Using Evolutionary Computing To Find The Ideal Shaman Healing Spell Rotation In World of Warcraft

Alan Dimitriev
alan.dimitriev@queensu.ca
Queen's University
Kingston, Ontario, Canada

ABSTRACT

This study proposes the creation of an evolutionary algorithm to find what the best single target healing spell rotation for shamans in World of Warcraft is based on percent healing generated over a given span of time. The algorithm implements a steady state model, with tournament selection to find parents, applying one or two point crossover and point mutations to the offspring. While the globally optimal spell rotation was not uncovered, the proposed algorithm seems capable of solving the task given more compute power and time.

1 PROBLEM DESCRIPTION

On November 23rd, 2004, the massively multiplayer online role playing game (MMORPG) 'World of Warcraft' (WoW) was released. The game became an instant success with millions of players buying subscriptions and creating characters in the game. The game has received numerous large expansions over time that have allowed it to maintain one of the largest player bases in the gaming industry, at one point having over 12 million active subscribers. These expansions have brought new content, and new ideas to the game but the core concept of the gameplay has remained unchanged in the last seventeen years. Players in the World of Warcraft create their own unique character that they use to fight monsters, complete quests, and level up, in an open world fantasy setting. A player character is completely customizable, from their appearance and race, to the class they play and the abilities they can use. Currently, there are twelve different classes within the game. Each of the twelve classes has unique play styles, abilities, and fill a specific 'role' within the game. World of Warcraft uses a traditional role playing game role system that revolves around three specific archetypes: tanks, healers, and damage dealers. The damage dealing role is rather self explanatory, their goal is to do damage. Damage dealers aim to use their abilities to deal as much damage as quickly as possible to monsters and enemies. Tanks are characters who have more health and more defensive abilities, allowing them to take all the incoming damage from enemies as to prevent the damage from hitting the damage dealers or healers who have significantly less health. Healers are characters whose abilities are based around healing their allies and ensuring that they do not die during combat. What role a player fills is dependent upon their class. Certain classes can only perform one role, while others can perform two, or even all of the roles. The role essentially determines what the main focus of the character's abilities are going to be (healers have abilities that heal, tanks have abilities that increase their defenses and draw enemy aggression, and damage dealers have spells that deal damage). The actual specific abilities available to a character (one a role has been decided) are dependent upon which of the twelve classes they are playing. Each class has its own unique aesthetic, play style, and abilities.

With such a varied amount of classes, roles, and abilities, a question that arises in any player's mind when playing World of Warcraft is what the ideal way to play the game is. In practise this is a complex and loaded question because of how many different concepts need to be accounted for. However, we can start to get an idea of 'the best way to play' by breaking the problem down into simpler individual segments that can then be addressed by algorithmic means. Thus, we can take the question of 'what is the ideal way to play World of Warcraft' and break it into a smaller sub-problem: 'what is the ideal order in which a player should use their abilities?' Even this sub-problem is dependent on multiple factors, such as what class the player is playing, what role the player is playing, and a host of other game mechanics. For the purpose of this study we will be working under the assumption that the character we want to optimize is a healer, specifically a healer whose class is 'Shaman'. In game shamans are powerful practitioners of elemental forces, using the powers of earth, air, wind and fire to destroy their enemies and heal their allies.

Our problem statement then becomes: 'What is the best single target healing ability rotation for shamans in World of Warcraft?' To answer this question we must first define specifically what we mean. 'Single target' means that we are only concerned with healing one target character at a time, we are not interested in healing any characters other than our singular target. A rotation in WoW is a specific order in which a player uses his abilities. One key thing to note is that while the term 'rotation' implies a cyclic nature to the sequence of abilities cast, it is not necessary that the sequence is expected to repeat immediately after it ends. In fact 'sequence' might be a more apt designation as to what we are trying to find, but 'rotation' is the common terminology within the game's community. So, we are thus trying to find the best rotation (or order/sequence) of healing abilities for shamans. To answer this we must first define what an ability is, and what our criteria is to decide which rotation is 'best'.

An 'ability' in WoW is an action that a character can take to produce some effect.

An example of how spell details are organized can be seen in Figure 1. In this case we see the ability description box for the spell (the terms 'ability' and 'spell' may be used interchangeably throughout the course of this paper) called 'Wellspring'. There are multiple important inferences to be gained from analyzing this description box, with the important pieces of information being:

Conference'17, July 2017, Washington, DC, USA
2021. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

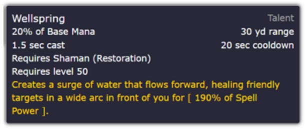

Figure 1: An ability effect box for the ability 'Wellspring', detailing the specific attributes and effects of the spell.

cast time, spell cooldown, and the spell effect. Cast time is how long it takes to cast a spell. While casting a spell a player character cannot perform any other actions, this means they cannot move, interact, or use another ability, until they have finished casting the initial spell. Wellspring in Figure 1 for example has a cast time of 1.5 seconds, meaning that for 1.5 seconds the player cannot use another ability. The next important trait of the spell to look at is the effect text in yellow. This describes what the spell actually does. One thing to note is that if a spell has a cast time then the spell effect does not take place until the cast has been completed. So 1.5 seconds after casting Wellspring allies in front of the player are healed for 190% spell power. For the purposes of this study 'spell power' is not relevant since it is based of multiple factors and is too variable to model. Instead we will focus our attention to the raw percentage value associated with that spell power. This raw percentage value is what we are interested in. When we define what is the 'best' spell rotation in this study, the 'best' rotation is the one that has the highest percentage healing value. The final important takeaway from the ability description box is the cooldown time. Wellspring has a cooldown of 20 seconds, meaning that for twenty seconds after you cast Wellspring you cannot cast it again. The player character can cast other abilities during this time, but until the cooldown for a specific spell has expired a player cannot cast that specific spell again. To summarize, we are interested in finding the sequence of ability casts in World of Warcraft for shamans that results in the maximal percent healing output.

The intricacy of answering this question stems from the variety of spells shamans have at their disposal. Table 1 displays the important details for nine different shaman healing spells. This study aims to map these nine spells to an ideal rotation. While other healing spells or effects do exist in the game, for our purposes it is unrealistic to model them in a controlled setting due to the number of factors involved. Finding the ideal order in which to use these abilities becomes a complicated task when details such as cast time and cooldowns afffect what can be done at any particular moment. This challenge is then exacerbated by the fact that not only do specific spells have their own cooldowns that dictate when they can and cannot be used, but there exists another cooldown to be aware of called the 'global cooldown'. The global cooldown triggers after any type of ability is used and prevents any spell from being used for 1.5 seconds following the completion of another. This mechanic is designed to prevent players from using multiple spells at once or in rapid sequence.

Due to the combination of global cooldowns, spell specific cooldowns, and cast times, the task of finding the ideal spell rotation becomes more and more convoluted, as well as becoming a time dependent problem. The ideal rotation to produce the most healing based on the game rules becomes dependent on how long a time span we wish to cover. If we are looking for the optimal spell sequence for a short amount of time it is likely that we would want to cast spells that provide immediate healing or spells that have the highest immediate percent healing return. While if we are interested in a longer span of time, it could be more beneficial to cast spells that provide more health over time initially so that we can accrue their benefits as time progresses. Since time has such a large impact on the distribution of the output, for the purposes of this study we will only be evaluating a five minute encounter. Most combat encounters in World of Warcraft vary in length from between two to eight minutes, with five minutes being a standard time for a regular boss in a given dungeon. This study will also be working under a couple other assumptions. The first being that we are only interested in single target healing, and that the target we are healing is taking constant damage. We will also assume that the target is always in range and our player character does not get interrupted during any casts. While this significantly hampers how realistic the application of our model is to the game in a realistic setting, it is unreasonable to model factors such as player movement, incoming enemy damage, stun or interrupt effects against our player character, and so on as this would essentially require modelling the entire gameplay experience.

2 EVOLUTIONARY ALGORITHM DESIGN

2.1 Representation and Fitness Evaluation

The first challenge when developing any evolutionary algorithm is to find the proper genetic representation of the solution space you are trying to model. For this study a simple list containing strings representing the specific spell used is suitable. What's important in this representation is that we also need the spell rotation to intrinsically be able to represent time. A workable solution is to have each index in the sequence list represent increments of 0.5 seconds. An example of this representation can be seen below, where the top list represents how the spells are stored and identified, and the bottom represents the time in seconds associated with each respective spell:

$$[HealingSurge, ChainHeal, HealingRain, Downpour]$$

$$[0.0, 0.5, 1.0, 1.5]$$

The initial 'Healing Surge' for example would be cast at the time 0.0 seconds. This time designation becomes important when evaluating the fitness of an individual.

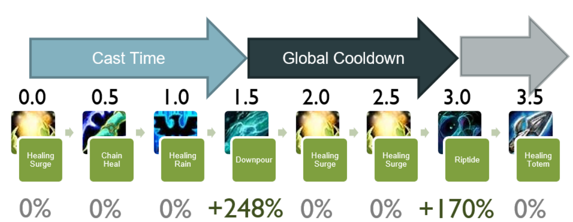

Figure 2: A visual example of the fitness evaluation of a given healing spell rotation.

Table 1: Shaman healing spell details.

Spell Name	Cast Time (sec)	Cooldown Time (sec)	Spell Effect (percent healing)
Healing Surge	1.5	0	+248%
Riptide	0	6	+170% (plus 132% over 18 seconds)
Unleash Life	0	15	+190% (plus increase next direct heal by 35%)
Chain Heal	2.5	0	+210%
Healing Wave	2.5	0	+300%
Healing Stream Totem	0	30	+47% every 2 seconds for 15 seconds
Healing Rain	2	10	+159% over 10 seconds
Downpour	1.5	15	+175%
Wellspring	1.5	20	+190%

Fitness for a given individual is calculated by sequentially going through the spells stored in the individual, evaluating their effects based on the rules of the game. Figure 2 shows an example of how this is evaluated. In this case 'Healing Surge' is cast at time 0.0. This spell notably has a 1.5 second cast time, meaning that for 1.5 seconds no other abilities can be activated and the actual spell effect of 'Healing Surge' also doesn't resolve until the cast is complete. Once the cast of 'Healing Surge' is completed at the 1.5 second mark the global cooldown then kicks in, preventing any abilities from being used for the next 1.5 seconds. It is only once this global coolddown is resolved at the 3.0 second mark that another ability (in this case 'Riptide') can be cast. In the example shown even though the individual rotation being evaluated consists of eight different spells, only two are actually used during the time frame specified by the individual.

The only mechanic that Figure 2 does not demonstrate is the spell specific cooldowns. In practice these are handled in the same way as the global cooldown or other cast times: if at a given point in time on the rotation it would be invalid to cast a spell due to it still being on cooldown from a previous use, then it will not be evaluated. Instead the evaluation will proceed to the next spell/time in the rotation. The result of this is a spell rotation that contains a spell cast every 0.5 seconds for a designated amount of time. The 0.5 index is chosen due to World of Warcraft's consistent implementation of timings for spells and cooldowns, since all cooldowns or cast times at a base level can be broken down into 0.5 increments. While this means that each individual in the population is composed mainly of spells that are not in fact ever cast, they are not prunned to remove what can be equated to a 'structural intron' to ensure that genetic diversity is preserved in the population. A spell at a specific time in one individual rotation may not be cast, however in a different rotation it might be a valid choice. This is why an evolutionary approach to this problem makes sense, there is too much variation and possibility in the order and timings that spells can be cast to easily compute the 'ideal' rotation through more traditional algorithmic means.

2.2 Parent and Survivor Selection

For this study a steady state model was selected to be the basis of our survivor selection. The reasoning being that since it is likely that there are beneficial patterns within a population (such as casting spells that provide effects over time at the beginning of a rotation) that might be harder to uncover if the entire population is replaced with every iteration. With a steady state model we would in theory be replacing the 'less fit' rotations in our population with the offspring of rotations that are 'more fit'. This allows for our rotations that produce the most percent healing to remain in our population as we experiment and progress towards finding more optimal combinations.

To achieve this 'steady-state' a parent selection mechanism is needed to not only find parents that have good fitness, but also one that finds less desirable rotations within the population that should be replaced. Given that we have a well defined external fitness function to evaluate our population members, 'tournament selection' serves as a method of parent selection that suits this problem well. Preliminary experiments with the number of tournaments and the tournament sizes suggested there was no increase in model efficiency or performance increases resultant from increasing either parameter, thus the final parameters for tournament selection were decided to be two independent tournaments to decide the two parent individuals, and a tournament size $k = 3$. Each generation six participants are chosen randomly from the total population, split into two groups of three to represent each tournament. Within each tournament, the fitness of each individual is evaluated, with a winner and loser for each tournament being decided based on which individual has the best (highest percent healing) or worst (lowest percent healing) fitness. The winners of the two tournaments serve as the parents in our model. Copies of them are made and these 'offspring' undergo recombination and mutation according to the parameters specified in the following sections. These offspring then replace the two losers from the tournaments in the original population. The benefit of this is that the 'best' individuals in a tournament are never replaced, allowing for rotations that produce a higher percent healing output to remain in our population.

2.3 Recombination

Due to how our healing spell rotations are represented (a simple list representation) our recombination mechanism does not have to be anything too complex. The only rule that our mechanism must abide by is that the overall length of the offspring must be the same as the parents. This is because since our rotations represent a specific length of time, if we performed a recombination that produced offspring of unequal length then those offspring would no longer be modeling the specific length of time we initially set

Table 2: Technical summary of implemented evolutionary algorithm.

Parameter	Description
Representation	List of strings of length 600 (each index is a 0.5 time increment)
Mutation	Point Mutation (pm = .5)
Recombination	One and two point crossover (crossover rate = 0.8)
Parent Selection	Tournament selection (2 tournaments, k = 3)
Survivor Selection	Steady state

out to model. With this in mind, standard crossover makes a case for being the most suitable recombination mechanism. As long as the crossover point is the same across both parents during recombination, then the output lengths of the offspring will also remain the same. This allows us to share genetic information between two individuals while staying within the bounds defined to ensure that our offspring are valid in length. For this study we implemented both one-point and two-point crossover as this seemed to produce the best results during preliminary trials. Each iteration any given pair of parents has an 80% chance to undergo recombination, if they do not undergo recombination then copies of the parents are returned as the offspring pre-mutation. If recombination for a given pair of parents is chosen, there is a 50/50 chance that they will then undergo either one-point or two-point mutation.

2.4 Mutation

Mutation for this problem is a straightforward concept. Each offspring has a mutation rate which determines the probability that each one undergoes mutation. Since our population individuals cannot contain varying lengths there is no macro-mutations that are applicable to our algorithm, instead each offspring has a chance to undergo micro-mutation in the form of a point mutation. In this point mutation a random spell within the rotation is selected to randomly be replaced by a different spell that is also selected at random. One important distinction is that a spell cannot be replaced by another copy of itself. Since tournament selection keeps higher fitness individuals around in the population and only replaces two individuals on each iteration we can get away with having a relatively high mutation rate of 0.5 to allow for more genetic diversity to be introduced into the population.

2.5 Evolutionary Algorithm Summary

Table 2 displays a technical summary of the parameters of the evolutionary algorithm implemented in this study. Since it is very possible that there is only one definitive 'optimal' or 'near-optimal' rotation that can be found as a result of implementing this algorithm, this study proposes a more broad result reporting metric. We aim to report the rotation that results in the single highest percent healing generation over the span of five minutes, but we also aim to report on the probability that a spell occurs at a specific time interval in the rotation. To do this we will run five trials of 50,000 iterations. Taking from these trials the best individuals and recording where spells occur in each rotation. While five trials seems rather low from the offset, each trial does take a relatively long amount of time to converge.

3 EVOLUTIONARY ALGORITHM RESULTS

The fitness values for the five documented trials can be seen in Table 3. While our evolutionary algorithm does not seem to converge at a global maximum, each trial produces a percentage healing output that is relatively close to the recorded maximum. It is impossible to fit even the 'intron free' versions of the rotations on this paper (and not really worth it since they intrinsically are hard to decipher) so the actual output of the program can be viewed in the submitted project directory. Figure 3 displays the counts of what spells were used on what cast amongst the first ten spell casts of the five trials. While there aren't many clear patterns that can be identified by looking at this figure alone (since it only accounts for on average the first 24 seconds of a 300 second rotation) but by looking at this and the raw output in the form of the intron free rotations there are some observations that can be noted. The spell 'Chain Heal' is never cast, in any of the rotations generated by the five trials. The spell 'Unleash Life' is almost always followed by either a 'Wellspring' or 'Riptide' cast. Healing totems on average are summoned after the first four casts.

Table 3: Results of evolutionary algorithm.

Trial Number	Fitness (Percent Healing)
1	43982.86666832964%
2	43566.93333511964
3	44086.600001629544
4	42996.9333348596
5	43173.33333504968

4 COMPARISON

In order to compare our evolutionary algorithm's performance against another model a simple greedy algorithm was created to act as a competitor of sorts to our proposed EA. For clarity purposes we will refer to our evolutionary algorithm as algorithm EA and the greedy algorithm as GA. The GA is not overly complex. The GA takes a 'greedy' approach to rotation generation, at each time interval in casts the spell that will result in the maximal percentage healing. Spells that heal over time have had any of their healing that occurs at delayed times counted towards their 'max' value. This is to incentivize casting these spells, as otherwise this greedy approach would only cast 'Healing Wave' as it has the highest base percentage healing output.

Due to its greedy nature and static problem, the GA always produces the same resultant rotation. The GA rotation simply casts

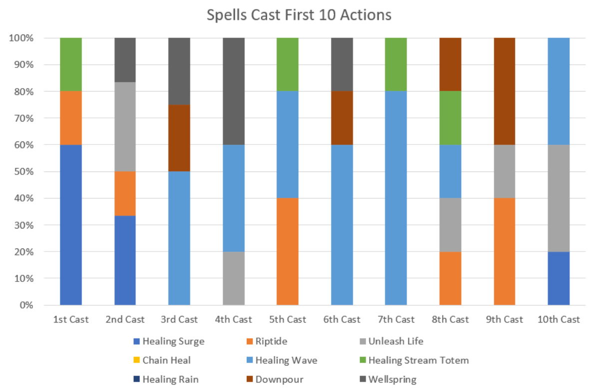

Figure 3: A chart displaying the frequencies of spell usage in the first 10 casts of a rotation.

'Healing Stream Totem' at the start of its sequence, and then proceeds to alternate between 'Riptide' and 'Healing Wave'. The GA's final fitness score is **26018.00000199013%**.

While, admittedly, the greedy algorithm GA used for comparison is significantly un-optimized, it demonstrates the problem with using more traditional approaches for this type of problem. There is simply too much variability within the solution space to allow a greedy algorithm to find the 'ideal' rotation. For this problem type, our evolutionary approach shows significantly more promise. Not only because it achieved a raw fitness value that was higher than the greedy approach, but also because of how its rotations are constructed. Traditional algorithms might prioritize spells that have appealing effects with a disregard for how their cooldowns, cast times, or impact on other spells affects the final fitness.

5 DISCUSSION

While it seems that there was not enough time to optimize our evolutionary algorithm perfectly (it fails to converge on a singular global optimum), it does significantly outperform the greedy approach and does seem to infer ideal spell combinations and uses by itself. The greedy approach ends up only utilizing three different spells and is only able to create around half the healing output of the evolutionary approach. This serves to highlight the idea that for certain problems, traditional algorithms are outmatched by evolutionary approaches. The ideal shaman rotation problem for example is one that has an incredibly vast solution space and is difficult to map by traditional means. However, this paper does show that near optimal solutions to the problem can be found using evolutionary computing. Not only that, but the trials documented provide insights that align with general though on how to actually play shaman. The fact that 'Chain Heal' is never cast in any of our final rotations makes logical sense when you analyze what the spell specifically does. 'Chain Heal' heals for a relatively low amount of percent healing and requires a 2.5 second cast. This makes it comparatively time inefficient compared to the other spells in the context of this study. In a real game scenario chain heal affects more than one target and is used for group healing. However, in our study (which is focused on finding the ideal rotation for healing one single target) it is objectively the worst spell available and thus it makes logical sense that our populations have evolved to omit this allele.

The spell 'Unleash Life' being followed by either a cast of 'Riptide' or 'Wellspring' which initially seemed strange, however upon further evaluation it actually bolsters a novel concept. Initially one would assume that 'Unleash Life', which provides a 35% increase to the next healing spell cast, would be followed by a spell like 'Healing Surge' which has a high base percentage healing. The idea being that 35% of 300% is more than 35% of 170%. However, the interesting thing to note about 'Riptide' and 'Wellspring' are that they are spells that have cooldowns and low (or in the case of 'Riptide' no) cast time. Meaning that each cast they are involved in limits how they can be cast after, whereas spells with a higher base percentage healing do not have cooldowns and thus can more flexibly be used throughout the rotation.

Another sign that our evolutionary algorithm is adapting to the game concepts is the timing of when it casts 'Healing Stream Totem'. If you look up a guide for restoration shaman in World of Warcraft, one of the ideas that is promoted quite often is the concept of using spells that have short cooldowns (note a spell not having a cooldown does not mean it has a 'short' cooldown, it means it has no cooldown) first followed by spells that have longer cooldowns. This is an instance where my own game intuition seems to have been proven wrong. Healing Stream Totem provides benefits over a relatively long stretch of time (15 seconds) and thus in my personal thought process made it a desirable target as a spell to be cast very early on in an encounter. The reasoning being that if a totem is activated immedietly then you will recieve those buffs for the next 15 seconds. The spell having no cast time also makes it seem like a harmless cast at the start of a rotation. However, our initial results from the first ten casts of our trials shown in Figure 3 shows that on average it ideally should be cast after certain low cooldown spells like 'Riptide' or 'Unleash Life'.

Ultimately, our proposed evolutionary algorithm gets close to finding a global optimum, the elusive 'mathematically proofed best shaman healing spell rotation' but fails to achieve this goal. It finds solutions that we believe are near optimal, but does not find a definitive 'best' solution. While it is possible that there is no true 'best' rotation, it is more likely that our algorithm just needs more trials in terms of trial iterations and the total number of trials to find a true optimum. However, we believe no large architectural change is required. The algorithm is demonstrably capable of uncovering trends and patterns within the solution space which is believed to be the key to finding the ideal solution.