Model-Based Difficulty Estimation of Indoor Bouldering Routes

Author
Resul Çelik - 6987648

First Supervisor
Prof. dr. M.J. van Kreveld

Second Supervisor
Dr. F. Staals

Third Supervisor
J. Vermeulen, MSc

Game and Media Technology Master's Thesis

Faculty of Science
Department of Information and Computing Sciences
Utrecht University
The Netherlands

January 12, 2022

Abstract

In this thesis, we describe an application-specific model to calculate the difficulty of a bouldering route. Current bouldering routes are graded by the route setters themselves which can introduce a bias. Our model, on the contrary, uses a set of tailormade rules to find the least-cost path for a given route, and extracts the difficulty from this path. This thesis introduces a novel data capture method, with which routes at an indoor bouldering gym can be recorded accurately. We evaluate our model by supplying these captured routes of varying difficulty to our model and comparing the actual order of grades with the order of predicted difficulty values. Several models of increasing complexity are tested for an increase in the accuracy of the difficulty estimation. Our results showed that a model that incorporates the type of each hold was able to predict the relative order of difficulty for each pair of routes in 70% of the cases.

Acknowledgements

I would like to thank my first supervisor, Marc van Kreveld, for his invaluable guidance and feedback throughout the course of my thesis. I am also grateful for my second and third supervisors, Frank Staals and Jordi Vermeulen. My sincere thanks to Boulderhal Bruut in Breda for providing the wall data and for allowing me to get in their way during their route setting sessions and collect some of their routes. Lastly, I would like to express my gratitude to all my friends and family from the bottom of my heart for all of their support throughout the years. I wouldn't be where I am today if not for them.

Contents

1 | Introduction

Rock climbing is an increasingly popular recreational and competitive sport with many different disciplines. One such discipline is indoor bouldering where a climber is tasked with finding a path up or across a wall aided by artificial shapes called holds. A collection of these holds of the same colour on the wall is called a route. The holds where the climber is supposed to start from and end at are labelled. A climber has started a route once the entire body has left the ground and the hands are holding the start holds. A route is finished if the climber touches the end hold with two hands. Once a route is started the climber has total freedom to use any combination of holds to reach the final hold.

Routes are created by route setters, who are experienced climbers that work at a climbing gym. To inform the climber of the approximate difficulty of the route, most gyms use a grading system. Once a route is set, the route setter assigns it a grade. This grading process is entirely subjective, and as such there can be a large discrepancy in the difficulty between routes. Several previous papers have looked at route difficulty classification. However, most research focusses on the Moonboard training equipment[1]. This is due to the relative ease of extracting a large database of routes. The downside of using the Moonboard dataset is that it is much more restricted than an actual bouldering wall.

This thesis is the first that looks at routes set on bouldering walls to extract difficulty information. By extracting various features from the routes and supplying them to a model, a digital approximation can be created of the route. These features include wall positions, hold position, hold direction and hold types. A least-cost path finding algorithm is used to greedily go through all possible hand-foot configurations (stances) to find the best path up the route. From this least-cost path the final difficulty is calculated.

This thesis uses 13 routes divided into five classifications. A classification is one or two Fontainebleau grades combined. To keep the model as simple as possible, a baseline model is created and then extended by various enrichments. These enrichments will be evaluated separately to find the best performing model.

1.1 Overview

The rest of thesis is laid out in the following order. In the next chapter, our research questions are introduced and explained. In Chapter 3 an overview of rock climbing is given as well as a brief history on the sport and its evolution. Next, Chapter 4 will summarize previous work done on studying rock climbing with an emphasis on automatic route synthesis and grade prediction. Chapter 5 explains what data we have collected from the routes to generate a route grade prediction. The next chapter explains the model that uses this data to generate the final grade prediction. These results will be given in Chapter 7. The results will be discussed in Chapter 8. From this discussion a conclusion of our model will be given in Chapter 9. Finally, the limitations of our model and possible future work will be described in Chapter 10.

[1] Moonbard, Advanced Indoor Training for Climbers since 2005, Moon Climbing Ltd (https://www.moonboard.com/)

2 | Research Questions

This thesis will seek to answer the following research question:

How much does adding complexity to a model increase the accuracy of difficulty estimation of an indoor bouldering route?

The main goals of this thesis are to generate a description of a model that can be utilised to assess the difficulty of routes set in an indoor bouldering gym and to create a model that is capable of using this data to accurately predict the difficulty of these routes. This model will use a set of common climber best-practices as a set of rules. These rules are implemented as a set of penalties. Therefore if these rules are not followed, the model increases the difficulty of the route.

Previous research has not looked at different aspects of a bouldering route and how it can affect grade predictions. Therefore a set of sub-questions are created to evaluate various properties:

- *How does adding hold properties affect the accuracy of difficulty estimation?*
- *How does adding climber properties affect the accuracy of difficulty estimation?*

In these sub-questions we have introduced two categories of properties: hold and climber properties. In the following sections, we will explain what these properties are. The detailed explanation on how these properties have been implemented can be found in Chapter 6.

2.1 Hold Properties

Hold properties are extra information captured for each hold. In the baseline model, we assume a simplified model for our holds where all of them are indistinguishable from each other. Each hold can be held from any angle, and the position you hold it at is exactly at the centre of the hold. Additionally, each hold has the exact same difficulty regardless of its hold type. The hold properties improve this baseline using three pieces of additional information, which include the hold type, distance to hold surface and the hold direction.

The hold type is an important measure of difficulty because some hold types are more commonly found at certain bouldering grades. Therefore each type will have a different difficulty level.

Holds can differ a lot in size, this might mean that the distance from centre to centre between two holds could exceed the arm span of a climber, but the distance between the actual hold surfaces might not.

Lastly, the hold direction. Most holds can't be held at any angle, or at the very least have a direction where they are the easiest to hold. Some holds even have slippery surfaces outside the hold surface to further discourage using these surfaces.

2.2 Climber Properties

Our baseline model does not differentiate between the different limbs. However, when climbing, a climber manoeuvres their limbs to be in an optimal position, to expand the least amount of energy. Therefore a limb property is added to the climber enrichment. This allows us to make the model aware of the different limb configurations, and add penalties to certain configurations. For example, although sometimes necessary, a climber does not prefer to hang upside down when climbing.

3 | Rock Climbing

As the reader might not be familiar with rock climbing and its various disciplines, this chapter familiarises them with the sport and its specific jargon. Rock climbing is the overarching name given to any climbing related sports such as mountaineering, ice climbing, rope climbing and bouldering. However, this chapter will mostly be focussing on indoor bouldering.

3.1 Sport Climbing

One of the disciplines within rock climbing is sport climbing. Sport climbing can be described as a sport where a climber is tasked to climb up, across or down a rock wall, either natural or artificial, to reach a predefined goal without falling. To add to the challenge, oftentimes only a set of marked holds are allowed to be used for the traversal. This set of holds is said to be a route.

Sport climbing is frequently separated into three disciplines, top roping, lead climbing and bouldering. The first two disciplines are performed on rock walls that can go up hundreds of meters. Due to this, safety gear such as a harness and rope is required. In the event of an accident this gear stops the climber from falling a dangerous distance. The last discipline, bouldering, does not require a rope. This is because the natural rock formation or artificial bouldering walls that this discipline is performed on usually extend a maximum of six meters. To ensure a soft landing, padded mats are placed down on the ground, in the approximate landing area of the climber.

This thesis focusses solely on bouldering, and therefore for more information on rope climbing it refers the reader to external sources.

3.2 Bouldering

Originally bouldering was not seen as a separate sport from rope climbing. It was seen as a training instrument. For example, to replicate a difficult rope climbing move at safe heights, and as such be able to train it more safely. Bouldering was also used to maintain finger strength and stamina during times when rope climbing was not available, such as during the winter months.

Early in the sport's history, natural rock formations were used to boulder. Later, indoor gyms were built specifically to train bouldering moves. The first indoor walls were constructed from bricks. Current iterations of the walls use plywood sheets. These sheets are backed by a lattice of metal scaffolding to provide structure. Several sheets are combined at different angles to create a wall with different challenges. An example of such a wall can be seen in Figure 3.1.

A grid of threaded holes drilled at regular intervals into the plywood sheets provide convenient points to attach holds to using bolts. These holds are usually made out of coloured plastic, although wood is also occasionally used. The plastic holds and walls are often textured to mimic a natural rock texture. Some holds can additionally have a slippery edges to discourage the climber from using these parts of the hold. Their shape can vary widely, from small holds only a couple of centimetres across to holds that are almost a meter in size. Although the shapes can vary widely they can often be classified to be of a certain type. The

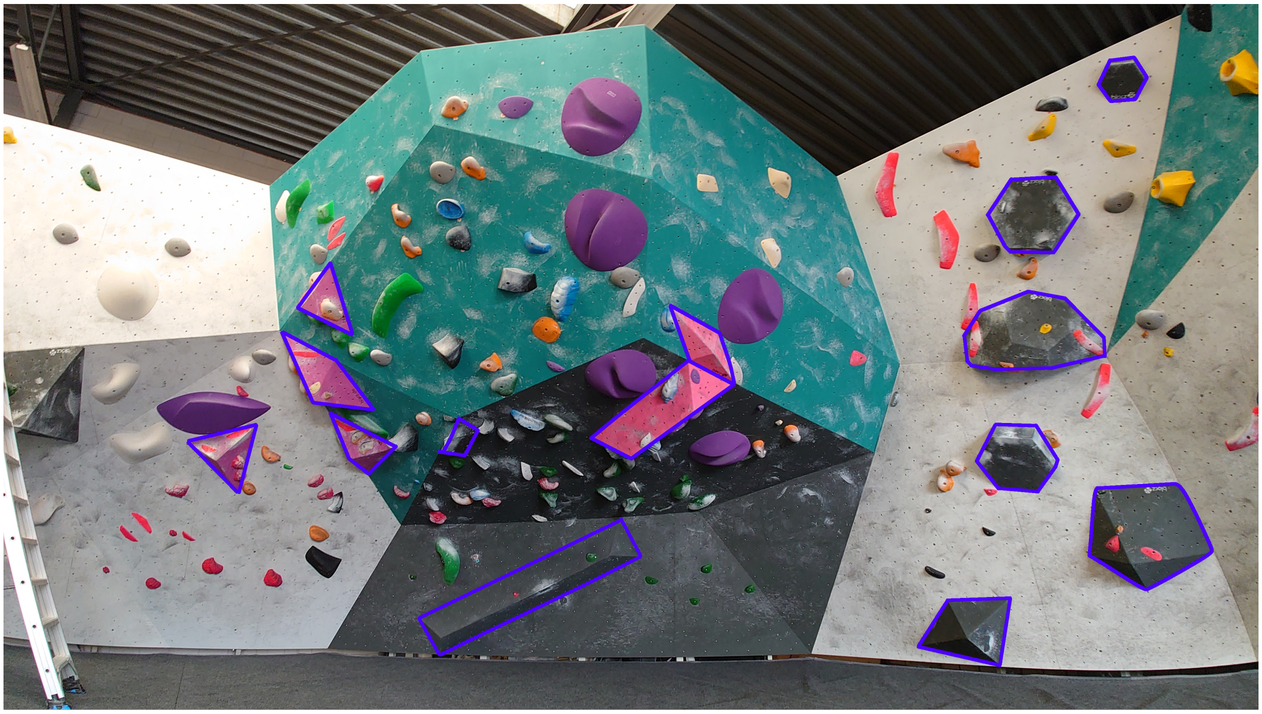


Figure 3.1: An example of an indoor bouldering wall. The coloured walls are made out of plywood sheets which are attached to a metal scaffolding at the back to create structure. Different coloured holds are attached to these walls. Each set of coloured holds defines a different route up the wall. The purple outlined volumes augment the walls and add new ways to create a route.

most common types are Jugs, Crimps, Pinches, Slopers, Pockets and Jibs. Figure 3.2 contains an example of each of these types.

Figure 3.2a illustrates a Jug this type of hold is characterised by a concave shape that fits the entire hand. This type of hold is the easiest to hold and as such is mostly used for beginner climbs. An example of a Crimp can be seen in Figure 3.2b these holds have a narrow flat edge to hold onto. Oftentimes only the fingertips can be placed on the edge. A Pinch, as seen in Figure 3.2c, is a hold that is meant to be pinched with the thumb and fingers on either side of the hold. Most often Slopers will have a convex shape. To hold onto Slopers the climber has to rely on the friction between the hold and their hand. Finally, an example of a Pocket can be seen in Figure 3.2e. These, like Jugs, have a deep pocket, but unlike Jugs they restrict the number of fingers they fit, often to less than three fingers.

Smaller plywood shapes, called volumes can be added to the wall to change the topology of the wall and to provide new challenges. Like the walls these often have bolt holes for holds to bolt into. Volumes can be used by a climber climbing any route, as they are considered to be parts of the wall.

Although each gym can have their own rules, most gyms make use of similar set of rules on how to climb a route. Each route needs a start and end point. Either one or two holds are marked as the start holds. These are the holds that the two hands have to be touching when a route is started. A route is said to have started when the two hands are on the start holds, and the feet are off the ground, either on the wall or likewise on a hold. Once a route is started, the climber is tasked to climb to the end. This can either be another hold that is marked as the end or, occasionally, the top of the wall is used as the final "hold". A route is said to be finished, or topped, when both hands touch the end hold for several seconds.

Figure 3.2: Examples of the five most common types of holds, as well as an example of a volume.

3.3 Route setting

To keep climbers engaged, indoor bouldering routes are occasionally changed. Old routes will then be removed from the wall and new ones created by a team of route setters. Route setters are expert climber that have a lot of experience with climbing and/or bouldering.

Route setting begins with the removal of the old routes, cleaning the holds, and sorting them for future use. A route setter then chooses several holds to use for the new route. These holds are bolted or screwed on to the wall. While attaching the hold, the route setter pays close attention to the position and angle it is attached to the wall. Even, slight changes to these parameters can drastically change the difficulty of the route. During the build the route setter will iteratively evaluate their route and make changes when desired. Once a route is finished the original setter or one of their fellow setters climbs the route and gives it a grade.

3.4 Grading

To inform a climber of the approximate difficulty of a route, a grading system is used. There are various grading scales used in different parts of the world. The most widely used system in North America is the

Hueco (also known as Vermin) scale. This is an open-ended scale system where a route is graded by a single number prefixed by a 'V', e.g. V4, V8. Like many other grading systems this system does not take into account danger or fear as part of the scale, only the physical aspects of the climb itself are evaluated for a grade.

Most of Europe uses the Fontainebleau scale, which is an open-ended grading system as well. This system also uses a numbering system to grade a route. However, it can make use of letters (a,b,c) to further divide the grade, e.g. 6a, 7b. Even further distinction can be made with a plus sign (+), e.g. 6c^{+}.

Even though both of the above-mentioned scales use a similar numbering style and set of criteria, they are not easily translated to one another. To make comparing academic research easier the International Rock Climbing Association (IRCRA) proposed [1] a reporting scale as well as a conversion table from all major grading systems. This table can be seen in Figure 3.3. This thesis will report what IRCRA grades were recorded in Chapter 5.

Climbing Group		Vermin	Font	IRCRA Reporting Scale	YDS	French/sport	British Tech		Ewbank	BRZ	UIAA	Metric UIAA	Watts
				1	5.1	1		2	4	I sup	I	1.00	
				2	5.2	2			6	II	II	2.00	
				3	5.3	2+			8	II sup	III	3.00	
				4	5.4	3-	3			III	III+	3.50	
Lower Grade (Level 1) Male & Female				5	5.5	3			10	IV	IV	4.00	
				6	5.6	3+		4	12	V	IV+ V-	4.33 4.66	0.00
				7	5.7	4			14		V	5.00	0.25
				8	5.8	4+					V+	5.33	0.50
		VB	< 2	9	5.9	5	5a		16	V sup	VI-	5.66	0.75
				10	5.10a	5+			18	VI	VI	6.00	1.00
		V0-	3	11	5.10b	6a		5b	19		VI+	6.33	1.25
	Intermediate (Level 2) Female	V0	4	12	5.10c	6a+					VII-	6.66	1.50
Intermediate (Level 2) Male		V0+	4+	13	5.10d	6b	5c		20	VI sup	VII	7.00	1.75
		V1	5	14	5.11a	6b+			21	7a			2.00
			5+	15	5.11b	6c		6a		7b	VII+	7.33	2.25
		V2	6A 6A+	16	5.11c	6c+			22		VIII-	7.66	2.50
	Advanced (Level 3) Female	V3	6B	17	5.11d	7a			23	7c	VIII	8.00	2.75
		V4	6B+	18	5.12a	7a+	6b		24	8a	VIII+	8.33	3.00
		V5	6C 6C+	19	5.12b	7b			25	8b			3.25
Advanced (Level 3) Male		V6	7A	20	5.12c	7b+			26	8c	IX-	8.66	3.50
		V7	7A+	21	5.12d	7c		6c	27	9a	IX	9.00	3.75
		V8	7B	22	5.13a	7c+			28	9b	IX+	9.33	4.00
	Elite (Level 4) Female		7B+	23	5.13b	8a			29	9c	X-	9.66	4.25
		V9	7C	24	5.13c	8a+			30	10a			4.50
Elite (Level 4) Male		V10	7C+	25	5.13d	8b	7a		31	10b	X	10.00	4.75
		V11	8A	26	5.14a	8b+			32	10c	X+	10.33	5.00
		V12	8A+	27	5.14b	8c			33	11a	XI-	10.66	5.25
		V13	8B	28	5.14c	8c+			34	11b	XI	11.00	5.50
	Higher Elite (Level 5) Female			29	5.14d	9a		7b	35	11c			5.75
Higher Elite (Level 5) Male		V14	8B+	30	5.15a	9a+			36	12a	XI+	11.33	6.00
		V15	8C	31	5.15b	9b			37	12b	XII-	11.66	6.25
		V16	8C+	32	5.15c	9b+			38	12c	XII	12.00	6.50

Figure 3.3: The various reporting scale (for both climbing and bouldering) as well as the conversion to the IRCRA reporting scale.

3.5 Definitions

Below is a list of all climber specific jargon used throughout this thesis as a reference.

- Hold - A coloured shape which the climber uses to ascend up a wall.
- Bouldering route/problem - A set of various holds that the climber is allowed to use to ascend the wall. A route starts with one or two holds marked as hand-holds and one, often the top-most, hold is marked as the finishing hold. A climber has started the route when the two hands are on the starting holds and both feet are off the ground. A climber has finished a route when both hands touch the finishing hold for a couple of seconds.
- Grade - The difficulty of a bouldering problem.
- Hooks - A toe-hook or heel-hook is a climbing move where a toe or heel respectively is placed on or behind a hold to create tensions
- Route setter - A person that comes up with and creates new routes.
- Move - The movement needed to reach the next stance. This can be a large jump or a subtle hand readjustment.
- Top - Finishing a route by touching the last hold with both hands in a controlled manner.
- Flagging - Extending the leg to shift the centre of mass of the climber.
- Smearing - Using the friction between the climbing shoe and the wall.
- Matching - Having two limb appendages on the same hold.
- Rock over - Shifting the centre of gravity from one foot to another. Often used to extend the reach of the climber.
- Barn-door - If the centre of mass is too far from the wall and all limb appendages attached to holds are on or close to the same vertical axis it can lead to a barn-door. In a barn-door the hands and feet act like hinges and the body swings away from the wall, similar to a barn door opening, often resulting in a fall.
- Dyno - An abbreviation of dynamic move. It is a move where a sudden impulse is needed to propel the climber towards a hold. Dyno's often require all limb appendages to detach from their holds, i.e. jump to the next hold.
- Crux - The most difficult move of a route. The difficulty of a route is often graded based on the crux' difficulty.
- Jug - A fairly deep hold whose topology is similar to that of a steep-walled pot or jug.
- Crimp - A shallow hold which may only support the tips of the fingers and might need to be locked-off (where the thumb reinforces the position, by pressing down on the forefingers)
- Sloper - A more rounded hold which is gripped with the palm of the hand or pads of the fingers, to create friction.
- Jib - A very small knob-like hold, usually used as a foot piece
- Pocket - A hold that only allows one to three fingers to be used.
- Pinch - A hold with two opposing faces meant to be pinched between thumb and fingers.
- Sidepull - When a hold is positioned so that its main gripping surface is away from the climber's body. Similar to the grip one would use to close a sliding glass door.
- Gaston - The opposite of a sidepull, with the gripping surface facing inward. Best explained as the grip used to pry open an elevator door.
- Beta - Information about how a route/problem must (or can) be climbed.
- Volume - Extensions added to a wall, holds can be bolted onto volumes, volumes can be used on any route.

- Flash - Topping a route at the first try, looking at other climbers or learning the beta is permitted.
- Onsight - Topping a route without having seen someone climb the problem or learning the beta.

4 | Related Work

In the following sections, we present existing work on automatic difficulty estimation and route synthesis of climbing and bouldering.

4.1 Route Synthesis

Designing new bouldering problems takes a significant amount of work, regardless of the route setters experience. It is a highly iterative process where the route is tested and modified in a cycle. This can be labour intensive as it involves bolting on or removing holds at various heights and angles. Multiple papers have tried to simplify or aid in this process by digitally synthesising new routes which can help kickstart the creative process and cut down on the labour-intensive work.

Phillips et al. [2] used a descriptive language called Climbing Route Description Language (CRDL) (Figure 4.1) to transcribe problems to a computer-readable format. Considerations were made to include as much detail as possible without being difficult enough to form a barrier when transcribing the route. The language only models the sequence of hand movements, as the authors believe foot holds can easily be set to support the hand movements. Exceptions are foot movements where the climber has to use a toe or heel-hook. Multiple transcribed problems are then combined using a chaotic variation algorithm to generate new problems. This CRDL representation gives the setters more freedom as it does not specify the exact placement of holds, but instead the sequence of movements required to get to the top. The authors remark that testing if this language is descriptive enough can be achieved if two routes created with the same description feel to be variations on the same premise. However, this notion was not tested. As this technique is used as an aid for a setter, important difficulty measures are left out, such as the placement of feet during a climb and the steepness of a wall. The route setter can augment the difficulty by choosing harder holds or set harder to reach foot holds. These abstractions make CRDL unsuitable for automatic difficulty detection. It also requires an experienced setter to transcribe the route, which introduces a bias on how it is supposed to be climbed.

Indoor boulder routes are typically reset after a certain amount of time, this means that capturing a large database will take a considerable amount of time. To increase the data available to them, several papers used the standardized training wall known as a Moonboard (Figure 4.2). These training walls are found in climbing gyms all over the world. This wall has 198 holds laid out in an 18x11 grid. The configuration of the holds stays fixed for an entire year. A series of led lights let the climber know which holds can be used to climb up. Climbers can create new routes and upload them to the internet for others to climb and rate.

Kin Ho Lo [3] used the Moonboard's dataset in their paper to allow them to train a variational autoencoder. Once trained, this autoencoder could be sampled to generate new climbing routes. An autoencoder is refined by passing the training data through a lower dimension latent space and trying to regenerate the training set using a decoder. For their method, the encoder reduced the 198 (18x11) vector of holds to a vector of length 16. By training the autoencoder on a large dataset the features from a route can be encoded into the lower dimensional latent space. From this space, the decoder can extract a route by sampling the latent space and expanding back to a vector of size 198. The model was trained using almost 17000 problems obtained from the Moonboard dataset. After training, 50 problems were generated by randomly sampling the latent space. A minimum of six holds was set as a constraint for the chosen routes, which reduced the viable routes from

```
Problem #13 from the CU–hosted RMR CCS
Climbing Competition in March, 2009.
A few large moves between moderate
crimps and slopers with thin/smeared
feet on a vertical wall. Set by Thomas.
Intermediate Difficulty.
———
R slopey ledge
L match
R medium crimp sidepull
L diagonal sloper
R crimp (big move)
L sloper (bad) sidewaysish
R crack sidepull
L wide pinch
R match
```

Figure 4.1: An example of the CRDL developed by Phillips et al. [2]. Instead of specifying the details of the route, an experienced route setter transcribes the moves and gives a description of the hold.

50 to 27. Five of them could not be uploaded to the Moonboard's site due to issues with the routes. One problem was an exact copy of a route already uploaded by a user. For another, it was almost impossible to reach the final hold from the penultimate holds. The other three lacked starting or finishing holds, which are required for a problem to be uploaded to the Moonboard database. Due to the dataset being skewed to the easier grades, the generator also seems to generate easier routes in general. However, the exact grading of the routes was not determined.

4.2 Route Finding

Path planning for a humanoid climber has applications ranging from real-world robotic climbers to simulating more realistic climbing in video games. As with the previous papers, it can also aid in the route setting process. Therefore several papers have explored route finding for virtual humanoid characters.

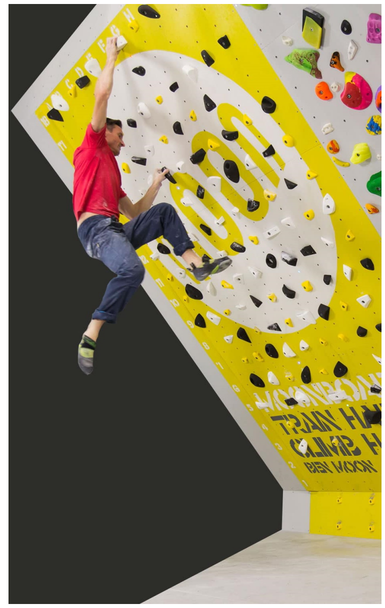
Pfeil et al. [4] designed a tool to interactively create new routes in a virtual environment. A user was able to add holds to a vertical wall using a simple mouse interface, and the software simulated potential climbing routes in real-time. Two hold types were defined, a normal hold for both hands and feet, and a hold only for foot placement.

Instead of physically simulating the entire climbing sequence, their technique finds *stances*. These stances are an abstraction of the pose of a climber. A stance is a tuple of the four holds that are attached to the hands and feet of the climber. A route is found by generating a sequence of stances from the start hold to the finish hold. The limbs of a 2D character with 12 bones are placed at the four holds defined by the stance to create a pose for the climber. A resting pose is mimicked by minimizing a cost function for the character's bone angles. The full path is subsequently visualized by linearly interpolating between these rest poses. Only stances that differ with one limb from the current stance are used as candidate stances for the transition.

To allow real-time interaction considerations had to be made. The virtual climber was only allowed to move one limb at a time and the hands and feet were constrained to the holds on the wall. This meant that more advanced techniques, e.g. dynamic movements, or moves using the wall as support could not be simulated. Additionally, each hold was modelled as a ball and socket joint, which means that there is no variation between holds, unlike actual routes. Furthermore, the interpolation did not respect human joint restrictions, which resulted in unrealistic intermediate poses according to an informal study with three experienced climbers.

Naderi et al. used a similar notion of stances in their paper [5] which explored finding the least cost path for a given problem. Their paper breaks the route-finding into two phases, a low-level and high-level planner.

(a)



(b)

Figure 4.2: The 2016 Moonboard configuration and a climber attempting to climb a route using this configuration. Images by courtesy of Moon Climbing Ltd.

The high-level planner employs the same stance representation as Pfeil et al. The authors do, however, allow free-hanging appendages that are not attached to any hold. This is achieved by building up the stances from all holds and one more special value which denotes a free appendage. The high-level planner creates a graph of plausible stances and uses a path planner to find the least-cost path to the top. This graph is generated by creating a stance for each unique 4-tuple of holds. A set of heuristics is subsequently used to cull as many stances as possible. These are the following heuristics:

- Foot holds cannot be higher than 0.1cm above the highest hand hold.
- If all appendages are attached to holds, there should at least be three unique holds in any given stance.
- Maximum distances between the appendages should not be exceeded.

A graph is generated by connecting the remaining stances. However, once again some connections are deemed impossible and as such are removed using another set of heuristics:

- At most two elements can differ from one stance to another.
- At least one hand and at least two appendages are connected when transitioning from one stance to another.

Using the generated graph a path planning algorithm finds the least cost route. The overall cost of the route is divided into three cost functions. A sum of all the node and edge costs and a dynamic cost which is initially set to zero. The node cost depends on what holds are in the stance graph and in what configuration, free hands are penalized more than free feet, as feet can be used for flagging. Non-crossing limbs are preferred as it is often cumbersome to cross one's limbs when climbing. Matching feet or limbs that are too close add to the overall cost of the node as these can lead to instability of the climber. The edge cost is calculated by observing the transition from one stance to another. To avoid making redundant moves the distance between the holds is calculated and included as a cost heuristic. Moving more than one limb at a time is considered difficult when it is not a simple ladder-climbing motion and contributes to the overall edge cost. A cost is included for only one or two unique holds attached to the limbs as this pose is often unstable and can lead to a barn-door situation. These two cost functions are then summed to get the total cost of a path in the graph. This results in several trees from the start node to several end nodes.

In the low-level planner, a physically simulated actor tries to follow the path found by the tree. Although the heuristics filter many of the impossible stances, not all of the stances and transitions are reachable by the physics controller. If the controller fails to perform a move, the dynamic cost of the transition is increased and the path planning algorithm is rerun to find the new least-cost trees.

Using this technique the authors were able to find plausible looking routes for a virtual climbing route in under a minute of CPU time. However, it has several limitations. As with the previous technique it did not make any distinction between the holds and they were all modelled with the same ball and socket joint. The heuristics were also regarded as more complicated than necessary by the authors. This model also excludes many stances to reduce the number of nodes and edges in the graph representation to reduce computational complexity. Dynamic moves and more exotic moves are not permitted by the set of graph generation heuristics and as such get culled.

Katsura et al. [6] used the same technique as the previous authors to improve the quality of routes found for Moonboard problems. The same cost function was used as the previous study, with the only change made to the node cost. Here an additional heuristic was added that takes into account the difficulty of the holds attached to the left and right hands. Because there was no prior research in incorporating the difficulty of the holds in such an algorithm, the authors decided to experiment with splitting the difficulty up into different partitions. The experiments were run with 2, 4, 8 and 16 partitions. For example with 2 partitions, a hold could either be labelled "easy" or "hard". Using a genetic algorithm trained on data captured from the Moonboard site, the difficulty value for each hold was determined. An interview with an undetermined amount of expert climbers showed that incorporating the difficulty of a hold in the route finding algorithm produced higher quality paths up a problem. A partition size of 8 produced the best route according to the climbers.

4.3 Difficulty Assessment

Different methods have been proposed to assess the difficulty of a boulder problem or climbing route. Kempen [7] tried this using Variable-Order Markov Models (VOMM). They employed a similar domain-specific language as Phillips et al. [2] to describe the route. As they used a similar language, they could reuse the data captured by Phillips et al. Due to the limited number of data points in the data, the model only had two classes. The routes are either classified as being easy or hard. To evaluate the performance of their model, the accuracy and a Matthews Correlation Coefficient (MCC) of the model are calculated. Due to the skewed number of entries for the two difficulties a model that always guesses that a route is easy will have an accuracy of 62%. Their models had an accuracy ranging between 58% and 62% and an MCC of 0.01-0.19. Although there seems to be some correlation judging from the MCC, the accuracy is barely higher than a random classifier.

Like Kin Ho Lo, Dobles et al. [8] used the Moonboard to capture a large amount of data. This allowed them to acquire over 13,000 routes. However, instead of generating new routes, their research focussed on grading the routes in the dataset. Each route is rated by the user that created the route before being uploaded to the Moonboard database, while 42% of the routes were also rated by others in the community. The difficulty was divided into 13 grades, with more than half of the routes being in the first four grades and only 1% in grades 10-13. Three different machine learning models were used to predict the difficulty of these routes, Naive Bayes, Softmax Classifiers, and a Convolutional Neural Network (CNN). The first two classifiers were implemented as a performance baseline. The authors note that the Naive Bayes model's view of the route, where each hold is assessed separately is not a correct assumption. A route is a series of connected moves, and therefore can't be assessed on a per hold basis. The three classifiers were tested using three metrics, accuracy, mean absolute error (MAE) and Kullback-Leibler divergence. They were additionally compared to the rating submitted to the Moonboard app. The user ratings are not comparable with the classifiers as users do more than just look at the holds, they climb the route as well. Both baseline classifiers overpredict difficulty of the more common grades in the dataset, while completely ignoring the ones with less common categories. All three classifiers achieved a similar accuracy of $\sim$35%. The Naive Bayes had an MAE of 1.73, while the other two were $\sim$1.40. However, the CNN and the Softmax classifiers had a significantly lower KL-divergence than the Naive Bayes, 0.020, 0.078 and 0.41 respectively.

Cheng-Hao Tai et al. [9] also used Machine learning to try to grade Moonboard problems. As with previous studies, the Moonboard database was used to gather a large number of routes. However, the data was still extremely skewed towards the easier grades. To counteract this imbalance, the higher grades were upsampled to obtain enough training data. The authors chose to use a Graph Convolutional Network (GCN) previously employed for Natural Language Processing. Their justification for this was that as a sentence is made up of a string of connected words, a climbing route is made out of a series of connected holds. Like a CNN, a GCN works by combining neighbouring nodes in a non-linear way to learn features of the graph. By combining multiple layers, the model can learn information from nodes connected to neighbouring nodes. Ten experimental setups of the GCN were tested as well as some baseline models. An F_1 and Area Under the Curve (AUC) score was calculated for each of the models. Surprisingly, the best model had four convolutional layers. This went against conclusions from previous studies [10, 11] which found that more than two layers did not improve the performance of their model. This could mean that the optimal layer size could be domain-specific.

Instead of feeding the entire 18x11 grid of the Moonboard to a neural network, Yi-Shiou Duh and Ray Chang [12] preprocess the problem into a sequence of moves. For better comparison, all grades are converted to the Heuco scale ranging from V4 to V14, i.e. 10 different grades of difficulty. These are then used to grade existing problems (GradeNet) as well as generate new problems (DeepRouteSet). The sequence of moves is graded based on the relative distances between the holds and the difficulty of each hold. Using a beam search algorithm the easiest route is chosen from these candidate moves. This move set was then fed into a neural network consisting of two LSTM layers, followed by six dense layers. The output of this is then used in two processes, it is first flattened for a first grade estimation. As well as fed into two more LSTM layers followed by two dense layers for a second grade prediction. Their GradeNet was able to reach an accuracy of 87.5% if off-by-one errors in the Heuco scale were allowed. Compared to previous techniques they performed significantly better than [8], and were on par with [9]. Their argument for allowing off-by-one errors was a small test performed with three expert climbers. These climbers were tasked to rate a blind problem without climbing it. The authors found that the climbers grading performance was on par with their model, both when taking into account the off-by-one error and without. Their DeepRouteSet generated a set of routes that were evaluated by a climbing expert to be high-quality reasonable problems.

Instead of a Moonboard, Scarff [13] gathered his dataset from outdoor bouldering problems, these will remain static for a long time, which means an increased amount of bouldering problems and climbing attempts. This resulted in almost 9000 routes climbed over 236000 times by 3000 climbers. The climber's and route's rating are initially set to a normally distributed random value. The route's rating is centred around its initial grade, while the climber's is centred around 0. Using the set of ascends of a climber for a particular route, the rating of the route and the climber is updated simultaneously. The route's rating is updated because the grading is subjective, and often the opinion of a small group of climbers. Using the WHR, an objective grade can be produced for the route using data from all climbers that attempted the problem.

Another way to assess the difficulty of a route is to track the user using sensors attached to the climber when climbing a problem. Ebert et al. [14] strapped 18 climbers with five sensors, one for each limb and a central one on the chest. These sensors recorded rotation, acceleration, and temporal information. Several features are extracted using this data. These features are used to train several supervised machine learning algorithms to be able to assess the difficulty of a problem. The study tracked 18 different climbers on a range of problems of the easiest three grade scales so that every participant could climb all test routes. The best algorithm was able to predict the grade of the problem with a 98% accuracy. However, as the authors only tested simple routes, there is no indication that this accuracy can be extrapolated to more challenging routes.

4.4 Open Problems

As we can see bouldering route difficulty assessment has been studied in the past. However, these have focussed on static routes by either using outdoor boulders or a standardised training wall. As of yet, there is no research on difficulty prediction of indoor bouldering routes set by route setters on a non-standardised wall. However, by looking at previous studies we can still gain some insight into what properties are important in predicting the difficulty of a route. From Kempen, we can learn that the CRDL introduced by Phillips et al.

does indeed not hold enough information to extract meaningful difficulty from. When assessing routes for the Moonboard Katsura et al. found that giving the holds a difficulty value between one and eight produced the best routes, which leads us to believe that the difficulty of a hold contains information that could be beneficial when assessing the difficulty of a route. This theory is further supported by the research done by Yi-Shiou Duh and Ray Chang where the difficulty of the hold is part of their GradeNet system. The stance representation employed by Pfeil et al. and Naderi et al. strikes a balance between extracting enough information for difficulty estimation, while not including too much information.

5 | Data collection

Previous techniques that studied the configuration of holds that make up a route for difficulty assessment either used an online tool to generate the routes or used publicly available data in the form of the Moonboard dataset. This thesis is the first to model actual routes set in a climbing gym in enough detail to extract the difficulty of those routes. Phillips et al. [15] also transcribed indoor climbing routes, but their goal was to generate new routes, and as such, we believe their modelling language does not contain enough information to extract accurate difficulty measures from. In this chapter we explain how the data was collected as well as an overview of what data was collected.

Every climbing gym has a different way of route setting. Additionally, many gyms use a proprietary grading scheme. Even if they use one of the default grading systems each route setter decides for themselves how hard a route is. This makes it difficult to compare different gyms to each other. Therefore all data for this thesis is recorded from a single bouldering gym. Although comparing routes is not an exact science, it is generally true that if a route belongs to a more difficult grade then it should also generally be harder to climb.

5.1 Collection Method

To accurately assess the difficulty of a bouldering route, as much information as possible should be extracted from the route itself as well as the wall it is attached to. In the sections below a description is given for all information collected from the wall and route.

5.1.1 Wall data

The wall data was extracted using a 3D model supplied by the owner of the gym. The walls are built up out of flat sheets of plywood. At regular intervals, bolt holes are drilled through this sheet of plywood. To secure a hold to the wall, most holds have a hole in the centre for a bolt to pass through and bolt into the wall. Smaller holds are screwed directly into the wood. Examples of both type of holds can be seen in Figure 5.1.

Using the 3D model, the vertices for each of the plywood sheets can be extracted to generate planes that represent the walls. From this representation the wall angle with respect to the floor can be extracted. For each wall an arbitrary edge is chosen as the basis for an axis. The x and y distances for three bolt holes, which we named the anchor holes, are recorded with respect to this axis on the plane of the wall. These three bolt holes are in an L-shape. As all bolt holes have a fixed distance between each other all other holes can be calculated from these three. Each of these walls was given a unique identifier, so that bolt holes match up to their real-life counterparts. An illustration of all extracted wall parameters can be seen in Figure 5.2.

Once this data has been collected it can be converted to a digital format. First, all vertices of the wall are connected in counter-clockwise order to create the plane of the wall. The same edge that served as the basis of the axis system is used to recreate the three anchor holes. The direction and distance are extracted from the anchor holes and subsequently used to create all other bolt holes that are present on the wall. This procedure is repeated for all walls.

(a) (b)

Figure 5.1: Examples of the two ways to secure a hold to the wall. Larger holds are attached to the wall using a bolt that goes through the centre of the hold as seen in Figure (a). Smaller holds do not have the space for this bolt and as such are attached to the wall using several screws driven directly through the plywood walls as seen in Figure (b).

5.1.2 Route data

Five consecutive colours were chosen to be recorded at the local climbing gym, ranging from 4c to 6b on the Font scale (13-17 on the IRCRA Reporting Scale). Overall, there are ten difficulty colours at the climbing gym ranging from 4 to 7B+ on the Fontainebleau scale (4-23 on the IRCRA Reporting Scale). To make the grade of a route more apparent at a glance, several consecutive grades are grouped into a colour. Five consecutive colours were recorded: green, yellow, orange, black, pink sorted from easier to harder. Although in theory, each colour has its own grade range, in practice each consecutive colour has some overlap with its direct neighbours. Instead of arbitrary colours, we decided to classify these five colours into an alphabetical range, sorted from easier to harder. A conversion for all different grading systems can be found in Table 5.1. From this point forward only the alphabetical classification will be used in this thesis.

Class	Colour	Font Scale	IRCRA Scale
A	Green	4c	13
		5a	14
B	Yellow	5b	14
		5c	15
C	Orange	5c	15
		6a	16
D	Black	6a	16
		6b	17
E	Pink	6b	17

Table 5.1: Conversions table of the different ways to grade a bouldering route used in this thesis. The Colour and Font Scale are used by the bouldering hall. These are converted into IRCRA Scale for compatibility with other papers. The Class scale will be used throughout the rest of this thesis to have an easy to understand classification.

5.1.3 Hold data

During the recording of routes several parameters are considered. How these parameters are recorded is described below.

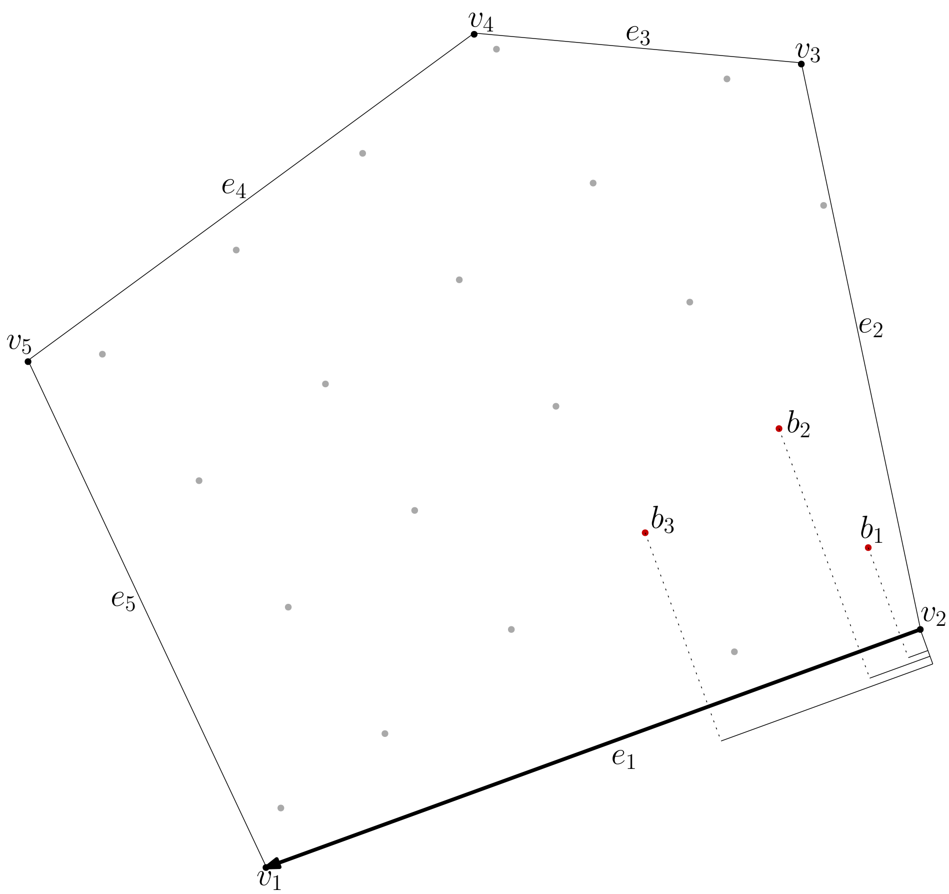


Figure 5.2: All information that is captured for a single wall. The wall vertices v_n, and the coordinates of anchor holes b_1, b_2, and b_3. These are capture with respect to an anchor vertex, in this case v_2. The distance along e_1 as well as the distance perpendicular to e_1 are measured. All other bolt holes are extrapolated from these three anchor holes.

First, as noted previously, each wall has been given a unique identifier. For each hold the ID of the wall the hold is attached to is noted. This is so that the hold can be placed at the right position in 3D space in the digital representation of the bouldering gym.

Second, the coordinates of the hold are recorded, this is achieved using the grid of bolt holes as a coordinate system. Starting at one of the extremities each hole in one direction increases the x-coordinate of the hold and moving in the perpendicular direction increases the y-coordinate. This results in a x,y-coordinate pair that can be directly translated into the 3D representation because all bolt holes have been recreated on the digital walls. However, as previously stated smaller holds do not use bolts to attach to the walls. For these, the nearest bolt hole coordinate is used. From this an offset is measured to get the precise position of the hold. Additionally, the start and end holds are marked. In our representation volumes are considered normal holds. Therefore they can use the same recording setup as the other holds.

Once the hold data is converted to a digital representation, the 3D coordinates of the holds can be extracted using the bolt hole coordinate system. As we know the vertices of the wall, we can construct three perpendicular vectors to represent the wall's coordinate system. Using this we can convert the 2D wall-centric coordinate system to 3D world coordinates. This will be necessary to accurately calculate the distance between different holds.

Next, the distance from the centre of the hold to the hold surface is measured. The hold surface is where a climber would hold on to when they are using the hold. Some of the holds can have multiple hold surfaces, e.g. pinches. The distance to each of these edges is separately recorded.

The type of the hold is another parameter that is noted. Although each hold is unique, and there are no hard boundaries between different hold types, each hold can only have one type. A hold can be on of the following five types: Jug, Sloper, Pocket, Pinch, Crimp.

Lastly, the direction of each hold surface is recorded. This is important as some holds are not easy to hold in certain orientations.

5.2 Collected Data

Using the above method the following data was captured from the bouldering gym.

In total 123 individual holds have been recorded, distributed over 13 routes. These routes are spread over 24 different walls. There are three routes captured of Classes B, C and E and two for Classes A and D.

6 | Model description

To be able to extract any difficulty information from the data captured in Chapter 5, a model has to be constructed that is able to make use of this data. The full pipeline, which converts the collected data to the final difficulty estimation, is described in the sections below. We describe multiple models, a single baseline model, which only uses the position of the holds for its difficulty estimation. This model is used as the basis for two enriched models that use additional data to improve the model's predictive capabilities.

6.1 Stance graph

We will be employing a similar abstraction as Pfeil et al. and Naderi et al. [4, 5] by using the stance representation for our model. This representation of a climber does not concern itself with the exact pose struck when climbing, instead it simply stores where the four limb appendages are placed. Using these stances, we can build a graph similar to the one that Naderi et al. built. However, due to the fundamentally different goals, our model employs different pruning criteria as well as edge and node cost functions.

6.1.1 Stance description

A stance is a 4-tuple of holds, $\sigma = [x_{lh}, x_{rh}, x_{lf}, x_{rf}]$, which denotes the positions of the four appendages, i.e. left hand, right hand, left foot, and right foot respectively. Each element in the stance can either be attached to a hold, $x_h \in \mathcal{H}$, or not attached to any hold, e.g. hanging or flagging, x_{-1}. Here $\mathcal{H}$ is the set that stores all holds.

For each permutation with repetition of holds a stance is created. Some of these stances are given a special meaning. A start stance, σ_{start}, is defined as a stance where both hands are attached to a start hold. If there are two start holds, both hands have to be on different holds. If there is only one, both hands have to be attached to this start hold. A stance is said to be a goal stance, σ_{goal}, if both hands are attached to the goal hold x_{goal}.

6.1.2 Graph Construction

From these stances a graph can be built that connects each stance with it neighbours. Here the graph nodes represent the stances, while the edges represent a transition from one stance to another. If two nodes are connected by an edge it means that there is a valid transition from one stance to another.

A three-step process is used to build the stance graph. In step 1, all possible stances are generated, regardless of the fact that they are physically possible or not. Each stance holds four holds in an ordered tuple. This tuple permits duplicate holds to be present. Therefore the number of stances generated in this step is $(|\mathcal{H}| + 1)^4$. There needs to be one additional component than the number of holds in the route, due to x_{-1} which allows a hand or foot to be from free from any hold.

The second step prunes the stances so that only physically possible stances are left. Here we introduce only one pruning criterion, which guarantees that the appendages cannot be further apart than the maximum allowed distance. There are three distances depending on the appendages in question. A hand-to-hand distance, d_{h2h}, a foot-to-foot distance, d_{f2f}, and a hand-to-foot distance, d_{h2f}.

In the third step the graph is generated using the stances that remain after pruning. We assume that once both hands are placed on the start hold or holds, the feet are off the ground. This means that any feet not attached to holds are either placed on the wall or hanging. For each stance a sphere with a radius of 2.85 meters is centred around its centre of mass (COM). Any hold within this sphere is marked as a candidate hold. Any stance that contains only candidate holds is connected with the current node. The sphere's radius is chosen as this is the furthest dyno recorded by the the Guinness World Records[1], and as such we can be confident that any hold outside this sphere is most likely unreachable by a human climber. This procedure is repeated for each node in the graph until all possible connections are made.

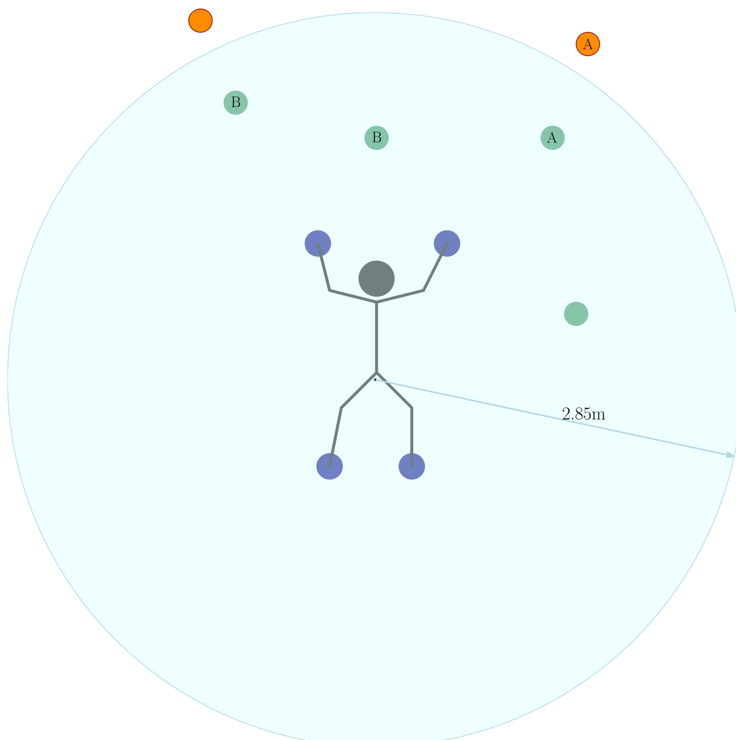


Figure 6.1: An illustration of the process of finding candidate holds and stances from the current stance. All green holds are in range, while the orange holds are out of range. Therefore stance A, i.e. all holds that are marked with A, would not be considered reachable, while stance B would and would be connected to the current stance in the graph.

6.2 Cost function

Once the stance graph is built, a path has to be generated that connects the rest stance to an end stance. However, not all paths that are possible are equal. Some paths use a winding path, while others make use of stances that would be hard to replicate by a climber. As climbers in real life are trying to minimize the energy expended during a climb, we can model the route finding as a least-cost path search. Here the cost of a route, λ_{route}, is based on a set of rules which, when violated, increase the cost of the path. These rules mimic best-practices in real life climbing which try to minimize energy expenditure.

We can break down the cost of a route into two parts, holding onto the current set of holds, and transitioning to a new hold or holds. The least cost path search represents this as two separate costs. A cost for each node in the graph λ_n and for each transition λ_e. Equation 6.2 calculates the cost of transitioning from σ_i to σ_j, two stances that are connected in the stance graph. The path finder can sum up all costs of the visited

[1]Guinness World Records, Farthest Dyno move in wall climbing - male (`https://www.guinnessworldrecords.com/world-records/farthest-dyno-move-in-wall-climbing-male`)

nodes to calculate the cost of a full path as in Equation 6.1. The sum of squares is used to ensure that the crux' difficulty weighs more heavily on the overall cost of the route. A crux is the hardest move in a route, so it would not be realistic to select a path that has a crux that is not necessary. Therefore this is penalized more heavily. Taking the sum of the cost of the entire path ensures that shorter routes are preferred.

$$\lambda_{route} = \lambda_n(\sigma_0) + \sum_{i=0}^{n-1} \lambda(\sigma_i, \sigma_{i+1})^2 \tag{6.1}$$

$$\lambda(\sigma_i, \sigma_j) = \lambda_e(\sigma_i, \sigma_j) + \lambda_n(\sigma_j) \tag{6.2}$$

Three different models will be tested against each other. A single baseline model will be used as the foundation of all subsequent models. In this baseline model, the limb configuration will not be considered, i.e. any stance using the same four holds will have the same cost in all configurations; the walls will be assumed to have a 90-degree angle with respect to the ground plane; and the holds will be assumed to all have the same shape and not have any dimensionality. The cost functions that make up this baseline model can be found in the following subsections. The enriched models in Section 6.5 will add extra cost functions to evaluate if the addition of extra information increases the predictive ability of the model.

6.2.1 Edge cost

The edge cost for the baseline model can be broken into two costs, the cumulative distance and the number of limbs changing.

The cumulative distance, in meters, between all holds, d_{dist}, is used as a cost λ_{dist} of transitioning between two stances. This cost is introduced to discourage making large moves to reduce the number of steps to get from the start to the finish in order to decrease the overall cost of the route. This means that the cost will be d_{dist}.

$$\lambda_{dist} = d_{dist} \tag{6.3}$$

If an appendage is not attached or if a detached appendage is attached, we can't calculate the distance between the two stances. Therefore we introduce a new variable, $d_{free2hold}$, which will be substituted as the distance in the instances where a free hand or foot is attached to a hold, or an appendage is detached from a hold.

The number of limbs changing, λ_{limbs}, increases the cost of a transition. As at least one limb has to change, this will not bring a cost with it. However, the more limbs change simultaneously, the more difficult a transition will be, so for each extra limb that moves, a cost, c_{limbs}, will be added to the transition cost. $\lambda_{limbs} = (\ell_{limbs} - 1) * c_{limbs}$, where ℓ_{limbs} is the number of limbs changing.

The full edge cost, σ_e, using these costs is calculated using Equation 6.4.

$$\lambda_e(\sigma_i, \sigma_j) = \lambda_{dist} + \lambda_{limbs} \tag{6.4}$$

6.2.2 Node cost

Finding a good set of holds to hold onto can be difficult depending on the particular set of holds. To reflect in the cost of the route, a node cost is added. This node cost consists of various cost functions. These functions are based on a set of best-practices, which, when disobeyed, make it harder to keep holding on. These cost functions are realized in the following set of cost functions.

It can be beneficial to have one limb free as it can show that you can make the next move. Therefore the cost of a free limb, λ_{free} will not penalize a single limb, but for every subsequent hand or foot that is detached the cost will increase, $\lambda_{free} = (\ell_{free} - 1) * c_{free}$, where ℓ_{free} is the number of free limbs.

A climber is most stable if each limb appendage is on a separate hold. However, as discussed earlier releasing one limb is needed to transition into another stance. Most of the time, three limbs is adequate to keeping balance. Even with two limbs a climber can preserve stability if opposing limbs are left on the holds, e.g. left hand and right foot. If both limbs that are released are on the same side, this can quickly lead to a barn-door situation. To represent this in the model a unique holds cost is added λ_{unique}. It is set to 0 if there are three or four unique holds, to c_{unique} if there are two opposing limbs on unique holds, $2 \cdot c_{unique}$ if the limbs are on the same side, and $4 \cdot c_{unique}$ if there is only one unique limb.

Another stance that can quickly lead to a barn-door is when all attached appendages are close to the same vertical axis. In this configuration, there is no way to counteract any torque around this axis which can, depending on the holds, quickly lead to slipping off of the holds. To find if this is the case for a stance, the centre of mass of the hold positions on the horizontal plane, COM_{hold}, is calculated using Equation 6.5.

$$COM_{hold} = \frac{\sum_{i=0}^{3} pos(x_i)}{h} \tag{6.5}$$

Where $pos(x)$ calculates the position of the hold in the xz-plane and h is the number of attached holds. If hold i is not attached a zero-vector is returned by $pos(x)$. If all distances from this centre of mass to each hold position is less than a threshold, t_{COM}, the alignment cost, λ_{align}, will be c_{align}, otherwise 0.

To be able to generate momentum to reach the next hold, the limbs have to not be entirely outstretched. Additionally, being too bunched up, i.e. having the limbs too close to each other, can also lead to a decreased ability to generate momentum. Therefore a cost, $\lambda_{stretched}$, is applied if the limbs are more than $t_{stretched}$ or less than $t_{min\_stretched}$ of d_{h2h}, d_{f2f} or d_{h2f}, depending on the limbs involved.

The full node cost is calculated using Equation 6.6.

$$\lambda_e(\sigma_i, \sigma_j) = \lambda_{free} + \lambda_{unique} + \lambda_{align} + \lambda_{stretched} \tag{6.6}$$

6.3 Path finding

Once the full stance graph is generated with a cost associated to each node and edge the path finding algorithm can start finding paths from the start stances σ_{start} to the goal stances σ_{goal}. As previously defined, a start stance is any stance where both hands are on the start holds(s). While a goal stance is where both hands are on the final hold.

Although the hands are locked to the start hold, the feet are free to find the best configuration to start from. However, to speed up the path finding, all four limbs are locked to a particular start stance. The same is done to a valid goal stance. This means that instead of having to find paths from all possible start stances to all possible goal stance, there is only one option. The algorithm finds these paths using a greedy algorithm, namely Dijkstra's algorithm [16].

For each path the algorithm finds, the total cost of the path, as well as the nodes visited, are stored. The total cost is used in the difficulty calculation in Section 6.4. The nodes visited are used to visualise the path that is found.

6.4 Route Difficulty

The cost function is necessary to finding the optimal path using the path finding algorithm. However, it could additionally give insight into the difficulty of a route. It can be argued that a problem of a higher grade forces moves that violate the rules laid out in Section 6.2. This will increase the overall score of the best path. Therefore it can be said that the difficulty is in a linear relationship with the cost of a particular path, which can be seen in Equation 6.7.

$$\mathcal{D} = \lambda_{route} \tag{6.7}$$

6.5 Enrichments

The difficulty equation introduced out in the above section uses a fair number of rules to come up with the score of a boulder problem. However, even with all this complexity it ignores several key features of a route that contribute to its difficulty. In this section, we lay out the enrichments that will be added to the difficulty calculation to include these features.

As described in Chapter 2, there are two different sets of properties that are defined: hold properties and climber properties. In the following sections the additional cost functions for these sets of properties will be explained.

6.5.1 Hold Properties

Hold Type

Although each hold has a unique shape, in general they can be categorised in the following five types, Jugs, Slopers, Pockets, Pinches, Crimps. In theory, any hold can be used for any difficulty, in practice, however, it is more likely to find certain types of holds at certain grades. Therefore we can assign each type of hold a certain cost, which increases with the general difficulty at which you can expect these holds. To calculate the difficulty of a stance, the type cost of each hold that is held is summed and an average is taken (Equation 6.8).

$$\lambda_{type}(\sigma_i) = \frac{\sum_{j=1}^{n} \tau(\sigma_{i,j}) \cdot c_{type}}{n} \tag{6.8}$$

Here $\tau(\sigma_{i,j})$ is the type cost of the j^{th} limb appendage of stance i. Each type has its own cost for a total of 5 different costs, τ_{jug}, τ_{sloper}, τ_{pocket}, τ_{pinch}, τ_{crimp}. c_{type} is a scalar value to increase the overall cost of the hold type. To get an average difficulty for a stance i, the total cost is divided by the number of attached holds.

Hold Vector

It is essential for a climber to keep their balance while climbing. To do this they need to counteract all forces acting upon them. Gravity is the most obvious force that acts upon the climber, however, they also need to neutralize all forces they exert on the holds. To model this we record the direction of the hold surface when measuring the bouldering routes. The hold surface is where the appendage will be placed when holding on to the hold, while the direction indicates the orientation of the limb. This orientation is converted into one of eight different orientations, four cardinal directions and four ordinal directions. Examples of two holds with their hold vectors and the eight orientations these vectors are converted into can be seen in Figure 6.2.

As a simplification our model will only be concerning itself with counteracting the gravity component. For this the vector for each hold in σ_i is compared to the three vectors that have at least part of their component pointing upwards, i.e. *NW*, *N* and *NE*.

Hold Distance

In the base model, the distance between holds is calculated from the centre of the hold. However, some holds can extend several tens of centimetres from this centre point. In certain cases this can lead to stances being culled during the graph construction phase (Section 6.1.2). To remedy this, the distance from the centre of the hold to the hold surface is measured. This distance is set as the radius of the hold. While culling the stances, instead of calculating the distance between two hold centres, the distance is calculated based on the distance between the closest extremes of the two holds.

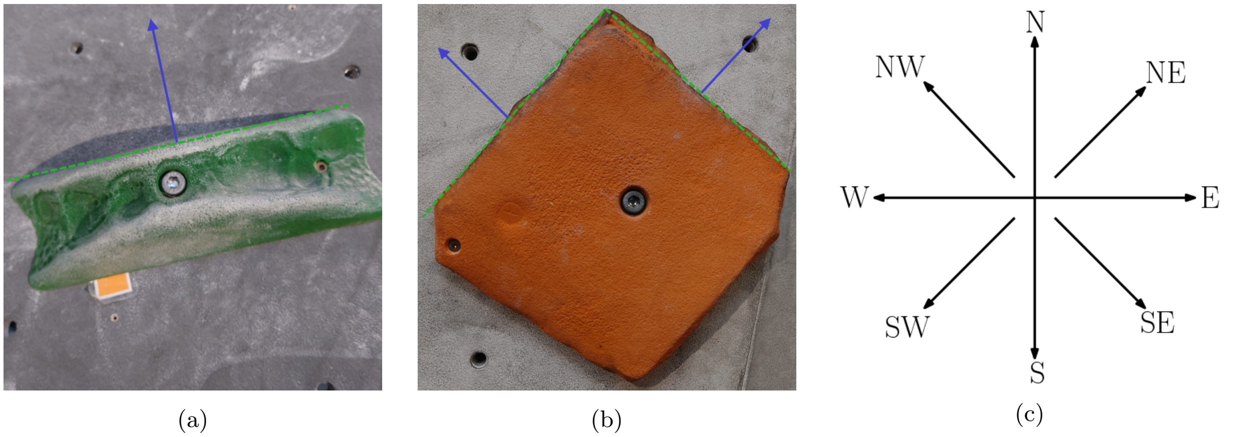


Figure 6.2: Examples of two hold and their hold vectors as well as the eight directions these hold vectors are converted into. The closest cardinal or ordinal direction for Figure a is vertically up, i.e. North. Figure c has two hold vectors one pointed close to *NE* and one pointing North-Westward.

6.5.2 Climber Properties

The climber properties add limb placement information. This information includes a distinction between the hands of the climber and their feet, as these two have very differing properties while climbing. In this section the cost functions that pertain to climber properties are explained.

Limbs Placement

Having access to the limb information, opens up a lot more avenues for adding rules to the model, both in the node cost calculation and the edge cost calculation.

The edge cost calculation can be enhanced with a body direction difference cost λ_{body_dir}. This cost compares the vector that goes from the average foot position to the average hand position in the starting and ending stance of a transition. The larger the angle between these two vectors, the more difficult a transition generally is. If the vector changes significantly from one stance to another, it can mean either the hand, feet, or both have moved a lot between stances. This kind of coordination move can be hard to execute.

The cost is set to c_{body_dir} if $\vec{v_i} \cdot \vec{v_j} > t_{body\_dir}$, 0 otherwise. Here t_{body_dir} is the threshold for the angle between the two vectors and $\vec{v_i}$ and $\vec{v_j}$ are the vectors from the average foot position to the average hand position in stance i and j respectively. If both feet or both hands are not attached to a hold, an average position can't be calculated. In these situations, the vector is assumed to go straight up. This is the most sensible, as gravity will pull down any limb that is not attached.

There also are several node costs that are associated with the way the limb appendages are placed on the holds, these are a match cost, cross cost, a cost for having the feet above the hands, and an orientation cost.

A hold is said to be matched if two limb appendages are on the same hold, which can be useful in many situations during a climb. Matching two hands is often more preferable than matching feet, because matching feet can result in an unstable position where the centre of mass is above a fulcrum point. To take this into account a matching cost, λ_{match}, is included in the node cost. This will be c_{feet_match} if the feet are matching and 0 otherwise.

Related to the matching cost is the cost for crossing limbs on the horizontal axis, λ_{cross}. Crossing limbs often leads to unstable positions where the hands or feet naturally want to pull the climber away from the wall. This cost, added when either a foot crosses another foot or a hand crosses the other hand on the horizontal plane, tries to mitigate this. $\lambda_{cross} = c_{cross}$ if the hands or feet are crossed along the horizontal axis, otherwise it is 0.

Although sometimes necessary, climbing with the feet above the hands uses a lot of energy, and as such should not be used unless absolutely necessary. Therefore a cost, λ_{feet_high}, is added if any foot is higher than the highest hand. This will be c_{feet_high} if any foot is higher than the highest hand, otherwise 0. If both hands are detached, we assume they are above the feet and λ_{feet_high} will be 0.

To further discourage hanging upside down, an orientation cost $\lambda_{orientation}$, is added. This calculates the angle between the up direction and the vector from the average foot position to the average hand position. If the angle exceeds $t_{orientation}$, the cost will be set to $c_{orientation}$ and 0 otherwise.

6.6 Parameter tuning

Sections 6.2 and 6.5 introduced a myriad of parameters that need tuning to result in plausible paths found by the path finder. This section explains how we tuned these parameters and gives an overview of the final values for each parameter.

There are various ways to tune these parameters to get correct results. Naderi et al. [5] seem to have used expert knowledge to find the set of values for their high level path planner. However, we can take advantage of the fact that we have real-world data for which the actual difficulty grade is known. With this data we can train our model to output correct grades when supplied with a route.

Various training methods were explored. For the final training we decided to train the model by randomly sampling different values for each of the parameters in a given range. The parameters were chosen in the range [1, 10]. This was to ensure that no parameter would be set to 0 and therefore not add to the overall score. The maximum range was chosen to be in the same magnitude as λ_{dist}. For the enrichment models, the values found during training of the baseline were fixed, while the new values were being trained.

The training algorithm requires us to supply a fitness function, which it used to calculate the performance of the model using the random values for the parameters. The fitness function examines all pairwise combination between two routes. Ideally passing the pair of routes through the model should result in the higher graded route to have a higher difficulty score. The number of correct classifications between these pairs is the fitness for a particular point in the parameter space.

All threshold values as well as the type cost for each of the different hold types were set manually. All other cost values were trained using the above mentioned training function. A table of all fixed values can be found in Table 6.1.

To see the effect of each of the different routes, the training is performed 13 times, each time with a different route removed from the training set. This results in a set of 13 parameter values for each of the different cost functions.

Parameter	Value	Parameter	Value
d_{h2h}	1.8	t_{body_dir}	$\frac{\pi}{2}$
d_{f2f}	1.5	$t_{orientation}$	$\frac{\pi}{4}$
d_{h2f}	2.3	τ_{jug}	0.3
$d_{free2hold}$	1.0	τ_{sloper}	0.5
t_{COM}	0.4	τ_{pocket}	0.5
$t_{stretched}$	0.9	τ_{pinch}	0.6
$t_{min\_stretched}$	0.2	τ_{crimp}	1.0

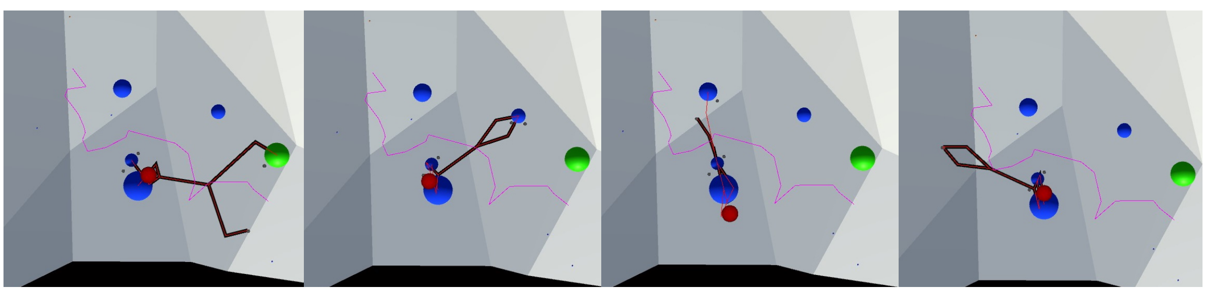
Table 6.1: An overview of all values for all of the fixed variables used throughout this chapter.

7 | Results and Evaluation

In this chapter we will utilize the model constructed in the previous chapter to generate difficulty estimations for the captured routes. Using the generated data the ability to correctly predict the difficulty of a route will be compared between the three models.

7.1 Baseline Model

As proposed in the previous chapter, we trained the model 13 times. The initial maximum fitness for each of the rounds started around the 60% mark, which increased to ∼70% after training. In the scatterplots in Figure 7.2 we can see a rising trend in most training cases except for the case where Class_E-2 is remove from the training set. Looking at the predicted path and difficulty of the route, it is always given a very high path cost and subsequently a large difficulty. In nearly all graphs we can see its difficulty to be close to double the second-hardest route. The scatterplot also reveals that harder routes are not by definition given a higher score, e.g. Class_C-1 a route in Class A is given a higher difficulty score than all other routes except for one.



(a) (b) (c) (d)

Figure 7.1: An example of a route where the model assumes the climber should climb upside down. The green sphere is the start hold, the orange sphere is the end hold. Blue sphere are all other holds. A stick figure represents the stance of the climber. The pink line shows the median position of all holds in each of the stances linked together.

As this base model does not differentiate between the different limbs, i.e. hands and feet, there are several routes where the found path leads the climber to go upside down, as seen in Figure 7.1. Visually this is incredibly jarring as this would not be something a climber would attempt on the actual route. However, as far as the model is concerned this is no harder than moving right side up.

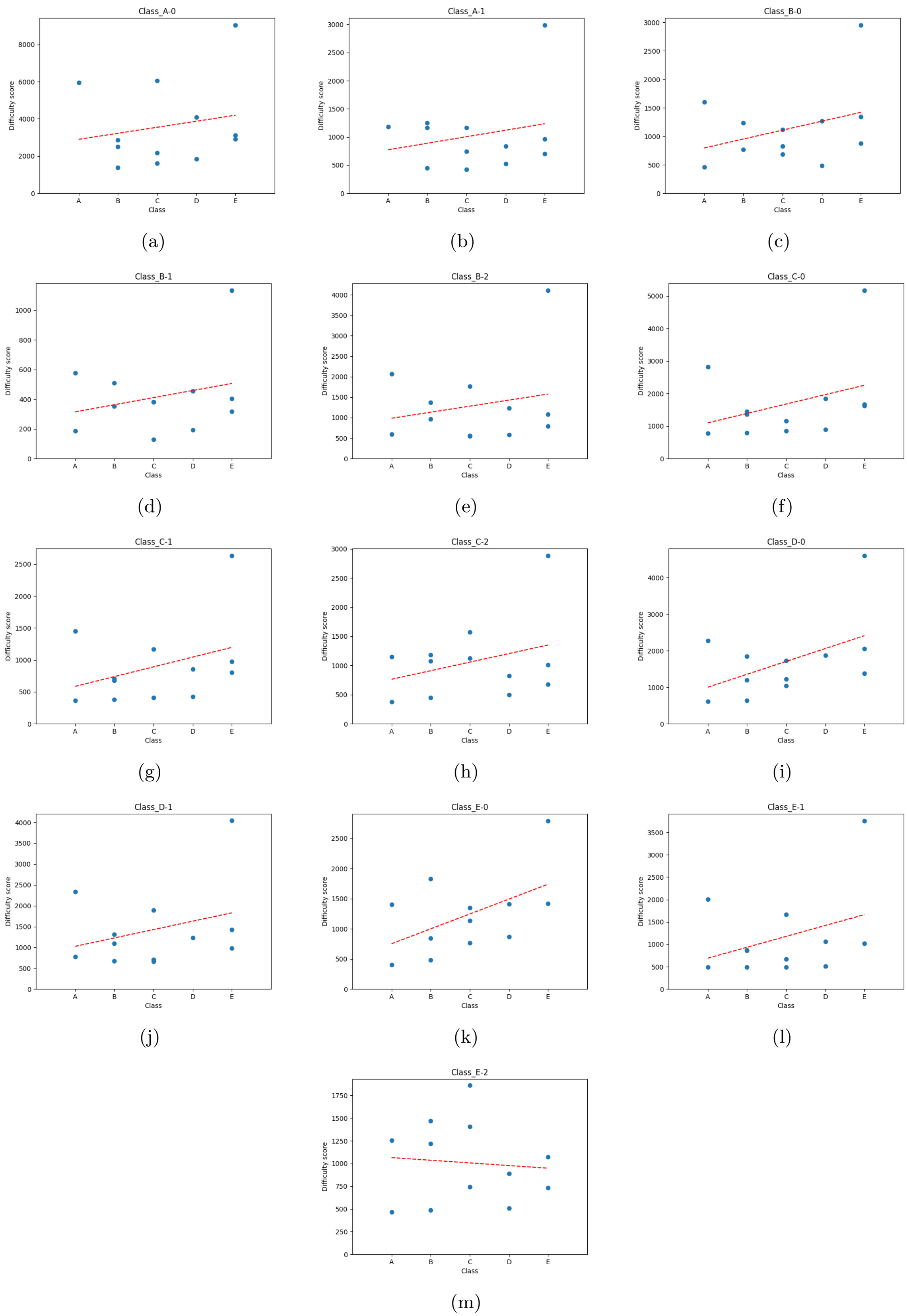


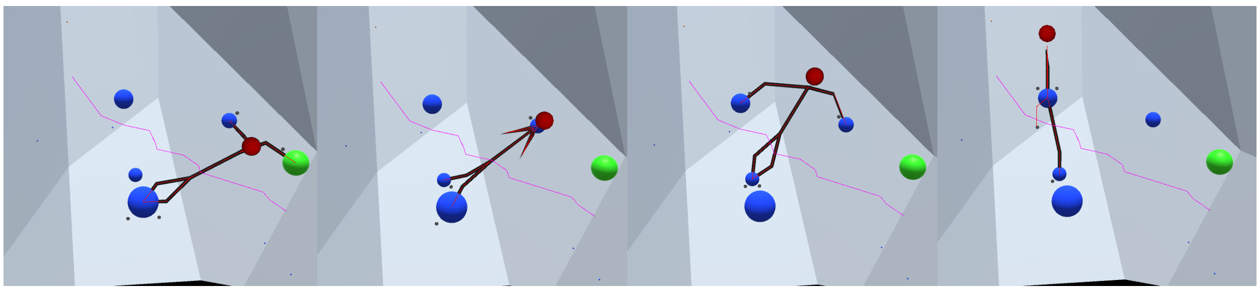
Figure 7.2: Scatterplot of baseline model.

Figure 7.3: Difficulty generated by the baseline model of all routes on a number line. Routes are distinguished by their colour as explained in Chapter 5. The five Classes are shown in five different colours. These are Green, Yellow, Orange, Black and Pink from easiest to hardest.

7.2 Climber Properties

As mentioned in Chapter 6 adding the climber properties makes the model aware of the difference between hand and feet. Which enables the model to add limb-related cost functions.

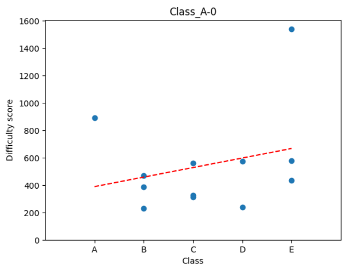
These newly added costs can be noticed in the different way the model finds a path for the route in Figure 7.1. The baseline model's path made an unnecessary flip upside down. A real climber would avoid unnecessary difficulty during a climb as this exhausts the climber more quickly and increases the risk of injury. However, with the extra data from the climber properties, the new path seen in Figure 7.4 finishes the climb without having to flip upside down. This makes the resulting path look more realistic.



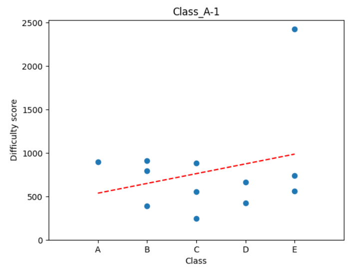
(a) (b) (c) (d)

Figure 7.4: With climber properties added to the model, the climber is aware of the difference between hands and feet, and is punished if they go upside down. This forces the model to find an alternative route, where the climber does not go upside down.

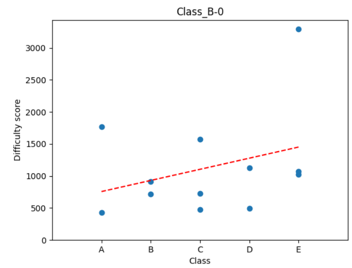
Unfortunately, the increase in realism in the found path does not translate in a significantly higher fitness once the model is trained with the new cost functions. Figure 7.10 shows the boxplot of all fitness values captured during training for the three models. In this figure we can see that the baseline's median is around 60% with a maximum value occasionally reaching 70%. The model that includes the climber properties does not appear to increase these metrics substantially.



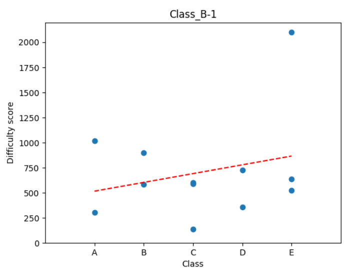
(a)



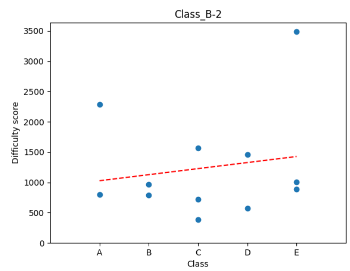
(b)



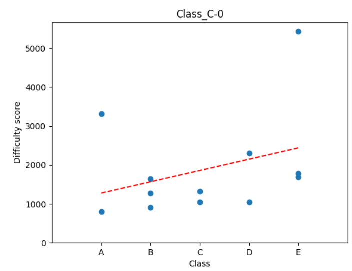
(c)



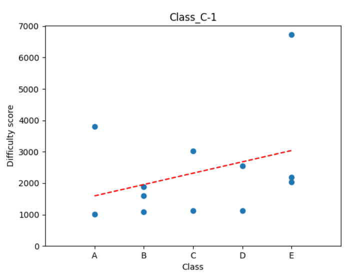
(d)



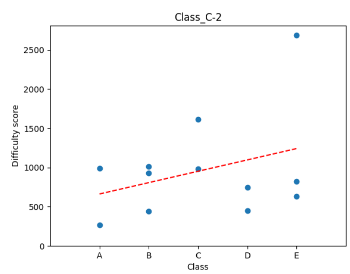
(e)



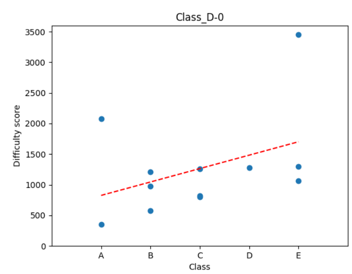
(f)



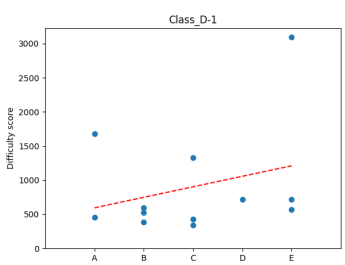
(g)



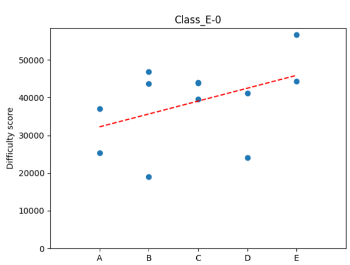
(h)



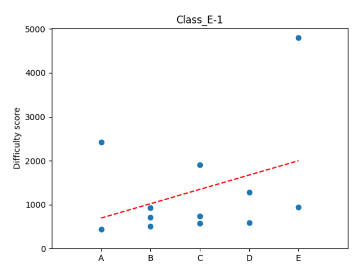
(i)



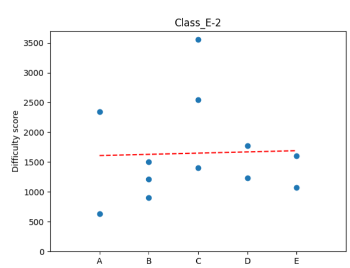
(j)



(k)



(l)



(m)

Figure 7.5: Scatterplot of climber properties model.

Figure 7.6: Difficulty generated by the climber properties model of all routes on a number line. Routes are distinguished by their colour as explained in Chapter 5. The five Classes are shown in five different colours. These are Green, Yellow, Orange, Black and Pink from easiest to hardest.

7.3 Hold Properties

Adding hold properties on the other hand does appear to have an impact on the model's ability to predict the difficulty of a route. In Figure 7.10 we can see that the average fitness of all models that incorporate hold properties is higher than their baseline counterpart. Additionally, the maximum value is also consistently higher.

Comparing the various scatterplots between the baseline and the hold properties tells a similar story. Although the maximum difficulty score still appears fairly erratic, the minimum difficulty score for each class does appear to increase than both the baseline and the climber properties models.

Figure 7.7 shows the percentage of the total cost that can be attributed to the edge cost for the different models. We can see that even though there are only two edge costs, as opposed to the four node costs, the total cost is mostly dictated by the edge cost. This can be mainly attributable to the distance cost (λ_{dist}), as all other costs only apply when a rule is violated, while the distance cost is a static cost for each move. However, we can see a downward trend in the enriched models. The climber properties add several more node costs, which means there is a bigger chance that one is violated, which reduces the overall share of the edge cost in the total cost. An even higher reduction can be seen in the model that includes hold properties. This is due to the hold cost. This is another static cost which is, this time, added to the node cost for each node that is visited.

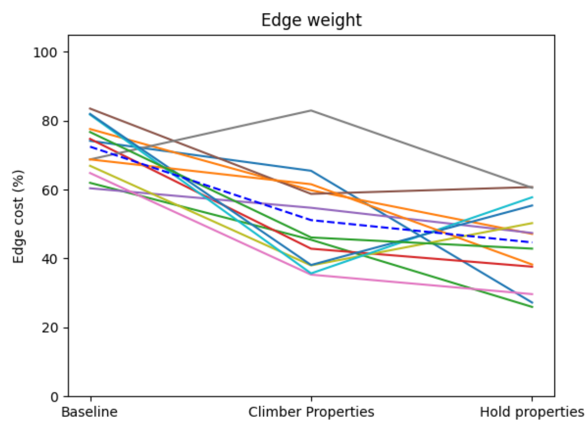


Figure 7.7: The percentage of the edge cost for the different routes in the three different models. The dashed line is the average of all routes.

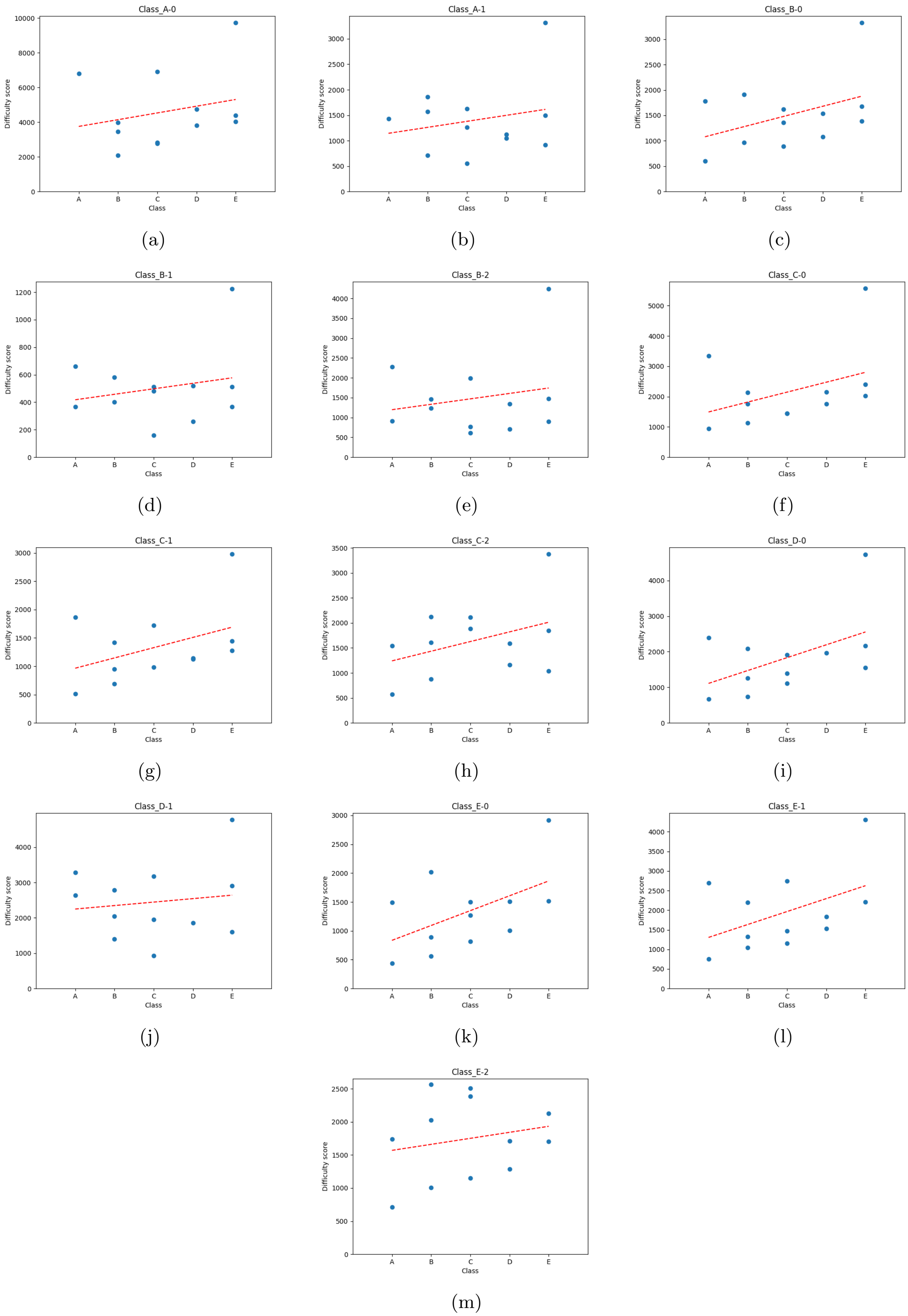
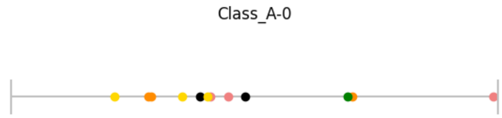


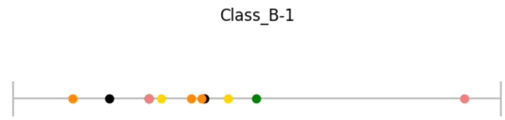
Figure 7.8: Scatterplot of hold properties model.



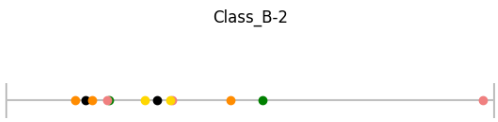
(a)

(b)

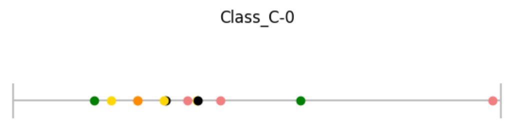
(c)



(d)



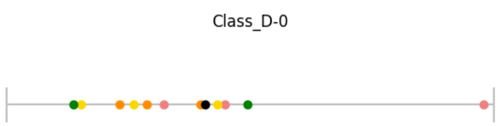
(e)



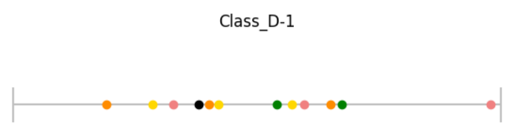
(f)

(g)

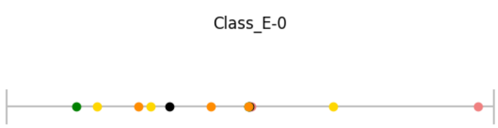
(h)



(i)



(j)



(k)

(l)

(m)

Figure 7.9: Difficulty generated by the hold properties model of all routes on a number line. Routes are distinguished by their colour as explained in Chapter 5. The five Classes are shown in five different colours. These are Green, Yellow, Orange, Black and Pink from easiest to hardest.

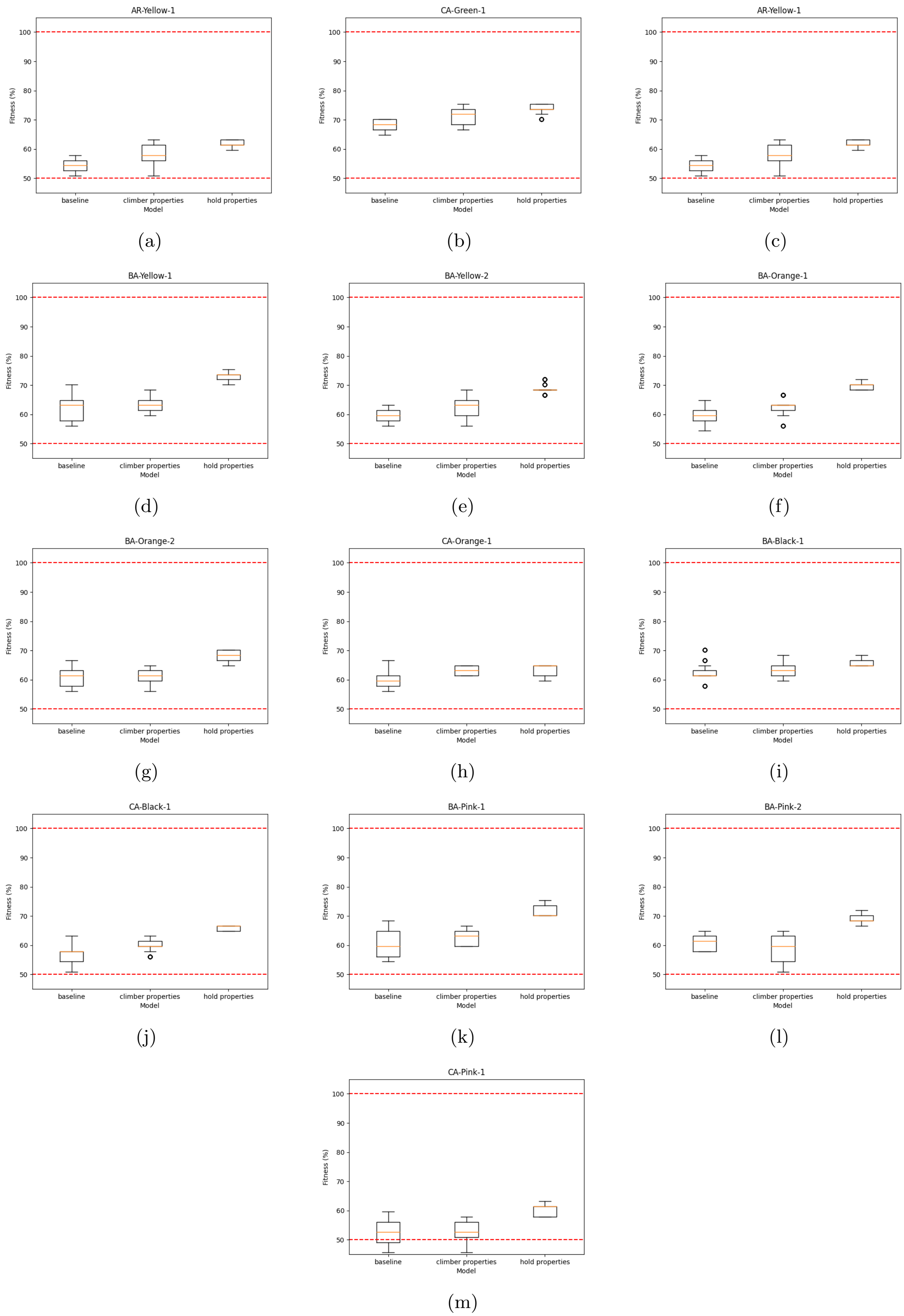


Figure 7.10: Boxplots comparing the fitness of the different models. Fitness is converted to a percentage. The horizontal red dashed lines are the 50% and 100% marks.

8 | Discussion

Looking at the boxplots in Figure 7.10, the baseline is already able to extract some characteristics from the model. Combining this model with the hold properties increases its predictive abilities significantly. However, the scatterplot and colour line tell a more nuanced story. In the scatterplots we can see that the minimum difficulty values do seem to exhibit a rising trend in most cases, however, the maximum values are about as erratic as the baseline model. This could mean that our model is able to learn that harder routes violate the best-practices more often, but fails to modulate the severity for each individual case.

A similar phenomenon can be seen in the colour lines. Except for the one Class E outlier, most routes are fairly close to each other with regards to their difficulty score. As previously mentioned, off-by-one errors are fairly common in climbing route classification. However, in the colour line we can see a plethora of routes that are flipped with multiple classes, and as such are classified much easier/harder than they actually are.

Observing the generated routes, several of these misclassifications can be explained. There is a Class A route that is consistently rated much higher than it should be. Visual inspection of the generated route explains why. The route setters intended a rock over manoeuvre for this route. From the initial position it is impossible to move to the next intended stance, however, once the centre of mass is moved to the other side, it is possible. Our model, unfortunately, does not have the ability to simulate this kind of move. Therefore it failed to find this manoeuvre and instead came up with an elaborate move set that added a significant amount of cost to the route. We anticipated this could happen and as such recorded the size of the holds so that we could mitigate this problem. However, from our testing, we can conclude that this problem is not as easily solved.

The Class E route that consistently is classified as the most difficult route by far, is set to only contain two dynamic moves to get from the start to the end. Our model adds a large cost to this kind of dynamic moves, to discourage these moves if there are easier alternatives. However, as this route consists of only this kind of move, it overestimates the difficulty of the route.

Our results are based on a very small dataset. From the data we have collected we can say that the hold properties increases the predictive abilities of the model. However, with additional route data, the confidence in our model can be increased.

9 | Conclusion

We have proposed and demonstrated a novel approach to difficulty estimation of bouldering routes set in an actual bouldering gym.

This thesis took a systematic approach to modelling a bouldering route and the various paths that can be used to climb the route. First, a baseline model was constructed that abstracted the climber to a stance, a 4-tuple representing the four limb appendages. This stance representation was then used to create a stance graph, where the nodes represented a specific stance, and the edges a transition between two stances. Edges were created if two stances were close enough together. Once the graph was generated a path planner was used to find the least cost path through this graph. The cost for each node and edge was calculated by a set of cost functions that were formulated using a set of best-practices in bouldering.

Second, two enrichment models were introduced that expanded the cost functions, these models added information pertaining to the holds, and information on the limbs of the climbers respectively. These extra costs changed the cost of the nodes and edges and changed the paths the model climber took to get to the end of the different routes. Which, in turn, changed the estimated difficulty of the routes.

Lastly, the models were assessed using various routes recorded at a local bouldering gym. These routes were systematically recorded and digitized after which it was supplied to the three models. This resulted in a difficulty score for each of the routes. To quantify the model's ability to accurately determine the difficulty of a bouldering route a set of experiments were run. During each experiment, a single route was removed from the set and the model was optimized using the remaining routes. A fitness function was introduced to optimize the models. Training the baseline model resulted in a fitness of around 65%, which means that 65% of all pairs in the training set were correctly ordered in difficulty. The climber properties model did not increase this fitness significantly. However, the model that took into account hold properties saw its average fitness rise to around 75%.

Using the results obtained in Chapter 7 we can now attempt to answer our original research questions. The baseline model was the simplest of the tested models, and the two other models were built on top of this model. This way we could see if adding complexity to the model also increased the accuracy of the model. From our results, we can conclude that adding additional climber properties does not significantly improve the predictive power of our model. However, adding hold properties does increase the performance of the model.

10 | Limitations and Future work

Our models focussed on a small set of routes due to the lengthy process of extracting a route at the gym. To improve confidence in our findings a larger database of routes should be created and used in our models. However, even in our small dataset, we have already highlighted some of the shortcomings of a set of rules based on expert knowledge. Therefore a larger dataset could also enable novel ways to extract node and edge cost data from the data itself. A larger dataset would also increase the training process's running time, which was already a bottleneck.

From our results, we can conclude that there is no reliable way to predict how adding cost functions will affect the predictive abilities of the model. Even though the visual results looked more natural than the baseline, the climber properties did not significantly increase the model's ability to predict the difficulty of the routes.

Our current data capturing method is a tedious and labour intensive process. An interesting avenue of research could be to automate the extraction of this route data to increase the number of routes in the database. As we have shown, a model that takes into account the position of the holds and the type of the holds has the potential to determine the difficulty of a route. This data could potentially be automatically extracted using computer vision. After which it would be trivial to convert it to the format the model expects.

The biggest limitation of our model is the time and space complexity of large interconnected graphs. On routes with more than $\sim$10 holds that are all quite close to one another, the resulting graph is very interconnected. This results in long processing times for the model. This means that larger routes, such as rope climbing routes, would require further pruning to reduce the number of connections.

Bibliography

[1] Nick Draper, David Giles, Volker Schöffl, Franz Konstantin Fuss, Phillip Watts, Peter Wolf, Jiří Baláš, Vanesa Espana-Romero, Gina Blunt Gonzalez, Simon Fryer, Maurizio Fanchini, Laurent Vigouroux, Ludovic Seifert, Lars Donath, Manuel Spoerri, Kelios Bonetti, Kevin Phillips, Urs Stöcker, Felix Bourassa-Moreau, Inmaculada Garrido, Scott Drum, Stuart Beekmeyer, Jean-Luc Ziltener, Nicola Taylor, Ina Beeretz, Franziska Mally, Arif Mithat Amca, Caroline Linhart, and Edgardo Abreu. Comparative grading scales, statistical analyses, climber descriptors and ability grouping: International rock climbing research association position statement. *Sports Technology*, 8(3-4):88–94, 2015.

[2] Caleb Phillips and Elizabeth Bradley. Strange beta: Chaotic variations for indoor rock climbing route setting. *AIP Conference Proceedings*, 1339, 2010.

[3] Kin Ho Lo. Embedding and generation of indoor climbing routes with variational autoencoder. *CoRR*, abs/2009.13271, 2020.

[4] Jonas Pfeil, Jun Mitani, and Takeo Igarashi. Interactive climbing route design using a simulated virtual climber. In *SIGGRAPH Asia 2011 Sketches*, SA '11, pages 2:1–2:2, 2011.

[5] Kourosh Naderi, Joose Rajamäki, and Perttu Hämäläinen. Discovering and synthesizing humanoid climbing movements. *ACM SIGGRAPH Computer Graphics*, 36, 2017.

[6] Daichi Katsura, Naoto Nishino, Daisuke Sakamoto, and Tetsuo Ono. Climbing pathfinding with the holds and a decision method of the difficulty level of the holds. In *International Workshop on Advanced Imaging Technology (IWAIT) 2021*, pages 288 – 293, 2021.

[7] Lindsay Kempen. A fair grade: assessing difficulty of climbing routes through machine learning. *Formal Methods and Tools, University of Twente*, 2018.

[8] Alejandro Dobles, Juan Carlos Sarmiento, and Peter Satterthwaite. Machine learning methods for climbing route classification. *Web link: http://cs229. stanford. edu/proj2017/finalreports/5232206. pdf*, 2017.

[9] Cheng Tai, Aaron Wu, and Rafael Hinojosa. Graph neural networks in classifying rock climbing difficulties. 2020.

[10] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[11] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. *CoRR*, abs/1809.05679, 2018.

[12] Yi-Shiou Duh and Ray Chang. Recurrent neural network for moonboard climbing route classification and generation. 2021.

[13] Dean Scarff. Estimation of climbing route difficulty using whole-history rating. *arXiv preprint arXiv:2001.05388*, 2020.

[14] André Ebert, Kyrill Schmid, Chadly Marouane, and Claudia Linnhoff-Popien. Automated recognition

and difficulty assessment of boulder routes. In *International Conference on IoT Technologies for HealthCare*, pages 62–68. Springer, 2017.

[15] Caleb Phillips, Lee Becker, and Elizabeth Bradley. strange beta: An assistance system for indoor rock climbing route setting. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013130, 2012.

[16] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.