2021

Game-theoretic Analysis of Effort Allocation of Contributors to Public Projects

Jared Soundy
University of Nebraska-Lincoln, jared.soundy@huskers.unl.edu

Chenhao Wang
University of Nebraska-Lincoln, cwang55@unl.edu

Clay Stevens
University of Nebraska - Lincoln, clay.stevens@huskers.unl.edu

Hau Chan
University of Nebraska-Lincoln, hchan3@unl.edu

Game-theoretic Analysis of Effort Allocation of Contributors to Public Projects

Jared Soundy* , **Chenhao Wang** , **Clay Stevens** and **Hau Chan**
University of Nebraska – Lincoln
{jared.soundy, clay.stevens}@huskers.unl.edu, {cwang55,hchan3}@unl.edu

Abstract

Public projects can succeed or fail for many reasons such as the feasibility of the original goal and coordination among contributors. One major reason for failure is that insufficient work leaves the project partially completed. For certain types of projects anything short of full completion is a failure (*e.g.,* feature request on software projects in GitHub). Therefore, project success relies heavily on individuals allocating sufficient effort. When there are multiple public projects, each contributor needs to make decisions to best allocate his/her limited effort (*e.g.,* time) to projects while considering the effort allocation decisions of other strategic contributors and his/her parameterized utilities based on values and costs for the projects. In this paper, we introduce a game-theoretic effort allocation model of contributors to public projects for modeling effort allocation of strategic contributors. We study the related Nash equilibrium (NE) computational problems and provide NP-hardness results for the existence of NE and polynomial-time algorithms for finding NE in restricted settings. Finally, we investigate the inefficiency of NE measured by the price of anarchy and price of stability.

1 Introduction

Often, we jointly work together on public projects. This ranges from small projects involving a few contributors (*e.g.,* a medium sized feature request on GitHub) to massive projects involving several nations (*e.g.,* International Space Station). Each contributor can allocate his/her available effort (*e.g.,* time) to the set of projects and determines the utility trade-off between reward (*e.g.,* satisfaction) and cost (*e.g.,* how fatiguing it is to allocate effort) per project. Joint projects can succeed or fail for many reasons: shifting requirements, management inexperience, *etc.,* [McConnell, 1996]. One common pivotal point that many projects have is that there is a minimal *threshold* or amount of work that must be completed before the project succeeds. This can apply to public projects, e.g., in software development [McConnell, 1996] and "Get Big Fast" type businesses [Spolsky, 2004]. The common thread is that a given project provides little to no utility until it is completed in its entirety. Project failure is not uncommon and contributors continue to exert effort in projects which may not succeed [McConnell, 1996]. This behavior seems irrational on its face, but effort allocation behavior among contributors to such projects is complex. To judge the rationality of a contributor's decision to allocate effort to a particular set of projects, these questions must be explored:

(**Modeling**) How would *strategic* contributors behave if a project's success depended solely on total contribution meeting a threshold? How would strategic contributors allocate effort to a given available set of projects with such requirements? (**Computation**) What are the corresponding (Nash) equilibrium computation questions and results?

Our goal in this paper is to address the above questions, theoretically and computationally, through game-theoretic models and analysis focusing on strategic interactions of strategic and rational contributors for public project contributions.

Our Contribution. To address the modeling and computational questions, we first introduce a game-theoretic Effort Allocation model of Contributors to Public Projects (EACPP) where a set D of contributors determines the number of discrete units of effort (*e.g.,* hours) to allocate to a set P of available projects with contribution threshold requirements. Each contributor $i \in D$ gains utility for each project $j \in P$ depending on i's effort allocation for project j, a_{ij}; i's intrinsic value of j, v_{ij}; i's skill related to j, s_{ij}; the cost for i of each unit of effort contributed to j, c_{ij}; j's contribution threshold, t_j; and most importantly, the effort allocation of other strategic contributors. Given the EACPP model, our interest is the Nash equilibrium (NE) computational questions, and the measurement for the efficiency of NEs. We first show–via novel reduction from partition problems–that the following are NP-complete, even in the restricted settings where each contributor has a single unit of effort to allocate to projects, skill is project-independent (*i.e.,* $s_{ij} = s_i$), and effort is costless (*i.e.,* $c_{ij} = 0$): (1) determining the existence of a NE, (2) determining the existence of a NE with at least k successful projects, (3) determining the existence of a NE with social welfare of at least V, and (4) determining the existence of a NE with a fixed effort allocation of contributors.

*Corresponding author

We then consider further restrictions on the model parameters. First, for the restricted settings where $v_{ij} = v_i$ or $v_{ij} = v_j$ under certain threshold and skill configurations, we show that a NE always exists and can be computed in polynomial time using an ordered sequential response algorithm (see Theorems 3 and 4). Second, when there are constant number of projects, we can find all NEs in pseudo-polynomial time in terms of the number n of contributors and parameter $r := \max_{j \in P}(1 + \sum_{i \in D} s_{ij})$ (see Theorem 5).

We consider a planner's problem of obtaining optimal outcomes that maximize either (i) the social welfare (*i.e.,* the sum of the utilities of the contributors) or (ii) the number of successful projects by allocating effort of contributors to projects. We show that the corresponding optimization problems are NP-hard (see Theorem 6). We investigate the efficiency of NEs by examining the price of anarchy (PoA) [Koutsoupias and Papadimitriou, 1999; Roughgarden, 2005], which is defined as the ratio between the optimal solution and the worst NE, and the price of stability (PoS) which is defined as the ratio between the optimal solution and the best NE [Anshelevich *et al.*, 2008] (See Section 5.) We show that they can be unbounded for both objectives, and identify the restricted settings where they are bounded or even equal to one.

1.1 Related Work

Below, we compare EACPP and our contributions to related work and highlight the key differences in several areas.

Project Games. The most related problem to EAPCC is project games [Bilò *et al.*, 2019] where each player selects a single project. Any project selected by at least one player is realized and the reward for the project is divided among contributing players. Similar to EACPP, players have skill (called weight) and reward is split proportionally based on contribution. The key difference from EACPP is that there is no threshold. One can view their model as a special case of ours under several restrictions on the model parameters, including zero threshold and cost for each project. Throughout the paper, we will elaborate the implication of their results.

Contests. There is a line of research where a set of contestants competes in a set of contests [DiPalantino and Vojnovic, 2009; Bernergård and Wärneryd, 2017; Morgan *et al.*, 2017; Azmat and Möller, 2009; Azmat and Möller, 2017; Chan *et al.*, 2020; Xu *et al.*, 2019]. In such settings, each contestant selects (a subset of) contests and/or determines the amount of effort to compete for prizes. In these settings, each contest is typically modeled using single contest semantics such as a Tullock contest, all pay-auction, and their variants (*e.g.,* see [Dechenaux *et al.*, 2015; Konrad, 2009]). However, these models do not consider threshold requirement for each contest, and the contest's success is not measured in terms of overall contributions of the contestants.

Congestion Games. In a congestion game (see *e.g.,* [Rosenthal, 1973; Fotakis, 2015; Kontogiannis and Spirakis, 2005; Monderer and Shapley, 1996; Mavronicolas *et al.*, 2007; Kleinberg and Oren, 2011]), there is a set of resources and a set of players. Each player's action set consists of some subsets of the resources. The player's goal is to select an action that minimizes the sum of the delay of each selected resource in the selected action where the delay is defined to be some increasing function of the number of players that selected the resource. Our EACPP is different from congestion games and their variations mainly in that there is a threshold for each resource/project and positive utility can only be obtained if there are sufficient contributions to the resource/project.

Networked Public Goods/Crowdfunding Games. Recent related works (see *e.g.,* [Yu *et al.*, 2020; Komarovsky *et al.*, 2015; Kempe *et al.*, 2020; Arieli *et al.*, 2018]) consider networked public goods games where each individual makes a binary decision to contribute to a public good project. An individual decision depends on (a function that maps) the number of other individuals in the individual's social network and the cost of contribution. Different from EACPP, public goods games and their variants consider only a single project with no threshold requirement or/and individual project model parameters related to values and effort contributions.

2 Game-theoretic Effort Allocation Models

In this section, we present a game-theoretic Effort Allocation model of Contributors to Public Projects (EACPP). In an EACPP game-theoretic model, we have a set $D = \{1, ..., n\}$ of n contributors and a set $P = \{1, ..., m\}$ of m projects indexing using i and j, respectively.

Contributors. Each contributor $i \in D$ allocates his/her (discrete) units of effort h_i (*e.g.,* work hours) among the set of public projects. Thus, the set of i's actions is $A_i = \{\mathbf{a}_i \in \{0, 1, ..., h_i\}^m \mid \sum_{j=1}^m a_{ij} = h_i\}$ where, given an allocation vector $\mathbf{a}_i \in A_i$, a_{ij} specifies the units of effort allocated to project $j \in P$. We let $A = \prod_{i=1}^n A_i$ be the joint-action set of effort allocation vectors. Contributor i's contribution depends on both the allocated effort and personal skills on the projects. Skill can be viewed as the efficiency of converting effort into contribution (*e.g.,* an experienced software developer is more efficient than a novice). For each $i \in D$, let the integer $s_{ij} \geq 0$ be i's skill for project j. Given allocation vector $\mathbf{a}_i \in A_i$, i's *contribution* to j is $a_{ij}s_{ij}$, representing the impact of i's skill on the effort they allocated to j. For each $i \in D$, there is a (per-unit effort) cost c_{ij} for each project $j \in P$ such that the total cost of j to i under $\mathbf{a}_i$ is defined by $a_{ij}c_{ij}$. Finally, there is an overall personal (satisfaction) value $v_{ij} \geq 0$ contributor i has for project j if j is successful.

Projects. Each project $j \in P$ has a contribution threshold $t_j \geq 0$ where the project is successful when the total contribution to j by all contributors exceeds the threshold. Given joint-action $\mathbf{a} \in A$, the total contribution on project j is $TC_j(\mathbf{a}) = \sum_{i=1}^n a_{ij}s_{ij}$. A project j succeeds if and only if $TC_j(\mathbf{a}) > 0$ and $TC_j(\mathbf{a}) \geq t_j$.

Utilities. Given an action profile $\mathbf{a} \in A$, a reward for contributor $i \in D$ for project $j \in P$ is defined to be

$$r_{ij}(\mathbf{a}) = \begin{cases} -c_{ij}a_{ij} & TC_j(\mathbf{a}) = 0 \\ -c_{ij}a_{ij} & TC_j(\mathbf{a}) < t_j \\ v_{ij}\frac{a_{ij}s_{ij}}{TC_j(\mathbf{a})} - c_{ij}a_{ij} & \text{otherwise.} \end{cases}$$

As motivated from existing literature (*e.g.,* project games [Bilò *et al.*, 2019] and Tullock contests [Dechenaux *et al.*, 2015; Konrad, 2009]), we consider a simple proportional rule to divide utility among contributors according to contribution made. This allows us to model many complex motivations, such as the prestige of being listed as a top contributor to a popular project or differential allocation of reward based on the proportion of work done. If i contributes additional work beyond t_j then i receives a larger proportion of the project, but the value of the project is not increased. The utility of contributor $i \in D$ is defined to be $u_i(\mathbf{a}) = \sum_{j=1}^{m} r_{ij}(\mathbf{a})$ which is the sum of the rewards over the projects.

Objectives. We quantify the quality of an action profile $\mathbf{a} \in A$ by considering two objectives, namely, the *social welfare* (SW) objective and the *projects successful* (PS) objective: (1) $SW(\mathbf{a}) = \sum_{i \in D} u_i(\mathbf{a})$ is the total utility of all contributors, and (2) $PS(\mathbf{a}) = \sum_{j \in P} \mathbb{1}[TC_j(\mathbf{a}) \geq t_j \wedge TC_j(\mathbf{a}) > 0]$ is the number of successful projects, where $\mathbb{1}[\cdot]$ is an indicator function. The planner wants to find optimal solutions that maximizes the objectives. We let $\mathbf{a}^{opt_{sw}} \in \mathrm{argmax}_{\mathbf{a} \in A} SW(\mathbf{a})$ and $\mathbf{a}^{opt_{ps}} \in \mathrm{argmax}_{\mathbf{a} \in A} PS(\mathbf{a})$ be the optimal solutions that maximize the SW and PS objectives, respectively.

Given an EACPP instance $\mathcal{I} = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$ (indices omitted), our goal is to compute a (pure-strategy) Nash equilibrium (NE).

Definition 1. *A joint-effort allocation vector $\boldsymbol{a} \in A$ is a (pure-strategy) Nash equilibrium (NE) if and only if $u_i(\boldsymbol{a}_i, \boldsymbol{a}_{-i}) \geq u_i(\bar{\boldsymbol{a}}_i, \boldsymbol{a}_{-i})$ for any $\bar{\boldsymbol{a}}_i \in A_i$ and each $i \in D$.*

Unfortunately, a NE might not always exist in an arbitrary EACPP. Below we provide an example.

Example 1. *Consider an instance $\mathcal{I} = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$ where $D = \{1, 2\}$, $P = \{1, 2\}$, $c_{ij} = 0$, $v_{ij} = v$, $h_i = 1$ for all $i \in D, j \in P$, $s_{1j} = s_1$, $s_{2j} = s_2$, and $t_j = s_1$ for all $j \in D$ for $s_1 > s_2$. In such a setting, contributor 1 alone is enough to make the projects successful whereas contributor 2 cannot. Contributor 2 always prefers to select the same project as contributor 1, and contributor 1 would always prefer to work on any one of the projects independently. It is easy to see that there is no NE.*

We note that our model can be further generalized and extended to cases where (a) the total contribution TC_j can be an arbitrarily non-decreasing function and (b) the proportional contribution term in the reward function r_{ij} can be any contest success function. However, as we will see in the next section, the computational questions related to NE are already NP-complete for various instances of our model.

We also remark that EACPP captures simultaneous strategic interactions of the contributors when they make effort allocation decisions at the same time in the complete information setting. It would be interesting to extend the models further to (a) consider sequential interactions where contributors determine effort allocation over time and/or (b) incomplete information where the model parameters (*e.g.,* values, skills, and costs) are drawn from some distributions.

3 Complexity of Computing a NE in EACPP

In this section, we consider the complexity questions related to the existence of a NE in EACPP. In particular, we show that the following questions are all NP-complete in EACPP:

1. Is there a NE?
2. Is there a NE where at least k projects are successful?
3. Is there a NE that obtains a social welfare of at least V?
4. Is there a NE that is consistent with a fixed assignment?

All the computational questions are NP-complete even in the case where $c_{ij} = 0$, $s_{ij} = s_i$, and $h_i = 1$ for all $i \in D$ and $j \in P$ (*i.e.,* each contributor has a cost of zero for each project with skill independent of the projects and can only select a single project). Our reductions use the known NP-complete Partition Problem (PP) [Garey and Johnson, 1979]:

Partition Problem (PP) Given a set $X = \{x_1, ..., x_n\}$ of n distinct positive integers, is there a partition of X to X_1 and X_2 such that $\sum_{x \in X_1} x = \sum_{x \in X_2} x$?

Theorem 1. *It is NP-complete to determine whether there is a NE in EACPP.*

Proof. This problem is in NP, because we can verify whether a given profile $\mathbf{a} \in A$ is a NE in polynomial time by checking for $m - 1$ deviations for each of the n contributors.

To prove our claim, we reduce from PP. Consider an instance of PP with $X = \{x_1, ..., x_n\}$ of n positive (unique) integers such that $x_1 > x_2 > ... > x_n$ (with the total sum divisible by 2), we reduce it to an instance of EACPP $I = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$ via the following:

- Let $m = n + 2$ be the number of projects;
- Let n be the number of contributors;
- Let $h_i = 1$ for each $i \in D$;
- Let $s_{ij} = x_i$ and $c_{ij} = 0$ for each $i \in D$ and $j \in P$;
- For each $j \in P \setminus \{1, 2\}$, let $x_{n-1} > t_j > x_n$ be some threshold and $t_1 = t_2 = T = \frac{\sum_{x \in X} x}{2}$;
- For each $i \in D \setminus \{n\}$ and $j \in P \setminus \{1, 2\}$, $v_{ij} = v_i > 0$ and $v_{i1} = v_{i2}$ such that $v_{i1} \frac{s_{ij}}{T} > v_i > v_{i1} \frac{s_{ij}}{T + 0.5}$;
- For n and $j \in P \setminus \{1, 2\}$, $v_{nj} = v_n > 0$ and $v_{n1} = v_{n2}$ such that $v_{n1} \frac{s_{nj}}{2T} > v_n$.

The next to last parameter settings ensure that contributor $i \in D \setminus \{n\}$ always prefers either project 1 or project 2 when the projects are successful with exactly T total contribution and, otherwise, one of the other projects (3, ..., m) without any other contributors. The last setting ensures that n prefers either project 1 or 2 as long the projects are successful.

PP solution $\implies$ NE. It is not hard to see that if there is partition of X into X_1 and X_2 with the same sum T, then we have a NE. In particular, for i with skill $x_i \in X_l$ for all j for some $l = 1, 2$, then $a_{il} = 1$ and $a_{ik} = 0$ for each other project k. It is easy to check that for $i \in D$, i has no incentive to deviate (see the last two items listed).

NE $\implies$ PP solution. Suppose there is a NE. We argue that, in any NE, n must either select successful project 1 or project 2. If this is not the case, then n must select one of the $\{3, ..., m\}$ projects. If there are other contributors in $\{3, ..., m\}$, it must be the case they are all in different projects and n will select one of them as a best response, say the project with i because each project j requires threshold $t_j > x_n$, which cannot be competed by n alone. However, i will deviate to another project without any contributor (which always exists since there are n other projects).

If there are no other contributors in $\{3, ..., m\}$, it must be the case that all of them are in either project 1 or project 2, but this cannot be a NE, since the threshold of each project is T and n is not in project 1 or 2. As a result, the projects must have threshold greater or less than T. In either case, this cannot be a NE as the contributors will deviate (by construction).

In order to sustain a NE where n is in project 1 or project 2, at least one of the projects has to be successful with exactly T total contribution (including x_n). The remaining contributors are either together in project 1 or 2 or in different projects of $\{3, ..., m\}$. It follows that one can construct a solution of PP by letting X_1 be the set of positive integers with x_i corresponding to $a_{il} = 1$ for $l(= 1$ or $2)$ project(s) with exactly T contribution with x_n. The remaining positive integers/contributors will be in X_2 which has the same sum as X_1 by construction. □

Theorem 2. *It is NP-complete to determine whether there is a NE with at least k successful projects in EACPP, determine whether there is a NE with social welfare at least V, or determining the existence of a NE with a fixed effort allocation of contributors.*[1]

We remark that positive results can be achieved in some more restricted settings. When $h_i = 1, c_{ij} = 0$, $s_{ij} = s_i$, $v_{ij} = v_j, t_j = 0$, a NE can be computed in polynomial time [Bilò *et al.*, 2019; Fotakis *et al.*, 2002]. In addition, when $h_i = 1, c_{ij} = 0, v_{ij} = v, t_j = 0$, the better response dynamics always converges to a NE [Bilò *et al.*, 2019].

4 Algorithms for Computing NE in EACPP

In Section 3, we have shown that determining the existence of a NE is NP-complete in instances $I = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$ where each $i \in D$ can only select one project (*i.e.,* $h_i = 1$), has zero-cost for all projects (*i.e.,* $c_{ij} = 0$ for all j), and has skill independent of the projects (*i.e.,* $s_{ij} = s_i$ for all j). In this section, we investigate EACPP where we can compute NE efficiently.

4.1 A Polynomial-Time Algorithm

The hardness results suggest that in order to derive an efficient algorithm for computing a NE, one must place further restrictions on the value v_{ij} and t_j in addition to the parameter settings in the hardness proofs. Our following result shows that if each project can be successfully completed by any player working alone using a single effort allocation and v_{ij} are restricted (*i.e.,* $v_{ij} = v_i$ or $v_{ij} = v_j$), then a NE always exists and can be computed via ordered sequential response (Algorithm 1). Algorithm 1 starts by ordering the contributors subject to some $parameter$, and lets each contributor, under the same ordering, sequentially select a best response. After all contributors have selected actions, it returns a NE.

Theorem 3. *Let $I = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$ be an EACPP instance where (1) $h_i = 1$, (2) $c_{ij} = 0$, and (3) $s_{ij} = s_i$ such that $s_1 \geq ... \geq s_n$ for all i, j. Algorithm 1 returns a NE for the following settings in polynomial time:*

Algorithm 1: Ordered Sequential Response

Input: $I = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$
Output: A NE profile $\mathbf{a} \in A$

1 Let $\mathbf{a} = \mathbf{0}$ # Set $\mathbf{a}$ to be a set of zero vectors.
2 Order contributors via $parameter$: $D = \{1, ..., n\}$
3 **for** $i = 1, ..., n$ **do**
4 Let $\mathcal{BR}_i(\mathbf{a}_{-i}) = \text{argmax}_{\mathbf{a}'_i \in A_i} u_i(\mathbf{a}'_i, \mathbf{a}_{-i})$
5 Select $\bar{\mathbf{a}}_i \in \mathcal{BR}_i(\mathbf{a}_{-i})$, set $\mathbf{a}_i = \bar{\mathbf{a}}_i$
6 #Note: if $v_{ij} = 0$; $\forall j \in P$, select $\bar{\mathbf{a}}_i$ with a project with the lowest total contribution
7 **end**

(a) $v_{ij} = v_j$ and $t_j \leq s_n$ for all i, j or
(b) $v_{ij} = v_i$ and $t_j \leq s_n$ for all i, j
with contributor's ordering induced by skills $s_1 \geq ... \geq s_n$.

Proof. We first consider the setting of (a) where $v_{ij} = v_j$, $s_1 \geq s_2 ... \geq s_n$, and $t_j \leq s_n$ for all i, j. Let i be the i^{th} contributor in the algorithm. For simplicity, we let $\mathbf{a}^{(0)} = \mathbf{0}, ..., \mathbf{a}^{(n)}$ be the action profile of Algorithm 1 after each iteration. We now show that, via induction, after (line 5) setting $\mathbf{a}_i = \bar{\mathbf{a}}_i$ such that $a_{ij} = 1$ and $a_{ik} = 0$, $\forall k \neq j$, $\bar{\mathbf{a}}_i \in \mathcal{BR}_i(\mathbf{a}_{-i})$, no other contributor $i' = 1, ..., i - 1$ would deviate. The basic case is straightforward when $i = 1$. Now we assume it is true up until $i - 1$. There are two possible cases we need to consider. If $a_{i'j} \neq a_{ij}$, we have that

$$u_{i'}(\mathbf{a}^{(i-1)}) = v_k \frac{s_{i'}}{TC_k(\mathbf{a}^{(i-1)})} \geq v_w \frac{s_{i'}}{TC_w(\mathbf{a}^{(i-1)}) + s_{i'}}$$

where $a_{i'k} = 1$, $k \neq j$, and $w \neq k$. The inequality continues to hold for $\mathbf{a}^{(i)}$ as the only difference is on the j project (with the addition of i) and $TC_j(\mathbf{a}^{(i)})$ is nondecreasing.

If $a_{i'j} = a_{ij} = 1$, it follows that $u_i(\mathbf{a}^{(i)}) =$

$$v_j \frac{s_i}{TC_j(\mathbf{a}^{(i)})} \geq v_w \frac{s_i}{TC_w(\mathbf{a}^{(i)}) + s_i} \geq v_w \frac{s_i}{TC_w(\mathbf{a}^{(i)}) + s_{i'}}$$

for any $w \neq j$ and i'. Thus, for any i', we can show that i' has no incentive to deviate from j by dividing $s_i > 0$ and multiplying $s_{i'} > 0$ from the above inequality. We can use a similar argument as above for part (b). □

In the above setting, we consider cases where $t_j \leq s_n$ for each project j. The threshold effectively does not exist because every player can overcome it. A natural question is whether we can identify instances where $t_j > s_n$ for which we can compute NE efficiently. We can answer this question affirmatively if the contributors have the same skill level (*i.e.,* $s_{ij} = s$) and the v_{ij} are restricted (*i.e.,* $v_{ij} = v_i$ or $v_{ij} = v_j$).

Theorem 4. *Let $I = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$ be an EACPP instance where (1) $h_i = 1$, (2) $c_{ij} = 0$, and (3) $s_{ij} = s$ for all i, j. One can compute a NE for the following settings in polynomial time:*
(a) $v_{ij} = v_j$ for all i, j or
(b) $v_{ij} = v_i$ for all i, j.

Proof (Sketch). Assume w.l.o.g. $t_1 \leq ... \leq t_m$. If every project requires at least two contributors (i.e, $t_j > s$ for each

[1]Omitted proofs can be found in the supplementary material at: https://digitalcommons.unl.edu/cseconfwork/332/

j), then setting the action of each contributor to a project (say j) is a NE as no contributor can unilaterally deviate to obtain a better utility. As such, we assume that there is a project, say k from the threshold ordering, that can be achieved by a single contributor (this can be verified in polynomial time). It follows that each project $1, ..., k$ can be achieved by a single contributor. Applying Algorithm 1 to current settings for setting (a) or (b), we obtain a NE by ordering the contributors arbitrarily, which can be proved using Theorem 3. $\square$

4.2 A Pseudo-Polynomial Algorithm

In general, we can use a standard enumeration approach to determine the existence of a NE, which would result in a running time of $O(nm^{n+1})$ that is exponential in the number of contributors. It turns out that we can derive an improved enumeration algorithm with a polynomial running-time in the number of contributors and a natural model parameter $r := \max_{j \in P}(1 + \sum_{i \in D} s_{ij})$, when the number of projects is bounded (and recall that s_{ij} is an integer). The assumption that m is bounded is natural, because in reality the planner often faces a small number of projects but a large number of contributors. We now give the pseudo-polynomial time algorithm when $h_i = 1$ and m is a constant.

Theorem 5. *When $h_i = 1$ and the number of projects m is a constant, there exists a pseudo-polynomial algorithm that returns all NE in $O(r^m(nmr^m + nm^2))$.*

Proof. We say vector $\mathbf{q} = (q_1, ..., q_m)$ is a potential-sum configuration, if $q_j \in \{0, 1, \ldots, r-1\}$ for any $j \in P$. Let Q be the set of all of such configurations. Clearly, $|Q| = r^m$. The entry of a configuration corresponds to the total contribution on a project (*i.e.,* $q_j = c$ means that project j receives a total contribution of c.). Note that for every $\mathbf{a} \in A$ there exists a $\mathbf{q} \in Q$ such that $TC_j(\mathbf{a}) = q_j$. Therefore by checking every $\mathbf{q} \in Q$ for the existence of a NE, every possible strategy profile is checked.[2] We note that, to check if a solution is a NE, it suffices to rely on the information of the corresponding configuration. Our algorithm consists of the following two steps: for every configuration $\mathbf{q} = (q_1, ..., q_m) \in Q$,

Step 1: Compute the set of best-response strategies/projects $b_i \subseteq P$ for the given $\mathbf{q}$ and each contributor $i \in D$. More specifically, we define and compute

$$r^*_{ij}(x) = \begin{cases} -c_{ij} & x = 0 \text{ or } x < t_j \\ v_{ij}\frac{s_{ij}}{x} - c_{ij} & \text{otherwise.} \end{cases}$$

$$b_i = \{j \in P \mid r^*_{ij}(q_j) \geq r^*_{ik}(q_k + s_{ik}), \forall k \in P \setminus \{j\}\}.$$

Let $\mathbf{b} = (b_1, \ldots, b_n)$ be the profile of best-response projects.

Step 2: Given the profile $\mathbf{b}$, we check if $\mathbf{b}$ can *induce* the potential-sum configuration $\mathbf{q}$, that is, there exist an action profile $\mathbf{a}$, such that for any $i \in D, j \in P$, (1) $a_{ij} = 1$ if and only if $j \in b_i$, and (2) $TC_j(\mathbf{a}) = q_j$.

The correctness is clear: if the answer in Step 2 is "yes", then configuration $\mathbf{q}$ gives a NE; if the answer is "no" for all $\mathbf{q} \in Q$, then there is no NE. Next, we analyze the time complexity. For each of the $r^m = |Q|$ configurations, the time for Step 1 is at most $O(nm^2)$ as we need to compare each project with $m-1$ other projects for each contributor.

Step 2 can be done efficiently using dynamic programming where (1) one can first order the contributor $1, ..., n$ and (2) create a binary table $T_i(\mathbf{q}) \in \{0, 1\}$ for each $\mathbf{q} \in Q$ of size r^m for each contributor i. Defining $T_0(\mathbf{0}) = 1$ (with entry of zero otherwise), $T_i(\mathbf{q}) = 1$ if only if there exists $T_{i-1}(\bar{\mathbf{q}}) = 1$ such that $\mathbf{q} = \bar{\mathbf{q}} + \mathbf{e}_k * s_{ik}$ for some project $k \in b_i$ where $\mathbf{e}_k$ is a binary vector of size m of all zero except the k^{th} entry. Table T_i can simply be constructed by looking at all the 1's $\bar{\mathbf{q}}$ entries of T_{i-1} and add s_{ik} to each entry for each $k \in b_i$. Because there are at most r^m configurations, and each of the n contributors have at most m choices/projects for contributing to the configurations, the time for this step is at most $O(nmr^m)$. To verify whether a given $\mathbf{q}$ can be achieved, one can check if $T_n(\mathbf{q}) = 1$ and the corresponding NE can be constructed via a standard backtracking procedure in dynamic programming.[3] The total time to check a given potential-sum $\mathbf{q}$ is $O(nmr^m + nm^2)$. Yielding $O(r^m(nmr^m + nm^2))$ to check all $\mathbf{q} \in Q$, which is polynomial in n and r, when m is a constant for each configuration. $\square$

5 Inefficiency of Nash Equilibria in EACPP

In this section, we consider the inefficiency, measured with respect to the planner's objective, induced by the strategic behavior of contributors. To measure such an inefficiency, we invoke the notion of the *price of anarchy* (PoA) [Koutsoupias and Papadimitriou, 1999] which is the ratio between the optimal "centralized" efficiency and the worst-case NE's efficiency. Also, we consider the *price of stability* (PoS) [Anshelevich *et al.*, 2008], which is the ratio between the optimal efficiency and the best NE's efficiency.

The following result shows that computing optimal solutions is NP-hard for either maximizing the social welfare, or maximizing the successful projects.

Theorem 6. *It is NP-hard to find an optimal solution that maximizes social welfare, or to find an optimal solution that maximizes the number of successful projects, even if $h_i = 1, c_{ij} = 0, v_{ij} = v$.*

Proof (Sketch). For the social welfare objective, we can reduce from the PP problem where we are given a set of n integers and a target T. The constructed EACPP instance consists of two projects with threshold T, where each of the n contributors can only select one project with value of T, and has skill s_i (corresponding to the integer) and zero cost for each project. One can obtain a social welfare no less than $2T$ if and only if there is a solution to the PP problem.

For the project successful objective, we can reduce from the 3-Partition Problem (3PP), which given a set of $3n$ positive integers, seeks a partition into n subsets of size 3 such that their sums are equal. It is easy to show that there is a solution to the 3-PP instance if and only if there is a solution of a constructed EACPP instance where all projects succeed. $\square$

[2] The mapping of $\mathbf{a}$ to $\mathbf{q}$ is neither one-to-one nor onto.

[3] If multiple NE correspond to the same $\mathbf{q}$, then there is one path from $T_n(\mathbf{q})$ to $T_0(\mathbf{0})$ for each NE.

In contrast, we remark that, for a more constrained setting where $h_i = 1, c_{ij} = 0, v_{ij} = v_j, t_j = 0$, maximizing social welfare can be done in polynomial time [Bilò *et al.*, 2019].

Next, we consider PoA and PoS which is defined to be $PoA^{sw} = \max_{\mathbf{a}\in\mathbb{NE}} \frac{SW(\mathbf{a}^{opt_{sw}})}{SW(\mathbf{a})}$, $PoS^{sw} = \min_{\mathbf{a}\in\mathbb{NE}} \frac{SW(\mathbf{a}^{opt_{sw}})}{SW(\mathbf{a})}$, and similarly for PoA^{ps} and PoS^{ps}, where $\mathbb{NE} \subseteq A$ is the set of all NEs. Clearly, $PoS \leq PoA$. Notice that we focus only on the instances in which NE exists.[4] The following results show that the PoS can be unbounded for either objective.

Theorem 7. *PoS^{ps} is unbounded, even if $h_i = 1$ and $s_{ij} = s$. PoS^{sw} is also unbounded, even if $h_i = 1, s_{ij} = s, c_{ij} = 0$.*

Proof. We first consider PoS^{ps}. Consider the instance $\mathcal{I} = (D, P, \{s_{ij}\}, \{c_{ij}\}, \{v_{ij}\}, \{t_j\})$ where $|D| = 2, |P| = 2, s_{ij} = 1, v_{i1} = 0, c_{i1} = 0, c_{i2} = 1$ for $i \in D, j \in P$. The values for project 2 are $v_{12} = 0, v_{22} = 3$. The thresholds are $t_1 = 3, t_2 = 2$. Note that project 1 can never be completed and project 2 can only be completed when $a_{12} = a_{22} = 1$. Agent 1 will never choose project 2 because it has a cost $c_{12} = 1$ and a value $v_{12} = 0$, while the cost of project 1 is $c_{11} = 0$. Agent 2 will choose project 2 only if the threshold is reached. Therefore, the only NE is $a_{11} = a_{21} = 1$ where 0 projects are completed. However, the optimal solution is $a_{12} = a_{22} = 1$ where 1 project is completed.

Next, we consider PoS^{sw}. Consider an instance $\mathcal{I} = (D, P, \{s_{ij}\}, \{v_{ij}\}, \{t_j\})$ where $|D| = |P| = 2, t_1 = 0, t_2 = 2, h_i = 1, s_{ij} = 1$ and $c_{ij} = 0$ for any $i \in D, j \in P$. For contributor 1, $v_{11} = 1, v_{12} = 0$, and for contributor 2, $v_{21} = 1, v_{22} = L$, where L is a sufficiently large number. We show that, there is a NE with $a_{11} = a_{21} = 1$, which has a social welfare 2. Clearly contributor 1 has no incentive to deviate because it has positive value only for project 1. For contributor 2, the utility is 1, and if it deviates to project 2, then the utility decreases to 0. Moreover, noting that the only way for both contributors having positive utility is $a_{11} = a_{21} = 1$, this is the unique NE. However, the optimal solution is that both contributors select project 2 (*i.e.,* $a_{12} = a_{22} = 1$), and the optimal social welfare is L. The ratio between the optimal social welfare and the social welfare in the NE is $\frac{L}{2}$, which tends to ∞ when $L \to \infty$. Thus, PoS^{sw} is unbounded. ☐

Despite the negative results above, we are able to identify instances in which the PoA is bounded for both objectives. For the social welfare objective, [Bilò *et al.*, 2019] proved that, when $h_i = 1, t_j = 0, c_{ij} = 0, v_{ij} = v$, it always has $PoS^{sw} = PoA^{sw} = 1$; and if the condition $v_{ij} = v$ is relaxed to $v_{ij} = v_j$, then it always has $PoA^{sw}(n, m) \leq 1 + \frac{\min\{n,m\}-1}{n}$. Further, for the successful project objective, we have the following results.

Theorem 8. *When $h_i = 1$ and $c_{ij} = 0$ for all $i \in D, j \in P$, we have $PoS^{ps} = n$.*

[4]We note that PoA^{sw} or PoS^{sw} typically assume both of the quantities in the fraction to be non-negative. If both quantities are negative, then we can flip the fraction. If the numerator and denominator are positive and negative, respectively, then we say that the PoA is unbounded. When the denominator is zero, we say that the value is one if the numerator is zero, otherwise it is unbounded.

Proof. We first prove $PoS^{ps} \leq n$. The optimal solution has at most $\min\{m, n\}$ successful projects. Note that we focus on instances in which NE exists. It suffices to prove the existence of a NE with at least one successful project.

If there is a project $j \in P$ and contributor i satisfying $s_{ij} \geq t_j$ and $v_{ij} > 0$, then every NE must have at least one successful project, otherwise contributor i can deviate to j and obtain a positive utility. So we only need to consider the case where that does not hold. We construct a NE $\mathbf{a}$ with at least one successful project as follows. Let $j^* \in P$ be an arbitrary project that has potential to be successful. For any $i \in D$, set $a_{ij^*} = 1$, and set $a_{ij} = 0$ for all $j \neq j^*$. Clearly, project j^* is successful. No contributor has incentive to deviate, because no project can be satisfied by a single contributor who has positive value to this project. Hence, action profile $\mathbf{a}$ is a NE with at least one successful project.

Now we prove $PoS^{ps} \geq n$. Consider an instance $\mathcal{I} = (D, P, \{s_{ij}\}, \{v_{ij}\}, \{t_j\})$ where $|D| = |P| = n, t_j = 0, s_{ij} = 1$ for $i \in D, j \in P$. For $i \in D$, set $v_{ij^*} = 1$ and $v_{ij} = 0$ for all project $j \neq j^*$. Clearly, the optimal solution that maximizes the number of successful projects is that every project is contributed by a unique contributor, and the number of successful projects is n. However, the unique NE is that all contributors contribute to project j^*, where every one has a positive utility, and the number of successful projects is 1. Hence, the PoS is at least n. ☐

6 Conclusion

In this paper, we consider a game-theoretic model of how contributors allocate effort (*e.g.,* time) to contribute to a set of public projects where each project is successful if the total contribution meets or exceeds a predetermined project threshold. We study several computational and efficiency questions associated with pure-strategy Nash equilibria (NE). In particular, we show that the computational problems (*i.e.,* existence of NE with/without required properties) are generally hard, and NE can be very inefficient (*i.e.,* PoS). We note that the major obstacle for obtaining positive results is the existence of *minimum threshold*, which ensures that no project receiving a total contribution below the threshold can be completed. Compared with Project Games studied in [Bilò *et al.*, 2019], the threshold version studied in our paper becomes a bit intractable in the sense of both NP-hardness and PoA. However, we show that when we further restrict the model parameters, various computational and efficiency NE questions become tractable and positive, respectively.

Regarding the future directions, many variants of EACPP could be investigated. Non-binary project outcomes could be modeled, and most notably, additional contributions beyond the minimum threshold could increase the quality or probability of success for a project. Other variants, such as public reward in addition to contributors' reward, and individuals' different skills (*e.g.,* programming, project management, marketing) for projects, are also possible. Such variations could be useful in a variety of applications, such as software engineering research [Stevens *et al.*, 2021]. Finally, with some modifications this model would be interesting to study as a cooperative game [Bachrach and Rosenschein, 2008].

References

[Anshelevich *et al.*, 2008] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[Arieli *et al.*, 2018] Itai Arieli, Moran Koren, and Rann Smorodinsky. The one-shot crowdfunding game. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 213–214, 2018.

[Azmat and Möller, 2009] Ghazala Azmat and Marc Möller. Competition among contests. *The RAND Journal of Economics*, 40(4):743–768, 2009.

[Azmat and Möller, 2017] Ghazala Azmat and Marc Möller. The distribution of talent across contests. *The Economic Journal*, 2017.

[Bachrach and Rosenschein, 2008] Yoram Bachrach and Jeffrey S Rosenschein. Coalitional skill games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 1023–1030, 2008.

[Bernergård and Wärneryd, 2017] Axel Bernergård and Karl Wärneryd. Self-allocation in contests. SSE Working Paper Series in Economics 2017:2, Stockholm School of Economics, 2017.

[Bilò *et al.*, 2019] Vittorio Bilò, Laurent Gourvès, and Jérôme Monnot. Project games. In *International Conference on Algorithms and Complexity*, pages 75–86. Springer, 2019.

[Chan *et al.*, 2020] Hau Chan, David C. Parkes, and Karim R. Lakhani. The price of anarchy of self-selection in tullock contests. In *AAMAS*, pages 1795–1797, 2020.

[Dechenaux *et al.*, 2015] Emmanuel Dechenaux, Dan Kovenock, and Roman M. Sheremeta. A survey of experimental research on contests, all-pay auctions and tournaments. *Experimental Economics*, 18(4):609–669, 2015.

[DiPalantino and Vojnovic, 2009] Dominic DiPalantino and Milan Vojnovic. Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, pages 119–128, 2009.

[Fotakis *et al.*, 2002] Dimitris Fotakis, Spyros Kontogiannis, Elias Koutsoupias, Marios Mavronicolas, and Paul Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In *ICALP*, pages 123–134. Springer, 2002.

[Fotakis, 2015] Dimitris Fotakis. *A Selective Tour Through Congestion Games*. Springer, 2015.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[Kempe *et al.*, 2020] David Kempe, Sixie Yu, and Yevgeniy Vorobeychik. Inducing equilibria in networked public goods games through network structure modification. In *AAMAS*, pages 611–619, 2020.

[Kleinberg and Oren, 2011] Jon Kleinberg and Sigal Oren. Mechanisms for (mis)allocating scientific credit. In *STOC*, page 529–538, 2011.

[Komarovsky *et al.*, 2015] Z. Komarovsky, V. Levit, T. Grinshpoun, and A. Meisels. Efficient equilibria in a public goods game. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 2, pages 214–219, 2015.

[Konrad, 2009] Kai A. Konrad. *Strategy and Dynamics in Contests*. Oxford University Press, 2009.

[Kontogiannis and Spirakis, 2005] Spyros Kontogiannis and Paul Spirakis. Atomic selfish routing in networks: A survey. In *Internet and Network Economics*, pages 989–1002, 2005.

[Koutsoupias and Papadimitriou, 1999] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *STACS 99*, pages 404–413, 1999.

[Mavronicolas *et al.*, 2007] Marios Mavronicolas, Igal Milchtaich, Burkhard Monien, and Karsten Tiemann. Congestion games with player-specific constants. In *Mathematical Foundations of Computer Science 2007*, pages 633–644, 2007.

[McConnell, 1996] Steve McConnell. *Rapid development: taming wild software shcedules*. Microsoft Press, 1996.

[Monderer and Shapley, 1996] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124 – 143, 1996.

[Morgan *et al.*, 2017] John Morgan, Dana Sisak, and Felix Várdy. The ponds dilemma. *The Economic Journal*, 2017.

[Rosenthal, 1973] Robert W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[Roughgarden, 2005] Tim Roughgarden. *Selfish Routing and the Price of Anarchy*. The MIT Press, 2005.

[Spolsky, 2004] Joel Spolsky. *Strategy Letter I: Ben & Jerry's vs. Amazon*. APress, 2004.

[Stevens *et al.*, 2021] Clay Stevens, Jared Soundy, and Hau Chan. Exploring the efficiency of self-organizing teams with game theory. In *ICSE-NIER 2021: 43nd International Conference on Software Engineering, New Ideas and Emerging Results*. ACM, 2021.

[Xu *et al.*, 2019] Jin Xu, Yves Zenou, and Junjie Zhou. Networks in conflict: A variational inequality approach. *SSRN Electronic Journal*, 2019.

[Yu *et al.*, 2020] Sixie Yu, Kai Zhou, P. Brantingham, and Yevgeniy Vorobeychik. Computing equilibria in binary networked public goods games. *AAAI*, 34:2310–2317, 2020.

Supplementary Material

Proof of Theorem 2

We prove the three claims one by one in the following.

Claim 1. *It is NP-complete to determine whether there is a NE with at least k successful projects.*

Proof. It is not hard to see that we can verify whether a given NE has at least k successful projects (both of which can be verified in polynomial time).

We reduce from the 3-PP problem. In a (feasible) 3-PP, we have a multiset $X = \{x_1, ..., x_{3n}\}$ of $3n$ positive integers with positive integer target sum $T = \frac{\sum_{x \in X} x}{n}$ and $\frac{T}{4} < x_i < \frac{T}{2}$.

We consider our EAPPC instance $I = (D, P, \{c_{ij}\}, \{s_{ij}\}, \{v_{ij}\}, \{h_i\}, \{t_j\})$ via the following:

- Let $m = n$ be the number of projects;
- Let $3n$ be the number of contributors;
- Let $h_i = 1$ for each $i \in D$;
- Let $s_{ij} = x_i$ and $c_{ij} = 0$ for each $i \in D$ and $j \in P$;
- For each $j \in P$, let $t_j = T$;
- For each $i \in D$ and $j \in P$, let $v_{ij} = v$ for some v.
- Let $k = m$ for the successful projects.

3-PP solution $\implies$ NE at least k projects. Given a solution, $X_1, ..., X_n$, each has a sum of T, for the 3-PP instance, we can construct a NE with at least k successful projects. In particular, for each $x_i \in X_l$, $a_{il} = 1$ for $l = 1, ..., n$ (and $a_{il'} = 0$ otherwise for $l' \neq l$). Since the total contribution is T for each project, each i with $a_{il} = 1$ would have no incentive to deviate (i.e., $v_{il} \frac{x_i}{T} \geq v_{il'} \frac{x_i}{T + x_i}$ for any l').

NE at least k projects $\implies$ 3-PP solution. Suppose there is a NE with at least k successful projects. This implies that all projects are successful, and each project has the total contribution of exactly T (i.e., if there is a project with total contribution $> T$, there will be a project with total contribution $< T$ where contributors in the project will have an incentive to deviate). For each project, there must be ≥ 3 contributors (since $\frac{T}{4} < x_i < \frac{T}{2}$). Since there are $3n$ contributors and n projects, each project must have exactly 3 contributors in the NE. Thus, we can construct a solution to 3-PP by letting X_j be x_i that corresponds to contributor i in project j. □

Claim 2. *It is NP-complete to determine whether there is a NE with social welfare of at least V.*

Proof. The proof uses a similar argument in the proof of Theorem 1 – we take an instance of PP and reduce it to an instance of EAPPC. The reduction is identical except when we set v_{ij} for each $i \in D$ and $j \in P$ (last two items of the list in the proof of Theorem 1) where we set the appropriate values via the following construction:

- For each $i \in D \setminus \{n\}$ and $j \in P \setminus \{1, 2\}$, $v_{ij} = v_i > 0$ and $v_{i1} = v_{i2}$ should be such that (1) $v_{i1} \frac{s_{ij}}{T} = s_{ij}$ ($\implies v_{i1} = v_{i2} = T$) and (2) $v_{i1} \frac{s_{ij}}{T} > v_i > v_{i1} \frac{s_{ij}}{T + 0.5}$;
- For n and $j \in P \setminus \{1, 2\}$, $v_{nj} = v_n > 0$ and $v_{n1} = v_{n2}$ such that (1) $v_{n1} \frac{s_{nj}}{2T} = s_{nj}$ and (2) $v_{n1} \frac{s_{nj}}{2T} > v_n$;
- Finally, we set $V = T$.

It is not hard to see that the arguments (in the proof of Theorem 1) would follow since there is a NE with a social welfare of at least V for our instance if and only if there is a solution for the PP instance. □

Our next computational question is related to a fixed assignment of some contributors to some projects. More specifically, given a partially fixed allocation profile $\mathbf{a}_S \in A_S$ for some contributors $S \subseteq D$, we would like to determine if there is a NE $\mathbf{a}^*$ that is consistent with the fixed allocation profile where $\mathbf{a}^*_S = \mathbf{a}_S$.

Claim 3. *It is NP-complete to determine whether there is a NE that is consistent with some fixed assignment $\mathbf{a}_S \in A_S$ for some contributors $S \subseteq D$ in EAPPC.*

Proof. The proof in this theorem uses the same reduction and construction as the proof of Theorem 1. After the construction, we fix the assignment of n to be in project 1 (i.e., $a_{n1} = 1$ and $a_{nj} = 0$ for other j).

To argue that a solution, say X_1 and X_2, of PP can be mapped to a NE where n is in project 1, notice that x_n must be in one of the two partitions, say in X_2. For the corresponding contributors in X_2, they will take the action of selecting project 1. Following the same argument as in the proof of Theorem 1, this is a clearly a NE and n is fixed to project 1. Next, to argue that a NE with n fixed in project 1 can be mapped to a solution of PP, notice that in any NE, n has to be in project 1 or project 2. If there is a NE where n is in project 2, we can swap the contributors of the two (homogeneous) projects and obtain another (isomorphic) NE. Following the proof of Theorem 1, we can construct a solution for PP using the NE. □