Talk: Knapsack Voting for Concurrent Block Proposal

Geoffrey Ramseyer[1], Chenghan Zhou[2], and Ashish Goel[3]

[1] Stellar Development Foundation, San Francisco, CA 94107 geoff@stellar.org
[2] Stanford University, Stanford, CA 94305 chzhou@stanford.edu
[3] Stanford University, Stanford, CA 94305 ashishg@stanford.edu

Abstract. Many consensus protocols elect one leader at each round to propose the next block of transactions. This design gives the leader a temporary monopoly over allocating block space, which raises fears of censorship and rent extraction. In response, a number of protocols have recently been formulated in which multiple leaders concurrently propose blocks at every round. We study the problem of how these concurrent proposals can be aggregated into a single, next block.

Our key observation is that this problem lies at a novel intersection between mechanism design and the branch of social choice theory known as Participatory Budgeting. The naive approach to the aggregation problem, concatenation, wastes limited blockchain resources. This social choice perspective enables new system designs with desirable properties, such as a weak form of strategy-proofness for block proposers, while ensuring system resources are not wasted.

But unlike the classic social choice setting, all parts of the system—block proposers and the end-users sending transactions—are strategic agents. A user's incentives when, for example, setting a transaction fee bid are closely tied to the aggregation method. Natural aggregation rules can lead to economically inefficient outcomes. We believe that understanding these incentives is a crucial prerequisite for deploying blockchains with multiple proposers, and pose open questions that we suspect will prove fruitful in this research direction.

Many public blockchains use consensus protocols that periodically elect a single leader, who proposes the next block in the chain. This design choice grants a *temporary monopoly* over allocation of the contents of a block to each leader. This monopoly can be extremely profitable, and raises fears of censorship of transactions. Instead, recent works (such as Fox et al. [3] and Neuder and Resnick [5]) have proposed electing multiple leaders at each round as a way of bypassing this monopoly. These works, along with common-sense intuition, argue that using multiple leaders can bring desirable properties, such as reducing the ability of one leader to censor a transaction or extract rents via higher transaction fees.

We study here the problem of how these multiple, concurrent block proposals can be combined into a single block. Concatenation, the naive approach to this problem, has the potential to waste the limited computational resources of today's blockchains. For example, a chain with the capacity to execute K

transactions per block could allocate K/N units of blockspace to each of N block proposers, but if multiple proposers include the same transactions (e.g., if they all select the highest-fee transactions), then the chain will only use a $1/N$ fraction of its capacity.

We argue that this problem is best viewed as a novel blend of mechanism design and social choice. Our first observation is that this problem is closely related to that of "Participatory Budgeting" (for a survey, see Aziz and Shah [2]). In the classic setting, an election administrator creates a menu of policy options, each of which has a financial cost. For example, a city might propose building a library for $1 million, spending $500,000 on road repairs, and spending $700,000 on bicycle infrastructure. The election has a budget limit. Voters are asked to decide how to allocate the budgeted money across projects, and then these votes are aggregated into an actual city budget.

A blockchain with multiple proposers closely maps onto this election paradigm. Each transaction (budget item) consumes a certain amount of blockchain gas, and each block has a total gas limit (budget). When there are multiple concurrent block proposers, each proposal looks like a voter's budget allocation, and the consensus protocol takes the role of the election administrator, aggregating proposed budgets into a final block.

This perspective on the problem opens up new designs for constructing blockchains by leveraging widely-used results from social choice theory. As an example, we show how Knapsack Voting (Goel et al. [4]) can enable a design that both includes input from multiple proposals (a basic property, required for any form of censorship-resistance) and additionally prevents a blockchain's limited computational resources from being wasted by repeated transactions. Furthermore, there is already extensive research on voter incentives in the participatory budgeting literature, which we can leverage in this blockchain setting. We show that Knapsack Voting, for example, maintains a weak strategy-proofness property when adapted to a blockchain.

What makes this problem novel and distinct from the social choice setting is that both the block proposers *and* the users that send transactions are strategic agents. A user can strategically choose their transaction fee bid, while a bicycle lane does not strategically choose its cost, and the classic participatory budgeting setting typically takes the set of budget menu items as an exogenous input. A user's strategic choice of fee bid is inherently coupled with the proposal aggregation process. We show by example that natural mechanism designs can lead to outcomes that are inefficient with regard to social welfare (of the users) and also give suboptimal fee revenue (for block proposers).

We believe that this combination of social choice and mechanism design poses a number of interesting challenges, and that understanding these incentive questions is a prerequisite for deploying systems with concurrent block proposers. This talk will describe this ongoing research direction, highlight some promising results to demonstrate the utility of this perspective, and propose several key, open questions that we suspect will be important for future system designs.

1 Model

We consider a basic model that captures the core features of the problem. At some protocol-defined frequency, a constant number N of blockchain "validators" are selected as "proposers" to propose the next block.

Each block has a "gas limit" K, which governs the size of each block. Every transaction has its own "gas cost". We assume here that the gas cost (or an upper bound thereof) of each transaction is computable in advance.

After selection, each of the N validators proposes a block of transactions. Each validator's proposal is limited to some gas limit $k \leq K$. These validators communicate with each other through some reliable medium, and come to agreement on the set of proposed blocks. Then, an "aggregation rule" takes all of these proposals together and produces an output block. We assume this aggregation rule is implementable as a deterministic function of the inputs (i.e., something that can run on-chain as part of the blockchain protocol). The output of this aggregation rule should be a block of transactions that respects the overall gas limit K. The process then repeats. These aggregation rules are the focus of our study.

We assume that transactions originate from "clients," which can make both public and private agreements with validators, that the protocol cannot restrict communication or agreements between clients and validators, and that clients and validators can send payments to each other, conditioned on the observable behavior of a validator (i.e. a client can make a payment, conditioned on its transaction's inclusion in a validator's proposal and the final block).

2 Knapsack Voting

As an example, we show that a process known as Knapsack Voting 4 can be naturally adapted to the block aggregation setting. In the classic setting, every voter allocates the entirety of a budget to their preferred projects (fractional allocations are allowed). Here, we generalize to limit each proposer to a predefined gas limit of $k \leq K$.

Any mechanism's performance depends on the behavior and incentives of the actors involved. We present this setting here as an example of the kinds of observations and questions that we believe this line of research can lead to. Assume for the moment that block proposers are purely profit-maximizing from transaction fees, and that fees are paid as fixed bids to each proposer, conditional on both the proposer including a transaction and the transaction's inclusion in the output block.

Given a set of N proposals, Knapsack voting proceeds as follows. The *score* of the ith unit of gas for a transaction is the number of proposals that include that transaction and give that transaction at least i units of gas (equivalently, each transaction is logically broken into unit-size increments). Knapsack greedily allocates gas to transactions in order of score, until it reaches the gas limit. Ties between transactions of equal score are broken by a well-defined, transitive tie-breaking rule (i.e. sorting by hash).

This mechanism satisfies a weak form of strategy-proofness. Following Theorem 2.10 of 4, consider a proposer i responding to the proposals of all other proposers. Let W_{-i} be the allocation of gas that would be selected without i's vote. Given this set, it is easy to compute a best response S_i^* for i which myopically maximizes i's revenue. We say that an allocation of the kth unit of gas to a transaction τ *dominates* the allocation of the jth unit of gas to transaction $\tau' \neq \tau$ if and only if $k \in W_{-i}$ and i receives more revenue for k than for j.

Theorem 2.1. *Under Knapsack Voting, there exists a best response S_i^* for i such that if k dominates $j \in S_i^*$, then $k \in S_i^*$.*

Theorem 2.1 is a generalization of Theorem 2.10 of Goel et al. 4 to our setting. The primary difference is that block proposers allocate fewer units of gas than the total amount of gas available.

One challenge with the Knapsack allocation rule (and indeed, with many potential allocation rules) is that some transactions will only be allocated a fraction of the gas that they require. The simplest solution is to interpret a fractional allocation as a probability of inclusion within a block. A recent result of Aziz et al. 1 (Theorem 3.2) gives a randomized rounding scheme that rounds a fractional allocation to an integral one (up to one item) that preserves the marginal probability of each item.

3 Auction Dynamics

A key feature that differentiates this problem setting from the classic participatory budgeting setting is that the set of possible transactions (and proposers' utilities, or payments) is not statically determined (i.e. by an election administrator). Instead, clients are strategic agents. A user, for example, decides strategically what fees to offer on their transactions, and these strategies interact with a blockchain's aggregation rule in novel ways. This is already a challenging problem even when proposers greedily maximize (expected) profit with a naive aggregation rule (concatenation); 3 gives one such example.

A single-proposer system can run mechanisms with good welfare properties, such as a classical second-price auction. We show by example that even in simple settings, moving from one block proposer to multiple can significantly reduce both overall welfare and validator fee revenue.

4 Acknowledgements

This research was supported by the Stanford IOG Research Hub, the Stanford Future of Digital Currency Initiative, and the Stellar Development Foundation.

References

Aziz, H., Lu, X., Suzuki, M., Vollen, J., Walsh, T.: Fair lotteries for participatory budgeting (2024), https://arxiv.org/abs/2404.05198

Aziz, H., Shah, N.: Participatory budgeting: Models and approaches. Pathways Between Social Science and Computational Social Science: Theories, Methods, and Interpretations pp. 215–236 (2021)

Fox, E., Pai, M., Resnick, M.: Censorship resistance in on-chain auctions. arXiv preprint arXiv:2301.13321 (2023)

Goel, A., Krishnaswamy, A.K., Sakshuwong, S., Aitamurto, T.: Knapsack voting for participatory budgeting. ACM Transactions on Economics and Computation (TEAC) **7**(2), 1–27 (2019)

Neuder, M., Resnick, M.: Concurrent block proposers in ethereum (Feb 2024), `https://web.archive.org/web/20240424114807/https://ethresear.ch/t/concurrent-block-proposers-in-ethereum/18777`