Tug of War Game

William Gasarch and Nick Sovich and Paul Zimand

October 6, 2009

Abstract

To be written later.

1 Introduction

Combinatorial games under auction play, introduced by Lazarus, Loeb, Propp, Stromquist, and Ullmann [1, 2], are two-party games in which the player who makes the next move is determined by the highest bid in an auction. The game itself consists of moving a token along the edges of a given graph until the token reaches some designated node. In those papers the bids are secret, and bidding ties are solved as follows. The first time the players bid the same amount, the first player moves, the second time the second player moves, and from there on the players alternate in moving for tying bids. In this work, we study the case in which the bids are not secret. At each step, Player 1 bids first, and Player 2 bids knowing the first player's bid. The player who bids more gets to move the token. In the case of a tie, the first player moves the token. Thus, Player 1 has the advantage that ties go in her favor, and Player 2 has the advantage that he knows Player 1's bid before bidding himself.

We restrict the analysis to a very simple graph (namely, a simple path). This game is called Tug of War (TGW). We consider two variants of this game. In the first (called money-to-the-adversary TGW), the winning bid goes to the other player, and in the second (called money-to-the-bank TGW), the winning bid goes to a bank (and thus disappears from the game).

Summary of results so far (August 23, 2009).

1. Section 2.1: Money to the adversary, continuous game, advantage ≥ 0, $\Rightarrow$ Left wins.
2. Section 2.2: Money to the adversary, continuous game, advantage < 0, $\Rightarrow$ Right wins.
3. Section 3.1: Money to the adversary, discrete game, advantage ≥ 0, $\Rightarrow$ Left wins.
4. Section 3.2: Money to the adversary, discrete game, advantage $<< 0$, $\Rightarrow$ Right does not necessarily win.
5. Section 4.2 and Section 4.3: Money to the bank, continuous game. Found algorithm to determine how large the ratio $\frac{d_L}{d_R}$ has to be for Left to win or draw. Found algorithm to determine how small the ratio $\frac{d_L}{d_R}$ has to be for Right to win or draw. Found strategy for each player.
6. Section 4.4: Money to the bank, continuous. Analyzed the game when token starts in the middle. If $d_L > d_R$ then left wins, if $d_L < d_R$, then Right wins.

7. Section 5: Money to the bank, discrete. Described algorithm for finding winner in any configuration.

1.1 The Game

1. The board is a line with n nodes marked off on it. The nodes are labeled in order $0, 1, 2, \ldots, n$; thus, the left and right end nodes are labeled 0 and n, respectively.

   $\circ - - - - - \circ - - - - - - \circ - - - - - - \circ - - - - - - \circ - - - - - - \circ - - - - - - \circ$

   0 $\qquad\qquad\qquad\qquad$ p $\qquad\qquad\qquad\qquad$ n

   The players are Left and Right. The game starts with the token at some arbitrary node p, $0 < p < n$. Of particular interest is the case when in there are an odd number of nodes and the token starts at the middle node.

   It is Left's goal to get the token to the 0 node. It is Right's goal to get the token to the n node. Left starts with d_L dollars. Right starts with d_R dollars.

2. A step of the game proceeds as follows:

   Left says how much he is willing to pay for a move, say b. b is a nonnegative real number that is $\leq$ how much Left has.

   (a) If Right agrees then token is moved to the left neighbor of the current node, but Right gets b dollars from Left. This is called a *Left move.*

   (b) If Right disagrees then he must pay Left an amount $> b$ but then gets to move the token to the right neighbor of the current node. This is called a *Right move.*

3. The game ends when the token is either on node 0, in which case Left wins, or on node n, in which case Right wins. The players play turns until one of them wins. (It could go on forever.)

NOTATION: Let $T = d_L + d_R$ be the total amount of money.
Let $N = n - 1$ be the number of edges in the graph.

Definition 1.1 *(a) A j-*run* for Left is a sequence of j Left moves followed by a Right move.*

*(b) A j-*run* for Right is a sequence of j Right moves followed by a Left move.*

2 Continuous Game

The game is said to be continuous if the bids are allowed to be arbitrary nonnegative real numbers.

If the token is at the middle node, then we consider that a fair allocation of money is when Left has $T/2$ dollars and Right has $T/2$ dollars. However, if the token is at the node which is at $1/3$ of the total length from the leftmost node and $2/3$ from the rightmost node, then it is natural to say that a fair allocation is when Left has $T/3$ dollars and Right has $2T/3$ dollars. Thus, the amount of money that Left and Right possess has to be considered relative to the position of the token. In general, if the token is at node p, then a fair allocation of money is when Left has $(p/N)T$ dollars and right has $((N-p)/N)T$ dollars. We define the *advantage* to be the number of dollars that Left has in addition to the fair allocation.

Definition 2.1 *The* advantage *at step t is (the amount of money Left has at step t) minus* $\frac{T}{N}\times$ *(the position of the token at step t).*

Note that if at step t, the node is at position p and the advantage is Δ, then Left has $p(T/N)+\Delta$ dollars and right has $(N-p)(T/N)-\Delta$ dollars.

It turns out that the advantage determines the winner of the game. If the advantage is ≥ 0, then Left has a winning strategy and if the advantage is <0, then Right has a winning strategy.

2.1 Left starts with at least as much money as Right

We first analyze the case when the advantage is ≥ 0 (or, in other words, when Left starts with at least as much money as Right relative to the position of the token). As announced, we show that in this case Left has a winning strategy. The proof consists of two steps.

We first show that if the advantage is 0, then Left has a strategy by which either she creates a small positive advantage, or she directly wins the game. We next show that if the advantage is >0, then Left has a strategy by which either she increases the advantage by at least a constant number of dollars, or she directly wins the game. Of course Left can repeat the increase-the-advantage phase as many times as she wants. Since the advantage cannot be larger than the total amount of money, it means that one of these phases will actually terminate with Left winning the game.

Concretely, the game consists of three phases. By phase, we mean a sequence of steps. In phase 1 ("creating an advantage"), the advantage is initially 0, and at the end the advantage has become $\Delta>0$. In phase 2 (increasing the advantage), the advantage is increased by a constant, so that by the end of the phase, $\Delta_{final}=\Delta_{start}+constant$. The constant depends on Δ (the advantage created in phase 1) and N, but not on time. The third phase ("steamrolling into victory") consists of a series of Left moves, at the end of which Left wins the game.

Phase 1 **(Creating a positive advantage):** $\Delta=0$. In this phase of the game, Left can create a positive advantage. Left's strategy is to bid $\frac{T}{N}$ dollars at each step.

Lemma 2.2 *Suppose that at some step t, the token is at node p and Left has* $p(T/N)$ *dollars. In other words, the advantage* Δ *is equal to* 0*. If Left bids* $\frac{T}{N}$ *at every node, then either*

(a) Left wins the game (by phase 3*, steamrolling into victory), or*

(b) after a certain number of steps, the advantage Δ *becomes* >0 *(by phase* 1*).*

Proof: There are two possible cases:

1. Right never chooses to outbid Left. Since Left has $p\frac{T}{N}$ dollars at step t, she can move the token up to node 0, which means that she wins the game.

2. Right outbids Left for the first time at node $i-1$ by an amount $\Delta'>0$ (in other words, there is a $(p-i+1)$-run for Left). At the end of this sequence of moves, the token is at node i. Left has spent $(p-i+1)\frac{T}{N}$ dollars to bring the token to node $i-1$, and in the last move Left receives $\frac{T}{N}+\Delta'$ dollars. So after this sequence of moves, Left has

$$p\frac{T}{N}-(p-i+1)\frac{T}{N}+\frac{T}{N}+\Delta'=i\frac{T}{N}+\Delta'.$$

So the advantage is $\Delta'>0$.

Phase 2 **(Increasing the advantage):** $\Delta > 0$. In this phase of the game, Left can increase the advantage Δ by at least $\frac{1}{2^{n-3}}\Delta$.

We first describe Left's strategy. Let $\epsilon_i = 1/(2^{i-1})$ for all $0 < i < n$. Left's strategy is as follows: when the token is at node i, she bids $\frac{T}{N} + \epsilon_i\Delta$.

Lemma 2.3 *Suppose that at some step t, the token is at position p and Left has $p(T/N) + \Delta$ dollars, for some $\Delta > 0$. In other words, the advantage is $\Delta > 0$. If Left bids $\frac{T}{N} + \epsilon_i\Delta$ at node i, for every $0 < i < n$, then either*

(a) Left wins the game (by phase 3), or

(b) after a certain number of steps, the advantage becomes at least $(1 + \frac{1}{2^{n-3}})\Delta$ (by phase 2).

Proof: As in the previous case, there are two situations.

1. Right never outbids Left. In this situation, Left has $p\frac{T}{N} + \Delta$ dollars at the initial step t.

   To bring the token to node 0, she needs to have

$$\left(\frac{T}{N} + \epsilon_p\Delta\right) + \left(\frac{T}{N} + \epsilon_{p-1}\Delta\right) + \ldots + \left(\frac{T}{N} + \epsilon_1\Delta\right) < p\frac{T}{N} + \Delta.$$

   Since Left has this amount of money, Left wins the game.

2. There is an m-run move for Left, where $0 \leq m < p$. We show that the advantage at the end of the run is bigger than Δ (the advantage at the beginning of the run), by at least $\frac{1}{2^{n-3}}\Delta$ dollars.

   Initially, the token is at node p and Left has $p\frac{T}{N} + \Delta$ dollars. After an m-run for Left, the token will be at node $p' = p - m + 1$, and Left has more than

$$\begin{aligned} &p\tfrac{T}{N} + \Delta - (\tfrac{mT}{N} + (\epsilon_p + \cdots + \epsilon_{p-m+1})\Delta) + \tfrac{T}{N} + \epsilon_{p-m}\Delta \\ &= (p - m + 1)\tfrac{T}{N} + (1 - (\epsilon_p + ... + \epsilon_{p-m+1}) + \epsilon_{p-m})\Delta \\ &= (p - m + 1)\tfrac{T}{N} + (1 + (\epsilon_{p-m} - (\epsilon_p + \ldots + \epsilon_{p-m+1}))\Delta \end{aligned}$$

   dollars.

   Now, $\epsilon_{p-m} - (\epsilon_p + \ldots + \epsilon_{p-m+1})$ turns out to be $= \frac{1}{2^m}$. Since $m < p < n$, it follows that $m \leq n-3$ and thus $\frac{1}{2^m} \geq \frac{1}{2^{n-3}}$. The conclusion follows.

Phase 3 **(Steamrolling to Victory):** We show that the advantage eventually becomes large enough that Left wins the game simply by outbidding Right at each step.

Theorem 2.4 *Consider the game when the token is initially at node p, $0 < p < n$, and Left initially has $p(T/N) + \Delta$ dollars, for some $\Delta \geq 0$. In other words, initially the advantage is $\Delta \geq 0$. Then Left has a strategy for winning the game.*

Proof: There are two cases: either $\Delta = 0$ or $\Delta > 0$. In the first case, by Lemma 2.2, Left has a strategy by which she either wins the game or, after a certain number of steps, the advantage becomes strictly positive, and thus the second case applies. In the second case, by Lemma 2.3, Left has a strategy by which she either wins the game or, after a certain number of steps, strictly increases the advantage. We show in the next claim that the advantage can only be increased finitely many times. This implies that Left wins the game.

Claim 2.5 *Case (b) of Lemma 2.3 can happen only a finite number of times..*

Proof: Let $\alpha = \frac{1}{2^{n-3}}$. Each time Case (b) happens the advantage is increased by a factor of $(1+\alpha)$. If Case (b) happens k times (where k is an arbitrary number), the advantage becomes

$$\Delta(1+\alpha)^k.$$

The above expression goes to ∞ when k goes to ∞, which means that if k is not bounded, eventually, the advantage would become $> T$. This cannot happen, because clearly, at any step t, the advantage is bounded by the total amount of money, T.

2.2 Right starts with more money than Left

We show that if the advantage is negative, then Right has a strategy to win the game. Observe that a negative advantage is equivalent to saying that Right has more money than Left, with respect to the current position of the token.

Theorem 2.6 *Consider the game in which the token starts at node p, $0 < p < n$, and Left initially has $p(T/N) - \Delta$ dollars for some $\Delta > 0$. In other words, the advantage is initially $-\Delta$. Then Right has a strategy for winning the game.*

Proof: The hypothesis states that at step $t = 0$, the token is at node p, Left has $p\frac{T}{N} - \Delta$ dollars, and Right has $(N-p)\frac{T}{N} + \Delta$ dollars.

Let $\gamma_k = (1/2)^{n-k+1}$, for $k = 0, 1, \ldots, n$. Right will play the game using the following strategy:

(a) If Left bids $\geq \frac{T}{N} + \gamma_i\Delta$ at node i, where $0 < i < n$, Right bids 0.

(b) If Left bids $< \frac{T}{N} + \gamma_i\Delta$ at node i, where $0 < i < n$, Right bids $\frac{T}{N} + \gamma_i\Delta$.

The next lemma shows that Right can indeed play the above strategy.

Lemma 2.7 *For all i, each time the token is at node i, Right has enough money to bid $\frac{T}{N} + \gamma_i\Delta$.*

Proof: We can think of Right as having two bank accounts, A and B. The initial $(N-i)\frac{T}{N} + \Delta$ dollars are distributed as follows. Bank account A has $(N-i)\frac{T}{N}$ dollars, which we think of as consisting of $N-i$ *units* where a unit is $\frac{T}{N}$ dollars. Bank account B starts with Δ dollars. At a move of type (a), Right adds $\frac{T}{N}$ (i.e., a unit) to A and $\gamma_i\Delta$ to B. At a move of type (b), Right subtracts $\frac{T}{N}$ from A and $\gamma_i\Delta$ from B. We show that for all i, each time the token is at position i, Right has at least $\frac{T}{N}$ in account A and at least $\gamma_i\Delta$ in account B.

Let us first look at account A. We fix position i where $0 < i < n$. We have two cases:

Case 1: $i \geq p$ (i.e., the position i is at or after the initial position p).

The first time the token moves from i to $i+1$, Right spends for the move out of his initial $N-i$ units. Let's consider the second time the token moves from i to $i+1$. There must have been an earlier step when the token moved from $i+1$ to i. At that time, Right gained a unit, which can be spent now. The same logic applies to all subsequent moves.

Case 2: $i < p$ (i.e., the position i is before the initial position p).

The first time the token is at i, it comes from $i+1$, meaning that a unit has been deposited in A. This unit is available for Right to spend as he moves right from i. The same argument applies to all subsequent times the token is at i.

Let us now consider account B. Again, we fix position i where $0 < i < n$. To move from i to $i+1$, Right needs to spend $(\frac{1}{2})^{n-i+1} \times \Delta$ from account B. Note that $\sum_{i=0}^{n-1} (\frac{1}{2})^{n-i+1} \times \Delta = \Delta[(\frac{1}{2})^{n+1} + \ldots + (\frac{1}{2})^2] < \Delta$. This implies that Right has enough money for the first move from i to $i+1$. For subsequent moves from i to $i+1$, there must have been a previous move from $i+1$ to i when Right deposited $(\frac{1}{2})^{n-i+1} \times \Delta$ in account B. Therefore, he has this amount to pay for the move.

Next, we show that using the above strategy Right decreases the advantage.

Lemma 2.8 *In a j-run for Right, for any $j \geq 0$, if the advantage at the beginning of the run is $-\Delta$, for some $\Delta > 0$, at the end of the run the advantage is $\leq -\Delta - (1/2)^{n+1}\Delta$.*

Proof: Suppose that before the j-run, the token is at node i and Left has advantage $-\Delta$. At the beginning of the run, Left has $i \times \frac{T}{N} - \Delta$ dollars. At the end of the j-run, the token is at node $i+j-1$. For the j Right moves, Left receives $j\frac{T}{N} + \gamma_i\Delta + \gamma_{i+1}\Delta + \ldots + \gamma_{i+j-1}\Delta$ dollars and in the final Left move of the run, Left spends $\frac{T}{N} + \gamma_{i+j}\Delta$ dollars. Thus, at the end of the run, Left has

$$
\begin{aligned}
&(i+j-1)\tfrac{T}{N} - \Delta - \Delta(\gamma_{i+j} - (\gamma_i + \ldots + \gamma_{i+j-1})) \\
&= (i+j-1)\tfrac{T}{N} - \Delta - (1/2)^{n-i+1}\Delta \\
&\leq (i+j-1)\tfrac{T}{N} - \Delta - (1/2)^{n+1}\Delta.
\end{aligned}
$$

(We used the fact that $\gamma_{i+j} - (\gamma_i + \gamma_{i+1} + \ldots + \gamma_{i+j-1}) = (1/2)^{n-i+1}$.) ∎

Now we can complete the proof of Theorem 2.6. Any series of steps can be broken down into a series of j-runs for Right, for various values of j (with the possible exception that the last run of the game may not end in a Left move). Lemma 2.8 implies that the advantage is negative by the end of each run, because initially the advantage is negative and each run decreases the advantage. This means that if the token reaches node 1 at the end of a run for Right, Left has less than $\frac{T}{N}$ dollars, and so the next move will be a Right move (by Right's strategy). Thus, Left cannot win the game.

After each run for Right the advantage decreases by at least $(1/2)^{2n+1} \times \Delta$. If this happens k times, the advantage becomes

$$-\Delta - k(1/2)^{2n+1}\Delta$$

The above expression converges to $-\infty$ when k goes to ∞. Since the advantage cannot be less than $-T$, it means that k, the number of runs, is finite. Thus, Right wins the game. ∎

Remark: The fact that Left bids first does not give Right an advantage. Right can bid $\frac{T}{N} + \gamma_i\Delta$ when the token is at position i regardless of how much Left bids, and the above analysis is still valid.

3 Discrete Game

We say that a game is discrete if d_L, d_R, and the bids are positive integer numbers.

Definition 3.1 *A configuration is given by the 4-tuplet (Money Left, Money Right, current position of the token, number of nodes).*

We define the *advantage* by analogy to the corresponding definition in the continuous case.

Definition 3.2 *Suppose that at step t in the game, the token is at node p. We call the* advantage *at step t to be (Left's Money at step t) $- \left\lceil \frac{pT}{N} \right\rceil$.*

3.1 Left starts with at least as much money as Right

In the discrete case, the game consists of two phases. In phase 1, Left increases her advantage by at least 1 dollar. Phase 2 consists of a sequence of Left moves, at the end of which Left wins. We do not have a phase that is analog to the "create the advantage" phase in the continuous version. In that version of the game, the increases in the advantage depended on the first phase, or how large the initial advantage was. In the discrete version, however, the increases in the advantage are not dependent on one another and therefore there is no difference between creating the initial advantage and increasing the advantage aftwerwards.

The following theorem is the discrete analog of Theorem 2.4.

Theorem 3.3 *Consider the game when the token is initially at node p, $0 < p < n$, and Left initially has $\left\lceil \frac{pT}{N} \right\rceil + \Delta$ dollars, for some $\Delta \geq 0$. In other words, initially the advantage is $\Delta \geq 0$. Then Left has a strategy for winning the game.*

Proof: Left's strategy is to bid $\left\lfloor \frac{T}{N} \right\rfloor$ at each step. We prove the following claim.

Claim 3.4 *Suppose that at some step t, the advantage is $\Delta \geq 0$. Then there exists a step $t' > t$, when either*

(a) the advantage is $\geq \Delta + 1$, or

(b) Left has won the game.

We observe that the claim establishes the theorem, because situation (a) can happen only a finite number of times (namely, at most T times), and thus situation (b) will eventually occur.

Proof of Claim:

The first observation is that Right cannot do $N - p$ Right moves (which would have him reach node N and thus win the game). Suppose the contrary is true. Then when the token is at node $N - 1$, Left would have at least

$$\left\lceil \frac{pT}{N} \right\rceil + (N - 1 - p)(\left\lfloor \frac{T}{N} \right\rfloor + 1),$$

because he starts with at least $\left\lceil \frac{pT}{N} \right\rceil$ dollars and at each of the next $N - 1 - p$ steps he wins at least $\left\lfloor \frac{T}{N} \right\rfloor + 1$ dollars.

There are three cases left to analyze.

Case 1: The next steps are m Right moves, with $0 \leq m < N - p$, followed by a Left move (in other words, an m-run for Right).

After the first m steps (which are Right moves), the token is at node $p + m$, and Left has

$$\left\lceil \frac{pT}{N} \right\rceil + \Delta + m(\left\lfloor \frac{T}{N} \right\rfloor + 1) \geq \left\lceil \frac{(p+m)T}{N} \right\rceil + \Delta.$$

Therefore, after these first m steps, the advantage does not decrease. In the last step, which is a Left move, the token is moved to node $p+m-1$, and Left has at least

$$\left\lceil \frac{(p+m)T}{N} \right\rceil + \Delta - \left\lfloor \frac{T}{N} \right\rfloor \geq \left\lceil \frac{(p-m+1)T}{N} \right\rceil + \Delta + 1.$$

Thus, at step $m+1$, the advantage has increased by at least 1.

Case 2: There are m Left moves, with $0 \leq m < p$, followed by a Right move (in other words, an m-run for Left).

At the end, the token is at node $p-m+1$, and Left has at least

$$\left\lceil \frac{pT}{N} \right\rceil + \Delta - (m-1)\left\lfloor \frac{T}{N} \right\rfloor + 1 \geq \left\lceil \frac{(p-m+1)T}{N} \right\rceil + \Delta + 1$$

Thus, the advantage has increased by at least 1 dollar, because he has at least 1 more than $\left\lceil \frac{(p-m+1)T}{N} \right\rceil$ at position $p-m+1$.

Case 3 (Trivial Case): The next p steps are all Left moves. In this case, obviously Left wins the game.

This ends the proof of the claim and of the theorem.

3.2 Right starts with more money than Left

In this section, we show that in contrast to the continuous case, it is possible for Right to have much more money than Left and still not win the game.

Theorem 3.5 *Consider the game with $2n+1$ positions. Let $k = n(n+1)/2$.*

(a) If Right has at most $k-1$ dollars, and the token is at node n, then Left can force a draw regardless of how much money he has. Thus, Right does not win if Left plays optimally. Note that this means that initially the advantage can be as small as $-\left\lceil \frac{n^2+n-2}{4} \right\rceil$, when Left has 0 dollars and Right has $\frac{n(n+1)}{2} - 1$ dollars.

(b) If Right has at least k dollars, Left has 0 dollars, and the token is at node n, then Right has a winning strategy.

Proof: (a) Clearly, it is most favorable for Right when Left starts with 0 dollars. We show in the following fact that in this case Left can force a draw.

Fact 3.6 *For any n, if the game starts in configuration $(0, k-1, n, 2n+1)$, Left has a strategy for a draw.*

Proof: Suppose that Right has a winning strategy. Then Right can play a game without repeating any configuration (by Lemma 3.12 NOTE LEMMA MOVED TO OBSERVATIONS. BUT IT'S OBVIOUS). The initial configuration is $(0, k-1, n, 2n+1)$. Left's optimal strategy is as follows. She bids 0 in the first round, forcing Right to bid 1 (otherwise, since Left wins when the bids are equal, Right would lose a position and not win any money from Left). Now, the game is $(1, k-1, n+1, 2n+1)$. Left will bid 1, and Right will respond with a bid of 2, since if he bid 0, he would return to the game $(0, k, n, 2n+1)$ and thus force a repetition. Left's general strategy is to bid $i-1$ dollars in round i, and Right is forced to bid i in order to prevent a draw; otherwise, if he bids 0, the game would return to the previous configuration. After $n-1$ steps, the game reaches the configuration $(\frac{n(n-1)}{2}, (k-1) - \frac{n(n-1)}{2}, 2n-1, 2n+1)$. Note that at this point, Right has $(k-1) - \frac{n(n-1)}{2} = n-1$ dollars. Now Left bids $n-1$, Right can only bid 0, and the game goes into the previous configuration. Thus, Right cannot win.

(b) Right's strategy is as follows: For $k = 0, 1, \ldots, n-1$, in configuration $(1+2+\ldots+k, (k+1)+\ldots+n, n+k, 2n+1)$, if Left bids $k+1$ or more, Right bids 0, and if Left bids k or less, Right bids $k+1$. In all other configurations, Right bids 0 regardless of what Left bids (the proof will involve only the first kind of configurations).

Suppose Right does not win with this strategy. This means that either Left has a strategy that induces a loop (in which case the game is a draw), or Left has a winning strategy. In case Left has a winning strategy, she also has a winning strategy with a minimal number of steps, which we call an *optimal Left win*.

We will show the following fact.

Fact 3.7 *The run of the game is either*

(1) a sequence of configurations $c_1 \to c_2 \to \ldots \to c_i \to \ldots \to c_j \to \ldots \to c_t$ *in which there exist* $i < j$ *such that the token positions in* c_i *and* c_j *are the same, and Left's money in* c_j *is less than Left's money in* c_i*, or*

(2) is the sequence $(0, 1+2+\ldots+n, n, 2n+1) \to (1, 2+\ldots+n, n+1, 2n+1) \to \ldots \to (1+2+\ldots+k, (k+1)+\ldots+n, n+k, 2n+1) \to \ldots \to (1+2+\ldots+n, 0, 2n, 2n+1)$.

This will finish the proof, because in both cases, we have a contradiction. In case (1), we obtain a contradiction, because that would not be an optimal Left win. This is so because Left can play in such a way that the game runs through the configurations $c_1 \to \ldots \to c_i \to c_{j+1} \to \ldots \to c_t$, which is a shorter run than $c_1 \to c_2 \to \ldots \to c_i \to \ldots \to c_j \to \ldots \to c_t$.

In case (2), the contradiction is that actually Right, and not Left, wins in the last configurations of the run.

We now prove the fact. The game starts with configuration $(0, 1+2+\ldots+n, n, 2n+1)$. Left can only bid 0, so Right will bid 1 and the game will enter configuration $(1, 2+\ldots+n, n+1, 2n+1)$. Now, Left bids 0 or 1, Right bids 2, and the new configuration is $(1+2, 3+\ldots+n, n+2, 2n+1)$. If Left bids 3, Right will bid 0 and the game will enter configuration $(0, 3+3+\ldots+n, n+1, 2n+1)$, which would induce a sequence of type (1) configurations. If Left bids $0, 1$, or 2, then Right bids 3, and the game goes into configuration $(1+2+3, 4+\ldots+n, n+3, 2n+1)$.

In general, suppose that at the kth move, configurations of type (1) have been avoided and the last transition was $(1+\ldots+(k-1), k+\ldots+n, n+k-1, 2n+1) \to (1+\ldots+k, (k+1)+\ldots+n, n+k, 2n+1)$. If Left bids $k+1$ or more, Right will respond with 0, and the game moves to $(x, k-x, n+k-1, 2n+1)$, with $x < 1+\ldots+(k-1)$. This means that the sequence of configurations is of type (1). If Left bids k or less, Right bids $k+1$, and the game moves to configuration $(1+2+\ldots+(k+1), (k+2)+\ldots+n, n+k+1, 2n+1)$. This finishes the proof of the fact.

3.3 Observations - probably will be eliminated in the final paper

Lemma 3.8 *If d is an integer multiple of $n*2^{2n-1}$, then the continuous strategy can be directly applied to the discrete case, since $\frac{id}{n}$ and $\frac{1}{2^i}$ are guaranteed to be integers for all i, n.*

Lemma 3.9 *If the game has gone on for more than $(2n-1)(2d+1)$ moves, and both players are playing optimally, then the game is a draw.*

Proof: There are $(2n-1)$ nodes such that if the token is at node i, with $1 \leq i \leq 2n-1$, the game has not yet ended. There are $(2d+1)$ combinations of dollar amounts for Left and Right. Thus, by the pigeonhole principle, if there have been more than $(2n-1)(2d+1)$ moves, then at least one configuration will have been repeated. We know from the previous lemma that this will result in a draw.

Lemma 3.10 *Let p be the position of the node, T the total amount of money, and N the highest node. If Left's money is $\geq \left\lceil \frac{pT}{N} \right\rceil + k$, for any $k > 0$, Left wins.*

Proof: We prove this by induction.

Base Case: Let $p = 0$. Then Left wins by the rules of the game. Now let $p = 1$. Left has $\geq \left\lceil \frac{T}{N} \right\rceil + k$ dollars, and he bids $\left\lceil \frac{T}{N} \right\rceil + k$. If Right lets Left win this bid, then $p = 0$ and Left wins the game. So Right will outbid Left, and now the token will be at $p = 2$, with Left having $2\left\lceil \frac{T}{N} \right\rceil + k + 1 > \left\lceil \frac{2T}{N} \right\rceil + k$. Thus, Left gains an advantage, with which he can eventually win the game.

Lemma 3.11 *If the game starts from the configuration $(M_L, M_R, k, 2n+1)$ and if $M_R < 2n-k$, Left wins regardless of M_L.*

Proof: Left always bids 0. Right can respond with a bid of 1 only at most M_R times and this is not enough to reach the rightmost node which is node $2n$. After Right has his finished his money, the token will only move to the left and thus Left wins.

Lemma 3.12 *If both players are playing optimally, and the same configuration occurs twice, then the game is a draw.*

Lemma 3.13 *If d is an integer multiple of $n * 2^{2n-1}$, then the continuous strategy can be directly applied to the discrete case, since $\frac{id}{n}$ and $\frac{1}{2^i}$ are guaranteed to be integers for all i, n.*

Lemma 3.14 *If the game has gone on for more than $(2n-1)(2d+1)$ moves, and both players are playing optimally, then the game is a draw.*

Proof: There are $(2n-1)$ nodes such that if the token is at node i, with $1 \leq i \leq 2n-1$, the game has not yet ended. There are $(2d+1)$ combinations of dollar amounts for Left and Right. Thus, by the pigeonhole principle, if there have been more than $(2n-1)(2d+1)$ moves, then at least one configuration will have been repeated. We know from the previous lemma that this will result in a draw.

Lemma 3.15 *Let p be the position of the node, T the total amount of money, and N the highest node. If Left's money is $\geq \left\lceil \frac{pT}{N} \right\rceil + k$, for any $k > 0$, Left wins.*

Proof: We prove this by induction.

Base Case: Let $p = 0$. Then Left wins by the rules of the game. Now let $p = 1$. Left has $\geq \left\lceil \frac{T}{N} \right\rceil + k$ dollars, and he bids $\left\lceil \frac{T}{N} \right\rceil + k$. If Right lets Left win this bid, then $p = 0$ and Left wins the game. So Right will outbid Left, and now the token will be at $p = 2$, with Left having $2\left\lceil \frac{T}{N} \right\rceil + k + 1 > \left\lceil \frac{2T}{N} \right\rceil + k$. Thus, Left gains an advantage, with which he can eventually win the game.

4 Money Goes To The Bank

4.1 Introduction

In this version of the game, the player with the highest bid pays the money to a bank instead of to the other player. The key observation is that the ratio $\frac{d_L}{d_R}$ determines whether Left wins, Right wins, or if it is a draw. Indeed suppose that d_L and d_R are such that, say, Left wins. Then if d'_L and d'_R are values such that $\frac{d'_L}{d'_R} = \frac{d_L}{d_R}$, then Left can play the same strategy just rescaled and she will win in the new situation as well. Note that this fact is not valid in the discrete version, because scaling may not be possible. We present an algorithm that determines when Left wins, when Right wins, and when the game is a draw, based on the ratio of the two players' money amounts ($\frac{d_L}{d_R}$). The algorithm can also be used to find the winning strategies for Left and Right, when such strategies exist.

4.2 Determining d_L/d_R so that Left wins, and Left's winning strategy

Definition 4.1 *For any $p \in \{1, \ldots, N-1\}$ and for any natural number t, let $V(p,t)$ be the set of numbers v such that if $\frac{d_L}{d_R} \geq v$, then Left wins in at most t steps when the token is at node p, Left has d_L dollars, and Right has d_R dollars. Let $v(p,t) = \inf V(p,t)$, in case $V(p,t) \neq \emptyset$. If $V(p,t) = \emptyset$, we set $v(p,t) = \infty$.*

Clearly, $V(p,t)$ can be of one of the following forms: (a) $V(p,t) = \emptyset$, (b) $V(p,t) = [v(p,t), \infty)$, or (c) $V(p,t) = (v(p,t), \infty)$. We will show that, in fact, case (c) never happens.

For each $p \in \{1, \ldots, N-1\}$, the sequence $v(p,t), t = 1, 2, \ldots$, is decreasing and consists of nonnegative terms. It follows that $\lim_{t\to\infty} v(p,t)$ exists and is a finite number, which we denote $v(p)$. The numbers $v(p,t)$, for $t = 1, 2, \ldots$, and $v(p)$ are helpful in finding a winning strategy for Left, if such a strategy exists. Suppose that the token is at node p. Consider first the case when $p \neq 1$ and $p \neq N-1$. If $d_L/d_R < v(p)$, then Left does not win (if Right plays well). So, suppose that $d_L/d_R \geq v(p)$. If Left has a winning strategy, there exists a number of steps t, in which the strategy is winning. This means that there exists t such that $d_L/d_R \geq v(p,t)$. Thus, Left's strategy is as follows:

> 1. Find t such that $\frac{d_L}{d_R} \geq v(p,t)$.
> 2. Find a number $x \in [0, L]$ so that
> $(\frac{d_L - x}{d_R} \geq v(p-1,t))$ AND $(\forall \epsilon \in (0, R-x], \frac{d_L}{d_R - x - \epsilon} \geq v(p+1,t))$;
> Bid x dollars.

Such a number x is guaranteed to exist, otherwise Left would not have a winning strategy. Indeed, suppose that for every $x \in [0, d_L]$,

$(d_L - x)/d_R < v(p-1,t))$ OR $d_L/(d_R - x) < v(p+1,t))$.

Then Right would respond either 0 (in the first situation) or $x + \epsilon$ (for an $\epsilon > 0$ such that $\frac{d_L}{d_R - x - \epsilon}$ is still less than $v(p-1,t)$), and the game would move into a configuration in which Left cannot win in t steps.

Let us now analyze the case when the token is at node $p = 1$. Then, Left's strategy is as follows.

1. If $d_L \geq d_R$, bid d_L.
2. Else, find t such that $\frac{d_L}{d_R} \geq v(1,t)$.
3. Find $x \in [0, d_L]$ such that $\frac{d_L}{d_R - x} \geq v(2,t)$.
4. Bid x.

In case $p = N - 1$, Left's strategy is the following:

1. Find t such that $\frac{d_L}{d_R} \geq v(N-2,t)$.
2. Find $x \geq d_R$ such that $\frac{d_L - x}{d_R} \geq v(N-2,t)$.
3. Bid x.

Next we explain how to compute the values $v(p,t)$.

We find a recurrence relation for $V(p,t)$ and $v(p,t)$. The general idea is that Left wins in $t+1$ steps, if she has a bid x such that for all responses of Right, the game moves into a configuration in which Left wins in t steps (or, she directly wins the game). A value of x that satisfies this requirement is called a *good bid*.

The initial conditions ($t = 1$) are that $V(1,1) = [1,\infty)$, $v(1,1) = 1$ and $V(p,1) = \emptyset$, $v(p,1) = \infty$, for $p \neq 1$, which can be readily checked.

Next, we determine $V(p,t+1)$ and $v(p,t+1)$, for $t \geq 1$. Depending on the values of p, $v(p-1,t)$ and $v(p+1,t)$, there are five possible cases:

Case 1: $p = 1$.

We find a recurrence relation for v(1, t+1). At node $p = 1$, Left wins in $\leq t+1$ steps if one of the following two conditions hold true:

1. $d_L \geq d_R \Leftrightarrow \frac{d_L}{d_R} \geq 1$. This means that Right cannot outbid Left, and Left is able to move the token to node 0 and win the game, or
2. There exists an x, $0 \leq x \leq d_L$, such that $\frac{d_L}{d_R - x - \epsilon} \geq v(2,t)$ for all ϵ, with $0 < \epsilon \leq d_R - x$ (which is equivalent to $\frac{d_L}{d_R - x} \geq v(2,t)$). This means that Right outbids Left at node $p = 1$ by bidding $x + \epsilon$, and afterwards Left can win the game in t steps.

From the second condition, we get the following two conditions: $x \leq d_L$ and $x \geq d_R - \frac{d_L}{v(2,t)}$. Putting these together, we find that a good bid x exists iff $d_L \geq d_R - \frac{d_L}{v(2,t)} \Leftrightarrow \frac{d_L}{d_R} \geq \frac{1}{1 + \frac{1}{v(2,t)}} \Leftrightarrow \frac{d_L}{d_R} \geq \frac{v(2,t)}{v(2,t)+1}$.

Thus, $v(1,t+1) = \min(1, \frac{v(2,t)}{v(2,t)+1}) = \frac{v(2,t)}{v(2,t)+1}$.

Case 2: $p = N - 1$.

At node $p = N - 1$, Left wins in $\leq t+1$ steps iff the following two conditions both hold true.

1. $d_L \geq d_R$. Otherwise, Right can outbid Left and move the token to node N, thus winning the game.

2. There exists x, $d_R \leq x \leq d_L$, such that $\frac{d_L - x}{d_R} \geq v(N-2,t)$. This means that Left bids an amount $x \geq d_R$ and moves the token to node $N-2$, and afterwards she can win the game in t steps.

From the second condition, we get that $x \leq d_L - v(N-2,t)d_R$. Since $x \geq d_R$, we get that a good bid x exists iff $d_R \leq d_L - v(N-2,t)d_R \Leftrightarrow \frac{d_L}{d_R} \geq 1 + v(N-2,t)$.

Thus, $v(N-1,t+1) = 1 + v(N-2,t)$.

Case 3: $p \neq 1, p \neq N-1, t \geq 1, v(p-1,t) \neq \infty$ and $v(p+1,t) \neq \infty$.

Intuitively, this means that, at nodes $p-1$ and $p+1$, Left can win in t steps for certain values of $\frac{d_L}{d_R}$.

At node p, Left wins in $\leq t+1$ steps iff there exists a bid x, $0 \leq x \leq d_L$, such that the following two conditions both hold true:

(1) Left wins from node $p-1$ in t steps with Left having $d_L - x$ dollars and Right having d_R dollars.

(2) (a) For all $\epsilon, 0 < \epsilon < d_R - x$, Left wins from node $p+1$ in t steps with Left having d_L dollars and Right having $d_R - x - \epsilon$ dollars or

(b) $x \geq d_R$.

We find the recursion relation for $v(p,t+1)$ using the fact that Left wins iff there exists x that satisfies [(1) AND (2,a)] OR [(1) AND (2,b)].

Conditions (1) and (2,a) From (1), we get that $x \leq d_L - v(p-1,t)d_R$, and from (2,a), $x \geq d_R - \frac{d_L}{v(p+1,t)}$. Putting these equations together, we get that x satisfying (1) AND (2,a) exists iff $\frac{d_L}{d_R} \geq \frac{1+v(p-1,t)}{1+\frac{1}{v(p+1,t)}} \Leftrightarrow \frac{d_L}{d_R} \geq \frac{v(p+1,t)+v(p-1,t)v(p+1,t)}{v(p+1,t)+1}$.

NOTE: In the second condition, we use the fact that $\frac{d_L}{d_R-x-\epsilon} \geq v(p+1,t) \Leftrightarrow \frac{d_L}{d_R-x} \geq v(p+1,t)$. We prove this now. $\frac{d_L}{d_R-x-\epsilon} > \frac{d_L}{d_R-x}$, so clearly if $\frac{d_L}{d_R-x} \geq v(p+1,t)$, then $\frac{d_L}{d_R-x-\epsilon} \geq v(p+1,t)$. Conversely, suppose that $\frac{d_L}{d_R-x} < v(p+1,t)$. Then there exists a small enough ϵ so that $\frac{d_L}{d_R-x-\epsilon} < v(p+1,t)$. This is a contradiction.

Conditions (1) AND (2,b) From (1), we get that $x \leq d_L - v(p-1,t)d_R$, and (2,b) $x \geq d_R$. So such an x exists, iff $d_R \leq d_L - v(p-1,t)d_R \Leftrightarrow \frac{d_L}{d_R} \geq 1 + v(p-1,t)$.

Thus, a good bid x exists iff $d_L/d_R \geq \min(\frac{v(p+1,t)+v(p-1,t)v(p+1,t)}{v(p+1,t)+1}, 1+v(p-1,t)) = \frac{v(p+1,t)+v(p-1,t)v(p+1,t)}{v(p+1,t)+1}$. Thus, $v(p,t+1) = \frac{v(p+1,t)+v(p-1,t)v(p+1,t)}{v(p+1,t)+1}$.

Case 4: $p \neq 1, p \neq N-1, v(p-1,t) \neq \infty$ and $v(p+1,t) = \infty$.

Intuitively, this means that at node $p-1$, Left can win the game in t steps for a certain $\frac{d_L}{d_R}$, but at node $p+1$, there is no $\frac{d_L}{d_R}$ such that Left wins the game in t steps.

At node p, Left wins the game in $t+1$ steps iff there exists an x, $d_R \leq x \leq d_L$, such that $\frac{d_L-x}{d_R} \geq v(p-1,t) \Leftrightarrow x \leq d_L - v(p-1,t)d_R$. Since we also have that $x \geq d_R$, we get that a good bid x exists iff $d_R \leq d_L - v(p-1,t)d_R \Leftrightarrow \frac{d_L}{d_R} \geq 1 + v(p-1,t)$.

Thus, $v(p, t+1) = 1 + v(p-1, t)$.

Case 5: $p \neq 1, p \neq N-1, t \geq 1, v(p-1, t) = \infty$ and $v(p+1, t) = \infty$.

Intuitively, this means that at node $p-1$, Left cannot win the game in t steps, and that at node $p+1$, Left cannot win the game in t steps. Clearly, if Left cannot win the game in t steps from nodes $p-1$ and $p+1$, then she cannot win from node p in $t+1$ steps either.

Thus, $v(p, t+1) = \infty$.

4.3 Determining $\frac{d_L}{d_R}$ so that Right wins, and Right's winning strategy

Definition 4.2 *For any $p \in 1, \ldots, N-1$ and for any natural number t, let $U(v, t)$ be the set of numbers u such that if $\frac{d_L}{d_R} \leq u$, then Right wins in at most t steps when the token is at node p, Left has d_L dollars, and Right has d_R dollars. Let $u(p, t) = \sup U(p, t)$, in case $U(p, t) \neq \emptyset$. If $U(p, t) = \emptyset$, we set $u(p, t) = -1$ (the value -1 is arbitrary, but ensures that the sequence $u(p, t)$ is increasing with t).*

$U(p, t)$ can be of one of the following forms: (a) $U(p, t) = \emptyset$, (b) $U(p, t) = (0, u(p, t))$, or (c) $U(p, t) = [0, u(p, t)]$. We will show that, in fact, case (c) never happens.

Next we describe Right's winning strategy, if there exists one. Let p be the position of the token and consider first that $p \neq 1$ and $p \neq N_1$. The sequence $u(p, t), t = 1, 2, \ldots$, is increasing and bounded, and thus it has a limit, which we denote $u(p)$. Right has a winning strategy iff $d_L/d_R < u(p)$. In case there is a winning strategy, Right determines it as follows: For whatever $x, 0 \leq x \leq d_L$, that Left bids, Right finds t such that $\frac{d_L - x}{d_R} < u(p-1, t)$ or $\frac{d_L}{d_R - x - \epsilon} < u(p+1, t)$ for some small ϵ. Note that the second relation is equivalent with $\frac{d_L}{d_R - x} < u(p+1, t)$.

In other words, at position p (with $p \neq 1$ and $p \neq N-1$), Right has a strategy to win the game if he can either win the game in t steps from position $p-1$ or win in t steps from position $p+1$. If Right can win from $p-1$ in t steps, he will bid 0 dollars, so that Left moves the token from p to $p-1$. If Right can win from $p+1$ in t steps, he will bid $x + \epsilon$ dollars, moving the token from p to $p+1$. Thus, Right's strategy is as follows. Let x be Left's bid. Then Right does the following:

1. Find t such that $\frac{d_L - x}{d_R} < u(p-1, t)$ or $\frac{d_L}{d_R - x} < u(p+1, t)$.
2. If $\frac{d_L - x}{d_R} < u(p-1, t)$, bid 0 dollars.
3. If $\frac{d_L}{d_R - x} < u(p+1, t)$, bid $x + \epsilon$ dollars.
4. If $\frac{d_L - x}{d_R} < u(p-1, t)$ and $\frac{d_L}{d_R - x} < u(p+1, t)$, bid either 0 or $x + \epsilon$ dollars.

Let us consider the case $p = 1$. In that case, Right's strategy is as follows.

1. Find t such that $\frac{d_L}{d_R - x} < u(p+1, t)$.
2. If $\frac{d_L}{d_R - x} < u(p+1, t)$, bid $x + \epsilon$ dollars.

Finally, the case $p = N - 1$.

1. If $x < d_R$, then bid $x + \epsilon$, and Right wins the game.
2. Else, bid 0.

Next we explain how to compute the values $u(p, t)$.

We find a recurrence relation for $U(p,t)$ and $u(p,t)$. The general idea is that Right wins in $t+1$ steps if for all Left bids x, $0 \le x \le d_L$, Right has a choice between bidding 0 or $x+\epsilon$ that will take the game into a configuration where he wins in t steps (or he directly wins the game).

The initial conditions ($t=1$) are that $U(N-1,1) = (0,1) \Rightarrow u(N-1,t) = 1$, and $U(p,1) = \emptyset$ for $p \ne N-1$, which can be readily checked.

Next, we determine $U(p,t+1)$ and $u(p,t+1)$, for $t \ge 1$. Depending on the values of p, $u(p-1,t)$ and $u(p+1,t)$, there are five possible cases:

Case 1: $p = N-1$

We find a recurrence relation for $u(N-1,t+1)$. There are two cases. The first case is when $U[N-2,t] = \emptyset$. In this case, Right wins iff $d_L/d_R < 1$, so $u(N-1,t+1) = 1$ and $U(N-1,t+1) = [0,1)$.

The second case is when $U[N-2,t] = [0, u(N-2,t))$. At node $N-1$, Right *does not win* in $t+1$ steps iff $d_L \ge d_R$ and there exists $x \in [d_R, d_L]$, such that $\frac{d_L - x}{d_R} \ge u(N-2,t)$. After calculations, we get that such an x exists iff $\frac{d_L}{d_R} \ge 1 + u(N-2,t)$. So Right wins in $t+1$ steps iff $\frac{d_L}{d_R} \ge 1 + u(N-2,t)$ and thus $u(N-1,t+1) = 1 + u(N-2,t)$ and $U(N-1,t+1) = [0, 1+u(N-2,t))$.

Case 2: $p = 1$

We find a recurrence relation for $u(1,t+1)$. Clearly, if $u(2,t) = -1$ (and $U(2,t) = \emptyset$, then Right cannot win in $t+1$ steps, and thus $u(1,t+1) = -1$ and $U(1,t+1) = \emptyset$. So, assume that $u(1,t+1) \ne -1$. At node $p=1$, Right wins in $t+1$ steps iff for all x, $0 \le x \le d_L$, both of the following conditions are true:

1. $d_L < d_R$. Otherwise, Left would outbid Right and win the game.
2. There exists an $\epsilon, 0 \le \epsilon \le d_R - d_L$, such that $\frac{d_L}{d_R - d_L - \epsilon} < u(2,t) \Leftrightarrow \frac{d_L}{d_R - d_L} < u(2,t)$. This means that for any bid x, Right can win the game from node 2 in t steps.

From the second condition, we get that $\frac{d_L}{d_R} < \frac{u(2,t)}{1+u(2,t)}$. So $u(1,t+1) = \frac{u(2,t)}{1+u(2,t)}$, and $U(1,t+1) = [0, \frac{u(2,t)}{1+u(2,t)})$ otherwise.

Case 3: $p \ne 1, p \ne N-1, U(p-1,t) = [0, u(p-1,t)), U(p+1,t) = [0, u(p+1,t))$

We find a recurrence relation for $u(p,t+1)$. To do this, we find out when Right *does not win* in $t+1$ steps and then negate the recurrence relation we obtain. Right does not win in $t+1$ steps iff there exists an $x, 0 \le x \le d_L$ such that both of the following conditions are true:

1. $\frac{d_L - x}{d_R} \ge u(p-1,t)$. This means that Right does not win in t steps from node $p+1$.
2. For all $\epsilon, 0 < \epsilon \le d_R - x$, $\frac{d_L}{d_R - x - \epsilon} \ge u(p+1,t) \Leftrightarrow \frac{d_L}{d_R - x} \ge u(p+1,t)$. This means that regardless of Right's bid, Right does not win the game in t steps from node $p-1$.

From the first condition, we get that $x \le d_L - u(p-1,t)d_R$, and from the second equation we get that $x \ge \frac{u(p+1,t)d_R - d_L}{u(p+1,t)}$. Putting these two inequalities together, it results that there exists an $x, 0 \le x \le d_L$ such that regardless of Right's bid, Right does not win the game in t steps if

$$d_L - u(p-1,t)d_R \ge \frac{u(p+1)d_R - d_L}{u(p+1,t)} \Leftrightarrow \frac{d_L}{d_R} \ge \frac{1 + u(p-1,t)}{1 + \frac{1}{u(p+1,t)}}$$

Thus, Right wins in $t+1$ steps iff $\frac{d_L}{d_R} < \frac{1+u(p-1,t)}{1+\frac{1}{u(p+1,t)}} = \frac{u(p+1,t)+u(p-1,t)u(p+1,t)}{1+u(p+1,t)}$. So $u(p,t+1) = \frac{u(p+1,t)+u(p-1,t)u(p+1,t)}{1+u(p+1,t)}$, and $U(p,t+1) = \left[0, \frac{u(p+1,t)+u(p-1,t)u(p+1,t)}{1+u(p+1,t)}\right)$.

Case 4: $p \neq 1, p \neq N-1, u(p-1,t) = \emptyset, u(p+1,t) = [0, u(p+1,t))$

We find a recurrence relation for $u(p,t+1)$. Again, we first find the recurrence relation in which Right *does not win* the game in $t+1$ steps and negate the equation to obtain $u(p,t+1)$. Right does not win the game in $t+1$ steps iff there exists $x, 0 \leq x \leq d_L$ such that for all $\epsilon, 0 < \epsilon \leq d_R - x$, $\frac{d_L}{d_R-x-\epsilon} \geq u(p+1,t) \Leftrightarrow \frac{d_L}{d_R-x} \geq u(p+1,t)$. From this equation, we get that $x \geq d_R - \frac{d_L}{u(p+1,t)}$. Combining this equation with the fact that $x \leq d_L$, we get that Right does not win in $t+1$ steps iff $\frac{d_L}{d_R} \geq \frac{1}{1+\frac{1}{u(p+1,t)}}$.

Thus, Right wins in $t+1$ steps iff $\frac{d_L}{d_R} < \frac{1}{1+\frac{1}{u(p+1,t)}}$. So $u(p,t+1) = \frac{1}{1+\frac{1}{u(p+1,t)}} \Leftrightarrow u(p,t+1) = \frac{u(p+1,t)}{1+u(p+1,t)}$, and $U(p,t+1) = [0, \frac{u(p+1,t)}{1+u(p+1,t)})$.

Case 5: $p \neq 1, p \neq N-1, U(p-1,t) = \emptyset, U(p+1,t) = \emptyset$

In this case, it is clear that $U(p,t+1) = \emptyset$ (and thus, $u(p,t+1) = -1$), because if Right cannot win the game from node $p-1$ in t steps or win from node $p+1$ in t steps, he cannot win from node p in $t+1$ steps either.

4.4 Analysis of the game in which the token starts at the middle node

We show that if the token starts at the middle node, if $d_L > d_R$ then Left wins, and if $d_L < d_R$ then Right wins. Numerical experiments show that Left also wins if $d_L = d_R$, but we don't have a proof for this situation.

Recall that we have denoted $v(p) = \lim_{t\to\infty} v(p,t)$, and $u(p) = \lim_{t\to\infty} u(p,t)$ for $p = 1, 2, \ldots, N-1$. We show that $v(N/2) = 1$ and $u(N/2) = 1$, which implies the above assertions.

Let us concentrate first on $v(N/2)$. Actually, we prove that $v(p) = \frac{1}{v(N-p)}$ for every $p \in \{1, N-1\}$. From this, it follows that $v(N/2) = 1$, by taking $p = N/2$

Next we show that $u(p) = v(p)$ for every $p \in \{1, N-1\}$ and thus $u(N/2) = v(N/2) = 1$.

Remark: The above facts (that are proved below) show that for every initial position p, there exists a real number $v(p) = u(p)$ such that if $d_L/d_R > v(p)$ then Left wins and if $d_L/d_R < v(p)$ then Right wins.

To prove the above facts, we introduce the following system of equations:

$$
\begin{aligned}
x_1 &= \tfrac{x_2}{1+x_2} \\
x_2 &= \tfrac{x_3+x_3x_1}{1+x_3} \\
&\vdots \\
x_p &= \tfrac{x_{p+1}+x_{p+1}x_{p-1}}{1+x_{p+1}} \\
&\vdots \\
x_{N-2} &= \tfrac{x_{N-1}+x_{N-1}x_{N-3}}{1+x_{N-1}} \\
x_{N-1} &= 1+x_{N-2}
\end{aligned}
\tag{1}
$$

The above system can be rewriten as follows:

$$
\begin{aligned}
x_2 &= \tfrac{x_1}{1-x_1} \\
x_3 &= \tfrac{x_2}{1-x_2+x_1} \\
&\vdots \\
x_{p+1} &= \tfrac{x_p}{1-x_p+x_{p-1}} \\
&\vdots \\
x_{N-1} &= \tfrac{x_{N-2}}{1-x_{N-2}+x_{N-3}} \\
x_{N-1} &= 1+x_{N-2}
\end{aligned}
\tag{2}
$$

Lemma 4.3 *(a)* $(v(1), v(2), \ldots, v(N-1))$ *satisfies the system of equations (1) (and the equivalent system of equations (2)).*

(b) $(u(1), u(2), \ldots, u(N-1))$ *satisfies the system of equations (1) (and the equivalent system of equations (2)).*

(c) $(1/v(N-1), 1/v(N-2), \ldots, 1/v(1))$ *satisfies the system of equations (1) (and the equivalent system of equations (2)).*

Proof: (a) We found that $v(1,t+1) = \frac{v(2,t)}{v(2,t)+1}$. If we take the limit as $t \to \infty$ of both sides of this equation, we obtain $v(1) = \frac{v(2)}{v(2)+1}$, which shows that $v(1)$ and $v(2)$ satisfy the first equation of the system above. Similarly,

$$v(p,t+1) = \tfrac{v(p+1,t)+v(p-1,t)v(p+1,t)}{v(p+1,t)+1}$$

which by taking the limit implies $v(p) = \frac{v(p+1)+v(p-1)v(p+1)}{v(p+1)+1}$.

This reasoning can be applied for all p other than $p=1$ (see above) and $p=N-1$, to be discussed below.

We check that $v(N-1)$ satisfies the last equation of the system. We take the limit as $t \to \infty$ of both sides of the following equation: $v(N-1,t+1) = 1 + v(N-2,t)$. This gives us that $v(N-1) = 1 + v(N-2)$, so clearly, $v(N-1)$ and $v(N-2)$ satisfy the last equation of the system.

(b) The sequences $u(p,t)$ satisfy the same recurrences as the sequences $v(p,t)$. Therefore, the same proof shows that the limit values $u(p)$ satisfy the sytem of equations (1).

(c) Similar to point(a), we have three cases to analyze: $p = 1, p = N-1$, and all other values of p between 1 and $N-1$.

1. $p = 1$. We must show that $\frac{1}{v(N-1)} = \frac{\frac{1}{v(N-2)}}{1+\frac{1}{v(N-2)}}$. This is equivalent to $v(N-1) = \frac{1+\frac{1}{v(N-2)}}{\frac{1}{v(N-2)}} = 1 + v(N-2)$, which is true by Lemma 4.3.

2. $p \neq 1, p \neq N-1$. We must show that $\frac{1}{v(N-p)} = \frac{\frac{1}{v(N-p-1)}+\frac{1}{v(N-p-1)}\frac{1}{v(N-p+1)}}{1+\frac{1}{v(N-p-1)}}$. This is equivalent, after calculations, to $v(N-p) = \frac{v(N-p+1)(1+v(N-p+1))}{1+v(N-p+1)}$, which is true by Lemma 4.3.

3. $p = N-1$. We must show that $\frac{1}{v(1)} = 1 + \frac{1}{v(2)}$. This is equivalent to $v(1) = \frac{v(2)}{v(2)+1}$, which is true by Lemma 4.3.

∎

Lemma 4.4 *The system of equations (1) (and the system of equations (2)) has a unique solution.*

Proof: We have shown that the system has a solution. It remains to show the uniqueness.

Suppose $(a_1, a_2, \ldots, a_{N-1})$ and $(b_1, b_2, \ldots, b_{N-1})$ are two distinct solutions to system (2). If $a_1 = b_1$, then from the form of the system (2) it can be immediately seen that $a_2 = b_2, \ldots, a_{N-1} = b_{N-1}$. So suppose $a_1 > b_1$. By induction, we show that $a_i > b_i$ and that $a_i - a_{i-1} > b_i - b_{i-1}$, for $2 \leq i \leq N-1$. This leads us to a contradiction because $a_{N-1} - a_{N-2}$ and $b_{N-1} - b_{N-2}$ are both equal to 1.

Base Case: First, we show $a_2 > b_2$. We have $a_2 = \frac{a_1}{1-a_1}$, and $b_2 = \frac{b_1}{1-b_1}$. Since $a_1 > b_1$, it follows that $b_2 > b_1$. Next, $a_2 - a_1 = \frac{a_1}{1-a_1} - a_1 = \frac{a_1^2}{1-a_1}$. By similar calculations, we get that $b_2 - b_1 = \frac{b_1^2}{1-b_1}$. Clearly, $\frac{a_1^2}{1-a_1} > \frac{b_1^2}{1-b_1}$, since $a_1 > b_1$. Thus, $a_2 - a_1 > b_2 - b_1$.

Induction Step: Suppose $a_k > b_k$ and $a_k - a_{k-1} > b_k - b_{k-1}$. We show that $a_{k+1} > b_{k+1}$ and $a_{k+1} - a_k > b_{k+1} - b_k$. We have $a_{k+1} = \frac{a_k}{1-(a_k - a_{k-1})}$ and $b_{k+1} = \frac{b_k}{1-(b_k-b_{k-1})}$. Since $a_k > b_k$ and $a_k - a_{k-1} > b_k - b_{k-1}$, we get that $a_{k+1} > b_{k+1}$. We also have

$$a_{k+1} - a_k = \frac{a_k}{1 - a_k + a_{k-1}} - a_k = \frac{a_k(a_k - a_{k-1})}{1 - (a_k - a_{k-1})}$$

and similarly, $b_{k+1} - b_k = \frac{b_k(b_k-b_{k-1})}{1-(b_k-b_{k-1})}$. Since $a_k > b_k$ and $a_k - a_{k-1} > b_k - b_{k-1}$, it follows that $a_k - a_{k-1} > b_k - b_{k-1}$.

End of Induction Proof.

Thus, the system of equations has a unique set of solutions. ∎

Now we can conclude the proof of the main assertion in this section.

Since the system of equations (1) has a unique solution, and $(v(1), \ldots v(N-1))$ and $(1/v(N-1), \ldots, 1/v(1)))$ both satisfy this system, it follows that $v(1) = \frac{1}{v(N-1)}, v(2) = \frac{1}{v(N-2)}, \ldots, v(N/2) = \frac{1}{v(N/2)}, \ldots, v(N-1) = \frac{1}{v(1)}$. In particular, $v(N/2) = 1/v(N/2)$ and thus $v(N/2) = 1$ (taking into account that $v(N/2) \geq 0$). The tuple $(v(1), \ldots v(N-1))$ also satisfies the system (1). Invoking again, the uniqueness of solutions, it follows that $u(N/2) = v(N/2) = 1$

5 Money To The Bank - Discrete Version

5.1 Introduction

In this version of the game, the players bid integer amounts of money (and also start with integer amounts of money), and the money goes to a bank instead of to the adversary. We were not able to find a simple formula that gives the winner of any configuration of the game (as we did in the continuous version of the game), but we have found an algorithm that determines the winner.

5.2 Algorithm for Determining the Winner

Lemma 5.1 *There are at most $d_L + d_R + N$ steps in the game.*

Proof: At each step of the game, at least one of the players will bid at least 1 dollar. This is so because if Left bids 0 dollars, then Right will bid 1 dollar (otherwise he would allow Left to move the token for free). There will be at most $d_L + d_R$ steps before both players have 0 dollars (if the game lasts that long). At that point, Left can move the token at each step until she has won the game, which requires at most N more steps.

5.2.1 Determining When Left Wins

We present a dynamic-programming algorithm that determines if Left wins the game starting in a given configuration, and, if this is the case, the minimum number of steps in which Left wins the game.

A configuration of the game is given by (p, m_1, m_2), where $p \in \{0, \ldots, N\}$ is the position of the token, $m_1 \in \{0, \ldots, d_L\}$ is the amount of dollars that Left has and $m_2 \in \{0, \ldots, d_R\}$ is the amount of dollars that Right has. We introduce the array L. L is a four-dimensional array that depends on p, the current position of the token, m_1, the current amount of money that Left has, m_2, the current amount of money that Right has, and t, the number of steps that are considered. We define

- $L[p][m_1][m_2][t] = s$ if Left wins the game in at most $s \leq t$ steps when the game starts in configuration (p, m_1, m_2) and
- $L[p][m_1][m_2][t] = 0$, if Left does not win the game in t steps starting in configuration (p, m_1, m_2).

We recursively determine $L[p][m_1][m_2][t]$ for all possible $t, 1 \leq t \leq d_L + d_R + N$, $p, 0 \leq p \leq N$, $m_1, 0 \leq m_1 \leq dL$, and $m_2, 0 \leq m_2 \leq dR$.

We first find the initial values $L[p][m_1][m_2][t]$ for $t = 1$. If $p = 1$ and $m_1 \geq m_2$, then $L[p][m_1][m_2][t] = 1$; in other words, if the token is at node 1 and Left has more money than Right, she can win in 1 step by bidding m_1. If either one of the two conditions are not true, then $L[p][m_1][m_2][t] = 0$.

The following algorithm calculates $L[p][m_1][m_2][t]$ for $t > 1$ based on the values $L[p][m_1][m_2][t-1]$.

```
for(p=1; p<= N-1; p++)
for(m1=0; m1<=dL; m1++)
for (m2=0; m2<=dR; m2++) {
// compute L[p][m1][m2][t]
if(L[p][m1][m2][t-1] != 0)   // Left wins in t-1 steps, so she wins in t steps
L[p][m1][m2][t]= L[p][m1][m2][t-1] ;
```

```
else {     // Left does not win in t-1 steps; see if she wins in t steps
good_move = 0;  //search for a good bid for Left so that she wins
k = 0;
while (!good_move && (k <= m1)) {
if (p==1){
if (k >= m2)
{ good_move = 1; L[p][m1][m2][t]= 1;}
else if (L[p+1][m1][m2-k-1][t-1] !=0)
{good_move=1; L[p][m1][m2][t] = 1 + L[p+1][m1][m2-k-1][t-1];}
else k++;
}
else if (p==N-1){
if ((k>=m2) && (L[p-1][m1-k][m2][t-1] != 0))
{good_move = 1; L[p][m1][m2][t] = 1 + L[p-1][m1-k][m2][t-1];}
else k++;
}
else if((L[p-1][m1-k][m2][t-1] != 0) && (k + 1 > m2)) {
good_move = 1;
L[p][m1][m2][t] = 1 + L[p-1][m1-k][m2][t-1];
}
else if ((L[p-1][m1-k][m2][t-1] != 0) && (L[p+1][m1][m2-k-1][t-1]!=0)) {
good_move=1;
if (L[p-1][m1-k][m2][t-1] >= L[p+1][m1][m2-k-1][t-1])
L[p][m1][m2][t] = 1 + L[p-1][m1-k][m2][t-1];
else
L[p][m1][m2][t] = 1 + L[p+1][m1][m2-k-1][t-1];
}
else
k++;  // try the next k as a possible good bid
}
if (good_move ==0)
L[p][m1][m2][t] = 0;
}
}
}
```

We provide the necessary explanations. If Left wins in $t-1$ steps, then she can clearly win in t steps as well. If Left does not win the game in $t-1$ steps, we must search for a *good bid* (which may or may not exist) that Left can make such that she wins in t steps. Let k denote Left's bid, and initially let $k = 0$. There are several possible situations:

1. Let $p = 1$. If $k \geq m_2$, then $L[p][m_1][m_2][t] = 1$. If Left does not have more money than Right, Right necessarily outbids Left (since Right would lose otherwise), and we must look at $L[p+1][m_1][m_2-k-1][t-1]$. If $L[p+1][m_1][m_2-k-1][t-1] \neq 0$, then we let $L[p][m_1][m_2][t] = 1 + L[p+1][m_1][m_2-k-1][t-1]$, because if Left can win in $s \leq t-1$ moves from $p = 2$ in

the present configuration of the game, she can also win in the previous configuration (at node 1) in $s+1$ steps. If $L[p+1][m_1][m_2-k-1][t+1] = 0$, then we increase k by 1 and repeat the process. If, by $k = m_1$, a good move still has not been found, we let $L[p][m_1][m_2][t] = 0$.

2. Let $p = N-1$. If $k \geq m_2$ and $L[p-1][m_1-k][m_2][t-1] \neq 0$ (in other words, if Left has more money than Right and, after outbidding Right to avoid losing the game, can win the game in $t-1$ steps), then we let $L[p][m_1][m_2][t] = 1 + L[p-1][m_1-k][m_2][t-1]$. If this is not the case, we increase k until both conditions are satisfied, and if this does not occur by the time $k = m_1$, we let $L[p][m_1][m_2][t] = 0$.

3. Let $p \neq 1$ and $p \neq N-1$. If $L[p-1][m_1-k][m_2][t-1] \neq 0$ (if by bidding k dollars, Left is able to win in $t-1$ steps from node $p-1$) and $k+1 > m_2$ (if Right is unable to outbid Left at node p), then we let $L[p][m_1][m_2][t] = 1 + L[p-1][m_1-k][m_2][t-1]$.

4. Let $p \neq 1$ and $p \neq N-1$. If $L[p-1][m_1-k][m_2][t-1] \neq 0$ and $L[p+1][m_1][m_2-k-1][t-1] \neq 0$ (in other words, if regardless of how Right responds to Left's bid of k at node p, Left can win from nodes $p-1$ or $p+1$ in $t-1$ steps), then we let $L[p][m_1][m_2][t] = 1 + \max(L[p-1][m_1-k][m_2][t-1], L[p+1][m_1][m_2-k-1][t-1])$.

5. If none of the above conditions are satisfied, increase k by 1 and repeat the procedure. If we have checked all k up through $k = m_1$ and have found no good bids, we let $L[p][m_1][m_2][t] = 0$.

5.2.2 Determining When Right Wins

We present a dynamic-programming algorithm, similar to the one in the previous section, that determines if Right wins, and, if this is the case, the minimum number of steps in which he wins.

We introduce the array R. R is a four-dimensional array defined similarly to L. We set

- $R[p][m_1][m_2][t] = s$, if Right can win in $s \leq t$ steps from configuration (p, m_1, m_2), and
- 0, otherwise.

We recursively determine $R[p][m_1][m_2][t]$ for all possible $t, 1 \leq t \leq d_L + d_R + N$, $p, 0 \leq p \leq N$, $m_1, 0 \leq m_1 \leq dL$, and $m_2, 0 \leq m_2 \leq dR$.

We first find the initial values $R[p][m_1][m_2][t]$ for $t = 1$. Right wins in 1 step if $p = N-1$ and $m_2 > m_1$ (in other words, if Right is one node away from winning and he has more money than Left), so in this case we let $R[p][m_1][m_2][t] = 1$. Otherwise, we let $R[p][m_1][m_2][t] = 0$.

The following algorithm shows how to calculate $R[p][m_1][m_2][t]$ for $t > 1$ based on the values $R[p][m_1][m_2][t-1]$.

```
for(p=1; p<=N-1; p++)
for(m1=0; m1<=dL; m1++)
for(m2=0; m2<=dR; m2++) {
//compute R[p][m1][m2][t]
if(!(R[p][m1][m2][old] == 0))
// If Right wins in t-1 steps, he also wins in t steps
R[p][m1][m2][cur]= R[p][m1][m2][old];
else {   // Right does not win in t-1 steps; we see if he wins in t steps
```

```
good_move = 0;
// good-move is true iff we find a good bid for Left (she wins or draws)
k = 0;
while (!good_move && (k <= m1)) {
if (p==1) {
if ((k >= m2) || R[p+1][m1][m2-k-1][old] == 0)
good_move = 1;
else k++;
}
else if (p== N-1) {
if ((k>=m2) && R[p-1][m1-k][m2][old] == 0)
good_move = 1;
else k++;
}
else if(  (R[p-1][m1-k][m2][old] == 0) && (k + 1 >= m2 || (R[p+1][m1][m2-k-1][old]==0)))
good_move = 1;
else
k++;
}
if (good_move ==0)
R[p][m1][m2][cur] = t;
else
R[p][m1][m2][cur] = 0;
}
}
}
```

We now explain the algorithm. If Right can win in $t-1$ steps, he can also win in t steps. If this is not the case, then Right can win in t steps from node p only if there are no *good bids* for Left such that Left has a winning or drawing strategy from node p. To find the existence (or lack of existence) of such good bids, we must analyze several cases. Let k denote Left's bid, and let $k = 0$ initially.

1. Let $p = 1$. If $k > m_2$ or $R[p+1][m_1][m_2 - k - 1][t-1] = 0$ (in other words, Right is able to outbid Left and move the token, but he cannot win from node $p+1$ in $t-1$ steps), then a good move for Left has been found, and we let $R[p][m_1][m_2][t] = 0$. Otherwise, we increase k by 1 and check the same criteria again.

2. Let $p = N - 1$. If $k > m_2$ and $R[p-1][m_1 - k][m_2][t-1] = 0$ (in other words, if Right cannot outbid Left and if he cannot win in $t-1$ steps from node $p-1$), then we have found a good move for Left, and we let $R[p][m_1][m_2][t] = 0$. Otherwise, we increase k by 1 and check the same criteria again.

3. Let $p \neq 1$ and $p \neq N - 1$. If $R[p-1][m_1 - k][m_2][t-1] = 0$ (Left outbids Right and Right cannot win in $t-1$ steps from node $p-1$) and either $k + 1 > m_2$ (Right cannot outbid Left) or $R[p+1][m_1][m_2 - k - 1][t-1] = 0$ (Right can outbid Left, but he cannot win in $t-1$ steps from

node $p+1$), then we have found a good bid for Left and we let $R[p][m_1][m_2][t] = 0$. Otherwise, we increase k by 1 and check the same criteria again.

4. If k reaches the value m_1+1 and we have still not found a good bid for Left, we set $R[p][m_1][m_2][t] = s$, where s is the current step.

References

[1] Lazarus, Loeb, Propp, Stromquist, and Ullman. Combinatorial games under auction play. *Games and Economic Behaviour*, 27:229–264, 1999. Available at `http://dept-info.labri.u-bordeaux.fr/~loeb/pull/a.html`.

[2] Lazarus, Loeb, Propp, and Ullman. Richman games. In *Games OF No Chance*, 1994. Available at `http://dept-info.labri.u-bordeaux.fr/~loeb/pull/a.html`.