D3: Affix & Item Optimization Analysis

How to determine expected improvement costs and gains

Introduction

This document is meant as a guide on how to accurately calculate the expected values of the costs and gains of enchanting items in Diablo 3. The guide mostly addresses ambitious players with already really good gear, such that the timer required for the calculation is small compared to the time potentially wasted on maxing out enchantment possibilities in the wrong order. It introduces and applies the mathematical tools that are also a prerequisite to a more difficult item reforging analysis.

The following calculations require the specific affix chances to be known in order to be applicable. Fortunately, they have successfully been datamined and can be looked up here. Without this great achievement, no accurate analysis on this subject would be possible.

Affix Rolls & Enchanting Basics

When an item is generated in Diablo 3, a certain number of stats called affixes are randomly chosen for it following a weighted probability distribution comprised of all eligible affixes (which can be looked up at d3planner for instance) each with their specific weight from the previously linked table. When an affix is chosen, it is removed from the pool for this item and can’t roll twice. This alters the probability distribution of the subsequent rolls and means that the entire sequence of rolls for all affix slots of one item is not a sequence of independent and identical random processes. The sequence as a whole instead follows a weighted hypergeometric distribution, but using the combined information of eligibility and weights of each affix for each item, the precise probability of a certain set of affixes on a certain item can be calculated as the sum of all the set’s permutations’ probabilities. This is most important for rerolling items as a whole, but through the presentation of a choice between two rolled affixes which cannot be the same during enchantment, it becomes relevant to enchantment as well. Enchanting can present two incompatible affixes, but never two identical ones.

Furthermore, affixes are separated in two categories, primary and secondary. They don’t influence each other except - and this is important - if Resistance to All is rolled as primary, single elemental resistances cannot be rolled as secondary. Strictly it is not known and not backed up by statistical testing whether the secondary affixes are rolled first and a single resistance realization restricts all resistance, or primaries are rolled first and an all resistance realization restricts single resistances, but the naming as “primary” and “secondary” affixes strongly suggests the latter to be the case, which hence will be assumed to be the case from now on.

There are several instances of affixes in the same category being mutually exclusive, like elemental damage bonuses on bracers and amulets and weapon damage on weapons. In these cases, a realization removes not only the realized affix itself from the set of eligible affixes but also others for subsequent rolls.

Apart from deciding which affixes are chosen, the game also decides the values of each affix. These follow a uniform distribution, meaning that every possible value is equally likely.

Probability & Math Basics

This chapter is a brief introduction to the discrete probability calculus tools we need for this analysis.

Discrete Random Variables, Expected Value

Let X be a discrete random variable that may assume any value x_i with probability p_i for a countable (not necessarily finite) number of possible values.

Then $E[X] = \sum_i p_i * x_i$ is the expected (or average) value of that random variable, if the sum exists. It is also the barycenter of the underlying probability distribution.

Benefit & Efficiency

Not a traditional subject of mathematics but important to our application to video games is the consideration of benefit and efficiency of certain stat increases for certain costs.
Benefit means the relative increase of a considered value function due to the increase in the stat. For most simple stats in Diablo 3, we can take Damage and Toughness as the value functions in question, with notable exceptions being attack speed, cooldown reduction, resource cost reduction and resource generation. In general, we have $ben = \frac{new\ value}{old\ value} - 1$. If competing upgrades with different benefits and costs are to be compared in regards to efficiency, there are two ways to define a metric for it:
$eff = \frac{ben}{cost}$ and $eff = \sqrt[cost]{1 + ben} - 1$.
It is often difficult to distinguish which is the proper one to use, and in fact, not even clear on the subject treated here (enchanting / rerolling efficiency). More on this later.

If an actual benefit is not certain but subject to a random variable, we consider the expected benefit instead.

The Calculation

Introductory Example

We have two dice and start rolling them, only one at a time, while reserving the other. We try to get higher numbers and after rolling another die, we can choose to keep the old one or the new one, whichever was better, and re-roll the other one. The first number rolled is 4. What is the expected benefit of the next roll?

$$ben = \frac{1}{6} * (0 + 0 + 0 + 0 + \frac{1}{4} + \frac{2}{4}) = \frac{1}{6*4} * \sum_{i=1}^{6-4} i = \frac{1}{6*4} * \frac{(6-4)*(6-4+1)}{2} = \frac{2*3}{6*4*2} = 1/8 = 0.125$$

Generalization to Benefit

We can generalize this to any uniform random variable with lower bound a, upper bound b, and previous maximum (current value on which we try to improve) x_0. Our example simply used the values a=1 (lowest number on a die), b=6 (highest number on a die) x_0=4 (initial roll).

$$ben = \frac{1}{(b-a+1)*x_0} * \sum_{i=1}^{b-x_0} i = \frac{(b-x_0)*(b-x_0+1)}{(b-a+1)*x_0*2}$$

Furthermore, we want to generalize to have eligible values from a to b not with a fixed step size of 1, but any given step size s. After some transformation, we get:

$$ben = \frac{(b-x_0)*(b-x_0+s)}{(b-a+s)*x_0*2}$$

The last thing that is specific and ought to be generalized in this formula is how the value is measured. In this case, we had the random variable itself being immediately equal to the value function used to calculate the benefit, which resulted in having x_0 in the divisor. Usually though, the value is not a proportional function of the random variable (the affix on the item we are rerolling), but just a linear one, meaning that it has a constant offset which doesn’t eliminate in the fraction. A good example of this would be the crit stats, Critical Hit Chance and Critical Hit Damage. Both contribute linearly to damage. This makes the benefit of a fixed increase ccb:

$ben_{cc} = \frac{1+(cc+ccb)*cd}{1+cc*cd} - 1 = \frac{1+cc*cd+ccb*cd}{1+cc*cd} - 1 = 1 + \frac{ccb*cd}{1+cc*cd} - 1 = \frac{ccb*cd}{1+cc*cd} = \frac{ccb}{\frac{1}{cd}+cc}$ and exchanging ccb with cdb, cc with cd and cd with cc for the reverse case (bonus on crit damage).

Replacing x_o in the divisor with the new divisor (cc+1/cd) for crit chance bonuses or (cd+1/cc) for crit damage bonuses gives us the expected value of the benefit of re-rolling a crit stat, i.e. under the condition that an enchantment roll offers the crit stat, what its expected benefit will be.

To get to the general expected benefit of rolling the enchantment in the first place, we still need to multiply the so far obtained benefit with the probability of being offered the missing crit stat on enchantment roll.

Enchantment Probabilities

Each time we enchant, we're being presented two non-identical affixes from the eligible pool. Remember that affix chances are weighted according to the initially linked table (note that the weights are all multiples of 100, so dividing all of them by 100 will make actual calculations easier).

When the enchantment is rolled, the first one could be the desired affix, or - if this was not the case - the second one could be after the first one being each of the other eligible affixes.

Suppose 4 eligible affixes A, B, C & D with specific weights w_A,w_B,w_C & w_D and w_T being the sum of all of them. We want the D. The total probability of it being offered is:

$$p_D = \frac{w_D}{w_T} + \frac{w_A * w_D}{w_T * (w_T - w_A)} + \frac{w_B * w_D}{w_T * (w_T - w_B)} + \frac{w_C * w_D}{w_T * (w_T - w_C)}$$

$$= \frac{w_D}{w_T} * \left(1 + \sum_{i=A,B,C} \frac{w_i}{w_T - w_i}\right) = \frac{w_D}{w_T} * \left(1 + \sum_{i=A,B,C} \frac{1}{\frac{w_T}{w_i} - 1}\right)$$

The sum iterates on all affixes in the pool with the exception of the desired affix itself, as this is already included in the first summand. As practically, there are many eligible affixes and many of them have equal weights, it is often feasible to group equal weights together and instead of considering the set of individual affixes, consider the multiset of affix weights instead, with multiplicities excluding the desired one (while it's still included in w_T):

$$p_D = \frac{w_D}{w_T} * \left(1 + \sum_i \frac{m_i}{\frac{w_T}{w_i} - 1}\right)$$

Unfortunately there are no general solutions to this for an item, as the result depends on which other affixes are realized on the item being enchanted, not just which item it is. Let's consider an example nevertheless:

Focus/Restraint with a socket, mainstat and crit chance - enchanting for crit damage. Eligible affixes open:

- 15 - weapon damage

- 40 - Vitality
- 10 - armor, res all *, life%, lph, lps, asi, chd, ad,cdr, rcr

Res all is only in if there is no secondary single resistance rolled.
w_T is then 15+40+10*(9 or 10) = 145 or 155.
The probability that one of the two presented affixes is crit damage is then:

$$p_{CHD} = \frac{10}{\{145,155\}} * \left(1 + \frac{1}{\frac{\{145,155\}}{40} - 1} + \frac{1}{\frac{\{145,155\}}{15} - 1} + \frac{\{9,10\}}{\frac{\{145,155\}}{10} - 1}\right) = \{14.917\%, 13.836\%\}$$

Multiplying this with the previously evaluated benefit would yield the corrected expected benefit for one enchant. To finish the example, suppose we have 50% crit chance and 500% crit damage including the current enchanted value of 35%. Also let's assume there is a secondary resistance fixed. Then we get for the expected benefit:

$$ben = 0.14917 * \frac{(0.5-0.35)*(0.5-0.35+0.01)}{(0.5-0.25+0.01)*2*(5+1/0.5)} = 9.836 * 10^{-4}$$

This is what we can compare to other values for other equipment parts that are still not enchanted to their maximum to decide which should be prioritized.
If the prices for enchanting or re-rolling competing items differs in a restraining resource, the proper efficiency metric ought to be used for the comparison instead.

Total Reforging Probabilities

When an entire item is re-rolled rather than just one of its affixes, regardless of whether this is achieved through reforging or obtaining it anew by any means, the total probability of any desired affix configuration can be determined with a calculation very similar to that for enchanting a single affix, with some slight but important differences.

The affix configuration can be visualized as a random process of as many consequential steps as there are affixes to configure. Because different permutations of the same combination of affixes can have different probabilities, they can't be grouped together. The total probability of any certain affix configuration is hence the sum of the probabilities of each of its permutations. Those in turn are calculated just like enchantment probabilities, but with more than two steps, and with incompatible affixes subsequently excluding each other too rather than just identical ones.

The number of summands that have to be included grows considerably as the option of enchanting introduces new degrees of freedom. In general, suppose m desirable primary affixes shall be realized on an item with m primary affix slots. Then one needs to consider all subsets of size m-1 of the set of desired affixes, and all permutations for each of the subsets enhanced by one variable position (whose numerator is the complement to the still available desired affixes' combined weight in the total available combined weight, and the following denominator becomes the expected value of the remaining weight after this variable draw). If some affixes are predefined on the item in question, only the remaining random affix slots are to be considered. If one of the predefined affixes is non-desired, it demands and consumes the enchanting option and hence requires all the random affixes on that item to be desired ones, in which case one still has to consider all permutations of m-1 subsets of the m desired affixes (to check, there are m-1 random affix slots), but without the variable "wild card" slots. To prove the exhaustiveness of a proposed term, one may check that the number of considered subsets of size m-1 from the set of size m is indeed exactly m (because any one of the m items in the set may be left out). In short, all branches of the related probability tree which yield a realization that through enchanting can be transformed into the desired realization have to be taken exactly once and added together. Equally probable branches may be grouped.

Example: Obtaining a Hellfire Amulet with crit chance, crit damage, a certain elemental damage, a slot and a desired passive (assuming 19 passives for the class, e.g. Barb, DH).

$$p_{affixes} = p_{passive} * 2 * (p_{c,c} + p_{c,el} + p_{el,c}) = \frac{10}{19} * (\frac{10*10}{205*195} * 2 + \frac{10*10}{205*165}) = 0.4189\% = 238.71^{-1}$$

In this case, since the values of the affixes haven't been considered yet, the two crit stats could be grouped together as they have equal weight, and if elemental comes second it also yields the same branch probability. Elemental first gives a higher chance, however, because it eliminates the other elements from the affix pool as well for the second pick. Secondary affixes have been disregarded. So, on average, for a class with 19 passives, one out of ~238.7 crafted Hellfire Amulets will have a desired passive and a set of primary affixes that can, through enchanting, be transformed into {crit chance, crit damage, elemental damage} for one specific element.

We can take this example a step further by including the values and asking for the probability of a potentially perfect Hellfire Amulet (in terms of primary affixes, i.e. not requiring it to be ancient and not posing requirements on the secondary affixes). For this, both rolled affixes are required to be rolled on their maximum value, since the enchantment has to be used to transform the primary attribute into the third missing affix. As the probability distribution over possible affix values is believed to be uniform, the probability of any individual possible value, including the highest one, is given by: $p_x = \frac{s}{b-a+s}$ in accordance to previous variable definitions. For a potentially perfect-affixed Hellfire Amulet we hence get:

$$p = \frac{5}{19} * (2 * \frac{10*10}{205*195} * \frac{0.01}{1-0.51+0.01} * \frac{0.005}{0.1-0.08+0.005} + (\frac{10*10}{205*195} + \frac{10*10}{205*165}) * \frac{0.01}{1-0.51+0.01} * \frac{0.01}{0.2-0.15+0.01}$$
$$+ (\frac{10*10}{205*195} + \frac{10*10}{205*165}) * \frac{0.005}{0.1-0.08+0.005} * \frac{0.01}{0.2-0.15+0.01}) = \frac{5}{19} * (\frac{4}{199{,}875} + \frac{8}{439{,}725} + \frac{16}{87{,}945}) = 5.793 * 10^{-4} = 17,261.93^{-1}$$

Meaning that on average, one out of about 17,262 crafted Hellfire Amulets will have two perfect primary affixes from {crit chance, crit damage, element} where element is one specific element you desire and a useful passive. None of the affixes benefits from the item rolling ancient, but if you were to impose this requirement as well, the probability is further divided through 10, meaning only one out of 172,619 amulets fits them. On a side note, looking at the conditional probabilities, such an amulet will have realized crit chance and elemental 82.64% of the time due to the probability of critical damage rolling perfectly being so low compared to the others.

The Hellfire example shows how various requirements and aspects can be logically incorporated into a probability formula:

Mathematical operation	**Logical meaning**
Multiplication (a * b)	“both a and b”. Connecting requirements, only accepting their intersection.
Addition (a + b)	“a or b” (inclusive “or”). The union of the two criteria.
Division ($\frac{a}{b}$)	“a out of b”. Only makes sense if a is a subset of b, returns the ratio of the magnitudes of subset a and set b.
Subtraction (a - b)	“a without b”. Only makes sense if b is a subset of a, returns the complement to b in a.

Furthermore, the example highlights the miniscule chances of perfect gear and that trying to obtain it is mostly futile. Instead, we want to grasp the probabilities of all realizations constituting an improvement over status quo and incorporate those with their benefit magnitudes into expected benefits, much like we did in the introductory example.

Benefit Functions for Various Affixes

In order to evaluate the benefits of increases of certain character stats through item affixes, their contribution to the three value functions (damage output, toughness and recovery) have to be examined. Unfortunately these cannot be unified as beating a certain Greater Rift level poses requirements on both your damage potential and your survivability (both in regards to burst damage and sustained damage) which cannot really be compensated by each other. Depending on whether dying too often (suddenly or over time) or taking too much time to kill enemies is what limits your progress, increasing the respective value function should be focused. The general structure of these value functions is:

$Toughness = Health \div (\Pi(incoming\ damage\ multipliers))$

$Recovery = (\Sigma(healing\ per\ time)) \div (\Pi(incoming\ damage\ multipliers))$

$Damage = (\Sigma(damage\ per\ time)) * \Pi(general\ outgoing\ damage\ multipliers)$

The summands within the sum in the damage formula are, in return, products. Most of them usually share a lot of factors, which may then be factorised into the external product. An example of this would be the mainstat damage multiplier that is a common factor to almost all sources of damage, the only exception I know of being Death's Bargain. It should be mentioned that these value and hence the benefit calculations can't be exact because they partially rely on variables which have to be estimated. A prominent example of this is how many enemies (on average, weighted with their hitpoints) are hit by a certain skill or the area damage effect it triggers. Another example are single elemental resistances, these require accurate proportions of the incoming damage element types to be properly evaluated, yet these proportions again can only be guessed. Nevertheless, most stats can be examined with satisfying precision. In this chapter, the relative benefits of increases in all stats in regard to those value functions will be determined. Character level 70 is assumed. X is always the increase in the examined stat.

Strength

Strength increases the character's armor by one per point and the damage of Barbarians and Crusaders by one percentage point per point, almost no exceptions. Only a miniscule deviation if Death's Bargain is used, and possibly some other odd old legendaries.

$$ben_T = ben_R = \frac{x}{\frac{\frac{3500}{AM} + \Sigma A_{equip} + Dex}{SM} + Str}$$

$$ben_{D(Barbarian,Crusader)} = \frac{x}{\frac{100}{SM} + Str} - \varepsilon$$

Where A is armor, AM is armor multiplier (e.g. from paragon, Tough as Nails), SM is strength multiplier (as from Finery), ε is the miniscule deviation (may be neglected).

Dexterity

Dexterity works the same way as Strength, only that its damage increase applies to Monks and Demon Hunters instead.

$$ben_T = ben_R = \frac{x}{\frac{3500}{AM} + \Sigma A_{equip} + Str*SM + Dex}$$

$$ben_{D(Demon\ Hunter, Monk)} = \frac{x}{100 + Dex} - \varepsilon$$

With homologue variable definitions. There is no Dexterity multiplier currently implemented into the game as of my knowledge, hence no such variable appears in the formula.

Intelligence

Intelligence increases the damage of Witchdoctors and Wizards by one percentage point per point. It also increases the resistances against all elements by 0.1 per point.

$$ben_{D(Witchdoctor, Mage)} = \frac{x}{\frac{100}{IM} + Int} - \varepsilon$$

$$ben_{T,R} = \frac{\sum_i^{elements} \left(\frac{w_i}{350 + (R_{o,i} + Int * \frac{IM}{10}) * RM}\right)}{\sum_i^{elements} \left(\frac{w_i}{350 + (R_{o,i} + (Int + x) * \frac{IM}{10}) * RM}\right)} - 1$$

Where Ro,i are the total resistances to element i from other sources than Intelligence, before multipliers. IM is Intelligence multiplier (Gruesome Feast), RM is Resistance multiplier (e.g. Mantra of Salvation), wi are the specific weights with which element i is to be considered. Evaluating these requires estimating the proportions of incoming damage elements against each other. If an elemental immunity amulet is used, the weight for that element can be set to 0, which is equivalent to setting the resistance against this element to positive infinity.

If the resistances against every element are very similar, the defensive benefit can be approximated by:

$$ben_{T,R} \simeq \frac{x}{Int+\frac{10}{IM}*(R_o+\frac{350}{RM})}$$

Since the possible sources for deviating single resistances are restrained to the limited secondary affixes on gear, whereas Intelligence as a source for common resistance to all elements grows with paragon level, the actual benefit will approach this value at higher levels.
For Ro, the (mean) Resistance against all elements from other sources than Intelligence, any mean value can be used since the premise of the single values not differing much from each other makes the differences between various mean metrics miniscule, and using an elaborate mean metric eliminates the practicability of an easy approximation, to the point where one may as well use the exact formula instead. For the sake of completeness however, the best standard mean to be used on this occasion is the Harmonic mean (optionally with weights for the elements), and the overall best possible evaluation for the mean Resistance is:

$$R_{mean} = \frac{\Sigma w_i}{\Sigma\frac{w_i}{350+R_i}} - 350$$

All those defensively beneficial properties of Intelligence are equivalent to Resistance to All Elements, except for the involvement of an Intelligence multiplier.

Armor

This is equivalent to the defensive aspect of Dexterity and Strength since those also increase armor, just that a strength multiplier only affects armor from Strength and not from item affixes.

$$ben_{T,R} = \frac{x}{\frac{3500}{AM}+A}$$

Here, A is the total armor from all sources (Dexterity, Strength, items) before multipliers.

Resistance to All Elements

See Intelligence. This time however, resistances don't have to be divided into their Intelligence and other parts, respectively. Instead, the entirety of Intelligence after any multiplier can simply be included in the current resistance value, only resistance multipliers stand separate.

$$ben_{T,R} = \frac{\sum_{i}^{elements}\left(\frac{w_i}{350+R_i*RM}\right)}{\sum_{i}^{elements}\left(\frac{w_i}{350+(R_i+x)*RM}\right)} - 1$$

Again, the approximation for similar values of each single elemental resistance is applicable under this condition:

$$ben_{T,R} \simeq \frac{x}{\frac{350}{RM}+R_{mean}}$$

With Rmean denoting their common mean value, the best metric for which can be found under Intelligence.

Vitality

Each point of Vitality increases maximum life before the Life % multiplier by 100. There also is a small constant amount of maximum life (before the multiplier) of 316 at level 70.

Unlike all previous defensive stats, Vitality does not apply to Recovery in the full extent as it applies to Toughness: Vitality only increases those components of health recovery which are linked to a fraction of maximum health as opposed to a fixed amount, hence the return on Recovery is significantly diminished.

Vitality also doesn't affect damage immediately, only indirectly through a few legendary item affixes. The most notable case of this is Heart of Iron being an integral part to all Thorns Crusader builds, therefore it's included here.

$$ben_{T} = \frac{x}{Vit+3.16}$$

$$ben_{R} = \frac{x}{Vit+3.16+\frac{rate_{flat}}{rate_{\%}*(1+Life\%)}}$$

$$ben_{D(Heart\ of\ Iron)} = \frac{x}{Vit + \frac{T_o + WD * \frac{WDM}{TDM}}{r_{HoI}}}$$

Where Vit is Vitality, rate_flat is the amount of life you regain per second independent of your maximum life (from LPS, LPH etc.), rate_% is the total rate at which you regain a percentage of your health in %/sec (from healing potions and abilities with % heals ; note that when this is 0, the formula becomes invalid and the benefit for Recovery from increased Vitality is 0 instead), Life% is any percentage bonus to your maximum life (from affixes and paragon) as a decimal fraction, T_o is flat Thorns damage from other sources, WD is weapon damage, WDM is weapon damage multiplier (how many times your weapon damage you apply per time in damage output after all), TDM is your thorns damage multiplier (how many times your thorns damage you apply per time in damage output after all), and r_HoI is the Vitality-to-Thorns conversion rate of your Heart of Iron as a decimal number (i.e. from 2.5 to 3, not in %).

Life %

Similarly to Vitality, Life% as a stat primarily only affects Toughness and only extends a benefit to Recovery if %-of-max-life-healing is present. As of my knowledge, there is no source of total life to damage conversion implemented at the current state, so Life % is a purely defensive stat.

$$ben_T = \frac{x}{1 + Life\%}$$

$$ben_R = \frac{x}{1 + Life\% + \frac{rate_{flat}}{rate_{\%} * (Vit + 3.16)}}$$

Keep in mind Life% is a decimal fraction whereas rate_% is in %/sec.

Critical Hit Chance & Damage

Any damage output that is based on weapon damage profits from critical hits, while any thorns-related damage does not. Critical hits are purely offensive.

$$ben_{D,cc} = \frac{x}{cc + \frac{1}{cd} + \frac{TD * TDM}{WD * WDM}}$$

$$ben_{D,cd} = \frac{x}{cd + \frac{1}{cc} + \frac{TD*TDM}{WD*WDM}}$$

Note that unlike in the Heart of Iron damage benefit formula, weapon damage multiplier here does not include the average critical hit damage multiplier. The third summand in the denominator indicates the ratio of net thorns damage output to net weapon damage output and is usually either very big (Thorns build), so the benefit of crit gets close to 0, or very small (weapon damage build), in which case this third summand can be neglected and the benefit of crit gets close to $\frac{x}{c1 + \frac{1}{c2}}$. Hybrid builds are very uncommon, but this formula allows to calculate benefits for them nevertheless.

Life per Second

Life per second is a passive, unconditional and continuous way of regenerating life. It only benefits Recovery, in a way that is straightforward:

$$ben_R = \frac{x}{\Sigma regen\ per\ second}$$

Where the denominator is the sum of all the life regeneration rates per second from all sources including LPS without the new amount x. LPH * hits per second is one summand, as is any healing effect, the average intake from health orbs, healing potions, etc.

Life per Hit

Similarly to LPS, LPH is a purely regenerative statistic. Its actual increase to nominal health regeneration per second depends on the character’s actual hits per second. As such, it practically only contributes to the recovery of builds which make use of a generator / normal attack as opposed to waiting for cooldowns and only attacking when their damage skill is up. In the latter case, the benefit of LPH depends more on the actual cooldown of the ability but can be generally considered to be close to nonexistent. Hence the following formula is for builds that do have a normal attack and do attack continuously.

$$ben_R = \frac{x}{LPH + \frac{\Sigma regen\ from\ other\ sources}{Attacks\ per\ Second}}$$

Weapon damage

The benefit of weapon damage is entirely offensive as of level 70 due to the complete disabling of Lifesteal effects. In accordance to previous variable definitions, we have:

$$ben_D = \frac{x}{WD + \frac{TD*TDM}{WDM*(1+cc*cd)}}$$

Similar to critical hit stats, this benefit gets close to 0 in Thorns builds, where the Thorns Damage and the Thorns Damage Multiplier make the denominator very large. For Weapon builds on the other hand, the second summand in the denominator can be effectively treated as 0, such that the benefit is then simply almost $\frac{x}{WD}$.

The only anomaly with the calculation of weapon damage benefits is that the absolute increase amount itself is not straightforward to calculate, since a weapon damage affix consists of both a lower and an upper bound, which both are uniform random variables. The effect of weapon damage can easily be treated using the expected value, for a given pair of min and max damage this is simply $\frac{min+max}{2}$, the arithmetic mean of the bounds. Due to the linearity of expectation, weapon damage amounts from different sources can be added together this way as well without any problems (treading weapon damage as a random variable):

$$E[WD_{total}] = \sum_{i=1}^{n} E[WD_i] = \frac{1}{2}\sum_{i=1}^{n} min_i + max_i$$

This makes it rather simple to evaluate the maximum benefit that can be obtained by iteratively enchanting weapon damage on a given item using the above benefit function, as the highest value x is the sum of the differences between highest and current minimum and maximum damage, divided by two.

For example, assume a weapon damage build where the current weapon damage on the weapon is 1800-2000, and weapon damage is also rolled on one ancient ring as 90-170, such that it could be increased up to 105-210. Then the average current damage is $\frac{1800+2000+90+170}{2} = 2030$, and it could be increased by up to $\frac{105+210-90-170}{2} = 27.5$. Since the build is focused on weapon damage, the potential benefit is 27.5/2030=0.01355=+1.355%. This can be compared to the potential benefit of enchanting other non-perfect affixes on that ring instead to see which one should be chosen.

However, while the potential and the expectation behave regularly, it is important to note that the expected damage itself is not a uniform random variable like all the other affix stats we've analysed so far, but rather the sum of two uniform random variables. As such, while the benefit function is indeed correct, the enchantment probabilities formulae do **not** apply here, as they were defined for uniformly distributed stats. With mean values closer to the center of the distribution being significantly more likely than those at the boundaries, the expected costs to raise them to the upper limit are greater than if the distribution was uniform, making the efficiency (not the potential benefit) of enchanting weapon damage lower than the enchantment probability formula would suggest.

In short, this means the equations here answer *if* weapon damage should be enchanted on a given item, but not yet *when* compared to other items.

Problems

The Efficiency Dilemma

A persistent problem remains which efficiency metric is the proper one to use for enchanting / re-rolling items in Diablo 3. The nature of this problem will be illustrated in the following.

There are two differing ways to interpret enchanting the same identical item:

- Expected cost and benefit of obtaining it with *the desired* affix configuration. The affix set probability is included in the cost by multiplying the cost with the probability's multiplicative inverse (expected value of a geometric distribution). (Higher cost, higher benefit)
- Expected cost and benefit of obtaining it with *any* affix configuration. The affix set probability is included in the benefit by multiplying the benefit with the probability. (Lower cost, lower benefit)

Crucially, since both interpretations describe the very same operation (rerolling/enchanting the very same item), we demand both to yield the same efficiency value. This is only true for the ratio formula.

On the other hand, let's compare one upgrade with cost equal to the sum of two other upgrades, such that for the same price one could either obtain the first one or the two others together. If the final result of the value function is equal for both cases, then we want their efficiencies to be equal as well, and conversely, if their efficiencies are equal, then likewise we want their value function results to be equal, too.
Suppose we have one upgrade costing 2 and providing a (relative) benefit of 3, and two upgrades each costing 1 and providing a benefit of 1. Then at the cost of 2, getting either the first or the other two will yield a final multiplier of 4: $1 + 3 = (1 + 1) * (1 + 1) = 4$
Using the ratio metric, option 1 has an efficiency of 3/2 while option two has 1/1, even though their outcome is the same and hence their efficiencies should be the same, too. Using the root metric, we get $eff_1 = \sqrt[2]{1 + 3} - 1 = \sqrt[1]{1 + 1} - 1 = eff_2 = 1$, so the equality is only true for the root formula.

This leads to the efficiency dilemma: Neither efficiency metric is satisfying all criteria we impose on it, warranting the question whether a different metric has to be found which does, and if so, which it is, or there are errors in our assumptions, and if so, which they are.

Fortunately, this issue can be largely evaded in Diablo 3. What we can say without doubt is that efficiency is a bivariate function of benefit and cost, with a positive partial derivative towards benefit, and a negative partial derivative towards cost. At equal cost, the option with the highest benefit is hence also the one with the highest efficiency. Normalizing our interpretation to the act of rerolling once means we do conserve the equality of costs across all options. This was also the method shown in the Focus/Restraint example in the previous chapter, the chance of the affix configuration was included in the benefit, not the cost.
Currently in Diablo 3, reforging or enchanting a legendary item costs the same amount of Death's Breath, Forgotten Souls and act cache materials, regardless of the item being reforged/enchanted, so the cost is indeed equal and benefits can be compared instead.
There are a few exceptions to this:

- Crafted legendary items. Those are cheaper to craft anew rather than reforging them.
- Hellfire amulets. It is faster to obtain the ingredients to craft new ones than to collect the bounty materials to reforge them. To compare the cost of a Hellfire amulet to that of reforging a legendary, their cost has to be translated to expected amount of time required to conduct the re-roll, which will be different and hence requires an efficiency metric to compare.
- Some legendary items very common in their category. In some borderline cases, it may require less time investment to obtain the Death's Breaths and ordinary materials to upgrade rare items of the same category or draw from Kadala until the item is obtained again. The costs in such cases can also be quantified using the expected value of the underlying geometric distribution (trying until you get the item once, 1/p) and translated to time investment, so the cost is comparable to that of reforging in principle, but still requires a proper efficiency function.
- Horadic cache legendary items. Similar to crafted ones, it's faster to farm them directly as opposed to reforging them.

Apart from these cases, the benefit dilemma can be successfully circumvented through cost normalization.

In those cases though, or when in doubt in general, I tentatively suggest using the ratio formula until the dilemma is resolved.

Multiple Methods of Acquisition

While reforging may be the prefered method of attempting to acquire new versions of a desired legendary item, it is not the only one. While completing bounties to gain the materials required for reforging, one also collects Death's Breaths and Bloodshards in lower quantities. After everything has been enchanted to the maximum, those Death's Breaths can be used to upgrade rare items instead and Bloodshards can be used on Kadala to attempt to obtain desired items through other means.

Because unlike reforging, upgrading and gambling have different costs for different items, comparing these requires a proper efficiency function.

Since the resources for acquiring new versions of the equipment in multiple methods are obtained at different rates greater than 0, benefits and efficiencies for improving your equipment via all available means is to be compared. The tools provided previously can be straightforward applied to any acquisition method, which grants an efficiency ranking of improving any of your non-perfect equipment parts, for every method independently.

A small problem from this independent consideration occurs for example when one item is clearly the best option to be improved via one method, but only by a slim margin the best via a second method. In that case, it would generally be advantageous to use the second method on the close contestor and only use the first method on that best-to-improve item, because the efficiency at which the first method translates resources to equipment power will fall off faster if the second method saps potential for improvement by being applied on the very same item, whereas if the second method is applied to the contestor instead, it operates at an only marginally lower translation efficiency and allows the first method to operate at a significantly higher translation efficiency for a longer time, hence yielding better results in the long run by deviating from the local optimum.

Such globally optimal deviations from local optima through distributive resource allocation can not be accounted for if each resource is considered independently. Mathematical methods to account for such interactions in a more unified approach need to be developed.

Closing Statements

This document is still unfinished. Some affixes' benefit functions are still missing. Beyond that, the scope of this document merely scratches the surface of the treated topic. My intent with this early release is to introduce people to and get them engaged with the interesting mathematical background of Diablo 3. I hope this encourages others to deal with the subject, and that it may get a theorycrafting community going that is so far, considering the tremendous number of players Diablo 3 has, relatively small.

I'd like to clarify that I don't claim this work to be without error, and would like to encourage inquiry wherever aspects require a closer look - so go ahead and ask any questions that come up, I'm aware this article may be demanding at times.

Make sure to leave comments and questions in the reddit release thread.

Changelog

2016-06-02 - initial release
2016-06-03 - added weapon damage benefit