Double-Constrained Quadratic Funding

Funding real-world Public Goods with minimal fixed costs

Loudverse EthDenver 2022 Buidl Team

Thomas Vieira @neodaoist
John @justjohn
Nick Adamson @nïck
Peter Laird @plaird.eth

Background

Quadractic Funding mechanisms for funding public goods have been explored and successfully implemented by projects such as Gitcoin as means to encourage democratic mechanisms to determine which projects provide the greatest public benefits. This paper proposes an extension to Quadratic Funding to account for two key real world economic constraints that apply to public goods projects:

- constraint on available funding pool
- per-proposal minimum viable funding constraint

Pool Constraint

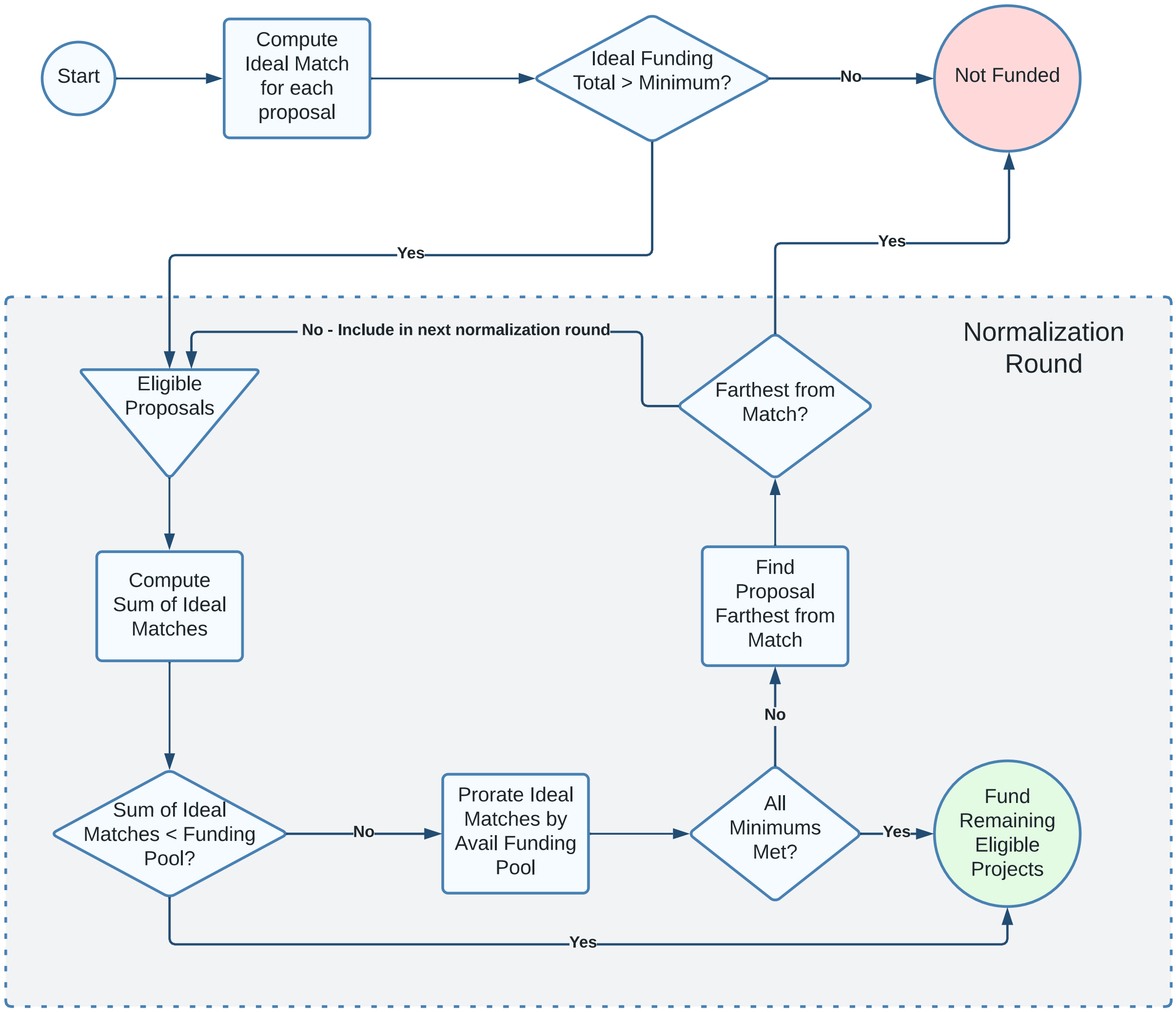
The pool constraint is the overall amount of funding available for the given funding round. The goal is to use the available funds to fund the greatest amount of viable public goods proposals possible.

Minimum Viable Cost Constraint

The Minimum Viable Cost Constraint is a constraint set by the proposer of a public goods project, representing the minimum amount of funds needed to complete the project. For example, the producer of a musical piece may need to hire a string quartet to perform the piece. If the funds raised are not sufficient, the project will not be successful and, more tragically, any funds that were contributed may be wasted.

Proposers are incentivized to keep their declared minimums as low as practical because proposals which do not have their minimums met are not funded. At the same time proposals with a greater public good value, as determined by the matching process, will provide their creators with greater rewards.

Approach



1. Double-Constrained Quadratic funding starts much like other forms of Quadratic Funding. For each proposal, an "Ideal Funding Total" is computed as the square of the sum of the square roots of each contribution to the proposal. This represents the individual contributions plus the amount of match the matching pool would contribute if the matching pool was not constrained (i.e. had infinite funds available)

$$IdealFundingTotal = (\sum_{i=1}^{n}\sqrt{Contribution_i})^2$$

$$IdealMatch = IdealFundingTotal - \sum CommunityContributions$$

2. Each proposal is checked to see if the Ideal Funding Total is greater than the Minimum Viable Cost for the proposal. Any proposals which fail this check are removed from the funding round as non-viable (depending on the implementation, the proposal could advance to another round

or the individual contributions could be refunded). Proposals which pass the check (those which could move forward if the funding pool has sufficient funds) move on to the Normalization rounds

3. Normalization: The sum of the Ideal Funding Total for each remaining proposal is computed and compared to the funding pool available amount. If the funding pool available amount is greater than the sum of the Ideal Funding Totals, then the normalization process ends and each proposal receives the full match.

4. Otherwise, a Pool Constraint Coefficient is computed which is the funding pool available funds divided by the Ideal Funding Totals sum. Each proposal is evaluated by taking the Ideal Match Amount times the Pool Constraint Coefficient to determine a pro-rated portion of the funding pool to apply. If all eligible proposals have met their minimum, exit and apply the prorated funding.

$$PoolConstraintCoefficient = \frac{PoolFundsAvailable}{\sum IdealMatch}$$

5. Otherwise, for each proposal that has not met its minimum, compute the difference between the proposed prorated funding and that proposal's minimum. The proposal with the greatest distance is closed as non-viable (deferring to a future round or funding as previously discussed). Then the normalization process is performed again with the remaining eligible proposals. This allows the full amount of funding available for matching to be applied to viable projects. Eventually the normalization rounds will end when either all both constraints are met (all remaining eligible proposals can be funded with the funds remaining in the pool) or when the eligible projects list drops to zero (which is the pedantic case where the pool lacks funds to fund even one project to its Minimum Viable Amount).

Advantages

- By allowing proposers to set the minimum viable amount, we ensure that only projects which are actually viable get community and organization fund matching. This avoids waste and coordination failures that might occur in other quadratic funding systems when proposers are awarded a match but lower than they expect, and may not be able to proceed with the project due to their own fixed costs.
- By applying matching funds even above the minimum viable amount, proposals with greater public good contributions receive greater funding, incentivizing and supporting creators to work on public good projects.

Risks

- Like all forms of Quadratic Funding, Sybil Attacks are a concern. This approach doesn't substantially increase or decrease this threat. More research is needed in this area