Robin Hood Reachability Bidding Games

Shaull Almagor
Technion
Haifa, Israel
shaull@technion.ac.il

Guy Avni
University of Haifa
Haifa, Israel
gavni@cs.haifa.ac.il

Neta Dafni
Technion
Haifa, Israel
netad@campus.technion.ac.il

ABSTRACT

Two-player graph games are a fundamental model for reasoning about the interaction of agents. These games are played between two players who move a token along a graph. In bidding games, the players have some monetary budget, and at each step they bid for the privilege of moving the token. Typically, the winner of the bid either pays the loser or the bank, or a combination thereof. We introduce Robin Hood bidding games, where at the beginning of every step the richer player pays the poorer a fixed fraction of the difference of their wealth. After the bid, the winner pays the loser. Intuitively, this captures the setting where a regulating entity prevents the accumulation of wealth to some degree.

We show that the central property of bidding games, namely the existence of a threshold function, is retained in Robin Hood bidding games. We show that finding the threshold can be formulated as a Mixed-Integer Linear Program. Surprisingly, we show that the games are not always determined exactly at the threshold, unlike their standard counterpart.

KEYWORDS

Bidding Games; Discounting; Reachability Games; Wealth Regulation

ACM Reference Format:
Shaull Almagor, Guy Avni, and Neta Dafni. 2025. Robin Hood Reachability Bidding Games. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

A *reachability game* is a 2-player game played on a graph, by placing a token on one of the vertices and moving it along the edges according to some predefined rules, where the goal of Player 1 (denoted 🚹1) is to reach a set of target vertices, and the goal of Player 2 (denoted 🚺2) is to prevent that. Reachability games are fundamental in automated synthesis of systems [15], where a system plays against an environment (e.g., controller synthesis [10], robotic planning [8], network routing [6], etc.).

In *bidding games* [2, 3, 11, 12], each player has a *budget* (a real value in $[0, 1]$, where the sum of the budgets is assumed to be normalized to 1) at any given moment, and the movement of the token in each step is determined by an *auction*, resulting in the higher bidder moving the token. We focus on *Richman* games [11], where the winner of the auction pays their bid to the loser (see e.g., [4] for other bidding mechanisms).

Bidding games are useful for modelling settings where agents compete for some resource (e.g., money, computational resources, etc.) and use these resources to direct the interaction.

A common phenomenon in bidding games is that if one of the players accumulates a high-enough portion of the budget, that player can force the game to reach any desired location. In loose terms, "the rich can do whatever they want". In some settings this phenomenon is desirable, e.g., when modelling the interaction between an attacker and defender of a security system, and the budget is computational resources – nothing prevents either party from hogging resources in order to win. In many other settings, however, the players operate under some regulating entity (e.g., a scheduler in an operating system, or monetary regulation), which prevents the accumulation of excessive wealth in order to achieve some fairness, or to inspire active participation in the game.

A standard means to regulate wealth is to redistribute some of the wealth of the rich to the poor, à la Robin Hood's *steal from the rich and give to the poor* [9, 16]. Note that this is not the same as taxation in that in standard taxation it is not at all clear that taxes go to the poor, nor is it the case that the poor are not taxed.

In this work, we introduce a variant of bidding games called *Robin Hood bidding games* which incorporates wealth regulations. In a Robin Hood bidding game, each auction is preceded by a *wealth redistribution* phase: the richer player pays the poorer player a constant fraction (denoted λ) of the difference between their budgets. The classical model of bidding games then corresponds to $\lambda = 0$. We only consider $0 \leq \lambda < \frac{1}{2}$, as $\lambda \geq \frac{1}{2}$ would mean that the richer player becomes the poorer (or equal, for $\lambda = \frac{1}{2}$), which is of little motivation.

Example 1.1. Consider the game depicted in Figure 1, starting in v_{left}, where the target for 🚹1 is v_1. The wealth redistribution factor is $\lambda = \frac{1}{8}$.

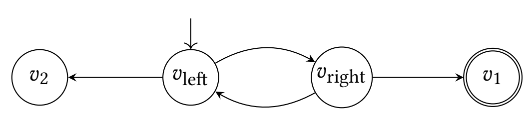


Figure 1: A Robin Hood game. The target for 🚹1 is v_1.

Observe that 🚹1 wins if the play reaches v_1, and loses if it reaches v_2 or oscillates indefinitely between v_{left} and v_{right}. Recall that we assume the sum of budgets of the players is 1. As we show in Section 3, 🚹1 needs a starting budget of at least 0.7 in order to win. We demonstrate why 🚹1 loses when starting with 0.6. At a glance, the optimal strategies for the players induce the play in Figure 3.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

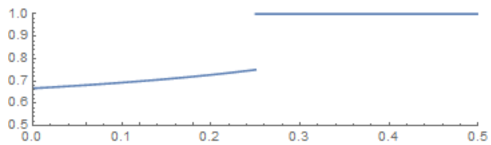


Figure 2: Threshold of v_{left} in Figure 1 as a function of λ.

Starting with budgets $\binom{0.6}{0.4}$, the first step is to apply wealth redistribution (WR, for short). Since $\lambda = \frac{1}{8}$ and $0.6 - 0.4 = 0.2$, then 🧍1 pays 0.025, so the new budgets are $\binom{0.575}{0.425}$. Note that if 🧍1 loses the bidding at v_{left}, she loses the game. Therefore, she must bid at least 0.425 (we use the common assumption that ties are broken in favor of 🧍1). Fortunately, she has sufficient budget for this bid. She moves the token to v_{right} and the new budgets are $\binom{0.15}{0.85}$ (see (1) in Figure 3). Next, WR is applied (with 🧍2 paying). In v_{right} 🧍2 must win the bidding, or he loses the game. To do so, he must bid strictly more than 🧍1. He bids $0.2375 + 0.001 = 0.2385$, wins the bid and moves the token to v_{left} (see (2)). Then, WR leaves 🧍2 as the richer, and he wins the game (3) by out-bidding 🧍1 and moving to v_2.

$$\underbrace{\binom{0.6}{0.4}}_{v_{\text{left}}} \xrightarrow{\text{WR}} \underbrace{\binom{0.575}{0.425}}_{v_{\text{left}}} \xrightarrow{(1)} \underbrace{\binom{0.15}{0.85}}_{v_{\text{right}}} \xrightarrow{\text{WR}} \underbrace{\binom{0.2375}{0.7625}}_{v_{\text{right}}} \xrightarrow{(2)} \underbrace{\binom{0.476}{0.524}}_{v_{\text{left}}} \xrightarrow{\text{WR}} \underbrace{\binom{0.4805}{0.5195}}_{v_{\text{left}}} \xrightarrow{(3)} v_2$$

Figure 3: A losing play for 🧍1.

The central question in the study of bidding games is the existence of a *threshold* function for the game: a function that assigns for each vertex v a value $\mathsf{Th}(v)$, such that if 🧍1 starts with budget more than $\mathsf{Th}(v)$ then she wins, and if she starts with less than $\mathsf{Th}(v)$, she loses. It is shown in [4] that every reachability bidding game has a threshold, and finding its value is in NP ∩ coNP.

We extend the known results regarding the existence of thresholds for reachability bidding games, and show that every Robin Hood reachability bidding games has a threshold (Section 4). We further show that computing this threshold can be done using Mixed Integer Linear Programming (MILP). Additionally, unlike previous works, we discuss what happens when the initial budget equals exactly the threshold (Section 5). We find, surprisingly, that the game might not be determined at the threshold (i.e., neither player has a winning strategy), a behavior that does not occur in standard bidding games where 🧍1 wins ties (nor in general turn based games without concurrent biddings [13]). Apart from the result itself, we believe it is important to draw attention to discussions about the behavior at the threshold. Indeed – it is often the case that optimal strategies work by reaching certain vertices exactly at their threshold. Still in Section 5, we show that given the threshold function and a vertex v, we can decide in polynomial time if the game is undetermined at v with budget $\mathsf{Th}(v)$, and if not – who the winner is.

In addition to these contributions, we observe curious behavior of the threshold function when λ is treated as a parameter. Specifically, in Section 3 we conduct an elaborate analysis of the example in Figure 1 and show that this function might be discontinuous (see Figure 2). We also demonstrate a toolbox for analyzing specific games when λ is a parameter.

Due to space constraints, some proofs and remarks appear only in the full version [1].

2 PRELIMINARIES

A *graph* is $G = (V, E)$ where V is a set of vertices and $E \subseteq V \times V$ is a set of edges. For $v \in V$ we denote by $\Gamma(v) = \{u \mid (v, u) \in E\}$ the set of *neighbors* of v. If $\Gamma(v) = \emptyset$ then v is a *sink*.

A *Robin Hood reachability bidding game* is $\mathcal{G} = \langle G, v_0, x_{\text{init}}, \lambda, T\rangle$, where $G = (V, E)$ is a finite graph, $v_0 \in V$ is an initial vertex, x_{init} is 🧍1's initial budget, $\lambda \in [0, \frac{1}{2})$ is the *wealth redistribution* factor, and $T \subseteq V$ is a set of *target* vertices for 🧍1. We assume for convenience that the vertices in T are sinks. We sometimes omit v_0 and x_{init}, when the discussion is not limited to specific initial vertex and budget.

Intuitively, the game is played between 2 players as follows. At each step, a token is placed on a vertex $v \in V$ (initially v_0), and each of the players has a *budget*, the budgets being described by a vector $\boldsymbol{w} \in [0, 1]^2$ (initially $\begin{pmatrix} x_{\text{init}} \\ 1-x_{\text{init}} \end{pmatrix}$). For clarity, we denote vectors in bold (e.g., $\boldsymbol{v}$).

The game proceeds in steps, each consisting of the following phases:

(1) Wealth Redistribution (abbreviated WR and denoted $\mathfrak{R}$): Each player's budget is updated using the operator

$$\mathfrak{R}(x) = (1 - 2\lambda)x + \lambda$$

(2) Bidding: Each player (concurrently) makes a bid within their budget. The player with the higher bid b wins the bidding (a tie is broken in favor of 🧍1) and pays b to the other player. The budgets are updated accordingly.
(3) Moving: The player who wins the bidding moves the token to a neighbor of v of their choice.

We remark that $\mathfrak{R}$ can be viewed as the linear operator on vectors given by the matrix $\mathfrak{R} = \begin{pmatrix} 1-\lambda & \lambda \\ \lambda & 1-\lambda \end{pmatrix}$. Indeed, in this view we have $\mathfrak{R}\binom{x}{1-x} = \begin{pmatrix} (1-2\lambda)x+\lambda \\ (1-2\lambda)(1-x)+\lambda \end{pmatrix}$. We abuse notation and use either view as convenient.

Formally, a *configuration* is a pair $(v, x) \in (V \times [0, 1])$ where v is the current vertex and x is 🧍1's budget (so the budget for 🧍2 is $1-x$). A *strategy*[1] for 🧍1 is a function $\sigma_1 : V \times [0, 1] \to [0, 1] \times V$ describing for each configuration (v, x) a bid $b \in [0, 1]$ and a neighbor u of v. That is, if $\sigma_1(v, x) = (b, u)$ then we require $b \le \mathfrak{R}(x)$ (as 🧍1's budget during the bidding is $\mathfrak{R}(x)$) and $(v, u) \in E$. A strategy σ_2 for 🧍2 is defined similarly, changing the budget requirement to $b \le \mathfrak{R}(1-x)$, as $\mathfrak{R}(1-x)$ is 🧍2's budget. Given an initial configuration (v_0, x_{init}) and strategies σ_1, σ_2 for 🧍1 and 🧍2 respectively, their induced play, denoted $\mathtt{play}(\sigma_1, \sigma_2, v_0, x_{\text{init}}) = (v_0, x_0), (v_1, x_1), \ldots$, is a (finite or infinite) sequence of configurations defined as per the steps above. Specifically, $x_0 = x_{\text{init}}$, and for every $n \ge 0$, if v_n has no outgoing edges, the sequence terminates. Otherwise, consider $(b_i, u_i) = \sigma_i(v_0, \mathfrak{R}(x_n))$ for $i \in \{1, 2\}$. Then, if $b_1 \ge b_2$ we have

[1]Note that our definition is restricted to *memoryless* strategies. Since we consider Reachability objectives, this is sufficient.

$(v_{n+1}, x_{n+1}) = (u_1, \mathfrak{R}(x_n) - b_1)$ and if $b_1 < b_2$ then $(v_{n+1}, x_{n+1}) = (u_2, \mathfrak{R}(x_n) + b_2)$. If the play reaches T then 👤1 *wins* the play, and otherwise 👤2 wins. For $i \in \{1, 2\}$, a strategy σ_i is *winning for Player i from* (v_n, x_{init}) if for every strategy σ_{1-i} for Player $1-i$, the induced play is winning for Player i.

Observe that the initial budgets of the players sum to 1. Moreover, their sum is maintained by the WR operation and after each bidding. Thus, their sum remains 1 throughout the play.

Given $\langle G, \lambda, T\rangle$, a *threshold function* is a function $\mathsf{Th}\colon V \to [0, 1]$ such that for every (v, x_{init}) the following holds for the game $\langle G, v, x_{\text{init}}, \lambda, \alpha\rangle$:

- If $x_{\text{init}} > \mathsf{Th}(v)$, 👤1 has a winning strategy from (v, x_{init}).
- If $x_{\text{init}} < \mathsf{Th}(v)$, 👤2 has a winning strategy from (v, x_{init}).

We call $\mathsf{Th}(v)$ *1-strong* if an initial budget of exactly $\mathsf{Th}(v)$ wins for 👤1, and *2-strong* if it wins for 👤2. If neither is true (the game is undetermined at the threshold), we call the threshold *weak*.

3 AN ENLIGHTENING EXAMPLE

In this section we expand the discussion regarding the game in Figure 1. This serves to gain familiarity with the model, but in addition – enables us to prove certain interesting properties of Robin-Hood games, namely Corollary 3.1. Moreover, the tools we present may be of use in analyzing other games (c.f., Remark 3.2). We remark that this section is quite technical, and the rest of our results do not depend on it. Thus, algebraically-averse readers can safely skip to Section 4 if they so choose.

We are interested in finding a threshold for v_{left} specifically, as a function of λ. We denote this threshold by $\tau(\lambda)$.

For $\lambda = 0$, it is shown in [4] that $\frac{2}{3}$ is a 1-strong threshold. Intuitively, this budget allows 👤1 to ensure the play oscillates between v_{left} and v_{right}, with her budget increasing with each move to v_{left}, until it is high enough to allow her to win two consecutive biddings and reach v_1. In the following, we show the existence of a threshold $\tau(\lambda)$ for every λ.

A play can end by either reaching v_1 or v_2 (and 👤1 wins or loses respectively), or oscillate infinitely between $v_{\text{left}}, v_{\text{right}}$, in which case 👤1 loses. Regardless, a play can be described by a finite or infinite sequence of iterations, each comprising four phases: (1) WR in v_{left}; (2) bidding in v_{left}; (3) WR in v_{right}; and (4) bidding in v_{right}. This sequence can be finite and be followed by a move into v_2 or v_1 which ends the play, or be infinite. For the n'th iteration, we denote the budget vectors as follows. In v_{left}: $\boldsymbol{w}^{(n)}_{\text{left}}$ and $\boldsymbol{w}^{(n)}_{\text{wr,left}}$, before and after WR, respectively; and similarly $\boldsymbol{w}^{(n)}_{\text{right}}$ and $\boldsymbol{w}^{(n)}_{\text{wr,right}}$ for v_{right}. We also denote the first entries of these vectors, that is, 👤1's budgets, by $x^{(n)}_{\text{left}}, x^{(n)}_{\text{wr,left}}, x^{(n)}_{\text{right}}, x^{(n)}_{\text{wr,right}}$.

3.1 Alternative Tie Breaking

It is useful to first consider the case that ties are not always broken in favor of 👤1, but instead in favor of 👤1 when in v_{left} and in favor of 👤2 when in v_{right} (resulting in moving from v_{left} to v_{right} and vice-versa).

We describe optimal strategies σ_1, σ_2 for the players and their resulting play. At any iteration n, when in v_{left} and about to bid, if $x^{(n)}_{\text{wr,left}} < \frac{1}{2}$ then 👤2 can bid $\frac{1}{2} + \epsilon$ and move to v_2, so 👤1 instantly loses. For this reason we only consider initial budgets of at least $\frac{1}{2}$ for this example. Note that having a budget of at least $\frac{1}{2}$ before or after WR is equivalent, and the same for a strict inequality. That is, $x > \frac{1}{2}$ if and only if $\mathfrak{R}(x) > \frac{1}{2}$ and $\mathfrak{R}(\frac{1}{2}) = \frac{1}{2}$. If 👤1 has at least $\frac{1}{2}$, she can win the bidding by matching 👤2's budget, namely bidding $1 - x^{(n)}_{\text{wr,left}}$. Moreover, she must do so or lose the game. 👤1 then pays that amount to 👤2, and moves to v_{right}. Similarly, when in v_{right} and about to bid, 👤1 wins instantly if $x^{(n)}_{\text{wr,right}} > \frac{1}{2}$, and otherwise 👤2 pays $x^{(n)}_{\text{wr,right}}$ to 👤1 and moves to v_{left}.

Starting with $\boldsymbol{w}^{(0)}_{\text{left}} = \binom{x_{\text{init}}}{1-x_{\text{init}}}$, the sequence of vectors (while oscillating between v_{left} and v_{right}) can therefore be described with the following steps (viewing $\mathfrak{R}$ as a matrix):

(Step 1) $\boldsymbol{w}^{(n)}_{\text{wr,left}} = \mathfrak{R}\boldsymbol{w}^{(n)}_{\text{left}}$ (Step 2) $\boldsymbol{w}^{(n)}_{\text{right}} = B_{\text{left}}\boldsymbol{w}^{(n)}_{\text{wr,left}}$

(Step 3) $\boldsymbol{w}^{(n)}_{\text{wr,right}} = \mathfrak{R}\boldsymbol{w}^{(n)}_{\text{right}}$ (Step 4) $\boldsymbol{w}^{(n+1)}_{\text{left}} = B_{\text{right}}\boldsymbol{w}^{(n)}_{\text{wr,right}}$

Where

$$B_{\text{left}} = \begin{pmatrix} 1 & -1 \\ 0 & 2 \end{pmatrix} \qquad B_{\text{right}} = \begin{pmatrix} 2 & 0 \\ -1 & 1 \end{pmatrix} \qquad \mathfrak{R} = \begin{pmatrix} 1-\lambda & \lambda \\ \lambda & 1-\lambda \end{pmatrix}$$

Overall, $\boldsymbol{w}^{(n+1)}_{\text{left}} = M\boldsymbol{w}^{(n)}_{\text{left}}$ for the matrix $M = B_{\text{right}}\mathfrak{R}B_{\text{left}}\mathfrak{R}$, which depends on λ. The matrix M has two eigenvalues:

- $E^{\mathtt{val}}_1 = 1$, with the (normalized) eigenvector $\boldsymbol{E}^{\mathbf{vec}}_1 = \begin{pmatrix} \frac{2\lambda-2}{4\lambda-3} \\ \frac{2\lambda-1}{4\lambda-3} \end{pmatrix}$.
- $E^{\mathtt{val}}_2 = 16\lambda^2 - 16\lambda + 4$, with the eigenvector $\boldsymbol{E}^{\mathbf{vec}}_2 = \binom{-1}{1}$.

Recall that the budget vectors belong to the affine subspace $W = \{(x, y) \in \mathbb{R}^2 \mid x + y = 1\}$, which is invariant under M. Every vector $\boldsymbol{w} \in W$ can be written as a linear combination of the form $\boldsymbol{w} = \boldsymbol{E}^{\mathbf{vec}}_1 + c\boldsymbol{E}^{\mathbf{vec}}_2$ for some $c \in \mathbb{R}$. Indeed, since $\boldsymbol{E}^{\mathbf{vec}}_1 \in W$ and $\boldsymbol{E}^{\mathbf{vec}}_2$ is the slope of $x + y = 1$, we have that $W = \{\boldsymbol{E}^{\mathbf{vec}}_1 + c\boldsymbol{E}^{\mathbf{vec}}_2 \mid c \in \mathbb{R}\}$.

We then have $M\boldsymbol{w} = \boldsymbol{E}^{\mathbf{vec}}_1 + cE^{\mathtt{val}}_2\boldsymbol{E}^{\mathbf{vec}}_2 = \boldsymbol{E}^{\mathbf{vec}}_1 + E^{\mathtt{val}}_2(\boldsymbol{w} - \boldsymbol{E}^{\mathbf{vec}}_1)$. Projecting this on the first coordinate and denoting $x_{\text{fix}} = \frac{2\lambda-2}{4\lambda-3}$, for every n we have $x^{(n+1)}_{\text{left}} = \mathtt{next}\left(x^{(n)}_{\text{left}}\right)$, where $\mathtt{next}(x) = x_{\text{fix}} + E^{\mathtt{val}}_2(x - x_{\text{fix}})$, and overall

$$x^{(n)}_{\text{left}} = x_{\text{fix}} + \left(E^{\mathtt{val}}_2\right)^n (x_{\text{init}} - x_{\text{fix}})$$

If the condition $x^{(n)}_{\text{wr,right}} > \frac{1}{2}$ is met for some n (and the play does not end before that), 👤1 wins. Similarly, if $x^{(n)}_{\text{wr,left}} < \frac{1}{2}$ (equivalently, $x^{(n)}_{\text{left}} < \frac{1}{2}$) then 👤1 loses. If neither of these occurs for any n then the play is infinite and 👤1 loses.

It is convenient to phrase the win condition $x^{(n)}_{\text{wr,right}} > \frac{1}{2}$ as a condition on $x^{(n)}_{\text{left}}$, which allows the analysis to focus on $x^{(n)}_{\text{left}}$ only. Note that $x^{(n)}_{\text{wr,right}}$ is an injective function of $x^{(n)}_{\text{left}}$, obtained by projecting the operator $\mathfrak{R}B_{\text{left}}\mathfrak{R}$ on the first coordinate. Its reverse function, denoted $f^{\text{rev}}(x)$, satisfies $x^{(n)}_{\text{left}} = f^{\text{rev}}\left(x^{(n)}_{\text{wr,right}}\right)$. It is easy to verify that $f^{\text{rev}}(x) = \frac{4\lambda^2-5\lambda+x+1}{2(2\lambda-1)^2}$, and it is increasing for all x. Therefore, 👤1 wins in the n'th iteration if and only if $x^{(n)}_{\text{left}} > f^{\text{rev}}(\frac{1}{2})$. It follows that 👤1 wins the play if and only if there exists n such that $x^{(n)}_{\text{left}} > f^{\text{rev}}(\frac{1}{2})$ and $x^{(n')}_{\text{left}} \geq \frac{1}{2}$ for all $n' < n$.

We now split the analysis according to the value of λ. Specifically, according to whether E_2^{val} is the dominant eigenvalue, i.e., whether $\lambda < \frac{1}{4}$ or $\lambda \geq \frac{1}{4}$.

The case $\frac{1}{4} \leq \lambda < \frac{1}{2}$. In this case, $f^{\mathrm{rev}}(\frac{1}{2}) \geq 1$ and therefore 👤1's winning condition is never met, and she loses for every initial budget.

The case $0 < \lambda < \frac{1}{4}$. In this case, we have $E_2^{\mathsf{val}} > 1$. Recall that $\boldsymbol{E}_1^{\mathsf{vec}} = \begin{pmatrix} x_{\mathrm{fix}} \\ 1-x_{\mathrm{fix}} \end{pmatrix}$ is an eigenvector with eigenvalue 1, and we claim that it is the threshold vector, that is, $x_{\mathrm{fix}} = \frac{2\lambda-2}{4\lambda-3}$ is a (2-strong) threshold. Note that x_{fix} is increasing and continuous in λ, and equals $\frac{2}{3}$ for $\lambda = 0$ and $\frac{3}{4}$ for $\lambda = \frac{1}{4}$.

Recall that $x_{\mathrm{left}}^{(n)} = x_{\mathrm{fix}} + \left(E_2^{\mathsf{val}}\right)^n (x_{\mathrm{init}} - x_{\mathrm{fix}})$, and 👤1 wins if and only if this value goes above $f^{\mathrm{rev}}(\frac{1}{2})$ (and does not go below $\frac{1}{2}$ before that). We now have $\frac{1}{2} < \frac{2}{3} \leq x_{\mathrm{fix}} < f^{\mathrm{rev}}\left(\frac{1}{2}\right) < 1$. It follows that:

- For $x_{\mathrm{init}} = x_{\mathrm{fix}}$, we have $x_{\mathrm{left}}^{(n)} \equiv x_{\mathrm{fix}} < f^{\mathrm{rev}}(\frac{1}{2})$ for all n, and so 👤1 loses.
- For $x_{\mathrm{init}} > x_{\mathrm{fix}}$, the sequence $x_{\mathrm{left}}^{(n)}$ increases unboundedly, eventually above $f^{\mathrm{rev}}(\frac{1}{2})$, at which point 👤1 wins.
- For $x_{\mathrm{init}} < x_{\mathrm{fix}}$, the sequence decreases unboundedly, eventually below $\frac{1}{2}$, at which point 👤1 loses.

We remark that this eigenvalue analysis also gives us insight regarding the behavior of the budget vectors throughout the play (see the full version for the details).

3.2 Correct Tie Breaking

We now return to the original tie breaking mechanism, where all ties are broken in favor of 👤1. The analysis is therefore changed to reflect the fact that in v_{right}, 👤2 must strictly outbid 👤1 in order to control the movement of the token.

In particular, When in v_{right} and $x_{\mathrm{wr,right}}^{(n)} < \frac{1}{2}$, 👤2 must bid strictly more than $x_{\mathrm{wr,right}}^{(n)}$, i.e. $x_{\mathrm{wr,right}}^{(n)} + \epsilon_n$ for some $\epsilon_n > 0$. Note that thus far, our analysis considered fixed optimal strategies, and hence a single play. Now, however, each strategy of 👤2 may choose different values for the ϵ_n's, thus inducing multiple plays that need to be analyzed.

The full analysis can be found in the full version. As it turns out, the values of the threshold remain unchanged, but the result of the game when starting with initial budget of exactly the threshold changes in favor of 👤1 for the case $0 \leq \lambda \leq \frac{1}{4}$.

In conclusion, we have that the thresholds are the following:

- For $0 \leq \lambda < \frac{1}{4}$: $\tau(\lambda) = \frac{2\lambda-2}{4\lambda-3}$ (increasing from $\frac{2}{3}$ to $\frac{3}{4}$, 1-strong threshold)
- For $\lambda = \frac{1}{4}$: $\tau(\lambda) = 1$ (1-strong threshold)
- For $\frac{1}{4} < \lambda < \frac{1}{2}$: $\tau(\lambda) = 1$ (2-strong threshold)

Note that $\tau(\lambda)$, depicted in Figure 2, is discontinuous at $\lambda = \frac{1}{4}$, which gives us the following.

Corollary 3.1. *There exists a game $\mathcal{G}$ and vertex v such that the threshold function of v is discontinuous as a function of λ.*

Remark 3.1. *The behavior of τ in this example can be given an economic interpretation: after a certain threshold (namely $\lambda = \frac{1}{4}$), the threshold (suddenly) becomes equivalent to that of any $\frac{1}{4} \leq \lambda < \frac{1}{2}$. This suggests that beyond a certain threshold, it no longer helps anyone to impose more tax. Naturally this does not fully extend to real-life economics, but it is a curiosity nonetheless.*

Remark 3.2. *The analysis carried out in this section crucially depends on obtaining a characterization of the play resulting from optimal strategies as a linear dynamical system. Since analyzing such systems is notoriously difficult, especially in high dimensions [14], automating this analysis algorithmically seems out of reach. Nonetheless, the tools we develop in this section may be used in other examples. Specifically, starting with alternative tie-breaking to avoid sinks, and using the dominant eigenvalues as a guide to the long-run behavior.*

4 EXISTENCE OF THRESHOLDS

The analysis in Section 3 demonstrates the threshold function for a specific game, but does not give a general technique for computing thresholds, nor shows that they always exist. In this section we present our main result, namely that every game has a threshold function.

The section is organized as follows. In Section 4.1 we describe an invariant dubbed *the average property* which gives a lower bound on the threshold. In Section 4.2 we restrict the discussion to games played on directed acyclic graphs (DAGs), in which case there exists a unique function satisfying the average property, and it constitutes a threshold. In Section 4.3 we turn to general graphs, and show the existence of a threshold by a reduction to the setting of DAGs. In Section 4.4 we show that the threshold satisfies the average property, and is obtained as the point-wise maximum over all functions satisfying this property. Finally, in Section 4.5 we use this characterization to compute the threshold using mixed-integer linear programming (MILP).

4.1 The Average Property

Consider a game $\mathcal{G} = \langle G, \lambda, T\rangle$, and assume it has a threshold $\mathsf{Th}\colon V \to [0,1]$. For a sink $v \in T$, it holds that $\mathsf{Th}(v) = 0$, since starting at the target, 👤1 instantly wins for any initial budget. For a sink $v \notin T$, we have $\mathsf{Th}(v) = 1$, since 👤1 instantly loses for any initial budget.

For a non-sink $v \in V$, $\mathsf{Th}(v)$ relates to the minimum and maximum values of Th among v's neighbors as follows. Intuitively, if 👤1 wins the bidding at v, it is optimal for her to choose the next vertex to have a minimal threshold. Similarly, 👤2 would choose the maximal threshold. As we show in this section, the budget needed *during the bidding phase* in v for 👤1 to win the game turns out to be exactly the average of these two values, and so the threshold (which is the budget before WR) is obtained by applying the reverse map $\mathfrak{R}^{-1}$ on this average (where $\mathfrak{R}^{-1}(x) = \frac{x-\lambda}{1-2\lambda}$). We remark that similar average properties typically arise in bidding games [2].

Definition 4.1. Let $G = (V, E)$ be a graph, $T \subseteq V$ a subset of the sinks, and $f\colon V \to [0,1]$. For a non-sink $v \in V$, let

$$v^+ = \operatorname*{argmax}_{u\in\Gamma(v)} f(u) \quad \text{and} \quad v^- = \operatorname*{argmin}_{u\in\Gamma(v)} f(u)$$

(we choose arbitrarily if the extrema are not unique). Let

$$f_{\mathrm{avg}}(v) = \frac{f(v^+) + f(v^-)}{2} \quad \text{and} \quad f_{\mathrm{pre}}(v) = \mathfrak{R}^{-1}(f_{\mathrm{avg}}(v))$$

We say that f satisfies the *average property* if for every $v \in V$:

- If v is a sink, then $f(v) = \begin{cases} 0 & v \in T \\ 1 & v \notin T \end{cases}$.
- If v is not a sink, then $f(v) = \begin{cases} 0 & f_{\text{pre}}(v) < 0 \\ f_{\text{pre}}(v) & f_{\text{pre}}(v) \in [0,1] \\ 1 & f_{\text{pre}}(v) > 1 \end{cases}$

  Equivalently, $f(v) = \max(\min(f_{\text{pre}}(v), 1), 0)$

Note that the "artificial" introduction of 1 and 0 as limits makes sense both because f's range is $[0,1]$, but also semantically. For example, if $f_{\text{pre}}(v) < 0$, it can be viewed as "P1 does not even need 0 in order to win, so 0 is certainly enough". Formally, since $\mathfrak{R}(x)$ is an increasing function, we have in this case that $\mathfrak{R}(0) > \mathfrak{R}(f_{\text{pre}}(v)) = f_{\text{avg}}(v)$. Therefore, starting with budget 0, during the first bidding P1 has more than $f_{\text{avg}}(v)$. In the following we show this is enough for P1 to win. The analysis for $f_{\text{pre}}(v) > 1$ is similar.

The following lemma provides a clear motivation for the average property and its relation to thresholds. Intuitively, it states that if f satisfies the average property, then starting from $x_{\text{init}} > f(v_0)$, P1 can guarantee that the current budget always remains above $f(v)$, and dually for P2 staying below $f(v_0)$. At first glance, this may seem to suggest that every function satisfying the average property is a threshold. This, however, is generally false: there may be multiple such functions, while the threshold is clearly unique.

Lemma 4.2. *Let $\mathcal{G} = \langle G, v_0, \lambda, T\rangle$ be a game, and let f be a function satisfying the average property. There exist strategies σ_1, σ_2 for Players 1 and 2, respectively, such that the following holds.*

(1) If $x_{\text{init}} > f(v_0)$ then for every strategy σ_2' of P2, every configuration (v, x) in $\texttt{play}(\sigma_1, \sigma_2', v_0, x_{\text{init}})$ satisfies $x > f(v)$.
(2) If $x_{\text{init}} < f(v_0)$ then for every strategy σ_1' of P1, every configuration (v, x) in $\texttt{play}(\sigma_1', \sigma_2, v_0, x_{\text{init}})$ satisfies $x < f(v)$.

Proof. Assume $x_{\text{init}} > f(v_0)$. We describe σ_1 inductively. Let (v, x) be a configuration such that v is not a sink and $x > f(v)$. Then $f(v) < 1$, and in particular $f(v) \geq f_{\text{pre}}(v)$ (as either $f(v) = f_{\text{pre}}(v)$ or $f(v) = 0 > f_{\text{pre}}(v)$). After WR, P1 has budget

$$\mathfrak{R}(x) > \mathfrak{R}(f(v)) \geq \mathfrak{R}(f_{\text{pre}}(v)) = f_{\text{avg}}(v)$$

We now describe the bid of P1. Let

$$f_{\text{diff}}(v) = \frac{f(v^+) - f(v^-)}{2}$$

P1 bids $f_{\text{diff}}(v)$ (note that $f_{\text{diff}}(v) \leq f_{\text{avg}}(v)$). If she wins the bidding, she moves to v^-, at which point her budget is x' satisfying

$$x' = \mathfrak{R}(x) - f_{\text{diff}}(v) > f_{\text{avg}}(v) - f_{\text{diff}}(v) = f(v^-)$$

As desired. Dually, if P1 loses the bidding, then P2 bid more than $f_{\text{diff}}(v)$, so P1's new budget is x' satisfying

$$x' > \mathfrak{R}(x) + f_{\text{diff}}(v) > f_{\text{avg}}(v) + f_{\text{diff}}(v) = f(v^+)$$

and the invariant is maintained regardless of the vertex P2 chooses to move to (since v^+ has maximal value of f among the neighbors).

Next assume $x_{\text{init}} < f(v_0)$, we describe σ_2. Given (v, x) such that $x < f(v)$, we have in particular $f(v) > 0$, and so $f(v) \leq f_{\text{pre}}(v)$. After WR, P1 has budget $\mathfrak{R}(x) < \mathfrak{R}(f(v)) \leq \mathfrak{R}(f_{\text{pre}}(v)) = f_{\text{avg}}(v) = f(v^+) - f_{\text{diff}}(v) \leq 1 - f_{\text{diff}}(v)$. Thus, P2 has budget at least $f_{\text{diff}}(v)$. He bids that amount, and upon winning moves to v^+. If P1 wins the bidding, her new budget is $x' < f_{\text{avg}}(v) - f_{\text{diff}}(v) = f(v^-)$ and so the invariant is maintained in the next vertex regardless of P1's choice (since v^- has minimal value of f among the neighbors). If P1 loses the bidding, she has less then $f(v^+)$, as desired. □

Consider a function f satisfying the average property, and recall that P1 wins in a play if and only if a vertex $v \in T$ is reached. Such vertices satisfy $f(v) = 0$ by Definition 4.1. Thus, if every configuration (v, x) in a play satisfies the invariant $x < f(v)$, then it cannot hold that $f(v) = 0$ for any vertex in that play, i.e. the play does not reach T. Using Lemma 4.2 we then have the following.

Corollary 4.3. *Let $\mathcal{G}$ be a reachability game, and let f be a function satisfying the average property. If $x_{\text{init}} < f(v_0)$, then P2 has a winning strategy.*

Note that a dual argument for P1 winning when $x_{\text{init}} > f(v_0)$ fails, as her losing does not require reaching $f(v) = 1$.

4.2 Games Played on Directed Acyclic Graphs

In this section we restrict attention to directed acyclic graphs (DAGs), and show the existence of thresholds in this case. We rely on these results in Section 4.3 where we generalize to all graphs. Intuitively, for a DAG, the restrictions given by the average property uniquely define a function inductively from the sinks backwards, and By Lemma 4.2 it constitutes a threshold. Hence, we have the following result (the complete proof can be found in the full version):

Lemma 4.4. *Consider a game $\mathcal{G} = \langle G, \lambda, T\rangle$ such that G is a DAG, then $\mathcal{G}$ has a unique function that satisfies the average property, and it is a threshold function.*

Example 4.5. We illustrate Lemma 4.4 in Figure 4 with $\lambda = \frac{1}{6}$ (the names of the vertices are relevant for Section 4.3). Observe that for $\lambda = \frac{1}{6}$ we have $\mathfrak{R}^{-1}(x) = \frac{3}{2}x - \frac{1}{4}$. Thus, in $(v_1, 1)$ we have $\mathsf{Th}_{\text{avg}}(v_1, 1) = \frac{1}{2}$, so $\mathsf{Th}(v_1, 1) = \mathfrak{R}^{-1}(\frac{1}{2}) = \frac{1}{2}$. In $(v_0, 1)$ we have $\mathsf{Th}_{\text{avg}}(v_0, 1) = 1$, so $\mathsf{Th}_{\text{pre}} = \mathfrak{R}^{-1}(1) = \frac{5}{4} > 1$, and therefore $\mathsf{Th}(v_0, 1) = 1$. Finally, $\mathsf{Th}_{\text{avg}}(v_0, 0) = \frac{3}{4}$ so $\mathsf{Th}(v_0, 0) = \mathfrak{R}^{-1}(\frac{3}{4}) = \frac{7}{8}$.

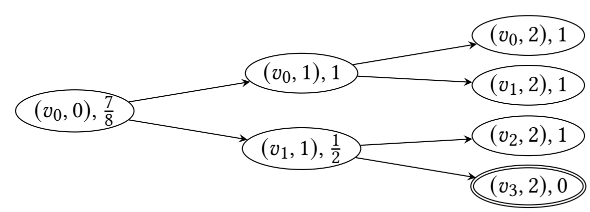


Figure 4: A DAG (in this case, a tree) and the unique values satisfying the average property, for $n = 2$ and $\lambda = \frac{1}{6}$.

4.3 Games Played on General Graphs

We are now ready for our main result.

Theorem 4.6. *Every game $\mathcal{G} = \langle G, \lambda, T\rangle$ has a threshold function which satisfies the average property.*

Proof. We construct a function $\mathsf{Th}(v)$, show that it satisfies the average property, and that it constitutes a threshold function.

The first step is to reduce the setting to that of DAGs. Denote $G = (V, E)$ and let $n \in \mathbb{N}$. We turn to define a game that is played on a DAG (specifically, the unravelling of G for n steps) and, intuitively, corresponds to the objective of winning in $\mathcal{G}$ within at most n steps. Consider the DAG $\mathcal{D}_n = (V \times \{0, \ldots, n\}, E_n)$, where $E_n = \{(u_i, v_{i+1}) \mid (u, v) \in E, 0 \leq i \leq n-1\}$. As an example, the underlying graph of the game depicted in Figure 5 yields the DAG depicted in Figure 4 for $n = 2$.

Next, we define the game $\mathcal{G}_n = \langle \mathcal{D}_n, \lambda, T \times \{0, \ldots, n\} \rangle$. By Lemma 4.4 we have that $\mathcal{G}_n$ has a threshold function Th_n.

Let $v \in V$. We consider the relation between $\mathsf{Th}_n((v, 0))$ and $\mathsf{Th}_{n+1}((v, 0))$. Assume 1 has a winning strategy σ_1 for $\mathcal{G}_n$, starting in configuration $((v, 0), x_{\text{init}})$. Following σ_1 ensures, in particular, that the play does not reach $(V \times \{n\}) \setminus (T \times \{n\})$, as those are sinks not belonging to the target. Observe that $\mathcal{D}_{n+1}$ is obtained from $\mathcal{D}_n$ by (possibly) adding outgoing edges only from $(V \times \{n\}) \setminus (T \times \{n\})$. The strategy σ_1 therefore wins, starting in $((v, 0), x_{\text{init}})$, in $\mathcal{G}_{n+1}$ as well. Intuitively, winning $\mathcal{G}$ in at most n steps particularly wins it in at most $n + 1$ steps. Thus, $\mathsf{Th}_n((v, 0)) \geq \mathsf{Th}_{n+1}((v, 0))$, i.e., the sequence $\{\mathsf{Th}_n((v, 0))\}_{n=0}^{\infty}$ is non-increasing. This sequence is also bounded from below by 0, and therefore converges. We define the threshold Th for $\mathcal{G}$ as the pointwise-limit

$$\mathsf{Th}(v) = \lim_{n \to \infty} \mathsf{Th}_n((v, 0))$$

Next, we prove that Th satisfies the average property. For a sink $v \in T$, we have $\mathsf{Th}_n((v, 0)) = 0$ for all n (since $(v, 0) \in T \times \{0, \ldots, n\}$), and so $\mathsf{Th}(v) = 0$ as needed. For a sink $v \notin T$, we have $\mathsf{Th}_n(v) = 1$ for all n (since $(v, 0)$ is a sink in $\mathcal{D}_n$ and does not belong to the target), and so $\mathsf{Th}(v) = 1$ as needed.

For the following, fix a non-sink v. We need to show (as per Definition 4.1) that

$$\mathsf{Th}(v) = \max\left(\min\left(\mathfrak{R}^{-1}\left(\frac{\mathsf{Th}(v^+) + \mathsf{Th}(v^-)}{2}\right), 1\right), 0\right) \qquad (1)$$

Note that $(v, 0)$ is not a sink in $\mathcal{D}_n$ for all $n \geq 1$. For every $n \geq 1$, let

$$v_n^+ = \operatorname*{argmax}_{u \in \Gamma(v)} \mathsf{Th}_n(u) \quad \text{and} \quad v_n^- = \operatorname*{argmin}_{u \in \Gamma(v)} \mathsf{Th}_n(u)$$

For every $u \in \Gamma(v)$, consider the sub-DAG of $\mathcal{D}_n$ starting in $(u, 1)$. Observe that this sub-DAG is isomorphic to the sub-DAG of $\mathcal{D}_{n-1}$ starting in $(u, 0)$ (with the difference only being the indices of the levels). It follows that $\mathsf{Th}_n((u, 1)) = \mathsf{Th}_{n-1}((u, 0))$. By the average property for $\mathcal{G}_n$, we have that $\mathsf{Th}_n((v_{,0}))$ is:

$$\max\left(\min\left(\mathfrak{R}^{-1}\left(\frac{\mathsf{Th}_{n-1}((v_n^+, 0)) + \mathsf{Th}_{n-1}((v_n^-, 0))}{2}\right), 1\right), 0\right)$$

Note that this is continuous as a function of $\mathsf{Th}_{n-1}((v_n^+, 0))$ and $\mathsf{Th}_{n-1}((v_n^-, 0))$. In order to show Equation (1), it is therefore enough to show

$$\lim_{n \to \infty} \mathsf{Th}_{n-1}((v_n^+, 0)) = \mathsf{Th}(v^+) \qquad (2)$$

$$\lim_{n \to \infty} \mathsf{Th}_{n-1}((v_n^-, 0)) = \mathsf{Th}(v^-) \qquad (3)$$

We show that Equation (2) holds, and the proof for Equation (3) is analogous. Note that v_n^+ might be a different vertex for each n, and so the left hand side of Equation (2) does not describe the limit of thresholds for a single vertex. Intuitively, however, there is a set of vertices that appear infinitely often in this limit whose corresponding limits are all equal, which enables us to conclude the claim (see the full version for the explicit argument).

Finally, we show that Th is a threshold function for $\mathcal{G}$. Assume $\mathcal{G}$ starts in (v, x_{init}).

If $x_{\text{init}} > \mathsf{Th}(v)$ then there exists n such that $x_{\text{init}} > \mathsf{Th}_n(v)$, meaning 1 has a strategy that wins in at most n steps, and in particular wins in $\mathcal{G}$.

If $x_{\text{init}} < \mathsf{Th}(v)$, then since Th satisfies the threshold property, it follows from Corollary 4.3 that 2 has a winning strategy. □

4.4 Characterization of Thresholds in Terms of the Average Property

Recall that for games played on DAGs, the threshold is the unique function satisfying the average property, and it can be computed inductively from the sinks. For general games with $\lambda = 0$, it is known [4] that there is still a unique function satisfying the average property, and that finding the threshold is in NP ∩ coNP (and in P for graphs with out-degree 2).

In stark contrast, this uniqueness no longer holds in Robin Hood games for $\lambda > 0$, as we now demonstrate. Consider the game in Figure 5, with $\lambda = \frac{1}{4}$. It can be verified that the numbers on the vertices are the thresholds. However, for any $t \in [0, 1]$, the function defined by $f(v_0) = t$ and $f(v) = \mathsf{Th}(v)$ for $v \neq v_0$ satisfies the average property. Indeed, this can be easily checked for $v \neq v_0$ (and also follows from Lemma 4.4, since from without v_0 the game is a DAG). For v_0, note that $\mathfrak{R}^{-1}(x) = 2(x - \frac{1}{4}) = 2x - \frac{1}{2}$, so

$$f_{\text{pre}}(v_0) = \mathfrak{R}^{-1}(f_{\text{avg}}(v_0)) = \mathfrak{R}^{-1}\left(\frac{t + \frac{1}{2}}{2}\right) = t = f(v_0)$$

and since $f(v_0) \in [0, 1]$, the average property holds.

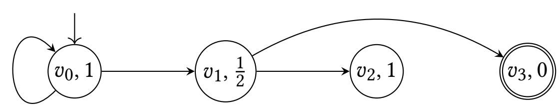


Figure 5: Infinitely many functions satisfying the average property, for $\lambda = \frac{1}{4}$. The numbers are the thresholds.

As mentioned above, the threshold in Figure 5 satisfies $\mathsf{Th}(v_0) = 1$, which coincides with the maximal value of t. As it turns out, this is not a coincidence.

Theorem 4.7. *Consider a game $\mathcal{G}$. The threshold* Th *is the pointwise maximum over the functions satisfying the average property.*

Proof. Let A be the set of functions satisfying the average property and let $m(v) = \sup_{f \in A} f(v)$. Since Th satisfies the average property, we have $\mathsf{Th}(v) \leq m(v)$. We additionally have $f(v) \leq \mathsf{Th}(v)$ for every $f \in A$; indeed, assume by way of contradiction that $\mathsf{Th}(v) < f(v)$, and let $\mathsf{Th}(v) < x_{\text{init}} < f(v)$. By the threshold definition, 1 has a winning strategy, but by Corollary 4.3, 2 also does. Therefore, $m(v) \leq \mathsf{Th}(v)$, and overall $\mathsf{Th}(v) = m(v)$. □

4.5 MILP for Computing the Threshold

While Theorem 4.6 shows the existence of thresholds, it uses a limit and is therefore not constructive. However, using Theorem 4.7 we can algorithmically compute the threshold using mixed-integer linear programming (MILP).

Theorem 4.8. *Given a game $\mathcal{G}$, we can efficiently construct a MILP instance $\mathcal{I}$ whose solution is the threshold function for $\mathcal{G}$.*

Proof. Consider a game $\mathcal{G}$ with vertices $V = v_1, \ldots, v_m$. The average property can be readily expressed as a set of constraints on the variables $\mathsf{Th}(v)$ for every $v \in V$ containing linear and min/max expressions (as per Definition 4.1). Observe that this complies with MILP, since e.g., the expression $\min\{x_1, x_2\}$ can be removed by introducing a new variable X, a variable $b \in \{0, 1\}$ and the following constraints:

$$X \leq x_1 \wedge X \leq x_2 \wedge X \geq x_1 - 2M(1-b) \wedge X \geq x_2 - 2Mb$$

Where M is a bound such that $|x_1|, |x_2| < M$. In our case, $M = \mathfrak{R}^{-1}(1)$ is such a bound. Indeed, the first two constraints ensure $X \leq \min\{x_1, x_2\}$, and the latter two ensure that either $X \geq x_1$ (if $b = 1$) or $X \geq x_2$ (if $b = 0$). The choice of M ensures that the two latter constraints are satisfiable.

Finally, we maximize the objective $\sum_{v \in V} \mathsf{Th}(v)$. The solution then equals the threshold by Theorem 4.7. □

Example 4.9. We demonstrate the construction of the MILP for the game depicted in Figure 5. For each v_i, we use a variable v_i to represent the value $f(v_i)$. The resulting MILP is in Table 1.

	maximize $v_1 + v_2 + v_3 + v_4$ subject to:
(C1)	$v_2 = 1 \wedge v_3 = 0$
(C2)	$v_1^- \leq v_2 \wedge v_1^- \leq v_3$ $v_1^- \geq v_2 - 2M(1 - b_1^-) \wedge v_1^- \geq v_3 - 2Mb_1^-$
(C3)	$-v_1^+ \leq -v_2 \wedge -v_1^+ \leq -v_3$ $-v_1^+ \geq -v_2 + 2M(1 - b_1^+) \wedge -v_1^+ \geq -v_3 + 2Mb_1^+$
(C4)	$v_1' \leq \mathfrak{R}^{-1}\left(\frac{v_1^+ + v_1^-}{2}\right) \wedge v_1' \leq 1$ $v_1' \geq \mathfrak{R}^{-1}\left(\frac{v_1^+ + v_1^-}{2}\right) - 2M(1 - b_1') \wedge v_1' \geq 1 - 2Mb_1'$
(C5)	$-v_1 \leq -v_1' \wedge -v_1 \leq 0$ $-v_1 \geq -v_1' + 2M(1 - b_1) \wedge -v_1 \geq 2Mb_1$
(C6)	$v_0^- \leq v_0 \wedge v_0^- \leq v_1$ $v_0^- \geq v_0 - 2M(1 - b_0^-) \wedge v_0^- \geq v_1 - 2Mb_0^-$ $-v_0^+ \leq -v_0 \wedge -v_0^+ \leq -v_1$ $-v_0^+ \geq -v_0 + 2M(1 - b_0^+) \wedge -v_0^+ \geq -v_1 + 2Mb_0^+$ $v_0' \leq \mathfrak{R}^{-1}\left(\frac{v_0^+ + v_0^-}{2}\right) \wedge v_0' \leq 1$ $v_0' \geq \mathfrak{R}^{-1}\left(\frac{v_0^+ + v_0^-}{2}\right) - 2M(1 - b_0') \wedge v_0' \geq 1 - 2Mb_0'$ $-v_0 \leq -v_0' \wedge -v_0 \leq 0$ $-v_0 \geq -v_0' + 2M(1 - b_0) \wedge -v_0 \geq 2Mb_0$
(C7)	$b_1^-, b_1^+, b_1', b_1, b_0^-, b_0^+, b_0', b_0 \in \{0, 1\}$

Table 1: The MILP for Example 4.9

(C1) expresses the average property requirement for the sinks. For the non-sink v_1, the requirement of the average property involves v_1^-, which can attain the value of either v_2 or v_3, and therefore introduces a new variable[2] and the constraints in (C2). The binary variable b_1^- gets the value 1 if $v_2 = \min(v_2, v_3)$, and 0 otherwise. The constraint (C3) similarly set $v_1^+ = -\min(-v_2, -v_3)$. Next, (C4) serves to express $v_1 = \min\left(\mathfrak{R}^{-1}\left(\frac{v_1^+ + v_1^-}{2}\right), 1\right)$, and (C5) deals with the maximum with 0. (C6) describes the analogue of (C2)–(C5) for v_0. Finally, (C7) puts the integer constraints on the various b_i variables.

We remark that in particular, we can solve a decision version of finding the threshold (i.e., comparing it to a given bound) in NP. Additionally, it is not hard to construct games for which the set of functions that satisfy the average property is not convex. This suggests (but does not prove) that formulating the problem in Linear Programming, or indeed finding a polynomial time solution, is unlikely.

5 INITIAL BUDGET OF EXACTLY THE THRESHOLD

Recall from Section 2 that the definition of a threshold function only considers the behavior strictly above or below the threshold. In this section, we study the behavior exactly at the threshold. We present two results. First, surprisingly, we show that when starting with exactly the threshold the game can be *undetermined* (i.e., no player has a winning strategy). Next, we show how to decide in polynomial time whether the threshold in each vertex is 1-strong, 2-strong, or weak.

Example 5.1. Consider the game in Figure 6a with $\lambda = \frac{1}{8}$. Here v_1 stands for an initial vertex of some game with a 1-strong threshold of $\frac{7}{18}$ (e.g., the game depicted in Figure 6b). The only solution to the average property then gives $\mathsf{Th}(v_0) = \frac{5}{18}$. We claim that starting with $x_{\text{init}} = \mathsf{Th}(v_0)$, the game is undetermined. Indeed, fix a strategy for **1**, we show that **2** can counter it and win. We remind that for a vertex v we have $\mathsf{Th}_{\text{diff}}(v) = \frac{\mathsf{Th}(v^+) - \mathsf{Th}(v^-)}{2}$, and that the reachability objective allows us to restrict the discussion to memoryless strategies. In v_0:

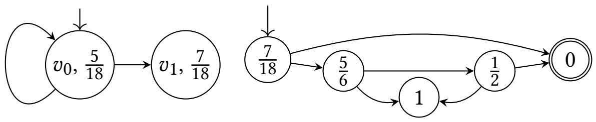
- If **1** bids at least $\frac{1}{18} = \mathsf{Th}_{\text{diff}}(v_0)$, **2** bids 0. **1** wins the bidding with a resulting budget of at most $\mathsf{Th}(v_0^-) = \mathsf{Th}(v_0) = \frac{5}{18}$. If she moves to v_1, she has strictly less than the threshold $\mathsf{Th}(v_1) = \frac{7}{18}$ and she loses. If she stays in v_0 indefinitely, she also loses.
- If **1** bids $\frac{1}{18} - \epsilon$ for $\epsilon > 0$, **2** bids $\frac{1}{18} - \frac{\epsilon}{2}$. He wins the bidding and moves to v_1, where **1**'s budget is $\mathsf{Th}(v_1) - \frac{\epsilon}{2}$, so **1** loses again.

Conversely, fix a strategy for **2**, and we show that **1** can counter it and win. In v_0:

- If **2** bids at least $\frac{1}{18} = \mathsf{Th}_{\text{diff}}(v_0)$, **1** bids 0. **2** then wins the bidding, and **1** has at least $\mathsf{Th}(v_0^+) = \mathsf{Th}(v_1) = \frac{7}{18}$. If **2** stays in v_0 then **1** has strictly more than the threshold and she wins. If **2** moves to v_1, **1** still wins since $\mathsf{Th}(v_1) = \frac{7}{18}$ is 1-strong.
- If **2** bids $\frac{1}{18} - \epsilon$ for $\epsilon > 0$, **1** bids $\frac{1}{18} - \frac{\epsilon}{2}$. She wins the bidding and stays in v_0, but increases her budget to $\mathsf{Th}(v_0) + \frac{\epsilon}{2}$, allowing her to win.

We conclude that no player wins from v_0 with $x_{\text{init}} = \frac{5}{18}$.

[2]Observe that for games with out-degree at most 2, we can exploit the symmetry between v_i^- and v_i^+, in that we do not need to encode which is the minimal and which is the maximal.



(a) Undetermined game **(b) A game with threshold $\frac{7}{18}$ for $\lambda = \frac{1}{8}$**

Figure 6: A game undetermined at the threshold for $\lambda = \frac{1}{8}$

We now turn to show that it is decidable in polynomial time whether the threshold at a vertex is 1-strong / 2-strong / weak. We start with some intuition based on Example 5.1. There, the reason 🚹1 does not have a winning strategy is that upon bidding $\mathsf{Th}_{\text{diff}}(v_0)$ and winning, she must go to $v_0^- = v_0$, so no progress is made. Conversely, she must bid $\mathsf{Th}_{\text{diff}}(v)$ in order not to lose immediately. The observant reader may see that if 🚹1 can follow a path consisting only of transitions of the form (v, v^-) until reaching T, then she can guarantee winning from the threshold. Indeed, we show that if there is such a path, then the threshold is 1-strong. The latter can be easily checked using graph reachability.

Next, if this condition fails, we need to distinguish between a 2-strong and a weak threshold. We show that this distinction can be made by reduction to a 2-player *turn based* reachability game, which are solvable in polynomial time. In the following we consider $\lambda > 0$. The case of $\lambda = 0$ is easier (see Proposition 5.3).

Theorem 5.2. *Let $\mathcal{G}$ be a game with $\lambda > 0$. Given the threshold function, it is possible to decide in polynomial time whether the threshold is 1-strong, 2-strong, or weak, for each vertex.*

Proof sketch. We partition the vertices V as follows. $V_1 = \{v \in V \mid \mathsf{Th}_{\text{pre}}(v) < 0\}$, $V_2 = \{v \in V \mid \mathsf{Th}_{\text{pre}}(v) > 1\}$ and $V_{\text{mid}} = V \setminus (V_1 \cup V_2)$. By adding self loops on sinks, we have that $T \subseteq V_1$. We show that for $v \in V_1$, after the first WR, 🚹1 has budget strictly above the threshold. Thus, $\mathsf{Th}(v) = 0$ and it is 1-strong. Analogously, for $v \in V_2$, after WR 🚹1 has budget strictly below the threshold, so $\mathsf{Th}(v) = 1$ and it is 2-strong. It remains to consider V_{mid}.

We obtain a graph G_{good} from G by keeping only edges of the form (u, u^-) for every u^- that minimizes the threshold among the neighbors of u. For $v \in V_{\text{mid}}$, we show that if V_1 is reachable from v in G_{good} then from budget $\mathsf{Th}(v)$ 🚹1 can either move along a path to V_1, maintaining a budget of exactly the threshold, or she might lose the bidding and gain budget strictly greater than the threshold. In either case she wins, so the threshold is 1-strong.

If V_1 is not reachable from v in G_{good}, we first show that $\mathsf{Th}(v)$ is not 1-strong. Intuitively, this is because in order to leave V_{mid}, the play either enters V_2 with at most the threshold (where 🚺2 wins since the threshold is 2-strong), or enters V_1 strictly below the threshold (and again 🚺2 wins).

Then, in order to decide whether $\mathsf{Th}(v)$ is 2-strong or weak, we show that if 🚺2 plays optimally, then his bids are fixed at each vertex, and maintain the invariant that the budget is always equal to the threshold. This allows us to restrict the set of configurations that need to be considered to a finite set, and reduces the game to a turn-based reachability game, where the goal of 🚹1 is to reach V_1, and V_2 is a sink. We prove that if 🚺2 wins this turn-based game then $\mathsf{Th}(v)$ is 2-strong, and otherwise it is weak.

Since deciding the winner in a turn-based reachability game can be done in polynomial time, we conclude the proof. The details appear in the full version. □

For $\lambda = 0$ things are simpler: if T is reachable then the threshold is 1-strong, and otherwise it is trivially 2-strong. This reduces the decision problem to graph reachability (see proof in the full version).

Proposition 5.3. *Let $\mathcal{G}$ be a game with $\lambda = 0$. If T is reachable from v then* $\mathsf{Th}(v)$ *is 1-strong.*

Finally, we remark that games on DAGs are determined at the threshold (see Proposition 4.6 in the full version).

6 DISCUSSION AND FUTURE RESEARCH

Robin Hood bidding games incorporate a regulating entity into bidding games, allowing the simulation of realistic settings that cannot be captured with standard bidding games. The introduction of wealth redistribution comes at a technical cost: analyzing the behavior of the game becomes much more involved (c.f., Section 3). Nonetheless, we are able to show that the model retains the nice property of having a threshold function, albeit the game may become undetermined exactly at the threshold.

Apart from establishing the theoretical and algorithmic foundations of this setting, our results shed light on various properties of the optimal strategies of the players. In particular, we show that when starting above the threshold, 🚹1 intuitively plays on an unwinding of the game to a DAG in order to reach T. However, when playing exactly from the threshold, 🚹1 needs a path to T along which she can afford to win all the biddings.

A natural future direction is to extend our framework to infinite-duration games, e.g., Büchi and parity games. For standard bidding games, winning in infinite-duration games reduces to an analysis of strongly connected components [3]. In the Robin Hood case, however, this no longer applies, suggesting that showing the existence of a threshold function is nontrivial, if there even exists one.

One view of WR is as a mechanism for changing the budgets of players outside the bidding phase. A different mechanism for achieving this is, introduced in [5], designates special vertices where agents can *charge* their budget. From an economical perspective, this can be seen as vertices where a player performs some "work" and receives a salary. Thus, combining the models would allow us to specify a richer economic dynamics. It would be interesting to examine whether this model retains nice algorithmic properties.

A different research direction concerns viewing wealth redistribution as a form of *discounting* [7]: in discounting, the value of future rewards decreases exponentially with time, according to some discount factor λ. Wealth redistribution can then be viewed as a discounting factor on the difference of the budgets of the agents. It may be of interest to consider other models of discounting in bidding games, e.g., a reward model for the agents where future-budgets are worth less than current budgets.

ACKNOWLEDGMENTS

This research was supported by the ISRAEL SCIENCE FOUNDATION (grants No. 989/22 and No. 1679/21).

REFERENCES

[1] Shaull Almagor, Guy Avni, and Neta Dafni. 2024. Robin Hood Reachability Bidding Games. arXiv:2412.17718 [cs.GT] https://arxiv.org/abs/2412.17718

[2] Guy Avni and Thomas A Henzinger. 2020. A survey of bidding games on graphs. In *31st International Conference on Concurrency Theory*, Vol. 171.

[3] Guy Avni, Thomas A Henzinger, and Ventsislav Chonev. 2019. Infinite-duration bidding games. *Journal of the ACM (JACM)* 66, 4 (2019), 1–29.

[4] Guy Avni, Thomas A Henzinger, and Đorđe Žikelić. 2021. Bidding mechanisms in graph games. *J. Comput. System Sci.* 119 (2021), 133–144.

[5] Guy Avni, Ehsan Kafshdar Goharshady, Thomas A Henzinger, and Kaushik Mallik. 2024. Bidding Games with Charging. In *35th International Conference on Concurrency Theory (CONCUR 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[6] Thomas Brihaye, Gilles Geeraerts, Marion Hallet, Benjamin Monmege, and Bruno Quoitin. 2019. Dynamics on Games: Simulation-Based Techniques and Applications to Routing. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik.

[7] John Broome. 1994. Discounting the future. *Philosophy & Public Affairs* 23, 2 (1994), 128–156.

[8] Georgios E Fainekos, Savvas G Loizou, and George J Pappas. 2006. Translating temporal logic to controller specifications. In *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 899–904.

[9] William Franko, Caroline J Tolbert, and Christopher Witko. 2013. Inequality, self-interest, and public support for "Robin Hood" tax policies. *Political research quarterly* 66, 4 (2013), 923–937.

[10] Antoine Girard. 2012. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica* 48, 5 (2012), 947–953.

[11] Andrew J Lazarus, Daniel E Loeb, James Propp, and Daniel Ullman. 1996. Richman Games. In *Games of No Chance*. Cambridge University Press, 439–449.

[12] Andrew J. Lazarus, Daniel E. Loeb, James G. Propp, Walter R. Stromquist, and Daniel H. Ullman. 1999. Combinatorial Games under Auction Play. *Games and Economic Behavior* 27, 2 (May 1999), 229–264.

[13] Donald A Martin. 1975. Borel determinacy. *Annals of Mathematics* 102, 2 (1975), 363–371.

[14] Joël Ouaknine and James Worrell. 2012. Decision problems for linear recurrence sequences. In *International Workshop on Reachability Problems*. Springer, 21–28.

[15] Amir Pnueli and Roni Rosner. 1989. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 179–190.

[16] Amit Poddar, Jeff Foreman, Syagnik Sy Banerjee, and Pam Scholder Ellen. 2012. Exploring the Robin Hood effect: Moral profiteering motives for purchasing counterfeit products. *Journal of Business Research* 65, 10 (2012), 1500–1506.