MASTER'S THESIS

submitted within the UNIGIS MSc programme
Interfaculty Department of Geoinformatics–Z_GIS University of Salzburg

SPATIAL SIMULATION OF A SKI RESORT: AN AGENT-BASED MODEL

by

ALINA HEINRICH, BSc
105286

Advisor:
Dr. Christian Neuwirth

Toronto ON, March 2022

Alina Heinrich: *Spatial Simulation of a Ski Resort: An Agent-Based Model,* A thesis submitted in partial fulfilment of the requirements of the degree of Master of Science (Geographical Information Science & Systems) – MSc (GIS), © March 2022

ACKNOWLEDGEMENTS

This master's thesis and the learning process behind it would not have been achievable without the support of my advisor, Dr. Christian Neuwirth. First of all, he proposed to investigate this study. His prompt feedback to all my messages, his comments, and his ideas always kept me on the right path. A very special thank you, Mr. Neuwirth!

Many thanks to PowerGIS Geografische Informationssysteme GmbH, especially Maximilian Mündler, who took the time for an online meeting to discuss my ideas for this thesis and provided me with the necessary GIS data, and an essential paper, which helped me calculate a critical parameter for the model.

I am profoundly grateful to my husband, Dr. Daniel Heinrich, who has always been calm, patient, and supportive of me from the beginning of my science journey to the last draft of this thesis.

I am also truly thankful to my colleague and friend Alexander Kreisel, MSc. for proofreading the thesis.

I appreciate the curiousness of my family, colleagues, and friends about the current condition of my study process. Thus, they motivated me in this way to keep researching and creating.

Regarding the LaTeXTemplate for this thesis, many thanks and acknowledgment go to the author André Miede.

Receiving a master's degree was not in my interest until I had heard about UNIGIS Salzburg – University of Salzburg. The distance-learning format, acceptable tuition fees, and outstanding course catalog made me choose this challenge easily. Special thanks go to the UNIGIS Office team, especially Eva Reibersdorfer-Adelsberger, Regina Hatheier-Stampfl and Danijela Ristic, for the phenomenal administrative support; to the supervisors for the design and creation of impressive modules and guidance throughout the study, and last but not least, my fellow students of our "self-help group".

ABSTRACT

Austria is a country that has always been associated with outdoor activities such as hiking, climbing, and skiing. Nevertheless, the infrastructure, especially for winter seasons, requires professional effort and can lead to excessively high costs. For example, the expenses which flow into cable transport investments can be worth billions of euros. Still, expansion and replacing older infrastructure are crucial for increasing the clients' – in this case the skiers' – satisfaction. One example of such improvements is the waiting time reduction in the queue and thus the increase of the skiing time.

The cableway companies are improving their project management to widen their supply; however, keeping the financial burden as low as possible is essential. That is why a computer model could help calculate the desirable results and possible risks before starting with actual construction or even before planning it.

We built a simple ABM (agent-based model) for a small part of a ski resort. We have decided to use the ABM method as it can be a beneficial tool to simulate spatial and behavioral systems. The structure of our model is based on the "Overview, Design concepts, Details" protocol. The ABM's experiments represent how the lift speed affects the waiting time of the lift station. The model simulates a part of a ski resort with two lifts and six slopes. The agents can move along this slope network as a skier and along the lift network as a gondola. The waiting line is also implemented; this parameter will be monitored and plotted. The simulation runs 105 times. The results show some correlation between the waiting time and lift speed. We could see that increasing the lift speed reduces the waiting time. However, if the number of skiers is above a certain threshold, the waiting time stays above the skiing time. The simulation is based on a simple model and is open for various adjustments. Some propositions are listed in the last chapter of this thesis.

KURZFASSUNG

Österreich ist ein Land, das schon immer mit Freizeit-Aktivitäten wie Wandern, Klettern und Skifahren in Verbindung gebracht wurde. Nichtsdestotrotz erfordert die Infrastruktur vor allem für die Wintersaison professionellen Aufwand und kann zu enormen Kosten führen. Beispielsweise können Investitionen die in Seilbahnen fließen, Milliarden von Euro betragen. Dennoch sind der Ausbau und der Austausch älterer Infrastruktur ausschlaggebend, um die Zufriedenheit der Kunden, in diesem Fall der Skifahrer, zu sichern. Ein Beispiel für derartige Verbesserungen ist die Verkürzung der Wartezeit in der Schlange von Liftanlagen und damit die Erhöhung der Skifahrzeit.

Die Seilbahngesellschaften verbessern ihr Projektmanagement um ihr Angebot zu erweitern; es ist jedoch wesentlich, die Kosten so gering wie möglich zu halten. Deshalb könnte ein Computermodell helfen, die wünschenswerten Ergebnisse und möglichen Risiken zu berechnen, bevor mit der eigentlichen Planung oder gar dem Bau begonnen wird.

Wir haben ein einfaches ABM (agentenbasiertes Modell) für einen Teil eines Skigebiets erstellt. Unsere Entscheidung fiel auf die ABM-Methode, da sie ein nützliches Werkzeug zur Simulation von räumlichen und Verhaltenssystemen sein kann. Die Struktur unseres Modells basiert auf dem ODD-Protokoll – "Übersicht, Designkonzepte, Details". Die Experimente des ABM stellen dar, wie sich die Liftgeschwindigkeit auf die Wartezeit der Liftstation auswirkt. Das Modell simuliert ein Skigebiet mit zwei Liften und sechs Pisten. Auf diesem Pistennetz können sich die Agenten als Skifahrer und auf dem Liftnetz als Gondel bewegen. Die Warteschlange ist ebenfalls implementiert; dieser Parameter wird überwacht und aufgezeichnet. Die Simulation läuft 105 Mal. Die Ergebnisse zeigen eine gewisse Korrelation zwischen der Wartezeit und der Liftgeschwindigkeit. Wir haben feststellen können, dass eine Erhöhung der Liftgeschwindigkeit zu einer Verkürzung der Wartezeit führt. Wenn die Anzahl der Skifahrer jedoch ober-

halb einer gewissen Schwelle liegt, bleibt die Wartezeit größer als die Skifahrzeit. Die Simulation basiert auf einem einfachen Modell und ist offen für diverse Anpassungen. Einige Thesen sind im letzten Kapitel dieser Arbeit aufgeführt.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

Part I

MOTIVATION & METHODOLOGY

1 INTRODUCTION

Being in the alpine part of Austria in the winter season, one may notice many people of all ages wearing bulky, colorful clothes, carrying their skis, snowboards, or toboggans in the middle of the city, at any time of the day. That is right, and it seems like everyone in Austria tries to be on the slopes in their free time.

Austria has built its first cable transport in 1926 (Lower Austria), and today their number reached 2900. There are different types of cableway systems, such as Funicular railroads, aerial lifts, and chairlifts (WKO.at, 2020). With its 435 ski resorts (Skiresort Service International, 2021), this country is the third-largest ski market in the world (Steiger & Scott, 2020). It is a famous winter tourism destination and offers all kinds of skiing possibilities. The beauty of the mountains, the quality of the slopes, and the well-developed infrastructure make the Alps so attractive. In the pre-pandemic 2018/2019 season, there were 72.9 million overnight stays by guests in accommodations in Austria, which is even slightly higher than in previous seasons (2017/18 - 71.9 million, 2016/17 - 68.6 million), and the cash turnover of the cableway companies in the same year amounts to €1.55 billion (Mohr, 2021a). According to a survey among business owners in the tourism sector on the development of winter tourism in the 2018/19 season, about 66% of the respondents were satisfied or very satisfied with the turnover (Mohr, 2021b). Due to their geographical location, some resorts like Arlberg offer over 300km of slopes (Mohr, 2021c); others invested considerably in faster and higher gondola lifts or snowmaking facilities (Falk, 2008). The expenses which flow in cable transport investments in the period 2007/08 reach €557 million (Federal Ministry Republic of Austria, 2022).

All of the factors mentioned above would not be enough for successful cooperation if the customers, in this case, skiers (and other snow athletes), were not enjoying spending time in a ski resort. The happiness of a consumer is a complicated phenomenon, which sociologists and psychologists have been studying for decades despite the obstacles that prevent consumer researchers from achieving goals in understanding happiness (Garner et al., 2021). Project managers face many challenges to satisfy the customer in a ski resort. One of the important goals is to minimize the waiting time so that athletes can use slopes as long as possible. Nevertheless, there is no or little improvement in using tools, which could help in project planning, while decision-makers make changes in a project based on their "intuition and experience" (Poulhès & Mirial, 2017). This work aims to design a model that can simulate a ski resort and help analyze the waiting time at lifts. The results could help the professionals in project planning and keep a service quality that satisfies the customer.

1.1 CURRENT STATE OF RESEARCH

The cableway companies are improving their infrastructure and expand their offer. Academic researchers are also trying to develop and test theories that help understand how to make a ski day more enjoyable and comfortable and help companies make decisions about changing existing infrastructure to prevent unnecessary investments. There are studies about transportation, visitor flow, and queues modeling. Some of these works use an agent-based model (ABM) as a method, which is relevant for this master's thesis, and the following section examines them.

Ski resort operators in Austria are trying to make their destinations more attractive, not only by offering daily slope grooming and guaranteed snow conditions using snowmakers (Hanzer et al., 2020). Most expensive ski resorts in Austria also keep day pass prices with an average of €45,20 relatively low, compared to Switzerland (€54,54), Italy (€46,29), and Germany (€46,29) (Statista Research Department, 2016) (prices are averaged over ski resorts and converted from British Pounds to Euro, rate 1 GBP=1.19 EUR, 2022.01.14 (European Commission, 2022)). Some ski resorts offer particular ski passes to limit waiting time (Poulhès & Mirial, 2017). Salzburg and the Kitzbühel Alps, for example, are selling their SuperSkiCard online to avoid waiting time at the ticket counter

(Schmittenhöhebahn AG, 2021). In Switzerland, there is a possibility to pay for cutting the queue of the lift-chair (Poulhès & Mirial, 2017). The expansion or even replacement of older infrastructure with faster and higher capacity lifts (Poulhès & Mirial, 2017) could be more effective in reducing waiting time, making ski resorts more popular, and increasing the satisfaction of athletes (Falk, 2011).

Alvarado-Valencia et al. (2017) also believe that waiting time has an impact on customer satisfaction and can bring related costs with it. They proposed a "hybrid" model that is agent-based and discrete-event. It is based on psychological studies, individual customer behavior, and includes hidden costs associated with waiting time. Among others, they found out that a time filler such as music or advertisement can reduce dissatisfaction and make customers stay longer in a line (Alvarado-Valencia et al., 2017). That is, however, in a cable lift queue not always possible.

Liu & Chen (2019) created an ABM simulation to detect the critical factor for the consumed time in a metro station. The motional parameters of the simulation were calibrated using data gained from video analysis (Liu & Chen, 2019). Waiting time was calculated by comparing pedestrians' velocities and relative positions on the video. Several scenarios, such as guidance to the passenger on choosing ways, infrastructure position, or improvement of the latter, show that it is at least theoretically possible to significantly decrease the "overall consumed time" in a metro station. Furthermore, emergency rescue managers can use this research as a tool for their projects (Liu & Chen, 2019).

We can use an ABM also as a tool for strategies to decrease the waiting time in queues of charging stations for hybrid and electric cars. García-Magariño et al. (2018) proposed pathfinding algorithms implemented in a booking system. This strategy, if the drivers would follow it, can reduce waiting time, even if sometimes one should take a longer path to the charging station (García-Magariño et al., 2018).

Using discrete simulation and linear programming Chetouane et al. (2010) were able to plan a ski resort, simulate its operation and also evaluate waiting times and slope usage rate. The researchers made several scenarios for different crowding levels.

They found that crowding level and lift speed are essential elements for reducing waiting time. (Chetouane et al., 2010).

Poulhès & Mirial (2017) created later the "first full multi-agent model", which simulates different scenarios implementing new infrastructure instead of existing ones. The agents' behavior is adaptive on two levels; each agent can choose a new destination, take the shortest path towards the target, and decide between available slopes and the lift. Furthermore, they calculated queue as an interaction between the agents' flow and lift capacity. With the simulation, they reached the goal and could decrease waiting time by replacing a chairlift. However, this model could be improved by calibration using research on social behavior and also adding the accommodations to the network (Poulhès & Mirial, 2017).

1.2 RESEARCH: GOALS AND QUESTIONS

From the sections above, we can see that the interest in ski resorts remains high. As long as there is a possibility of going skiing, whether natural snow or artificial, people will keep their desire for winter vacation. Nevertheless, the competition between resorts is high, and decision-makers should strive to make their choices with scientific proof. Therefore, we propose building an ABM to simulate a ski resort. Our model should answer if and how the lift speed would change waiting time at the lift station. It is apparent that faster infrastructure would increase clients' satisfaction (by decreasing waiting); however, our assignment is to deliver quantitative results over assumptions. It is also interesting to see at what time the lift speed can be kept low to decrease the energy consumption and at what level it is necessary to heighten the speed so that the skiers' day remains satisfactory. With the simulations' results and further possible research based on our proposal, this work provides the data for public and private decision-makers, who could use the results as a tool to predict the consequences of further management projects, and as assistance in the planning of a ski resort before constructing them.

1.3 THESIS STRUCTURE

This thesis is organized in five chapters. After this introduction, chapter 2 "Methods" explains the methods and tools which were used to design the research. A detailed description of the model and the ODD (Overview, Design concepts, Details) protocol components are documented in chapter 3 "Model Description". This protocol contains information about initial parameterization, used data, and processes explanation. The choice of parameters and observers for the experiments is described in chapter 4 "Simulations". Furthermore, the results which are presented in plots are also recorded on these pages. Finally, chapter 6 "Summary and outlook" summarizes the results to answer the original goal of the work and to provide some ideas for the improvement of this work in further research. In the appendix the reader will find the whole code for the model (appendix A), the NetLogo interface (appendix B), and study area map (appendix C).

2

METHODS

2.1 ABM AS A METHOD FOR SPATIAL ANALYSIS

One of the goals of this master's thesis is to develop a model with different scenarios based on simple social behavior and various parameters. Agent-based modeling can be a valuable tool to reach this target as it can represent people, animals, plants (Nicholls et al., 2017), as well as their behavior and interactions (Balbi et al., 2013) in one simulation.

Agent-based modeling is object-oriented programming. The main advantage of the ABM is that the implementation (technical description, attributes, and data) is separated from the interface (input, output, and monitoring) (see figure 1), which makes the modeling user-friendly (An, 2012).

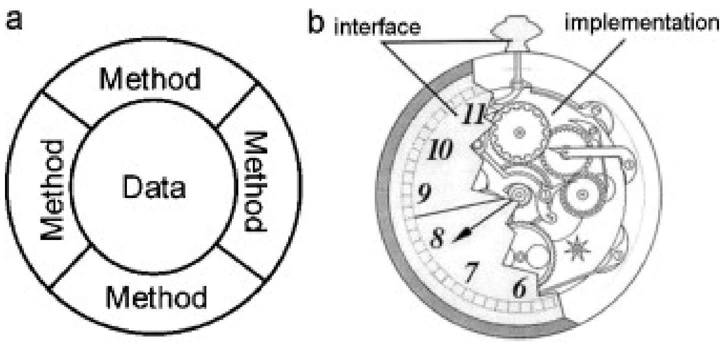

Figure 1: Interface of the clock (An, 2012)

This method was already widely used as a decision-making tool in industrial projects (Chetouane et al., 2010), in multiple complex studies such as by Poulhès & Mirial (2017) in the winter tourism industry research, by Li et al. (2015) in their work about visitors in forests, or even in bicycle accidents investigations by Wallentin & Loidl (2016) and many other research studies. Moreover, agent-based modeling allows one to simulate the actions of different agents over time and measure the resulting system behavior (Crooks et al., 2008). One of the advantages of an ABM is its ability to represent complex spatial systems in a more straightforward way (a model). Therefore, we can develop the model as a basic "if-then" model but can also have a more cognitively rich behavioral structure (Abar et al., 2017). However, it is essential to document the structure of the model correctly, as it is helpful for science research to use reproducible observations (Grimm et al., 2006).

2.2 RESEARCH TOOLS

There is a long list of software that we can use to create an agent-based simulation. The paper of Abar et al. (2017) helps to get an overview and to make a decision for specific software. Several platforms were compared based on other surveys in the paper mentioned above. Table 1 lists some of the most widely used instruments for an ABM.

Table 1: Comparison of ABM tools (extracted and shortened from Abar et al., 2017)

ABM Tool	Code Language	OS	Level	Scope/Domain
NetLogo	Scala code compilation to Java byte-code	Java Virtual Machine, Windows, Linux, Mac OS X	Simple	2D/3D simulations in social and natural sciences
Gama	GAML	Mac OS X, Windows, Linux	Moderate	spatially explicit agent-based simulations; land-use/land-planing, social, institutional, economical, ecological, biophysical systems
Swarm	Java, Objective-C	Windows, Linux, Mac OS X, Solaris, IRIX, HPUX 9/10/11	Complex	Simulations of complex adaptive systems in social, biological sciences, consumer behaviour with social network effects etc.
Mason	Java.net	Java-enabled platforms, Windows, Linux, Mac OS X	Complex	General multipurpose 2D/3D simulations (e.g. social, physical modeling, artificial intelligence, robotics, etc.)

For this work, the *NetLogo* software was chosen. This instrument has an easy-to-use interface, is friendly for a beginner, and it is well suited for medium-scaled models (Abar et al., 2017). In addition, the download and installation are also uncomplicated and do not bring any difficulties. A user can program the model in NetLogo directly; it does not require any additional software (Wilensky, 1999).

The description of the model, which follows in the next chapter, is based on the ODD standard protocol. It was proposed by (Grimm et al., 2006); it combines two elements: general considerations and mathematical description. The protocol contains three main blocks: Overview, Design Concepts, and Details with seven elements: Purpose, State variables and scales, Design concepts, Initialization, Input, and Submodels.

Furthermore, we used GIS data to create the ski resort network. The pre-processing of the geodata, such as CRS transformation, adjustment of slopes, and lift axes, was done with the help of the open-source software QGIS. In addition, the software RStudio was a helpful tool for this work to create histograms and plots. The flowchart representing the model's processes and the thesis were designed and created using LaTeX.

Part II

MODELING

3 MODEL DESCRIPTION

The model description follows the ODD protocol (Grimm et al., 2006) for the comprehensible structure and possibility to reproduce the model. We also took the updated version of the protocol into account (Grimm et al., 2010). Generally, the ABM in this work represents a ski resort and simulates skiing on the slopes and waiting in a queue at the lift station. For this thesis, we chose Fanninberg ski resort in Salzburg, Austria as a study area (see figure 2). We used geodata, which PowerGIS provided as shapefiles, to create slopes and lift networks.

Figure 2: Trail map Fanningberg (Skiresort Service International, 2022). The model includes only lifts A and B, and slopes 5, 8, 9, 11, 12, and 13

3.1 OVERVIEW

3.1.1 *Purpose*

The purpose of this model is to simulate a ski resort. Comparing the model's results, we aim to calculate ski-lift waiting time and analyze the changes depending on lift speed and the number of skiers. This comparison should show if and how the lift speed could change waiting time at the lift station.

3.1.2 *Entities, state variables, and scales*

The model's entities are `skiers`, `slopenodes`, and `liftnodes`. The agents of type `skiers` have a `target`, `speed`, `ski-speed`, `xcor` and `ycor` (location) as state variables. The `target` is the `slopenode` or the `liftnode` that the agent is currently moving towards during simulation. The `speed` is the current speed of the skier in m/s. When the skier runs down the slope, the speed changes to `ski-speed` (also in m/s). Once the skier is riding the lift, the `speed` changes to `lift-speed` which in this model is a global, user definable variable.

The `slopenodes` and `liftnodes` are created from the GIS dataset. We generate links between the nodes to build the network of slopes and lift axes. The `slopenodes` have a boolean variable `start-node?`, which informs the agents where the slope begins. The `liftnodes` have two variables: `waiting-list`, and `lift-capacity`. In the `waiting-list` we save all the waiting skiers. The `lift-capacity` is the maximum capacity in skiers per hour at the maximum lift speed of 5 m/s; we took the values 2362 people per hour for Zirbenjet and 2400 people per hour for Samsonbahn (Skiresort Service International, 2021). We calculated the spatial extent from the extent of the GIS dataset using `gis:set-transformation`. The NetLogo world is 521 by 271 patches large, and one patch is 4.956 meters wide and high. The skiers' location is not restricted to a patch but can be a fraction of a patch. NetLogo saves all numbers as double-precision floats (Wilensky, 2022a). One time step (tick) represents 0.1 seconds. Table 2 shows an overview of all entities and agents with their initial values.

Table 2: Overview of the entities and their initial value

type	entity	description	init. value
interface	num-skiers	user definable number of skiers (from 100 to 3000)	
	lift-speed	user definable lift speed (from 3 to 5 m/s)	
globals	liftaxis-dataset	for creating lift network from an axis file	
	slope-dataset	for creating slope network from a slope file	
	slope-layer-dataset	for slopes presentation only	
	currenttime	current time of the simulation	0:00
	endtime	time when the simulation ends	1:20
	resettime	time after which all plots, waiting time & rides are reset & relevant monitoring begins	0:20
	s-per-tick	time unit which is used to calculate how many seconds are in one tick	0.1
	m-per-patch	spatial unit which is used to calculate how many meters are in one patch (from widths of the world/widths of patches)	∼4.9
	valley-stations	list of stations	0
	total-waiting-ticks	number of ticks all skiers collectively have been waiting	0
	total-skiing-ticks	number of ticks all skiers collectively have been skiing	0
	rides	number of lift rides, used to calculate average waiting time	0
	lift-capacities	list of all lift capacities	2362, 2400
agentsets	breed [liftnodes liftnode]	liftnodes is the name of the breed, liftnode is a single member of the breed (Wilensky, 1999). The nodes are created from vertices of the lift data & used as a target for the skiers	
	breed [slopenodes slopenode]	slopenodes are created from the slope network vertices	
	breed [skiers skier]	all skiers	
	breed [forests forest]	trees for the presentation of a forest only; not essential for the model	
turtles-own	waiting-list	list with all waiting skiers	empty list
	lift-capacity	lift capacity at 5 m/s lift speed	one of lift-capacities
	start-node?	true for the first node of each slope, false for all other slopenodes	true or false
	target	current node the skier moves towards	random slopenode
	speed	current skier speed in m/s	ski-speed
	ski-speed	speed in m/s skiers running down the slope	random (mean 4.4m/s, st.dev. 2m/s)

3.1.3 *Process overview and scheduling*

The processes of the model are structured as in the flowchart in figure 3.

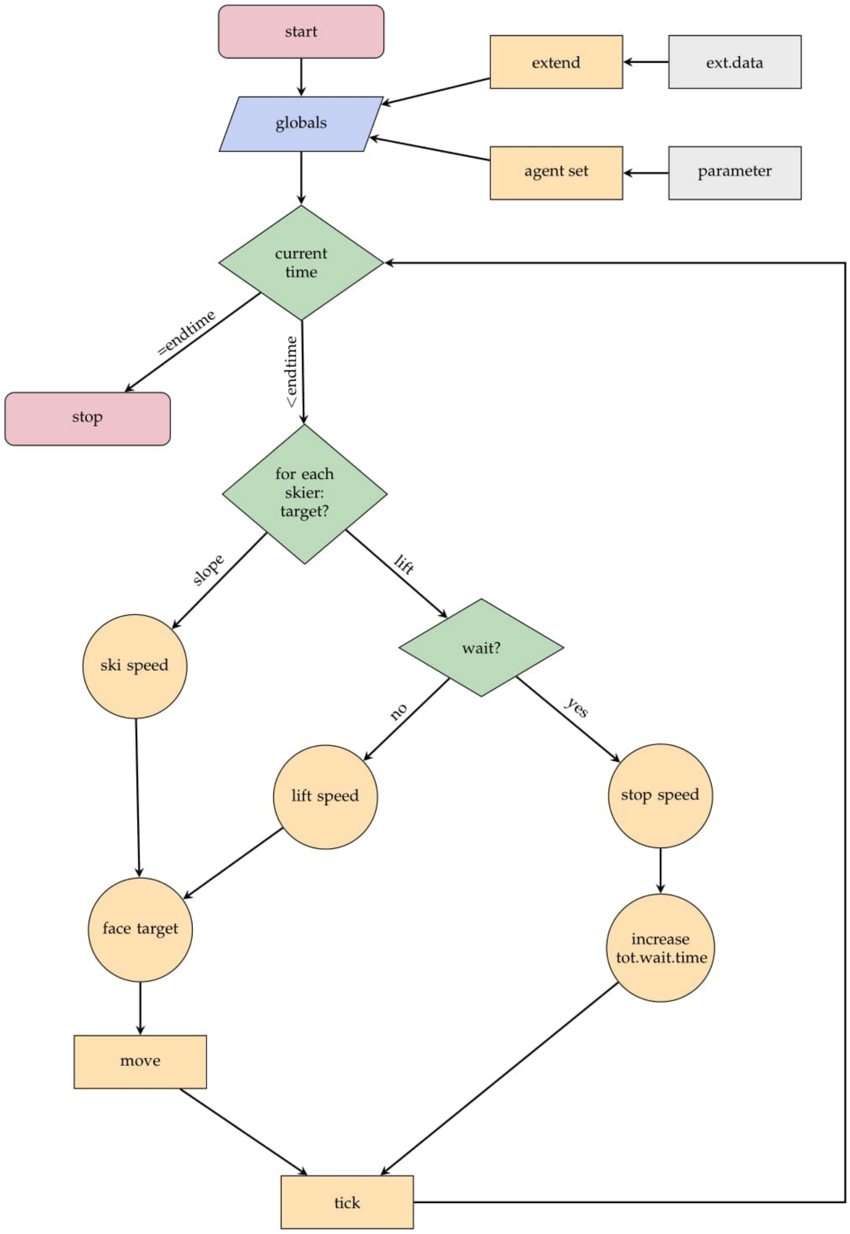

Figure 3: Processes of the ABM

Two main processes are implemented in the model: `phase-lift` and `move-skier`. The first process, `phase-lift` includes functions that are responsible for checking the departure time and sending the first six skiers from each lift's waiting list up to the top station. The movement of the skier, the second main process `move-skier`, is based on searching, facing, and moving towards the target. After the second process finishes, time increments by the discrete amount of one tick (0.1 seconds), and the processes start over. Listing 1 shows the processes of the model. The second line is a timekeeper. Lines three to eight belong to the initialization. The stop condition is saved in lines nine and ten.

Listing 1: Main processes of the model

```
1  to go
2     set currenttime time:plus currenttime s-per-tick "seconds"
3     if time:is-equal? currenttime resettime
4     [
5      set total-waiting-ticks 0
6      set rides 0
7      clear-all-plots
8     ]
9     if time:is-after? currenttime endtime
10    [stop]
11
12    phase-lift
13    move-skier
14    tick
15 end
```

3.2 DESIGN CONCEPTS

3.2.1 *Basic*

This work aims to simulate and analyze agents' waiting time depending on the lift network's capacity and the total number of agents. The model allows simulating the movement of the agents on the network. The map contains two lift axes. Six slopes connect the ends of the lift axes (top station) with the valley stations. The skiers move at different speeds along the network, and stay in a queue at the valley station. In certain intervals, up to six skiers will be removed from the waiting list and continue their movement as a gondola at the lift speed. The number of agents and lift speed is predefined and can be changed for different scenarios in the interface. Furthermore, the variety of given parameters helps analyze how and if the moving flow changes throughout the simulation.

3.2.2 *Emergence*

In the present ABM, no emergent behaviors are implemented.

3.2.3 *Adaptation*

In the present ABM, no adaptive behaviors are implemented.

3.2.4 *Objectives*

In the present ABM, no objectives are implemented; however, it is assumed that the agent's most important goal is to spend as much time as possible on a slope and as little as possible in a waiting line. This goal should be reached with changing of parameters in different scenarios.

3.2.5 *Learning*

In the present ABM, no learning behaviors are implemented.

3.2.6 *Prediction*

In the present ABM, agents are not able to predict future conditions.

3.2.7 *Sensing*

In the present ABM, no sensing is implemented.

3.2.8 *Interaction*

In the present ABM, no interactions among agents are implemented.

3.2.9 *Stochasticity*

During initialization, the skiers are placed on a random `slopenode`: `move-to one-of slopenodes`. They move with normal distributed `ski-speed` with mean 4.4 m/s and standard deviation 2 m/s: `ski-speed abs random-normal 4.4 2`. The choice of the slope to ski after riding a lift is also random: `set target one-of slopenodes in-radius 5 with [start-node? = true]`.

3.2.10 *Collectives*

Grouping, e. g. according to age, interests, or affiliation, is not implemented for this ABM. However, each gondola in the simulation has six chairs; according to this capacity, a group of up to six agents occupies the lift chair at one time.

3.2.11 *Observation*

Three plots and seven monitors are implemented in the model to observe and analyze the values during the simulation: a clock that shows the current time; the total average waiting time per skier; average waiting time per ride; total average rides per skier; the total number of waiting skiers; total skiing time and finally total average skiing time per skier. Table 3 shows an overview of collected data and their units.

Table 3: Overview of the observed data

Monitor	Unit	Plot	Unit
clock	hh:mm:ss	total-waiting-time in min	min
total average waiting time per skier in min	min	num of waiting skiers	number of people
num of waiting skiers	number of people	average waiting time in min	min
total average rides per skier	number		
average waiting time per ride in min	min		
total skiing time in min	min		
total average skiing time per skier in min	min		

3.3 DETAILS

3.3.1 *Initialization*

Initially, we transformed the CRS of the data into EPSG:31255 - MGI/Austria GK Central using QGIS. The slopes, which were given as a polygon file, were converted to a polyline. Furthermore, the data was adjusted to only contain relevant lifts – Zirbenjet, Samsonbahn – and slopes – Weißpriacher-Steilhang, Radersteilhang, Familienabfahrt, Stöffei-Steilhang, Schmiedabfahrt, and Liebestal. Therefore, the gradient and difficulty level of the slopes will not be taken into account during simulation.

In the beginning, the global variables, spatial extent, network (from geodata files), and agents (from the model-defined parameters) will be initialized. The `set-up` starts the model, which updates all monitors and plots. The first twenty minutes or 1200 ticks are reserved for the initialization of the model, and only one hour of a day will be represented. During the initialization period, the agents already move down the slopes to the lift stations, where they populate the queues and then ride the lifts up to the top stations. Thus, they distribute themselves in the whole ski resort.

The spatial extent, the network, the lift capacity, and the speed of the skiers remain the same throughout the simulation. The user can change the number of agents `num-skiers` and the `lift-speed` for each run (one hour and twenty minutes). The number of agents can be set between 100 and 3000; furthermore, the runs will proceed in 100 steps to receive more data for further analysis. As the initial value for the lift speed, we took the actual 5 m/s (Skiresort Service International, 2021), and we decreased it to 3 m/s in 0.5 steps for further scenarios.

Due to a lack of data, the skiers' speed was calculated as proposed from Von Allmen (1976). It is assumed that the speed depends on different skiing ability levels (see table 4). According to this proposal, the expected speed is corrected by dividing by four since experienced agents ski cross-slope and beginners need more pauses. The standard deviation and mean were calculated from the corrected speed values from table 4. Hence, each skiers `ski-speed` is randomly drawn from a normal distribution with mean 4.4 m/s and standard deviation 2 m/s (see listing 2). The histogram in figure 4 represents the distribution of the agents' speed. The initial values of all entities are shown in table 2 on page 17.

Listing 2: Initial setup for skier

```
to setup-skier
   set shape "skier"
```

```
    set size 5
    set target one-of slopenodes
    move-to target
    set ski-speed abs random-normal 4.4 2
    set speed ski-speed
end
```

Table 4: Experience level, expected speed and corrected speed (extracted and shortened from Von Allmen, 1976)

level	exp. speed (m/s)	cor. speed (m/s)
First practice	8	2
Beginner	10	2.5
Low Intermediate	12.5	3.2
Intermediate	16	4
Advanced Intermediate	20	5
Advanced	25	6.3
Expert	31.5	8

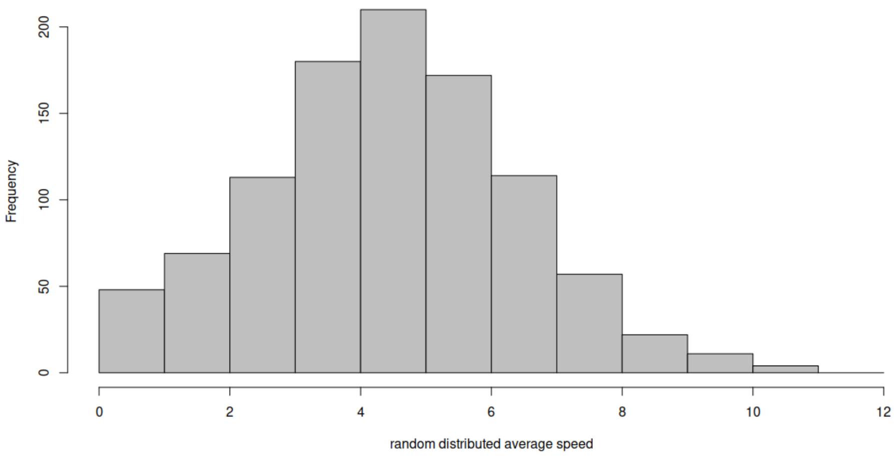

Figure 4: Random distributed corrected speed for 1000 skiers

3.3.2 *Input data*

The model does not use any external data which could change over time. However, some geodata was needed to recreate the network. The shapefiles “slopes” and “axes” were provided by Mündler (2021). The data was first converted (CRS) and preprocessed in QGIS and loaded in NetLogo using `extension [gis]`. The world's extent was adjusted to the extent of the geodata.

3.3.3 *Submodels*

The processes that were mentioned in section 3.1.3 are represented in two submodels: `phase-lift` and `move-skier`.

`phase-lift`

At every time step, the total waiting time is raised by the number of waiting skiers (see line 3 in listing 3). Then, each lift checks if it is time for a gondola to leave the station by evaluating `if ticks mod chair-period = 0 []`. Every lift has its own `chair-period` defined by

$$\text{chair-period} = \frac{\text{chair-capacity}}{(\text{current-lift-capacity}/3600\,{}^{s}/_{h}) \cdot \text{s-per-tick}}. \quad (1)$$

In those intervals, the first waiting skiers (up to six) will be removed from the waiting list. Then, they move as a lift chair with predefined `lift-speed`.

Listing 3: Submodel `phase-lift`

```
1  to phase-lift
2    ask turtle-set valley-stations [
3      set total-waiting-ticks total-waiting-ticks + length waiting-list
4      let current-lift-capacity lift-capacity * lift-speed / 5
5      let chair-capacity 6
6      let chair-period int(chair-capacity / ( current-lift-capacity / 60 / 60
          ) / s-per-tick)
7      if ticks mod chair-period = 0 [
8        repeat chair-capacity [
9          if not empty? waiting-list [
10           ask first waiting-list [
11             set speed lift-speed
12             set rides rides + 1
13             set shape "pentagon"
14             set color black
15           ]
16           set waiting-list butfirst waiting-list
17         ]
18       ]
19     ]
20   ]
```

```
end
```

move-skier

The agents are only moving along the network, searching for and facing their respective targets. Each `target` is either one of the `liftnodes` or one of the `slopenodes`. Depending on the current part of the network (slope or lift), the `speed` of the agent is different. We should note that the moving direction is the same as that of each segment to the vertex, which should be examined during pre-processing using QGIS.

The movement transpires in the following way: in each time step the agent turns towards its `target` (`face target`) and moves forward at its current `speed` (`forward speed / m-per-patch * s-per-tick`). If the distance to the `target` is less than the distance to be traveled in one time step (`if distance target < speed / m-per-patch * s-per-tick []`), which means the current `target` is reached, the skier chooses the next node as its new `target` (`target one-of [out-link-neighbors] of target`). After arriving at the last `slopenode` the agent randomly chooses one of the `liftnodes` within five patches (about 25 meters) radius. Afterwards, the agent adds itself to the `waiting-list` and its `speed` is set to zero. When the skier reaches the last `liftnode`, it searches for all start slope nodes within five patches (about 25 meters) radius and randomly chooses one of these nodes. Furthermore, the skier's `speed` is set to `ski-speed`. See complete code for this submodel in listing 4.

Listing 4: Submodel `move-skier`

```
to move-skier
   ask skiers [
      let current_node_type [breed] of target
      if distance target < speed / m-per-patch * s-per-tick [
         set target one-of [out-link-neighbors] of target
      ]

      if target = nobody [
         if current_node_type = slopenodes [
            set target one-of liftnodes in-radius 5
            ask target [
               set waiting-list lput myself waiting-list
            ]
            set speed 0
```

```
        ]

        if current_node_type = liftnodes [
          set target one-of slopenodes in-radius 5 with [start-node? = true]
          set speed ski-speed
          set shape "skier"
          set color one-of base-colors
        ]
      ]
      if speed = ski-speed [
        set total-skiing-ticks total-skiing-ticks + 1
      ]
      face target
      forward speed / m-per-patch * s-per-tick
    ]
end
```

4 SIMULATIONS

Different scenarios were created for the ABM in this work with the tool BehaviorSpace in NetLogo. It allows the user to select all required observers and parameters for the simulation and run all experiments simultaneously. In this way, we could perform 105 runs and visualize different scenarios from the table or spreadsheet, which was automatically generated as an output.

4.1 SIMULATION TO FIND PROPER RESET TIME

4.1.1 *Parameters*

To find out the proper time to start the model, we first run four experiments with two variables: number of skiers 1000 and 3000 (`["num-skiers" 1000 3000]`), and lift speed 3 m/s and 5 m/s (`["lift-speed" 3 5]`). Those numbers represent meaningful minimum and maximum values. As observers we took the total number of waiting skiers (`count skiers with [ speed = 0 ]`), the number of waiting skiers at one of the stations (`[length waiting-list] of item 0 valley-stations`), and the number of waiting skiers at the other station (`[length waiting-list] of item 1 valley-stations`).

As it was already mentioned in section 3.2.9, the agents are randomly located in the resort. To be reasonably sure that the average numbers of waiting skiers, i.e., the output from the simulation to find a proper reset time, are accurate, we also calculated their confidence intervals. Therefore, we repeated the previous simulation with the same variables 100 times. However, we only took the total number of waiting skiers (`count skiers with [ speed = 0 ]`) as observers. Using RStudio, we calculated the confidence interval with a 95% confidence level (function CI) of the number of waiting skiers for each time step (tick).

4.1.2 *Output*

At the beginning of the simulation, the skiers are randomly located only on the slopes, which means that there is no agent in the waiting line or in the lift. The skiers then arrive at the station faster than they are leaving. For that reason, we decided to let the simulation run for a certain time until the number of waiting skiers has reached a steady state. In the following, we present plotted results. We extracted them from the spreadsheet which BehaviorSpace created (see table 5). The spreadsheet contains all necessary information for the model, like header, version, name, date, dimensions, and finally, the list with all measurements for all simulation steps (Wilensky, 2022b).

Table 5: Output, created from BehaviorSpace. The table contains the waiting list length for each time step. This is used to determine the proper reset time. Here, only the first five of 192004 entries are presented.

BehaviorSpace results (NetLogo 6.2.0)						
slopesandaxis1.1.nlogo						
resetTimeExtreme						
01/11/2022 20:53:22:329 -0500						
min-pxcor	max-pxcor	min-pycor	max-pycor			
-260	260	-135	135			
[run number]	num-skiers	lift-speed	[step]	count skiers with [speed = 0]	[length waiting-list] of item 0 valley-stations	[length waiting-list] of item 1 valley-stations
1	1000	3	0	0	0	0
1	1000	3	1	39	22	17
1	1000	3	2	39	22	17
1	1000	3	3	39	22	17
1	1000	3	4	39	22	17
...						

The plot in figure 5 shows the results of four model runs. The runs are separated by color. The highest line of each color on the graphic represents the total number of waiting skiers (both stations); the two lower lines of each color visualize the number of waiting skiers at one and the other station, respectively. According to this result, we decided that the proper time to reset the observers and start monitoring from the beginning to receive correct values is after 12000 ticks (twenty minutes).

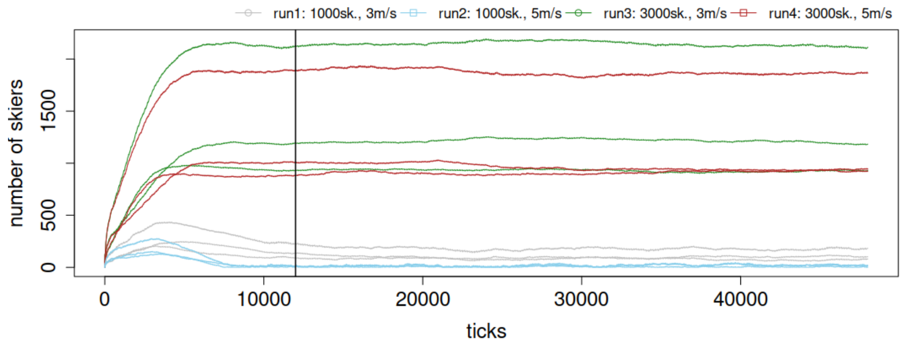

Figure 5: The highest line of each color on the graphic represents the total number of waiting skiers (both stations); the two lower lines of each color visualize the number of waiting skiers at one and the other station, respectively. After 12000 ticks all waiting lists have reached a steady state.

The 95% confidence interval of the total number of waiting skiers was computed for each of the four experiments and at each time step. The lines in figure 6 represent the total number of waiting skiers at both stations, averaged over 100 runs. The confidence intervals are too small to be discerned on this plot. For that reason, we calculated the average upper and lower values for each experiment to see the approximate differences. From table 6, we can see that the average size of confidence intervals varies between 4.47 and 6.86 (skiers).

Table 6: Average lower and upper values for each number of waiting skiers

averaged	**1000sk., 3m/s**	**1000sk., 5m/s**	**3000sk., 3m/s**	**3000sk., 5m/s**
upper	227.10	49.53	2079.91	1818.94
lower	222.54	45.06	2074.21	1812.08
difference	4.56	4.47	5.70	6.86

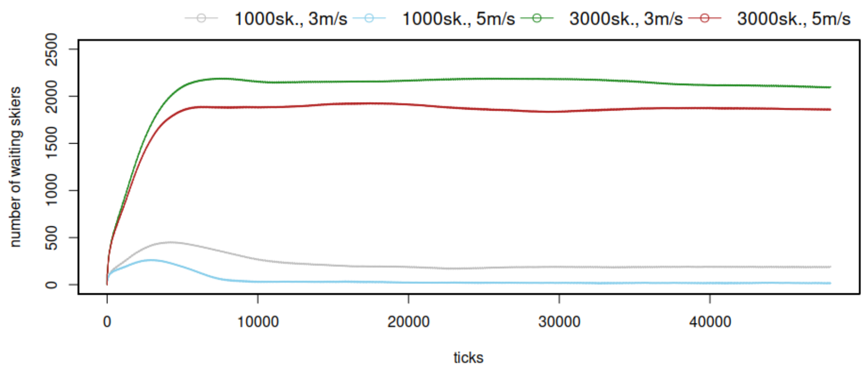

Figure 6: The graph represents the total number of waiting skiers at both stations, averaged over 100 runs. The gray line visualizes the number of waiting skiers out of 1000 skiers in the resort at a lift speed of 3 m/s; the blue line is for 1000 skiers and 5 m/s; the red line is for 3000 skiers and 3 m/s; and finally, green line is for 3000 skiers at 5 m/s. Confidence intervals are smaller than the line thickness.

4.2 SIMULATION TO DETERMINE WAITING TIME

4.2.1 *Parameters*

For our main experiments, we added the following variables to the BehaviorSpace: 100 to 3000 skiers in 100 steps (`["num-skiers" [100 100 3000]]`) and lift speed from 3 m/s to 5 m/s in 0.5 steps (`["lift-speed" [3 0.5 5]]`). For the observers, we selected the total average rides per skier (`rides / num-skiers`), average waiting time per ride (`total-waiting-ticks / rides * s-per-tick / 60`), total average waiting time per skier (`( total-waiting-ticks * s-per-tick / 60 ) / num-skiers`), and total average skiing time per skier (`( total-skiing-ticks * s-per-tick / 60 ) / num-skiers`).

4.2.2 *Output*

After 105 runs for the main experiments, we again used table output from BehaviorSpace (see table 7) for further analyses.

Then, using RStudio, we created several plots that help analyze the waiting and skiing time versus the number of skiers and lift speed, respectively. We did not need to change any parameters

Table 7: Output, created from BehaviorSpace. The table contains information to calculate the reset time for the simulation to determine waiting time. Here only first five of 157 entries are presented

BehaviorSpace results (NetLogo 6.2.0)							
slopesandaxis1.1.nlogo							
szenarios100to3000							
01/14/2022 14:33:35:616 -0500							
min-pxcor	max-pxcor	min-pycor	max-pycor				
-260	260	-135	135				
[run number]	num-skiers	lift-speed	[step]	(total-waiting-ticks * s-per-tick / 60) / num-skiers	rides / num-skiers	total-waiting-ticks / rides * s-per-tick / 60	(total-skiing-ticks * s-per-tick / 60) / num-skiers
1	100	3	48000	0.4751	3.64	0.130521978021978	37.38235
2	100	4	48000	0.362266666666667	3.84	0.0943402777777778	46.8715166666667
3	100	4.5	48000	0.399766666666667	4.58	0.0872852983988355	45.49305
4	100	3.5	48000	0.396116666666667	3.64	0.10882326007326	41.9992333333333
5	100	5	48000	0.36975	4.74	0.0780063291139241	47.59955
...							

or make any new model runs, thanks to the BehaviorSpace that could generate one output with all information required for the analysis.

First, we created an overview of lift speed, waiting time, and skiing time against the number of skiers (see figure 7). Looking at this plot, we can already recognize a non-linear correlation between two main parameters: waiting time and the number of skiers. Furthermore, we can see that from about 1000 skiers, the skiing time falls while waiting time increases constantly. At a lift speed of 3 m/s (4 m/s) [5 m/s], skiers spend more time waiting in line than skiing when there are more than 1300 (1700) [2100] people in the resort.

Let us take a closer look at the waiting and skiing time vs. the number of skiers for different lift speeds (see figure 8). These five plots show that for up to 1000 skiers in a resort, there is no need to speed up the lifts considering there is no waiting time in this

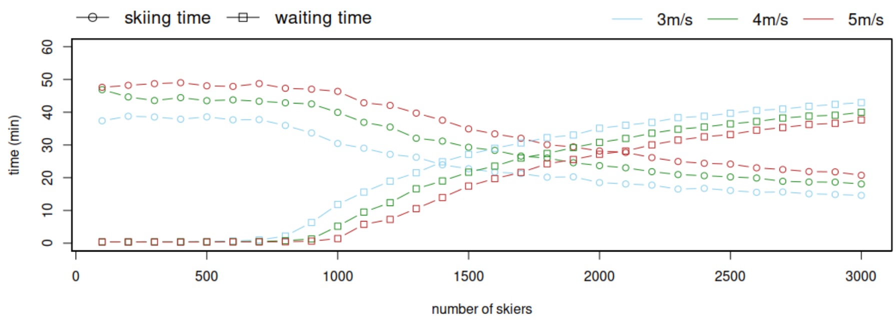

Figure 7: Total average waiting and skiing time per skier vs. number of skiers at different lift speed

case. However, the waiting time increases rapidly between 1000 and 2000 skiers from 0 to ca. 30 minutes, yet between 2000 and 3000 skiers it increases only by about another 10 minutes. At a lift speed of 3 m/s (4 m/s) [5 m/s], the skiing time remains stable until 700 (900) [1000] skiers, and then decreases from about 38 to 13 (43 to 16) [48 to 20] minutes.

It is also interesting to observe these two parameters under another perspective: figure 9 shows the same waiting and skiing time but plotted against the lift speed. These plots also show how the waiting time decreases from 10 to almost 0 min at 1000 skiers, from 35 to 25 min at 2000 skiers, and from 42 to 38 min at 3000 skiers; however, we can also see how skiing time increases with higher lift speed. Considering that the simulated ski resort is small, there is almost no change in waiting time with less than 1000 skiers.

We were also interested in knowing how the average waiting time per ride has changed during the simulation and how it depends on the number of skiers and lift speed. These results are interesting because while the waiting time per ride vs. lift speed changes non-linear (see figure 10), the same parameter as a function of the number of skiers changes in a linear way after reaching a speed-dependent threshold (see figure 11): first, it is zero till 800, 900, and 1000 skiers at 3 m/s, 4 m/s, and 5 m/s, respectively, but then it grows linearly to about 45, 32, and 25 minutes.

We were also able to evaluate how many lift rides a skier can take in one hour. We notice that there is again a dependency between the number of rides and increasing lift speed and number of skiers, but more interesting is that skiers can ride a maximum of 2 (2000 skiers and 5 m/s) or 1.5 (3000 skiers and 5 m/s) times, however, if there are 1500 skiers the number reaches 3 rides per hour and for 1000 skiers the number of rides more than doubles to 4.5 (at 5 m/s) rides per hour (see figure 12). Furthermore, the correlation between the number of rides and the lift speed seems to be linear.

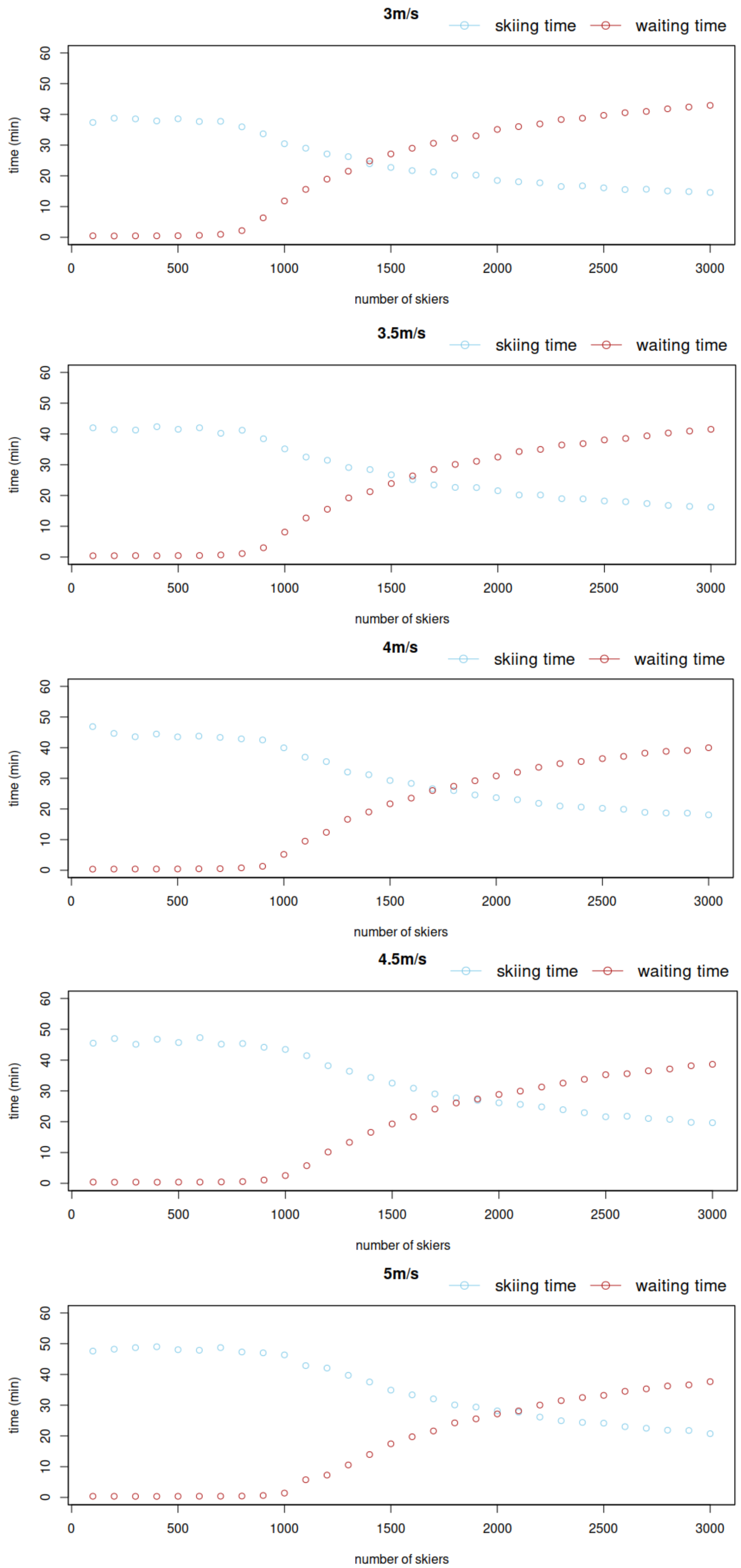

Figure 8: Total average waiting and skiing time per skier vs. number of skiers for each lift speed

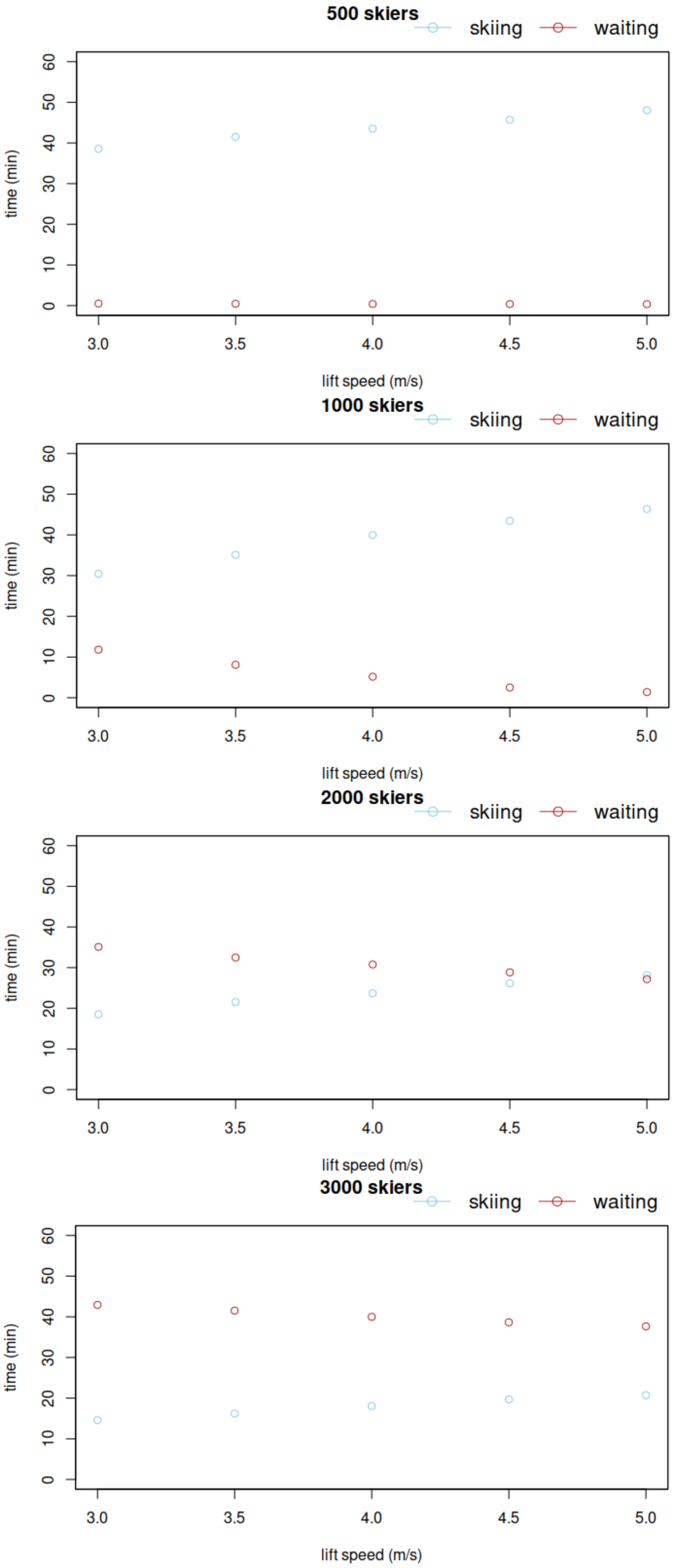

Figure 9: Total average waiting and skiing time per skier vs. lift speed for different numbers of skiers

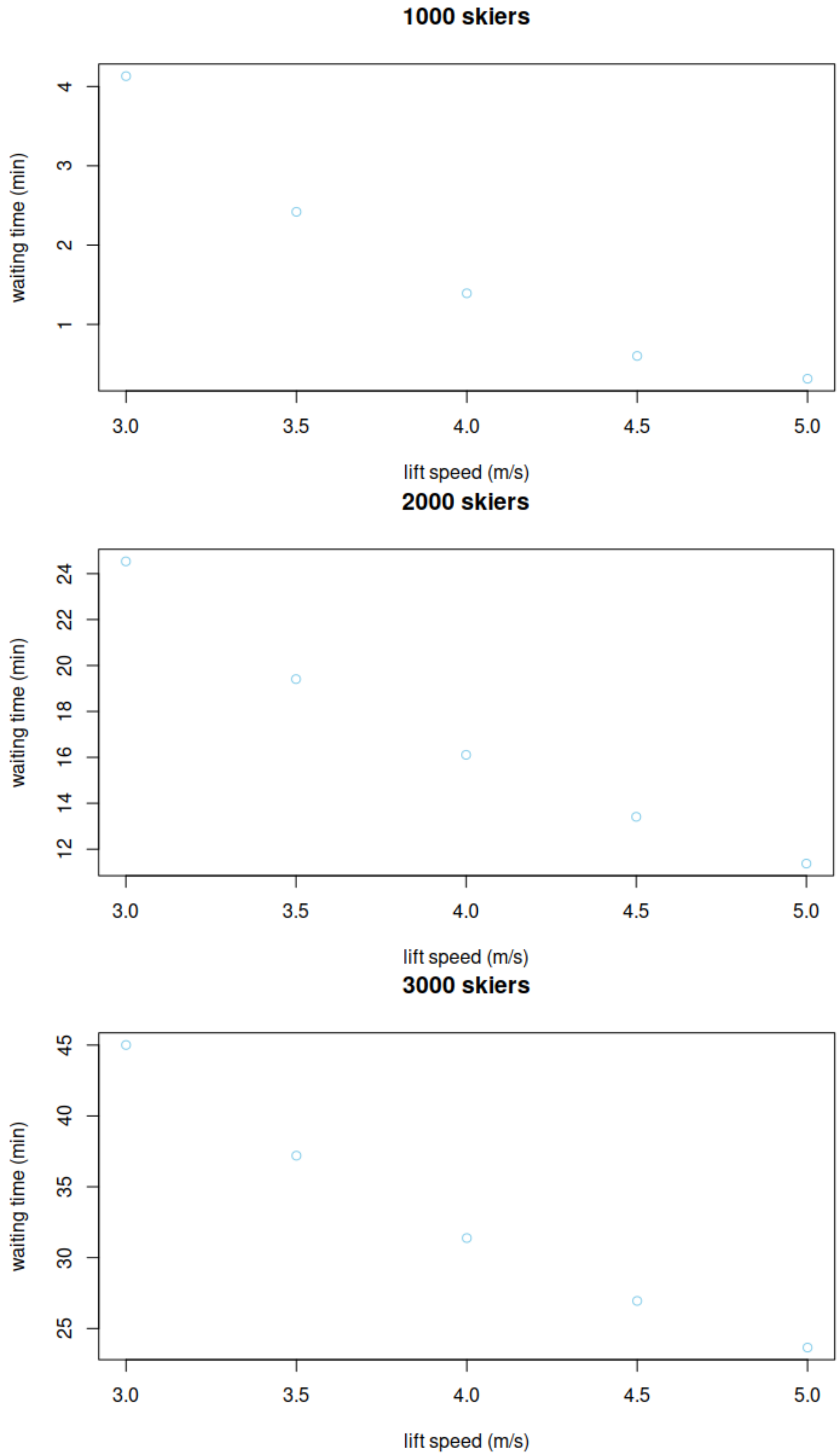

Figure 10: Total average waiting time per ride vs. lift speed for different numbers of skiers

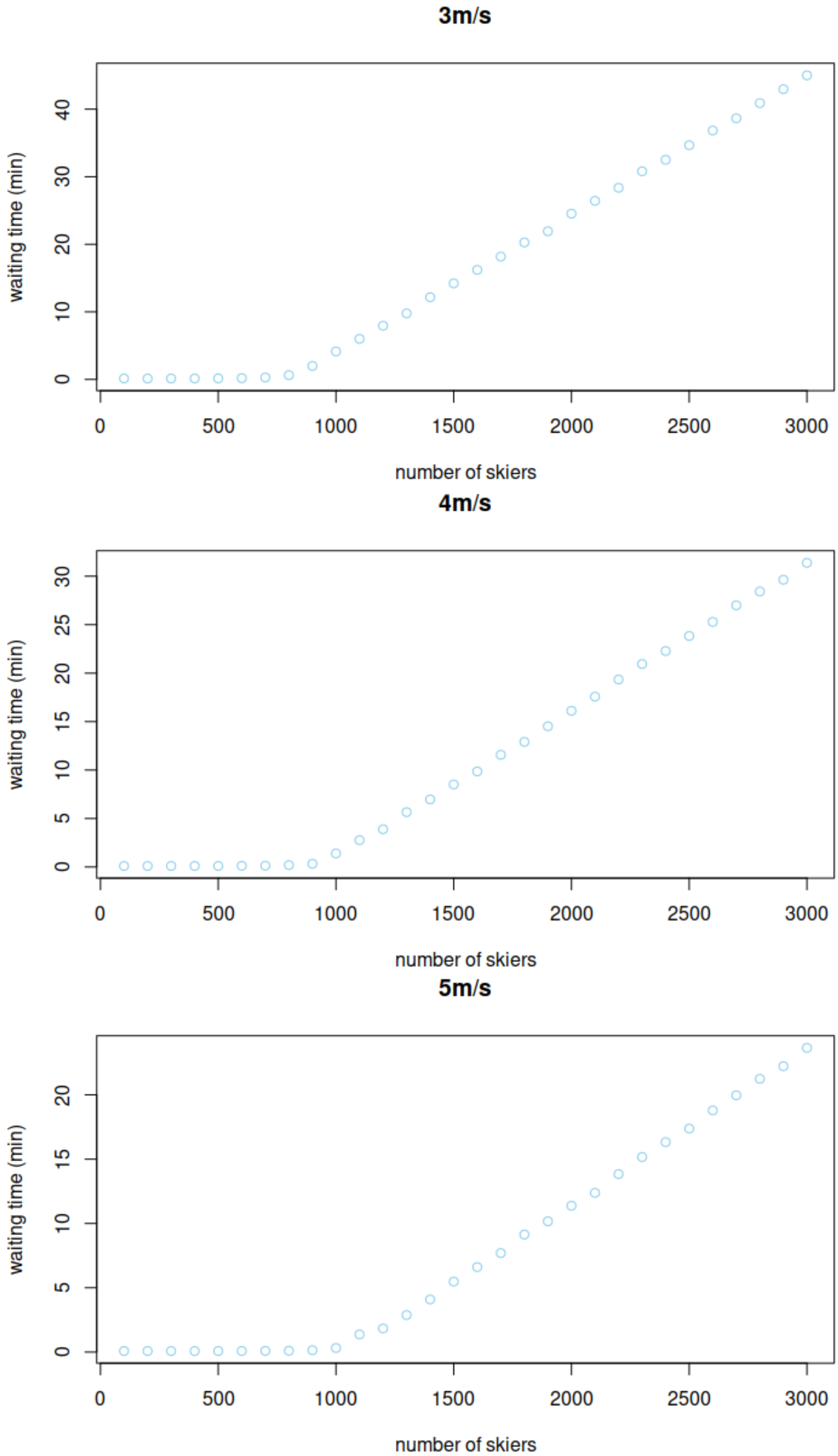

Figure 11: Total average waiting time per ride vs. number of skiers for different numbers of skiers

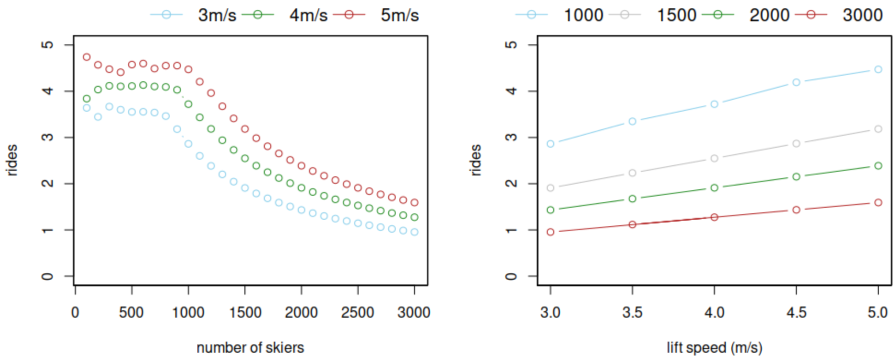

Figure 12: Total average number of rides per skier vs. lift speed and number of skiers

5 DISCUSSION

about the ***methods*** To create a ski resort simulation, we decided to use agent-based modeling. This method successfully helped us design spatial simulations with different parameters and observers. The software NetLogo turned out to be clear and user-friendly. The BehaviorSpace was a beneficial tool, too. Its fast and easy-created output gave us all the necessary data to analyze the chosen observers. However, we found the syntax of the code a little bit underdeveloped. We were missing some error messages, highlighting, and suggestions.

about the ***simulation to find proper reset time*** At the beginning of this thesis, we decided to simulate only one hour in the middle of the skiing day. Therefore, we faced the problem of choosing all agents' initial locations: on the slope, in the lift, or in the waiting line. To avoid unproven assumptions, we randomly located all the agents on the slopes only and created no skiers in the lifts or waiting lines. However, this initial state underestimates the lengths of waiting lines at the beginning of each run. For that reason, we ran the simulation for a specific time, during which the agents started to move down the slope, fill the stations, and take the lifts. At the beginning of the simulation, more skiers were arriving at the stations than leaving, but after several simulated minutes, the situation on slopes and at lifts became closer to reality. It is essential to note that both lifts have very similar capacities; otherwise, agents' random choice of the slope would lead to overcrowding of the lift with lower capacity.

about the ***simulation to determine waiting time*** With the results of these simulations, we can answer this thesis's question: "if and how the lift speed would change waiting time at the lift station". From the plots in chapter 4, we could already determine *if* the lift speed changes the waiting time. The latter is indeed changing depending on lift speed; however, it is worth mentioning that in our resort, the waiting time does not change and remains close to zero with up to 700 skiers no matter how high the lift speed is. The non-linear trend answered the question "how" the waiting time changes. Furthermore, we can notice that the waiting time always displays a rapid change from a certain level, like for example in figure 11 for the total average waiting time per ride for different numbers of skiers. We can see an apparent kink in the plotted data. The sudden and steady increase of the waiting time means that the model shows a high sensitivity at this point. The reason is that with up to 700 (900) [1000] people in the resort at 3 m/s (4 m/s) [5 m/s] lift speed, the gondolas are not fully occupied. Above that threshold, the lifts are running at capacity, and any additional skiers populate the waiting lines. Furthermore, the plotted results show us a level from which the waiting time is higher than the skiing time (see figure 8). First of all, it is a sign that a particular amount of skiers is too high for the infrastructure capacities in this ski resort. Therefore, a better adjustment of the number of agents and the size of the ski resort to each other would probably lead to more balanced results. Moreover, after the waiting/skiing time crossing, the waiting time increases only slowly.

Part III

CONCLUSION

6 SUMMARY AND OUTLOOK

6.1 SUMMARY

At the beginning of this research, we have dealt with the question of whether different lift speeds could influence the waiting time of skiers at a lift station in a ski resort. We built an ABM using NetLogo that simulates a ski resort containing a slopes and lifts network and agents. We designed two main processes: movements of lifts and skiers. We also implemented such parameters as lift speed and monitored the waiting time. The description of the model's elements was designed based on the ODD protocol.

The output was created by BehaviorSpace, which allowed us to consider several scenarios with different numbers of skiers and various lift speeds. We used RStudio to visualize the results. In this way, we could find the correlation between the observers. The results shown in the plots in section 4.2.2 demonstrate apparently that the waiting time scales down when increasing the lift speed. However, the trend is non-linear.

Summing up, we could say that agent-based modeling is a handy tool to create a spatial simulation with social behavior. We designed the model to represent a ski resort with moving skiers and infrastructure. All methods and software we chose helped us reach the goal of this thesis.

6.2 OUTLOOK

For the purpose of this master's thesis, we created a straightforward model, and there is room for several approaches. First of all, the ability to change the number of such entities as stations, slopes, and lifts or even their replacement could be implemented. The model could represent the whole ski resort and provide a more realistic picture of the chosen area. Furthermore, it could help to prevent levels from where the waiting time becomes higher than skiing time. The results could be influenced by including other infrastructures like restaurants and accommodations; in this case, waiting time could change depending on how many people are on a break.

An exciting aspect could also be how much the lift speed influences the energy consumption; using technical data from the manufacturers, one could calculate the dependency and quantitatively prove our assumptions. Finally, a possible extension in an agent's behavior could be the memory and ability to make decisions rather than determine the next slope to ski randomly.

Finally, more data, such as ski-pass access, could make validation possible, which is interesting to compare our theoretical assumptions.

Part IV

APPENDIX

A

NETLOGO SCRIPT

Listing 5: Complete code for the agent-based model in a ski resort

```
;this model is to simulate waiting time of skiers in a skiresort.
;it was created for a skiresort with two lifts (six chairs each) but
;can be expanded as required
;autor: Alina Heinrich
;date: 2021-2022
extensions [ gis time]

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~DEFINE GLOBAR VARIABLE FOR ALL AGENTS~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

globals [
   liftaxis-dataset
   slope-dataset
   slope-layer-dataset
   currenttime
   endtime
   resettime   ;after this time monitoring begins
   s-per-tick   ;seconds per tick
   m-per-patch   ;meter per patch
   valley-stations   ;list of all valley stations
   total-waiting-ticks
   total-skiing-ticks
   rides
   lift-capacities
]

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~CREATE AGENTSETS~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

breed [liftnodes liftnode]
breed [slopenodes slopenode]
breed [skiers skier]
breed [forests forest]

liftnodes-own [
   waiting-list   ;list with waiting skiers
   lift-capacity
]

slopenodes-own [
   start-node?
]
```

```
skiers-own [
   target   ;skier moves to
   speed   ;current speed in m/sec
   ski-speed   ;down the slope
]

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~PROCEDURE STEPS~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to set-up
   clear-all
   setup-globals

   ask patches [set pcolor white]
   ;~~load geo data
   set liftaxis-dataset gis:load-dataset "axes.shp"
   set slope-dataset gis:load-dataset "slopelines.shp"
   set slope-layer-dataset gis:load-dataset "slope_layer.shp"   ;for slopes
        presentation only
   let env (gis:envelope-union-of(gis:envelope-of liftaxis-dataset)
   gis:envelope-of slope-dataset)
   gis:set-transformation env (list (min-pxcor + 4) (max-pxcor - 4) (min-pycor + 4) (
        max-pycor - 4))
   ask patches gis:intersecting slope-layer-dataset  [
      set pcolor blue + 4
   ]
   ;~~
   ;~~create forest
   create-forests 350
   [
      setxy random-pxcor random-pycor
      while [any? patches with [pcolor != white] in-radius 6]
      [
         setxy random-pxcor random-pycor
      ]
      set shape "tree"
      set color green + 4
      set size 10
   ]
   ;~~

   ;~~convert patches in meter
   let extend-in-m item 1 gis:world-envelope - item 0 gis:world-envelope   ;how wide
        is the world in meter
   let extend-in-p max-pxcor - min-pxcor   ;how wide is the world in patches
   set m-per-patch extend-in-m / extend-in-p
   ;~~

   create-liftnetwork
   create-slopenetwork

   ;create user defined number of skiers
   create-skiers num-skiers [
      setup-skier
```

```
   ]
   ;~~

   reset-ticks
end

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~DEFINE VALUES FOR GLOBAL VARIABLES~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to setup-globals
   set currenttime time:create "2000/01/01 0:00"
   set endtime time:create "2000/01/01 1:20"
   set resettime time:create "2000/01/01 0:20"
   set s-per-tick 0.1
   set valley-stations []
   set rides 0   ;value null causes "divide by zero"  error at beginning of simulation
   set lift-capacities list 2362 2400
end

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~CREATE NETWORK FROM LIFTDATA~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to create-liftnetwork
   foreach gis:feature-list-of liftaxis-dataset [   ;loop over the list of lifts (each is a polyline)
      liftfeature ->
      let nodelist []
      let previous-node nobody
      foreach gis:vertex-lists-of liftfeature [   ;loop over the list of each segment /pair of coordinates
         liftsegment ->
         foreach liftsegment [   ;loop over the list of vertices
            liftvertex -> let location gis:location-of liftvertex
            create-liftnodes 1 [   ;create 1 node at location of the vertex
               set xcor item 0 location
               set ycor item 1 location
               set nodelist lput liftnode who nodelist
               set hidden? true
               ifelse previous-node = nobody [
               ][
                  create-link-to previous-node   ;create link between nodes
               ]
               set previous-node self
            ]
         ]
         ask previous-node [
            set waiting-list []   ;set waiting list on the last node, which is the valley station
            set valley-stations lput self valley-stations   ;add valley station to list of all valley stations (for plotting)
            set lift-capacity first lift-capacities   ;save first value of capacities list
```

```
            set lift-capacities but-first lift-capacities   ;delete first value from capacities list
         ]
      ]
      ask links [set color blue]
   ]
end

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~CREATE NETWORK FROM SLOPEDATA~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to create-slopenetwork
   foreach gis:feature-list-of slope-dataset [
      slopefeature ->
      let previous-node nobody
      foreach gis:vertex-lists-of slopefeature [
         slopesegment ->
         foreach slopesegment [
            slopevertex -> let location gis:location-of slopevertex
            create-slopenodes 1 [
               set xcor item 0 location
               set ycor item 1 location
               set hidden? true
               ifelse previous-node = nobody [
                  set start-node? true
               ][
                  create-link-from previous-node [hide-link]
               ]
               set previous-node self
            ]
         ]
      ]
   ]
end

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~PROCEDURE TO SETUP SKIER~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to setup-skier
   set shape "skier"
   set size 5
   set target one-of slopenodes   ;slopenode is a target to move to for skier
   move-to target
   set ski-speed abs random-normal 4.4 2   ;skier moves with the speed 4.4m/s with 2m/s stand. dev.
   set speed ski-speed
end
;~~

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~PROCEDURE TO RUN THE MODEL~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to go
```

```
  ;~~ set time of the model. It runs one hour and 30min.
  ;first 30 min are to setup the model, after 30 min monitoring begins
  set currenttime time:plus currenttime s-per-tick "seconds"
  if time:is-equal? currenttime resettime
  [
    set total-waiting-ticks 0
    set rides 0
    clear-all-plots
  ]
  if time:is-after? currenttime endtime
  [stop]
  ;~~

  phase-lift
  move-skier
  tick
end

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~PROCEDURE TO MOVE SKIERS~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to move-skier
  ask skiers [
    let current_node_type [breed] of target
    if distance target < speed / m-per-patch * s-per-tick [  ; when the skier gets close to their target
      set target one-of [out-link-neighbors] of target  ; choose a new target
    ]

    if target = nobody [  ; if the skier is at the end of the lift or slope
      if current_node_type = slopenodes [  ; if the skier is at the end of slope
        set target one-of liftnodes in-radius 5  ; choose a random lift near the skier
        ask target [
          set waiting-list lput myself waiting-list  ; and put skier on the waiting list of the lift
        ]
        set speed 0
      ]

      if current_node_type = liftnodes [  ; if the skier is at the end of the lift
        set target one-of slopenodes in-radius 5 with [start-node? = true] ; choose a random slope near the skier
        set speed ski-speed
        set shape "skier"
        set color one-of base-colors
      ]
    ]
    if speed = ski-speed [
      set total-skiing-ticks total-skiing-ticks + 1
    ]
    face target
    forward speed / m-per-patch * s-per-tick
  ]
end
```

```

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~PROCEDURE TO RIDE THE LIFT~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

to phase-lift
;  ask liftnodes with [waiting-list != 0] [
      ask turtle-set valley-stations [   ;loop over all valley stations
         set total-waiting-ticks total-waiting-ticks + length waiting-list   ;
              increase total waiting ticks by number of waiting skiers
         let current-lift-capacity lift-capacity * lift-speed / 5   ;if lift speed is
               higher or lower than standart 5m/s, scale lift capacity accordingly
         let chair-capacity 6   ;this model represents 6 seats per gondola
         let chair-period int(chair-capacity / ( current-lift-capacity / 60 / 60 ) /
              s-per-tick)   ;calculate every how many ticks a gondola leaves the
              station
         if ticks mod chair-period = 0 [   ;this is true if it is time for a gondola
              to leave the station
            repeat chair-capacity [
               if not empty? waiting-list [   ;check if someone is in the waiting
                    list
                  ;tell the skier to proceed as a gondola
                  ask first waiting-list [
                     set speed lift-speed
                     set rides rides + 1
                     set shape "pentagon"
                     set color black
                  ]
                  set waiting-list butfirst waiting-list   ;delete that skier from
                       the waiting list
            ]
         ]
      ]
   ]
end
```

B

INTERFACE

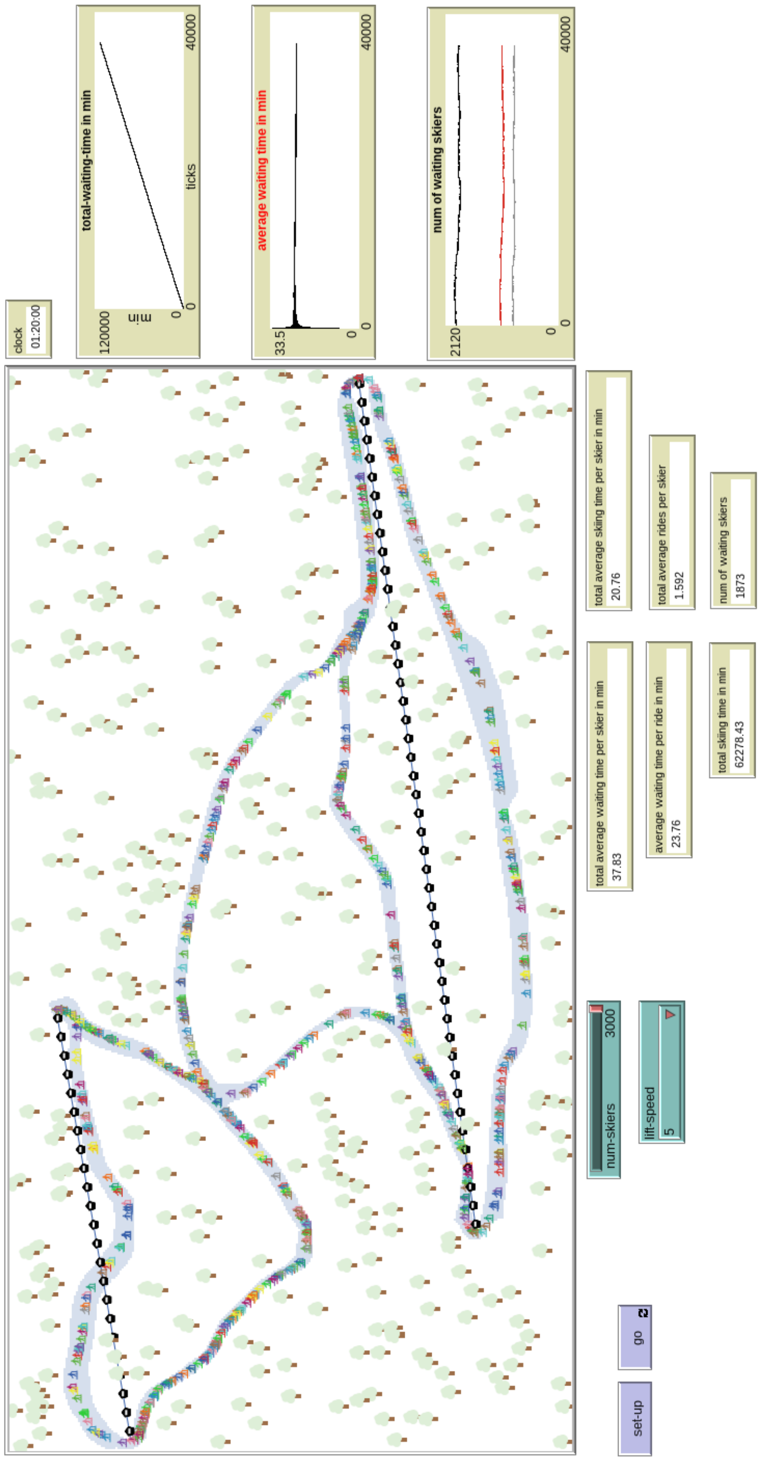

Figure 13: Model's interface in NetLogo

C

MAP OF THE STUDY AREA

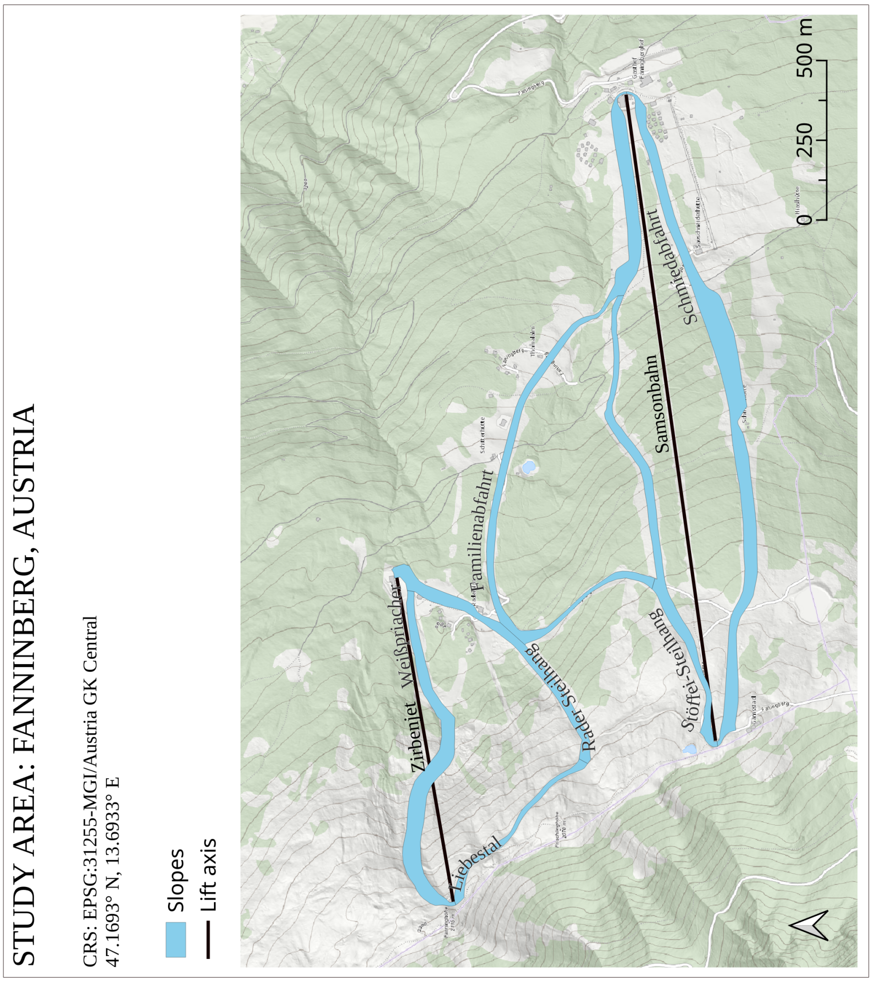

Figure 14: Slopes and axes network in Fanninberg which were used for the model.

BIBLIOGRAPHY

Abar, S., G. K. Theodoropoulos, P. Lemarinier & G. M. O'Hare (2017): "Agent Based Modelling and Simulation tools: A review of the state-of-art software." In: *Computer Science Review* 24, pp. 13–33. DOI: 10.1016/j.cosrev.2017.03.001.

Alvarado-Valencia, J. A., G. C. Tueti Silva & J. R. Montoya-Torres (2017): "Modeling and simulation of customer dissatisfaction in waiting lines and its effects." In: *Simulation: Transactions of the Society for Modeling and Simulation International* 93.2, pp. 91–101. DOI: 10.1177/0037549716678477.

An, L. (2012): "Modeling human decisions in coupled human and natural systems: Review of agent-based models." In: *Ecological Modelling* 229, pp. 25–36. DOI: 10.1016/J.ECOLMODEL.2011.07.010.

Balbi, S., C. Giupponi, P. Perez & M. Alberti (2013): "A spatial agent-based model for assessing strategies of adaptation to climate and tourism demand changes in an alpine tourism destination." In: *Environmental Modelling and Software* 45, pp. 29–51. DOI: 10.1016/j.envsoft.2012.10.004.

Chetouane, F., R. Bernier & S. Collart-Dutilleul (2010): "Simulation and Optimisation of Lift Networks in a Provincial Skiing Resort." In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 43.17, pp. 22–27. DOI: 10.3182/20100908-3-PT-3007.00006.

Crooks, A., C. Castle & M. Batty (2008): "Key challenges in agent-based modelling for geo-spatial simulation." In: *Computers, Environment and Urban Systems* 32.6, pp. 417–430. DOI: 10.1016/j.compenvurbsys.2008.09.004.

European Commission (2022): *Exchange rate.* URL: https://ec.europa.eu/info/funding-tenders/procedures-guidelines-tenders/information-contractors-and-beneficiaries/exchange-rate-inforeuro%7B%5C_%7Dde (visited on 01/14/2022).

Falk, M. (2008): "A hedonic price model for ski lift tickets." In: *Tourism Management* 29.6, pp. 1172–1184. DOI: 10.1016/j.tourman.2008.02.021.

Falk, M. (2011): "International Price Differences in Ski Lift Tickets." In: *Swiss Journal of Economics and Statistics* 147.3, pp. 303–336. DOI: https://doi.org/10.1007/BF03399348.

Federal Ministry Republic of Austria (2022): *Cableways*. URL: https://www.bmk.gv.at/en/topics/transport/cableways.html (visited on 02/07/2022).

García-Magariño, I., G. Palacios-Navarro, R. Lacuesta & J. Lloret (2018): "ABSCEV: An agent-based simulation framework about smart transportation for reducing waiting times in charging electric vehicles." In: *Computer Networks* 138, pp. 119–135. DOI: 10.1016/j.comnet.2018.03.014.

Garner, B., C. Thornton, A. Luo Pawluk, R. Mora Cortez, W. Johnston & C. Ayala (2021): "Utilizing text-mining to explore consumer happiness within tourism destinations." In: *Journal of Business Research* 139, pp. 1366–1377. DOI: 10.1016/J.JBUSRES.2021.08.025.

Grimm, V., U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske & S. F. Railsback (2010): "The ODD protocol: A review and first update." In: *Ecological Modelling* 221.23, pp. 2760–2768. DOI: 10.1016/j.ecolmodel.2010.08.019.

Grimm, V. et al. (2006): "A standard protocol for describing individual-based and agent-based models." In: *Ecological Modelling* 198.1-2, pp. 115–126. DOI: 10.1016/j.ecolmodel.2006.04.023.

Hanzer, F., C. M. Carmagnola, P. P. Ebner, F. Koch, F. Monti, M. Bavay, M. Bernhardt, M. Lafaysse, M. Lehning, U. Strasser, H. François & S. Morin (2020): "Simulation of snow management in Alpine ski resorts using three different snow models." In: *Cold Regions Science and Technology* 172. DOI: 10.1016/J.COLDREGIONS.2020.102995.

Li, S., V. Colson, P. Lejeune, N. Speybroeck & S. O. Vanwambeke (2015): "Agent-based modelling of the spatial pattern of leisure visitation in forests: A case study in Wallonia, south Belgium." In: *Environmental Modelling and Software* 71, pp. 111–125. DOI: 10.1016/j.envsoft.2015.06.001.

Liu, J. & X. Chen (2019): "Simulation of passenger motion in metro stations during rush hours based on video analysis." In: *Automation in Construction* 107. DOI: 10.1016/j.autcon.2019.102938.

Mohr, M. (2021a): *Österreich - Gästeübernachtungen Saisons 2020/2021*. URL: https://de.statista.com/statistik/daten/studie/295500/umfrage/verteilung-der-uebernachtungen-in-oesterreich-nach-herkunftslaendern/ (visited on 12/07/2021).

Mohr, M. (2021b): *Österreich - Zufriedenheit mit dem Umsatz aus dem Wintertourismus 2018/19*. URL: https://de.statista.com/statistik/daten/studie/729308/umfrage/umfrage-zur-zufriedenheit-mit-dem-umsatz-aus-dem-wintertourismus-in-oesterreich/ (visited on 12/07/2021).

Mohr, M. (2021c): *Statistiken zum Wintertourismus in Österreich*. URL: https://de.statista.com/themen/3247/wintertourismus-in-oesterreich/ (visited on 12/07/2021).

Mündler, M. (2021): *Personal correspondence with Maximilian Mündler, PowerGIS*. (Visited on 11/18/2021).

Nicholls, S., B. Amelung & J. Student (2017): "Agent-Based Modeling: A Powerful Tool for Tourism Researchers." In: *Journal of Travel Research* 56.1, pp. 3–15. DOI: 10.1177/0047287515620490.

Poulhès, A. & P. Mirial (2017): "Dynaski, an Agent-based Model to Simulate Skiers in a Ski area." In: *Procedia Computer Science* 109, pp. 84–91. DOI: 10.1016/j.procs.2017.05.298.

Schmittenhöhebahn AG (2021): *SuperSkiCard*. URL: https://www.superskicard.com/de/verkauf.html (visited on 12/08/2021).

Skiresort Service International (2021): *Ski resorts Austria*. URL: https://www.skiresort.info/ski-resorts/austria/ (visited on 12/07/2021).

Skiresort Service International (2022): *Trail map Fanningberg*. URL: https://www.skiresort.info/ski-resort/fanningberg/trail-map/ (visited on 02/04/2022).

Statista Research Department (2016): *Most expensive European ski resorts 2016*. URL: https://www.statista.com/statistics/639048/most-expensive-european-ski-resorsts/ (visited on 12/08/2021).

Steiger, R. & D. Scott (2020): "Ski tourism in a warmer world: Increased adaptation and regional economic impacts in Austria."

In: *Tourism Management* 77. DOI: 10.1016/j.tourman.2019.104032.

Von Allmen, B. (1976): "Zur Bemessung und Bestimmung der Belastbarkeit von Schigebieten." In: *International aerial tramway reviewnternational aerial tramway review*, pp. 215–221.

Wallentin, G. & M. Loidl (2016): "Bicycle-bicycle accidents emerge from encounters: An agent-based approach." In: *Safety* 2.2. DOI: 10.3390/safety2020014.

Wilensky, U. (1999): *NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.* URL: http://ccl.northwestern.edu/netlogo/ (visited on 11/10/2021).

Wilensky, U. (2022a): *NetLogo 6.2.2 User Manual: Programming Guide*. URL: https://ccl.northwestern.edu/netlogo/docs/programming.html (visited on 01/24/2022).

Wilensky, U. (2022b): *NetLogo 6.2.2. User Manual: BehaviorSpace Guide*. URL: https://ccl.northwestern.edu/netlogo/docs/behaviorspace.html (visited on 01/24/2022).

WKO.at (2020): *Bergbahnen: Geschichte und aktuelle Zahlen*. URL: https://www.wko.at/branchen/transport-verkehr/seilbahnen/bergbahn-seilbahn-oesterreich.html (visited on 12/07/2021).

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/