Quan Nguyen
quan@wustl.edu

Research note
April 24, 2020

Summary:

- Section 1 discusses a number of problems encountered when simulations involving Beta distributions with small parameters are run.
- Section 2 covers the distinction between the leximin assignment on normalized costs and leximin on unnormalized costs.

1 More on the PoF with different Beta distributions

1.1 Beta distributions with small parameters

Samples drawn from Beta distributions where $\alpha, \beta << 1$ tend to either 0 and 1. This causes two specific problems:

- The last columns in a generated matrix might contain values that are interpreted as 1, and the frequency of this event increases along with the dimension of the matrix. A single experiment showed a 20-by-5 matrix contained 18 ones in its last column, while a 30-by-6 matrix contained all ones in its last column.

  This could cause the greedy leximin assignment to generate an incorrect solution since the policy uses the arbitrary order of the agents to break ties.

- The original leximin assignment (running Gurobi) could be used instead. The solver has adjustable tolerance parameters whose default values are `1e-4`. This value is too high for our purposes, and it is not obvious how a tolerance parameter should be set since the coefficients in our matrices can get increasingly small.

  An experiment showed that this solver can indeed fail to lexically minimize the largest costs, specifically when $Beta(0.01, 0.1)$ is used.

When $\alpha, \beta \geq 0.1$, these two problems are not yet found to occur in our simulations.

One potential solution is to refine the greedy leximin assignment. Specifically, when tie-breaking is necessary, the algorithm can move on to the next intervention and start assigning agents with minimum costs with respect to that intervention until the number of tying agents in the previous intervention does not exceed its remaining capacity.

1.2 PoF with Bernoulli costs

For an instance of the assignment problem where costs are drawn from a Bernoulli distribution, the resulting cost matrix only consists of values in the set $\{0, 1\}$. Here we notice that the leximin assignment actually corresponds to the efficient assignment, as both minimize

the number of agents with positive costs.

We can prove this claim by assuming the opposite. Firstly, as the efficient assignment minimizes the total cost across all agents, each of which in this case is either 0 or 1, the assignment minimizes the number of agents with cost 1 (which is equal to the total cost).

Now, assume that the leximin assignment is different from the efficient assignment, which means under the leximin assignment, the number of agents with cost 1 (denoted as k_L) is strictly greater than that under the efficient assignment (denoted as k_E, where $k_E < k_L$). This is a contradiction since the leximin assignment in this case does not minimize the cost of at least $k_L - k_E$ agents (whose costs could be 0 but are 1). In other words, the leximin assignment cannot be different (in terms of total cost) from the efficient assignment, and the PoF is thus always 1.

2 Normalized leximin vs. unnormalized leximin

There exists a distinction between computing the leximin assignment on the normalized costs (which is what we have been doing so far) and computing the leximin assignment on the original costs. Specifically, the former sequentially minimizes the largest increases from the lowest cost of a given agent, while the latter simply minimizes the largest costs of individual agents in order.

For an instance of the assignment problem with n agents, we denote $L_n(i)$ as the assignment of agent i under the leximin assignment on the normalized costs and $L_u(i)$ as that under the leximin assignment on the original, unnormalized costs, for all $i = 1, ..., n$.

2.1 Price of Fairness for L_u

We run the simulations that we had for L_n, this time for L_u. Specifically, we consider the mean PoF(L_u) when costs are generated from different Beta distributions.

Figure 1: Mean PoF(L_u)

We notice a slight difference between this heat map and the one for PoF(L_n): high values for PoF still correspond to small values of α and β, but this time a specific region where $\alpha > \beta$ has a higher mean PoF. Overall, PoF(L_u) is also higher than PoF(L_n).

2.2 Bounds for the difference in total cost

We are interested in the ratio of the total cost between the two assignments: $C(L_u)/C(L_n)$. In this section, we will prove that:

$$\frac{1}{n-1} \leq \frac{C(L_u)}{C(L_n)} \leq n$$

where n is the number of agents in the assignment problem. We can prove the first inequality:

$$\frac{C(L_u)}{C(L_n)} \leq n$$

by employing a similar argument for the upper-bound of PoF(L_n). Specifically, due to the definition of the assignment, the bottleneck cost under L_u is minimized and therefore is at most the bottleneck cost under L_n, which is in turn at most the total cost of the assignment, $C(L_n)$. Summing the actual costs under L_u for the n agents, we obtain the inequality above.

We show that $C(L_u)/C(L_n)$ can get arbitrarily close to n via the following sample matrix:

	1	**2**	**3**	**...**	**n - 1**	**n**
1	0	0	0	...	0	$1-\epsilon$
2	0	0	0	...	$1-\epsilon$	1
3	0	0	0	...	1	1
...	...	...	...	...	...	...
n - 1	0	$1-\epsilon$	1	...	1	1
n	$1-\epsilon$	1	1	...	1	1

Table 1: Matrix for large $C(L_u)$

Under L_u, each agent i is assigned to intervention $n-i+1$, and the whole assignment corresponds to the anti-diagonal of the matrix. Under L_n, agent n is assigned to intervention n, and every other agent i is assigned to intervention $n-i$ (making up the anti-diagonal moved up by one row). These two assignments result in the following total costs: $C(L_u) = (1-\epsilon)\, n$ and $C(L_n) = 1$, thus proving the tightness of the bound above.

The second inequality:

$$\frac{C(L_n)}{C(L_u)} \leq n-1$$

can be proven using the proof for the upper-bound for PoF(L_n) itself. To show the tightness of this bound, a more complicated sample matrix is needed:

	1	**2**	**3**	**...**	**n - 1**	**n**
1	0	$1-2\epsilon$	$1-2\epsilon$	...	$1-2\epsilon$	$1-\epsilon$
2	$n\epsilon$	$n\epsilon$	$n\epsilon$	...	$n\epsilon$	1
3	$(n-1)\epsilon$	$(n-1)\epsilon$	$(n-1)\epsilon$	...	1	1
...	...	...	...	...	...	...
n - 1	3ϵ	3ϵ	1	...	1	1
n	2ϵ	1	1	...	1	1

Table 2: Matrix for large $C(L_n)$

L_u again corresponds to the anti-diagonal assignment with the total cost of:

$$\begin{aligned} C(L_u) &= 1 - \epsilon + n\epsilon + (n-1)\epsilon + ... + 3\epsilon + 2\epsilon \\ &= 1 + \left(\frac{n^2 + n - 4}{2}\right)\epsilon \end{aligned}$$

The cost matrix above also has the following normalized values:

	1	**2**	**3**	**...**	**n - 1**	**n**
1	0	$1-2\epsilon$	$1-2\epsilon$	...	$1-2\epsilon$	$1-\epsilon$
2	0	0	0	...	0	$1-n\epsilon$
3	0	0	0	...	$1-(n-1)\epsilon$	$1-(n-1)\epsilon$
...	...	...	...	...	...	...
n - 1	0	0	$1-3\epsilon$	...	$1-3\epsilon$	$1-3\epsilon$
n	0	$1-2\epsilon$	$1-2\epsilon$	...	$1-2\epsilon$	$1-2\epsilon$

Table 3: Normalized matrix for large $C(L_n)$

Here L_n corresponds to the anti-diagonal moved down by one cell: agent 1 is assigned to intervention 1, agent i is assigned to intervention $n-i+2$, for all $i > 1$. Thus $C(L_n) = n-1$, and $C(L_n)/C(L_u)$ approaches $n-1$ as ϵ approaches 0 from the right.

2.3 Simulations

We generate the cost matrices by drawing from various Beta distributions, sort the rows, run the two versions of the leximin assignment in question and record the ratio $C(L_u)/C(L_n)$. The following heat map visualizes the average ratio across 500 simulations for each of the (α, β) parameter combination that defines the Beta distribution of costs. (The matrices are of the size 30-by-5 with interventions having equal capacities.)

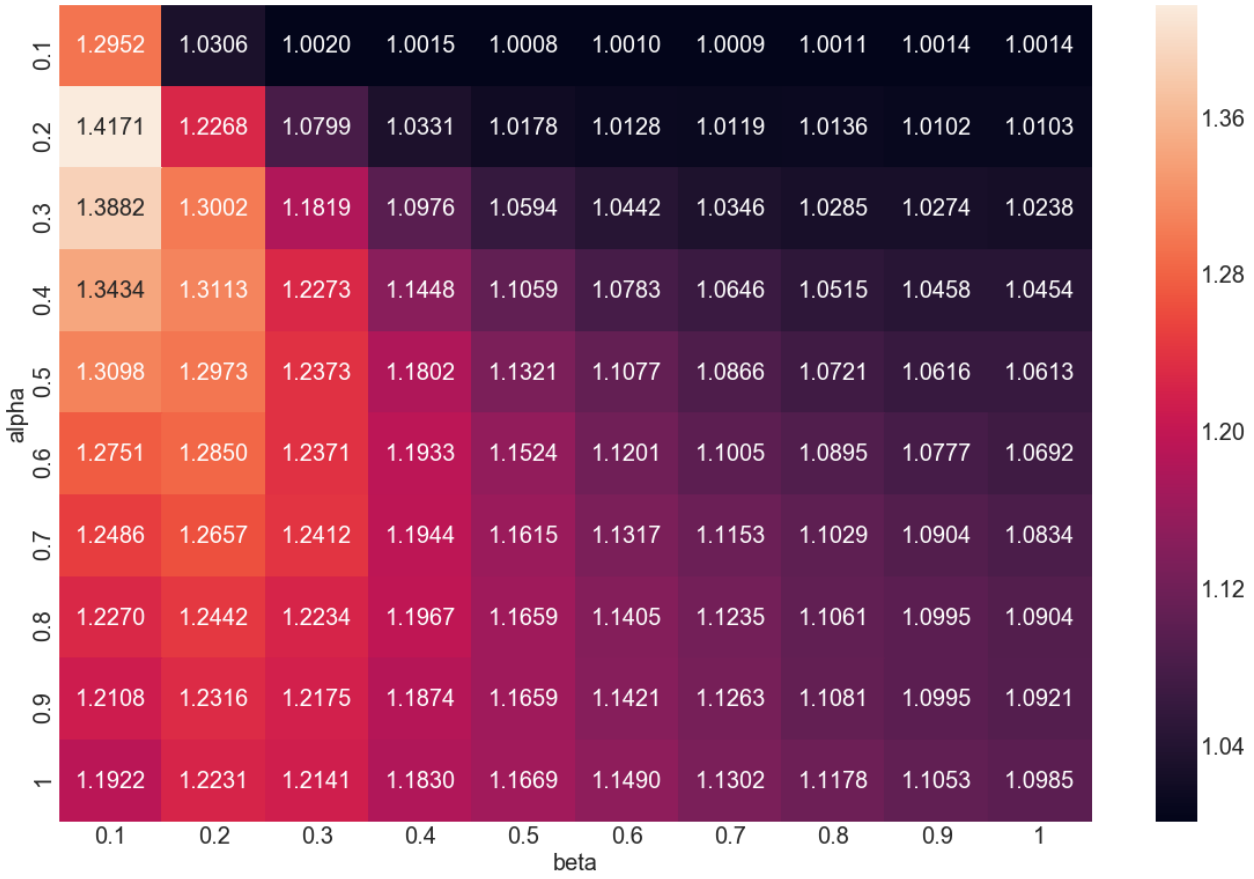

Figure 2: Ratio of leximin costs

We see that generally, the ratio does not fall below 1, indicating that L_n incurs a lower cost than L_u. More interestingly, the maximum of the ratio centers around the area where β is relatively small while $\alpha > \beta$.

2.4 Intuition

When the rows in the matrix are sorted, the greedy leximin assignment iterates through the interventions from the least effective to the most and fill out their respective capacities with the agents having lowest normalized (for L_n) or unnormalized (for L_u) costs.

Say given a fixed matrix, we follow the execution of the two assignments in parallel. The first discrepancy between the two assignments occurs in intervention j when agent i^*, who minimizes the normalized cost in column j, is different from agent i_*, who minimizes the

unnormalized cost:

$$i^* = \arg\min_i x_{ij} - x_{i1}$$
$$i_* = \arg\min_i x_{ij}$$
$$i^* \neq i_*$$

Due to their definitions, we thus have the following inequalities for the (un)normalized costs for intervention j and the lowest costs among the two agents:

$$\begin{aligned} x_{i^*j} &> x_{i_*j} && \text{(unnormalized costs)} \\ x_{i^*j} - x_{i^*1} &< x_{i_*j} - x_{i_*1} && \text{(normalized costs)} \\ x_{i^*1} &> x_{i_*1} && \text{(lowest costs)} \end{aligned}$$

The following visualizes the four numbers in two number lines:

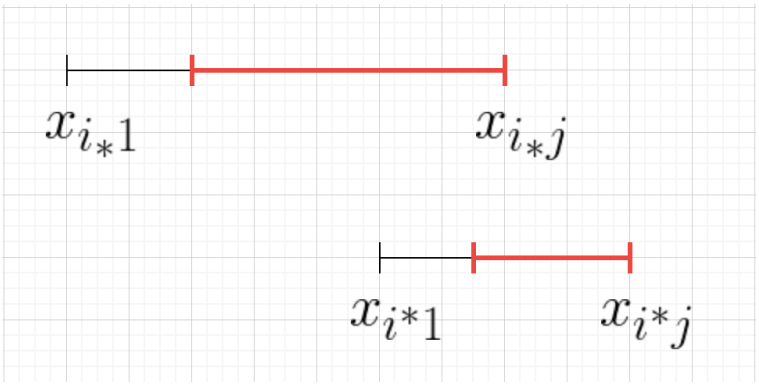

Figure 3: L_u vs. L_n

The endpoints of the two long segments respectively denote the four numbers: x_{i_*1}, x_{i_*j}, x_{i^*1}, and x_{i^*j}. Assume that the left endpoints of the two shorted red segments respectively denote the cost of the intervention that L_n assigns agent i_* to (which has the shortest distance from each agent's lowest cost in that intervention) and the cost of the intervention that L_u assigns agent i^* to (which is the lowest cost in that intervention).

We argue that since $x_{i^*j} - x_{i^*1} < x_{i_*j} - x_{i_*1}$ and the left endpoint of the first red segment is optimized to be close to x_{i_*1}, the first red segment is more likely to be longer than the second. This indicates that L_n benefits more (in terms of the total cost) with respect to the assignment of agent i_* than L_u does with respect to the assignment of agent i^*. This difference might accumulate over many assignments and result in a large decrease in total cost when we go from L_u to L_n.

Roughly speaking, as L_u lexically minimizes the large costs, many effective interventions will be assigned to agents with large average costs. L_n, on the other hand, only considers the distances from each agent's lowest cost, so if an agent has their lowest cost relatively higher than other agents' lowest costs, they are not guaranteed to be assigned to a good intervention.

This is illustrated by the last agent in Table 1, who starts out with a lowest cost of $1 - \epsilon$: L_u assigns this agent to the best intervention to ensure that no one is assigned a cost of 1, while L_n assigns this agent to the last intervention, since with respect to $1 - \epsilon$, that intervention (or any other intervention) only increases their cost by a small amount.

Additionally, we could explain the fact that bimodal Beta distributions with $\alpha > \beta$ are more likely to cause $C(L_u)$ to be significantly higher than $C(L_n)$: such a Beta distribution will result in some agents having high costs in the first column and thus cause the normalization to have a big effect on the cost matrix (this corresponds to agent i^* in our example), while at the same time, some other agents will have large normalized costs (corresponding to agent i_*). A balance of these two factors leads to the region close to but below the diagonal of the matrix of α and β that are less than 1.