Optimizing Single-Round, Single-Target Gag Choices Interactively

Dr. Jonathan Helianthicus Doe, IV

2018, March

1 Posing the problem

- $opt[n][hp][orgs]$ is the value of the largest-valued gag in the set of n gags which when combined deals at least hp damage and contains at most $orgs$ organic gags, such that the value of the largest-valued gag is minimized.
- G is the set of gags, including the "empty gag", i.e. passing.
- G' is the subset of G that is organic.
- $dmg(\cdot)$ takes in a gag as an argument and returns the net HP loss its use incurs (it's actually not this simple, but it works well enough to create the proper recurrence).

Constant for the whole problem: cog level, "is the cog v2.0?", "is the cog lured?", G, G'.

$$opt[n][hp][orgs] = \min_{gag \in G} \{\max\left(opt[n-1][hp - dmg(gag)][orgs - \mathbb{1}_{G'}(gag)],\ gag\right)\}.$$

2 Defining a well-order on the set G

In order to make sense of the "min" and "max" above we want to define a well-order on the set of gags G that captures the intuitive notion that higher level gags are worth more, that some gag types (squirt, sound, etc.) are worth more than others, and that organic gags are worth more than non-organic gags. Preferably, we care about those three aspects in exactly that order, which suggests that we can simply define a lexicographical order.

We could represent each gag, then, as a triplet (l, τ, o) where l is the level of the gag (e.g. for a squirting flower $l = 1$, for a foghorn $l = 6$, etc.), τ is the type of the gag (squirt, sound, etc.), and $o = 1$ when the gag is organic and $o = 0$ otherwise. The only gag not represented this way is "pass", which is clearly just the least element of G.

Then we would only need to choose a perference ordering for the possible gag types τ; for example we might use $\{\text{drop} \leq \text{trap} \leq \text{squirt} \leq \text{throw} \leq \text{sound}\}$. Now l, τ, and o have obvious orderings and the lexicographical ordering on the set of gags $(l, \tau, o) \in G$ emerges.

In reality, however, this is a somewhat *crude* ordering; it is very reasonable to assert foghorn > cake ($(6, \text{sound}, 0) > (6, \text{throw}, 0)$), but also assert cream pie > elephant trunk ($(5, \text{throw}, 0) > (5, \text{sound}, 0)$). There are other examples like this too, and because of that it is probably best to simply hand-pick an ordering (at worst having $5 \cdot 7 = 35$ elements, since we exclude toonup and lure).

3 The slightly mysterious $dmg(\cdot)$ function

The $dmg(\cdot)$ function used in the definition of opt in section 1 actually conceals an additional parameter that opt takes, which is a map M from gag types τ to indicators d that have 3 possibilities:

- This gag type has not been used at all yet.
- This gag type has been used once, for x damage.

- This gag type has been used more than once, for x damage.

This is sufficient to define $dmg(\cdot)$ and have it correctly calculate "yellow" and "orange" damage:

$$opt[n][hp][orgs][M] = \min_{gag \in G} \left\{ \max\left(opt[n-1][hp - dmg(gag)][orgs - \mathbb{1}_{G'}(gag)][update(M, gag)],\ gag \right) \right\}.$$

The problem as a whole is then $opt[n][hp][orgs][\varnothing]$.

4 Tiebreaking