Reordering snacks is effective and just

Dr. Tom Murphy VII Ph.D.*

1 April 2016

Concrete

It's time to get serious about *snacks* and *stats*.

Keywords: snacking, permutations, comestability theory

1 Introduction

Some workplaces of the future now offer a feature called Snacks. With Snacks, a kitchenette is provided near workers, which supplies running water and an array of small food packs. These foods are free of (dollar) charge, but various spoken and unspoken rules govern worker interactions with the foods.

This presents a challenge, since some foods are more desirable than others. Specifically, say that one yogurt food comes in four varieties: Classic, Diet, Cherry-Vanilla, and Caffeine Free. Furthermore, say that one user `<3`'s Cherry-Vanilla variety yogurt food and `>=3`'s Caffeine Free variety yogurt food. This worker is happiest when he begins his office work while eating Cherry-Vanilla. If there is no Cherry-Vanilla, the worker may resort to Classic yogurt food, reducing his task-ready disposition, and thus performance. If all flavors have been exhausted but Caffeine Free, then the worker may take no yogurt at all, knowing that Caffeine Free brings more displeasure than even hunger. This creates a negative affect, which may cause the worker to actively sabotage the work of his peers. This provides poor Return on Investment (ROI).

One strategy is for this worker, who we will call Sal, to take all of the favored Cherry-Vanilla foods from the kitchenette to his desk at the beginning of the day. This strategy is called Hoarding. This ensures that Sal may eat all his Cherry-Vanilla flavors. However, this behavior is considered unfair, for one reason that all other workers are completely deprived of Cherry-Vanilla flavor. It is perceived that Sal and all other workers should have equivalent access to the shared resource, except for the moment that he is selecting a food (for he is "first in line," and lines are fair). Moreover, Sal should take only one food at a time (for it is "Please help yourself. We ask that you take only one piece so that others may enjoy it as well," which is fair). We also perceive that Sal should take food only when he is actually hungry (for it is "waste not, want not," and you can't spell "aphorism" without "a fair is," mmm?).

Given these rules, there are still things that Sal can do to influence the chance that he gets the Cherry-Vanilla flavors. In particular, this paper investigates the strategy of Reordering, where Sal selects his favored snack, and also changes the order of the foods in the kitchenette. The thought is that while everyone retains equivalent access to the foods, other workers are less likely to select Cherry-Vanilla due to the decreased visibility and/or increased effort in finding them.

Note that the author does not reorder habitually reorder snacks; this question is of abstract philosophical interest. We consulted the wisdom of Judge John Hodgman, who ruled [1]:

> Why don't you just go out to lunch and buy the food you want with your own money?
> . . . Stop with the personal e-mails and get back to work.

We did not find this to be a satisfactory argument.

In this paper, I first provide a short argument why Sal's behavior may be considered fair, using an unjustifiable but common assumption. I then give a formal model for Snacks, which can be used to conduct controlled experiments. I then show that under suitable conditions, Sal's behavior benefits both him and the workplace, in a Utilitarian sense.

1.1 Why Reorder?

As suggested before, Reordering benefits Sal because his favorite snacks are less likely to be eaten by others. The policy can be thought of as fair because it still provides everyone equivalent access to the snacks. But did Sal make the snacks worse for everyone by putting the good snacks at the back? The problem seems intractible until we consider an unjustifiable but standard simplifying assumption: Each worker's preferences for snacks are independent and identically distributed. This means that Sal's preference for Cherry-Vanilla gives us no information about the rest of the workforce's preferences. Since all preferences are equally likely, reordering the snacks improves access to some snacks and hinders access to others; these effects cancel out. Thus under suitable assumptions, *any* reordering of snacks is fair and harmless to other workers. The reason why we believe that this particular policy is beneficial is that other workers may be indifferent among various flavors, and simply take the one that is most convenient. When indifferent, taking a variety that someone else substantially favors is globally sub-optimal. This is similar to the principle of "decoy beer", whereby cheap OK beer (e.g. Red Stripe) is placed at the front of the fridge in front of the premium good beer (which may be unsuitable for casual drinkers anyway).

We can go further and demonstrate this scientifically.

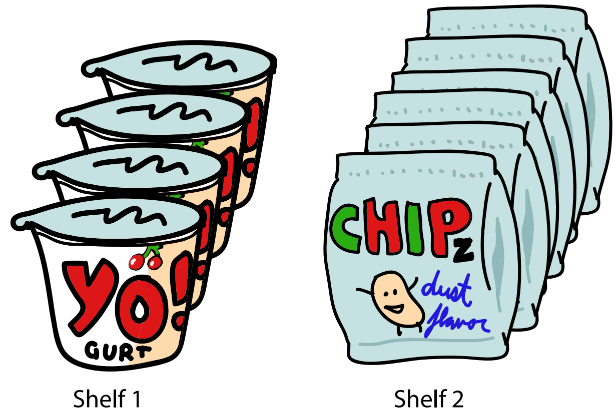

Figure 1: Two shelves with snacks. Shelf 1 has four snacks; potentially different types of Yö!gurt brand yogurt. Shelf 2 has six snacks, potentially different types CHIPz brand natural particle chips. I can tell you that the first three on Shelf 1 are Cherry flavor, and the last one is the Diet Cherry variety. All of the CHIPz are different varieties: Dust flavor, Dirt flavor, Sticks flavor, Cobweb flavor, Sand flavor, and Powder flavor. But because of the way snacks are arranged, you can only see the variety of the snack in front.

1.2 Game of Scones

To investigate whether Reordering helps Sal and the rest of the workplace, we could run an experiment. Unfortunately this would be very expensive; we would need to find many workforces that are comparable, with similar food preferences, randomly assign some to the experimental group (where some fraction implement the Reordering policy) and then somehow judge their happiness. This would take a long time, and if the policy or experimental controls turn out to be harmful, might impact real GDP. For the effect sizes we see later, a live experiment is unlikely to show significance.

Instead, we develop a simple model that captures important aspects of the Snacks program, implement this on a computer, and then run millions of simulations.

The simplified model is as follows.

A simulation consists of an array S of shelves, each of which is stocked with different varieties of food (Figure 1). The varieties are just given as integers from 0–V_i, where each food type (shelf) may have a different number of varieties. In an early version of the simulation each food and variety is given a name, so we might have a shelf consisting of many sodas, like[1]

- Diet Kake
- Diet Thuck Lite
- Mango Slooch
- Mr. Sleepe Black
- Caffeine–Free Droob Ultra
- Caffeine–Free Spask Black
- Dr. Drarb Classic
- Drorp Lite
- Strawberry Sad
- Vanilla Grerb
- Cherry Prote Lite
- Diet Grobe
- Dr. Brosh Lite
- Duq Ultra
- Grape Ding Lite
- Mrs. Broop

[1] All of these brands now Copyright Trademark Patent Pending Dr. Tom Murphy VII Ph.D., 2016!

- Diet Pap
- Grape Drax Classic

We also have an array W of workers. Each worker has a preference function P_{ij}, one floating point value for each snack variety. This value may be negative, indicating an aversion to that snack. Nominally, these values are in dollars, for scale.

For simplicity, workers all get hungry at the same rate (although their hunger strikes randomly). When a worker hungers, she

1. Selects a shelf at random.
2. Sets her gaze upon the foremost variety on that shelf. She has some value v for this item, given by P.
3. She can see the number of items on the shelf, but not what varieties they are. From this number, she estimates what value v' she would get from skipping the current variety (for this round) and setting her gaze upon the next one.
4. If $v' > v$, she does so, and repeats from step 3.
5. If not, this is the food provisionally selected for this shelf, with value v.
6. If there are unvisited shelves remaining, she estimates the value of abandoning this shelf and trying the next one, v''.
7. If $v'' > v$, she moves to the next shelf and returns to step 2.
8. If she finishes with a selected food, she may reorder the items on the shelf of that food arbitrarily. She removes the selected food, if any, and eats it.

No player retains any knowledge of the organization of a shelf between rounds.[2]

We have not yet said how the worker estimates the value of a shelf. But observe the following properties:

- If estimates are accurate, workers select a rational choice of food to maximize their own happiness.
- If a user has a dramatically favored snack, she is willing to search deep within a shelf for it.
- If a user has some snacks she favors and some she does not, she will be less willing to give up a good snack to find her favorite snack, because she might get stuck with a worse snack.

However, it also has an undesirable property:

- If a worker has a flat distribution of preferences, she will search the whole shelf. This is because there is no risk of getting stuck with a bad snack; she likes them all. This extends in a soft way to nearly flat distributions.

This does not match our intuitions of how real workers behave. Most of the time, an indifferent worker will just take a food that is "good enough;" this is known as "satisficing."[2] The argument for the global value of Sal reordering hinges on such indifference, in fact. To prevent this, a worker's estimate of the value of continuing to search the shelf will include a small cost to search each item. This can be thought of as the cost of the physical labor or the displaced opportunity cost, or an estimate of the risk that an interruption causes her to have to stop searching before she selects a snack.

The above requires an estimate of a shelf's value, both for the case where the worker may continue searching a shelf and the case where she continues to the next shelf. This can be computed with a recurrence relation. Since the worker cannot see beyond the snack her gaze is upon, this only depends on which shelf this is and the number of items on it. The expected value $E_s(n)$ for looking through n items on shelf number s is

$$E_s(0) = 0$$

$$E_s(n) = \sum_{i=1}^{V_s} \mathrm{P}(\text{item } i) \times \max(P_{si}, E_s(n-1) - c)$$

where V_s is the number varieties for shelf s, P(item i) is the probability of selecting variety i in the next slot, P_{si} is the worker's preference for variety i from shelf s, and c is the small cost of looking at all. The content of the recurrence is simple: At each step, for each possible item, the worker can either can take that item with the value given by the preference function P, or keep going (but now there will be one fewer item).

[2] This is not an accurate assumption in reality, but seems to only disadvantage those who reorder snacks from benefiting from their own treachery. A very advanced strategy might rearrange items on the shelf in order to encode information about what has been reordered, for example, by coding a specific unlikely pattern at the front of a shelf (or prior shelf) to foreshadow the hidden booty. Ultra-advanced strategies might place misleading codes to confuse other workers and cause them to make suboptimal choices. Hyper-advanced strategies might use steganographic techniques or cryptographic signatures to hide codes or make them tamper-proof. Of course, this does not matter in a real workplace because workers can remember extremely simple facts themselves.

Since we stipulate that the worker remembers nothing between rounds, the only probability distribution that makes sense for P(item i) is the uniform one, so the general case becomes

$$E_s(n) = \sum_{i=1}^{V_s} \frac{\max(P_{si}, E_s(n-1) - c)}{V_s}$$

Since this only depends on the preference function, we can compute this when the worker is born and print it on their birth certificate and employee badge.

1.3 Experimenting

I implemented the rules above in about 1000 lines of JavaScript, including a UI, random number generation, and the experimentation harness.[3]

Some other parameters need to be set: `NUM_SHELVES`, `NUM_PEOPLE` and `MAX_VARIETIES`, all self-explanatory; `MIN_ITEMS` and `MAX_ITEMS`, the bounds on the number of snacks initially stocked per shelf (it seems that `MIN` should be at least 2, for the program is Snacks, plural); `COST_TO_LOOK`, the penalty from the recurrence for estimating a shelf's value. We also have `PREF_MEAN` and `PREF_STDDEV` which are the parameters for the generation of preference functions (each is Gaussian—preferences are allowed to be negative). Finally, `MEAN_WAIT` gives the average amount of time between hunger events, wait times cannot be negative and so are distributed as $\Gamma(2.0, \texttt{MEAN\_WAIT})$. We run each simulation for three eight-hour work days.

I then generated random scenarios and evaluated different snack reordering policies. One extreme policy sorts the shelf in reverse preference order, so that the worker's favorite snacks are at the back. This seems unnecessary (takes $O(n \lg n)$ time, unless perhaps using "radix sort") and unrealistic. It also leads to substantial interference between workers, as one worker's resorting completely undoes another's. Another relaxation of this, more realistic, is where the worker only moves his favorite variety to the depths of the shelf. This is analogous to a "Move-to-Back List" [3], except not really, since the worker somehow finds all of the instances of their favorite variety in the whole shelf before moving them to the back. The most realistic policy generalizes this last one, and only moves the favorite snack when it passes some "outlier" threshold for how much it is preferred over the other snacks. This is the policy I ran many experiments for.

[3]It can be found at `sourceforge.net/p/tom7misc/svn/HEAD/tree/trunk/snacks/`.

The first thing to notice from the experiments is that the policy doesn't even work! OMFG! For the reasonable parameter settings I started with, not only is the overall happiness (value of snacks eaten) neutral to slightly negative when the policy is in effect, but the worker employing the reordering strategy *also eats neutral to worse snacks*! This doesn't concur with anyone's perspective on the problem (except maybe John Hodgman's)—at least the "selfish" worker who reorders snacks should be benefiting from it, right?

Normally, this is where a heroic scientist would (a) implicate the model, seeing as how it fails to have intuitive properties[4] or (b) appeal that "it seems more work is needed in this fertile area of study" or (c) give up. But owing to Draconian SIGBOVIK deadlines, this heroic scientist resorts to another unjustifiable but common technique: "Tweaking" the model parameters until the experiments turn out as expected (i.e., positive). Better yet, this process can be automated!

Search. Overnight in 4 separate browser tabs I ran a simple "hyper-parameter search" to try to find the best settings of the parameters described previously. I selected parameter values uniformly at random, and then interpolated against the current local maximum (by averaging the random point with the best point for each successive "heads" coin-flip during intialization). The goal was to find parameters that simultaneously improved two scores in the experiment: The value of the snacks eaten by the resorting worker, and the total value of snacks eaten across all workers. (Specifically, I maximized the minimum of these two). The best settings of the model had these values:

`COST_TO_LOOK`	0.001
`MAX_ITEMS`	37
`MAX_VARIETY`	4
`MEAN_WAIT`	36305.715
`MIN_ITEMS`	3
`MRATIO`	1.010
`NUM_PEOPLE`	2
`NUM_SHELVES`	15
`OUTLIER_RATIO`	2.199
`PREF_MEAN`	1.320
`PREF_STDDEV`	5.625

[4]I think that the model does not capture two important aspects. First is that Sal *does know* that he's reordered the snacks and can likely find his favorites by just looking immediately to the back. Second is that traces of the model suggest that most workers look through nearly all of the snacks due to the (rational) expectation that they will improve their selection. There does not seem to be a good setting of the `COST_TO_LOOK` penalty that suitably discourages workers from looking through the whole shelf while still allowing a resorter to find his deeply-placed favorites.

This an interesting instance: There are only two workers, but lots of shelves and lots of snacks. Most interestingly, the mean hunger time is very unusually large (in the highest 97th percentile of the probability distribution): 36,305 seconds is over 10 hours! This means that in the course of a three-day simulation, we only expect each player to eat about 2.5 times. There is a very good chance that the workers never interact (never visit the same shelf) and more than a $^1/_4$ chance that a worker never eats anything.

Repeating many more simulations with these parameter values suggests—but does not prove—that this may just be a nearly degenerate case in the simulation where the policies of the workers do not matter, and what we are seeing is pure noise. However, when I stopped the simulation arbitrarily after 1 million rounds, it produced nice smooth-looking distributions, consistent with a good sample size (Figure 2). The player implementing the resorting policy ate 0.015% better snacks on average, and the overall workplace ate 0.031% better snacks on average! This truly is a victory for snack reordering and the scientific method! $\square$

References

[1] J. Hodgman, "Judge John Hodgman on tech-firm snacks," *New York Times Magazine*, Jan 2016. [Online]. Available: nytimes.com/2016/01/24/magazine/judge-john-hodgman-on-tech-firm-snacks.html

[2] H. A. Simon, "Rational choice and the structure of the environment," *Psychological review*, vol. 63, no. 2, p. 129, 1956.

[3] R. Rivest, "On self-organizing sequential search heuristics," *Communications of the ACM*, vol. 19, no. 2, pp. 63–67, 1976.

[4] J. Kruschke, *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan.* Academic Press, 2014.

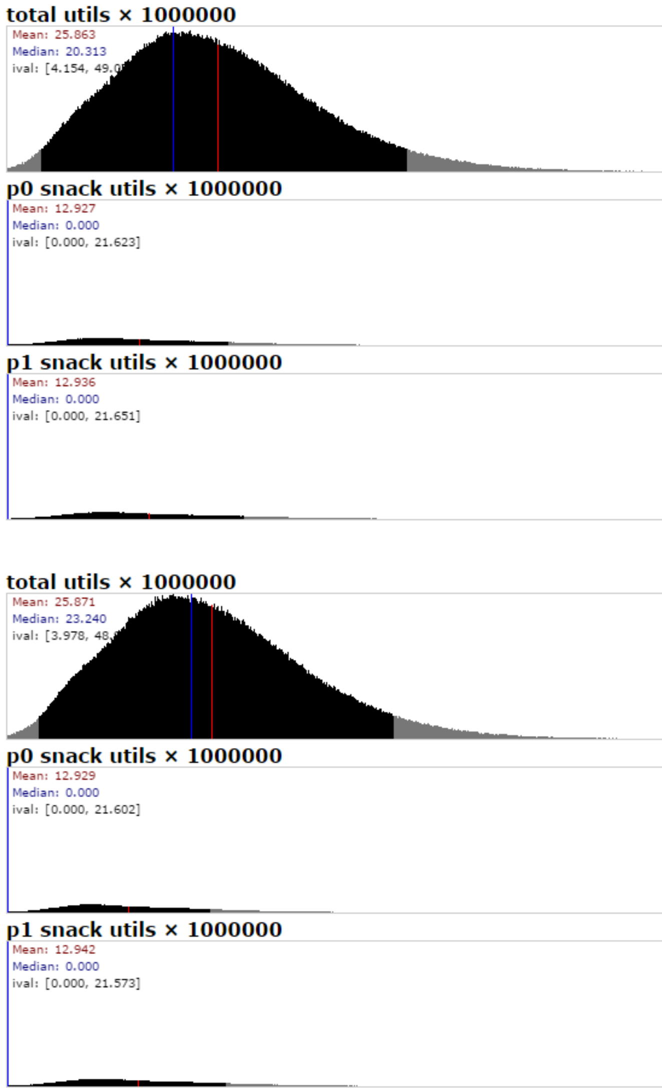

Figure 2: Histogram of one million experiments.

The top group is the control (no resorting) and the bottom is the experiment (worker 0 moves his favorite to the back of the shelf if its value exceeds the outlier ratio). Note how smooth the simulated outcomes are. This kind of graph is just the kind of statistics you write home about. Within each group, the big lump is the total snacks eaten by the two workers, and then the snacks eaten by each worker individually. Note a dramatic peak at the value 0 for the two workers, corresponding to *no snacks*; this happens because of the extreme hunger inteval of 10 hours. A overall value of zero is very unlikely because the simulation can only advance time (and thus end) when a worker has a chance to eat. Each histogram shows its mean and median, as well as darkening the 95% highest density interval [4]. An advanced technique would difference the values of the experiment and control and show a distribution of that statistic, but again, Draconian SIGBOVIK page size limitations preclude this. Normally we would use the 95% intervals to make the comparison between control and experiment, but this appears to be unreadable due to the very smooth, nice looking distribution occluding it. Therefore we compare the means, seeing 0.031%, 0.015%, and 0.046% improvement in snack consumption respectively for the three pairs. Note that resorting is actually *altruistic* in this experiment, helping most the worker who does not sort!

Without getting into too much math, regarding the frequentist standard of statistical significance typically used, we can say with confidence that $p > 0.05$.