Profit Sharing with Thresholds and Non-monotone Player Utilities

Elliot Anshelevich[*] John Postl[†]

July 8, 2014

Abstract

We study profit sharing games in which players select projects to participate in and share the reward resulting from that project equally. Unlike most existing work, in which it is assumed that the player utility is monotone in the number of participants working on their project, we consider non-monotone player utilities. Such utilities could result, for example, from "threshold" or "phase transition" effects, when the total benefit from a project improves slowly until the number of participants reaches some critical mass, then improves rapidly, and then slows again due to diminishing returns.

Non-monotone player utilities result in a lot of instability: strong Nash equilibrium may no longer exist, and the quality of Nash equilibria may be far away from the centralized optimum. We show, however, that by adding additional requirements such as players needing permission to leave a project from the players currently on this project, or instead players needing permission to a join a project from players on that project, we ensure that strong Nash equilibrium always exists. Moreover, just the addition of permission to leave already guarantees the existence of strong Nash equilibrium within a factor of 2 of the social optimum. In this paper, we provide results on the existence and quality of several different coalitional solution concepts, focusing especially on permission to leave and join projects, and show that such requirements result in the existence of good stable solutions even for the case when player utilities are non-monotone.

1 Introduction

Resource selection games, in which players choose which project, market, or group to participate in, and then receive utility based on the number of people who choose the same strategy as them, have been heavily studied in algorithmic game theory (see for example [1, 2, 10, 14, 16]). Closely related to such games are coalition formation games, in which players choose which coalition to participate in, and player utility depends on the members of their coalition (such utilities are often called *hedonic* [3–6, 8, 11, 13]). In most such games, player utility is assumed to either only *increase* with the number of players that choose the same group or project, or only *decrease* with the number of players that choose the same group or project (for example in [14], where more people choosing the same project means more competition).

In many important situations, however, player utility may not be a monotone function in the number of players who choose the same strategy. For example, consider the scenario where players choose which project to work on (or form teams in order to submit funding proposals). It is mostly true that more participants will improve the overall outcome of the project; in most existing work the overall success of the project is assumed to either be convex or concave nondecreasing as a

[*] Computer Science Dept, Rensselaer Polytechnic Institute, Troy, NY 12180

[†] Computer Science Dept, Rensselaer Polytechnic Institute, Troy, NY 12180

function of the number of participants. However, many projects exhibit "threshold" or "phase-transition" behavior: until there is a critical mass of participants there will be very little progress, and after that critical mass, each additional participant only makes a marginal amount of difference to the project's success. In such a scenario, and assuming that the benefit (e.g., credit) from a project's success is divided equally among the participants, the utility of a player as a function of the number of project participants may increase until this threshold is reached, and then begin to decrease. In this paper, we study such resource selection games, but for player utility functions which do not have to be monotone, and thus are much more general.

Our Model. More concretely, consider the following simple profit sharing game in which multiple projects are available to the players. There are n identical players and m projects; the strategy set of each player is $\{\emptyset, 1, 2, \ldots, m\}$. Each project k has a payoff function $p_k : \mathbb{N}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that is monotone nondecreasing in the number of players working on the project: this function represents the total benefit or success of this project. Each player selects a project and shares the reward equally with everyone else working on that project. That is, the utility of a player i that selects strategy $k \neq \emptyset$ is $u_i(\mathbf{s}) = \frac{p_k(x_k(\mathbf{s}))}{x_k(\mathbf{s})}$, where $x_k(\mathbf{s})$ is the number of players selecting k in solution $\mathbf{s}$. This payoff scheme is very robust and can model many different situations. For example, if the project payoff function is convex, then the individual reward function will also be increasing. On the other extreme, if the project payoff function is constant, then the individual reward function is strictly decreasing. Finally, if the project payoff function is a "threshold" function, then the individual reward function will alternate between being increasing and decreasing. More generally, the individual payoff function may not even be single-peaked: it can increase, decrease, and then increase again.

While games of this type with either convex or concave project payoff functions p_k have been shown to have very nice properties, arbitrary non-decreasing payoff functions cause a lot of instability. First, observe that Nash equilibrium is not a good solution concept for threshold payoff functions (and tends to perform very poorly compared to the social optimum), since players typically cannot satisfy thresholds with unilateral deviations. Just as teams form to tackle difficult projects, we need to consider group deviations. Unfortunately, unlike for the case when p_k are constant or convex, strong Nash equilibrium does not exist when p_k are threshold functions. To illustrate this, consider the following simple example of $n \geq 3$ players and two projects.

Example 1. Project A is a small project that any number of people can work on and share the credit, while project B is a large project that will only succeed with a large team. Formally, project A has total value of 2 no matter how many players choose to work on it, while project B requires n players (any fewer players working on project B would get nothing), and has a total payoff of n if there are n players working on it. In other words, p_A is a constant function, while p_B is a threshold function with the threshold being n. If everyone chooses to work on project B, then the utility of each player is 1, and so they would each rather unilaterally switch to project A. It is not difficult to verify that the only Nash Equilibrium solution is for all players to pick project A, and thus is very far from optimum. Perhaps even worse, strong Nash equilibrium does not exist: if everyone chooses project A, then all the players could together deviate to project B, and obtain 1 utility instead of $\frac{2}{n}$.

Contractual Stability. Given the motivation of players forming teams to work on projects, or forming teams to submit grant proposals, the above bad example does not seem satisfactory to the authors. While it is true that strong Nash equilibrium does not exist above, this occurs because of a strange player interaction. When all players choose project A, they together decide to switch to project B; everyone is better off. After this occurs, a single player says: "Aha! Now that you have left project A unoccupied, I will leave project B," a side-effect of which is that all the other players receive nothing, since every player is an integral part of the team. It is unusual that team members drop out once a grant proposal has been funded: this is because there is often a contractual obligation for them to perform the work, and because by dropping out they will incur the bad will and shame from the other members of the team. If players are not allowed to leave a project without the permission of the other project members, then the above example has a wonderful stable solution where everyone chooses project B. On the other hand, someone is usually not allowed to join a grant proposal or a team project without permission of its members. If this were true in the above example, then once again a coalitionally stable solution exists. This line of thought suggests the study of different coalitional stability concepts in which players must obtain permission to leave or join projects.

In this paper, we provide an analysis of this generalized profit sharing game. In particular, we provide results on the existence and quality of many different coalitional solution concepts, focusing especially on permission to leave and join projects. These ideas are captured precisely by solution concepts from *hedonic games* literature [3–6, 8, 11, 13]; to our knowledge we are the first to consider them in the context of non-cooperative games.

1.1 Our Contributions

For the Profit Sharing model defined above, we introduce stability concepts which include permission to join and permission to leave. For permission to leave, players cannot leave a project with other players remaining on it, unless the utility of those players does not decrease as a result of their leaving. Permission to leave can be thought of as the enforcement of a contract: when a player decides to work on a project with a group, she is essentially entering into a contract with them to complete the project, and she will only be let out of this contract if it benefits the rest of the participants. Using the terminology from hedonic games, we call strong Nash stability with permission to leave *strong contractual Nash stability (SCNS)*. Unlike strong Nash equilibrium, we show that *SCNS* always exists. Furthermore, we show there always exists a *SCNS* solution which is within a factor of 2 of the social optimum, thus implying that the price of stability with respect to *SCNS* is at most 2. We show that this bound is asymptotically tight. Our results indicate that adding contractual obligations to not leave projects before they are completed not only results in the creation of coalitionally stable solutions, but also in the creation of high welfare stable solutions.

For permission to join, players cannot join a project with other players already on it unless the utility of those players does not decrease as a result of their joining. Permission to join is a natural idea, since a person cannot typically work on a team with other people unless they allow her to join their group. Again borrowing the terminology from coalition formation, we call strong Nash stability along with permission to join *strong individual stability (SIS)*. We show that *SIS* solutions always exist as well. However, the quality of *SIS* can be very bad; the price of stability with respect to *SIS* can be n, the number of players. This is because a player can join a high value project by itself, and since others cannot join to lower the player's utility from that project, it has no incentive to ever switch. The other players might be left on projects with high thresholds that cannot be satisfied without everyone's cooperation, which means they are receiving very little utility potentially. Essentially, permission to join does not encourage cooperation as well as permission to leave does.

Finally, we consider what happens when both permission to leave and permission to join are in effect: we call the corresponding solution concept *strong contractual individual stability (SCIS)*. Not surprisingly, this stability notion is very strong: we show that the centrally optimal solution is always *SCIS*. However, since every *SIS* state is a *SCIS* state, the price of anarchy with respect to

SCIS can be as high as n.

Our results show that unlike in cases where p_k is convex or concave, in profit sharing where the project benefit can contain threshold effects (or in general when player utilities are not monotone), it is crucial to have some coordinating mechanism like permission to leave a project or permission to join a project. Without such a mechanism coalitionally stable solutions are not guaranteed to exist, and Nash equilibrium can be very inefficient. Once permission to leave is added through contractual obligations, however, this results in the creation of high-quality solutions which are resilient even to deviations by coalitions.

1.2 Related Work

Our model is a generalization of the well-studied market sharing game [2, 10, 16]. In the market sharing game, each market has an associated value that is shared equally among the players that have selected that particular market. In other words, it is a special case of our model in which the payoff functions are constants. The threshold model can be thought of as the market sharing game in which the value of the market is not awarded to the players until a sufficient number of players have selected the market. The market sharing game is a simple example of a monotone valid-utility game [16, 18], but our model is not: the utility of a player can be less than their marginal contribution to the total welfare, and the total player utility is not submodular.

The stability concepts used to analyze our model are borrowed primarily from hedonic games literature; consult [3–6, 8, 11, 13] and their references. Hedonic games are coalition formation games in which how much a player values a group depends solely on who is in her group and is independent of how the remaining players are partitioned into groups. Our model has this property as well, since a player's payoff depends only on how many people are on her chosen project. Hedonic games differ from the non-cooperative game we study primarily in that they are cooperative games in which players form groups rather than select projects: the utility of a player depends only on the members of the group, and not on which project they select together. In fact, in many hedonic game formulations, the players have ordinal preferences over groups rather than utilities. Hedonic games literature consists primarily of results characterizing the existence of stability concepts such as the core, (strong) Nash stability, (strong) individual stability, (strong) contractual Nash stability, etc. We use these concepts to model permission to join and to leave projects in our model. However, the *quality* of stable solutions for hedonic games has received little attention; in contrast we consider the price of stability and price of anarchy [17] with respect to these stability concepts.

Numerous other games are motivated by project/group selection, and, in some cases, apply concepts from hedonic games literature to non-cooperative models. Kutten et al. [15] define several group formation games where there is only a single project with a payoff function that is essentially our threshold payoff function. Players form groups (that require permission to join) that compete for this project, but only one group receives a payoff. Chalkiadakis et al. [7] define a cooperative game that is similar to our threshold model, but it has additional constraints on the project payoff functions, and there is an infinite number of each type of project. Augustine et al. [1] define non-cooperative project selection games based on monotone convex cooperative games and provide quality of equilibria and convergence results for them. Kleinberg and Oren [14] define a project selection game with decreasing player payoff functions, unlike ours which may not be monotone, and their results focus on redesigning project payoff functions to ensure that the optimal solution is stable. Finally, Feldman et al. [9] define a class of non-cooperative games called hedonic clustering games, and provide quality of equilibrium results for them. Their model has little in common with ours, but they borrow heavily from hedonic games literature and apply concepts like price of anarchy, as we do.

2 Definitions

Model. As mentioned in the Introduction, the Profit Sharing game we consider consists of n identical players, and m projects, with each player strategy being to choose one of these projects. We let $[n] = \{1, 2, \dots, n\}$ and $[m] = \{1, 2, \dots, m\}$. Each project has a non-decreasing payoff function p_k. We define $x_k(\mathbf{s})$ to be the number of players who choose project k in state s; then the utility of a player i with $s_i = k$ is $u_i(\mathbf{s}) = \frac{p_k(x_k(\mathbf{s}))}{x_k(\mathbf{s})}$. We allow players to opt out of playing by selecting the null strategy $\emptyset$ where for all players i, $u_i(\emptyset, s_{-i}) = 0$.

While our results will generally hold for arbitrary non-decreasing functions p_k, we will sometimes refer to the important special case of threshold payoff functions. We define a threshold payoff function for a project k to be

$$p_k(x) = \begin{cases} 0, & \text{if } x < t_k \\ c_k, & \text{otherwise} \end{cases}$$

where $t_k \in \mathbb{Z}_{>0}$ is the threshold of project k and $c_k \in \mathbb{R}_{\geq 0}$ is the value of the project k once the threshold has been met, which will be split equally among all players on project k.

Stability Concepts. We now introduce several stability concepts. Many of these concepts are adapted from hedonic games literature [3–6, 8, 11, 13] to fit into the framework of non-cooperative game theory. These concepts take standard non-cooperative game theory equilibrium concepts such as *Nash equilibrium* and *strong Nash equilibrium* and add permission to join and/or permission to leave. "Individually stable" refers to permission to join, and "contractually stable" refers to permission to leave a project.

A state is a *Nash equilibrium* or *Nash stable (NS)* if no player can change her strategy to improve her utility. That is, if for every player i and every strategy $s'_i \in [m]$, we have $u_i(\mathbf{s}) \geq u_i(s'_i, s_{-i})$.

A state is *individually stable (IS)* if for every player i and every strategy $s'_i \in [m]$, either $u_i(\mathbf{s}) \geq u_i(s'_i, s_{-i})$ or there exists player j with $s_j = s'_i$ such that $u_j(\mathbf{s}) > u_j(s'_i, s_{-i})$.

A state is *contractually Nash stable (CNS)* if for every player i and every strategy $s'_i \in [m]$, either $u_i(\mathbf{s}) \geq u_i(s'_i, s_{-i})$ or there exists player j with $s_j = s_i$ such that $u_j(\mathbf{s}) > u_j(s'_i, s_{-i})$.

A state is *contractually individual stable (CIS)* if for every player i and every strategy $s'_i \in [m]$, either $u_i(\mathbf{s}) \geq u_i(s'_i, s_{-i})$ or there exists player j with $s_j = s'_i$ such that $u_j(\mathbf{s}) > u_j(s'_i, s_{-i})$ or there exists player j with $s_j = s_i$ such that $u_j(\mathbf{s}) > u_j(s'_i, s_{-i})$.

A state is a *strong Nash equilibrium* or *strong Nash stable (SNS)* if for every non-empty subset of players $C \subseteq [n]$ and for every $s'_C \in [m]^C$, there exists a player $i \in C$ such that $u_i(\mathbf{s}) \geq u_i(s'_C, s_{-C})$.

A state is *strong individually stable (SIS)* if for every non-empty subset of players $C \subseteq [n]$ and for every $s'_C \in [m]^C$, there exists a player $i \in C$ such that $u_i(\mathbf{s}) \geq u_i(s'_C, s_{-C})$ or there exists player $j \notin C$ with $s_j = s'_i$ such that $u_j(\mathbf{s}) > u_j(s'_C, s_{-C})$.

A state is *strong contractually Nash stable (SCNS)* if for every non-empty subset of players $C \subseteq [n]$ and for every $s'_C \in [m]^C$, there exists a player $i \in C$ such that $u_i(\mathbf{s}) \geq u_i(s'_C, s_{-C})$ or there exists $j \notin C$ with $s_j = s_i$ such that $u_j(\mathbf{s}) > u_j(s'_C, s_{-C})$.

A state is *strong contractually individually stable (SCIS)* if for every non-empty subset of players $C \subseteq [n]$ and for every $s'_C \in [m]^C$, there exists a player $i \in C$ such that $u_i(\mathbf{s}) \geq u_i(s'_C, s_{-C})$ or there exists player $j \notin C$ with $s_j = s'_i$ such that $u_j(\mathbf{s}) > u_j(s'_C, s_{-C})$ or there exists $j \notin C$ with $s_j = s_i$ such that $u_j(\mathbf{s}) > u_j(s'_C, s_{-C})$.

Price of Anarchy and Price of Stability. The global objective function we consider (i.e., the social welfare) is simply $u(\mathbf{s}) = \sum_i u_i(\mathbf{s})$. Let $\mathcal{N}(\mathcal{G}, SC)$ denote the set of states that are stable with respect to the stability concept SC. The *price of anarchy with respect to SC* of $\mathcal{G}$ is $\frac{u(\mathbf{s}^*)}{\min_{\mathbf{s} \in \mathcal{N}(\mathcal{G}, SC)} u(\mathbf{s})}$. The *price of stability with respect to SC* of $\mathcal{G}$ is $\frac{u(\mathbf{s}^*)}{\max_{\mathbf{s} \in \mathcal{N}(\mathcal{G}, SC)} u(\mathbf{s})}$

3 Properties of Profit Sharing with Thresholds

First, we note that Profit Sharing model is an exact potential game [17], with the standard potential function $\Phi(\mathbf{s}) = \sum_{k\in[m]} \sum_{l=1}^{x_k(\mathbf{s})} \frac{p_k(l)}{l}$, which implies that pure Nash equilibrium always exists. As we saw in Example 1, there is no such guarantee for strong Nash equilibrium.

Claim 2. *A Nash equilibrium always exists. A strong Nash equilibrium need not exist.*

As mentioned in Example 1 in the Introduction, the quality of Nash equilibrium can be very bad even for threshold payoff functions. Although strong Nash equilibria are not guaranteed to exist (and in fact don't exist for even very simple cases), when they do, they are efficient.

Theorem 3. *The price of stability with respect to* NS *is at most n, and the price of anarchy with respect to* NS *is unbounded.*

Proof. For the price of anarchy, consider Example 1 in the Introduction, but set the value of project A to 0. If all of the players are working on project A, then it is a Nash equilibrium, since any unilateral deviations to project B will not benefit the players.

To establish a lower bound of n on the price of stability, consider the same example, but with value of project A replaced by $1+\epsilon$, for some $\epsilon > 0$. Using the same argument as Example 1, it can be demonstrated that the only Nash equilibrium is the state in which every player works on project A.

We will now give an algorithm that creates a Nash equilibrium that is within a factor of n of optimal at most.

Let $\mathbf{s}^t$ denote the state in which t players are working on projects, while the remaining players have strategy $\emptyset$. Start with the state $\mathbf{s}^0$ in which the strategy of every player is $\emptyset$. While there exist players that are not working on a project, select the project k that maximizes

$$q_k(\mathbf{s}^t) = \max_{y=x_k(\mathbf{s}^t)+1\ldots n-t} \frac{p_k(y)}{y}.$$

Assign $y - x_k(\mathbf{s}^t)$ additional players to k, where y is the maximum value between $x_k(\mathbf{s}^t)+1$ and $x_k(\mathbf{s}^t)+n-t$ that maximizes $\frac{p_k(y)}{y}$. Call the resulting state $\mathbf{s}^{t+y-x_k(\mathbf{s}^t)}$.

First, we will demonstrate that the state created by the algorithm $\mathbf{s} = \mathbf{s}^n$ is a Nash equilibrium. Let r be the project which had players added to it in the last step of the algorithm. We observe that for every project $k \neq r$, it must be the case that

$$\frac{p_k(x_k(\mathbf{s}))}{x_k(\mathbf{s})} > \frac{p_k(x_k(\mathbf{s})+1)}{x_k(\mathbf{s})+1}.$$

This is because at the end of each step of our algorithm, we assigned enough players to project k to maximize the player payoff function; if assigning one more player would have increased the player payoff, the algorithm would have done so. The sole exception to this property is the final step in which all of the remaining players are assigned. It could be the case that the above property does not hold for project r, because the "extra" player did not exist to be assigned to the project.

We claim no player will want to deviate to another project. Consider a player i working on any project k. Since the algorithm never removes players from projects, q_k is monotone decreasing for each project as the algorithm progresses: this is because at each point it is the absolute maximum given the number of players remaining. Thus we can suppose, without loss of generality, that player i was in the last batch of players assigned to k in state $\mathbf{s}^t$. Then for all $k' \neq r$, we know that

$$\begin{aligned} u_i(\mathbf{s}) &= q_k(\mathbf{s}^t) \\ &\geq q_{k'}(\mathbf{s}^t) \\ &\geq q_{k'}(\mathbf{s}) \\ &= \frac{p_{k'}(x_{k'}(\mathbf{s}))}{x_{k'}(\mathbf{s})} \\ &> \frac{p_{k'}(x_{k'}(\mathbf{s})+1)}{x_{k'}(\mathbf{s})+1} \end{aligned}$$

Thus, i will not deviate to project $k' \neq r$. Now consider i deviating to project r, and consider $q_r(\mathbf{s}^t)$. Since i is assigned to project $k \neq r$ at that point in the algorithm, this means that at least $x_r(\mathbf{s}) - x_r(\mathbf{s}^t) + 1$ players are available at that point, and so $q_r(\mathbf{s}^t) \geq p_r(x_r(\mathbf{s})+1)/(x_r(\mathbf{s})+1)$. Thus, $u_i(\mathbf{s}) = q_k(\mathbf{s}^t) \geq q_r(\mathbf{s}^t) \geq p_r(x_r(\mathbf{s})+1)/(x_r(\mathbf{s})+1)$, and so player i would not have incentive to deviate to project r.

We will now demonstrate that $u(\mathbf{s}) \geq \frac{1}{n}u(\mathbf{s}^*)$. Let l denote the largest amount of utility that any single player can achieve in any solution. We observe that at least one player is receiving l utility after the first step of the algorithm is completed. Since p_k is non-decreasing, we know that $u(\mathbf{s}) \geq l$. Clearly, $ln \geq u(\mathbf{s}^*)$, which completes our proof. □

Theorem 4. *The price of stability and price of anarchy with respect to* SNS *is at most 2, and this bound is tight.*

Proof. Let $\mathbf{s}$ be a strong Nash equilibrium. We begin by proving for every pair of players i, j, $u_i(\mathbf{s}) \geq \frac{1}{2}u_j(\mathbf{s})$. That is, every player's utility is within a factor of 2 of each other's. If $s_i = s_j$, then clearly they have the same utility. Let $k = s_j \neq s_i$. Since $\mathbf{s}$ is a Nash equilibrium, we observe that

$$\begin{aligned} u_i(\mathbf{s}) &\geq \frac{p_k(x_k(\mathbf{s})+1)}{x_k(\mathbf{s})+1} \\ &\geq \frac{p_k(x_k(\mathbf{s}))}{x_k(\mathbf{s})+1} \\ &\geq \frac{p_k(x_k(\mathbf{s}))}{2 \cdot x_k(\mathbf{s})} \\ &= \frac{1}{2}u_j(\mathbf{s}). \end{aligned}$$

Next, we let L denote the set of projects with more players on them in the optimal solution $\mathbf{s}^*$ than in $\mathbf{s}$. Formally, $L = \{k : x_k(\mathbf{s}^*) > x_k(\mathbf{s})\}$. Let R denote the set of projects with at least as many players on them in $\mathbf{s}$ as $\mathbf{s}^*$. That is, $R = \{k : x_k(\mathbf{s}^*) \leq x_k(\mathbf{s})\}$. Finally, let $X(L), X(R)$ denote the sets of players working on the projects in L and R, respectively.

For every project $k \in L$, we can always have a coalition of players C deviate from their projects in $\mathbf{s}$ to project k to create a solution $\mathbf{s}'$ with $x_k(\mathbf{s}') = x_k(\mathbf{s}^*)$. However, since $\mathbf{s}$ is a strong Nash equilibrium, there exists a player $i_k \in C$ such that $u_{i_k}(\mathbf{s}) \geq \frac{p_k(x_k(\mathbf{s}^*))}{x_k(\mathbf{s}^*)}$. Using the fact that any player's utility is within a factor of 2 of any other player's, we derive that

$$\begin{aligned} \sum_{k \in L} p_k(x_k(\mathbf{s}^*)) &\leq \sum_{k \in L} x_k(\mathbf{s}^*) \cdot u_{i_k}(\mathbf{s}) \\ &\leq \sum_{i \in X(L)} 2 \cdot u_i(\mathbf{s}). \end{aligned}$$

Since p_k is non-decreasing, then clearly

$$\sum_{k \in R} p_k(x_k(\mathbf{s}^*)) \leq \sum_{k \in R} p_k(x_k(\mathbf{s})) = \sum_{i \in X(R)} u_i(\mathbf{s}).$$

Combining these two previous equations allows us to derive that $u(\mathbf{s}^*) \leq 2 \cdot u(\mathbf{s})$. The corresponding lower bound is the same as the one in Theorem 6. □

3.1 Existence and Quality of Contractually Stable Solutions

We prove the main results of this paper in this section. Since strong Nash stable states do not necessarily exist, we add permission to leave to the solution concept to give us *SCNS*. This puts a restriction on the types of deviations that coalitions are allowed to make. Namely, players cannot abandon their old projects if it means the players remaining on those projects will be harmed.

We will use a lexicographic improvement argument [12] to show that a *SCNS* state always exists. In this paper, we will need two different notions of lexicographic ordering, as defined below.

Lexicographic Orderings. Define $v(\mathbf{s})$ to be the vector of player payoffs, i.e., $v(\mathbf{s}) = (u_1(\mathbf{s}), u_2(\mathbf{s}), \ldots, u_n(\mathbf{s}))$. We let $\hat{v}(\mathbf{s})$ denote the sorted vector in which the entries of $v(\mathbf{s})$ are sorted in non-decreasing order, i.e., $\hat{v}_1(\mathbf{s}) \leq \hat{v}_2(\mathbf{s}) \leq \cdots \leq \hat{v}_n(\mathbf{s})$. Then we say that $\mathbf{s}$ is *min-lex* strictly larger than $\mathbf{s}'$ if there exists an index k such that for all $i < k$, $\hat{v}_i(\mathbf{s}) = \hat{v}_i(\mathbf{s}')$ and $\hat{v}_k(\mathbf{s}) > \hat{v}_k(\mathbf{s}')$. We say that a solution $\mathbf{s}$ is the *min-lex maximizer* if there does not exist a solution $\mathbf{s}'$ such that $\mathbf{s}'$ is min-lex strictly larger than $\mathbf{s}$. Similarly, we say that $\mathbf{s}$ is *max-lex* larger than $\mathbf{s}'$ if there exists an index k such that for all $i > k$, $\hat{v}_i(\mathbf{s}) = \hat{v}_i(\mathbf{s}')$ and $\hat{v}_k(\mathbf{s}) > \hat{v}_k(\mathbf{s}')$.

Intuitively, min-lex order ranks solutions by how well the players with the least utility are doing, while max-lex ranks solutions by how well the players with the most utility are doing. The min-lex maximizer maximizes the minimum utility received by any player, while the max-lex maximizer maximizes the the maximum utility received by any player.

We will see that when permission to leave a project must be granted by its participants, any improving deviation results in a min-lex improvement. That is, the minimum utility never becomes worse. Then any state that is a min-lex local maximum is *SCNS*. Thus, our model will always converge to a *SCNS* under best response dynamics.

Theorem 5. *A* SCNS *state always exists.*

Proof. We claim that the min-lex maximizer $\mathbf{s}$ is a *SCNS* state. For a coalition C, let $X(C)$ be the set of players j such that $s_j = s_i$ for some $i \in C$. In other words, $X(C)$ is the set of players who share projects with at least one member of C.

Suppose there exists a coalition C with an improving deviation from state $\mathbf{s}$ to state $\mathbf{s}'$, such that for every player $j \notin C$ who is on the same project as any player $i \in C$ in state s, it must be that $u_j(\mathbf{s}') \geq u_j(\mathbf{s})$. In other words, players who are not part of the coalition will allow the players in C to leave their projects, because they do not suffer due to this coalitional deviation. This would occur, for example, if the project payoff function is a threshold function, and after some players of C leave, the threshold is still satisfied. Thus, for all players $i \in C \cup X(C)$, it must be that $u_i(\mathbf{s}') \geq u_i(\mathbf{s})$.

The utility of some players not in C must have decreased in $\mathbf{s}'$ since $\mathbf{s}$ is the min-lex maximizer. Thus, there must be some project k such that $p_k(x_k(\mathbf{s}))/x_k(\mathbf{s}) > p_k(x_k(\mathbf{s}'))/x_k(\mathbf{s}')$.

Because we have permission to leave, we know for every project that had players deviate from it, the utility of the players that remained on those projects did not decrease. Then only projects that gained players from $\mathbf{s}$ to $\mathbf{s}'$ can have players who lost utility. Since every player in the deviating coalition increased her utility, it must have been the players who were already part of the project in $\mathbf{s}$ whose utility decreased (that is, there must exist at least one project that gained players and was non-empty in $\mathbf{s}$). However, for every player i whose utility decreased, there exists a player $j \in C$ who joined her project such that $u_i(\mathbf{s}') = u_j(\mathbf{s}') > u_j(\mathbf{s})$. Thus, all of the players whose utility decreased from $\mathbf{s}$ to $\mathbf{s}'$ on this project still have $> u_j(\mathbf{s})$ utility, which means that $\mathbf{s}'$ is min-lex larger than $\mathbf{s}$, a contradiction. □

We will now show that there are always good *SCNS* solutions, by demonstrating that the min-lex maximizer is always within a factor of 2 of the optimal solution. Notice that since *SNS* solutions usually do not exist, this result does not follow from Theorem 4.

Theorem 6. *The price of stability with respect to* SCNS *is at most 2, and this bound is tight.*

Proof. To prove the upper bound, we show that the social welfare of the min-lex maximizer is at least $\frac{1}{2}$ that of the optimal solution. Let $\mathbf{s}$ denote the min-lex maximizer. Let $\mathbf{s}^*$ denote the optimal solution. We claim that $u(\mathbf{s}) \geq \frac{1}{2}u(\mathbf{s}^*)$. Suppose, by way of contradiction, that $u(\mathbf{s}) < \frac{1}{2}u(\mathbf{s}^*)$.

Let $l = \min_{i\in[n]} u_i(\mathbf{s})$. Let A denote the set of players in $\mathbf{s}^*$ that receive $\leq l$ utility. Let B denote the set of players in $\mathbf{s}^*$ that receive $> 2l$ utility. Since $\mathbf{s}$ is the min-lex maximizer, we know that A is nonempty.

We claim that B is nonempty. We observe that

$$ln \leq \sum_{i\in[n]} u_i(\mathbf{s}) = u(\mathbf{s}) < \frac{1}{2}u(\mathbf{s}^*),$$

which implies that the average utility of a player in $\mathbf{s}^*$ is strictly larger than $2l$.

We will now reassign players in A to create a new solution such that they receive $> l$ utility using the following algorithm.

We start with $\mathbf{s}^*$. Let M' denote the set of projects that the players in B are assigned to. For each player $i \in A$, we move it to any project in M' where the utility of that player will be $> l$. That is, if a project $k \in M'$ currently has x_k players assigned to it, then we do not assign players to it if $\frac{p_k(x_k+1)}{x_k+1} < l$. We repeat this process until every player in A is assigned to a project in M' or until we cannot assign a player in A to a project in M' such that she receives $> l$ utility. We call the resulting solution $\mathbf{s}'$.

We claim that our algorithm always assigns every player in A to a project in M' such that her utility is $> l$. Suppose not. We begin by observing that

$$\begin{aligned} 2ln < u(\mathbf{s}^*) &= \sum_{i\in A} u_i(\mathbf{s}^*) + \sum_{i\in B} u_i(\mathbf{s}^*) + \sum_{i\notin A\cup B} u_i(\mathbf{s}^*) \\ &\leq |A|l + \sum_{i\in B} u_i(\mathbf{s}^*) + 2\left(n - |A| - |B|\right) l \\ \implies \sum_{i\in B} u_i(\mathbf{s}^*) &= \sum_{k\in M'} p_k\left(x_k(\mathbf{s}^*)\right) > |A|l + 2|B|l. \end{aligned}$$

Since there are players in A that could not be assigned to projects in M', it must be the case that for all $k \in M'$, $\frac{p_k(x_k(\mathbf{s}')+1)}{x_k(\mathbf{s})'+1} < l$, because otherwise we would have assigned that player to that project. Furthermore, since not every player in A was reassigned, $\sum_{k\in M'} x_k(\mathbf{s}') < |A| + |B|$. Combining these facts together allows us to derive that

$$\begin{aligned}|A| + 2|B| &> \sum_{k \in M'} x_k(\mathbf{s}') + |M'| = \sum_{k \in M'} \big(x_k(\mathbf{s}') + 1\big) \\ &> \sum_{k \in M'} \frac{p_k(x_k(\mathbf{s}') + 1)}{l} \geq \sum_{k \in M'} \frac{p_k(x_k(\mathbf{s}^*))}{l} \\ &= \frac{1}{l} \sum_{i \in B} u_i(\mathbf{s}^*) > |A| + 2|B|\end{aligned}$$

which is a contradiction. Then our algorithm terminated with every player in A reassigned to a project in M'. Then every player in $\mathbf{s}'$ has strictly larger than l utility, which contradicts that $\mathbf{s}$ is the min-lex maximizer. We conclude that $\mathbf{s}$ is within a factor of 2 of the optimal solution.

The corresponding lower bound comes from the following example: suppose there are n projects with $p_A(x) = n + \epsilon$ for some $\epsilon > 0$ and $p_k(x) = 1$, for all other projects k. The optimal solution $\mathbf{s}^*$ is assigning one player to each project, and $u(\mathbf{s}^*) = 2n - 1 + \epsilon$. The sole *SCNS* state is every player working on project A, because they are always guaranteed to receive > 1 utility by working on A. Thus, as n goes the infinity and ϵ goes to 0, the price of stability approaches 2. □

3.2 Existence and Quality of Individually Stable Solutions

We now examine what happens when we add permission to join to strong Nash stability. The resulting concept is called *SIS*. We are able to show the our game has a similar lexicographic improvement property with respect to *SIS*, which implies *SIS* states always exist. However, unlike *SCNS* which uses min-lex, *SIS* uses max-lex. In other words, the utility of the players with the most utility always increases with improving deviations.

Theorem 7. *A* SIS *state always exists.*

Proof. We claim that the max-lex maximizer $\mathbf{s}$ is a *SIS* state. For a deviating coalition C, let $X(C)$ be the set of players $j \notin C$ such that $s_j = s'_i$ for some $i \in C$. In other words, $X(C)$ is the set of players who have a player from C joining their projects.

Suppose there exists a coalition C with an improving deviation from state $\mathbf{s}$ to state $\mathbf{s}'$, such that for every player $j \notin C$ who is working on the same project as any player $i \in C$ in state $\mathbf{s}'$, it must be that $u_j(\mathbf{s}') \geq u_j(\mathbf{s})$. In other words, players who are not part of the coalition will allow the players in C to join their projects, because they do not suffer due to this coalitional deviation. Thus, for all players $i \in C \cup X(C)$, it must be that $u_i(\mathbf{s}') \geq u_i(\mathbf{s})$.

The utility of some players not in C must have decreased in $\mathbf{s}'$ since $\mathbf{s}$ is the max-lex maximizer. Thus, there must be some project k such that $p_k(x_k(\mathbf{s}))/x_k(\mathbf{s}) > p_k(x_k(\mathbf{s}'))/x_k(\mathbf{s}')$.

Because we have permission to join, we know that for every project that had players join it, the utility of the players on those projects did not decrease. Then only projects that had some players leave it can have players that lost utility. However, for every player i whose utility decreased this way, there exists a player $j \in C$ with $s_j = s_i$ and $u_j(\mathbf{s}') > u_j(\mathbf{s}) = u_i(\mathbf{s})$. Thus, for any player who lost utility in $\mathbf{s}'$, there exists a player who had the same utility in $\mathbf{s}$ whose utility strictly increased in $\mathbf{s}'$. This implies that $\mathbf{s}'$ is max-lex larger than $\mathbf{s}$, a contradiction. □

We will now see that the quality of *SIS* states is not as good as the quality of *SCNS* states. This is because dynamics that would lead to players deviating to good, high threshold projects together cannot occur. For example, if a player is alone on a project with low utility which gives slightly more than what they would receive on the high value, high threshold project, then they have no incentive to deviate. If there are no other projects to join, the remaining players have no options: they cannot meet the threshold of the high value project, nor can they join the low value project to drive the value down and force the lone player there to deviate to the high value project. Thus, many players are stranded without any projects to join.

Theorem 8. *The price of stability and price of anarchy with respect to* SIS *is at most n, and this bound is tight.*

Proof. Example 1 with the value of project A set to $1+\epsilon$ for $\epsilon > 0$ suffices to show that the price of stability is at least n, except the stable solutions are states in which one player is working on project A and the remaining players are working on B or nothing.

We claim that the social welfare of max-lex maximizer is at least $\frac{1}{n}$ the social welfare of the optimal solution. Let l denote the maximum utility that a player receives in the max-lex maximizer $\mathbf{s}$. Then $u(\mathbf{s}) \geq l$. The maximum utility a player receives in the optimal solution cannot exceed l, since $\mathbf{s}$ is the max-lex maximizer. Thus, $u(\mathbf{s}^*) \leq ln$, which implies that $u(\mathbf{s}) \geq \frac{1}{n}u(\mathbf{s}^*)$. □

3.3 Permission to Join and Leave Results

We finally analyze what happens when we must have both permission to join and permission to leave a project. We observe that the optimal solution becomes stable, but there still exist stable, low quality states since *SCIS* is more general than *SIS*.

Theorem 9. *The optimal solution* $\mathbf{s}^*$ *is* SCIS*.*

Proof. The only deviations permitted are ones in which the utility of every player in the deviating coalition strictly increases, and the utility of every player that is not in the deviating coalition, but is on a project that gains or loses players, does not decrease. The utility of players on projects which neither gain nor lose players remains unchanged, clearly. Thus, the utility of every player must remain the same or increase, which means the value of the resulting solution after the improving deviation would be strictly larger than the value of the optimal solution, a contradiction. □

Corollary 10. *The price of stability with respect to* SCIS *is 1, while the price of anarchy with respect to* SCIS *is n.*

4 Conclusion

Our results show that in profit sharing games when player utilities are not monotone, it is crucial to have some coordinating mechanism like permission to leave a project or permission to join a project. Without such a mechanism no coalitionally stable solutions exist, and even Nash equilibrium can be very bad. Once permission to leave is added through contractual obligations, however, this results in the creation of high-quality solutions which are resilient even to deviations by coalitions. Making sure that people honor their commitments seems to have more effect than excluding people from joining your project, as the former will always have good stable solutions, while the latter may still have high price of stability. Finally, requiring permission to both leave and join a project is perhaps too constraining: a very large number of solutions become stable due to the paucity of allowed deviations, leading to price of stability of 1 but to high price of anarchy.

A natural extension of our model is to allow even more general project payoff functions. In particular, what if we allow the project payoff function to be non-monotone as well? For example, at a certain point, having too many people working on the same project can make communication and organization more difficult, and there may not be enough tasks for everyone, so adding more people may actually *hurt* the project. This captures the idea of the project having a capacity, for example. Due to the extremely powerful stability concept, the centrally optimal solution is still *SCIS* even for this more general case. Unfortunately, even for simple non-monotone project payoff functions, the price of stability with respect to *SCNS* can be horrible. Suppose there is a single project with the payoff function $p_A(x) = 1$ if $x < n$ and $p_A(n) = \epsilon$. Then we need one player to remain unassigned to the project, but that player will always prefer to receive some utility rather than none, so the only *SCNS* solution is having all players work on the project. One extension that seems to avoid such issues is project payoff functions with hard capacities: that is, non-decreasing functions which, after a certain point, suddenly become 0. We conjecture that the price of stability with respect to *SCNS* for this case is still small, and consider this a good future direction.

Acknowledgements

We thank George Karakostas for interesting discussions on the topic of thresholds and capacities. This work was supported in part by NSF awards CCF-0914782, CCF-1101495, CNS-1017932, and CNS-1218374.

References

[1] J. Augustine, N. Chen, E. Elkind, A. Fanelli, N. Gravin, and D. Shiryaev. Dynamics of profit-sharing games. In *IJCAI '11*, pages 37–42.

[2] B. Awerbuch, Y. Azar, A. Epstein, V. S. Mirrkoni, and A. Skopalik. Fast convergence to nearly optimal solutions in potential games. In *EC '08*, pages 264–273.

[3] H. Aziz. Stable marriage and roommate problems with individual-based stability. arXiv: 1204.1628

[4] H. Aziz and F. Brandl. Existence of Stability in Hedonic Coalition Formation Games. In *AAMAS '12*, pages 763–770, 2012.

[5] F. Bloch and E. Diamantoudi. Noncooperative formation of coalitions in hedonic games. *International Journal of Game Theory*, 40(2):263–280, 2010

[6] A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201-230, 2002.

[7] G. Chalkiadakis, E. Elkind, E. Markakis, M. Polukarov, and N. R. Jennings. Cooperative Games with Overlapping Coalitions. *Journal of Artificial Intelligence Research*, 39(1):179–216, 2010.

[8] J. H. Drèze and J. Greenberg. Hedonic Coalitions: Optimality and Stability. *Econometrica*, 48(4):987–1003, 1980.

[9] M. Feldman, L. Lewin-Eytan, and J. S. Naor. Hedonic clustering games. In *SPAA '12*, pages 267–276.

[10] M. X. Goemans, L. Li, V. S. Mirrokni. and M. Thottan. Market sharing games applied to content distribution in ad-hoc networks. In *JSAC '06*, 24(5):1020–1033.

[11] J. Hajduková. Coalition formation games: A survey. *International Game Theory Review*, 8(4):613-641, 2006.

[12] T. Harks, M. Klimm, and R. Moehring. Strong Nash Equilibria in Games with the Lexicographical Improvement Property. In *WINE '09*, pages 463–470.

[13] M. Karakaya. Hedonic coalition formation games: A new stability notion. *Mathematical Social Sciences*, 61(3):157-165, 2011.

[14] J. Kleinberg and S. Oren. Mechanisms for (Mis)allocating Scientific Credit. In *STOC '11*, pages 529–538.

[15] S. Kutten, R. Lavi, A. Trehan. Composition Games for Distributed Systems: the EU Grant Games. In *AAAI '13*, pages 1–16.

[16] V. S. Mirrokni and A. Vetta. Convergence issues in competitive games. In *RANDOM-APPROX '04*, pages 183–194.

[17] Éva Tardos and Tom Wexler. Network formation games. In Noam Nisan, Éva Tardos, Tim Roughgarden, and Vijay Vazirani, editors, *Algorithmic Game Theory,* Chapter 19. Cambridge University Press, 2007.

[18] A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *FOCS '02*, pages 416–425.