The Three Woes of Bitcoin's Fee Market (and How BlockDAGs Can Fix Them) – Part III: BlockDAG Fee Markets

written by Parallel Thoughts (Shai Deshe) on Functor Network
original link: https://functor.network/user/912/entry/958

> This post was written as a cursory version of a future part in my open book, understanding blocckDAGs and GHOSTDAG

« Part II

In the previous posts we analyzed block*chain* fee markets and isolated some unsavory properties that manifest as their equilibria: race-to-the-bottom, starvation, and price aberration. We named these *the three woes*. In the current post, we do the parallel analysis for block*DAGs*. In the next post, we discuss how these differences affect the manifestation of the tree woes, and conclude the discussion.

The current post is by far the most mathematically involved in the series. To make it welcoming to a less technical crowd, I delegated the computational work to clearly separate sections and did my best to ensure the post is consistent and self-contained if they are skipped.

This post double-serves both as a layman exposition and for recording new observations that I am currently inspecting. In particular, I add interesting computations and results to this post as I find them. These extra-hard sections, marked with a double star, can rely on arbitrarily advanced math (though currently, it doesn't go beyond advanced undergrad), and are *not* written as "educational content". They are mostly here so I could easily share them with colleagues.

Pure Fee Markets With Multiple Leaders

We now need to extend the mining game we defined for blockchains to accommodate several miners. For simplicity, we assume that there are N blocks per round (in reality, the number of blocks per round fluctuates around an average that can change with the network conditions), where each block was created by a different miner, and that the miners do not cooperate in any way.

The offering rounds remain the same, but the payoff rounds change. Here, each miner indeed reports a choice of ℓ transactions. But if several miners include the same transaction, its fee will go to one of them uniformly randomly.

Remark: OK, it is not *exactly* uniformly randomly. What *actually* happens is that the *miners themselves* are *randomly shuffled*: they are numbered from 1 to N in some random way. If two miners include the same transaction, the

one with the *lower number* gets it. In other words, we don't resolve the conflict per transaction, but per block. If Alice and Bob both included the same two transactions, then either Alice or Bob will get the fee for *both* transactions. There is no scenario where Alice wins one fee, and Bob the other. However, if we consider only *one of the transactions*, it *will* go with probability half to either Alice or Bob (assuming there are no more miners). In formal terms: the probability that one transaction goes to Bob *depends* on the probabilities that other transactions go to Bob, but the *marginal distribution* of a *single* transaction is uniform across all miners who included it.

This dynamic kind of shakes the ground at the feet of the miners, which is a good thing because the entire point I'm trying to make is that Bitcoin miners are *just a bit too comfortable*. Note that the source of unease is *not* the randomness itself, miners, above all, are *very used* to randomness. What makes this setting truly interesting is that now the choices of one miner *affect* the profit of another miner. Bob's expected value *depends* on Alice's choices. It is the consequences of *this* phenomenon that we seek to understand.

Equilibria Analysis

Strictly speaking, we need to analyze the equilibria strategies for both users and miners. Like before, we can't do a precise analysis of the users, and while approximate models are possible, we will be satisfied with a qualitative discussion. This discussion will start very similarly to the case for block *chains*, but will have *wildly* different outcomes.

But analyzing the miner now becomes much more demanding, compared to the trivial equilibrium we had for blockchains (which was "just take the most profitable transactions").

At the payoff round, the state of the game is a list of m fees, which we will denote $F_1, \ldots, F_m$. Say that each miner only has to select one transaction (that is, $\ell = 1$). A *strategy* is specified by a list of *probabilities* $p_1, \ldots, p_m$, which we interpret as "include the transaction paying a fee of F_m with probability p_m". If ℓ is two, then we need to specify a probability for any *pair* of transactions, and so on. In fact, as long as ℓ is sufficiently smaller than m, the number of possible choices grows *exponentially* with ℓ (the exact number is given by a binomial coefficient).

To avoid this complexity, we will first assume that each miner only chooses *one* transaction. That is, that $\ell = 1$. Computing the exact answer for higher values of ℓ is quite difficult, but we will see how it can be simply approximated very well.

Another bane of our existence is that, since there are N miners, and the strategy of each miner has m probabilities, then there are in total $N \cdot m$ probabilities in their combined strategies. Maximizing for that many variables is still quite daunting.

Fortunately, we can leverage the symmetry of the problem. Since all miners have the exact same information and options available to them, we can assume that there exists a *symmetric* equilibrium, and in fact, it is the only stable equilibrium (though actually proving this requires some heavy machinery). This means that we only need to solve for m variables, describing a *single* strategy $(p_1, \ldots, p_m)$ that *all* miners use.

Even under these assumptions, solving the general case is still quite a headache. So instead, we will consider three much simpler yet very illuminating cases:

1. Two miners selecting one out of two transactions ($N = 2$, $\ell = 1$, $m = 2$)
2. Two miners selecting one transaction ($N = 2$, $\ell = 1$, general m)
3. Uniform general case (general n, ℓ, and m, but $F_1 = \ldots = F_m$)

First Case

This is the simplest case that is not trivial. It contains enough complexity to exhibit the consequences of multiple leaders quite nicely.

Recall that we consider two minutes that select one out of two transactions. Let us denote the fees they are paying x and y, and assume that $x \geq y$. We can describe a strategy by a single number p, which we understand as the probability of including x, whereby we include y with probability $1 - p$.

What is the best strategy?

Before jumping into the calculus, let us try some natural candidates and see how they fare.

Strategy I - Maximum: Select x. That is, $p = 1$.

This strategy ignores the existence of the other miner and follows the optimal strategy for the single miner case. What is its expected profit?

Recall that we consider the case where both miners apply the same strategy. Hence, both miners will select x, and the probability for each miner to gain the fee is $1/2$, resulting in an expected profit of $x/2$.

For $x = 100$ and $y = 1$, the expected profit is $x/2 = 50$. What about $x = 100$ and $y = 99$? Well, the expected profit doesn't really *care* about the actual value of y, as long as it is lower than x, so it remains 50.

Strategy II - Uniform: Select either x or y with the same probability. That is, $p = 1/2$.

The miners become aware of the fact that collisions harm their expected profit and try to avoid it by flipping a fair coin. What would be the expected profit here?

There are four cases, each happening with the same probability of $1/4$:

- Both miners selected x, expected profit is $x/2$

- Both miners selected y, expected profit is $y/2$
- I chose x and other miner chose y, expected profit is x
- I chose y and other miner chose x, expected profit is y

Since all events are equally probable, the expectation is just their simple average.

$$\frac{1}{4}\left(\frac{x}{2} + x + \frac{y}{2} + y\right) = \frac{3}{8}(x+y).$$

For $x = 100$ and $y = 1$ this comes out 37.875, which is clearly *worse* than the maximal strategy. But for $x = 100$ and $y = 99$, this comes out \$74.625%, which is *better*!

So if neither of these strategies is optimal, what is? We will work out the answer using some high-school-level pre-calculus: we will write the expected outcome as a function of p and derive it to find its maximum. I will defer the calculation itself to the next section (that you can skip if you don't feel like doing math).

The upshot of the calculation is the following:

Strategy III - Optimal: select x with probability

$$p = \frac{x}{x+y}.$$

The expected profit from this strategy is

$$\frac{1}{2}\left(x + y - \frac{xy}{x+y}\right).$$

Let's see how we did! First, for $x = 100$ and $y = 1$ we get a profit of 50.005 which is ever so slightly higher than the maximal strategy. For $x = 100$ and $y = 99$ we get 74.6257 which is again just above the random strategy. The impressive feat is that it is better than *both* at the same time.

In hindsight, it should not come as a surprise that the maximal strategy is almost optimal first case: if one transaction is 100 times more profitable than the second, then surely this offsets the expected profit lost by a chance of collision. It should also come as no surprise that the uniform strategy is almost optimal for the second: if the transactions pay about the same, then choosing one over the other doesn't meaningfully change the profit, so it is better to do whatever possible to reduce the chance of collisions.

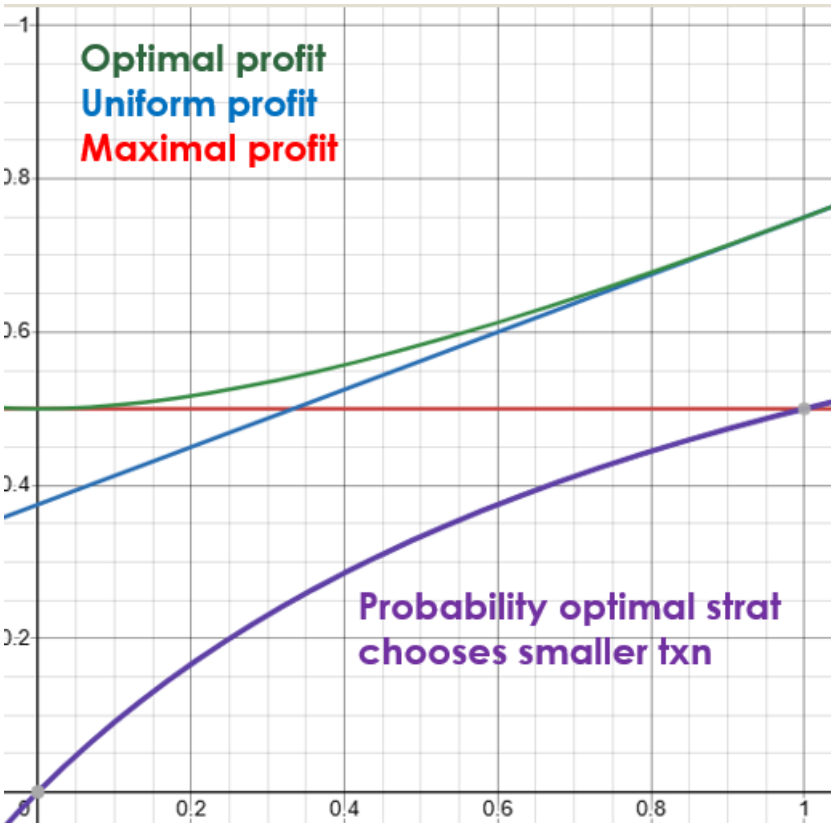
What happens around the middle, say at $x = 100$ and $y = 50$? Here the maximal strategy still expects 50, but now the uniform strategy nets 56.2, while the optimal strategy gives 58.3.

We finally note that the optimal strategy doesn't actually depend on the values x and y, just on the ratio between them. We can see this by rewriting the expression we got:

$$p = \frac{x}{x+y} = \frac{1}{1+y/x}.$$

The optimal strategy is the same whether $x = 100$ and $y = 50$, or whether $x = 1$ and $y = 0.5$. Multiplying x and y by the same number α *doesn't change* the optimal strategy (though the expected *profit* of this strategy does get multiplied by α).

For that reason, we can just assume $x = 1$, and that $y < 1$, and plot the expected profit from the optimal, uniform, and maximal strategies as a function of y:



I also plotted the probability y is chosen as a function of y.

We see that the optimal strategy kind of *morphs* from the maximal strategy around 0 to the uniform strategy around 1, which makes a lot of sense.

Remark: We can also compute that the shift in optimality between the uniform and maximal profit happens exactly when $x = 3y$. For example, if $x = 30$ and $y = 10$ we easily see that the maximal and uniform strategy both have an expected profit of 15. The optimal strategy still beats both with 16.25.

The Computation* One easy way to generalize the computations we did for the maximal and uniform strategies is to use a *probability table*. In such a table, each row represents a choice by one miner, and each column represents a choice by the second miner. The matching cell is the expected profit conditioned on these choices. The title of each row/column is the probability that the miner chose this option, so the probability of each cell is the *product* of the probabilities written in the title of its respective row and column. Sounds complicated? An example will make it much simpler.

For the maximum strategy, we have the following table:

	1	0
1	$x/2$	y
0	x	$y/2$

So we see that the expectation $x/2$ obtained in the top-left corner has probability $1 \cdot 1 = 1$, and the rest have probability 0.

When we switch to a different strategy, we do not change the values in the cells. The expected profit *given some particular selection by the miners* remains the same. The only thing that happens is the probability that this selection will actually occur.

For the uniform strategy, we get the following table:

	$1/2$	$1/2$
$1/2$	$x/2$	y
$1/2$	x	$y/2$

We see that here all cells have the same probability of $1/4$, making the total probability just the sum of all cells divided by four. In other words, their average value.

In the general case, we have a probability p to choose x, so we obtain this table:

	p	$1-p$
p	$x/2$	y
$1-p$	x	$y/2$

So now we follow the same procedure, we multiply each cell by the headers of its row and column, and sum the results:

$$p \cdot p \cdot \frac{x}{2} + p\,(1-p) \cdot y + (1-p) \cdot p \cdot x + (1-p) \cdot (1-p) \cdot \frac{y}{2}$$

and you can verify that this simplifies to

$$\frac{1}{2}\left(\left(2p-p^2\right)x+\left(1-p^2\right)y\right).$$

I will leave it to the reader to derive this expression with respect to p, compare it to zero, and solve for p. The final result is:

$$p=\frac{x}{x+y},$$

as expected.

Substituting this value of p into the equation we found for the expectation gives the expression we got for the optimal expectation, as the reader is invited to verify.

Second Case

Now we have two miners ($N=2$) selecting one ($\ell=1$) out of m transactions.

The solution, it turns out, is best expressed in terms of what's called a *harmonic average*. It seems a bit weird at first, but it has a nice motivation.

The standard way to average numbers is called the *arithmetic mean*, defined for a list of numbers $F_1,\ldots,F_n$ as

$$A(F_1,\ldots,F_n)=\frac{1}{n}\left(F_1+\ldots+F_n\right).$$

The harmonic mean is a different way to average numbers, that is more suitable to our computation. It is defined like this:

$$H(F_1,\ldots,F_n)=\frac{n}{\frac{1}{F_1}+\ldots+\frac{1}{F_1}}.$$

I wrote a beautiful segment about the harmonic mean and how you already know, but then I realized that it interrupted the entire flow of the post, so I deferred it to an optional section below. Suffice it to say, if we think of the everyday arithmetic mean as the correct way to average speeds, then the harmonic mean is the correct way to average *rates*.

As usual, I will defer the computation to a skippable section below.

The result is the following strategy:

$$p_i=1-\frac{m-1}{m}\frac{H\left(\overline{F}\right)}{F_i}.$$

Before we move on, we note that one can (quite easily) find for any $m > 2$ values $F_1, \ldots, F_m$ such that, according to the formula above, $p_i < 0$.

We call such instances *degeneracies*. We handle degeneracies in a later section, but for now, we will remain in the regime where no degeneracies happen, and this solution works.

The expected profit for each miner in this case is given by

$$\frac{m}{2}\left(A\left(F_1,\ldots,F_m\right) - \left(\frac{m-1}{m}\right)^2 H\left(F_1,\ldots,F_m\right)\right).$$

.

The Harmonic Mean and How You Already Know It The harmonic mean might seem a bit... alien, but you've actually used it before without even realizing it. When you were a kid, you were probably given an exercise like this: if one builder builds one wall in three days, and another builder builds one wall in two days, how long would it take both builders to build two walls together?

We can't just add the days up, that makes no sense. To correctly compute the answer, we realize that if a builder builds a wall in 3 days, then they build $\frac{1}{3}$ walls in a day. The second builder builds $\frac{1}{2}$ walls in a day, so together they build $\frac{1}{3} + \frac{1}{2}$ walls in a day. We translate this quantity back from "walls per day" to "days per wall" by taking the reciprocal again. This gives us how long it would take to build *one* wall, so we multiply it by two to get:

$$2 \cdot \frac{1}{\frac{1}{3} + \frac{1}{2}} = H(3,2).$$

The key insight is that the arithmetic mean is the correct way to average *speed*, while the harmonic mean is the correct way to average *rates*. To hit the nail on the head, you can convince yourself that

$$H\left(F_1,\ldots,F_n\right) = \frac{1}{A\left(\frac{1}{F_1},\ldots,\frac{1}{F_n}\right)}.$$

The Computation** Say that there are two miners selecting one of the transactions paying $F_1, \ldots, F_m$. Given a strategy $p_1, \ldots, p_m$. Say the first miner selected F_i, conditioned on that, their expected profit becomes $F_i((1-p_i) + p_i/2)$. Since they choose F_i with probability p_i, their expected profit *from this transaction alone* is obtained by multiplying throughout by p_i, getting $F_i\left(p_i(1-p_i) + \frac{1}{2}p_i^2\right)$.

To make things a bit simpler (and more readily generalizable) we can define the function

$$S_2\left(p\right) = \left(p\left(1-p\right) + \frac{1}{2}p^2\right)$$

and obtain that the expected profit from F_i alone is exactly $F_i \cdot S(p_i)$.

Hence, the total profit is given by the function:

$$f(p_1, \ldots, p_m) = \sum_{i=1}^{m} F_i \cdot S_2(p_i).$$

We want to maximize this function with respect to the constraint $g(p_1, \ldots, p_n) = p_1 + \ldots + p_m - 1 = 0$ (we will ignore the constraints $p_i \geq 0$ for now).

By using the method of Lagrange multipliers, we can find that there is some real λ such that for each i, j we get the equation

$$\frac{\partial f}{\partial p_i} = \frac{\partial f}{\partial p_j},$$

but since the only part of f that depends on p_i is the summand $F_i \cdot S_2(p_i)$ these conditions can be rewritten as

$$\frac{\partial f}{\partial p_i} = \frac{\partial f}{\partial p_j},$$

$$F_i \frac{d}{dp_i}\left(p_i - \frac{1}{2}p_i^2\right) = F_j \frac{d}{dp_j}\left(p_j - \frac{1}{2}p_j^2\right).$$

One can verify that after differentiating and isolating p_j we find the relation

$$p_j = 1 - \frac{F_i}{F_j}(1 - p_i)$$

which we plug into the constraint g to obtain

$$\begin{aligned}
1 &= \sum_{j=1}^{m} p_j \\
&= \sum_{j=1}^{m} \left(1 - \frac{F_i}{F_j}(1 - p_i)\right) \\
&= m - (1 - p_i) F_i \sum_{j=1}^{m} \frac{1}{F_j} \\
&= m - (1 - p_i) F_i m \frac{\sum_{j=1}^{m} \frac{1}{F_j}}{m} \\
&= m - (1 - p_i) \frac{F_i}{H(F_1, \ldots, F_n)} m
\end{aligned}$$

which after some rearrangement becomes

$$p_i = 1 - \frac{m-1}{m} \frac{H\left(F_1, \ldots, F_m\right)}{F_i}$$

as needed.

By plugging this optimal strategy into f we obtain the maximal expected profit.

Third Case

The general case is a bit involved but becomes much simpler if we assume all transactions pay the same fee F. In this case, the problem becomes much more symmetric and it is easy to conclude that the uniform strategy is the equilibrium (to prove that one can e.g. use the observation that an equilibrium strategy for transaction selection must be symmetric, and show that non-uniform strategies provide lower expected profit).

We can ask ourselves, if we add a new transaction, how high would it have to be to be non-degenerate? It is easiest to express the value of the transaction as $\alpha \cdot F$ and see how low α could be without hitting degeneracy. Say we have m transactions of value F and we add a single transaction of value αF. Then for the transaction to have positive inclusion probability it must satisfy

$$\alpha F \geq \frac{m}{m+1} H\left(\overbrace{F, \ldots, F}^{m \text{ times}}, \alpha F\right).$$

I will leave the computation to you, but it comes out $\alpha \geq \frac{m}{m+1}$.

This doesn't look very good, does it? Even if there is a decently small queue of 99 transactions, we get that to be included one must have $\alpha = 99\%$.

Well, we should not be disappointed by seeing only a slight improvement, because two leaders are only slightly better than one. What happens when there are many?

I didn't explicitly write down the degeneracy formula for n miners selecting one out of m transactions, but there is one. If we plug m transactions of value F and a single transaction of value αF into this formula, we get that the αF transaction has positive probability if

$$\alpha \geq \left(\frac{m}{m+1}\right)^{n-1}.$$

This is already much better! It might not seem like much, since there are typically a lot more transactions (which could be in the thousands) compared with leaders per block (which are, at best, in the low dozens). However, the thing is that this expression decreases exponentially with n while being only linearly close to 1.

For example, if we consider again the case of $m = 100$ and now increase the number of leaders from 2 to 10 we see that α significantly reduced from 99% to... 91.4%. Wait... that's still not very good either, is it? Of course, because we are only limiting ourselves to one transaction per block! In fact, if $n = 10$ and $m = 100$ we are essentially considering a situation where the network is congested ten times over!

Now, increasing ℓ is where things get really complicated, even for two miners and uniform transactions. What causes the complication is the dependence between events. Since a miner can include both the second and seventh transactions, we have to take into account all possible overlaps with all other miners which creates a terrible mess. However, one can prove that even through that mess if all transactions pay a fee of F, a transaction with fee αF has positive inclusion chances if

$$\alpha \geq \left(\frac{m}{m+1}\right)^{\ell(n-1)}.$$

This is looking much better. We can now compute the degeneracy not as a function of the absolute number of transactions, but their fraction of the total capacity. That is, we consider $k \cdot \ell$ uniform transactions, If $k = n$ we get that the network is exactly in full capacity. If $k = n/2$ we get that we are at half capacity, etc.

One can use the formula above to prove that if $m = k \cdot \ell$, the bound on α is approximately $e^{-n/k}$. What does this mean?

If the network is exactly fully congested, that is $k = n$, we get a threshold of about $1/e \approx 37\%$ which is already a noticeable improvement, but that's just the worst case! If the network is only half congested, that is, $k = n/2$, this becomes $1/e^2 \approx 13.5\%$! Even when the network is at twice the full capacity we get that the threshold $1/\sqrt{e} \approx 60\%$. Heck, even at ten times the throughput, we get the bound $1/\sqrt[10]{e} \approx 90\%$ Compare this to Bitcoin, where once congestion hits, we are at a round 0%.

General Case For Selecting One Transaction**

For completeness, I will provide the computation for a general number of miners sampling one out of a general number of transactions m paying arbitrary fees $F_1, \ldots, F_m$.

For technical reasons, it will actually be more comfortable to talk about $N + 1$ miners. This will allow us to assume the point of view of one miner, and write their profit in terms of what the other N miners do.

The computation starts similarly. We ask ourselves, for the strategy $(p_1, \ldots, p_m)$, what is the expected profit from F_i specifically for a particular miner.

First of all, the probability that the miner even selects F_i is p_i. If none of the other miners selected F_i, which happens with probability $(1 - p_i)^N$, then the

profit is F_i. If exactly one miner also chose p_i, then the expected profit is $F_i/2$. For each *given* miner, the probability that this miner selected F_i while the rest selected something else is $p_i(1-p_i)^n$. Since there are N ways to choose one of the N miners, we get that the total probability of the event "exactly one other miner selected F_i" is $N \cdot (1-p_i)^{N-1}$. We can extend this logic and say that the probability that exactly j different miner selected F_i is given by $\binom{N}{j} p_i^j (1-p_i)^{N-j}$ and in this case the expected profit is $\frac{F_i}{j+1}$.

We can encode all this into the function

$$S_N(p) = p \sum_{j=0}^{N} \frac{1}{j+1} \binom{N}{j} p^j (1-p)^{N-j}$$

and get that the expected profit for the strategy $p_1, \ldots, p_m$ is given by

$$f(p_1, \ldots, p_m) = \sum_{i=1}^{m} F_i \cdot S_N(p_i).$$

If we try to proceed in the computation as is, we will run into notational inferno. Instead, we first try to find a nicer expression for S_N. This is a bit delicate:

$$\begin{aligned}
S'_N(p) &= \sum_{j=0}^{N} \frac{1}{j+1} \binom{N}{j} \frac{d}{dx} p^{j+1} (1-p)^{N-j} \\
&= \sum_{j=0}^{N} \frac{1}{j+1} \binom{N}{j} \left((j+1) p^j (1-p)^{N-j} - (N-j) p^{j+1} (1-p)^{N-j-1} \right) \\
&= \sum_{j=0}^{N} \frac{1}{j+1} \binom{N}{j} \left((j+1) p^j (1-p)^{N-j} + (j+1) p (1-p)^{N-j-1} - (N+1) p^{j+1} (1-p)^{N-j-1} \right) \\
&= \sum_{j=0}^{N} \frac{1}{j+1} \binom{N}{j} \left((j+1) p^j (1-p)^{N-j} + \frac{p}{1-p} (j+1) p^j (1-p)^{N-j} - \frac{1}{1-p} (N+1) p^{j+1} (1-p)^{N-j} \right) \\
&= \sum_{j=0}^{N} \frac{1}{j+1} \binom{N}{j} \left(\frac{1}{1-p} (j+1) p^j (1-p)^{N-j} - \frac{1}{1-p} (N+1) p^{j+1} (1-p)^{n-j} \right) \\
&= \frac{1}{1-p} \sum_{j=0}^{N} \binom{N}{j} p^j (1-p)^{N-j} - \frac{1}{1-p} (N+1) \sum_{j=0}^{N} \frac{1}{j+1} \binom{N}{j} p^{j+1} (1-p)^{n-j} \\
&= \frac{1}{1-p} - \frac{N+1}{1-p} S'_N(x)
\end{aligned}$$

The upshot is that $S_N(x)$ is the unique solution to the separable differential equation

$$y' = \frac{1}{1-x} + \frac{n+1}{1-x} y,$$

with respect to the boundary condition $y(0) = 0$ (where uniqueness follows trivially by Picard's theorem).

Solving this equation is direct, so I will leave it to the reader, but it follows that the solution corresponding to these boundary conditions is

$$S_N(p) = \frac{1-(1-x)^{N+1}}{N+1}.$$

Hence, the function we want to maximize is:

$$f(p_1, \ldots, p_m) = \sum_{i=1}^{m} F_i S_N(p_i),$$

and we do it again by using Lagrange multipliers.

I will spare you the details, but tell you the result. First, we introduce the *power mean*:

$$M_\alpha(F_1, \ldots, F_m) = \left(\frac{1}{m}\sum x_i^\alpha\right)^{1/\alpha}.$$

(You are welcome to check that M_1 and M_{-1} are the arithmetic and harmonic means respectively.)

By doing a computation very similar to the one we did for two miners, we find that the optimal strategy is given by

$$p_i = 1 - \sqrt[N-1]{\frac{M_{-1/(N-1)}(F_1, \ldots, F_m)}{F_i}}\frac{m-1}{m},$$

but to make this formula just a bit easier on the eyes, we return to the assumption that the *total* number of miners is N and obtain the solution

$$p_i = 1 - \sqrt[N]{\frac{M_{-1/N}(F_1, \ldots, F_m)}{F_i}}\frac{m-1}{m}$$

. The expected profit is then given by

$$\frac{m}{N+1}\left(A(F_1, \ldots, F_m) - \left(1-\frac{1}{m}\right)^{N+1} M_{-1/N}(F_1, \ldots, F_m)\right),$$

and the non-degeneracy condition is

$$F_i \geq \left(1-\frac{1}{m}\right)^N M_{-1/N}(F_1, \ldots, F_m).$$

Degeneracy Reexamined**

Consider a situation where there are $m-1$ transactions with a fee 1 and an mth transaction with a fee F. Clearly if $F=1$ then there are no degeneracies, how low/high can F be without creating degeneracies? We know that no degeneracies exist for the case $m=2$ so we will assume there are at least three transactions in total.

One can easily compute that in this case, we have

$$M_{-1/N}\left(F_1,\ldots,F_m\right)=\left(\frac{1}{m}\left((m-1)+\frac{1}{\sqrt[N]{F}}\right)\right)^{-N}$$

so the non-degeneracy conditions become

$$\begin{aligned}F_i &\geq \left(1-\frac{1}{m}\right)^N M_{-1/N}\left(F_1,\ldots,F_m\right)\\ &=\left(1-\frac{1}{m}\right)^N\left(\frac{1}{m}\left((m-1)+\frac{1}{\sqrt[N]{F}}\right)\right)^{-N}\\ &=F\left(\frac{m-1}{\sqrt[N]{F}\,(m-1)+1}\right)^N.\end{aligned}$$

Note that we only need to verify this holds for $F_i=1$ and $F_i=F$, as we assumed all fees are one of these two. One can check that the non-degeneracy condition we get by substituting $F_i=1$ *always* holds. This tells us that F can be as high as we want without creating a degeneracy. This is already surprising: if there are at least two transactions, and they are all paying the same fee, then adding one new transaction, no matter how astronomic it fee might be, cannot make the other transactions degenerate. There is always motivation to leave *some* probability for the lower paying transactions. This is not true if we add more than one transaction though. Actually, it is not hard to show that if there are $m_F>1$ transactions of value F, we obtain the condition

$$\left(1+\frac{1}{m_F-1}\right)^N\geq F.$$

The non-degeneracy condition we get for $F_i=F$ is easily shown to be

$$F\geq\left(1-\frac{1}{m-1}\right)^N.$$

Again, we shift our perspective a bit to assume that there are m transactions with fee 1, to which an additional, $(m+1)$th transaction with fee F was added, to obtain that the condition for the newly added transaction to be non-degenerate is simply

$$F\geq\left(1-\frac{1}{m}\right)^N.$$

This is already much better than the *single* leader (single transaction case), where if $F < 1$ it instantly becomes degenerate, whereas if $F > 1$ then it instantly renders all other transactions degenerate. However, the lower bound seems to approach 1 pretty fast. If

$$m \ll N$$

, then we get that F is only allowed to be smaller than the average by a tiny amount.

But here's the thing: in the $\ell = 1$ case, assuming $m \ll N$ is the same as assuming that the network is extremely congested. Obviously, if there are a million times more transactions paying a fee of 1 than the network can contain, there is no reason to acknowledge any transaction paying a fee of 0.99. To get a true grasp of how degeneracy plays out, we need to increase the throughput.

The accurate way to do that is to accurately solve the problem for miners with larger blocks, which turns out *extremely difficult*. However, what if we instead just increase the number of miners? More accurately, how inaccurate would it be to consider a single miner selecting ℓ transactions to ℓ miners selecting one transaction?

Recall that in our situation, all transactions but one pay the same fee, so we can say that up to a very small fix (at least for reasonably large values of m), the next transaction is selected uniformly. It is well known that if I sample uniformly from m different options, then it would take approximately $\sqrt{m}$ samples before I sample the same thing twice, and below that threshold, this probability drops *sharply*. In other words, if

$$\ell \ll m$$

, then uniformly sampling ℓ different transactions (which is what ℓ miners selecting one transaction each do) and uniformly sampling ℓ different transactions *without repetition* (which is what a single miner selecting ℓ transaction does) is practically the same, up to a very small correction, whose magnitude is dominated by the probability that two of the ℓ miners selected the same transaction.

A second observation is that the bound stays correct even for smaller values of m, it just becomes *loose*. Setting F above the bound we are about to derive will *still* assure non-degeneracy. It is just that we could have gone *even lower* and still not become degenerate. Finding the precise non-degeneracy bound for F when m is not much larger than ℓ (say, at most twice as large) seems very difficult, and I assume this needs to be tackled numerically.

The bound is obviously given by

$$F \geq \left(1 - \frac{1}{m}\right)^{\ell \cdot N},$$

which is still not terribly illuminating. It would benefit us to understand the bound in terms of the *congestion factor* $x = \frac{m}{\ell \cdot N}$. Substituting this into the bound, we get the condition:

$$
\begin{aligned}
F &\geq \left(1 - \frac{1}{m}\right)^{\ell \cdot N} \\
&= \left(1 - \frac{1}{\ell \cdot N \cdot x}\right)^{\ell \cdot N} \\
&= \left(\left(1 - \frac{1}{\ell \cdot N \cdot x}\right)^{\ell \cdot N \cdot x}\right)^{1/x} \\
&\approx e^{-1/x}
\end{aligned}
$$

I stress again that this is an *upper bound*. It gets very accurate very fast and is practically the correct answer for $x > 2$, but it overshoots the required fee for smaller values of x. Nevertheless, by inspecting the graph of the function $e^{1/x}$ we find that even if the demand for the network is at *double* its capacity, and it is filled with transactions that pay a fee of one dollar, you could get away with a fee of 70 cents. At *ten times its* capacity, you could still get away with paying less than 90 cents. When the network is *exactly at capacity*, this bound becomes loose, yet it *still* tells us that we would need to pay at most 35 cents. For a less-than-saturated network, this bound becomes increasingly over-estimating, and other methods (probably numeric) are required.

Part IV »

> If you find this content educational, interesting, or useful, please consider supporting my work.