

[Gas auctions](#) in the mempool are nothing new. However, unlike Bitcoin, tx fees on Ethereum are not monotonic in the sense that the exclusion of a tx from the block can actually increase the miner's revenue from that block. Consider the following contract code:

```
if(x) throw else revert;
```

The inclusion in a block of a transaction to this contract that has a high gas allowance and a high gas price results in a disincentive for the validator to include any transaction before it that might result in changing the condition x

```
from true
```

```
to false
```

```
.
```

One problem with the above code is that throw

is a special opcode in that unlike all the other opcodes the validator does not need to perform the corresponding amount of work to get the gas reward. But naïve solutions such as burning the remaining gas rather than awarding it to the miner would not work, as any recognizable infinite loop can be used as “throw

by convention”.

Another, far more serious problem is a gas price that is potentially far beyond the costs of validation. Again, naïve solutions treating blocks that include transactions with gas prices far in excess of the minimum or some average (e.g. the median) won't work, as they would merely compel the validators to only include bribe transactions in certain blocks.

The general problem is buying up a scarce resource (the block gas limit

) at a high price conditioned on outcomes of on-chain calculations. Note that this is far worse than off-chain bribes for excluding transactions because the commitments to pay the bribe are enforced by the blockchain and any coordination problems that collective bribing might incur can be solved through smart contracts. Capping gas prices won't work either, because the resource won't become less scarce with price controls and there are many other ways in which validators can charge for inclusion in the block besides the default mechanism.

With ordering-critical but not timing-critical transactions a viable solution is validating ourselves. That is instead of trying to get a sequence of transactions into blocks validated by other validators, we may want to include a block of our own in the blockchain. This does require either mining power or validator stake (depending on the type of the blockchain). However, this merely shifts the all-pay auction from tx fees to hashing power or validator stake.

The problem seems fundamental to Turing-complete blockchains with all-pay auctions for tx inclusion being an inevitable fact of life. One way of keeping all-pay auctions from runaway bidding is credible commitments to very high bids. With account abstraction, it becomes possible for third parties to pay for the execution of transactions. Thus, one can subscribe (for a fee) to services with sufficiently deep war chests that explicitly offer to “bid to death” for transactions clearly marked as originating from subscribers. As long as the commitment is observable and credible, actual bidding wars can be avoided.