A Case Study in Plural Governance Design

Joel Miller[1], **Chris Kanich**[1], and **Glen Weyl**[2]

[1]Department of Computer Science, University of Illinois at Chicago
[2]Plural Technology Collaboratory, Microsoft Research
{jmill54,ckanich}@uic.edu, glenweyl@microsoft.com

Abstract

We describe the development and application of a plural algorithm for collective decision making. Specifically, we describe a modification to an algorithm called Quadratic Funding (QF), which was used by a crowdfunding platform named Gitcoin from 2018-2023. QF suffered drawbacks in practice due to its non-plural nature. Our modification to it, Connection Oriented Quadratic Funding (CO-QF), has been in use at Gitcoin since 2023, where it has directed the flow of 2.9 million dollars to crowdfunded projects. We describe the plural design principles behind CO-QF. Through qualitative interviews with Gitcoin community members, we explore how CO-QF provided a framework for aligning public decisions with diverse sets of viewpoints, allowing it to solve the problems suffered by QF. We conclude by discussing how CO-QF represents a plural framework for public decision making in general, including decisions about AI alignment.

1 Quadratic Funding and its implementation at Gitcoin

Quadratic Funding (QF) is a mechanism for awarding funding to public goods. While classical results from economics cast doubt on the ability of markets and 1-person-1-vote schemes to optimally fund public goods, QF funds public goods optimally under a set of assumptions common in the economics literature [3]. QF first accepts individual contributions to a public good from a set of agents N. Let $c_{i,p} \geq 0$ denote the contribution of agent $i \in N$ to the public good p, in dollars. Then QF awards p

$$\left(\sum_{i \in N} \sqrt{c_{i,p}} \right)^2$$

in dollars of funding. With two or more contributors the funding awarded by QF is greater than the sum of individual contributions, so QF relies on the assumption that the mechanism designer has a pool of subsidy cash to award to public goods. One can split the funding awarded by QF into *direct donations* (the sum of the c_i's) and the amount added by QF, which we call the *subsidy amount*. The subsidy amount can can be calculated as $(\sum_i \sqrt{c_{i,p}})^2 - \sum_i c_{i,p}$ or equivalently via algorithm 1.

A crowdfunding platform called Gitcoin[1] began using quadratic funding to award money to open-source software projects in 2018. Within a set time period, individual users donate to projects, and then Gitcoin awards the projects extra cash (obtained from corporate and NGO sponsors such as the EFF and the Ethereum Foundation) based on the output of QF. Gitcoin organizes its crowdfunding into a series of "rounds" with fixed subsidy pools. Each round admits a fixed set of projects (which are manually vetted) and runs for a fixed time period. During that time period, users are free to donate any amount to any projects they see fit. After the end of the round, QF is used to obtain a "raw" subsidy amount for each project (via algorithm 1). These raw subsidy amounts are then normalized

[1] gitcoin.co

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Algorithm 1: Calculation of QF subsidy amounts

Input: $c_{i,p}$ for $i \in N$ – contributions to the project p
Output: `subsidy` – the raw amount of subsidy funding awarded to project p

```
subsidy ← 0;
for i ∈ N do
    for j ∈ N do
        if i ≠ j then
            subsidy ← subsidy + √(c_{i,p} · c_{j,p});
return subsidy
```

so that their sum matches the size of that round's subsidy pool. [2] In total, each project is awarded the donations made by individuals plus its portion of the subsidy pool. Gitcoin runs its own funding rounds, and external organizations also run funding rounds using Gitcoin's infrastructure. Each round has one or more "round managers" who oversee that round's operation. External round managers made up the bulk of our interviewees: interviewees with IDs P2-P6 are external round managers, and P1 is a Gitcoin employee. (for more on our research methods see Appendix B).

2 Theoretical and Practical Issues with Quadratic Funding

While Gitcoin's use of QF had benefits over other schemes and afforded the platform public interest, the algorithm also suffered alignment problems due to its non-plural nature. We will explain this issue from both the theoretical and practical perspective.

Theoretically, QF derives its optimality (in part) from the assumption that all individuals are selfish and that, outside of the public good under consideration, all their consumption is of private goods. However, if individuals are social because of altruism, coordination or because beyond the present application they participate in networks of social consumption, QF loses its optimality (proof in Appendix C). Moreover, QF specifically loses its optimality by *overshooting* the optimal funding amount. The theory suggests, then, that a group of coordinated agents can shift funding outcomes in their favor, even if this shift does not align with the preferences of the community as a whole.

In practice, this is precisely what happened. Gitcoin's donor base participates in a range of shared goods in communities beyond the reach of the platform, allowing for the type of coordination discussed above. Moreover, at a practical level, the fact of the matter is that QF awards projects super-linear increases in funding for linear increases to that project's donor pool. This meant that groups of Sybil agents[3] or coordinated humans could dominate funding outcomes. Accordingly, the algorithm was sometimes disparagingly referred to as a "popularity contest". P3 brought up the term, saying "*I think the biggest complaint we hear with vanilla QF is it's a "popularity contest*". P3 went on to elaborate that this complaint was a result of QF lacking the "*granularity of the signal*" necessary to distinguish "*popular*" projects from projects that "*went out and did amazing things in the world*". P6 also independently brought up the term "popularity contest" and gave a similar account of how QF outcomes could become distorted. P4 echoed that sentiment, noting that "*there is of course the classic thing with Quadratic Funding which is that, you know, if you're a project that has lots of friends... you can slightly farm the mechanism*".

P1, a Gitcoin employee involved in Sybil and fraud defense, emphasized that "*you can't really disentangle [technical and social attacks] clearly*". Indeed, P1 went on to explain that while Gitcoin did implement helpful technical Sybil defense measures, they did not fully solve the problems mentioned above.

The core issue with QF, then, is that it has a non-plural view of the world. In a landscape of coordinated groups, QF tended to award the largest coordinated groups the maximal amount of funding, leaving smaller groups with scraps. However, as P3 points out, the projects with large coordinated groups of donors were *not* necessarily the projects doing "amazing things in the world".

[2]Most rounds also implement a "matching cap" which prohibits any project from taking more than some percent of the subsidy pool – if so, funding in excess of this amount is re-distributed to other projects below the matching cap.

[3]Sybil agents are multiple accounts controlled by the same human.

In order to solve the problems with QF, we modified the algorithm to prioritize projects with a plurality of support from diverse areas.

3 CO-QF: a Plural modification of QF

To understand our plural modification to QF, it helps to revisit QF's calculation of the subsidy amount, displayed in algorithm 1. Lines 4 and 5 of this algorithm, which add $\sqrt{c_{i,p} \cdot c_{j,p}}$ to project p's subsidy amount if and only if $i \neq j$, implement a type of "individual-level" plurality. That is, insofar as plurality is about prioritizing viewpoints held across unique groups, QF accomplishes this goal under the assumption that each individual constitutes their own completely unique social group. However, the previous section shows that most individuals are in fact connected/cooperative with others along important social axes. A plural approach to QF, then, should incorporate this social information.

Specifically, our conception of plurality draws on the sociologist Georg Simmel, who saw identity as being defined by one's memberships in social groups. For Simmel, "the groups to which an individual belongs form, so to speak, a system of coordinates, in such a way that each newly added group determines the individual more precisely and more unambiguously" [12]. Our plural algorithm uses mappings between individuals and the (relevant) groups they are members of as input.

Formally, assume we have a set G of groupings of agents. Each element $g \in G$ is a subset of N, and for each $i \in g$, we have a weight $w_{i,g}$ describing the strength of i's membership in g, normalized so that $\sum_{g \in G} w_{i,g} = 1$. Our modification to QF implements a plural perspective by awarding more funding to projects liked by *different pairs social groups*, instead of projects liked by different pairs of individuals (as is the case in normal QF). See algorithm 2 for a technical description.

Algorithm 2: Calculating subsidy amounts under Connection-Oriented QF (CO-QF)

Input: $c_{i,p}$ for $i \in N$ (contributions), G (social groupings), and w (weights in social groups)
Output: `subsidy` (the "raw" amount of subsidy funding awarded to project p)

```
subsidy ← 0;
for g ∈ G do
    for h ∈ G do
        g_sum ← 0;
        h_sum ← 0;
        for i ∈ g \ h do
            g_sum ← g_sum + c_{i,p} · w_{i,g};
        for j ∈ h \ g do
            h_sum ← h_sum + c_{j,p} · w_{j,h};
        subsidy ← subsidy + √(g_sum · h_sum)
return subsidy
```

For each pair of social groups g and h, we check if members of *just one group, but not the other* have contributed to the project. `subsidy` only increases if agents on both sides of this symmetric difference support the project – in other words, if there is *agreement across difference* on the merit of the project (relative to g and h). In this way, projects with support from a plurality of social groups are prioritized.

4 Implementing CO-QF at Gitcoin

Implementing CO-QF at Gitcoin required a choice of G – a set of relevant social groups that would help the platform achieve pluralistic outcomes. After trying a few options (detailed in Appendix D), the platform settled on using the projects themselves as the social groups: G is the set of projects in a given funding round, and if a user i contributes $c_{i,p} > 0$ to a project p, they are placed in p's group with a weight $w_{i,p}$ of $c_{i,p} / \sum_{q \in G} c_{i,q}$. G is reset for each new round. The raw subsidy amounts returned by CO-QF are normalized via the same rules laid out in section 1.

All five external round manager interviewees felt positively about CO-QF, and three specifically noted that CO-QF gave a better signal of the community's desires compared to QF. P2 found that CO-QF was helpful with getting to "*the most accurate depiction of what the total voting pool wanted most*" and "*the best picture of what the community would get the most benefit from*" while sidelining "*votes from single sources or just outside sources that didn't really encompass the true sentiment of the broader group.*" P3 had similar thoughts:

> *"I feel like [CO-QF] gives us this more holistic view of the signal of the community – what does the community really want? ... So for a project with 500 people donating for some potential future [quid-pro-quo reward], who donated to one project ... in QF [that project] would run away with the money. In CO-QF, all of the sudden, we look and say, hmm, something's going on over there, and yes they have a lot of signal, but it's not the signal of this greater community."* – P3

Both of these round managers praised the ability of CO-QF to capture a clearer signal of what their community as a whole wanted. Importantly, neither of them self-identified as knowing about plurality as a normative concept, indicating that they were less biased to assume a plural approach would work better for this purpose. P6, who did identify as familiar with plurality, also felt that CO-QF gave a more trustworthy signal of overall community preferences. Similarly, P4 praised CO-QF's ability to reduce funding for "*projects that ... I kind of knew are well coordinated clusters that maybe would outperform in [QF], but we didn't want them to*." P5 complimented CO-QF's results as feeling "*fair and equitable*", but did not offer an opinion on CO-QF's ability to reflect community sentiment.

CO-QF also had pain points. All interviewees voiced concerns about the opacity of the algorithm and two interviewees were not sure it was appropriate to use donation data to calibrate G. One interviewee felt that CO-QF's tendency to produce drastically more even funding distributions could have a negative impact on the health of the ecosystem as a whole.

5 Conclusion: Using CO-QF for plural AI alignment

This case study covers the context of public goods funding, but CO-QF can be applied in any public decision-making context including AI alignment. Achieving AI alignment and accountability is not merely a question of implementing the "correct" technical solution: instead, it is an evolving process that requires coordination between complex networks of lawmakers, stakeholders, civic agencies and internal groups [10, 9]. We have demonstrated that CO-QF shows promise at delivering clear signals of a community's overall desires, so a developer wishing to understand the most pressing policies to implement in their AI system could run CO-QF both among each relevant group or at the meta-group level, letting individuals spend artificial money to support various policy standpoints. It seems promising that the results would help the developer align their AI system pluralistically with the needs of the community while balancing conflicting interests. [4] At a smaller scale, CO-QF could also be used to resolve annotator disagreements.

On the other hand, while CO-QF shows promise at finding policy choices with support from diverse groups, it is worth remembering that "universal" policy choices of this type may not always be necessary. In a setting where different policy choices can be applied to different groups, it may be more effective to simply let each group choose their own policy.

Adopting a "Simmelian" notion of identity (i.e., identity as an intersection of social groups) was helpful in our context. Of course, our notion of social groups is an abstraction, since from CO-QF's perspective, each group is equal to every other group, but in reality groups themselves form meta-groups and hierarchical structures. However, while we see the benefits to modeling more sophisticated social information, we also urge caution. The act of making diverse social structures "legible" to an algorithm can have adverse effects [11]. Moreover, technical systems that model the world a certain away frequently have the circular effect of inducing their subjects to behave *more like* the agents in that very model [7] or, conversely, may create incentives to undermine those models [6, 8]. In the context of algorithms which use social information, the danger of this type of feedback loop is clear: the act of dividing people into groups could also lead people to identify more with those groups, exacerbating group tensions, or to people suppressing these group affiliations strategically.

While more testing is still needed to validate the plural aspects of CO-QF, current results show that the algorithm constitutes a promising first step towards a robust mechanism for all types plural public decision making, AI alignment included.

[4] For a discussion of CO-QF's relationship to a similar public decision making mechanism called Quadratic Voting, which could also be used for plural AI alignment, see Appendix E.

References

[1] Gary S Becker. “A theory of social interactions”. In: *Journal of political economy* 82.6 (1974), pp. 1063–1093.

[2] Neil Bruce and Michael Waldman. “The rotten-kid theorem meets the Samaritan’s dilemma”. In: *The Quarterly Journal of Economics* 105.1 (1990), pp. 155–165.

[3] Vitalik Buterin, Zoë Hitzig, and E Glen Weyl. “A flexible design for funding public goods”. In: *Management Science* 65.11 (2019), pp. 5171–5187.

[4] Gary Charness and Matthew Rabin. “Understanding social preferences with simple tests”. In: *The quarterly journal of economics* 117.3 (2002), pp. 817–869.

[5] Francis Ysidro Edgeworth. *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. 10. CK Paul, 1881.

[6] C. A. E. Goodhart. “Problems of Monetary Management: The UK Experience”. In: *Monetary Theory and Practice: The UK Experience*. London: Macmillan Education UK, 1984, pp. 91–121. ISBN: 978-1-349-17295-5. DOI: 10.1007/978-1-349-17295-5_4. URL: https://doi.org/10.1007/978-1-349-17295-5_4.

[7] Robert Hurley et al. *The Cybernetic Hypothesis*. Vol. 28. MIT Press, 2020.

[8] Donald MacKenzie and Alice Bamford. “Counterperformativity”. In: *New Left Review* 113 (2018), pp. 97–121.

[9] Jacob Metcalf et al. “Taking Algorithms to Courts: A Relational Approach to Algorithmic Accountability”. In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. FAccT ’23. Chicago, IL, USA: Association for Computing Machinery, 2023, pp. 1450–1462. ISBN: 9798400701924. DOI: 10.1145/3593013.3594092. URL: https://doi.org/10.1145/3593013.3594092.

[10] Inioluwa Deborah Raji et al. “Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing”. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 33–44.

[11] James C Scott. *Seeing like a state: How certain schemes to improve the human condition have failed*. yale university Press, 1998.

[12] Georg Simmel. *Über sociale Differenzierung. Sociologische und psychologische Untersuchungen*. Leipzig: Duncker & Humblot, 1890.

A Conflict Of Interest Statement

Joel Miller is employed by Gitcoin as a part-time contractor, where they implement and help maintain the CO-QF tool used by round managers to calculate funding distributions. Chris Kanich and Glen Weyl have no conflicts of interest.

B Research Methods

Under IRB approval, we conducted semi-structured interviews with one Gitcoin employee and five external round managers. We talked to the employee in order to understand Gitcoin's technical Sybil and fraud defense strategy. We talked to the external round managers in order to understand the benefits and drawbacks of QF and CO-QF. Our interview with the round managers consisted of four main questions, with clarifying questions asked as needed:

1. *What's your relationship with Gitcoin and the surrounding community?*
2. *What do you like and dislike about QF?*
3. *What do you like and dislike about CO-QF?*
4. *What's your familiarity with the concept of plurality?*

We opted to ask broad, open-ended questions about QF and CO-QF so as to not bias interviewees towards thinking any specific issues or benefits were more important. Interviewees volunteered their time. Interviews ranged between 15 and 60 minutes in length. We performed a thematic analysis on interview responses, looking for patterns in interviewee viewpoints on QF and CO-QF. See table 1 for an overview of each participant's role in the Gitcoin community. If a participant has had multiple roles in the community, they are listed chronologically.

Participant ID	Role(s) in Gitcoin Community
P1	Volunteer, Employee
P2	External Round Manager
P3	Grantee, External Round Manager
P4	External Round Manager
P5	External Round Manager
P6	Grantee, External Round Manager

Table 1: Anonymized list of interview participants

Although Gitcoin also runs funding rounds where paid employees act as round managers and assess the outcomes of QF and CO-QF, we opted not to interview any Gitcoin-employed round managers. We made this decision in order to reduce the potential bias of participants, with the idea being that Gitcoin employees may be biased to hold more favorable viewpoints of CO-QF. The one Gitcoin employee we did interview was asked mainly about their experience with security and Sybil defense. We also opted to not interview any current grantees due to the risk that they could be biased towards speaking more positively about QF or CO-QF depending on which algorithm would have given their project more funding (P3 and P6 were grantees in the past, but not simultaneously with their round manager roles). Round managers and their respective rounds served diverse parts of the Gitcoin/ "Web3" ecosystem.

C QF does not fund public goods optimally in the presence of pro-social utilities

First, we will briefly overview the core result of [3]. Focus on the case of one good. Let $v_i(f)$ denote agent i's personal valuation for f dollars of funding to the public good. As in [3], assume that v_i is concave, smooth, and increasing for all i. Let $v(f) = \sum_i v_i(f)$ denote the society-wide valuation function for funding to the public good (note that v is also concave, smooth, and increasing). Here, "optimality" is defined as utilitarian social welfare maximization. The proof of QF's optimality in [3] shows that when every agent donates to maximize their own utility, i.e. when agent i donated so as to maximize

$$v_i\left(\left(\sum_{j\in N}\sqrt{c_j}\right)^2\right) - c_i$$

then QF's unique Nash equilibrium results in a funding level f_p^* such that $v'(f^*) = 1$ – in other words, QF funds the public good exactly until the point where another dollar of funding to the good would not be worth the marginal benefit it gave to society, thereby maximizing utilitarian social welfare.

We model pro-social utilities with the "sympathy coefficient" model first posited in 1881 by Edgeworth [5] and later used and adapted by many other economists, e.g. [1, 2, 4]. Let $\beta \in (0, 1]$ denote the amount that agents internalize the valuation functions of others. Then suppose each agent i has the valuation function

$$\hat{v}_i(f) = v_i(f) + \beta \cdot \sum_{j \neq i} v_j(f)$$

which sums their utility with a β fraction of the utilities of others to reflect a pro-social outlook. To see why QF does not maximize utilitarian social welfare when agents donate using this new valuation function, we can follow the math in section 4.1 of [3]. Agent i's contribution will be chosen to maximize

$$\hat{v}_i\left(\left(\sum_{j \in N} \sqrt{c_j}\right)^2\right) - c_i$$

which will have to satisfy

$$\frac{\hat{v}_i{}'(f)\left(\sum_{j \in N} \sqrt{c_j}\right.}{2\sqrt{c_i}} \Leftrightarrow \hat{v}_i{}'(f) = \frac{\sqrt{c_i}}{\sum_{j \in N} \sqrt{c_j}}$$

by differentiation. Then summing across agents yields

$$\sum_{i \in N} \hat{v}_i{}'(f) = 1 \Leftrightarrow (1 + (n-1)\beta)v'(f) = 1 \Leftrightarrow v'(f) = \frac{1}{1 + (n-1)\beta} < 1$$

Since v is concave, smooth, and increasing, its slope is 1 before it is any constant less than 1. In other words, here QF has overshot the optimal funding level. This theoretical result mirrors the qualms commonly expressed with standard QF in practice, where projects with large coordinated groups of backers are perceived as getting more funding than they deserve.

D Other ways of choosing social groups

Before settling on using donation data to instantiate G and w, Gitcoin tried two different approaches.

First, since many Gitcoin donors use blockchain tools and applications, the platform looked to public blockchain-based sources of data. The platform specifically looked at using data from POAP (Proof Of Attendance Protocol) [5] and Guild [6]. Both of these tools are allow users to publish public, cryptographically verifiable attestations to various social arrangements on the Ethereum blockchain. The chief difference between the two is that POAP tends to be used for logging information related to specific events (i.e., who attended a certain conference), whereas Guild tends to be used to register members of an organization. However, only a small fraction of Gitcoin donors had published any records relating to POAP or Guild on the Ethereum blockchain, so there simply was not enough data to work with.

Next, the platform experimented with using donation data, but in a different way. Within a round, each user was assigned a binary string corresponding to the set of projects they donated to. So under this scheme, in a round with m projects, G is the set of binary vectors of length m, and each user i is assigned to the exactly one group, namely the group

$$(\mathbf{1}\{c_{i,p} > 0\})_{1 \leq p \leq m}$$

where $\mathbf{1}\{c_{i,p} > 0\}$ is an indicator variable that is 1 if i donated to project p, and 0 otherwise.

This scheme was discarded because of a security flaw. A group of coordinating individuals (or Sybils) all aiming to support a project p can easily appear diverse from each other with the following strategy: each agent donates a significant amount to p, and some small amount to a unique set of other projects. Since G has 2^m elements, any group of colluding agents of size less than 2^{m-1} can easily donate such that they are all put in different groups, thereby executing an attack comparable in severity to what is possible under standard QF. In contrast, while CO-QF can still be attacked under the current choice of G, the severity of the worst attack is significantly reduced.

[5] https://poap.xyz/

[6] https://guildprotocol.io/

E A simple extension to plural Quadratic Voting

Quadratic Voting (QV) is a voting mechanism closely related to QF, and the modifications to QF discussed in this paper also directly apply to make QV more pluralistic. QV sees agents spend "voice credits" on an issue, which it converts to effective votes at quadratic costs. So if agent i spends v_i voice credits towards an issue, then the effective number of votes given to that issue is

$$\sum_{i \in N} \sqrt{v_i}$$

Notice that if we let the v_i values denote contributions (i.e., letting $c_i = v_i$), then the above formula can be re-written as the root of the total amount of funding under vanilla QF:

$$\sum_{i \in N} \sqrt{v_i} = \sqrt{\left(\sum_{i \in N} \sqrt{c_i} \right)^2}$$

At first glance, this might seem like a needlessly complicated way to re-write the formula for QV. However, the point is that instead of putting the formula for vanilla QF inside the square root on the RHS of the above equality, we could plug in CO-QF instead. In other words, this way of generalizing QV (as the root of the funding amount given by *some* QF formula) opens the door to the use of CO-QF for voting as well. However, at the end of the day, both QF and QV are very similar algorithms for the same essential context — public decision making — and either mechanism can be used in either context (i.e., a community could quite plausibly vote using QF or fund public goods with QV).