DEVELOPING AND OPTIMIZING A SIMPLE MODEL FOR THE UNDEAD DEFENSE

ALIEN@SYSTEM

Abstract. Development of a model to evaluate the quality of a spawn distribution for the Undead Defense and the creation of a spawn distribution based on this model.

1. The Front Line Function

Definition 1. *A* Model *of the spawn distribution shall be a 4-tuple of real functions defined for all integers in* $[1, 20]$*, called* P_1 *to* P_4*, respectively. Within the specified interval, the functions fulfil the equation*

$$\sum_{i=1}^{4} P_i = 1 \tag{1.1}$$

at all times.

This is based directly on the implementation of the Undead Defense spawn within the game, with each of the P_i representing the probability that a spawned unit is created in the Spawn Zone of index i.

Based on these values, one has to estimate the flow of the battle as it evolves over time to evaluate the quality of any particular model. Ideally, the front line progresses backwards steadily from round one to round twenty, without any obvious jumps or stagnations. As a comparison, we can define a *base function* defined over [1,20] which describes an idealised movement of the front line during the game. As the simplest and most sensible possibility, the base function used in this article is a linear function, rising from the start of area one, at the coordinate 0, in round one to the end of area four, at the coordinate 4, in round twenty.

We can compare our front line curve to this base function and use the least squares method to find the optimal parameters to fit our curve to the desired shape. To find the front line curve, let us assume the following combat model:

Proposition 1. *The amount of killed enemy units is directly proportional to the amount of units spawned in that area. There is no influence of other variables. Defending units do not move between areas. No units are retained between rounds.*

This is an extremely simplified assumption which does not hold for very low or high numbers of units, but has the advantage of simplifying all following equations. One could find a better model using Lancester's law to take the actual amount of attackers entering that area into account.

For this article, the linear approximation shall be used as sufficiently accurate. Based on that, we immediately see that for a given Model, the amount of killed

Date: October 4, 2013.

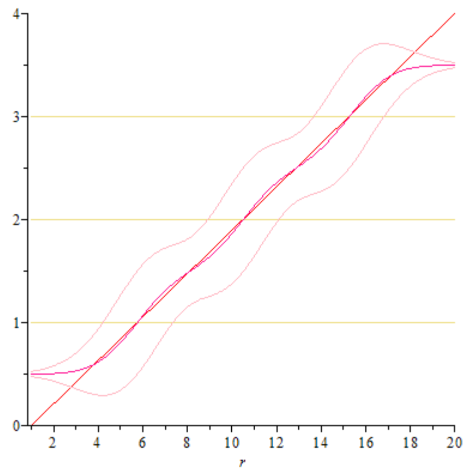

FIGURE 1. Example of base function and front line function

units is directly related to our P_i. Assuming that the spawn locations are in the centre of the corresponding areas, at the coordinates 0.5, 1.5 etcetera, we arrive at the following definition:

Definition 2. *The* front line function F_M *of a model* M *is a function on* $[1, 20]$ *defined as*

$$\sum_{i=1}^{4} P_i \cdot (i - 0.5) \tag{1.2}$$

Similarly, the front line width *shall be the standard deviation thereof, being*

$$\sqrt{\sum_{i=1}^{4} P_i \cdot (i - 0.5 - F_M)^2} \tag{1.3}$$

In figure 1, we can see an example for a front line function, the base function and the front line width as a 2σ-environment around the front line function. As is obvious from the definition, the front line function cannot rise above 3.5 and similarly cannot fall below 0.5. This produces an unavoidable discrepancy between our base function and front line function and obviously prevents a perfect fit.

To remove this problem, one can introduce a constant s and remove the ill-fitting first and last data points and only consider the interval $[1 + s, 20 - s]$ when using the least squares method. This increases the accuracy in the center part while increasing it at beginning and end. By choosing an s of 3, we can find optimal results for the area in which linearity is possible.

2. Optimization of the Current Model

The presently used model assumes that, as the combat progresses through the four areas, we can use normal distributions to describe the weight of an area in respect to time. As it is reasonable to assume, the first function should be focused on round 1, when almost all Defenders should spawn in the first area, and similarly

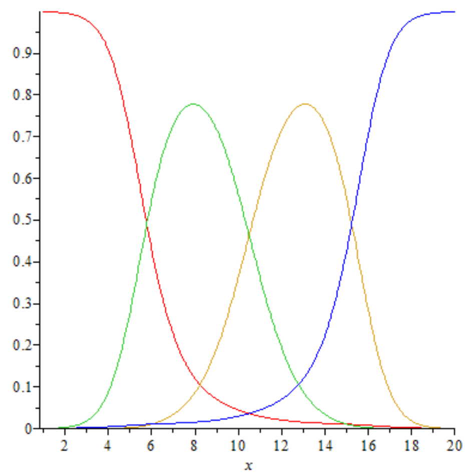

FIGURE 2. The P_i of the current model over the round number

the fourth function should be centred on round 20, for analogous reasons. The two other function are centred somewhere between, but their exact location might vary. We therefore shall leave them as parameters of our Model and arrive at the following proto-probability functions:

$$f_1(r) = \frac{1}{\sigma_1} \cdot \phi\left(\frac{r-1}{\sigma_1}\right) \tag{2.1}$$

$$f_2(r) = \frac{1}{\sigma_2} \cdot \phi\left(\frac{r-\mu_2}{\sigma_2}\right) \tag{2.2}$$

$$f_3(r) = \frac{1}{\sigma_3} \cdot \phi\left(\frac{r-\mu_3}{\sigma_3}\right) \tag{2.3}$$

$$f_4(r) = \frac{1}{\sigma_4} \cdot \phi\left(\frac{r-20}{\sigma_4}\right) \tag{2.4}$$

As these functions do not fulfil our required condition (1.1), we have to first normalize them by dividing each function through the total, arriving at

$$P_i = \frac{f_i}{\sum_{i=1}^{4} f_i} \tag{2.5}$$

We have thus found the Model corresponding to our assumptions and can optimize. Using the Fit command of Maple with the base function and truncation described above results in the following constants for ideal fit:

	1	2	3	4
μ	1	8.95	12.05	20
σ	4.81	1.88	1.88	4.81

Interestingly, the values for μ_2 and μ_3 do not coincide with the intersection of the base function and the spawn coordinate. This stems from the function being skewed out of shape by the normalization, resulting in a shift of their maxima.

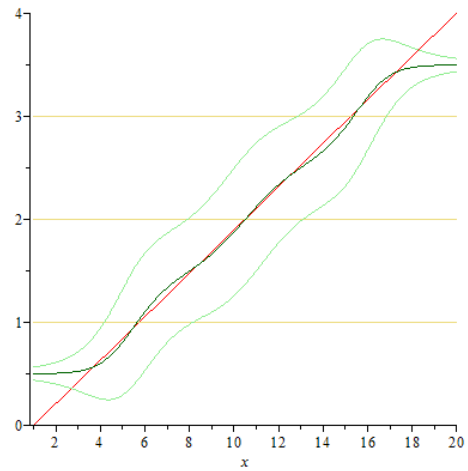

FIGURE 3. Front line curve for the current model, best fit

As is evident in figure 3, the thus achieved front line curve follows the base function nicely already. However, there are still dents where it doesn't follow the desired shape perfectly.

3. Finding a New Model

The assumptions the current model is based on are to some extend arbitrary and hindering themselves. While the assumption of Gaussian shapes for the spawn rates in the areas is a sound decision, the following necessary step of normalizing them causes skewing. For sufficiently badly chosen parameters of σ_i, the front line function will stop being monotonous.

Therefore, a new approach might be necessary. Starting from the base function, it is easy to conclude that for an ideal Model, the death density curve of the attackers will be a distribution that has as its mean the base function. As no better data can be derived from our model, it is natural to assume that the death density is a Gaussian distribution that follows the base function.

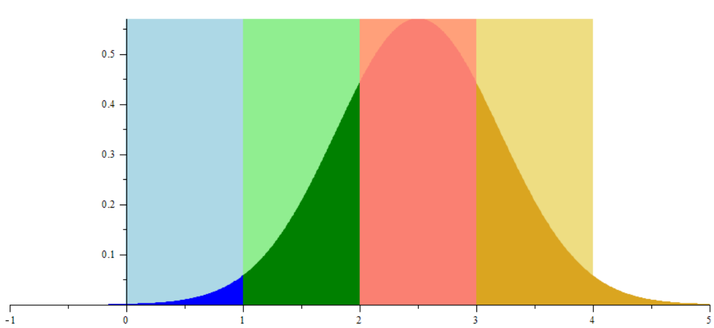

FIGURE 4. Death density curve and the resulting integrals for the P_i

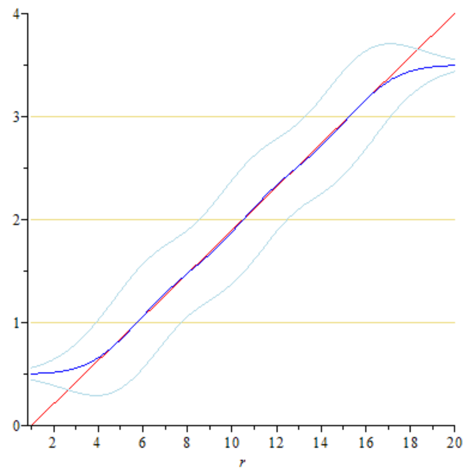

FIGURE 5. Front line curve for the new model, best fit

The amount of units necessary to spawn within any area can be of course easily derived from the death density function by integrating it over the intervals the areas cover.

This is illustrated in figure 4, with the saturated areas representing the integrals and the less saturated ones representing the domains. For the purpose of illustration, the standard deviation of the distribution has been increased compared to the optimum. As the Gaussian function extends over the limited area, the integrals of the outer points are also extended towards infinity.

As the integral of a Gaussian distribution is the Error function, we find our P_i. Since the integral over the distribution is always 1, it is not necessary to normalize the results. Therefore, our Model now consists of the following, where B is our base function:

$$P_1 = \frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{1-B}{\sigma\sqrt{2}}\right) \tag{3.1}$$

$$P_2 = \frac{1}{2}\operatorname{erf}\left(\frac{2-B}{\sigma\sqrt{2}}\right) - \frac{1}{2}\operatorname{erf}\left(\frac{1-B}{\sigma\sqrt{2}}\right) \tag{3.2}$$

$$P_3 = \frac{1}{2}\operatorname{erf}\left(\frac{3-B}{\sigma\sqrt{2}}\right) - \frac{1}{2}\operatorname{erf}\left(\frac{2-B}{\sigma\sqrt{2}}\right) \tag{3.3}$$

$$P_4 = \frac{1}{2} - \frac{1}{2}\operatorname{erf}\left(\frac{3-B}{\sigma\sqrt{2}}\right) \tag{3.4}$$

The only free parameter left is the width of the input distribution. Just as in the current model, choosing it too small will result in the front line curve having large steps. Choosing it too large will result in a lazily rising curve that does not follow our base function nicely. Using the same method as above, we can find the best fit to our base function with Maple, which has a σ of 0.369. The resulting curve can be seen in figure 5. As we can see, it follows the slope closer than the other model.

4. Comparing the Two Models

As can be seen on the comparison of the two front line curves in figure 6 on page 7, the new model (blue) is generally closer to the base function than the current model (green). The transition into and out of the slope is also smoother for the new model. In terms of the measurement defined in the first section, the new model is superior to the current.

There are however other factors to consider. For example, the front line width of the new curve is narrower than before. This can be seen even more clearly in figure 7, which compares the standard deviations of both distributions directly. Both follow a shape with three tips which is caused by the discrete spawn distribution. When the majority of all units spawn in one location, the deviation is lower than it is when the base function is between areas and two spawn points produce about equally many defenders.

As we can see, the old model is generally higher than the new one, has a greater variation between the extrema and falls off faster towards both ends. Those effects are however not necessarily bad, as a wider front width gives the heroes a greater influence when it comes to pushing forward.

Another factor to consider is the computational cost. The new model requires a calculation of the error function, which is non-elemental and not readily available in Jass. The old model only requires elementary operations, which make it possible to be calculated at runtime. Given however the complexity of the operations, a faster method might be to create a lookup table with another program and import it into the map before compilation. I that case, the computational complexity is no longer relevant.

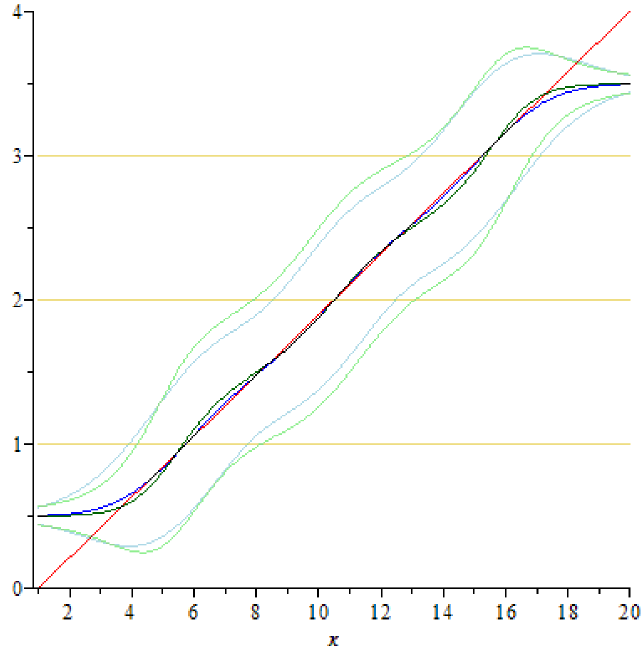

FIGURE 6. Comparison of the best-fitting front line curves of both models

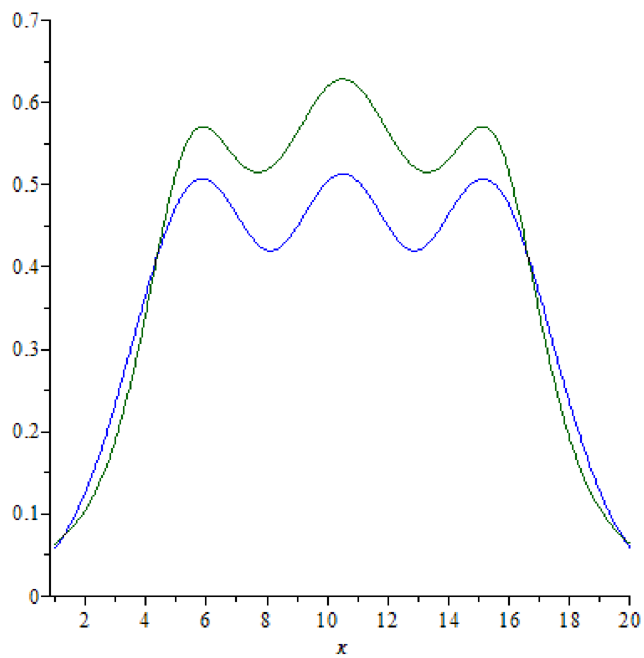

FIGURE 7. Comparison of front line widths of both models.