

Geo-information Science and Remote Sensing

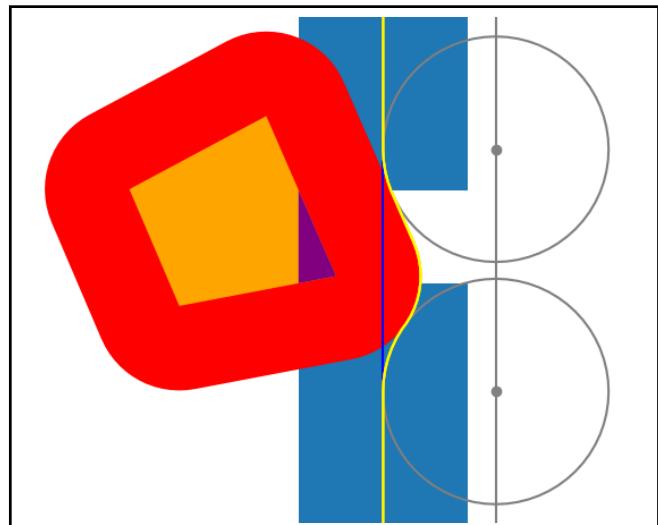
Thesis Report GIRS-2019-21

Path planning on agricultural fields with obstacles

An application of computational geometry algorithms

Jelmar Versleijen

3th July 2019



Path planning on agricultural fields with obstacles

An application of computational geometry algorithms

Jelmar Versleijen

Registration number 951221883040

Supervisor:

Dr. ir. Sytze de Bruin

A thesis submitted in partial fulfilment of the degree of Master of Science
at Wageningen University and Research Centre,
The Netherlands.

3th July 2019
Wageningen, The Netherlands

Thesis code number: GRS-80436
Thesis Report: GIRS-2018-21
Wageningen University and Research Centre
Laboratory of Geo-Information Science and Remote Sensing

Abstract

The objective of path planning in agriculture is to provide the optimal placement of paths in a field with a fixed width, acting as a traffic lane for agricultural machinery. However, calculating an optimal path plan is a complex task as geometric field properties and machinery characteristics have to be considered. Literature describes various cellular decomposition algorithms in which the agricultural field is broken down into simple subfields, thereby guaranteeing complete coverage. However, these methods do not consider in-field obstacle circumventing, overlap area or uncovered area and typically focus on number of turns. In this thesis, a combination of an extended trapezoidal spatial decomposition algorithm and in-field obstacle avoidance with smooth Dubins' paths is introduced for generating a path plan. The optimal path plan is defined as the path plan with the least cost according to a weighted sum objective function that includes three factors, i.e., overlap area, uncovered area and number of turns. A script was developed for generating path plans with various factor prioritizations and results were compared to path plans observed in literature and in practice. The developed solution provided a better solution in terms of minimizing overlap area and uncovered area and a minor improvement in minimizing number of turns on comparison fields without curved paths. In a follow-up study, curved path algorithms should be further investigated as they were observed to be effective in further reducing number of turns and possibly overlap area and uncovered area whilst simultaneously providing a flexible way of accounting for specific agricultural operation restrictions.

Keywords:

Agriculture, Computational geometry, Coverage planning, Optimization, Path planning, Robotics

Table of contents

Abstract	iv
Abbreviations	vi
Definitions.....	vii
Symbols.....	vii
1. Introduction	1
1.1 Context and background	1
1.2 Problem definition	2
1.2.1 Coverage path planning.....	2
1.2.2 Route planning.....	3
1.2.3 Obstacles	3
1.3 Research objectives and research questions.....	5
2. Methodology	6
2.1 Definition of an optimal path plan	7
2.1.1 Components and coverage of an agricultural field (requirement 1)	7
2.1.2 Definition of obstacles (requirement 3 and 4)	8
2.1.3 Movement throughout an agricultural field (requirement 5).....	9
2.1.4 Optimal path under the given restrictions (requirement 6)	10
2.1.5 Technical implementation	10
2.1.6 Testing	10
2.2 Path coverage generating algorithm.....	13
2.2.1 Coverage of a convex field.....	13
2.2.2 Spatial decomposition of a field using a sweep line.....	15
2.2.3 Merging subfields	17
2.2.4 Obstacle avoidance	18
2.2.5 Technical Implementation.....	23
2.2.6 Testing	25
2.3 Path plan comparison.....	26
2.3.1 General approach.....	26
2.3.2 Fields from literature.....	26
2.3.3 Field from practice.....	27
2.3.4 Comparison.....	28
3. Results	30
3.1 Definition of an optimal path plan	30

3.1.1	Field 1739	30
3.1.2	Field 5458	33
3.1.3	Field 1087	35
3.2	Path coverage generating algorithm.....	37
3.2.1	Performance	37
3.2.2	Field decomposition	37
3.2.3	Obstacle avoidance	40
3.3	Path plan comparison.....	43
3.3.1	Field from literature	43
3.3.2	Farmer's field.....	49
4.	Discussion	52
4.1	Definition of an optimal path plan	52
4.2	Path coverage generating algorithm.....	53
4.2.1	Field decomposition	53
4.2.2	Obstacle avoidance	55
4.3	Path plan comparison.....	56
5.	Conclusion and recommendations	58
	References	60
	Appendix A: Path plan examples.....	65

Abbreviations

BRP	Basisregistratie Gewaspercelen
CTF	Controlled Traffic Farming
DSS	Decision Support System
GAOS	Geo-spatial Arable Field Optimization Service
GEOS	Geometry Engine Open Source
ICT	Information and Communication Technology
MSW	Minimum Sum of Widths
OGC	Open Geospatial Consortium

Definitions

Coverage Path Planning	Envelops both route and path planning.
Edge	Line which makes up the exterior and interior.
Exterior	The outer boundary of a polygon.
Headland	Part of the field at an edge where most turning manoeuvres can be executed.
Interiors	Obstacle features (holes) in a polygon.
Path planning	Placement of tracks within an agricultural field.
Swath	The area that is worked continuously by an agricultural field during operation.
Spatial decomposition	Breaking down of a field geometry in multiple, smaller fields whilst maintaining the original exterior shape.
Route planning	The visiting sequence of tracks.
Track / traffic lane / (driving) path	Predefined paths in an agricultural field used by machinery during operations.
Turning radius	The distance describing the circle when the steering wheel of the vehicle is at its maximum steering angle, measured from the midpoint between the rear wheels.
Vertex	Point which connects edges.
Width	Minimum altitude of a convex polygon.

Symbols

\mathcal{D}	Convex decomposition of a non-convex polygon.
o	Overlap area of a field.
P	Path plan
\mathcal{R}_{aco}	Radius of the auxiliary circle obstacle.
\mathcal{R}_o	Radius of obstacle.
r	Minimum turning radius of an agricultural vehicle.
S	Dimensionless sum of overlap, uncovered and number of turns.
t	Number of turning manoeuvres.
u	Uncovered area of a field.
$\lambda_1 [1/m^2]$	Weight factor for overlap area.
$\lambda_2 [1/m^2]$	Weight factor for uncovered area.
λ_3	Weight factor for number of turns.
\mathcal{W}	Width of a convex polygon
w	Swath width

1. Introduction

1.1 Context and background

Rising food demands lead to increasing challenges for agricultural innovation (Gebbers et al., 2010; Iglesias et al., 2012; Tilman et al., 2011). These challenges are not limited to productivity, but include, amongst others, biodiversity (Haenke et al., 2009; Velten et al., 2018), ecological network connectivity (Foltête, 2018; von Haaren et al., 2006), efficient resource usage (Jiang et al., 2018; Lin et al., 2017; Scherer et al., 2018) and visual perception of the agricultural landscape (Assandri et al., 2018; Borin et al., 2010; Stilma et al., 2009). These challenges also involve complex decision-making problem in farm management where a balance has to be struck between environmental impact and economic gain. The increasing amount of data that is accumulated by modern Information and Communication Technology (ICT) can contribute in reducing uncertainty in these challenges and ultimately help ensuring global food security and closing the yield gap (Fountas et al., 2006), i.e., the difference between theoretical maximum sustainable yield and achieved yield (van Ittersum et al., 2013).

The process of closing the yield gap can be referred to as an optimisation problem. Application of precision agriculture can contribute to closing the yield gap. Precision agriculture can be described as data driven Decision Support Systems (DSS) in agriculture provide means of maximizing productivity, whilst minimizing environmental impact (Pierce et al., 1999). Amongst others, the spatial configuration of agricultural field tracks is a key element in such optimisation process. The variation in geometric field characteristics influences the difficulty of maximizing productivity of agricultural machinery (Bochtis et al., 2009). To comply with the growing demands within the agricultural sector, optimising operational efficiency is a must. Operational efficiency refers to the ratio between achieved productivity during a field operation and the theoretical maximum productivity (Witney, 1988). Traditional agricultural planning and management need to be revised in order to take advantage of the full potential of modern machinery in agriculture (Bochtis et al., 2014; Sørensen et al., 2010).

Controlled traffic farming (CTF) is an important advancement in agriculture to minimize soil degradation caused by heavy machinery during in-field operations whilst simultaneously maximizing productivity (Tullberg et al., 2018; Tullberg et al., 2007). CTF refers to the traffic management in agricultural fields which separates crop zones from traffic lanes (tracks) (Taylor, 1983). Using agricultural machine guidance systems, CTF has proven to be an effective strategy in reducing environmental impact, mainly by achieving higher yields and decreased soil compaction (Gasso et al., 2014). The application of (semi-)autonomous agricultural vehicles in CTF operations help reducing overlap in the coverage of fields, which contributes to limiting environmental impact of fertilizers and pesticides.

Various software packages have been developed to assist farmers in optimizing usage of their agricultural fields. However, development remains costly and therefore limiting widespread adaptation of such tools for precision agriculture. Analysis of geospatial data is essential in these systems. Progress made over the last 20 years in standardization of

geographic information by the Open Geospatial Consortium (OGC) provides a solid foundation for precision agriculture software development (Nash et al., 2009). The geo-spatial arable field optimization service (GAOS) provides an example of successfully adopting these standards in practice in a web-based environment for optimizing track placement within agricultural fields (de Bruin et al., 2014).

Coverage path planning has its roots in robotics and revolves around finding a path over all points of a given area whilst avoiding obstacles (Galceran et al., 2013; Yu, 2015). It has many applications in robotics, ranging from smaller household aid robots such as lawnmowers (Shiu et al., 2008) or vacuum cleaners (Joon Seop et al., 2004) to larger robots such as autonomous agricultural machinery (Oksanen et al., 2009). A CTF system incorporates both path and route planning, the former of which will be emphasized in this thesis.

1.2. Problem definition

The aim of this research is to contribute to the development of efficient and computationally effective algorithms for determining the optimal path plan of an agricultural field with obstacles.

1.2.1 Coverage path planning

One of the earliest literature on coverage path planning defined six requirements for a mobile robot in a coverage problem (Cao et al., 1988). Although the study revolved around a filling operation, the concept still remains applicable in agricultural operations. The requirements are as follows:

1. The robot must move through the entire area (complete coverage).
2. The robot must fill the entire region without an overlapping path.
3. The robot must traverse the entire area, both sequentially and continuously without repetition of paths.
4. All obstacles must be avoided within the area.
5. Only simple movements (e.g., circles or straight lines) should be used for easier controls.
6. Under the available conditions, an optimal path is desired.

Recent studies on coverage path planning in agriculture are generally focussed on optimizing field decomposition of complex field shapes (non-convex and/or with obstacles) (Bochtis et al., 2008). Methods such as the Minimum Sum of Width (MSW) decomposition (Yu, 2015) or “split and merge” (Oksanen, 2007) have been proposed to reduce a complex geometric field in multiple subfields with distinct track directions. These methods may provide a better solution compared to a single direction track for a complex geometric field in terms of complete coverage of a field with minimal travel distance for an agricultural vehicle. These studies concluded that by minimizing the number of turns, the travel distance and operation time can be minimized, resulting in maximal operational efficiency.

The objective of path planning in agriculture is to provide the optimal placement of tracks in a field with a fixed width, acting as a traffic lane for agricultural machinery (Jensen et al., 2018). Combining a coverage plan with software for autonomous vehicle navigation allows

for increased operational efficiency (Bochtis et al., 2008; Keicher et al., 2000). However, calculating an optimal path plan is a complex task as geometric field properties and machinery characteristics have to be considered. This problem bears resemblance to the "lawn mower" problem which revolves around finding the shortest path in a given region and a cutter with a given shape which needs to cover the entire region. This is proven to be NP-hard (Arkin et al., 2000). Some commercial floor cleaning robots use a random path approach which eliminates the need for expensive algorithms and sensors (Palacin et al., 2005). Given enough time, it will eventually cover the whole area. However, one can imagine that this approach is not feasible for agricultural operations. Path planning in this thesis is focussed on achieving the optimal condition for coverage in a 2D space. Projecting a 2D generated path on 3D terrain will lead to slight increases or decreases in path length depending on the elevation (Hameed et al., 2016).

1.2.2 Route planning

Whereas path planning refers to the placement of tracks, route planning describes the visiting sequence and direction of the predefined paths for an operation (Jensen et al., 2018). This is also referred to as the driving pattern. Depending on the type of operation, this may introduce additional complexity. Different operations involve vehicles with distinct machine characteristics, such as the minimum turning radius or loading capacity. For example, when executing an operation with a servicing unit, the available capacity should be accounted for when calculating an optimal route plan in order to avoid a forced stop of machinery halfway a track due to capacity limitations and thereby reducing operational efficiency (Spekken et al., 2013). The optimal routing plan can be calculated and optimised given a path plan and various machine characteristics and/or soil properties such as wetness. Because of limited resources, route planning is considered outside the scope of this thesis.

1.2.3 Obstacles

Coverage path planning would be greatly simplified if all fields would be rectangular. However, in the real world, boundaries are imposed by nature or man-made. Natural boundaries may include anything ranging from natural heritage sites, protected cultural landscape to rivers or rocks which can be very pronounced in countries such as Finland (Oksanen, 2007). Man-made boundaries may include infrastructure, urban areas, wind turbines, electricity pylon (figure 1) or less obvious borders such as municipal boundaries. An increasing demand on sustainable energy development urges for more wind turbine placement (Felder et al., 2011). However, placement is a controversial topic which is met by strong opposition by some communities (Righter, 1996). The ever increasing societal relevance on renewable energy sources may lead to placement on agricultural plots, as was demonstrated by "windpark Noordoostpolder" in Flevoland, the Netherlands ("Duurzame energiewinning," 2017). Rather than removing agricultural land, dual-purpose land usage for food and renewable energy production should be considered. A dual- or even multi-purpose agricultural land use may be inevitable as agricultural land is seen as one of the most suitable land types for placement of wind turbines (Bucksteeg, 2019).



Figure 1: examples of in-field obstacles on agricultural fields; left wind turbine (EPSG:4326: X: 5.595644, Y: 52.692736), right electricity pylon EPSG:4326: X: 5.647720, Y: 51.963021) (Google Earth, 2019).

This possible development affirms the need for obstacle avoidance methods in coverage path planning. It has been proven that circumventing smaller obstacles can increase operational efficiency (Liu et al., 2018; Zhou et al., 2014). Increased efficiency can be obtained if there is no need for headlands around obstacles. Headlands are required for agricultural machinery to allow for turning space to navigate from one track to the other. Various headland manoeuvres are possible when changing tracks depending on the machine characteristics for smooth connections from one track to the other (Sabelhaus et al., 2013). The respective width and length of a manoeuvre influences the operational efficiency (Parafos et al., 2018). By circumventing obstacles, the number of headlands can be reduced, thereby reducing the number of turns needed to cover a field. There is need for more research on combining spatial decomposition and merging methods with alternative methods for obstacle avoidance.

1.3 Research objectives and research questions

The objective of this research is to determine an optimal path plan including headland placement for agricultural machinery on a field with obstacles.

1. What defines an optimal path plan in fields with obstacles?
2. What algorithm can be used to find the optimal path plan on a field with obstacles?
3. How does the path plan generated by the developed algorithm compare to path plans observed in practice and in literature?

2. Methodology

This chapter covers methods for answering each research question and describes which steps will be performed in order to realise the overall objective. A schematic representation of these steps is presented in figure 2.

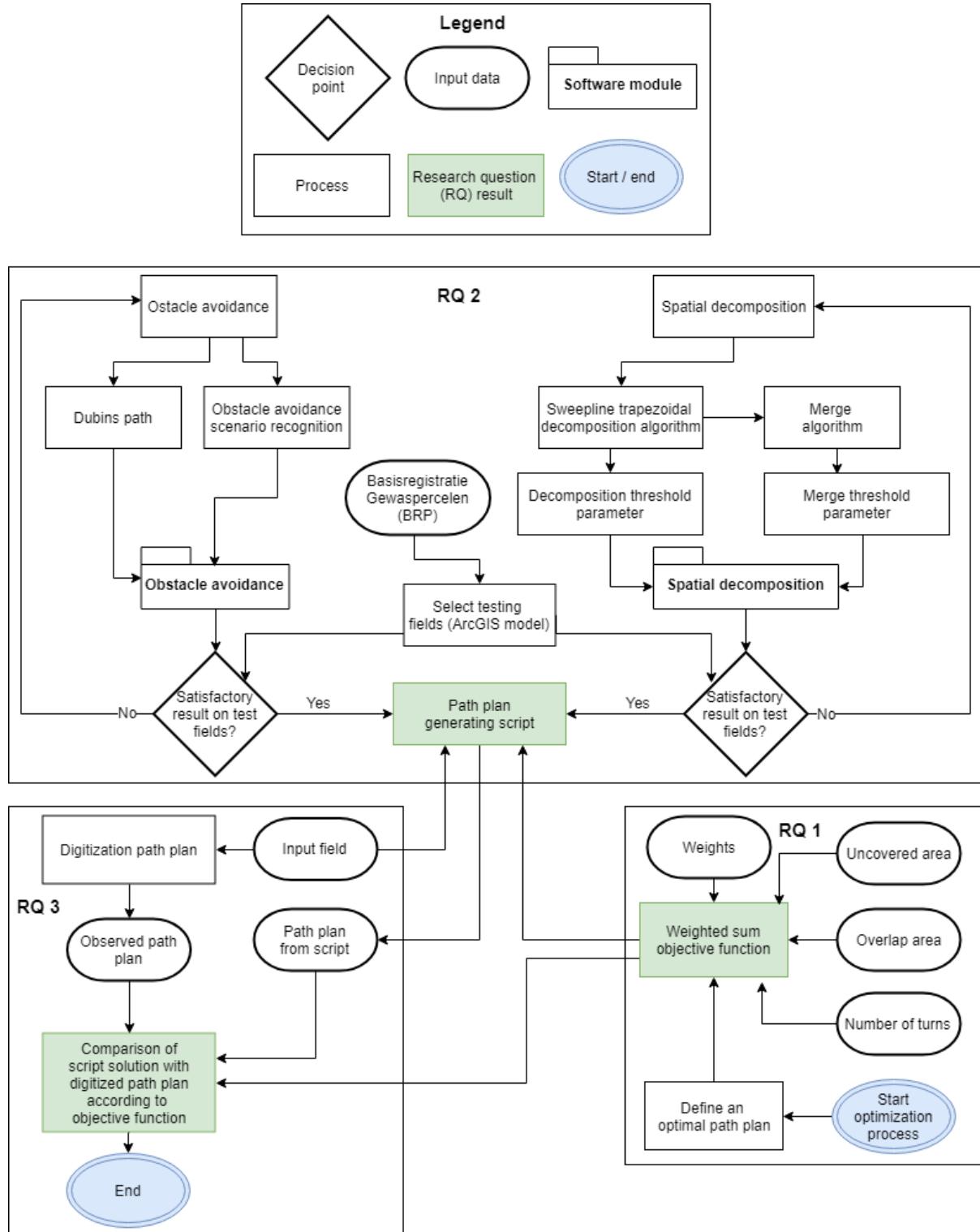


Figure 2: flowchart of methodology.

2.1 Definition of an optimal path plan

For a simple, rectangular field, placing tracks parallel to the longest edge will result in the minimal number of turns. When dealing with increasingly difficult field geometry in agricultural coverage problems, not all requirements can be satisfied and thus have to be prioritized (Oksanen, 2007). In this thesis, a distinction was made between route planning and path planning as stated in the problem definition. Therefore, only requirement 1, 3, 4, 5 and 6 from section 1.2.1 will be considered. In addition to the requirements, a constraint is that the vehicle is not allowed to leave the field geometry. The definition of these requirements is elaborated below.

2.1.1 Components and coverage of an agricultural field (requirement 1)

An agricultural field can be defined as a 2D space which may include holes (obstacles). A field is represented as a polygon. The polygon consists out of multiple, straight-line segments, also referred to as edges, which form a closed geometry. Each edge is connected to another edge. A connection between edges is called a vertex or corner. The outer edges of a field are referred to as the exterior polygon. In this thesis, the entire area within the polygon is targeted to be covered.

An area which is used for turning is referred to as a headland. Headlands are placed at the edge of a field. Lines which the agricultural machinery use are referred to as paths, which are usually placed between headlands. The swath width determines the space between paths and is depended on the agricultural operation. The paths should be placed in a way that the complete field can be covered (worked land). Figure 3 provides an illustration of the basic terminology.

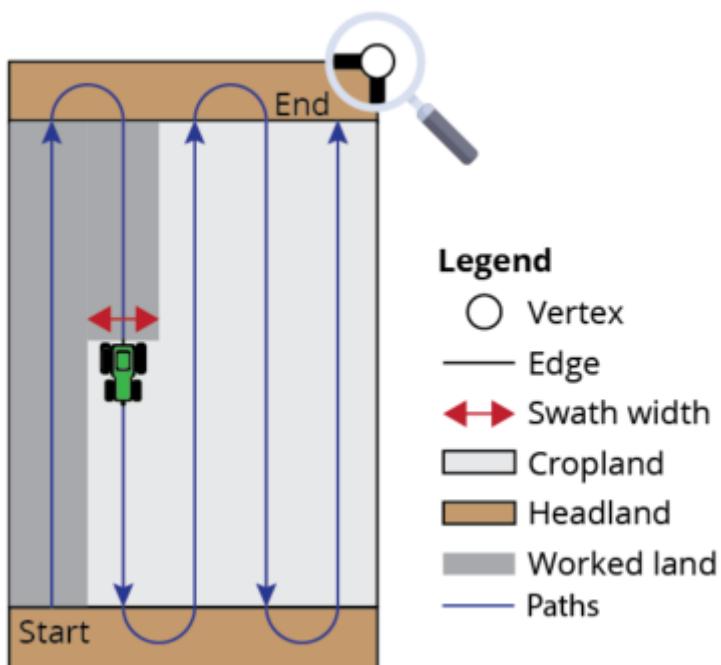


Figure 3: basic notation of agricultural field components.

Generally, headlands should be minimized as this results in the least number of turns and therefore increased operational efficiency (Oksanen, 2013). Figure 4 provides an example of this situation. Although solution figure 4(a) results in the minimum number of turns, solution figure 4(b) would ensure complete coverage. The uncovered area in figure 4(a) comes at a cost that depends on the crop type (de Bruin et al., 2009). Swath width is essential in determining the total area of uncovered land and the number of turns in a field. As stated at the beginning of this chapter, not all requirements can be simultaneously satisfied and therefore they are prioritized depending on user preference.

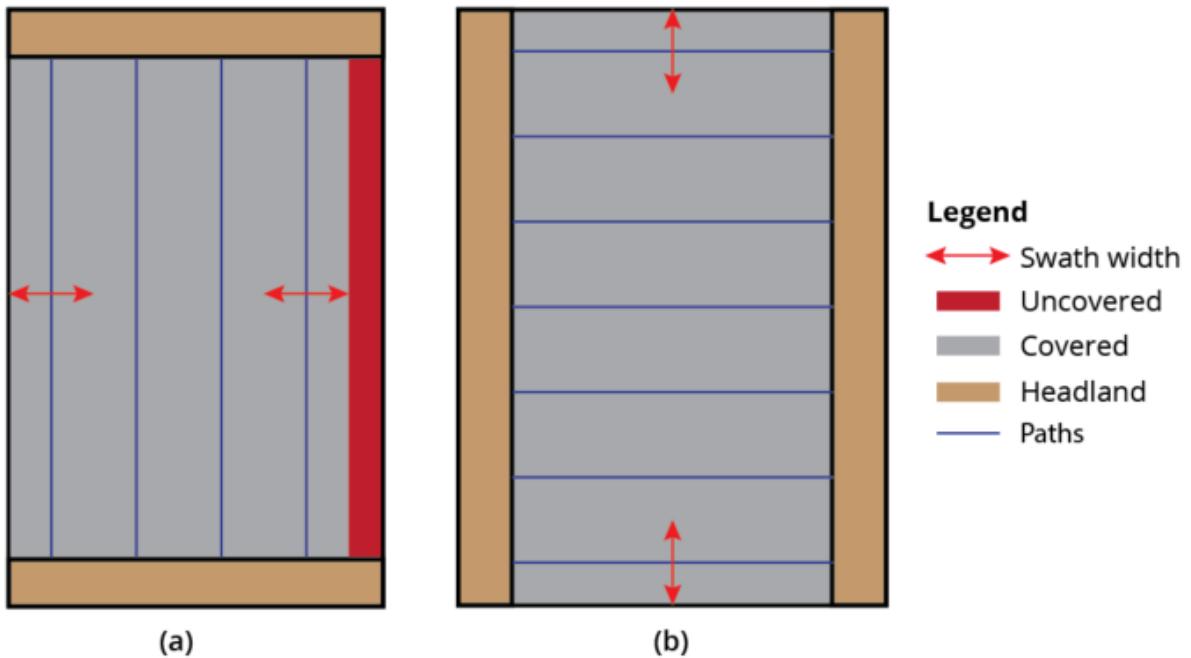


Figure 4: complete coverage in relation to number of turns depending on path direction.

2.1.2 Definition of obstacles (requirement 3 and 4)

An obstacle can be defined as an area within the field which must be avoided by the agricultural machinery. Stated more formally, an obstacle area is an interior polygon inside an agricultural field. Coverage paths cannot cross obstacle areas. Additionally, half a swath width minimum distance should be kept from an obstacle to ensure safety of the machinery.

Obstacle avoidance solutions are shown in figure 5. Whereas solution figure 5(a) has to execute six turn manoeuvres, solution figure 5(b) has only four by altering the path direction slightly. Additionally, solution figure 5(a) introduces uncovered land (red) by having to place additional headlands. A downside to this approach is that the altered path will overlap (yellow) an already covered area and introduce a small piece of uncovered land due to turning. When encountering larger obstacle areas, this strategy may not work and an additional headland has to be placed along the edge of the interior polygon. Depending on the prioritization of requirements, either solution may be optimal. When the number of turns and/or uncovered area must be minimized, solution figure 5(b) is optimal. However, when minimizing overlap, solution figure 5(a) remains favourable.

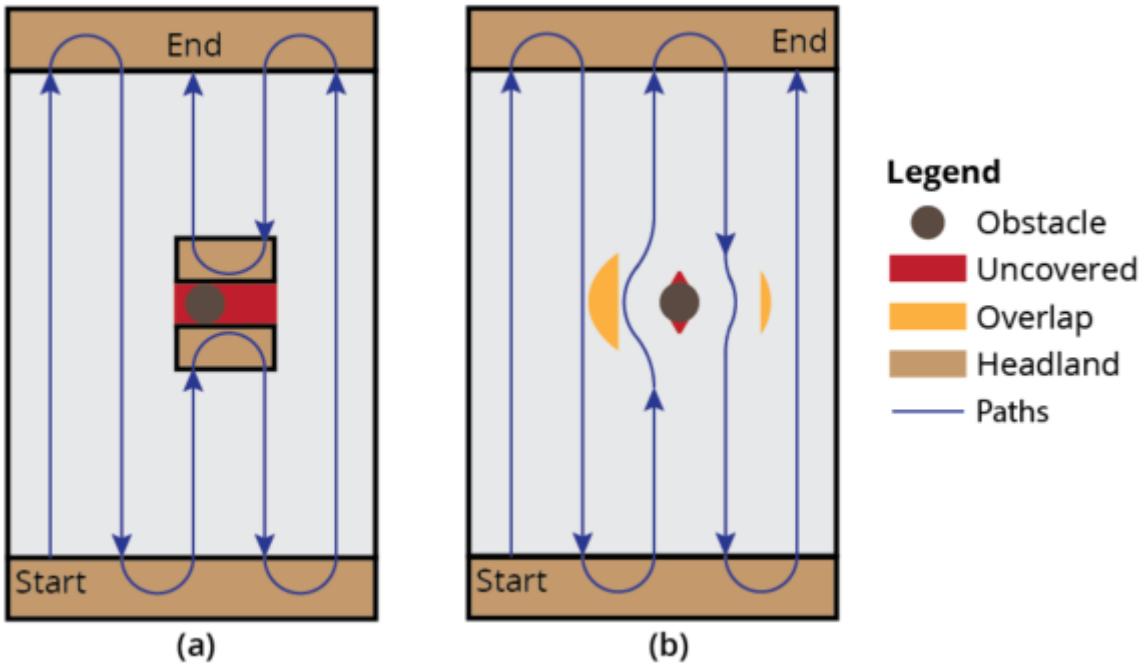


Figure 5: coverage path plan variety on a field with an obstacle.

2.1.3 Movement throughout an agricultural field (requirement 5)

A turn can be defined as the path between tracks. Turn manoeuvres are time consuming and can be defined as non-working distance. Circular movements are needed when turning. The minimum turning radius of a vehicle is used as input for the algorithm to determine the optimal path. Left and right are modelled as going around a circle. The circle represents the minimum turning radius. The method of modelling a vehicle with linear and circular movements will provide the shortest path between 2 points in 2D space and is thus provides the optimal path. Figure 6 illustrates this where the shortest path is found given a minimum turning radius with prescribed initial (a) and terminal (b) tangents assuming that the vehicle can only travel forward. This is also referred to as the Dubins path (Dubins, 1957). Using car analogy, a path can be described using 'left turn' (L), 'right turn' (R) or 'straight' (S). Considering this, the path in figure 6 can be denoted as RSR.

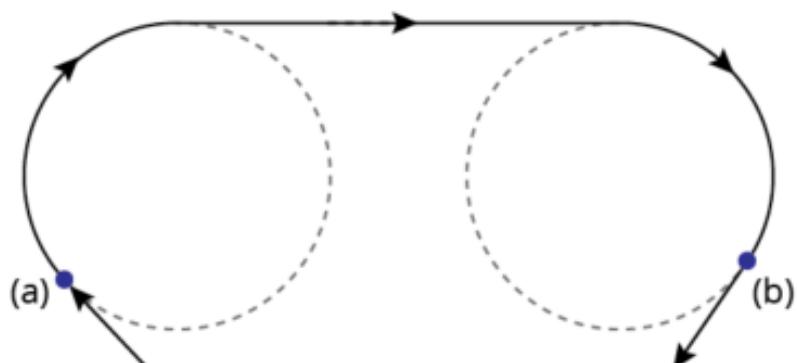


Figure 6: shortest path between point a and b using an RSR Dubins path.

2.1.4 Optimal path under the given restrictions (requirement 6)

As the optimal path plan depends on the prioritization of the user, a penalty-based objective function $S(P)$ was introduced, where S is the cost for path plan P which is to be minimized. The three factors, overlap $o[m^2]$, uncovered $u[m^2]$, number of turns t were summarized in a weighted sum objective function as shown in equation 1. Weights $\lambda_1[1/m^2]$, $\lambda_2[1/m^2]$ and λ_3 are assigned to represent prioritization. For each path plan, S will be calculated. Assigning a higher weight, or penalty, to a factor will prioritize it when generating a path plan for a given field.

$$\text{Minimize } S(P) = (o * \lambda_1) + (u * \lambda_2) + (t * \lambda_3) \quad (1)$$

Where $o [m^2]$ denotes overlap area in square meters, $u [m^2]$ denotes uncovered area in square meters and t denotes number of turns. The proposed solution will produce multiple path plans P for a given field. The dimensionless sum S of the products for the three weighted factors was minimized to result in the optimal path plan.

2.1.5 Technical implementation

The objective function from 2.1.4 was implemented in a Python environment. It is part of a larger script which is elaborated in 2.2.5. The input weights for each factor are provided by the user when running the script.

2.1.6 Testing

A workflow was designed in ArcGIS Pro model builder which preprocessed the Basisregistratie gewaspercelen (BRP), loosely translated as "Core Registration of Crop fields" from the Dutch ministry of economic affairs, before calculating a path plan. The general idea was to select interesting use cases by selecting fields which intersected power pylons and/or wind turbines. The selection was further narrowed down by dropping fields which were marked as (temporarily) grassland as these provide no clear path plan. Additionally, fields which contained a large number of power pylons or wind turbines and already had interior features were selected as the most interesting test cases.

The BRP did not always provide valid geometry in a sense that power pylons were not always erased from the field geometry or coordinates sequences were invalid. A workflow was designed in ArcGIS Pro which erased the power pylons from fields assuming all power pylons had a roughly equal base of 7×7 meters. The resulting set of fields was used to test the algorithm during development.

Three fields were selected for demonstrating the effects of changing the weight parameters. The input field was treated as if headlands were already removed. The testing fields were manually selected by visual assessment of the field geometry complexity. Three fields were selected which contained obstacles and were relatively geometrically complex compared to other fields in the BRP. Table 1 provides an overview of the fields selected.

Table 1: testing field description.

Name	Figure	Area m ²	Obstacle count	Centroid XY (RD_new)
Field 1739	7	78970	2	195518.136, 360392.574
Field 5458	8	44543	1	247401.575, 475119.554
Field 1087	9	50210	1	195846.908, 364275.225

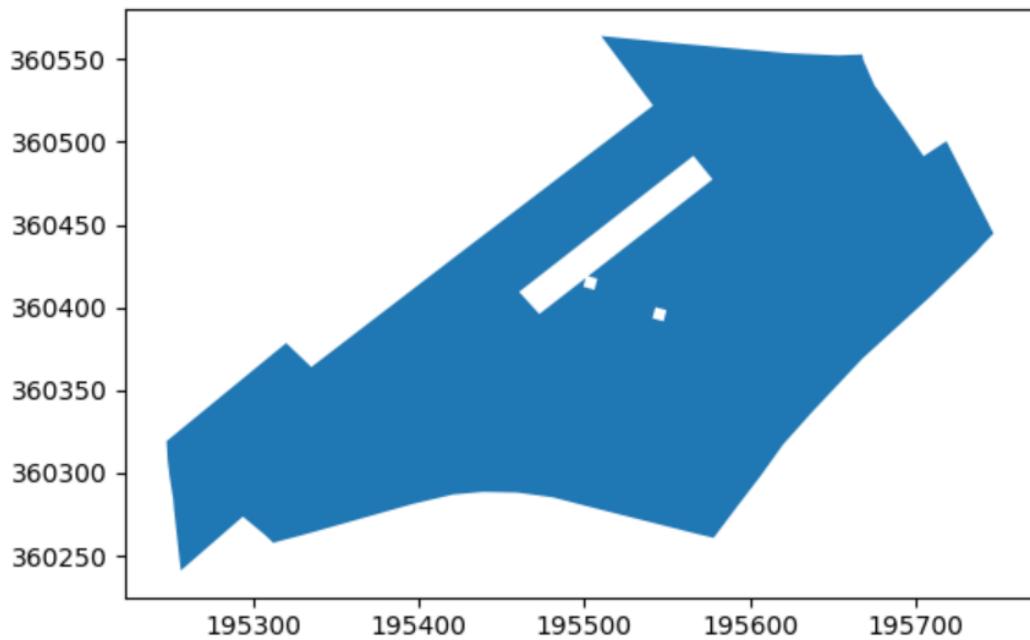


Figure 7: field 1739.

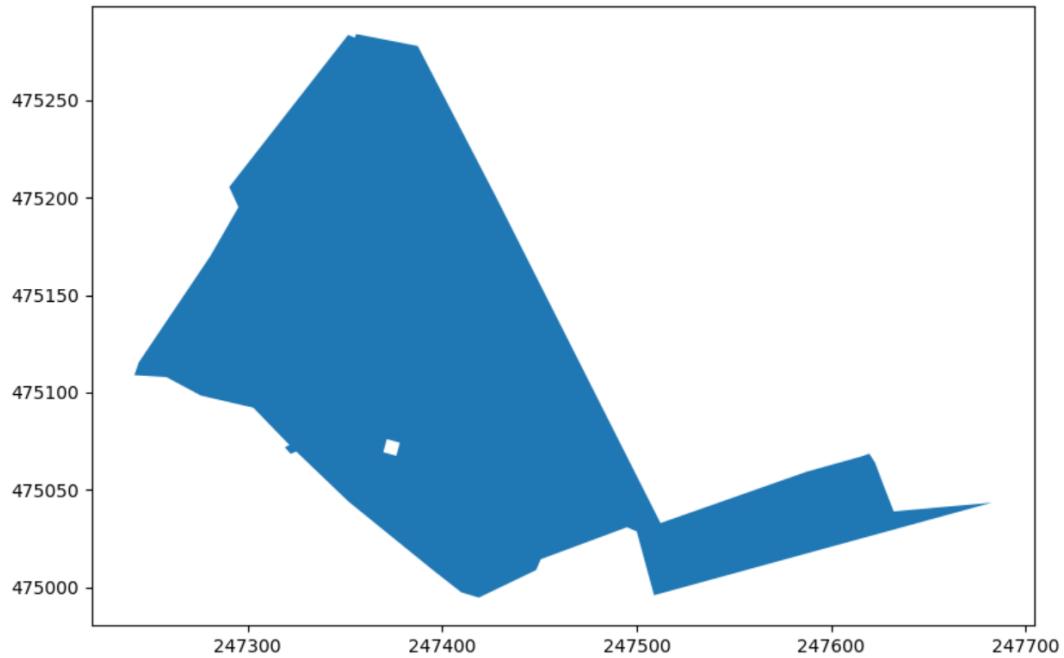


Figure 8: field 5458.

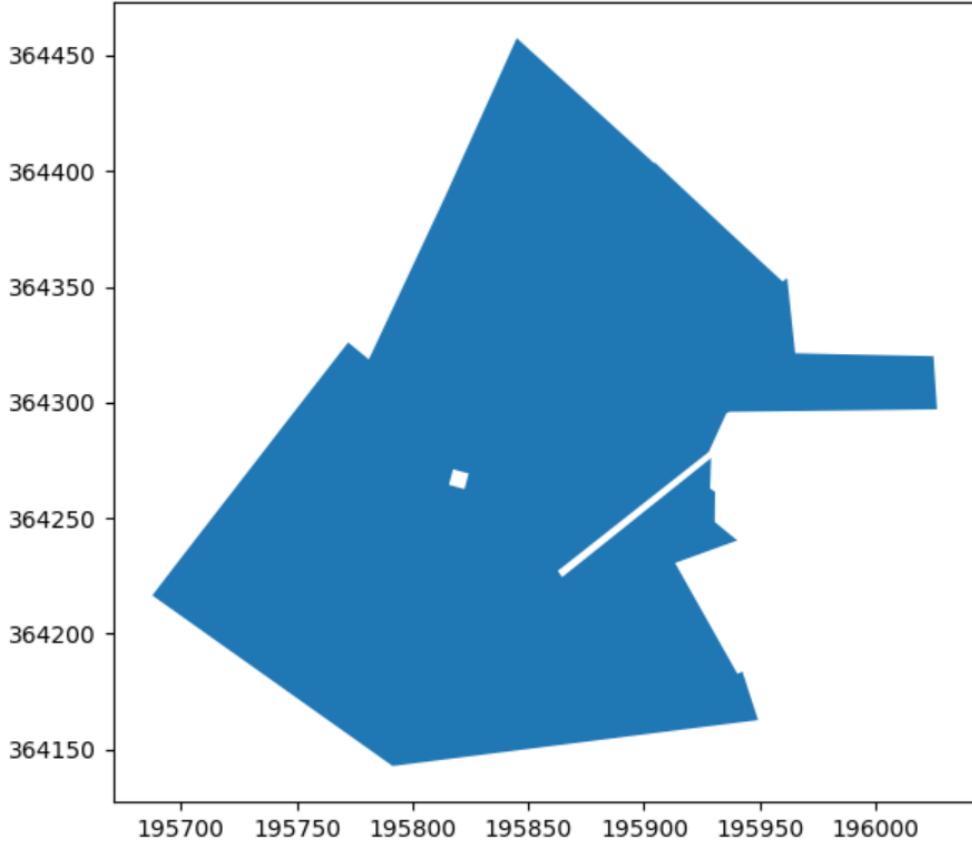


Figure 9: field 1087.

Three scenarios were designed to test the sensitivity of the objective function to changes made in the factor weights. An overview of the weight settings is provided in table 2. Each scenario revolves around minimizing one of the three factors. Each scenario and its respective weight settings were used to calculate a path plan for the selected test fields. A default swath width of 3 meters (usual for potato crops) and a turning radius of 6 meters (adopted from Oksanen (2007)) were used for each scenario.

Table 2: three testing scenarios with varying weight settings.

	Overlap (o)	Uncovered (u)	Number of turns (t)
Scenario 1	99999	1	1
Scenario 2	1	99999	1
Scenario 3	1	1	99999

2.2 Path coverage generating algorithm

Path coverage generation aims to find the optimal path plan of an agricultural field with obstacles, considering machine characteristics such as the maximum turning radius. Two methods were considered: Minimum Sum of Widths (MSW) (Yu, 2015) and 'Split and Merge' (Oksanen, 2007). Both methods adopt a similar approach, with a few minor differences. The general approach can be described as finding the optimal decomposition of a complex geometric field into multiple simple shaped subfields. Complex can be described as non-convex whereas simple refers to a convex shape. Both methods apply the divide and conquer algorithm design paradigm, albeit with different optimization approaches. The decomposition process consists out of multiple steps:

1. A sweep line is applied to the input field perpendicular to the edges of the input field (Yu, 2015) or in a range between 0-180 degrees and optimized using heuristics (Oksanen, 2007).
2. The sweep line is used for decomposition of the input field in multiple, subfields. This will result in multiple sets of subfields, depending on the number of sweep line directions performed.
3. For each of the resulting sets of subfields, subfields are merged based on a given rule set to reduce the number of subfields.
4. The sets of subfields are evaluated using the Minimum Sum of Widths (Yu, 2015) or a weighted sum (Oksanen, 2007) that quantifies 'optimal' in terms of total area of the subfield, distance covered by an agricultural vehicle in a subfield and efficiency expressed in time needed to cover a subfield are used to determine the optimal decomposition of a field.

In this thesis, an extension to the Minimum Sum of Widths method was used as the methodology could be implemented and extended with limited resources.

2.2.1 Coverage of a convex field.

Spatial decomposition of a field was executed in order to minimize the number of turns which have to be made. The complex input geometry is split in multiple, convex geometries which are referred to as subfields. For coverage of a convex subfield, the problem can be formulated as finding the width of the field and its corresponding direction. In context of this research, width refers to the shortest edge of the smallest bounding rectangle. Tracks are placed perpendicular to this edge. The number of turns required in a convex polygon scales with the width of the field. Figure 10 provides an illustration of the width of a polygon, determined using a smallest bounding rectangle.

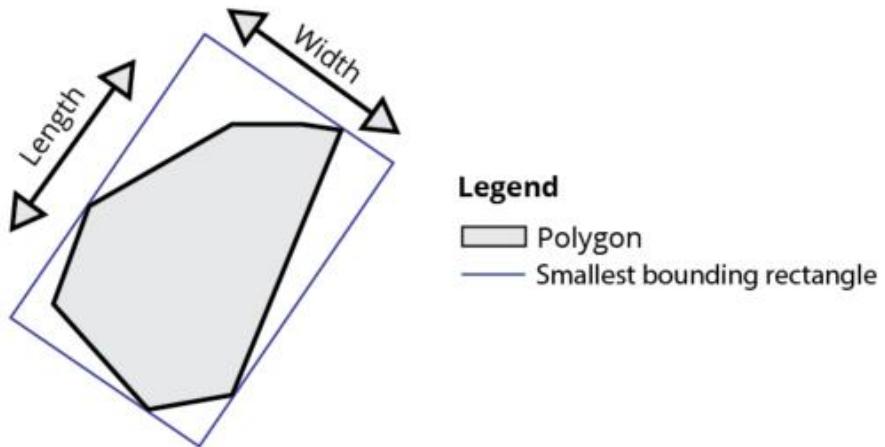


Figure 10: determining the width of a polygon using the smallest bounding rectangle.

Calculating the smallest bounding rectangle was achieved by using the rotating calliper algorithmic paradigm method shown in figure 11 (Toussaint, 1983). An antipodal pair is formed when one blade of the Vernier calliper is flat against the polygon edge with the opposite edge or point touching the other blade. A complete rotation of the polygon is done to determine the smallest bounding rectangle and the corresponding width. Tracks were placed perpendicular to the width of the resulting subfields, resulting in the least number of turns for a given subfield.

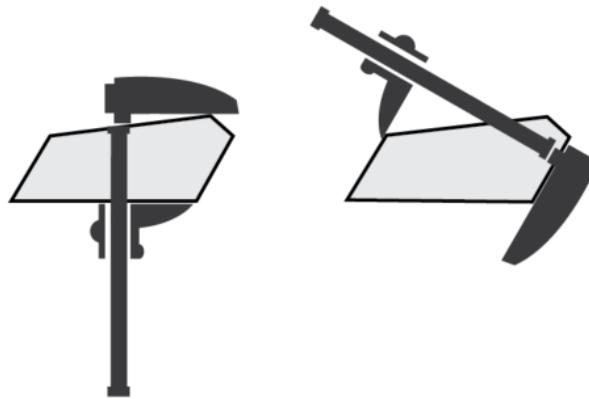


Figure 11: illustration of rotating calliper method to determine the width of a polygon.

Ideally, a path should be perpendicular to a headland (90°) as this ensures no uncovered or overlapping areas (Hameed et al., 2011). However, when working with non-rectangular fields this will not be the case. Depending on the angle, additional uncovered or overlapped areas will be introduced as shown in figure 12. Overlapped areas will result in excess use of resources such as pesticides or fertilizers. Uncovered areas will ensure no overlap but at a cost of leaving potential cultivated land untouched. Additionally, such areas may act as sources of weed invastation. Whether situation figure 12(a) is favourable over figure 12(b) is dependent on the user preferences and is covered in research question 1. Overlapped or uncovered areas can be calculated by intersection of the path with the headlands. Overlap and uncovered areas increase with headland angles increasingly differing from 90° . Swaths will seldom fit perfectly in a given (sub) field geometry as shown in situation b. In order to minimize this effect, paths were placed starting from the longest edge parallel to the defined driving direction until the paths go out of bounds or are too short. Too short is defined as

when the driving path is shorter than twice the swath width and will generally be not worth the effort of covering as introducing extra headland manoeuvres are expensive (Bochtis et al., 2019a; Bochtis et al., 2008).

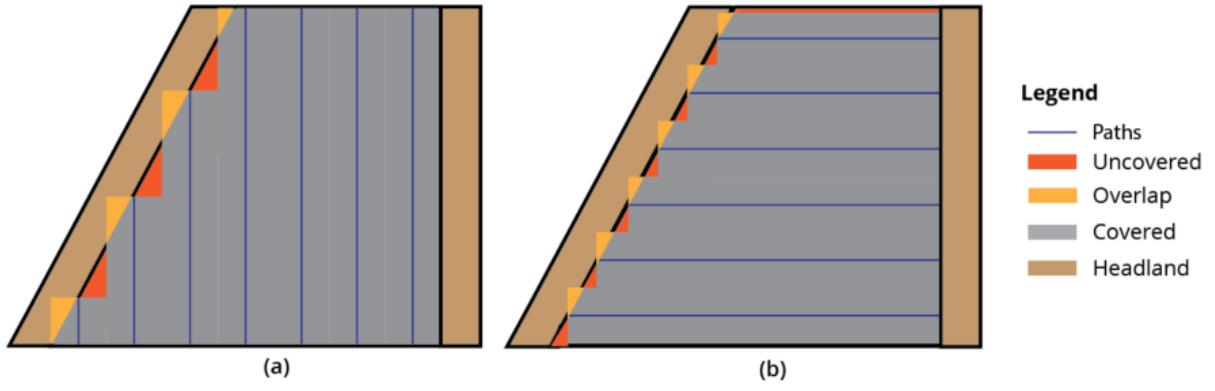


Figure 12: uncovered area & overlap area introduced by track to headland angle with two different path directions on the same field.

2.2.2 Spatial decomposition of a field using a sweep line.

A sweep line is an algorithmic design paradigm that refers to a line that is swept across Euclidian space (Berg et al., 2008). A common application of a sweep line in computational geometry is creating a Voronoi diagram (Fortune, 1987). In this thesis, a straight sweep line approach was adopted to decompose an input field. The optimal sweep line direction is unknown and has to be searched for. A sweep line algorithm was applied to the input field, in perpendicular direction to each boundary edge and in-field obstacle edge. This was done by rotating the input field for each unique edge direction, parallel to the x-axis. This is proven to contain the best sweep line direction, given that the tracks are placed parallel to one of the edges (Huang, 2001). Convex trapezoidal subfields were formed. Prior to applying the sweep line, vertices of the input field were assigned an event code. If an interior angle exceeds 180° , the geometry will be split at that vertex point (event 1). If the interior angle was less or equal to 180° , the vertex was ignored by the sweep line (event 0). Figure 13 provides an example of assigning event codes to vertices.

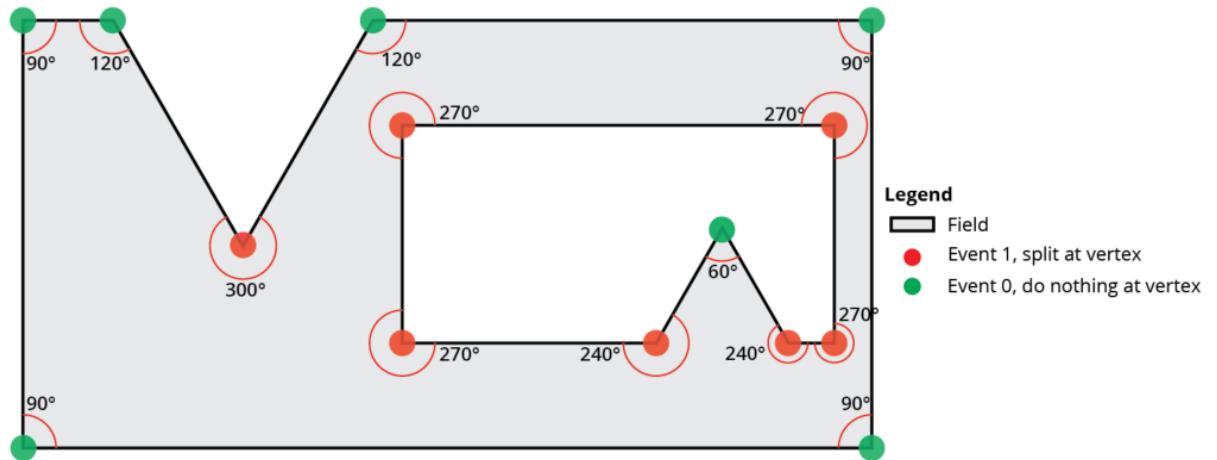


Figure 13: interior angle per vertex of a field with an obstacle feature. If the interior angle of the field exceeds 180° , the field will be split at that vertex during the sweep line algorithm execution (event 1). If it is equal or lower than 180° , the vertex is ignored (event 0).

Figure 14 provides an illustration of the decomposition process using a sweep line where figure 14(a) is decomposed in seven subfields illustrated by the dashed lines in figure 14(b).

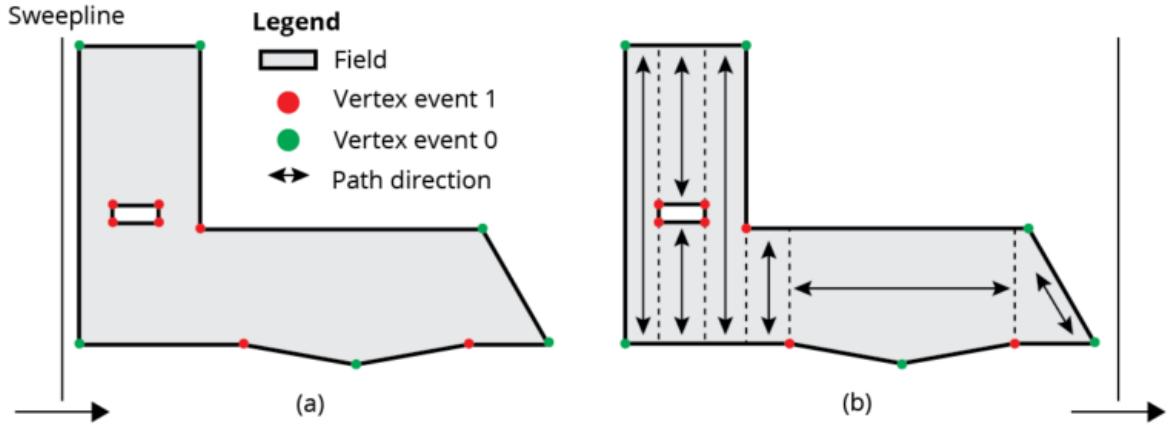


Figure 14: input field with an in-field obstacle (a) is decomposed into multiple convex subfields at vertex event 1 and separated by the dashed line (b) using a sweep line parallel to the y-axis. Optimal path direction per subfield is indicated by arrows.

The process of forming trapezoidal subfields of the input geometry can be described as exact cellular decomposition (Latombe, 1991). A drawback of exact cellular decomposition, however, is the creation of many subfields (Galceran et al., 2013). Therefore, exact cellular decomposition was used in combination with a separate merging algorithm. This allowed to reduce the amount of subfields and to account for specific restrictions for an agricultural coverage problem.

In Yu (2015), the sum of widths for the resulting subfields from exact cellular decomposition is calculated for each direction of the sweep line. When working with non-convex geometries, the problem was formulated as finding the convex layout \mathcal{D} of a non-convex polygon whilst minimizing the sum S of widths \mathcal{W} of the convex subfields i as shown in equation 2. The lowest sum was accepted as the optimal decomposition of the field. This would result in the minimal number of turns needed to cover the field. However, as discussed in chapter 2.1, this does not necessarily result in the optimal overall situation when considering overlap and uncovered area.

$$\text{Minimize } S(\mathcal{D}) = \sum_{i=1} \mathcal{W}_i \quad (2)$$

Therefore, an extension to this method was proposed which reduces the number of redundant subfields formed. A decomposition threshold value was introduced which allowed the user to provide the angle at which a subfield would be formed during the sweep line algorithm execution. By default, the MSW methodology used a 180° threshold, which would result in convex subfields. However, when working with more complex field geometries which contain many vertices at low intervals, it may provide an undesirable number of narrow subfields. Several subfields will contain the same optimal driving direction. In order to avoid this scenario, a decomposition threshold was introduced which controlled the assignment of event codes to vertices. Assignment of event code 1 (split at vertex) can be set at a higher interior angle than 180° using the decomposition threshold, thereby reducing the number of subfields.

Figure 15 demonstrates the implications of the decomposition threshold. In figure 16(a), vertices are marked for decomposition if it exceeds 180°, while in figure 15(b) this occurs at 200°. Two narrow subfields were formed in figure 15(a). This is avoided in figure 15(b) by increasing the threshold value of assigning event 1 (split at vertex) at 200°.

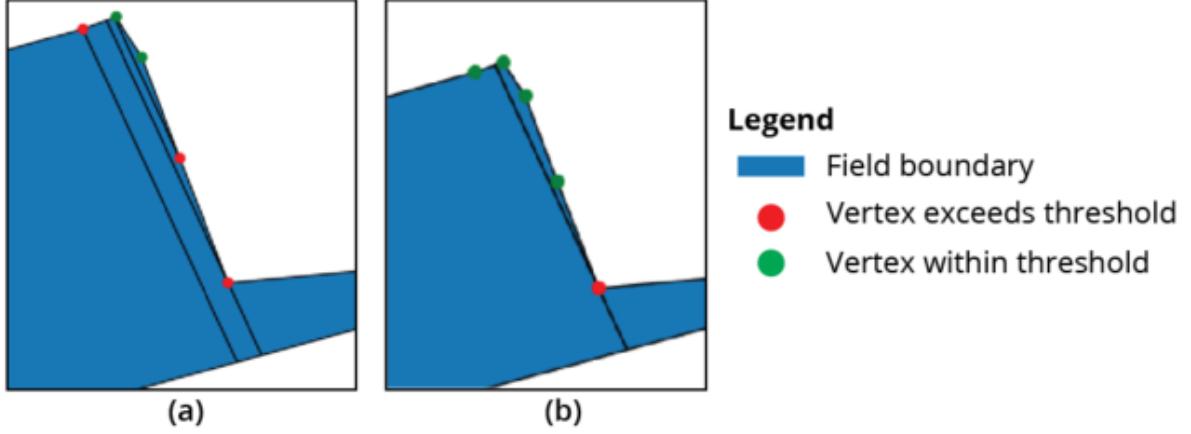


Figure 15: zoomed in section of a field with two decomposition threshold settings. Assigning event code 1 (split at vertex) at 180° in a, compared to assigning event code 1 at 200° in b. The creation of two narrow subfields was prevented in b compared to a.

2.2.3 Merging subfields

When a complex field is decomposed in multiple subfields, the algorithm finds the optimal coverage for each of the individual subfields. However, this does not necessarily mean that this results in the overall optimal solution as headlands may be formed in-field at adjacent subfields with different path directions (Hofstee et al., 2009). In addition, smaller subfields, with different path directions are to be avoided. Both situation are undesirable as introducing extra headland manoeuvres are expensive (Bochtis et al., 2019a; Bochtis et al., 2008). Therefore, subfields will be optimized using a set of requirements. After spatial decomposition of a field, the merging process was executed in three phases. The order of execution is important in order to reduce the number of redundant or impractical subfields.

The first phase will merge adjacent fields which have identical optimal driving direction angles. The second phase merges small subfields which would be impractical to cover separately. Small subfields were defined as subfields which have an area of less than 200 m². The subfield is joined to the adjacent field with which it shares the longest edge. Rather than merging it with the largest adjacent field, this helps to optimize the selection based on optimal driving direction. The third phase merges subfields if the optimal driving direction angles are within a defined merge threshold value. An upper and lower threshold is calculated per subfield. For example, a merge threshold is defined at 10°. A subfield with an optimal track direction of 90° will be merged with an adjacent field of not only 90° degrees, but anything equal to, or between 80° and 100°. After merging, the largest subfield angle was selected as the new optimal driving direction angle. When subfields were merged, the previous threshold values were saved. When a subfield which was merged before is selected to be merged again, all previous angle thresholds must be within the threshold value of the subfield to which it is being compared. Figure 17 provides an example of this situation.

Subfields figure 16(b) and figure 16(c) are merged first in phase 1 as their angles are exactly equal. Now the resulting subfield figure 16(bc) can be merged with figure 16(d) and figure 16(a) given a threshold of 12 degrees. However, as the field figure 16(a) and figure 16(d) are not within the threshold of 12 degrees, these should not be merged. Therefore subfield figure 16(bc) will be merged with either figure 16(a) or figure 16(d) but not both. Although not necessarily optimal, the restriction will prevent merging outside of the defined threshold in niche cases.

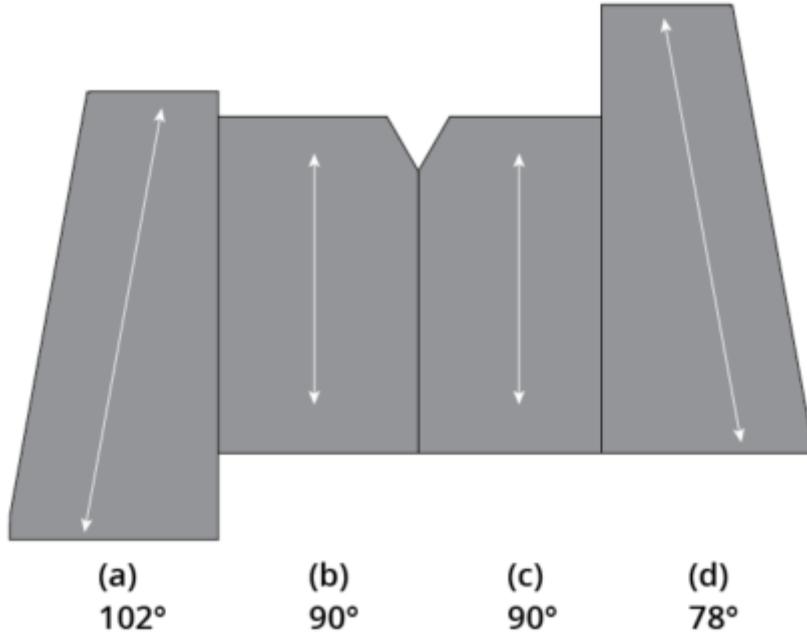


Figure 16: four subfields with optimal path direction indicated by white arrows and degrees.

2.2.4 Obstacle avoidance

Depending on the direction of the path and width of obstacles, placement of headland can be avoided and thereby increasing operational efficiency (Zhou et al., 2014). This scenario is shown in figure 17. In situation figure 17(a), the obstacles were circumvented, thus no extra path is needed. In situation figure 17(b), extra paths were introduced to avoid the obstacle. A threshold value (t) is introduced depending on the swath width. A bounding box was drawn around the obstacle with a parallel edge to the track direction to determine the width. Obstacles were classified in four categories with respective measures as shown in table 3.

Table 3: obstacle categories and their respective measures taken when encountering the situations.

Category	Measure
a	If the width of the obstacle (Δd) $< t$, the obstacle is circumvented.
b	Multiple obstacles close together are treated as one obstacle if the swath width is larger than the space between obstacles.
c	An obstacle near the edge of the field and is clipped out if the space between the edge and obstacle is less than the swath width. The field geometry is updated.
d	If the obstacle cannot be circumvented fully, extra headlands are generated.

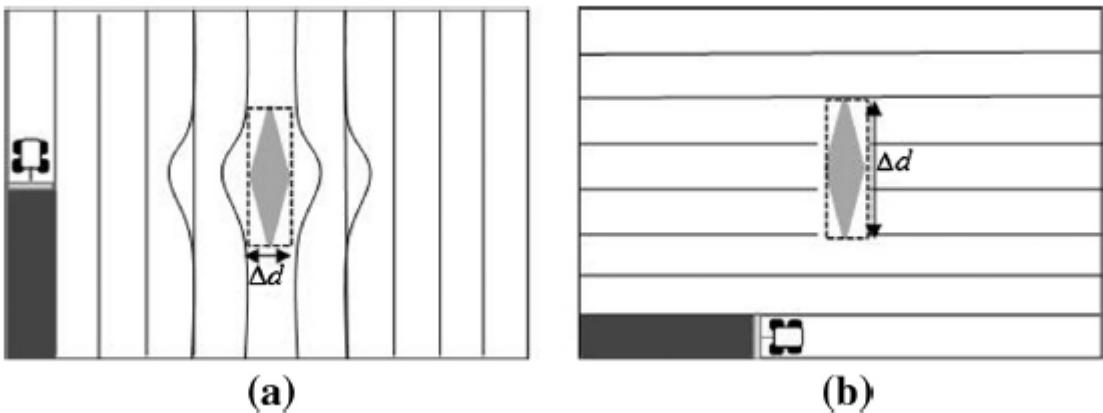
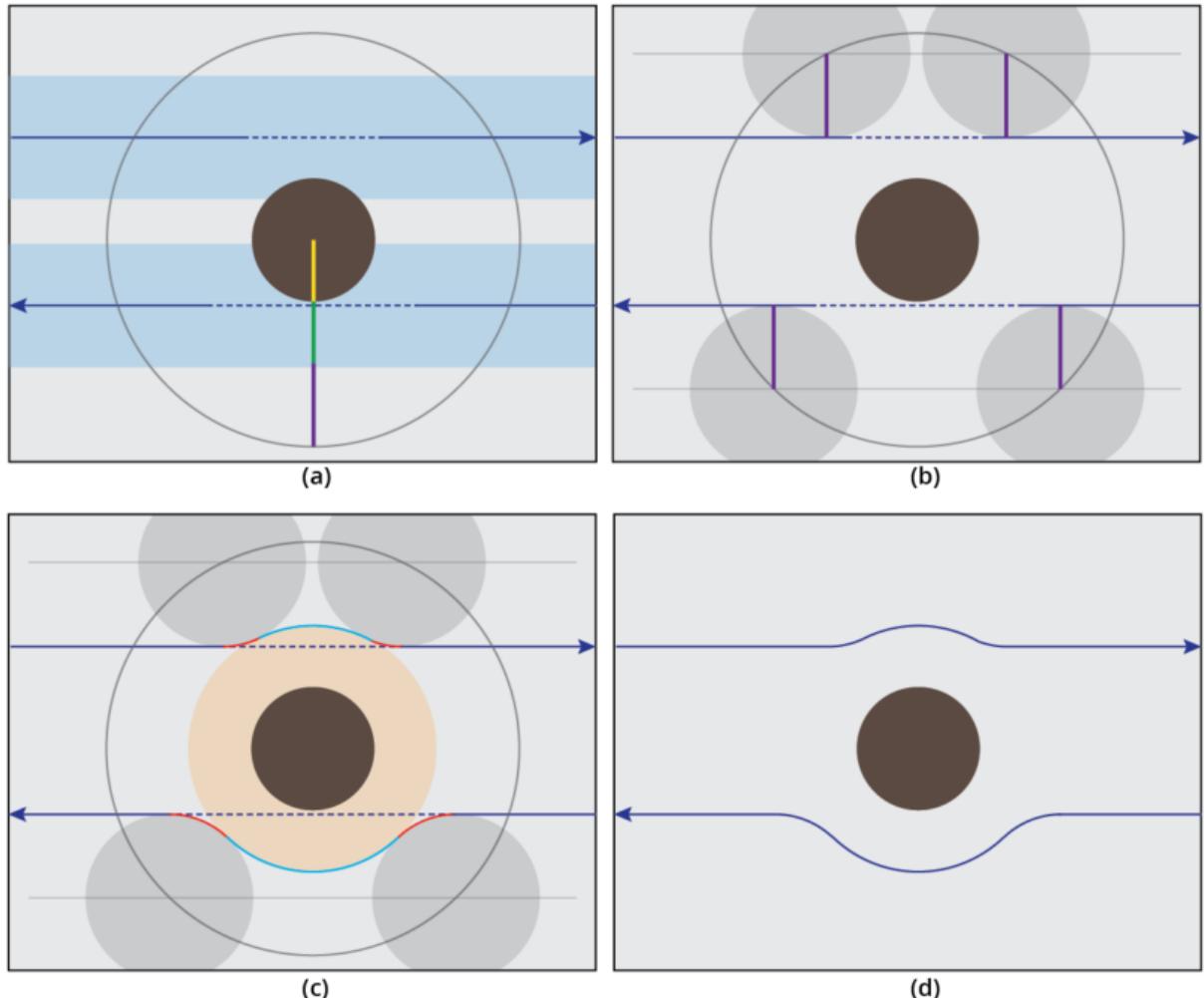


Figure 17: path direction determines if an obstacle is avoided. In a the obstacle is avoided as the swath width is larger than the width of the obstacle in this direction. In b, extra headland around the obstacle needs to be placed as the swath width is smaller than the width of the obstacle in this direction (Zhou et al., 2014).

In category *a* of table 3, Dubins path was used for generating smooth turns. This method revolves around determining the shortest curve between two points in a 2D space with a constraint on curvature of the path depending on the turning radius of the agricultural vehicle. Its application has been demonstrated in agricultural coverage and routing problems such as optimization of headland turning (Backman et al., 2015; Zhang et al., 2014) and obstacle avoidance (Liu et al., 2018). Figure 18 describes the method for circumventing circular obstacles categorized as *a* of table 3. The process is broken down into four consecutive steps.



Legend

- | | |
|-----------------------------|-----------------------------|
| → Path | — Radius of obstacle |
| ■ Swath | — Minimum turning radius |
| ○ Auxiliary circle obstacle | — Swath width / 2 |
| ● Obstacle | — New path mid turn |
| ○ Obstacle buffer | — New path begin & end turn |
| ● Auxiliary circles path | — Auxiliary lines |

Figure 18: obstacle circumventing using Dubins path. In a, auxiliary circle obstacle R_{aco} is calculated using the swath width, radius of obstacle and the turning radius of the vehicle. In b, auxiliary circles and lines are placed for determining a smooth entry and exit of the new path mid turn segment in c. New path begin, mid and end turn are merged with the original path in d to circumvent the obstacle. Adapted from (Liu et al., 2018).

Firstly in figure 18(a), *auxiliary circle obstacle* is calculated. The radius of *auxiliary circle obstacle* R_{aco} can be stated more formally by equation 3. Where R_o refers to the radius of the *obstacle* (yellow), w to the *swath width* (green) and r to *minimum turning radius* (purple) of an agricultural vehicle.

$$R_{aco} = R_o + r + \frac{w}{2} \quad (3)$$

Secondly, in figure 18(b), two *auxiliary lines* and four *auxiliary circles path* are calculated. *Auxiliary lines* are calculated by a parallel offset of the *path* with a distance equal to the *minimum turning radius* in the direction away from the *obstacle*. The *auxiliary circles path* (grey) radius is equal to the *minimum turning radius*. The *auxiliary circles path* (grey) are placed at the four intersections of the *auxiliary lines* with the *auxiliary circle obstacle*. Thirdly, in figure 18(c), the *new path mid, begin & end turn* (blue and red) which make up the *obstacle circumvention path* are calculated. *New path begin & end turn* are calculated by splitting the *auxiliary circles path* between the intersection of the *path* and *obstacle buffer* (orange). *Obstacle buffered with swath width / 2* and taking its outer edge. *New path mid turn* is calculated by taking the outer edge of the *obstacle buffer* between the intersections of the *auxiliary circles path*. Lastly, in figure 18(d), the *new path mid, begin & end turn* and *path* are joined together to form the *obstacle circumventing path*.

As a non-circular obstacle does not have a radius, the *auxiliary circle obstacle* cannot be calculated. Consequently, the *auxiliary circles path* intersection points cannot be calculated. Therefore, a variation of the method above is used when encountering a non-circular obstacle as shown in figure 19. The *obstacle* and *path* were rotated so the longest edge of the *path* is parallel to the y-axis. In figure 19(a), the *intersection buffered obstacle* is calculated by intersecting the left half of the *swath* with the *buffered obstacle*. The *buffered obstacle* represents the distance which must be kept by the *driving path* at all time to not collide with the *obstacle*. In other words, if the *path* follows the outer boundary of the *buffered obstacle*, the *swath* will not collide with the *obstacle*. The *intersection buffered obstacle* is therefore the part of the *driving path* geometry which must be mutated in order to avoid collision. In order to do so, *auxiliary circles* were generated which represent the minimum turning radius. The highest and lowest intersection points are determined for the *intersection buffered obstacle*. Using these points, the auxiliary circles middle points can be calculated:

Auxiliary point top = *top_intersection_point x coordinate - turning radius, top_intersection_point y coordinate + turning radius.*

Auxiliary point bottom = *bottom_intersection_point x coordinate - turning radius, bottom_intersection_point y coordinate - turning radius.*

The points were buffered according to the turning radius to form the auxiliary circles. The auxiliary circles were moved towards the *intersection buffered obstacle* parallel to the y-axis until they intersect as shown in figure 19(c) and figure 19(d). The *new path begin & end turn* were selected by splitting the auxiliary circles on the *path* and the outer boundary of

intersection buffered obstacle. The parts which intersected both the *intersection buffered obstacle* and the *path* and did not exceed the maximum and minimum x-coordinate of the *intersection buffered obstacle* were selected. The *new path mid turn* consists out of the segment which did not intersect with the *path*. The *new path mid, begin & end turn* and *path* are joined to form the new path and corresponding *swath* as shown in figure 19(c).

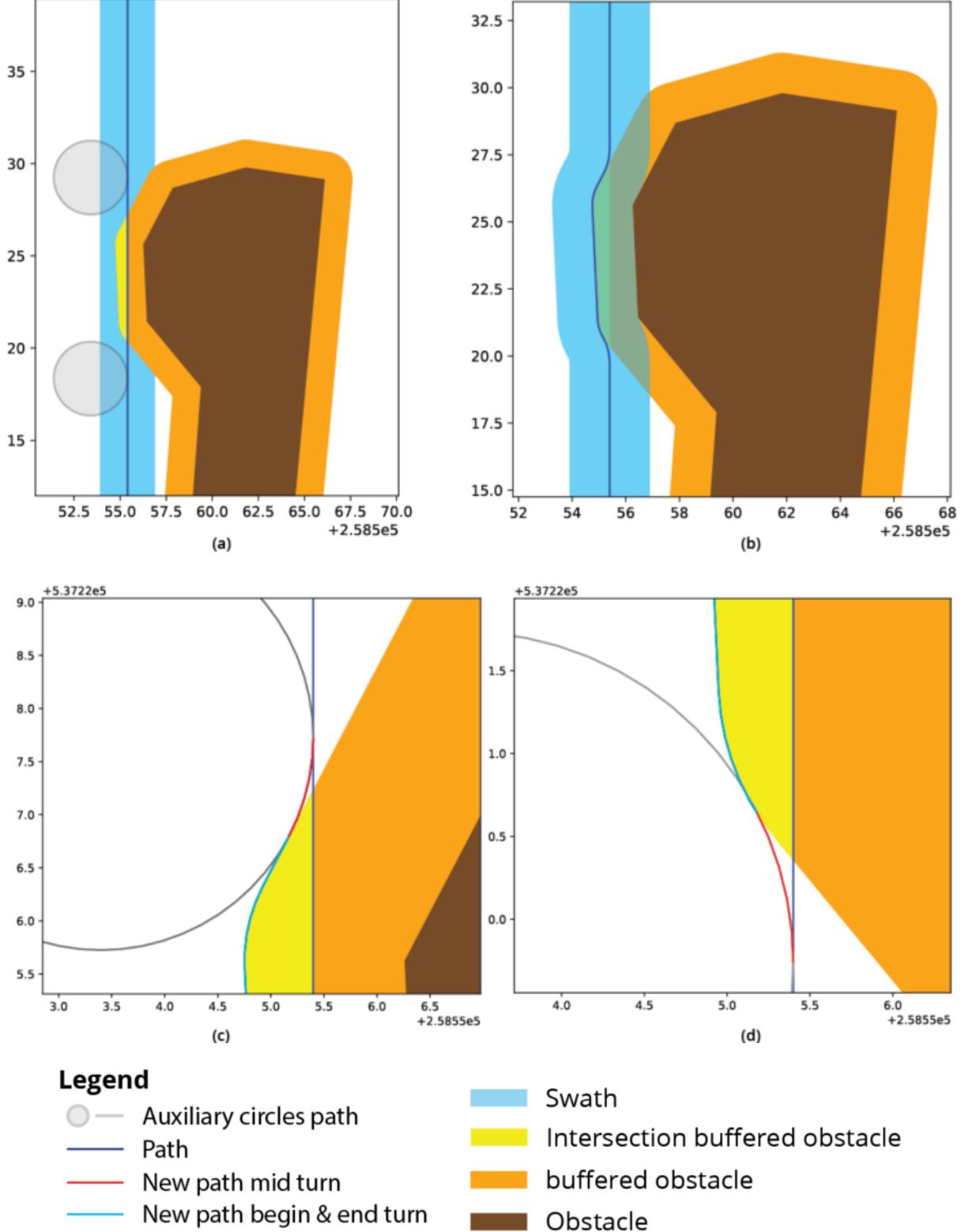


Figure 19: obstacle avoidance using Dubins path.

When a path intersects with an obstacle categorized as d in table 3, the path geometry must be shortened in order to avoid collision. The *path* and *obstacle* were rotated so the longest edge of the path was parallel to the y-axis. Figure 20 demonstrates the process of shortening the path. In this scenario, the obstacle cannot be circumvented as the path would intersect with adjacent paths. Therefore, the *path* and *swath* were trimmed by taking the symmetrical difference between the *swath/path* and the *stretched intersection*.

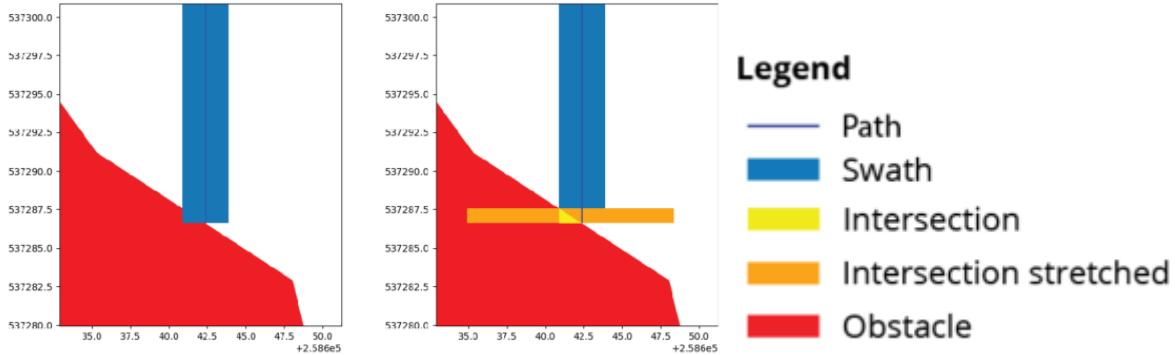


Figure 20: trimming the path and swath when an obstacle cannot be avoided. The intersection of the swath with the obstacle is calculated. A stretched bounding box is calculated for the intersection. The symmetrical difference between the swath/path and the intersection stretched is calculated.

2.2.5 Technical Implementation

For practical reasons, a Python-based approach was chosen in this thesis to implement the methodology. A working environment was set up in PyCharm 2018.3.3 (Community Edition) on a Windows machine using Python 3.6. The following Python libraries were used:

- Shapely 1.6.4, an open source library for computational geometry in a 2D plane. It is dependent on the Geometry Engine Open Source (GEOS) (Gillies, 2019b).
- Fiona 1.8.6, an open source library which is used for reading and writing geospatial data (Gillies, 2019a). It is dependent on GDAL (Geospatial Data Abstraction Library)/OGR.
- GeoPandas 0.4.0, an open source library which extends data types used by the well-known Pandas library to be compatible with geometrical types (e.g., point, line, polygon) (Geopandas, 2018). Geometric operations are defined by Shapely.
- Math, an always available library for mathematical functions used for calculations regarding angles.

The script was created by chaining a collection of custom-built functions. The modules decomposition and obstacle avoidance are described in a simplified version below.

Pseudo code field decomposition

Input: field geometry, swath width, merge threshold, decomposition threshold, weights
Output: path plan including paths (linestrings) and their corresponding subfields (polygons), overlap/uncovered geometry (polygons) and a cost according the objective function.

- Check user input for errors
- Read field geometry from geodatabase and check for errors (vertex order of interior/exterior features).

- Extract interior (obstacles) and exterior (field outer boundary) features.
- Get list of all interior and exterior edge angles in radians.
- For angle in angle list:
 - Rotate interior/exterior so the angle is perpendicular to x-axis
 - Assign decomposition event per vertex (interior/exterior), 1 of vertex angle > decomposition threshold, else 0.
 - Sort vertices based on x-coordinate. Small to large.
 - Apply sweep line algorithm on sorted vertex list. Generate a line geometry if vertex interior angle > decomposition threshold. If not, do nothing.
 - Create polygons of the line collection and determine the optimal driving angle according to the width of the smallest bounding rectangle.
 - Merge neighbouring subfields which are within the threshold value of similar driving path direction.
 - Generate paths as lines according to the optimal path angle, try to fit the paths so uncovered area is minimized.
 - Create paths as polygons according to swath width, correct paths if it encounters an obstacle or field boundary.
 - Calculate cost by intersecting the paths with the rotated geometry and calculating the uncovered / overlap area and using the weighted sum objective function.
 - Compare cost with previous angles.
- Recalculate original geometry orientation for the generated polygons and lines.
- Return path plan with least cost.

Pseudo code obstacle avoidance

Input: path (which collides with an obstacle), swath width, minimum turning radius, interiors.

Output: path polygons, path linestrings.

- Prepare for avoiding obstacle, rotate features so the paths longest edge is parallel to the y-axis.
- Categorize obstacle situation (table 3).
- Shorten the paths, shown in figure 20.
- Calculate avoidance path, see figure 18 & 19.
 - Create buffer first around the obstacle/sliver which represents the minimum distance to be kept.
 - Get the overlapping part of the buffered obstacle with the path.
 - Generate auxiliary circles by selecting the highest and lowest point of the intersection buffered obstacle with the path.
 - Keep moving circles towards obstacle until it does intersect.
 - Select the obstacle avoidance line segments by taking the boundary of polygon geometries and splitting them in separate linestrings.
 - Select top & bottom segment by splitting it with intersection buffered obstacle and selecting the line segment which touches both the path and intersection buffered obstacle.

- Select middle segments which are furthest away from the original path.
- Select the segments from the original path which do not need to be edited.
- Join segments to form the new path which avoids the obstacles.
- Recalculate original geometry orientation the new path.
- Return path.

2.2.6 Testing

The test fields introduced in section 2.1.6 were used for demonstrating the effects of changing the merge threshold, the decomposition threshold and the turning radius. The influence of chosen thresholds for merging and decomposition depends on the input geometry. To demonstrate the effects of varying thresholds, ten different settings were tested. An overview of the settings is provided in table 4. For the sake of clarity and conciseness, field 1087 (figure 9) was selected in particular to demonstrate the effects as this field proved to be the best, visual, example. Important to note is that for all test cases a swath width of 3 meters and weight settings from scenario 3 (minimize turns), chapter 2.1.6. were used. Three different settings for each parameter were tested. Each setting ranging from little to no impact, to a large impact.

Table 4: parameter settings per test case.

Test cases	Merge threshold (degrees)	Decomposition threshold (degrees)	Minimal turning radius (meters)
Setting 1, default	0	180	6
Setting 2	15	180	6
Setting 3	30	180	6
Setting 4	60	180	6
Setting 5	0	205	6
Setting 6	0	230	6
Setting 7	0	255	6
Setting 8	0	180	3
Setting 9	0	180	9
Setting 10	0	180	12

Using the outcome of the visual assessment, it was determined to fix the merge threshold at 15 degrees and the decomposition threshold at 205 degrees during the comparison of path plans in section 3.3.

2.3 Path plan comparison

Results of the algorithm were compared with a selected farmer practice and path plans found in literature.

2.3.1 General approach

A total of three fields and their respective path plans were compared; one from practice and two from literature. The novelty of the developed algorithm lies in the combination of field decomposition with obstacle avoidance. Therefore, the selection of fields used in this chapter all contain obstacles. Additionally, the fields from literature were selected based on the type of methodology applied, e.g., spatial decomposition or obstacle avoidance. The field from observed in practice was selected based on its crop type (potato, as these provide clear path plans on aerial imagery) and geometrical shape. In order to compare path plans, the observed paths were manually traced using ArcGIS Pro. Due to the time consuming nature of manually digitizing path plans, only three fields were compared.

Default parameter values (used if no alternative was provided) were set as follows. The swath width default was set to 3 meters to match a common potato crop (four ridges, each 75 cm wide). The default turning radius was set to 6 meters (Oksanen, 2007). Since headland were not taken into account in the test fields from literature (Liu et al., 2018; Oksanen, 2007), these were assumed to have been removed. For calculating overlap and uncovered area (as described in figure 12, chapter 2.2.1), headlands were assumed to be located around the edges of the outer borders of the input field.

2.3.2 Fields from literature

Fields from literature are often provided in vector formats which can be geo-referenced in GIS software.

Field Oksanen

Field Oksanen (Oksanen, 2007) shown in figure 21 was chosen to assess decomposition. Swath width was set on 2.5 meters (Oksanen, 2007). From the several examples provided by Oksanen (2007) this field was selected as it contained an easily digitized obstacle which allowed a realistic comparison.

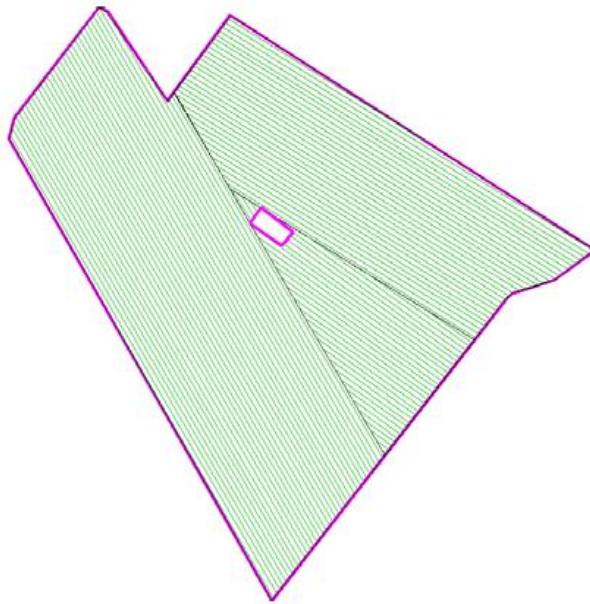


Figure 21: field from literature focussed on decomposition (Oksanen, 2007).

Field Liu

Field Liu (Liu et al., 2018) shown in figure 22 was chosen to assess obstacle avoidance. Liu et al. (2018) includes a table with the coordinates of vertices representing exterior and interior boundaries. The swath width was set to 10 meters and the turning radius was set to 3 meters. The given path plan (red lines, figure 22) was converted into separate paths by manually removing the turning manoeuvres.

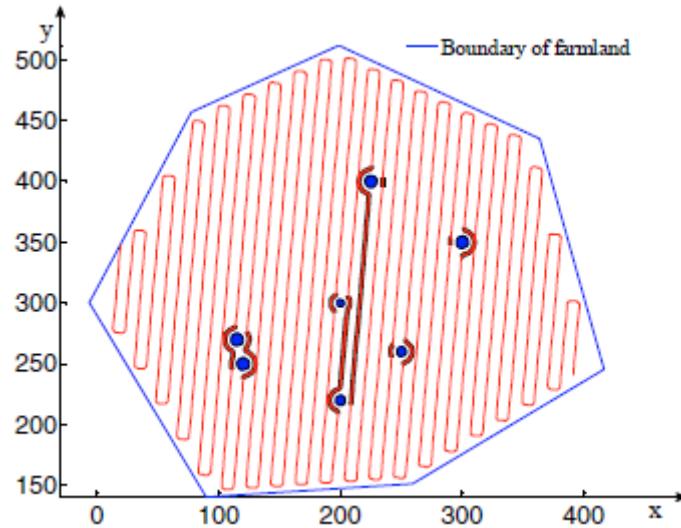


Figure 22: field from literature focussed on obstacle avoidance (Liu et al., 2018).

2.3.3 Field from practice

An aerial image for a selected crop year was retrieved via Google Earth. The BRP geometry of the selected field was split in subfields according to the observed path plan on the aerial image. For each resulting subfield, a paths were digitized.

Field near Linne

Figure 23 shows the selected field, which is near the village Linne in Limburg. From left to right, it contains one natural obstacle, one power pylon infield and one power pylon on the edge. The obstacles were categorized as d , d , c according to the obstacle classification table 3 in section 2.2.4. A long path to the middle further increases the field complexity. According to BRP, it contained potatoes were grown in 2018; accordingly, the default swath width was used. The field was split into four subfields. Interesting to note is that the longest subfield contained a curved path which the developed algorithm is unable to generate.

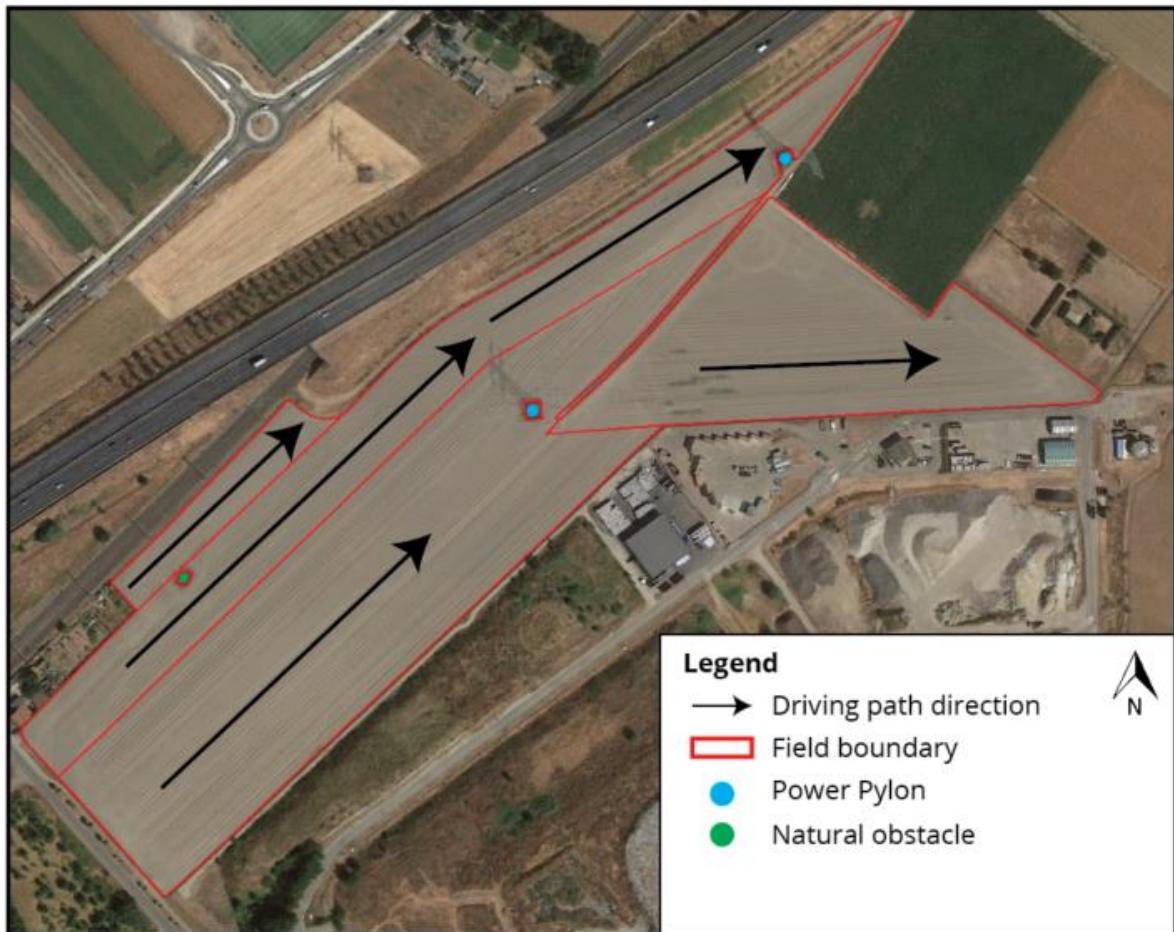


Figure 23: aerial image from field with obstacle with the respective BRP boundary split in subfields. EPSG:4326: X: 5.934042, Y: 51.144340) (Google Earth, 2019).

2.3.4 Comparison

The three factors from the objective function, uncovered area, overlap area, number of turns, were calculated using the objective function with the digitized fields as input. For each of the testing fields, the developed algorithm generated 3 path plans according to the scenarios formulated in chapter 2.1.6, each of which minimizes one of the three factors; overlap uncovered and number of turns. The dimensionless sum S of equation 1 was calculated given a set of weights reflecting the costs of the path plan which were used for comparison of the results of the developed script with the references from literature or farmer's practice. Various weight settings were tested as shown in table 5. For field Liu, the

merge and decomposition thresholds were ignored as the input geometry was already convex.

Table 5: input parameters per testing field used for comparison.

	Swath width (meters)	Turning radius (meters)	Headlands	Merge threshold (degrees)	Decomposition threshold (degrees)
Field Oksanen	2.5	6	No	15	205
Field Liu	10	3	No	n/a	n/a
Field near Linne	3	6	No	15	205

3. Results

This chapter elaborates the results per research question.

3.1 Definition of an optimal path plan

Three different scenarios were tested on three different fields with varying geometries. Each scenario, 1, 2 and 3, revolves around minimizing one of the three factors, overlap, uncovered or number of turns, respectively. The script succeeded in minimizing each of the factors in their corresponding scenario. The output cost from equation 1 was not presented as comparing different scenarios from the same field to each other does not make sense for a dimensionless sum.

3.1.1 Field 1739

Table 6 provides an overview of the results of the three scenarios for field 1739. The number of turns, percentage overlap and uncovered area are presented for the resulting path plan for each scenario.

Table 6: results of various scenarios on field 1739 in terms of overlap area, uncovered area, number of turns and number of subfields.

	See figure(s)	% overlap	% uncovered	Number of turns	Number of subfields
Scenario 1, minimize overlap	24, 25	0.32	1.13	189	2
Scenario 2, minimize uncovered	26	0.35	0.56	192	3
Scenario 3, minimize number of turns	27	0.53	0.97	111	1

Interestingly, minimizing number of turns in scenario 3 was achieved by not splitting up the field at all. This is explained by the rectangular obstacle which is oriented parallel with its longest edge to the exterior field boundaries longest edge. Figure 25 shows a successful obstacle circumvention. Scenario 1 and 2 both adopted a similar main driving direction. Scenario 2 only differs from 1 by having a small rectangular subfield at the bottom left. The exterior field complexity introduced a number of too small paths in scenario 1 at the top right and bottom left part. The small subfield near the large obstacle could not fit perfectly which left a large uncovered area which resulted in a high percentage of uncovered area.

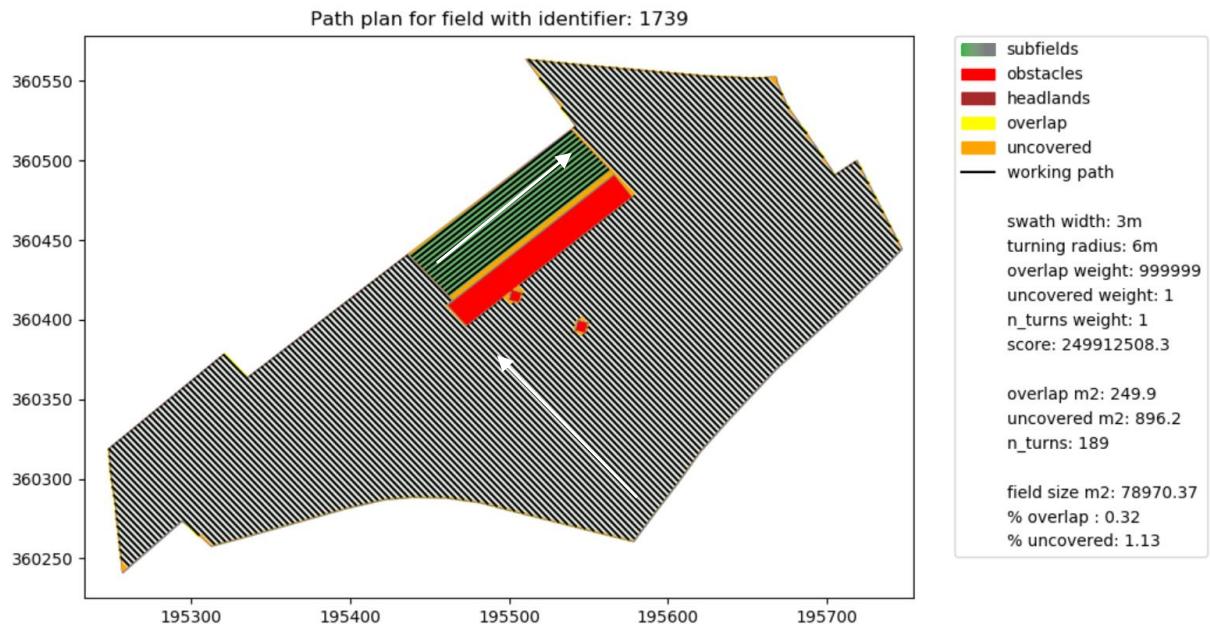


Figure 24: path plan for field 1739, scenario 1, minimizing overlap area. White arrows indicate path direction per subfield.

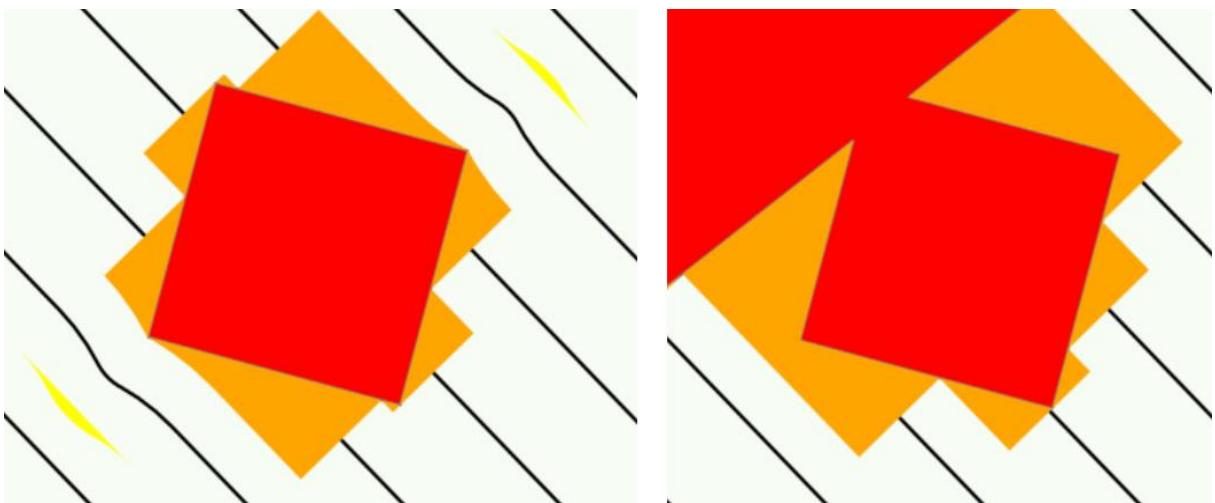


Figure 25: obstacle situation path plan field 1739, scenario 1.

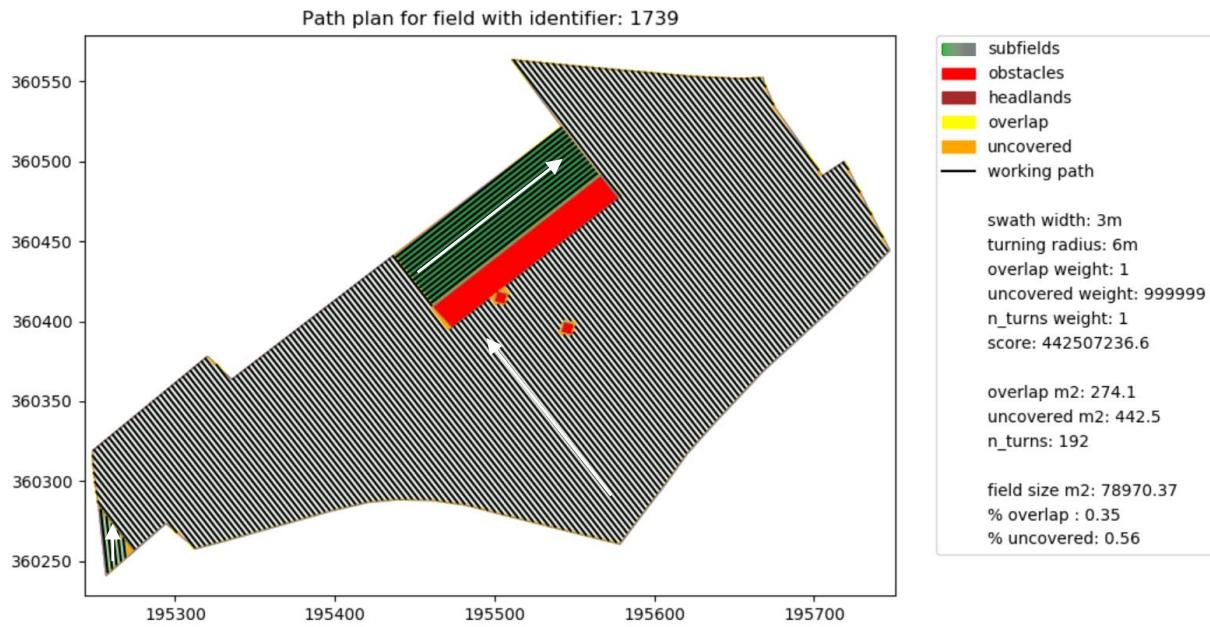


Figure 26: path plan for field 1739, scenario 2, minimizing uncovered area. White arrows indicate path direction per subfield.

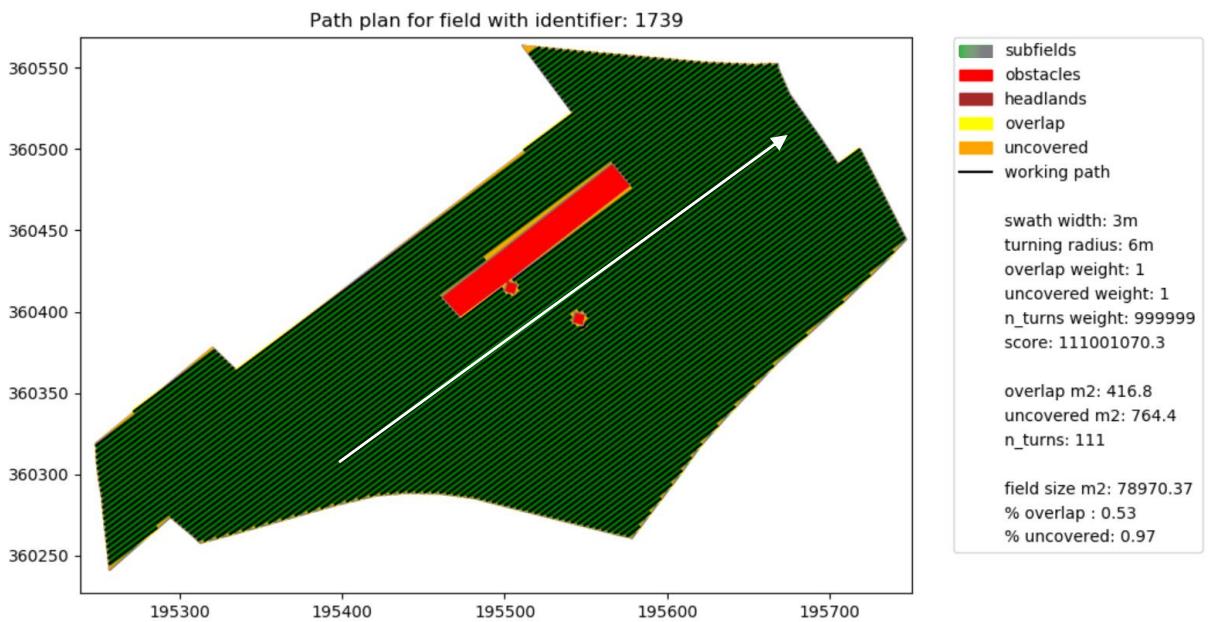


Figure 27: path plan for field 1739, scenario 3, minimizing number of turns. White arrows indicate path direction per subfield.

3.1.2 Field 5458

Table 7 provides an overview of the results of the three scenarios for field 5458. The number of turns, percentage overlap and uncovered area are presented for the resulting path plan for each scenario.

Table 7: results of various scenarios on field 5458 in terms of overlap area, uncovered area, number of turns and number of subfields.

	See figure(s)	% overlap	% uncovered	Number of turns	Number of subfields
Scenario 1, minimize overlap	28	0.53	0.72	112	1
Scenario 2, minimize uncovered	29	0.61	0.65	113	2
Scenario 3, minimize number of turns	30	0.64	0.92	88	3

Scenario 1 and 2 both adopted a similar main driving direction. As one of the edge is much longer than the others, the paths were chosen approximately perpendicular to that edge to minimize overlap area and uncovered area. In scenario 2, a slightly different angle path angle was adopted which resulted in 2 subfields, which have almost the same optimal path angle. Scenario 3 resulted in a path plan with 3 subfields and adopted a path angle almost parallel to the longest edge. However, the apparent longest edge was split into multiple segments. This resulted in an increased uncovered area compared to the other scenarios, as the largest subfield introduces a stretched, thin uncovered area.

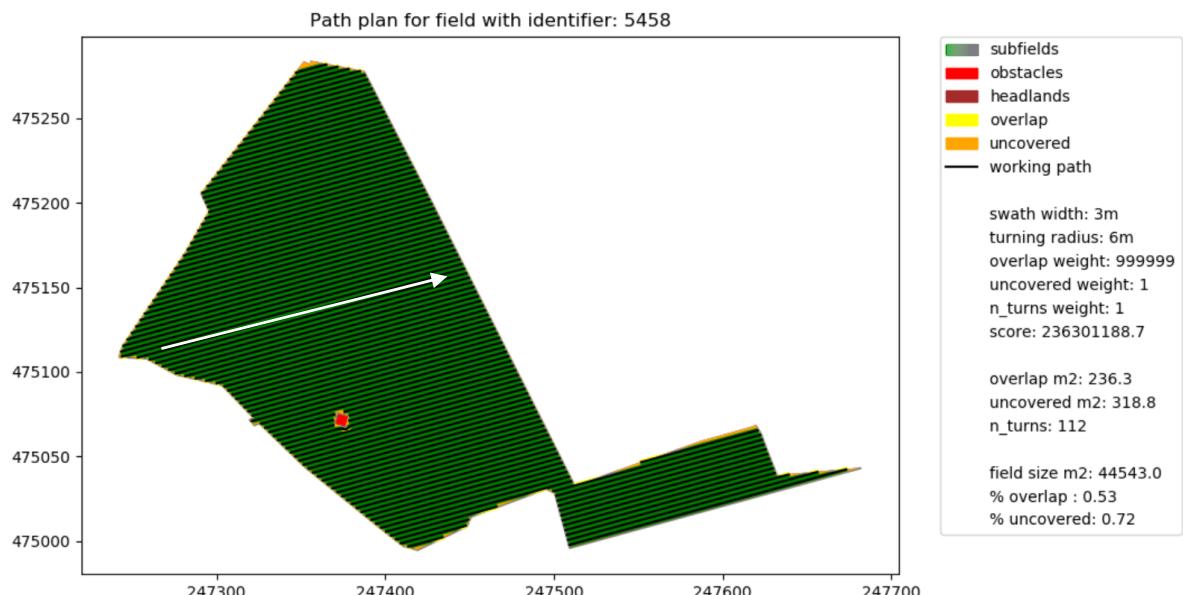


Figure 28: path plan for field 5458, scenario 1, minimizing overlap area. White arrows indicate path direction per subfield.

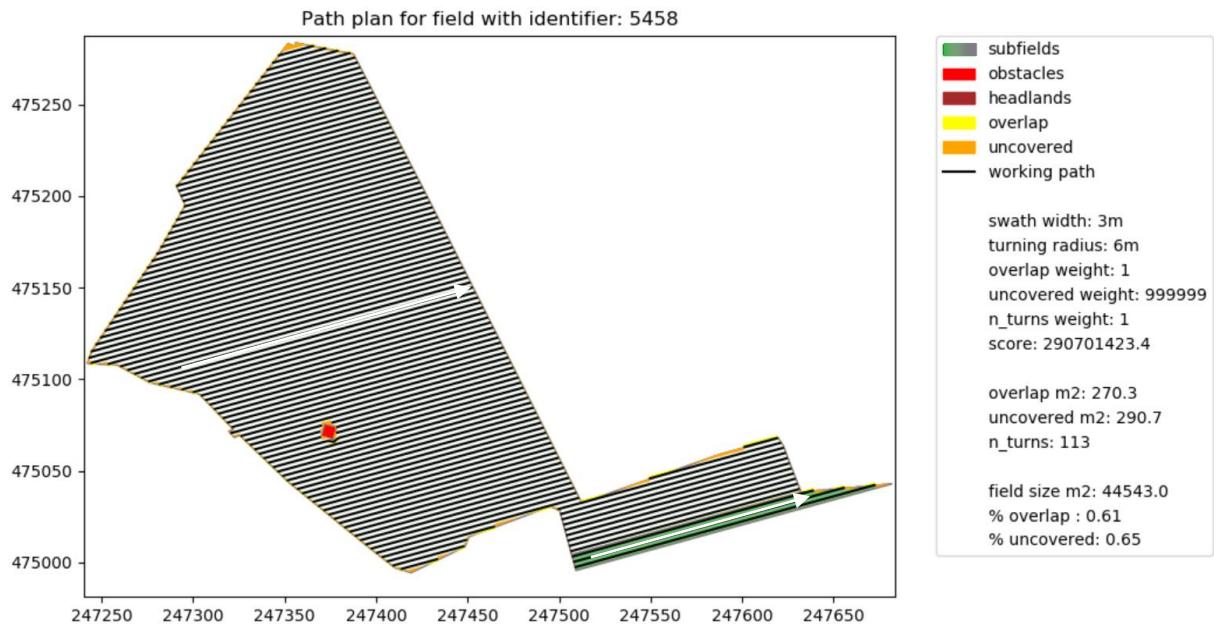


Figure 29: path plan for field 5458, scenario 2, minimizing uncovered area. White arrows indicate path direction per subfield.

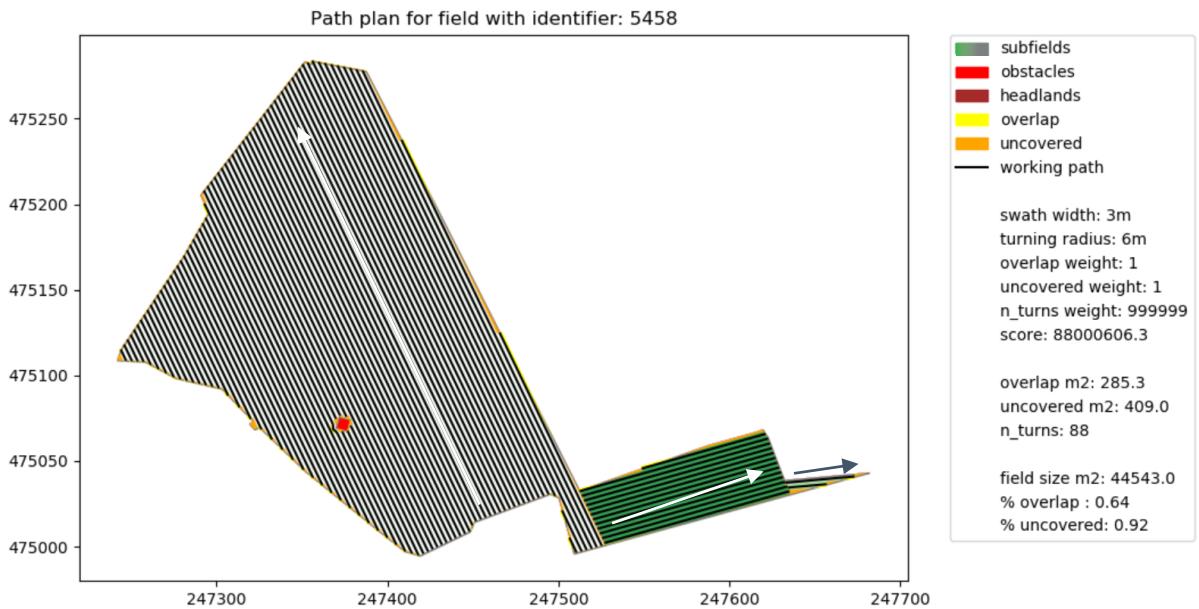


Figure 30: path plan for field 5458, scenario 3, minimizing number of turns. White & grey arrows indicate path direction per subfield.

3.1.3 Field 1087

Table 8 provides an overview of the results of the three scenarios for field 1087. The number of turns, percentage overlap and uncovered area are presented for the resulting path plan for each scenario.

Table 8: results of various scenarios on field 1087 in terms of overlap area, uncovered area, number of turns and number of subfields.

	See figure(s)	% overlap	% uncovered	Number of turns	Number of subfields
Scenario 1, minimize overlap	31	0.53	0.70	143	3
Scenario 2, minimize uncovered	32	0.82	0.59	129	3
Scenario 3, minimize number of turns	33	0.55	0.72	112	4

Field 1087 provided a large variety regarding the decomposition of fields. Interesting to note is that scenario 1 and 3 both have very similar percentages of overlap and uncovered, whilst scenario 3 achieved this in much fewer turns. All scenarios share the creation of triangles at the corners which are caused by the irregular shape and numerous smaller angles ($< 90^\circ$) at the vertices of the exterior.

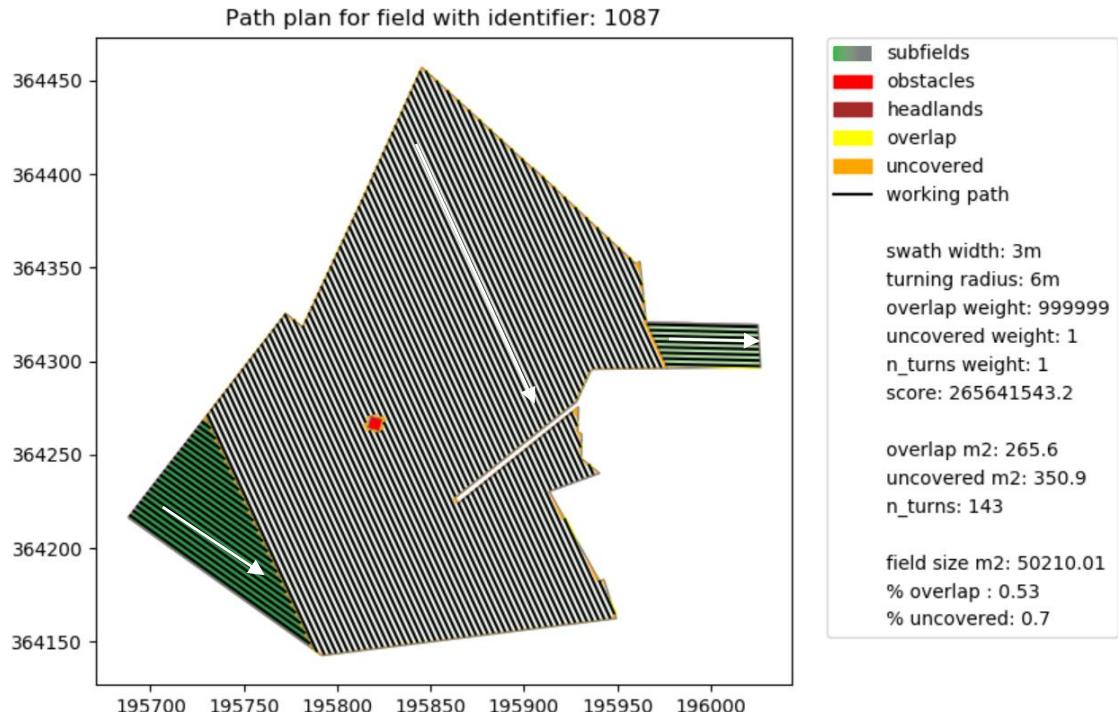


Figure 31: path plan for field 1087, scenario 1, minimizing overlap. White arrows indicate path direction per subfield.

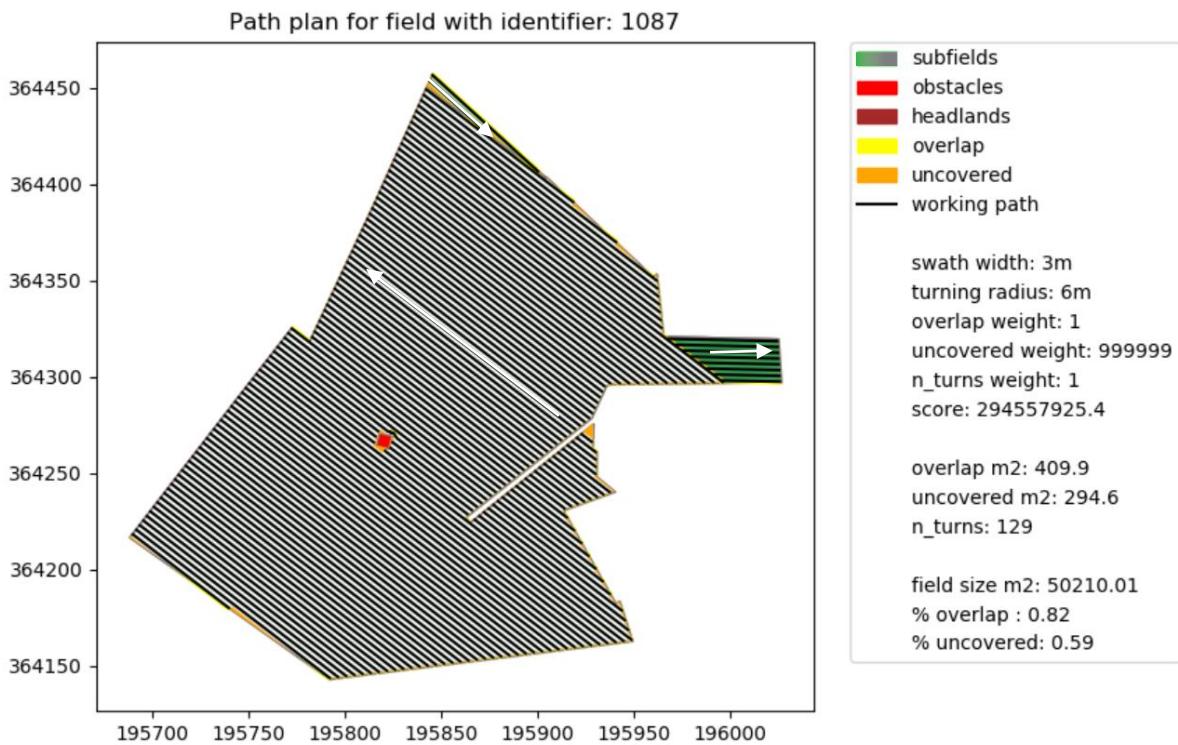


Figure 32: path plan for field 1087, scenario 1, minimizing uncovered. White arrows indicate path direction per subfield.

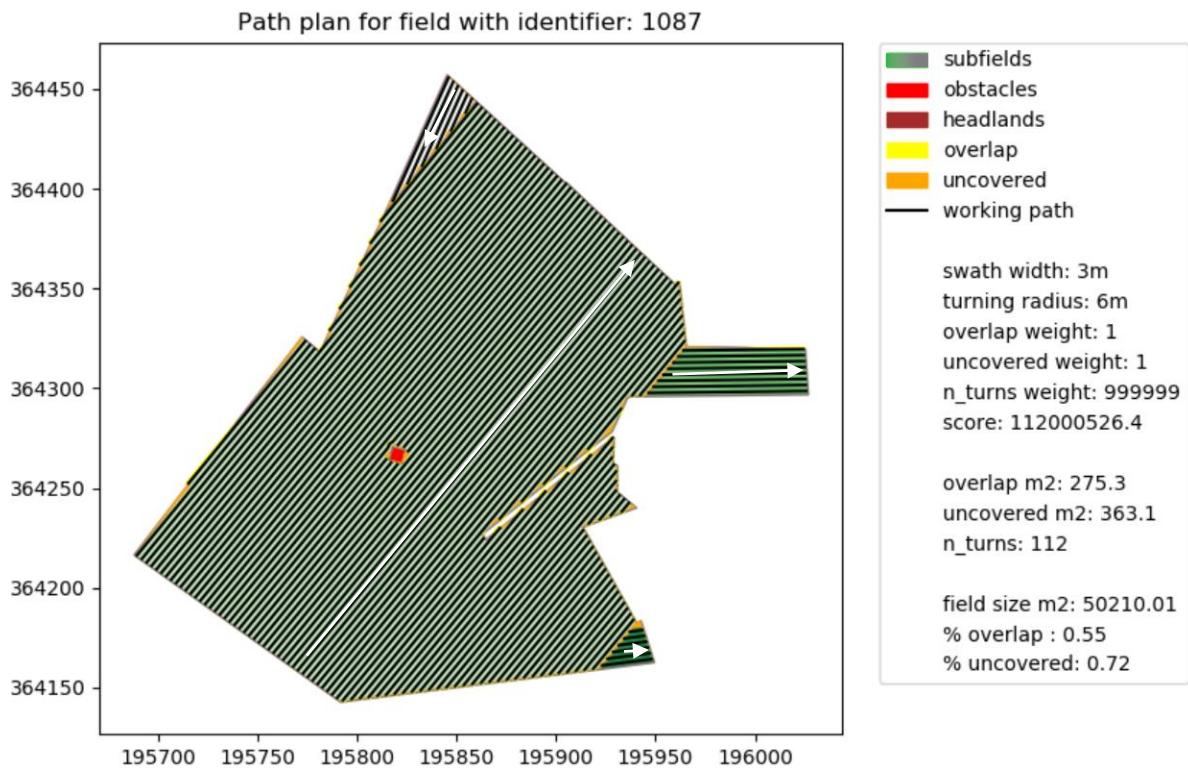


Figure 33: path plan for field 1087, scenario 3, minimizing number of turns. White arrows indicate path direction per subfield.

3.2 Path coverage generating algorithm

Below, performance, field decomposition and obstacle avoidance are described in separate subsections. A common legend (figure 34) is provided which applies to all figures in this section.

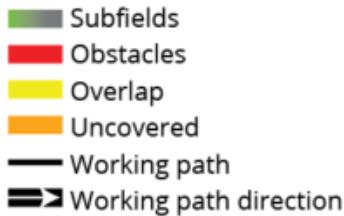


Figure 34 Common legend of figures in section 3.2.

3.2.1 Performance

The proposed methodology was successfully implemented in a Python environment. Feasible results were obtained within reasonable time. Table 9 provides an overview of general performance. It was noted that the obstacle avoidance module was the most expensive to execute; the higher the number of obstacles, the longer computation time was needed. The same was observed for number of angles in relation to the total computation time. This was to be expected, as each angle adds an iteration to the number of possible solutions compared.

Table 9: performance figures for testing fields.

Field	Number of obstacles	Number of angles	Time (seconds)
1739, figure 7	2	69	84.34
5458, figure 8	1	40	28.16
1087, figure 9	1	38	24.15

The script was found to be robust and it rarely crashed. Especially the spatial decomposition module was reliable. Occasional crashes were caused by the obstacle module. This was observed in situations having a long narrow obstacle (e.g., concrete path) with a smaller obstacle (e.g., power pylon) at a distance less than the swath width. In most situations, these crashes could be avoided by skipping the current sweep line angle and the script would be able to continue with the next angle. The downside was that a possible optimal solution would be skipped. The script notified the user when iterations were skipped.

3.2.2 Field decomposition

For convex fields, the optimal solution was observed to be parallel to the longest edge, which was to be expected considering no subfields would have to be created. For convex fields, the three different scenarios of section 2.1.6 produce the same path plans. It was found that geometrically simple fields (almost convex), generally obtain in a similar path plan as convex

fields, given that a clear longest edge would be present. For geometrically complex fields (many obstacles, irregular shape), the path plans tend to differ more between the scenarios.

Figure 35 shows four examples of a varying merging threshold (0, 15, 30 and 60 degrees) on path plans generated for field 1087. Table 10 shows the corresponding overlap area, uncovered area and number of turns per test case. Situation figure 35(c) reduces the number of subfields from 4 to 3 compared to figure 35(a) and figure 35(b) whilst still retaining the same number of turns. Situation figure 35(d), however, reduced the number of subfields to two but introduced an extra turn compared to figure 35(a), figure 35(b), and figure 35(c). In addition, the total overlap and uncovered area changed when subfields were removed as a result of changing track to headland angle.

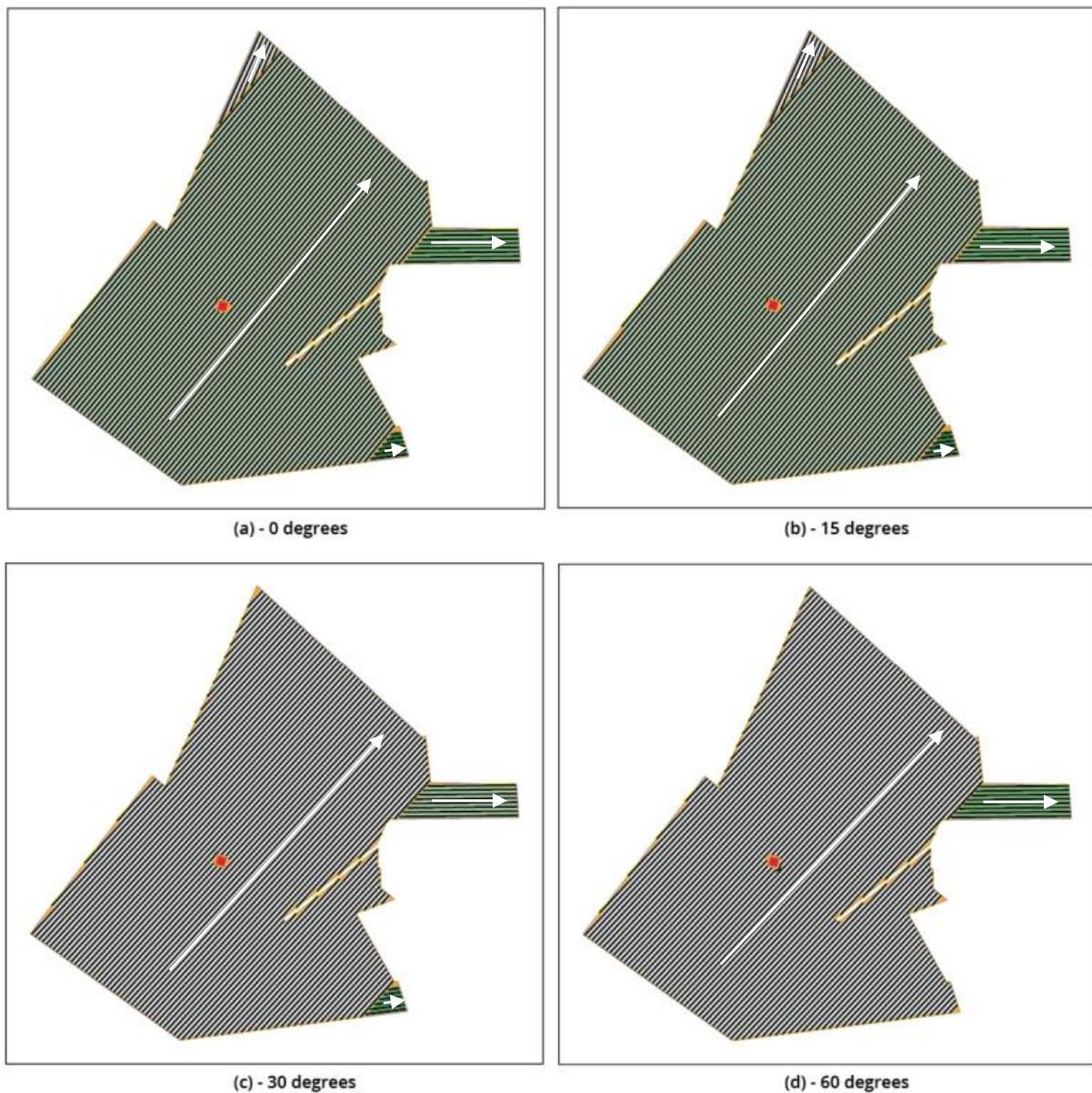


Figure 35: spatial decomposition and path plans with varying merge thresholds for field 1087.

Table 10: overlap, uncovered and number of turns for different merge thresholds applied to field 1087.

Merge threshold (degrees)	Overlap (%)	Uncovered (%)	Number of turns
0	0.55	0.72	112
15	0.55	0.72	112
30	0.55	0.67	112
60	0.61	0.64	113

Figure 36 shows four examples of different decomposition thresholds. Table 11 shows the Corresponding overlap area, uncovered area and number of turns per test case. Not surprisingly, the 195, 210 and 240 thresholds (figure 36, b, c and d) all produced the same outcome. This is to be expected, as the number of vertices marked for decomposition remain the same for these test cases. These vertices determine the creation of subfields by the sweep line during spatial decomposition and will therefore all produce identical path plans. Note that a difference of 180 degrees (figure 36(a)) to 195 degrees (figure 36(b)) reduced the total number of vertices needed for decomposition 15 from 40. By increasing the threshold from 180 to 195 degrees, the creation of the smaller subfield in the top left corner was prevented but it was replaced by two other, triangle shaped subfields.

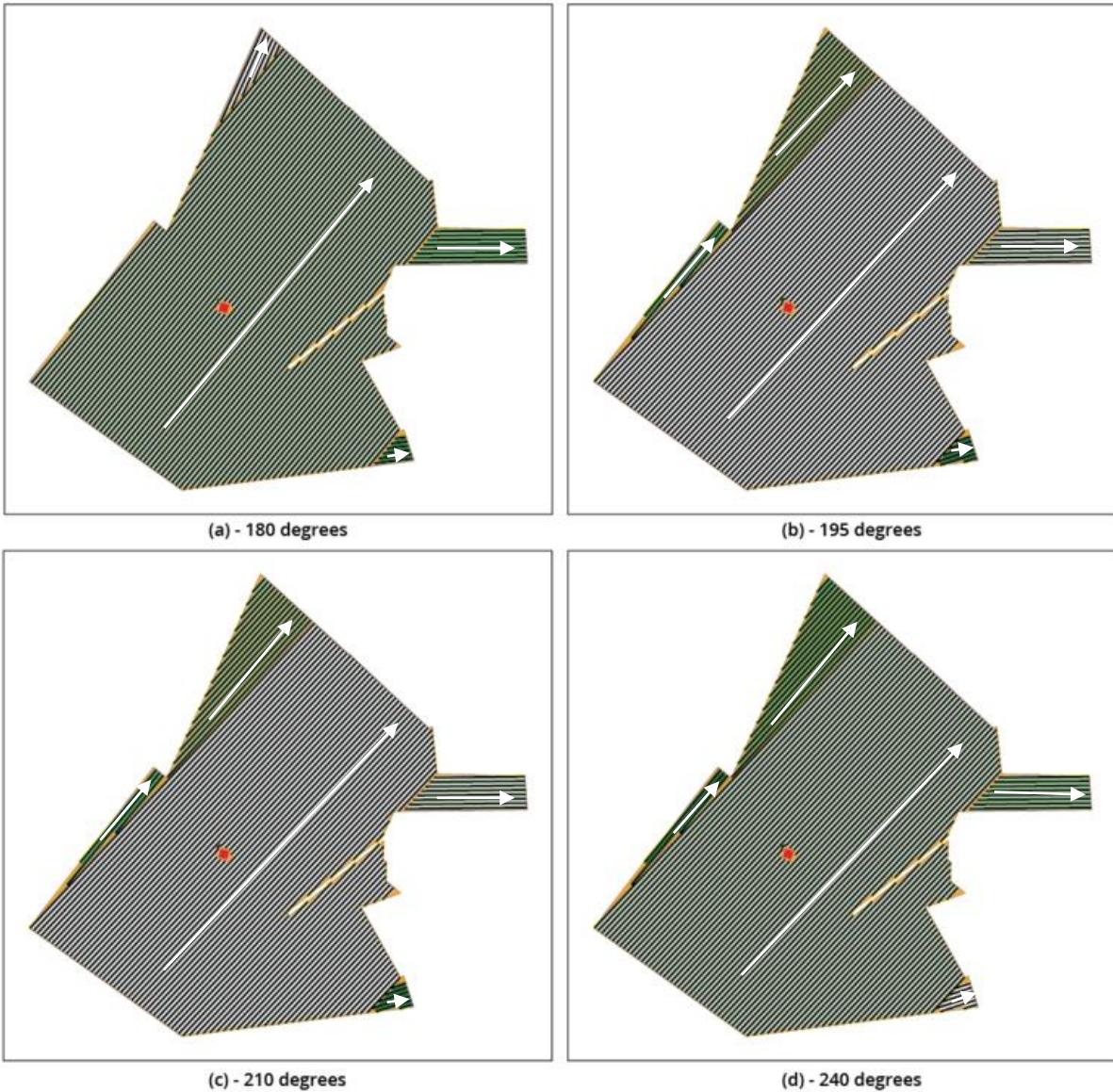


Figure 36: spatial decomposition and path plans with varying decomposition thresholds for field 1087.

Table 11: overlap, uncovered and number of turns per testing case for changing decomposition threshold.

Merge threshold (degrees)	Overlap (%)	Uncovered (%)	Number of turns	Number of vertices exceeding decomposition threshold
180	0.55	0.72	112	40
195	0.86	0.86	112	15
210	0.86	0.86	112	15
240	0.86	0.86	112	15

3.2.3 Obstacle avoidance

Figure 37 shows the effect of changing the minimum turning radius. The implemented methodology successfully handled the changing parameter as reflected by the increasing size of the overlap area in relation to the increasing minimum turning radius.

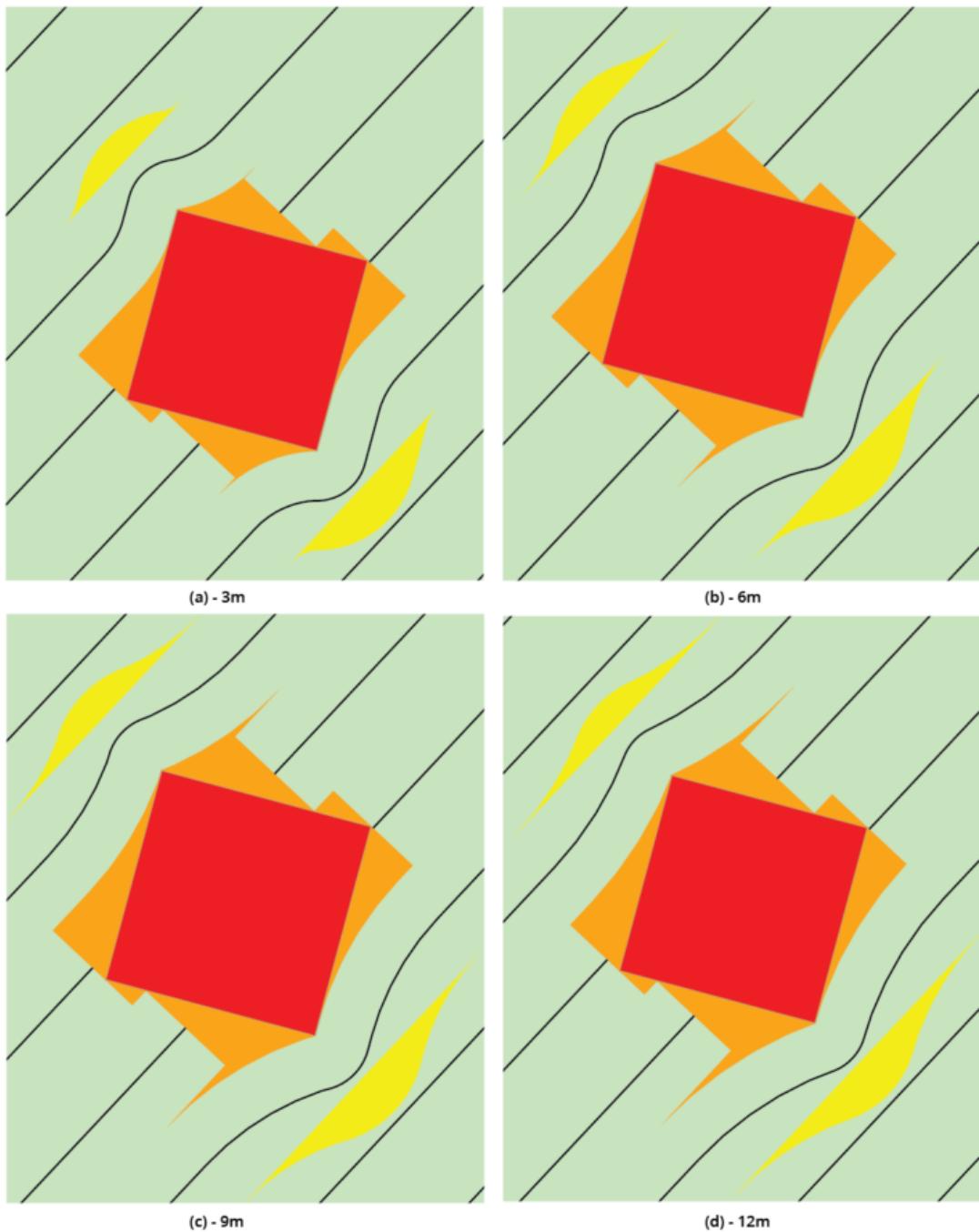


Figure 37: obstacle circumventing for different minimum turning radii.

In addition to circumventing smaller obstacles, the module was also able to successfully avoid larger obstacles using a larger avoidance path as is demonstrated in figure 38.

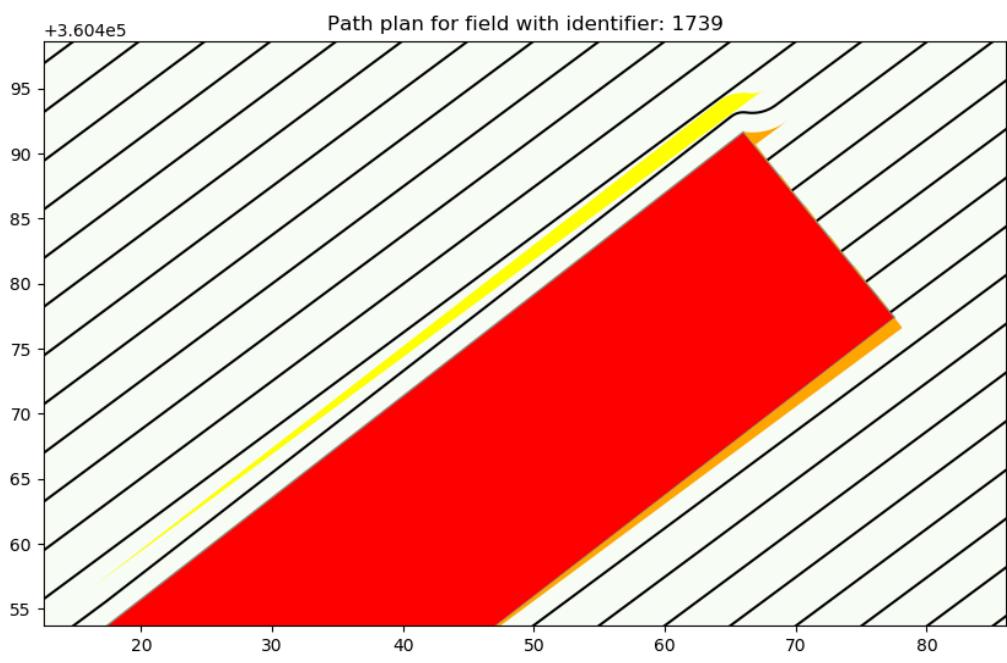


Figure 38: obstacle circumventing applied on a larger obstacle.

3.3 Path plan comparison

The selected fields from literature and from practice and their respective path plans were digitized. The output was used to calculate uncovered area, overlap area and the number of turns which were used as input for calculating a cost using the objective function (equation 1). The results are compared with the output of the developed algorithm and are presented in this section. Scenario 1, 2 and 3 represent minimizing overlap, uncovered and number of turns respectively in accordance with section 2.1.6, table 2. The script managed to provide a better or similar path plan according to the objective function when compared to fields which did not include curved paths. A general legend is provided (figure 39) for this chapter.

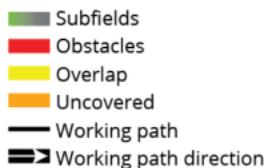


Figure 39: General legend chapter 3.3.

3.3.1 Field from literature

Field Oksanen

Figure 40 shows the digitized field and path plan taken from Oksanen (2007). It contains 4 subfields and one obstacle. The results of the objective function are shown in table 12.

Table 12: comparison result field Oksanen.

	Source	Overlap m ²	Uncovered m ²	Number of turns	Cost ^a	Percentage change ^b
Scenario 1	Oksanen (2007)	145.4	359.6	95	14534960	-3.02
	Script	141.0	193.4	139	14095632	
Scenario 2	Oksanen (2007)	145.4	359.6	95	35957858	-57.05
	Script	208.0	154.4	118	15444710	
Scenario 3	Oksanen (2007)	145.4	359.6	95	9500410	-0.001
	Script	157.4	226.0	95	9500288	

^a Cost is calculated by the objective function (equation 1, section 2.1.4).

^b The change of cost of a path plan, using the cost of path plan from Oksanen (2007) as starting value, compared to the cost of the path plan generated by the script.

In scenario 1 (figure 41) and 2 (figure 42), the algorithm managed to provide a better result according to the objective function. Especially, scenario 2 provided a substantial reduction in uncovered area using a path plan with only 1 subfield. Path plans of scenario 1 and 2 were both similar, only a slight change of path direction is apparent. The cost computed for scenario 3 (figure 43) was close to Oksanen's solution (figure 40), although the general path plan and the number of subfields were different. Oksanen's solution provided two larger subfields, with paths parallel to long edges and two smaller triangular subfields between

them. The larger triangular subfield in the bottom right side only has a slightly different path direction than the adjacent field above. The smaller triangular subfield in the middle of the field also has a slightly different path direction compared to the largest adjacent subfield in the bottom left side. This resulted in extra overlap area and uncovered area.

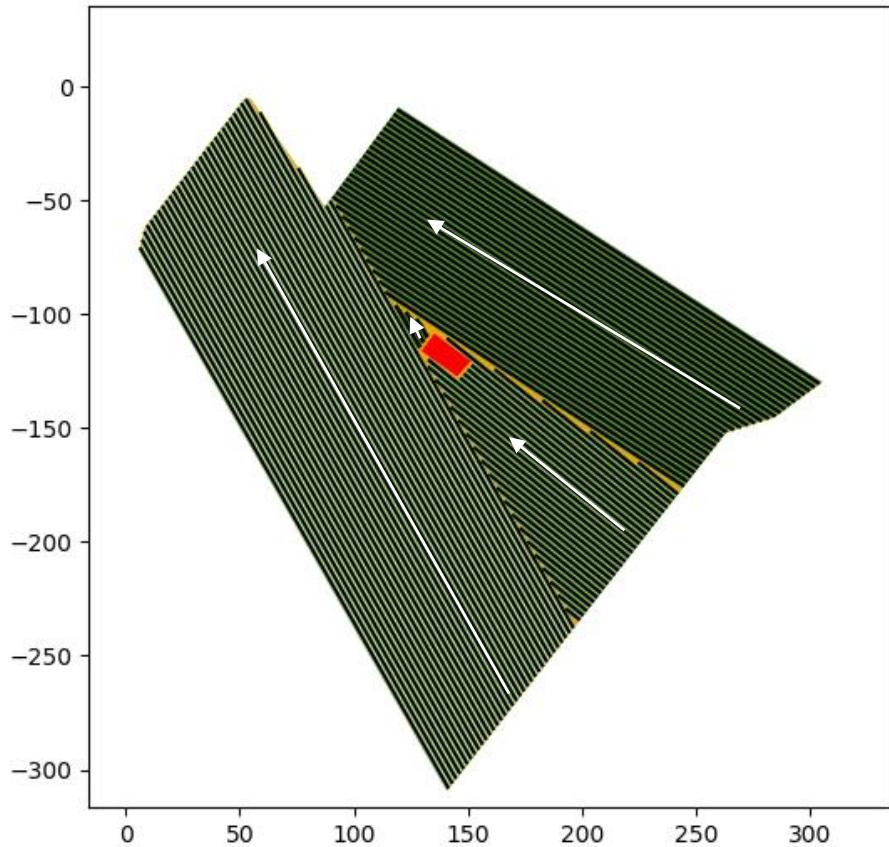


Figure 40: digitized path plan for field Oksanen.

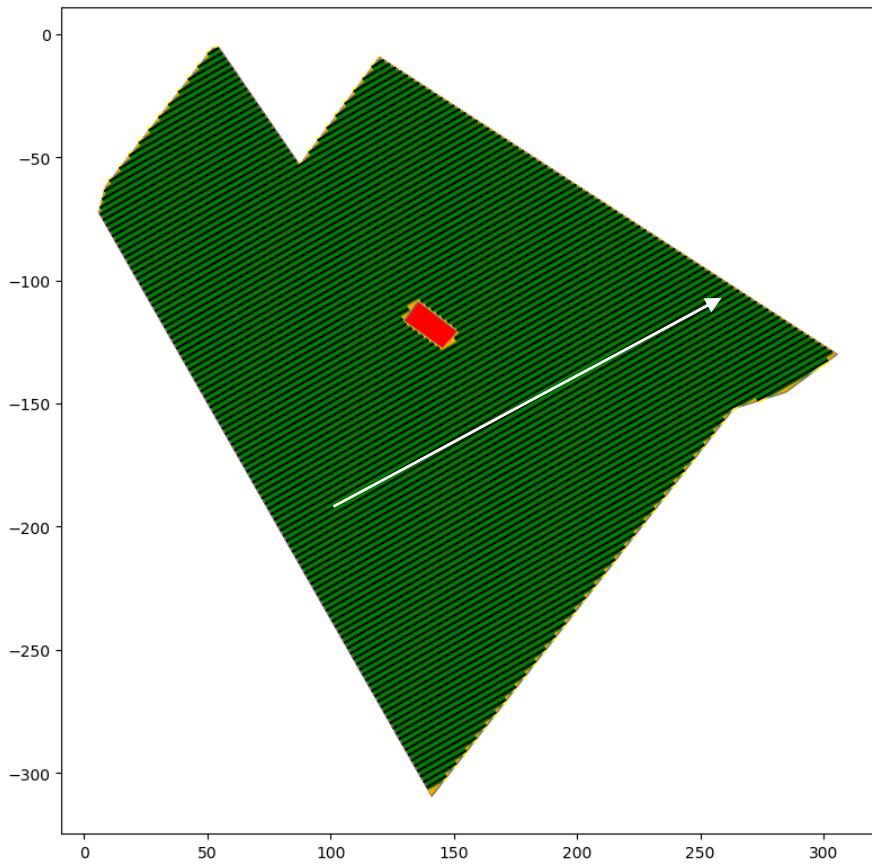


Figure 41: path plan field Oksanen, scenario 1, minimizing overlap area.

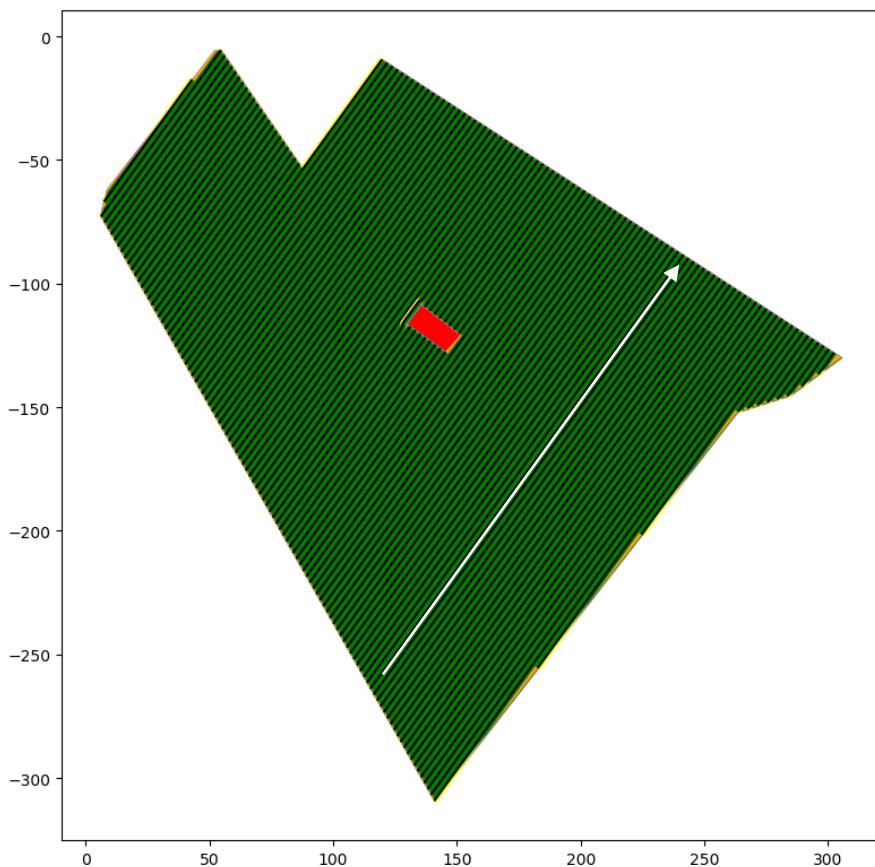


Figure 42: path plan field Oksanen, scenario 2, minimizing uncovered area.

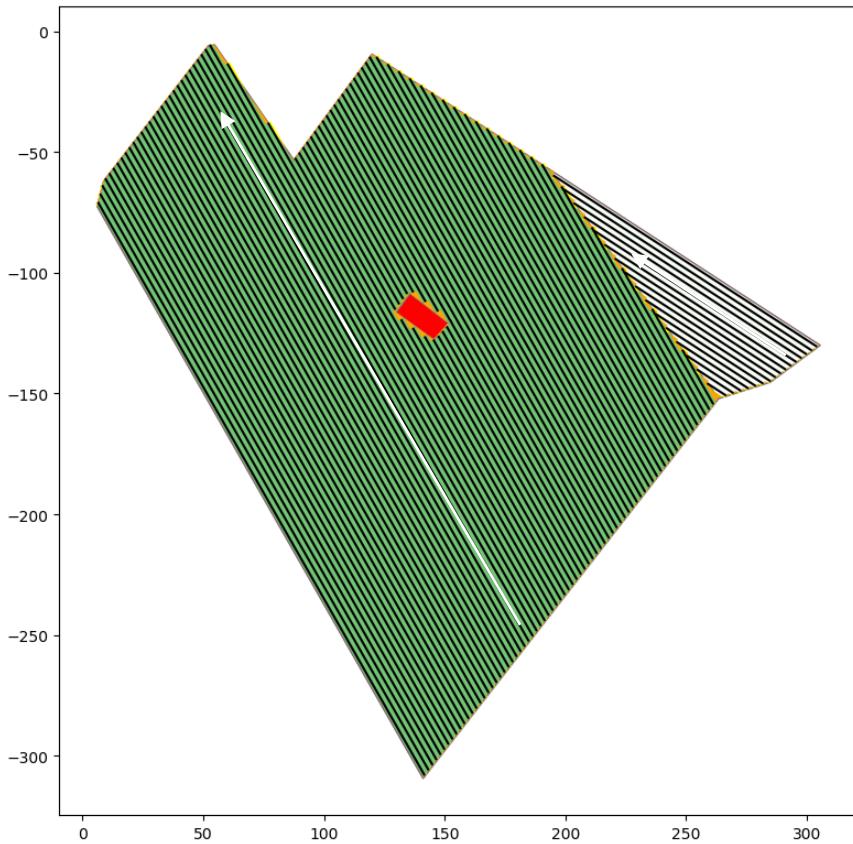


Figure 43: path plan for field Oksanen, scenario 3, minimizing the number of turns.

Field Liu

The second field was from literature focussed on obstacle avoidance. Figure 44 shows the result containing the digitized field and path plan taken from (Liu et al., 2018). It has no subfields since the input geometry is convex. Costs are shown in table 13.

Table 13: comparison result field Liu.

	Source	Overlap m ²	Uncovered m ²	Number of turns	Cost ^a	Percentage change ^b
Scenario 1	Liu et al. (2018)	2127.5	1728.4	39	212749639	-65.91
	Script	725.1	876.7	42	72506610	
Scenario 2	Liu et al. (2018)	2127.5	1728.4	39	172840438	-49.28
	Script	725.1	876.7	42	87669953	
Scenario 3	Liu et al. (2018)	2127.5	1728.4	39	3903817	-2.62
	Script	855.0	850.2	38	3801667	

^a Cost is calculated by the objective function (equation 1, section 2.1.4).

^b The change of cost of a path plan, using the cost of the path plan from Liu et al. (2018) as starting value, compared to the cost of the path plan generated by the script.

In scenario 1 and 2 (figure 45), the algorithm managed to provide a substantial reduction in costs. Both scenarios resulted in the same path plan. The large uncovered area apparent in the solution by Liu et al. (2018) was avoided. Interesting to note is that the path direction in this solution (figure 44) would never be generated by the script as the direction is not parallel to any of the edges. In scenario 3 (figure 46), the number of turns was reduced by one by changing the path direction. Obstacles were not always circumvented completely and number of turns could be further reduced for all path plans computed by the script.

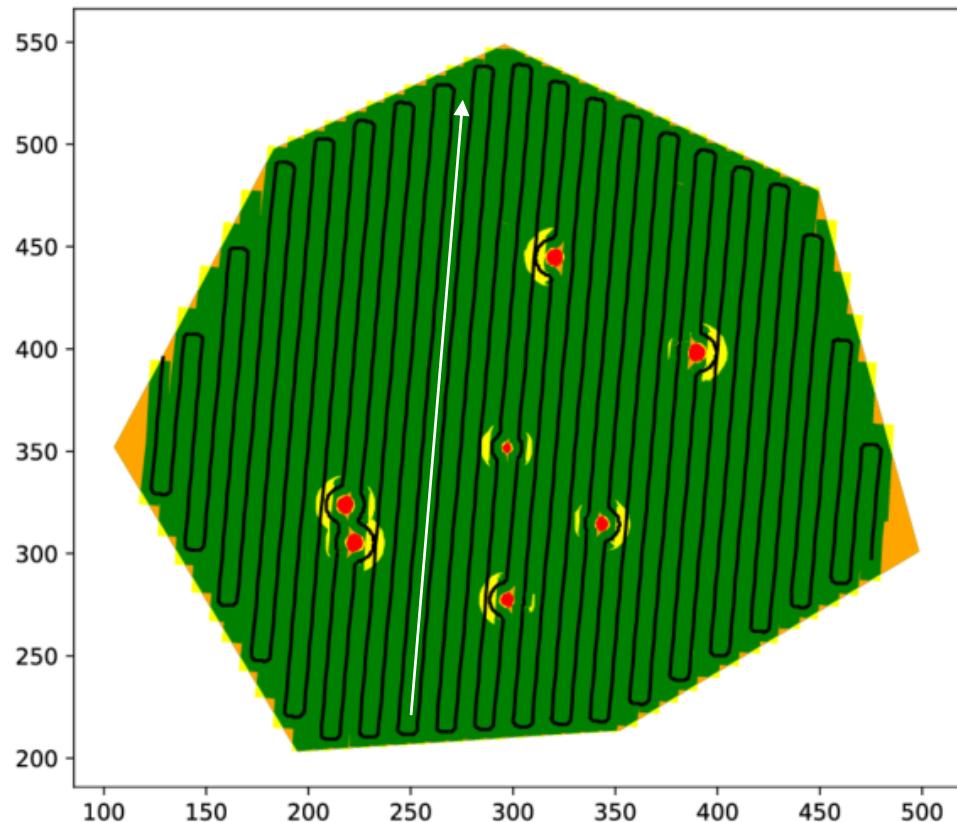


Figure 44: digitized path plan field Liu.

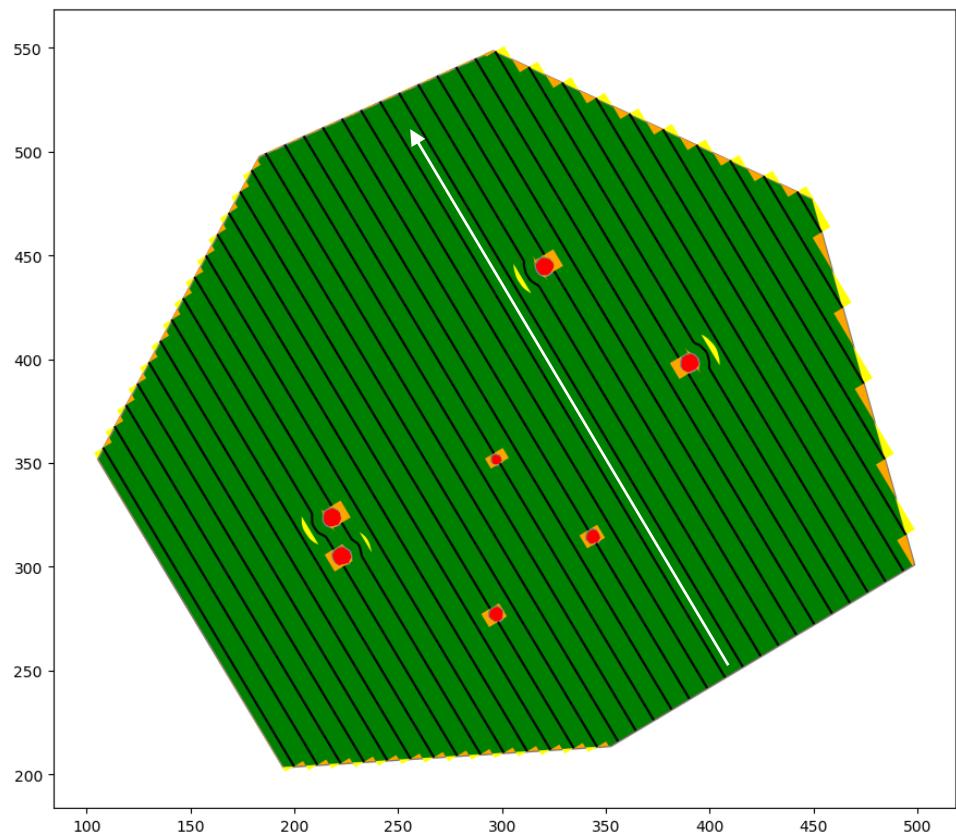


Figure 45: path plan field Liu, scenario 1 and 2, minimizing overlap area and uncovered area (which produced identical results).

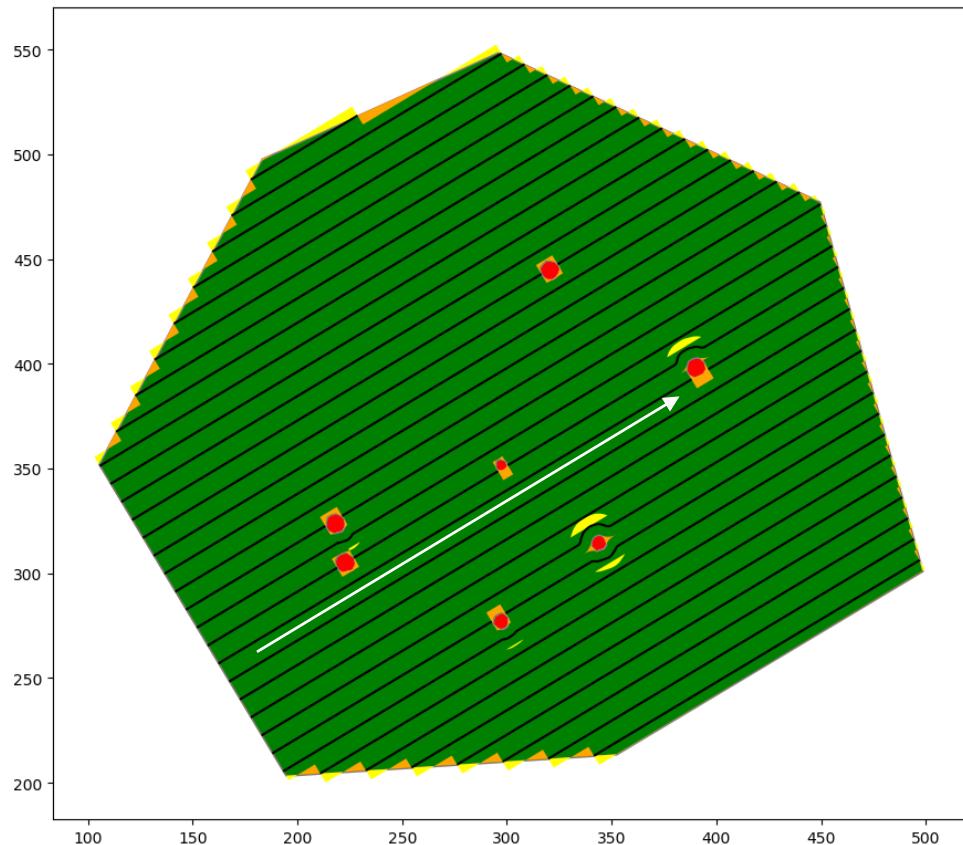


Figure 46: path plan field Liu, scenario 3, minimizing the number of turns.

3.3.2 Farmer's field

Field near Linne

Figure 47 shows the result containing the digitized field and path plan from the field near Linne. It contains 4 subfields. The results of the objective function are shown in table 14.

Table 14: comparison result field near Linne.

	Source	Overlap m²	Uncovered m²	Number of turns	Cost^a	Percentage change^b
Scenario 1	Field Linne	1214.4	1190.6	118	121441723	-49.68
	Script	611.1	927	252	61106349	
Scenario 2	Field Linne	1214.4	1190.6	118	119065493	-46.26
	Script	726.0	639.8	124	63981896	
Scenario 3	Field Linne	1214.4	1190.6	118	11802287	5.07
	Script	726.0	639.8	124	12401242	

^aCost is calculated by the objective function (equation 1, section 2.1.4).

^bThe change of cost of a path plan, using the cost of the path plan from the field near Linne as starting value, compared to the cost of the path plan generated by the script.

In scenarios 1 (figure 48) and 2 (figure 49), the algorithm produced a lower cost according to the objective function. Scenarios 1 and 2 produced a substantial reduction of overlap and uncovered area, respectively, as reflected in the percentage change of the cost. Scenario 1 resulted in a path plan with six subfields. Although limiting overlap, a dramatic increase in number of turns was observed for a minor decrease in overlap compared to the farmer's path plan. Scenario 2 produced the same path plan as scenario 3, resulting in two subfields. The script did not produce a better solution in scenario 3, which is caused by not supporting curved paths. However, the downside of the curved path approach used by the farmer was that it substantially increased the total uncovered and overlapping areas.

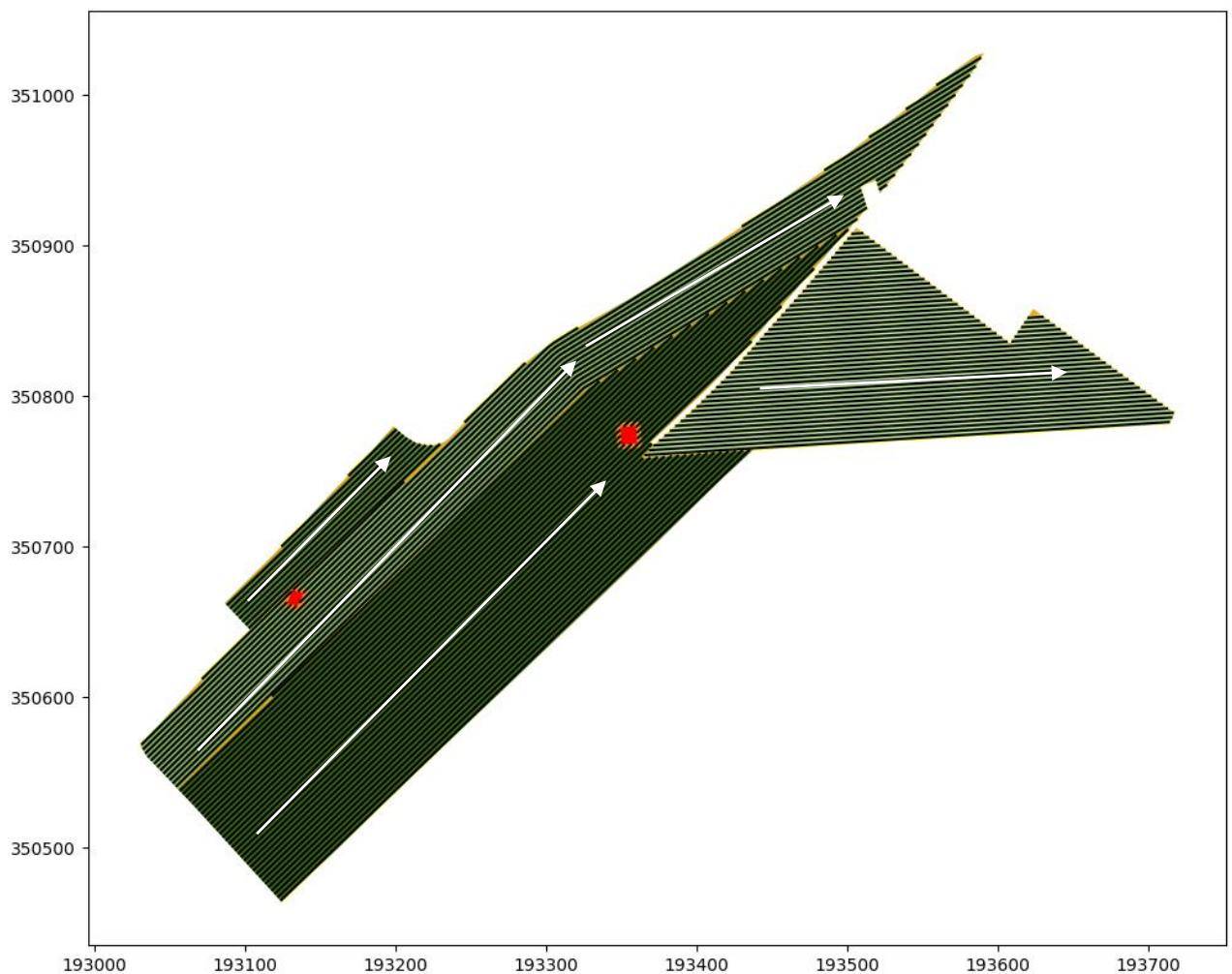


Figure 47: digitized path plan for field near Linne.

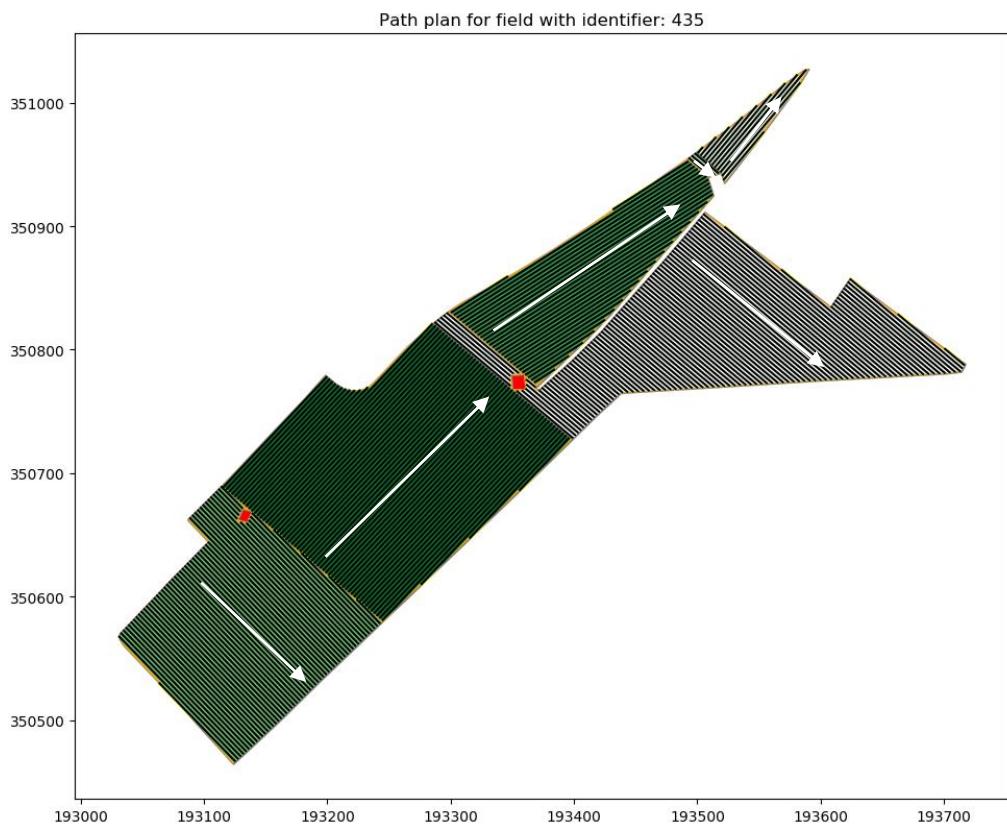


Figure 48: path plan field Linne, scenario 1, minimizing overlap area.



Figure 49: path plan field Linne, scenario 2 and 3, minimizing uncovered area and the number of turns (which produced identical results).

4. Discussion

This chapter discusses the methods and their results per research questions.

4.1 Definition of an optimal path plan

An optimal path plan was defined as the path plan with the minimum cost according to a weighted sum objective function equation 1. The cost was calculated given a set of weights provided by the user and assigned to the three factors, overlap area, uncovered area and number of turns.

It was proven to be successful in providing a path plan with minimal overlap area (figures 24, 28 and 31), uncovered area (figures 26, 29 and 32) or number of turns (figures 27, 30 and 33) for a given field as demonstrated in section 3.1. The cost, however, is heavily dependent on the weights provided by the user. Using the weights, three optimal solutions can be found which minimize each of the three factors respectively. Finding an intermediate solution remains difficult as this means finding a balance between the dimensionless factors. This balance varies greatly per input field geometry. This means that the current implementation of the objective function is only suited for generating three optimal path plans per field, wherein each path plan minimizes one of the three factors. Rather than using a dimensionless number, the cost could be expressed by assigning each of the three factors a cost in euros (de Bruin et al., 2009). Depending on the operation, costs per factor can be determined to provide a more realistic solution for the optimal path plan.

The optimal path plan when minimizing the factors overlap and uncovered area was mainly determined by the track to headland angle (described in section 2.2.1, figure 12). This would often result in similar path plans as observed for scenario 1 and 2 for field 1739 (figure 24 and 26) and 5458 (figure 28 and 29) in section 3.1.1 and 3.1.2. In this thesis, the input field was treated as if the headland was already removed. It was assumed that a headland was placed all around the exterior of field. When calculating the optimal situation, distance covered in the headlands was not included in this thesis. Overlap and uncovered was calculated by simply cutting the path when it touched the exterior as illustrated in figure 20, section 2.2.4. Realistic headland manoeuvres are more complicated and depending on the turning manoeuvres used (Sabelhaus et al., 2013), introduce different overlap, uncovered area and non-working time (Bochtis et al., 2008). In addition, this assumption on headland placement results in unnecessary placed headlands as when a path is parallel to an edge, where headlands are not needed.

This thesis was solely focussed on calculating the optimal path plan. However, a given path plan which is optimal according to the objective function may be sub-optimal depending on the complexity of the agricultural operation. The methodology did not account for the requirements of a path plan with a cooperating machine operation (Jensen et al., 2012), capacitated field operation (Jensen et al., 2015) or servicing units (Spekken et al., 2013). Each operation may introduce restrictions on the placement of paths. Defining an optimal path plan for an agricultural operation is arguably more difficult than for instance an autonomous vacuum cleaner (Yakoubi et al., 2016). The latter are not limited by a large turning radius, soil compaction (Sivarajan et al., 2018) or problems when projecting 2D generated path plan on a 3D environment (Hameed et al., 2016; Han, 2019), to name a few. In defining an optimal

path plan in this thesis, wet spots which may introduce additional soil compaction or undesirable path directions as a result of elevation differences were not taken into account.

When calculating the optimal path plan for minimizing the number of turns in a field, full coverage on non-convex fields was observed to result in small corners with short paths which are costly to operate in comparison to other paths (Spekken et al., 2012). However, this was not accounted for when calculating the cost using the objective function. Figures 26, 30, and 33 from section 3.1 provided examples of small, triangular subfields with short paths. Each path, no matter the length, were assigned the same cost. In a more realistic scenario, determining the optimal situation of a path plan involves calculating its corresponding routing plan as stated above. Rather than expressing the optimal situation in minimal number of turns, the optimal situation should include the minimal non-working time (Spekken et al., 2013) or non-working distance (Bochtis et al., 2008) in headlands.

4.2 Path coverage generating algorithm

To calculate the optimal path plan, an extension to the Minimum Sum of Widths (Yu, 2015) method (trapezoidal decomposition using a sweep line) was proposed. It provided fine-tuning parameters regarding field decomposition and subfield merging. An additional module was added for obstacle avoidance based on Dubins path (Dubins, 1957). The coverage path script was successfully implemented in combination with the objective function.

4.2.1 Field decomposition

Field decomposition was executed using a sweep line algorithm (figure 14, section 2.2.2). The direction is determined by the edge angles of the input geometry relative to the x-axis. For each unique angle, the sweep line was used perpendicular to that direction. This was proven to result in the lowest number of turns (Huang, 2001), but may not necessarily be an optimal solution when considering more factors such as overlap and uncovered area. Moreover, when considering more complex agricultural situations which include cooperating machine, or capacitated vehicle operations, more sweep line directions will have to be executed as the optimal situation can no longer be expressed in minimal number of turns. Sweep line decomposition angle optimization was demonstrated in Oksanen (2007) and optimized using heuristics.

A decomposition threshold was introduced in section 2.2.2 in an effort to reduce the unnecessary creation of subfields. Whereas the MSW method would split a field at vertices when the interior angle exceeded 180 degrees, the decomposition threshold allowed the user to control this value (figure 13 section 2.2.2). However, optimal settings are largely depended on the digitization of the input field. Fields provided by the BRP varied in level of detail. Seemingly straight edges were commonly composed of multiple, smaller edges which would exceed the interior angle of 180 degrees. If an input geometry contained many vertices in relative small intervals, many narrow subfields were created which would all share the same width and therefore the same optimal path direction. To some extent, the introduced decomposition threshold helped in reducing these scenarios. However, the decomposition threshold did not work when encountering vertices which connected very

short edges. These could have very large interior angles, whilst the resulting subfields would be very small. An example of this situation is shown in figure 35(a, b and c). The small triangular subfield at the bottom is a result of a very short edge connected to a larger edge (vertex at the top of the triangle). In the current implementation, these smaller subfields are handled separately in the merging process (section 2.2.3) which tries to merge it with adjacent fields regardless of optimal track direction. An improvement to the decomposition threshold would be to only consider vertices for decomposition if they connected larger edges (relatively compared to other edges of the field). This would speed up and simplify the merging process whilst also improving the feasibility of the resulting subfields. Simplifying field geometry using algorithms that preserved topology were observed to help in reducing the amount of seemingly unnecessary vertices during development. However, this was not implemented as this would be contradictory with the idea of precision agriculture and precise path planning if the input geometry was edited in order to provide optimal coverage.

A merge threshold was introduced in section 2.2.3 in an effort to reduce the unnecessary creation of subfields. The MSW method only merged adjacent subfields when the width was exactly equal to each other. The merge threshold provided a way of merging adjacent field when the direction was almost equal to each other, depending on user preference. The approach could be improved by adding additional conditions for merging adjacent subfields, rather than just the optimal driving direction. For instance, it was observed that smaller subfields did not always provide a decrease in number of turns needed for complete coverage, whilst it did introduce in-field turns and therefore extra overlap or uncovered area. Field 5458, scenario 3, minimizing number of turns (figure 30) from section 3.1.2 provides an example. A small triangular shaped subfield is formed at the bottom which did not help reducing the number of turns. If it would have the same path direction as the adjacent rectangular field, this would not make a difference. These subfields did, however, have a different optimal path direction than the adjacent fields and would therefore not be merged. An extra condition would be necessary to avoid these situations. This could be done by testing if removing a subfield would actually increase the number of subfields. This could improve the feasibility of the output path plan as it is unlikely that a farmer will proceed to work a small subfield, which introduces an extra in-field turns, whilst it does not reduce the number of turns for the overall path plan. The need for global field optimization rather than local subfield optimization was further emphasized in Hofstee et al. (2009).

Galceran et al. (2013) provides an overview on coverage path planning in robotics. It was observed that spatial decomposition is key in providing complete coverage. Path planning in agriculture was no exception to this observation. Exact cellular decomposition techniques were observed to be most popular, namely trapezoidal decomposition (Hameed et al., 2013; Hofstee et al., 2009; Oksanen et al., 2009; Yu, 2015). Each study added various restrictions on the decomposition or merging process. In the current thesis, a similar approach was used by enhancing trapezoidal decomposition with various restrictions on merging subfields (figure 35, section 3.2.2) and a decomposition threshold value which determines when field is split at a vertex (figure 36, section 3.2.2). Literature tended to approach the coverage problem with varying levels of field geometry detail. In Yu (2015), very regular field geometries were presented with low vertex density which may not be applicable in a real-life situation. In contrast, Oksanen et al. (2009) incorporated more complex, realistic fields. The simplification

of field geometries is contradictory with the idea of precision agriculture in which the complete field surface area, however complex, must be worked. It was demonstrated in Oksanen (2007) that exact cellular decomposition provided increasingly unsatisfactory results in relation to increasing field geometry complexity. This was observed during development when testing the script on large, curved fields of the BRP with high vertex density. Examples are shown in appendix A, field 4162, which only has one distinct path direction and 5220, which contains ten subfields with different path directions.

The developed solution did not support the creation of curved paths which is a downside considering that fields with straight edges do not really exist. Moreover, this means that application is limited to straight, row based crops. The advantage of curved paths was demonstrated in Oksanen et al. (2009) using a predictive recursive online approach and in Jin et al. (2011) using a seed curve. Long and curved paths will provide an advantage if there are many short edges. In other words, if there is no clear longest edge, the current implementation will not provide satisfactory results.

4.2.2 Obstacle avoidance

When a small swath width is used in combination with a large minimum turning radius, the obstacle avoidance path may not be feasible. This is due to the obstacle avoidance method being specifically designed for small, circular obstacle avoidance only (Liu et al., 2018).

Theoretically, this can occur in any case where $\text{swath width} < \text{minimum turning radius}$ with an increasing difference leading to increasing chance of a problematic situation. In a niche case where the obstacle area which would collide with the agricultural machinery would be narrow and steep (e.g. a narrow, stretched triangle), the avoidance path may not ensure the safety of the agricultural equipment. As only the start and end of the avoidance path are calculated using the minimum turning radius using Dubins path, the middle part may not be possible given the requirements of an agricultural vehicle. The middle part is calculated only by buffering it with the swath width, which will introduce an impossible path in case of a thin, stretched obstacle as the resulting curvature will be too steep. However, such cases were observed to be a rare occurrence in the fields used of the BRP and were not observed in this thesis.

The current implementation does not account for changes in velocity which is assumed to be fixed. Accounting for changes in velocity is also known as the "Dubins Car" approach (Patsko et al., 2018). Moreover, the time needed for turning the steering wheel is neither accounted for. Dubins path is further limited to a 2D environment which was observed to be a downside in comparison to a 3D path planning environment as stated above. The complexity of generating smooth curves in a 3D environment considerably increases as a result of velocity and acceleration interaction, in addition to computational efficiency challenges (Marino et al., 2016; Zhang et al., 2018).

4.3 Path plan comparison

A large improvement in the scenarios 1 and 2, which minimized overlap area and uncovered area respectively. This was shown in figures 41, 42 in section 3.3.1, figure 45 in section 3.3.2 and figures 48, 49 in section 3.3.3. In practice and in literature, number of turns was observed to be the most costly element of a field operation (Bochtis et al., 2008; Oksanen et al., 2009; Yu, 2015). This explains the large differences in cost in these scenarios, as overlap area and uncovered area were not included in determining an optimal path plan for the comparison fields in section 3.2.1 and 3.2.2 (Liu et al., 2018; Oksanen, 2007).

The large uncovered area of the path plan in field Liu (figure 44, section 3.2.2) in comparison to the script can be explained by the two larger areas at the western and eastern edge of the field. These areas are 404.7 m^2 and 510.3 m^2 respectively. Unsurprisingly, this resulted in a 49.28% decrease in cost of the path plan (table 13) from the script of scenario 2 (figure 45), making it favourable over the path plan from field Liu. In scenario 1 (figure 45), the path plan of field Liu has a considerable increase in overlap area due to the many obstacle avoidance manoeuvres. This resulted in a 65.91% decrease in cost in the path plan from the script. Although the field had no clear width, by changing the path direction, a 2.62% decrease in cost was achieved in scenario 3 (figure 46) in the path plan from the script.

Important to note is that in the solution provided by the script, the avoidance part of the working path would never be closer to the adjacent paths than the swath width divided by 2. This was due to an error in the implementation which trimmed the path too early when the path intersected an obstacle, instead of circumventing the obstacle. The effect of this was especially visible in section 3.3.2, figures 45 and 46. The swath width was set to 10 meters, which equalled 5 meters of actual avoidance space in the implemented method. In field Liu, the path will try to circumvent an obstacle if it does not have to cross another path, thereby giving it another 5 meters of avoidance space. This is detailed in figure 50. Situation figure 50(a) represents the solution by the developed algorithm, where the bottom path which collides with the obstacle is cut short. This is contrast with situation figure 50(b) in field Liu where the driving path to the left of the obstacle, which would otherwise collide with the obstacle, is edited in order to circumvent it. This results in more obstacle avoidance options, consequently lowering the number of turns needed for coverage whilst increasing the total amount of overlap area.

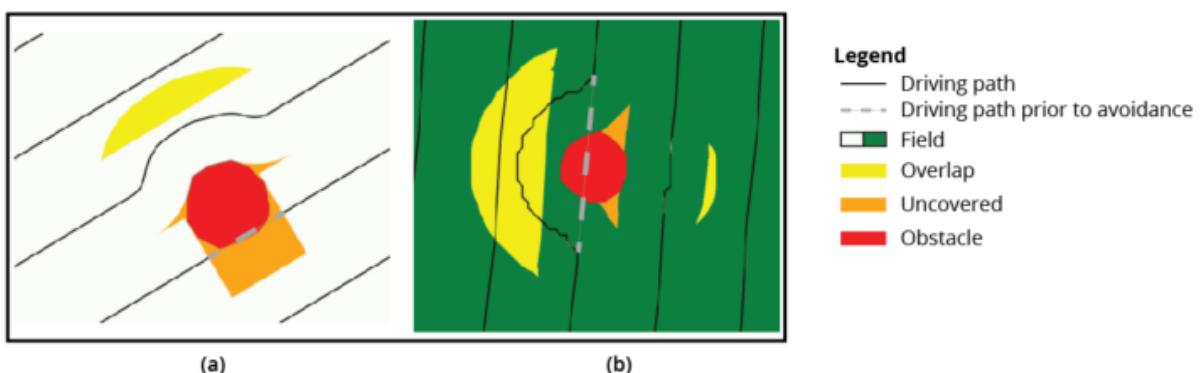


Figure 50: obstacle avoidance of the algorithm (a) compared to field Liu obstacle avoidance (b).

The process of digitizing path plans proved to be challenging. Field Linne was difficult to digitize because of manually tracing the curved paths, which had to be correctly aligned to prevent overlap, was not a build-in feature in GIS software or Python. A more scalable path plan digitization method is needed for comparison on a larger scale. A deep learning approach was proven to be effective in a comparable scenario for digitizing road centrelines from aerial imagery (Liu et al., 2019) and for geographically-regularized semantic segmentation of aerial imagery (Volpi et al., 2018).

Both fields from literature in section 2.3.2 were provided in an unprojected vector format called Encapsulated PostScript which is commonly used in graphic software such as Adobe Illustrator. This introduced difficulties such as missing line segments which had to be manually corrected in ArcGIS Pro and the Python working environment. This did influence the results in section 3.3.1 and 3.3.2 and should therefore be kept in mind when interpreting the costs according to the objective function.

5. Conclusion and recommendations

This thesis proposed a method to determine an optimal path plan on an agricultural field with obstacles, allowing user prioritization of three factors to minimize either overlap area, uncovered area or number of turns. An obstacle avoidance method using Dubins path was demonstrated to be successful in reducing the number of turns on fields with obstacles. Although the method is restricted to a 2D environment with straight and circular movement only, it was observed to provide feasible path plans for almost any input geometry, incorporating minimum turning radius, factor prioritization, and parameter settings for tuning spatial decomposition and merging parameters. Below, brief answers are provided to the research questions posed in chapter 1.

1. What defines an optimal path plan in fields with obstacles?

An optimal path plan is defined as the path plan which results in minimal overlap area, uncovered area and number of turns. The total cost of a path plan is determined using a weighted sum containing the three factors and a set of weights in a dimensionless sum.

2. What algorithm can be used to find the optimal path plan on a field with obstacles?

A sweepline algorithm was used for trapezoidal decomposition of the input field in combination with a merging algorithm. A decomposition and merge threshold were used to fine-tune these algorithms. Obstacles avoidance manoeuvres were calculated using Dubins path when the obstacle intersection width of the path was less than the swath width.

3. How does the path plan generated by the developed algorithm compare to path plans observed in practice and in literature?

Path plans were compared in three different scenarios, each limiting one of the three factors. The script managed to provide a better or similar path plan according to the objective function when compared to fields which did not include curved paths.

Several recommendations for further research are presented. An optimal path plan was calculated by using a dimensionless sum of the three factors. Alternatively, an optimal path plan can be expressed economically in a cost scenario in which overlap area, uncovered area and number of turns are each assigned a cost in euros (de Bruin et al., 2009). Depending on the agricultural operation, different costs can be assigned to provide a realistic estimation of economic efficiency of a path plan (Bochtis et al., 2019b).

Curved paths should be further investigated as they were observed to provide a better path plan in test field Linne and for more complex fields in literature compared to the proposed method in this thesis. Additionally, curved paths would provide increased flexibility in obstacle avoidance, soil compaction minimization or handling possible terrain elevation differences. Successful application was demonstrated in a 2D environment Oksanen (2007) and 3D environment Jin et al. (2011) which provided promising results when compared to trapezoidal decomposition with straight paths.

In order to provide a more scalable comparison method, further research could be conducted on digitizing path plans observed from satellite imagery. A deep learning approach was proven to be effective in a comparable scenario for digitizing road centrelines from aerial imagery (Liu et al., 2019) and for geographically-regularized semantic segmentation of aerial imagery (Volpi et al., 2018).

References

- Arkin, E. M., Fekete, S. P., & Mitchell, J. S. B. (2000). Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17(1), 25-50. doi:[https://doi.org/10.1016/S0925-7721\(00\)00015-8](https://doi.org/10.1016/S0925-7721(00)00015-8)
- Assandri, G., Bogliani, G., Pedrini, P., & Brambilla, M. (2018). Beautiful agricultural landscapes promote cultural ecosystem services and biodiversity conservation. *Agriculture, Ecosystems & Environment*, 256, 200-210. doi:<https://doi.org/10.1016/j.agee.2018.01.012>
- Backman, J., Piirainen, P., & Oksanen, T. (2015). Smooth turning path generation for agricultural vehicles in headlands. *Biosystems Engineering*, 139, 76-86. doi:<https://doi.org/10.1016/j.biosystemseng.2015.08.005>
- Berg, M. d., Cheong, O., Kreveld, M. v., & Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*: Springer-Verlag TELOS.
- Bochtis, D., Sørensen, C. A. G., & Kateris, D. (2019a). 3 - Effectiveness and Efficiency of Agricultural Machinery. In D. Bochtis, C. A. G. Sørensen, & D. Kateris (Eds.), *Operations Management in Agriculture* (pp. 47-78): Academic Press.
- Bochtis, D., Sørensen, C. A. G., & Kateris, D. (2019b). 4 - Cost of Using Agricultural Machinery. In D. Bochtis, C. A. G. Sørensen, & D. Kateris (Eds.), *Operations Management in Agriculture* (pp. 79-115): Academic Press.
- Bochtis, D. D., & Sørensen, C. G. (2009). The vehicle routing problem in field logistics part I. *Biosystems Engineering*, 104(4), 447-457. doi:<https://doi.org/10.1016/j.biosystemseng.2009.09.003>
- Bochtis, D. D., Sørensen, C. G. C., & Busato, P. (2014). Advances in agricultural machinery management: A review. *Biosystems Engineering*, 126, 69-81. doi:<https://doi.org/10.1016/j.biosystemseng.2014.07.012>
- Bochtis, D. D., & Vougioukas, S. G. (2008). Minimising the non-working distance travelled by machines operating in a headland field pattern. *Biosystems Engineering*, 101(1), 1-12. doi:<https://doi.org/10.1016/j.biosystemseng.2008.06.008>
- Borin, M., Passoni, M., Thiene, M., & Tempesta, T. (2010). Multiple functions of buffer strips in farming areas. *European Journal of Agronomy*, 32(1), 103-111. doi:<https://doi.org/10.1016/j.eja.2009.05.003>
- Bucksteeg, M. (2019). Modelling the impact of geographical diversification of wind turbines on the required firm capacity in Germany. *Applied Energy*, 235, 1476-1491. doi:<https://doi.org/10.1016/j.apenergy.2018.11.031>
- Cao, Z. L., Huang, Y., & Hall, E. L. (1988). Region filling operations with random obstacle avoidance for mobile robots. *Journal of Robotic Systems*, 5(2), 87-102. doi:10.1002/rob.4620050202
- de Bruin, S., Lerink, P., J. La Riviere, I., & Vanmeulebrouk, B. (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:<https://doi.org/10.1016/j.biosystemseng.2013.07.009>
- de Bruin, S., Lerink, P., Klompe, A., van der Wal, T., & Heijting, S. (2009). Spatial optimisation of cropped swaths and field margins using GIS. *Computers and Electronics in Agriculture*, 68(2), 185-190. doi:<https://doi.org/10.1016/j.compag.2009.06.001>
- 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. doi:10.2307/2372560
- Duurzame energiewinning. (2017). Retrieved from <https://www.windparknoordoostpolder.nl/>
- Felder, F. A., Andrews, C. J., & Hulkower, S. D. (2011). Chapter 2 - Which Energy Future? In F. P. Sioshansi (Ed.), *Energy, Sustainability and the Environment* (pp. 31-61). Boston: Butterworth-Heinemann.

- Foltête, J.-C. (2018). A parcel-based graph to match connectivity analysis with field action in agricultural landscapes: Is node removal a reliable method? *Landscape and Urban Planning*, 178, 32-42. doi:<https://doi.org/10.1016/j.landurbplan.2018.05.016>
- Fortune, S. (1987). A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1), 153. doi:10.1007/bf01840357
- Fountas, S., Wulfsohn, D., Blackmore, B. S., Jacobsen, H. L., & Pedersen, S. M. (2006). A model of decision-making and information flows for information-intensive agriculture. *Agricultural Systems*, 87(2), 192-210. doi:<https://doi.org/10.1016/j.agsy.2004.12.003>
- Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12), 1258-1276. doi:<https://doi.org/10.1016/j.robot.2013.09.004>
- Gasso, V., Oudshoorn, F. W., Sørensen, C. A. G., & Pedersen, H. H. (2014). An environmental life cycle assessment of controlled traffic farming. *Journal of Cleaner Production*, 73, 175-182. doi:<https://doi.org/10.1016/j.jclepro.2013.10.044>
- Gebbers, R., & Adamchuk, V. I. (2010). Precision Agriculture and Food Security. *Science*, 327(5967), 828. doi:10.1126/science.1183899
- Geopandas. (2018). Geopandas 0.4.0. Retrieved from <http://geopandas.org/>
- Gillies, S. (2019a). Fiona 1.8.6 documentation. Retrieved from <https://fiona.readthedocs.io/en/latest/index.html>
- Gillies, S. (2019b). Shapely 1.6.4 documentation. Retrieved from <https://shapely.readthedocs.io/en/latest/index.html#>
- Google Earth. (2019). Google Earth. Retrieved from <https://earth.google.com/web/>
- Haenke, S., Scheid, B., Schaefer, M., Tscharntke, T., & Thies, C. (2009). Increasing syrphid fly diversity and density in sown flower strips within simple vs. complex landscapes. *Journal of Applied Ecology*, 46(5), 1106-1114. doi:doi:10.1111/j.1365-2664.2009.01685.x
- Hameed, I., Bochtis, D., & Sørensen, C. (2011). *Driving Angle and Track Sequence Optimization for Operational Path Planning Using Genetic Algorithms* (Vol. 27).
- Hameed, I. A., Bochtis, D., & Sørensen, C. A. (2013). An Optimized Field Coverage Planning Approach for Navigation of Agricultural Robots in Fields Involving Obstacle Areas. *International Journal of Advanced Robotic Systems*, 10(5), 231. doi:10.5772/56248
- Hameed, I. A., la Cour-Harbo, A., & Osen, O. L. (2016). Side-to-side 3D coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. *Robotics and Autonomous Systems*, 76, 36-45. doi:<https://doi.org/10.1016/j.robot.2015.11.009>
- Han, J. (2019). An efficient approach to 3D path planning. *Information Sciences*, 478, 318-330. doi:<https://doi.org/10.1016/j.ins.2018.11.045>
- Hofstee, J. W., Spätjens, L. E. E. M., & Ijken, H. (2009). *Optimal path planning for field operations*. Paper presented at the Joint International Agricultural Conference, (JIAC2009), Wageningen, The Netherlands, 6 - 8 July, 2009, Wageningen.
- Huang, W. H. (2001). *Optimal line-sweep-based decompositions for coverage algorithms*. Paper presented at the Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on.
- Iglesias, A., Quiroga, S., Moneo, M., & Garrote, L. (2012). From climate change impacts to the development of adaptation strategies: Challenges for agriculture in Europe. *Climatic Change*, 112(1), 143-168. doi:10.1007/s10584-011-0344-x
- Jensen, A., Riviere, I. J. I., Bruin, S. d., Zhou, K., Jørgensen, M. S., Pedersen, H., & Sørensen, C. G. (2018). *An online DSS for optimisation of traffic in fields with controlled traffic farming (CTF)*. Paper presented at the CIGR 2018: XIX. World Congress of CIGR (Commission Internationale du Génie Rural) Program & Abstracts' Book.
- Jensen, M. A. F., Bochtis, D., Sørensen, C. G., Blas, M. R., & Lykkegaard, K. L. (2012). In-field and inter-field path planning for agricultural transport units. *Computers & Industrial Engineering*, 63(4), 1054-1061. doi:<https://doi.org/10.1016/j.cie.2012.07.004>

- Jensen, M. F., Nørremark, M., Busato, P., Sørensen, C. G., & Bochtis, D. (2015). Coverage planning for capacitated field operations, Part I: Task decomposition. *Biosystems Engineering*, 139, 136-148. doi:<https://doi.org/10.1016/j.biosystemeng.2015.07.003>
- Jiang, Y., Hao, K., Cai, X., & Ding, Y. (2018). An improved reinforcement-immune algorithm for agricultural resource allocation optimization. *Journal of Computational Science*, 27, 320-328. doi:<https://doi.org/10.1016/j.jocs.2018.06.011>
- Jin, J., & Tang, L. (2011). *Coverage Path Planning on Three-Dimensional Terrain for Arable Farming* (Vol. 28).
- Joon Seop, O., Yoon Ho, C., Jin Bae, P., & Zheng, Y. F. (2004). Complete coverage navigation of cleaning robots using triangular-cell-based map. *IEEE Transactions on Industrial Electronics*, 51(3), 718-726. doi:10.1109/TIE.2004.825197
- Keicher, R., & Seufert, H. (2000). Automatic guidance for agricultural vehicles in Europe. *Computers and Electronics in Agriculture*, 25(1), 169-194. doi:[https://doi.org/10.1016/S0168-1699\(99\)00062-9](https://doi.org/10.1016/S0168-1699(99)00062-9)
- Latombe, J.-C. (1991). Exact Cell Decomposition. In *Robot Motion Planning* (pp. 200-247). Boston, MA: Springer US.
- Lin, H.-C., & Hülsbergen, K.-J. (2017). A new method for analyzing agricultural land-use efficiency, and its application in organic and conventional farming systems in southern Germany. *European Journal of Agronomy*, 83, 15-27. doi:<https://doi.org/10.1016/j.eja.2016.11.003>
- Liu, C., Zhao, X., Du, Y., Cao, C., Zhu, Z., & Mao, E. (2018). Research on static path planning method of small obstacles for automatic navigation of agricultural machinery. *IFAC-PapersOnLine*, 51(17), 673-677. doi:<https://doi.org/10.1016/j.ifacol.2018.08.119>
- Liu, R., Miao, Q., Song, J., Quan, Y., Li, Y., Xu, P., & Dai, J. (2019). Multiscale road centerlines extraction from high-resolution aerial imagery. *Neurocomputing*, 329, 384-396. doi:<https://doi.org/10.1016/j.neucom.2018.10.036>
- Marino, H., Salaris, P., & Pallottino, L. (2016). Controllability analysis of a pair of 3D Dubins vehicles in formation. *Robotics and Autonomous Systems*, 83, 94-105. doi:<https://doi.org/10.1016/j.robot.2016.05.015>
- Nash, E., Korduan, P., & Bill, R. (2009). Applications of open geospatial web services in precision agriculture: a review. *Precision Agriculture*, 10(6), 546. doi:10.1007/s11119-009-9134-0
- Oksanen, T. (2007). *Path Planning Algorithms for Agricultural Field Machines*.
- Oksanen, T. (2013). Estimating operational efficiency of field work based on field shape. *IFAC Proceedings Volumes*, 46(18), 202-206. doi:<https://doi.org/10.3182/20130828-2-SF-3019.00065>
- Oksanen, T., & Visala, A. (2009). Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8), 651-668. doi:doi:10.1002/rob.20300
- Palacin, J., Palleja, T., Valganon, I., Pernia, R., & Roca, J. (2005, 18-22 April 2005). *Measuring Coverage Performances of a Floor Cleaning Mobile Robot Using a Vision System*. Paper presented at the Proceedings of the 2005 IEEE International Conference on Robotics and Automation.
- Parafatos, D. S., Hübner, R., & Griepentrog, H. W. (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:<https://doi.org/10.1016/j.compag.2018.07.035>
- Patsko, V. S., & Fedotov, A. A. (2018). Attainability Set at Instant for One-Side Turning Dubins Car. *IFAC-PapersOnLine*, 51(32), 201-206. doi:<https://doi.org/10.1016/j.ifacol.2018.11.381>
- Pierce, F. J., & Nowak, P. (1999). Aspects of Precision Agriculture. In D. L. Sparks (Ed.), *Advances in Agronomy* (Vol. 67, pp. 1-85): Academic Press.
- Righter, R. W. (1996). *Wind energy in America: A history*: University of Oklahoma Press.

- Sabelhaus, D., Röben, F., Meyer zu Hellen, L. P., & Schulze Lammers, P. (2013). Using continuous-curvature paths to generate feasible headland turn manoeuvres. *Biosystems Engineering*, 116(4), 399-409. doi:<https://doi.org/10.1016/j.biosystemseng.2013.08.012>
- Scherer, L. A., Verburg, P. H., & Schulp, C. J. E. (2018). Opportunities for sustainable intensification in European agriculture. *Global Environmental Change*, 48, 43-55. doi:<https://doi.org/10.1016/j.gloenvcha.2017.11.009>
- Shiu, B.-M., & Lin, C.-L. (2008). *Design of an autonomous lawn mower with optimal route planning*.
- Sivarajan, S., Maharlooei, M., Bajwa, S. G., & Nowatzki, J. (2018). Impact of soil compaction due to wheel traffic on corn and soybean growth, development and yield. *Soil and Tillage Research*, 175, 234-243. doi:<https://doi.org/10.1016/j.still.2017.09.001>
- Sørensen, C. G., & Bochtis, D. D. (2010). Conceptual model of fleet management in agriculture. *Biosystems Engineering*, 105(1), 41-50. doi:<https://doi.org/10.1016/j.biosystemseng.2009.09.009>
- Spekken, M., & de Bruin, S. (2013). Optimized routing on agricultural fields by minimizing maneuvering and servicing time. *Precision Agriculture*, 14(2), 224-244. doi:10.1007/s11119-012-9290-5
- Spekken, M., & Molin, J. (2012). *OPTIMIZING PATH PLANNING BY AVOIDING SHORT CORNER TRACKS*.
- Stilma, E. S. C., Smit, A. B., Geerling-Eiff, F. A., Struik, P. C., Vosman, B., & Korevaar, H. (2009). Perception of biodiversity in arable production systems in the Netherlands. *NJAS - Wageningen Journal of Life Sciences*, 56(4), 391-404. doi:[https://doi.org/10.1016/S1573-5214\(09\)80006-7](https://doi.org/10.1016/S1573-5214(09)80006-7)
- Taylor, J. H. (1983). Benefits of permanent traffic lanes in a controlled traffic crop production system. *Soil and Tillage Research*, 3(4), 385-395. doi:[https://doi.org/10.1016/0167-1987\(83\)90040-5](https://doi.org/10.1016/0167-1987(83)90040-5)
- Tilman, D., Balzer, C., Hill, J., & Befort, B. L. (2011). Global food demand and the sustainable intensification of agriculture. *Proceedings of the National Academy of Sciences*, 108(50), 20260-20264. doi:10.1073/pnas.1116437108
- Toussaint, G. (1983). *Solving Geometric Problems with the Rotating Calipers* (Vol. 83).
- Tullberg, J., Antille, D. L., Bluett, C., Eberhard, J., & Scheer, C. (2018). Controlled traffic farming effects on soil emissions of nitrous oxide and methane. *Soil and Tillage Research*, 176, 18-25. doi:<https://doi.org/10.1016/j.still.2017.09.014>
- Tullberg, J. N., Yule, D. F., & McGarry, D. (2007). Controlled traffic farming—From research to adoption in Australia. *Soil and Tillage Research*, 97(2), 272-281. doi:<https://doi.org/10.1016/j.still.2007.09.007>
- van Ittersum, M. K., Cassman, K. G., Grassini, P., Wolf, J., Tittonell, P., & Hochman, Z. (2013). Yield gap analysis with local to global relevance—A review. *Field Crops Research*, 143, 4-17. doi:<https://doi.org/10.1016/j.fcr.2012.09.009>
- Velten, S., Schaal, T., Leventon, J., Hanspach, J., Fischer, J., & Newig, J. (2018). Rethinking biodiversity governance in European agricultural landscapes: Acceptability of alternative governance scenarios. *Land Use Policy*, 77, 84-93. doi:<https://doi.org/10.1016/j.landusepol.2018.05.032>
- Volpi, M., & Tuia, D. (2018). Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144, 48-60. doi:<https://doi.org/10.1016/j.isprsjprs.2018.06.007>
- von Haaren, C., & Reich, M. (2006). The German way to greenways and habitat networks. *Landscape and Urban Planning*, 76(1), 7-22. doi:<https://doi.org/10.1016/j.landurbplan.2004.09.041>
- Witney, B. (1988). *Choosing and using farm machines*. London: Longman.
- Yakoubi, M. A., & Laskri, M. T. (2016). The path planning of cleaner robot for coverage region using Genetic Algorithms. *Journal of Innovation in Digital Ecosystems*, 3(1), 37-43. doi:<https://doi.org/10.1016/j.jides.2016.05.004>

- Yu, X. (2015). *Optimization Approaches for a Dubins vehicle in coverage planning problem and traveling salesman problems*.
- Zhang, H., & Yang, S. (2018). Smooth path and velocity planning under 3D path constraints for car-like vehicles. *Robotics and Autonomous Systems*, 107, 87-99.
doi:<https://doi.org/10.1016/j.robot.2018.05.013>
- Zhang, X., Chen, J., Xin, B., & Peng, Z. (2014). A memetic algorithm for path planning of curvature-constrained UAVs performing surveillance of multiple ground targets. *Chinese Journal of Aeronautics*, 27(3), 622-633. doi:<https://doi.org/10.1016/j.cja.2014.04.024>
- Zhou, K., Leck Jensen, A., Sørensen, C. G., Busato, P., & Bothtis, D. D. (2014). Agricultural operations planning in fields with multiple obstacle areas. *Computers and Electronics in Agriculture*, 109, 12-22. doi:<https://doi.org/10.1016/j.compag.2014.08.013>

Appendix A: Path plan examples

The following figures provide some complete examples on path plans for various fields. For most fields, three scenarios are calculated, minimizing overlap, uncovered and number of turns. Two very complex fields are included at the end which only minimize number of turns. White arrows are provided to clarify path direction per field.

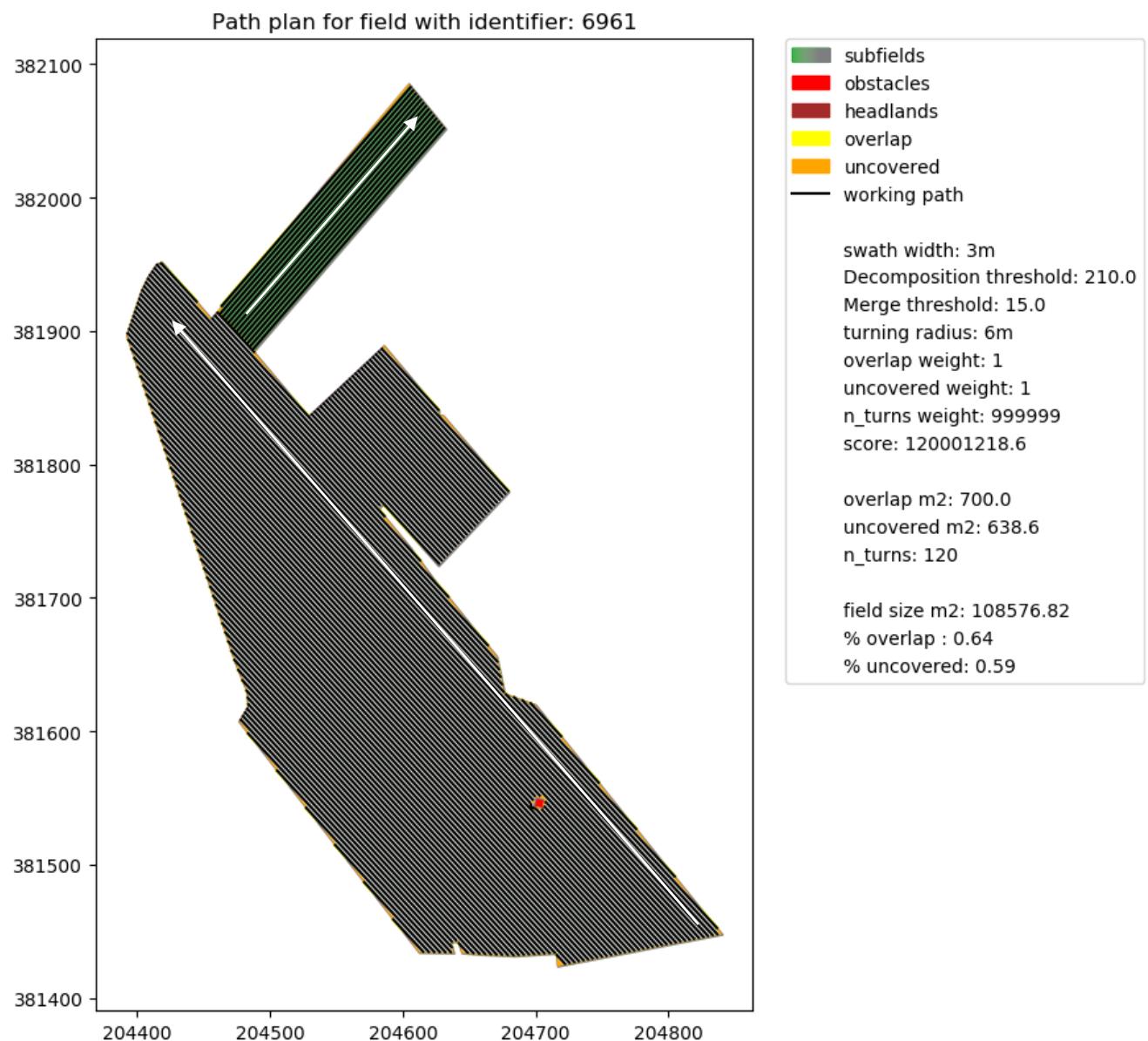
Field 3124

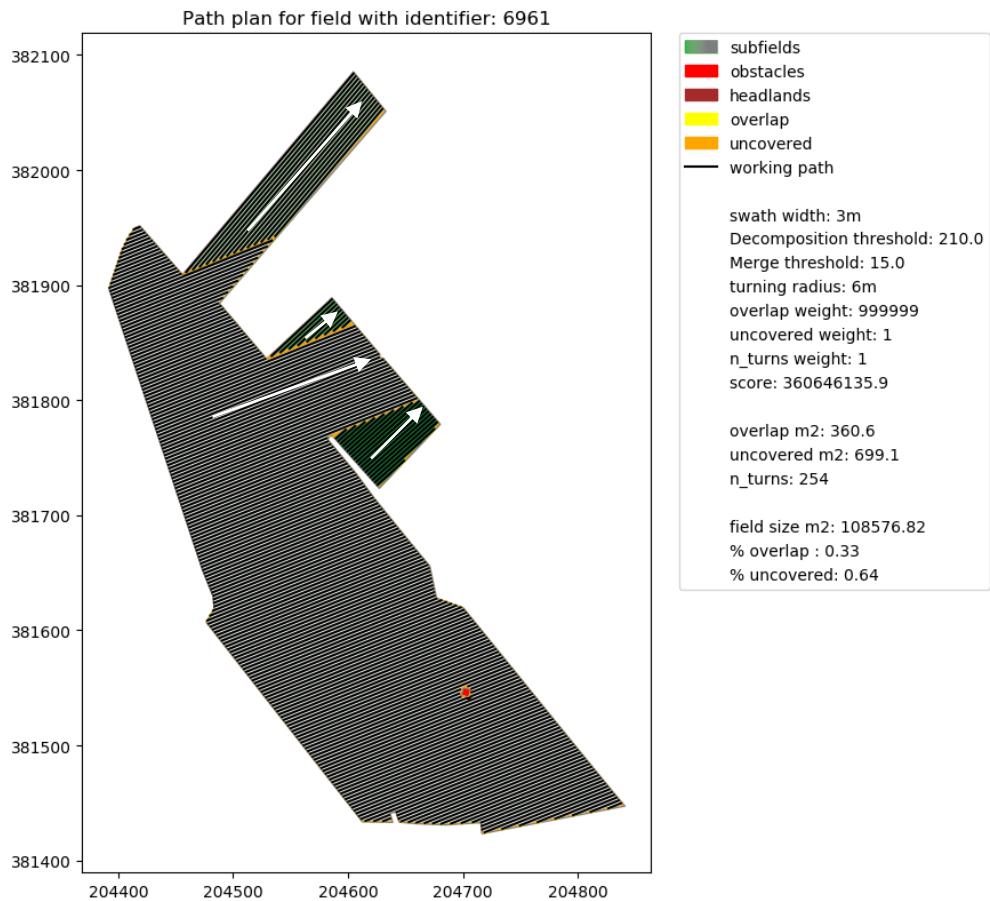
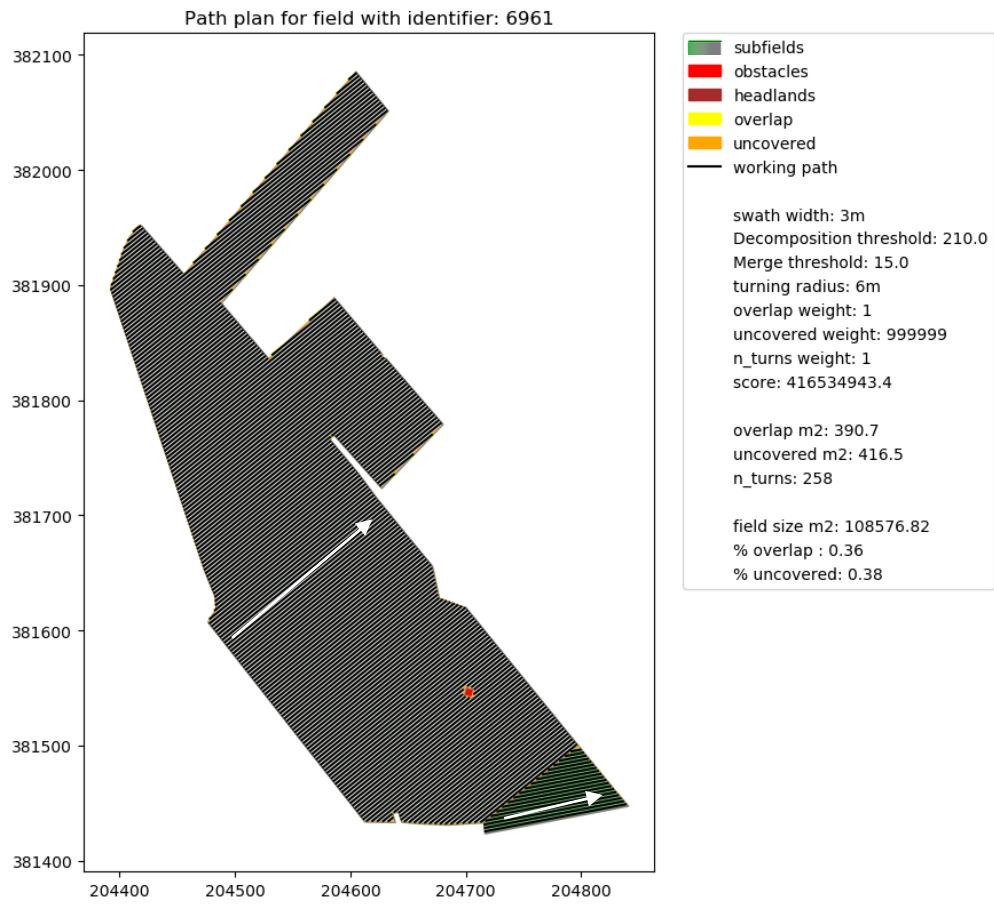
Average calculation time: 88.34 seconds (minimizing overlap/uncovered produced the same path plan, therefore only the latter scenario is presented).



Field 6961

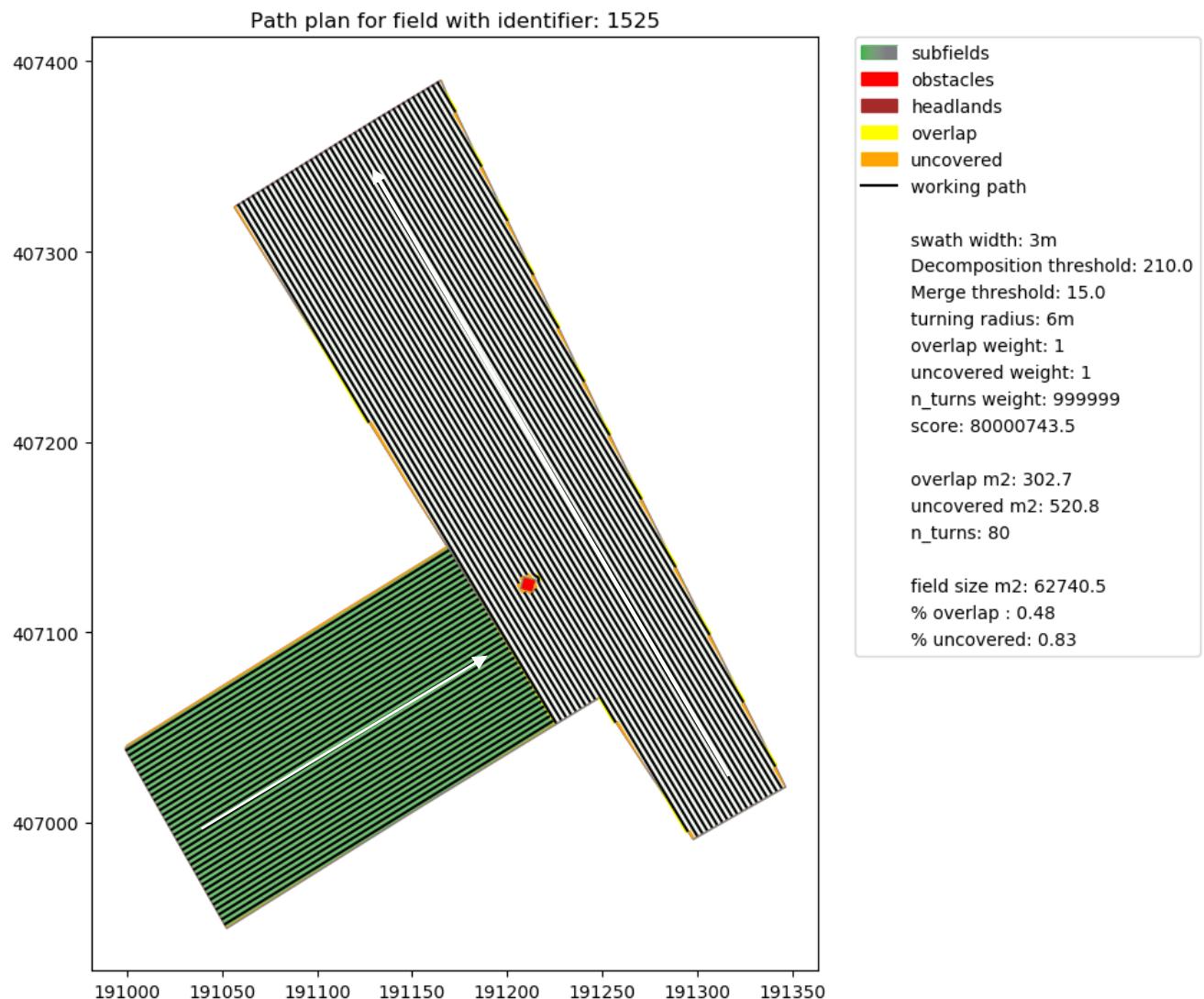
Average calculation time: 64.53 seconds.



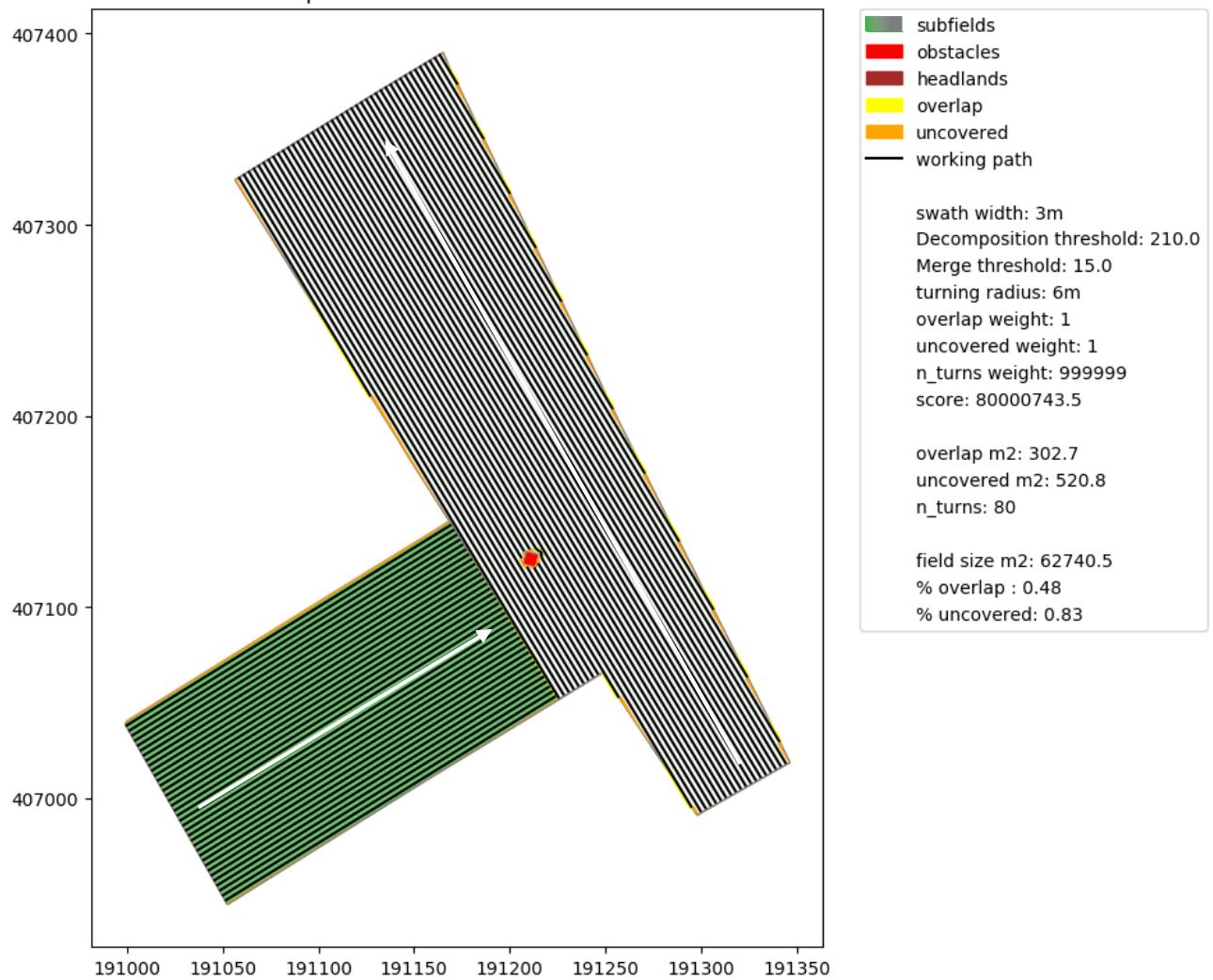


Field 1525

Average calculation time: 15.67 seconds (minimizing overlap/number of turns produced the same path plan, therefore only the latter scenario is presented).

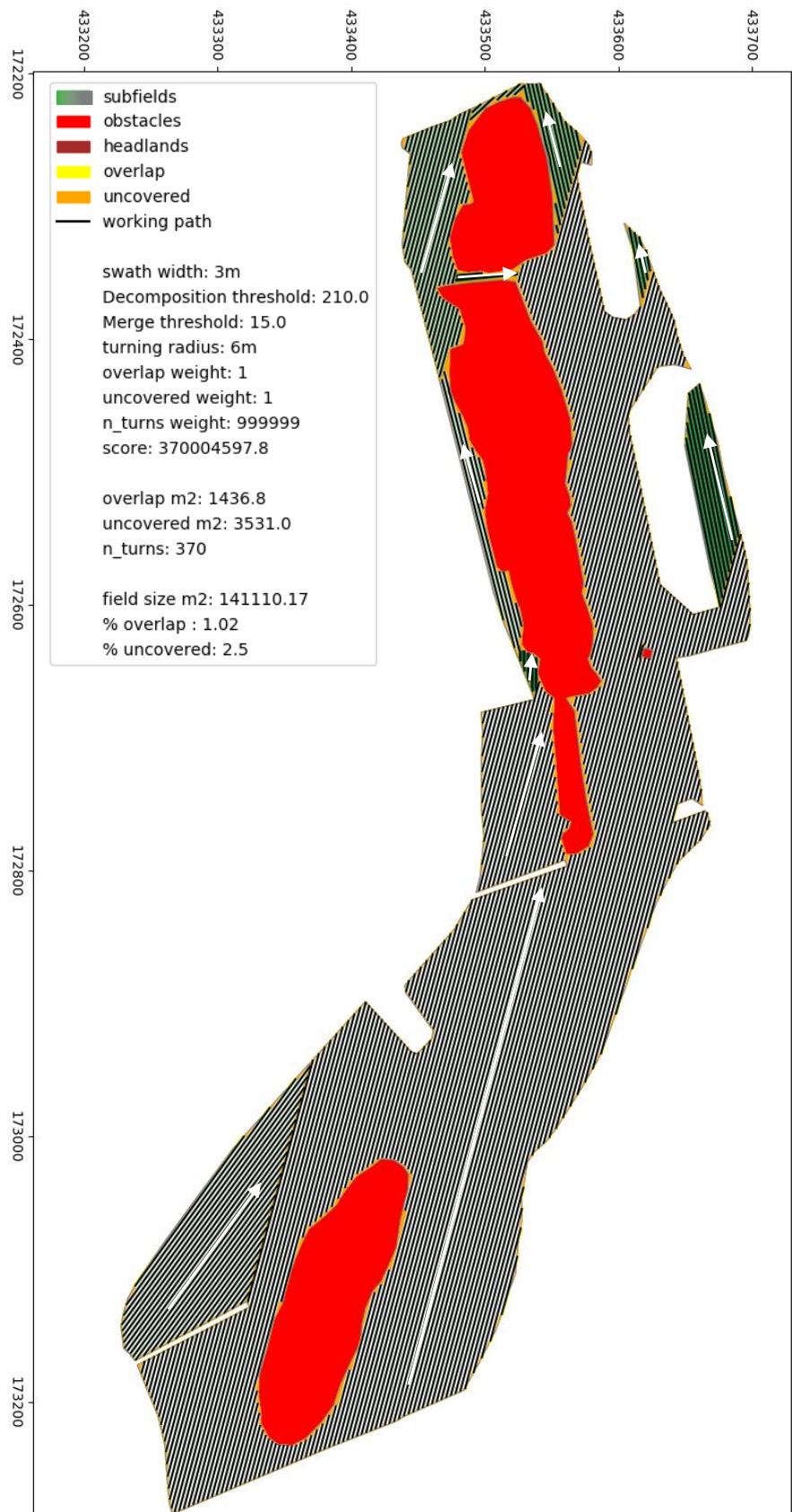


Path plan for field with identifier: 1525



Field 4162

Calculation time 1844 seconds, minimizing number of turns.



Field 5220

Calculation time 400 seconds, minimizing number of turns.

