turns with Dubins and Reeds–Shepp curves, while Mier et al. (2025b) plans turns with continuous curvature paths. Integrating headland coverage into the ACPP benchmark remains future work.

In terms of computational scalability, the current approaches which employ exhaustive angle search may not scale efficiently to larger or more complex fields. In these cases, heuristic methods –such as choosing the direction of the field's longest edge (Bochtis and Vougioukas, 2008)– can produce effective solutions even for those fields. Additionally, Fields2Benchmark functions as an offline planner and does not yet integrate real-time sensor feedback or adaptive re-planning, limiting its immediate applicability in dynamic environments where field conditions can change rapidly.

Furthermore, a wider support for other ACPP applications like orchard navigation (Guevara et al., 2020) or UAV-based sensing (Vélez et al., 2024) by using grid-based methods could further promote the adoption of the Fields2Benchmark. Although the current implementation relies on classical algorithms, this choice was intentional to provide a reliable baseline. The modular design of Fields2Benchmark permits the integration of new state-of-the-art methods. Novel techniques which are currently being explored are planned to be incorporated in future releases.

3.6 Conclusions

This work presents a standardized, open-source benchmark for ACPP, integrated with Fields2Cover to support the research community. The benchmark facilitates the comparison of algorithms across a wide range of conditions, including support for complex fields, capacitated robots, and use case decisions, like the use of predefined swaths, the selection of the coverage angle and different headland width options. By linking inputs, decisions, algorithms, and outputs, it helps researchers understand not just which methods perform best, but why.

Our benchmark further provides:

- **A general open-source ACPP benchmark tool**: Fields2Benchmark provides a documented codebase with ready-to-use tools for evaluating ACPP algorithms. It includes a dataset of 350 real-world fields, including non-convex fields and fields with obstacles. Results are recorded in database files and images to facilitate visual and statistical analysis.
- **Modular Framework**: The benchmark includes 16 algorithms and 8 objective functions, distributed across five independent modules, i.e., Field Decomposition, Headland Generation, Swath Generation, Route Planning, and Path Planning. Each module solves a single step of the ACPP problem, making its algorithms and objective functions interchangeable. This modularity enables easy integration of new algorithms and objective functions.
- **Support for use case customization**: This benchmark accommodates diverse operational constraints by accepting use case decisions to modify the general ACPP. It allows predefined or algorithmically generated swaths, adjustable coverage angles, and multiple options to determine the headland widths.
- **Integration with Fields2Cover**: To encourage community adoption, the framework of Fields2Benchmark extends the Fields2Cover library by adding support for non-convex fields and capacitated operations while maintaining the modular structure of Fields2Cover [4].

By openly releasing the code and datasets, this benchmark is expected to become a core tool for consistent evaluation of ACPP algorithms, encouraging future contributions by the research community.

[4] `https://github.com/Fields2Cover/Fields2Cover`

Chapter 4

Continuous curvature path planning for headland coverage with agricultural robots

This chapter is based on:

G. Mier, R. Fennema, J. Valente, and S. de Bruin (2025b). "Continuous Curvature Path Planning for Headland Coverage With Agricultural Robots". *Journal of Field Robotics* 42.3, 641–656. DOI: 10.1002/rob.22489

Abstract

We introduce a methodology for headland coverage planning for autonomous agricultural robot systems, which is a complex problem often overlooked in agricultural robotics. At the corners of the headlands, a robot faces the risk to cross the border of a field while turning. Though potentially dangerous, current papers about corner turns in headlands do not tackle this issue. Moreover, they produce paths with curvature discontinuities, which are not feasible by non-holonomic robots. This chapter presents an approach to strictly adhere to field borders during the headland coverage, and three types of continuous curvature turn planners for convex and concave corners. The turning planners are evaluated in terms of path length and uncovered area to assess their effectiveness in headland corner navigation. Through empirical validation, including extensive tests on a coverage path planning benchmark as well as real-field experiments with an autonomous robot, the proposed approach demonstrates its practical applicability and effectiveness. In simulations, the mean coverage area of the fields went from 94.73%, using a constant offset around the field, to 97.29% using the proposed approach. Besides providing a solution to the coverage of headlands in agricultural automation, this chapter also extends the covered area on the mainland, thus increasing the overall productivity of the field.

4.1 Introduction

In recent years, the role of agricultural automation in transforming farming methodologies has been crucial. The advent of autonomous agricultural robot systems (Fig. 4.1) has marked a significant advancement in agriculture, facilitating the enhancement of operational efficiency and crop yields (Oliveira et al., 2021). These systems autonomously and precisely traverse agricultural fields, executing tasks like harvesting (Nørremark et al., 2022; Rahman et al., 2019) and spraying (Spekken and de Bruin, 2013) as required by the specific location.

Agricultural fields are generally categorized into two main sections: the mainland, i.e., the primary area of operations, and the headlands, predominantly used for turning between the swaths covering the mainland. The optimization of mainland coverage has been the primary focus of many studies (Höffmann et al., 2023; Höffmann et al., 2024), whereas the headland areas that are crucial for comprehensive field management, have been much less studied.

Headlands are typically formed as concentric tracks, their number and dimensions varying according to the operational requirements and the space needed for tractor maneuvers (Paraforos et al., 2018). Research on headland management has mainly concentrated on optimizing turns between swaths, employing various strategies like route planning to minimize non-working time (Bochtis and Vougioukas, 2008), selecting headland turning types to decrease expenses (Spekken et al., 2015), and path planners for turns in the headland avoiding getting out of the field when the tractor is equipped with mounted implements (Trendafilov et al., 2023). However, comprehensive approaches for effective between-track maneuvers on the headlands and particularly maneuvers along corners remain sparse.

To date, most turning planners rely on circular segments (Jeon et al., 2021; Nilsson and Zhou, 2020), or non-continuous-curvature curves (Pour Arab et al., 2022), such as Dubins (Dubins, 1957) or Reeds-Shepp (Reeds and Shepp, 1990) paths, comprising sequences of circles and straight lines. These paths, however, are unsuitable for non-holonomic robotic systems owing to sudden directional changes. To mitigate this issue, Fraichard and Scheuer (2004) suggested an adaptation of Reeds-Shepp's curve using clothoids for smoother transitions. While this has been for swath-to-swath turns in agricultural fields (Sabelhaus et al., 2013), it has not been used for within-headland navigation. Also Non-Uniform Rational B-Splines (NURBs) have been proposed (Höffmann et al., 2022) for smoother headland paths, focusing solely on forward movement. However, these do not effectively cover the corners as the method focuses on path planning around the headland rather than the coverage of it. For corner coverage in headlands, Jeon et al. (2021) introduced a back-and-forth maneuver, neglecting physical constrains on speed of curvature changes.

While some studies go beyond simply considering maximum curvature and operational width (e.g., Jeon et al. (2021)), they have so far disregarded the dimensions of both the tractor and the implement for preventing overstepping field boundaries during turns. Crossing field boundaries unexpectedly poses safety risks to machinery, external property, and farmers. To our knowledge and as shown in Table 4.1, our work is the first that ensures field borders not to be crossed, thanks to a more complete representation of the robot and implement geometry. Additionally, our approach maximizes field coverage by utilizing backward movements while maintaining continuous curvature along the paths. As previously mentioned, non-continuous curvature paths are unfeasible for non-holonomic vehicles, such as tractors, which may explain why Pour Arab et al. (2022) did not test their algorithm in a real environment, and Jeon et al. (2021) experiments relied on a human driver, instead of autonomous robots.

Table 4.1: Comparison between headland coverage path planning solutions. (*) Concave corners are treated as convex corners, ignoring the field borders. (**) Using human driver.

Approaches	*Pour Arab et al. (2022)*	*Höffmann et al. (2022)*	*Jeon et al. (2021)*	**Our approach**
1. Continuous curvature paths	✗	✔	✗	✔
2. Allow reverse movement	✔	✗	✔	✔
3. Support for concave corners	✗*	✔	✗	✔
4. Implement not in center of robot	✗	✗	✔	✔
5. Coverage area ≠ Implement area	✗	✗	✗	✔
6. Multiple cost functions	✗	✗	✗	✔
7. **Guarantee field borders are not crossed**	✗	✗	✗	✔
8. Tested autonomously on real experiment	✗	✗	✗**	✔

This study builds on existing non-curvature-continuous corner path planners (Jeon et al., 2021), integrating solutions for continuous curvature turns to achieve headland coverage using real-world robotic implementations. The proposed method employs clothoids for generating smooth transitions between intersecting lines (Baykal et al., 1997), a technique commonly used in road alignment, but scarcely in agricultural path planning. We introduce a method for improving the coverage of the headland area while constraining the robot and the attached implement to stay within the field borders. To do so, a minimal safety offset for each border of the field is computed, considering the type of turn and the robot geometry and three types of continuous curvature turns for headland corners. The selection method for the corner turn planner employs two objective functions, i.e., path length and the area uncovered.

This chapter substantially contributes to the field of headland coverage path planning in agricultural operations, offering the following key advancements:

- **Continuous curvature corner path planners**: Three different types of corner turn planners (C-type, B-type, and X-type corner turns) are presented. Those corner

Figure 4.1: The AgBot 5.115T2 of the company AgXeed B.V (The Netherlands) with a mounted Amazone Cenius 3000 Super cultivator combined with a seed drill.

turn planners are suitable for both convex and concave corners while generating paths with continuous curvature.

- **Method to constrain the agricultural robot to field borders**: A thorough analysis is presented, quantifying the degree to which agricultural implements exceed field borders during corner maneuvers. Additionally, a strategy is provided to mitigate instances of exiting the field borders by offsetting the edges of the border the distance exceeded by the robotic system.
- **Automatic turn selection**: The study proposes dual objective functions – area coverage and travel distance – for selecting among C-type, B-type, and X-type corner turns. This systematic evaluation of each turn type against these objectives aims to optimize field coverage and enhance navigation efficiency.
- **Simulated and real field experiments**: The methods have been tested in simulation using the Nilsson and Zhou (2020) benchmark, and on a real field with an autonomous robot. These experiments prove the validity of the proposed methods.

4.2 Methods

The proposed algorithm is represented in the block diagram of Fig. 4.2. The inputs of the diagram are the robot data (Sec. 4.2.1), the number of headland tracks and the field shape.

The first step to plan the headland coverage path is to collect the corners (Sec. 4.2.3) and transform them to a local system (Sec. 4.2.4). For each corner, the path to cover the corner is computed, depending if the corner is convex (Sec. 4.2.5) or concave (Sec. 4.2.6). To avoid overpassing the corner edges, the central path is moved an offset inwards the field. The computation of those offsets is explained on Sec. 4.2.7. This chapter proposes three planners. Choosing the best planner for each corner requires to define a cost function. In the case of headland coverage, the objective of the turns is to cover the maximum area while driving the shortest path. This trade-off is handled using a sum of normalized cost functions (Sec. 4.2.9). After selecting the planners for the corners, the offsets are used to move inwards the field borders (Sec. 4.2.8). Finally, the turns are recomputed using the new offsets, and the headland coverage path is obtained by concatenating the turns of each corner (Sec. 4.2.8).

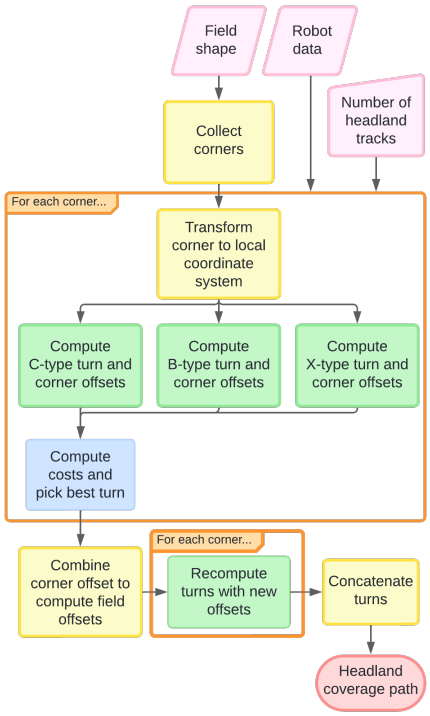


Figure 4.2: Block diagram of the proposed algorithm. Inputs are drawn in pink, geometric transformations in yellow, corner planners in green, cost functions in blue and output in red.

4.2.1 Robot definitions

The geometric model of the robotic system is composed of two rectangular components representing the robot and the implement fixed to it (see Fig. 4.3). The robotic system design assumes symmetry in the motion direction, characterized by three widths: robot's (W_R), implement's (W_I), and overall robotic system's (W_S), the latter being the larger of the former two. The center of the implement is located a distance c_I from the robot's

Figure 4.3: Autonomous robot, represented by a blue box, equipped with a mounted implement, indicated by a green box. The coverage area of the implement is highlighted in yellow. The centers of both the robot and the implement are marked with symbols 'R' (for robot) and 'I' (for implement), respectively.

center. When the implement is positioned behind the robot, c_I is negative. The center of the robot also serves as the center of rotation of the robotic system.

The robot and implement lengths are denoted as L_R and L_I, respectively. The positions of the implement front, I_f, and back, I_b, relative to the robot center are $I_f = c_I + \frac{L_I}{2}$ and $I_b = c_I - \frac{L_I}{2}$. The length of the robotic system is thus defined as $L_S = S_f - S_b$, with $S_f = \max(\frac{L_R}{2}, I_f)$ and $S_b = \min(-\frac{L_R}{2}, I_b)$.

The implement operates in two states: 'on' (active) and 'off' (inactive), with state changes involving a temporal transition. In its active state, the implement covers an area of width W_C and length L_C, centered around the implement's geometrical center. Coverage is achieved when the full working length, L_C, has passed a location. The center of the front part of the implement's coverage area is defined as $C_f = c_I + \frac{L_C}{2}$, and the center of the back part as $C_b = c_I - \frac{L_C}{2}$.

The robot is non-holonomic, meaning that turning is subject to bounded, continuous curvature (κ) and a bounded rate of curvature change (σ), reflecting operational and physical constraints. The maximum curvature, according to the state of the implement, is denoted as κ^{on}_{max} or κ^{off}_{max}, while the greatest rate of curvature change is denoted as σ_{max}. When active, the implement requires smaller maximum curvature to prevent machinery damage ($\kappa^{on}_{max} < \kappa^{off}_{max}$).

This research involves testing on an AgBot robot equipped with an implement (cultivator and seed drill). The dimensions and operational capabilities of the robot-implement combinations are summarized in Table 4.2.

Table 4.2: Dimensions and operational capabilities of the AgBot robot and the Amazone implement (cultivator and seed drill).

Robot Width	W_R	3m
Robot Length	L_R	4m
Implement Width	W_I	3.15m
Implement Length	L_I	3.6m
Coverage Width of Implement	W_C	3m
Coverage Length of Implement	L_C	3m
Implement Center Offset	C_I	-3.8m
Maximum Curvature with Implement Off	$\kappa_{\max}^{off}$	0.5 m^{-1}
Maximum Curvature with Implement On	$\kappa_{\max}^{on}$	0.05 m^{-1}
Maximum Curvature Change Rate	$\sigma_{\max}$	0.1 m^{-2}

4.2.2 Smooth turns

A smooth turn (see Fig. 4.4) is a turn without curvature discontinuities. To transition between two points with a smooth turn, clothoids are used. Clothoids, also known as Euler spirals, are defined by linearly changing curvatures along their length, making smooth transitions from straight to curved paths and vice versa, thus avoiding curvature discontinuities. A clothoid is described by the following equations (Fraichard and Scheuer, 2004):

$$\kappa_t^\eta = t\,\kappa_{max} \tag{4.1}$$

$$\theta_t^\eta = \frac{(t\,\kappa_{max})^2}{2 * \sigma} \tag{4.2}$$

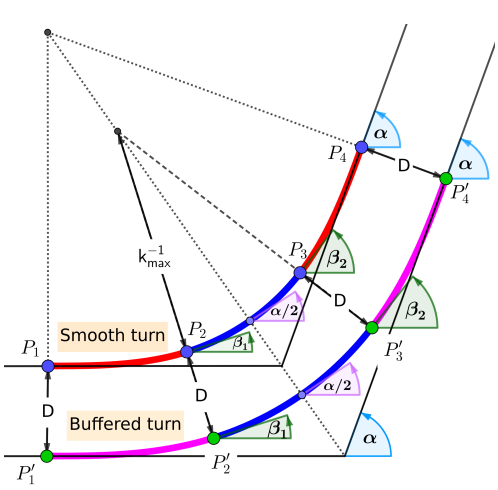
$$x_t^\eta = \sqrt{\frac{\pi}{\sigma}}\,C_f(\sqrt{\frac{(t\,\kappa_{max})^2}{\pi * \sigma}}) \tag{4.3}$$

$$y_t^\eta = \sqrt{\frac{\pi}{\sigma}}\,S_f(\sqrt{\frac{(t\,\kappa_{max})^2}{\pi * \sigma}}) \tag{4.4}$$

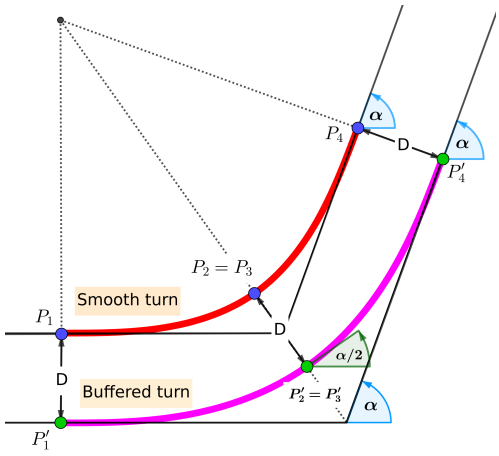
where C_f and S_f represent the Fresnel integrals, while κ_t^η, θ_t^η, x_t^η, and y_t^η indicate the curvature, angle, x-axis position, and y-axis position of the clothoid at a normalized segment length $t \in [0, 1]$, respectively.

A smooth turn starts and exits with transitions between straight lines and circular segments. However, for a turn with an angular difference α, only when the angle of the clothoid at maximum curvature (θ_{max}) is less than half of α, a circular segment of $\alpha - 2\theta_{max}$ is incorporated (Fig. 4.4a). Otherwise, the turn does not need a circular segment (Fig. 4.4b), and both clothoids (Equations 4.1-4.4) have a t equal to:

$$t = \frac{\sqrt{\alpha * \sigma}}{\kappa_{max}}$$



(**a**) Smooth turns made by two clothoids and one circular segment. Between points P_2 and P_3, the circular segment has κ_{max} curvature.



(**b**) Smooth turns made by two clothoids. P_2 and P_3 are the same, and the curvature on that point is $\leq \kappa_{max}$, so no circular segment is used.

Figure 4.4: Smooth turn and buffered turn. Clothoid are indicated in red; circular segments in blue; magenta denotes buffered clothoids. The buffering distance between both turns is always D. Both turns start with angle 0 and end the turn with angle α. From P_1 to P_2 the curvature increases, and from P_3 to P_4 the curvature decreases. As the curve is symmetric, $\alpha = \beta_1 + \beta_2$

For a left turn, the first clothoid segment uses a positive rate of curvature change, $\sigma = \sigma_{max}$, while for a right turn the negative value is used, $\sigma = -\sigma_{max}$. The sign of σ for the second clothoid segment in a sequence is always the opposite of that of the first clothoid.

This study introduces a variation of the standard clothoid by introducing a buffering distance D. This adjustment maintains the angle but alters the curvature and the x and y coordinates as described in the following equations:

$$\begin{aligned} \kappa_t^\zeta &= \frac{t\,\kappa_{max}}{D\,t\,\kappa_{max} + 1} \\ x_t^\zeta &= x_t^\eta + D\,\sin(\theta_t^\eta) \\ y_t^\zeta &= y_t^\eta + D\,(1 - \cos(\theta_t^\eta)) \end{aligned}$$

Buffered clothoids ensure parallel paths at a distance D, thus minimizing overlap between swaths while maintaining continuity in curvature properties. These buffered clothoids can replace standard clothoids in smooth turns, as calculated using the same method (Fig. 4.4).

Henceforth, this chapter will refer to smooth turns as standard left/right turns if they use standard clothoids, while those employing buffered clothoids being designated as buffered left/right turns.

4.2.3 Covering field headlands

Headlands consist of N concentric headland tracks, where each track is represented by a vector of points forming edges connected by direct lines, with the first and last points identical. A headland track cannot contain any self-intersection. The outermost headland track is the one nearest to the field border. Inner tracks are created by inwardly moving the outermost track by a distance corresponding to the robot's coverage width, W_C.

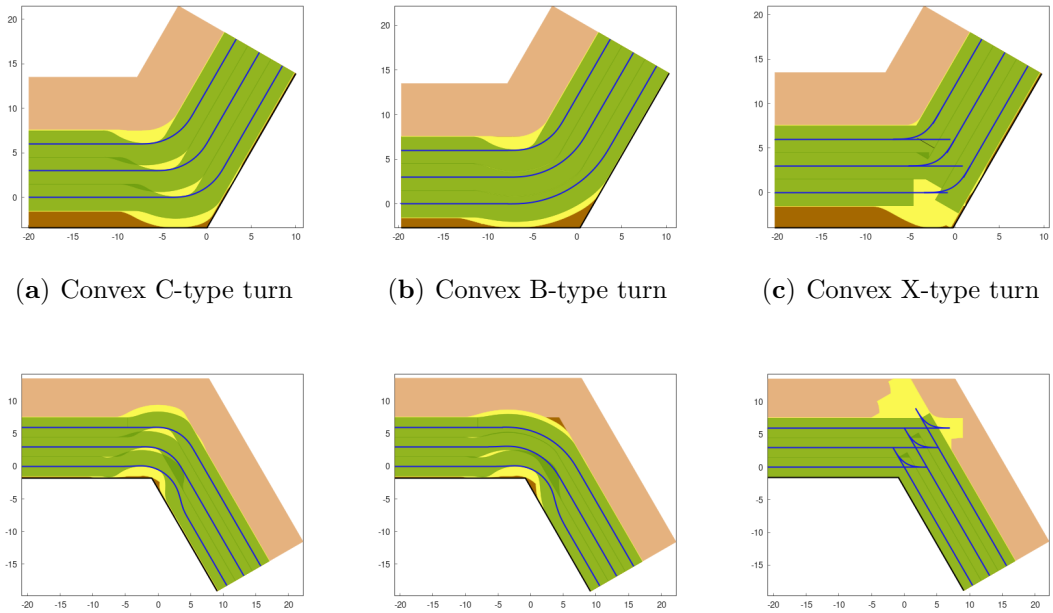
Before creating the inner headland tracks, the computation of the outermost one involves the inward displacement of each edge by a specific offset δ_j, being j the index of the edge. It is presupposed that edges are sufficiently large to prevent reduction to zero due to this displacement. The offset for each edge is calculated based on the maximum offsets at its corners, incorporating both start and end offsets (δ^{cs} and δ^{ce}). These offsets are subject to the robotic system parameters and the planners selected for each corner.

Once the offsets for each edge are determined, corner turns are computed. This entails considering the offsets and planner parameters to ensure the robot's effective navigation of each turn, thereby covering the headlands efficiently while keeping the robot and implement within field borders.

4.2.4 Corner Path Planning

For each corner in the field, a corner turn is independently planned from the rest of the field. Then, that corner turn depends only on the robot parameters and the geometry of the corner. A corner in a field is delineated by two successive edges, and characterized by three points on the xy-plane, C_1, C_2 and C_3. The corner angle, $\alpha \in (-\pi, \pi]$, represents the angular deviation between the two edges of the corner. A corner is deemed convex if $\alpha > 0$, and concave if $\alpha < 0$. The turn planning strategy diverges based on the corner's nature - convex or concave.

For convenience, the corner points are transformed by xy-shifts and a rotation to a local coordinates system where the middle point is located at (0, 0), and the first edge of the corner is parallel to the X-axis. Note that this transformation ensures that distances and angles between the points of the corner are maintained. The transformed points of the corner are denoted as C_1', C_2' and C_3'. The border of the field near the first edge of the corner, after the transformation, is located at $y' = -\delta^{cs}$, parallel to the first edge, while the second edge parallels the second border edge at δ^{ce} distance. The computed path is transformed back to the xy-plane coordinate system using the inverse transformation, first rotating and then inverting the xy-shifts.



(**a**) Convex C-type turn (**b**) Convex B-type turn (**c**) Convex X-type turn

(**d**) Concave C-type turn (**e**) Concave B-type turn (**f**) Concave X-type turn

Figure 4.5: Robot motions for corner convex turns, $\alpha = \pi/3$ (a-c), and concave turns, $\alpha = -\pi/3$ (d-f), using each type of planner for three headland tracks. Green area is the area covered. Yellow is the area traversed. Dark brown is the headland area, and light brown is the field area.

4.2.5 Convex Corner Path Planning

Convex C-type and B-type corner turns

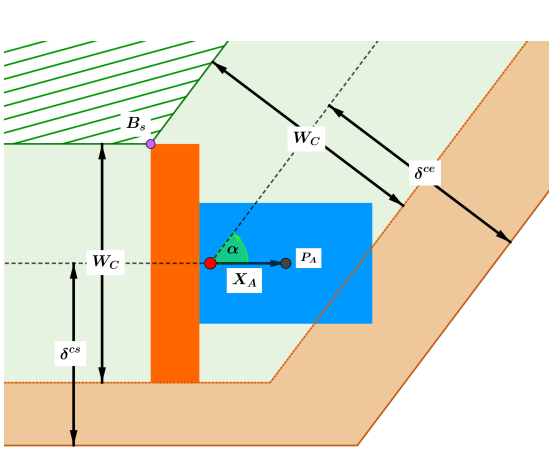
The Convex C-type corner turn (Fig. 4.5a) involves a single smooth left turn. Through the turn, the implement is continuously being used. On one hand, the time required to deactivate and activate the implement is saved. On the other hand, the turn has to be smoother when the implement is on, to avoid damaging the machinery.

This method has potential disadvantages, such as the creation of uncovered areas between swaths. To mitigate this, Jeon et al. (2021) suggested buffering the inner circular segment for subsequent corner turns. This adjustment shifts the uncovered area outside the already covered territory.

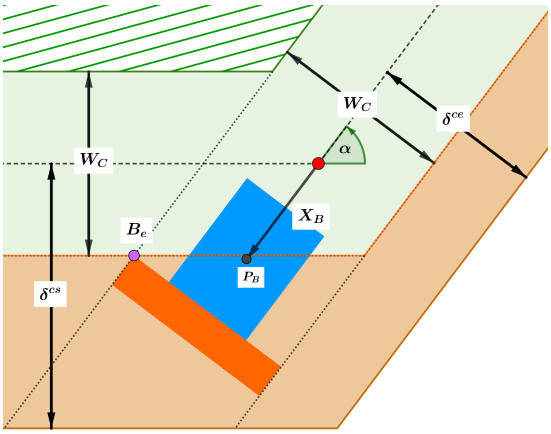
Following the approach of Jeon et al. (2021), a variation of the Convex C-type corner turn is proposed: the B-type corner turn (Fig. 4.5b). In this variant, the left smooth turn is replaced by a buffered one, using buffered clothoids (Fig. 4.4) with $D = i * W_C$, where D represents the buffered distance, and i is the number of headland swaths from the inner corner to the planned corner.

Convex X-type Corner

In the Convex X-type Corner strategy (Fig. 4.6), the robot initially advances forward following the start headland swath a distance X_A, to P_A (Fig. 4.6a). Then the implement is turned off and the robot continues moving backward until the start of the forward-moving left turn. After the backward smooth turn (Fig. 4.6b), the robot go backwards on the end headland swath a distance X_B, to P_B, turns on the implement, and ends in with forward movement to continue covering the end headland swath.



(**a**) Position of the robot at the start of the turn. The rear corner of the implement touches B_s to guarantee complete coverage of the start swath. At point P_A the robot turns off the implement.



(**b**) Position of the robot at the end of the turn. The front corner of the implement touches B_e. At point P_B the implement is turned on to continue covering the end swath.

Figure 4.6: Convex X-type corner turn. The robot is represented in blue, and the implement in orange. The green area represents the covered part of headland swaths; the brown is the uncovered headland area; the striped green area is the mainland. α represents the angle between both swaths, W_I is the width of the implement, and δ^{cs} and δ^{ce} are the offset of the turn at the start and the end swaths.

The region where the headland swaths intersect is subject to overlapping due to the robot traversing both paths. An area is only considered covered when the entire coverage area of the implement has passed over it. To reduce this overlap, given that the implement is mounted at the robot's rear, the variable X_A is selected to cover the start headland swath excluding its intersection with the end headland swath. In contrast, X_B is assigned to completely cover the end headland swath, including the overlapping region. Therefore, X_A is determined by the rear of the coverage area, while X_B is determined by its front part.

The values for X_A and X_B are calculated using the following equations:

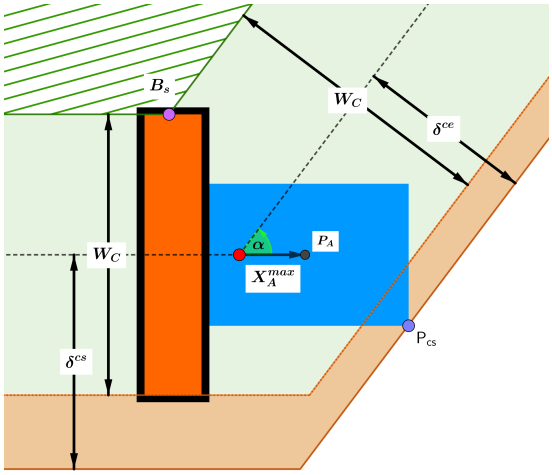
$$X_A = C_b - \frac{W_C}{2} * \left(\frac{1}{|\sin\alpha|} - \frac{1}{|\tan\alpha|} \right) \tag{4.5}$$

$$X_B = C_f - \frac{W_C}{2} * \left(\frac{1}{|\sin\alpha|} + \frac{1}{|\tan\alpha|} \right) \tag{4.6}$$

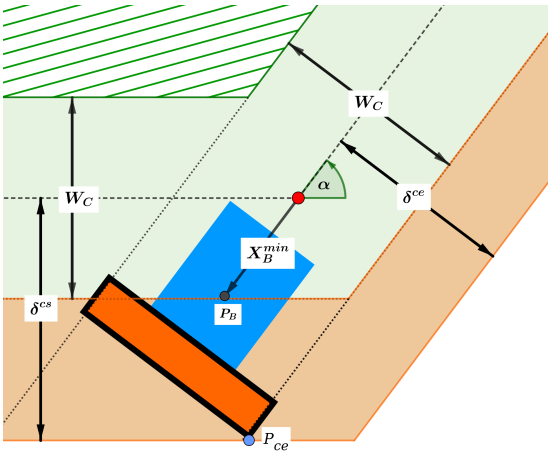
While X_A and X_B depend on the coverage area of the implement, constrains using the robot physical dimensions are applied to prevent the robot from leaving the operational area (Fig. 4.7). Specifically, $X_A \leq X_A^{\max}$ and $X_B \geq X_B^{\min}$, where:

$$X_A^{\max} = \frac{\delta^{ce}}{\sin\alpha} - \max\left(\frac{L_R}{2} + \left|\frac{W_R}{2\tan\alpha}\right|,\ I_f + \left|\frac{W_I}{2\tan\alpha}\right| \right)$$
$$X_B^{\min} = -\frac{\delta^{cs}}{\sin\alpha} + \min\left(\frac{-L_R}{2} + \left|\frac{W_R}{2\tan\alpha}\right|,\ I_b + \left|\frac{W_I}{2\tan\alpha}\right| \right)$$

If $X_A^{\max} < 0$ or $X_B^{\min} > 0$, the turn causes the robot to exceed the field's borders, requiring an increase in either δ^{cs} or δ^{ce}.



(**a**) The front of the robot touches the border of the field (P_{cs}), which limits the implement to completely cover the start swath.



(**b**) The back of the robot touches the border of the field (P_{ce}), constraining the robot movement.

Figure 4.7: Cases where the robot is physically constrained by the border of the field. The robotic system is represented as a blue box for the robot, a black box for the implement, and an orange box for the coverage area of the implement. α represents the angle between both swaths, W_C is the coverage width of the implement, and δ^{cs} and δ^{ce} are the offset of the turn at the start and the end swaths.

4.2.6 Concave Corner Path Planning

Concave C-type and B-type corner turns

The Concave C-type corner turn (Fig. 4.5d and Fig. 4.8) is a sequence of three smooth turns, that includes a left turn up to an inflection angle γ_s, a subsequent right turn through an angle totaling $\alpha + \gamma_s + \gamma_e$, and a final left turn by an inflection angle γ_e.

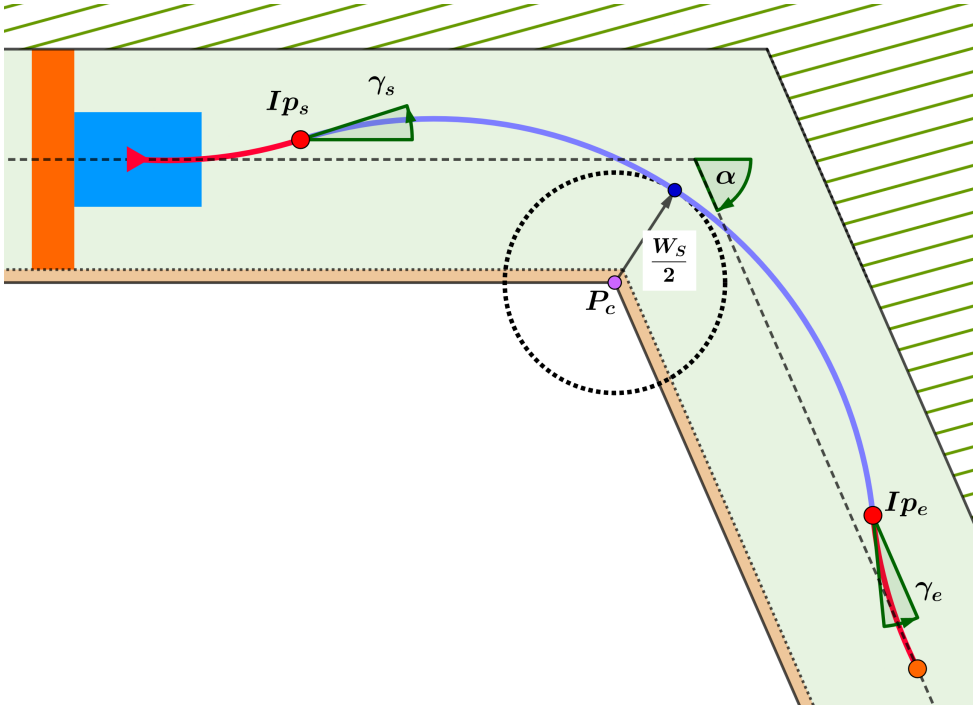


Figure 4.8: Concave C-type turn made by two left turns (red lines), and one right turn (purple line). Green area is the area of the headland swaths, brown area is the area between the border and the headland swath, and the green striped area is the mainland. The border area has a width of δ^{cs} on the starting line, and δ^{ce} on the ending line. P_C is the corner of turn. The dotted circle around P_C, with radius $W_S/2$, represents the collision region of the corner with the robot. If the path enters into the dotted circle, the robot would collide with the border of the field.

The inflexion angles, γ_s and γ_e, prevent the robot from colliding with the corner of the field. To determine the inflexion angles that minimize the length of the planned corner turn, an exhaustive search is employed, exploring angles within the range of $[0, \frac{\pi}{2}]$ in increments of 0.0125π radians. During this search, a planned path is only valid if:

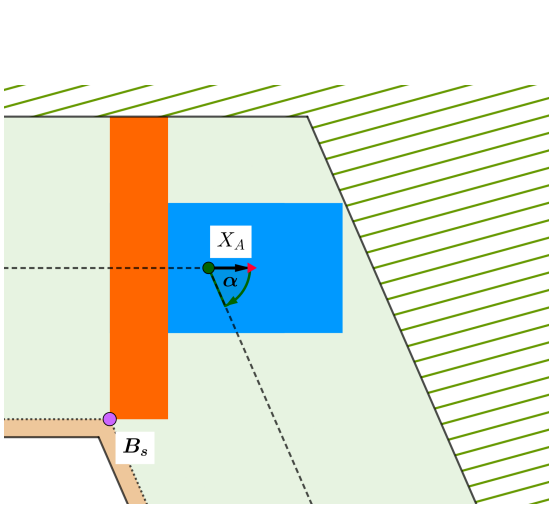
- The robot does not cross a line at $y = -\delta^{cs}$ while executing a left turn of angle γ_s.
- Similarly, it avoids crossing a line at $y = -\delta^{ce}$ during a left turn of angle γ_e, assuming that the robot is going backwards during the turn.

- The path maintains a minimum distance of half of the robotic system's width from the border corner.

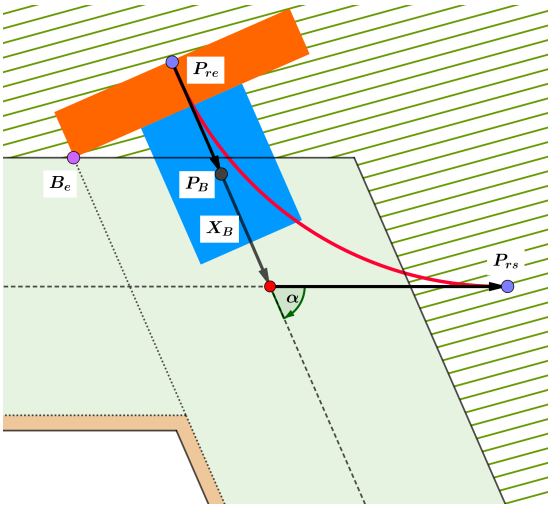
As in the convex case, a variation of the Concave C-type corner turn is proposed, called Concave B-type corner turns (Fig. 4.5e), using buffered turns instead.

Concave X-type corner turn

In a Concave X-type corner turn (Fig. 4.5f and Fig. 4.9), the robot initially advances with its implement operational. This is followed by the deactivation (lifting) of the implement at P_A and moves towards point P_{rs} (Fig. 4.9a), either forward or in reverse. Subsequently, the robot performs a reverse left turn, proceeds to point P_{re} (Fig. 4.9b) in either direction, reactivates the implement at P_B, and resumes forward motion.



(**a**) Start position of the robot, where the rear part of the implement touches B_s.



(**b**) End position of the robot, where the front part of the implement touches B_e.

Figure 4.9: Diagram of Concave X-type corner turn. The turn starts at P_A, turning off the implement. Then the robot continues until P_{rs}, does a smooth turn backwards until P_{re}, and follows forward to P_B, where the implement is turned on again.

The calculations for X_A and X_B in this scenario follow the same formulas as in the Convex X-type corner turn, specifically Equations 4.5 and 4.6. Unlike in the convex scenario, these variables are not bounded due to the lack of collision risk with the edges of the field.

4.2.7 Computing edge offset

To analyze the robot's required offsets during corner turns, a simplified path is used instead of the complete one, to reduce computations. The simplified path starts at (0,0) with zero angle, involving a left turn (can be buffered turn), and ends at angle τ. While in convex turns $\tau = \alpha$, in concave turns τ has not the same value as α.

For each corner of the robotic system along the simplified path, the minimum y-value is computed. This minimum occurs when the robot's angle $\in [0, \tau]$. As the y-value function is convex for the defined range, employing the Bisection Method or any convex optimizer is effective for finding this minimum. The minimum of all the corners of the robotic system is used as the minimum offset required by the robot to prevent crossing the borders of the field.

For the calculation of offsets δ^{cs} and δ^{ce} on convex turns, a forward smooth turn is planned from angle 0 to α. For X-type corner turns, the maximum curvature of the turn is κ_{off}, while for C-type and B-type corner turns use κ_{on} as maximum curvature.

Offsets for Concave C-type & B-type corner turns are calculated using the same method as the usual turn planner, an exhaustive search, with the path length as the cost function. During each iteration, δ^{cs} and δ^{ce} are computed for the respective combinations of γ_s and γ_e.

Regarding the offsets for the Concave X-type turn, both the start and end offsets are set to half the width of the robotic system, denoted as $\frac{W_S}{2}$. This ensures that the robot maintains a safe distance from field edges, preventing any part of the robot from extending beyond the field limits during the turning maneuver.

4.2.8 Corner turns on inner headland rings

On Sec. 4.2.4, turns are planned for the outer headland ring, which is closer to the border. Usually, headlands have more than one headland rings. The computation of corner turns on inner headland rings are equivalent to the turns on outer headland rings, using modified start and end offsets, δ^{cs} and δ^{ce}:

$$\delta_i^{cs} = (i-1) * W_C + \delta_1^{cs}$$
$$\delta_j^{ce} = (j-1) * W_C + \delta_1^{ce}$$

Being δ_i^{cs} and δ_j^{ce} the offsets of the start and end edge on the i^{th} and j^{th} headland ring, respectively. The values i and $j \in [1, N]$, being 1 the outer ring and N the total number of headland rings. Whenever the corner is made within the same headland ring, $i = j$. To change the headland ring on the corner, $i \neq j$.

4.2.9 Cost functions

Following the nomenclature on Sec. 4.2.4, a corner is defined by three points, C_1, C_2 and C_3. Adapting it to a corner with more than one headland tracks (Fig. 4.10), the points of a corner in the i^{th} headland track are denoted as $C_1^{(i)}$, $C_2^{(i)}$ and $C_3^{(i)}$. Points $C_1^{(1)}$, $C_2^{(1)}$ and $C_3^{(1)}$ define the outer corner. The points on the border of the field are B_1, B_2 and B_3. The

distance from B_1 to $C_1^{(i)}$ is δ_i^{cs}, and δ_i^{ce} from B_3 to $C_3^{(i)}$. The border of the mainland is defined by the points M_1, M_2 and M_3. Those points are placed parallel to the corner points, $\overrightarrow{C_1^{(i)}C_2^{(i)}} \parallel \overrightarrow{M_1M_2}$ and $\overrightarrow{C_2^{(i)}C_3^{(i)}} \parallel \overrightarrow{M_2M_3}$, and at a distance $|\overrightarrow{C_j^{(i)}M_j}| = W_C * (N - i + 0.5)$, for $i \in [1, N]$, $j \in \{1, 3\}$, being N the number of headland swaths.

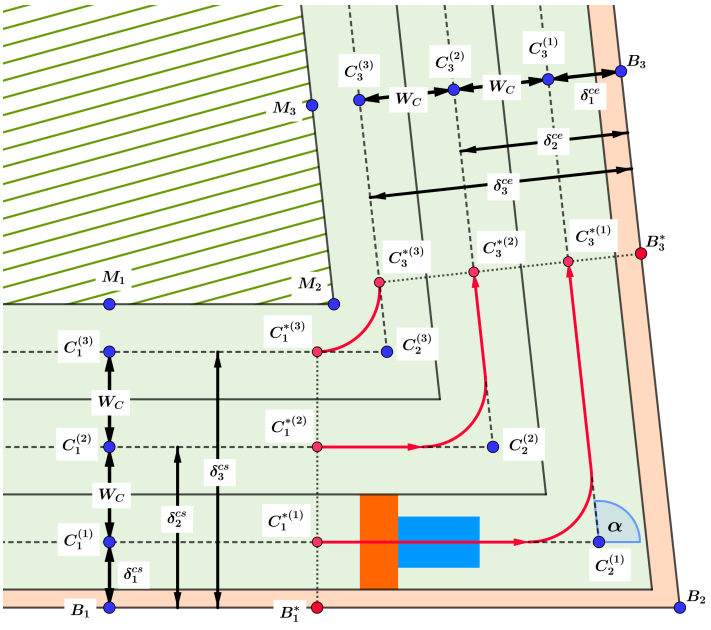


Figure 4.10: Corner with three headland tracks. Points and distances are labeled following the nomenclature used in this work. The striped green area represents the mainland and the headland swaths are on green. Between the outer border and the headland swaths there is light brown area, which is a not-covered headland area to prevent the robot crossing the field borders. The blue and orange boxes are the robot and the implement, respectively.

Additionally, to simplify the following explanation some functions and variables are used:

- $Path_\rho(C_1, C_2, C_3)$: returns the path made by the robot to drive from point C_1 to C_3, using the planner $\rho \in c, b, x$.
- $A_{cov}(Path_\rho(C_1, C_2, C_3))$: returns the region covered by the robot during the path $Path_\rho(C_1, C_2, C_3)$.
- $Area(a)$ returns the area of a polygon a.
- H: returns the region of the headland around the corner made by C_1, C_2 and C_3. This area is the polygon: (M_1, M_2, M_3, B_3, B_2, B_1, M_1).

The choice of planner for each corner is dictated by the normalized sum of the cost functions: the path length and the uncovered area.

The path length cost function sums up the length of the turn path for each headland track corner, i.e. $f_L(\rho) = \sum_{i=1}^{N} |Path_\rho(C_1^{(i)}, C_2^{(i)}, C_3^{(i)})|$. Conversely, the cost function for uncovered area calculates the area that remains unattended by the robot after executing all the turns at the headland ring corners, i.e. $f_A(\rho) = Area(H - \bigcup_{i=1}^{N} A_{cov}(Path_\rho(C_1^{(i)}, C_2^{(i)}, C_3^{(i)})))$.

$Path_\rho(C_1, C_2, C_3)$ is composed of a forward movement from C_1 to C_1^*, then a turn that finishes at C_3^*, and then another forward movement until C_3. If C_1 is closer to C_2 than C_1^* or C_3 is closer to C_2 than C_3^*, then the turn is unfeasible because $\overrightarrow{C_1 C_1^*}$ or $\overrightarrow{C_3^* C_3}$ are not forward moves. Moreover, $\overrightarrow{C_1^{*(1)} C_1^{*(i)}} \perp \overrightarrow{B_1 B_2}$, and $\overrightarrow{C_3^{*(1)} C_3^{*(i)}} \perp \overrightarrow{B_2 B_3}$.

Both cost functions depend on the length of the edges of the field, $|\overrightarrow{B_1 B_2}|$ and $|\overrightarrow{B_2 B_3}|$. The vectors $\overrightarrow{C_1^{(i)} C_2^{(i)}}$ and $\overrightarrow{C_2^{(i)} C_3^{(i)}}$ change according to the length of those edges. The computation of the cost functions can be split between the cost of the forward movements and the cost of the turn, $f(\rho, B_1, B_2, B_3) = f^S(B_1, B_1^*) + f^T(\rho, B_1^*, B_2, B_3^*) + f^S(B_3^*, B_3)$, where f is the cost function, f^T is the cost function on the turn, and f^S on a straight forward movement. Specifically, for the path length and the uncovered area:

$$\begin{aligned} f_L^S(B_1, B_1^*) &= N * |\overrightarrow{B_1 B_1^*}| \\ f_L^S(B_3^*, B_3) &= N * |\overrightarrow{B_3^* B_3}| \\ f_A^S(B_1, B_1^*) &= (\delta_1^{cs} - W_C/2) * |\overrightarrow{B_1 B_1^*}| \\ f_A^S(B_3^*, B_3) &= (\delta_1^{ce} - W_C/2) * |\overrightarrow{B_3^* B_3}| \end{aligned}$$

The function $f^T(\rho, B_1^*, B_2, B_3^*)$ does not depend on the position of B_1^*, B_2 and B_3^*, but only requires the angle α between the subsequent line segments, simplifying notation to $f^T(\rho, \alpha)$. This function has to be computed for each planner and angle, and its results can be cached to save computational resources.

To normalize the cost functions for all the angles and for a given pair of edges, the path length is divided by the length of the corners ($\sum_{i=1}^{N}(|\overrightarrow{C_1^{(i)} C_2^{(i)}}| + |\overrightarrow{C_2^{(i)} C_3^{(i)}}|)$), and the uncovered area by the area of the headland, $Area(H)$.

$$\begin{aligned} F_L(\rho, B_1, B_2, B_3) &= \frac{f_L(\rho, B_1, B_2, B_3)}{\sum_{i=1}^{N}(|\overrightarrow{C_1^{(i)} C_2^{(i)}}| + |\overrightarrow{C_2^{(i)} C_3^{(i)}}|)} \\ F_A(\rho, B_1, B_2, B_3) &= \frac{f_A(\rho, B_1, B_2, B_3)}{Area(H)} \end{aligned}$$

Note that $C_1^{(i)}$, $C_2^{(i)}$, $C_3^{(i)}$, and H depends on the edge lengths ($|\overrightarrow{B_1 B_2}|$ and $|\overrightarrow{B_2 B_3}|$) and the angle of the turn, α. A second normalization is made to scale both objective functions between $[0, 1]$ before adding them to the combined cost function, $G(\rho, \alpha)$.

$$G(\rho, \alpha) = \frac{F_A(\rho, \alpha) - \min(F_A)}{\max(F_A) - \min(F_A)} + \frac{F_L(\rho, \alpha) - \min(F_L)}{\max(F_L) - \min(F_L)}$$

The optimal planner for a corner (given angle α and edge lengths) is the one that has the smallest combined cost, while being feasible to execute.

4.3 Results

4.3.1 Experiments

Several experiments were conducted to test the algorithms developed in this chapter. The objective was to verify whether the corner turn planners effectively plan the coverage of the headlands without crossing the field borders.

Field experiments were conducted using the AgBot robot equipped with an Amazone cultivator and seed drill (Fig. 4.1). The specifications of the robotic system are given in Table 4.2. Computer code was implemented using C++ and the Fields2Cover library (Mier et al., 2023b). Matlab R2023a was used for visualization purposes. The laptop used for experiments was an MSI GF627RE with an Intel(R) Core(TM) i7-7700HQ CPU, running Ubuntu 22.04.5.

The metrics used for evaluating the algorithms' performance were:

- Absence of collision between the robotic system and the field borders: Collisions cause the robot to exit the field area. The absence of collisions proved that the planners could execute turns without crossing the field borders.
- Path length and uncovered headland area: As previously explained, these metrics focused on maximizing headland coverage while minimizing operational time.

The tests involved three experiments:

- Simulated planning in single corners: This experiment evaluated each corner planner presented in this work for individual corners. The tests concerned the influence of the number of headland tracks, the angle of the corner and the length of the edges on the election of the best corner planner. The experiment included three sub-experiments. The first studied the effect of the corner angle on the normalization process of the cost functions. The second compared the influence of edge lengths and number of headland tracks on the combined cost function. The third identified the best corner planner according to the number of headland tracks, the corner angle and the edge lengths.
- Simulated headland coverage: In this experiment, 54 fields from the Nilsson and Zhou (2020) benchmark were used to test the headland coverage path planners. Complete

headland coverage was compared using a constant offset with the best planner for each corner, a single corner planner with the minimum offsets, and the best planner with minimum offset. The benefits of the proposed methods for headland coverage were assessed.

- Real-Life Experimentation: A custom field was laid out for real-life testing, designed to challenge navigation along convex and concave turns. The field, located near Oirlo, in the province of Limburg, the Netherlands, was part of a larger test field. The field borders were marked using tape, and the corners were identified using RTK GNSS. During the experiment, the cultivator was partly lifted to have it just touch the ground, leaving a visible trail while preventing equipment damage. The path was recorded using the robot on-board RTK-GNSS. From this, the total path length was derived. An Unmanned Aerial Vehicle (UAV), with a camera and a RTK-GNSS, was used for tracking the worked area. The goal of this experiment was to assess the methodology's applicability, verifying that the robot can indeed follow the headland coverage path without exiting the field.

In the second and third experiments, the mainland was assumed to be covered using adjacent parallel straight swaths, which was not detailed in this study.

4.3.2 Simulated planning in single corners

In the first experiment, the influence of the corner angle was studied on the normalization process of each cost function for each corner turn planner. Fixed values of three headland tracks and 100 meters of length for both edges were used for this test. Those turns were evaluated on two cost functions: path length and uncovered headland area. Both cost functions were normalized using the method explained in Section 4.2.9.

Fig. 4.11 shows the costs for turns with corner angle $\in (-\pi, \pi)$, and the normalization impact on cost functions, particularly for extreme corner angles. The left side of the figure illustrates the raw cost values, i.e., path length and uncovered area, as a function of the corner angle. The right side demonstrates the effect of normalization on these cost functions. Negative angles denote concave turns, and positive angles indicate convex turns. Normalization significantly influences cost evaluation, particularly for angles substantially different from zero. For extreme cases near $-\pi$, normalization makes the large values more easily comparable.

As expected, in Fig. 4.11, X-type turns had the longest path lengths, which was due to their complex maneuvering requirements. The uncovered area in concave X-type turns was almost negligible, since the required offset equals half the difference between the coverage width and the robot width. In this example, the convex B-type turns covered more headland area than convex C-type turns for $\alpha \in [0, 0.46\pi]$. Both outperformed convex X-type turns by covering more area owing to their smaller maximum curvature, which implied

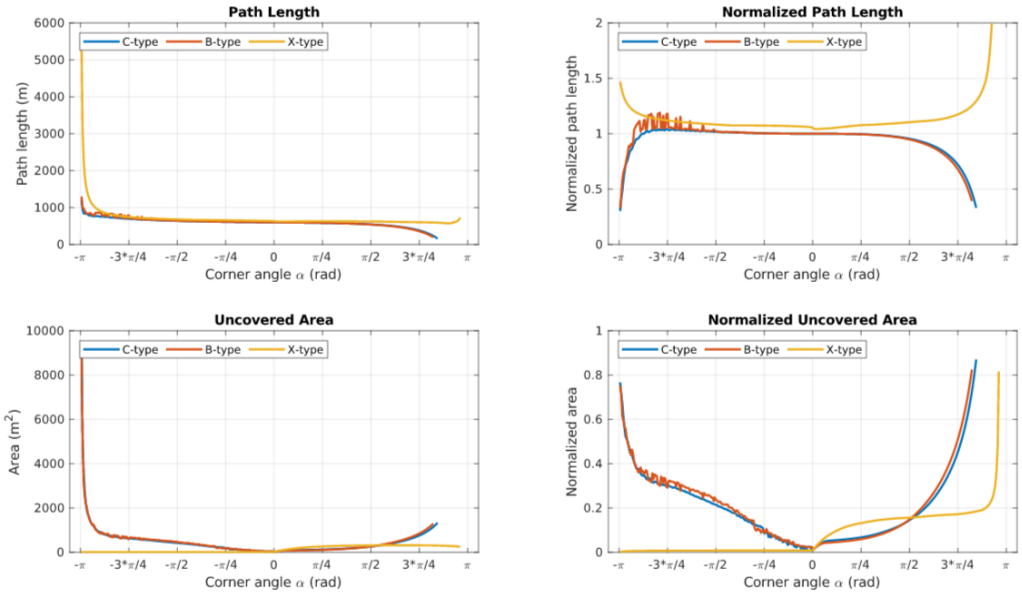


Figure 4.11: Path length, uncovered area, and normalized cost functions comparison for corner turns (X-type, C-type, B-type) with 100 m edge lengths and three headland tracks.

a smaller offset required for turning within the field limits. On the other hand, convex X-type turns covered more area for $\alpha \in [0.52\pi, 0.96\pi)$. In the range $\alpha \in (0.46\pi, 0.52\pi)$, C-type was the planner that left less area uncovered. Beyond this range, all three corner turns became unfeasible. The optimization results of the concave B-type exhibited noise due to optimization challenges. This can be particularly observed in the normalized cost functions of Fig. 4.11 and suggests potential improvements to the optimizer for finding the inflection angles.

In the second experiment, corner planners were compared using the combined cost function, across different headland tracks, corner angles, and edge lengths. Moreover, this experiment showed that for certain values corner turns cannot be planned due to space constrains. This comparison was made for 1, 2, and 3 headland tracks and for 20, 60, and 200 m of edge lengths.

Fig. 4.12 displays combined cost function values under these conditions, revealing preferences for different turn types based on the corner angle, edge length, and headland track count. The intersection points of cost function curves shifted with these parameters, indicating optimal planner choices in diverse scenarios.

When only one headland track was used, B-type and C-type turns were similar, as clothoids are not buffered. In the case where edges were 20 m, some angles were unfeasible to make using the B/C-type turns due to the lack of space required. For convex turns, B-type was preferred to C-type from angle 0 to a point where both lines intersected. For bigger angles (after other line intersection between C-type and X-type), X-type was the best planner.

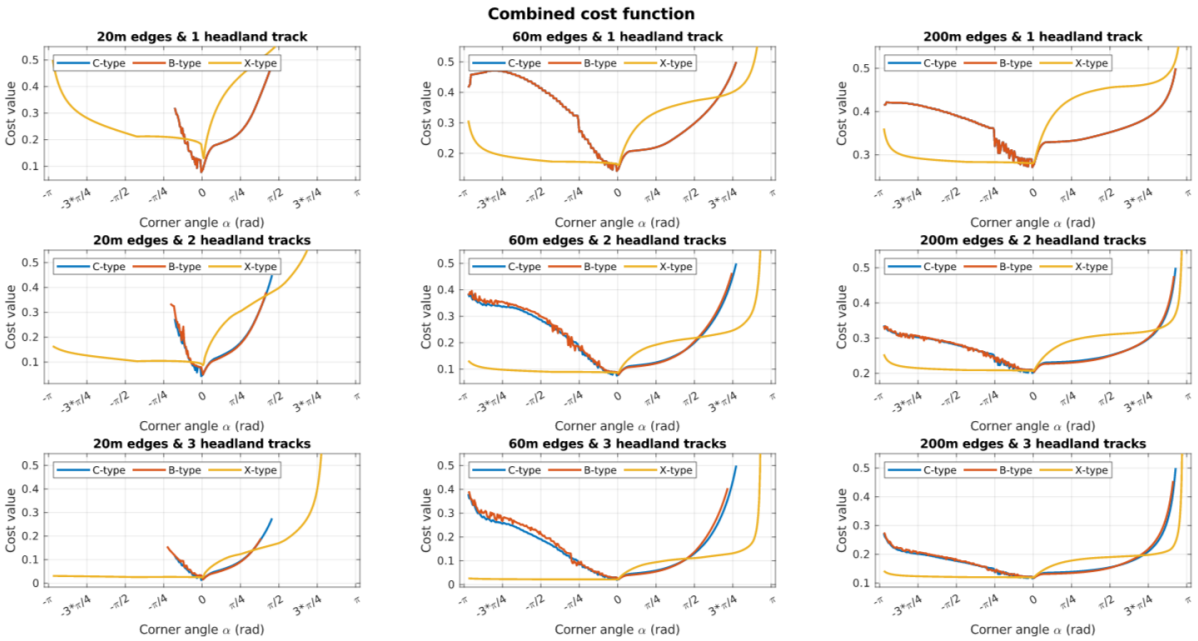


Figure 4.12: Combined cost function evaluation across various corner angles, planners, edge lengths, and headland tracks.

This behavior can also be seen in Fig. 4.11. These two inflexion points depend on the number of headland tracks and the length of the edges. Longer edges or less headland tracks moved the intersection point between C-type and X-type to bigger angles. For concave angles, there was only one intersection point between C-type and X-type. This intersection point was located near the angle 0, being preferred concave C-type only for small angles.

The third experiment (Fig. 4.13) identified optimal planners based on corner angle and edge lengths. The results corroborated with those from Fig. 4.11 and Fig. 4.12. X-type was often the preferred planner for concave corners, except at small negative angles. For convex corners, B-type was favored for smaller angles, while sharp convex corners suit X-type turns. Between the regions where B-type and X-type were preferred, C-type turns had a space in which it was the optimal planner. If the corner was too acute and edge lengths were short, none of the planners generated feasible paths.

The computing time required by each planner is shown in Fig. 4.14. Each turn was computed 5 times to obtain the average of time it takes to plan it. Due to C-type and B-type turns produced similar results, the C-type planner is not represented to simplify the plot. Moreover, the edges length did not modify the computing times of the planners. As shown in Fig. 4.14, for the same planner, the number of headland swaths proportionally increased the computing time, while for the same convexity, the corner angle did not affect it. Comparing the planners, the concave B-type planner is 10^5 times slower than the other turn planners because of its optimization process.

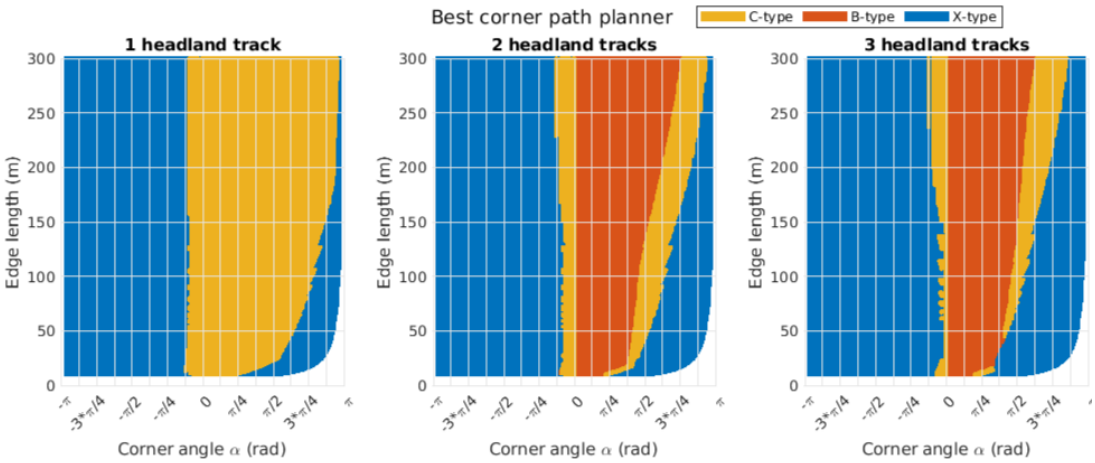


Figure 4.13: Optimal planner selection based on corner angle and edge length, indicating preferred turn types

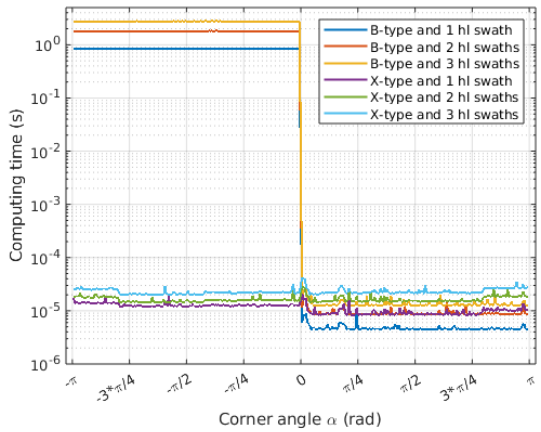


Figure 4.14: Computing time required by the turn planners to compute a turn with swaths length of 200m, according to the planner used and the number of headland swaths. Y-axis is in logarithm scale.

4.3.3 Simulated headland coverage

Despite a corner path was planned using the geometric properties of the corner instead of the whole field boundary, the yield of the field was affected by all the corner paths planned.

Fig. 4.15 showcases a field from the benchmark (Nilsson and Zhou, 2020), illustrating the impact of planner choice on each corner. The field on Fig. 4.15 was down-scaled to 1 ha for visualization purposes. The path demonstrated various turn types, with the coverage and uncovered areas distinctly marked. Starting at the bottom left corner, going in clockwise order from in-to-out order, the path did three B-type turns, followed by

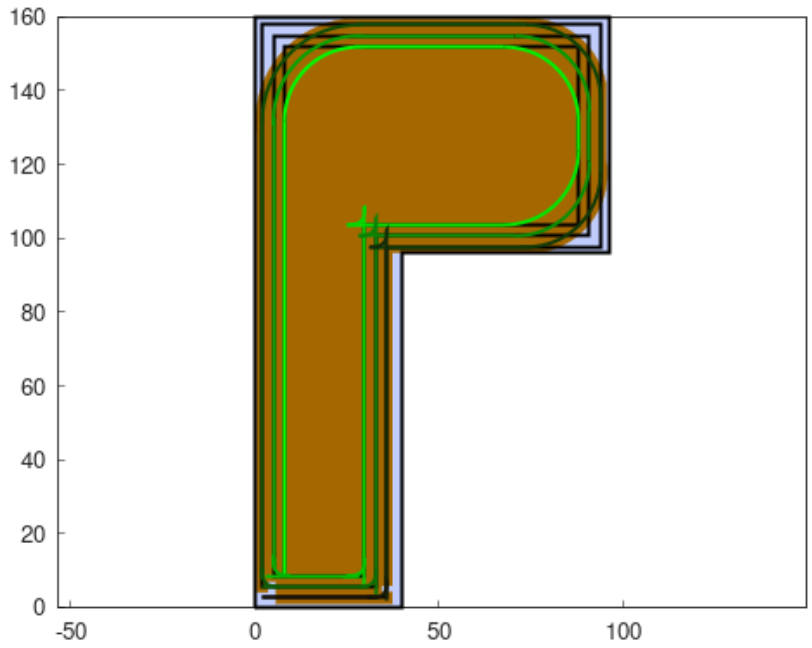


Figure 4.15: Field ELE_A from Nilsson and Zhou (2020), scaled to 1 ha, showing the coverage path of the headlands with marked covered (brown) and uncovered areas (light blue). The path color is light green at the beginning and gets darker in each new segment.

three X-type turns. This visualization aids in understanding the practical implications of different planners.

In this experiment the covered area was compared for complete fields, for three different cases:

- Constant offset. The best corner planner was chosen for each corner, but the border offsets had a constant value equal to $\delta_j = I_L$. This value was used as a safety measure to guarantee that the robot did not cross the borders.
- Single corner path planner (C-type, B-type and X-type). Only one corner planner was used for all the corners. Border offsets were computed to be the minimum required to adhere to field borders.
- Proposed approach. The best corner planners were used, computing the minimum border offset required.

This comparison allowed to analyze the benefits of each improvement presented on this work, according to the percentage of covered area of the field.

Results of this comparison are shown in Table 4.3. The approach with constant offsets had the worst coverage values, as the constant safety distance left more area uncovered

Table 4.3: Coverage area comparison on the Nilsson and Zhou (2020) benchmark with three headland tracks. Values are expressed as a percentage of covered area divided by the total area of the field.

	Const offset	C-type	B-type	X-type	Proposed approach
Max	96.62%	98.71%	**99.05%**	97.98%	**99.05%**
Mean	94.73%	96.88%	97.01%	95.76%	**97.29%**
Min	92.89%	93.24%	92.80%	94.35%	**95.21%**

near the borders. When only one corner planner was used, coverage results depended on the field geometry, being the B-type corner planner the one with best maximum and mean coverage percentages. Note that the maximum coverage for the B-type planner was equal to the maximum coverage for the proposed solution. This result was obtained for a square field, on which the B-type turn allowed the robot to approach field borders smoothly, requiring less extra space for the border offsets. Finally, our proposed approach obtained the best maximum (99.05%), mean (97.29%) and minimum (95.21%) coverage over the other approaches compared. In comparison, our approach was able to cover **2.56%** more field area than the constant offset approach.

These results confirmed the proposed approach behavior in simulations, maximizing the total covered area of the agricultural fields while observing their borders.

4.3.4 Real-Life Experimentation

The third experiment was conducted to assess the proposed methodology's applicability in real life. The AgBot robot and the Amazone implement (Fig. 4.1) were instructed to follow a headland coverage path plan. Owing to spatial limitations, the headland path comprised only a single track. A visual comparison was performed between the reference planned path and the path executed by the robot.

The experiment was conducted on June 26, 2023 under sunny and windy conditions. Figure 4.16 illustrates the experiment's outcomes. Due to the wind, the tape to mark the borders of the field bent inward along some edges. Except for three segments, the robot was able to follow the corner turns planned autonomously. Twice, the cultivator got tangled with the inward bent tape (green path in Fig. 4.16). The third time, the human operator manually steered the robot out of the field (yellow path in Fig. 4.16); this was due to a miscommunication about the size of the complete experiment. All autonomous maneuvers along the corners were successful, though. The robot successfully navigated a total path of 506.6 m without exiting the field borders, as shown by the buffered path representation in Fig. 4.16.

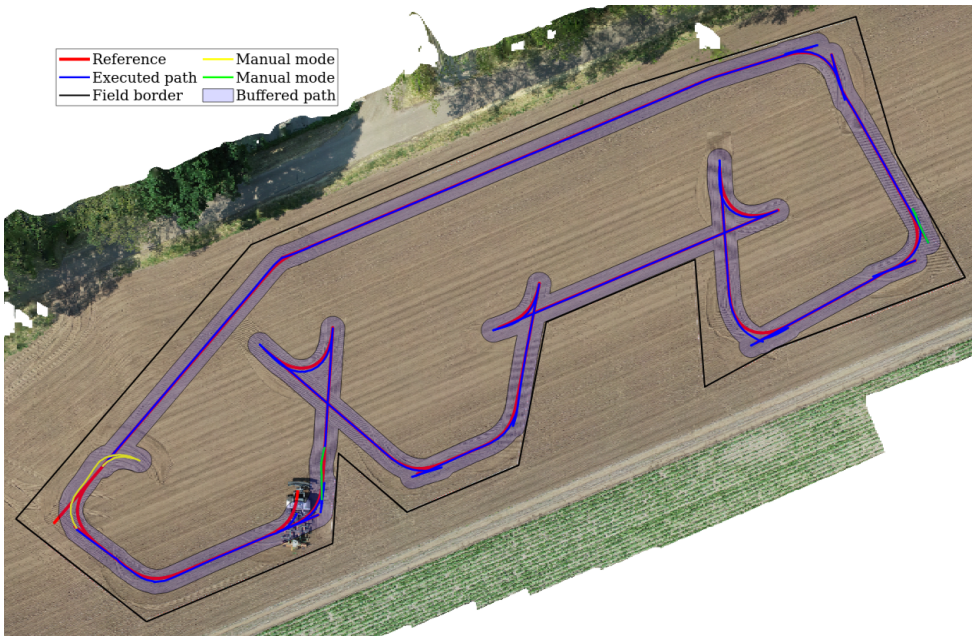


Figure 4.16: Drone image of the field experiment, showing planned and actual robot paths. The path of the robot was buffered by half of its width to verify that the robot remained inside the field.

4.4 Conclusions and Future work

In this chapter, we have introduced a headland coverage path planning approach, focusing on maneuvers along the field borders. First, three types of continuous curvature turn planners were developed for convex and concave headland corners. These corner planners employ clothoids for generating smooth transitions between intersecting lines, a technique notably used in road alignment but rarely applied in agricultural path planning. Then, a new method was presented to quantify and correct the distance that a robot requires to perform a corner turn. The method stands out for its potential to ensure the robot's confinement within field borders, especially with larger implements, by calculating a safety offset for each turn. This feature is critical in agricultural operations to prevent damage to crops and borders. Moreover, a selection mechanism for turn planners was explored, considering the area uncovered and the traveled distance as objective functions. Tests conducted with a coverage path planning benchmark and real-field trials using an autonomous robot demonstrated the efficacy of the proposed approach.

In contrast to previous literature, this work considered the dimensions of the robotic system to compute where field limits were trespassed, and offered a method to correct it. Though maximizing field coverage improves yield, some country regulations may promote or require fixed offsets near certain obstacles, i.e. ditches, limiting the benefits of our approach.

The test to measure the planners performance indicates that field size does not modify the computing time of the headland path, only the number of headland tracks and the concave C-type and B-type turns. Despite C-type and B-type planners for concave corners have a slow computing process -requiring seconds to compute-, it is still manageable on real-time environments. Selecting faster optimizer for these planners is a challenge that will be tackle in future works.

Field experiments show how the proposed offline path planning approach behaves in real-world agricultural settings with a commercial robot and implement. Other components from the robotic system, such as navigation and control, are out of the scope of this study. Future research could assess the complete robotic system – combining path planning with the robot control and navigation system. This would give the opportunity to address other challenges such as wet soil, unexpected obstacles, and imprecise global localization.

The main limitation of the proposed approach applies to corners with short edges, where the robot does not have enough space to turn safely. Future research involves research on this topic, planning the headland coverage path for corners with short edges or fields with smooth edges. Moreover, corner path planners presented in this chapter use a fixed maximum curvature to plan the corner turns. Smaller maximum curvature produces smoother turns, reducing the border offset required. An advanced approach would modify the maximum curvature of sections of the turn dynamically, to further minimize the area uncovered and the path length.

Following our commitment with the open-source community, the methods implemented for this chapter will be provided on the next major release of the Fields2Cover library (version 3.0.0) (Mier et al., 2023b).

Chapter 5

Soil2Cover: Coverage path planning minimizing soil compaction for sustainable agriculture

This chapter is based on:

G. Mier, S. Vélez, J. Valente, and S. de Bruin (2025c). "Soil2Cover: Coverage path planning minimizing soil compaction for sustainable agriculture". *Precision Agriculture* 26.4, 1–21. DOI: 10.1007/s11119-025-10250-4

Abstract

Soil compaction caused by heavy agricultural machinery poses a significant challenge to sustainable farming by degrading soil health, reducing crop productivity, and disrupting environmental dynamics. Field traffic optimization can help abate compaction, yet conventional algorithms have mostly focused on minimizing route length while overlooking soil compaction dynamics in their cost function. This study introduces Soil2Cover, an approach that combines controlled traffic farming principles with the SoilFlex model to minimize soil compaction by optimizing machinery paths. Soil2Cover prioritizes the frequency of machinery passes over specific areas, while integrating soil mechanical properties to quantify compaction impacts. Results from tests on 1000 fields demonstrate that our approach achieves a reduction in route length of up to 4-6% while reducing the soil compaction on headlands by up to 30% in both single-crop and intercropping scenarios. The optimized routes improve crop yields whilst reducing operational costs, lowering fuel consumption and decreasing the overall environmental footprint of agricultural production. The implementation code will be released with the third version of Fields2Cover, an open-source library for the coverage path planning problem in agricultural settings.

5.1 Introduction

Soil compaction poses a major challenge to agricultural land management by degrading soil health and reducing crop productivity. Heavy machinery traffic is the main cause of both topsoil and subsoil compaction, affecting around 68 million hectares globally–half of which lies in Europe (FAO, 2015). Compaction increases soil density and reduces porosity, thereby impairing aeration, drainage, and root growth (Shah et al., 2017). It also disrupts plant growth by altering enzyme activity (Wang et al., 2019). Moderate compaction has been found to reduce crop yields by 5–40% (Nawaz et al., 2023; Van Orsouw et al., 2022). Given the multitude of impacts, minimizing soil compaction is crucial for sustainable farming and climate change mitigation (Machmuller et al., 2015).

Various techniques have been proposed to combat soil compaction. Deep tillage using chisel plows or subsoilers, and practices that improve soil organic matter can temporarily relieve compaction but often incur high costs and may further degrade soil structure (Shaheb et al., 2021). In contrast, preventive strategies that limit compaction before it occurs offer a more sustainable solution. A direct method to avoid compaction is to control the movement of heavy machinery. Controlled Traffic Farming (CTF) confines machine traffic to specific lanes, thereby protecting the bulk of the field from repeated passes (Gasso et al., 2013). This practice not only protects soil structure but also enhances crop productivity and sustainability. Successful CTF implementation requires optimized machine paths to minimize soil disturbance.

Coverage Path Planning (CPP) aims to generate routes that ensure complete field coverage with minimal overlap or missed areas (Ariza-Sentís et al., 2024). Recent research has introduced various CPP algorithms to address specific agricultural challenges. For instance, Juman et al. (2017) improved autonomous navigation in oil palm plantations using D-lite algorithms for real-time path planning to address labour shortages. Similarly, Jeon et al. (2024) developed a polygonal path planner for unmanned tillage in paddy fields, achieving a similar efficiency as manual operation. Bochtis et al. (2010b) explored algorithmic solutions for in-field navigation to optimize paths for agricultural service units, improving efficiency and reducing compaction in large-scale machinery operations. Although these approaches have improved operational efficiency, most do not directly address soil compaction and overlook the cumulative impact of machinery passes (Chatzisavvas et al., 2023).

A few studies have integrated soil characteristics directly into path planning models. Bochtis et al. (2012) developed a decision support system (DSS) that uses electrical conductivity maps to assess soil sensitivity to compaction, optimizing machinery routes to reduce compaction risks. Similarly, Spekken et al. (2016) employed the RUSLE model to assess erosion, considering factors such as soil type and topography to optimize machine paths on steep terrains. However, these models do not explicitly account for the nonlinear nature of soil responses under repeated mechanical stress.

In contrast, the SoilFlex model (Keller et al., 2007) addresses this non-linear behaviour by simulating soil compaction and stress distribution using analytical stress propagation equations. It incorporates mechanical soil properties to predict displacement and rut depth, offering a comprehensive view on soil behavior under machinery loads.

In this chapter, we describe Soil2Cover, an advanced coverage path planner that integrates the SoilFlex model into a novel soil compaction cost function. Our contributions are threefold:

- We design a cost function that captures the nonlinear behavior of soil compaction using SoilFlex, modeling the effect of repeated machinery passes.
- We develop a dual-graph route planning strategy that minimizes both route length and soil compaction, thereby reducing unnecessary re-tracing of field areas.
- We evaluate our approach on 1000 real-world fields under both single-crop and intercropping scenarios.

This chapter quantifies the benefits of integrating the SoilFlex model by comparing routes avoiding soil compaction against minimal path length routes, where the latter conform to the most commonly used criterion in literature (Filip et al., 2020). The proposed approach supports sustainable farming practices—such as intercropping—and the use of autonomous agricultural machinery, thereby advancing precision farming and sustainable soil management practices.

5.2 Materials and Methods

5.2.1 General overview

The flow diagram of Figure 5.1 shows the Soil2Cover method for solving an agricultural route planning problem while minimizing soil compaction. Required input data includes soil composition, robot load, crop distribution, field boundaries, and robot width. The information is processed by the SoilFlex model to develop a cost function that minimizes soil compaction and helps maintain soil health. A Headland Generator uses the field boundary and robot width to distinguish between the inner field area and the headland. Next, the Swath Generator creates swaths, accommodating different crop types in case of strip cropping or intercropping. Subsequent steps involve computing the weights for the path and coverage graphs that are essential for route planning. The coverage routing directly affects the soil compaction component of the cost function. An optimizer repeatedly assesses the cost function to compute the optimal route by continuously refining until the most efficient path for field coverage is determined. The loop terminates when subsequent iterations yield identical routes. Green and yellow blocks in the flowchart distinguish

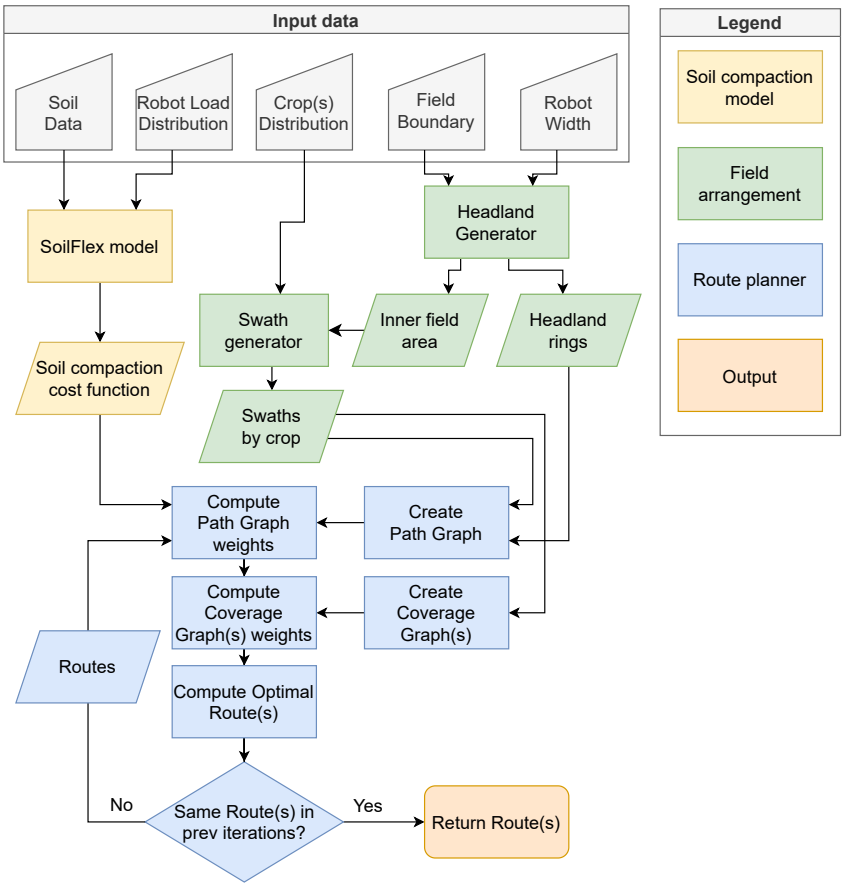


Figure 5.1: Flowchart of the Soil2Cover method. Block colours distinguish functionalities (see Legend).

field area distribution from soil management, emphasizing the method's integration with precision agriculture.

5.2.2 Soil compaction under a single wheel

The path of agricultural vehicles directly affects soil disturbance. For a single wheel, a direct relationship can be established between the area of soil disturbed, and the path distance travelled over previously-undisturbed soil. Specifically, the disturbed area is approximately equal to the product of the width of the tyre and the path length (Mier et al., 2023a).

This relationship extends to soil compaction, stating that for a tyre producing constant normal stress along a straight path, the bulk density change is proportional to the path length and the bulk density change at a point within the disturbed area. This approximation is only valid if soil properties are homogeneous over the disturbed area.

To compute the bulk density change, this work employs the SoilFlex model (Keller et al., 2007). The SoilFlex model estimates soil stress and compaction by assessing the

Table 5.1: Relevant measures of the AgBot robot

Specification	Value
Total Weight	7800kg
Track Length	2.55m
Track Width	0.61m
Idlers Per Track	2
Idler Radius	0.55m
Rollers Per Track	4
Roller Radius	0.3m

distribution of vertical stress beneath agricultural machinery. Under repetitive passes with the same load, the soil bulk density asymptotically approaches its saturation level using a logarithmic expression. The equations of this model are developed in Appendix 1.

For the scope of this research, the soil compaction produced by a single tyre following a path over homogeneous soil is approximated using the path length and the bulk density change below the tyre centre at 20 cm depth. This simplifying assumption is consistent with the SoilFlex model and our subsequent analyses and discussions.

5.2.3 Soil compaction cost function

Tractors are typically characterized by a quad-tyre configuration or caterpillar tracks arranged in two lanes. Hence, the soil compaction affects twice the area compared to a single wheel lane. For brevity, this chapter elaborates only on the case of caterpillar tracks. Particularly, the AgBot (Fig. 5.2) of the AgXeed company is used as an example. However, the methodology is equally applicable to any other type of wheeled or tracked vehicle.

Each AgBot track is composed of two idlers and four rollers (Table 5.1). The compaction made by a track on a traversed point is equal to the difference between the initial bulk density and the bulk density after idlers and rollers have passed. Note that the order in which each wheel stress is applied affects the final bulk density. An initial bulk density, ρ_0, is used to compute the bulk density after the first idler passed, ρ_1. Next, ρ_1 is the initial bulk density for computing the bulk density after the first roller passed, ρ_2. By the same reasoning, roller 2, 3 and 4 produce ρ_3, ρ_4 and ρ_5, respectively, and the second idler produces ρ_6. Consequently, the variation of bulk density made by the track is $\Delta\rho_t = \rho_6 - \rho_0$.

Let $T_\rho(\rho_{init})$ be a function that –given an initial bulk density ρ_{init}– returns the bulk density after the track passes. The function $B_\rho(n)$ is then defined to return the bulk density change made by a track after passing a point n times:

Table 5.2: Load distribution values for each wheel in AgBot tracks. Each value is the fraction of load applied to that wheel.

Load Distribution Type	Idler 1	Roller 1	Roller 2	Roller 3	Roller 4	Idler 2
Uniform	0.167	0.167	0.167	0.167	0.167	0.167
Trapezoidal	0.016	0.086	0.139	0.192	0.245	0.319
Triangular	0.050	0.102	0.184	0.265	0.347	0.050

Figure 5.2: Load distribution on tracks according to Uniform, trapezoidal and triangular distributions

$$B_\rho(n) = \begin{cases} B_\rho(n-1) + T_\rho(B_\rho(n-1)) & \text{, if n} \geq 1 \\ T_\rho(\rho_{init}) & \text{, otherwise} \end{cases} \tag{5.1}$$

Even when the load of the robot is evenly distributed across two tracks, the load distribution within each track (Fig. 5.2) may be uneven. Uniform load distribution weights for each wheel in a track were compared against trapezoidal (Wong et al., 2019) and triangular (Keller and Arvidsson, 2016) load distributions of weight, using the AgBot data listed in Table 5.2.

Figure 5.3 shows the comparative analysis of how the three load distributions modify the bulk density values obtained, starting with a bulk density of $B_\rho(0) = [1g/cm^3]$. Each load distribution is represented in a row. Each line in a plot represents a different total load applied. In the columns, the plot shows the evaluated values of $B_\rho(n)$ and $T_\rho(B_\rho(n))$, respectively. Note that the y-axis in the second-row plots has logarithm scale.

Even though an increase in stress produces greater soil compaction, the ratio between the results from different loads is constant. Therefore, for a field with a soil with homogeneous properties and assuming that the robot has constant weight, the weight of the robot has no impact on the route that minimizes the soil compaction.

However, the load distribution model does produce different soil compaction results. In contrast to the uniform load distribution, the triangular and trapezoidal load distributions yield similar results. Since an uniform load distribution is unrealistic, this study adopts a trapezoidal load distribution.

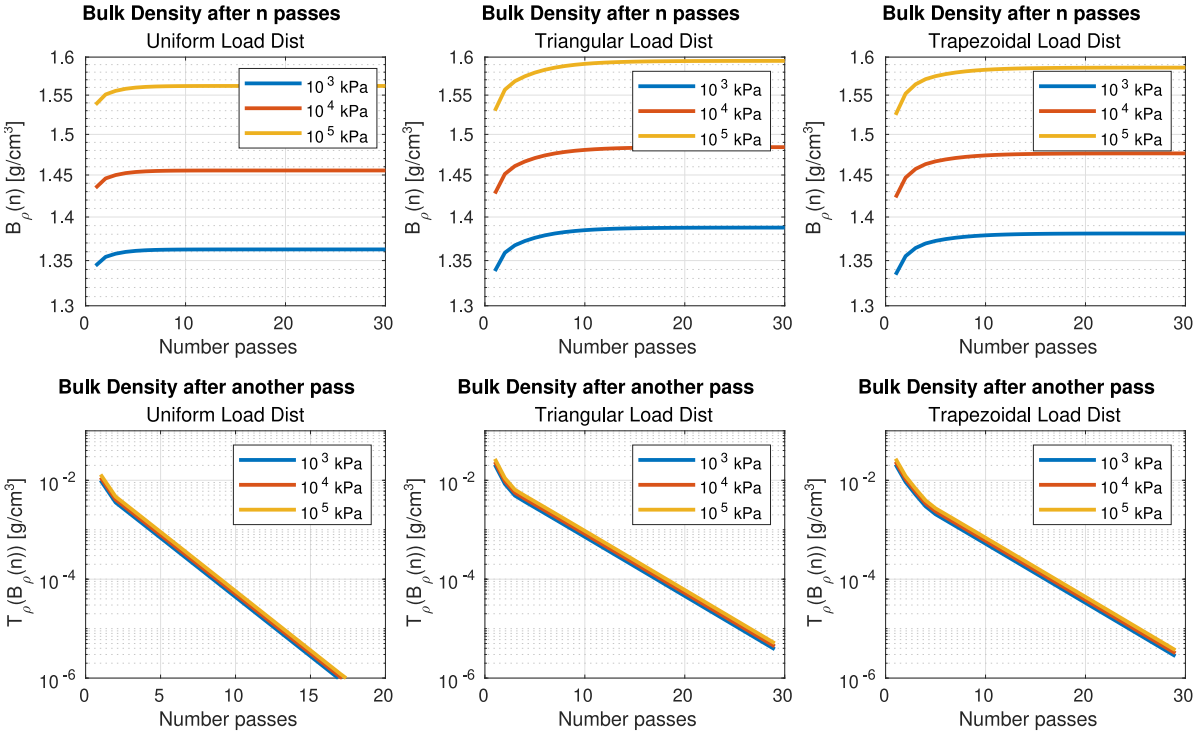


Figure 5.3: SoilFlex-computed bulk density after n passes with the robot. The number of passes is represented by the x-axis. The columns refer to results using Uniform, Triangular and Trapezoidal load distributions, respectively.

For each route segment, the soil compaction cost is computed as:

$$f_c(n) = \Omega_L * B_\rho(n) \tag{5.2}$$

where Ω_L is the effective travel length of the route segment, and $B_\rho(n)$ is the increase in soil bulk density after n passes, i.e, it models the compaction resulting from repeated machinery passes. Since the track width is constant, Ω_L implicitly quantifies the area compacted by the machine passes. The total soil compaction cost produced by a complete route is the sum of the costs of all the route segments.

5.2.4 Headland generation

For several operations, revisiting a previously covered swath would damage the crop. Therefore, manoeuvres between swaths occur in the headlands. A headland is an area reserved to make turns from swath to swath or to travel around the field without damaging the crop in the inner field.

Following common practice (Nilsson and Zhou, 2020), the headland is here generated by inward buffering the borders of the field (including obstacles) by three times the width of

the robot. Moreover, a linear ring between the inner and outer borders of each headland is created. In this work, this line is called a headland ring, and is denoted by $H^i_{p=j}$, where $i \in [1, I]$ is the index of the headland, with 1 referring to the outer border, and I to the inner border, and $j \in [1, J_i]$ the j^{th} point in the i^{th} headland ring, with J_i being the number of points in the headland ring i.

5.2.5 Swaths planner

Swaths are used by the robot to cover the field while traversing the inner field during the operation. Swaths are generated by intersecting parallel straight lines with the area of the inner field, until the entire field is covered. The distance between consecutive lines equals the width of a crop strip. The orientation of the swaths can be predetermined, for example in the case of orchards or already cultivated land, or it can be optimized according to an objective function (Mier et al., 2023b). The objective function used in Soil2Cover for planning the swaths in the inner field is the sum of the lengths of the swaths. This is consistent with the fact that for a single pass the soil compaction is proportional to the length of the route. Swath angle optimization uses an exhaustive search with a step size of $\pi/180$ (Mier et al., 2023b).

5.2.6 Single crop

In this section, the route planner for a single crop is explained following the pseudocode presented in Algorithm 1. The initialization steps (#1) in this algorithm refer to the Sections 5.2.3, 5.2.4 and 5.2.5, respectively.

Algorithm 1 Soil2Cover Route Optimization

Require: Field geometry, crop distribution, soil properties, robot width, number max of passes P_{max}

Initialize:
- Pre-compute the soil compaction cost function $B_\rho(i)$, for $i \in [0, P_{max}]$.
- Generate headland rings from the field boundary.
- Generate swaths using the swath planner $\Leftarrow$ Find Swaths that minimize sum of swath lengths.

Create both graphs:
- Construct the initial Path Graph G^P with vertices from headland rings, and swaths.
- Compute the distance $\Omega_L(e)$ for all edges $e \in G^P$.
- Set traversal count $n(e) = 0$ for all edges $e \in G^P$.
- Construct the initial Coverage Graph G^C with vertices from swaths.

for each edge $e \in G^P$ **do**
- Compute the Euclidean distance $\Omega_L(e)$.
- Set traversal count $n(e) = 0$.

end for
repeat
    for each edge $e \in G^P$ **do**
        Update weight: $w(e) = \Omega_L(e) \cdot B_\rho\big(n(e)\big)$.
    end for
    Compute all-pairs shortest paths in G^P.
    Update Coverage Graph G^C:
    for each pair of vertices $v_i, v_j \in G^C$ **do**
        if v_i and v_j belong to the same swath **then**
            Set $w(v_i, v_j) = 0$.
        else
            Set $w(v_i, v_j)$ equal to the least cost from v_i to v_j in G^P.
        end if
    end for
    Solve the Travelling Salesman Problem (TSP) on G^C to determine the optimal swath coverage order.
    Construct the complete route R by concatenating the least cost paths between consecutive swath nodes from G^P.
    for each edge e **do**
        Update the traversal count: $n(e) \leftarrow$ (number of passes on e in route R).
    end for
until the stopping criterion is met (i.e., no significant reduction in bulk density between iterations or $P_{\max}$ is reached)
return Optimized route R

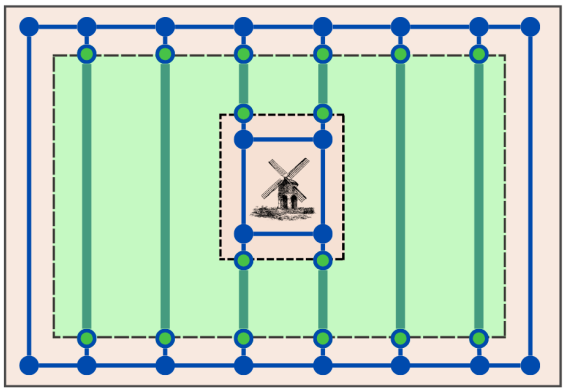


Figure 5.4: Path Graph example. Nodes and edges of the Path Graph are shown in blue. Green points with a blue border are part of both the Path Graph and the Coverage Graph. Swaths are represented by green lines; these are not edges of the Path Graph. The headlands are shown in light brown; the inner field is green.

Graph definition

A graph, defined by $G = (V, E)$, is a structure that contains a set of nodes, called vertices "V", and the relationship between pairs of vertices, called edges "E". Each edge has an associated weight representing the cost to traverse it. If two vertices are not connected, the weight between them is $+\infty$

In Soil2Cover, a 2-bidirectional-graphs approach is used to solve the route planning problem in agriculture. Those graphs are the Path Graph (G^P) and the Coverage Graph (G^C).

Path Graph

The Path Graph, $G^P = (V^P, E^P)$, (Fig. 5.4) is used to find the shortest path to travel through the headlands between swaths. Vertices of G^P are points, and edges are line segments. The Path Graph is populated with nodes and edges in three steps.

First, headland rings points ($H^i_{p=j}$) are added to V^P, and the connection between consecutive points ($H^i_{p=j} \leftrightarrow H^i_{p=j+1}$) in the same headland ring are added to E^P.

Second, for each swath, and for each point on it (s_p), the closest point on the headland rings is found (s'_p). s_p and s'_p are added to V^P and the connection between s_p and s'_p is added to E^P. This procedure also applies to the start and end points of the route.

Third, redundant edges are removed. Since the soil compaction cost function requires knowing how many times an edge has been traversed, every pair of overlapping edges is transformed into non-overlapping segments. Then, for each node $v^P_j \in V^P$ and edge $e^P_{ik} \in E^P$, with $i \neq j \neq k$, where v^P_j is the j^{th} vertex and e^P_{ik} is the edge connecting vertices i and k. For each pair v^P_j and e^P_{ik}, if the point corresponding to v^P_j is located on

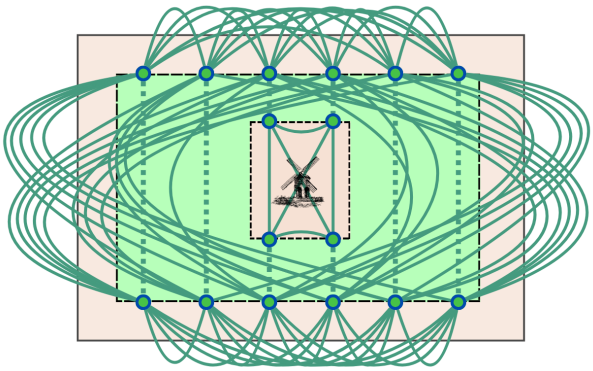


Figure 5.5: Coverage Graph represented over a field. Green lines are connections between nodes, dotted green lines are edges with 0 cost, connecting nodes on the same swath. Every pair of nodes on the same headland is connected.

the segment that joins the points v_i^P and v_k^P , the edge e_{ik}^P is removed from E^P, and the edges e_{ij} and e_{jk} are added to E^P. We consider that v_j^P is on the segment (v_i^P, v_k^P) if the distance from the point to the segment is less than δ_{tol}.

The weight of an edge $e_{ij}^P \in E^P$ is the Euclidean distance between v_i^P and v_j^P.

Coverage Graph

The Coverage Graph, $G^C = (V^C, E^C)$ (Fig. 5.5), is used to define the space of possible routes.

Each swath is represented by its two endpoints. A swath is considered visited after both endpoints are visited, with their order determining the traversal direction. V^C contains both endpoints of every swath and the field entry point. In addition, an important property is that $V^C \subset V^P$, meaning the nodes on the Coverage Graph are also represented in the Path Graph (Bochtis and Vougioukas, 2008).

The weight of the edge $e_{ij}^C \in E^C$, being v_i^C and v_j^C points of the same swath is 0. Otherwise, the weight is the cost of the shortest path from v_i^C to v_j^C in the G^P. The shortest path between each pair of vertices in G^P is computed using the Floyd-Warshall algorithm (Floyd, 1962). The Floyd-Warshall algorithm efficiently computes the shortest paths between all pairs of vertices in a weighted graph. If two vertices are not connected in G^P, they are neither connected in G^C.

Finding the shortest route

The Coverage Graph problem is equivalent to the Travelling Salesman Problem (TSP). The TSP is the problem of finding, for a given graph, the shortest Hamiltonian path, which is a path that visits all the vertices of the graph. This problem is NP-hard, as there is no known method to find a solution in polynomial time. Fortunately, there are open-source solutions

like Or-tools (Perron, 2011), which provide optimizers to find near-optimal solutions in reasonable time.

Once the optimizer returns the coverage order of the swaths, the transition path between swaths is searched in the G^P.

5.2.7 Minimizing soil compaction for single crop

To find a route that minimizes soil compaction, Soil2Cover uses an iterative algorithm. First, the function $B_\rho(n)$, previously defined, returns the difference of bulk density between the initial bulk density and its value after the n^{th} pass (Fig. 5.3). Next, G^P and G^C are created following the same procedure as in Section 5.2.6. The weights of G^P are replaced by the cost of passing through each edge one more time. Equivalently, the cost of the edge e_{ij} equals $d_{ij} * B_\rho(n_{ij})$, being n_{ij} the number of passes between vertices i and j, and d_{ij} is the Euclidean distance between vertex i and j. In the first iteration, the costs of G^P to minimize soil compaction are proportional to the costs of G^P to minimize distance.

G^P is used to compute the weights of G^C, then find an initial coverage route, and generate the complete route. Once the coverage route is generated, it is split between segments that correspond to edges in E^P. For each edge e^P_{ij}, n_{ij} is updated with the number of times that segment is traversed in the route, regardless of its direction. The latter numbers are used to update the weights of G^P. G^P is again used to generate the new weights of G^C and to produce a new route.

This repeats until the obtained route remains unchanged for two consecutive cycles. The method returns the route that has minimized the total soil compaction.

5.2.8 Minimizing soil compaction with strip cropping

In the case where multiple row crops are planted in alternating strips, M refers to the number of crops in the field. Each crop has its own swaths S_i, where $1 \leq i \leq M$, covered either by different robots or by the same robot at different times. For this case, Soil2Cover has a common Path Graph, G^P, for all crops, while creating a unique Coverage Graph, G^C_i, for each crop ($1 \leq i \leq M$). Note that generating G^P with points from all swaths (regardless of crop) ensures $V^C_i \subset V^P$ for all crops i. The algorithm followed is the same as for a single crop, with the difference being that the route for all crops is generated before updating the weights of G^P. Consequently, n_{ij} is the number of times an edge e_{ij} in E^P has been traversed, regardless of which robot visited the edge. The rest of the steps and the stopping condition remain similar.

5.2.9 Experiments

To select a representative set of fields, we retrieved boundary polygons from the `https://github.com/Courseplay/CourseGenerator` repository in `.xml` format. Each file in

this repository contains multiple fields. We retained only fields with an area between 1 and 20 ha and exported them to `.wkt` (well-known text) format, yielding 1567 fields. We retained the first 1000 fields to limit computation time while still capturing a broad range of shapes and sizes for our experiments.

Both single crop with 3m row width (AgBot width) and strip cropping with two crops in 6m width as in Campanelli et al. (2023) were considered. In both cases, the robot operation concerned one, two and three passes over the crop cycle. As soil compaction reduces by reusing previous paths, robots operations for the same crop were constrained to follow the same route. To include this case in Soil2Cover, $f_c(n_{ij})$ is replaced by $\sum_{k=1}^{2} f_c(2 * n_{ij} + k)$ or $\sum_{k=1}^{3} f_c(3 * n_{ij} + k)$, for simulating repeating the operation two and three times respectively.

All analyses and graphics were produced using C++, GDAL (GDAL/OGR contributors, 2022) and Or-tools (Perron, 2011), and with Matlab 2023b. The laptop used for experiments was an MSI GF627RE with an Intel(R) Core(TM) i7-7700HQ CPU, running Ubuntu 22.04.5.

Experiments were done using the Fields2Cover library, a software tool specifically designed to optimize coverage path planning (CPP) for agricultural machinery. It allows considering various factors like soil properties, vehicle characteristics, and environmental constraints to implement path planning algorithms (Mier et al., 2023b).

5.3 Results

5.3.1 Single crop

Figure 5.6 shows the ratios of soil compaction and route length between different routes. Let R_A be the baseline route obtained by optimizing solely for route length, with soil compaction S_A and route length L_A. For any alternative route R_B, with soil compaction S_B and route length L_B, its ratio is defined as (S_B/S_A, L_B/L_A). Thus, a ratio of (1, 1) implies that R_B has identical soil compaction and route length to the baseline route R_A. A value below 1 for either component indicates an improvement (i.e., lower soil compaction or shorter route) relative to the baseline.

After optimizing the route for 1000 fields with a single pass, 66 of them produced different solutions depending on the cost function. Unexpectedly, for single crops, optimizing for soil compaction produced better results in both soil compaction and route length compared to route length optimization. This likely occurred because modifying the weights in G^C helped the optimizer escape local minima. The average soil compaction improvement for the 66 fields with one, two, and three passes were [0.964, 0.971, 0.978], while route length improvement were [0.994, 0.995, 0.996].

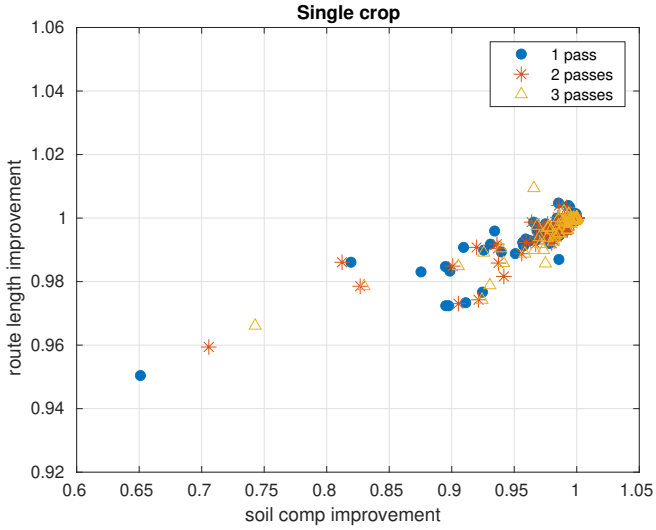


Figure 5.6: Ratios of soil compaction and route length, between routes obtained optimizing soil compaction and routes optimizing route length. Routes account for 1, 2 and 3 passes, for a single crop at 3m width.

Figure 5.7 shows the obtained route for a field optimized for route length and for soil compaction. On this field, when the route minimizes soil compaction, the bottom right headlands are not travelled, to reduce the damage to the soil in that area.

5.3.2 Strip cropping of two crops in 6m wide strips

Figure 5.8 shows the route improvement ratios for a strip cropping scenario with two crops grown in alternating 6m wide strips. For the one-pass case, 66 out of the 1000 fields yielded a different route than the baseline (i.e., the route obtained when minimizing path length), while for the three-pass case, 130 fields were assigned a different route. For the former 66 fields, we computed the average ratios of the soil compaction and route length values (with the ratios defined as the metric for the soil compaction optimized route divided by that for the baseline route, $(S_B/S_A,\ L_B/L_A)$). The average soil compaction ratios for one, two, and three passes were [0.964, 0.966, 0.973], respectively, while the corresponding average route length ratios were [0.991, 0.992, 0.995]. Recall that ratios below 1 indicate improvements compared to the baseline.

Figure 5.9 shows the routes through three fields, for the two considered cost functions. The first case concerns a small rectangular field; To minimize the soil compaction, the optimizer produces a path without lateral headland passes. The second field contains two large obstacles. In this case, the swath coverage order changes significantly: the first robot begins in the northwest corner to minimize route length, but in the southeast corner to minimize soil compaction. In the last field, Soil2Cover employs the same strategy as in the first, avoiding the right edge to prevent additional soil compaction.

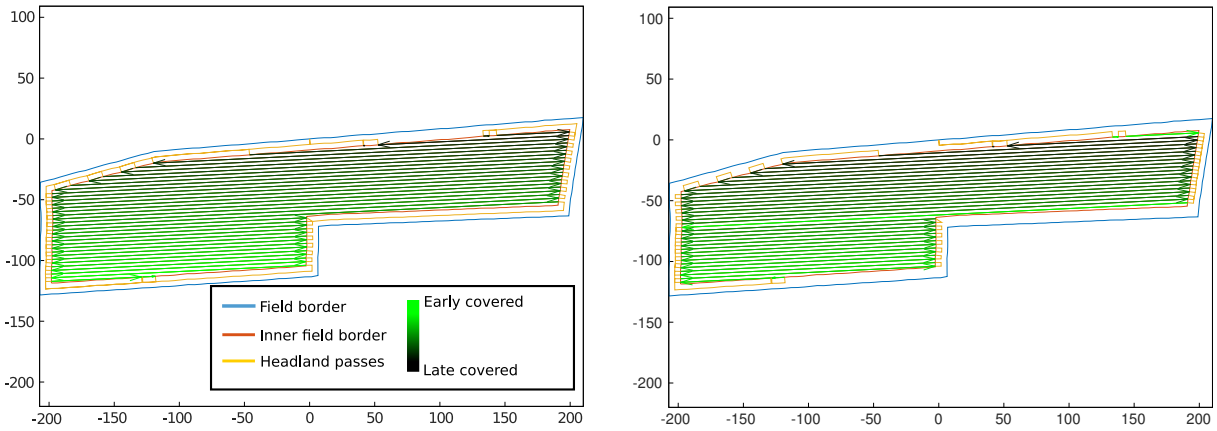


Figure 5.7: Optimized routes for the same field according to two objectives. Left: minimal length route. Right: minimal soil compaction route. The coordinates in both sub-figures are in meters relative to the start and end point of the route.

5.4 Discussion

We have demonstrated that routing autonomous tractors along fixed paths can reduce soil compaction risk, thereby preserving soil health and boosting crop yields (Fig. 5.6). Our approach reduces deterioration of soil structure and improves efficiency by reducing operational route length. Predictable traffic routes integrate seamlessly with precision farming systems, enhancing input efficiency and supporting sustainable management. Although farmers recognize the need for Controlled Traffic Farming (CTF) practices that limit autonomous tractors circulation to permanent tracks (McPhee et al., 2020), operational challenges often hinder their adoption (Tamirat et al., 2022). In this context, an autonomous route planning algorithm that adheres to CTF principles represents a key means for preserving soil health.

5.4.1 Novelty of the work

The integrated SoilFlex model (Keller et al., 2007) treats soil compaction as a complex, nonlinear process. By accounting for soil properties, equipment weight, and pass frequency, Soil2Cover offers a meticulous representation of field conditions and a deeper understanding of machinery's impact on soil. Unlike previous methods (Bochtis et al., 2012; Plessen, 2018; Spekken et al., 2016), it minimizes soil damage by evaluating changes in bulk density from vehicle passes—focusing on the disproportionate impact of initial passes on surface layers (Patel and Mani, 2011; Pulido-Moncada et al., 2019). Simulations on 1000 fields using a single 3-meter machine, Soil2Cover improved route efficiency by up to 4–6% and reduced compaction by up to 30% (Fig. 5.6). Even without explicit compaction optimization, the algorithm consistently found efficient routes that reduce route length and environmental impact.

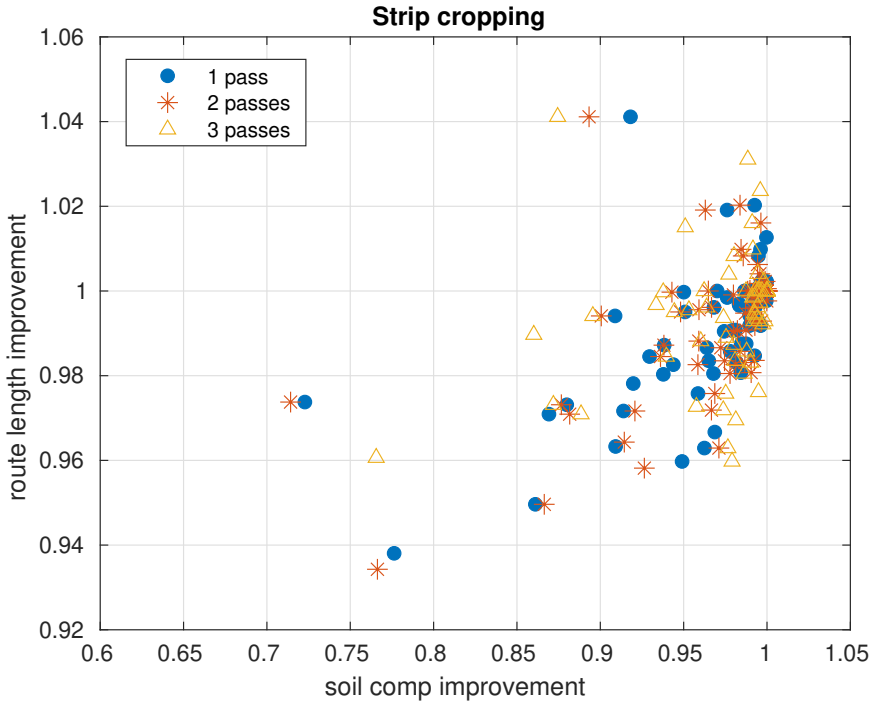


Figure 5.8: Ratios of soil compaction and route length, between routes obtained optimizing soil compaction and routes optimizing route length. Routes account for 1, 2 and 3 passes, for two crops at 6m width.

The algorithm also handles complex field shapes and obstacles, ensuring efficient land coverage while protecting soil quality–a critical requirement for sustainable agriculture. Its two-graph method, using the Floyd-Warshall algorithm, outperforms distance-only approaches by directly minimizing soil damage. Moreover, by integrating the SoilFlex model, Soil2Cover treats soil as a heterogeneous continuous system and offers route planners a new strategy to mitigate soil compaction and preserve soil structure.

5.4.2 Intercropping and strip cropping

Soil2Cover also supports routes for intercropping. It customizes routes by accounting for each crop's machinery and compaction sensitivity. While intercropping (or mixed cropping) and strip cropping are known to boost productivity and preserve soil fertility (Brooker et al., 2015; Hernández-Ochoa et al., 2022), these practices demand tailored machinery to meet each crop's unique requirements. For example, perennial plants, such as fruit trees, require distinct management compared to annual crops (Hauggaard-Nielsen et al., 2012; Ma et al., 2007; Martin-Gorriz et al., 2022; Wei et al., 2024).

In our intercropping simulation—planting crops in 6m strips (Campanelli et al., 2023)—soil compaction improvements matched those in single-machine trials, with path lengths varying by up to 6% and compaction dropping by up to 30% (Fig. 5.6). These results further

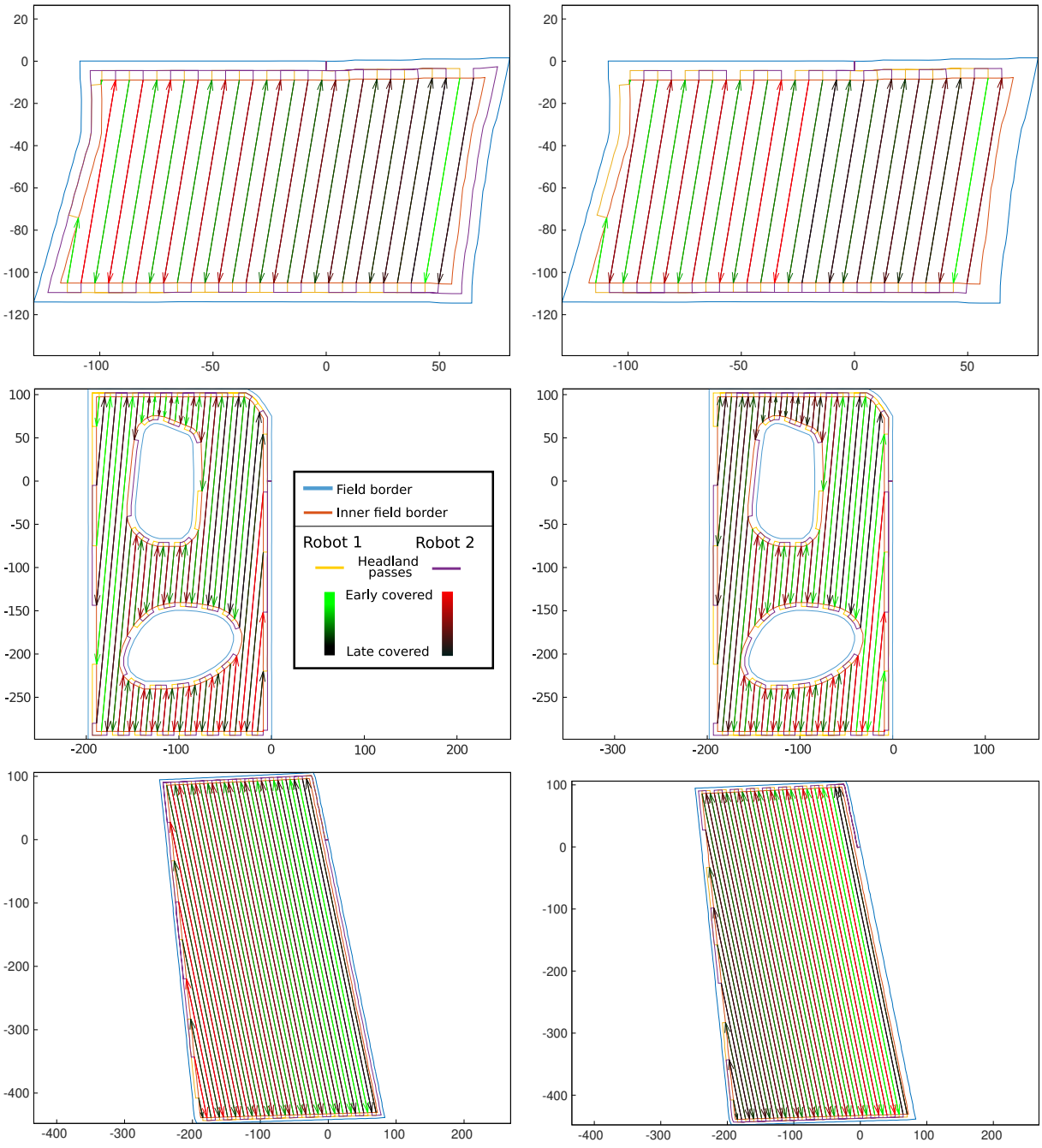


Figure 5.9: Optimized routes on three fields (rows) according to two cost functions (route length, left; soil compaction, right). In these scenarios, two robots operate in the same field, doing strip cropping. Swaths are assigned alternately: odd-numbered swaths to one robot, even-numbered swaths to the other. Both robots start at the origin of the local coordinate system. Axes are in meters.

validate Soil2Cover's capacity to enhance machinery use while reducing operational time, fuel consumption, and environmental impact.

Additionally, the technique can integrate with low-pressure tyres, advanced tracks, and gantry conveyors to further reduce compaction (Mileusnić et al., 2022). Adjusting SoilFlex inputs enables more optimized paths. The algorithm's flexibility also permits integration with multi-robot soil mapping (Roberts-Elliott et al., 2022) and real-time wheel track identification (Zhang, 2024), enabling dynamic path adaptation based on fresh field data. Future versions may consider diverse agricultural vehicles, as some operations permit tyres with lower inflation pressures to expand the tyre-soil contact area and cut compaction (Shaheb et al., 2021). In this work, the approach was tested on straight swaths. The method is expected to work on curved swaths as well, and this will be tested in future research.

5.5 Conclusions

Soil2Cover provides a key tool for sustainable soil management. By integrating the SoilFlex model and targeting machinery pass frequency rather than merely tyre load, it offers a precise strategy to reduce soil compaction–a major threat to productivity and soil health.

Simulations on 1000 fields showed that Soil2Cover improved route efficiency by 4–6% in 6–13% of the fields and reduced compaction by up to 30% in both single-machine and intercropping scenarios. Its robust performance in complex fields demonstrates its capacity to protect soil while guiding robotic tractors with high precision. Notably, its cost function captures the nonlinear dynamics of soil compaction and the outsized impact of initial passes, which is vital for planning routes that prioritize soil health over simple distance minimization. Moreover, by customizing routes for intercropping, Soil2Cover optimizes yield while safeguarding soil health.

Future work will test Soil2Cover across diverse field conditions and soil types to confirm its robustness. A next step would be to relax the assumption of homogeneous soil properties across the entire field. This advancement can be combined with real-time in-field sensing of soil and weather conditions to further optimize operational timing, reduce compaction, cut costs, and increase yields.

Appendix

5.A SoilFlex equations

The type of tractor wheel affects the pressure applied to the ground. In the case of wheels, SoilFlex (Keller et al., 2007) describes the vertical stress as:

$$\sigma(y) = C_A * \left(\frac{w(x)}{2} - y\right) * e^{-\gamma*((w(x)/2)-y)} \text{ , if } |y| \leq \frac{w(x)}{2} \tag{5.3}$$

$$\sigma(x) = \sigma_{x=0,y} * (1 - \left(\frac{x}{l(y)/2}\right)^{\alpha}) \text{ , if } |x| \leq \frac{l(y)}{2} \tag{5.4}$$

while the vertical stress for caterpillar tracks are defined by Keller and Arvidsson (2016) as:

$$\sigma(y) = \sigma_{max} * \left(1 + \frac{|y| - W/2}{W/2} * (1 - a)\right)/a \text{ , if } |y| \leq \frac{W}{2} \tag{5.5}$$

$$\sigma(x) = \sigma_{x=0,y} * \cos\left(\frac{2\pi x}{L}\right)/2 \text{ , if } |x| \leq \frac{L}{2} \tag{5.6}$$

where C_A, γ, α and a are tyre parameters (Keller and Arvidsson, 2016; Keller et al., 2007), $\sigma_{x=0,y}$ is the stress under the centre of the tyre, $l(y)$ and L are the length of the tyre, and $w(x)$ and W are the width of the tyre.

Vertical stress is distributed over the contact area, $A_{contact}$, discretized into i elements with A_i area and σ_i normal stress (Söhne, 1953). The radial normal stress, $\sigma_{r,i}$ at depth z, assuming negligible horizontal stress, is

$$\sigma_{r,i} = \frac{\xi \sigma_i A_i}{2\pi r_i^2} \cos^{\xi-2} \theta_i \tag{5.7}$$

where ξ is concentration factor, r_i is the distance from the centre of A_i to the desired point, and θ_i is the angle between the normal load vector and the vector from the centre of A_i to the desired point (Keller et al., 2007).

Using the principal stresses and the first stress invariant (Koolen and Kuipers, 1983), the mean normal stress, p, is:

$$p = \frac{1}{3}\sum_{i=0}^{n}\sigma_{r,i} \tag{5.8}$$

SoilFlex employs O'Sullivan and Robertson (1996) to model rebound and recompression effects of the soil, based on the virgin compression line (VCL), recompression line (RCL), and steeper recompression line (RCL').

$$VCL: \qquad \upsilon = N - \lambda_n \ln p \tag{5.9}$$

$$RCL: \qquad \upsilon = \upsilon_{init} - \kappa \ln p \tag{5.10}$$

$$RCL': \quad \upsilon = \upsilon_{YL} - \sqrt{\lambda_n \kappa} \ln p \tag{5.11}$$

where υ is the specific volume, υ_{init} is the initial specific volume and υ_{YL} is the volume at the intersection of the yield line and the recompression line. N is the specific volume at $p = 1kPa$, λ_n is the compression index and κ is the recompression index (Keller et al., 2007). Finally, the Bulk Density, ρ, is computed as

$$\rho = \frac{\rho_s}{\upsilon} \tag{5.12}$$

where ρ_s is the density of solids.

In SoilFlex, parameters such as the compression index λ_n, the recompression index κ, the initial specific volume υ_{init}, and the load distribution factors (C_A, γ, α, etc.) are used to compute bulk density changes. Greater values for compression or recompression indices increase predicted soil deformation under load. Conversely, parameter sets indicating lower soil susceptibility (e.g., smaller λ_n) yield smaller incremental bulk density changes, making certain routes more favorable. Thus, adjusting these model parameters to represent the local soil type and machinery characteristics directly influences route selection, allowing Soil2Cover to prioritize or avoid certain zones based on their predicted compaction risk.

Chapter 6

Synthesis

6.1 Revisiting the Research Objectives

This thesis contributes to the development of the coverage path planning problems for agricultural vehicles. It addresses four research objectives. The first objective unifies the ACPP problem in a modular and generic framework (Chapters 2 and 3). The second objective raises the scientific transparency of ACPP studies by releasing an open-source library (Chapter 2) and an open-source benchmark (Chapter 3). The third research objective (Chapter 4) increases field coverage by improving headland coverage, planning a continuous-curvature path that keeps the robot inside the field area. Lastly, the fourth research objective (Chapter 5) aims to reduce soil compaction caused by the robot's coverage route.

1. **Integrating coverage path planning algorithms into a unified framework**
2. **Enhancing the scientific transparency of coverage path planning research**
3. **Improving headland coverage and headland turning feasibility**
4. **Reducing soil compaction through coverage path planning optimization**

6.2 Main findings

6.2.1 Integrating coverage path planning algorithms into a unified framework

The Agriculture Coverage Path Planning (ACPP) problem has been addressed using many algorithms, yet historically the solutions have remained fragmented, sparse, and difficult to integrate or compare (Utamima and Djunaidy, 2022). Cellular decomposition, boustrophedon patterns, genetic optimizations, and other methods answer the same general task, but relying on different inputs, outputs and objective functions (Höffmann et al., 2024). Researchers propose algorithms that solve specific problem, which split the research line into sub-problems. For instance, Hameed et al. (2013b) orient the swaths in the inner field, Spekken and de Bruin (2013) optimize the headland area according to the orientation of the swaths, and Bochtis et al. (2015) sort the generated swaths to find the shortest route. Due to the sheer number of sub-problems, direct comparison between methods is hard, and forces researchers to re-implement other algorithms beyond the original scope of their work.

Chapter 2 breaks this pattern with Fields2Cover, an open-source library that splits CPP into four modules: headland generation, swath generation, route planning, and path planning. In Chapter 3, Fields2Cover is expanded with a new module, called field decomposition. Each module aims to solve one of the sub-problems of ACPP, abstracting the researcher from the complete problem and allowing them to make fair comparison

between their methods and others. Every module exposes a common interface, so a new algorithm can be plugged in without touching the rest. Fields2Cover also provides a general workflow that executes the modules in order, and manages the inputs and outputs data. Currently, the unified framework used by Fields2Cover supports: convex and non-convex fields, straight and curved swaths, and robots with implements and limited capacity.

Chapters 4 and 5 were developed using the Fields2Cover framework from Chapter 2. Additionally, Chapter 3 releases a benchmarking tool that compares algorithms in the framework, integrating many state-of-the-art methods into a public solution. By giving one baseline for data, metrics, and workflow, the framework now turns scattered ideas into parts of a single, testable solution. Therefore, future work can now target scientific gaps rather than rebuild algorithms that were already developed.

6.2.2 Enhancing the scientific transparency of coverage path planning research

Coverage path planning research gains strength when readers can inspect every step of the experimental pipeline. In science, many studies do not release the full source code or the raw data that support their results (Elliott, 2022). This limited disclosure prevents an independent researcher from repeating the experiments, checking intermediate computations, or adapting the methods to new conditions (Finger et al., 2025). A transparent workflow should facilitate replication by other researchers. Reproducibility builds trust and speeds progress because later studies can start from a verified baseline rather than rebuild earlier tools from scratch (Ambrósio et al., 2021).

Even though this thesis promotes transparency as one of its pillars, some practical hurdles limits that complete openness. Chapters 2 and 4 use the public dataset made by Nilsson and Zhou (2020). However, a researcher must request this dataset by email, which limits its accessibility and transparency. Chapter 5 faced a different difficulty when it sourced fields from the Courseplay repository. The files required format conversion before use, and the license for the transformed data did not allow republication, so external teams cannot inspect exactly the same inputs. Such cases show that even well-intended releases can leave gaps that block strict replication. Different datasets, file formats, and hidden preprocessing steps produce results that cannot be compared with confidence across publications.

Chapter 2 aims to improve the transparency in ACPP with the Fields2Cover library, which contains every algorithm described in this thesis. Each routine includes full documentation and unit tests, which guide other researchers through the code paths and clarify implementation details that often remain implicit. The public repository invites independent review, bug reporting, and direct extensions.

Chapter 3 extends my commitment to openness with a fully open-access benchmark of fields that load directly in Fields2Cover, and a benchmarking tool to compare ACPP algorithms under the same metrics.

Public input data, and well-maintained open-source code improve transparency and trust on the published results. This clear and unified workflow sets a higher bar for evidence in coverage path planning and pushes the community to adopt those transparency standards on their own.

6.2.3 Improving headland coverage and headland turning feasibility

Most coverage path planners focus on the inner field because that zone produces the highest yield (Höffmann et al., 2024). Usually, the headland is created by inward moving the field border by a number of track widths multiplied by the coverage width of the robot (Paraforos et al., 2018). The repeated offsets build a band of parallel tracks that are quick to compute but have sharp corners (Jeon et al., 2021). Such corners exceed the maximum curvature and their rate of change, so a non-holonomic robot cannot follow them (Höffmann et al., 2022). To generate a feasible headland coverage path, the planner should smooths the corners of the tracks, and respect the curvature and curvature rate constraints.

Chapter 4 introduces three headland corner path planners that are able to plan smooth paths for convex and concave corners, taking into account the curvature and curvature rate constraints, and the robot and implement dimensions. A selection criteria is proposed to pick the best headland corner path planner for each corner based on the area covered and the path length.

Moreover, when a robot with a long implement fixed to its rear follows a corner on a headland track, its tail swings far beyond the wheels (Ha et al., 2018). If the outermost headland track is just half a coverage width away from the field borders, the robot following the turn can push the tool beyond the border. This could cause collisions with posts, fences or ditches. To avoid this risk, farmers often leave a constant offset between the outermost headland track and the border to avoid such strikes. However, computing this safety offset is not easy. Leaving too much clearance between the first headland track and the border reduces the yield, whilst leaving too little leads to collisions with obstacles.

Chapter 4 contributes with a method for finding the minimal constant offset needed to prevent the robot implement to swing outside the field. Using a constant offset tuned to the smallest safe value, the mean covered area did not exceed 94.73%. To increase the covered area, another approach is proposed. In the latter, the offset is optimized separately for each field edge. Thanks to this approach, the mean of the fields area covered improved to 97.29%, which marks a 2.56% gain of over the constant–offset baseline.

This chapter demonstrates the value of optimizing the area reserved for the headlands. By shaping them according to the machinery dimensions and constraints, the inner field area can be increased, leading to higher yields.

6.2.4 Reducing soil compaction through coverage path planning optimization

Most soil compaction comes from repeated passes by heavy farm machines (Calleja-Huerta et al., 2023). The compacted layer blocks root growth, hinders water infiltration, lowers yields, and increases runoff and gas emissions (Shah et al., 2017). Despite these well-known impacts on soil health and productivity, classical coverage path planners often ignore this harm. They simply assume that soil compaction can be addressed by an objective function proportional to the path length. This assumption was also adopted in Chapters 2 and 3.

However, the shortest path is not necessarily the least damaging path. Soil responds in a nonlinear way to repeated loads (O'Sullivan and Robertson, 1996). Its reaction also shifts with composition and with the bulk density caused by earlier passes (Larson et al., 1980). To reduce the soil compaction owing to coverage operations, the route planner has to integrate a soil model that predicts the compaction.

Chapter 5 introduces Soil2Cover, a route planner that integrates the SoilFlex model (Keller et al., 2007). SoilFlex gives a physically-based estimate of how each pass increases the soil bulk density. Soil2Cover converts that estimate into a compaction cost for every headland segment. It starts with the shortest coverage route, for which the soil compaction objective function is computed over the entire route. Next, it repeats the search to find a new cost. The loop stops when the soil compaction objective functions of two consecutive iterations are very similar.

This iterative procedure guarantees a good result. If the compaction based search cannot reduce the soil damage, the algorithm still returns the length-optimal path. Experiments in Chapter 5 show that Soil2Cover can reduce the overall soil compaction caused during an operation. The benefit appears in both monoculture and intercropping layouts. Consistent with controlled traffic farming, the main finding of Chapter 5 is to minimize the disturbed soil area because the first pass compacts the soil the most.

6.3 Reflections and outlook

6.3.1 Open-source as a dissemination-driven strategy

This thesis research was done in collaboration with AgXeed (*AgXeed website* n.d.). The company needed robust coverage path planning for autonomous robots. By agreement, every paper, dataset, and line of code had to appear under open access and an open source

license. As a researcher, my aim was not only to release the material without cost, but also to persuade others to adopt it. To reconcile both aims, a one-year embargo on the code was initially imposed before its public release.

I chose GitHub as the main platform to promote transparency and collaboration (Dabbish et al., 2012). Popularity on this platform is indicated by stars that developers grant to projects they value (Borges and Valente, 2018). Figure 6.3.1 presents the evolution of the Fields2Cover star count.

To help newcomers, I prepared a detailed `README`[1] and a documentation site[2] that describe the purpose of the library and its installation, use, and citation. Clear guidance lowers the access barrier and raises visibility (Wang et al., 2023). I then announced the release on LinkedIn (Mier, 2022b), Reddit (Mier, 2022c), and developer blogs (Mier, 2022d). In one Reddit thread I invited the community to "roast my project" (Mier, 2022a), expecting that critical debate about best practices would spark lively feedback, a pattern noted by Chen et al. (2020). The ensuing discussion served as an informal peer review, guided improvements in the first release, and reinforced confidence in the code while boosting its profile.

The work on the code and the clear guides soon started attracting others to use the Fields2Cover library. Under direct supervision, six master students completed their thesis using Fields2Cover as a baseline. One of those student thesis led to the paper Mier et al. (2025b) about headland coverage path planning (Chapter 4). External teams also followed the online guides. Burro.ai integrated Fields2Cover into its mowing robot, and its Chief Technology Officer reported a significant reduction in development time, stating "Working with Fields2Cover was great! It is easy to use and very performant, the gain of time for our team was very considerable as we developed Cortador!" (Pita, 2024). Students at Osnabrück University of Applied Sciences likewise employed the library to build a stone collecting robot (Feldmann et al., 2024).

Interest soon extended beyond ground robots. Teams equipped unmanned aerial vehicles with Fields2Cover for tasks that range from inspecting farmland (Vélez et al., 2024) to searching from small objects in crop fields (Essen et al., 2025) to locating missing vessels (Alejo, 2024). Unmanned surface vehicles have used the library for bathymetric surveys (Zhao et al., 2024), and even tourists benefit when scenic spot tours are optimised through Fields2Cover (Song, 2024). All these examples show how Fields2Cover can inspire ideas far beyond its original scope.

Researchers often choose Fields2Cover as a benchmark for novel planners (Cai et al., 2024; Song, 2024). Few, however, contribute their implementations to the main repository. Two reasons stand out. The core is written in C++ with a Python interface, so many authors

[1] `https://github.com/Fields2Cover/Fields2Cover/blob/main/README.rst`
[2] `https://fields2cover.github.io/`

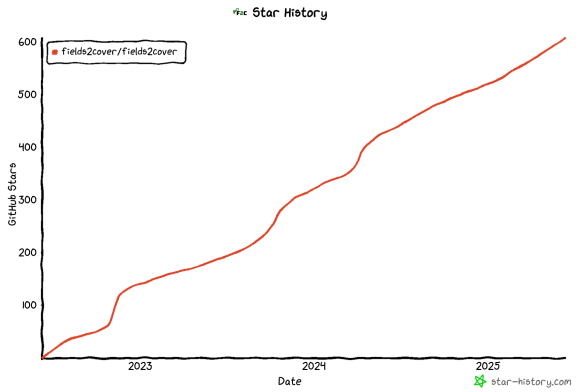


Figure 6.3.1: Star count of the Fields2Cover project over time. Graph generated with Star-history.com[3]

prototype in Python yet they lack the resources to port their code into C++. Furthermore, the current documentation cites only my own work, which may let other researchers think that new contributions will not gain credit. Weak community norms also surfaced when another team plagiarized large parts of the Fields2Cover paper (Mier et al., 2023b). After contacting the journal editor this was solved; for privacy reasons I do not reveal the specific work here. The project is thus often used, but has not yet formed a strong developer community around it.

Fields2Cover version 3 concludes the major releases prepared by the founding team. Limited updates may lead to a period of reduced activity (Aggarwal et al., 2014). The repository will continue to receive bug fixes, yet substantive progress must now arise from new maintainers or from a successor project that satisfies emerging requirements while still supporting research on agricultural coverage path planning.

The Fields2Cover project was designed as a tool to optimize path for agricultural robots. The interest generated through open dissemination library may show the agricultural sector's demand for automated solutions.

6.3.2 Agricultural robotics and automated farming

Traditionally, farmers have relied heavily on intuition and personal experience to manage their fields and make day-to-day operational choices (Nuthall, 2012). However, this method carries risks of subjective judgments and inconsistent outcomes due to varying individual skills and experience levels (Nuthall and Old, 2018).

To overcome these limitations, authors like Bochtis et al. (2012) and Zhai et al. (2020) have proposed Decision Support Systems (DSS) to help managing farms. Kukar et al. (2019) compared thirteen DSS surveying their water resources management, climate change

[3]Source: `https://www.star-history.com/#fields2cover/fields2cover&Date`

adaptation, food waste control and applications for agricultural mission planning. These DSS usually rely on another systems to perform the tasks when required, specially in cases where machinery is employed. Those machines need a tool to plan the coverage path, taking into account the data provided by the DSS.

Thanks to the release of open-source tools like Fields2Cover, ROS (Stanford Artificial Intelligence Laboratory et al., 2018), and Nav2 (Macenski et al., 2020), the machinery can be automated at less cost and faster. Although Fields2Cover already provides a ROS 1 interface, Macenski contacted me to work on the integration of Fields2Cover with Nav2 and ROS 2. As Nav2 is a well-recognized project dedicated to integrate path planners into a common package, the inclusion of Fields2Cover was a natural step. Nav2 also provides access to the BehaviorTree.CPP library (Faconti et al., n.d.), which allows to implement logic rules that execute routine according to triggers. This library is perfect for implementing a DSS that launches Fields2Cover when needed. The integration of Fields2Cover in Nav2 is still work in process, and it is being developed in the opennav_coverage repository (Macenski et al., n.d.).

ROS (Stanford Artificial Intelligence Laboratory et al., 2018) is the standard in robotics. Tt provides many drivers to use sensors and actuators out-of-the-box, and allows communicating those messages with other routines. This connection between hardware and software is key, as many companies like AgXeed (*AgXeed website* n.d.), John Deere (*John Deere website* n.d.), and Naio (*Naio technologies website* n.d.), are producing self-driving tractors. Those vehicles have to read their sensors and actuate devices appropriately, whilst operating without human drivers and executing complex field operations based on pre-defined, data-driven plans. The ROS interface of Fields2Cover provides a communication link between the robots and the Fields2Cover library to configure generate coverage paths according to the current status of the field. Automated machinery then follows optimized paths, drastically reducing labor demands and freeing farmers to focus on higher-level farm management and strategic decision-making tasks (Ghobadpour et al., 2022).

It is important to note that farmers tend to be risk-averse when confronted with uncertainty (Yan, Yan, et al., 2023). To adopt new technologies, they need to be confident that an autonomous tractor will not destroy their crop, miss a swath or waste energy by zigzagging inefficiently. Real-world demonstrations, field days, and peer testimonials are important to show that new technology works reliably (Mgendi et al., 2022). Early adopters play a role in proving the concept. Over time, as more users come on board, the technology can become standard practice (DeLay et al., 2022). In fact, adoption of other guidance technologies like auto-steering grew steadily once farmers saw the clear benefits of reduced overlap (McFadden and Rosburg, 2025). The hope is that as autonomous capabilities and coverage path planning software prove themselves, acceptance will follow a similar trajectory. To reduce uncertainty concerned with the adoption of agricultural

robots, the process has to be evidence-driven. For automated systems and DSS to be effective, timely and dynamic data from the field are essential.

6.3.3 Remote sensing for dynamic coverage path planning

Most agricultural coverage paths are planned long before machines reach the field and they rely on geometry and crop data that can be months old (Oksanen and Visala, 2009). By the time the robots start their work, the recorded field geometry may differ from the actual situation, and the crop may have grown. Up-to-date information is therefore essential. Remote sensing can supply such information (Barrientos et al., 2011).

Future developments should enable up-to-date data from different sources to be processed by a Decision Support System (DSS) compliant with Fields2Cover. The DSS comes before the coverage path planner, listens to remote-sensing feeds, schedules each operation and reconfigures the planner with the new information (Fountas et al., 2006). Satellites, aerial drones, in-field probes and on-board sensors deliver live outlines, hazards and crop traits, keeping every mission safe, and grounded in farm practices (Mulla, 2013). Before the operation starts, the DSS checks the latest satellite scenes (Georgi et al., 2018) and aerial imagery (Cyclomedia, 2025). These images reveal border shifts caused by erosion, or earthworks. Field geometry strongly influences cultivation cost, thus precise outlines cut that cost and prevents the path to be planned outside the field (Ptacek et al., 2024).

Not every hazard is visible ahead of time. While the operation runs, lidar and cameras mounted on the robot must recognize unexpected threats and trigger local avoidance (Han et al., 2023). A deer that freezes in the lights (DeVault et al., 2020) forces the robot to pause until the path clears. In a multi-robot fleet a disabled teammate makes the others to split the uncovered area between the remaining robots (Soitinaho et al., 2024). Ground-nesting birds need gentle treatment. Drones mark the nests first (Scholten et al., 2019), then the planner lifts the implement while crossing those points or shifts the path to stay clear. Additional bird-friendly practices mow from the center outward, so wildlife can escape and leave strips of uncultivated land between headlands and the field edge (Hyde and Campbell, 2012).

Remote and in-situ sensors can also support agronomic decisions making. On-the-go probes measure organic matter, main nutrients and volume of water on the ground (Adamchuk et al., 2004). These values can be used by the route planner presented in Chapter 5 to generate coverage path that compact less the soil. This information can also be used to trigger certain operations. For example, soil organic matter is a proxy of nitrogen (Chiriac et al., 2025). When low nitrogen is sensed, a fertilization event can be launched (Olt et al., 2024).

Satellite and UAV imagery can be used to derive yield maps (Mathivanan and Jayagopal, 2022). For example, using satellite images Gaso et al. (2019) predict the wheat grain

yield, and Gaso et al. (2021) predict soybean yield. These predictions can be done using only UAV data (Nevavuori et al., 2020), or combine the data from satellites and UAV to achieve better precision (Maimaitijiang et al., 2021). Managers use these maps to choose fleet size, fix rental dates and place unloading or refueling points when several days and several machines are required on a large farm (Papadopoulos et al., 2024). This planning saves money and optimizes the use of the available resources.

The yield maps can also be used to inform the capacitated route planning problem. Chapter 3 presents a route planner for capacitated operations, but here the yields of the swaths were assumed to known before the planning process. Yields are usually not known before hand, so the route planner should rely on yield estimates provided by a yield map. Yield estimates are stochastic; each cell on the map may be characterized by both a mean and a standard deviation. Underestimation can fill the capacity halfway a swath, whereas overestimation can send the robot half empty to an unloading depot. Repeated replanning may avoid such situation. The machine should measure the real mass flow, communicate with the DSS to update the yield map with the measured data, and recompute the coverage path with Fields2Cover for the remaining area. Such a feedback loop can also be used to improve the prediction model during the operation.

Merging real-time sensing with adaptive planning turns a static coverage task into a living workflow. The system follows current geometry, protects wildlife and deploys inputs with precision. This dynamic responsiveness not only improves efficiency but also leads directly to tangible environmental and economic benefits.

6.3.4 Environmental and economic benefits of agricultural automation

Being the CPP problem a core element of agricultural automation, it brings direct benefits for sustainable agriculture. First, optimizing the path length implies that robots burn less fuel and cut engine hours, which reduces emissions (Guo et al., 2019). At the same time, coverage planning can also help to reduce the skipped and overlapped areas. Skipped/overlapped areas is a huge problem in fields with undulating topography, in which parallel swaths do not fully cover the complete area owing to the commonly used planar projection of the swaths (Hameed et al., 2016). A skipped area means that it is not treated, reducing yield owing to unused space. On the other hand, overlapping area means an area is covered twice or more often, which leads to spending extra resources.

Once the machine follows an efficient path, on-board sensors and farm records reveal where the crop actually needs help. Variable Rate Technology (VRT) uses these data to vary the application of seed, fertilizer, or pesticide on the move (Pawase et al., 2023). Inputs fall only on patches where they are needed, so far less chemicals are left unused. Avoiding surplus chemical to reach the soil surface implies that nutrient runoff into nearby streams or groundwater is also reduced (Getahun et al., 2024). The field grows healthier plants and the farmer pays lower invoices for the used agrochemicals.

Controlled Traffic Farming fixes traffic lanes that are reused season after season (Bochtis et al., 2010a). Concentrating traffic in this way means the bulk of the field is not disturbed by the machinery (Antille et al., 2019). Untouched soil remains loose and porous, which lets rainwater soak in and slows erosion. Better aeration also reduces anaerobic pockets where nitrous oxide tends to form, so greenhouse gas release drops (Pulido-Moncada et al., 2022). In the long term a field with stable structure supports stronger roots and higher biological activity.

The environmental gains translate into solid economic returns. Lower resources use reduce operating costs while precise agronomy holds yields steady. Controlled studies on commercial farms report substantial annual savings that more than cover the cost of agricultural robots (Vahdanjoo et al., 2023). Because profit and environmental stewardship rise together, farmers have a clear incentive to adopt automated coverage planning.

6.3.5 Social and ethical dimensions

The shift to robotic and optimized field operations has broad social implications for farming communities. On one hand, reducing labor burdens by automating coverage tasks can alleviate the strain on farmers and farm workers (Bogue, 2024). Many agricultural regions face labor shortages or an aging farmer population (Bousmah and Grenier, 2022). Automation can help ensure that tasks like spraying and mowing get done within available time windows even with limited labor. This can improve the work-life balance of farmers, who might otherwise spend long days driving machinery. It also enhances safety, reducing human exposure to dangers like pesticide spray (Damalas and Koutroubas, 2016) or equipment rollovers (Irwin et al., 2019). In this sense, the technology contributes to the social sustainability of farming, making the occupation more attractive to a younger, tech-oriented generation. Indeed, the automation of the field might draw new talent into agriculture, seeing it as a high-tech industry (Sahoo et al., 2022).

On the other hand, there could be concerns about job displacement for equipment operators (Marinoudi et al., 2019). In the short term, the more likely scenario is that those roles are replaced by skilled technicians maintaining the systems. Training and re-skilling programs might be needed so that the existing agricultural workforce can take on these new roles (McLandsborough, 2023). Moreover, Samuels and Thomson (2021) find a positive relationship between agricultural mechanization and democracy, stating that this technological revolution may improve people's power on the decision-making process of their communities.

It is also important to consider the equitable access to these advancements. Carolan (2020) predicts that when farm owners consider human labor expensive and scarce, they would want to automate their farms, making farm work underpaid and undesirable. If only large, wealthy farms adopt autonomous equipment, smaller farms could be left at a competitive disadvantage, potentially exacerbating social inequalities in rural areas (Rotz

et al., 2019). Policymakers and cooperatives might consider models to make technology accessible, for example, robot-as-a-service contracting, where a service provider uses these tools on smaller farms for a fee (Milella et al., 2024).

I launched Fields2Cover convinced of its social value. Automating field work, I aimed to lighten farmers' loads and increase food production. However, despite the software being open, the robots that run its coverage paths remain costly. This price wall reduces adoption in low-income regions, while wealthier nations may gain extra market power and widen the global gap. Open-hardware efforts such as Acorn (Fields, 2025), which publishes full build instructions for precision-farming robots, offer a way forward. Partnering Fields2Cover with such projects would reduce entry costs and let farmers everywhere benefits from the project.

After addressing these ethical and social aspects of the agricultural automation, I want to make a final statement in this thesis: Automation will improve people's lives only when we shape it for open reach and steer it by human needs.

References

Acar, E. U., H. Choset, Y. Zhang, and M. Schervish (2003). "Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods". *The International journal of robotics research* 22.7-8, 441–466. DOI: 10.1177/02783649030227002.

Adamchuk, V. I., J. W. Hummel, M. T. Morgan, and S. K. Upadhyaya (2004). "On-the-go soil sensors for precision agriculture". *Computers and electronics in agriculture* 44.1, 71–91. DOI: 10.1016/j.compag.2004.03.002.

Aggarwal, K., A. Hindle, and E. Stroulia (2014). "Co-evolution of project documentation and popularity within github". In: *Proceedings of the 11th working conference on mining software repositories*, 360–363. DOI: 10.1145/2597073.2597120.

AgXeed website (n.d.). https://agxeed.com/. Accessed: January 9, 2026.

Ajibade, S., B. Simon, M. Gulyas, and C. Balint (2023). "Sustainable intensification of agriculture as a tool to promote food security: A bibliometric analysis". *Frontiers in Sustainable Food Systems* 7, 1101528. DOI: 10.3389/fsufs.2023.1101528.

Alejo, D. (2024). Private communication. Fields2Cover was used as part of the Sky-Eye team on MBZIRC Challenge 2023.

Ambrósio, L., H. Linhares, J. M. N. David, R. Braga, W. Arbex, M. M. Campos, and R. Capilla (2021). "Enhancing the reuse of scientific experiments for agricultural software ecosystems". *Journal of Grid Computing* 19.4, 44. DOI: 10.1007/s10723-021-09583-x.

Antille, D., S. Peets, J. Galambošová, G. Botta, V. Rataj, M. Macak, J. Tullberg, W. Chamen, D. White, P. Misiewicz, et al. (2019). "Soil compaction and controlled traffic farming in arable and grass cropping systems". *Agronomy Research* 17.3, 653–682. DOI: 10.15159/AR.19.133.

Ariza-Sentís, M., S. Vélez, R. Martínez-Peña, H. Baja, and J. Valente (2024). "Object detection and tracking in Precision Farming: a systematic review". *Computers and Electronics in Agriculture* 219, 108757. DOI: 10.1016/j.compag.2024.108757.

Backman, J., P. Piirainen, and T. Oksanen (2015). "Smooth turning path generation for agricultural vehicles in headlands". *Biosystems Engineering* 139, 76–86. DOI: 10.1016/j.biosystemseng.2015.08.005.

Baehnemann, R., L. Liu, and D. Kleiser (n.d.). *Ethz-asl/polygon_coverage_planning GitHub*. https://github.com/ethz-asl/polygon_coverage_planning. Accessed: January 9, 2026.

Barrientos, A., J. Colorado, J. d. Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente (2011). "Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots". *Journal of Field Robotics* 28.5, 667–689. DOI: 10.1002/rob.20403.

Baykal, O., E. Tari, Z. Çoşkun, and M. Şahin (1997). "New transition curve joining two straight lines". *Journal of Transportation Engineering* 123.5, 337–345. DOI: 10.1061/(ASCE)0733-947X(1997)123:5(337).

Beazley, D. M. et al. (1996). "SWIG: An Easy to Use Tool for Integrating Scripting Languages with C and C++." In: *Tcl/Tk Workshop*. Vol. 43, 74.

Bochtis, D. and T. Oksanen (2009). "Combined coverage path planning for field operations". In: *Proc. Joint International Agricultural Conference, JIAC*, 521–527. DOI: 10.3920/978-90-8686-664-9.

Bochtis, D., C. Sørensen, P. Busato, I. Hameed, E. Rodias, O. Green, and G. Papadakis (2010a). "Tramline establishment in controlled traffic farming based on operational machinery cost". *biosystems engineering* 107.3, 221–231. DOI: 10.1016/j.biosystemseng.2010.08.004.

Bochtis, D., C. Sørensen, and S. Vougioukas (2010b). "Path planning for in-field navigation-aiding of service units". *Computers and Electronics in Agriculture* 74.1, 80–90. DOI: 10.1016/j.compag.2010.06.008.

Bochtis, D. and S. Vougioukas (2008). "Minimising the non-working distance travelled by machines operating in a headland field pattern". *Biosystems engineering* 101.1, 1–12. DOI: 10.1016/j.biosystemseng.2008.06.008.

Bochtis, D., H. W. Griepentrog, S. Vougioukas, P. Busato, R. Berruto, and K. Zhou (2015). "Route planning for orchard operations". *Computers and electronics in agriculture* 113, 51–60. DOI: 10.1016/j.compag.2014.12.024.

Bochtis, D. D., C. G. Sørensen, P. Busato, and R. Berruto (2013). "Benefits from optimal route planning based on B-patterns". *Biosystems Engineering* 115.4, 389–395. DOI: 10.1016/j.biosystemseng.2013.04.006.

Bochtis, D. D., C. G. Sørensen, and O. Green (2012). "A DSS for planning of soil-sensitive field operations". *Decision Support Systems* 53.1, 66–75. DOI: 10.1016/j.dss.2011.12.005.

Bogue, R. (2024). "Robots addressing agricultural labour shortages and environmental issues". *Industrial Robot: the international journal of robotics research and application* 51.1, 1–6. DOI: 10.1108/IR-10-2023-0245.

Borges, H. and M. T. Valente (2018). "What's in a github star? understanding repository starring practices in a social coding platform". *Journal of Systems and Software* 146, 112–129. DOI: 10.1016/j.jss.2018.09.016.

Bormann, R., F. Jordan, J. Hampp, and M. Hägele (2018). "Indoor coverage path planning: Survey, implementation, analysis". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1718–1725. DOI: 10.1109/ICRA.2018.8460566.

Bostelmann-Arp, L., C. Steup, and S. Mostaghim (2023). "Multi-objective seed curve optimization for coverage path planning in precision farming". In: *Proceedings of the Genetic and Evolutionary Computation Conference*, 1312–1320. DOI: 10.1145/3583131.3590490.

Bousmah, I. and G. Grenier (2022). "Labor shortages and immigration: The case of the Canadian agriculture sector". *Agribusiness* 38.1, 220–235. DOI: 10.1002/agr.21719.

Brooker, R. W., A. E. Bennett, W.-F. Cong, T. J. Daniell, T. S. George, P. D. Hallett, C. Hawes, P. P. Iannetta, H. G. Jones, A. J. Karley, et al. (2015). "Improving intercropping: a synthesis of research in agronomy, plant physiology and ecology". *New Phytologist* 206.1, 107–117. DOI: 10.1111/nph.13132.

Cai, Z., C. R. Cardenas, K. Leo, C. Zhang, K. Backman, H. Li, B. Li, M. Ghorbanali, S. Datta, L. Qu, et al. (2024). "NEUSIS: A Compositional Neuro-Symbolic Framework for Autonomous Perception, Reasoning, and Planning in Complex UAV Search Missions". *arXiv preprint arXiv:2409.10196*. DOI: 10.48550/arXiv.2409.10196.

Calleja-Huerta, A., M. Lamandé, O. Green, and L. Munkholm (2023). "Impacts of load and repeated wheeling from a lightweight autonomous field robot on the physical properties of a loamy sand soil". *Soil and Tillage Research* 233, 105791. DOI: 10.1016/j.still.2023.105791.

Campanelli, G., I. Iocola, F. Leteo, F. Montemurro, C. Platani, E. Testani, and S. Canali (2023). "Strip cropping in organically managed vegetable systems: agronomic and environmental effects". *Renewable Agriculture and Food Systems* 38, e31. DOI: 10.1017/S1742170523000248.

Carolan, M. (2020). "Automated agrifood futures: Robotics, labor and the distributive politics of digital agriculture". *The Journal of Peasant Studies* 47.1, 184–207. DOI: 10.1080/03066150.2019.1584189.

Chakraborty, S., D. Elangovan, P. L. Govindarajan, M. F. ELnaggar, M. M. Alrashed, and S. Kamel (2022). "A comprehensive review of path planning for agricultural ground robots". *Sustainability* 14.15, 9156. DOI: 10.3390/su14159156.

Chatzisavvas, A., M. Louta, and M. Dasygenis (2023). "Path planning for agricultural ground robots–Review". In: *AIP Conference Proceedings*. Vol. 2909. 1. AIP Publishing. DOI: 10.3390/su14159156.

Chen, G., C. K. Lo, and L. Hu (2020). "Sustaining online academic discussions: Identifying the characteristics of messages that receive responses". *Computers & Education* 156, 103938. DOI: 10.1016/j.compedu.2020.103938.

Chiriac, O. P., M. Pittarello, B. Moretti, and L. Zavattaro (2025). "Factors influencing nitrogen derived from soil organic matter mineralisation: Results from a long-term experiment". *Agriculture, Ecosystems & Environment* 381, 109444. DOI: 10.1016/j.agee.2024.109444.

Choset, H. (2000). "Coverage of known spaces: The boustrophedon cellular decomposition". *Autonomous Robots* 9.3, 247–253. DOI: 10.1023/A:1008958800904.

Christiaensen, L., Z. Rutledge, and J. E. Taylor (2020). "The future of work in agriculture: Some reflections". *World Bank Policy Research Working Paper* 9193.

Clephas, T., C. López, and Nobleo (n.d.). *Nobleo/full_coverage_path_planner GitHub*. https://github.com/nobleo/full_coverage_path_planner. Accessed: January 9, 2026.

Collaboration, O. S. (2015). "Estimating the reproducibility of psychological science". *Science* 349.6251, aac4716. DOI: 10.1126/science.aac4716.

Crüwell, S., J. van Doorn, A. Etz, M. C. Makel, H. Moshontz, J. C. Niebaum, A. Orben, S. Parsons, and M. Schulte-Mecklenbeck (2019). "Seven easy steps to open science". *Zeitschrift für Psychologie*. DOI: 10.1027/2151-2604/a000387.

Cyclomedia (2025). *Ortho Aerial data*. https://www.cyclomedia.com/en/producten/data-visualisatie/aerial-data.

Dabbish, L., C. Stuart, J. Tsay, and J. Herbsleb (2012). "Social coding in GitHub: transparency and collaboration in an open software repository". In: *Proceedings of the ACM 2012 conference on computer supported cooperative work*, 1277–1286. DOI: 10.1145/2145204.2145396.

Damalas, C. A. and S. D. Koutroubas (2016). *Farmers' exposure to pesticides: toxicity types and ways of prevention*. DOI: 10.3390/toxics4010001.

de Bruin, S., P. Lerink, A. Klompe, T. van der Wal, and S. Heijting (2009). "Spatial optimisation of cropped swaths and field margins using GIS". *Computers and Electronics in Agriculture* 68.2, 185–190. DOI: 10.1016/j.compag.2009.06.001.

de Bruin, S., P. Lerink, I. J. La Riviere, and B. Vanmeulebrouk (2014). "Systematic planning and cultivation of agricultural fields using a geo-spatial arable field optimization service: Opportunities and obstacles". *Biosystems Engineering* 120, 15–24. DOI: 10.1016/j.biosystemseng.2013.07.009.

DeLay, N. D., N. M. Thompson, and J. R. Mintert (2022). "Precision agriculture technology adoption and technical efficiency". *Journal of Agricultural Economics* 73.1, 195–219. DOI: 10.1111/1477-9552.12440.

DeVault, T. L., T. W. Seamans, and B. F. Blackwell (2020). "Frontal vehicle illumination via rear-facing lighting reduces potential for collisions with white-tailed deer". *Ecosphere* 11.7, e03187. DOI: 10.1002/ecs2.3187.

Dubins, L. E. (1957). "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents". *American Journal of mathematics* 79.3, 497–516. DOI: 10.2307/2372560.

Economic, U. N. D. of and S. Affairs (2021). *World Population Prospects 2017 - Volume I: Comprehensive Tables*. United Nations. DOI: 10.18356/9789210001014.

Elliott, K. C. (2022). "A taxonomy of transparency in science". *Canadian Journal of Philosophy* 52.3, 342–355. DOI: 10.1017/can.2020.21.

ESRI (2025). *ESRI Imagery*. https://www.esri.com/en-us/home. Accessed: 2025-03-06.

Essen, R. van, E. van Henten, L. Kooistra, and G. Kootstra (2025). "Adaptive path planning for efficient object search by UAVs in agricultural fields". *Smart Agricultural Technology*, 101075. DOI: 10.1016/j.atech.2025.101075.

Faconti, D., M. Colledanchise, E. de Oliveira, and A. Sasine (n.d.). *BehaviorTree.CPP GitHub*. https://github.com/BehaviorTree/BehaviorTree.CPP. Accessed: January 9, 2026.

FAO, I. (2015). "Food and Agricultural Organization of the United Nations and Intergovernmental Technical Panel on Soils, Rome, Italy". *Status of the World's Soil Resources (SWSR)–Main Report*.

Farm hack website (n.d.). https://farmhack.org. Accessed: 11-06-2025.

Fecher, B. and S. Friesike (2014). *Open science: one term, five schools of thought*. Springer International Publishing, 17–47. DOI: 10.1007/978-3-319-00026-8_2.

Feldmann, J., S. Dickebohm, and F. Tepe (2024). *Steinesammler-Roboter – Interdisziplinäres Projekt*. Hochschule Osnabrück University. https://www.hs-osnabrueck.de/loesungen-fuer-morgen/mehr-loesungen-fuer-morgen/steinesammler-roboter/.

Field, T., J. Leibs, J. Bowman, and D. Thomas (n.d.). *Rosbag package*. http://wiki.ros.org/rosbag. Accessed: January 9, 2026.

Fields, T. (2025). *Acorn precision farming rover*. Github. https://github.com/Twisted-Fields/acorn-precision-farming-rover.

Filip, M., T. Zoubek, R. Bumbalek, P. Cerny, C. E. Batista, P. Olsan, P. Bartos, P. Kriz, M. Xiao, A. Dolan, et al. (2020). "Advanced computational methods for agriculture machinery movement optimization with applications in sugarcane production". *Agriculture* 10.10, 434. DOI: 10.3390/agriculture10100434.

Finger, R., A. Henningsen, J. Höhler, R. Huber, J. Rommel, and C. Grebitus (2025). "Open science in agricultural economics". *Q open* 5.3, qoae029. DOI: 10.1093/qopen/qoae029.

Floyd, R. W. (1962). "Algorithm 97: shortest path". *Communications of the ACM* 5.6, 345. DOI: 10.1145/367766.368168.

Fountas, S., D. Wulfsohn, B. S. Blackmore, H. Jacobsen, and S. M. Pedersen (2006). "A model of decision-making and information flows for information-intensive agriculture". *Agricultural Systems* 87.2, 192–210. DOI: 10.1016/j.agsy.2004.12.003.

Fraichard, T. and A. Scheuer (2004). "From Reeds and Shepp's to continuous-curvature paths". *IEEE Transactions on Robotics* 20.6, 1025–1035. DOI: 10.1109/TRO.2004.833789.

Galceran, E. and M. Carreras (2013). "A survey on coverage path planning for robotics". *Robotics and Autonomous systems* 61.12, 1258–1276. DOI: 10.1016/j.robot.2013.09.004.

Gaso, D. V., A. G. Berger, and V. S. Ciganda (2019). "Predicting wheat grain yield and spatial variability at field scale using a simple regression or a crop model in conjunction with Landsat images". *Computers and Electronics in Agriculture* 159, 75–83. DOI: 10.1016/j.compag.2019.02.026.

Gaso, D. V., A. de Wit, A. G. Berger, and L. Kooistra (2021). "Predicting within-field soybean yield variability by coupling Sentinel-2 leaf area index with a crop growth model". *Agricultural and forest meteorology* 308, 108553. DOI: 10.1016/j.agrformet.2021.108553.

Gasso, V., C. A. Sørensen, F. W. Oudshoorn, and O. Green (2013). "Controlled traffic farming: A review of the environmental impacts". *European Journal of Agronomy* 48, 66–73. DOI: 10.1016/j.eja.2013.02.002.

GDAL/OGR contributors (2022). *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation. DOI: 10.5281/zenodo.5884351. URL: https://gdal.org.

Georgi, C., D. Spengler, S. Itzerott, and B. Kleinschmit (2018). "Automatic delineation algorithm for site-specific management zones based on satellite remote sensing data". *Precision agriculture* 19, 684–707. DOI: 10.1007/s11119-017-9549-y.

Getahun, S., H. Kefale, and Y. Gelaye (2024). "Application of precision agriculture technologies for sustainable crop production and environmental sustainability: A systematic review". *The Scientific World Journal* 2024.1, 2126734. DOI: 10.1155/2024/2126734.

Ghobadpour, A., G. Monsalve, A. Cardenas, and H. Mousazadeh (2022). "Off-road electric vehicles and autonomous robots in agricultural sector: trends, challenges, and opportunities". *Vehicles* 4.3, 843–864. DOI: 10.3390/vehicles4030047.

Google (n.d.). *Protobuf on C++*. https://developers.google.com/protocol-buffers. Accessed: January 9, 2026.

Greenzie (n.d.). *Greenzie/boustrophedon_planner GitHub*. https://github.com/Greenzie/boustrophedon_planner. Accessed: January 9, 2026.

Grossman, R. L., Y. Gu, J. Mambretti, M. Sabala, A. Szalay, and K. White (2010). "An overview of the open science data cloud". In: *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, 377–384. DOI: 10.1145/1851476.1851533.

Guevara, L., M. M. Michałek, and F. A. Cheein (2020). "Headland turning algorithmization for autonomous N-trailer vehicles in agricultural scenarios". *Computers and Electronics in Agriculture* 175, 105541. DOI: 10.1016/j.compag.2020.105541.

Guo, D., J. Wang, J. B. Zhao, F. Sun, S. Gao, C. D. Li, M. H. Li, and C. C. Li (2019). "A vehicle path planning method based on a dynamic traffic network that considers fuel consumption and emissions". *Science of the Total Environment* 663, 935–943. DOI: 10.1016/j.scitotenv.2019.01.222.

Ha, J., C. Lee, A. Pal, G. Park, and H. Kim (2018). "Development of optimized headland turning mechanism on an agricultural robot for Korean garlic farms". *Journal of Biosystems Engineering* 43.4, 273–284. DOI: 10.5307/JBE.2018.43.4.273.

Hameed, I., D. Bochtis, C. G. Sørensen, A. L. Jensen, and R. Larsen (2013a). "Optimized driving direction based on a three-dimensional field representation". *Computers and electronics in agriculture* 91, 145–153. DOI: 10.1016/j.compag.2012.12.009.

Hameed, I., D. Bochtis, C. G. Sørensen, and M. Nøremark (2010). "Automated generation of guidance lines for operational field planning". *Biosystems engineering* 107.4, 294–306. DOI: 10.1016/j.biosystemseng.2010.09.001.

Hameed, I. A. (2017). "Coverage path planning software for autonomous robotic lawn mower using Dubins' curve". In: *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 517–522. DOI: 10.1109/RCAR.2017.8311915.

Hameed, I. A., D. Bochtis, and C. Sorensen (2011). "Driving angle and track sequence optimization for operational path planning using genetic algorithms". *Applied Engineering in Agriculture* 27.6, 1077–1086. DOI: 10.13031/2013.40615.

Hameed, I. A., D. Bochtis, and C. A. Sørensen (2013b). "An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas". *International journal of advanced robotic systems* 10.5, 231. DOI: 10.5772/56248.

Hameed, I. A., A. la Cour-Harbo, and O. L. Osen (2016). "Side-to-side 3D coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths". *Robotics and Autonomous Systems* 76, 36–45. DOI: 10.1016/j.robot.2015.11.009.

Han, C., W. Wu, X. Luo, and J. Li (2023). "Visual navigation and obstacle avoidance control for agricultural robots via LiDAR and camera". *Remote sensing* 15.22, 5402. DOI: 10.3390/rs15225402.

Hauggaard-Nielsen, H., A. Johansen, M. S. Carter, P. Ambus, and E. S. Jensen (2012). "Strip cropping of alternating perennial grass–clover and annual rye–vetch intercrops

when grown within an organic farming system". *Field Crops Research* 136, 1–11. DOI: 10.1016/j.fcr.2012.07.003.

He, Z., Y. Bao, Q. Yu, P. Lu, Y. He, and Y. Liu (2023). "Dynamic path planning method for headland turning of unmanned agricultural vehicles". *Computers and Electronics in Agriculture* 206, 107699. DOI: 10.1016/j.compag.2023.107699.

Hernández-Ochoa, I. M., T. Gaiser, K.-C. Kersebaum, H. Webber, S. J. Seidel, K. Grahmann, and F. Ewert (2022). "Model-based design of crop diversification through new field arrangements in spatially heterogeneous landscapes. A review". *Agronomy for Sustainable Development* 42.4, 74. DOI: 10.1007/s13593-022-00805-4.

Höffmann, M., S. Patel, and C. Büskens (2022). "Weight-Optimized NURBS Curves: Headland Paths for Nonholonomic Field Robots", 81–85. DOI: 10.1109/ICARA55094.2022.9738525.

– (2023). "Optimal Coverage Path Planning for Agricultural Vehicles with Curvature Constraints". *Agriculture* 13.11, 2112. DOI: 10.3390/agriculture13112112.

– (2024). "Optimal guidance track generation for precision agriculture: A review of coverage path planning techniques". *Journal of Field Robotics*. DOI: 10.1002/rob.22286.

Hossain, A., T. J. Krupnik, J. Timsina, M. G. Mahboob, A. K. Chaki, M. Farooq, R. Bhatt, S. Fahad, and M. Hasanuzzaman (2020). "Agricultural land degradation: processes and problems undermining future food security". In: *Environment, climate, plant and vegetation growth*. Springer, 17–61. DOI: 10.1007/978-3-030-49732-3_2.

Huang, W. H. (2001). "Optimal line-sweep-based decompositions for coverage algorithms". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*. Vol. 1. IEEE, 27–32. DOI: 10.1109/ROBOT.2001.932525.

Hyde, D. and S. Campbell (2012). "Agricultural practices that conserve grassland birds". *Michigan State University Extension: East Lansing, MI, USA*.

Intel (n.d.). *oneAPI Threading Building Blocks*. https://github.com/oneapi-src/oneTBB. Accessed: January 9, 2026.

Ioannidis, J. P. (2005). "Why most published research findings are false". *PLoS medicine* 2.8, e124. DOI: 10.1371/journal.pmed.0020124.

IPA, F. (n.d.). *Ipa320/ipa_coverage_planning GitHub*. https://github.com/ipa320/ipa_coverage_planning. Accessed: January 9, 2026.

Irwin, A., L. Caruso, and I. Tone (2019). "Thinking ahead of the tractor: Driver safety and situation awareness". *Journal of agromedicine* 24.3, 288–297. DOI: 10.1080/1059924X.2019.1604279.

Janulevičius, A. and K. Giedra (2009). "The slippage of the driving wheels of a tractor in a cultivated soil and stubble". *Transport* 24.1, 14–20. DOI: 10.3846/1648-4142.2009.24.14-20.

Jensen, M. F., D. Bochtis, and C. G. Sørensen (2015). "Coverage planning for capacitated field operations, part II: Optimisation". *Biosystems Engineering* 139, 149–164. DOI: 10.1016/j.biosystemseng.2015.07.002.

Jensen-Nau, K. R., T. Hermans, and K. K. Leang (2020). "Near-optimal area-coverage path planning of energy-constrained aerial robots with application in autonomous environmental monitoring". *IEEE Transactions on Automation Science and Engineering* 18.3, 1453–1468. DOI: 10.1109/TASE.2020.3016276.

Jeon, C.-W., H.-J. Kim, C. Yun, X. Han, and J. H. Kim (2021). "Design and validation testing of a complete paddy field-coverage path planner for a fully autonomous tillage tractor". *Biosystems Engineering* 208, 79–97. DOI: 10.1016/j.biosystemseng.2021.05.008.

Jeon, C.-W., H.-J. Kim, C. Yun, S.-J. Park, Y. B. Hwang, and X. Han (2024). "Autonomous paddy field puddling and leveling operations based on full-coverage path generation and tracking". *Precision Agriculture* 25.1, 235–256. DOI: 10.1007/s11119-023-10066-0.

Jiaping, R. (n.d.). *RJJxp/CoveragePlanning GitHub*. https://github.com/RJJxp/CoveragePlanning. Accessed: January 9, 2026.

Jin, J. (2009). "Optimal field coverage path planning on 2D and 3D surfaces". DOI: 10.31274/etd-180810-3122.

Jin, J. and L. Tang (2010). "Optimal coverage path planning for arable farming on 2D surfaces". *Transactions of the ASABE* 53.1, 283–295. DOI: 10.13031/2013.29488.

– (2011). "Coverage path planning on three-dimensional terrain for arable farming". *Journal of field robotics* 28.3, 424–440. DOI: 10.1002/rob.20388.

John Deere website (n.d.). https://www.deere.com/en/autonomous/. Accessed: January 9, 2026.

Juman, M. A., Y. W. Wong, R. K. Rajkumar, and C. Y. H'ng (2017). "An integrated path planning system for a robot designed for oil palm plantations". In: *TENCON 2017-2017 IEEE Region 10 Conference*. IEEE, 1048–1053. DOI: 10.1109/TENCON.2017.8228012.

Keller, T. and J. Arvidsson (2016). "A model for prediction of vertical stress distribution near the soil surface below rubber-tracked undercarriage systems fitted on agricultural vehicles". *Soil and Tillage Research* 155, 116–123. DOI: 10.1016/j.still.2015.07.014.

Keller, T., P. Défossez, P. Weisskopf, J. Arvidsson, and G. Richard (2007). "SoilFlex: A model for prediction of soil stresses and soil compaction due to agricultural field traffic including a synthesis of analytical approaches". *Soil and Tillage Research* 93.2, 391–411. DOI: 10.1016/j.still.2006.05.012.

Keller, T., M. Sandin, T. Colombi, R. Horn, and D. Or (2019). "Historical increase in agricultural machinery weights enhanced soil stress levels and adversely affected soil functioning". *Soil and Tillage Research* 194, 104293. DOI: 10.1016/j.still.2019.104293.

Khosravani Moghadam, E., M. Vahdanjoo, A. L. Jensen, M. Sharifi, and C. A. G. Sørensen (2020). "An Arable Field for Benchmarking of Metaheuristic Algorithms for Capacitated Coverage Path Planning Problems". *Agronomy* 10.10, 1454. DOI: 10.3390/agronomy10101454.

Kistowski, J. v., J. A. Arnold, K. Huppler, K.-D. Lange, J. L. Henning, and P. Cao (2015). "How to build a benchmark". In: *Proceedings of the 6th ACM/SPEC international conference on performance engineering*, 333–336. DOI: 10.1145/2668930.2688819.

Koolen, A. and H. Kuipers (1983). *Agricultural Soil Mechanics*. English. Advanced series in agricultural sciences 13. Springer Verlag.

Kottman, C. (2002). *OGC Simple Features*.

Kukar, M., P. Vračar, D. Košir, D. Pevec, Z. Bosnić, et al. (2019). "AgroDSS: A decision support system for agriculture and farming". *Computers and Electronics in Agriculture* 161, 260–271. DOI: 10.1016/j.compag.2018.04.001.

Larson, W., S. Gupta, and R. Useche (1980). "Compression of agricultural soils from eight soil orders". *Soil Science Society of America Journal* 44.3, 450–457. DOI: 10.2136/sssaj1980.03615995004400030002x.

Latombe, J.-C. (1991). *Robot Motion Planning*. Springer Science & Business Media. DOI: 10.1007/978-1-4615-4022-9.

Li, Y., Z. Guo, F. Shuang, M. Zhang, and X. Li (2022). "Key technologies of machine vision for weeding robots: A review and benchmark". *Computers and Electronics in Agriculture* 196, 106880. DOI: 10.1016/j.compag.2022.106880.

Linker, R. and T. Blass (2008). "Path-planning algorithm for vehicles operating in orchards". *Biosystems engineering* 101.2, 152–160. DOI: 10.1016/j.biosystemseng.2008.06.002.

Lu, E., L. Xu, Y. Li, Z. Tang, and Z. Ma (2020). "Modeling of working environment and coverage path planning method of combine harvesters". *International Journal of Agricultural and Biological Engineering* 13.2, 132–137. DOI: 10.25165/j.ijabe.20201302.5210.

Ma, K.-Z., S.-G. Hao, H.-Y. Zhao, and L. Kang (2007). "Strip cropping wheat and alfalfa to improve the biological control of the wheat aphid Macrosiphum avenae by the mite Allothrombium ovatum". *Agriculture, ecosystems & environment* 119.1-2, 49–52. DOI: 10.1016/j.agee.2006.06.009.

Macenski, S., F. Martín, R. White, and J. G. Clavero (2020). "The marathon 2: A navigation system". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2718–2725. DOI: 10.1109/IROS45743.2020.9341207.

Macenski, S., M. Morcos, and B. Bush (n.d.). *Open Navigation's Nav2 Complete Coverage GitHub*. https://github.com/open-navigation/opennav_coverage. Accessed: January 9, 2026.

Macenski, S., T. Foote, B. Gerkey, C. Lalancette, and W. Woodall (2022). "Robot operating system 2: Design, architecture, and uses in the wild". *Science robotics* 7.66, eabm6074. DOI: 10.1126/scirobotics.abm6074.

Machmuller, M. B., M. G. Kramer, T. K. Cyle, N. Hill, D. Hancock, and A. Thompson (2015). "Emerging land use practices rapidly increase soil organic matter". *Nature Communications* 6.1, 6995. DOI: 10.1038/ncomms7995.

Maimaitijiang, M., V. Sagan, and F. B. Fritschi (2021). "Crop yield prediction using satellite/UAV synergy and machine learning". In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 6276–6279. DOI: 10.1109/IGARSS47720.2021.9554735.

Marinoudi, V., C. G. Sørensen, S. Pearson, and D. Bochtis (2019). "Robotics and labour in agriculture. A context consideration". *Biosystems Engineering* 184, 111–121. DOI: 10.1016/j.biosystemseng.2019.06.013.

Martin-Gorriz, B., J. A. Zabala, V. Sánchez-Navarro, B. Gallego-Elvira, V. Martínez-García, F. Alcon, and J. F. Maestre-Valero (2022). "Intercropping Practices in Mediterranean Mandarin Orchards from an Environmental and Economic Perspective". *Agriculture* 12.5, 574. DOI: 10.3390/agriculture12050574.

Mathivanan, S. K. and P. Jayagopal (2022). "Utilizing satellite and UAV data for crop yield prediction and monitoring through deep learning". *Acta Geophysica* 70.6, 2991–3004. DOI: 10.1007/s11600-022-00911-7.

McFadden, J. and A. Rosburg (2025). "Automation, productivity, and profitability: evidence from farmers' use of auto-steer systems". In: *Precision agriculture'25*. Wageningen Academic, 1075–1081. DOI: 10.1163/9789004725232_141.

McLandsborough, A. (2023). "Nothing Runs like a (Autonomous) Deere: How Government-Sponsored Reskilling Can Cushion the Blow of Technological Advancement for Kansan Farm Workers". *Kan. JL & Pub. Pol'y* 33, 197.

McPhee, J. E., D. L. Antille, J. N. Tullberg, R. B. Doyle, and M. Boersma (2020). "Managing soil compaction – A choice of low-mass autonomous vehicles or controlled traffic?" *Biosystems Engineering* 195, 227–241. DOI: 10.1016/j.biosystemseng.2020.05.006.

Meuth, R. J. and D. C. Wunsch (2008). "Divide and conquer evolutionary TSP solution for vehicle path planning". In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE, 676–681. DOI: 10.1109/CEC.2008.4630868.

Mgendi, G., S. Mao, and F. Qiao (2022). "Does agricultural training and demonstration matter in technology adoption? The empirical evidence from small rice farmers in Tanzania". *Technology in Society* 70, 102024. DOI: 10.1016/j.techsoc.2022.102024.

Mier, G., J. Valente, and S. de Bruin (2023a). "Optimizing agricultural coverage path to minimize soil compaction". In: *Precision agriculture'23*. Wageningen Academic Publishers, 467–472. DOI: 10.3920/978-90-8686-947-3.

Mier, G. (2022a). *Could you roast my project, Fields2Cover library*. Reddit post. https://www.reddit.com/r/cpp/comments/v944ju/could_you_roast_my_project_fields2cover_library/.

– (2022b). *Fields2Cover release announcement*. LinkedIn post. https://www.linkedin.com/posts/gonzalo-mier_tractor-fields2cover-opensource-activity-6940600294858756096-ntxh.

– (2022c). *Fields2Cover release announcement*. Reddit post. https://www.reddit.com/r/Python/comments/x4czll/fields2cover_the_first_opensource_coverage_path/.

– (2022d). *Fields2Cover release announcement*. Dev.to post. https://dev.to/gonzalomier/how-to-install-fields2cover-on-linux-for-coverage-path-planning-on-agriculture-3dc0.

Mier, G., A. M. Casado Faulí, J. Valente, and S. de Bruin (2025a). "Fields2Benchmark: An open-source benchmark for coverage path planning methods in agriculture". *Smart Agricultural Technology*, 101156. DOI: 10.1016/j.atech.2025.101156.

Mier, G., R. Fennema, J. Valente, and S. de Bruin (2025b). "Continuous Curvature Path Planning for Headland Coverage With Agricultural Robots". *Journal of Field Robotics* 42.3, 641–656. DOI: 10.1002/rob.22489.

Mier, G., J. Valente, and S. de Bruin (2023b). "Fields2Cover: An open-source coverage path planning library for unmanned agricultural vehicles". *IEEE Robotics and Automation Letters* 8.4, 2166–2172. DOI: 10.1109/LRA.2023.3248439.

Mier, G., S. Vélez, J. Valente, and S. de Bruin (2025c). "Soil2Cover: Coverage path planning minimizing soil compaction for sustainable agriculture". *Precision Agriculture* 26.4, 1–21. DOI: 10.1007/s11119-025-10250-4.

Milella, A., S. Rilling, A. Rana, R. Galati, A. Petitti, M. Hoffmann, J. L. Stanly, and G. Reina (2024). "Robot-as-a-service as a new paradigm in precision farming". *IEEE access* 12, 47942–47949. DOI: 10.1109/ACCESS.2024.3381511.

Mileusnić, Z. I., E. Saljnikov, R. L. Radojević, and D. V. Petrović (2022). "Soil compaction due to agricultural machinery impact". *Journal of Terramechanics* 100, 51–60. DOI: 10.1016/j.jterra.2021.12.002.

Mukherjee, A. and S. Stern (2009). "Disclosure or secrecy? The dynamics of open science". *International Journal of Industrial Organization* 27.3, 449–462. DOI: 10.1016/j.ijindorg.2008.11.005.

Mulla, D. J. (2013). "Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps". *Biosystems engineering* 114.4, 358–371. DOI: 10.1016/j.biosystemseng.2012.08.009.

Munafò, M. R., B. A. Nosek, D. V. Bishop, K. S. Button, C. D. Chambers, N. Percie du Sert, U. Simonsohn, E.-J. Wagenmakers, J. J. Ware, and J. P. Ioannidis (2017). "A manifesto for reproducible science". *Nature human behaviour* 1.1, 0021. DOI: 10.1038/s41562-016-0021.

Musker, R. and B. Schaap (2018). "Global Open Data in Agriculture and Nutrition (GODAN) initiative partner network analysis". *F1000Research* 7, 47. DOI: 10.12688/f1000research.13044.1.

Naio technologies website (n.d.). https://www.naio-technologies.com. Accessed: January 9, 2026.

Nawaz, M. F., G. Bourrie, and F. Trolard (2013). "Soil compaction impact and modelling. A review". *Agronomy for sustainable development* 33, 291–309. DOI: 10.1007/s13593-011-0071-8.

Nawaz, M. M., M. A. Noor, H. Latifmanesh, X. Wang, W. Ma, and W. Zhang (2023). "Field traffic-induced soil compaction under moderate machine-field conditions affects soil properties and maize yield on sandy loam soil". *Frontiers in Plant Science* 14, 1002943. DOI: 10.3389/fpls.2023.1002943.

Nevavuori, P., N. Narra, P. Linna, and T. Lipping (2020). "Crop yield prediction using multitemporal UAV data and spatio-temporal deep learning models". *Remote sensing* 12.23, 4000. DOI: 10.3390/rs12234000.

Nielsen, L. D., I. Sung, and P. Nielsen (2019). "Convex decomposition for a coverage path planning for autonomous vehicles: Interior extension of edges". *Sensors* 19.19, 4165. DOI: 10.3390/s19194165.

Nilsson, R. S. and K. Zhou (2020). "Method and bench-marking framework for coverage path planning in arable farming". *Biosystems Engineering* 198, 248–265. DOI: 10.1016/j.biosystemseng.2020.08.007.

Nørremark, M., R. S. Nilsson, and C. A. G. Sørensen (2022). "In-Field Route Planning Optimisation and Performance Indicators of Grain Harvest Operations". *Agronomy* 12.5, 1151. DOI: 10.3390/agronomy12051151.

Nuthall, P. L. (2012). "The intuitive world of farmers–the case of grazing management systems and experts". *Agricultural Systems* 107, 65–73. DOI: 10.1016/j.agsy.2011.11.006.

Nuthall, P. and K. Old (2018). "Intuition, the farmers' primary decision process. A review and analysis". *Journal of Rural Studies* 58, 28–38. DOI: 10.1016/j.jrurstud.2017.12.012.

O'Sullivan, M. and E. Robertson (1996). "Critical state parameters from intact samples of two agricultural topsoils". *Soil and Tillage Research* 39.3-4, 161–173. DOI: 10.1016/S0167-1987(96)01068-9.

Oksanen, T. et al. (2007). *Path planning algorithms for agricultural field machines*. Helsinki University of Technology.

Oksanen, T. and A. Visala (2009). "Coverage path planning algorithms for agricultural field machines". *Journal of field robotics* 26.8, 651–668. DOI: 10.1002/rob.20300.

Oliveira, L. F., A. P. Moreira, and M. F. Silva (2021). "Advances in agriculture robotics: A state-of-the-art review and challenges ahead". *Robotics* 10.2, 52. DOI: 10.3390/robotics10020052.

Olt, J., O. Liivapuu, I. Virro, and T. Lillerand (2024). "Designing a Fertilizing Robot Application Considering Energy Efficiency". *Rigas Tehniskas Universitates Zinatniskie Raksti* 28.1, 258–268. DOI: 10.2478/rtuect-2024-0021.

OpenStreetMap contributors (2025). *OpenStreetMap*. https://www.openstreetmap.org. Accessed: 2025-03-06.

Papadopoulos, G., S. Arduini, H. Uyar, V. Psiroukis, A. Kasimati, and S. Fountas (2024). "Economic and environmental benefits of digital agricultural technologies in crop production: A review". *Smart Agricultural Technology*, 100441. DOI: 10.1016/j.atech.2024.100441.

Paraforos, D. S., R. Hübner, and H. W. Griepentrog (2018). "Automatic determination of headland turning from auto-steering position data for minimising the infield non-working time". *Computers and electronics in agriculture* 152, 393–400. DOI: 10.1016/j.compag.2018.07.035.

Patel, S. and I. Mani (2011). "Effect of multiple passes of tractor with varying normal load on subsoil compaction". *Journal of Terramechanics* 48.4, 277–284. DOI: 10.1016/j.jterra.2011.06.002.

Pawase, P. P., S. M. Nalawade, G. B. Bhanage, A. A. Walunj, P. B. Kadam, A. G. Durgude, and M. R. Patil (2023). "Variable rate fertilizer application technology for nutrient management: A review". *International Journal of Agricultural and Biological Engineering* 16.4, 11–19. DOI: 10.25165/j.ijabe.20231604.7671.

Perron, L. (2011). "Operations research and constraint programming at google". In: *Principles and Practice of Constraint Programming–CP 2011: 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings 17*. Springer, 2–2. DOI: 10.1007/978-3-642-23786-7_2.

Pita, G. (2024). Private communication.

Plessen, M. (2018). "Partial field coverage based on two path planning patterns". *Biosystems engineering* 171, 16–29. DOI: 10.1016/j.biosystemseng.2018.04.010.

– (2025). "Smoothing of headland path edges and headland-to-mainfield lane transitions based on a spatial domain transformation and linear programming". *Biosystems Engineering* 257, 104229. DOI: 10.1016/j.biosystemseng.2025.104229.

Pour Arab, D. and C. Essert (2024). *Agricultural Fields 2D and 3D Models Dataset*. DOI: 10.5281/zenodo.10949632.

Pour Arab, D., M. Spisser, and C. Essert (2022). "Complete coverage path planning for wheeled agricultural robots". *Journal of Field Robotics*. DOI: 10.1002/rob.22187.

Ptacek, M., F. Frick, H. Pahl, C. Stetter, S. Wimmer, and J. Sauer (2024). "'ShapeCost-TUM': A calculation tool for field geometry dependent cultivation and transport costs". *Computers and Electronics in Agriculture* 225, 109254. DOI: 10.1016/j.compag.2024.109254.

Pulido-Moncada, M., L. J. Munkholm, and P. Schjønning (2019). "Wheel load, repeated wheeling, and traction effects on subsoil compaction in northern Europe". *Soil and Tillage Research* 186, 300–309. DOI: 10.1016/j.still.2018.11.005.

Pulido-Moncada, M., S. O. Petersen, and L. J. Munkholm (2022). "Soil compaction raises nitrous oxide emissions in managed agroecosystems. A review: Soil compaction raises nitrous oxide emissions in managed agroecosystems. A review". *Agronomy for Sustainable Development* 42.3, 38. DOI: 10.1007/s13593-022-00773-9.

Rahman, M. M., K. Ishii, and N. Noguchi (2019). "Optimum harvesting area of convex and concave polygon field for path planning of robot combine harvester". *Intelligent service robotics* 12.2, 167–179. DOI: 10.1007/s11370-018-00273-4.

Reeds, J. and L. Shepp (1990). "Optimal paths for a car that goes both forwards and backwards". *Pacific journal of mathematics* 145.2, 367–393. DOI: 10.2140/pjm.1990.145.367.

Roberts-Elliott, L., G. P. Das, and A. G. Millard (2022). "Agent-Based Simulation of Multi-robot Soil Compaction Mapping". In: *Annual Conference Towards Autonomous Robotic Systems*. Springer, 251–265. DOI: 10.1007/978-3-031-15908-4_20.

Rotz, S., E. Gravely, I. Mosby, E. Duncan, E. Finnis, M. Horgan, J. LeBlanc, R. Martin, H. T. Neufeld, A. Nixon, et al. (2019). "Automated pastures and the digital divide: How agricultural technologies are shaping labour and rural communities". *Journal of Rural Studies* 68, 112–122. DOI: 10.1016/j.jrurstud.2019.01.023.

Sabelhaus, D., F. Röben, L. P. M. zu Helligen, and P. S. Lammers (2013). "Using continuous-curvature paths to generate feasible headland turn manoeuvres". *Biosystems engineering* 116.4, 399–409. DOI: 10.1016/j.biosystemseng.2013.08.012.

Sahoo, P. K., D. K. Kushwaha, Y. NrusinghCharanPradhan, M. Kumar, M. Mahendra-Jatoliya, and I. Mani (2022). "Robotics application in agriculture". In: *55 Annual Convention of Indian Society of Agricultural Engineers and International Symposium*, 60–76.

Samuels, D. J. and H. Thomson (2021). "Lord, peasant... and tractor? Agricultural mechanization, Moore's thesis, and the emergence of democracy". *Perspectives on Politics* 19.3, 739–753. DOI: 10.1017/S1537592720002303.

Santos, L., N. Ferraz, F. N. dos Santos, J. Mendes, R. Morais, P. Costa, and R. Reis (2018). "Path planning aware of soil compaction for steep slope vineyards". In: *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 250–255. DOI: 10.1109/ICARSC.2018.8374191.

Schneider, M., A. Chan, and M. Körner (2023). *EuroCrops*. Zenodo. DOI: 10.5281/zenodo.10118572.

Scholten, C., A. Kamphuis, K. Vredevoogd, K. Lee-Strydhorst, J. Atma, C. Shea, O. Lamberg, and D. Proppe (2019). "Real-time thermal imagery from an unmanned aerial vehicle can locate ground nests of a grassland songbird at rates similar to traditional methods". *Biological Conservation* 233, 241–246. DOI: 10.1016/j.biocon.2019.03.001.

Shah, A. N., M. Tanveer, B. Shahzad, G. Yang, S. Fahad, S. Ali, M. A. Bukhari, S. A. Tung, A. Hafeez, and B. Souliyanonh (2017). "Soil compaction effects on soil health and cropproductivity: an overview". *Environmental Science and Pollution Research* 24.11, 10056–10067. DOI: 10.1007/s11356-017-8421-y.

Shaheb, M. R., R. Venkatesh, and S. A. Shearer (2021). "A review on the effect of soil compaction and its management for sustainable crop production". *Journal of Biosystems Engineering*, 1–23. DOI: 10.1007/s42853-021-00117-7.

Sitek, D. and R. Bertelmann (2014). "Open access: A state of the art". *Opening science: The evolving guide on how the Internet is changing research, collaboration and scholarly publishing*, 139–153. DOI: 10.1007/978-3-319-00026-8_9.

Slaughter, D. C., D. Giles, and D. Downey (2008). "Autonomous robotic weed control systems: A review". *Computers and electronics in agriculture* 61.1, 63–78. DOI: 10.1016/j.compag.2007.05.008.

Smith, P., R. M. Poch, D. A. Lobb, R. Bhattacharyya, G. Alloush, G. D. Eudoxie, L. H. Anjos, M. Castellano, G. M. Ndzana, C. Chenu, et al. (2024). "Status of the World's Soils". *Annual Review of Environment and Resources* 49. DOI: 10.1146/annurev-environ-030323-075629.

Söhne, W. (1953). "Druckverteilung im boden und bodenverformung unter schlepperreifen". *Grundlagen der Landtechnik-Konstrukteurhefte* 5.

Soitinaho, R., V. Väyrynen, and T. Oksanen (2024). "Heuristic cooperative coverage path planning for multiple autonomous agricultural field machines performing sequentially dependent tasks of different working widths and turn characteristics". *Biosystems Engineering* 242, 16–28. DOI: 10.1016/j.biosystemseng.2024.04.007.

Song, C. (2024). "Scenic spot path planning and journey customization based on multilayer hybrid hypernetwork optimization". *PloS one* 19.12, e0308135. DOI: 10.1371/journal.pone.0308135.

Spekken, M. and S. de Bruin (2013). "Optimized routing on agricultural fields by minimizing maneuvering and servicing time". *Precision agriculture* 14, 224–244. DOI: 10.1007/s11119-012-9290-5.

Spekken, M., S. de Bruin, J. P. Molin, and G. Sparovek (2016). "Planning machine paths and row crop patterns on steep surfaces to minimize soil erosion". *Computers and Electronics in Agriculture* 124, 194–210. DOI: 10.1016/j.compag.2016.03.013.

Spekken, M., J. P. Molin, and T. L. Romanelli (2015). "Cost of boundary manoeuvres in sugarcane production". *Biosystems engineering* 129, 112–126. DOI: 10.1016/j.biosystemseng.2014.09.007.

Stanford Artificial Intelligence Laboratory et al. (2018). *Robotic Operating System*. https://www.ros.org. Version ROS Melodic Morenia.

Stelter, A. (n.d.). *Ipiano/coverage-planning GitHub: QuickOpp Implementation*. https://github.com/Ipiano/coverage-planning. Accessed: January 9, 2026.

Tamirat, T. W., S. M. Pedersen, R. J. Farquharson, S. de Bruin, P. D. Forristal, C. G. Sørensen, D. Nuyttens, H. H. Pedersen, and M. N. Thomsen (2022). "Controlled traffic farming and field traffic management: Perceptions of farmers groups from Northern and Western European countries". *Soil and Tillage Research* 217, 105288. DOI: 10.1016/j.still.2021.105288.

Trendafilov, K., G. Tihanov, V. Stoykova, and G. Shivacheva (2023). "Algorithm for optimizing the movement of a mounted machine-tractor unit in the headland of an irregularly shaped field". *INMATEH-Agricultural Engineering* 70.2. DOI: 10.35633/inmateh-70-31.

Utamima, A. and A. Djunaidy (2022). "Agricultural routing planning: A narrative review of literature". *Procedia Computer Science* 197, 693–700. DOI: 10.1016/j.procs.2021.12.190.

Vahdanjoo, M., R. Gislum, and C. A. G. Sørensen (2023). "Operational, economic, and environmental assessment of an agricultural robot in seeding and weeding operations". *AgriEngineering* 5.1, 299–324. DOI: 10.3390/agriengineering5010020.

Van Orsouw, T. L., V. L. Mulder, J. M. Schoorl, G. J. Van Os, E. A. Van Essen, K. H. Pepers, and G. B. Heuvelink (2022). "Practical Implications of the Availability of Multiple Measurements to Classify Agricultural Soil Compaction: A Case-Study in The Netherlands". *Agronomy* 12.7, 1669. DOI: 10.3390/agronomy12071669.

Vasquez, J. I. (n.d.). *Irvingvasquez/ocpp GitHub*. https://github.com/irvingvasquez/ocpp. Accessed: January 9, 2026.

Vélez, S., G. Mier, M. Ariza-Sentís, and J. Valente (2024). "Integrated framework for multipurpose UAV Path Planning in hedgerow systems considering the biophysical environment". *Crop Protection*, 106992. DOI: 10.1016/j.cropro.2024.106992.

Versleijen, J. and S. de Bruin (2019). "Path planning on agricultural fields with obstacles". *Geo-Information Science and Remote Sensing Thesis Report GIRS-2019–2021*.

Wang, M., D. He, F. Shen, J. Huang, R. Zhang, W. Liu, M. Zhu, L. Zhou, L. Wang, and Q. Zhou (2019). "Effects of soil compaction on plant growth, nutrient absorption, and root respiration in soybean seedlings". *Environmental Science and Pollution Research* 26, 22835–22845. DOI: 10.1007/s11356-019-05606-z.

Wang, N., Z. Jin, T. Wang, J. Xiao, Z. Zhang, H. Wang, M. Zhang, and H. Li (2025). "Hybrid path planning methods for complete coverage in harvesting operation scenarios". *Computers and Electronics in Agriculture* 231, 109946. DOI: 10.1016/j.compag.2025.109946.

Wang, T., S. Wang, and T.-H. P. Chen (2023). "Study the correlation between the readme file of GitHub projects and their popularity". *Journal of Systems and Software* 205, 111806. DOI: 10.1016/j.jss.2023.111806.

Wei, W., T. Liu, S. Zhang, L. Shen, X. Wang, L. Li, Y. Zhu, and W. Zhang (2024). "Root spatial distribution and belowground competition in an apple/ryegrass agroforestry system". *Agricultural Systems* 215, 103869. DOI: 10.1016/j.agsy.2024.103869.

Wong, J. Y., P. Jayakumar, and J. Preston-Thomas (2019). "Evaluation of the computer simulation model NTVPM for assessing military tracked vehicle cross-country mobility". *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 233.5, 1194–1213.

Yan, S., G. Yan, et al. (2023). "Uncertainty aversion and farmers' innovative seed adoption: Evidence from a field experiment in rural China". *Journal of Integrative Agriculture* 22.6, 1928–1944. DOI: 10.1016/j.jia.2023.04.004.

Yasutomi, F., M. Yamada, and K. Tsukamoto (1988). "Cleaning robot control". In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1839–1841. DOI: 10.1109/ROBOT.1988.12333.

Zhai, Z., J. F. Martínez, V. Beltran, and N. L. Martínez (2020). "Decision support systems for agriculture 4.0: Survey and challenges". *Computers and Electronics in Agriculture* 170, 105256. DOI: 10.1016/j.compag.2020.105256.

Zhang, W. (2024). "Identification of wheel track in the wheat field". *Scientific Reports* 14.1, 900. DOI: 10.1038/s41598-024-51601-x.

Zhao, L., Y. Bai, and J. K. Paik (2024). "Optimal coverage path planning for USV-assisted coastal bathymetric survey: Models, solutions, and lake trials". *Ocean Engineering* 296, 116921. DOI: 10.1016/j.oceaneng.2024.116921.

Zhou, K., A. L. Jensen, D. D. Bochtis, and C. G. Sørensen (2015). "Quantifying the benefits of alternative fieldwork patterns in a potato cultivation system". *Computers and Electronics in Agriculture* 119, 228–240. DOI: 10.1016/j.compag.2015.10.012.

Acknowledgements

Supervisors

I want to start by thanking my supervisors, who selected me for the PhD position without knowing me and gave me a new chance to follow the academic path. Sytze, I am deeply grateful to have learned from you. During my PhD, I had the privilege of knowing you both professionally and personally. Every revision you made, often leaving my documents thoroughly marked, revealed the effort you put into teaching and improving the scientific quality of your students' work. Once I defined your teaching style as a river. Now, I would like to define your personality with two (apparently opposite) qualities: openness and closeness. Openness, because from the beginning you were transparent and willing to discuss any topic or ideas. Closeness, because your guidance always came from the heart, giving advice with care and a sense of family.

Joao, I also want to thank you for your guidance and for the mini-projects we carried out together. You brought creativity and motivation even when spirits were low. I will always remember programming the RefiBot in the afternoons, our drone class for kids in Utrecht, and our trip to Italy, where we repeatedly fell into tourist traps while seeking underground sites. The PhD would not have been the same without you.

Collaborators

I am grateful to AgXeed, which partially funded this PhD, and to the wonderful people I had the chance to work with there. Lars, your willingness to explore new ideas and constant support made a deep impression on me. I remember our first meetings where you spoke of improving farmers' lives through automation, and your motivation helped shape the ethical part of this work toward open science. I was fortunate to collaborate with talented developers Johan, and later Charlotte and Jeroen, and I thank them for their technical support during this phase of my life.

Coworkers

Many coworkers from the GRS and INF group have contributed to this journey. To all PhD students, postdocs, and professors, I thank you for always being available for coffee, advice, and conversation. Special thanks to Truus, Antoinette, and Gerlinde, whose endless support makes the GRS what it is. As a line from the Spanish film *Amanece, que no es poco* says: "Todos somos contingentes, pero tú eres necesario." (We are all contingent, but you are necessary.)

From the GRS, I want to highlight a few people. Deborah, I admire your spirit and resilience, as you keep moving forward even under difficult circumstances. Ximena, my paranymph, your support in the final phase of this PhD meant the world. Our conversations offered comfort and understanding, and I hope I returned the same to you. Paulina, we shared fewer university hours but bonded through aerial sports. Please, continue to challenge authority wherever you go. Sebastian, sharing language, room, and my first promotor with you made the early PhD struggles lighter. Robert, I thank you for countless lunches at Eastern Express, advice, and gossip. Tianyi, "Good morning".

Though my time in the INF group was brief, I had the chance to work closely with remarkable people such as Mar, Sergio, Jurrian, and Christos. Mar, I loved ice-skating with you while discussing collaborations. Sergio, the big blue giant, transformed a paper project into weekly meetings full of imagination. Thank you for inviting me to your DnD group.

Friends

I am grateful to those who helped make Wageningen feel like home. Laura, who introduced me to aerial hoop and became my first Dutch friend. Bhaggawent, who proves that dreams are worth following. Josephine from De Keuken, who always offered soup and a impressive story with a life lesson. The Trang and Tri siblings from Mama Bowls, who fed me curry katsu every week. Lisa from Pole Jungle, who kept me fit in the aerial playground. Many others welcomed me warmly, including Luci, Marta, Anastasia, Maria, and Aureo. Special thanks to Marina, my paranymph, who has been there since my first day. From walks along the dike to library days and our few night parties, your presence helped me overcome lonely and low moments in Wageningen.

I also want to acknowledge friends who remained close despite distance. Julio, your debates on humanity and our discussions about science have made me a better engineer and person. Barbara, your kindness matches the length of our conversations. Ana, whom I randomly share a similar academic path with, thank you for helping me finish the last paper of this thesis. Raquel and Sèlia, who have followed my path and made life feel connected despite distance. Peluchito, who showed how to face the crazy side of life with a smile. Andrea,

who inspired me to explore the world relentlessly. Other people who contributed to this thesis with their support were Paula, Eva, Pilar, Kike, Pache, Carmen and Gerardo.

Family

Quiero agradecer a mis padres. A mi madre, que me enseño desde el cariño mas extremo a como estar sentado siempre al borde de la silla, listo para hacer siempre lo que haga falta. A mi padre, por el interes que siempre puso en mejorar mi futuro y a mi como persona. A mi hermano y a Diana, por apoyarme siempre desde la distancia para lo que hiciese falta. A Lucas y Julia, por todo el amor que dan al mundo desde tan chicos. A mis abuelos Trini y Cristobal, que en paz descasen, por quererme incondicionalmente. Sin vosotros y vuestros valores no hubiese llegado tan lejos.

Marta, thank you for being a true companion through these years. Your energy filled our home and pushed me forward when work drained me. I treasure our daily routines, salsa nights, and trips around the world. I also thank Paco, Maria José, Judit, Tomás, Bruno, Lauri, and Iris for welcoming me into your lives. I want to remember Marta's grandma, María, who never missed a chance to tell me to shave my beard, and whose voice I still hear with a smile.

About the author

Gonzalo Mier was born on October 31st, 1994, in Chiclana de la Frontera, Cádiz, Spain. He grew up close to the sea, where he spent much of his time playing hockey and tinkering with small builds. A friend's father once crafted a wooden vending machine, a moment that sparked Gonzalo's drive to create things that worked. Later, in school, he helped build a small smart-home model, which strengthened his wish to work with machines.

He studied Electronic Engineering, Robotics and Mechatronics at the University of Seville. He enjoyed hands-on classes in robotics and electronics, though he leaned toward computer science, wary of sparks, crashes, and other hardware mishaps. He graduated in 2016.

Drawn by the chance to live in a larger city and deepen his skills, he moved to Madrid and earned a Master's in Artificial Intelligence at the Polytechnic University of Madrid. His thesis joined reinforcement learning with robotics to control an underactuated acrobot, feeding his interest in the point where AI and mechanics meet.

After his master's degree, he returned to Seville to join the robotics research group at Pablo de Olavide University. He worked on an autonomous robot that mapped the sewers of Barcelona and designed a social path planner for a museum guide robot. Both projects shared a clear aim: improve people's daily lives. During this time, he spent three months at Honda Research Institute Japan, where he worked with Haru, a small social robot.

Seeking new ground, he moved to the Netherlands to start a PhD at Wageningen University. His work linked robotics with path planning to support automation in farming. He released the Fields2Cover library, which drew broad attention to coverage planning and helped push progress in autonomous tractors.

After completing his PhD, Gonzalo began a postdoctoral position focused on expanding coverage path planning to uncertain settings. Alongside his research, he teaches courses on data structures and machine learning. Outside work, he enjoys travel, board games, and aerial sports such as aerial hoop and pole dance.

Peer-reviewed Journal Publications

Mier, G., A. M. Casado Faulí, J. Valente, and S. de Bruin (2025a). "Fields2Benchmark: An open-source benchmark for coverage path planning methods in agriculture". *Smart Agricultural Technology*, 101156. DOI: 10.1016/j.atech.2025.101156.

Mier, G., R. Fennema, J. Valente, and S. de Bruin (2025b). "Continuous Curvature Path Planning for Headland Coverage With Agricultural Robots". *Journal of Field Robotics* 42.3, 641–656. DOI: 10.1002/rob.22489.

Mier, G., J. Valente, and S. de Bruin (2023a). "Optimizing agricultural coverage path to minimize soil compaction". *Precision agriculture'23*, 467–472. DOI: 10.3920/978-90-8686-947-3_58.

Mier, G., J. Valente, and S. de Bruin (2023b). "Fields2Cover: An open-source coverage path planning library for unmanned agricultural vehicles". *IEEE Robotics and Automation Letters* 8.4, 2166–2172. DOI: 10.1109/LRA.2023.3248439.

Mier, G., S. Vélez, J. Valente, and S. de Bruin (2025c). "Soil2Cover: Coverage path planning minimizing soil compaction for sustainable agriculture". *Precision Agriculture* 26.4, 1–21. DOI: 10.1007/s11119-025-10250-4.

Pantos, C., J. Doornbos, **G. Mier**, and J. Valente (2023). "The ReFiBot makers guide: Fostering academic open science and circularity with a robotic educational kit". *HardwareX* 16, e00484. DOI: 10.1016/j.ohx.2023.e00484.

Vélez, S., **G. Mier**, M. Ariza-Sentís, and J. Valente (2024). "Integrated framework for multipurpose UAV Path Planning in hedgerow systems considering the biophysical environment". *Crop Protection*, 106992. DOI: 10.1016/j.cropro.2024.106992.

Other Scientific Publications

Ariza-Sentís, M., **G. Mier**, S. Vélez, and J. Valente (2024a). "Comparative Analysis of Uavs, Ugvs and Tractors for Precision Spraying in Vineyards: Addressing Economic, Energy, and Sustainability Aspects with Battery Constraints". In: SSRN. DOI: 10.2139/ssrn.4813501.

Ariza-Sentís, M., S. Vélez, H. Baja, **G. Mier**, R. G. Valenti, and J. Valente (2024b). "Optimización avanzada de drones para la detección de racimos usando algoritmos basados en la inteligencia de enjambre". In: 326. Tierras: Agricultura, Interempresasmedia, 56–59.

Mier, G., S. Vélez, J. Pereira, and S. de Bruin (2024a). "Optimized Route Planning to minimize Soil Compaction in single cropping and intercropping". In: 2024 IEEE International Conference on Robotics and Automation (ICRA), ICRA ; Conference date: 13-05-2024 Through 17-05-2024. DOI: 10.5281/zenodo.11175505.

Mier, G., S. Vélez, J. Valente, and S. de Bruin (2024b). "Planificación optimizada de rutas para minimizar la compactación del suelo". In: 1086. Editorial Agrícola Española, 78–80.

PE&RC Training and Education Statement

With the training and education activities listed below the PhD candidate has complied with the requirements set by the Graduate School for Production Ecology and Resource Conservation (PE&RC) which comprises a minimum total of 30 ECTS (= 20 weeks of activities)

Review/project proposal

- Fields2Cover: Robust and efficient coverage paths for autonomous agricultural vehicles (6 ECTS)

In-depth / Topical / On-site Postgraduate Courses

- Geostatistics, PE&RC, 2023 (1.5 ECTS)
- Algorithms, Coursera, 2024 (2.1 ECTS)

Methodological / Statistical Postgraduate Courses

- Design of Experiments, WIAS/PE&RC, 2023 (0.8 ECTS)
- Uncertainty Analysis and Statistical Validation of Spatial Environmental Models, PE&RC, 2024 (1.5 ECTS)

Invited review of journal manuscripts

- Biosystems Engineering, Agricultural coverage path planning, 2025 (1 ECTS)

Competence, skills and career-oriented activities

- Supervising BSc and MSc thesis students, WUR, 2022 (0.64 ECTS)
- Reviewing a Scientific Manuscript, WUR, 2023 (0.1 ECTS)
- Entrepreneurship in and outside Science, WUR, 2023 (1.1 ECTS)

- Scientific Publishing, WUR, 2024 (0.3 ECTS)

Scientific Integrity/Ethics in science activities

- Scientific Integrity, WUR, 2021 (0.6 ECTS)
- Ethics in the Age of AI, Coursera, 2023 (0.93 ECTS)

PE&RC Retreat, PE&RC Day, and other PE&RC events

- PE&RC Day, 2020-2023 (1.2 ECTS)
- PE&RC PhD Weekend - first term, 2021 (0.9 ECTS)
- PE&RC PhD Weekend - midterm, 2022, 2023 (1.2 ECTS)

National/local scientific meetings, seminars, and discussion group

- Vision and Robotics Exchange group meeting, 2022 (0.1 ECTS)
- AgroFood Robotics group meeting, 2021-2023 (8.2 ECTS)

International symposia, workshops and conferences

- NCG Symposium, Wageningen, the Netherlands, 2022 (1.3 ECTS)
- IEEE/RSJ International Conference on Intelligent Robots and Systems, Detroit, EEUU, 2023 (3.5 ECTS)
- European Conference on Precision Agriculture, Bologna, Italy, 2023 (1.9 ECTS)
- International Conference on Robotics and Automation, Yokohama, Japan, 2024 (2.5 ECTS)

Societally relevant exposure

- Workshop: I Children's and Youth Science Fair; Instituto Cervantes of Utrecht and the CENL-SWNL, 2023 (0.9 ECTS)
- Article in "Interempresas" magazine, 2024 (1 ECTS)
- Article in "Agricultura", 2024 (1 ECTS)

Lecturing/Supervision of practicals/tutorials

- Programming in Python, 2021 (0.3 ECTS)
- Artificial Intelligence, 2022-2024 (5.1 ECTS)

BSc/MSc thesis supervision

- Research topic 1: Coverage Path Planning on Headlands for Autonomous Field Operations
- Research topic 2: Soil impact of headland turning operations in path-planning for an autonomous tracked vehicle
- Research topic 3: Conditional route planning for unmanned agricultural vehicles
- Research topic 4: Workflows for coverage path planning for complex agricultural fields

- Research topic 5: Coverage path planning for capacitated harvest vehicles for the inner field
- Research topic 6: Obstacle avoidance for autonomous agricultural vehicles

Authorship statement

Chapter 1. General introduction. I wrote the first draft of the general research question and its general scientific and social perspective. My promotor and co-promotor commented on that draft. I revised the text. I have used Goblin.tools and ChatGPT to provide language support, and to enhance the clarity of the text. I am fully responsible for the final content and interpretation of the research.

Chapter 2. Fields2Cover. I contributed to defining the research question, proposed the methodology and the experimental design, carried out the experiments, and did the data analysis. I made the figures. I wrote the first draft, and solved the comments of my co-authors. I am fully responsible for the final content and interpretation of the research.

Chapter 3. Fields2Benchmark. I proposed the research question, the methodology and the experimental design, and carried out the experiments. I did the data analysis with one of the co-authors. That co-author and I made the figures. That co-author and I wrote the first draft. I solved the comments of my co-authors. I have used Goblin.tools and ChatGPT to provide language support, and to enhance the clarity of the text. I am fully responsible for the final content and interpretation of the research.

Chapter 4. Headland coverage planner. I contributed to defining the research question, proposed the methodology and the experimental design, and carried out the experiments. I did the data analysis. I made the figures. I wrote the first draft. I solved the comments of my co-authors. I have used Goblin.tools and ChatGPT to provide language support, and to enhance the clarity of the text. I am fully responsible for the final content and interpretation of the research.

Chapter 5. Soil2Cover. I proposed the research question, the methodology and the experimental design, carried out the experiments, and did the data analysis. I made the figures. One of the co-authors and I wrote the first draft. I solved the comments of my co-authors. I have used Goblin.tools and ChatGPT to provide language support, and to enhance the clarity of the text. I am fully responsible for the final content and interpretation of the research.

Chapter 6. General discussion. I wrote the first draft of the text after discussing subjects to be included with my promotor and co-promotor. I revised the text once, after comments of my promotor and co-promotor. I have used Goblin.tools and ChatGPT to provide language support, and to enhance the clarity of the text. I am fully responsible for the final content and interpretation of the research.

This research is part of the project "Fields2Cover: Robust and efficient coverage paths for autonomous agricultural vehicles" (with project number ENPPS.LIFT.019.019 of the research programme Science PPP Fund for the top sectors which is (partly) financed by the Dutch Research Council (NWO)).

Financial support from Wageningen University for printing this thesis is gratefully acknowledged.

Cover design by Marta Fanega-Valencia.

Printed by ProefschriftMaken.

This thesis has been printed on FSC-certified paper.