CS269I: Incentives in Computer Science
Lecture #10: Incentives in Crowdsourcing*

Tim Roughgarden†

October 26, 2016

1 Bitcoin with Large Transaction Fees

We continue with more incentive issues in Bitcoin mining, which will segue naturally into a discussion of crowdsourcing.

1.1 Flat Reward vs. Transaction Fees

Recall that Bitcoin mining refers to the activity of finding valid blocks. A block contains a bunch of transactions and the hash of (effectively, a pointer to) the previous block. A block is valid if it hashes to a number close to 0 (at least ℓ leading zeroes, where ℓ, currently around 70, is tuned to keep the average rate of valid block creation around 10 minutes). The belief is that the only way to find valid blocks is by exhaustively searching through all possible nonces (every block includes a nonce).

Recall the reward for finding a new valid block and adding it to the blockchain:

1. A flat reward that does not depend on the contents of the block (other than it being valid). When Bitcoin debuted this reward was 50 BTC, but the protocol dictates that this amount gets cut in half every four years. It was cut to 25 BTC in 2012, and to 12.5 BTC just this past summer. At current exchange rates, this amounts to an 8K or 9K USD reward per block.

2. The sum of the transaction fees of the transactions in the block. Currently, transaction fees are non-zero but typically constitute only a few percent of the overall reward.

Looking toward the future, we know that the flat reward will keep dropping (all the way to 0), and it's natural to expect that transaction fees will continue to rise (to keep miners motivated to authorize blocks). New issues arise once transaction fees constitute the lion's share of block rewards (see also [2]).

†Department of Computer Science, Stanford University, 474 Gates Building, 353 Serra Mall, Stanford, CA 94305. Email: tim@cs.stanford.edu.

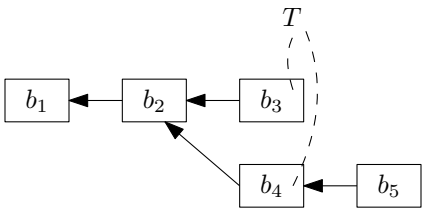

Figure 1: A miner can attempt to capture the fee T by extending b_2, including T in its new block, and extending the fork to beyond the original chain.

1.2 Fee Sniping

For a first example of how incentives can go awry with large transaction fees, suppose there is a transaction T with an unusually large transaction fee, say bigger than the typical sum of transaction fees in a block. Suppose some miner successfully authorizes this transaction in the block b_3, which gets appended to the blockchain (Figure 1). Suppose you're some other miner. If you mine honestly, then you will try to extend the block b_3 (with a block that cannot contain T). Why might you not want to do this? It might be more lucrative to try to extend the previous block b_2 with a block b_4 containing T, and then another block b_5 extending b_4.[1] If you succeed in appending both b_4 and b_5 before any other miner extends b_3, then b_3 is orphaned, b_4 and b_5 get authorized, and you collect T's large transaction fee.[2] The success probability of this attack is α^2, where α is the mining power of the miner (i.e., the fraction of the overall computational cycles owned by the miner).

If many miners are attempting the fee sniping attack for the same transaction, then the outcome is quite undesirable: very few miners would be doing any useful work in extending the last block of the blockchain, resulting in low throughput and high latencies, if not outright anarchy.

1.3 Selfish Tie-Breaking and Undercutting

Suppose there is a fork in the blockchain, with blocks b_4 and b_5 both extending b_3 (Figure 2). Recall the intended behavior specified by the Bitcoin protocol: a user should regard the

[1] So this is another attack involving deliberate forking, as in the double-spend attack and the 51% attack from last lecture.

[2] Note that if other miners tie-break between blocks according to which one was heard about earlier, as is intended, then appending b_4 is insufficient to authorize T (since b_3 was announced earlier).

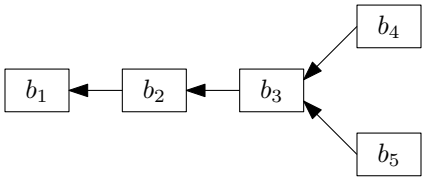

Figure 2: A fork in the block chain.

longest branch as the valid one, breaking ties according to the block that it heard about first. When there is a fixed reward per block, then it's all the same to a miner, and there is no reason not to follow this prescribed behavior. What's different when there are significant transaction fees?

Given the choice of two blocks to extend, with different amounts of transaction fees, a miner may be incentivized to extend the block with less transaction fees in it (regardless of whether or not it heard about this block first).[3] Whichever block has less transaction fees leaves more fees to be claimed by the next miner who extends the block. An extreme case would be where one block authorizes all of the currently outstanding transactions (leaving no rewards for other miners, until new transactions arrive), while another block authorizes only half of them (say).

Suppose you know that other miners are tie-breaking selfishly, as above. Then there is an incentive to *undercut* other miners. For example, if the current blockchain is $b_1 \leftarrow b_2 \leftarrow b_3$ and b_3 has a total of x transaction fees, and not many high-value transactions are left over, then a miner has an incentive to extend b_2 with a block b_4 that has only $x - \epsilon$ transaction fees, in the hope that other miners will extend b_4 in lieu of b_3. But then of course, other miners have an incentive to undercut further, appending blocks to b_2 with ever-decreasing transaction fees. This can viewed as a form of unraveling (recall Lecture #2), and in the worst case would lead to miners producing only blocks with very few transactions, reducing throughput and increasing latencies.

Do you think such undercutting issues will actually arise in the future? (We can't really test this now, since transaction fees are relatively small.) If so, what would you do to mitigate the problem?

1.4 Transaction Withholding

How do miners know about the outstanding transactions in the first place? Recall that when a new Bitcoin transaction is submitted, the intended behavior is to broadcast the transaction to all users via a peer-to-peer network. Conceptually, imagine a breadth-first search from the originating node to the rest of the nodes in the network. Nodes are supposed to forward announcements of new transactions to their neighbors, but are they incentivized to do so?

The issue is that spreading news about a transaction increases the amount of competition for authorizing it. Consider the extreme case where you're the only one who knows about a transaction that comes with a very lucrative fee. It's tempting to just keep the transaction a secret, and work on it privately. For example, a miner with mining power $\alpha = .01$ could expect to authorize the transaction in less than a day. (This is not so good for the senders and receivers in the transaction, since authorization would require tens of hours rather than tens of minutes.) This issue does not arise with the flat reward—if your chance of finding a block and your reward for a block are both independent of who knows about which transactions, then there's no reason not to forward transactions to everyone else.

[3] Last lecture we saw two genres of deviations from the Bitcoin protocol: deliberately creating a fork, and block withholding. Dishonest tie-breaking is a third type of deviation.

2 Crowdsourcing and the DARPA Network Challenge

The incentive for Bitcoin transaction withholding (Section 1.4) stems from the competing goals of Bitcoin to get tasks accomplished (i.e., validate blocks) and to recruit workers to complete tasks (i.e., miners). This tension is not specific to Bitcoin and is common in *crowdsourcing* systems, where the goal is to recruit a team of workers to complete one or more tasks.

Crowdsourcing competitions were thrust into prime time in 2009, with the DARPA Network Challenge (a.k.a. the "Red Balloons" contest). Here's how the challenge worked: at a prescribed time, DARPA dropped 10 red weather balloons at public locations around the U.S.[4] The first group to identify the locations of all 10 balloons received 40K USD in prize money.

There were somewhere between 50 and 100 teams in all. The winning team, from MIT, was also the team that thought most seriously about incentives, and in particular the need to incentivize both task completion *and* recruitment [3]. The team found the 10 balloons in a little under 9 hours.[5]

The main idea is familiar from "finder's fees." Monetary rewards were distributed on a per-balloon basis. For a given balloon, a 2K reward was given to the finder of the balloon, 1K to the recruiter of the balloon-finder, 500 dollars to the recruiter's recruiter, and so on.[6] The user pool can be viewed as a tree, according to who recruited whom (Figure 3). Note that the geometrically decreasing rewards ensure that the team does not pay out more than the 40K of prize money. (All rewards to users were explicitly contingent on the team actually winning the challenge.) The remaining money was donated to charity. Note that, in principle, a similar scheme could be used in Bitcoin to incentivize the broadcasting of transactions through the peer-to-peer network (provided one can keep track of who was the first to send the transaction to whom).

So if you're a user with the MIT team, is it in your interest to recruit more users, or are you better off keeping the competition to yourself? The answer depends somewhat on the details of the situation. In the extreme case, if you're the only one who knows about the task and have a 100% chance of completing it, then you don't ever want to recruit more users (they might complete the task before you). But suppose the probability of any given user completing a task is quite small (as with locating a red balloon), and that the probability that the task is completed by a user or one of her descendants is proportional to the size of her recruitment subtree. Then, it's better in expectation to recruit than to not recruit (do you see why?).

[4] One of them was actually in Union Square, San Francisco, but most were in more obscure locations like Katy, TX.

[5] The second-place team, from Georgia Tech, did not explicitly incentivize users to recruit other users. They started their recruitment efforts several weeks before the MIT team, but wound up with a significantly smaller number of contributors. Strategies for other teams varied; for example, one team managed to locate 6 balloons just by monitoring social media sites like Twitter [4].

[6] Anyone could register (with an email address) at the team's Web site. Upon registering, the new user would be given a link that they could share via email, Twitter, etc. This link was personalized, so anybody who registered through this link was considered to be recruited by the corresponding user.

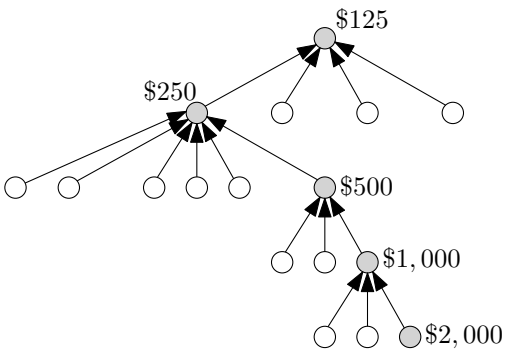

Figure 3: The recruitment graph is a tree, and the ancestors of the finder of a balloon get rewarded based on their distance from the finder.

3 Malicious Attacks in Competitive Crowdsourcing

One issue with competitive crowdsourcing contests like the DARPA Network Challenge, where there are multiple teams competing for the same prize, is incorrect and malicious submissions. This is not the focus of this lecture, but for completeness, here's a short summary of what happened.

The winning MIT team had to deal with many incorrect balloon-sighting submissions, including plenty of Photoshopped balloon pictures. Presumably some of the incorrect submissions were honest mistakes, some were greedy people hoping to get lucky, and some were malicious submissions by other teams in the challenge. The MIT team was able to filter the false submissions using relatively straightforward ideas—waiting for multiple independent sightings of a balloon before confirming it, comparing the geographic location of the IP address of the submitter against the location where the alleged balloon was sighted, etc. One nice aspect of the competition structure was that the false submissions did not have any effect on how many correct submissions were being submitted (since an honest participant doesn't even know about the false submissions).

Malicious attacks were a bigger issue in the 2011 DARPA Shredder Challenge. Here, each team was given 5 documents that had been cross-shredded, and the winner was the first to reconstruct all of them. (Like in the movie *Argo*.) A team from UCSD approached the challenge using crowdsourcing—they designed a game where players could experiment with moving the shreds of paper around, for all other players to see, and with different players building on each other's progress. Even though the UCSD team only found out about the competition two weeks after it started, within days they had catapulted to the top 3 on the leaderboard (out of roughly 100 teams). Subsequently, the UCSD system experienced multiple waves of "attacks," in the form of a user (apparently a competitor in the contest) undoing the progress made by previous users and basically making a mess of the puzzle. The UCSD team tracked every move made in the puzzle, so it was easy enough to roll back the malicious attacks. But the damage had been done: morale among the puzzle's users dropped, and eventually the UCSD team had to introduce a reputation system and let only the most proven users contribute moves. This reduced the number of contributors and hence

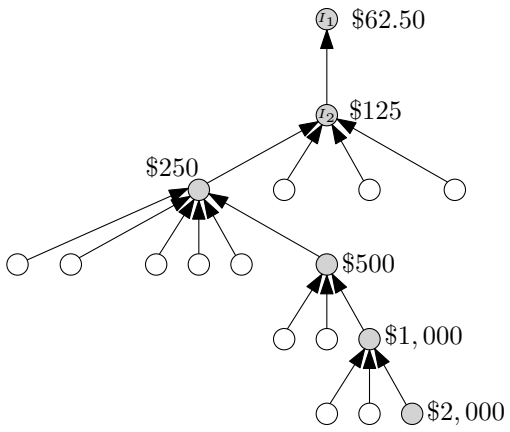

Figure 4: By creating an additional false identity I_2, and recruiting from that identity, the root node now collects the rewards from I_1 and I_2, increasing her payout by 50%.

the rate of progress. Ultimately, the UCSD team placed second; it's reasonable to speculate that they would have won the contest, had their been no malicious attacks on their system.[7]

It's not clear that these types of attacks are relevant for Bitcoin—fake transactions are easy to detect, and do not interfere with the processing of valid transactions.

4 Sybil Attacks

4.1 The Attack

Now let's drill down on incentive issues within a single team, rather than between different teams. For example, suppose that the MIT team was the only one in the challenge. We already argued that it's better to recruit than not. But are there any more sophisticated deviations from the intended behavior of "join and recruit everyone you know?"

The issue is *Sybil attacks*. As discussed last time, a Sybil attack is when a user of a system manipulates it by creating multiple identities. Whenever creating identities is cheap (like public keys in Bitcoin, or email addresses for MIT's system in the DARPA Network Challenge), Sybil attacks are a possibility. We mentioned last time that in Bitcoin, in the current regime of negligible transaction fees, Sybil attacks are not a problem. (Since influence over the system is determined by your computational power, not by how many identities you have.) What about in the recursive incentive scheme?

Here's an easy way to boost your reward under the MIT team's recursive incentive scheme. Create two identities, I_1 and I_2. Use I_1 to recruit I_2. Use I_2 to recruit everybody that you would have recruited originally. The result is that you get the same reward R that you would have gotten anyways (through I_2), plus an extra $R/2$ reward (through I_1). See Figure 4. Thus creating a second identity is guaranteed to boost your reward by 50%. And why stop

[7]The winning team had a totally different approach, based on computer vision techniques.

at 2 identities? By creating many identities and a recruitment chain through them, you boost your reward by (arbitrarily close to) 100%, effectively stealing all of the money that would otherwise be going to your ancestors and to charity.[8] Such attacks were not actually observed in the DARPA Network Challenge, but this might be partially due to the short time frame (4 days total).

4.2 Preferring Short Chains

One way to modify the recursive incentive scheme to be more robust to Sybil attacks is to reward only short chains [1]. Since the attack above with multiple identities creates long chains, there is intuition that this should help.

More precisely, the new reward scheme will be parameterized by a depth cut-off D, which is the maximum-length chain that is eligible for a reward. Bigger values of D mean that more reward will be paid out, but also that more people will get recruited.

Suppose a user x completes a task, and that the user's chain of ancestors (her recruiter, her recruiter's recruiter, etc.) has d users (not counting x herself). If $d > D$, then no rewards are given. Otherwise, a reward of 1 is given to everyone in the chain except for x, who gets a reward of $(D+1)-d$ (which is at least 1). In this case, the payout is $D+1$ (independent of d). Note that if a user is already at depth D in the recruitment tree, then there is no point in recruiting further.

We mention only some high-level intuition about why (and when) this reward scheme will work well; see [1] for the precise statements and arguments. As in Section 2, imagine that the probability of any given user completing the task is small, and that the probability that a user or one of her descendants completes the task is proportional to the size of her recruitment subtree. Suppose also that every user is capable of recruiting at least two other users, and so the size of a recruitment tree grows exponentially with its depth. Suppose some user creates a second identity (who recruits the first, as above). The benefit to the user is that, when she receives a reward, this reward is now 1 larger than it otherwise would have been. On the other hand, adding the second identity effectively deletes the leaves from her original recruitment subtree (Figure 5). Since every user has at least two children, this removes more than half of the users from the recruitment tree. Thus the second identity decreases the probability of obtaining a reward by at least 50%. Since the second identity at most doubles the reward received, the user would have been better off without it.

5 The Wisdom of the Crowd

Our last topic is slightly different but still quite related to crowdsourcing. You've surely heard the phrase "the wisdom of the crowd" bandied about. What does it mean? And is the crowd actually so "wise?"

[8] Replacing the "50%" rule in the recursive incentive scheme with a smaller percentage reduces the benefit of Sybil attacks (though the benefit remains non-zero), but also reduces the incentive to recruit additional users.

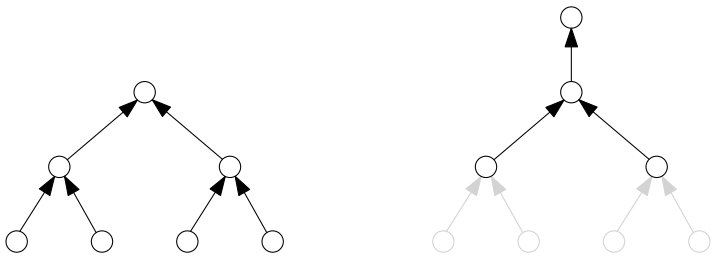

Figure 5: If the root node creates a second identity, the last layer in her recruitment tree (here layer 3) cannot be recruited anymore because it would not lead to a short chain anymore.

5.1 The Independent Case

The winning strategy in the DARPA Network Challenge illustrates that crowds can be powerful (even if not "wise," they at least have tremendous search power). But the original intuition for why a crowd can be "wiser" than any of its individuals is much more basic.

Suppose we want to estimate an unknown parameter μ, such as the "true quality" of a restaurant. Suppose each player i is an unbiased estimator of μ, in the sense that she corresponds to a random variable X_i with $\mathbf{E}[X_i] = \mu$. For example, X_i could represent player i's actual experience at the restaurant (e.g., as measured by the number of stars in her Yelp review). Also, and this is crucial, let's assume that the X_i's are independent random variables.

A natural way to aggregate the opinions of the crowd—the random variables $X_1, \ldots, X_n$—is via the average $\frac{1}{n}\sum_{i=1}^{n} X_i$. By the linearity of expectation, the average is again an unbiased estimator of μ:

$$\mathbf{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}[X_i] = \frac{1}{n}\cdot n \cdot \mu = \mu.$$

The variance, meanwhile, satisfies

$$\mathbf{Var}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] = \frac{1}{n^2}\sum_{i=1}^{n}\mathbf{Var}[X_i]\,,$$

where we are using that the X_i's are independent. (Variances do not generally add for non-independent random variables.) So for example, if the X_i's all have the same variance V (e.g., because they are i.i.d.), then the variance of the average is only V/n, far lower than that of any individual's estimate. As n grows large, the average of the crowd's estimate is overwhelmingly likely to be very close to the "ground truth" μ.[9]

[9]Does this mean that for restaurants with lots of Yelp reviews, the average number of stars is an excellent predictor of the restaurant's quality?

5.2 Herding and Information Cascades

Have you ever gone to a new city and seen two adjacent and similar-seeming restaurants, with one totally full and the other totally empty? What's up with that? Does the empty restaurant truly suck, or could there be another explanation?

Or how about: does rigid conformity require social pressure, or could it arise through purely rational decision-making?

Here's a thought experiment—stylized, but designed to clearly highlight a phenomenon that appears also in many other models and in the real world. I have two urns. (And in a technical course like this, urns are only ever good for holding balls.) The "red urn" has two red balls and one blue ball. The "blue urn" has two blue balls and one red ball. I flip a fair coin and use it to pick one of the two urns at random. (If you prefer, we're adopting the uniform prior over the two urns.) Whichever urn I picked, it will be fixed for the remainder of the experiment.

Players now arrive in a sequence. When player i arrives, she privately observes one randomly chosen ball from the urn. She then announces to everybody her guess ("red urn" or "blue urn"). Note that at the time of making a guess, the player knows the announcements of the first $i-1$ players (but not their observations) and her own observation. At the end of the experiment, a player receives a reward of 1 if she guessed correctly, and 0 otherwise.

Assume that all players are rational (i.e., want to maximize expected reward), and that all players assume that all other players are also rational. What happens?

Player #1. The first player has nothing to go on but her own observed ball. The intuitively obvious thing to do is to guess whatever color was observed. Formally, this can be justified using a simple application of Bayes' rule.[10]

Player #2. The second player hears the guess of the first player and also has her own independent observation. Assuming that the first player is rational, she announced whatever ball she drew. Thus the second player effectively has at her disposal two i.i.d. samples from the urn. Clearly, she should guess "red" if both samples are red and "blue" if both samples are blue. What if there is one of each color? By symmetry, there is no reason to favor one guess over the other. For simplicity, in this case, let's assume that the player breaks the tie by announcing the color of the ball she observed. Thus, just like the first player, in all cases the second player announces the color of the ball that she observed.

Player #3. The third player hears the guesses of the first two players and also has her own independent observation. Assuming that then previous players behave as above, they announced the color of the ball drawn. Thus, the third player has access to three i.i.d. samples from the urn. The intuitively obvious thing to do is to guess whichever color is the majority (again, formally justified using Bayes' rule). A key point is: if the first two players

[10] We are using the assumption that each of the two urns was originally equally likely. For example, if there was a 99% chance at the beginning of picking the red urn, then the first player should guess "red" even if she observed a blue ball.

draw the same color (and announce accordingly), then *the third player announces the same color, independent of the color of her ball.*

Player #4 and beyond. Suppose the draws (and hence announcements) of the first two players are the same color. Then the fourth player knows that the third player will say the same color, independent of the color of her ball. So just like the third player, in this case the fourth player has 3 i.i.d. samples from the urn (the first two announcements and her own observation) and no other information. By the same reasoning, when the first two balls have the same color, the fourth player will also guess this color, independent of the color of her ball. This "follow the herd" behavior will continue with all future players.

Note that the probability of drawing the blue ball twice from the red urn is $\frac{1}{9}$. Thus with more than 10% probability, all players guess incorrectly about which urn it is! This remains true no matter how many players there are, and despite the fact that on average two-thirds of these players will draw a ball of the opposite (i.e., correct) color.

This stylied thought experiment serves as a cautionary tale—when information cascades are present, the crowd may not be so wise. Our anecdotes suggest that, in group decision-making, it's important that individuals come up with their own independent opinions before coming together to try to reach consensus.

References

[1] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar. On bitcoin and red balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pages 56–73, 2012.

[2] M. Carlsten, H. A. Kalodner, S. M. Weinberg, and A. Narayanan. On the instability of Bitcoin without the block reward. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 154–167, 2016.

[3] G. Pickard, W. Pan, I. Rahwan, M. Cebrian, R. Crane, A. Madan, and A. Pentland. Time-critical social mobilization. *Science*, 334:509–512, 2011.

[4] J. C. Tang, M. Cebrian, N. A. Goacobe, H.-W. Kim, T. Kim, and D. Wickert. Reflecting on the DARPA Red Balloon Challenge. *Communications of the ACM*, 54(4):78–85, 2011.