MATHEMATICAL GAME MODELING AND OPTIMIZATION

Optimizing Player Decisions in Old School Runescape

Palfore

7^{th} November, 2020

Contents

Introduction

Runescape (RS) is a popular *Massively Multiplayer Online Role-Playing Game* (MMORPG) that was first publicly released on January 4'th, 2001 by the video game developer Jagex Limited. Ranked as the 5'th most popular MMORPG in 2020 by several sources, this game's unique mechanisms and game play make it still successful nearly 20 years after it's incarnation [1, 2, 3]. On the 20th of November 2012, a total overhaul to the game's combat system - an integral part of gameplay - caused a great divide among it's players. As a result, the game bifurcated into two versions: Runescape 3, and *Old School Runescape* (OSRS). The latter was released on February 22, 2013 and reverted to the old mechanics. The relative player counts over time can be found in Ref. [4]. OSRS currently contains the majority of players. To limit the already-large scope, this text will only focus on that version.

In typical role-playing fashion, the majority of game play centers around fighting monsters and bosses, training skills, completing quests, playing mini-games, and collecting items. This game is played over the course of months or years. In a few years, there will even be some players who have played for *decades*. As the player base gained a more comprehensive understanding of the game, their mentality has generally shifted from one of discovery to one of *efficiency*. Many tools have been created with the goal of improving player efficiency, optimizing game play, and maximizing success in difficult challenges.

There are 23 skills that a player can train [5]. A player is rewarded experience for certain actions related to a given skill. For example, cutting an `Oak Tree` yields 37.5 experience per log chopped. 83 Experience is required to go from level 1 to level 2, while reaching the maximum level of 99 requires 13,034,431 total experience [6]. The experience required to level up increases exponentially - hence the drive for efficiency [6]. There are several combat skills that directly influence a player's fighting ability. Quests are often completed for the special items, new training methods, and experience rewards they provide. They have skill requirements and often make use of combat in defeating difficult bosses. And so even in this basic overview, the complexity of the interactions and relations between different actions a player can perform becomes apparent.

Figure 1: Some relevant interfaces/images that play a central role in game play. The skill panel (left) shows the player's levels in the 23 skills along with their total level (Image slightly modified from Ref. [5]). The combat skills, attack, strength, defence, ranged, prayer, magic, and health (in the middle column), are respectively outlined in red. The quest panel (middle) shows the player's quests that are completed, in progress, and not started (green, yellow, and red, respectively. Image from Ref. [7]). A character that a player would control in a 3D world is shown on the right.

To understand player decisions and optimize them, we will be mathematically modeling the in-game mechanics. A surprising variety of mathematical concepts and techniques will be encountered. Additionally, algorithms derived from computer science are required to solve some of these problems. This serves as an exciting *field* to explore, with some very interesting results and visuals. Some of the details require high level mathematical solutions/descriptions. With the interest of being digestible by a broad audience with varying proficiencies, many arduous solutions are moved to the appendix at the end of each part.

This text accompanies several additional resources:

1. Python open-source codebase, *OSRSmath*: https://github.com/Palfore/OSRSmath.
2. Video Series, *Optimizing Runescape*: https://www.youtube.com/watch?v=7N9UJX70Z5I&list=PLm3INE_scU5s8NQWmw0fxKtA_6SVxDOc7&ab_channel=Palfore.
3. Discord Chatroom: https://discord.gg/4SXcKQh

Part I

Combat

List of Combat Related Terms and Associated Values

1. Combat Class: One of [`melee, ranged, magic`]
2. Attack Type: One of [`stab, slash, crush, ranged, magic`]
3. Combat Style: The name of the attack.
   (a) For the `melee` combat class: One of [`punch, kick, chop, hack, smash, block, pound, pummel, slash, lunge, jab, swipe, fend, spike, impale, reap, flick, lash, deflect, bash, focus, scorch`]
   (b) For the `ranged` combat class: One of [`accurate, rapid, longrange, short fuse, medium fuse, long fuse, flare`]
   (c) For the `magic` combat class: One of [`spell, spell (defensive), blaze, accurate, longrange`]. The latter two only apply to `Powered staves`.
4. Attack Style:
   (a) For the `melee` combat class: One of [`accurate, aggressive, defensive, controlled`]
   (b) For the `ranged` combat class: One of [`accurate, rapid, longrange`]
   (c) For the `magic` combat class: One of [`standard, defensive`]
5. Attack Speed: The number of ticks between attacks. Integer between 1 and 15.
6. Attack Interval: The number of second between attacks. Real number between 0.6s and 9s.
7. Attribute: When referring to an opponent/monster, their `attribute` is one of [`Demon, Draconic, Fiery, Kalphite, Leafy, Penance, Shade, Undead, Vampyre, Xerician`]. In addition, we expand this to also include properties like: [`On slayer task, In wilderness`].

Chapter 1

Overview

In this chapter, we will discuss the various factors involved in combat. We will consider combat in two stages. The first considers an autonomous fight in which the player performs no actions once the initial conditions of the fight have been specified. Analyzing this system will allow us to calculate quantities like the expected number of attacks required to defeat an opponent, and the probability of winning a fight. The second considers active player decisions that occur during combat. This will allow us to investigate the effect of performing actions on the aforementioned quantities. *Policies* may be defined to mathematically model a player's decision. As an example, a player may use a healing item any time throughout a fight. To handle this, we can consider a specific policy whereby the player will use a healing item when health is below some threshold. Investigating this threshold will give us insight into how players should use healing mechanics.

It is interesting to note that although the descriptions of in-game mechanics likely have no real-world connections (since they are somewhat arbitrarily decided by the game's developers), the mathematics that can be applied to the dynamic variables resulting from these mechanics can actually be applied and generalized to real-world settings. We will begin with a discussion of the most relevant mechanics, however there is an additional large body of information that can be found on the Official Wiki that provides a greater overview.

1.1 Autonomous Mechanics

1.1.1 Combat Skills, Combat Triangle and Attack Styles

Combat is built around the so-called *Combat Triangle* which describes the relation between the three classes of combat in the game [8]. A Melee fighter makes use of close quarters combat, typically wielding swords, daggers, halberds, etc. A Ranged fighter makes use of bow and arrow, crossbows, and thrown objects to deal damage at a distance. Finally, a Mage will make use of staves and magical spells to do damage, also at a distance. The combat triangle refers to the notation that melee users are (generally) weak to magic, which is weak to ranged, which is weak to melee, and is depicted in Fig. 1.1.

Some skills provide benefits to all fighters, while others are specific to the style:

1. Attack, L_a: Increases the accuracy of a melee attacker.
2. Strength L_s: Increases the maximum damage a melee attacker can do in a single attack.

3. Ranged L_r: Increases the accuracy and maximum damage of a ranged attacker.

4. Magic L_m: Most spells have a constant damage (with more powerful spells being unlocked at higher levels), also some scale with magic level. Accuracy however is generally increased with higher magic. In addition, defence against magical attacks is partially determined by the player's magic level.

5. Defence L_d: Decreases the probability that the opponent will have a sucessful attack.

6. Prayer L_p: Acts as a depleteable resource that can boost combat skills.

7. Hitpoints L_h: Increases the amount of damage a player can receive before they lose a fight.

The set of all combat levels is denoted $\{L\}$.

Every weapon has a set of *attack styles* that allow a player to change which combat skill they train. In addition, the attack style may provide a small bonus to combat. Prayer is the only skill that cannot be trained directly through combat. Hitpoints is another exception in that a proportion of the experience awarded to the skill associated with the player's attack style is given to hitpoints.

1.1.2 Equipment Bonuses

Let's begin discussing a fighter's equipment by defining an *item*, $\mathcal{I}$. Equipable items can be worn in one of 11 slots. We let $\mathcal{I}^{\text{slot}}$ represent the item in a given *slot*, where

$$\text{slot} \in \{\text{head, cape, neck, ammo, weapon, torso, shield, legs, hands, feet, ring}\}. \tag{1.1}$$

Each item has some associated equipment bonuses. Most of these are constant, however some bonuses are conditional. The constant bonuses can be represented as a vector:

$$\begin{aligned} \vec{\mathcal{I}}_c^{\text{slot}} = (&A_{\text{stab}}, A_{\text{slash}}, A_{\text{crush}}, A_{\text{magic}}, A_{\text{ranged}}, &(1.2)\\ &D_{\text{stab}}, D_{\text{slash}}, D_{\text{crush}}, D_{\text{magic}}, D_{\text{ranged}}, &(1.3)\\ &S_w, S_r, S_m, P, w, r). &(1.4) \end{aligned}$$

There are many terms to define, so we will explain them here. A, D, and S refers to the attack, defensive, and strength bonuses, respectively. The attack and defence bonuses are associated with the different attack types, while the strength bonuses are associated with the combat class [SEE LIST OF TERMS]. The first three attack and defence bonuses listed are associated with melee combat, the last two are associated with magic, and ranged, respectively. There is a strength bonus associated with each combat class. In the order above we have: melee/warrior, ranged, then magic. The prayer bonus, P affects how long bonuses from the prayer skill can last without recharging. w is the weight of the item. Finally, if the item is a weapon, r is the attack rate given by $r = 1/s$, where s is the weapon attack speed. If it is not a weapon, $r = 0$. Note that we use the rate since every other bonuses improves fighter ability. This allows us to use a basic comparison operator (at the cost of using real numbers instead of integers).

The total constant equipment bonuses that a fighter has, E_c is given as the sum over all the slots,

$$\vec{E}_c = \sum_{\text{slot}\in\{\text{slots}\}} \vec{\mathcal{I}}_c^{\text{slot}} \mathcal{E}. \tag{1.5}$$

The in-game interface indicating these values is shown in Fig. 1.1. There are a number of conditional effects that may not appear in this interface.

The conditional bonuses can be further divided into special/attribute bonuses and equipment set bonuses. Monsters may have a particular weakness due to their so-called attribute. For example, a `Iron dragon` would be `dragonic`, and `dragonbane` weapons would provide an accuracy and damage multiplier. In this sense, we can consider these bonuses to be dependent on information that the item itself does not know, and so we represent these special bonuses as an operator $\hat{\mathcal{I}}_s$. When acting on a fighter's environment $\mathcal{E}$, these bonuses become concrete:

$$\vec{E}_s = \hat{\mathcal{I}}_s \mathcal{E} \tag{1.6}$$

Set effects are also similar except that they are conditional on equipment the player is wearing. For this reason, (and the fact that there are other special cases), we group all these effects into the special bonus operator, $\hat{\mathcal{I}}_s$ from above. The total bonuses from all the player's items can be represented as:

$$\vec{E} = \vec{E}_c \cup \vec{E}_s \tag{1.7}$$

$$= \sum_{\text{slot } \in \, \{\text{slots}\}} \vec{\mathcal{I}}_c^{\text{slot}} \cup \hat{\mathcal{I}}_s \mathcal{E}, \tag{1.8}$$

The definition of environment is intentionally vague, as there are a myriad of conditions, essentially limited only by developer imagination and infrastructure. Some of these conditions/dependencies include: attacker & opponent equipment & levels, attack style (which implies combat class), whether a particular `Diary` is completed, the `attribute` of the opponent, and so on. The elements and details of $\vec{E}_s$ are also purposefully vague, as there is an additional caveat that makes these a bit trickier to handle both mathematically but more-so programatically. Unlike the constant bonuses, which can be added together, special bonuses are generally multiplicative but also make use of intermediate *flooring*. This makes the special bonus operator non-commutative, since the order does effect the rounding.[1] This means that a vector representing special bonuses would essentially have as many elements as the number of special items! So it is often easier to work on each bonus type with different methods. For this reason, special effects and constant bonuses are treated independent, making the union above more symbolic than practical.

1.1.3 Ticks and Attack Speed

At a fundamental level the entire game operates on a tick-based system. Every 0.6 seconds (called a tick) the game updates. This discretizes the possible game states, and typically means we will be dealing with sums in place of integrals, and recursive equations in place of differential equations.

Once an attacker begins combat with an opponent, the fight continues until either is defeated, or one runs away. The attacks occur at an interval associated with the weapon. Different weapons have different *Attack Speeds*, typically between 3-9 game ticks (1.8s - 5.4s). The attacker's attack speed $\mathcal{A}|s$, is the number of ticks between attacks. On each attack, the player's accuracy will determine the probability of a *successful hit*. On a successful hit, a number between 0 and the player's maximum hit will be uniformly sampled as the damage the player does.

A notable consequence of this tick-based system is that a series of precise player actions known as tick-manipulation allows players to perform multiple actions in a single tick, or to take advantage of mechanisms like tick-eating, allowing a player to survive otherwise fatal attacks.

[1]The *specific* ordering of the flooring operations is taken from ref. [bitter-dps calc]. Although, this author is unsure if that ordering is arbitrary, but we assume not. [Reference max hit section?]

Figure 1.1: The attack styles (left), equipment slots and associated equipment bonuses (middle) along with a depiction of the combat triangle (right). The attack styles for the `Dragon Claws` are Chop, Slash, Lunge, Block and give experience specifically to Attack, Strength, shared, and Defence, respectively. Shared means experience is split equally. In the equipment panel, the player is not wearing any equipment which results in 0 bonuses for all attributes. Starting with the bottom left of the combat triangle, a mage has advantage over the melee equipment typically worn by a melee fighter, a melee warrior has an advantage over the equipment typically worn by a ranged fighter, and ditto for ranged to mage.

1.1.4 Summary

A fighter has some combat skill levels and will (typically) equip some armour and a weapon. They will select an attack style, which selects the skill they will receive experience in, and which equipment bonuses plays a roll in the accuracy calculation. The problem then reduces to considering an accuracy and maximum hit. Once a fight begins, an attack occurs every couple of ticks. If the attack is successful, a uniform integer between 0 and their max hit is delivered to the opponent, reducing their current hit points. Once a fighter's health reaches 0, the fight is over.

1.2 Agency

1.2.1 Special Attacks

Certain weapons have the ability to use a special attack, typically dealing additional damage, but may also reduce the opponent's levels temporarily.

1.2.2 Temporary Boosts and Healing

Potions provide temporary boosts to skill levels.

1.2.3 Item Switching and Movement

Different items, moving around. Attack delays etc.

Chapter 2

Maximum Hits

2.1 Melee

$$m_0 = \left\lfloor \frac{1}{2} + \frac{64 + E_{\text{strength}}}{640} \left\lfloor \left\lfloor \bar{L}_{\text{strength}} B_{\text{prayer}} + B_{\text{stance}} \right\rfloor B_{\text{void melee}} \right\rfloor \right\rfloor \tag{2.1}$$

2.2 Ranged

The maximum ranged hit is given by:

$$m = \left\lfloor c_0 + c_1 L_r^{\text{eff}} + c_2 S_r + c_3 L_r^{\text{eff}} S_r \right\rfloor \tag{2.2}$$

$$L_s^{\text{eff}} \equiv \lfloor (L_r + B_{\text{potion}}) B_{\text{prayer}} B_{\text{other}} + \mathcal{S} \rfloor \tag{2.3}$$

$$\{c_i\} = \left\{ 1.3, \frac{1}{10}, \frac{1}{80}, \frac{1}{640} \right\}. \tag{2.4}$$

For ranged,

$$\mathcal{S} = \begin{cases} 3 & \text{if style is } \texttt{accurate} \\ 0 & \text{Otherwise} \end{cases}. \tag{2.5}$$

Note that if the attack style is set to rapid, the weapon attack speed is increased by 1 tick. A list of B_{potion}, B_{prayer}, and (incomplete) B_{other} can be found in Ref. [9].

2.3 Magic

Magic differs slightly, so we need a few additional definitions. First we define m_{spell} as the base max hit of the player's spell/staff. Some of these depend on the player's magic level. A list of these can be found in Ref. [10]. Then there are several special items, listed below as an associated bonus B_{other} and either an additive toggle $\bar{\delta}_{\text{item}}$ which is 1 or 0 based on the accompanying condition or a multiplicative toggle δ_{item} which is either $B_{\text{other}}^{\text{item}}$ or 1 based on the accompanying condition.

1. $B_{\text{other}}^{\text{chaos}} = 3, \bar{\delta}_{\text{chaos}}$ if a `bolt` spell is used along with `Chaos gauntlets`.
2. $B_{\text{other}}^{\text{tome}} = 1.5, \delta_{\text{tome}}$ if a `fire` spell is used along with a `Tome of fire`.

3. $B^{\text{castlewars}}_{\text{other}} = 1.2, \delta_{\text{castlewars}}$ if a `Castle wars bracelet` is worn while attacking a `flag bearer`.

4. $B^{\text{salve}}_{\text{other}} = \textit{varies}, \delta_{\text{salve}}$ if any variant of the `salve amulet` is worn while attacking an `undead`.

5. $B^{\text{slayer}}_{\text{other}} = 1.15, \delta_{\text{slayer}}$ if any variant of the `imbued black mask` is worn while attacking `slayer task monster`.

Then the maximum magic hit is given by:

$$m = \left\lfloor \left\lfloor \left\lfloor \left\lfloor (m_{\text{spell}} + B^{\text{chaos}}_{\text{other}} \bar{\delta}_{\text{chaos}}) * (1 + S_m) \right\rfloor \delta_{\text{salve}} \bar{\delta}_{\text{salve}} + (1 - \bar{\delta}_{\text{salve}}) \delta_{\text{slayer}} \right\rfloor \delta_{\text{tome}} \right\rfloor \delta_{\text{castlewars}} \right\rfloor \tag{2.6}$$

Chapter 3

Accuracy

Chapter 4

Models

4.1 Important Quantities

4.2 Crude

4.3 Averaged Piecewise

4.4 Piecewise

4.5 Markov Chain

Chapter 5

Optimizing Player Equipment

In Section 1.1.2 we discussed the basic formulation of equipment bonuses. In this chapter, we will be interested in optimizing the player's choice of equipment to maximize some metric. Some examples of these metrics would be combat experience per hour, number of kills per hour, probability of winning, and so. Since the goal of the player may vary heavily, we aim to produce a general framework that can maximize arbitrary objective functions.

There are a large number of items that a player can equip in each slot, typically ranging from 10-100 considerations. Exhaustively considering each possible combination of equipment would result in roughly 10^{10} to 100^{10} possible sets. It is obvious then, that brute force would not work. We aim to reduce the number of possible equipments *loadouts*, $\mathcal{L}$ that we must consider. A loadout is simply the set of equipment that a fighter is wearing:

$$\mathcal{L} = \{I_{\text{slot}} \,|\, \text{slot} \in \text{slots}\}. \tag{5.1}$$

To do this, we will reduce redundant equipment choices for each slot such that we end with a set of *possibly* optimal loadouts. The only way to determine which of those is the actual optimal solution, $\mathcal{L}^*$ is to numerically evaluate the value of the objective, f, for each possibly optimal loadout. This defines our optimization problem as:

$$\mathcal{L}^* \leftarrow \underset{\mathcal{L}\in\{\mathcal{L}\}}{\text{argmin}}\, f(\mathcal{L}). \tag{5.2}$$

So we will begin by reducing the size of the set $\{\mathcal{L}\}$.

5.1 The Projection Vector

Different optimization problems require different information about the player's equipment. For example, when straightforwardly optimizing pure damage output (typically measured in kills per hour), the defensive bonuses of the player are irrelevant. By contrast, trying to maximize the probability of killing an opponent would require consideration of those defensive bonuses. Furthermore, a fighter only attacks with one attack style at a time. So clearly, not all bonuses are required for every optimization. For this reason, we introduce a *projection* vector, $\vec{p}$, that will *select* the bonuses that matter. Recall that our bonuses are divided into two groups: constant bonuses, and special bonuses. In Section 1.1.2, we discussed how we treat special bonuses differently. So we will ignore them for now, and focus on the constant bonuses, $\vec{\mathcal{I}}_c^{\text{slot}}$. To select out the desired bonuses, we make $\vec{p}$ the same length as $\vec{\mathcal{I}}_c^{\text{slot}}$ so that each element in $\vec{p}$

corresponds to an equipment bonus. We can do so-called one-hot encoding, where we set elements to either 0 or 1 depending on whether the corresponding equipment bonus should be considered.

An element-wise multiplication (also known as a Hadamard product) of these two vectors, $\vec{\mathcal{I}}_c^{\mathrm{slot}} \odot \vec{p}$ contains the bonuses of the item that we want, and 0's for bonuses we don't want to consider. To compare different items we need to define a comparison operator, $\hat{C}(\cdot, \cdot)$ such that:

$$\hat{C}(\vec{\mathcal{I}}_{c,1}^{\mathrm{slot}}, \vec{\mathcal{I}}_{c,2}^{\mathrm{slot}}; \vec{p}) = \begin{cases} 1 \text{ if } \exists\, i \in [1, n] \,|\, (\vec{\mathcal{I}}_{c,1}^{\mathrm{slot}} \odot \vec{p})_i > (\vec{\mathcal{I}}_{c,2}^{\mathrm{slot}} \odot \vec{p})_i \\ 0 \text{ otherwise,} \end{cases} \tag{5.3}$$

where n is the number of bonuses for an item can have, and $(\cdot)_i$ represents the i'th bonus. By simply comparing the individual item bonuses, this term indicates whether some item, $\vec{\mathcal{I}}_{c,1}^{\mathrm{slot}}$, provides some advantage over another, $\vec{\mathcal{I}}_{c,2}^{\mathrm{slot}}$. If a given item doesn't provide some advantage over any other item then it doesn't need to be included in the set of equipment to consider. Note that $\hat{C}(\vec{\mathcal{I}}_{c,1}^{\mathrm{slot}}, \vec{\mathcal{I}}_{c,1}^{\mathrm{slot}}; \vec{p}) = 0$.

5.2 Set Reduction

With this, we can reduce the set of possible items in given slot, $\{\mathcal{I}_j^{\mathrm{slot}}\}_{j=1}^{N_{\mathrm{slot}}}$, where N_{slot} is the number of possible items in that slot. We can define the following matrix,

$$\boldsymbol{A}^{\vec{p}} = A_{ij}^{\vec{p}} = \hat{C}(\vec{\mathcal{I}}_{c,i}^{\mathrm{slot}}, \vec{\mathcal{I}}_{c,j}^{\mathrm{slot}}; \vec{p}), \tag{5.4}$$

that is also one-hot encoded with 1's representing that the item in column i provides some advantage over the item in row j, for our optimization problem.

These individual item comparisons aren't too important, instead we care if an item may provide some advantage over *any* other in the set. So, an item shouldn't be considered if the comparison yields 0 when compared against all other items:

$$\sum_{i=1}^{N} \hat{C}(\vec{\mathcal{I}}_{c,j}^{\mathrm{slot}}, \vec{\mathcal{I}}_{c,i}^{\mathrm{slot}}; \vec{p}) = 0 \tag{5.5}$$

$$\implies \sum_{i=1}^{N} A_{ij}^{\vec{p}} = 0 \tag{5.6}$$

$$\implies \mathbf{1} \cdot \boldsymbol{A}_j^{\vec{p}} = 0 \tag{5.7}$$

where $\mathbf{1}$ is a N_{slot}-dimensional vector containing ones, and $\boldsymbol{A}_j^{\vec{p}}$ is the j'th column in $\boldsymbol{A}^{\vec{p}}$. Then the reduced set of items for a given slot is then:

$$\{\mathcal{I}_j^{\mathrm{slot}}; \vec{p}\}_{j=1}^{\bar{N}_{\mathrm{slot}}} = \{\mathcal{I}_j^{\mathrm{slot}} \,|\, \mathbf{1} \cdot \boldsymbol{A}_j^{\vec{p}} \neq 0, \;\; 1 \leq j \leq N_{\mathrm{slot}}\}, \tag{5.8}$$

where $\bar{N}_{\mathrm{slot}}$ is the number of items left in this reduced set. The possibly optimal loadout set can be constructed as a "n-ary" Cartesian product:

$$\{\bar{\mathcal{L}}; \vec{p}\} = \{(\mathcal{I}^{\mathrm{head}}, ..., \mathcal{I}^{\mathrm{ring}}) | \mathcal{I}^{\mathrm{slot}} \in \{\mathcal{I}_j^{\mathrm{slot}}; \vec{p}\}_{j=1}^{\bar{N}_{\mathrm{slot}}} \, \forall \, \mathrm{slot} \in \{\mathrm{slots}\}\} \tag{5.9}$$

As mentioned earlier, the choice of $\vec{p}$ depends on the problem at hand. However, only one attack style is generally considered and so often $\vec{p}$ can be chosen to only consider one attack style.[1] A natural formulation would be to iterate over each attack style. We

[1] If the fighter switches weapons, or has multiple attacks, considering more than one attack style would then be required.

can, for example, use $\vec{p}_{\text{stab}}$ to denote a projection vector which has stab as the only non-zero attack bonus. Then, the total set of considerations is the union of the specific attack style sets:

$$\{\bar{\mathcal{L}}; \vec{p}\} = \bigcup_{a \in \{\text{attack styles}\}} \{\bar{\mathcal{L}}; \vec{p_a}\}. \tag{5.10}$$

This makes our optimization:

$$\mathcal{L}^* \leftarrow \operatorname*{argmin}_{\mathcal{L} \in \{\bar{\mathcal{L}}; \vec{p}\}} f(L), \tag{5.11}$$

which has reduced a search space containing billions of possibilities to one containing (empirically) less than 1,000.

Chapter 6

Optimizing Training Order

6.1 Dijkstra's algorithm

Appendices

Appendix A

Justifying the Recursive Model Approximation

We begin with Eq. (**??**) which we will restate here with $h_m \to x_n$:

$$x_{n+1} = x_n - \frac{x_n}{2}\left(2 - \frac{x_n + 1}{M + 1}\right). \tag{A.1}$$

This formulation looks similar to Newtons method for finding the root of $f(x)$, which is given as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{A.2}$$

After some algebra we find that,

$$\frac{f(x_n)}{f'(x_n)} = x_n(1 - \gamma x_n). \tag{A.3}$$

This can be easily solved for $f(x)$ since this is a separable equation:

$$\frac{df}{f} = \frac{dx}{x(1 - \gamma x)} \tag{A.4}$$

$$\int \frac{df}{f} = \int \frac{dx}{x(1 - \gamma x)} \tag{A.5}$$

$$\ln(f) = \ln|x| - \ln|\gamma x - 1| + C \tag{A.6}$$

$$f(x) = e^C |x||\gamma x - 1| \tag{A.7}$$

The error, ϵ in the next iteration of Newtons method is given as,

$$\epsilon_{n+1} = \epsilon_n^2 \left|\frac{f''(r)}{2f'(r)}\right|, \tag{A.8}$$

where r is the root we desire. In this case the root we want is,

$$\lim_{n \to \infty} x_n = 0 \tag{A.9}$$

since this corresponds to the health reaching zero. The ratio becomes,

$$\left|\frac{f''(r)}{2f'(r)}\right| = \gamma, \tag{A.10}$$

which simplifies the error equation to,

$$\epsilon_{n+1} = \gamma \epsilon_n^2. \tag{A.11}$$

This is exactly the same recursive equation we had before! Except that now we have error in health instead of health. So how does this connect? Since we already know the root value (which is 0), the error becomes the upper bound on the health. This means that if we have an error of, for example, 1 that the health must be below that. So, by solving $\epsilon = h$ we can find the number of iterations required to reach below that health. This solution is already given earlier but in the context of error we have,

$$\epsilon_n = \frac{1}{\gamma}\left(\frac{1}{2} - \gamma\right)^{2^n} \tag{A.12}$$

$$n = \log_2 \log_{\frac{1}{2}-\gamma}(\gamma\epsilon). \tag{A.13}$$

This has done two things: justify the use exclusion of the h_m term in the original recursive equation (instead of excluding h_m^2), and provided a second interpretation of the meaning of $h = 1$. The error comes from a Taylor series, but interestingly all higher order terms die off so this is actually exact. This suggests that for higher precision, the assumption that ϵ is small is violated and the Taylor series formulation no longer holds.

Appendix B

Power Reduction in the Piecewise Recursive Model

The average damage described in Section **??** can be expanded to give,

$$
\begin{aligned}
\langle D \rangle_{\text{overall}} &= \frac{1}{h_0}\left(\sum_{n=M+1}^{h_0}\frac{M}{2} + \sum_{n=1}^{y}\frac{n}{2}\left(2-\frac{n+1}{M+1}\right)\right) && \text{(B.1)}\\
&= \frac{1}{h_0}\left(\frac{M}{2}(h_0-y) + \sum_{n=1}^{y} n - \frac{1}{2}\sum_{n=1}^{y} n\frac{n+1}{M+1}\right) && \text{(B.2)}\\
&= \frac{1}{2h_0}\left(Mh_0 - My + y(y+1) - \frac{1}{2(M+1)}\sum_{n=1}^{y}(n^2+n)\right) && \text{(B.3)}\\
&= \frac{1}{2h_0}\left(Mh_0 - My + y(y+1) - \frac{y(y+1)}{2(M+1)} - \frac{y(n+1)(2y+1)}{6(M+1)}\right) && \text{(B.4)}\\
&= \frac{1}{2h_0(M+1)}\left(M^2h_0 - M^2y + My^2 + yM + Mh_0 - My - \frac{y^3-y}{3}\right) && \text{(B.5)}\\
&= \frac{y(y+1)}{h_0(M+1)}\left(\frac{M(M+1)h_0}{2y(y+1)} + \frac{(y-M)M}{2(y+1)} - \frac{y-1}{6}\right) && \text{(B.6)}\\
&= \frac{y(y+1)}{h_0(M+1)}\left(\frac{M(M+1)h_0}{2y(y+1)} + \frac{(y-M)M}{2(y+1)} + \frac{y+1}{2} - \frac{1}{3}(2y+1)\right) && \text{(B.7)}
\end{aligned}
$$

where $y = \min(M, h_0)$. In the second line, we used:

$$
\begin{aligned}
\sum_{a+1}^{b} 1 &= \begin{cases} b-a & \text{if } b > a \\ 0 & \text{else} \end{cases} && \text{(B.8)}\\
&= b - \begin{cases} a & \text{if } b > a \\ 0 & \text{else} \end{cases} && \text{(B.9)}\\
&= b - \min(a,b) && \text{(B.10)}
\end{aligned}
$$

To finish, let's focus on,

$$\frac{M(M+1)h_0}{2y(y+1)} + \frac{(y-M)M}{2(y+1)} + \frac{y+1}{2} \tag{B.11}$$

$$= \frac{1}{2y(y+1)}\Big[M(M+1)h_0 + y(y-M)M + y(y+1)^2\Big] \tag{B.12}$$

$$= \frac{1}{2y(y+1)}\Big[M^2h_0 + Mh_0 + My^2 - M^2y + y^3 + 2y^2 + y\Big]. \tag{B.13}$$

This is a hard equation to simplify since the M's and h_0's are implicitly embedded in the y's, but if you play with it long enough you can "discover" a way to simplify it - a form *power reduction* that relies on getting rid of as many y's as possible.

B.1 Power Reduction

I'd like to preface the next part by saying the final result can easily be determined by plugging in m as the min, and h_0 as the min and combining the result. In this instance it works out nicely, but we will focus on general machinery to solve these problems assuming the solution was not so nice. Our goal here is to pull the m's and h_0's out of y. To do this, let's see if there is a way to construct y^2 from the other variables, specifically only using y^1. We know that if $M < h_0$, we need a term like M^2, and in the opposite case, we need a term like h_0^2,

$$y^2 \sim M^2 \text{ or } h_0^2. \tag{B.14}$$

Based on this, we should be able to use min to switch between these two. So if we write the first term using y, we'd have something like My, which is true when M is the minimum. If it isn't the minimum, there should be a second term which cancels the now Mh_0 term plus the required h_0^2 term:

$$y^2 = My + h_0(y-M) \quad (!) \tag{B.15}$$

Using the same logic, we can inductively deduce,

$$\boxed{y^{n+1} = My^n + h_0^n(y-M) = My^n + h_0^n y - Mh_0^n.} \tag{B.16}$$

(and as an identity for the math people, with $\gamma = \min(a,b)$):

$$\boxed{\gamma^{n+1} = a\gamma^n + b^n(\gamma - a) = a\gamma^n + b^n\gamma - ab^n} \tag{B.17}$$

In fact, this holds for max as well, or any *similar* piece-wise function. Writing this as a recursive sequence by letting $g(n) = \gamma^n$ yields,

$$g(n+1) = ag(n) + b^n(g(1) - a), \tag{B.18}$$

Under the initial condition $g(1) = \gamma$, WolframAlpha gives the general solution as,

$$g(n) = a^n + (\gamma - a)\frac{a^n - b^n}{a - b} \tag{B.19}$$

$$\boxed{g(n) = a^n + (\gamma - a)\sum_{i=0}^{n-1} a^{n-i-1}b^j,} \tag{B.20}$$

where the second line uses the difference of powers formula. This could have been solved by hand, but we've had enough fun with recursion in the other sections! This yields,

$$\begin{aligned}
g(1) &= \gamma && \text{(B.21)}\\
g(2) &= a^2 + (\gamma - a)(a+b) && \text{(B.22)}\\
&= a^2 + a\gamma - a^2 + b\gamma - ab && \text{(B.23)}\\
&= a\gamma + b(\gamma - a) && \text{(B.24)}\\
g(3) &= a^3 + (\gamma - a)\frac{a^3 - b^3}{a-b} && \text{(B.25)}\\
&= a^3 + (\gamma - a)(a^2 + ab + b^2) && \text{(B.26)}\\
&= a^3 + \gamma a^2 + \gamma ab + \gamma b^2 - a^3 - a^2 b - ab^2 && \text{(B.27)}\\
&= \gamma a^2 + \gamma ab + \gamma b^2 - a^2 b - ab^2. && \text{(B.28)}
\end{aligned}$$

These agree with the original iterative equation. Okay, so this is a bit overkill since at most y^3 appears, so having general powers isn't too helpful. Nonetheless, we can now reduce the powers of y in the original equation, and see how that simplifies things.

B.2 Simplifying

We can now reduce the bracketed term in Eq. B.13:

$$\begin{aligned}
& M^2 h_0 + M h_0 + M y^2 - M^2 y + y^3 + 2y^2 + y && \text{(B.29)}\\
&= M^2 h_0 + M h_0 + M^2 y + h_0 y M - h_0 M^2 - M^2 y + y M^2 + y M h_0 + y h_0^2 + && \text{(B.30)}\\
&\qquad - M^2 h_0 - h_0^2 M + 2My + 2h_0 y - 2h_0 M + y && \text{(B.31)}\\
&= \left(-M^2 y + y M^2 + M^2 y + y M h_0 + y h_0^2 + 2My + 2h_0 y + y + h_0 y M\right) + && \text{(B.32)}\\
&\qquad \left(M^2 h_0 + M h_0 - h_0 M^2 + -M^2 h_0 - h_0^2 M - 2h_0 M\right) && \text{(B.33)}\\
&= \left(2My + 2h_0 y + y + 2yMh_0 + M^2 y + y h_0^2\right) + \left(-M^2 h_0 - h_0^2 M - h_0 M\right) && \text{(B.34)}
\end{aligned}$$

Having eliminated the "hidden" variables, let's try to re-group into powers of y:

$$\begin{aligned}
&= y^2 + \left(My + h_0 y + y + 2yMh_0 + M^2 y + y h_0^2\right) + \left(-M^2 h_0 - h_0^2 M\right) && \text{(B.35)}\\
&= 2y^2 + y + 2yMh_0 + M^2 y + y h_0^2 + -M^2 h_0 - h_0^2 M + M h_0 && \text{(B.36)}\\
&= y^2 M + y + y^2 + y^2 h_0 + y^2 + M h_0 && \text{(B.37)}\\
&= y\left(yM + y h_0 + y + M + h_0 + 1\right) && \text{(B.38)}\\
&= y\left(y+1\right)\left(M + h_0 + 1\right) && \text{(B.39)}
\end{aligned}$$

Putting this into the corresponding term in Eq. B.7 gives

$$\frac{1}{2y(y+1)} y(y+1)\left(M + h_0 + 1\right) = \frac{1}{2}\left(M + h_0 + 1\right) \tag{B.40}$$

and so finally we arrive at,

$$\boxed{\langle D \rangle_{\text{overall}} = \frac{y(y+1)}{h_0(M+1)}\left[\frac{1}{2}\left(M + h_0 + 1\right) - \frac{1}{3}(2y+1)\right]}. \tag{B.41}$$

Appendix C

Fighting Probabilities

C.1 Definitions

$$c_+ = \frac{a}{m+1} \tag{C.1}$$

$$c_* = mc_+ \tag{C.2}$$

$$c_0 = 1 - c_* \tag{C.3}$$

$$\tag{C.4}$$

C.2 Recursive Equation

The probability the player does n damage to their opponent is given by:

$$P(X = n) = \begin{cases} c_0, & \text{if } n = 0 \\ c_+, & \text{if } 1 \le n \le m \\ 0, & \text{for } n < 0 \text{ or } m < n \end{cases} \tag{C.5}$$

To be explicit, this tells us that c_0 is the probability of hitting a zero, and c_+ is the probability of doing damage. In one turn, the opponent can be brought to a given health i, from their initial health h according to the transition probability,

$$\pi_{h,i} = \begin{cases} P(X = h - i) & \text{if } i > 0 \\ P(X \ge h) & \text{if } i = 0 \end{cases} \tag{C.6}$$

as given in Nukelawe's work. The probability they are killed in L turns can be given by the sum of the probabilities the opponent was brought to i, then killed in $L-1$ turns:

$$
\begin{aligned}
P_{h,L} &= \sum_{i=0}^{\infty} \pi_{h,i} P_{i,L-1} && \text{(C.7)} \\
&= \cancel{\pi_{h,0} P_{0,L-1}} + \pi_{h,h} P_{h,L-1} + \sum_{i=1}^{h-1} \pi_{h,i} P_{i,L-1} + \sum_{i=h+1}^{\infty} \pi_{h,i} P_{i,L-1} && \text{(C.8)} \\
&= c_0 P_{h,L-1} + \sum_{i=1}^{h-1} \pi_{h,i} P_{i,L-1} + \cancel{\sum_{i=h+1}^{\infty} \pi_{h,i} P_{i,L-1}} && \text{(C.9)} \\
P_{h,L} &= c_0 P_{h,L-1} + c_+ \sum_{i=\max(h-m,1)}^{h-1} P_{i,L-1} && \text{(C.10)} \\
& && \text{(C.11)}
\end{aligned}
$$

where in the second line, the $i = 0, h$ terms are explicitly considered, and the remaining sum is split in two. In the third line, $\pi_{h,i}$ would correspond to healing (and the $n = 0$ condition in Eq. (C.5)) and is therefore zero. In the final line, we get the lower bound by considering that

$$
\begin{aligned}
1 \leq i \leq h-1 &\implies \pi_{h,i} = c_+ \text{ if } 1 \leq h-i \leq m \text{ otherwise } 0, && \text{(C.12)} \\
&\implies 1 \leq h-i \text{ and } h-i \leq m && \text{(C.13)} \\
&\quad \therefore i \leq h-1 \text{ and } h-m \leq i. && \text{(C.14)}
\end{aligned}
$$

Since the first condition is already met, we have that $i \geq h-m$, but i also cannot be below 1, hence $i \geq \max(h-m, 1)$.

C.3 Solution

The recursive equation to solve is:

$$
P_{h,L} = c_0 P_{h,L-1} + c_+ \sum_{i \in I_h} P_{i,L-1}, \quad L \geq 2, h \geq 1 \tag{C.15}
$$

$$
\tag{C.16}
$$

where I_h is the set of integers satisfying $h-1 \geq i \geq \max(h-m, 1)$. The initial conditions are given by:

$$
\begin{aligned}
P_{h,1} &= c_+ \max(m-h+1, 0) && \text{(C.17)} \\
P_{1,L} &= c_* c_0^{L-1} && \text{(C.18)} \\
P_{1,1} &= c_* && \text{(C.19)}
\end{aligned}
$$

Using a generating function:

$$g(x,y) = \sum_{h=1}^{\infty}\sum_{L=1}^{\infty} P_{h,L} y^h x^L \tag{C.20}$$
$$= \sum_{h=1}^{\infty}\left(P_{h,1}y^h x + \sum_{L=2}^{\infty} P_{h,L} y^h x^L\right) \tag{C.21}$$
$$= \sum_{h=1}^{\infty} P_{h,1}y^h x + \sum_{h=1}^{\infty}\sum_{L=2}^{\infty} P_{h,L} y^h x^L \tag{C.22}$$
$$= xyP_{1,1} + \sum_{h=2}^{\infty} P_{h,1}y^h x + \sum_{L=2}^{\infty}\left(P_{1,L}yx^L + \sum_{h=2}^{\infty} P_{h,L} y^h x^L\right) \tag{C.23}$$
$$= xyP_{1,1} + x\sum_{h=2}^{\infty} P_{h,1}y^h + \sum_{L=2}^{\infty} P_{1,L}yx^L + \sum_{L=2}^{\infty}\sum_{h=2}^{\infty} P_{h,L} y^h x \tag{C.24}$$
$$= xyP_{1,1} + x\sum_{h=2}^{\infty} P_{h,1}y^h + y\sum_{L=2}^{\infty} P_{1,L}x^L + \sum_{L=2}^{\infty}\sum_{h=2}^{\infty} P_{h,L} y^h x^L \tag{C.25}$$
$$\tag{C.26}$$

These are the boundaries of a grid (corner + top + left) plus a sum over the interior.

C.3.1 Corner

Let's focus on each term at a time, starting with the corner:

$$xyP_{1,1} = xyc_* \tag{C.27}$$

C.3.2 Top

For the top, lets first note that $\max(m-h+1, 0)$ is non-zero when $m-h+1 \geq 1 \implies m \geq h$. Then,

$$x\sum_{h=2}^{\infty} P_{h,1}y^h = c_+ x\sum_{h=2}^{\infty}\max(m-h+1,0)y^h \tag{C.28}$$
$$= c_+ x\sum_{h=2}^{m}(m-h+1)y^h \tag{C.29}$$
$$\tag{C.30}$$

C.3.3 Left

Now the left:

$$
\begin{aligned}
y\sum_{L=2}^{\infty} P_{1,L}x^L &= y\sum_{L=2}^{\infty} c_* c_0^{L-1}x^L && \text{(C.31)}\\
&= y\frac{c_*}{c_0}\sum_{L=2}^{\infty}(c_0x)^L && \text{(C.32)}\\
&= y\frac{c_*}{c_0}\left[\sum_{L=0}^{\infty}(c_0x)^L - 1 - c_0x\right] && \text{(C.33)}\\
&= y\frac{c_*}{c_0}\left[\frac{1}{1-c_0x} - 1 - c_0x\right] && \text{(C.34)}\\
&= y\frac{c_*}{c_0}\frac{1}{1-c_0x}\left[1-(1-c_0x)-(1-c_0x)c_0x\right] && \text{(C.35)}\\
&= y\frac{c_*}{c_0}\frac{1}{1-c_0x}\left[c_0^2x^2\right] && \text{(C.36)}\\
&= y\frac{c_*}{c_0}\frac{c_0^2x^2}{1-c_0x} && \text{(C.37)}\\
& && \text{(C.38)}
\end{aligned}
$$

C.3.4 Interior

Now the interior:

$$
\begin{aligned}
\sum_{L=2}^{\infty}\sum_{h=2}^{\infty} P_{h,L}y^hx^L &= \sum_{L=2}^{\infty}\sum_{h=2}^{\infty}\left(c_0P_{h,L-1} + c_+\sum_{i\in I_h} P_{i,L-1}\right)y^hx^L && \text{(C.39)}\\
&= \sum_{L=2}^{\infty}\sum_{h=2}^{\infty} c_0P_{h,L-1}y^hx^L + c_+\sum_{L=2}^{\infty}\sum_{h=2}^{\infty}\sum_{i\in I_h} P_{i,L-1}y^hx^L && \text{(C.40)}\\
\mathcal{I}(x,y) &= G(x,y) + R(x,y). && \text{(C.41)}
\end{aligned}
$$

Let us also tackle this individually, starting with the 'g' term:

$$\sum_{L=2}^{\infty}\sum_{h=2}^{\infty} c_0 P_{h,L-1} y^h x^L = c_0 \sum_{L=1}^{\infty}\sum_{h=2}^{\infty} P_{h,L} y^h x^{L+1} \tag{C.42}$$

$$= c_0 x \sum_{L=1}^{\infty}\sum_{h=2}^{\infty} P_{h,L} y^h x^L \tag{C.43}$$

$$= c_0 x \sum_{L=1}^{\infty}\left(\sum_{h=1}^{\infty} P_{h,L} y^h x^L - P_{1,L} y x^L\right) \tag{C.44}$$

$$= c_0 x \sum_{L=1}^{\infty}\sum_{h=1}^{\infty} P_{h,L} y^h x^L - c_0 x y \sum_{L=1}^{\infty} P_{1,L} x^L \tag{C.45}$$

$$= c_0 x g(x,y) - c_* c_0 x y \sum_{L=1}^{\infty} c_0^{L-1} x^L \tag{C.46}$$

$$= c_0 x g(x,y) - c_* c_0 x^2 y \sum_{L=1}^{\infty} c_0^{L-1} x^{L-1} \tag{C.47}$$

$$= c_0 x g(x,y) - c_* c_0 x^2 y \sum_{L=0}^{\infty} c_0^{L} x^{L} \tag{C.48}$$

$$= c_0 x g(x,y) - \frac{c_* c_0 x^2 y}{1 - c_0 x} \tag{C.49}$$

$$\tag{C.50}$$

For R, we will first need a term that tells us whether h is in the set I, i.e. does h satisfy $h - 1 >= \max(m - h, 1)$? You will actually see that we need the more general $h - 1 >= \max(m - h + n, 1)$. We will call this condition $\delta^n_{m,h}$ which is 1 if satisfied and 0 otherwise. Empirically, this can be expressed as:

$$\delta^n_{m,h} = \begin{cases} 0 & \text{if } h = 1 \\ 0 & \text{if } n > m \\ 1 & \text{otherwise} \end{cases} \tag{C.51}$$

Since $h >= 2$,

$$\delta^n_{m,h} = \delta^n_m = \begin{cases} 1 & \text{if } n \le m \\ 0 & \text{otherwise} \end{cases} \tag{C.52}$$

Now solving $R(x,y) = c_+ \sum_{L=2}^{\infty} \sum_{h=2}^{\infty} \sum_{i \in I} P_{i,L-1} y^h x^L$ gives:

$$R(x,y) = c_+ \sum_{L=2}^{\infty} \sum_{h=2}^{\infty} \sum_{i=\max(h-m,1)}^{h-1} P_{i,L-1} y^h x^L \tag{C.53}$$

$$= c_+ x \sum_{L=1}^{\infty} \sum_{h=2}^{\infty} \sum_{i=\max(h-m,1)}^{h-1} P_{i,L} y^h x^L \tag{C.54}$$

$$= c_+ xy \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \sum_{i=\max(h-m+1,1)}^{h} P_{i,L} y^h x^L \tag{C.55}$$

$$= c_+ xy \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \left(P_{h,L} \delta_m^1 + \sum_{i=\max(h-m+1,1)}^{h-1} P_{i,L} \right) y^h x^L \tag{C.56}$$

$$= c_+ xy \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} P_{h,L} x^L \delta_m^1 + c_+ xy \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \sum_{i=\max(h-m+1,1)}^{h-1} P_{i,L} y^h x^L \tag{C.57}$$

$$\tag{C.58}$$

Notice that the $h = 1$ term in the second set of sums yields 0.

$$= c_+ xyg(x,y)\delta_h^1 + c_+ xy \sum_{L=1}^{\infty} \sum_{h=2}^{\infty} \sum_{i=\max(h-m+1,1)}^{h-1} P_{i,L} y^h x^L \tag{C.59}$$

$$= c_+ xyg(x,y)\delta_h^1 + c_+ xy^2 \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \sum_{i=\max(h-m+2,1)}^{h} P_{i,L} y^h x^L \tag{C.60}$$

$$= c_+ xyg(x,y)\delta_h^1 + c_+ xy^2 \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \left(\sum_{i=\max(h-m+2,1)}^{h} P_{i,L} \right) y^h x^L \tag{C.61}$$

$$= c_+ xyg(x,y)\delta_h^1 + c_+ xy^2 \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \left(P_{i,L} \delta_m^2 + \sum_{i=\max(h-m+2,1)}^{h} P_{i,L} \right) y^h x^L \tag{C.62}$$

$$= c_+ xyg(x,y)\delta_h^1 + c_+ xy^2 \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} P_{i,L} \delta_m^2 + c_+ xy^2 \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \sum_{i=\max(h-m+2,1)}^{h} P_{i,L} y^h x^L \tag{C.63}$$

$$= c_+ xyg(x,y)\delta_h^1 + c_+ xy^2 g(x,y)\delta_m^2 + c_+ xy^2 \sum_{L=1}^{\infty} \sum_{h=1}^{\infty} \sum_{i=\max(h-m+2,1)}^{h} P_{i,L} y^h x^L \tag{C.64}$$

$$\tag{C.65}$$

These series continues until the ‘engine’ producing terms has no more. The number of terms in this series is given by the maximum n that is non-zero:

$$\arg\max_n \delta_m^n = m, \tag{C.66}$$

and so,

$$R(x,y) = c_+ xg(x,y) \sum_{i=1}^{m} y^i \tag{C.67}$$

Now let's combine everything:

$$g(x,y) = xyc_* + c_+x\sum_{h=2}^{m}(m-h+1)y^h + \cancel{y\frac{c_*}{c_0}\frac{c_0^2x^2}{1-c_0x}} + c_0xg(x,y) - \cancel{\frac{c_*c_0x^2y}{1-c_0x}} + c_+xg(x,y)\sum_{i=1}^{m}y^i \tag{C.68}$$

$$= c_+x\sum_{h=1}^{m}(m-h+1)y^h + c_0xg(x,y) + c_+xg(x,y)\sum_{i=1}^{m}y^i \tag{C.69}$$

Isolating for $g(x,y)$:

$$g(x,y) - c_0xg(x,y) - c_+xg(x,y)\sum_{i=1}^{m}y^i = c_+x\sum_{h=1}^{m}(m-h+1)y^h \tag{C.70}$$

$$\left(1 - c_0x - c_+x\sum_{i=1}^{m}y^i\right)g(x,y) = c_+x\sum_{h=1}^{m}(m-h+1)y^h \tag{C.71}$$

$$g(x,y) = x\frac{c_+\sum_{h=1}^{m}(m-h+1)y^h}{1-\left(c_0+c_+\sum_{i=1}^{m}y^i\right)x} \tag{C.72}$$

$$g(x,y) = \sum_{h=1}^{\infty}\sum_{L=1}^{\infty}P_{h,L}y^hx^L = x\frac{c_+\sum_{h=1}^{m}(m-h+1)y^h}{1-\left(c_0+c_+\sum_{i=1}^{m}y^i\right)x} \tag{C.73}$$

To simplify, let's define:

$$T(y) = c_+\sum_{h=1}^{m}(m-h+1)y^h \tag{C.74}$$

$$B(y) = c_0 + c_+\sum_{i=1}^{m}y^i \tag{C.75}$$

$$\tag{C.76}$$

Now $g(x,y)$ can be written as,

$$g(x,y) = T(y)\frac{x}{1-B(y)x}. \tag{C.77}$$

C.3.5 Obtaining Power Series

To find the series representation of this, we will require the following identities:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k \tag{C.78}$$

$$(x-y)^n = \sum_{k=0}^{n} \binom{n}{k} (-1)^k x^{n-k} y^k \tag{C.79}$$

$$\frac{1}{(1-z)^\beta} = \sum_{k=0}^{\infty} \binom{k+\beta-1}{k} z^k \tag{C.80}$$

$$\sum_{k=0}^{n} r^k = \frac{1-r^{n+1}}{1-r} \tag{C.81}$$

Let us start with:

$$\frac{1}{1-B(y)x} = \sum_{k=0}^{\infty} B^k(y) x^k \tag{C.82}$$

$$\tag{C.83}$$

We need to workout $B^k(y)$:

$$B^k(y) = \left(c_0 + c_+ \sum_{j=1}^{m} y^j\right)^k \tag{C.84}$$

$$= \sum_{i=0}^{k} \binom{k}{i} c_0^{k-i} c_+^i \left(\sum_{j=1}^{m} y^j\right)^i \tag{C.85}$$

$$= c_0^k \sum_{i=0}^{k} \binom{k}{i} \left(\frac{c_+}{c_0}\right)^i \left(\sum_{j=1}^{m} y^j\right)^i \tag{C.86}$$

$$\tag{C.87}$$

Now, we would like to handle the last term,

$$\left(\sum_{j=1}^{m} y^j\right)^i = \left(\sum_{j=0}^{m} y^j - 1\right)^i \tag{C.88}$$

$$= \left(\frac{1-y^{m+1}}{1-y} - 1\right)^i \tag{C.89}$$

$$= \left(\frac{1-y^{m+1}-1+y}{1-y}\right)^i \tag{C.90}$$

$$= \frac{(y-y^{m+1})^i}{(1-y)^i} \tag{C.91}$$

$$= (y-y^{m+1})^i \cdot \sum_{j=0}^{\infty} \binom{j+i-1}{j} y^j \tag{C.92}$$

$$= (y-y^{m+1})^i \cdot \sum_{j=0}^{\infty} \binom{j+i-1}{j} y^j \tag{C.93}$$

$$= \sum_{l=0}^{i} \binom{i}{l} (-1)^l y^{i+lm} \cdot \sum_{j=0}^{\infty} \binom{j+i-1}{j} y^j \tag{C.94}$$

$$= \sum_{j=0}^{\infty} \sum_{l=0}^{i} (-1)^l \binom{i}{l} \binom{j+i-1}{j} y^{i+j+lm} \tag{C.95}$$

$$\tag{C.96}$$

where in the first line, Eq. (C.80) was used. In the second line, Eq. (C.79) was used. We also note that the expansion requires $i > 1$. We can simplify this by defining,

$$a^i_{l,j} \equiv (-1)^l \binom{i}{l} \binom{j+i-1}{j} \tag{C.97}$$

leaving us with

$$\left(\sum_{j=1}^{m} y^j\right)^i = y^i \sum_{j=0}^{\infty} \sum_{l=0}^{i} a^i_{l,j} y^{j+lm} \tag{C.98}$$

This needs to be cast as a regular power series. For this, we turn to a visual proof [Omitted], which yields:

$$\sum_{j=0}^{\infty} \sum_{l=0}^{i} a^i_{l,j} y^{j+ml} = \sum_{j=0}^{\infty} \left(\sum_{l=0}^{\min(\lfloor j/m \rfloor, i)} a^i_{l,j-ml} \right) y^j. \tag{C.99}$$

If we define,

$$A_{j,i} \equiv \sum_{l=0}^{\min(\lfloor j/m \rfloor, i)} a^i_{l,j-ml}, \tag{C.100}$$

then,

$$\sum_{j=0}^{\infty} \sum_{l=0}^{i} a^i_{l,j} y^{j+ml} = \sum_{j=0}^{\infty} A_{j,i} y^j. \tag{C.101}$$

And so,

$$\left(\sum_{j=1}^{m} y^j\right)^i = y^i \sum_{j=0}^{\infty} A_{j,i} y^j. \tag{C.102}$$

Expanding fully gives,

$$\left(\sum_{j=1}^{m} y^j\right)^i = \sum_{j=0}^{\infty} \left[\sum_{l=0}^{\min(\lfloor j/m \rfloor, i)} (-1)^l \binom{i}{l} \binom{j-ml+i-1}{j-ml}\right] y^{i+j}. \tag{C.103}$$

Returning to $B^k(y)$:

$$\begin{aligned}
B^k(y) &= c_0^k \sum_{i=0}^{k} \binom{k}{i} \left(\frac{c_+}{c_0}\right)^i \left(\sum_{j=1}^{m} y^j\right)^i && \text{(C.104)}\\
&= c_0^k \left[1 + \sum_{i=1}^{k} \binom{k}{i} \left(\frac{c_+}{c_0}\right)^i \sum_{j=0}^{\infty} A_{j,i} y^{i+j}\right] && \text{(C.105)}\\
&= c_0^k \left[1 + \sum_{j=0}^{\infty} \sum_{i=1}^{k} \binom{k}{i} \left(\frac{c_+}{c_0}\right)^i A_{j,i} y^{i+j}\right] && \text{(C.106)}\\
& && \text{(C.107)}
\end{aligned}$$

To handle this, we'll define,

$$D_{i,j}^k = \binom{k}{i} \left(\frac{c_+}{c_0}\right)^i A_{j,i}, \tag{C.108}$$

making,

$$B^k(y) = c_0^k \left[1 + \sum_{j=0}^{\infty} \sum_{i=1}^{k} D_{i,j}^k y^{i+j}\right], \tag{C.109}$$

and make use of the identity [visual proof is also omitted]:

$$\sum_{j=0}^{\infty} \sum_{i=1}^{k} a_{i,j} y^{i+j} = \sum_{j=1}^{\infty} \left(\sum_{i=1}^{\min(j,k)} a_{i,j-i}\right) y^j. \tag{C.110}$$

This leaves us with,

$$B^k(y) = c_0^k \left[1 + \sum_{j=1}^{\infty} \left(\sum_{i=1}^{\min(j,k)} D_{i,j-i}^k\right) y^j\right]. \tag{C.111}$$

We further simplify:

$$F_{j,k} \equiv \sum_{i=1}^{\min(j,k)} D_{i,j-i}^k, \tag{C.112}$$

leaving us finally with:

$$B^k(y) = c_0^k \left[1 + \sum_{j=1}^{\infty} F_{j,k} y^j\right]. \tag{C.113}$$

So our original formula becomes,

$$\frac{1}{1-B(y)x} = \sum_{k=0}^{\infty} c_0^k \left[1 + \sum_{j=1}^{\infty} F_{j,k} y^j\right] x^k \tag{C.114}$$

$$\implies \frac{x}{1-B(y)x} = \sum_{k=1}^{\infty} c_0^{k-1} \left[1 + \sum_{j=1}^{\infty} F_{j,k-1} y^j\right] x^k \tag{C.115}$$

$$= \sum_{k=1}^{\infty} c_0^{k-1} x^k + \sum_{k=1}^{\infty}\sum_{j=1}^{\infty} c_0^{k-1} F_{j,k-1} y^j x^k \tag{C.116}$$

$$\tag{C.117}$$

Multiplying by $T(y)$ gives back our generating function:

$$g(x,y) = \frac{T(y)x}{1-B(y)x} = T(y)\sum_{k=1}^{\infty} c_0^{k-1} x^k + T(y)\sum_{k=1}^{\infty}\sum_{j=1}^{\infty} c_0^{k-1} F_{j,k-1} y^j x^k \tag{C.118}$$

$$= c_+ \sum_{h=1}^{m}(m-h+1)y^h \sum_{k=1}^{\infty} c_0^{k-1} x^k + T(y)\sum_{k=1}^{\infty}\sum_{j=1}^{\infty} c_0^{k-1} F_{j,k-1} y^j x^k \tag{C.119}$$

$$= \sum_{h=1}^{m}\sum_{k=1}^{\infty}\left[c_+ c_0^{k-1}(m-h+1)\right] y^h x^k + T(y)\sum_{k=1}^{\infty}\sum_{j=1}^{\infty} c_0^{k-1} F_{j,k-1} y^j x^k \tag{C.120}$$

$$\tag{C.121}$$

Focusing on the second series:

$$T(y)\sum_{k=1}^{\infty}\sum_{j=1}^{\infty} c_0^{k-1} F_{j,k-1} y^j x^k = c_+ \sum_{h=1}^{m}(m-h+1)y^h \sum_{k=1}^{\infty}\sum_{j=1}^{\infty} c_0^{k-1} F_{j,k-1} y^j x^k \tag{C.122}$$

$$= \sum_{h=1}^{m}\sum_{k=1}^{\infty}\sum_{j=1}^{\infty} c_+ c_0^{k-1}(m-h+1) F_{j,k-1} y^{j+h} x^k \tag{C.123}$$

$$= \sum_{k=1}^{\infty}\left[\sum_{j=1}^{\infty}\sum_{h=1}^{m} c_+ c_0^{k-1}(m-h+1) F_{j,k-1} y^{j+h}\right] x^k \tag{C.124}$$

$$\tag{C.125}$$

Defining

$$H_{j,h}^k \equiv c_+ c_0^{k-1}(m-h+1) F_{j,k-1}, \tag{C.126}$$

makes,

$$T(y)\sum_{k=1}^{\infty}\sum_{j=1}^{\infty}c_0^{k-1}F_{j,k-1}y^jx^k = \sum_{k=1}^{\infty}\left[\sum_{j=1}^{\infty}\sum_{h=1}^{m}H_{j,h}^{k}y^{j+h}\right]x^k \tag{C.127}$$

$$= \sum_{k=1}^{\infty}\left[\sum_{j=1}^{\infty}\sum_{h=1}^{\min(j,m)}H_{j-h,h}^{k}y^{j}\right]x^k \tag{C.128}$$

$$= \sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\left(\sum_{h=1}^{\min(j,m)}H_{j-h,h}^{k}\right)y^jx^k \tag{C.129}$$

$$\tag{C.130}$$

So,

$$g(x,y) = \sum_{L=1}^{\infty}\sum_{h=1}^{m}c_+c_0^{L-1}(m-h+1)y^hx^L + \sum_{L=1}^{\infty}\sum_{h=1}^{\infty}\left(\sum_{i=1}^{\min(h,m)}H_{h-i,i}^{L}\right)y^hx^L \tag{C.131}$$

$$= \sum_{L=1}^{\infty}\sum_{h=1}^{\infty}\theta(m-h)c_+c_0^{L-1}(m-h+1)y^hx^L + \sum_{L=1}^{\infty}\sum_{h=1}^{\infty}\left(\sum_{i=1}^{\min(h,m)}H_{h-i,i}^{L}\right)y^hx^L \tag{C.132}$$

$$= \sum_{L=1}^{\infty}\sum_{h=1}^{\infty}\left[\theta(m-h)c_+c_0^{L-1}(m-h+1) + \sum_{i=1}^{\min(h,m)}H_{h-i,i}^{L}\right]y^hx^L \tag{C.133}$$

$$\tag{C.134}$$

where $\theta(x)$ is the Heaviside function. And at last, we obtain:

$$P_{h,L} = \theta(m-h)c_+c_0^{L-1}(m-h+1) + \sum_{i=1}^{\min(h,m)}H_{h-i,i}^{L} \tag{C.135}$$

$$\boxed{\therefore P_{h,L} = c_+c_0^{L-1}\max(m-h+1,0) + \sum_{i=1}^{\min(h,m)}H_{h-i,i}^{L}} \tag{C.136}$$

We can unravel H using:

$$H_{j,h}^{k} \equiv c_+c_0^{k-1}(m-h+1)F_{j,k-1}, \tag{C.137}$$

$$F_{j,k} \equiv \sum_{i=1}^{\min(j,k)}D_{i,j-i}^{k}, \tag{C.138}$$

$$D_{i,j}^{k} = \binom{k}{i}\left(\frac{c_+}{c_0}\right)^iA_{j,i}, \tag{C.139}$$

$$A_{j,i} \equiv \sum_{l=0}^{\min(\lfloor j/m\rfloor,i)}a_{l,j-ml}^{i}, \tag{C.140}$$

$$a_{l,j}^{i} \equiv (-1)^l\binom{i}{l}\binom{j+i-1}{j} \tag{C.141}$$

$$\tag{C.142}$$

to get:

$$
\begin{aligned}
H^L_{h-i,i} &= c_+ c_0^{L-1}(m-i+1)F_{h-i,L-1} && \text{(C.143)}\\
&= C_i c_0^{L-1} \sum_{p=1}^{\min^{L-1}_{h-i}} D^{L-1}_{p,h-i-p} && \text{(C.144)}\\
&= C_i c_0^{L-1} \sum_{p=1}^{\min^{L-1}_{h-i}} \binom{L-1}{p} \left(\frac{c_+}{c_0}\right)^p A_{h-i-p,p} && \text{(C.145)}\\
&= C_i c_0^{L-1} \sum_{p=1}^{\min^{L-1}_{h-i}} \binom{L-1}{p} \left(\frac{c_+}{c_0}\right)^p \sum_{l=0}^{\min^p_{\lfloor (h-i-p)/m \rfloor}} a^p_{l,h-i-p-ml} && \text{(C.146)}\\
&= C_i c_0^{L-1} \sum_{p=1}^{\min^{L-1}_{h-i}} \binom{L-1}{p} \left(\frac{c_+}{c_0}\right)^p \sum_{l=0}^{\min^p_{\lfloor (h-i-p)/m \rfloor}} (-1)^l \binom{p}{l} \binom{h-i-ml-1}{h-i-ml-p} && \text{(C.147)}\\
&= C_i c_0^{L-1} \sum_{p=1}^{\min^{L-1}_{h-i}} \binom{L-1}{p} G(h,i,p) && \text{(C.148)}\\
& && \text{(C.149)}
\end{aligned}
$$

where $C_i = c_+(m-i+1)$ and

$$
G(h,i,p) \equiv \left(\frac{c_+}{c_0}\right)^p \sum_{l=0}^{\min^p_{\lfloor (h-i-p)/m \rfloor}} (-1)^l \binom{p}{l} \binom{h-i-ml-1}{h-i-ml-p}. \tag{C.150}
$$

C.4 Summing over L

Often, the following quantity needs to be evaluated for further applications, like the probability of winning a fight:

$$
\sum_{L=b}^{\infty} P_{h,L} = P_{h,1} \sum_{L=b}^{\infty} c_0^{L-1} + \sum_{i=1}^{\min(h,m)} \sum_{L=b}^{\infty} H^L_{h,i}, \tag{C.151}
$$

Tackling the first term, we get:

$$
\begin{aligned}
\sum_{L=b}^{\infty} c_0^{L-1} &= \sum_{L=b-1}^{\infty} c_0^L && \text{(C.152)}\\
&= \frac{c_0^{b-1}}{1-c_0} && \text{(C.153)}\\
& && \text{(C.154)}
\end{aligned}
$$

For the second term, we need:

$$
\begin{aligned}
\sum_{i=a}^{\infty} \binom{i}{k} x^i &= \sum_{i=1}^{\infty} \binom{i}{k} x^i - \sum_{i=1}^{a-1} \binom{i}{k} x^i, \quad 1 \le k \le a && \text{(C.155)}\\
&= \frac{x^k}{(1-x)^{k+1}} - \sum_{i=1}^{a-1} \binom{i}{k} x^i && \text{(C.156)}\\
& && \text{(C.157)}
\end{aligned}
$$

Focusing on the second term,

$$\sum_{L=b}^{\infty} H_{h,i}^{L} = C_i \sum_{L=b}^{\infty} c_0^{L-1} \sum_{p=1}^{\min_{h-i}^{L-1}} \binom{L-1}{p} G(h,i,p) \tag{C.158}$$

$$= C_i \sum_{L=b}^{\infty} \left[\sum_{p=1}^{\min_{h-i}^{L-1}} c_0^{L-1} \binom{L-1}{p} G(h,i,p) \right] \tag{C.159}$$

$$= C_i \sum_{L=b}^{h-i} \sum_{p=1}^{L-1} c_0^{L-1} \binom{L-1}{p} G(h,i,p) + C_i \sum_{L=\max_{h-i+1}^{b}}^{\infty} \sum_{p=1}^{h-i} c_0^{L-1} \binom{L-1}{p} G(h,i,p) \tag{C.160}$$

$$= C_i \sum_{L=b}^{h-i} \sum_{p=1}^{L-1} c_0^{L-1} \binom{L-1}{p} G(h,i,p) + C_i \sum_{p=1}^{h-i} \left[\sum_{L=\max_{h-i+1}^{b}}^{\infty} c_0^{L-1} \binom{L-1}{p} \right] G(h,i,p) \tag{C.161}$$

$$= C_i \sum_{L=b}^{h-i} \sum_{p=1}^{L-1} c_0^{L-1} \binom{L-1}{p} G(h,i,p) + C_i \sum_{p=1}^{h-i} \left[\sum_{L=\max_{h-i}^{b-1}}^{\infty} c_0^{L} \binom{L}{p} \right] G(h,i,p) \tag{C.162}$$

$$= C_i \sum_{L=b}^{h-i} \sum_{p=1}^{L-1} c_0^{L-1} \binom{L-1}{p} G(h,i,p) + C_i \sum_{p=1}^{h-i} \left[\sum_{L=0}^{\infty} c_0^{L} \binom{L}{p} - \sum_{L=0}^{\max_{h-i}^{b-1}-1} c_0^{L} \binom{L}{p} \right] G(h,i,p) \tag{C.163}$$

$$= C_i \sum_{L=b}^{h-i} \sum_{p=1}^{L-1} c_0^{L-1} \binom{L-1}{p} G(h,i,p) + C_i \sum_{p=1}^{h-i} \left[\frac{c_0^{p}}{(1-c_0)^{p+1}} - \sum_{L=0}^{\max_{h-i}^{b-1}-1} c_0^{L} \binom{L}{p} \right] G(h,i,p) \tag{C.164}$$

$$= C_i \sum_{L=b}^{h-i} \sum_{p=1}^{L-1} c_0^{L-1} \binom{L-1}{p} G(h,i,p) + C_i \sum_{p=1}^{h-i} \left[\frac{c_0^{p}}{(1-c_0)^{p+1}} - \sum_{L=1}^{\max_{h-i}^{b-1}-1} \binom{L}{p} c_0^{L} \right] G(h,i,p) \tag{C.165}$$

Having removed any infinities, this equation is now computationally feasible. It can be used in further calculations like the probability a player kills their opponent before they are killed.

C.5 Summary

First, given a player's max hit of m and an opponent's initial health of h, we define:

$$c_+ = \frac{a}{m+1} \tag{C.166}$$
$$c_* = mc_+ \tag{C.167}$$
$$c_0 = 1 - c_* \tag{C.168}$$
$$C_i = c_+(m - i + 1) \tag{C.169}$$

where c_0 is the probability of doing zero damage, and c_+ is the probability of doing any positive amount of damage. Then, the probability of killing in L turns is given by the recursive equation:

$$P_{h,L} = c_0 P_{h,L-1} + c_+ \sum_{i=\max h-m,1}^{h-1} P_{i,L-1}, \quad L \geq 2, h \geq 1 \tag{C.170}$$

The boundary conditions are given by:

$$P_{h,1} = c_+ \max(m - h + 1, 0) \tag{C.171}$$
$$P_{1,L} = c_* c_0^{L-1} \tag{C.172}$$
$$P_{1,1} = c_* \tag{C.173}$$

This has the following solution:

$$P_{h,L} = P_{h,1} c_0^{L-1} + \sum_{i=1}^{\min(h,m)} H_{h,i}^L, \tag{C.174}$$

where

$$H_{h,i}^L \equiv c_+ c_0^{L-1}(m - i + 1) \sum_{p=1}^{\min_{h-i}^{L-1}} \binom{L-1}{p} G(h, i, p) \tag{C.175}$$

$$G(h, i, p) \equiv \left(\frac{c_+}{c_0}\right)^p \sum_{l=0}^{\min_{\lfloor (h-i-p)/m \rfloor}^{p}} (-1)^l \binom{p}{l} \binom{h - i - ml - 1}{h - i - ml - p}. \tag{C.176}$$

The probability of winning a fight with an opponent, and drawing is given by:

$$P_{\text{win}} = \sum_{L=1}^{\infty} \sum_{l=L+1}^{\infty} P_{h_{\text{player}},l}^{m_{\text{opponent}}} P_{h_{\text{opponent}},L}^{m_{\text{player}}} \tag{C.177}$$

$$P_{\text{draw}} = \sum_{L=1}^{\infty} P_{h_{\text{player}},L}^{m_{\text{opponent}}} P_{h_{\text{opponent}},L}^{m_{\text{player}}}, \tag{C.178}$$

where $P_{\text{win}} + P_{\text{lose}} + P_{\text{draw}} = 1$ and swapping opponent values for player values turns P_{win} into P_{lose}. The sum $\sum_{L=a}^{\infty} P_{h,L}$ can also be expressed in terms of a finite sum.

C.6 Comments

It would be useful to compute the time complexity required for evaluation. There is an infinite sum in both P_{win} and P_{lose} that must simply be truncated during evaluation. It would be nice to either find analytic solutions or determine appropriate cutoffs.

Part II

Woodcutting

Part III

Firemaking

Chapter 1

Wintertodt

Wintertodt is a skilling boss that is primarily fought (in a manner similar to a minigame moreso than combat) for `firemaking` experience, either solo or as part of a (large) group. Experience is gained from actions during the fight, with a bonus awarded for players that have contributed enough. The bonus provides experience and `supply crates` which contain valuable resources.

The general game play is as follows: A player will chop `logs`, and either collect them "as is" or fletch them into `kindling` (providing more points and experience per action, but costing time). They will then burn them in a `brazier`. The cold occasionally snuffs out the fire and the player will have to relight it, granting points and `firemaking` experience. Each `braizer` has an associated `Pyromancer` who needs to be revived before relighting, if they died.Sometimes the `braizer` will actually break (due to the high temperatures) and the player will repair it, providing `construction` experience and some points. During the battle the player will take damage scaling with their health and firemaking level.

We will be attempting to determine the number of games required to get to 99 firemaking. To handle the scope of this training method, we will be using the following considerations:

1. Points will only be obtained through firemaking (so fixing/relighting `braziers` and healing `Pyromancers` will be ignored). In practice, this is reasonable since on large worlds other players typically perform this anyway.

Definitions:

1. The player's `woodcutting` and `firemaking` levels be represented by L_{wc} and L_{fm} respectively.

2. Each action rewards xp proportional to the skill level. The proportionalities for cutting, fletching, burning a log, and burning a kindling are given by $c = 0.3, f = 0.6, l = 3, k = 3.8$.

3. The `firemaking` experience for logs and kindling are $E_l = lL_{fm}$ and $E_k = kL_{fm}$.

4. Assume the player has a target of T points during a game. $P_l = 10$ points are given for burning a log, $P_k = 25$ are given for burning kindling.

5. The bonus experience at the end of a round is given by BL_{fm}, $B = 100$. It's important to note that the level may change during the game.

Let's first assume that the player fletches no logs. We note that a player may level up during the course of a fight, and so for an exact solution, we need to consider the experience gained on a per action basis. We will label the experience at k kills, and a actions as E_a^k. Starting with E_0^k experience, the player's experience after a actions during a kill is governed by:

$$E_a^k(E_0^k) \mid E_{n+1}^k = E_n^k + l\mathcal{L}(E_n^k). \tag{1.1}$$

There is no current analytic formula for $\mathcal{L}$, so we will leave this as a recursive equation for now. To obtain the target number of points, the player needs to burn T/P_l logs. The total experience gained after a fight (including the bonus experience) is then,

$$E_{\lceil T/P_l \rceil}^k(E_0) + B\mathcal{L}(E_{\lceil T/P_l \rceil}^k(E_0)). \tag{1.2}$$

Finally, the experience gained after a kill is:

$$E_0^{k+1} = E_{\lceil T/P_l \rceil}^k(E_0) + B\mathcal{L}(E_{\lceil T/P_l \rceil}^k(E_0)). \tag{1.3}$$

1.1 Applications

1. Time to max from 50. (Number of kills) times (time per kill estimate).
2. Number of crates from 50.
3. Value of creates given base stats: 1, 40, 75, 99.

Part IV

Mining

Part V

Quests

Bibliography

[1] 10 most played mmorpgs of 2020. https://bestreamer.com/gaming/most-played-mmorpg-2019/2/. Retrieved October 02, 2020.

[2] Top 6 most popular mmorpgs sorted by population (2020). https://altarofgaming.com/all-mmos-sorted-by-population-2018/. Retrieved October 02, 2020.

[3] Ranking the 15 best mmorpgs of all time. https://www.thegamer.com/best-mmorpgs-ever-wow-runescape/. Retrieved October 02, 2020.

[4] Misplacedme. Relative player counts in runescape. https://www.misplaceditems.com/rs_tools/graph/.

[5] Skills. https://oldschool.runescape.wiki/w/Skills.

[6] Experience. https://oldschool.runescape.wiki/w/Experience.

[7] Quests. https://oldschool.runescape.wiki/w/Quests.

[8] Combat triangle. https://oldschool.runescape.wiki/w/Combat_triangle.

[9] Maximum ranged hit. https://oldschool.runescape.wiki/w/Maximum_ranged_hit.

[10] Maximum magic hit. https://oldschool.runescape.wiki/w/Maximum_magic_hit.