NUS Summer Research 2026

Momin Ahmed

August 7, 2026

Contents

1 First Problem Statemen - PoS Sharding

Consider n nodes - each node i has "weight" (stake) w_i. There are malicious/adversary/black nodes with fraction of total stake at most f (e.g. 0.25) - we are not aware of them.
We wish to partition (or assign to committees) the nodes such that each committee has at most $0 \leq f < g < 1$ (e.g. 0.5) fraction of malicious stake. Meanwhile, we also want to minimize the maximum number of nodes in any committee.

2 Idealistic Setting : Known Adversaries

Assume we already know the color of all the nodes - i.e. we know black (malicious) and white (honest) nodes. Then, the problem is still not easy.

To see this, first recall the "**Multi-bag Min-knapsack**" problem: Given n bags, each bag i with threshold t_i, we want to fill the bag with items of weight w_i such that the total weight in bag i is at least t_i, and we want to minimize the total cost. Not only is this problem NP-hard, but it is also hard to determine if a "valid" solution exists (i.e. if we can cross the threshold for each bag).
Similar problems include bin covering problem, load balancing problem and so on.

2.1 NP-Hardness

Using this, we can determine if a "valid" solution exists - which is np-hard.

2.2 Approximation

Modify the Multi-bag Min-knapsack problem such that we want to minimize the maximum cost of any bag, instead of the total cost.
Using this as black box, we cah iteratively apply this algorithm to find our solution.

1. For each malicious node, create a new bag with threshild $\frac{w_i}{g} - w_i$.
2. Introduce "virtual" honest nodes that are extremely "long" (i.e. heavy) and extremely expensive - this is so that valid solution exists.
3. Apply the above algorithm. If no virtual nodes are used, we are done.
4. Otherwise, remove the virtual nodes. Now each "excess" bag becomes an honest "node" (with weight = excess amount, and cost = total cost of nodes in that bag) and each deficient bag becomes a malicous "node" (i.e. a new bag for the next iteration).
5. Repeat until no virtual nodes are used.

Note that in the iterative process, the number of bags must reduce by atleast 1, so the algorithm terminates.

2.3 Greedy Algorithm

We also present an algorithm for which the analysis and correctness is not clear yet.

1. Assign the "best" block to the "biggest" hole.
2. Repeat until all nodes are assigned.

Note that "best" block means it is the largest block that can either fit in the hole or just exceed it. Also note that after each assignment, we readjust the holes before the next iteration - otherwise there is a simple counter example.

3 Naive-Pessimistic Idea

We use 2 observations for this idea. Firstly, we can get a bounded approximation of the number of nodes in each committee. Secondly, we can also attain a bound on the total stake in each committee.
Thus, the idea is as follows:

1. Assign nodes uniformly randomly to k committees
2. Check if there exists a "valid" combination of $\frac{|M|}{k}(1+\epsilon)$ nodes that holds stake more than $\frac{1}{k}(1-\delta)$. If so, reject and repeat.

Here M is the set of malicious nodes and "valid" simply means we "ignore" nodes that have weight more than f since they must be honest.
For the algorithm to be efficient (i.e. not reject the assignment too many times), we must make the following assumption:

k-Regularity : For any valid $\frac{|M|}{k}(1+\epsilon)$ nodes which contains at least an honest node, the total stake is at most $\frac{1}{2k}(1-\delta)$.

Intuitively, this says that there is not too much concentration of wealth.
Since we do not know M, we can get an upperbound on $|M|$ by summing items from lightest to heaviest until we get f stake.
Issue: We can analyze the probability of failure and it's effeciency - both work out fine if the distribution is not too biased. But this assumption may be too strong. Essentially, we are giving the adversary too much power by considering the worst case in every committee.
Idea: We can split the nodes to either achieve k-regularlity, or come close to it.

4 Distinct Classes

We can form distinct classes of nodes to simply the problem.
Let ϵ be the smallest non negative weight. Then we have the classification

$$C_1 = [\epsilon, 2\epsilon) \quad , \quad C_2 = [2\epsilon, 4\epsilon) \quad , \ldots , \quad C_i = [2^{i-1}\epsilon, 2^i\epsilon)$$

The algorithms works as follows:

1. For each class, assign nodes randomly to k committees such that each committee has exactly $\frac{|C_i|}{k}$ nodes from class C_i.
2. Merge randomly.

4.1 Restrictive Case

Let $f = \frac{1}{4}, g = \frac{1}{2} - \delta$ (δ is choosing carefuly enough that you must either reach 1/2 or cross it). There are 2 classes and we want to form k committees. Assume there are $2k$ large nodes, and many (say pk) small nodes - the total stake held by each class is $\frac{1}{2}$. Assume that all heavy nodes have equal weight and all light nodes have equal weight ($w_H = \frac{1}{4k}, w_L = \frac{1}{2pk}$).

4.1.1 Optimal Adversary Strategy

Claim:
Let E denote the event that some committee contains corrupted weight greater than or equal to 1/2. Then there exists an optimal corruption strategy maximizing $\Pr(E)$ that corrupts only heavy nodes.
Proof:
First, note that it is optimal for the adversary to fully utilize the budget. Thus, any optimal strategy will corrupt H heavy nodes and L light nodes such that

$$w_H H + w_L L = f = \frac{1}{4} \quad \Rightarrow \quad L = \frac{1}{4w_L} - \frac{w_H}{w_L} H$$

Consider any strategy τ that corrupts H heavy nodes and $L > 0$ light nodes. Construct the strategy τ' that corrupts $H+1$ heavy nodes and $L - \frac{w_H}{w_L}$ light nodes. We wish to show that $\Pr(E)|_{\tau'} \geq \Pr(E)|_{\tau}$. Fix a committee C. Then, under either strategy, we have

$$\begin{aligned}\Pr(E) &= \Pr(2 \text{ heavy malicious nodes}) + \Pr(1 \text{ heavy malicious node and at least hallight malicious nodes}) \\ &\quad + \Pr(\text{ No heavy malicious nodes and all light malicious nodes})\end{aligned}$$

Under strategy τ, we have

$$\Pr(E)|_{\tau} = \frac{\binom{H}{2}}{\binom{2k}{2}} + \frac{\binom{H}{1}\binom{2k-H}{1}}{\binom{2k}{2}} \cdot \frac{\binom{L}{\frac{pk}{4}}}{\binom{pk}{\frac{pk}{4}}} + \frac{\binom{2k-H}{2}}{\binom{2k}{2}} \cdot \frac{\binom{L}{\frac{pk}{2}}}{\binom{pk}{\frac{pk}{2}}}$$

Under strategy τ', we have

$$\Pr(E)|_{\tau'} = \frac{\binom{H+1}{2}}{\binom{2k}{2}} + \frac{\binom{H+1}{1}\binom{2k-H-1}{1}}{\binom{2k}{2}} \cdot \frac{\binom{L-\frac{w_H}{w_L}}{\frac{pk}{4}}}{\binom{pk}{\frac{pk}{4}}} + \frac{\binom{2k-H-1}{2}}{\binom{2k}{2}} \cdot \frac{\binom{L-\frac{w_H}{w_L}}{\frac{pk}{2}}}{\binom{pk}{\frac{pk}{2}}}$$

4.1.2 Analysis

With the adversary corrupting k heavy nodes, we have

$$\Pr(E) = 1 - \frac{2^k}{\binom{2k}{k}} \approx 1 - \frac{\sqrt{\pi k}}{2^k}$$

We can compare this with the uniformly random assignment strategy. Under the corruption of only heavy nodes, we have

$$\Pr(E) = 1 - \frac{k!}{k^k} \approx 1 - \frac{\sqrt{2\pi k}}{e^k}$$

Here is a table for reference :

k	$1 - \frac{2^k}{\binom{2k}{k}}$	$1 - \frac{\sqrt{\pi k}}{2^k}$	$1 - \frac{k!}{k^k}$	$1 - \frac{\sqrt{2\pi k}}{e^k}$
1	0.000000000	0.113773075	0.000000000	0.077862991
2	0.333333333	0.373342931	0.500000000	0.520248912
3	0.600000000	0.616252485	0.777777778	0.783844089
4	0.771428571	0.778443269	0.906250000	0.908179003
5	0.873015873	0.876146022	0.961600000	0.962233866
6	0.930735931	0.932162382	0.984567901	0.984780560
7	0.962703963	0.963363499	0.993880101	0.993952476
8	0.980108779	0.980416185	0.997596741	0.997621632
9	0.989469354	0.989614527	0.999063343	0.999071972
10	0.994457555	0.994526377	0.999637120	0.999640130

Table 1: Comparison of exact probabilities and their Stirling approximations for $k = 1, \ldots, 10$.

Claim : For all $k \geq 1$, we have $1 - \frac{2^k}{\binom{2k}{k}} < 1 - \frac{k!}{k^k}$.

Proof : We wish to show that

$$\frac{2^k}{\binom{2k}{k}} > \frac{k!}{k^k}, \implies \frac{2^k k! k^k}{(2k)!} > 1.$$

Observe that

$$(2k)! = \prod_{i=1}^{k} (2i-1)(2i),$$

so, we obtain

$$\frac{2^k k! k^k}{(2k)!} = \frac{2^k k! k^k}{\prod_{i=1}^{k}(2i-1)(2i)} = \frac{k^k}{\prod_{i=1}^{k}(2i-1)}.$$

Therefore it remains to show that

$$k^k > \prod_{i=1}^{k}(2i-1).$$

Consider the k positive numbers

$$1, 3, 5, \ldots, 2k-1.$$

Their arithmetic mean is

$$\frac{1+3+\cdots+(2k-1)}{k}.$$

Since the sum of the first k odd numbers is k^2, the arithmetic mean equals

$$\frac{k^2}{k} = k.$$

By the arithmetic–geometric mean inequality,

$$\left(\prod_{i=1}^{k}(2i-1)\right)^{1/k} \leq k.$$

Raising both sides to the kth power yields

$$\prod_{i=1}^{k}(2i-1) \leq k^k.$$

Moreover, equality in AM–GM holds if and only if all the numbers are equal. Since

$$1, 3, 5, \ldots, 2k-1$$

are not all equal when $k > 1$, the inequality is strict:

$$\prod_{i=1}^{k}(2i-1) < k^k.$$

Consequently,

$$\frac{2^k k! k^k}{(2k)!} = \frac{k^k}{\prod_{i=1}^{k}(2i-1)} > 1.$$

Hence

$$\frac{2^k}{\binom{2k}{k}} > \frac{k!}{k^k},$$

which implies

$$1 - \frac{2^k}{\binom{2k}{k}} < 1 - \frac{k!}{k^k}.$$

This completes the proof. $\square$

4.1.3 Alg > Uniform Random Assignment

If α (the strategy of corrupting only heavy nodes) is optimal in both cases, the above claim shows Alg > Uniform Random Assignment.

5 Variance Guided Appraoch

Inherently, the problem is about deviating from the mean. We wish to design an algorithm such that there is great concerntration around the mean - while the adversary wishes to choose the strategy that maximizes the deviation from the mean.
Let $\tau_1, \ldots, \tau_m$ be the classes where $\tau_i = (c_i k, w_i)$ - $c_i k$ is the number of nodes in this class and w_i is their weight. Further denote by x_i the number of corrupt nodes in class τ_i.

5.1 Fixed-size Random Assignment

Let X_i be the number of corrupt nodes in class τ_i in a committee and x_i be the total number of corrupt nodes of class i (in total). Note that x_i follows a hypergeometric distribution. Thus, we have

$$Var[X_i] = c_i \cdot \frac{x_i}{c_i k}(1 - \frac{x_i}{c_i k}) \cdot \frac{c_i k - c_i}{c_i k - 1} = x_i(c_i k - x_i) \cdot \frac{k-1}{k^2(c_i k - 1)}$$

We are interested in maximizing $Var[\sum_i x_i w_i]$ since that would indicate the adversary's optimal strategy. Thus we have the following Quadratic Programming problem:

$$\begin{aligned}
\text{Maximize:} \quad & \sum_{i=1}^{m} w_i^2 \cdot x_i(c_i k - x_i) \cdot \frac{k-1}{k^2(c_i k - 1)} \\
\text{Subject to:} \quad & \sum_{i=1}^{m} w_i x_i \leq f \\
& \sum_{i=1}^{m} c_i k w_i = 1 \\
& 0 \leq w_i \leq 1 \quad \forall i \\
& c_i, x_i \geq 0 \quad \forall i.
\end{aligned}$$

Note that the maximum variance is achieved when $x_i = \frac{c_i k}{2}$. Thus, we simplify the problem:

$$\begin{aligned}
& \text{Maximize:} \quad \sum_{i=1}^{m} w_i^2 \cdot x_i(c_i k - x_i) \cdot \frac{k-1}{k^2(c_i k - 1)} \\
\iff & \text{Maximize:} \quad \sum_{i=1}^{m} w_i^2 \cdot \frac{k-1}{k^2(c_i k - 1)} \cdot \left(-(x_i - \frac{c_i k}{2})^2 + (\frac{c_i k}{2})^2 \right) \\
\iff & \text{Minimize:} \quad \sum_{i=1}^{m} w_i^2 \cdot \frac{k-1}{k^2(c_i k - 1)} \cdot (x_i - \frac{c_i k}{2})^2
\end{aligned}$$

Let $y_i = x_i - \frac{c_i k}{2}$. Then first note that

$$\sum_{i=1}^{m} w_i y_i = \sum_{i=1}^{m} w_i x_i - \sum_{i=1}^{m} w_i \frac{c_i k}{2} \leq f - \frac{1}{2}$$

Thus, we have the following Quadratic Programming problem:

$$\begin{aligned}
\text{Minimize:} \quad & \sum_{i=1}^{m} w_i^2 \cdot \frac{k-1}{k^2(c_i k - 1)} \cdot y_i^2 \\
\text{Subject to:} \quad & \sum_{i=1}^{m} w_i y_i \leq f - \frac{1}{2} \\
& \sum_{i=1}^{m} c_i k w_i = 1 \\
& 0 \leq w_i \leq 1 \quad \forall i \\
& c_i \geq 0 \quad \forall i \\
& y_i \leq 0 \quad \forall i.
\end{aligned}$$

Here the last constraint follows since we may assume that $x_i \leq \frac{c_i k}{2}$; otherwise we can replace x_i by $c_i k - x_i$, obtaining the same objective value while satisfying the budget constraint.
Assuming the adversary uses the entire budget, Cauchy–Schwarz gives

$$\left(\sum_{i=1}^{m} w_i^2 \frac{k-1}{k^2(c_i k - 1)} y_i^2 \right) \left(\sum_{i=1}^{m} \frac{k^2(c_i k - 1)}{k-1} \right) \geq \left(\sum_{i=1}^{m} w_i y_i \right)^2,$$

and therefore

$$\sum_{i=1}^{m} w_i^2 \frac{k-1}{k^2(c_i k-1)} y_i^2 \geq \frac{\left(f-\frac{1}{2}\right)^2}{\sum_{i=1}^{m} \frac{k^2(c_i k-1)}{k-1}}.$$

Equality holds if and only if there exists a constant λ such that

$$y_i = \frac{\lambda}{w_i} \cdot \frac{k^2(c_i k-1)}{k-1}.$$

To determine λ,

$$\begin{aligned} f-\frac{1}{2} &= \sum_{i=1}^{m} w_i y_i \\ &= \lambda \sum_{i=1}^{m} \frac{k^2(c_i k-1)}{k-1}, \end{aligned}$$

so that

$$\lambda = \frac{f-\frac{1}{2}}{\sum_{i=1}^{m} \frac{k^2(c_i k-1)}{k-1}}.$$

Hence,

$$\begin{aligned} x_i &= \frac{c_i k}{2} + \frac{\lambda}{w_i} \cdot \frac{k^2(c_i k-1)}{k-1} \\ &= \frac{c_i k}{2} + \frac{f-\frac{1}{2}}{w_i \sum_{j=1}^{m} \frac{k^2(c_j k-1)}{k-1}} \cdot \frac{k^2(c_i k-1)}{k-1} \\ &= \boxed{\frac{c_i k}{2} - \frac{\frac{1}{2}-f}{w_i} \cdot \frac{c_i k-1}{\sum_{j=1}^{m}(c_j k-1)}} \end{aligned}$$

For this solution to be feasible, it is necessary that

$$x_i \geq 0$$

for every i, i.e.,

$$\frac{c_i k}{2} - \frac{\frac{1}{2}-f}{w_i} \cdot \frac{c_i k-1}{\sum_{j=1}^{m}(c_j k-1)} \geq 0 \iff \frac{c_i k}{2} \geq \frac{\frac{1}{2}-f}{w_i} \cdot \frac{c_i k-1}{\sum_{j=1}^{m}(c_j k-1)} \iff \frac{c_i k w_i}{2} \geq \frac{(\frac{1}{2}-f)(c_i k-1)}{\sum_{j=1}^{m}(c_j k-1)}.$$

The total variance is therefore

$$\begin{aligned} \sum_{i=1}^{m} w_i^2 x_i (c_i k - x_i) \cdot \frac{k-1}{k^2(c_i k-1)} &= \sum_{i=1}^{m} w_i^2 \left(\frac{c_i k}{2} - \frac{\frac{1}{2}-f}{w_i} \cdot \frac{c_i k-1}{\sum_{j=1}^{m}(c_j k-1)}\right)\left(\frac{c_i k}{2} + \frac{\frac{1}{2}-f}{w_i} \cdot \frac{c_i k-1}{\sum_{j=1}^{m}(c_j k-1)}\right) \cdot \frac{k-1}{k^2(c_i k-1)} \\ &= \sum_{i=1}^{m} w_i^2 \left(\left(\frac{c_i k}{2}\right)^2 - \left(\frac{\frac{1}{2}-f}{w_i} \cdot \frac{c_i k-1}{\sum_{j=1}^{m}(c_j k-1)}\right)^2\right) \cdot \frac{k-1}{k^2(c_i k-1)} \\ &= \sum_{i=1}^{m} \left(\left(\frac{c_i k w_i}{2}\right)^2 - \left((\frac{1}{2}-f) \cdot \frac{c_i k-1}{\sum_{j=1}^{m}(c_j k-1)}\right)^2\right) \cdot \frac{k-1}{k^2(c_i k-1)} \end{aligned}$$

Note that the above equation is only correct if x_i's are feesible (otherwise it is not the maximum). Thus when implementing, we need to remove the negative values and re-run the algorithm.

5.2 Uniformly Random Assignment

It is not sufficient here to study the variance of the weights of corrupted nodes since the stake at each committee is not fixed. Instead, we need to study the distribution of the ratio of corrupt to honest nodes. Let x_i be the ratio of corrupt nodes to honest nodes in class τ_i.

6 Closing Remarks

Overall, the variance guided approach looks promising - under certain assumptions (such as $np > 5$), we can use the normal approximation to get a good estimate of the probability of failure.
Additionally, we also verified using handcrafted examples that $ALG_{5.1}$ performs better than other natural ideas that do not consider classes.
Future Work: The main missing piece is formation of classes. Essentially, forming too few classes increases the variance (which adversary can exploit) while forming too many classes gives the distribution a more rigid structure that the adversary can exploit. By simple computation, we can see that if ck items have exactly the same weight, then it is better to put them in the same class. If the items varry "too much" it is better to put them in separate classes.

7 Second Problem Statement - Duplex CFLOOD

Refer to the following paper : `https://www.comp.nus.edu.sg/~yuhf/TRA4-16.pdf`.
The paper considers the simplex model : In each round, nodes are set to be either receiving or sending. Then, simultanously, the messages are sent from sending node to recieving node (if an edge exists in that round).
We are interested now in the duplex model : In each round, nodes can simultaneously send and receive messages.
It can be verified that the same strategy no longer holds - thus this is a new non-trivial problem.

8 Class of reductions that do no work

Expanding on the non-triviality, consider the following class of reductions : "Under the same cycle promise (and same "public" nodes (A_τ, B_τ)), the chain is disconnected from the even side while the odd side (if any) remains connected".
In the paper, the chain is a simple chain with 3 nodes and 2 edges. We may replace it by any complicated graph with any complicated adversary rules - we claim that this will not work.
To see this, we can first assume that the CFLOOD algorithm in duplex mode has all nodes both sending and receiving. Now either Alice or Bob will not be able to match the true adversary (otherwise the problem will be trivially solved without communication). WLOG assume Bob "mistimes" an edge (removed or added), by our assumption, both adjacent nodes will get spoilt. In particular, nodes on both sides of the cut will get spoilt so B_τ also gets spoilt - thus the simultaion can not continue.

9 Useful tools

Consider allowing more than two "public" nodes. Public nodes are nodes that Alice (Bob) can simulate correctly and tell Bob (Alice) the outgoing messages. This can help stop spoiltness from spreading. Of course, using more public nodes decrease the lower bound.
Within this, consider allowing "dynamic public" nodes - that is, some nodes may be public only in some specific agreed upon rounds. This is to help reduce the number of public nodes used overall.