Mechanisms for (Mis)allocating Scientific Credit

Jon Kleinberg *　　　Sigal Oren †

November, 2010

Abstract

Scientific communities confer many forms of *credit* — both implicit and explicit — on their successful members, and it has long been argued that the motivation provided by these forms of credit helps to shape a community's collective attention toward different lines of research. The allocation of scientific credit, however, has also been the focus of long-documented pathologies: certain research questions are said to command too much credit, at the expense of other equally important questions; and certain researchers (in a version of Robert Merton's *Matthew Effect*) seem to receive a disproportionate share of the credit, even when the contributions of others are similar.

Here we show that the presence of each of these pathologies can in fact increase the collective productivity of a community. We consider a model for the allocation of credit, in which individuals can choose among *projects* of varying levels of importance and difficulty, and they compete to receive credit with others who choose the same project. Under the most natural mechanism for allocating credit, in which it is divided among those who succeed at a project in proportion to the project's importance, the resulting selection of projects by self-interested, credit-maximizing individuals will in general be socially sub-optimal. However, we show that there exist ways of allocating credit out of proportion to the true importance of the projects, as well as mechanisms that assign credit out of proportion to the relative contributions of the individuals, that lead credit-maximizing individuals to collectively achieve social optimality. These results therefore suggest how well-known forms of misallocation of scientific credit can in fact serve to channel self-interested behavior into socially optimal outcomes.

*Department of Computer Science, Cornell University, Ithaca NY 14853. Email: kleinber@cs.cornell.edu. Supported in part by a John D. and Catherine T. MacArthur Foundation Fellowship, a Google Research Grant, a Yahoo! Research Alliance Grant, and NSF grants IIS-0705774, IIS-0910664, and CCF-0910940.

†Department of Computer Science, Cornell University, Ithaca NY 14853. Email: sigal@cs.cornell.edu. Supported in part by NSF grant CCF-0910940.

1 Introduction

As a scientific community makes progress on its research questions, it also develops conventions for allocating *credit* to its members. Scientific credit comes in many forms; it includes explicit markers such as prizes, appointments to high-status positions, and publication in prestigious venues, but it also builds upon a broader base of informal reputational measures and standing within the community [3, 12, 15]. The mechanisms by which scientific credit is allocated have long been the subject of fascination among scientists, as well as a topic of research for scholars in the philosophy and sociology of science. A common theme in this line of inquiry has been the fundamental ways in which credit seems to be systematically *misallocated* by scientific communities over time — or at least allocated in ways that seem to violate certain intuitive notions of "fairness." Two categories of misallocation in particular stand out, as follows.

1. *Certain research questions receive an "unfair" amount of credit.* In other words, a community will often have certain questions on which progress is heavily rewarded, even when there is general agreement that other questions are equally important. Such issues, for example, have been at the heart of recent debates within the theoretical computer science community, focusing on the question of whether conference program committees tend to overvalue progress on questions that display "technical diffculty" [1, 9].

2. *Certain people receive an "unfair" amount of credit.* Robert Merton's celebrated formulation of the *Matthew Effect* asserts, roughly, that if two (or more) scientists independently or jointly discover an important result, then the more famous one receives a disproportionate share of the credit, even if their contributions were equivalent [14, 15].[1] Other attributes such as affiliations or academic pedigree can play an analogous role in discriminating among researchers.

There is a wide range of potential explanations for these two phenomena, and many are rooted in hypotheses about human cognitive factors: a fascination with "hard" problems or the use of such problems to identify talented problem-solvers in the first case; the effect of famous individuals as focal points or the confidence imparted by endorsement from a famous individual in the second case [14, 22].

A model of competition and credit in science. One can read this state of affairs as a story of how fundamental human biases lead to inherent unfairness, but we argue in this paper that it is useful to bring into the discussion an alternate interpretation, via a natural formal model for the process by which scientists choose problems and by which credit is allocated.

We begin by adapting a model proposed in influential work of Kitcher in the philosophy of science [11, 12, 21], and with roots in earlier work of Peirce, Arrow, and Bourdieu [2, 5, 17]. Kitcher's model has some slightly complicated features that we do not need for our purposes, so we will focus the discussion in terms of the following closely related model; it is designed as a stylized abstraction of a community of n researchers who each choose independently among a set of m open problems to work on.

- The m open problems will also be referred to as *projects*. Each project j has an *importance* w_j (also called its *weight*), and a probability of success q_j (with a corresponding failure probability $f_j = 1 - q_j$). We assume these numbers are rational. The researchers will initially be modeled as identical, but we later consider generalizations to individuals with different problem-solving abilities.

- Each researcher must choose a single project to work on. We model researchers as working independently, so if k_j researchers work on project j, there is a probability of $(1 - f_j^{k_j})$ that at least one of them succeeds.

[1] This is a kind of rich-get-richer phenomenon, and Merton's use of the term "Matthew Effect" is derived from *Matthew 25:29* in the New Testament of the Bible, which says, "For unto every one that hath shall be given, and he shall have abundance: but from him that hath not shall be taken away even that which he hath."

- In the event that multiple researchers succeed at project j, one of them is chosen uniformly at random to receive an amount of credit equal to the project's importance w_j. (We can imagine there is a "race" to be the first to solve the problem, and the credit goes to the "winner" ; alternately, we get the same model if we imagine that all successful researchers divide the credit equally.)

Suppose that researchers are motivated by the amount of credit they receive: each researcher chooses a project to work on to maximize her expected amount of credit, given the choices of all other researchers. The selection of projects is thus a game, in which the players are the researchers, the strategies are the choices of projects, and the payoffs are the expected amount of credit received. This game-theoretic view forms the basis of Kitcher's model of scientific competition; the view itself was perhaps first articulated explicitly in this form by the social scientist Pierre Bourdieu [3, 5], who wrote that researchers' motivations

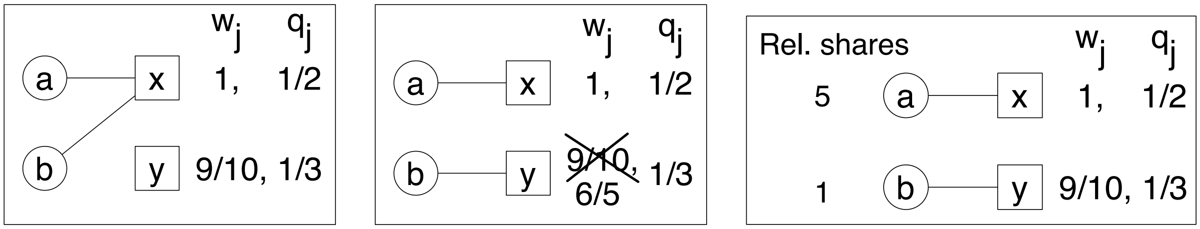
> are organized by reference to – conscious or unconscious – anticipation of the average chances of profit ... Thus researchers' tendency [is] to concentrate on those problems regarded as the most important ones ... The intense competition which is then triggered off is likely to bring about a fall in average rates of symbolic profit, and hence the departure of a fraction of researchers towards other objects which are less prestigious but around which the competition is less intense, so that they offer profits of at least as great.

Like the frameworks of Bourdieu and Kitcher, our model is a highly simplified version of the actual process of selecting research projects and competing for credit. We are focusing on projects that can be represented as problems to be solved; we are not modeling the process of collaboration among researchers, the ways in which problems build on each other, or the ways in which new problems arise; and we are not trying to capture the multiple ways in which one can measure the importance or difficulty of a problem. These are all interesting extensions, but our point is to identify a tractable model that contains the fundamental ingredients in our discussion: a competition for credit among projects of varying difficulty, in a way that causes credit-seeking individuals to distribute themselves across different projects. We will see how phenomena that are complex but intuitively familiar can arise even when a community has a single, universally agreed-upon measure of importance and difficulty across projects.

Credit as a mechanism for allocating effort. Our main focus is to extend this class of models to consider the issues raised at the outset of the paper, and in particular to the two sources of "unfairness" discussed there. The model we have described thus far is based on an intuitively fair allocation of credit that doesn't suffer from either of these two pathologies: all researchers are treated identically, and the credit a successful researcher receives is equal to the community's agreed-upon measure of the importance of the problem solved. In other words, no problems are overvalued relative to their true importance, and no researchers are *a priori* favored in the assignment of credit.

As a first thought experiment, suppose that we were allowed to design the rules by which credit was assigned in a research community; are these "fair" rules the ones we should use? The following very small example shows the difficulties we quickly run into. Suppose, for simplicity, that we are dealing with a community consisting of two players a and b, and two projects x and y. Project x is more important and also easier; it has $w_x = 1$ and $q_x = 1/2$. Project y is less important and more difficult; it has $w_y = 9/10$ and $q_y = 1/3$. Figure 1(a) shows the unique Nash equilibrium for this research community: both players work on x, each receiving an expected payoff $3/8$ (since project x will be solved with probability $3/4$, and a and b are equally likely to receive credit for it.)

If we were in charge of this research community, arguably the natural objective function for us to care about would be the *social welfare*, defined as the total expected importance of all projects successfully completed. And now here's the difficulty: the unique Nash equilibrium does not maximize social welfare. It produces a social welfare of $3/4$, whereas if the players divided up over the two different projects, we would obtain a social welfare of $1/2 + 3/10 = 4/5$.



(a) *Projects with original weights.* (b) *Projects with modified weights.* (c) *Players with modified credit shares.*

Figure 1: In (a), self-interested players do not reach a socially optimal selection of projects. However, if the weight of project y is increased (b), or if one of the players is guaranteed a sufficiently disproportionate share of the credit in the event of joint success (c), then a socially optimal assignment of players to projects arises.

Can we change the way credit is assigned so as to create incentives for the players under which the resulting Nash equilibrium maximizes social welfare? In fact, there are two natural ways to do this, and each should be recognizable given the discussion at the beginning of the introduction.

- First, we could declare that the credit received for succeeding at a project will not be proportional to its importance. Instead, in our example, we could decide that success at the harder project y will bring an amount of credit equal to $w'_y \neq w_y$. If $w'_y > 9/8$, then the unique Nash equilibrium is socially optimal (Figure 1(b)).
- Alternately we could declare that if players a and b both succeed at the same project, they will not split the credit equally, but instead in a ratio of c to 1. (Equivalently, if they both succeed, player a is selected to receive all the credit with probability $c/(c+1)$ and player b with probability $1/(c+1)$.) If $c > 4$, then it is not worth it for b to try competing with a on project x, and b will instead work on project y, again leading to a socially optimal Nash equilibrium (Figure 1(c)).

This example highlights several points. First, we can think of the amount of credit associated with different projects as something malleable; by choosing to have certain projects confer more credit, the community can create incentives that cause effort to be allocated in different ways. Second, it is clearly the case that actual research communities engage in this shaping of credit, not just at an implicit level but through a variety of explicit mechanisms: the decisions of program committees and editorial boards about which papers to accept, the decisions of hiring committees about which people to interview and areas to recruit in, and the decisions of granting agencies about funding priorities all serve to shift the amounts of credit assigned to different kinds of activities. In this sense, a research community is, to a certain extent, a kind of "planned economy" — it is much more complex than our simple model, but many of its central institutions have the effect of deliberately implementing and publicizing decisions about the allocation of credit for different kinds of research topics.

What we see in the example is that the "fair" allocation of credit can be at odds with the goal of social optimality: if the community believes that as a whole it is being evaluated according to the total expected weight of successful projects, then by rewarding its participants according to these same weights, it produces a socially sub-optimal outcome. The two alternate ways of assigning credit above correspond to the two forms of "unfairness" discussed at the outset: overvaluing certain projects (in our example, the harder and less important project), and overvaluing the contributions of certain researchers. If done appropriately in this example, either of these can be used to achieve social optimality.

As a final point on the underlying motivation, we are not claiming that research communities are overtly trying to assign credit in a way that achieves social optimality, or arriving at credit allocations in general through explicit computation. It is clear that the human cognitive biases discussed earlier — in favor of certain topics and certain people — are a large and likely dominant contributor to this. What we do see, however, is that social optimality plays an important and surprisingly subtle role in the discussion about these issues: institutions such as program committees and funding agencies do take into account the goal of shaping the kind of research that gets done, and to the extent that these cognitive biases can sometimes — paradoxically — raise the overall productivity of the community, it arguably makes such biases particularly hard to eliminate from people's behavior.

Social optimality and the misallocation of credit: General results. Our main results begin by establishing that the two kinds of mechanisms suggested by the example in Figure 1 are each sufficient to ensure social optimality in general — that is, in all instances. For any set of projects, it is possible to assign each project j a *modified weight* w'_j, potentially different from its real weight w_j, so that when players receive credit according to these modified weights, all Nash equilibria are socially optimal with respect to the real weights. It is also possible to assign each player i a weight z_i so that when players divide credit on successful projects in proportion to their weights z_i, all Nash equilibria are again socially optimal. This makes precise the sense in which our two categories of credit misallocation can both be used to optimize social welfare.

These results in fact hold in a generalization of the basic model, in which the players are heterogeneous and have different levels of *ability* at solving problems. In this more general model, a player's success at a project depends on both her ability and the project's difficulty: each player i has a parameter $p_i \leq 1$ such that her probability of succeeding at project j is equal to the product $p_i q_j$. Beyond this, the remaining aspects of the model remain the same; in particular, if multiple players all succeed at the same project, then one is selected uniformly at random to receive the credit. (That is, their ability affects their chance of succeeding, but not their share of the credit.) For this more general game, there still always exist re-weightings of projects and also re-weightings of credit shares to players that lead to socially optimal Nash equilibria.

Our results make use of the fact that the underlying game, even in its more general form with heterogeneous players, is both a *congestion game* [16, 18] and a *monotone valid-utility game* [8, 23, 24]. However, given the motivating setting for our analysis, we have the ability to modify certain parameters of the game — as part of a research community's mechanism for allocating credit — that are not normally under the control of the modeler. As a result, our focus is on somewhat different questions, motivated by these credit allocation schemes. At the same time, there are interesting analogies to issues in congestion games from other settings. Re-weighting the amount of credit on projects can be viewed as a kind of "toll" system, interpreting the effort of the researchers as the "traffic" in the congestion game. The crux of our analysis for re-weighting the players is to begin by considering an alternate model in which an ordering is defined on the players, and the first player in this ordering to succeed receives all the credit. This suggests interesting potential connections with the theory of *priority algorithms* introduced by Borodin et al. [4]; although the context is quite different, we too are asking whether there is a "greedy ordering" that leads to optimality. A related set of questions was considered by Strevens in his model of sequential progress on a research problem, working within Kitcher's model of scientific competition [21].

We also consider some of the structural aspects of the underlying game; among other results, we show that the price of anarchy of the game is always strictly less than 2 (compared with a general upper bound of 2, which can sometimes be attained, for fully general monotone valid-utility games). For the case of identical players, we also show that the ratio of the price of stability to the price of anarchy (i.e. the welfare of the best Nash equilibrium relative to the worst) is at most $3/2$. In particular, this implies that when there exists a Nash equilibrium that is optimal, there is no Nash equilibrium that is less than $2/3$ times optimal.

Finally, we consider a still more general model, in which player success probabilities are arbitrary and unrelated: player i has a probability p_{ij} of succeeding on project j. We show that there exist instances of this general game in which no re-weighting of the projects yields a social optimal Nash equilibrium. Similarly, there are instances in which no strict ordering on the players yields a socially optimal Nash equilibrium, although the power of more general re-weighting of players remains open.

Interpreting the model. With any simple theoretical model of a social process — in this case, credit among researchers — it is important to ask whether the overall behavior of the model captures fundamental qualitative aspects of the real system's behavior. In this case we argue that it captures several important phenomena at a broad level. First, it is based on the idea that institutions within a research community can and do shift the amount of credit that different research topics receives, and in a number of cases with the goal of creating corresponding incentives toward certain research directions. Second, it argues that some of the typical ways in which credit is misallocated can interact in a complex fashion with social welfare, and that these misallocations can in fact play an important role in the maximization of welfare.

Moreover, there is a rapidly widening scope for the potential application of explicitly computational approaches to credit-allocation, as we see an increasing number of intentionally *designed* systems aimed at fostering massive Internet based-collaboration — these include large open-source projects, collaborative knowledge resources like Wikipedia, and collective problem-solving experiments such as the Polymath project [10]. For example, a number of credit-allocation conventions familiar from the scientific community have been built into Wikipedia, including the ways in which editors compete to have articles "featured" on the front page of the site [20], and the ways in which they go through internal review and promotion processes to achieve greater levels of status and responsibility [6, 13]. While the framework in this paper is only an initial foray in this direction, the general issue of designing credit-allocation schemes to optimize collective productivity becomes an increasingly wide-ranging question.

Finally, the model offers a set of simple and, in the end, intuitively natural interpretations for the specific ways in which misallocation can lead to greater collective productivity. The re-weighting of projects not only follows the informal roadmap contained in Pierre Bourdieu's quote above, but sharpens it. Even without re-weighting of projects, the effect of competition does work to disperse some number of researchers out to harder and/or less attractive projects, which helps push the system toward states of higher social welfare. But the point is that this dispersion is not optimally balanced on its own; it needs to be helped along, and this is where the re-weighting of projects comes into play. The re-weighting of players is based on a different point — that when certain individuals are unfairly marginalized by a community, it can force them to embark on higher-risk courses of action, enabling beneficial innovation that would otherwise not have happened. In all these cases, it does not mean that such forms of misallocation are fair to the participants in the community, only that they can sometimes have the effect of increasing the community's overall output.

2 Identical Players

We first consider the case of the *Project Game* defined in the introduction when all players are identical, and then move on to the case in which players have different levels of ability. Recall that w_j denotes the weight (i.e. importance) of project j, and f_j denotes the probability that any individual player fails to succeed at it. Thus, when there are k players working on project j, the contribution of project j to the social welfare is $w_j(1 - f_j^k)$, and we denote this quantity by $\sigma_j(k)$.

We denote the choices of all players by a *strategy vector* $\vec{a}$, in which player i chooses to work on project a_i. As is standard, we denote by a_{-i} the strategy vector $\vec{a}$ without the i^{th} coordinate and by j, a_{-i} the strategy vector $a_1, \ldots, a_{i-1}, j, a_{i+1}, \ldots, a_n$. We use $K_j(\vec{a})$ to denote the set of players working on project j in strategy vector $\vec{a}$, and we write $k_j(\vec{a}) = |K_j(\vec{a})|$. The social welfare obtained from strategy vector $\vec{a}$ is $u(\vec{a}) = \sum_{j \in M} \sigma_j(k_j(\vec{a}))$. Since each player is equally likely to receive the credit on a project, the payoff, or utility, of player i under strategy vector $\vec{a}$ is $u_i(\vec{a}) = \dfrac{\sigma_{a_i}(k_{a_i}(\vec{a}))}{k_{a_i}(\vec{a})}$.

We make a few observations about these quantities. First, as noted in the introduction, $u_i(\vec{a})$ is the utility of i regardless of whether we interpret the credit as being assigned uniformly at random to one successful player on a project, or divided evenly over all successful players. Moreover, since the players divide up the social welfare among themselves, we have $\sum_{i \in N} u_i(\vec{a}) = u(\vec{a})$. Since a player's utility depends solely on the number of other players choosing her project, it is not hard to verify that the game with identical players is a congestion game, and hence has pure Nash equilibria. Finally, it will be useful in some of the proofs to define the marginal utility $r_j(k)$ from joining project j when k players are already working on it; this is $r_j(k) = (1 - f_j) f_j^k$. Notice that $r_j(k)$ is decreasing in k.

We begin by developing some basic properties of the social optimum and of the set of Nash equilibria with identical players; we then build on this to prove bounds on the price of anarchy (the ratio of the social welfare of the worst Nash equilibrium to the social optimum) and the price of stability (the analogous ratio of the best Nash equilibrium to the social optimum). After this, we provide algorithms for re-weighting projects and re-weighting players so as to produce Nash equilibria that are socially optimal.

Before proceeding, we first state four basic claims about the game with identical players. The simple proofs of these are given in the appendix.

Claim 2.1 *The Project Game with Identical Players is a monotone valid-utility game.*

Claim 2.2 *The social optimum can be achieved by the following greedy algorithm: players are assigned to projects one at a time, and in each iteration an assigned player is placed on a project j with the greatest current marginal utility $r_j(k_j)$.*

Claim 2.3 *A Nash equilibrium can be computed in polynomial time by the following algorithm: players choose projects one at a time in an arbitrary order, and in each iteration the current player i chooses a project that maximizes his utility based on the choices made by earlier players.*

Claim 2.4 *For every two different Nash equilibria $\vec{a}$ and $\vec{b}$ and for every two projects j,l such that $k_j(\vec{a}) > k_j(\vec{b})$ and $k_l(\vec{a}) < k_l(\vec{b})$, we have the following relationships: $k_j(\vec{a}) = k_j(\vec{b}) + 1$ and $k_l(\vec{b}) = k_l(\vec{a}) + 1$.*

The Price of Anarchy and Price of Stability. From Claim 2.1, by a result of Vetta [24], it follows that the price of anarchy (PoA) of the game is at most 2. Here we provide a strengthened analysis of the price of anarchy that yields several consequences:

(i) a bound of $1 + \frac{c-1}{c}$ on the PoA for instances in which the worst Nash equilibrium has at most c players assigned to any single project;

(ii) as a corollary of (i), a general upper bound of $2 - \frac{1}{n}$ on the PoA for any instance; and

(iii) a bound of $\frac{3}{2}$ between the price of anarchy and the price of stability (PoS) for any instance.

None of (i)-(iii) hold for monotone valid-utility games in general.

We first show that these bounds are tight, by means of the following example. Consider an instance with n players and n projects; all projects are guaranteed to succeed (i.e. $q_j = 1$ for all j); and the weights of the projects are defined so that $w_1 = 1$ and $w_j = 1/n$ for $j > 1$. The socially optimal assignment of players to projects in this game is for each player to work on a different project, yielding a social welfare of $2 - \frac{1}{n}$. On the other hand, it is a Nash equilibrium for every player to work on project 1, yielding a social welfare of 1. Furthermore, in the case of this example when $n = 2$, the social optimum is also a Nash equilibrium, establishing a gap of $3/2$ between the PoA and PoS in this case. (We also note that for general n, if we increase the weight of project 1 by arbitrarily little, then we obtain an example in which the PoS is arbitrarily close to $2 - \frac{1}{n}$.)

To prove the upper bounds in (i)-(iii), we use Roughgarden's notion of *smoothness* [19].

Definition 2.5 *A monotone valid-utility game is (λ, μ)-smooth if for every two strategy vectors $\vec{a}$ and $\vec{b}$, we have $\sum_{i \in N} u_i(b_i, a_{-i}) \geq \lambda u(\vec{b}) - \mu u(\vec{a})$.*

The following is a useful claim based on Roughgarden's paper (the simple proof is in the appendix):

Claim 2.6 *If a monotone valid-utility game is (λ, μ)-smooth then for every Nash equilibrium $\vec{a}$ and every strategy vector $\vec{b}$, we have $\frac{u(\vec{b})}{u(\vec{a})} \leq \frac{1+\mu}{\lambda}$. Applying this bound with $\vec{a}$ equal to the worst Nash equilibrium and $\vec{b}$ equal to the optimal assignment, it follows that the price of anarchy is at most $\frac{1+\mu}{\lambda}$.*

In the appendix, we prove a sequence of claims about the smoothness of the Project Game with identical players, leading up to the following result.

Theorem 2.7 *The Project Game with Identical Players is (λ, μ)-smooth for $\lambda = 1$ and*

$$\mu = \max_{\{l \mid k_l(\vec{a}) > k_l(\vec{b}) > 0\}} \frac{k_l(\vec{a}) - k_l(\vec{b})}{k_l(\vec{a}) - k_l(\vec{b}) + 1}.$$

Consequences (i) and (ii) above follow directly from Theorem 2.7 together with Claim 2.6.

To obtain consequence (iii), we call a game *weakly-(λ, μ)-smooth* provided the (λ, μ)-smoothness condition holds just for all Nash equilibria $\vec{a}$ and $\vec{b}$, rather than all arbitrary strategy vectors. Now, for any two Nash equilibria $\vec{a}$ and $\vec{b}$, Claim 2.4 implies that the number of players working on each project in $\vec{a}$ and $\vec{b}$ can differ by at most one. Hence, by Theorem 2.7 we have that the Project Game with Identical Players is weakly-(λ, μ)-smooth for $\mu = \frac{1}{2}$. We can now apply Claim 2.6 with $\vec{a}$ equal to the worst Nash equilibrium and $\vec{b}$ equal to the best Nash equilibria to get that $\frac{u(\vec{b})}{u(\vec{a})} \leq \frac{3}{2}$.

Re-weighting Projects to Achieve Social Optimality. We now describe a mechanism for re-weighting projects so as to achieve social optimality. As discussed in the introduction, we show that it is possible to assign new weights $\{w'_j\}$ to the projects so that when utilities are allocated according to these new weights, all Nash equilibria are socially optimal. Note that the re-weighting of projects only affects players' utilities, not the social welfare, as the latter is still computed using the true weights $\{w_j\}$.

The idea is to choose weights so that when players are assigned according to the social optimum, they all receive identical utilities. The following re-weighting accomplishes this: we compute a socially optimal assignment $\vec{o}$, and define $w'_j = \frac{k_j(\vec{o})}{(1 - f_j^{k_j(\vec{o})})}$ for $k_j(\vec{o}) > 0$ and $w'_j = 0$ otherwise. In the appendix, we prove the following result.

Theorem 2.8 *With these weights, all Nash equilibria achieve the social welfare of assignment $\vec{o}$.*

It is interesting to reflect on the qualitative interpretation of these new weights for an instance with n players and a very large set of projects of equal weight and with success probabilities $q_1 \geq q_2 \geq q_3 \geq \cdots$ decreasing to 0. In this case, there will be a largest j^* for which the optimal assignment places any players on j^*, and computational experiments with several natural distributions of $\{q_j\}$ indicate that the number of players assigned to projects increases roughly monotonically toward a maximum approximately near j^*. This means that the credit assigned to projects must increase toward j^*, and then be chosen so as to discourage players from working on projects beyond j^*. Moreover, the value of j^* grows with n, the number of players. Hence we have a situation in which the research community can be viewed, roughly, as establishing the following coarse division of its projects into three categories: "too easy" (receiving relatively little credit), "just right" (near j^*, receiving an amount of credit that encourages extensive competition on these projects), and "too hard" (beyond j^*, receiving an amount of credit designed to dissuade effort on these projects). Moreover, smaller research communities reward easier problems (since j^* is smaller), while larger communities focus their rewards on harder problems.

Re-weighting Players to Achieve Social Optimality. We now discuss the companion to the previous analysis: a mechanism for re-weighting the players to achieve social optimality. Recall that this means we assign each player i a weight z_i, and when a set S of players succeeds at a project j, we choose player $i \in S$ to receive the credit w_j with probability $\frac{z_i}{\sum_{h \in S} z_h}$.

When players are identical, we can base the re-weighting mechanism on the optimality of the greedy algorithm expressed in Claim 2.2. That is, if we were to assign an absolute order to the players, and announce the convention that credit would go to the first player in the order to succeed at a project, then the players' simultaneous choices would simulate the greedy algorithm to achieve social optimality: the first player in the announced order would choose a project without regard to the choices of other players; the second player would choose as though the first player would win any direct competition, but without regard to the choices of any other players; and so forth. Now, instead of an order, we need to define weights on the players; but we can approximately simulate the order using sharply decreasing weights in which $z_i = \epsilon^i$ for an $\epsilon > 0$ chosen to be sufficiently small. The effect of these sharply decreasing weights is to ensure that a player i gets almost no utility from a project j if a player of higher weight also succeeds at j, and i gets almost all the utility from j if i is the player of highest weight to succeed at j. From this, we can show that each player's utility is roughly what it would be under an order on the players. In the appendix we prove that we can indeed find such an ϵ as required.

Theorem 2.9 *With $\epsilon > 0$ sufficiently small and the re-weighting of players defined by $z_i = \epsilon^i$, all Nash equilibria of the resulting game are socially optimal.*

Even given the informal argument above, the proof is complicated by the fact that, with positive weights on all players, their strategic reasoning is more complex than it would be under an actual ordering. To prove Theorem 2.9, we consider the relationship between the actual utilities of the re-weighted players for a given strategy vector $\vec{a}$, denoted $\widetilde{u_i}(\vec{a})$, and their "ideal" utility under the order we are trying to simulate, denoted $\widehat{u_i}(\vec{a})$. Recalling that the projects' weights and success probabilities are rational, let d be their common denominator. In the appendix, we first show that if these two different utilities are close enough with respect to d, then our approximate simulation of the an order using weights will succeed:

Claim 2.10 *If for every player i and project j we have $\widehat{u_i}(j, a_{-i}) - \frac{1}{4d^{n+1}} \leq \widetilde{u_i}(j, a_{-i}) \leq \widehat{u_i}(j, a_{-i}) + \frac{1}{4d^{n+1}}$, then any Nash equilibrium in the game with the weights $\{z_i\}$ is also an optimal assignment.*

We then prove that it is possible to choose ϵ sufficiently small that the bounds in Claim 2.10 will hold:

Claim 2.11 *There exists an ϵ such that for every player i and project j: $\widehat{u_i}(j, a_{-i}) - \frac{1}{4d^{n+1}} \leq \widetilde{u_i}(j, a_{-i}) \leq \widehat{u_i}(j, a_{-i}) + \frac{1}{4d^{n+1}}$.*

3 Players of Heterogeneous Abilities

We now consider the case in which players have different levels of ability. Recall from the introduction that in this model, each player i has a parameter $p_i \leq 1$, and her probability of success at project j is $p_i q_j$. As before, player i receives credit for her selected project a_i if she succeeds at it and is chosen, uniformly at random, from among all players who succeed at it. Player i's utility is the expected amount of credit she receives in this process.

Recall that $K_j(\vec{a})$ is the set of players working on project j in strategy vector $\vec{a}$; we write $s_j(K_j(\vec{a})) = w_j(1 - \prod_{i \in K_j(\vec{a})}(1 - p_i q_j))$ for the contribution of project j to the social welfare, so that the overall social welfare of $\vec{a}$ is $u(\vec{a}) = \sum_{j \in M} s_j(K_j(\vec{a}))$. We denote the marginal utility of adding player i to project j by $s_j(i|K_j(\vec{a})) = s_j(K_j(\vec{a}) \cup \{i\}) - s_j(K_j(\vec{a})) = w_j p_i q_j \prod_{l \in K_j(\vec{a})}(1 - p_l q_j)$ and we use $u(j|a_{-i}) = s_j(i|K_j(a_{-i}))$ to denote the marginal utility of player i choosing project j when the rest of the players choose a_{-i}.

To begin with, we can show the following basic facts about this general version of the game.

Claim 3.1 *The Project Game with Different Abilities is a monotone valid-utility game.*

Claim 3.2 *The Project Game with Different Abilities is a congestion game.*

Claim 3.3 *Computing the social optimum for the Project Game with Different Abilities is NP-hard.*

The proof of Claim 3.1 is very similar to the proof of Claim 2.1; the only part that changes in a non-trivial way is the proof that the utility function is submodular, and we include a proof of this fact in the appendix. We prove Claims 3.2 and 3.3 in the appendix. Note that Claim 3.2 is less clear-cut than in the case of identical players, since now the payoffs depend not just on the number of players sharing a project but on their identities. To bypass this we prove that the utility functions for the Project Game with Different Abilities obey a certain structural property that, by results of Monderer and Shapley [16], implies that the game is a congestion game.

There is a useful closed-form way to write i's utility, as follows. First, suppose that in strategy vector $\vec{a}$, player i selects project j, and let S denote the other players who select j. Then in order for i to receive the credit of w_j for the project, she has to succeed (with probability $p_i q_j$); moreover, some subset S' of the other players on j will succeed (with probability $\prod_{h \in S'} p_h q_j$) while the rest will fail (with probability $\prod_{h \in \{S - S'\}}(1 - p_h q_j)$), and i must be selected from among the successful players (with probability $\frac{1}{|S'|+1}$). Thus we have

$$u_i(\vec{a}) = w_j p_i q_j \sum_{S' \subseteq S} \left(\frac{1}{|S'|+1} \prod_{h \in S'} p_h q_j \prod_{h \in \{S - S'\}} (1 - p_h q_j) \right).$$

This summation over all sets S' is a natural quantity that is useful to define separately for future use; we denote it by $c_j(S)$ and refer to it as the *competition function* for project j. The competition function represents the expected reduction in credit to a player on project j due to the competition from players in the set S; instead of the expected credit of $w_j p_i q_j$ that i would receive if she worked on j in isolation, she instead gets $w_j p_i q_j c_j(S)$ when the players in S are also working on j. Thus, with a_i denoting the project chosen by i, and $K_{a_i}(\vec{a})$ denoting the set of all players choosing project a_i, we have $u_i(\vec{a}) = w_{a_i} p_i q_{a_i} c_{a_i}(K_{a_i}(\vec{a}) - i)$.

Re-weighting Projects to Achieve Social Optimality. We now describe how to re-weight projects, creating new weights $\{w'_j\}$, so as to make a given social optimum $\vec{o}$ a Nash equilibrium. First, since the relative values of the project weights are all that matters, we can choose any project x arbitrarily and set its new weight w'_x equal to 1. We will set the weights w'_j of the other projects so that every player's favorite alternate project (and hence the target of any potential deviation) is x.

Now, among all the players working on another project $j \neq x$, which one has the greatest incentive to move to x? It is the player $i \in K_j(\vec{o})$ with the lowest ability p_i, since all players $i' \in K_j(\vec{o})$ experience the same competition function $c_x(K_x(\vec{o}))$, but i experiences the strongest competition from the other players in $K_j(\vec{o})$. This is because they all have ability at least as great as i, so i has the most to gain by moving off j.

Motivated by this, for a strategy vector $\vec{a}$ and a project j, we define $\delta_j(\vec{a})$ to be the player $i \in K_j(\vec{a})$ of minimum ability p_i. We define $w'_x = 1$ and for every other project $j \neq x$, we define

$$w'_j = \frac{q_x c_x(K_x(\vec{o}))}{q_j c_j(K_j(\vec{o}) - \delta_j(\vec{o}))}. \tag{1}$$

In the appendix, we take the informal argument above and make it precise, proving

Theorem 3.4 *The optimal assignment $\vec{o}$ is a Nash equilibrium in the game with the given weights $\{w'_j\}$.*

Re-weighting Players to Achieve Social Optimality. It is also possible to re-weight the players so as to make the social optimum a Nash equilibrium. Because the greedy algorithm no longer computes the social optimum, it is no longer enough to use weights to approximately simulate an arbitrary ordering on the players. However, we can use an extension of this plan that incorporates two additional ingredients: first, we base the greedy ordering on the socially optimal assignment, and second, we do not use a strict ordering but rather one that groups the players into *stages* of equal weight.

The algorithm for assigning weights is as follows. In the beginning, we fix an optimal assignment $\vec{o}$ and a sufficiently small value of $\epsilon > 0$ (to be determined below), and we declare all players to be *unassigned*. The algorithm then operates in a sequence of *stages* $c = 1, 2, \ldots$. At the start of stage c, some players have been given weights and been assigned to projects, resulting in a partial strategy vector $\vec{a^c}$ consisting only of players assigned before stage c. We show that at the start of stage c, each unassigned player would maximize her payoff by choosing a project from the set

$$X_c = \{j \mid w_j \prod_{h \in K_j(\vec{a^c})} (1 - p_h q_j) q_j = \max_l w_l \prod_{h \in K_l(\vec{a^c})} (1 - p_h q_l) q_l\}.$$

Thus in stage c, the algorithm does the following. It first computes this set of projects X_c. Then, for each project $j \in X_c$ for which there exists a player i such that $o_i = j$ and i is unassigned, it assigns i to project j, and sets $z_i = \epsilon^c$.

It would be natural to try proving that with these weights, the assignment $\vec{o}$ is a Nash equilibrium. However, this is not necessarily correct. In the final stage c^* of the algorithm, it may be that the number of unassigned players is less than $|X_{c^*}|$, and in this case some of the unassigned players might go to projects other than the ones corresponding to $\vec{o}$. However, in the appendix we prove that there always exists an optimal assignment $\vec{o'}$ derived from $\vec{o}$ that is a Nash equilibrium with these weights.

Theorem 3.5 *There is an optimal assignment $\vec{o'}$ that is a Nash equilibrium in the game with weights $\{z_i\}$.*

4 A Further Generalization: Arbitrary Success Probabilities

Finally, we consider a further generalization of the model, in which player i has an arbitrary success probability p_{ij} when working on project j. The strategies and payoffs remain the same as before, subject to this modification. Also, this generalization is a monotone valid-utility game and congestion game; however, we omit the proofs since they are very similar to the proofs for the case from the previous section.

An interesting feature of this generalization is that one can no longer always make the social optimum a Nash equilibrium by re-weighting projects. To see why, consider an example in which there are two players 1 and 2, and two projects a and b. We have $w_a = w_b = 1$ and success probabilities $p_{1a} = 1$, $p_{1b} = 0.5$, $p_{2a} = 0.5$, and $p_{2b} = 0.1$. Now, the social optimum is achieved if player 1 is assigned to a and player 2 is assigned to b. But this gives too little utility to player 2, and in order to keep player 2 on b, we need to re-weight the projects so that $w'_b \geq 2.5 w'_a$. In this case, however, player 1 also has an incentive to move to b, proving that no re-weighting can enforce the social optimum.

The case of re-weighting players is an open question. In Sections 2 and 3, we used the re-weighting of players in a limited way, to simulate an ordering. We can show via an example that there are instances in this more general model where no strict ordering on the players will produce the social optimum. To see why, consider an instance with three players and two projects a and b with weights $w_a = 1$ and $w_b = 0.56$.

Players 1 and 2 each have success probabilities 0.5 for a and 0.9 for b, and player 3 has a success probability of 0.6 for a and 1 for b. In this case, one can verify that the unique social optimum assigns players 1 and 2 to a and player 3 to b, but in any ordering, the player ordered first will choose the wrong project.

However, one can potentially make use of player weights in more complex ways, and so we have the following open questions.

Question 4.1 *In the model with arbitrary success probabilities:*

(a) For every instance, does there exist a re-weighting of the players so that there is a social optimum that is a Nash equilibrium? If not, can this be done by re-weighting both the players and *the projects?*

(b) Does there exist a constant $c < 2$ such that for all instances, it is possible to re-weight only the projects so that the price of anarchy is at most c?

As one interesting partial result on the re-weighting of players in this model, we can show the following.

Theorem 4.2 *If there exists a social optimum $\vec{o}$ that assigns each player to a distinct project, then it is possible to re-weight the players so that $\vec{o}$ is a Nash equilibrium.*

The proof, given in the appendix, uses an analysis of the alternating-cycle structure of a bipartite graph on players and projects, combined with ideas from the proof of Theorem 3.5.

Finally, as a further insight into the structure of this general case, we pursue an analogy with the bound of $2 - \frac{1}{n}$ on the price of anarchy in the case of identical players: we show that even in the case of arbitrary success probabilities, the price of anarchy is strictly better than the general bound for monotone valid-utility games implies.

Theorem 4.3 *In every instance of the Project Game with arbitrary success probabilities, the price of anarchy is strictly less than* 2.

Acknowledgments. We thank Harry Collins, Michael Macy, Trevor Pinch, and Michael Strevens for valuable insights and suggestions on the relevant connections to work in the philosophy and sociology of science, and Larry Blume, David Easley, and Bobby Kleinberg for valuable advice on the game-theoretic aspects of the problem.

References

[1] Scott Aaronson, Allan Borodin, Bernard Chazelle, Oded Goldreich, Shafi Goldwasser, Richard Karp, Michael Kearns, Christos Papadimitriou, Madhu Sudan, and Salil Vadhan. Statement on conceptual contributions in theory, 2008. On-line at http://scottaaronson.com/blog/?p=315.

[2] Kenneth J. Arrow. Economic welfare and the allocation of resources for invention. In National Bureau of Economic Research, editor, *The Rate and Direction of Inventive Activity: Economic and Social Factors*, pages 609–626. Princeton University Press, 1962.

[3] Jesus Zamora Bonilla. The economics of scientific knowledge. In Uskali Mäki, editor, *Handbook of the Philosophy of Science: The Philosophy of Economics*. Elsevier, to appear.

[4] Allan Borodin, Morten N. Nielsen, and Charles Rackoff. (incremental) priority algorithms. *Algorithmica*, 37(4):295–326, 2003.

[5] Jean-Pierre Bourdieu. The specificity of the scientific field and the social conditions of the progress of reason. *Social Science of Information*, 14(6):19–47, 1975.

[6] Moira Burke and Robert Kraut. Mopping up: Modeling wikipedia promotion decisions. In *Proc. CSCW'08: ACM Conference on Computer-Supported Cooperative Work*, pages 27–36, 2008.

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

[8] Michel X. Goemans, Li Li, Vahab S. Mirrokni, and Marina Thottan. Market sharing games applied to content distribution in ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(5):1020–1033, 2006.

[9] Oded Goldreich. Innovations in Computer Science (ICS), 2009. On-line at http://www.wisdom.weizmann.ac.il/∼oded/ics.html.

[10] Tim Gowers, Gil Kalai, Michael Nielsen, and Terence Tao. The polymath blog. http://polymathprojects.org/about/.

[11] Philip Kitcher. The division of cognitive labor. *Journal of Philosophy*, 86(1):5–21, January 1990.

[12] Philip Kitcher. *The Advancement of Science*. Oxford University Press, 1993.

[13] Jure Leskovec, Dan Huttenlocher, and Jon Kleinberg. Governance in social media: A case study of the Wikipedia promotion process. In *Proc. 4th International Conference on Weblogs and Social Media*, 2010.

[14] Robert K. Merton. The Matthew effect in science. *Science*, 159(3810):56–63, January 1968.

[15] Robert K. Merton. *The Sociology of Science: Theoretical and Empirical Investigations*. University of Chicago Press, 1973.

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

[17] Charles S. Peirce. Note on the theory of economy of research. In Arthur W. Burks, editor, *Collected Papers of Charles Sander Peirce*, volume 7, pages 76–83. Harvard University Press, 1958.

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

[19] Tim Roughgarden. Intrinsic robustness of the price of anarchy. In *Proc. 41st ACM Symposium on Theory of Computing*, pages 513–522, 2009.

[20] Klaus Stein and Claudia Hess. Does it matter who contributes: A study on featured articles in the German Wikipedia. In *Proc. 18th ACM Conference on Hypertext and Hypermedia*, pages 171–174, 2007.

[21] Michael Strevens. The role of the priority rule in science. *Journal of Philosophy*, 100(2):55–79, 2003.

[22] Michael Strevens. The role of the Matthew Effect in science. *Studies in History and Philosophy of Science*, 37:159–170, 2006.

[23] Éva Tardos and Tom Wexler. Network formation games and the potential function method. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay Vazirani, editors, *Algorithmic Game Theory*, pages 487–516. Cambridge University Press, 2007.

[24] Adrian Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proc. 43rd IEEE Symposium on Foundations of Computer Science*, pages 416–425, 2002.

5 Appendix: Proofs of Results from Section 2

Proof of Claim 2.1. We need a bit of additional notation: the quantity $u(a_i|a_{-i})$ will denote the marginal contribution of player i to the overall utility, given the choices made by all other players. In other words, it is $r_{a_i}(k_{a_i}(\vec{a}) - 1)$.

The definition of a monotone valid-utility game [23, 24] requires verifying four properties of the utility functions, as follows.

1. $u(\vec{a})$ *is submodular:* Since $u(\vec{a})$ is the summation of the projects' separate utilities, it is enough to prove that the utility of every project is submodular. For identical players this is settled by the simple observation that $r_j(k)$ is decreasing in k.

2. $u(\vec{a})$ *is monotone:* Naturally, a project's success probability can only increase when more players are working on it.

3. $u_i(\vec{a}) \geq u(a_i|a_{-i})$: For this, we notice that $\sigma_j(k_j(\vec{a}))$ can be written as the sum of the marginal utilities contributed by the players on project j when they arrive to it in any order, and a player's utility is the average of such contributions over all arrival orders. Since the utility is submodular, the smallest of these contributions occurs when i arrives last in the order. In this case it is equal to $u(a_i|a_{-i})$. Hence this quantity is at most $u_i(\vec{a})$ as required.

4. $u(\vec{a}) \geq \sum_i u_i(\vec{a})$*:* In this game, by the definition of a player's utility, they are equal.

■

Proof of Claim 2.2. Assume towards a contradiction that the assignment resulting from the greedy algorithm $\vec{a}$ is sub-optimal. Let $\vec{o}$ be an optimal assignment. Recall that the assignments are insensitive to the identity of the players. Hence, since the two assignments have different utilities it has to be the case that there exist two projects b, c such that:

- $k_b(\vec{a}) > k_b(\vec{o})$
- $k_c(\vec{a}) < k_c(\vec{o})$

Let i be the last player the algorithm assigned to project b. In the iteration when i was assigned to project b there were at most $k_c(\vec{a})$ players working on project c. As we noted at the beginning of Section 2, the function $r_j(k)$ is decreasing in k for all projects j, and hence $r_b(k_b(\vec{a})-1) \geq r_c(k_c(\vec{a}))$. By this decreasing property we also have that $r_b(k_b(\vec{o})) \geq r_b(k_b(\vec{a})-1)$ and that $r_c(k_c(\vec{a})) \geq r_c(k_c(\vec{o})-1)$. Because $\vec{o}$ is the optimal assignment then $r_b(k_b(\vec{o})) \leq r_c(k_c(\vec{o})-1)$. Hence we have that $r_b(k_b(\vec{o})) = r_c(k_c(\vec{o})-1)$.

Denote the assignment resulting from moving a player from project c to project b in $\vec{o}$ by $\vec{o_1}$. $\vec{o_1}$ is also an optimal assignment, and hence there are two more projects b' and c' as before. Now we can move a player from c' to b' to obtain an optimal assignment $\vec{o_2}$. We continue similarly. Since each time we use this argument the number of players working on every project j is monotone decreasing or increasing, after a finite number of steps we reach assignment $\vec{o_l}$ which is an optimal assignment identical to $\vec{a}$. This contradicts the initial assumption that $\vec{a}$ was not an optimal assignment. ■

Proof of Claim 2.3. Denote the assignment the algorithm computes by $\vec{a}$. Assume towards a contradiction that player i, who is currently assigned to project j, can increase his payoff by switching to project l: $u_i(j, a_{-i}) < u_i(l, a_{-i})$. Since all the players are identical we can assume without loss of generality that

i was the last player who chose project j. Denote by $\vec{a'}$ the assignment vector at the iteration at which it was player i's turn to choose a project. Since player i chose project j we have that $u_i(j, a'_{-i}) \geq u_i(l, a'_{-i})$. Noticing that $k_l(\vec{a}) \geq k_l(\vec{a'})$ and that the utility function is submodular, we obtain a contradiction. ■

Proof of Claim 2.4. As assignments are insensitive to the identity of the players there exists a player i such that $a_i = j$ and $b_i = l$. Since $\vec{a}$ is a Nash equilibrium we have $u_i(j, a_{-i}) \geq u_i(l, a_{-i})$. On the other hand $\vec{b}$ is also a Nash equilibrium, and hence $u_i(j, b_{-i}) \leq u_i(l, b_{-i})$. Recall that j and l are two projects such that $k_j(\vec{a}) > k_j(\vec{b})$ and $k_l(\vec{a}) < k_l(\vec{b})$. Therefore, because a player's utility decreases as the number of players working on the project increases, we have

$$u_i(j, a_{-i}) \leq u_i(j, b_{-i}) \leq u_i(l, b_{-i}) \leq u_i(l, a_{-i}) \leq u_i(j, a_{-i}).$$

Therefore, $u_i(j, a_{-i}) = u_i(j, b_{-i})$. This implies that $k_j(\vec{a}) = k_j(j, a_{-i}) = k_j(j, b_{-i}) = k_j(\vec{b}) + 1$. Similarly for $u_i(l, b_{-i}) = u_i(l, a_{-i})$ we have that $k_l(\vec{b}) = k_l(l, b_{-i}) = k_l(l, a_{-i}) = k_l(\vec{a}) + 1$. ■

Proof of Claim 2.6. Since $\vec{a}$ is a Nash equilibrium then $\sum_{i \in N} u_i(b_i, a_{-i}) \leq \sum_{i \in N} u_i(\vec{a})$. Recall also that $\sum_{i \in N} u_i(\vec{a}) = u(\vec{a})$. Hence we have that $u(\vec{a}) \geq \lambda u(\vec{b}) - \mu u(\vec{a})$. By rearranging the terms we get that: $u(\vec{b}) \leq \frac{1+\mu}{\lambda} u(\vec{a})$. ■

Proof of Theorem 2.7. In order to prove Theorem 2.7, we need several claims. First, by noticing that

$$\sum_{j \in M} \left(\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) \right) = \sum_{i \in N} u_i(b_i, a_{-i}) \geq \lambda u(\vec{b}) - \mu u(\vec{a}) = \sum_{j \in M} \left(\lambda \sigma_j(k_j(\vec{b})) - \mu \sigma_j(k_j(\vec{a})) \right),$$

we derive stricter condition for a game to be (λ, μ)-smooth; this condition will turn out to be easier to work with.

Claim 5.1 *Suppose that in a monotone valid-utility game, it is the case that for every strategy vectors $\vec{a}$ and $\vec{b}$, and project j, we have*

$$\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) \geq \lambda \sigma_j(k_j(\vec{b})) - \mu \sigma_j(k_j(\vec{a})).$$

Then the game is (λ, μ)-smooth.

The next two claims compare two strategy vectors $\vec{a}$ and $\vec{b}$ with respect to a given project j in two cases: when $k_j(\vec{a}) > k_j(\vec{b}) > 0$ (Claim 5.2) and when $k_j(\vec{a}) \leq k_j(\vec{b})$ (Claim 5.3).

Claim 5.2 *If $k_j(\vec{a}) > k_j(\vec{b}) > 0$ then $\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) \geq \sigma_j(k_j(\vec{b})) - \frac{k_j(\vec{a}) - k_j(\vec{b})}{k_j(\vec{a}) - k_j(\vec{b}) + 1} \sigma_j(k_j(\vec{a}))$*

Proof: Since, all players who are working on project j in $\vec{b}$ are already working on it in $\vec{a}$, we have $\forall i \in K_j(\vec{b}) : \; u_i(b_i, a_{-i}) = u_i(\vec{a}) = \dfrac{\sigma_j(k_j(\vec{a}))}{k_j(\vec{a})}$. Thus, $\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) = k_j(\vec{b}) \dfrac{\sigma_j(k_j(\vec{a}))}{k_j(\vec{a})}$. Hence, we need to show that

$$\frac{k_j(\vec{b})}{k_j(\vec{a})} \sigma_j(k_j(\vec{a})) + \frac{k_j(\vec{a}) - K_j(\vec{b})}{k_j(\vec{a}) - k_j(\vec{b}) + 1} \sigma_j(k_j(\vec{a})) \geq \sigma_j(k_j(\vec{b})).$$

Since $\sigma_j(k)$ is monotone, we have that $\sigma_j(k_j(\vec{a})) \geq \sigma_j(k_j(\vec{b}))$. Therefore, it is enough to show that

$$\frac{k_j(\vec{b})}{k_j(\vec{a})} + \frac{k_j(\vec{a}) - k_j(\vec{b})}{k_j(\vec{a}) - k_j(\vec{b}) + 1} \geq 1$$

Because $k_j(\vec{b}) > 0$, we have $\frac{k_j(\vec{a}) - k_j(\vec{b})}{k_j(\vec{a}) - k_j(\vec{b}) + 1} \geq \frac{k_j(\vec{a}) - k_j(\vec{b})}{k_j(\vec{a})}$ and the claim follows. □

Claim 5.3 *If $k_j(\vec{a}) \leq k_j(\vec{b})$ then $\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) \geq \sigma_j(k_j(\vec{b}))$*

Proof: If $k_j(\vec{a}) = k_j(\vec{b})$ then the claim is trivial. Otherwise, $k_j(\vec{a}) < k_j(\vec{b})$. The utility of the $k_j(\vec{b}) - k_j(\vec{a})$ players who were not working on project j in $\vec{a}$ is $\frac{\sigma_j(k_j(\vec{a})+1)}{k_j(\vec{a})+1}$. The utility of the $k_j(\vec{a})$ players that were working on project j in $\vec{a}$ is $\frac{\sigma_j(k_j(\vec{a}))}{k_j(\vec{a})}$. The latter utility decreases as more players are working in the project. Therefore, $\forall i \in K_j(\vec{b}) : \ u_i(b_i, a_{-i}) \geq \frac{\sigma_j(k_j(\vec{a})+1)}{k_j(\vec{a})+1}$. Similarly, since $k_j(\vec{a}) + 1 \leq k_j(\vec{b})$ then $\frac{\sigma_j(k_j(\vec{a})+1)}{k_j(\vec{a})+1} \geq \frac{\sigma_j(k_j(\vec{b}))}{k_j(\vec{b})}$. Thus, $\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) \geq k_j(\vec{b}) \frac{\sigma_j(k_j(\vec{a})+1)}{k_j(\vec{a})+1} \geq k_j(\vec{b}) \frac{\sigma_j(k_j(\vec{b}))}{k_j(\vec{b})} = \sigma_j(k_j(\vec{b}))$. □

Finally, we can complete the proof of Theorem 2.7. Claims 5.2 and 5.3 establish the following:

- If $k_j(\vec{a}) > k_j(\vec{b}) > 0$ then $\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) \geq \sigma_j(k_j(\vec{b})) - \frac{k_j(\vec{a}) - k_j(\vec{b})}{k_j(\vec{a}) - k_j(\vec{b}) + 1} \sigma_j(k_j(\vec{a}))$
- If $k_j(\vec{a}) \leq k_j(\vec{b})$ then $\sum_{i \in K_j(\vec{b})} u_i(b_i, a_{-i}) \geq \sigma_j(k_j(\vec{b}))$

Hence by taking μ as the maximum of the appropriate fractions from Claim 5.2, the theorem follows. ■

Proof of Theorem 2.8. We show more generally that if the utility of all players is identical when playing a strategy vector $\vec{a}$ then $\vec{a}$ is a Nash equilibrium. Furthermore we also show that all Nash equilibria assign to every project j exactly $k_j(\vec{a})$ players.

Indeed, denote the utility of each player in $\vec{a}$ by x: that is, for every player i, we have $u_i(\vec{a}) = x$. We also have that for every project $j \neq a_i$ $u_i(j, a_{-i}) < x$. This holds for each j because either $k_j(\vec{o}) = 0$, in which case $w'_j = 0$, or else $k_j(\vec{o}) > 0$, in which case there are already players assigned to j, and for such projects j a player's utility function is strictly decreasing in the number of players working on j. Therefore, $\vec{a}$ is a Nash equilibrium. As a corollary of Claim 2.4 we have that if there exist Nash equilibria $\vec{a}$ and $\vec{b}$ assigning different numbers of players to some project, then there exists a player i such that $a_i \neq b_i$ and $u_i(a_i, a_{-i}) = u_i(b_i, a_{-i})$. But this is impossible since we have that $u_i(a_i, a_{-i}) = x$ and $u_i(b_i, a_{-i}) < x$. ■

Proof of Theorem 2.9. We follow the outline of the proof discussed in Section 2. We first write down a closed-form expression for player i's utility after re-weighting by player weights $\{z_i\}$. The utility is

$$\widetilde{u_i}(\vec{a}) = w_{a_i} q_{a_i} \sum_{S \subseteq \{K_{a_i}(\vec{a}) - i\}} \frac{z_i}{(\sum_{h \in S} z_h) + z_i} q_{a_i}^{|S|} (1 - q_{a_i})^{k_{a_i}(\vec{a}) - |S| - 1}$$

Notice that the re-weighting affects only the players utilities, not the social welfare.

Now, as discussed in Section 2, we define weights for the players so as to simulate an approximate ordering on them. For some arbitrary ordering of the players we assign for every player i the weight $z_i = \epsilon^i$. We first show that if the utilities $\widetilde{u}$ can be closely bounded in terms of the utility when the players are actually ordered, then this re-weighting scheme achieves the goal of making all Nash equilibria optimal. Following this, we prove some technical claims to show that we can indeed bound the utilities as desired. First we need some definitions:

Definition 5.4

- $prev_i(S) = \{h \in S | z_i < z_h\}$ *where* $S \subseteq N$. *This is the set of players before player* i.
- $succ_i(S) = \{h \in S | z_i > z_h\}$ *where* $S \subseteq N$. *This is theset of players after player* i.
- $\widehat{u}_i(\vec{a}) = r_{a_i}(|prev_i(K_{a_i}(\vec{a}))|)$. *This is the marginal contribution of player* i *to the social welfare.*

Claim 2.10. If for every player i and project j we have $\widehat{u}_i(j, a_{-i}) - \frac{1}{4d^{n+1}} \leq \widetilde{u}_i(j, a_{-i}) \leq \widehat{u}_i(j, a_{-i}) + \frac{1}{4d^{n+1}}$, then any Nash equilibrium in the game with the weights $\{z_i\}$ is also an optimal assignment.

Proof: The proof resembles the proof that the greedy algorithm achieves the socially optimal assignment. Let $\vec{a}$ be a Nash equilibrium. Among all possible optimal assignments, we say that an optimal assignment $\vec{o}$ is *most similar* to $\vec{a}$ if for every two projects j and l such that $k_j(\vec{o}) > k_j(\vec{a})$ and $k_l(\vec{o}) < k_l(\vec{a})$ it is the case that $r_j(k_j(\vec{o}) - 1) > r_l(k_l(\vec{a}))$. This is a most similar assignment to $\vec{a}$ since we cannot create a more similar assignment with the same utility by moving a player from project j to project l.

Let $\vec{o}$ be most similar to $\vec{a}$, and assume towards a contradiction that $\vec{a}$ is not an optimal assignment. Hence there exist two projects j and l such that:

- $k_j(\vec{o}) > k_j(\vec{a})$
- $k_l(\vec{o}) < k_l(\vec{a})$

For those two projects, since $\vec{o}$ is an optimal assignment, $r_j(k_j(\vec{o}) - 1) > r_l(k_l(\vec{o}))$. By the previous statement and the decreasing property of r_j we conclude that also $r_j(k_j(\vec{a})) > r_l(k_l(\vec{a}) - 1)$. Let player i be the player with the highest index working on project l in $\vec{a}$. Since player i is the player with the minimal weight working on project l, we have $\widehat{u}_i(l, a_{-i}) = r_l(k_l(\vec{a}) - 1)$. Since $\vec{a}$ is a Nash equilibrium we have that $\widetilde{u}_i(l, a_{-i}) \geq \widetilde{u}_i(j, a_{-i})$. By the assumption stated in the claim, we have

$$r_l(k_l(\vec{a}) - 1) + \frac{1}{4d^{n+1}} \geq \widetilde{u}_i(l, a_{-i}) \geq \widetilde{u}_i(j, a_{-i}) \geq \widehat{u}_i(j, a_{-i}) - \frac{1}{4d^{n+1}} \geq r_j(k_j(\vec{a})) - \frac{1}{4d^{n+1}}$$

Hence, $r_l(k_l(\vec{a}) - 1) + \frac{1}{4d^{n+1}} \geq r_j(k_j(\vec{a})) - \frac{1}{4d^{n+1}} \implies r_j(k_j(\vec{a})) - r_l(k_l(\vec{a}) - 1) \leq \frac{1}{2d^{n+1}}$. But this is a contradiction: each of $r_j(k_j(\vec{a}))$ and $r_l(k_l(\vec{a}) - 1)$ are products of at most $n + 1$ terms of common denominator d, and they are not equal, so they must differ by at least $\frac{1}{d^{n+1}}$. $\square$

Next we are going to prove that there exists an ϵ for which the bounds assumed in the previous claim holds.

Claim 2.11. There exists and ϵ such that for every player i and project j: $\widehat{u}_i(j,a_{-i}) - \frac{1}{4d^{n+1}} \leq \widetilde{u}_i(j,a_{-i}) \leq \widehat{u}_i(j,a_{-i}) + \frac{1}{4d^{n+1}}$

Proof: The following definitions will be helpful in simplifying the utility function:

Definition 5.5 $X_i(j;S;\vec{a}) = \frac{z_i}{z_i+\sum_{h\in S} z_h} q_j^{|S|+1} \cdot (1-q_j)^{k_j(a_{-i})-|S|}$ *is the probability that only player i and the players in S succeed at the project, and player i is the one chosen to receive the credit.*

Using the definition we now have $\widetilde{u}_i(\vec{a}) = w_{a_i} \sum_{S\subseteq K_{a_i}(a_{-i})} X_i(j;S;\vec{a})$.

By using $prev_i(S)$ and $succ_i(S)$ we can break up the player's utility in the following manner:

$$\widetilde{u}_i(j,a_{-i}) = w_j \left(\sum_{S\subseteq succ_i(K_j(\vec{a}))} X_i(j;S;\vec{a}) + \sum_{\substack{S\subseteq K_j(a_{-i}) \\ S\cap prev_i(K_j(\vec{a}))\neq\emptyset}} X_i(j;S;\vec{a}) \right)$$

This is a convenient representation of a player's utility since it partitions the successful player sets into two types:

- $S \subseteq succ_i(K_j(\vec{a}))$: for such a set S, player i's weight is *dominant* in S, and hence she gets most of the utility.
- $S \subseteq K_j(a_{-i})$ and $S \cap prev_i(K_j(\vec{a})) \neq \emptyset$: for such a set S, player i's weight is *dominated*, and hence she gets only a very small fraction of the utility.

Lemma 5.6 *For every player i and project j:*

1. $\widetilde{u}_i(j,a_{-i}) \geq \frac{1}{1+2\epsilon}\widehat{u}_i(j,a_{-i})$
2. $\widetilde{u}_i(j,a_{-i}) \leq \widehat{u}_i(j,a_{-i}) + w_j \frac{2^{k_j(a_{-i})}\epsilon}{1+\epsilon}$

Proof:

1. We show that $w_j \sum_{S\subseteq succ_i(K_j(\vec{a}))} X_i(j;S;\vec{a}) \geq \frac{1}{1+2\epsilon}\widehat{u}_i(j,a_{-i})$. From this we can conclude that $\widetilde{u}_i(j,a_{-i}) \geq \frac{1}{1+2\epsilon}\widehat{u}_i(j,a_{-i})$. We first write an alternative expression for $\widehat{u}_i(j,a_{-i})$:

$$\begin{aligned}\widehat{u}_i(j,a_{-i}) &= \underbrace{w_j q_j (1-q_j)^{|prev_i(K_j(\vec{a}))|}}_{=r_j(|prev_i(K_j(\vec{a}))|)} \underbrace{\sum_{S\subseteq succ_i(K_j(\vec{a}))} q_j^{|S|} \cdot (1-q_j)^{|succ_i(K_j(\vec{a}))|-|S|}}_{=1} \\ &= w_j \sum_{S\subseteq succ_i(K_j(\vec{a}))} q_j^{|S|+1} \cdot (1-q_j)^{k_j(a_{-i})-|S|}\end{aligned}$$

The resulting expression is an upper bound on $w_j \sum_{S\subseteq succ_i(K_j(\vec{a}))} X_i(j;S;\vec{a})$ since it assumes that $z_i > 0$ and the rest of the weights are 0. By the definition of $succ_i$, we have $z_h \leq \epsilon z_i$ for all $h \in$

$succ_i(K_j(\vec{a}))$. By simple rearrangement of terms, we can bound the weight coefficients $\dfrac{z_i}{z_i + \sum_{h \in S} z_h}$ in $\widetilde{u}_i(j, a_{-i})$:

$$\forall S \subseteq succ_i(K_j(\vec{a})) : \ \frac{z_i}{z_i + \sum_{h \in S} z_h} \geq \frac{z_i}{z_i + \sum_{h \in succ_i(K_j(\vec{a}))} z_h} \geq \frac{z_i}{z_i + \sum_{h=1}^{|succ_i(K_j(\vec{a}))|} \epsilon^h z_i}$$

$$= \frac{1}{1 + \sum_{h=1}^{|succ_i(K_j(\vec{a}))|} \epsilon^h} > \frac{1}{1 + 2\epsilon}$$

The last inequality holds for $\epsilon < 0.5$. Hence we have that $\sum_{S \subseteq succ_i(K_j(\vec{a}))} X_i(j; S; \vec{a}) \geq \dfrac{1}{1+2\epsilon} \widehat{u}_i(j, a_{-i})$.

2. By the previous analysis we have

$$w_j \sum_{S \subseteq succ_i(K_j(\vec{a}))} X_i(j; S; \vec{a}) \leq \widehat{u}_i(j, a_{-i})$$

The claim follows by showing that:

$$w_j \left(\sum_{\substack{S \subseteq K_j(a_{-i}) \\ S \cap prev_i(K_j(\vec{a})) \neq \emptyset}} X_i(j; S; \vec{a}) \right) \leq w_j \frac{2^{k_j(a_{-i})} \epsilon}{1 + \epsilon}$$

Since $S \cap prev_i(K_j(\vec{a})) \neq \emptyset$ for all S, in each set S there exists at least one player $h \in S$ such that $z_h > z_i$:

$$\forall S \ \frac{z_i}{z_i + \sum_{h \in S} z_h} \leq \frac{z_i}{z_i + \min_{\{h \in S | z_h > z_i\}} z_h} \leq \frac{z_i}{z_i + \frac{z_i}{\epsilon}} = \frac{\epsilon}{1 + \epsilon}$$

Since $q_j^{|S|+1} \cdot (1 - q_j)^{k_j(a_{-i}) - |S|} < 1$ for every one of the $2^{k_j(a_{-i})}$ subsets S the claim holds.

□

Lemma 5.7 *If* $\epsilon \leq min_{l \in M} \left(\dfrac{1}{4d^{n+1} w_l 2^{k_l(\vec{a})}} \right)$ *then* $\widehat{u}_i(j, a_{-i}) - \dfrac{1}{4d^{n+1}} \leq \widetilde{u}_i(j, a_{-i}) \leq \widehat{u}_i(j, a_{-i}) + \dfrac{1}{4d^{n+1}}$

Proof: By part 1 of Lemma 5.6 we have that for every project j:

$$\widetilde{u}_i(j, a_{-i}) \geq \frac{1}{1 + \frac{2}{4d^{n+1} w_l 2^{k_l(\vec{a})}}} \widehat{u}_i(j, a_{-i}) = \frac{4d^{n+1} w_l 2^{k_l(\vec{a})}}{4d^{n+1} w_l 2^{k_l(\vec{a})} + 2} \widehat{u}_i(j, a_{-i}) =$$

$$= \widehat{u}_i(j, a_{-i}) - \frac{2}{4d^{n+1} w_l 2^{k_l(\vec{a})} + 2} \geq \widehat{u}_i(j, a_{-i}) - \frac{1}{4d^{n+1}}$$

By part 2 of Lemma 5.6 we have that for every project j:

$$\widetilde{u}_i(j, a_{-i}) \leq \widehat{u}_i(j, a_{-i}) + \frac{\frac{w_j 2^{k_j(\vec{a})}}{4d^{n+1} w_l 2^{k_l(\vec{a})}}}{1 + \frac{1}{4d^{n+1} w_l 2^{k_l(\vec{a})}}}$$

Since by the definition of ϵ we have $w_j 2^{k_j(\vec{a})} \leq w_l 2^{k_l(\vec{a})}$, it follows that $\frac{w_j 2^{k_j(\vec{a})}}{4d^{n+1} w_l 2^{k_l(\vec{a})}} \leq \frac{1}{4d^{n+1}}$. Hence

$$\begin{aligned}
\widetilde{u_i}(j, a_{-i}) &\leq \widehat{u}_i(j, a_{-i}) + \frac{\frac{1}{4d^{n+1}}}{1 + \frac{1}{4d^{n+1} w_l 2^{k_l(\vec{a})}}} \\
&= \widehat{u}_i(j, a_{-i}) + \frac{1}{4d^{n+1}} \frac{4d^{n+1} w_l 2^{k_l(\vec{a})}}{4d^{n+1} w_l 2^{k_l(\vec{a})} + 1} \\
&= \widehat{u}_i(j, a_{-i}) + \frac{w_l 2^{k_l(\vec{a})}}{4d^{n+1} w_l 2^{k_l(\vec{a})} + 1} \\
&\leq \widehat{u}_i(j, a_{-i}) + \frac{1}{4d^{n+1}}.
\end{aligned}$$

□

This finishes establishing the necessary properties of ϵ, and hence establishes Claim 2.11 and therefore Theorem 2.9. □ ■

6 Appendix: Proofs of Results from Section 3

Proof of Claim 3.1. As noted in Section 3, the only significant change to the proof, compared with the proof of Claim 2.1, is in showing that the utility function is submodular. We prove this here by showing that $u(\vec{a})$ has decreasing marginal utility. Recall that $u(\vec{a})$ is the summation of the projects' separate utilities. Hence it is enough to prove that the utility of every project is submodular. More formally, We need to show that for every two sets of players $S \subseteq S'$ and for every project j and player i, we have $s_j(i|S) \geq s_j(i|S')$. To prove this, we observe that $w_j p_i q_j \prod_{l \in S}(1 - p_l q_j) \geq w_j p_i q_j \prod_{l \in T'}(1 - p_l q_j)$ since $1 \geq \prod_{l \in \{S' - S \cap S'\}}(1 - p_l q_j)$. ■

Proof of Claim 3.2. As discussed in Section 3, the utility of a player i depends not only on the number of other players working on i's project, but also on their identities. As a result, to establish that the game is a congestion game, we use a different characterization of congestion given by Monderer and Shapley in Corollary 2.9 of their paper [16]. Using the notation and terminology we have defined for the Project Game, the corollary can be written as follows.

Theorem 6.1 (Adapted from Monderer-Shapley) *The Project Game is an (exact) potential game if for every two players i, j, projects $x_i \neq y_i, x_j \neq y_j$ and strategy vector $a_{-i,j}$:*

$$u_i(y_i, x_j, a_{-i,j}) - u_i(x_i, x_j, a_{-i,j}) + u_j(y_i, y_j, a_{-i,j}) - u_j(y_i, x_j, a_{-i,j}) +$$
$$u_i(x_i, y_j, a_{-i,j}) - u_i(y_i, y_j, a_{-i,j}) + u_j(x_i, x_j, a_{-i,j}) - u_j(x_i, y_j, a_{-i,j}) = 0$$

We now use this to prove that the Project Game with Different Abilities is an exact potential game, from which the claim follows, since by another result of Monderer and Shapley, every finite exact potential game is isomorphic to a congestion game.

Recall that the utility of a player i is affected only by the players who are working on the same project as i. Hence, we should differentiate in the condition given in Theorem 6.1 between the cases in which players i and j are working on the same project and those in which they are not. By symmetry we can assume without loss of generality that $x_i \neq y_j$ and that $y_i \neq x_j$. Therefore we are left with the following cases:

1. $x_i \neq x_j$ and $y_i \neq y_j$. By rearranging the terms we get:

$$\underbrace{u_i(y_i, x_j, a_{-i,j}) - u_i(y_i, y_j, a_{-i,j})}_{=0} + \underbrace{u_i(x_i, y_j, a_{-i,j}) - u_i(x_i, x_j, a_{-i,j})}_{=0} +$$
$$\underbrace{u_j(y_i, y_j, a_{-i,j}) - u_j(x_i, y_j, a_{-i,j})}_{=0} + \underbrace{u_j(x_i, x_j, a_{-i,j}) - u_j(y_i, x_j, a_{-i,j})}_{=0} = 0$$

For example, $u_i(y_i, x_j, a_{-i,j}) - u_i(y_i, y_j, a_{-i,j}) = 0$ since $K_{y_i}(y_i, x_j, a_{-i,j}) = K_{y_i}(y_i, y_j, a_{-i,j})$ which is what the utility of player i depends on.

2. $x_i = x_j$ and $y_i \neq y_j$. By using Lemma 6.2 below and the previous argument we have that:

$$\underbrace{u_i(x_i, y_j, a_{-i,j}) - u_i(x_i, x_j, a_{-i,j}) + u_j(x_i, x_j, a_{-i,j}) - u_j(y_i, x_j, a_{-i,j})}_{=0} +$$
$$\underbrace{u_i(y_i, x_j, a_{-i,j}) - u_i(y_i, y_j, a_{-i,j})}_{=0} + \underbrace{u_j(y_i, y_j, a_{-i,j}) - u_j(x_i, y_j, a_{-i,j})}_{=0} = 0$$

3. $x_i \neq x_j$ and $y_i = y_j$. This case is symmetric to case 2.

4. $x_i = x_j$ and $y_i = y_j$. This case can be proved by using Lemma 6.2 twice, similar to case 2.

Finally, we conclude with the lemma needed in the analysis of the cases above.

Lemma 6.2 *For any three projects b, c, d such that $b \neq c$ and $b \neq d$:*

$$u_i(b, b, a_{-i,j}) - u_i(b, c, a_{-i,j}) = u_j(b, b, a_{-i,j}) - u_j(d, b, a_{-i,j})$$

Proof: By splitting the utility function to two parts (i.e., depending on whether player j succeeds in the project or fails in the project) we have that:

$$u_i(b, b, a_{-i,j}) = w_b p_i q_b \sum_{S \subseteq K_b(a_{-i,j})} \left(\frac{1}{|S|+1} \prod_{l \in S} p_l q_b \prod_{l \in \{K_b(a_{-i,j})) - S\}} (1 - p_l q_b) \right) (1 - p_j q_b) +$$
$$w_b p_i q_b \sum_{S \subseteq K_b(a_{-i,j})} \left(\frac{1}{|S|+2} \prod_{l \in S} p_l q_b \prod_{l \in \{K_b(a_{-i,j})) - S\}} (1 - p_l q_b) \right) p_j q_b$$
$$= \underbrace{w_b p_i q_b \sum_{S \subseteq K_b(a_{-i,j})} \left(\frac{1}{|S|+1} \prod_{l \in S} p_l q_b \prod_{l \in \{K_b(a_{-i,j})) - S\}} (1 - p_l q_b) \right)}_{=u_i(b,c,a_{-i,j})} -$$
$$w_b p_i q_b \sum_{S \subseteq K_b(a_{-i,j})} \left(\frac{1}{(|S|+1)(|S|+2)} \prod_{l \in S} p_l q_b \prod_{l \in \{K_b(a_{-i,j})) - S\}} (1 - p_l q_b) \right) p_j q_b$$

Similarly we have:

$$u_j(b, b, a_{-i,j}) = u_j(d, b, a_{-i,j}) - w_b p_j q_b \sum_{S \subseteq K_b(a_{-i,j})} \left(\frac{1}{(|S|+1)(|S|+2)} \prod_{l \in S} p_l q_b \prod_{l \in \{K_b(a_{-i,j})) - S\}} (1 - p_l q_b) \right) p_i q_b$$

Hence, $u_i(b,b,a_{-i,j}) - u_i(b,c,a_{-i,j}) = u_j(b,b,a_{-i,j}) - u_j(d,b,a_{-i,j})$.

□

This completes the proof of Claim 3.2. ■

Proof of Claim 3.3. We use a reduction from the *Subset Product* problem, whose NP-completeness is established in Garey and Johnson [7]. The Subset Product problem is defined as follows: given a set of n natural numbers $X = \{x_1, ..., x_n\}$ and a target number Q^*, does there exist $S \subseteq X$ such that $\prod_{x_i \in S} x_i = Q^*$?

As a first step, we show that the closely related *Multiplicative Number Partition* problem (MNP) is NP-complete. In MNP, we are again given a set of n natural numbers $X = \{x_1, ..., x_n\}$, but now we are asked whether there is a partition (S,T) of X such that $\prod_{x_i \in S} x_i = \prod_{x_j \in T} x_j$. We can show that MNP is NP-complete by a reduction from Subset Product, by analogy with the corresponding reduction from Subset Sum to (Additive) Number Partition. That is, given an instance of Subset Product with a set X and a target Q^*, we define $P = \prod_{x_i \in X} x_i$. Notice that if P is not divided by Q^* without a remainder then there is no subset as needed. Hence, we can assume without loss of generality that Q^* divides P. We then show that we can solve MNP for $X' = X \cup \{x_{n+1} = P^2/Q^*, x_{n+2} = P \cdot Q^*\}$ if and only if we can solve Subset Product. We prove this using the following lemma:

Lemma 6.3 *For a partition (S', T') of X' :* $\prod_{x_i \in S'} x_i = \prod_{x_j \in T'} x_j \iff \prod_{x_i \in \{S' - x_{n+1}\}} x_i = Q^*$

Proof: Notice that x_{n+1} and x_{n+2} should be in different sets because $x_{n+1} \cdot x_{n+2} > P$. We assume without loss of generality that $x_{n+1} \in S'$. Define $Y = \prod_{x_i \in \{S' - x_{n+1}\}} x_i$. By the definition of Y we have that $\prod_{x_j \in \{T' - x_{n+2}\}} x_j = \frac{P}{Y}$. By substituting in the equality $\prod_{x_i \in S'} x_i = \prod_{x_j \in T'} x_j$ we get that:

$$\frac{P^2}{Q^*} \cdot Y = P \cdot Q^* \cdot \frac{P}{Y} \iff \frac{Y}{Q^*} = \frac{Q^*}{Y}$$

Since both Y and Q^* are positive we get that $Y = Q^*$. □

By the previous lemma we conclude that for $S = \{S' - \{x_{n+1}\}\}$, which by the construction is a subset of X, we have that $\prod_{x_i \in S} = Q^*$. It follows that MNP is NP-complete.

We also need the following simple lemma about products over partitions of a set of natural numbers $X = \{x_1, ..., x_n\}$. In this lemma, we use ΠS as a shorthand for $\prod_{i \in S} x_i$.

Lemma 6.4 *A partition (S,T) minimizes $\Pi S + \Pi T$ if and only if it minimizes $|\Pi S - \Pi T|$*

Proof: Let (S,T) and (S',T') be two partitions of X. The following three inequalities are equivalent, since all terms are products of natural (and hence non-negative) numbers:

$$\Pi S + \Pi T < \Pi S' + \Pi T'$$

$$(\Pi S + \Pi T)^2 < (\Pi S' + \Pi T')^2$$

$$(\Pi S)^2 + 2\Pi S \cdot \Pi T + (\Pi T)^2 < (\Pi S')^2 + 2\Pi S' \cdot \Pi T' + (\Pi T')^2$$

Since both (S,T) and (S',T') are partitions of X, we have that: $\Pi S \cdot \Pi T = \Pi S' \cdot \Pi T'$, and hence we can subtract four times this common product from both sides of the previous inequality to get three more equivalent inequalities:

$$(\Pi S)^2 - 2\Pi S \cdot \Pi T + (\Pi T)^2 < (\Pi S')^2 - 2\Pi S' \cdot \Pi T' + (\Pi T')^2$$

$$(\Pi S - \Pi T)^2 < (\Pi S' - \Pi T')^2$$
$$|\Pi S - \Pi T| < |\Pi S' - \Pi T'|.$$

□

Finally, we prove that the socially optimal assignment in the Project Game with Different Abilities is NP-hard. We do this by a reduction from MNP to the special case of the optimal assignment in which we have n players and 2 identical projects. In this special case we assume both projects have a weight of 1 and success probability 1, and player i has a failure probability $\bar{p}_i$.

Given an instance of MNP, we create an instance of this special case of the optimal assignment problem by defining for every number x_i a player i with failure probability $\bar{p}_i = \frac{1}{x_i}$. The optimal solution to the assignment of players to projects is a partition (S, T) that maximizes the social welfare: $(1 - \prod_{i \in S} \bar{p}_i) + (1 - \prod_{j \in T} \bar{p}_j)$. This implies that the optimal partition actually minimizes $\prod_{i \in S} \bar{p}_i + \prod_{j \in T} \bar{p}_j$. By Lemma 6.4 we have that the assignment (S, T) that minimizes $\prod_{i \in S} \bar{p}_i + \prod_{j \in T} \bar{p}_j$ also minimizes $|\prod_{i \in S} \bar{p}_i - \prod_{j \in T} \bar{p}_j|$. Given a partition (S, T) that is an optimal solution to the identical projects variant it is possible to compute in poly time the value of $|\prod_{i \in S} \bar{p}_i - \prod_{j \in T} \bar{p}_j|$. If it is 0 we can tell that the answer to the MNP is yes. Otherwise the answer is no. The answer to MNP is yes if and only if $|(\prod S) - (\prod T)| = 0$ for the optimal partition (S, T) because

$$\prod_{i \in S} \frac{1}{x_i} = \prod_{j \in T} \frac{1}{x_j} \iff \prod_{i \in S} x_i = \prod_{j \in T} x_j$$

■

Proof of Theorem 3.4. To prove this, we will show that if a player did want to move to another project, he would choose to move to project x. After establishing this, it is enough to show that all the players working on project x in the optimal assignment don't want to move to another project, and that the rest of the players don't want to move to project x.

Before proceeding with these arguments, however, we state and prove a technical lemma giving an inductive form for the competition function that will be useful in the subsequent arguments.

Lemma 6.5 *For any project j, set of players S, and player $h \notin S$, we have*

$$c_j(S+h) = c_j(S) - p_h q_j \sum_{S' \subseteq S} \left(\frac{1}{(|S'|+1)(|S'|+2)} \prod_{i \in S'} p_i q_j \prod_{i \in \{S - S'\}} (1 - p_i q_j) \right)$$

Proof:

$$c_j(S+h) = (1 - p_h q_j) \sum_{S' \subseteq S} \left(\frac{1}{(|S'|+1)} \prod_{i \in S'} p_i q_j \prod_{i \in \{S - S'\}} (1 - p_i q_j) \right) +$$
$$p_h q_j \sum_{S' \subseteq S} \left(\frac{1}{(|S'|+2)} \prod_{i \in S'} p_i q_j \prod_{i \in \{S - S'\}} (1 - p_i q_j) \right) =$$
$$c_j(S) - p_h q_j \sum_{S' \subseteq S} \left(\frac{1}{(|S'|+1)(|S'|+2)} \prod_{i \in S'} p_i q_j \prod_{i \in \{S - S'\}} (1 - p_i q_j) \right)$$

□

We now show that a player i working on a project other than x views x as his best alternate project.

Lemma 6.6 *For any player i such that $o_i \neq x$, and for every project, $j \neq o_i$, we have $u_i(x, o_{-i}) \geq u_i(j, o_{-i})$*

Proof: We need to show that $w'_x p_i q_x c_x(K_x(\vec{o})) \geq w'_j p_i q_j c_j(K_j(\vec{o}))$. By setting the weights to their values according to Formula (1), we get that:

$$p_i q_x c_x(K_x(\vec{o})) \geq \frac{q_x c_x(K_x(\vec{o}))}{q_j c_j(K_j(\vec{o}) - \delta_j(\vec{o}))} p_i q_j c_j(K_j(\vec{o}))$$

By rearranging the terms we have that:

$$c_j(K_j(\vec{o}) - \delta_j(\vec{o})) \geq c_j(K_j(\vec{o})).$$

Intuitively, this inequality follows from the fact that as more players work on a project, it is less likely that a specific player will be the one to succeed at it. Formally, it follows from Lemma 6.5 above. □

Finally, we show that players on project x don't want to leave x, and players not on x don't want to move to x (and hence, by Lemma 6.6, don't want to move anywhere else either).

Lemma 6.7

1. *All players who are working in the optimal assignment on project x don't want to move to a different project.*
2. *All players who are working in the optimal assignment on project different than x don't want to move to project x.*

Proof: Assume towards a contradiction that there exists a player i who prefers to work on project $j \neq o_i$. This means that $w'_{o_i} p_i q_{o_i} c_{o_i}(K_{o_i}(\vec{o}) - i) < w'_j p_i q_j c_j(K_j(\vec{o}))$. For each case we set w'_{o_i} and w'_j to their values according to Formula (1) and get to a contradiction by rearranging the terms.

1. We set $w'_{o_i} = 1$ and $w'_j = \dfrac{q_x c_x(K_x(\vec{o}))}{q_j c_j(K_j(\vec{o}) - \delta_j(\vec{o}))}$ and get the following inequality:

$$p_i q_x c_x(K_x(\vec{o}) - i) < \frac{q_x c_x(K_x(\vec{o}))}{q_j c_j(K_j(\vec{o}) - \delta_j(\vec{o}))} p_i q_j c_j(K_j(\vec{o}))$$

   After rearranging the inequality we get that:

$$\frac{c_x(K_x(\vec{o}) - i)}{c_x(K_x(\vec{o}))} < \frac{c_j(K_j(\vec{o}))}{c_j(K_j(\vec{o}) - \delta_j(\vec{o}))}$$

   The contradiction follows by noticing that $c_x(K_x(\vec{o}) - i) > c_x(K_x(\vec{o}))$ by Lemma 6.5; however, $c_j(K_j(\vec{o})) < c_j(K_j(\vec{o}) - \delta_j(\vec{o}))$

2. We set $w'_{o_i} = \frac{q_x c_x(K_x(\vec{o}))}{q_{o_i} c_{o_i}(K_{o_i}(\vec{o}) - \delta_{o_i}(\vec{o}))}$ and $w'_j = 1$ and get the following inequality:

$$\frac{q_x c_x(K_x(\vec{o}))}{q_{o_i} c_{o_i}(K_{o_i}(\vec{o}) - \delta_{o_i}(\vec{o}))} p_i q_{o_i} c_{o_i}(K_{o_i}(\vec{o}) - i) < p_i q_x c_x(K_x(\vec{o}))$$

   After rearranging the inequality we get that:

$$\frac{c_{o_i}(K_{o_i}(\vec{o}) - i)}{c_{o_i}(K_{o_i}(\vec{o}) - \delta_{o_i}(\vec{o}))} < 1$$

$$\frac{c_{o_i}(K_{o_i}(\vec{o}) - i)}{c_{o_i}(K_{o_i}(\vec{o}) - \delta_{o_i}(\vec{o}))} = \frac{c_{o_i}(\{K_{o_i}(\vec{o}) - i - \delta_{o_i}(\vec{o})\} + \delta_{o_i}(\vec{o}))}{c_{o_i}(\{K_{o_i}(\vec{o}) - i - \delta_{o_i}(\vec{o})\} + i)}$$

By Lemma 6.5 we have that as p_h is greater the amount we subtract from $c(S)$ is greater. Therefore since by definition $p_i \geq p_{\delta_{o_i}(\vec{o})}$, we have

$$c_{o_i}(\{K_{o_i}(\vec{o}) - i - \delta_{o_i}(\vec{o})\} + \delta_j(\vec{o})) > c_{o_i}(\{K_{o_i}(\vec{o}) - i - \delta_{o_i}(\vec{o})\} + i)$$

and this is a contradiction.

□

Since this establishes that all players want to stay with their current projects, it follows that $\vec{o}$ is a Nash equilibrium under the modified weights, and hence the proof of Theorem 3.4 is complete. ■

Proof of Theorem 3.5. We define the players' utilities and contributions to the optimum very similarly to how we defined them for the case of identical players. The definition of $prev_i$ is the same as in Definition 5.4.

Definition 6.8

$$\widetilde{u_i}(\vec{a}) = w_{a_i} p_i q_{a_i} \sum_{S \subseteq \{K_{a_i}(\vec{a}) - i\}} \left(\frac{z_i}{(\sum_{l \in S} z_l) + z_i} \prod_{l \in S} p_l q_{a_i} \prod_{l \in K_{a_i}(\vec{a}) - |S| - i} (1 - p_l q_{a_i}) \right)$$

Definition 6.9

$$\widehat{u}_i(\vec{a}) = w_{a_i} p_i q_{a_i} \prod_{\{l \in prev_i(K_{a_i}(\vec{a}))\}} (1 - p_l q_{a_i})$$

We now analyze the game with weights computed by the allocation algorithm described in Section 3. We already know that all unassigned players favor the same projects. We use this fact to ensure at each stage that players work on the projects they are supposed to work on in the optimal assignment. As before by giving the players different weights we impose an order on them. This order is a bit more complex than in the case of identical players, but we will show that the weights assure each player's utility is very similar to his contribution to the social welfare in the stage he was allocated.

One might hope to prove that with these weights the assignment $\vec{o}$ is a Nash Equilibrium. However, as suggested in Section 3, this is not necessarily correct. It is possible that in the last stage c^* of the algorithm there are fewer than $|X_{c^*}|$ unassigned players; in the other words, it is possible that there are more projects maximizing the utility than there are players remaining. In this case some of players might go to different projects than in $\vec{o}$. To solve this problem we define a new strategy vector $\vec{o'}$ (Definition 6.10) which we show has the same social welfare as $\vec{o}$ (Claim 6.11) and is a Nash equilibrium (Claim 6.14).

Definition 6.10 *$\vec{o'}$ is constructed as follows:*

- *For every player i that was not assigned in the last stage of the algorithm, we define $o'_i = o_i$.*
- *For every project $j \in X_{c^*}$ we compute the value*

$$c_j(\vec{o'}) = \sum_{S \subseteq \{K_j(\vec{o'})\}} \frac{z^*}{(\sum_{l \in S} z_l) + z^*} \prod_{l \in S} p_l q_j \prod_{l \in \{K_j(\vec{o'}) - S\}} (1 - p_l q_j)$$

where z^* is the weight defined for players that were assigned last.

- *Sort all the projects in X_{c^*} by their value for $w_j c_j(\vec{o'})$*
- *Assign every unassigned player to one of the top projects in X_{c^*} according to the sorting.*

Claim 6.11 *$\vec{o'}$ is an optimal assignment (i.e., $u(\vec{o'}) = u(\vec{o})$).*

Proof: By the construction of $\vec{o'}$ the only players that might not work on the same projects as in $\vec{o}$ are those that were assigned last. Also, by the construction, all these players are assigned to projects in X_{c^*}. Notice that all projects in X_{c^*} maximize $w_j \prod_{\{l \in K_j(\vec{a}^{c^*-1})\}}(1 - p_l q_j)q_j$. Hence, the contribution of the players assigned last is the same regardless of which specific project in X_{c^*} they are working on. Therefore $\vec{o'}$ is an optimal assignment. □

The next natural step is to prove that there exists an ϵ for which $\vec{o'}$ is a Nash equilibrium. However, before doing that we need to adjust a lemma we had for the case of identical players to the current setting:

Lemma 6.12 *Let d be the common denominator of all terms in the sets $\{w_j : j \in M\}$ and $\{p_i q_j : i \in N, j \in M\}$. There exists an ϵ such that for every project j and player i with a unique weight on project j, we have*

$$\widehat{u}_i(j, o'_{-i}) - \frac{1}{4d^{n+1}} \le \widetilde{u}_i(j, o'_{-i}) \le \widehat{u}_i(j, o'_{-i}) + \frac{1}{4d^{n+1}}$$

We omit the proof since it is very similar to the case of identical players. To see this, we present the adjusted definition for $X_i(j; S; \vec{a})$:

Definition 6.13 $X_i(j; S; \vec{a}) = \dfrac{z_i}{z_i + \sum_{l \in S} z_l} p_i q_j \prod_{l \in S} p_l q_j \prod_{l \in \{K_j(a_{-i}) - S\}}(1 - p_l q_j)$

Using this definition it is easy to derive a proof similar to the proof of Lemma 5.6. The lemma follows by using Lemma 5.7 as is.

Claim 6.14 *There exists an ϵ for which $\vec{o'}$ is a Nash equilibrium.*

Proof: Assume towards a contradiction that $\vec{o'}$ is not a Nash equilibrium. Thus, there exists a player i and a project $j \neq o'_i$ such that $\widetilde{u}_i(j, o'_{-i}) > \widetilde{u}_i(\vec{o'})$. By the weighting algorithm we have that $\widehat{u}_i(\vec{o'}) \ge \widehat{u}_i(j, o'_{-i})$. To see this, assume player i was assigned in stage c. If $c < c^*$, then $o'_i = o_i$ and o_i was one of the projects maximizing the marginal contribution to social welfare; if $c = c^*$, then by the definition of $\vec{o}$, the project o'_i must have been one of the projects maximizing this marginal contribution. So in either case we have

$$w_{o'_i} \prod_{\{l \in K_{o'_i}(\vec{a^c})\}} (1 - p_l q_{o'_i}) q_{o'_i} \ge w_j \prod_{\{l \in K_j(\vec{a^c})\}} (1 - p_l q_j) q_j.$$

By multiplying both sides with p_i we have that $\widehat{u}_i(\vec{o'}) \ge \widehat{u}_i(j, o'_{-i})$. Therefore we are left with two cases to consider:

- $\widehat{u}_i(\vec{o'}) > \widehat{u}_i(j, o'_{-i})$: This means that player i was assigned at a different stage than all the players working on project j were. Hence, player i has a unique weight on project j. Since every player always has a unique weight on the project he is allocated to by using the assumption of the claim, we get that:
$$\widehat{u}_i(j, o'_{-i}) + \frac{1}{4d^{n+1}} \ge \widetilde{u}_i(j, o'_{-i}) > \widetilde{u}_i(\vec{o'}) \ge \widehat{u}_i(\vec{o'}) - \frac{1}{4d^{n+1}}$$
This implies that $\widehat{u}_i(\vec{o'}) - \widehat{u}_i(j, o'_{-i}) < \dfrac{1}{2d^{n+1}}$. But by the definition of d, since $\widehat{u}_i(\vec{o'})$ and $\widehat{u}_i(j, o'_{-i})$ are not equal, they must differ by at least $\dfrac{1}{d^{n+1}}$, a contradiction.

- $\widehat{u}_i(\vec{o'}) = \widehat{u}_i(j, o'_{-i})$: If player i was not assigned in the last stage of the algorithm, then by Claim 6.15 we have that there exists another player with the same weight as player i working on project j. Now, Claim 6.16 delivers the desired contradiction since $\widetilde{u}_i(j, o'_{-i}) \leq \widetilde{u}_i(\vec{o'})$. If player i was assigned in the last stage then by the construction of $\vec{o'}$ he wouldn't want to move to any project in X_{c^*} that no other player which has the same weight is working on. As with players in earlier stages, player i cannot benefit by working on a project that some other player with the same weight is already working on.

□

Claim 6.15 *In every stage c of the algorithm, except for the last stage, for every project $j \in X_c$ there exists an* unassigned *player i such that $o_i = j$.*

Proof: Assume towards a contradiction that in some stage c there exists a project $j \in X_c$ for which all the players working on it in $\vec{o}$ have already been assigned. If i is a player left *unassigned* after stage c then $u(j, o_{-i}) > u(\vec{o})$. This is because in each stage the projects in the set X_c maximize the marginal contribution. Since the utility is submodular, the marginal contribution of the projects can only decrease in every stage. Hence, player i's marginal contribution to project j is greater than his contribution to project o_i. Also, by removing player i from project o_i the marginal contribution of the rest of the players working on o_i can only increase. From this we conclude that $u(j, o_{-i}) > u(\vec{o})$, in contradiction to $\vec{o}$ being an optimal assignment. □

Claim 6.16 *For every two players i and c that have the same weight, $\widetilde{u}_i(o_c, o_{-i}) \leq \widetilde{u}_i(\vec{o})$ if for every player i and project j such that the weight of player i is unique among players working on project j we have $\widehat{u}_i(j, o'_{-i}) - \frac{1}{4d^{n+1}} \leq \widetilde{u}_i(j, o'_{-i}) \leq \widehat{u}_i(j, o'_{-i}) + \frac{1}{4d^{n+1}}$*

Proof: Let $z_i = z_c = z^*$. We have

$$\widetilde{u}_i(o_c, o_{-i}) = (1 - p_c q_{o_c}) \left(w_{o_c} p_i q_{o_c} \sum_{S \subseteq \{K_{o_c}(\vec{o}) - c\}} \left(\frac{z^*}{(\sum_{l \in S} z_l) + z^*} \prod_{l \in S} p_l q_{o_c} \prod_{l \in \{K_{o_c}(\vec{o}) - c - S\}} (1 - p_l q_{o_c}) \right) \right) +$$

$$p_c q_{o_c} \left(w_{o_c} p_i q_{o_c} \sum_{S \subseteq \{K_{o_c}(\vec{o}) - c\}} \left(\frac{z^*}{(\sum_{l \in S} z_l) + 2z^*} \prod_{l \in S} p_l q_{o_c} \prod_{l \in \{K_{o_c}(\vec{o}) - c - S\}} (1 - p_l q_{o_c}) \right) \right)$$

By rearranging the terms we have that $\widetilde{u}_i(o_c, o_{-i}) =$

$$\widetilde{u}_i(o_c, o_{-i,c}) - w_{o_c} p_c q_{o_c} p_i q_{o_c} \sum_{S \subseteq \{K_{o_c}(\vec{o}) - c\}} \left(\prod_{l \in S} p_l q_{o_c} \prod_{l \in \{K_{o_c}(\vec{o}) - c - S\}} (1 - p_l q_{o_c}) \left(\frac{z^*}{(\sum_{l \in S} z_l) + z^*} - \frac{z^*}{(\sum_{l \in S} z_l) + 2z^*} \right) \right)$$

By considering the empty set in the summation we get that:

$$\widetilde{u}_i(o_c, o_{-i}) \leq \widetilde{u}_i(o_c, o_{-i,c}) - \frac{1}{2} w_{o_c} p_c q_{o_c} p_i q_{o_c} \prod_{\{l \in K_{o_c}(\vec{o}) - c\}} (1 - p_l q_{o_c})$$

By the definition of the common denominator we have $w_{o_c} p_c q_{o_c} p_i q_{o_c} \prod_{\{l \in K_{o_c}(\vec{o}) - c\}} (1 - p_l q_{o_c}) \geq \frac{1}{d^{n+1}}$ and hence

$$\widetilde{u}_i(o_c, o_{-i}) \leq \widetilde{u}_i(o_c, o_{-i,c}) - \frac{1}{2d^{n+1}}$$

By the assumption we have that

$$\widetilde{u}_i(o_c, o_{-i,c}) \le \widehat{u}_i(o_c, o_{-i,c}) + \frac{1}{4d^{n+1}} = \widehat{u}_i(o_c, o_{-i}) + \frac{1}{4d^{n+1}}.$$

Because player i and c have the same weight we have that according to the algorithm $\widehat{u}_i(o_c, o_{-i}) = \widehat{u}_i(\vec{o})$. Therefore we have

$$\widetilde{u}_i(o_c, o_{-i}) \le \widehat{u}_i(\vec{o}) - \frac{1}{4d^{n+1}} \le \widetilde{u}_i(\vec{o}).$$

□

Because we have established that ϵ with the desired properties exists, this completes the proof of Theorem 3.5. ■

7 Appendix: Proofs of Results from Section 4

Proof of Theorem 4.2. As in other results on re-weighting players, we use the weights to simulate an ordering on the players. That is, we arrange the players in some specific order, and then we announce that all the credit on a project will be allocated to the first player in the order to succeed at it. We first describe how to construct such an ordering for which every Nash equilibrium in the resulting game is socially optimal, and then we show how to approximately simulate this order using weights.

Let $\vec{o}$ be an optimal assignment of players to projects in which there is at most one player working on each project. The following lemma establishes that there must be some player i who would choose his own project o_i if he were placed first in the order.

Lemma 7.1 *If in the optimal assignment there is at most one player working on each project then there exists a player i such that* $\max_j w_j p_{i,j} \le w_{o_i} p_{i,o_i}$

Proof: Assume towards a contradiction that such a player does not exist. Then for every player i there exists a project g_i such that $w_{g_i} p_{i,g_i} > w_{o_i} p_{i,o_i}$. Since in the optimal assignment there is at most one player working on each project, we can picture the assignment as a matching between the projects and the players. Consider the bipartite graph which has the players on the left side, the projects on the right side and both the edges of the optimal matching $\{(i, o_i)\}$ and edges from each player to his preferred project $\{(i, g_i)\}$. We color the edges in the first of these sets blue and the edges in the second of these sets red. This bipartite graph has $2n$ nodes and $2n$ edges, and it therefore contains a cycle C. The cycle C has interleaving red and blue edges, because each player on C has exactly one incident blue edge and one incident red edge. Hence, we can form a new perfect matching between players and projects by re-matching each player on C with the project to which he is matched using his red edge rather than his blue edge. Since all the players strictly prefer the projects to which they are connected by red edges, the social welfare of this new matching is greater than the social welfare of the blue matching, which contradicts the optimality of the blue matching.

□

Given this lemma, we can construct the desired ordering by induction. We identify a player i with the property specified in Lemma 7.1 and place him first in the order. Since he knows he will receive all the credit from any project he succeeds at, he will choose his own project in the optimal solution o_i. We now remove i and o_i from consideration and proceed inductively; the structure of the optimum on the remaining players is unchanged, so we can apply Lemma 7.1 on this smaller instance and continue in this way, thus producing an ordering.

The remainder of the proof is similar to the analysis for the case of identical players: we simulate the ordering $i_1, i_2, \ldots, i_n$ using weights by choosing a sufficiently small $\epsilon > 0$ and assigning player i_c (the c^{th} player in the order) a weight of $z_{i_c} = \epsilon^c$.

We now show:

Claim 7.2 *There exists an ϵ for which any Nash equilibrium in the game with the weights $\{z_i\}$ is an optimal assignment.*

Proof: For this proof we use similar definitions to the proof for re-weighting identical players. As before, we define d to be the common denominator of all probabilities and weights. We use the result of Claim 2.11:

$$\widehat{u}_i(j, a_{-i}) - \frac{1}{4d^{n+1}} \leq \widetilde{u}_i(j, a_{-i}) \leq \widehat{u}_i(j, a_{-i}) + \frac{1}{4d^{n+1}}.$$

We omit the proof since it is similar to the corresponding proofs for identical players and players of different abilities.

Assume towards a contradiction that $\vec{a}$ is a Nash equilibrium but $u(\vec{a}) < u(\vec{o})$. Let i be the player with the greatest weight such that $\widehat{u}_i(\vec{a}) < \widehat{u}_i(\vec{o})$. Since $u(\vec{a}) < u(\vec{o})$ such a player exists. Note that i's weight is greater than the weight of any other player working on o_i (i is the misplaced player of the highest weight) and hence $\widehat{u}_i(\vec{o}) = \widehat{u}_i(o_i, a_{-i})$. By using Claim 2.11 and since $\vec{a}$ is a Nash equilibrium:

$$\widehat{u}_i(o_i, a_{-i}) - \frac{1}{4d^{n+1}} \leq \widetilde{u}_i(o_i, a_{-i}) \leq \widetilde{u}_i(\vec{a}) \leq \widehat{u}_i(\vec{a}) + \frac{1}{4d^{n+1}}$$

This implies that $\widehat{u}_i(o_i, a_{-i}) - \widehat{u}_i(\vec{a}) \leq \frac{1}{2d^{n+1}}$, which is a contradiction since by the definition of d we have that $\widehat{u}_i(o_i, a_{-i}) - \widehat{u}_i(\vec{a}) \geq \frac{1}{d^{n+1}}$. □

Since we have established the existence of a sufficiently small ϵ for use in the construction of weights, this completes the proof of Theorem 4.2. ■

Proof of Theorem 4.3. Vetta's analysis of general monotone valid-utility games establishes that the price of anarchy (PoA) is ≤ 2 for all such games. In order to show that the PoA is < 2 for our game, we first describe a variation on Vetta's original proof establishing the upper bound $PoA \leq 2$. The proof is composed of a chain of inequalities such that when $PoA = 2$ all the inequalities are equalities. We then show that in our game it is not possible to have all these inequalities be equalities without reaching a contradiction.

Thus, to begin, we describe the variant of Vetta's proof that will be amenable to this strategy. This begins with the following definitions.

Definition 7.3

- $\vec{a} \oplus \vec{b}$: *For vectors $\vec{a}$ and $\vec{b}$ defined on the same set of players, $\vec{a} \oplus \vec{b}$ is a new vector in which each player i such that $a_i \neq b_i$ is duplicated and one copy of i works on a_i while the other works on b_i.*
- $u(o_i|\vec{a}) = u(o_i + \vec{a}) - u(\vec{a})$. *This is the contribution to the social welfare of adding a duplicate of player i that works on o_i to the players in $\vec{a}$.*

Now, here is an argument that $PoA \leq 2$ for all monotone valid-utility games. First, we recall the four properties that characterize valid utility games, as expressed also in the proof of Claim 2.1:

1. $u(\vec{a})$ is submodular.

2. $u(\vec{a})$ is monotone.

3. $u_i(\vec{a}) \geq u(a_i|a_{-i})$.

4. $u(\vec{a}) \geq \sum_i u_i(\vec{a})$.

Let $\vec{o}$ be the strategy vector that maximizes the social welfare and let $\vec{a}$ be a Nash equilibrium. Since the utility function is monotone submodular we know that $u(\vec{o}) \leq u(\vec{a} \oplus \vec{o})$. The bound of 2 follows by showing that $u(\vec{a} \oplus \vec{o}) \leq 2 \cdot u(\vec{a})$.

First,

$$u(\vec{a} \oplus \vec{o}) \leq u(\vec{a}) + \sum_{i:o_i \neq a_i} u(o_i|\vec{a}).$$

By decreasing marginal utility we have that:

$$u(o_i|\vec{a}) \leq u(o_i|\vec{a}_{-i})$$

Also by decreasing marginal utility we have that:

$$u(o_i|a_{-i}) \leq u_i(o_i, a_{-i}) \leq u_i(\vec{a}),$$

where the first inequality follows from the third requirement of a monotone valid-utility game, and the second inequality follows from the fact that $\vec{a}$ is a Nash equilibrium.

Finally, by the fourth requirement for a monotone valid utility game we get that $\sum_{i:o_i \neq a_i} u_i(\vec{a}) \leq u(a)$, and this concludes the proof that $PoA \leq 2$.

We now move on to show that for our game, in fact the price of anarchy is strictly less than 2. The following definition will be useful.

Definition 7.4 *$\pi(\vec{a})$ is the set of all projects that at least one player in $\vec{a}$ is working on. $\pi(\vec{a}) = \{j|\exists i\ a_i = j\}$*

Assume towards a contradiction that there exists an instance of the Project Game for which there exists a Nash equilibrium $\vec{a}$ and an optimal assignment $\vec{o}$ such that $u(\vec{o}) = 2u(\vec{a})$. Moreover, supposing such an example exists, we choose one with the minimum possible number of players. Given this minimality condition, it follows that for each player i, there is some project j for which $p_{i,j} > 0$, since otherwise player i does not contribute to the social welfare in any assignment, and so we can remove player i and have a smaller instance for which the price of anarchy is still equal to 2.

We have the following lemma.

Lemma 7.5 *For all players i, we have $p_{i,a_i} > 0$*

Proof: As noted above, we know that for every player i there exists a project j such that $p_{i,j} > 0$. Assume towards a contradiction that there exists a player i such that $p_{i,a_i} = 0$. Hence, $u_i(\vec{a}) = 0$, and since $\vec{a}$ is a Nash equilibrium, we have that $u_i(j, a_{-i}) = 0$ for all projects j. This implies that $p_{i,j} = 0$ for all projects j, which contradicts the fact that $p_{i,j} > 0$ for some project j. □

Now, since $u(\vec{o}) = 2u(\vec{a})$, Vetta's proof of $PoA \leq 2$ implies that $u(\vec{o}) = u(\vec{a} \oplus \vec{o})$. Therefore for every player i such that $a_i \neq o_i$, we have $u(a_i|\vec{o}) = 0$. Since $p_{i,a_i} > 0$ by Lemma 7.5, we have $s_{a_i}(K_{a_i}(\vec{o})) = w_{a_i}$. If $s_{a_i}(K_{a_i}(\vec{o})) = w_{a_i}$ then there exists a player l such that $o_l = a_i$ and $p_{l,a_i} = 1$. This brings us to the following intermediate corollary:

Corollary 7.6 *For every project $j \in \pi(\vec{a})$ there exists a player l such that $o_l = j$ and $p_{l,j} = 1$*

By Vetta's proof of $PoA \leq 2$ we also have that $u(o_i|a_{-i}) = u_i(o_i + a_{-i})$. Recall that the utility of a player is the average of his marginal contributions to the social welfare over all possible orderings of the players. Since the utility is submodular the smallest term in this average is $u(o_i|a_{-i})$. Hence, if $u(o_i|a_{-i}) = u_i(o_i + a_{-i})$ it has to be the case that $s_{o_i}(K_{o_i}(a_{-i})) = 0$. When combined with Lemma 7.5 this implies that $K_{o_i}(a_{-i})$ is empty. Together with Corollary 7.6 we have that for every project $j \in \pi(\vec{a})$ there exists a player l such that $u_l(o_l + a_{-l}) = w_{o_l}$. Since $\vec{a}$ is a Nash equilibrium this also implies that $u_l(\vec{a}) = w_{o_l}$. Therefore $(n - |\pi(\vec{a})|)$ players have a utility of 0 working on $\vec{a}$. If $|\pi(\vec{a})| = n$ then all players have the same utility as they have in the optimal assignment, and hence $\vec{a}$ has to be an optimal assignment. Otherwise, since there is at least one player i such that $u_i(\vec{a}) = 0$, we have a contradiction to Lemma 7.5. This contradiction completes the proof of Theorem 4.3. ■