Note: as a new user I had some post formatting limits, so it's much better to read this posthere, as it's more complete

Motivation

Quadratic funding is a powerful mechanism for resolving some collective action problems. But it has a major limitation - it relies on some third party, that provides a matching pool of funds.

In the most dangerous collective action problems, we don't have such third party helping us from above. Those situations already involve the most powerful actors, so we can't expect someone more powerful to resolve the conflict, like a galactic mom.

Some examples:

- global superpowers trying to coordinate to fight climate change
- AI organisations coordinating to pay [AI alignment tax] (more info [here])
- for example by funding safety research
- or creating some large dataset together, that's useful for alignment
- or funding methods which are thought to be safer, like STEM AI or tool AI
- for example by funding safety research
- or creating some large dataset together, that's useful for alignment
- or funding methods which are thought to be safer, like STEM AI or tool AI
- in general, escaping [inadequate equilibria] (see [this post] for many great examples)
- and most importantly, conflict between transformative AI systems or their owners

(This example may be the most important, but also the hardest to imagine as those systems don't exist yet. [This post] (section "1. Introduction") does a good job of describing this scenario. To quote it: "The size of losses from bargaining inefficiencies may massively increase with the capabilities of the actors involved."

Solution

One thing we can try in this situation, is to create a smart contract where each party says "I'll pay more if others pay more". This way, if you decide to increase your contribution by 1$, it causes the pot to grow by more than 1$, because your dollar caused other agents to contribute some more. This leverage, in some situations can be enough to make someone pay, because the value they get out of the bigger pot is higher than what they have to pay.

Some properties that it would be nice to have in such a system are:

- continuity - every increase in your payment causes an increase in others' payments
- known payment limit - you won't have to pay more than some limit you chose
- everyone is incentivised to contribute something - just like in quadratic funding, small contributions get a high leverage (it can get arbitrarily high, as you'll see later) - so even if you're only willing to pay if you get >100x leverage, there is always some contribution size that gives you such a high leverage

A very simple system that has these properties is given by those equations:

$$h=\sum_{i}^{}\sqrt{payment_i(h)}$$

$$payment_i(h)=\frac{limit_i}{\frac{\pi}{2}}arctan(h*saturation_speed_i)$$

- payment_i(h)

is the amount that i'th agent has to pay

- limit_i

is the i'th agent payment limit

- saturation_speed_i

tells how quickly i'th agent's limit will be approached as new agents make contributions (the choice of this parameter is underspecified for now, and is discussed in Parameter choice

section)

- given all those parameters, we find h

that satisfies the two equations above

It turns out, this system has a pretty graphical representation:

[

https://raw.githubusercontent.com/filyp/coordinated-quadratic-funding/main/animations/solution_finding.gif

(image larger than 4 MB)

](https://raw.githubusercontent.com/filyp/coordinated-quadratic-funding/main/animations/solution_finding.gif)

Each quarter-circle represents one agent's contribution. Area of a quarter-circle is the payment limit - the maximum amount this agent can pay. The yellow areas are what they currently pay in this particular situation. The squares on the right have the same areas as the respective sectors. So the height of the tower of squares represents h

- the sum of square roots of payments. The distance of a quarter-circle's center to the right corner is \frac{1}{saturation_speed}
- for small saturation_speed

, the quarter-circle is put further to the left and you can see that they saturate more slowly.

The animation shows the procedure for finding the solution to those two equations. We start with some arbitrary h

, then compute the payments (yellow sectors), then compute h

, recompute payments, recompute h

, and so on, until we converge on the stable solution.

On the next animation, you see what happens when someone new joins the smart contract. Their contribution increases h

, which makes others pay more. (Here the procedure of finding the solutions is ommited, and just the final solutions are shown).

(animation missing)

Here you can see the nice feature of quadratic funding: for small contributions, the leverage can get arbitrarily large. (To be precise, we compute the leverage on the margin

, so how the pot changes if you pay 0.01$ more.)

leverage_i=\frac{d\ \sum_{j}^{}payment_j}{d\ payment_i}

Because of this feature, the amount that you're willing to pay is roughly proportional to how much you care for the common resource (see [this] explanation of QF for the precise argument).

You can find the code for this algorithm here.

Example

Here you can see an example of such a contract from start to finish:

(animation missing)

There are 5 agents joining the contract one by one. You can see that the early contributions saturate quickly - what those agents finally pay is close to their payment limit. But there are always some less saturated contributions (the late ones), which provide some leverage to the newcomers, so the contract is alive.

Future work

Quadratic funding problems

Unfortunately, this mechanism inherits all the problems that ordinary quadratic funding has, like Sybil attacks and influence buying, but there is ongoing research trying to solve them [1.] [2.]. If we fail to solve those problems, we can always fall back to linear funding (compute h

as the sum of payments, instead of the sum of square roots of payments). This would be more robust and still enable coordination in some kinds of scenarios.

Parameter choice

Each contribution is specified by two parameters: limit

, and saturation_speed

. The limit

should be chosen by the contributor, but how the saturation_speed

is set, is left open. If its choice was left to the contributor too, it would be always optimal for them to choose the lowest saturation_speed

they can. So instead it should be set by the algorithm in a systematic way.

For example if we set it constant for all the contributors (which corresponds to all the quarter-circles having the same center), there may come a point where all of them become almost fully saturated and the leverage for new contributors vanishes. But this may rarely be a problem if the number of agents is small.

Alternatively, if each new contribution gets a smaller saturation_speed

than the previous ones (quarter-circles get placed more to the left), there will always be some unsaturated quarter-circles, so there always be a nice leverage for new contributors. But now, everyone is incentivised to wait for others to pay first, because being on the left means you pay less. This could create a deadlock where everyone is waiting for everyone.

If we made a simulation of how agents behave in this system, we could test several methods of setting saturation_speed

, and see which one results in the highest pot at the end.

Strategic thinking

Another potential problem is strategic thinking. Agents can think: “even if I don’t pay, the other agents will fund this anyway”. This problem is definitely smaller than in traditional fundraisers because of the leverage that this mechanism gives. But still, if many other agents join this contract after you, the real

leverage you get (what would happen counterfactually if you didn’t contribute) will be smaller than the immediate leverage you had at the time of joining the contract (the amount that the pot increased divided by what you paid). This real

leverage is much harder to compute, because it requires simulating what would happen if you didn’t contribute, which requires simulating agents’ strategies.

A solution would be to modify the algorithm to make the leverages predictable, so that everyone would know for sure

they will get the leverage they signed up for. This would prevent strategic thinking, and also make agents more willing to trust this system.

Coordinate where there is no pool of funds

This approach can be used directly where we have a shared resource which can be improved by throwing money at it. But what about situations which aren’t directly about money, like coordinating not to do some harmful thing?

Here, we would need to quantify what it means to do this harmful thing, and this quantification needs to be continuous. For example when countries coordinate to prevent climate change, we could count how much CO2 each is emitting - this number quantifies harm, in a continuous way. And if those measures could be reliably verified by some oracle, we could construct a system analogous to the one above: “I will emit less, if you emit less”.

An example for AI safety, could be performance on some alignment benchmark. AI organizations deploying their models, could say: “I will squeeze a few more points on this benchmark, if you squeeze some more”.

Of course it’s hard to keep such promises exactly - you probably will undershoot or overshoot the promised number. For this reason, there also need to be some rewards and penalties for missing the target.

Acknowledgements

Many thanks to Matthew Esche and Rasmus Hellborn for all their suggestions!