Utrecht University

Master Thesis Proposal
MSc Computing Science

# Where do they go?

## A geometric seed dispersal model

*Author:*
C.P. (Tamara) Florijn
*Student number:*
5856442

*Supervisors:*
Frank Staals
Sarita de Berg
*Second corrector:*
Marc van Kreveld

Faculty of Science

December 10, 2021

# Abstract

C.P. (Tamara) Florijn

*Where do they go?*
*A geometric seed dispersal model*

Climate change can force plants to migrate to other areas via seed dispersal, the process of seeds moving away from the parent plant. Seed dispersal might be stopped by objects on the terrain. Other factors such as temperature could also limit the possible destinations.

How can we find the most likely destinations for the plant seeds? Which barriers should be removed to have the greatest gain in plant growth? The aim for this project is to find a good geometric model of representation to answer these and more geometrical and ecological questions. The approach could include approximation algorithms, dynamic algorithms or algorithms dealing with uncertainty.

**Keywords:** Plant migration, seed dispersal, climate change, geometric algorithms, theoretical computer science, obstacles

# Contents

# Chapter 1

# Introduction

Climate change might force plants to migrate to other regions to survive, since the environmental conditions change. Plants do not walk around but need to use the spreading of their seeds to migrate. This process is called *seed dispersal*. Seed dispersal could be an important factor for plants to survive climate change (Thuiller et al., 2008) .

Seeds can disperse via self-dispersal, such as the explosive mechanism of the plant species *Impatience capensis*, better known as jewelweed, or via external factors such as wind, water, animals and humans. Not all seeds travel the same distance; some rare events make the seeds dispersal very far. The *long distance dispersal* indicates the dispersal of 1% of the seeds that travelled the furthest.

The main objective of this thesis is to calculate where the seeds disperse to in a given environment over a period of time. This includes finding an appropriate model and designing and analyzing algorithms to solve geometric and ecological challenges.

As an orientation in the ecological field, we spoke to ecologist Wiemer Wamelink from Wageningen University & Research and ecologist Laurens Sparrius from the Dutch research institute FLORON. Wamelink and his colleagues developed a seed dispersal model called DIMO (Wamelink et al., 2014). DIMO formed the basis of our geometrical approach to seed dispersal.

## 1.1 Relevance

Computational seed dispersal models can provide a deeper understanding in the way seeds disperse, and predict where the seeds and plants will grow. Ecologist Wieger Wamelink made a remark in our conversation (see Chapter 2) that in the past, researchers investigated this mostly out of academic interest. Now the purpose roughly comes down to two points: preventing plant migration and stimulating plant migration.

Invasive species, such as the Japanese knotweed, can cause serious trouble, Wamelink explained. Generic models of seed dispersion can help focus the management and manipulation of these invasive species. For example, well-placed barrier zones can stop the spreading (Gosper et al., 2005).

On the other hand, to maintain a diversity of plant species, stimulation of plant growth is more relevant than ever because of climate change, deforestation and other limitations of habitat. Plants have limited migration speed that differs per species. Modelling seed dispersal can give more insight in the status of endangered species that cannot keep up the pace: the tolerance for temperature changes is limited. As an example, McGuire et al. (2016) evaluated the connectivity of the US landscape and calculated the effect that corridors would have on the plant movement capacity.

## 1.2 Research questions

As we mentioned before, we spoke to ecologist Wieger Wamelink from Wageningen University & Research. Wamelink developed a seed dispersal model called DIMO, short for DIspersal MOdel. DIMO simulates the dispersal of plants through a grid map over a period of time. The user can model barriers such as roads to indicate where seeds cannot disperse to. A more elaborate explanation on DIMO can be found in Section 3.2.

In our conversation with Wamelink, he elaborated on some of the challenges they faced with the construction and calculation of their ecological model DIMO. The recurrent problem was the long computation time. For example, he regretted that the model was not fast enough to run from the Penultimate Glacial Period (the glacial age that occurred before the Last Glacial Period) till now.

Also, Wamelink mentioned that the input maps indicating the barriers and the regions where plants can grow, are kept constant because of computation time constraints. In reality, the barriers might change over time, but this is not integrated in the model.

Inspired by this conversation and directed by our literature review, we formulated our research questions. We decided to focus on a geometrical representation instead of representations such as a graph or a grid model, because the problem is continuous in its essence. In this way, we can use the geometry and techniques from computational geometry to solve the problem.

1. Design a suitable geometrical model for seed dispersal on the basis of the ecological model DIMO.
2. Design algorithms to answer questions such as:
    a) What is the region that a certain plant occurs in after a period of time?

    b) What obstacle removal would enlarge the reached region for a specific plant the most?
3. What is the complexity of the output maps of the designed algorithms? And what is the time complexity of the algorithms? The complexity will be analyzed in terms of the size of the input, such as the number of input plants, the complexity of the vegetation maps and the requested number of time rounds.

We will look at different variations of the geometrical seed dispersal model, including:

a) Polygonal obstacles

In the current model of DIMO, the user can define subdivisions so that the region is disconnected. We will more generally focus on polygonal obstacles.

b) Convex distance functions

The ecological model DIMO takes wind into account in the dispersal of seeds. To model that geometrically, we will investigate the problem using other distance measures such as the convex distance function.

c) A dynamic obstacle map

In accordance with the challenge Wamelink touched upon, we will design algorithms that can handle a dynamic obstacle map. Can we efficiently remove or add obstacles every time step while running the model? And can we efficiently modify the obstacles, for example if only a constant number of vertices moves in each time step?

d) A dynamic habitat suitability map

Not all regions are suitable for plants to grow. If a seed is settled in the ground, it might need to wait until the region is suitable for the plant to germinate and start growing. In relation to the previous subquestion, we will look into a dynamic habitat suitability map that can change (slightly) every time step.

# Chapter 2

# Research orientation

To orient on the ecological subject of seed dispersion, we contacted the ecologists Wieger Wamelink and Laurens Sparrius. Both took the time to explain in detail how their field was structured, how the seed dispersal models contribute to science, and what challenges lay ahead. In this chapter, we give the summaries of the conversations.

## 2.1 Wieger Wamelink

On September 14, 2021, we spoke to Wieger Wamelink, ecologist at Wageningen University & Research. Together with his colleagues, he designed a model for plant dispersion, called DIMO. We will discuss the three main themes of the conversation: the benefits, the technical challenges and the limitations of the seed dispersal model.

**What is the benefit of knowing or predicting seed dispersion?**

'In the past, the benefit came down to scientific interest to explain why plant species occur at certain places. Now, we have a new incentive, namely climate change. Each plant species has an optimum in environmental temperature. If the temperature rises two degrees Celcius, we will lose over two hundred species in the Netherlands. Species either adapt, they migrate or they go extinct. As an example, in the Netherlands the tree species Birch loses its leaves already in summer recent years, which we predicted with our model.'

'Sometimes barriers prevent the spreading of seeds. These barriers could be natural, such as rivers, or human-made, such as roads. To restore the flow of seeds, we search for ways to reconnect the nature reserves or forests. In Germany, we now have a project in a very fragmented area, in which we try to find the best ways to reconnect the area again.'

'We usually do not focus on trying to prevent invasive species from spreading. Usually humans are a large factor, which makes prevention difficult. It is outside the scope of our research.'

**What are technical or geometric challenges you faced in building the model?**

'We chose a grid model as a basis, mostly because we are used to it. Also, many input maps have a grid representation. Water maps, on the contrary, use a polygon representation. Converting the polygons to grid representation takes a lot of time.'

'Some plants can travel very far, such as an orchid. All the grid cells need to be calculated one by one which takes a lot of time. And time is the bottleneck, not space.'

‘When we model barriers, we use line segments in the grid model. Sometimes the barriers do not connect well over the different grids, resulting in unwanted gaps that produce errors. This takes a lot of time to fix.’

‘Using a 3D terrain would take too much time. Whether a barrier is high or low does matter, but we model that using a degree of permeability for barriers.’

**Limitations: what would you hope for in a plant dispersion model?**

‘It would help us a lot if we could switch easily between the representation of polygons and grids. For example, if we could use the polygon representation to solve a sub-problem with the water card, then switch back efficiently to a grid model when we present it to outsiders.

‘At this moment, we use the current vegetation map for both the past and the future. Updates occur frequently, such as a parameter update or an input map change. It would be handy if the model could do these updates dynamically. Note: the hundred maps are not random, there is a line of logical update in it.’

‘It has always been a goal for me to compute the model from the Penultimate Glacial Period (the glacial age that occurred before the Last Glacial Period), but for now that takes too much time.’

## 2.2 Laurens Sparrius

On September 21, 2021, we spoke to ecologist Laurens Sparrius, acquainted with the Dutch research institute FLORON. FLORON maps the distribution of common plants in the Netherlands. The data is collected by hundreds of people, mostly volunteers.

In our conversation, Sparrius elaborated on the structure and the purpose of the organisation. The data in the databases of Floron describe plant observations. These consist of the name of the plant, a time stamp and a location.

The Floron Verspreidingsatlas Vaatplanten[1] presents this data in a visually attractive way to support easy use. A user can make a variety of requests, such as asking the system to provide a map for one specific plant over a longer period of time, or limit the region to see an overview of plants that grow there.

When we asked for geometric or technical challenges, Sparrius mentioned the challenge of showing the data clearly and efficiently. As an example, the location of the observations all have an inaccuracy. Some observations are accurate up to a few meters, while other have an uncertainty radius of a few hundred meters. This is shown as an octagon, but is stored as a centre point and a radius. Since these challenges lay in the context of data visualisation, we will not focus our research on this direction.

[1] `https://www.verspreidingsatlas.nl/1398`

# Chapter 3

# Ecological background

In this part, we will discuss a myriad of ecological models with different representations and goals. Then, we will give a detailed overview of the DIMO model.

## 3.1 Overview

The earliest research to model the spread of plant populations assumed that plants spread via a Gaussian distribution. This assumption led to the problem that the actual migration of plants was much faster than the models predicted, the so called *Reid's paradox of rapid plant migration* (Reid, 1899; Clark et al., 1998). Rare *long-distance dispersal* events helped to explain the faster spreading rate.

### 3.1.1 Modelling spreading plants

We will first shortly introduce the DIMO model that is developed by ecologist Wieger Wamelink, whom we spoke to, and his colleagues (Wamelink et al., 2014). DIMO simulates the dispersal and establishment of plants through a possibly fragmented landscape – grid map – over a period of time in steps of one year. Each run concerns one plant species with specific parameters. The model could be helpful in making or evaluating policy. In Section 3.2 we will give a more elaborate overview of the model.

Andújar et al. (2017) constructed a geometrical model to predict the spreading of Johnsongrass (*Sorghum halepense*), an aggresively spreading weed, in maize fields in Central Spain. In contrast with the DIMO model, the authors used varying patch sizes and varying patch densities.

Just as the previous model, Somerville et al. (2020) investigated the prevention of weed spreading. Their review described both static and spatio-temporal models. One of the problems they tackled was where to use spray to stop the spreading of weed. The challenge is to use enough spray without waste.

### 3.1.2 Habitat connectivity

Climate change could demand higher plant migration rates than possible, especially in regions that are disconnected because of, for example, roads. Dullinger et al. (2015) compared different levels of habitat fragmentation in their models to evaluate the migration rates of plants. They modelled a hypothetical landscape using a grid raster. Their results suggested to focus on the movement of species throughout the whole countryside instead of only protected areas, because of the importance of connectivity.

Albert et al. (2015) collected data to assess what plant traits affect seed dispersal. Their predictive models can help decide how to restore habitat connectivity. As an example, if mostly land animals instead of birds transport the seeds, then it is efficient to restore corridors for animals; in the other case it might not be. Note, this is not a geometric model, but a systematic literature review.

### 3.1.3 Other spreading models

Not only seed disperse, also other substances can spread over a region. Teggi et al. (2018) created a dispersion model for pollutants. The sources are modelled as polygons with corresponding emission rates. Whereas DIMO and other models use a representation of presence (1) or absence (0) of plants, the output of this model consists of a spatial map with the concentration of pollutants in the region.

## 3.2 DIMO Model

Wamelink and his colleagues (2014) developed a dispersal model called DIMO. DIMO simulates the dispersal of plants on a 2D map over a period of time. Each run concerns one plant species with specific parameters. The output map shows the age of the plant per grid cell. At the start of the model, the initial source locations of the plant are given in a 2D landscape, see for example Figure 3.1a. The region is represented by a homogeneous grid; each cell of 250m $\times$ 250m indicates the presence or absence of a species. A function defines the time of arrival of the dispersing seeds from the source locations. This function depends on the dispersal by wind and by animals, possibly with hindrance of obstacles.

Figure 3.1b shows the plant age map after 10 years with a uniform wind, but the user can also use another function for the wind. The dispersal by wind is defined by meters per time unit per direction (given in degrees). The landscape can have different compartments for different wind functions. An example of the effect of wind is shown in Figure 3.1c. The dispersal by animals is given by a radius $R$ in meters per time unit. Obstacles, either for animals or for the wind, limit the spreading. In the current version of DIMO, the region is split into compartments such that seeds cannot spread from one compartment to the other, see for example an output map with the influence of barriers in Figure 3.1f.

Not all regions are suitable for the germination and growing of plants, which is defined in the *habitat suitability map*, specific per plant species. This is different from a barrier, since the germination is limited, not the dispersal. Plants can only germinate in a suitable area.

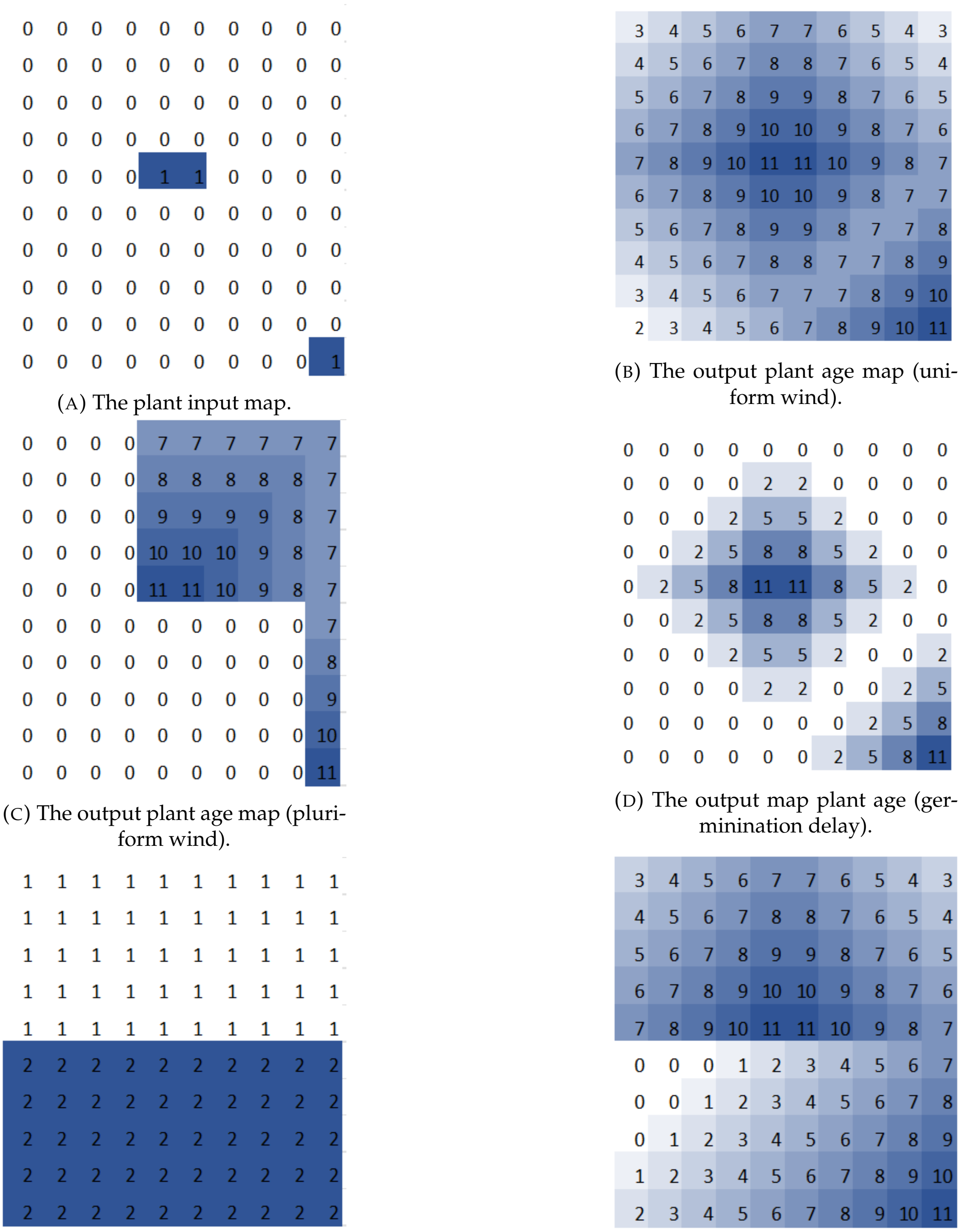
If a seed arrives on a location, the plants do not start growing immediately. The parameter *germination delay* denotes the time that a seed stays in the seed bank before it germinates. If the plant germinates, the plant should be counted as present. The parameter *the age of the first flowering* defines the time between germination and being a reproductive source plant itself. However, this parameter is not included in the current implementation version of DIMO. Figure 3.1d shows an example of a plant age map with germination delay. Since seeds cannot survive forever in the seed bank, the parameter *seed longevity* indicates the time a seed can survive dormant in the seed bank.

In short, for each plant, the input for DIMO consists of:

- A map of the landscape.
- A set of points, the source locations of the plant, including the age of the plant (Figure 3.1a).
- A set of points, the locations of the seeds of the plant, including the age of the seed.
- A habitat suitability map.
- A set of barriers, both for animals and for the wind (Figure 3.1e).
- Distance functions $c_i$ for the wind speed and force for compartment $i$ in the landscape.
- Parameters for the wind dispersal distance, animal dispersal distance, the seed-bank longevity and the germination delay.

In DIMO, the output consists of

- A plant age map that gives the number of years the plant is present as reproductive plant (Figure 3.1b).
- Suitability map, only the suitable areas reached by the plant.
- Seed-bank-life map that denotes the number of years left that the seed can survive in the seed bank.
- Seed-distribution map: the points the plant reached in the last round.



|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

(A) The plant input map.

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | 6 | 7 | 7 | 6 | 5 | 4 | 3 |
| 4 | 5 | 6 | 7 | 8 | 8 | 7 | 6 | 5 | 4 |
| 5 | 6 | 7 | 8 | 9 | 9 | 8 | 7 | 6 | 5 |
| 6 | 7 | 8 | 9 | 10 | 10 | 9 | 8 | 7 | 6 |
| 7 | 8 | 9 | 10 | 11 | 11 | 10 | 9 | 8 | 7 |
| 6 | 7 | 8 | 9 | 10 | 10 | 9 | 8 | 7 | 7 |
| 5 | 6 | 7 | 8 | 9 | 9 | 8 | 7 | 7 | 8 |
| 4 | 5 | 6 | 7 | 8 | 8 | 7 | 7 | 8 | 9 |
| 3 | 4 | 5 | 6 | 7 | 7 | 7 | 8 | 9 | 10 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

(B) The output plant age map (uniform wind).

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 7 | 7 | 7 | 7 | 7 | 7 |
| 0 | 0 | 0 | 0 | 8 | 8 | 8 | 8 | 8 | 7 |
| 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 8 | 7 |
| 0 | 0 | 0 | 0 | 10 | 10 | 10 | 9 | 8 | 7 |
| 0 | 0 | 0 | 0 | 11 | 11 | 10 | 9 | 8 | 7 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 |

(C) The output plant age map (pluriform wind).

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 2 | 5 | 5 | 2 | 0 | 0 | 0 |
| 0 | 0 | 2 | 5 | 8 | 8 | 5 | 2 | 0 | 0 |
| 0 | 2 | 5 | 8 | 11 | 11 | 8 | 5 | 2 | 0 |
| 0 | 0 | 2 | 5 | 8 | 8 | 5 | 2 | 0 | 0 |
| 0 | 0 | 0 | 2 | 5 | 5 | 2 | 0 | 0 | 2 |
| 0 | 0 | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 5 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 8 |
| 0 | 0 | 0 | 0 | 0 | 0 | 2 | 5 | 8 | 11 |

(D) The output map plant age (gerinination delay).

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |

(E) The input map for barriers.

|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | 6 | 7 | 7 | 6 | 5 | 4 | 3 |
| 4 | 5 | 6 | 7 | 8 | 8 | 7 | 6 | 5 | 4 |
| 5 | 6 | 7 | 8 | 9 | 9 | 8 | 7 | 6 | 5 |
| 6 | 7 | 8 | 9 | 10 | 10 | 9 | 8 | 7 | 6 |
| 7 | 8 | 9 | 10 | 11 | 11 | 10 | 9 | 8 | 7 |
| 0 | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

(F) The output plant age (barriers)

FIGURE 3.1: Visualisation of the DIMO model. The numbers show the age of the plant in the grid cell. (A) The plant input map shows the set of grids that contain the source plants, represented by "1". (B) The output plant age map with uniform wind. (C) The output plant age map with pluriform wind. The wind force is present in the direction North, North-East and East. (D) The output map plant age with a germination delay of 2. (E) The input map for barriers. Here, the region is divided into two regions, 1 and 2. The seeds cannot spread from region 1 to 2, and vice versa. (F) The output plant map when the barriers given in Figure 3.1e are used. Here, one can see that the seeds do not travel from compartment 1 to compartment 2.

# Chapter 4

# Geometric background

In the geometric part of the literature review, we will discuss geometrical problems that have been (partly) solved before. First, we will shortly mention marching algorithms for grids, since the model in DIMO has a grid representation. Then we will discuss algorithms for computing the visibility polygon. Even though most visibility polygon algorithms do not involve calculating shortest paths, they do relate to seed dispersal models in the context of "reachable regions".

Because calculating shortest paths in polygons is relevant, possibly using other metrics than Euclidean distance, we will review shortest path algorithms after that. One of these metrics is the convex distance function, which we will inspect in more detail. Finally, we will elaborate on Minkowski sums, because this operation corresponds closely with the way plant regions expand in one time step.

## 4.1 Marching algorithms

DIMO, the ecological model we based our approach on, represents the landscape using grids. Therefore, we explored marching algorithms for grids, which is called marching squares in 2D or marching cubes in 3D (Lorensen and Cline, 1987). Because our model is in 2D, we will only discuss marching squares here.

Marching squares generates contours in a field. The input is a collection of square grid cells, each with a numerical value. The marching square algorithm converts the array to a binary image: vertices with a value greater than a given threshold are marked '1', the vertices below the threshold are marked '0'.

The contours of the final image are computed locally using the sixteen different combinations of values for four vertices.

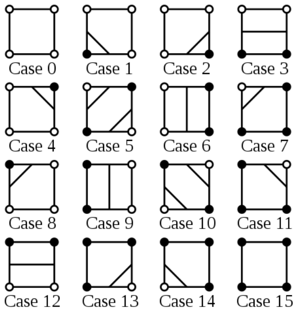


FIGURE 4.1: Sixteen combinations of binary values of four vertices. This look-up table shows locally how the contour line forms.

We found marching algorithms were not so relevant for the geometric seed dispersal model, since we are interested in computing the numerical or binary map that represents the presence or absence of plants. Possibly, marching algorithms could be applied in the final phase of the seed dispersal model, but we will not go into that now.

## 4.2 Visibility polygon

Given a point $p$ in a simple polygon $P$. The point visibility polygon $V$ is the set of points in $P$, such that each point $v \in V$ is visible from $p$, that is that the segment $vp$ does not intersect any obstacle or boundary.

If the polygon is simple and no obstacles are present, then the visibility polygon can be found in $O(n)$ time with $n$ the number of vertices of $P$, using the algorithm of Lee (1983) with corrections by Joe and Simpson (1987).

In the case of obstacles, Lee (1978) constructed a rotational plane sweep algorithm to compute the visibility graph. The visibility graph has an edge between every pair of vertices of $P$ that are mutually visible. For each vertex $p$, we rotate a half-line $\rho$ around $p$ to treat the vertices of the obstacles one by one. Each vertex event, we check whether the obstacle vertex is visible from $p$. This can be computed efficiently using a balanced search tree on the edges that intersect with the half-line $\rho$. For a set $S$ of polygonal obstacles with $n$ vertices, one rotational sweep can be computed in $O(n \log n)$. The visibility graph can then be computed in $O(n^2 \log n)$ time.

Heffernan and Mitchell (1995) provide an optimal algorithm to compute the visibility in the plane. If the $h$ holes are pair-wise disjoint simple polygons, then the visibility polygon can be computed in $O(n + h \log h)$ time and $O(n)$ space.

Relating back to the seed dispersal problem, one can see that seeds do not reach an infinite region in one step. This limited range of vision variation of the visibility polygon is discussed by Ben-moshe and Katz (2004). The problem is as follows. Given are a simple polygon $P$ with $n$ vertices, and a set $S$ of $m$ sites in $P$. Each site has an associated range of vision. A range-restricted visibility graph can be computed in $O(n \log n + m^{3/2} + k)$ time, with $k$ the size of the output, and $O(n + m \log m)$ storage.

## 4.3 Distance measures

As we explained in Section 3.2, DIMO takes the force of the wind into account in the dispersal of seeds. We will discuss several distance measures and evaluate the suitability for our geometric seed dispersal model, specifically in the context of including the effect of wind.

The Euclidean distance is a popular distance measure and usually fits well in settings applied to the real world. Let $i = (i_x, i_y)$ and $j = (j_x, j_y)$ be two points in $\mathbb{R}^2$. Then the Euclidean distance between $i$ and $j$ is defined by

$$d(i,j) = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}.$$

Another popular distance measure is the Manhattan distance. The Manhattan distance is defined as

$$d(i,j) = |i_x - j_x| + |i_y - j_y|.$$

The generalization of the Euclidean and Manhattan distance is known as the Minkowski distance, given by

$$d_p(i,j) = \sqrt[p]{|i_x - j_x|^p + |i_y - j_y|^p}.$$

This measure is also called the $L_p$ *norm*. The Manhattan distance measure corresponds with $L_1$ and the Euclidean distance measure corresponds with $L_2$.

The discussed measures are symmetric, positive and adhere to the triangle inequality. The symmetric property indicates that the distance from $i$ to $j$ is equal to the distance from $j$ to $i$. The positive property means that the distance between two distinct points is positive, and the distance from any point to itself is 0. The triangle inequality states that the distance from $i$ to $j$ via $k$ is at least as long as the shortest distance from $i$ to $j$. The combination of these properties is known as a *metric*.

### 4.3.1 Geodesic distances

Given are two points, $p$ and $q$, that both lie in a polygon $P$. The geodesic distance between $p$ and $q$ is defined as the length of the shortest path from point $p$ to $q$ that is completely contained in polygon $P$.

### 4.3.2 Convex distance function

As we said before, we would like to model the influence of wind. Because wind and headwind exist, we need to let go of the property of symmetry. Therefore we will look into a notion that is called the *convex distance function*, which exactly characterizes distance functions satisfying the triangle inequality (Barequet et al., 2001). Since the distance measure satisfies triangle inequality, the shortest distance between two points in a polygon will be a polygonal path.

In Figure 4.2, an example of a convex distance polygon is shown. Let $C$ be a convex distance function body that contains the center $O$ in its interior. The boundary of $C$ denotes all the points with distance 1 from the center. To calculate the distance from a point $p$ to a point $a$ using a convex distance function, we translate $C$ and shoot a ray from $p$ through $a$. Let $v$ be the intersection point between the ray and $C$. Then a convex distance function $d_C$ is defined by

$$d_C(p,a) := \frac{||a-p||}{||v-p||},$$

where $||a-p||$ is the Euclidean distance from $a$ to $p$. $d_C(p,a)$ is the factor that $C$ must be contracted or stretched to touch $a$. (Ma, 2000).

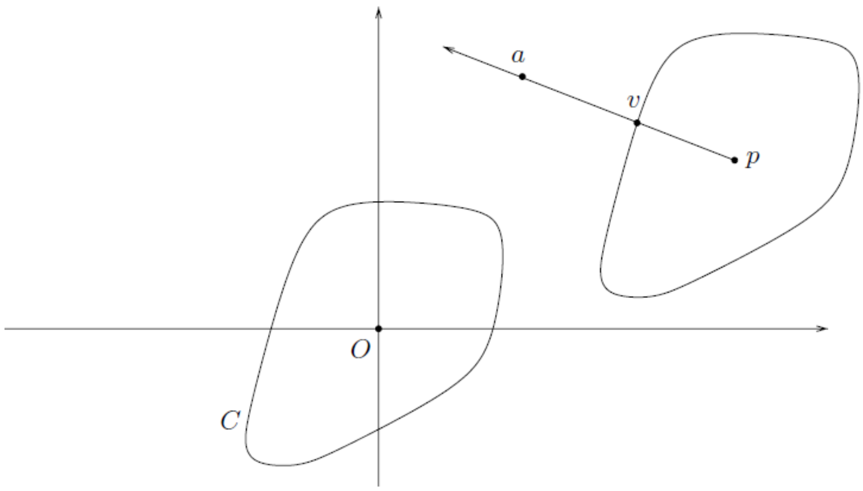


FIGURE 4.2: A convex distance function. (Ma, 2000).

## 4.4 Shortest paths

In this section, we will discuss articles that examine shortest path algorithms. First, we will discuss geodesic distances. Then we will elaborate on continuous Dijkstra, a technique that can be used to handle shortest paths in case of obstacles. Another important concept that we will include is Voronoi diagrams. At last, we will shortly mention algorithms that calculate shortest path with multiple convex distance functions, also called *anisotropic regions*.

### 4.4.1 Geodesic distances

Geodesic distances have been studied extensively in the past. Problems in the geodesic setting that have been solved optimally regarding their time complexity include computing the shortest path tree of a point in a simple polygon (Guibas et al., 1986) and, very recently, an optimal algorithm for geodesic Voronoi diagrams in simple polygons (Oh, 2019). A more elaborate explanation on Voronoi diagrams can be found in Section 4.4.3.

Using the geodesic distance, Arge and Staals (2017) addressed a problem that has its roots in ecology or, even more specific, in the spreading of plants. We are given a threshold $\epsilon$, two sets of $m$ points in $\mathbb{R}^2$, a set of "red" points $R$, the plant locations, and a set of "blue" points B, the possible destination locations, in a simple polygon with $n$ vertices. Each point $p \in R \cup B$ also has a real value $p_v$, representing for example temperature. The question is to find, for every plant location (red point) a closest destination location (blue point), provided that the ecological value of the blue point differs at most $\epsilon$ from the ecological value of the red point.

The algorithm they provide can handle deletions and insertions in $O(\sqrt{m} \log^3 n)$ time and process geodesic nearest neighbour queries in $O(\sqrt{m} \log m \log^2 n)$, with a space usage of $O(m \log n + n)$.

### 4.4.2 Continuous Dijkstra

We will continue this overview of literature in the direction of problems where polygonal obstacles are present. Hershberger and Suri (1999) provide an optimal algorithm called continuous Dijkstra, to compute a planar map that encodes all shortest paths from a fixed source point to all other points in the plane. If $n$ is the number of vertices in the polygonal obstacles, then the map can process queries in $O(\log n)$ time.

The algorithm uses an efficient wavefront propagation. It simulates a "bubble" expanding from one source point. The wavefront at time $t$ consist of all the points that have shortest distance $t$ to the source vertex $v$. The boundary of the wavefront is a set of circular arcs, as can be seen in Figure 4.3. The arrows in this figure show that the meeting point between these arcs is either a straight line or a hyperbola.

To simulate the wavefront, events are processed one by one. At these events, the topology of the wavefront might change. These events are categorized as *wavefront-wavefront collisions* or *wavefront-obstacle collisions*. Collisions between wavefronts can happen between arcs that are neighbours or non-neigbours. Collisions between non-neighbouring arcs are not easy to detect and process. Therefore, the authors introduced the concept of an *approximate wavefront*. Only in the second phase of the algorithm, the exact collision events are computed using Voronoi diagram techniques.

Using an efficient data structure called a *quad-tree-style* subdivision, the edges and vertices of the obstacles can be saved and searched through efficiently.

The algorithms takes $O(n \log n)$ time and $O(n \log n)$ space. $O(n \log n)$ running time is optimal. Wang (2021) settled the open problem for the optimal space requirement by introducing an algorithm with $O(n \log n)$ time and $O(n)$ space complexity.

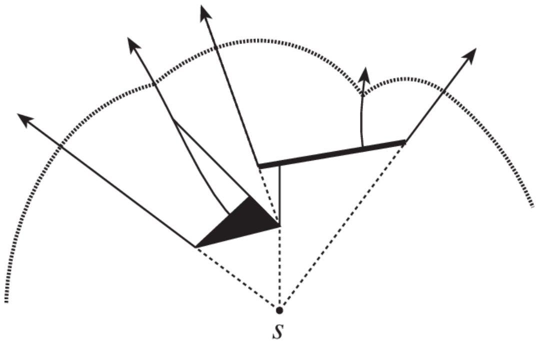


FIGURE 4.3: The wavefront of the algorithm continuous Dijkstra. (Hershberger and Suri, 1999).

### 4.4.3 Voronoi diagrams

A Voronoi diagram partitions the plane with a given set $S$ of points into regions. Each $p \in S$ has a corresponding cell in the Voronoi diagram that contains all the points in the plane that are closer to $p$ than any other point in $S$.

The geodesic Voronoi diagram of $m$ points in a simple polygon with $n$ vertices can be computed (optimally) in $O(n + m \log m)$ time using a sweep line algorithm (Oh, 2019).

### Convex distance function

Since we are interested in using different distance measures, we researched how to construct bisectors and Voronoi diagrams using a convex distance measure.

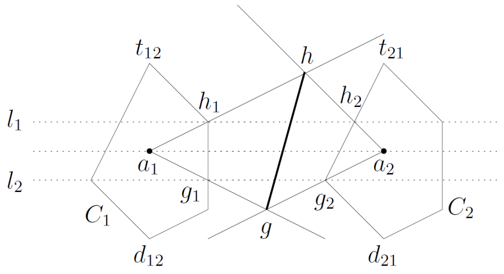


FIGURE 4.4: Construction of bisector between $a_1$ and $a_2$. (Ma, 2000).

The complete construction of the bisector using a sweep line costs $O(n)$ time if $C$ is a polygon with $n$ vertices. Since the bisector can have $n-2$ segments, this running time is optimal (Ma, 2000). First, we will sketch how to construct one segment of the bisector, see Figure 4.4. This process can be repeated to form the complete bisector.

Given are two points $a_1$ and $a_2$ and assume w.l.o.g. that they are on a horizontal line. We will construct the bisector of these two points, see also Figure 4.4. Let $h_1$ be a vertex of $C_1$ and let $h_2$ be the intersection point of the horizontal line through $h_1$, and $C_2$. Find the intersection point $h$ of the line through $a_1h_1$ and the line through $a_2h_2$. Point $h$ is then on the bisector. Choose vertex $g_2$ of $C_2$ in such a way that the area between the horizontal line through $h_1$ and the horizontal line through $g_2$ does not contain any vertices. Then, find the intersection between the horizontal line through $g_2$ and $C_1$, $g_1$. Then, find intersection point $g$ for the line through $a_1g_1$ and $a_2g_2$. In the case of Figure 4.4, point $h_1$ is a vertex of $C_1$ and $g_2$ is a vertex of $C_2$, but this is not necessarily so. The line segment $hg$ is part of the bisector between $a_1a_2$.

In the degenerate case that one of the edges of $C$ is parallel to $a_1a_2$, the bisector is not a line but a non-bounded region, see Figure 4.5. If you expand the reflection of $C$ in $p$ and $q$, they coincide instead of intersect at one point. Therefore, we introduce the concept of a chosen bisector, based on lexicographical rules, and pick one of the edges of the non-bounded region as the bisector.

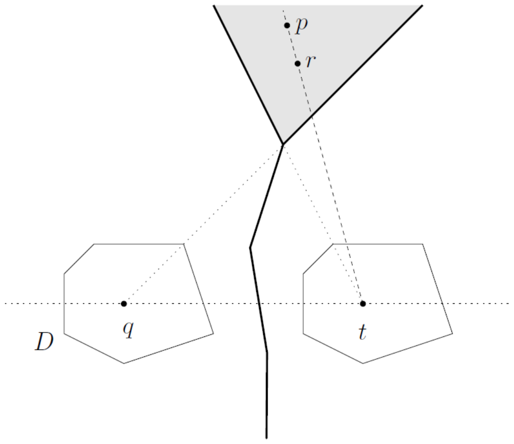


FIGURE 4.5: In the degenerate case, the bisector is not a line but a non-bounded region. (Ma, 2000).

Now that we know we can construct a bisector when using convex distance functions, the next question is how to efficiently compute the Voronoi diagram. Chew and Dyrsdale (1985) provide a divide-and-conquer algorithm that can compute the Voronoi diagram in $\Theta(m \log m)$ time, with $m$ the number of sites, given that the intersection of two bisectors can be found in constant time. Dehne and Klein (1997) provide a sweep line algorithm that can handle Voronoi diagrams for an arbitrary nice metric within (optimal) $O(m \log m)$ time. This concept is called *abstract Voronoi diagrams* and includes symmetric convex distance functions (Klein, 1988). This approach can be generalized to handle arbitrary convex distance functions (Ma, 2000).

### 4.4.4 Anisotropic regions

In the previous section, we assumed that the complete region was associated with one convex distance function. In the anisotropic setting, different convex distance functions might be used for different subregions. If more than one convex distance function is allowed, the triangle inequality does not hold anymore, see also Figure 5.5. The seed dispersal problem in the anisotropic setting is more difficult, and is usually approached by approximation instead of finding an exact solution. For example, Cheng et al. (2008) found an algorithm to approximate shortest paths in anisotropic regions. In our research, we will focus on exact algorithms and will not go into anisotropic regions further.

## 4.5 Minkowski sum

In the seed dispersal problem, seeds spread themselves in discrete time steps. Suppose the shape of the wind force is given by a polygon $R$. That means that every plant from round $t$ will spread seeds around itself in the form of polygon $R$. We can compute the region of plants in round $t+1$, say $B_{t+1}$, by taking the Minkowski sum of the region of plants in round $t$, say $B_t$, with polygon $R$.

We can visualise the Minkowski sum as sliding polygon $R$ along the boundary of $B_t$. An example can be seen in Figure 4.6. The formal definition of the Minkowski sum of sets $S_1 \subset \mathbb{R}^2$ and $S_2 \subset \mathbb{R}^2$ as given in de Berg et al. (2008) is

$$S_1 \oplus S_2 := \{p + q : p \in S_1, q \in S_2.\}$$

Since polygons are planar sets, the definition also applies to polygons. The Minkowski sum of two convex polygons with $n_1$ and $n_2$ vertices is convex itself and can be computed in $O(n_1 + n_2)$ time. If one of the polygons is convex and the other one is non-convex, then the time complexity is $O(n_1 \cdot n_2)$. If both are non-convex, then the running time is $O({n_1}^2 n_2^2)$. (de Berg et al., 2008).

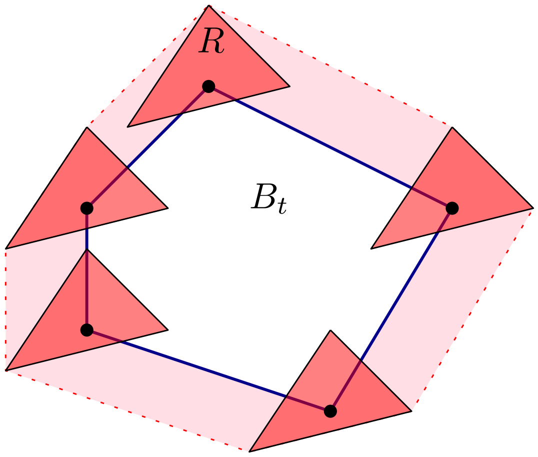


FIGURE 4.6: Visualisation of a Minkowski sum of a polygon $B_t$ and $R$.

# Chapter 5

# Preliminary results

In this chapter, we include some of the results we found so far. An important part of the research is formalising the ecological model into a geometrical model. We will share an example of such a formalization of one of the variations here. The chapter will close off with visualisations of some realisations we had during the process.

## 5.1 Sketch formalisation simplest problem

Given are a region $W$, which we will represent as a convex polygon with $n$ vertices, and a point $p$, the initial source plant. At $t = 0$, the whole region $W$, except for point $p$, does not contain plants. Assume point $p$ has age 1 at time $t = 0$. Each time step $t$, each point in the region with plants at time $t$, we call it $B_t$, disperse their seeds within a radius $r$. That is, the plant region $B_{t+1}$ is the Minkowski sum of $B_t$ and a disk with radius $r$. The region with seeds will be reproductive plants and spread seeds themselves after one time step. We want to answer the following question: at any time $t$, what is the plant region $B_t$, and what age do the plants have? The output map is a subdivision of $W$, where each region has a corresponding age with value $[0, t+1]$.

### Sketch solution

This problem for a convex polygon $W$ is the same as growing a disk with radius $r$ each step. The radius of the disk with plants is $t \cdot r$ at time $t$ starting with time $t = 0$. For example, this means that $B_1$ is a disk with radius $r$ and age 1 with center $p$ ($p$ has age 2), given that the region with plants did not yet intersect with the boundary of $W$. The output consists of concentric circles, with point $p$ in the center, intersected with $W$. The output map has complexity $O(n + t)$, since every edge of $W$ intersects at most twice with the disk in round $t$, and one edge can intersect all $t$ circles.

If $W$ is not convex, then it will not work to simply intersect $W$ with the concentric circles, as is shown in Figure 5.1. Also, concentric circles of a different radius can appear in a non-convex $W$, an example is presented in Figure 5.2. Instead of 'classic' disks, we then need to look at geodesic disks. Geodesic disks with center $p$ and radius $r$ are defined as all the points that have geodesic distance $r$ to point $p$.

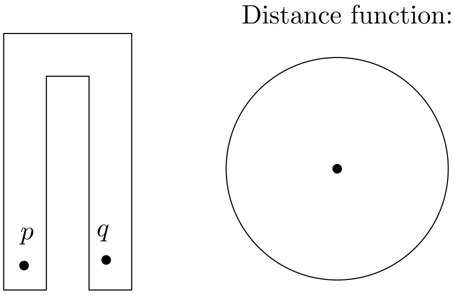


FIGURE 5.1: Taking just the intersection with the distance polygon does not work in this case.

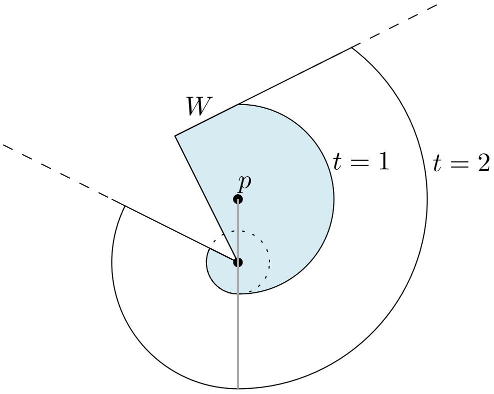


FIGURE 5.2: If world $W$ is not convex, then concentric circles of a different radius than distance function $r$ can appear.

## 5.2 Observations

### DIMO vs. wave propagation

As one can see in Figure 5.3, the model of DIMO is not the same as a wave propagation.

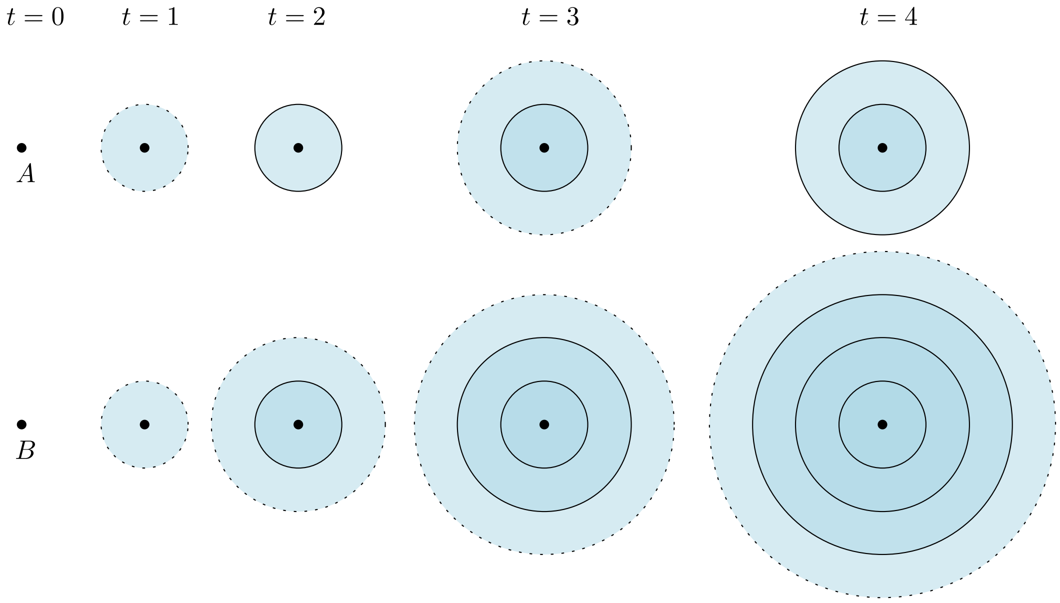


FIGURE 5.3: The plants from point $A$ expand according to the model of DIMO, and the plants from point $B$ expand as a wave propagation. In this figure, the dotted line represents the border of a region with seeds. A continuous line represents a region where plants are present. The germination delay is 1.

### Shortest path with convex distance measure

A shortest path in the Euclidean metric may not be a shortest path in a convex distance measure, as can be seen in Figure 5.4. Though $pq$ via $s$ is shorter in the Euclidean metric, with a specific convex distance function (on the right) $pq$ is shorter via $r$.

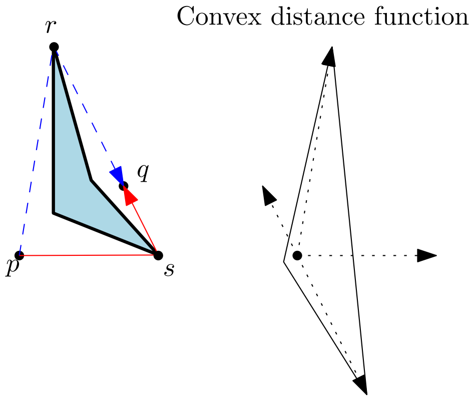


FIGURE 5.4: Shortest path with convex distance measure may be different from the shortest path in Euclidean metric.

### Triangle inequality

Triangle inequality does not hold in an anisotropic region, where each region has a different convex distance function. In Figure 5.5, the shortest distance from $p$ to $r$ in Euclidean metric is a straight line segment. If we take the convex distance function into account, then the path from $p$ to $r$ via $q$ is shorter than the direct path.

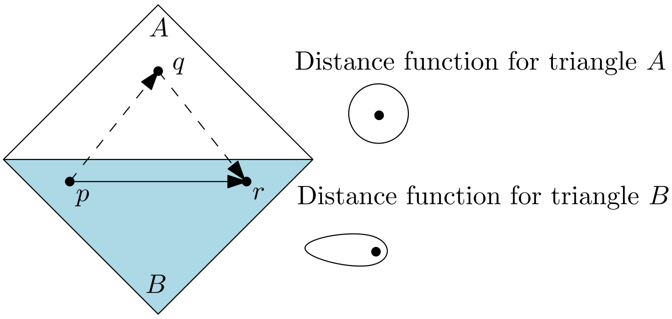

A
q
p
r
B
Distance function for triangle A
Distance function for triangle B

FIGURE 5.5: Triangle does not hold in anisotropic regions.

### Habitat suitability

In DIMO, the user can add a habitat suitability map as an input to the model and appoint regions that are not suitable for plants to grow. Suppose the starting region of the plants is convex. Will the region reached at each time $t$, $B_t$, be convex? No, not necessarily, as shown in Figure 5.6.

Suppose the filled, blue region $G$ is not suitable. If $G$ is completely contained at a certain time $t'$, then $B_t$ will continue to be convex, since the seeds can blow over $G$ without limitations, see the left example in Figure 5.6.

Suppose $G$ intersects $B_{t'}$ for a certain $t'$. Then $B_{t''}$ for $t'' \geq t'$ will also have that form and stay non-convex.

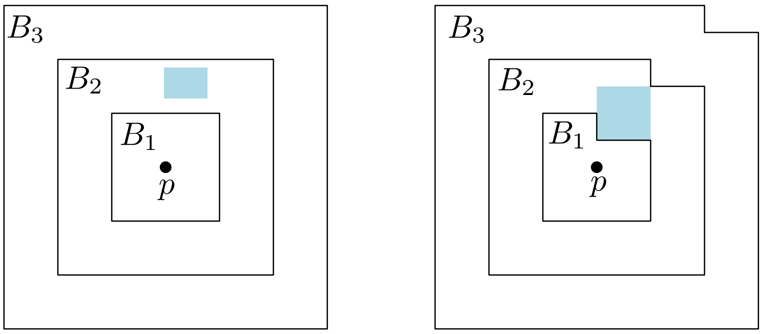


FIGURE 5.6: Here, the filled, blue region is not suitable for plant growth. The addition of a habitat suitability map can cause the reached region to become and stay non-convex.

# Chapter 6

# Planning

In this chapter, we will give an insight in some of the questions that will direct our research. Next to that, we will give an overview of our planning.

## 6.1 Approach

Our approach will be question-directed. In the coming weeks, we will think about and work on the following questions:

1. What is the complexity of the output map of the seed dispersal model with the variations given in Section 1.2?
2. Given the variations in Section 1.2, what would be a naive way to solve them?
    - What time complexity would the algorithms have?
3. Given the naive algorithms designed in subquestion 2, what adaptions could make the algorithm faster?
    - What time complexity would they have?
4. Would the approach of Arge and Staals (2017) still work if the distance measure is a convex distance function?
5. Will the addition of germination delay increase the complexity of the output map, or will it only change the values of the ages of the plants?
6. Allowing more than one convex distance function in a polygon makes it more difficult to find exact solutions. Would it be possible to make restrictions on for example the number or the complexity of a convex distance function that makes the problem easier?

## 6.2 Planning

### Till December 10 | W49

December 10, 2021 is the final date for our proposal.

### December 13 | W50

We will start on the first subquestion as written down in the previous section. What is the complexity of the output maps of the different variations?

### December 20 | W50 - W51

Holiday.

### January 3 | W1 - W2

We will continue with the question what the complexity of the output maps of the different variations is. Next to that, we will start on designing a naive algorithm to solve them.

### January 17 | W3 - W5

In these three weeks, we will focus on subquestion 2 and 3. Next to that, we will spend some time to answer subquestion 4.

### February 7 | W6 - W8

In these weeks, we will focus on incorporating a dynamic obstacle map.

### February 28 | W9

Flex time.

### March 7 | W10 - W11

Holiday.

### March 21 | W12 - W15

In these last few research weeks, we will look into the question, what obstacle removal would enlarge the reached region for a specific plant the most?

### April 19 | W16

In this week, we will specifically plan some time to write.

### April 25 | W17

Flex time.

### May 2 | W18 - W19

Since we cannot predict way in advance what research directions will the most interesting, we will leave these two weeks open for other questions and algorithms to be explored.

### May 22 | W20 - W21

Final writing phase and draft version.

### June 6 | W23 - W24

Flex time, before the final deadline June 17.

# Bibliography

Albert, A., A. Mårell, M. Picard, and C. Baltzinger (2015). “Using basic plant traits to predict ungulate seed dispersal potential”. In: *Ecography* 38, pp. 440–449. DOI: 10.1111/ecog.00709.

Andújar, D., X. Rodriguez, V. Rueda-Ayala, C. San Martín, A. Ribeiro, C. Fernández-Quintanilla, and J. Dorado (2017). “A Geometrical Model to Predict the Spatial Expansion of Sorghum Halepense in Maize Fields”. In: *Gesunde Pflanzen* 69.2, pp. 73–81. DOI: 10.1007/s10343-017-0388-6.

Arge, L. and F. Staals (2017). “Dynamic Geodesic Nearest Neighbor Searching in a Simple Polygon”.

Barequet, G., M. T. Dickerson, and M. T. Goodrich (2001). “Voronoi diagrams for convex polygon-offset distance functions”. In: *Discrete & Computational Geometry* 25.2, pp. 271–291. DOI: 10.1007/s004540010081.

Ben-moshe, B. and M. J. Katz (2004). “Computing the visibility graph of points within a polygon”. In: *In Symposium on Computational Geometry*. ACM Press, pp. 27–35.

Cheng, S., H. Na, A. Vigneron, and Y. Wang (2008). “Approximate Shortest Paths in Anisotropic Regions”. In: *SIAM Journal on Computing* 38.3, pp. 802–824. DOI: 10.1137/06067777X.

Chew, L. P. and R. L. ( Dyrsdale (1985). “Voronoi diagrams based on convex distance functions”. In: *Proceedings of the first annual symposium on Computational geometry*. Association for Computing Machinery, pp. 235–244. DOI: 10.1145/323233.323264.

Clark, J. S., C. Fastie, G. Hurtt, S. T. Jackson, C. Johnson, G. A. King, M. Lewis, J. Lynch, S. Pacala, C. Prentice, E. W. Schupp, T. Webb, and P. Wyckoff (1998). “Reid’s Paradox of Rapid Plant Migration”. In: *BioScience* 48.1, pp. 13–24. DOI: 10.2307/1313224.

“Computational Geometry” (2008). In: *Computational Geometry: Algorithms and Applications*. Ed. by M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. Springer, p. 297. DOI: 10.1007/978-3-540-77974-2_1.

Dehne, F. and R. Klein (1997). *"The Big Sweep": On the Power of the Wavefront Approach to Voronoi Diagrams*.

Dullinger, S., N. Dendoncker, A. Gattringer, M. Leitner, T. Mang, D. Moser, C. A. Mücher, C. Plutzar, M. Rounsevell, W. Willner, N. E. Zimmermann, and K. Hülber

(2015). "Modelling the effect of habitat fragmentation on climate-driven migration of European forest understorey plants". In: *Diversity and Distributions* 21.12, pp. 1375–1387. DOI: 10.1111/ddi.12370.

Gosper, C. R., C. D. Stansbury, and G. Vivian-Smith (2005). "Seed dispersal of fleshy-fruited invasive plants by birds: contributing factors and management options". In: *Diversity and Distributions* 11.6, pp. 549–558. DOI: 10.1111/j.1366-9516.2005.00195.x.

Guibas, L., J. Hershberger, D. Leven, M. Sharir, and R. Tarjan (1986). "Linear time algorithms for visibility and shortest path problems inside simple polygons". In: *Proceedings of the second annual symposium on Computational geometry - SCG '86*. the second annual symposium. ACM Press, pp. 1–13. DOI: 10.1145/10515.10516.

Heffernan, P. J. and J. S. B. Mitchell (1995). "An Optimal Algorithm for Computing Visibility in the Plane.pdf". In: 24.1, pp. 184–201.

Hershberger, J. and S. Suri (1999). "An Optimal Algorithm for Euclidean Shortest Paths in the Plane". In: *SIAM Journal on Computing* 28.6, pp. 2215–2256. DOI: 10.1137/S0097539795289604.

Joe, B. and R. B. Simpson (1987). "Corrections to Lee's visibility polygon algorithm". In: *BIT* 27.4, pp. 458–473. DOI: 10.1007/BF01937271.

Klein, R. (1988). "Abstract voronoi diagrams and their applications". In: *Computational Geometry and its Applications*. Ed. by H. Noltemeier. Springer, pp. 148–157. DOI: 10.1007/3-540-50335-8_31.

Lee, D. (1978). *Proximity and reachability in the plane.* University of Illinois at Urbana-Champaign.

Lee, D. (1983). "Visibility of a Simple Polygon". In: *Computer Vision, Graphics, and Image Processing* 22, pp. 207–221.

Lorensen, W. E. and H. E. Cline (1987). "Marching cubes: A high resolution 3D surface construction algorithm". In: *ACM SIGGRAPH* 21.4, pp. 163–169.

Ma, L. (2000). "Bisectors and Voronoi Diagrams for Convex Distance Functions". PhD thesis. University of Hagen.

McGuire, J. L., J. J. Lawler, B. H. McRae, T. A. Nuñez, and D. M. Theobald (2016). "Achieving climate connectivity in a fragmented landscape". In: *Proceedings of the National Academy of Sciences* 113.26, pp. 7195–7200. DOI: 10.1073/pnas.1602817113.

Oh, E. (2019). "Optimal Algorithm for Geodesic Nearest-point Voronoi Diagrams in Simple Polygons". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Ed. by T. M. Chan. Society for Industrial and Applied Mathematics, pp. 391–409. DOI: 10.1137/1.9781611975482.

Reid, C. (1899). *The origin of the British flora*. In collab. with Gerstein - University of Toronto. London Dulau. 238 pp.

Somerville, G. J., M. Sønderskov, S. K. Mathiassen, and H. Metcalfe (2020). “Spatial Modelling of Within-Field Weed Populations; a Review”. In: *Agronomy* 10.7. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 1044. DOI: 10.3390/agronomy10071044.

Teggi, S., S. Costanzini, G. Ghermandi, C. Malagoli, and M. Vinceti (2018). “A GIS-based atmospheric dispersion model for pollutants emitted by complex source areas”. In: *Science of The Total Environment* 610-611, pp. 175–190. DOI: 10.1016/j.scitotenv.2017.07.196.

Thuiller, W., C. Albert, M. B. Araújo, P. M. Berry, M. Cabeza, A. Guisan, T. Hickler, G. F. Midgley, J. Paterson, F. M. Schurr, M. T. Sykes, and N. E. Zimmermann (2008). “Predicting global change impacts on plant species’ distributions: Future challenges”. In: *Perspectives in Plant Ecology, Evolution and Systematics* 9.3, pp. 137–152. DOI: 10.1016/j.ppees.2007.09.004.

Wamelink, G. W. W., R. Jochem, W. Geertsema, A. H. Prins, W. A. Ozinga, J. van der Greft-van Rossum, J. Franke, A. H. Malinowska, A. H. Prins, D. C. J. van der Hoek, and C. J. Grashof-Bokdam (2014). “DIMO, a plant dispersal model”. In: *Statutory Research Tasks Unit for Nature & the Environment (WOT Natuur & Milieu)*.

Wang, H. (2021). “Shortest Paths Among Obstacles in the Plane Revisited”. In: *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Ed. by D. Marx, pp. 810–821. DOI: 10.1137/1.9781611976465.

## Figures

*Figure 4.3:* "The wavefront of the algorithm continuous Dijkstra". Original source: Figure 7 in Hershberger and Suri, 1999

*Figure 4.1:* "Marching Squares Algorithm illustration." by Nicoguaro (2015), available at https://en.wikipedia.org/wiki/Marching_squares#/media/File:Marching_squares_algorithm.svg under CC BY-SA 4.0. Full terms at https://creativecommons.org/licenses/by-sa/4.0/

*Figure 4.2:* "A convex distance function". Original source: Figure 1.1.1 in Ma, 2000.

*Figure 4.4* "Construction of bisector". Original source: Figure 2.2.1.1 in Ma, 2000.

*Figure 4.5:* "The degenerate case". Original source: Figure 2.1.3.1 in Ma, 2000.