



Minimal Risk Betting Analysis in Poker Tournaments



Uttam Bhetuwal^{1*}, Angel Chavez¹, Isaac Dobes¹

This paper focuses on minimal risk strategies for tournament style No-Limit Texas Hold'em. In order to obtain the most favourable outcome, we write linear programs which provide a set of constraints for a conservative player to follow so that a favorable outcome is obtained. We also model strategies by constructing stack functions, which map a strategy to its expected outcome. We prove various properties of stack functions, which provide evidence in support of the conjecture that the minimal risk strategy corresponds to the line segment connecting $(0, y_0)$ to (n, A) , where y_0 is the initial number of chips a player has in the tournament, and A is the desired amount of chips a player has at the end of the tournament.

INTRODUCTION

The game of poker is exciting for many and offers the potential for monetary gain. Nonetheless, the risk of losing large quantities of money deters many from taking part in the game. To mitigate risk, we analyze and model minimal risk strategies.

In this section, we first describe the rules of No-Limit Texas Hold'em (NLTH) and its implications in a tournament setting. Then we define risk and conservative play, and characterize the Zero Risk strategy.

After defining the Zero Risk strategy, we formulate the linear programs which provide a guideline for a player attempting to minimize risk. We then take a more theoretical approach and use various metrics to measure components of risk: aggressiveness, C^1 norms, and total variation.

The Rules of NLTH

No-Limit Texas Hold'em poker is an n player ($2 \leq n \leq 10$) game of incomplete information, that uses the standard 52 card playing deck. At the beginning of a hand, each player is dealt two cards. Player only have knowledge of cards dealt to them.

Once the players are dealt their two cards, two players are required to place bets, called the small blind and big blind bets. The big blind is always twice the amount of the small blind. Both blinds rotate around the table to different players after each hand to ensure all players will eventually be place a small and big blind. Once the big blind bets, a round of betting begins: players must either (1) call meaning they bet an equal number of chips as the player before them, (2) raise meaning they bet more chips then the player before them, (3) fold meaning they do not bet any chips and forfeit out of the hand, they also lose any chips they have bet in the hand, or (4) check meaning they defer the decision to call, bet, or fold, a decision must be made eventually. A round of betting is complete once every player calls, and no player desires to raise. Note that for the small blind to continue playing must also call or raise the big blind (or subsequent raises of the big blind); if they fold, then they lose the small blind they were forced to bet in the beginning. After this round of betting is complete, three community cards are placed face up on the table and similar rounds of betting proceed. After the rounds of betting are complete, the player with the best five card hand then wins the pot, which is the accumulation of all chips bet from the beginning of the hand. The best five card hand (two personal cards and three community cards) is determined by the usual poker rules.

NLTH Tournament Rules

In a NLTH tournament, many (sometimes over a thousand) players at different tables of nine each start with the same amount of chips and continuously play hands until they either run out or are the last player remaining, in which case they win the tournament and accept a cash prize. To facilitate betting and hasten the game, blind bets increase after a certain period of time. The period of time in which the blinds are a fixed amount is called a level, when the time period is over the players move into a new level and the blinds increase. In the beginning of the tournament, after a certain number of levels, an ante bet is introduced The ante bet re-

Address correspondence to:

¹Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, United States

*uttambhetuwal@gmail.com



doi:10.22186/jyi.39.9.27-33



Submission date: April 2020

Acceptance date: September 2020

Publication date: September 2021

Except where otherwise noted, this work is licensed under <https://creativecommons.org/licenses/by/4.0>



quires that all players bet a specified number of chips at the beginning of every hand; the blind bets still apply. The ante bets never decrease after a level, and like the blind bets, can and often increases after a level.

Risk and Conservative Strategies

Definition 1

Risk per hand is defined as the probability of losing the hand multiplied by the value bet in the hand. For a given hand i , the risk of hand i is denoted by $r(i)$.

Remark: The blind and ante amounts are not accounted for in this definition of risk; since they are mandatory bets they are considered sunk costs.

Definition 2

A strategy is a sequence of decisions allowable to a player in the game. The **overall risk of a strategy** is the expected value of the risk of the strategy, given by

$$\sum_{i=1}^N r(i)$$

where N denotes the total number of hands played in the duration of the strategy.

Definition 3

A **conservative strategy** is a strategy which attempts to minimize risk.

The Zero Risk Strategy

In the Zero Risk strategy, a player checks when possible, and folds otherwise. The only time a player bets chips is when they are a blind, or if an ante bet is required by all players. Since blind and ante bets are sunk costs, in this strategy, a player voluntarily bets exactly 0 chips, incurring a risk of 0 for every hand; hence the name.

THE LOSS FUNCTION

In this section, we derive the loss function, which calculates how many chips a player will have at the end of a level, n , utilizing the Zero Risk strategy.

First Order Linear Difference Equations

To derive the loss function, we use the following theorem. The proof of the theorem can be found in Jensen (2011).

Theorem 4

Let a_n and d_n , $n \in \mathbb{N}_0$, be real valued sequences. Then the linear first order difference equation

$$y_n = b_{n-1} \cdot y_{n-1} + d_{n-1}$$

with initial value $y_0 \in \mathbb{R}$ has the unique solution

$$y_n = \left(\prod_{k=0}^{n-1} a_k \right) y_0 + \sum_{k=0}^{n-1} \left(\prod_{j=k+1}^{n-1} a_j \right) d_k.$$

Derivation of Loss Function

Since we assume that we lose every hand we bet and we only bet the required amount via blind and ante, we can use the following recursive equation to find the number of chips left at the end of level n :

$$y_n = y_{n-1} - h \cdot a_n - f_n^s \cdot s_n - 2 \cdot f_n^b \cdot s_n$$

where h is the number of hands played in a level (this is assumed to be a constant), a_n is the ante amount in level n , s_n is the small blind amount in level n (note that for any level n , the big blind amount in level n is twice the small blind amount in level n), f_n^s is the small blind frequency in level n (that is, how many times we are the small blind), and f_n^b is the big blind frequency in level n (that is, how many times we are the big blind). The terms on the right hand side of the equation, excluding y_{n-1} , make up the term c_{n-1} in the recursive function in Theorem 4.

Applying Theorem 4, we have the formula:

$$y_n = y_0 - \sum_{k=1}^n c_k$$

where $c_k = h \cdot a_k + f_k^s \cdot s_k + 2 \cdot f_k^b \cdot s_k$. Note that this formula calculates the amount of chips left at the end of level n by subtracting the cost of the n levels from the initial amount of chips a player begins with.

Position and Blind Frequency

Definition 5

For any $n \in \mathbb{N}$, φ_n := initial position in level n with respect to the small blind.

That is, if a player has $\varphi_n = 0$, this means that they start level n as the small blind; if a player has $\varphi_n = k$, $k \in [1, \dots, 8]$, then they are k hands dealt away from being the small blind.

Remark: If a player is currently the small blind, then they will be the big blind in the next hand. The player who is the big blind is now 8 hands away from being the small blind again.

Proposition 6

$$\varphi_n = \varphi_1 - (n - 1)r \pmod{9}$$

Proof: To prove the proposition, we use induction on the level n .

Base Case ($n = 1$): $\varphi_1 = \varphi_1 - (1 - 1) \cdot r \pmod{9}$.

Induction Hypothesis: Suppose that for $k \in \mathbb{Z}$, $\varphi_k = \varphi_1 - (k - 1)r \pmod{9}$.

Induction Step: When $n = k + 1$,

$$\begin{aligned} \varphi_1 - kr \pmod{9} &= \varphi_1 - [(k - 1) + 1]r \pmod{9} \\ &= \varphi_1 - (k - 1)r + r \pmod{9} \\ &= \varphi_k + r \pmod{9}, \text{ by the induction hypothesis} \\ &= \varphi_k + 1 \pmod{9} \end{aligned}$$



By division algorithm $h = 9q + r$, for some $q, r \in \mathbb{Z}$ with $0 \leq r < 9$. This indicates that position, from one level to the next, is solely determined by the remainder.

Proposition 7

Let $r = h \pmod{9}$, ϕ_1 be a player's initial position at the beginning of the tournament, and ϕ_n be the player's initial position at the beginning of level n . Then the following are true:

1. if $\phi_n < r$, then $f_n^s = \lfloor h/9 \rfloor$, otherwise $f_n^s = \lfloor h/9 \rfloor$
2. if $\phi_n < 8 + r \pmod{9}$ or $\phi_n = 8$, then $f_n^b = \lfloor h/9 \rfloor$

Proof:

1. Consider the case where 9 is a divisor of h . Then there exists $k \in \mathbb{N}$ such that $h = 9k$ which implies that $r = 0$. It is clear that each player is the small blind $k = h/9 = \lfloor h/9 \rfloor = \lfloor h/9 \rfloor$ many times.

If 9 is not a divisor of h , then by the division algorithm, $h = 9k + r$, $0 < r < 9$. $9k$ hands are played and thus every player is the small blind k times. After the $9k^{\text{th}}$ hand, r hands remain to be played. The player with $\phi_n = 0$ and the players with $\phi_n \leq r - 1$ will have the opportunity to be the small blind once more; they are the small blind $k + 1$, or equivalently $\lfloor h/9 \rfloor$, many times. Players with $\phi_n > r - 1$ will not be able to play as the small blind again in the level; they are the small blind a total of k or $\lfloor h/9 \rfloor$ times.

2. Consider the case where 9 is a divisor of h . Then there exists $k \in \mathbb{N}$ such that $h = 9k$, which implies that $r = 0$. Therefore every player will play each position (including position 8) exactly $k = h/9 = \lfloor h/9 \rfloor = \lfloor h/9 \rfloor$ times.

If 9 is not a divisor of h , then by the division algorithm, let $h = 9k + r$, $0 < r < 9$. $9k$ hands are played and thus every player is the big blind k times. After the $9k^{\text{th}}$ hand, r hands remain to be played. The player who started as position 8 will be the big blind again after $9k$ hands, so the player with $\phi_n = 8$ will be the big blind $k + 1$ or $\lfloor h/9 \rfloor$ times. Now there are $r - 1$ remaining hands to be played in the level. For players that started with initial positions $r - 1$ or fewer hands away from the big blind, they will also be the big blind $k + 1$ or $\lfloor h/9 \rfloor$ times. By definition of position, these are players with initial position $\phi_n < r - 1 = 8 + r \pmod{9}$. All of the other players won't have the chance to become big blind again. As a result, they only get to be big blind $k = \lfloor h/9 \rfloor$ times.

Since the number of hands per level, h , is a constant, in conjunction with Proposition 7, the only input for the function required by the user is the betting structure, which in practice will be known to the player prior to entering the tournament. Therefore, if a player is utilizing the Zero Risk strategy in a tournament, they can use our function to determine how long they will last in the tournament.

INTEGER LINEAR PROGRAM (ILP)

While minimizing risk to zero may sound optimal, this notion

is refuted in practice. In reality, a player utilizing the Zero Risk Strategy will often last only a few levels (about 5-7 using the World Series of Poker betting structure). This is why it is necessary to introduce risk and voluntary betting. Nonetheless, a player may still want to minimize risk subject to the following constraint: survive to the level n with at least A chips, $n, A \in \mathbb{N}$. We refer to such a constraint as a **survival constraint**. We have tried and tested different iterations of ILP to get a minimal risk betting strategy.

Minimum Hand ILP

To satisfy the survival constraint, one method we have developed is to minimize the number of hands played in a tournament. Let w_i be the number of hands a player wins in level i , v_i be the minimum value of winning a hand in level i ($v_i = 9 \cdot a_i + 3 \cdot s_i$, as each player is required to bet the ante and the big blind is twice the small blind), and l_i be the amount of chips lost in a hand in level i :

$$l_i = \begin{cases} a_i + 2s_i & \text{if } \phi_i = 8 \\ a_i + s_i & \text{if } \phi_i = 0 \\ a_i & \text{otherwise} \end{cases}$$

For programming purposes, we approximate l_i to be $a_i + 1/3s_i$. We therefore formulate the Minimum Hand ILP as follows:

$$\min \sum_{i=1}^n w_i \tag{1}$$

$$\text{s.t. } \sum_{i=1}^n [w_i \cdot v_i - l_i \cdot (h - w_i)] + y_0 \geq A \tag{2}$$

$$\sum_{i=1}^k [w_i \cdot v_i - l_i \cdot (h - w_i)] + y_0 > 0 \quad \forall k \in [n] \tag{3}$$

$$w_i \in \mathbb{Z}_{\geq 0} \quad \forall i \in [n] \tag{4}$$

Note that the survival constraint for this LP is given in lines 2 and 3. In subsequent linear programs, the LP's survival constraint to these two lines. Also, the vector $\mathbf{w} \in \mathbb{Z}^n$ consisting of the components w_i , $1 \leq i \leq n$, is a solution vector if it satisfies the survival constraints.

Issues with the Minimum Hand ILP

Due to the betting structure of tournaments, winning a hand in a later level results in a greater payoff than winning a hand in an earlier level. Therefore, to minimize the required number of hands won in n levels subject to the survival constraint, the above ILP will dictate that a player wins most of the hands in the latter levels, clustering these hands in "close" proximity. This is problematic because playing with greater frequency gives more information to other play-

ers. This decreases the players' probability of winning the hand, thus increases the overall risk. Furthermore, requiring a player to play hands in close proximity limits what hands the player can choose to play, potentially forcing the player to play worse hands, thereby reducing their probability of winning the hand, which increases risk. Since the Minimum Hand ILP clusters the required hands won in later levels and does not require the player to win hands in earlier levels, the player is in a later level and forced to win all hands played. They may not have enough chips to bet and call opponents, making it difficult if not impossible to satisfy the survival constraints.

Uniformly Distributed ILP

To prevent clustering of required hands won, we develop another integer linear program which determines the smallest number of required hands won for a level w^* , so that if exactly w^* hands are won each level, then a player would satisfy the survival constraints. We call the value w^*/h the minimum ratio, since it is the ratio of the smallest number of required hands won per level to the number of hands dealt in a level. We formulate below what we refer to as the Uniformly Distributed Linear Program (UDLP), which we claim (and later prove) solves the problem of finding w^* .

$$\begin{aligned} \min \quad & w_1 \\ \text{s.t.} \quad & w_1 = w_i \quad \forall i \in [n] \\ & \sum_{i=1}^n [w_i \cdot v_i - l_i \cdot (h - w_i)] + y_0 \geq A \\ & \sum_{i=1}^k [w_i \cdot v_i - l_i \cdot (h - w_i)] + y_0 > 0 \quad \forall k \in [1, \dots, n] \\ & w_i \in \mathbb{Z}_{\geq 0} \quad \forall i \in [n] \end{aligned}$$

Minimum Ratio Proposition

Since $w_i = w_1$ for all $i \in [n]$, this LP has the corresponding solution vector $\mathbf{w} \in \mathbb{Z}^n$ with each component having the same value.

Proposition 8

Given the survival constraints for n levels in a tournament, the UDLP returns w^* .

Proof: Suppose the UDLP has the corresponding solution vector $\mathbf{w} \in \mathbb{Z}^n$, where each component has the value $w_1 \in \mathbb{Z}$. Assume for the sake of contradiction that there exists a solution vector $\mathbf{v} \in \mathbb{Z}^n$, such that $\max_{1 \leq i \leq n} v_i < w_1$. Let $\tilde{\mathbf{v}} \in \mathbb{Z}^n$ be the vector with every $1 \leq i \leq n$ component having the value $\max_{1 \leq i \leq n} v_i$. This satisfies all of the constraints in the UDLP, which contradicts the objective of the UDLP, since w_1 is not minimal.

Minimum Ratio ILP

Let the corresponding solution vector to the UDLP be $\mathbf{w} \in \mathbb{Z}^n$,

with each component having the value w^* . There may exist a solution vector satisfying the survival constraints and having maximum (component-wise) value w^* , while having other components with smaller values. Below we formulate an integer linear program whose solution corresponds to the solution vector consisting of the smallest of such components.

$$\begin{aligned} \min \quad & \sum_{i=1}^n w_i \\ \text{s.t.} \quad & w_i \leq w^* \quad \forall i \in [n] \\ & \sum_{i=1}^n [w_i \cdot v_i - l_i \cdot (h - w_i)] + y_0 \geq A \\ & \sum_{i=1}^k [w_i \cdot v_i - l_i \cdot (h - w_i)] + y_0 > 0 \quad \forall k \in [n] \\ & w_i \in \mathbb{Z}_{\geq 0} \quad \forall i \in [n] \end{aligned}$$

Notice the similarities between the Minimum Ratio ILP and the Minimum Hand ILP. Because of the objective of the LP and the constraint $w_i \leq w^*$ for all i , we reduce clustering in latter levels, which is the primary defect of the minimum hand LP. Furthermore, since the Minimum Ratio ILP has an upper bound on the number of hands a player can win in a level, it requires that the player win hands in earlier levels, which prevents the player's chips stack from falling too close to zero. This ensures that the player will have sufficient chips to place bets in latter levels.

Example

To illustrate the deficiencies of the Minimum Hand ILP and the superiority of the Minimum Ratio ILP, we ran an example. We attempted to use realistic numbers in our example, with an initial chip amount of $y_0 = 25000$ and a desired ending chip amount of $A = 100000$. We wanted to last 8 levels with 63 hands per level. The betting structure of this example is as follows: antes are $a_i = 0$ if $i \leq 3$ else $25 \cdot 2^{i-2}$, blinds are $s_i = 10 \cdot 2^{i-2}$. With this information we ran the Minimum Hand ILP which returned

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
0	0	0	0	1	6	7	11

Table 1. A solution to Minimum Hand ILP.

Running the UDLP gave $w^* = 15$. We then ran the Minimum Ratio ILP, which returned

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
0	0	0	0	2	9	9	9

Table 2. A solution to Minimum Ratio ILP.

Notice that there is a greater clustering of hands in the later levels in the Minimum Hand ILP. Even though the Mini-

imum Ratio ILP requires that we win more hands, since these 12 hands are more evenly distributed throughout all levels, we do not risk having our chip stack falling too close to zero and not being able to survive later levels.

Consider the chip stack of a player following the Minimum Hand ILP strategy it is straight forward to show the chip stack at the end of the levels is (0, 25000), (1, 24811), (2, 24370), (3, 23551), (4, 15550), (5, 2144), (6, 605), (7, 7717), and (8, 102953). We can graph this data.

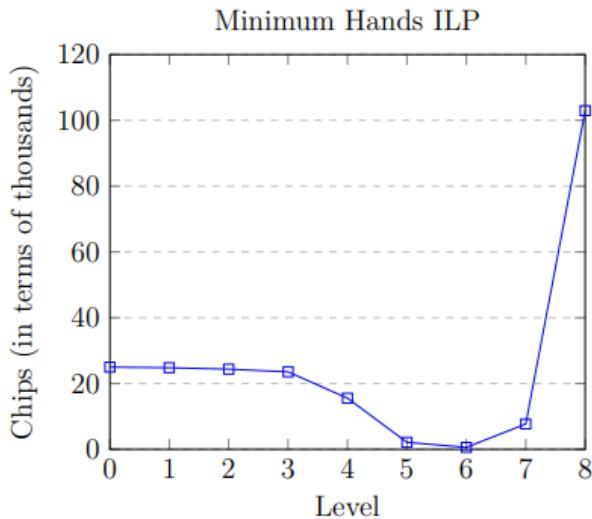


Figure 1.

Consider the chip stack of a player following the Minimum Ratio ILP strategy. It is straight forward to show the chip stack at the end of the levels is (0, 25000), (1, 24811), (2, 24370), (3, 23551), (4, 15550), (5, 4677), (6, 18339), (7, 45717), and (8, 100419). We can graph this data.

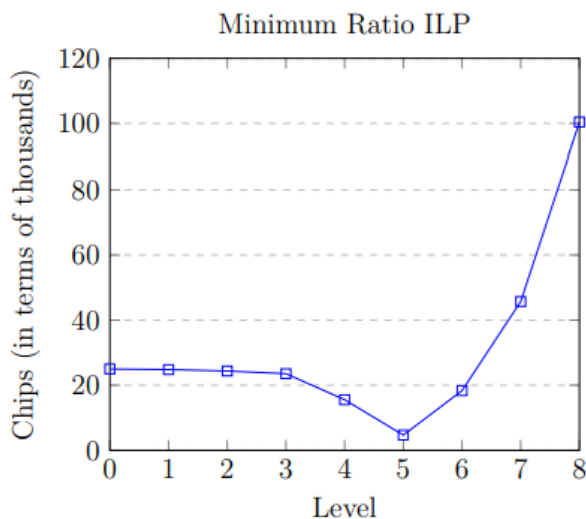


Figure 2.

STACK FUNCTIONS

Recall that a strategy is a sequence of decisions. Here, a strategy is defined on the number of hands, nh , in a tournament (there are h hands per level, and n levels). Therefore, a strategy is a sequence of betting decisions per hand. At each hand, the outcome of the hand is a random variable $X: \Omega \rightarrow Z$ defined by $\omega \mapsto \alpha$, where $\omega \in \Omega$ is a point in the sample space of possible betting amounts, and α is the amount a player has left at the end of the hand. The outcome of a strategy is the stochastic process $\{X_t; t \in [nh]\}$; that is, the outcome of the strategy is the sequence of random variables mapping the outcome of a given hand to a chip amount. The expected outcome of a strategy S , $E(S)$ is the sequence $\{E(X_t) : t \in [nh]\}$, where E is the expected value operator. The **discrete stack function** maps a strategy to its expected outcome. Hence, a discrete stack function will return a set of nh points corresponding to expected chip amounts at given hands based on the decisions made in the strategy.

Polynomial Interpolation

Given a discrete stack function S defined on n levels (and therefore nh hands), we use Lagrange Interpolation to construct a smooth curve that best fits the outputs of the discrete function.

Theorem 9

MAT 772 (2011); *Mayers and Suli (2003)* Let x_0, x_1, \dots, x_n be $n + 1$ distinct numbers, and let $f(x)$ be a function defined on a domain containing these numbers. Then the polynomial defined by

$$p_n(x) = \sum_{j=0}^n f(x_j) \mathcal{L}_{n,j}(x)$$

is the unique polynomial of degree n that satisfies $p_n(x_j) = f(x_j)$, $j = 0, 1, \dots, n$, where

$$\mathcal{L}_{n,j}(x) = \prod_{k=0, k \neq j}^n \frac{x - x_k}{x_j - x_k}$$

The polynomial $p_n(x)$ is called the interpolating polynomial of $f(x)$.

Given a strategy S with the corresponding discrete stack function S_1 , we construct the strategy's **continuous stack function** S_2 by applying the above theorem. As we consider hand 0 to be the beginning of the tournament where every player has y_0 many chips, we define $f(x)$ on $[0, \dots, nh]$ and $f(x) = S_1$. Since continuous stack functions are polynomials, they are continuously differentiable on their domain $[0, nh] \subseteq \mathbb{R}$. The function space $C^1([0, nh])$ is the set of all continuously differentiable functions defined on $[0, nh] \subseteq \mathbb{R}$. Hence it contains all continuous stack functions under consideration.

Aggressiveness

Definition 10

Tao (2008) The function space $C^1([0, nh])$ is equipped with the C^1 norm

$$\|f\|_{C^1 \infty ([0, nh])} := \|f\|_{L^\infty ([0, nh])} + \|f'\|_{L^\infty ([0, nh])}$$

where $\|f\|_{L^\infty ([0, nh])} := \sup\{|f(x)| : x \in [0, nh]\}$ is the supremum norm.

The norm measures the magnitude of a stack function and the magnitude of its derivative. The magnitude of a stack function's derivative is called the **aggressiveness** of a stack function. It measures the maximum magnitude of chips a player bet and won or bet and lost in a tournament.

Theorem 11

The stack function characterized by the line segment y_0A is the continuous stack function which minimizes aggressiveness among all stack functions satisfying the survival constraints. We make the assumption that such this stack function exists since we are trying to determine the ideal, not necessarily practical, minimal risk strategy.

Proof: Let S be a stack function which exactly meets the survival constraints. Let y_0A be the line segment connecting y_0 to A . By the Mean Value Theorem, there exists z in the domain of S such that $S'(z) = A - y_0 / n - 0$. If $S(z) \notin y_0A$, then there exists $\epsilon > 0$ such that

$$|S'(z - \epsilon)| > \left| \frac{A - y_0}{n - 0} \right| \text{ or } |S'(z + \epsilon)| > \left| \frac{A - y_0}{n - 0} \right|$$

Hence, $\max_{x \in \text{dom}(S)} \{|S'(x)|\} > |A - y_0 / n - 0|$. Therefore, y_0A minimizes aggressiveness among all possible strategies which exactly meet the survival constraints.

Corollary 12

The stack function characterized by the line segment y_0A is the continuous stack function which minimizes the C^1 norm of all stack functions which exactly meet the survival constraints.

The corollary follows immediately from the theorem since all stack functions exactly meeting the survival constraints must contain the endpoints $(0, y_0)$ and (nh, A) , which are the maximum and minimum values of the line segment y_0A respectively.

Total Variation

While aggressiveness is a decent metric for analyzing a player's most ambitious period during the game, the overall betting pattern is not accounted for as it only measures the maximum magnitude of the derivative at a single point. To get a better picture of the player's strategy, we introduce the

total variation function that measure a player's overall betting pattern.

Definition 13

Let f be a differentiable function defined on the interval $[a, b]$. The **total variation** of f from a to b is given by

$$V_b^a(f) = \int_a^b |f'(x)| dx.$$

We interpret the total variation as a measure of the overall amount bet in a strategy. While it does not explicitly give the betting amount, it provides a metric for comparing stack functions. Some nice properties of total variation are $V_b^a(f) = V_b^a(-f)$ and $V_b^a(f) = V_b^a(f + c)$ for any function f and any constant $c \in \mathbb{R}$. We use $V(f)$ to denote the total variation of f on the domain of f .

Theorem 14

Let $A = (x_a, y_a), B = (x_b, y_b) \in \mathbb{R}^2$ with $x_a < x_b$. Let AB be the line segment connecting A to B . Let f be a differentiable function defined on $[x_a, x_b]$ with $f(x_a) = y_a$ and $f(x_b) = y_b$. Then

$$V(f) \geq V(AB).$$

Proof: Case 1: $y_a \leq y_b$. We first calculate $V(AB)$. Since $|AB| = y_b - y_a / x_b - x_a$ then $V(AB) = y_b - y_a$. We now bound $V(f)$

$$\begin{aligned} V(f) &\geq \int_{x_a}^{x_b} f'(x) dx \\ &\geq f(x) \Big|_{x_a}^{x_b} \\ &\geq y_b - y_a \\ &\geq V(AB) \end{aligned}$$

Case 2: $y_a > y_b$. We reflect AB and f on the $y = y_a$ line. The reflection of f is $f^r(x) = 2y_a - f(x)$, and the reflection of AB is $AB^r(x) = 2y_a - AB(x)$. Similarly, $|AB^r| = y_a - y_b / x_b - x_a$, which implies that $V(AB^r) = y_a - y_b$.

$$\begin{aligned} V(f^r) &\geq \int_{x_a}^{x_b} (f^r)'(x) dx \\ &\geq f^r(x) \Big|_{x_a}^{x_b} \\ &\geq y_a - y_b \\ &\geq V(AB^r) \end{aligned}$$



By properties of total variation, $V(AB) = V(AB^c)$ and $V(f) = V(f^c)$, which implies that $V(AB) \leq V(f)$.

Corollary 15

The stack function characterized by the line segment y_0A minimizes total variation of all stack function exactly meeting the survival constraints.

Connection to Risk

Given a stack function P , increasing the aggressiveness of P increases the risk of P since the value bet is greater. The C^1 norm contains the aggressiveness and the magnitude of the stack function. The greater the magnitude of a stack function, the more chips are bet and won or bet and lost to get to that critical point, which implies that a stack function P with greater C^1 norm incurs greater risk. Similarly, if P has greater total variation, P acquires greater risk as more chips were bet. Since the stack function corresponding to the line segment y_0A minimizes aggressiveness, the C^1 norm, and total variation, we have the following conjecture:

Conjecture 16

The strategy corresponding to the stack function characterized by the line segment y_0A minimizes risk.

CONCLUSION

We started this paper with concrete and practical results. However, these results not ideal the Zero Risk strategy minimizes risk, but it is not a feasible long term strategy. The linear programs provide a criterion for conservatively play, that is too survive longer in a tournament but not necessarily end with more chips. The linear programs do not completely minimize risk, and do not provide any insight on how to win. Since the concrete results were not ideal, we analyzed betting patterns in a more abstract setting by approximating the expected outcomes of strategies with continuous stack functions. While this analysis is not as concrete and not fully concluded, it is promising. Since strategies are finite sequences and there are only finitely many decisions a player can make, it is theoretically possible to iterate through all possible strategies and find the expected outcome of each strategy. If our conjecture is correct, then the strategy which best fits the line segment y_0A would be the optimal minimal risk strategy.

ACKNOWLEDGMENTS

We would like to thank Carnegie Mellon University and the Summer Undergraduate Applied Mathematics Institute (SUAMI) program for hosting and funding our research. In particular, we would also like to thank Dr. Winger.

REFERENCES

Jensen, A. (2011). Lecture notes on difference equations. Available: <http://people.math.aau.dk/matarne/11-imat/notes2011a.pdf>.
 MAT 772, F. -. L. . N., Fall Semester. (2011). The gnats and gnu document preparation system. <http://www.math.usm.edu/lambers/mat772/fall10/lecture5.pdf>.

Mayers, D., & Suli, E. (2003). An introduction to numerical analysis. Cambridge University Press.
 Tao, T. (2008). Function spaces. https://terrytao.files.wordpress.com/2008/03/function_spaces1.pdf.