Designing Refund Bonus Schemes for Provision Point Mechanism in Civic Crowdfunding

Sankarshan Damle,
Moin Hussain Moti
International Institute of Information Technology
Hyderabad, India
{sankarshan.damle,moin.moti}@research.iiit.ac.in

Praphul Chandra
KoineArth
Bangalore, India
praphulcs@koinearth.com

Sujit Gujar
International Institute of Information Technology
Hyderabad, India
sujit.gujar@iiit.ac.in

ABSTRACT

Civic crowdfunding (CC) is a popular medium for raising funds for civic projects from interested agents. With Blockchains gaining traction, we can implement CC in a reliable, transparent, and secure manner with smart contracts (SCs). The fundamental challenge in CC is free-riding. PPR, the proposal by Zubrickas [23] of giving refund bonus to the contributors, in the case of the project not getting provisioned, has attractive properties. However, as observed by Chandra et al. [10], PPR faces a challenge wherein the agents defer their contribution until the deadline. We define this delaying of contributions as a race condition. To address this, their proposal, PPS, considers the temporal aspects of a contribution. However, PPS is computationally complex, expensive to implement as an SC, and it being sophisticated, it is difficult to explain to a layperson. In this work, our goal is to identify all essential properties a refund bonus scheme must satisfy in order to curb free-riding while avoiding the race condition. We prove Contribution Monotonicity and Time Monotonicity are sufficient conditions for this. We propose three elegant refund bonus schemes satisfying these two conditions leading to three novel mechanisms for CC - PPRG, PPRE, and PPRP. We show that PPRG is the most cost-effective mechanism when deployed as an SC. We show that under certain modest assumptions on valuations of the agents, in PPRG, the project is funded at equilibrium.

1 INTRODUCTION

Crowdfunding is the practice of funding a project by raising voluntary contributions from a large pool of interested participants and is an active area of research [5, 6, 17, 20]. The participants are incentivized to contribute towards crowdfunding for private projects by offering them rewards. Using crowdfunding in order to raise funds for civic (non-excludable) projects, however, introduces the free-riding problem – since we cannot exclude non-contributing participants from enjoying the benefits of the public project. Thus, strategic participants, henceforth *agents*, may not contribute. If we can address this challenge, civic crowdfunding (CC) can lead to greater democratic participation. It also contributes to citizens' empowerment by allowing them to increase their well-being by solving societal issues collectively. In this paper, we focus on solving the challenge of free-riding in CC implemented using blockchain-based smart contracts.

With the advancement of the *blockchain* technology, CC projects are now being deployed using *smart contracts* (SC). A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract [18]. Since a crowdfunding project as an SC is on a trusted publicly distributed ledger, it is open and auditable, making the contributions of the agents and the execution of the payments transparent as well as anonymous. Besides, as there is no need for any centralized, trusted third party, this reduces the cost incurred in setting up the project. WeiFund [3] and Starbase [19] are examples of decentralized crowdfunding platforms on public blockchains like *Ethereum*. In this paper, our focus is to study game-theoretic challenges in CC, especially over a blockchain. Our work builds on the literature, which studies the lack of proper incentives for contributions towards public goods. Over the years, researchers have addressed such interaction as a game and analyzed equilibrium strategies of the agents in it [7, 10, 11, 24].

In the baseline approach, the social planner uses the voluntary contribution mechanism with a provision point, *provision point mechanism* (PPM) [7]. The social planner sets up a target amount, referred to as the provision point. If the net contribution by the agents crosses this provision point, the social planner executes the project. We call this as *provisioning* of the project. If the provision point is not reached, we refer to it as the project being *under-provisioned*. In the case of under-provisioning, the social planner returns the contributions. PPM has a long history of applications. However, it consists of several inefficient equilibria [7, 8, 16].

Zubrickas proposes the Provision Point mechanism with Refund bonus (PPR) in which the author introduces an additional *refund bonus* to be paid to the contributing agents, along with their contributions, in the case of under-provisioning of the project [24]. This refund bonus induces a simultaneous move game in PPR, in which the project is provisioned at equilibrium. As observed by Chandra et al. [10], PPR may fail in online settings such as Internet-based platforms (e.g., [21, 22]) since in such a setting, an agent can observe the current amount of funds raised. Hence, in online settings, strategic agents in PPR would choose to defer their contributions till the end to check the possibility of free-riding and would contribute only in the end in anticipation of a refund bonus. Such postponement in contributions leads to a scenario where every strategic agent competes for a refund bonus at the deadline. We refer to this

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 2020, Auckland, New Zealand

https://doi.org/doi

scenario as a *race condition*. In online settings, as the agents can observe the history of the contributions, it induces a sequential game, and hence, we refer to these settings as *sequential settings*.

Provision Point mechanism with Securities (PPS) by Chandra et al. [10] introduces a class of mechanisms using complex prediction markets [4] which incentivizes an agent to contribute as soon as it arrives at the platform, thus avoiding the race condition. The challenge with the practical implementation of sophisticated mechanisms such as PPS is, as it uses complex prediction markets, it is not only difficult to explain to a layperson but also computationally expensive to implement, primarily as an SC.

The introduction of the refund bonus is vital in these mechanisms for CC as it incentivizes agents to contribute and helps avoid free-riding. Hence, in this paper, we focus on provision point mechanisms with a refund bonus. Our primary goal is to abstract out conditions that *refund bonus schemes* should satisfy to avoid free-riding as well as the race condition. We believe that such a characterisation would make it easier to further explore simpler as well as computationally efficient mechanisms for CC. Towards this, we introduce, *Contribution Monotonicity* and *Time Monotonicity*. Contribution monotonicity states that an agent's refund should increase with increase in its contribution. Further, time monotonicity states that an agent's refund should decrease if it delays its contribution. We prove these two conditions are *sufficient* to provision a public project via crowdfunding in a sequential setting at equilibrium and also to avoid the race condition.

We propose three elegant refund bonus schemes which satisfy the above two conditions. These schemes are straightforward to explain to a layperson, and are computationally efficient to implement as an SC. With these three schemes, we design novel mechanisms for CC, namely *Provision Point mechanism with Refund through Geometric Progression* (PPRG); *Provision Point mechanism with Refund based on Exponential function* (PPRE), and *Provision Point mechanism with Refund based on Polynomial function* (PPRP). We analyze the cost-effectiveness of these mechanisms, as well as PPS, when deployed as SCs and show that PPRG is the most cost-effective. We measure the performance of these mechanisms by *provision accuracy*, the fraction of the projects that are successfully provisioned using the mechanism. We simulate PPRG, PPRE, PPRP, and PPS and show that PPRG has a similar provision accuracy as PPS.

Contributions.

- We define Contribution Monotonicity (Condition 1) and Time Monotonicity (Condition 2) for refund bonus schemes. We prove that it is sufficient for the scheme to satisfy these conditions to ensure that it can implement crowdfunding in sequential setting such that the project is provisioned at equilibrium (Theorem 4.1).
- We design three novel mechanisms for CC, PPRG, PPRE and PPRP based on refund bonus schemes satisfying Condition 1 and Condition 2. We show that PPRG is highly cost-efficient in comparison to PPS, PPRE and PPRP as it consumes significantly less gas when implemented as a smart contract (Section 4.3).
- We identify a set of strategies which are sub-game perfect for PPRG (Theorem 5.1).
- We simulate PPRG, PPRE, PPRP, and PPS and show that PPRG has similar provision accuracy as PPS (Section 6).

2 PRELIMINARIES

We focus on Civic Crowdfunding (CC) which involve provisioning of projects without coercion where agents arrive over time and *not* simultaneously i.e., CC in a sequential setting. Similar to [7, 10, 24], we also assume that apart from knowing the history of contributions, i.e., the provision point and the total amount remaining towards the project's provision at any time, neither agents have *any* information regarding the project's provision nor do they know about how many agents are yet to arrive or the sequence in which the agents arrive. Thus, every agent's belief is symmetric towards the project's provision.

2.1 Model

A social planner (SP) puts a proposal for crowdfunding of a civic project P on web-based crowdfunding platform; that is, we are dealing with sequential settings. SP seeks voluntary contributions towards it. The proposal specifies a target amount H necessary for the project to be provisioned, referred to as the *provision point*. It also specifies deadline (T) by which the funds need to be raised. If the target amount is not achieved by the deadline, the project is not provisioned, which we refer to as the project being under-provisioned. In the case of under-provisioning, the SP returns the contributions.

A set of agents $N = \{1, 2, \ldots, n\}$ are interested in the crowdfunding of P. An Agent $i \in N$ has value $\theta_i \geq 0$ if the project is provisioned. It arrives at time y_i to the project, observes its valuation (θ_i) for it *as well as* the net contribution till y_i. However, no agent has knowledge about any other agent's arrival or their contributions towards the project.

Agent i may decide to contribute $x_i \geq 0$ at time t_i, such that $y_i \leq t_i \leq T$, towards its provision. Let $\vartheta = \sum_{i=1}^{i=n} \theta_i$ be the total valuation, and $C = \sum_{i=1}^{i=n} x_i$ be the sum of the contributions for the project. We denote h^t as the amount that remains to be funded at time t.

A project is provisioned if $C \geq H$ and under-provisioned if $C < H$, at the end of deadline T. SP keeps a budget B aside to be distributed as a refund bonus among the contributors, if the project is under-provisioned. This setup induces a game among the agents as the agents may now contribute to get a fraction of the budget B in anticipation that the project may be under-provisioned.

Towards this, let $\sigma = (\sigma_1, \ldots, \sigma_n)$ be the vector of strategy profile of every agent where Agent i's strategy consists of the tuple $\sigma_i = (x_i, t_i)$, such that $x_i \in [0, \theta_i]$ is its voluntary contribution to the project at time $t_i \in [y_i, T]$. We use the subscript $-i$ to represent vectors without Agent i. The payoff for an Agent i with valuation θ_i for the project, when all the agents play the strategy profile σ is $\pi_i(\sigma; \theta_i)$. Note that, in this work, we assume that every agent only contributes once to the project. We justify this assumption while providing the strategies for the agents (Section 5.1). We leave it for future study to explore the effect of splitting of an agent's contribution to the project's provision and its individual payoff.

Let $\mathcal{I}_X$ be an indicator random variable which takes the value 1 if X is true and 0 otherwise. Further, let $R : \sigma \rightarrow \mathbf{R}^n$ denote the refund bonus scheme. Then the payoff structure for a provision point mechanism with a refund bonus scheme $R(\cdot)$ and budget B, for every Agent i contributing x_i and at time t_i, will be

$$\pi_i(\sigma;\theta_i) = \mathcal{I}_{C\geq H}(\theta_i - x_i) + \mathcal{I}_{C<H}\left(R_i(\sigma)\right), \qquad (1)$$

where $R_i(\sigma)$ is the share of refund bonus for Agent i as per $R(\sigma)$ such that $R(\sigma) = (R_1(\sigma), \ldots, R_n(\sigma))$. We use $R(\cdot)$ to denote a refund bonus scheme and $R_i(\cdot)$ to denote Agent i's share of the refund bonus as per $R(\cdot)$ whenever the inputs are obvious.

2.2 Important Game Theoretic Definitions

The following definitions are required for the understanding of the results presented in this paper.

Definition 2.1 (Pure Strategy Nash Equilibrium (PSNE)). A strategy profile $\sigma^* = (\sigma_1^*, \ldots, \sigma_n^*)$ is said to be a Pure Strategy Nash equilibrium (PSNE) if for every Agent i, it maximizes the payoff $\pi_i(\sigma^*;\theta_i)$ i.e., $\forall i \in N$,

$$\pi_i(\sigma_i^*, \sigma_{-i}^*;\theta_i) \geq \pi_i(\sigma_i, \sigma_{-i}^*;\theta_i)\ \forall \sigma_i, \forall \theta_i.$$

The strategy profile for the Nash Equilibrium is useful in a simultaneous move game. However, for sequential settings, where the agents can see the actions of the other agents, they may not find it best to follow the PSNE strategy. For this, we require a strategy profile which is the best response of every agent at any time during the project i.e., the best response for every sub-game induced during it. Such a strategy profile is said to be a *Sub-game Perfect Equilibrium.*

Definition 2.2 (Sub-game Perfect Equilibrium (SPE)). A strategy profile $\sigma^* = (\sigma_1^*, \ldots, \sigma_n^*)$, with $\sigma_i^* = (x_i^*, t_i^*)$, is said to be a sub-game perfect equilibrium if for every Agent i, it maximizes the payoff $\pi_i(\sigma_i^*, \sigma^*_{-i|H^{t_i^*}};\theta_i)$ i.e. $\forall i \in N$,

$$\pi_i(\sigma_i^*, \sigma^*_{-i|H^{t_i^*}};\theta_i) \geq \pi_i(\sigma_i, \sigma^*_{-i|H^{t_i^*}};\theta_i)\ \forall \sigma_i, \forall H^t, \forall \theta_i.$$

Here, H^t is the history of the game till time t, constituting the agents' arrivals and their contributions and $\sigma^*_{-i|H^{t_i^*}}$ indicates that the agents who arrive after t_i^* follow the strategy specified by σ_{-i}^*. Informally it means that, at every stage of the game, irrespective of what has happened, it is Nash Equilibrium to follow the SPE strategy for every agent.

Note that in this work, we focus on deterministic strategies over randomized strategies. We believe, it is better if game-theoretic equilibria are achieved by deterministic strategies than randomized strategies, albeit the latter introduces a richer strategy space. Such randomized strategies may consist of a joint probability distribution over the amount and the time of contribution. However, this requires the agents to perform complex randomization. The agents will also require assurance over the correctness of the randomization, with the prescribed equilibrium. Therefore, we believe that for such complex games, deterministic strategies are better. We leave it for future work to further explore other randomized strategies.

3 RELATED WORK

We focus on the class of mechanisms which require the project to aggregate a minimum level (Provision Point) of funding before the SP can claim it. There is an extensive literature on the design for mechanisms with provision points for CC. Morgan [15] incentivizes agent contribution for civic projects using state lotteries such that a higher contribution leads to a higher likelihood of winning. The game induced attains a unique equilibrium. In [14], agents contribute in a round-robin fashion such that an equilibrium exists where an agent contributes *iff* other agents make their equilibrium contributions. Our work is most closely related to the PPM, PPR and PPS mechanisms.

3.1 Provision Point Mechanism (PPM)

PPM [7] is the simplest mechanism in this class where agents contribute voluntarily. Agents gain a positive payoff only when the project gets provisioned and a payoff of zero otherwise i.e., $R^{PPM}(\sigma) = ((0) \mid \forall i \in N)$. Then the payoff structure of PPM, for every Agent i, is,

$$\pi_i(\cdot) = \mathcal{I}_{C\geq H} \times (\theta_i - x_i)$$

where, $\pi_i(\cdot)$ and x_i are Agent i's payoff and contribution respectively. PPM has been shown to have multiple equilibria and also does not guarantee strictly positive payoff to the agents. It has led the mechanism to report under-provisioning of the project, i.e., the provision point not being reached [13].

3.2 Provision Point Mechanism With Refund (PPR)

PPM does not guarantee strictly positive payoff for agents. Thus, as civic goods are non-excludable, the agents do not have an incentive to contribute, and may free-ride leading to the project not being provisioned. PPR [24] improved upon this by offering refund bonuses to the agents in case the project doesn't get provisioned and rewarded payoff like PPM otherwise. The refund bonus scheme is directly proportional to agent's contribution and is given as $R_i^{PPR}(\sigma) = \left(\frac{x_i}{C}\right) B\ \forall i \in N$, where $B > 0$ is the total budget. Then the payoff structure of PPR, for every Agent i, can be expressed as,

$$\pi_i(\cdot) = \mathcal{I}_{C\geq H} \times (\theta_i - x_i) + \mathcal{I}_{C<H} \times R_i^{PPR}(\sigma).$$

In PPR, an agent has no knowledge of other agents' contribution. Thus, as shown in [10], PPR collapses to a one-shot simultaneous game where every agent delays its contribution till the deadline. This results in each agent attempting to contribute at the deadline, leading to a *race condition*, defined as follows.

Definition 3.1 (Race Condition). A strategy profile $\sigma^* = (\sigma_1^*, \ldots, \sigma_n^*)$ is said to have a race condition if $\exists S \subseteq N$ with $|S| > 1$, for which $\forall i \in S$ the strategy $\sigma_i^* = (x_i^*, t)$, with x_i^* as the equilibrium contribution, is the PSNE of the induced game i.e., $\forall \sigma_i, \forall i \in S$,

$$\pi_i(\sigma_i^*, \sigma_{-i}^*;\theta_i) \geq \pi_i(\sigma_i, \sigma_{-i}^*;\theta_i)$$
$$\text{where } t \in [\bar{y}, T] \text{ s.t. } \bar{y} = \max_{j\in S} y_j.$$

Here, $\sigma_i = (x_i^*, t_i)\ \forall t_i \in [y_i, T]$.

For PPR, $S = N$ and $t = T$, i.e., the strategy $\sigma_i^* = (x_i^*, T)\ \forall i \in N$ constitutes a set of PSNE of PPR in a sequential setting. This is because the refund bonuses here are independent of time of contribution. Thus, agents have no incentive to contribute early. Such strategies lead to the project not getting provisioned in practice and are therefore undesirable.

3.3 Provision Point Mechanism With Securities (PPS)

PPS [10] addresses the shortcomings of PPR by offering early contributors higher refund than a late contributor for the same amount. The refund bonus of a contributor is determined using securities from a cost based complex prediction market [4] and is given as $R_i^{PPS}(\sigma) = (r_i^{t_i} - x_i)\ \forall i \in N$ where, t_i and $r_i^{t_i}$ are Agent i's time of contribution and the number of securities allocated to it, respectively. $r_i^{t_i}$ depends on the contribution x_i and the total number of securities issued in the market at the time contribution t_i denoted by q^{t_i}. Then the payoff structure of PPS, for every Agent i, can be expressed as,

$$\pi_i(\cdot) = \mathcal{I}_{C \geq H} \times (\theta_i - x_i) + \mathcal{I}_{C<H} \times R_i^{PPS}(\sigma)$$

To set up a complex prediction market in the context of CC, PPS requires a cost function (C_0) satisfying [10, CONDITIONS 1-4,6-7]. This cost function can either be based on the *logarithmic* scoring rule [10, Eq. 3] or the *quadratic* scoring rule [10, Eq. 4].

PPS awards every contributing agent securities for the project not getting provisioned. These securities are dependent on the agent contribution i.e., the greater the contribution, the higher the number of securities are allocated to the agent. Each of these securities pay out a unit amount if the project is not provisioned. However, setting up such a market and computing securities to be allotted is computationally expensive and costly to implement as a smart contract. Hence, we want to look for more desirable refund bonus schemes.

4 DESIRABLE PROPERTIES OF REFUND BONUS SCHEMES

Motivated by the theoretical guarantees of PPR and PPS, in this paper we look for provision point mechanism with refund bonus schemes. We first identify the desirable properties for such schemes.

A *desirable* refund bonus scheme should not just restrict the set of strategies in a way that the project is provisioned at equilibrium, but should also incentivize *greater* as well as *early* contributions, so as to avoid the race condition, from all interested agents. A refund bonus scheme without these, would fail in a sequential (web-based) setting, similar to PPR, and hence these are essential for a provision point mechanism's implementation online. We formalize these desirable properties as the following two *conditions* for a refund bonus scheme $R(\sigma)$ where $\sigma = ((x_i, t_i) \mid \forall i \in N)$ such that $x_i \in (0, H]$, $t_i \in [y_i, T]\ \forall i \in N$ and with budget B.

Condition 1 (Contribution Monotonicity). *The refund must always increase with the increase in contribution so as to incentivize greater contribution i.e.,* $\forall i \in N$, $R_i(\sigma) \uparrow$ *as* $x_i \uparrow$. *Further, if* $R_i(\cdot)$ *is a differential in* $x_i\ \forall i$, *then,*

$$\frac{\partial R_i(\sigma)}{\partial x_i} > 0\ \forall t_i. \tag{2}$$

Condition 2 (Time Monotonicity). *The refund must always decrease with the increase in the duration of the project so as to incentivize early contribution i.e.,* $R(\sigma)$ *must be a monotonically* decreasing *function with respect to time* $t_i \in (0, T), \forall x_i, \ \forall i \in N$ *or*

$$R_i(\sigma) \downarrow \text{ as } t_i \uparrow \text{ and } \exists\ t_i < T, \text{and } \Delta t_i \text{ s.t.,}$$
$$\frac{R_i\left((x_i, t_i + \Delta t_i), \sigma_{-i}\right) - R_i\left((x_i, t_i), \sigma_{-i}\right)}{\Delta t_i} < 0 \tag{3}$$

Note that, with Condition 2 we impose that there does not exist any time $t \in [0, T]$ such that there is race among the agents to contribute at t. We now analyze the consequence of such a refund bonus scheme on the characteristics of the game induced by it.

4.1 Sufficiency of the Refund Bonus Scheme

In this subsection, we show that a refund bonus scheme satisfying Conditions 1 and 2, is sufficient to implement civic crowdfunding projects in sequential settings. For this, let G be the game induced by the refund bonus scheme $R(\cdot)$, for the payoff structure as given by Eq. 1. We require G to satisfy the following properties.

Property 1. *In G, at equilibrium, the total contribution equals the provision point i.e.,* $C = H$.

Property 2. *G must avoid the race condition.*

Property 3. *G is a sequential game.*

With these properties, we present the following theorem.

Theorem 4.1. *Let G be the game induced by a refund bonus scheme* $R(\cdot)$ *for the payoff structure as given by Eq. 1, and with* $\vartheta > H, B > 0$. *If* $R(\cdot)$ *satisfies Conditions 1 and 2, Properties 1, 2 and 3 hold.*

Proof: In Steps 1, 2 and 3, we show that $R(\cdot)$ satisfying Condition 1 is sufficient to satisfy Property 1 and Condition 2 is sufficient to satisfy Properties 2 and 3.

- *Step 1*: As $\vartheta > H$, from Eq. 1, at equilibrium $C < H$ cannot hold, as $\exists i \in N$ with $x_i < \theta_i$, at least. Such an Agent i could obtain a higher refund bonus by marginally increasing its contribution since $R(\cdot)$ satisfies Condition 1 and $B > 0$. For $C > H$, any agent with a positive contribution could gain in payoff by marginally decreasing its contribution. Thus, at equilibrium $C = H$ or G satisfies Property 1.

- *Step 2*: Every Agent i contributes as soon as it arrives, since $R(\cdot)$ satisfies Condition 2 i.e., $\forall i \in N$,

$$\pi_i\left((x_i, y_i), \sigma_{-i}\right) > \pi_i\left((x_i, t), \sigma_{-i}\right)\ \forall t \in (y_i, T].$$

In other words, the best response $\forall i \in N$ is the strategy $\sigma_i = (x_i, y_i)$. Thus, as per Definition 3.1, G avoids the race condition or G satisfies Property 2.

- *Step 3*: Since G satisfies Property 2, it avoids the race condition. Hence, it can be implemented in a sequential setting or G is a sequential game. □

Necessity. Theorem 4.1 shows that Condition 1 is sufficient to satisfy Property 1 and Condition 2 is sufficient to satisfy Properties 2 and 3. We believe that these conditions are not *necessary* and provide an argument for the same in the complete version of the paper [12]. However, a formal proof remains elusive.

Theorem 4.1 shows that a refund bonus scheme satisfying Conditions 1 and 2 avoids the race condition (Property 2) and induces a

Mechanism	Refund Scheme	Parameters	Covergence of Sum	Based On
PPRG	$R_i^{PPRG}(\cdot) = \left(\frac{x_i + a\times(1/\gamma)^{i-1}}{C+K_1}\right)B$	$a > 0, 1/\gamma < 1, K_1 = \frac{a\gamma}{\gamma-1}$	$\sum_{i=1}^{\infty}\left(x_i + a(1/\gamma)^{i-1}\right) = C + K_1$	Geometric Progression (GP)
PPRE	$R_i^{PPRE}(\cdot) = \left(\frac{x_i + K_2\times e^{-t_i}}{C+K_2}\right)B$	$K_2 > 0$	$\sum_{i=1}^{\infty}(x_i) + \int_{t=t_1}^{\infty}(K_2 e^{-t}dt) \leq C + K_2$	Exponential Function (EF)
PPRP	$R_i^{PPRP}(\cdot) = \left(\frac{x_i + K_3\times\frac{1}{i(i+1)}}{C+K_3}\right)B$	$K_3 > 0$	$\sum_{i=1}^{\infty}\left(x_i + K_3\frac{1}{i(i+1)}\right) = C + K_3$	Polynomial Function (PF)

Table 1: Various Refund schemes satisfying Condition 1 and Condition 2 for an Agent i. Note that, in R^{PPRG} and R^{PPRP}, the subscript i denotes the order of the contribution.

sequential game (Property 3). Thus, a mechanism deploying such a refund bonus scheme can be *implemented sequentially*, i.e., over web-based (or online) platforms. Additionally, refund bonus schemes should also be clear to explain to a layperson. Moreover, these should be computationally efficient and cost-effective when deployed as a smart contract. Through this generalized result on refund bonus schemes, we show the following proposition.

Proposition 4.2. *PPS satisfies Condition 1 and Condition 2.*

Proof: Since every cost function used in PPS for crowdfunding must satisfy $\frac{\partial(r_i^{t_i} - x_i)}{\partial x_i} > 0, \forall i$ [10, CONDITION-7], PPS satisfies Condition 1.

For Condition 2, observe that $\forall i$, from [10, Eq. 6]

$$(r_i^{t_i} - x_i) = C_0^{-1}(x_i + C_0(q^{t_i})) - q^{t_i} - x_i. \tag{4}$$

In Eq. 4, as $t_i \uparrow$, $q^{t_i} \uparrow$ as it is a monotonically non-decreasing function of t and thus R.H.S. of Eq. 4 decreases since R.H.S. of Eq. 4 is a monotonically decreasing function of q^{t_i} [10, Theorem 3 (Step 2)]. Thus, PPS also satisfies Condition 2. □

Corollary 4.3. *PPS avoids the race condition and thus can be implemented sequentially.*

Proof: The authors prove in [10, Theorem 3] that PPS can be implemented sequentially without using Condition 1 and 2. However, from Proposition 4.2, and the fact that PPS payoff structure follows Eq. 1, we see from Theorem 4.1 that PPS can be implemented in a sequential setting. □

In the next subsection, we present three novel refund schemes satisfying Conditions 1 and 2 and the novel provision point mechanisms based on them.

4.2 Refund Bonus Schemes

Table 1 presents three novel refund schemes for an Agent i contributing x_i at time t_i as well as the mechanisms which deploy them. Note that, we require all the refund bonus schemes to converge to a particular sum that can be pre-computed. This convergence allows these schemes to be *budget balanced.* The parameters a, γ, K_1, K_2, K_3 and B are mechanism parameters (for their respective mechanisms) which the SP is required to announce at the start.

The refund schemes presented deploy three mathematical functions: geometrical, exponential and polynomial decay. $R^{PPRG}(\cdot)$ and $R^{PPRP}(\cdot)$ refunds the contributing agents based on the sequence of their arrivals (similar to PPS), while the refund scheme $R^{PPRE}(\cdot)$ refunds them on the basis of their time of contribution. This allows us to compare the evolution in the refund share, in comparison to PPR and PPS, with respect to the increase in time, for an Agent i. The evolution in the refund share of PPRG, PPRE and PPRP, in comparison to PPR and PPS, with respect to the increase in time, for an Agent i is depicted in Figure 1. To compare the refund shares of different schemes we keep Agent i's contribution x_i, the budget B and the provision point H same for all, with $K_1 = K_2 = K_3$.

The horizontal axis in Figure 1 represents the time at which Agent i contributes. For PPRG and PPRP, this is equivalent to the sequence in which the agents contribute, i.e., the axis represents $\tilde{t}_i$, as formally defined later in Claim 2. For PPRE, the horizontal axis is the epoch of time at which Agent i contributes, i.e., t_i. For PPS, the horizontal axis is also the sequence of agents contributing, just like in PPRG and PPRP. Each Agent $\tilde{t}_j$ ($\tilde{j} < \tilde{i}$) is issued a constant number of securities, i.e., the number of outstanding securities in the market increases by a constant number as the number of agents contributing increases.

As evident in Figure 1, the refund scheme of PPRG decreases gradually when compared to refund schemes of PPRE and PPRP. Thus, PPRG can provide significantly greater refund share for more number of agents, for the *same* bonus budget. In Section 6, we study the impact of such a refund share distribution by simulating the three mechanisms in a Reinforcement Learning environment.

4.2.1 *Sufficiency Conditions.*

We now show that PPRG satisfies Conditions 1 and 2.

Claim 1. *$R^{PPRG}(\sigma)$ satisfies Condition 1 $\forall i \in N$.*

Proof: Observe that $\forall i \in N$,

$$\frac{\partial R_i^{PPRG}(\sigma)}{\partial x_i} = \frac{B}{C + K_1} > 0\ \forall t_i.$$

Therefore, $R^{PPRG}(\cdot)$ satisfies Condition 1 $\forall i$. □

Claim 2. *$R^{PPRG}(\sigma)$ satisfies Condition 2.*

Proof: For every Agent $i \in N$ arriving at time y_i, its share of the refund bonus given by $R^{PPRG}(\cdot)$ will only decrease from that point in time, since its position in the sequence of contributing agents can only go down, making it liable for a lesser share of the bonus, for the same contribution. Let $\tilde{t}_i$ be the position of the agent arriving at time y_i, when it contributes at time t_i. While $\tilde{t}_i$ will take discrete values corresponding to the position of the agents, for the purpose of differentiation, let $\tilde{t}_i \in \mathbf{R}$. Now, we can argue that at every epoch of time t_i, Agent $\tilde{t}_i$ will contribute to the project. With

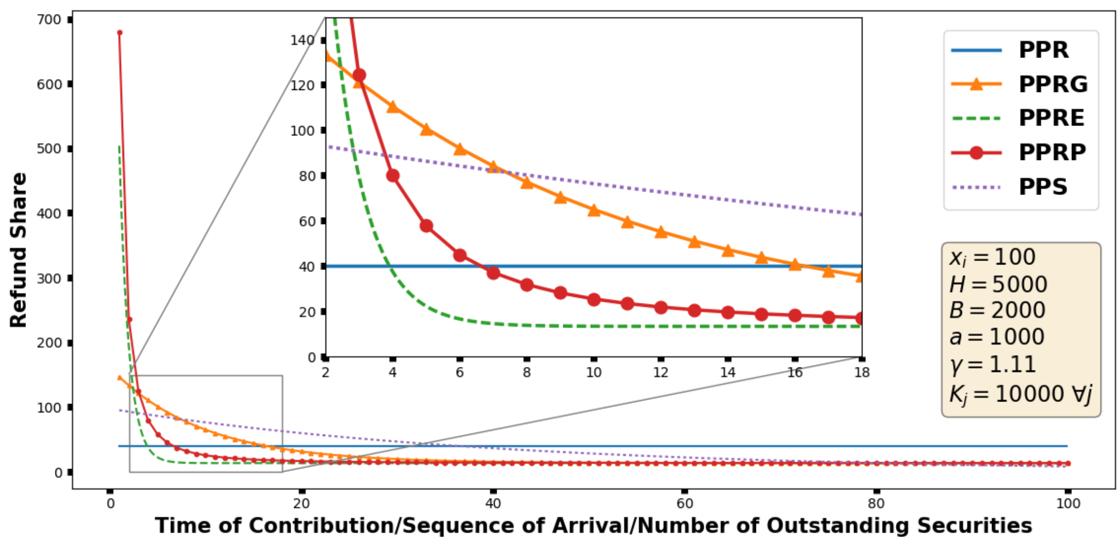


Figure 1: Evolution of the refund share for an Agent i for different provision point mechanisms.

this, $R^{PPRG}(\cdot)$ can be written as,

$$R_i^{PPRG}(\sigma) = \left(\frac{x_i + a \times (1/\gamma)^{\tilde{t_i}-1}}{C+K}\right)B.$$

Further observe that $\forall i \in N$,

$$\frac{\partial R_i^{PPRG}(\sigma)}{\partial \tilde{t_i}} = -\left(\frac{a \times (1/\gamma)^{\tilde{t_i}}}{C+K_1}\right)B < 0\ \forall x_i.$$

Therefore, $R^{PPRG}(\cdot)$ satisfies Condition 2. □

Similarly, it can be shown that R^{PPRE} and R^{PPRP} also satisfy Conditions 1 and 2. The formal proofs are presented in [12]. In the next subsection, we compare the efficiency of the refund schemes presented, in terms of their gas consumption when deployed as a smart contract over Ethereum.

4.3 Gas Comparisons

Every smart contract is compiled to a bytecode and is then executed on EVM (Ethereum Virtual Machine). EVM is sandboxed and completely isolated from the rest of the network and thus, every node runs each instruction independently on EVM. For executing every instruction, there is a specified cost, expressed in the number of *gas units*. Gas is the name for the execution fee that senders of transactions need to pay for every operation made on an Ethereum blockchain. Gas and ether are decoupled deliberately since units of gas align with computation units having a natural cost, while the price of ether fluctuates as a result of market forces. The Ethereum protocol charges a fee per computational step that is executed in a contract or transaction to prevent deliberate attacks and abuse on the Ethereum network [1].

We present a hypothetical cost comparison between PPS, PPRG, PPRE and PPRP based on the Gas usage statistics given in [9, 23]. Towards it, the cost in Gas units is as follows for the relevant operations: ADD: 3, SUB: 3, MUL: 5, DIV: 5, EXP(x): 10 + 10 $*$ $log(x)$ and LOG(x): $365 + 8 *$ size of x in bytes. Table 2 presents the comparison.

Note that, we need not require any exponential calculations in PPRG. Towards this, the SP can have a variable (say *val*) to store the previous GP term. For instance, when the first agent contributes it is allocated $a \times (1/\gamma)^0$. Post this, $val = a \times (1/\gamma)^0$. The second agent to contribute is then allocated $a \times (1/\gamma)^1$ or $val \times (1/\gamma)$ after which *val* is updated with this value. Thus, in PPRG, we can replace an exponential operation with multiplication operation which is significantly less expensive.

For every agent, PPRG takes 21 gas units, PPRP takes 31 gas units, PPRE takes at least 31 gas units and PPS takes at least 407 gas units. When implemented on smart contract, PPS is an expensive mechanism because of its logarithmic scoring rule for calculating payment rewards. PPRG, PPRP, and PPRE, on the other hand, use simpler operations and therefore have minimal operational cost.

PPRG's cost efficiency when deployed as a smart contract, along with the fact that it allocates significant refund shares for a greater number of agents when compared to PPRE and PPRP, makes it more desirable out of the three introduced. Hence, in the next section, we formally describe PPRG and analyze it.

5 PPRG

In this section, we describe the mechanism *Provision Point mechanism with Refund through Geometric Progression* (PPRG), for crowdfunding a civic project. PPRG incentivizes an interested agent to contribute as soon as it arrives at the crowdfunding platform. In PPRG, for the same contribution of Agent i and Agent j, i.e., $x_i = x_j$, the one who contributed earlier obtains a higher share of the refund bonus. These difference in shares is allocated using the terms of an

Operation	PPS		PPRG		PPRE		PPRP	
	Operations	Gas Consumed	Operations	Gas Consumed	Operations	Gas Consumed	Operations	Gas Consumed
ADD	2	6	2	6	2	6	2	6
SUB	2	6	0	0	0	0	0	0
MUL	2	10	2	10	2	10	3	15
DIV	2	10	1	5	1	5	2	10
EXP(x)	2	$10 + 10 \times (log(x))$	0	0	1	$10 + 10 \times (log(x))$	0	0
LOG(x)	2	$365 + 8 \times$ (bytes logged)	0	0	0	0	0	0
	Total Gas:	407 (at least)	**Total Gas:**	21	**Total Gas:**	31 (at least)	**Total Gas:**	31

Table 2: Gas Consumption comparison between PPS, PPRG, PPRE and PPRP for an agent. All values are in Gas units.

infinite geometric progression series (GP) with common ratio < 1. We now formally describe the refund bonus scheme in detail.

Refund Bonus Scheme. The sum of an infinite GP with $a > 0$ as the first term and $0 < 1/\gamma < 1$ as the common ratio, is,

$$K_1 = a \times \sum_{i=0}^{\inf} (1/\gamma)^i = \frac{a\gamma}{\gamma - 1}.$$

With this, we propose a novel refund bonus scheme,

$$R_i^{PPRG}(\sigma) = p_i = \left(\frac{x_i + a \times (1/\gamma)^{i-1}}{C + K_1} \right) B \tag{5}$$

for every Agent $i \in N$, B as the total bonus budget allocated for the project by the SP and where $\sigma = ((x_i, t_i) \mid \forall i \in N)$. The values a and γ are mechanism parameters which the SP is required to announce at the start of the project.

5.1 Equilibrium Analysis of PPRG

The following theorem provides the equilibrium analysis of PPRG,

Theorem 5.1. *For PPRG, with the refund p_i as described by Eq. 5 $\forall i \in N$, satisfying $0 < B \leq \vartheta - H$ and with the payoff structure as given by Eq. 1, a set of strategies* $\left\{ \left(\sigma_i^* = (x_i^*, y_i) \right) : \text{ if } h^{y_i} = 0 \text{ then } x_i^* = 0 \text{ otherwise } x_i^* \leq \frac{\theta_i (H+K_1) - aB \times (1/\gamma)^{i-1}}{H+K_1+B} \right\} \forall i \in N$ *are sub-game perfect equilibria, such that at equilibrium $C = H$. In this, x_i^* is the contribution towards the project, y_i is the arrival time to the project of Agent i, respectively.*

Step 1: Since $R^{PPRG}(\cdot)$ satisfies Condition 1 (Claim 1) and Condition 2 (Claim 2) and has a payoff structure as given by Eq. 1, from Theorem 1 we get the result that PPRG induces a sequential move game and thus, can be implemented in a sequential setting.

Step 2: From Claim 2, the best response for any agent is to contribute as soon as he arrives i.e., at time y_i.

Step 3: We assume that each agent is symmetric in its belief with respect to the provision of the project. Moreover, from Theorem 4.1, agents know that the project will be provisioned at equilibrium. Therefore, for any agent, its equilibrium contribution becomes that x_i^* for which its provisioned payoff is *greater than or equal to* its not provisioned payoff. Now, with $C = H$ at equilibrium,

$$\theta_i - x_i^* \geq p_i = \left(\frac{x_i^* + a \times (1/\gamma)^{i-1}}{C + K_1} \right) B$$

$$\Rightarrow x_i^* \leq \frac{\theta_i (H + K_1) - aB \times (1/\gamma)^{i-1}}{H + K_1 + B}$$

Step 4: Summing over $x_i^* \leq \frac{\theta_i (H+K_1) - aB \times (1/\gamma)^{i-1}}{H+K_1+B}$, $\forall i$ we get,

$$B \leq \frac{(H + K_1)\vartheta - H^2 - HK_1}{H + K_1}.$$

as $\sum_{i \in N} x_i^* = H$. From the above equation, we get

$$0 < B \leq \frac{(H + K_1)\vartheta - H^2 - HK_1}{H + K_1}$$
$$\Longrightarrow 0 < B \leq \vartheta - H,$$

as a sufficient condition for existence of Nash Equilibrium for PPRG.

Step 5: We now show that the strategies as defined in Theorem 5.1 are sub-game perfect through the following scenarios.

- For an Agent i entering the project such that $h^{y_i} = 0$ (i.e., $C = H$), its best response is contributing 0.
- For an Agent i entering the project such that $h^{y_i} > 0$ with $x_i^* > h^{y_i}$, its best response is contributing h^{y_i}. Observe that, Agent i will contribute the maximum contribution required, h^{y_i}, since its not provisioned payoff increases as its contribution increases (Claim 1). Therefore, for a contribution less than h^{y_i}, Agent i will receive *lesser* payoff in comparison for the contribution h^{y_i}.
- Lastly, for an Agent i entering the project such that $h^{y_i} > 0$ with $x_i^* \leq h^{y_i}$, its best response is contributing x_i^* (as defined in Theorem 5.1). This is because for the contribution x_i^*, its provisioned payoff is *equal* to its not provisioned payoff. For this scenario, with backward induction, it is the best response for every Agent i to follow the same strategy in which their provisioned payoffs are equal to their not provisioned payoffs, irrespective of h^{y_i}. □

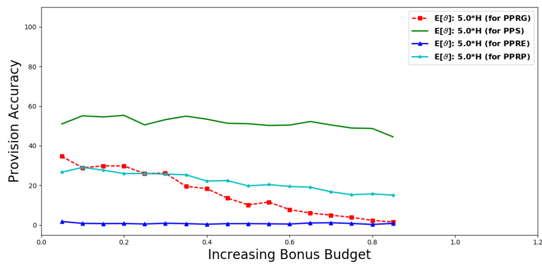
Note.

- Observe that, as the refund bonus decreases with time (Claim 2), each agent in PPRG is better off contributing once instead of breaking up its contribution. This follows as we assume that each agent's belief for the project's provision is symmetric and does not vary throughout the mechanism.
- In Theorem 5.1, we identify a set of pure-SPE at which the project is provisioned. However, we do not claim that these are the only set of pure-SPE possible. We leave it for future work to explore other possible pure-SPE at which the project gets provisioned.

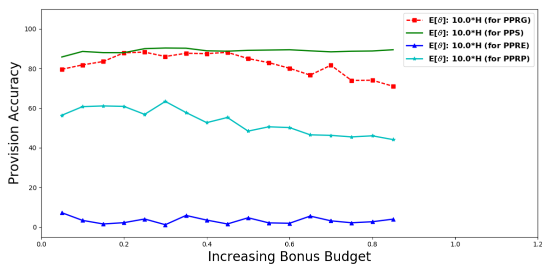
The equilibrium analysis of PPRE and PPRP follows similar to Theorem 5.1 and is presented in [12].

6 SIMULATION ANALYSIS

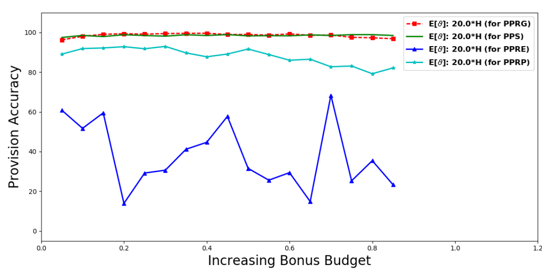
In Section 4.3 we analyzed PPRG, PPRE, and PPRP in a hypothetical cost comparison with respect to PPS if they were implemented as



(a)



(b)



(c)

Figure 2: Comparison of provision accuracy of PPRG, PPRE and PPRP with PPS for (a) $E[\vartheta] = 5 * H$ **(b)** $E[\vartheta] = 10 * H$**; and (c)** $E[\vartheta] = 20 * H$**.**

smart contracts. In this section, we compare PPRG, PPRE, PPRP, and PPS for provision accuracy using a civic crowdfunding proprietary simulator built in partnership with *KoineArth* [2].

However, it is very challenging to test civic crowdfunding mechanisms in a real-world environment because of the irreversible nature of the civic properties and decisions made in the process. Therefore, we employ Reinforcement Learning (RL) based simulations to test and compare the applicability and performance of the mechanisms. RL is an area of machine learning where agents interact with an environment and learn through a trial and error process where each action is rewarded or penalized based on its consequences on the game.

In this simulator, we create a Reinforcement Learning environment for PPRG, PPRE, PPRP, and PPS where agents learn to participate in the mechanisms. Agents go through repetitive iterations and learn their best strategy through rewards distributed by the corresponding mechanism. We run the simulation of 25 agents for all the mechanisms and obtain comparison results between PPRG, PPRE, PPRP with respect to PPS. In order to measure the performance of these mechanisms, we define the quantity *provision accuracy*. For a mechanism $\mathcal{M}$, the provision accuracy is defined as the fraction of the civic projects provisioned by $\mathcal{M}$ over the total number of projects simulated. The results of the simulation are shown in Figure 2.

Among PPRG, PPRE, and PPRP, it is clear to see that PPRG shows better provision accuracy. In case when the *total expected valuation* ($E(\vartheta)$) is low (5 times the provision point), PPRP shows slightly better accuracy. However, the gain in the accuracy only comes at the expense of a budget very close to the maximum possible budget, i.e., $B = E(\vartheta) - H$. Such a budget is difficult to get in realistic circumstances. Note that, the equilibrium contributions are such that the provisioned payoff equals the not provisioned payoff (as defined in Theorem 5.1). Therefore, the difference in the accuracy can be attributed to the greater refund share provided by PPRG, for the same budget. This increases the not provisioned payoff for the agents, thereby incentivizing them to increase their contributions. Thus, we conclude that PPRG performs better than PPRE and PPRP.

When compared to PPS, PPRG shows significantly good provision accuracy when $E(\vartheta)$ is high (10 times provision point, for instance). When PPS shows a slightly higher accuracy, it again comes at the expense of a budget close to the maximum possible budget, B. For a reasonable budget of approximately $0.5 \times B$ or less, both the mechanisms share similar accuracy. Thus, PPS and PPRG perform equally in terms of provision accuracy, for a rational budget.

7 CONCLUSION

Motivated by the theoretical guarantees of PPR [24] and PPS [10], we looked for provision point mechanisms for CC with refund bonus schemes. We introduced two conditions, namely Contribution Monotonicity and Time Monotonicity, for refund bonus schemes in provision point mechanisms. We proved that these two conditions are sufficient to implement provision point mechanisms with refund bonus to possess an equilibrium that avoids free-riding and race condition (Theorem 4.1). With this, we proposed three simple refund bonus schemes based on geometric progression, exponential and polynomial functions. With these schemes, we designed novel mechanisms, namely, PPRG, PPRE and PPRP. We showed that PPRG has much less cost when implemented as a smart contract over Ethereum framework. We identified a set of sub-game perfect equlibria for PPRG in which the project is provisioned at equilibrium (Theorem 5.1). To measure the performance of these mechanisms, we introduced a notion of provision accuracy. Our simulations showed that, whenever there is a hefty valuation for the project under consideration, with small refund bonus budgets, PPRG achieves the same provision accuracy as PPS. We leave it for future work to explore other refund bonus schemes having simplicity and efficiency as PPRG and much higher provision accuracies when the aggregate of the agents' valuations is just over target value.

REFERENCES

[1] [n. d.]. Ethereum Homestead Documentation. ([n. d.]). http://ethdocs.org/en/latest/

[2] [n. d.]. KoineArth Inc. ([n. d.]). https://koinearth.com

[3] 2015. WeiFund - Decentralised Fundraising. (2015). http://weifund.io/

[4] Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan. 2013. Efficient market making via convex optimization, and a connection to online learning. *ACM Transactions on Economics and Computation* 1, 2 (2013), 12.

[5] Saeed Alaei, Azarakhsh Malekian, and Mohamed Mostagir. 2016. A Dynamic Model of Crowdfunding. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC '16)*. ACM, New York, NY, USA, 363–363. https://doi.org/10.1145/2940716.2940777

[6] Itai Arieli, Moran Koren, and Rann Smorodinsky. 2017. The Crowdfunding Game. In *Web and Internet Economics - 13th International Conference, WINE 2017, Bangalore, India, December 17-20, 2017, Proceedings*.

[7] Mark Bagnoli and Barton L Lipman. 1989. Provision of public goods: Fully implementing the core through private contributions. *The Review of Economic Studies* 56, 4 (1989), 583–601.

[8] Earl R Brubaker. 1975. Free ride, free revelation, or golden rule? *The Journal of Law and Economics* 18, 1 (1975), 147–161.

[9] Vitalik Buterin et al. 2014. A next-generation smart contract and decentralized application platform. *white paper* (2014).

[10] Praphul Chandra, Sujit Gujar, and Y Narahari. 2016. Crowdfunding Public Projects with Provision Point: A Prediction Market Approach.. In *ECAI*. 778–786.

[11] Praphul Chandra, Sujit Gujar, and Yadati Narahari. 2017. Referral-Embedded Provision Point Mechanisms for Crowdfunding of Public Projects. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 642–650.

[12] Sankarshan Damle, Moin Hussain Moti, Praphul Chandra, and Sujit Gujar. 2018. Designing refund bonus schemes for provision point mechanism in civic crowdfunding. *arXiv preprint arXiv:1810.11695* (2018).

[13] Paul J Healy. 2006. Learning dynamics for mechanism design: An experimental comparison of public goods mechanisms. *Journal of Economic Theory* 129, 1 (2006), 114–149.

[14] Leslie M Marx and Steven A Matthews. 2000. Dynamic voluntary contribution to a public project. *The Review of Economic Studies* 67, 2 (2000), 327–358.

[15] John Morgan. 2000. Financing public goods by means of lotteries. *The Review of Economic Studies* 67, 4 (2000), 761–784.

[16] David Schmidtz. 1991. The Limits of Government (Boulder. (1991).

[17] Wen Shen, Jacob W Crandall, Ke Yan, and Cristina V Lopes. 2018. Information Design in Crowdfunding under Thresholding Policies. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 632–640.

[18] Smart Contract. 2006. Smart Contract — Wikipedia, The Free Encyclopedia. (2006). https://en.wikipedia.org/wiki/Smart_contract

[19] Starbase. 2016. Starbase. (2016). https://starbase.co/ Available at https://starbase.co/.

[20] Roland Strausz. 2017. A theory of crowdfunding: A mechanism design approach with demand uncertainty and moral hazard. *American Economic Review* 107, 6 (2017), 1430–76.

[21] Wikipedia contributors. 2019. GoFundMe — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=GoFundMe. (2019).

[22] Wikipedia contributors. 2019. Kickstarter — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Kickstarter. (2019).

[23] Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151 (2014), 1–32.

[24] Robertas Zubrickas. 2014. The provision point mechanism with refund bonuses. *Journal of Public Economics* 120 (2014), 231–234.