WTF is EIP-1559 *actually* doing?

Theo Diamandis
work with Guillermo Angeris, Tarun Chitra, Alex Evans, and Ciamac Moallemi

Columbia Cryptoeconomics Conference 2024

EIP-1559 solves a specific **online optimization problem**

EIP-1559 solves a specific **online optimization problem** with a **particular algorithm**

EIP-1559 solves a specific **online optimization problem** with a **particular algorithm** that is **low regret**.

Outline

Pricing algorithms solve the dual of a particular **resource allocation problem** (how to pack ‘optimal’ blocks).

Outline

Formalizing block building

- Mempool with n possible txns
  - Txn j consumes gas $a_j \in \mathbb{R}_+$, has utility $q_j \in \mathbb{R}_+$
  - Vector $x \in \{0,1\}^n$ encodes which txns are included in the block
  - Set of allowable txns is $S \subseteq \{0,1\}^n$ (gas limit, MEV constraints, etc)

Formalizing block building

- Mempool with n possible txns
  - Txn j consumes gas $a_j \in \mathbb{R}_+$, has utility $q_j \in \mathbb{R}_+$
  - Vector $x \in \{0,1\}^n$ encodes which txns are included in the block
  - Set of allowable txns is $S \subseteq \{0,1\}^n$ (gas limit, MEV constraints, etc)
- Gas usage of the block is $y = a^T x$.
- Network's unhappiness with this usage given by loss function $\ell(y)$, for example:

$$\ell(y) = \begin{cases} 0 & y = \mathrm{target} \\ \infty & \text{otherwise} \end{cases}$$

The resource allocation problem

$$
\begin{aligned}
\text{maximize} \quad & q^T x - \ell(y) \\
\text{subject to} \quad & y = a^T x \\
& x \in S.
\end{aligned}
$$

The resource allocation problem

$$
\begin{array}{ll}
\text{maximize} & q^T x - \ell(y) \\
\text{subject to} & y = a^T x \\
& x \in S.
\end{array}
$$

- **Objective:** Maximize utility of included txns minus loss incurred by the network

The resource allocation problem

$$\begin{array}{ll} \text{maximize} & q^T x - \ell(y) \\ \text{subject to} & y = a^T x \\ & x \in S. \end{array}$$

- **Objective:** Maximize utility of included txns minus loss incurred by the network
- **Constraints:** Utilization y is resource usage of included txns, and x is in the set of allowable txns $S \subseteq \{0, 1\}^n$

The resource allocation problem

$$\begin{array}{ll} \text{maximize} & q^T x - \ell(y) \\ \text{subject to} & y = a^T x \\ & x \in S. \end{array}$$

- But network designer cannot solve this in practice!
  - Doesn't decide which txns are in a block (block builders do this)
  - Doesn't know utilities q

The resource allocation problem

$$\begin{array}{ll} \text{maximize} & q^T x - \ell(y) \\ \text{subject to} & y = a^T x \\ & x \in S. \end{array}$$

- But network designer cannot solve this in practice!
  - Doesn't decide which txns are in a block (block builders do this)
  - Doesn't know utilities q
- Goal: set prices so that this problem is solved optimally on average

Duality theory: relaxing constraints to penalties

$$\begin{array}{ll} \text{maximize} & q^T x - \ell(y) \\ \text{subject to} & y = a^T x \\ & x \in S. \end{array}$$

- Network designer cares about utilization y, based on txns x
- Block builders only care about which txns they can include

Duality theory: relaxing constraints to penalties

$$\begin{array}{ll} \text{maximize} & q^T x - \ell(y) \\ \text{subject to} & y = a^T x \\ & x \in S \end{array}$$

- Network designer cares about utilization y, based on txns x
- Block builders only care about which txns they can include
- We will 'decouple' utilization of network and that of tx producers

Duality theory: relaxing constraints to penalties

$$\begin{array}{ll} \text{maximize} & q^T x - \ell(y) \\ \text{subject to} & y = a^T x \\ & x \in S \end{array}$$

- Network designer cares about utilization y, based on txns x
- Block builders only care about which txns they can include
- We will 'decouple' utilization of network and that of tx producers
- Correctly set penalty $\rightarrow$ dual problem = original problem & utilizations are equal

Dual decouples tx producers and network

- Dual problem is to find the prices $p \in \mathbb{R}_+$ that minimize dual function $f(p)$

Dual decouples tx producers and network

- Dual problem is to find the prices $p \in \mathbb{R}_+$ that minimize dual function $f(p)$
- From before, p are the prices for violating prev. constraint $y = a^T x$
  - Relaxing constraint to penalty $\rightarrow$ pay per unit violation

Dual decouples tx producers and network

- Dual problem is to find the prices $p \in \mathbb{R}_+$ that minimize dual function $f(p)$
- From before, p are the prices for violating prev. constraint $y = a^T x$
  - Relaxing constraint to penalty $\rightarrow$ pay per unit violation
- Dual problem is to minimize dual function $f(p)$,

$$\text{minimize} \quad f(p) \quad = \quad \underbrace{\ell^*(p))}_{\text{network}} \; + \; \underbrace{\sup_{x \in S} \, (q - p \cdot a)^T x}_{\text{tx producers}},$$

where ℓ^* is the conjugate function of ℓ (typically closed form)

Second term: block building problem

- Maximize net utility (utility minus cost) subject to tx constraints

$$\begin{array}{ll} \text{maximize} & (q - p \cdot a)^T x \\ \text{subject to} & x \in S. \end{array}$$

- Exact problem solved by block producers! $\rightarrow$ Network can observe $x^\star$
- Thus, we can compute the gradient of f:

$$\nabla f(p) = y^\star(p) - a^T x^\star(p)$$

Outline

Online problem

- But the transactions are not static! At each block:

Online problem

- But the transactions are not static! At each block:
  1. **Network chooses prices** p_t

Online problem

- But the transactions are not static! At each block:
  1. **Network chooses prices** p_t
  2. Users submit txns (with utilities q_t, gas usage a_t), possibly adversarially

Online problem

- But the transactions are not static! At each block:
  1. **Network chooses prices** p_t
  2. Users submit txns (with utilities q_t, gas usage a_t), possibly adversarially
  3. Network receives payoff $f_t(p_t)$ (from duality)

Online problem

- But the transactions are not static! At each block:
  1. **Network chooses prices** p_t
  2. Users submit txns (with utilities q_t, gas usage a_t), possibly adversarially
  3. Network receives payoff $f_t(p_t)$ (from duality)
- New goal: dynamically choose prices p_t to minimize the average loss,

$$\text{minimize} \quad \frac{1}{T}\sum_{t=1}^{T} f_t(p_t).$$

Outline

Natural idea: mirror descent

- Next price p_{t+1} minimizes linear approx. of $f_t(p)$ plus a regularizer $D(p, p_t)$:

$$p_{t+1} = \operatorname*{argmin}_p \hat{f}_t(p) + \frac{1}{2\eta} D(p, p_t)$$

where $\hat{f}_t(p) = f_t(p_t) + \nabla f_t(p_t)^T (p - p_t)$

Natural idea: mirror descent

- Next price p_{t+1} minimizes linear approx. of $f_t(p)$ plus a regularizer $D(p, p_t)$:

$$p_{t+1} = \operatorname*{argmin}_p \hat{f}_t(p) + \frac{1}{2\eta} D(p, p_t)$$

where $\hat{f}_t(p) = f_t(p_t) + \nabla f_t(p_t)^T (p - p_t)$

- $D(p, p_t)$ is a strongly convex function that keeps p_{t+1} close to p_t

Natural idea: mirror descent

- Next price p_{t+1} minimizes linear approx. of $f_t(p)$ plus a regularizer $D(p, p_t)$:

$$p_{t+1} = \operatorname*{argmin}_p \hat{f}_t(p) + \frac{1}{2\eta} D(p, p_t)$$

  where $\hat{f}_t(p) = f_t(p_t) + \nabla f_t(p_t)^T (p - p_t)$

- $D(p, p_t)$ is a strongly convex function that keeps p_{t+1} close to p_t
- Example: recover vanilla gradient descent by setting $D(p, p_t) = \|(p - p_t)\|_2^2$

Back to EIP-1559

- Choose the loss function and regularizer:

$$\ell(y) = \begin{cases} 0 & y = 15M \\ \infty & \text{otherwise} \end{cases}$$

$$D(p, p_t) = p \cdot \log(p/p_t) - p$$

Back to EIP-1559

- Choose the loss function and regularizer:

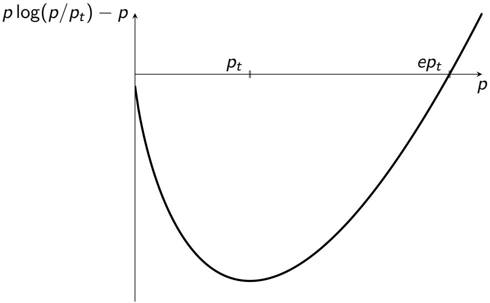
$$\ell(y) = \begin{cases} 0 & y = 15M \\ \infty & \text{otherwise} \end{cases} \qquad D(p, p_t) = p \cdot \log(p/p_t) - p$$

- Recover the EIP-1559 & EIP-4844 update rule:

$$p_{t+1} = p_t \cdot \exp\left(\eta(\underbrace{a^T x}_{\text{usage}} - \underbrace{15M}_{\text{target}})\right)$$

- Lots of questions...

What's this regularizer doing?



- 'Bregman divergence' associated with the negative entropy function
- Asymmetric; penalizes decreases more than increases
  - May explain positive overshoot of EIP-1559 (Leonardos et al. 2023)

More questions

- Is this the right regularizer?

More questions

- Is this the right regularizer?
- Is this loss function a good one?

More questions

- Is this the right regularizer?
- Is this loss function a good one?
- Is mirror descent a good update rule?

More questions

- Is this the right regularizer?
- Is this loss function a good one?
- Is mirror descent a good update rule?
- With what benchmark do we measure 'good'?

More questions

- Is this the right regularizer?
- Is this loss function a good one?
- Is mirror descent a good update rule?
- With what benchmark do we measure 'good'?
- What should η be?

Outline

Benchmark: regret

- Metric: *regret* of the network:

$$\mathrm{regret}(T) = \frac{1}{T}\left(\sum_{t=1}^{T} f_t(p_t) - \min_{p^\star} \sum_{t=1}^{T} f_t(p^\star)\right)$$

- Interpretation: difference between our update rule and the best fixed prices $p^\star$
  - Knowing $p^\star$ requires omniscience: assumes you know all future txns!
  - Analogy: online vs. offline learning

Choose the right step size, get low regret

- Choose the step size $\eta > 0$ such that

$$\eta \leq \frac{1}{\mathit{target}} \cdot \frac{1}{\sqrt{2T}} \cdot \sqrt{\log(p^{\max}) - 1}$$

- Then we have

$$\mathrm{regret}(T) \leq \frac{C}{\sqrt{T}}$$

- This regret goes to zero as T gets large!

Does EIP-1559 have the right step size?

- Recall that EIP-1559 is (approximately) given by

$$p_{t+1} = p_t \cdot \exp\left(\frac{1}{8}\left(\frac{\text{usage} - \text{target}}{\text{target}}\right)\right) = p_t \cdot \exp\left(\frac{1}{8 \cdot 15M}\left(a^T x - 15M\right)\right)$$

Does EIP-1559 have the right step size?

- Recall that EIP-1559 is (approximately) given by

$$p_{t+1} = p_t \cdot \exp\left(\frac{1}{8}\left(\frac{\text{usage} - \text{target}}{\text{target}}\right)\right) = p_t \cdot \exp\left(\frac{1}{8 \cdot 15M}\left(a^T x - 15M\right)\right)$$

- We get $O(1/\sqrt{T})$ regret with

$$\eta = \frac{1}{\text{target}} \cdot \frac{1}{\sqrt{2T}}$$

Does EIP-1559 have the right step size?

- Recall that EIP-1559 is (approximately) given by

$$p_{t+1} = p_t \cdot \exp\left(\frac{1}{8}\left(\frac{\text{usage} - \text{target}}{\text{target}}\right)\right) = p_t \cdot \exp\left(\frac{1}{8 \cdot 15M}\left(a^T x - 15M\right)\right)$$

- We get $O(1/\sqrt{T})$ regret with

$$\eta = \frac{1}{\text{target}} \cdot \frac{1}{\sqrt{2T}}$$

- Implies EIP-1559's choice of T is only 32 blocks!
  - Regret decays slowly with T; this T is likely too small... (and η is too large)

Outline

The optimization problem
- Single block
- Multiple blocks (online)

The algorithm

The regret (*i.e.*, why the algorithm works well)

Conclusion

What is EIP actually doing?

The Ethereum base fee algorithm is...

What is EIP actually doing?

The Ethereum base fee algorithm is...

- Solving the 1d resource allocation problem

What is EIP actually doing?

The Ethereum base fee algorithm is...

- Solving the 1d resource allocation problem
- ...with mirror descent
  - regularizer: Bregman divergence generated by the negative entropy function

What is EIP actually doing?

The Ethereum base fee algorithm is...

- Solving the 1d resource allocation problem
- ...with mirror descent
  - regularizer: Bregman divergence generated by the negative entropy function
- ...in a way that's low regret (for an appropriate step size)

What's next?

Lot's of open questions...

- Are we solving the right problem?
  - Multidimensional fees? What are the resources?
  - Another loss function?

What's next?

Lot's of open questions...

- Are we solving the right problem?
  - Multidimensional fees? What are the resources?
  - Another loss function?
- Is the step size well calibrated?
  - Theory here and empirical results (*e.g.*, Pai & Resnick 2024) suggest it's not

What's next?

Lot's of open questions...

- Are we solving the right problem?
  - Multidimensional fees? What are the resources?
  - Another loss function?
- Is the step size well calibrated?
  - Theory here and empirical results (*e.g.*, Pai & Resnick 2024) suggest it's not
- Should we choose another regularizer?

For more info, check out our paper!

Paper

Thank you!

Theo Diamandis
Research Partner, Bain Capital Crypto

tdiamandis@baincapital.com
@theo_diamandis
theodiamandis.com