Contribution to Open-Source Product Development

Mohammadi, Mohammad Ali

31 August 2020

Contribution to Open-Source Product Development

Mohammad Ali Mohammadi *

August, 2020

Abstract

The situation of private contribution to an open-source project is considered. This case is distinguished from a simple case of public good provision through the various underlying incentives of the participants. In the modelling, these incentives come into play to form various equilibria from which conclusions on their roles can be derived. Specifically, it is shown that the agent's willingness to participate, her own need from the project, and the community's altruism play important roles in driving the project forward.

Keywords: Open-source, contribution, altruism, simultaneous game

JEL Codes: L17, D64, C70

1 Introduction

The issue of contribution to open-source was perhaps one of the more confounding issues that an economist could observe. To quote Tirole on the subject, "*To an economist, the behavior of individual programmers and commercial companies engaged in open source projects is initially startling*" (Lerner et al. 2002). Fortunately, this very work was perhaps the inception of an effort to analyze the situation using the toolbox of economics.

*ma.mohamadi.ut@gmail.com

Since then, various works have tried to identify incentives of individual programmers in investing in open-source projects, projects that potentially take significant time and effort to contribute to, yet, the individual is not always compensated, at least not monetarily, for this contribution.

The situation is thus immediately reminiscent of a public good problem where contributions are private. However, in spite of the literature on public good widely indicating the inefficiency of private contributions in this case, open-source projects are often-times of sufficient maturity and quality, in many cases even surpassing their proprietary alternative in features, stability, and usability. Therefore, it would appear that private provision of a public good is only part of the mechanism at play here. As agents are willingly substituting paid endeavour with voluntary contributions, sufficiently enough to yield the situation just described, there may be other mechanisms at play as well.

Nevertheless, studying the issue is of paramount importance, not simply because a careless analysis of it would contradict the entire literature on public good and the reason for the existence of government, but also because it would seem that its mechanisms are quite strong, the employment of which in other scenarios could yield the same high social benefits, perhaps. Moreover, one needs to know whether the situation is stable, that is, driven by fundamental incentives that are unlikely to change as the state of the world would largely do. Furthermore, by understanding the inner workings of this situation, it may be possible to offer insights for improving this already beneficial case.

In this work, after reviewing the related literature on the economics of open-source, I review the situation of contributing to an open-source project. Afterwards, I offer a simple model of contribution to open-source, offering various insights on the various incentives and mechanisms at play. In the modelling, I try to be as close to the real situation in the open-source community in order to distinguish the problem from a simple problem of public good provision. The agent's specific benefit from the project, her various gains from the contribution effort including expertise gain, and gains from altruism, are identified by the modelling.

2 Literature Review

There has been no shortage of attention paid to the subject of open-source from various perspectives. However, most of the discourse comes from a managerial perspective, if not outright from a software engineering one that tries to understand the phenomena. Nevertheless, it is clear that the subject is best studied in the field of economics where agents' intrinsic incentives are involved. In spite of that, the study of the subject in economics has been quite limited. Most of the related published works in economic journals are surveys of contributor motivations. An example of such works is David et al. 2008 which finds that experience gains, possibility of monetary payment (for instance, through donations), education attainment etc. are among the motivations of a contributor to open-source. Moreover, the work claims there is strong evidence of heterogeneity among the agents, which it analyzes by clustering the contributors into various groups, therefore, indicating that models of open-source viewing the software as a whole, may be contradicted by various empirical findings.

As mentioned, however, there are presently only a few theoretical models of open-source contribution to begin with. Moreover, most of these works are, in fact, models of private contributions to public good rather than models incorporating aspects of open-source contribution specifically. Perhaps the earliest game-theoretic model of open-source contribution is the work of Justin Pappas Johnson 2002. The novelty of this early work is only in that it is the first of its kind to consider open-source contribution in the framework of a public benefit provision. As stated before, however, this means that the results of the work are not in fact unique to open-source contribution.

A brief statement of the underlying model of this work is as such: There is only one potential enhancement to the software which is valued at v_i by the agent and whose cost of development would require c_i investment. The agent assumes a probability of π^i for the development happening when she is not the one contributing. Therefore, her strategy is her probability of making the development herself, ψ_i, in a mixed strategy Bayesian Nash equilibrium (since values are private). Consequently, as there are no other elements in the payoffs, only the ratio of value-to-cost will matter, quite reminiscent of the "reporting a crime" game. In fact, this way of modelling leads only to similar strategies, equilibria and conclusions, albeit with private v/c ratios for each agent. For instance, with added agents (more potential contributors), the probability of contribution similarly declines. The paper, however, tries to argue that this decline approaches zero rather rapidly. Nevertheless, the observations of the model can be stated in a handful of scenarios unrelated to open-source.

With this kind of modelling, one has no choice but to see the difference between open-source contribution and general private contribution to public good, in the lower costs or higher values of contribution that exist. That is, the discourse usually following such results is that, for instance, valuations are higher since there are experience gains as well, or that costs are lower since these are net costs and additional benefits are included, thus explaining the higher levels of contribution to a public good in the case of open-source. This line of reasoning is not very informative. Firstly, one needs to see explicitly the consequences of a decomposition of the valuations or costs into the speculated terms. Secondly, such an argument does not explain the *dynamics* of open-source, whereas it is not just one-time contributions that matter, rather, how an open-source project can reach maturity levels beyond that of proprietary alternatives.

The first issue is slightly mitigated in the work of Bitzer et al. 2007. The modelling of the paper is rather simple. Possible sources of contributor motivation are collectively modelled inside cost. Afterwards, both infinite and finite time scenarios are considered, with time being continuous and individuals discounting the future. When the open-source alternative is introduced, its private value is added to the value of using the proprietary alternative. Consequently, depending on the specific valuation, there are agents who would never want to contribute regardless of the strategic interactions; agents whose discounted benefit of using the open-source product is less than their discounted cost of contribution. This leaves the problem with a finite number of agents.

Various benefits of contribution are modelled in the decomposition of the cost term into a general effort cost c, a "play" value of making the contribution p, and a "gift" value g.

The play value is any experience or practice gains from making the contribution. The gift value can mean a number of things, all having the effect of lowering costs. Most important interpretations of the gift value include either increased social status of ”gifting” to the open-source community, or ”altruism”, whether simple, in that contributors to open-source are simply more altruistic than contributors elsewhere, or reciprocal altruism, meaning that the contributor makes the contribution in the hopes of another contributor ”returning the favor”.

The paper then assumes all parameters to be common knowledge, stating that without the option of waiting, that is, in a simultanous one-time choice, this would be simply a ”chicken” game where eventually any one of the agents may pay and the rest free-ride. By allowing delay in making the decision, it may be possible to identify the agent yielding sooner and thus making the contribution, ending the game. In the infinite time, similar to the static problem, anyone may volunteer, as the waiting option exists forever. In the finite time, however, once the game ends, no further benefits can accrue. The idea is then to consider the time beyond which an agent wouldn't contribute in any case and then consider the largest of said time points. Just before that point in time, the agent with the largest of such time points finds it optimal to finally yield and contribute, since there is no one after her who would. At previous points in time, all other agents understand this outcome. Therefore, none of them will make the contribution. Given that time is discounted, this agent will contribute immediately.

Through an identification of the agent with the largest time point, the paper then proceeds to makes ceteris paribus conclusions on the role of each of the underlying motivations, finding that, higher gain from software, larger gift benefit, longer finite time horizon (interpreted as being younger), lower effort cost and higher play values, all increase the possibility of the agent being the one making the contribution. The intuition is simple: The agent who is still willing to contribute to the project after so much time has passed and the value of contribution has reduced so much, must incur the largest benefit from the open-source and contributing to it.

Many criticisms of the work Bitzer et al. 2007 come to mind. The assumption of perfect information in the work is perhaps too strong. There are certainly aspects of uncertainty and imperfect information even in the rather open, open-source community. Moreover, as stated before, this work too, only considers single contributions, still ignoring part of the dynamics of problem. Most importantly, while the work manages to show a decomposition of the cost, there still is little interaction between these parts and the overall game. That is, they only matter as far as reducing the cost term does, but not in any interactive way with the rest of the game or other agents. That is to say, the work is still not as strongly applicable to the issue of open-source contribution, rather than general private contribution in the presence of waiting option.

The issue of open source dynamics is studied in the work of Athey et al. 2014. This work seems to have a better incorporation of the unique aspects of open-source, however, its game-theoretic framework is perhaps too simplistic, dealing only with myopic agents with specific *needs* from the project, making independent decisions based on per-period payoffs and the state in which they are. The idea of agents with needs arriving to a project is more consistent with the real world scenario, as will be explained in the next section. On the other hand, the work has strong probabilistic foundations, incorporating a Markov process for the evolution of altruism, and ultimately, deriving open-source dynamics from such evolutions. A criticism of this work is that it only characterises the outcome, given the initial state, through employing a dynamic system approach, thus missing any discussion on the specifics of the interactions between the agents.

Other works exist on the economics of open-source as well. The work of Lerner et al. 2002, has more of a big-picture view of the issue, focusing on many aspects of open-source that are not necessarily related to contributor incentives, such as private firm's relation with the open-source communities and projects. The work of Justin P. Johnson 2006 ascribes part of the success of open-source to the ease of reviewing a peer's work, where through lower costs of information disclosure, it is shown that the open-source approach leads to greater social surplus.

Other works not directly related to the discourse, whose underlying ideas, nevertheless, have been considered, include Economides et al. 2006 which discusses the competition between open-source and proprietary software in greater detail. Additionally, the work of Dufwenberg et al. 2017, though mostly a purely theoretical work, offers insights on the modelling of altruism. Moreover, the work of Makris 2009 indicates a scenario where private provisions to a public good may not actually be inefficient, given the right assumptions.

3 Background

Before proceeding to the details of the model itself, it is perhaps useful to review a rough sketch of the process of open-source contribution. The inception of the project can safely be assumed to be random: A simultaneous public good game where one of the people in need of the project finally contributes the code basis. This is quite consistent with the real-world situation, as the initial code quality differs widely depending on which of the potential agents finally chose to submit the code (for instance, sometimes the documentation quality of projects addressing the same needs, or their capabilities, differ in the submitted initial code bases). Alternatively, one can consider the reasoning of Bitzer et al. 2007 to find the initial contributor and thus the initial code quality. Nevertheless, further contributions are what will determine the true dynamics of the project. Consequently, an agent arriving to a project of present quality q_t, is faced with a decision to contribute further to the project or not.

In a more complicated situation, this *potential contributor* is faced with this decision, in the presence of a myriad of other present contributors, alternatively named as *active contributors*, and other potential contributors as well. In what follows, I will consider a game being played between only one active contributor and one potential contributor, and the rationale will be discussed. Nevertheless, among the general motivations for the potential contributor are, first and foremost, need N for specific improvements in the project, the ”play” value of contribution, social interaction and improving social status among peers, profile building, experience gains, gift benefits (as previously discussed), and among the deterrents are lack of expertise, unfamiliarity with the code base, various constraints, and most importantly, free-riding incentives and uncertainty. Accordingly, given her own state and the state of the project, a potential contributor will decide whether to contribute or not.

The emphasis on the specific need of a potential contributor is what distinguishes this work from the rest of the literature. Even in the work of Athey et al. 2014, this need is not as crucial in determining the strategies or outcomes. Rather, in their work, need is only what determines the agent's Poisson arrival process, which, while making sense, still means that an agent's need do not play any role beyond her participation in the game. Need is an important distinction in the open-source environment precisely due to how it is responded to. An agent seeking the addition of a new feature to the software, a bug fix in the present code base, or any general enhancement like improved documentation of the code or usage guide of the software, will need to inform the active developers of the issue. ”Issue” is in fact the precise name given to all types of user needs in the open-source community. What advances the project as a whole is having more fixed issues and what each active developer does, after perhaps having worked on the code basis, is to solve more and more issues, or identify new issues in the project. As explained, issue, contrary to what the name implies, does not have to mean a ”fault” in the software that needs to be fixed. Its best interpretation is exactly a need that the project, at its current state, fails to answer.

Consequently, the rationale for considering the game as only being played between one active developer and one potential developer is as follows. Firstly, one should ask why the potential contributor has shown willingness to contribute in the first place. The most likely scenario is not that the contributor obtains simply an overall benefit from the project, as in, benefiting from roads or clean air, for instance. Rather, the more logical scenario is an initial *need* of the potential contributor that has drawn her to the project in the first place. Most contributors to open-source seem focused on contributing to projects that fall under the same specific category (for instance, it is quite rare to see a contributor to computer vision repositories to contribute to signal processing repos, or a contributor to video players projects to contribute to video editing software or vice versa, even though the expertises may sometimes align while benefits almost always align.)

Secondly, the specific need of the potential contributor makes it sufficient to consider only the marginal active developer. This is due to the process by which an agent expresses her need to the community, as briefly mentioned before. The detail is that once the agent "opens" a new issue in the respective repository, she will describe her problem in detail, and interested agents (whether active or potential) will discuss in the same topic about possible solutions to the problem. Eventually, only a few people will remain either with the expertise or willingness to solve the problem. Accordingly, what is meant by the marginal active developer, is the developer whose area of expertise and attention is close to the agent. An important note here is that through this process, the actual cost of solving the problem is discovered. If expertise exceeds the difficulty level of solving the problem, this cost will be the same for all agents, as, for instance, the precise location in the code that is causing the bug and the solution method have already been discussed and any capable developer will likely only need to put in the work to solve the issue. Secondly, the common value of the issue to all agents is also discovered by a similar argument, though its private value for the agent in need may be higher.

The question is then, can we find a mechanism where a potential contributor contributes in the hopes of having her own problem solved? In other words, can she expect a scenario where her contribution advances the project in a way that her problem too, is resolved? Accordingly, two approaches will be possible: Either crossing a threshold in project quality solves her problem as well, or perhaps, through coordination with other agents, a more capable agent will be willing to solve her problem. This is the main mechanism that is searched for in this work.

4 The Model

The model is as follows: A potential contributor, p, with specific need, or problem, p_p, and ability, or expertise, e_p, exists. There is a function $P : E \mapsto \{P\}$, mapping a value of ability to all problems that can be solved by possessing said ability. Assume a finite number of distinct problems related with a project. A sorting on this set can be done through the function P, where $p_i > p_j$ if and only if $p_i, p_j \in P(e_2)$ for some e_2 while $p_i \notin P(e_1)$ such that $p_j \in P(e_1)$ for some $e_1 < e_2$, which is simply an attempt at stating that some problems are harder than others and that each problem can be expressed with a real value. Calling the function that takes any e to the hardest problem that can be solved with it as defined above, $p : E \mapsto P$, we can assume that the agent that tries to solve any problem $p_k > p(e)$ receives a very high negative payoff, denoted by $-c_K$. On the other hand, the payoff of solving any problem within one's ability is $v(p_k) - c(p_k)$, where $v(.)$ and $c(.)$ are the value and cost functions. These are continuous function that should take their input as the real value assigned to each problem, based on its difficulty.

The active contributor is of similar endowments except that, since she is actively involved with the project itself, at least presently, she does not have a problem of her own that she wants to fix. Her ability is denoted by e_a. A trivial case is when the potential contributor is capable of solving her own problem, that is, $p_p < p(e_p)$. Ignoring this case, and recalling the discussion on focusing the attention to the marginal active contributor, assume $p_p < p(e_a)$ while $p_p > p(e_p)$. That is, the active contributor is capable of solving the potential contributor's need without great penalty while the same is not true for the potential contributor.

Additionally, we have to give some basic properties of the value and cost functions. As is conventional, valuations are diminishing and so the function is concave, while the cost function is convex. Both functions start from zero, that is, no contribution bears neither cost nor benefits. This leads to the two functions intersecting at some point K as seen in the below figure:

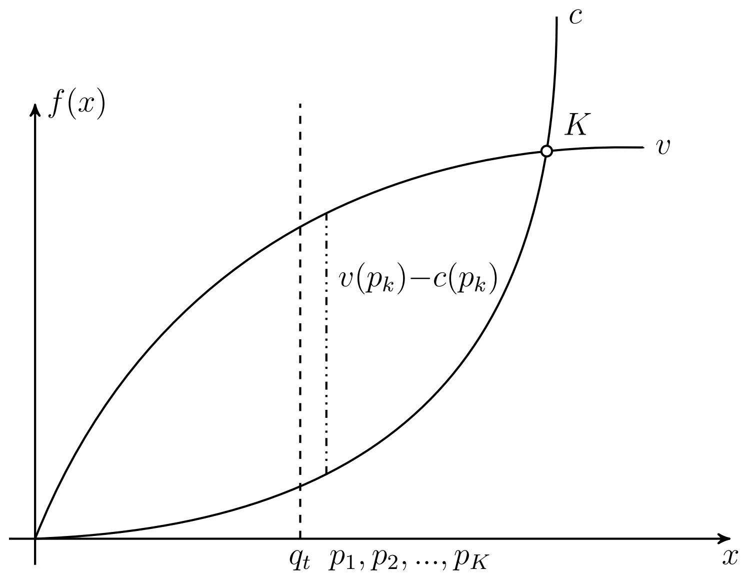

Figure 1: The shape of the value and cost functions as well as the position of present project quality and available associated problems.

This intersection helps to further limit the number of possible problems that can be solved, as solving a problem beyond p_K bears necessarily negative payoff. Moreover, this means that the highest possible value gain from the project will be $v(p_K)$. This is consistent with the idea of a project reaching full maturity and future contributions in this saturated stated not being worth the effort.

The present state of the project q_t will have a decisive role in the discussions to follow. In fact, it is an additional condition, or requirement, that is always assumed to hold through the analysis:

Assumption 1. *The project is in a state of sufficient maturity, q_t, such that $v(p_j)-c(p_j) < v(p_i) - c(p_i)$ for $p_j > p_i$.*

By the mean value theorem, it is guaranteed that at some point $x > 0$, we will have $v'(x) = c'(x)$, as both functions start at zero, and one is concave while the other is convex. This is precisely the point after which contribution payoff is decreasing, thus all project states after this point satisfy this assumption.

The next section provides a motivating scenario for the analysis at hand.

4.1 A Motivating Scenario

Consider a potential contributor with problem p_2 with the usual constraints discussed before. Assume the problems of the project, given the constraints, are only p_1 and p_2 where the active contributor is capable of solving both. Then, a simple model would be to consider the interaction as a simultaneous-move game with actions $\{NC, p_1, p_2\}$ representing no contribution, solving problem p_1, or solving p_2 respectively. For the potential contributor, the value of having her problem solved is possibly the greatest value that she can obtain from the project besides other factors of project state. Denoting this by v_K, c_K is then the possibly infinite cost of trying to solve the issue herself, which will ultimately fail as well. The K index represents the highest conceivable value or cost associated with a project, though I have not denoted it by p_K specifically, to avoid a discussion on these end values and simply assume them to be quite large in general. Another important case is when both developers work on the same problem. In this case, no coordination beyond cost and value discovery is assumed; as is often the case in software development, there is no cost sharing in solving a single problem (see Brooks, "The Mythical Man-Month"). Simply, two different solutions (perhaps merely through the different coding styles) are developed for the same problem.

4.1.1 Nash Equilibria

With the discussion above, the payoff table will be as below:

a \ p	NC	p_1	p_2
NC	$0, 0$	$v(p_1)$, $\underline{v(p_1) - c(p_1)}$	$0, -c_K$
p_1	$\underline{v(p_1) - c(p_1)}$, $\underline{v(p_1)}$	$v(p_1) - c(p_1)$, $v(p_1) - c(p_1)$	$\underline{v(p_1) - c(p_1)}$, $v(p_1) - c_K$
p_2	$v(p_2) - c(p_2)$, v_K	$\underline{v(p_1) + v(p_2) - c(p_2)}$, $\underline{v_K + v(p_1) - c(p_1)}$	$v(p_2) - c(p_2)$, $v_K - c_K$

Figure 2: Payoff table for a simultaneous game between the active and potential contributor.

We can see that there are in fact two equilibria of the problem. In fact, both of these equilibria are likely played in the real world. One equilibrium is having the potential developer not engaging with the community and only benefiting from the small contributions that active developers make, while her problem remains unsolved. In a situation where time is at play as well, this likely would mean that her problem too, is eventually addressed, as with the easier problems having being solved, it then would be optimal for the active agent to contribute to solving further problems. It is simply the case that this solution may come much later and at decreased values. The second equilibrium represent a case where the potential contributor has engaged with the community, offering to solve p_1, in the hopes that her own problem, p_2, gets solved by the more adept active developer. Clearly, payoffs from this equilibrium are larger for both players.

In fact, the two equilibria indicate the two kinds of potential contributors that exist.

It's only that one kind prefers free-riding while the other kind is in fact willing to engage with the community. The modelling of why she would be willing is a whole different study, however. Nevertheless, a question that arises from the analysis, is can the inefficient equilibrium be ruled out? The answer is yes. We only need to focus on the second kind of potential contributor. Her engagement with the community can be modelled as a stage preceding an actual problem assignment game, where the potential contributor is simply deciding whether she wants to contribute or not. The tree of this game is given below:

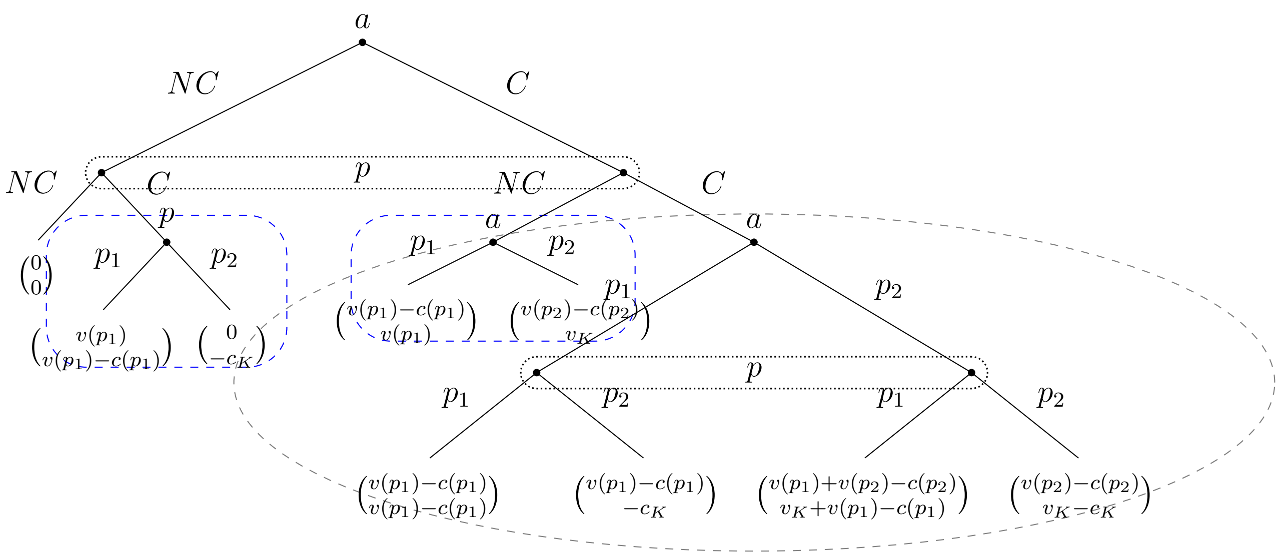

Figure 3: The game as played between an interested potential contributor and the active developer.

As is shown, there are three subgames besides the main game. The solutions of the two smaller subgames are clear. The solution to the right-most subgame is given in the below table:

p

		p_1	p_2
a	p_1	$v(p_1) - c(p_1)$, $\underline{v(p_1) - c(p_1)}$	$\underline{v(p_1) - c(p_1)}$, $v(p_1) - c_K$
	p_2	$\underline{v(p_1) + v(p_2) - c(p_2)}$, $\underline{v_K + v(p_1) - c(p_1)}$	$v(p_2) - c(p_2)$, $v_K - c_K$

Subgame 1

Figure 4: Subgame of the problem assignment from the main game in figure 3.

Replacing these three subgames with their respective payoffs, another simultaneous decision between contributing or not, remains, the solution of which is given below:

a \ p	NC	C
NC	$0, 0$	$v(p_1)$, $\underline{v(p_1) - c(p_1)}$
C	$\underline{v(p_1) - c(p_1)}$, $v(p_1)$	$\underline{v(p_1) + v(p_2) - c(p_2)}$, $\underline{v_K + v(p_1) - c(p_1)}$

Main game

Figure 5: The game in figure 3 with subgames payoffs substituted.

By informing the agent of the consequences through the subgames, it has been possible to rule out the pure free-riding equilibrium. This is possible due to the engagement of the contributor with the active contributor, before considering the payoffs of solving either of the problems or not contributing at all.

4.2 The General Case

It may seem that the original mechanism that I have been searching for, by which a potential contributor contributes in the hopes of having her own problem solved, has been found. This is not true, as I have only considered a project with two possible problems only. It is important to note that contribution is a dominant strategy except when free-riding or solving the same problem. It is just that contributing to the easier problem is also preferred. This can lead to a scenario where agents' needs are rarely met. In the below table, let us consider a more general case, where four problems are possible in the project. The choice of four problems is so that the agent's problem p_3, and her highest possible contribution, solving p_2, lie between a range of problems, which seems to be quite general enough (even though the discourse can be done quite easily in a general case of K problems, it would likely lack any further insights.)

a \ p	p_1	p_2	p_3	p_4
p_1	$v(p_1)-c(p_1)$, $v(p_1)-c(p_1)$	$\underline{v(p_2)+v(p_1)-c(p_1)}$, $\underline{v(p_1)+v(p_2)-c(p_2)}$	$\underline{v(p_1)-c(p_1)}$, $v(p_1)-c_K$	$\underline{v(p_1)-c(p_1)}$, $v(p_1)-c_K$
p_2	$\underline{v(p_1)+v(p_2)-c(p_2)}$, $\underline{v(p_2)+v(p_1)-c(p_1)}$	$v(p_2)-c(p_2)$, $v(p_2)-c(p_2)$	$v(p_2)-c(p_2)$, $v(p_2)-c_K$	$v(p_2)-c(p_2)$, $v(p_2)-c_K$
p_3	$v(p_1)+v(p_3)-c(p_3)$, $\underline{v_K+v(p_1)-c(p_1)}$	$v(p_2)+v(p_3)-c(p_3)$, $v_K+v(p_2)-c(p_2)$	$v(p_3)-c(p_3)$, v_K-c_K	$v(p_3)-c(p_3)$, v_K-c_K
p_4	$v(p_1)+v(p_4)-c(p_4)$, $\underline{v(p_4)+v(p_1)-c(p_1)}$	$v(p_2)+v(p_4)-c(p_4)$, $v(p_4)+v(p_2)-c(p_2)$	$v(p_4)-c(p_4)$, $v(p_4)-c_K$	$v(p_4)-c(p_4)$, $v(p_4)-c_K$

Subgame with four problems

Figure 6: The subgame of a game similar to the one in figure 3, with four possible choices of contribution for the two agents.

As can be seen from the figure, neither agent is willing to play by (p_3, p_2). This is especially an interesting behavior from the potential contributor, as her payoff would be higher than either of the two equilibria. Moreover, it is only her free-riding incentive that pushes her from this outcome to the equilibrium to the left of it. As for the active contributor, it would seem that her main concern is solving the easiest issue first. Consequently, I need to consider another aspect of the mechanism that seems to be missing from the present modelling, one by which the potential contributor can hope to have her problem solved by expressing her willingness to contribute more.

4.3 The Role of Altruism

Consider a game where the potential contributor can express the least amount that she is willing to contribute to the project, before proceeding to the actual problem assignment subgame. In other words, she will state that she is not interested in solving problems that are easier than a threshold in the subgame that follows. Let us call this threshold $\beta(e_p, p_p)$, meaning that, among other factors, this threshold will depend both on her own ability (as far as choosing which problems to solve go), and the actual problem that she is willing to solve (representing the difficulty of solving her problem). With this indication, she and the active contributor enter the problem assignment subgame. The tree of the game is shown below:

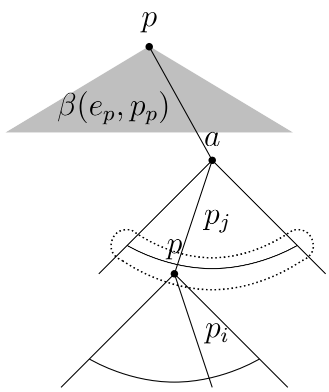

Figure 7: A game where the potential contributor first expresses her least contribution interest.

Notice that once in the subgame, the set of problems that the potential contributor can choose has reduced towards harder problems. There is a commitment issue here, therefore, alternatively, assume large social disutility by contributing any less, for instance, through being seen by the community as unreliable. Nevertheless, in the first stage the agent is free to avoid any subgame as she desires in any case so limiting her options in the subgame should not pose a great issue.

Additionally, one must note that without a new mechanism, the agent's indication is pointless and is similar to the case of playing the game simply twice in a row: In any subgame, the best response of the active agent to any $p_j \neq p_1$ by the potential contributor, is simply p_1 as she obtains the fixed value of $v(p_j)$ anyways, while assumption 1 ensures she chooses the first and easiest problem. If $p_j = p_1$, she will simply choose p_2 as contribution is dominant. Therefore, any equilibrium of any of the subgames either has the active agent contributing p_1 or p_2. In such a scenario, the potential contributor is only deciding on her own added contribution, which, through similar arguments, leads her to choose whichever of p_1 or p_2 that has not been chosen by the active developer. This would imply slow project growth, as previously discussed, which is contrary to the state of many open-source projects.

What's missing from the analysis is the reciprocal altruism that follows the potential contributor's indication. In reality, an active contributor may view the project as partly her property, therefore, making an indication by an outside agent to contribute to it, a sort of altruism. There are a handful of ways to model this altruism, the most appropriate one of which appears to be gift exchange. As mentioned in the literature review, gift exchange can mean a variety of things. Here, it is perhaps appropriate to take the term quite literally: The potential contributor is indicating that her gift will at least be of a certain quality, while the active developer wishes to return this kindness in the form of a gift of her own.

Accordingly, let us add a new term representing the gift exchange effect. I do not need to model this effect in the potential contributor's utility as soon will be shown, since she is already motivated by solving her own problem. The representation of this effect in the active contributor's utility is by the term $g(p_i, p_j, \alpha)$ where the first argument represents own contribution and the second term represents the other agent's contribution. Moreover, it is such that $g(p_i, p_j) > 0$ for $p_i > p_j$, and $g(p_i, p_j) < 0$ for $p_i < p_j$, and $D_\alpha g(p_i, p_j, \alpha) > 0$. The parameter α can be viewed as inherent altruism of the active developer. Additionally, it is assumed that the gift value is concave in own contribution, that is $D_{11}g(.) \leq 0$, and increasing. With this term added, we can try to argue whether an equilibrium exists, where the agents play any strategies except p_1 and p_2. Before proceeding, two assumptions are due:

Assumption 2. *Own contribution in the gift value is still sufficiently important at the inception of the exchange, where project status is q_t. That is, $g'(q_t; p_j) > |v'(q_t) - c'(q_t)|$ for all $p_j \in P$.*

Recalling that all $q_t > x$ where $v'(x) = c'(x)$, satisfy assumption 1, this condition is equivalent to saying that a change in sign in the derivative of the utility of the active developer, $v'(x) - c'(x) + g'(x; p_i)$, is due, indicating a possible maximum.

Assumption 3. *Potential contributor's payoff from having her problem solved, v_K, exceeds all combinations of benefits and costs. That is, $v_K - c(p_K) > v(p_K)$, simplified as $v_K > 2v(p_K)$.*

Recalling the distinction between v_K and c_K, and $v(p_K)$ and $c(p_K)$, there should be no ambiguity here. This statement simply means that the potential contributor prefers the outcome of having her problem solved, to any other possible outcome of the game, if presented with the option of choosing.

Proposition 1. *With the preceding assumptions, an equilibrium may exist where the agent's problem is solved while she contributes a value $p_i \neq p_1, p_2$, depending on the altruism level of the active contributor.*

Proof. Consider the best response of the active developer after having entered the subgame that is induced by the potential contributor choosing a β_p, to an action of the potential contributor, $p_i \geq \beta_p$. As the functions are smooth and continuous, and with assumption

2, for any p_i, there exists a p_i^*, such that $v'(p_i^*) - c'(p_i^*) + g'(p_i^*; p_i) = 0$ (since the overall expression is concave and with the assumption mentioned, has a maximum that is yet to come). As problems were assumed to be finite, then the best response is choosing the problem that is closest to this value.

With this situation, it is clear what the potential contributor has to do. She has to choose p_i such that the closest value to p_i^* is in fact, her own problem, p_p. In other words, we have:

$$
\begin{aligned}
&p_i \to p_i^* : v'(p_i^*) - c'(p_i^*) + g'(p_i^*; p_i) = 0 \\
&(p_i, p_i^*) : if \ |p_i^* - p_p| \leq |p_i^* - p_j| \ \ \forall j \\
&\qquad\qquad then, \quad \beta(e^p, p_p) = p_i
\end{aligned}
$$

That is, each of the problems that the potential contributor can solve, induce a value for which the active contributor finds contributing precisely by that value (though possibly unavailable in the problems set) optimal. Then, the potential contributor, knowing all pairs of available problems and their induced values, will choose whichever problem that would lead to the active contributor choosing the potential contributor's problem, p_p, to solve (that is, the active contributor finding p_p closer to the induced value and thus the problem choice that maximises her payoff). Clearly, there is no deviation from this outcome for the active agent as her utility is being conditionally maximised. Moreover, because of assumption 3, the potential contributor prefers this outcome to any other outcome as well, thus, maximizing her own payoff, indicates a willingness to contribute to at least $\beta_p = p_i$ as obtained from the above strategy. From assumption 1, this is in fact the contribution that she will make, not deviating in the subgame either. □

The phrase "may exists" is present because whether such a contribution is feasible for the potential contributor will depend on the level of altruism of the active developer. If she is quite highly altruistic, then she will be willing to solve even harder problems for less contributions, which might be where the potential contributor is constrained to, due to her ability, e_p. Similarly, if the active contributor is too altruistic, the potential entrant may be able to convince her to solve her problem even with the previously low contribution levels, p_1 and p_2.

It is clear that the assumptions made are only sufficient and an equilibrium as described may exist under weaker conditions as well. However, as I have forgone specifying the actual function forms of the various terms, my goal has been to only characterise some of the properties loosely, and to make clear the overarching workings of the problem.

4.3.1 Welfare Dynamics

It is interesting to make a note on welfare here. Assuming that altruism decays (as is done so in the work of Athey et al. 2014), it is interesting to note that with the above interactions, it is possible for a project to advance beyond levels that would be otherwise possible. That is, assume that utility gain from gift exchange from this period vanishes completely for the active developer in the next period. Nevertheless, potential developer's problem has been solved which increases her utility while the cost of its development had been offset by the gift exchange process in the previous period. Moreover, further people using the project now have access to a project of higher quality, at least more immediately now. Whereas, had the active developer been on her own, the project would have only advanced by resolving the simplest problems instead, resulting in inferior social surplus.

4.3.2 The Role of Play Value

While I ignored the role of play value, it is clear that it can make contributing to harder problems rational, as the cost of contributing the harder problem is offset by the gain from simply having worked on it, thus, likely requiring even weaker conditions on altruism to lead to the same equilibrium as previously described.

4.3.3 The Role of Uncertainty

Perhaps the most important extension that can be made to the work would be uncertainty. Specifically, there is uncertainty on how altruistic the active developer is going to be. Higher project status could be indicative of a higher α in the gift exchange term, thus requiring conditioning on project quality. As explained before, due to the process through which problems are discovered, it is quite unlikely that there is uncertainty regarding values, costs, or abilities.

5 Conclusion

In this work, I introduced both a different modelling of the open-source contribution problem as well as some slight hints on how to improve its various aspects. The main conclusions of the work indicate the important role that reciprocal altruism, as well as individual and specific contributor need play in the development process of a project. By focusing on contributions that are not one-timed or sufficient, rather, by assuming contribution to be dominant in most cases, I have shown how the interaction between the agents around a project matter in advancing it beyond what is expected of it in similar situations. I have tried to make assumptions that are quite unique to open-source, distinguishing it from a general scenario of private contribution to a public good. Consequently, the results seem more in line with what is observed in various open-source projects and communities. One interesting extension would be in the veins of Justin P. Johnson 2006, by considering the procedure of value and cost discovery in the model as well, thus indicating even larger social benefits. Other possible extensions include incorporating the uncertainty on the developer's altruism, incorporation of potential contributor altruism, a more sophisticated discussion on play value by specifically including its effect in the modelling, and a modelling of the more intricate roles of initial project quality.

References

[1] Susan Athey et al. "Dynamics of Open Source Movements". In: *Journal of Economics & Management Strategy* 23.2 (2014), pp. 294–316.

[2] Jürgen Bitzer et al. "Intrinsic Motivation in Open Source Software Development". In: *Journal of Comparative Economics* 35.1 (Mar. 1, 2007), pp. 160–169.

[3] Paul A. David et al. "Community-Based Production of Open-Source Software: What Do We Know about the Developers Who Participate?" In: *Information Economics and Policy*. Empirical Issues in Open Source Software 20.4 (Dec. 1, 2008), pp. 364–398.

[4] Martin Dufwenberg et al. "Reciprocity Networks and the Participation Problem". In: *Games and Economic Behavior*. Special Issue in Honor of John O. Ledyard 101 (Jan. 1, 2017), pp. 260–272.

[5] Nicholas Economides et al. "Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry". In: *Management Science* 52.7 (July 1, 2006), pp. 1057–1071.

[6] Justin P. Johnson. "Collaboration, Peer Review and Open Source Software". In: *Information Economics and Policy* 18.4 (Nov. 1, 2006), pp. 477–497.

[7] Justin Pappas Johnson. "Open Source Software: Private Provision of a Public Good". In: *Journal of Economics & Management Strategy* 11.4 (2002), pp. 637–662.

[8] Josh Lerner et al. "Some Simple Economics of Open Source". In: *The Journal of Industrial Economics* 50.2 (2002), pp. 197–234.

[9] Miltiadis Makris. "Private Provision of Discrete Public Goods". In: *Games and Economic Behavior*. Special Section of Games and Economic Behavior Dedicated to the 8th ACM Conference on Electronic Commerce 67.1 (Sept. 1, 2009), pp. 292–299.