Cogstruction modefications documentation

Lily Stejko

June 29, 2022

Abstract

Don't need one, just for taking notes.

1 Learning algorithm weights

1.1 Original fitness function

Shorthand reference for section

SOF = standard_obj_fxn

Inputs

iB_w = inv_build_weight = 7000.0[1]

iF_w = inv_flaggy_weight = 2000.0[1]

iE_w = inv_exp_weight = 3.0[1]

Weights

B_w = build_weight

F_w = flaggy_weight

E_w = exp_weight

Object parameters

B_r = build_rate

F_r = flaggy_rate

E_r = exp_mult

1.1.1 Fitness function

$SOF(B_w, F_w, E_w) = B_r * B_w + F_r * F_w + E_r * E_w$

1.1.2 Calculating weights

Using the following variables.

Formula

$$\begin{bmatrix} 1 & 1 & 1 \\ iB_w & -iF_w & 0.0 \\ 0.0 & iF_w & -iE_w \end{bmatrix} \begin{bmatrix} B_w \\ F_w \\ E_w \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Linear algebra form

$B_w + F_w + E_w = 1$ (Makes sure all values sum up to 1)

$iB_w * B_w - iF_w * F_w = 0$ (Scale lock the ratio between B_w and F_w)

$iF_w * F_w - iE_w * E_w = 0$ (Scale Lock the ratio between F_w and E_w)

[1] The reason for giving the variables their starting weights is not known. The original author has abandoned the project and is not taking any questions about it at this time.

1.2 Alternative fitness function

Shorthand reference for section

F = Fitness Formula

Inputs

The input weights are input at the ratio desired.

B_{w_i} = input_build_weight

F_{w_i} = input_flaggy_weight

E_{w_i} = input_exp_weight

Adj_v = adjust_parameters[1] *(True by default)*

Weights

B_w = build_weight

F_w = flaggy_weight

E_w = exp_weight

Object parameters

B_{r_c} = build_rate for a given cog

F_{r_c} = flaggy_rate for a given cog

E_{r_c} = exp_mult for a given cog

B_r = build_rate for a cog array

F_r = flaggy_rate

E_r = exp_mult

SQ_b = usable_suares

Constants and calculated variables

C = Construction Value

Af = Affixes *(5.5 for current best cogs, $5 + 50\%$ chance of a sixth)*

Af_b = Base Affixes *(5 for current best cogs)*

Sn = Number of stats on a gear that an affix can be assigned to *(Known const, 3)*

b = Expected booster source ratio *(assumed 2)*[2]

1.2.1 Fitness function

$$F = B_w * \frac{Sn*B_r}{Af*SQ_b} + F_w * b\left(\frac{Sn*F_r}{Af*SQ_b*b}\right)^{1.25} +$$

$$E_w * b\left(\left(0.0003808514 * \left(\frac{Sn*E_r}{Af*SQ_b*b}\right)^{3.968829}\right) + \left(\frac{Sn*E_r}{Af*SQ_b*b}\right)^{1.075056} + 0.8517081\right)$$

Applying the known values:

$$F = B_w * \frac{3*B_r}{5.5*SQ_b} + F_w * 2 * \left(\frac{3*F_r}{11*SQ_b}\right)^{1.25} +$$

$$E_w * 2 * \left(\left(0.0003808514 * \left(\frac{3*E_r}{11*SQ_b}\right)^{3.968829}\right) + \left(\frac{3*E_r}{11*SQ_b}\right)^{1.075056} + 0.8517081\right)$$

1.2.2 Calculating weights

$$[B_w, F_w, E_w] = \frac{[B_{w_i}, F_{w_i}, E_{w_i}]}{B_{w_i} + F_{w_i} + E_{w_i}}$$

[1] A bool value used to determine if the values of B_r, F_r, and E_r should be adjusted to reflect their non-linear relationships to gear crafter levels.

[2] for simplicity, indicating that approximately $\frac{1}{2}$ of the expected value of the grid comes from cog base values

1.2.3 Fitness function derivation

The "construction value" (C) is the base value that determines the power of an single Affix for a given gear type crafted at a given construction level. The formula to calculate the construction value at a given skill level is:

$C = \frac{\left((\frac{level}{3})+0.7\right)^{1.3+0.05*Af_b}}{4} + 3^{Af_b-2}$

The actual bonus for a given affix can range from 40% to 300% of that value.

Formulas for B_r, F_r, and E_r:

$B_{r_c} = C$, the relationship is 1 to 1 (and therefor linear)

$F_{r_c} = C^{0.8}$, the relationship is not linear.

$E_{r_c} = C^{0.4} + LOG10(C) * 10 - 5$, the relationship is not linear.

The inverse formulas are:

$C = B_{r_c}$

$C = F_{r_c}^{1.25}$

$C = b * 388.428W\left(0.145974 * \sqrt[25]{10^{E_{r_c}}}\right)^{\frac{5}{2}}$

The last formula contains a W which is too complex to use in practice but

$C = (0.0003808514 * E_{r_c}^{3.968829}) + E_{r_c}^{1.075056} + 0.8517081$

is much faster to calculate and is a good approximation formula for realistic values.

The cog array:

Every cog array has the same number of cogs, SQ_b.

To get the average values of an average cog we take $\frac{[B_r, F_r, E_r]}{SQ_b} = \left[\frac{B_r}{SQ_b}, \frac{F_r}{SQ_b}, \frac{E_r}{SQ_b}\right]$

We can generally assume top level cogs for every cog, if the player is using lower level cogs, that's ok! This formula is just trying to get some way to bring B_r, F_r, and E_r into an approximately consistent range.

To calculate the average affix, multiply by $\frac{Sn}{Af}$: $\left[\frac{Sn*B_r}{Af*SQ_b}, \frac{Sn*F_r}{Af*SQ_b}, \frac{Sn*E_r}{Af*SQ_b}\right]$

Since we assume that $\frac{1}{b}$ of the effect comes from a bonus:

$\left[\frac{Sn*B_r}{Af*SQ_b*b}, \frac{Sn*F_r}{Af*SQ_b*b}, \frac{Sn*E_r}{Af*SQ_b*b}\right]$

Applying the inverse formulas and reapplying b to the final result:

$$\left[\frac{Sn*B_r}{Af*SQ_b}, b\left(\frac{Sn*F_r}{Af*SQ_b*b}\right)^{1.25}, b * 388.428W\left(0.145974 * \sqrt[25]{10^{\frac{Sn*E_r}{Af*SQ_b*b}}}\right)^{\frac{5}{2}}\right]$$

Using the more efficient approximation of the last formula

$$\left[\frac{Sn*B_r}{Af*SQ_b}, b\left(\frac{Sn*F_r}{Af*SQ_b*b}\right)^{1.25}, b\left(\left(0.0003808514 * \left(\frac{Sn*E_r}{Af*SQ_b*b}\right)^{3.968829}\right) + \left(\frac{Sn*E_r}{Af*SQ_b*b}\right)^{1.075056} + 0.8517081\right)\right]$$

These can now be multiplied by $[B_w, F_w, E_w]$ and the result can be summed.