Threshold Task Games: Theory, Platform and Experiments

Kobi Gal
Ben-Gurion University of the Negev, Israel &
University of Edinburgh, U.K.
kgal@inf.ed.ac.uk

Ta Duy Nguyen
National University of Singapore
taduy@comp.nus.edu.sg

Quang Nhat Tran
National University of Singapore
e0072418@u.nus.edu

Yair Zick
National University of Singapore
zick@comp.nus.edu.sg

ABSTRACT
Threshold task games (TTGs) are a class of cooperative games in which participants form coalitions to complete tasks associated with different rewards and thresholds for success. We provide efficient algorithms for computing approximately optimal coalition structures in TTGs. We also present non-trivial bounds on the cost of stability for this class. We put our theoretical results to practice; we design a web-based framework which allows human players to interact in a collaborative task-based model. Our analysis of human play in two different countries shows that players succeed in general to form optimal coalition structures, and converge to approximately stable payoff divisions.

ACM Reference Format:

1 INTRODUCTION
A group of players needs to collaborate in order to complete a set of tasks. Their objective is twofold: first, they must form coalitions — disjoint groups of players, each working on a separate task; second, if players are self-interested, they must decide on a reasonable way of dividing revenue from their tasks.

Example 1.1. Alice, Bob and Claire sign up for an online freelance programming website (e.g. fiverr.com). The website currently has two programming tasks: a simple script task $t_1$ that requires a total of 3 hours to complete, and pays $5, and a complex programming task $t_2$ requiring 7 hours of work, paying $15. It is possible to complete a task more than once (so $t_1$ can be assigned to all three players). Alice and Bob can contribute 3 hours each, whereas Claire can contribute 4 hours. Assuming that the task hour load is easily divisible between the players, the (non-unique) best partition of players into work groups would be to assign Alice and Claire to $t_2$, and have Bob work alone on $t_1$. The next step would be to decide how one should divide task revenue amongst the players. One could reasonably argue that Claire should receive a higher share of the revenue than Alice, as she contributed more hours to the task; moreover, in order to complete $t_2$, Claire has to be assigned to it. However, each person could have worked on $t_1$ alone and receive $5 for their efforts. Clearly, finding an appropriate payoff division is not a straightforward task.

Cooperative game theory studies situations in which players form coalitions and share revenue. One can model scenarios like that described in Example 1.1 using a resource-task based model. Player $i$ controls a resource (its weight $w_i$), and a group of players can complete a task (and obtain a reward) if their combined weight exceeds a certain threshold. The value of a group of players (also called a coalition) is the value of the best task it can complete given its resources. In the literature, these are known as threshold task games (TTGs) [11, 22]. Despite their intuitive appeal, there has been little work analyzing solution concepts for TTGs, nor how humans behave when playing them. Our work directly addresses this gap.

1.1 Our Contribution
We study the cost of stability in TTGs: this measures the relative overhead required in order to stabilize the underlying TTG; that is, the amount of additional subsidy required in order to guarantee that there exists some payoff division that assigns each coalition a payoff exceeding its value. In Section 3.1, we present a simple algorithm for computing coalition structures (partitions of the player set) that guarantee at least $\frac{1}{2}$ of the optimal social welfare for TTGs (note that computing an optimal coalition structure for TTGs is NP-complete [17]). Next, in Section 3.2 we provide a tight bound on the cost of stability for TTGs.

Finally, we describe an online platform called Business Cats which provides a negotiation environment for playing TTGs simply and intuitively (Section 4).1 We recruit participants in two countries to play TTG sessions on the Business-Cats platform and analyze the results. Our analysis (Section 5) shows that human players form nearly-optimal coalition structures, and arrive at core-stable payoff divisions. We identify key criteria contributing to successful play; for example, players tend to favor power preserving proposals: players will often refuse proposals that do not assign them a payoff that is commensurate with their value in the game, as measured by their relative weight value. These results provide evidence for the use of empirical negotiation frameworks to support theoretical results and induce good play from people.

1.2 Related Work
Our work examines both theoretical properties and actual gameplay in TTGs. On the theoretical side, Nguyen and Zick [24] bound

---

1The platform is available at http://business-cats.comp.nus.edu.sg/
the cost of stability in TTGs, and provide efficient algorithms for computing the optimal coalition structure. Their analysis requires several limiting assumptions on the structure of player weights and tasks. We do away with these assumptions in Sections 3.1 and 3.2 via careful analysis and novel techniques.

Threshold task games were introduced by Chalkiadakis et al. [11]; their work departs from the classic cooperative game model, allowing players to split resources amongst several tasks. The only work we are aware of that studies a TTG model in the classic cooperative (also known as a Threshold task games, a subclass of threshold task games, we are aware of that studies a TTG model in the classic cooperative via careful analysis and novel techniques.

Given a cooperative game \( G = (N,v) \), a partition of players into coalitions is called a coalition structure. We say that a coalition structure \( CS^* \) is optimal if it maximizes social welfare; that is, \( CS^* \in \arg\max_{CS \in \Pi(N)} \sum_{S \in CS} v(S) \), where \( \Pi(N) \) denotes the set of partitions of \( N \). For brevity, we overload notation and write \( v(CS) = \sum_{S \in CS} v(S) \). We let \( \text{OPT}(G) \) be the value of an optimal coalition structure over \( G \). We refer to the problem of finding an optimal coalition structure (also known as the coalition structure generation problem) as \( \text{OptCS} \). We say that a coalition structure \( CS^* \) is \( \beta \)-optimal for \( G \) if \( v(CS^*) \geq \beta \text{OPT}(G) \). As a solution concept, \( \text{OptCS} \) assumes that a (benevolent) central planner tries to find the best way of partitioning players into groups, ignoring any strategic considerations. If one assumes that players form coalitions in order to receive part of the obtained reward, it is reasonable to assume selfish player behavior. Given a coalition structure \( CS \), an imputation for \( CS \) is a vector \( x \in \mathbb{R}_+^n \) satisfying \( \sum_{i \in S} x_i = v(S) \) for all \( S \in CS \). In other words, players in \( S \in CS \) generate the revenue \( v(S) \), and may divide it amongst themselves in any way they see fit; however, they may not transfer any of the utility generated by \( S \) to non-members of \( S \). The tuple \( (CS,x) \) is called an outcome of \( G \). Let us denote by \( I(G) \) the set of all outcomes for \( G \).

The core is a subset of outcomes in \( I(G) \) defined by,

\[
\text{Core}(G) = \left\{ (CS,x) \in I(G) : x(S) \geq v(S), \forall S \subseteq N \right\}.
\]

The constraints describing the core are also referred to as stability conditions, and outcomes in the core are referred to as stable. Core stability is a natural requirement: if \( (CS,x) \) is not stable, then there exists some coalition \( T \) whose members can generate more revenue than what they are assigned. In other words, \( T \)'s members could ensure a strictly better outcome for themselves if they choose to work together, rather than under the coalitions they were assigned to under \( CS \). It is well-known that if \( (CS,x) \in \text{Core}(G) \), then \( CS \) must be optimal \([12]\); in that respect, the core resolves both the \( \text{OptCS} \) objective and coalitional strategic considerations. Unfortunately, the core may be empty.

**Example 2.1.** Consider a 3 player game where \( v(S) = 0 \) if \(|S| \leq 1 \), and \( |S| = 2 \) otherwise. An optimal coalition structure has a value of 1. Consider any valid imputation \((x_1,x_2,x_3)\), such that \( \sum_{i=1}^3 x_i = 1 \). It must be the case that \( x_1, x_2 \) or \( x_3 \) are strictly greater than 0; with no loss of generality, assume that \( x_1 > 0 \). In that case, \( x_2 + x_3 < 1 = v(2,3) \), and therefore \((x_1,x_2,x_3)\) cannot satisfy the stability conditions. This implies the core is empty. It is easy to verify that any imputation that pays players strictly less than \( \frac{1}{2} \) in total cannot stabilize the game, whereas the payoff \((\frac{1}{4}, \frac{1}{4}, \frac{1}{2})\) is stable. Therefore, paying each player \( \frac{1}{4} \) is a minimal stable payoff division.

We note that this game can be described in the language of weights and thresholds: each player has a weight of 1, and there is a single task. If the total weight of players in a coalition \( S \) is \( \geq 2 \), they can complete the task and obtain a reward of 1; otherwise they coalition has a value of 0.

As Example 2.1 implies, stabilizing a game may require an additional external subsidy. The minimal total payoff needed can be found by solving the following linear program

\[
\min \sum_{i \in N} x_i \quad \text{s.t.} \quad x(S) \geq v(S) \quad \forall S \subseteq N. \tag{1}
\]

Let \( V^* \) be the optimal value of (1); the relative cost of stability (\(\text{CoS} \)) of a game \( G \) is the ratio between \( V^* \) and \( \text{OPT}(G) \); \(\text{CoS}(G) = \frac{V^*}{\text{OPT}(G)} \). We note that if the optimal solution to (1) has a value \( V^* > \text{OPT}(G) \) — i.e. the core is empty — then the corresponding payoff

\[
\text{OPT}(G) \quad \text{or} \quad \text{OPT}(G) \quad \text{as \( G \) is the ratio between \( V^* \) and \( \text{OPT}(G) \).}
\]

Subsequent works (e.g. [10, 23]) utilize the relative definition we use here.
We begin by proving that there exists a contiguous coalition structure $C^*$ in WVGs, and a what follows, we present an efficient algorithm for computing a coalition structure $G$. If that were the case, then the core would not be empty. In other words, stable payoffs that pay a total of $V^*$ do not describe viable payoff divisions.

### 2.2 Threshold Task Games

**Threshold Task Games (TTGs)** are a subclass of cooperative games. In TTGs, each player $i \in N$ has a weight $w_i \in \mathbb{Z}_+$. In addition, we are given a set of tasks $T = \{t_1, \ldots, t_m\}$, where each task $t_j \in T$ has a threshold $q_j \in \mathbb{Z}_+$ and a value $v_j \in \mathbb{Z}_+$. The characteristic function $v$ is $v(S) = \max_{j \in |m|} \{v_j : w(S) \geq q_j\}$. In other words, the value of a coalition $S$ is the value of the best task it can complete, given the resources it controls. It is useful to think of $T$ as a set of task types rather than actual tasks: if a coalition $S$ completes the task $t_j$, other coalitions are free to complete it as well. We assume that if two tasks $t_j$ and $t_k$ have $q_j \leq q_k$ then $v_j \leq v_k$; otherwise no coalition will ever actually complete $t_k$. Finally, we assume that all tasks in $T$ can be completed; that is, for every $t_j \in T$, $q_j \leq w(N)$. Weighted voting games (WVGs) are perhaps the best known example of TTGs; they are simply TTGs with one task with a value of 1. We forgo that assumption, and allow the value of winning coalitions in WVGs to be any positive number.

### 3 APPROXIMATE SOLUTION CONCEPTS IN TTGS

We now present the main theoretical results in this paper: bounds on the cost of stability, and efficient approximation algorithms for the OptCS problem in TTGs. Throughout this section, we assume that $w_1 \geq \cdots \geq w_n$ and that $q_1 \geq \cdots \geq q_m$.

#### 3.1 Approximately Optimal Coalition Structures

Nguyen and Zick [24] present a $\frac{1}{2}$ approximation algorithm for WVGs, and a $< \frac{1}{2}$ approximation for OptCS in general TTGs; in what follows, we present an efficient algorithm for computing a $\frac{1}{2}$ approximation to the OptCS problem in TTGs. The proof of Theorem 3.3 employs a simple dynamic programming algorithm for computing a $\frac{1}{2}$ optimal coalition structure; however, the proof of its correctness (in particular, Lemma 3.1) is non-trivial. We are interested in a special type of coalition structures: we say that a coalition structure $C$ is contiguous if there exist values $i_1 < \cdots < i_T = n$ such that $C$ equals

$$\{i_1, \ldots, i_t\}, \{i_1 + 1, \ldots, i_2\}, \ldots, \{i_{T-1} + 1, \ldots, i_T = n\}.$$  

We begin by proving that there exists a contiguous coalition structure that has value at least $\frac{1}{2}$ of the optimal value (Lemma 3.1). The proof of Theorem 3.3 is completed by a dynamic programming algorithm that finds the best contiguous coalition structure.

**Lemma 3.1.** Given a TTG $G = (N, v)$, there exists a contiguous coalition structure $C^*$ such that $v(C^*) \geq \frac{1}{2} \text{OPT}(G)$.

**Proof.** Let $C^* = \{C_1, \ldots, C_k, L\}$ be an optimal coalition structure where $L$ is the only coalition of value 0. We assume that $C^*$ is an optimal coalition structure that assigns minimal weight to singleton coalitions, and that no players can be transferred to $L$ without decreasing the value of $C^*$. We assume w.l.o.g. that $C_1, \ldots, C_k$ work on tasks $t_1^*, \ldots, t_k^*$ such that $q_1^* \geq \cdots \geq q_k^*$; in particular, this implies that the task values are weakly decreasing as well: $v_1^* \geq \cdots \geq v_k^*$. We now construct a $\frac{1}{2}$-optimal contiguous coalition structure, where players complete tasks in $t_1^*, \ldots, t_k^*$. The idea is to show that under the contiguous coalition structure, all of the odd-numbered tasks are completed. If this is the case, $\frac{1}{2}$ optimality is guaranteed. Suppose that $k$ is even: then $v_1^* \geq \cdots \geq v_k^*$, and we have

$$\sum_{j=1}^{k} 2v_j^* - \sum_{j=1}^{k} v_j^* = \text{OPT}(G) \iff \sum_{j=1}^{k} v_j^* - \frac{1}{2} \text{OPT}(G)$$

When $k$ is odd the above bound holds for the indices $1, \ldots, k-1$, with the addition of the (odd numbered) $k$-th task. We begin with a simple observation: since $C_1, \ldots, C_k$ complete tasks $t_1^*, \ldots, t_k^*$, it must be the case that

$$w(N \setminus L) \geq \sum_{j=1}^{k} q_j^*.$$  

In other words, the combined weight of the coalitions $C_1, \ldots, C_k$ must be, at the very least, as great as the combined thresholds of the tasks that they complete. Furthermore, it is no loss of generality to assume that if $\{i\}$, $\{i'\}$ are singleton coalitions in $C^*$ working on $t_j^*$, respectively, then $q_j^* \geq q_{j'}^*$ implies that $w_i \geq w_i'$. In other words, heavier singletons are assigned to higher-threshold tasks (otherwise, we can simply switch singleton players’ assigned tasks and both tasks can still be completed). We utilize the following proposition.

**Proposition 3.2.** Suppose that $C^*$ is an optimal coalition structure assigning minimal weight to singleton coalitions, where heavier singleton players work on heavier tasks; let $t_j^*$ be the task completed by a singleton $s_r$. Let

$$\tilde{w}_i = \begin{cases} w_i & \text{if } i \notin \{s_1, \ldots, s_q\} \\ q_j^* & \text{if } i = s_r \end{cases}$$

Then for every $i, i' \in N$, $w_i \geq w_i' \iff \tilde{w}_i \geq \tilde{w}_i'$.

**Proof.** Since heavier singletons work on heaver tasks, reducing singletons’ weight to be their task thresholds preserves their relative weight. Let $Q(s_r) = \{i \notin \{s_1, \ldots, s_q\} : w_i \leq w_{s_r}\}$. Suppose that $q_j^* < w_i$ for some $i \in Q(s_r)$; since $i$ can complete $t_j^*$, and $s_r$ has a greater weight than $i$, we can assign $i$ to $t_j^*$, and assign $s_r$ to the coalition that $i$ was in (since $w_{s_r} \geq w_i$, that coalition has a weakly higher value). This results in an optimal coalition structure that puts a strictly lower weight on singletons, a contradiction.

We assume w.l.o.g. that weights are already reduced to $\tilde{w}_i$ as per Proposition 3.2; this reduction still respects the weight order, as argued in Proposition 3.2. Under this weight configuration, all tasks in $C^*$ can still be completed by the same players; furthermore, for all $j \in [k]$ and all $i \in C_j$, $w_i \leq q_j^*$; in particular, if we let $A(j) = \{i \in N : w_i \leq q_j^*\}$, then $C_j \subseteq A(j)$ for all $C_j \in C^*$.

Consider Algorithm 1; its input is the thresholds $q_1^* \geq \cdots \geq q_k^*$ of the tasks completed in an optimal coalition structure, and player
weights \( w_1 \geq \cdots \geq w_n \); its output is a \( \frac{1}{2} \) optimal contiguous coalition structure. Algorithm 1 first splits players’ weight amongst tasks, forming a fractional coalition structure (lines 7–22). Player \( i \) allocates a weight of \( c(j, i) \) to task \( t_j^i \); \( c(j, i) \) equals either the remaining weight that player \( i \) can allocate — this occurs if \( w_i > q_j^i \), or if the task \( t_j^i \) requires more resources than what player \( i \) has — or just enough to complete \( t_j^i \). Due to this condition, we are guaranteed that player \( i \) never assigns positive weight to tasks such that \( w_i > q_j^i \), and in particular, players may not split their weight between more than two tasks. Since we assume that \( C_j \subseteq A(j) \), the following invariant must hold:

\[
\sum_{i \in N} r_i \geq \sum_{j \in [k]} \max\{q_j^i - y_j, 0\}
\]

Note that \( R_j = \max\{q_j^i - y_j, 0\} \) is the weight required to complete task \( t_j^i \); simply put, the invariant states that players always maintain enough weight to complete all tasks. The coalition structure formed during this process is guaranteed to complete all optimal tasks, and is contiguous; however, it is not a valid partition of the players, as some players might commit fractions of their weight to multiple tasks.

In other words, when we terminate in line 22, we complete all tasks \( t_1^i, \ldots, t_k^i \) using contiguous blocks of players; however, some tasks may have fractional player weights assigned to them. However, note that our weight allocation procedure never assigns fractions of a player’s weight to more than two contiguous blocks. In the DeFrac procedure, we ensure that every odd fractional coalition is assigned all players that assigned any weight to it. Since no player splits her weight to more than two coalitions, we are guaranteed that the procedure in lines 25–32 results in a valid coalition structure, where all odd-numbered tasks are completed. Since the value of the odd-numbered tasks is at least \( \frac{1}{2} \) the value of the optimal coalition structure (as argued above), we are done. □

Lemma 3.1 efficiently constructs a \( \frac{1}{2} \)-optimal contiguous coalition structure using Algorithm 1; however, in order to do so, it requires a given non-contiguous optimal coalition structure, which is NP-hard to compute, even when there is a single task [17]. To overcome this issue, we show that it is possible to compute an optimal contiguous coalition structure in polynomial time.

**Theorem 3.3.** There exists an efficient algorithm computing a \( \frac{1}{2} \) optimal coalition structure for any TTG \( G \).

**Proof.** We show that an optimal contiguous coalition structure \( CS^* \) can be found in polynomial time; the value of \( CS^* \) is at least the value of the contiguous coalition structure output by Algorithm 1, and thus by Lemma 3.1 has a value \( \geq \frac{1}{2} OPT(G) \). Let \( X(i, j) \) be the revenue extracted by an optimal contiguous coalition structure using only players in \( \{i, \ldots, n\} \) and tasks in \( \{t_1, \ldots, t_n\} \). Let \( \ell \) be the first index such that \( \{i, \ldots, \ell - 1\} \) can complete \( t_j \), and generate a value of \( v_j \); in this case the optimal revenue is \( v_j + X(\ell, j) \). If \( t_j \) is not completed by a coalition of the form \( \{i, \ldots, \ell - 1\} \), then the contiguity constraints require that no other coalition complete \( t_j \); in this case, the optimal value is \( X(i, j + 1) \). Therefore, \( X(i, j) = \max\{v_j + X(\ell, j), X(i, j + 1)\} \) where \( \ell = \min\{s : w_s + \cdots + w_{s-1} \geq q_j\} \). It is easy to keep track of the actual coalitions formed from the

\[
\text{Algorithm 1 Construct a } \frac{1}{2} \text{ optimal Contiguous CS}
\]

**Input:** thresholds \( q_j^i \geq \cdots \geq q_k^i \); weights \( w_1 \geq \cdots \geq w_n \)

1. for \( i \in N \) do
   2. \( r_i \leftarrow w_i \)
   3. end for

4. for \( j \in [k] \) do
   5. \( y_j \leftarrow 0 \)
   6. end for

7. \( j \leftarrow 1 \); \( i \leftarrow 1 \)
8. while \( i < n \) and \( j < k \) do
   9. if \( r_i < q_j^i - y_j \) or \( w_i > q_j^i \) then
      10. \( c(j, i) \leftarrow r_i \)
   11. else
      12. \( c(j, i) \leftarrow q_j^i - y_j \)
   13. end if
   14. \( r_i \leftarrow r_i - c(j, i) \)
   15. \( y_j \leftarrow y_j + c(j, i) \)
   16. if \( r_i = 0 \) then
      17. \( i \leftarrow i + 1 \)
      18. end if
   19. if \( y_j > q_j^i \) then
      20. \( j \leftarrow j + 1 \)
      21. end if
   22. end while
23. return DeFrac((c(j, i))_{i \in N, j = 1, \ldots, k})

**Function DeFrac((c(j, i))_{i \in N, j = 1, \ldots, k}):**

24. for \( \ell = 1, \ldots, k \) do
   25. if \( \ell \) is odd then
      26. \( S_\ell \leftarrow \{i \in N : c(\ell, i) > 0\} \)
   27. else
      28. \( S_\ell \leftarrow \{i \in N : c(\ell, i) = w_i\} \)
   29. end if
   30. end for
31. \( L \leftarrow N \setminus \bigcup_{\ell=1}^{k} S_\ell \)
32. return \( CS = \{S_\ell : S_\ell \neq \emptyset\} \cup \{L\} \)

33. end function

\( X(i, j) \) values; since \( X(1, 1) \) is the value of the optimal contiguous coalition structure, we are done. □

### 3.2 The Cost of Stability in TTGs

In what follows, we analyze the cost of stability for TTGs. Nguyen and Zick [24] show that \( \text{Cos}(G) \leq 2 \) for TTGs where players are not allowed to form singleton coalitions. We now show how one can forgo this assumption. Given a task \( t_j = (q_j, v_j) \), and a coalition \( S \subseteq N \), let \( \text{OPT}(S, t_j) \) be the optimal revenue of the WVG restricted to \( S \), where players may only complete the task \( t_j \). We begin with the following lemma.

**Lemma 3.4.** Suppose we are given \( k \) disjoint subsets of players \( T_1, \ldots, T_k \) and \( k \) tasks \( t_1, \ldots, t_k \) such that for every \( i \in T_s \), \( w_i \leq q_s \) and \( q_1 < \cdots < q_k \). Assume that for every \( s \in [2..k] \) and every player \( i \in T_s \), let \( x_1 = a_i w_i \) with \( a_i \leq \frac{q_s}{w_i} \). Then

\[
\sum_s OPT(T_s, t_s) \geq \frac{1}{2} \sum_s x(T_s) - \frac{q_k}{2}
\]
We are now ready to prove our main result for this section.

**Proof.** For \( s \in [2..k] \), let the players in \( T_s \) with weight \( w_i \leq q_s \) play a WVG with task \( t_s \). Each winning coalition weighs less than \( 2q_s \), and there is at most one losing coalition. If there is no losing coalition, we have \( w(T_s) \leq 2q_s \frac{\text{OPT}(T_s, t_s)}{v_i} \) or, \( \text{OPT}(T_s, t_s) \geq \frac{v_i}{2q_s} w(T_s) \geq \frac{x(T_s)}{2} \).

If there is one losing coalition, any player \( i \) in this losing coalition has \( w_i \geq q_{s+1} \) and thus \( v(i) \geq v_{s+1} \). Therefore \( w(T_s) \leq 2q_s \frac{\text{OPT}(T_s, t_s)+v_{s+1}}{v_i} + q_s \), or, \( \text{OPT}(T_s, t_s) \geq \frac{v_i}{2q_s} w(T_s) - \frac{v_i}{2} + v_{s+1} \geq \frac{x(T_s)}{2} - \frac{v_i}{2} + \frac{v_{s+1}}{2} \).

Since \( v_{s+1} > v_{s+1} \), we have in any case for \( s \in [2..k] \), \( \text{OPT}(T_s, t_s) \geq \frac{x(T_s)}{2} - \frac{v_i}{2} + \frac{v_{s+1}}{2} \). Similarly, if \( s = 1 \), we have \( \text{OPT}(T_1, t_1) \geq \frac{x(T_1)}{2} - \frac{v_1}{2} \).

Summing up the inequalities, we have

\[
\sum_s \text{OPT}(T_s, t_s) \geq \frac{1}{2} \sum_s x(T_s) - \frac{v_k}{2}.
\]

\[\square\]

We are now ready to prove our main result for this section.

**Theorem 3.5.** For any TTG \( G \), CoS(G) \( \leq 2 \).

**Proof.** We can assume with no loss of generality that \( max: \ w_i \leq \max_j q_j \). Given a payoff division \( \bar{x} \in \mathbb{R}^n \), we call a task \( t = (q, v) \) critical if there exists \( S \subset N \) such that \( w(S) \geq q \) and \( x(S) = v \). A coalition \( S \) whose value is derived by completing a critical task will also be referred to as critical.

Let \( a_1 \) be the minimum number such that the payoff \( x(i) = a_1 w_i \) is stable. There must exist a critical task for \( x_1 \). Let \( t_1 = (q_1, v_1) \) be the critical task with the highest threshold, and let \( S_1 \) be a critical task for \( t_1 \) under \( x_1 \). Since \( x_1(S_1) = a_1 w_1(S_1) \), we have \( a_1 = \frac{x_1(S_1)}{w(S_1)} \leq \frac{a_1}{q_1} \). Let \( T_1 = \{ i \in N : w_i \leq q_1 \} \). Because we assume that \( t_1 \) is the highest-threshold critical task, every critical set must be a subset of \( T_1 \).

Given \( j \) pairs \( (T_j, \alpha_1), \ldots, (T_j, \alpha_j) \), if \( \cup_{i=1}^j T_i \neq N \), we form \( (T_{j+1}, \alpha_{j+1}) \) as follows: Let \( \alpha_{j+1} \) be the minimum number such that the following payoff division is stable:

\[
x_{j+1}(i) = \begin{cases} a_1 w_i & \text{if } i \in T_s \text{ for some } s \leq j, \\ a_{j+1} w_i & \text{otherwise}. \end{cases}
\]

(3)

Let \( t_{j+1} = (q_{j+1}, v_{j+1}) \) be the critical task with the highest threshold under \( x_{j+1} \). Let \( T_{j+1} = \{ i \in N \setminus \cup_{i=1}^j T_i : w_i \leq q_{j+1} \} \).

We note that \( q_1 < \cdots < q_j < q_{j+1} \) and \( a_1 > \cdots > a_j > a_{j+1} \). Let \( S_{j+1} \) be a critical set for \( t_{j+1} \); then \( x_{j+1}(S_{j+1}) = v_{j+1} \geq a_{j+1} w(S) \geq a_{j+1} q_{j+1} \).

Hence we have \( a_{j+1} \leq \frac{v_{j+1}}{q_{j+1}} \).

Suppose that in the end we have \( k \) pairs \( (T_1, \alpha_1), \ldots, (T_k, \alpha_k) \). Consider the payoff \( x(i) = a_i w_i \) if \( i \in T_j \); this payoff division is stable by our construction: we always maintain the stability constraints in the reduction process, as per (3). Under \( \bar{x} \), the critical task with highest threshold is \( t_k \) corresponding to \( T_k \). Let \( S_k \) be a critical set corresponding to \( t_k \). In particular we have \( x(S_k) = v_k \). Let \( T_k' = T_k \setminus S_k \). We can verify that \( T_k', \ldots, T_k' \) satisfy the conditions in Lemma 3.4 Therefore we have

\[
\sum_s \text{OPT}(T_s', t_s) \geq \frac{1}{2} \sum_s x(T_s) - \frac{v_k}{2}
\]

Hence

\[
\text{OPT}(G) \geq v(S_k) + \sum_s \text{OPT}(T_s', t_s)
\]

\[
\geq v_k + \frac{1}{2} \sum_s x(T_s) - \frac{v_k}{2}
\]

\[
= \frac{1}{2} (x(S_k) + v_k) + \frac{1}{2} \sum_s x(T_s) - \frac{v_k}{2} = \frac{1}{2} x(N).
\]

and \( x(S) \leq 2 \text{OPT}(G) \).

\[\square\]

Nguyen and Zick [24] additionally show that for any \( \epsilon > 0 \) there exists some WVG \( G \) for which \( \text{CoS}(G) \geq 2 - \epsilon \), thus the bound in Theorem 3.5 is tight. Bachrach et al. [3] analyze the cost of stability when players are not allowed to form coalition structures; their analysis assumes that all players form a single coalition (\( N \), also called the grand coalition) and generate a revenue of \( \epsilon(N) \). Under this assumption, the cost of stability can be rather high.

**Example 3.6.** Consider an \( n \)-player WVG where \( w_i = 1 \) for all \( i \in N \), and \( q = 1 \). In this case, any (non-empty) coalition is winning, and in particular, \( v(N) = 1 \). If we restrict players to form the grand coalition, then the cost of stability is \( n \). However, if we allow players to form coalition structures then the cost of stability is \( 1 \): players form singleton coalitions and get paid 1 each, a core-stable outcome.

To conclude, when one allows players to form coalition structures (rather than sticking to forming the grand coalition as in [3]), the cost of stability in threshold task games decreases dramatically. In addition, there exist efficient algorithms for computing \( \frac{1}{2} \)-optimal coalition structures. These results indicate that while exact solutions for TTGs may not exist (in the case of the core), or are computationally intractable to compute (in the case of the optimal coalition structure), arriving at good approximate solutions is possible. However, theoretical guarantees say very little about how humans actually play TTGs. We explore this issue in Section 4.

### 4 THE BUSINESS CATS PLATFORM

To investigate human play in TTG, we design and implement a web-based negotiation game called Business Cats\(^3\). Our objective was to create a game environment that (1) closely emulates TTGs (2) is intuitive and fun to play and (3) allows players to strategically reason about their bargaining power. As seen in Figure 1, players

\[\text{Available at http://business-cats.comp.nus.edu.sg/}. \]
play the roles of cats; each cat owns a ladder (corresponding to the weight of the player) that can be used to climb ledges of different heights (corresponding to task thresholds). There are fish of different values on top of the ledges (corresponding to task rewards). Players negotiate to form coalitions, combining their ladders to reach the fish and share the rewards.

Our design closely emulates the TTG formulation as described in Section 2.2; moreover, it allows for intuitive and simple gameplay: this is evidenced from extensive feedback and design sessions with our experimental subjects, both in lab settings and on Amazon Mechanical Turk. Our interface models an iterative bargaining process, allowing players to easily reason about their bargaining power. Indeed, prior work on collaborative interactions among human players uses repeated interaction in order to facilitate strategic gameplay [4, 22]. This is not unreasonable: as Aumann and Maschler [2] describe, stable payoff divisions can be arrived at via a natural process of coalitions offering proposals and counter-proposals. In particular, such a sequence of proposals and counter-proposals converges to an outcome in the bargaining set [15]; this cooperative solution concept contains the core; thus, if the core is not empty, payoff divisions in the core can be arrived at via a simple bargaining process. The bargaining process described by Aumann and Maschler [2] is very similar to the one we implement in the Business Cats platform, with the exception that individual players (rather than coalitions) propose and counter-propose. To conclude, the Business Cats platform satisfies all of our stated desiderata.

The game consists of proposal and response phases. Unlike the turn-based process used by Mash et al., we allow players to initiate a proposal at any time they wish, as long as no other proposal is active; in other words, there is only one ‘live’ proposal at any given point in time. A proposal specifies a prospective task to work on, the coalition members that will work on it, and the proposed share of rewards. In order for a proposal to be submitted, the invited members must be able to complete the proposed task; the total payoff must equal the task value, and the proposer must receive a positive payoff. Upon submitting a proposal, all players see the proposal (and cannot submit additional proposals); players who were offered a positive share of the payoffs must respond to the proposal with an accept/reject⁴. Responders cannot see each other’s response until 30 seconds, the proposal is automatically rejected. If all responders accept the proposal, the coalition is formed, involved players receive the share they accepted, and are removed from the game. If the remaining players can still complete some task, the remaining players can still submit proposals; otherwise the game terminates, and uninvolved players receive a payoff of 0. If some responder rejects the proposal, a cool-down period (30 seconds for proposer, 15 seconds for the rest) is initiated, after which new proposals can be submitted. The cool-down period ensures that players have time to consider the game state, and prevents players from spamming proposals. The proposal history (including responses), and the current pending proposal are visible to every player. The game automatically terminates after 5 minutes; however, the vast majority of instances ended well before the 5-minute mark, taking less than 1 minute on average.

Example 4.1. Figure 2 shows an example run of the Business Cats gameplay of the instance shown in Figure 1. This instance has three players: the blue cat has a weight of 5, the red and green cats have a weight of 20. There are two tasks: \(t_1\) has a threshold \(q_1 = 40\) and value \(v_1 = 8\), \(t_2\) has a threshold \(q_2 = 45\) and value \(v_2 = 10\). The green cat made the first proposal, offering to have all players complete \(t_1\) with the payoff division \((2, 2, 4)\); however, the red cat rejected this offer. The green cat subsequently proposed all players completing \(t_2\) with the payoff division \((2, 2, 6)\); the red cat rejected this offer as well. Finally, the red cat proposed all players completing \(t_2\) with the payoff division \((2, 4, 4)\); all players accepted this proposal, and the game ended.

<table>
<thead>
<tr>
<th>Player:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Proponent</td>
<td>Task</td>
<td>Result</td>
</tr>
<tr>
<td>Initial Cool-off period – 15 sec.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1st</td>
<td>(40, 8)</td>
<td>Proposal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response</td>
</tr>
<tr>
<td>Cool-off period – 15/30 sec.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd</td>
<td>(45, 10)</td>
<td>Proposal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response</td>
</tr>
<tr>
<td>Cool-off period – 15/30 sec.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3rd</td>
<td>(45, 10)</td>
<td>Proposal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response</td>
</tr>
<tr>
<td>No more tasks can be completed – game ends.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: An example run of the Business Cats gameplay.

5 EXPERIMENTS

We generate TTGs instances, varying the number of players (3–5); weights (positive multiples of 5, no greater than 25); number of tasks (1–4); task thresholds (positive multiples of 5, no greater than 100); task values (from the number of players to 10) and whether singleton coalitions were available. We constrain our randomization so that greater task thresholds imply greater task values. During the experiments, games instances were randomly drawn from this pre-generated pool of games.

5.1 Design

We recruit 104 participants (undergraduate students from the National University of Singapore and Ben-Gurion University of the Negev, Israel). IRB approval was obtained from the institutions running the study. Participants were given a detailed tutorial, and were required to pass a comprehension quiz in order to play. Participants play a random series of games with different configurations, and are randomly matched to other players. Each participant receives a show-up fee equivalent to $6.5 US as well as a bonus (between 0 and $6.5) dependent on their total revenue in all the games they play. In total, we collected data from 857 game configurations. We wish to study how people play the game in terms of how they form and respond to proposals. Specifically, we form the following hypotheses: first, that people would generally form “good” coalition structures in terms of total revenue, and coalitional stability, as defined in Sections 3.1 and 3.2; second, that people respond to

---

⁴The proposing player is naturally assumed to accept their own proposal.
offers in a way that respects the power relationship in the game, as defined by their relative weight.

5.2 Analysis

We analyze the games played in terms of the number of optimal and stable coalitions formed, and how people accepted and made offers with respect to the power relationships in the game. We present aggregate results across the two countries as there was no significant difference in player behavior.

Successful Coalitions and Optimality: Tables 1a and 1b show the games collected with respect to number of players and tasks, and the percentage of games for which players reach the optimal coalition structure. These tables show that as the number of players and tasks grow, player performance decreases (as measured by overall welfare). Figure 3 shows a histogram of the welfare achieved by players during gameplay, normalized by OPT(G). Here, the x axis is the ratio between the actual social welfare achieved in the game and the optimal social welfare. Figure 3 shows that in more than 75% of instances, players were able to extract at least 75% of the optimal revenue. In 51.7% of the game instances, players were able to reach the optimal coalition value. On average, players were able to extract 87% of the optimal value.

Not shown in Figure 3 is the fact that in 86% of the cases, participants formed a non-trivial coalition structure — i.e. one containing more than one coalition. When the grand coalition \( N \) was formed, it was the optimal choice 86% of the time. Together these results support the first part of hypothesis 1, showing that players were able to create approximately optimal coalition structures for different number of players and tasks.

Stability Analysis: In this section we constrain the analysis to the games in which there was a non empty core (537 games). To measure the distance from a payoff division \( \hat{x} \) to a stable payoff, we solve the following optimization problem (using a linear programming formulation):

\[
\begin{align*}
\min & \sum_{i \in N} |\alpha_i| \\
\text{s.t.} & \forall S \subseteq N: \sum_{i \in S} x_i + \alpha_i \geq v(S), \\
& \sum_{i \in N} x_i + \alpha_i = \text{OPT}(G)
\end{align*}
\]

where \( \text{OPT}(G) \) is the value of an optimal coalition structure. Given a solution to (4), \( \hat{x}^* \), we let \( \sum_{i \in N} |\alpha_i| / \text{OPT}(G) \) be the distance of \( \hat{x} \) from a stable payoff division.

In general, 32.5% of the coalitions achieved stable outcomes, and more than 50% were at a distance of < 20% from a stable payoff. This result supports the second half of the first hypothesis, showing that our bargaining process was able, on average, to arrive at approximately stable outcomes.

Power Analysis: To study the second hypothesis, we first consider the acceptance ratio for players of different weights in the game. Figure 4 shows the CDF of acceptance rates for the players with the highest and lowest weight in a game. For example, players with the highest weight in a game instance accepted a 40% share of the profits in approximately 20% of instances, whereas the lowest-weight players did so for approximately 70% of instances. In both cases, acceptance rates rise with share percentage, with a “jump” to more than 50% acceptance rate for shares above 30% for lowest-weight players, and approximately 50% for highest-weight players. This confirms that players with higher weights do, in general, understand their relative power in the game.
We also analyze how players respond to offers in the game. We apply the definition of Mash et al. [22] and say an offer \((x_i, x_j)\) made to responders \(i\) and \(j\) with weights \((w_i, w_j)\), is power preserving if \(x_i \leq x_j\) whenever \(w_i \leq w_j\). Overall, 213 out of 1824 (11.7%) were non power preserving, which is considerably lower than the 40% ratio reported for WVGs [22]. In addition, 63.8% of non power preserving offers were rejected, whereas the rejection rate for power preserving offers was just 32.6%. Put together, these results confirm the second hypothesis: people generally make offers that align with responders’ weight, and responders were more likely to accept such offers. We mention that similar results hold when other measures of player power (such as the Shapley or Banzhaf values) are used.

**Active Participation:** While player negotiation tactics are diverse, we do note that by actively proposing, players can significantly improve their prospects. That is, when players initiate proposals, they are more likely to receive better payoffs. Within the 857 games played, there were 1655 instances of (passive) players ending the game without making a proposal, and 1569 instances of (active) players making at least one offer. Active participation directly affects one’s prospective payoff; one simple measure of this property is one’s likelihood to benefit from cooperation. If player \(i\) does form coalitions with others, the highest payoff that they can secure is \(v(\{i\})\). If player \(i\) secures a payoff strictly greater than \(v(\{i\})\) then they strictly benefit from cooperation. Passive players, who make no proposals, strictly benefit from cooperation in only 61% of instances; on the other hand, active players (those who made at least one proposal) strictly benefit from cooperation in 75% of the instances.

The results of our empirical study indicate that human players react efficiently, approximately stable outcomes in TTDGs, and that actual bargaining processes, mirroring theoretical bargaining processes [2], yield approximately stable outcomes.

### 5.3 Case studies

As we show in Section 5.2, Business Cats players were able to achieve overall good performance. However, as one might expect, human play exhibits a high level of complexity. We observe instances where players (a) propose suboptimal coalitions (as was the case in the first proposal in Example 4.1); (b) accept unreasonable proposals, especially later during game progression; or (c) reject reasonable proposals. We provide two examples from our collected data that illustrate some good/bad behaviors.

**Example 5.1.** The game consists of four players with weights \(w_1 = w_2 = w_3 = 15, w_4 = 20\), and two tasks \(t_1 = (35, 4); t_2 = (45, 7)\). Player 4 has a strictly higher weight, and greater influence on the other players. However, player 4 consistently refused to form coalitions where their share was equal to that of others. Eventually, players 1, 2, and 3 agreed to complete \(t_2\), excluding player 4, and receiving a reward of 2, 3, 2, respectively.

In Example 5.1, player 4 had a potentially decisive role, but received zero payoff. Indeed, player weights are not the sole determinant of payoffs. In section 5.2 we show that payoffs are affected by other factors such as player propensity to actively initiate proposals.

**Example 5.2.** The game consists of three players \((w_1 = 25, w_2 = 10, w_3 = 15)\), and two tasks \((t_1 = (25, 3); t_2 = (45, 8))\). In this example, Player 2 offered to form a coalition with players 1 and 3 to complete \(t_2\), offering both players a payoff of 2. Player 1 rejected this offer. Player 2 subsequently offered to form the same coalition offering player 1 a value of 4 and player 3 a value of 2. This coalition was successful.

In Example 5.2, player 1 could form a singleton coalition and complete \(t_1\) with a revenue of 3. However, player 1 accepted an offer to form a coalition with players 2 and 3, once they were offered a better payoff. Indeed, the resulting outcome is core stable (and, in particular, optimal).

### 6 CONCLUSIONS AND FUTURE WORK

In this work, we conduct a thorough theoretical and empirical analysis of threshold task games. We provide an efficient algorithm computing a \(\frac{1}{4}\)-optimal coalition structure, and provide tight bounds on the cost of stability. The Business Cats platform is the first publicly available interactive simulation of a TTG environment; the code and data are currently anonymized (to maintain a proper review process); however, we intend to release them to the research community upon publication. Our analysis indicates that people tend to form coalitions that are nearly stable and optimal, and that players who actively make proposals tend to obtain higher rewards. It is our hope that more platforms like ours will be designed and used to test cooperative game-theoretical models and hypotheses in the wild. This is an important direction to pursue if one is interested in seeing the translation of theoretical notions from cooperative game theory into practical applications.

### ACKNOWLEDGMENTS

The authors would like to thank Pang Hangzhi for her contribution in building the Business Cats platform.

The research work has been supported by an NRF Research Fellowship (no. R-252-000-750-733). The authors would like to thank the anonymous reviewers of AAMAS 2020, whose comments greatly improved the paper.

### REFERENCES


