Probabilistic Optimization of Table Tennis Matches

Mingchung Xia

Page Count: 17

Introduction

Table tennis is an Olympic racket sport in which two opposing competitors return a ball back and forth over a net-separated table. A full *match* of table tennis consists of any number of *games*,[1] where the competitors play until some number of points to win each game. Originally, each game was played until 21 points for a competitor to win, where each match was a 5-game 3-win format. However, after the 2000 summer Olympics in Sydney, the International Table Tennis Federation (ITTF) amended a new set of regulations for the point scoring system of table tennis — an 11-point system, in a 7-game 4-win format. The reasoning of this point system change was disclosed as a way to make table tennis more entertaining and suitable as a television sport. The shorter games, paired with more opportunities for advertisement breaks, appeared to be justifiable logic for this change. Nonetheless, what went unnoticed by many was the influence that this change would have on the competitors of table tennis. As each competitor of table tennis is always unequal in terms of skill level, their win probability for each point of a game is fundamentally different. Hence, the altercation of the point system for each game impacts the probabilistic nature of a match. The aim of this exploration is to model a table tennis match through a system of a set number of games and points, to optimize the probability of the weaker competitor winning a match. Probability distributions, multivariable calculus, programming, and simulation comprise the tools that are applied to this exploration.

*This investigation omits the table tennis rule that each game must be won by 2 points. i.e, 11-10 is a completed game.

[1] Matches are not to be confused with games. Matches are composed of some number of games. Each game is played to some number of points.

Modelling a Table Tennis Match

In a table tennis match, two opposing competitors differ in terms of skill level, each having a different win probability of a game. Instead of looking at the whole match, how do we model each singular game that is played? Let there be two competitors: Competitor A, and Competitor B. We know that if Competitor A wins a particular point in a game, Competitor B cannot win that particular point. The opposite holds true. Therefore, let the event that Competitor A wins a point have p probability of occuring. In a game, the probability sample space U of each point only contains two events, where either Competitor A or Competitor B wins. These events are mutually exclusive. By definition, if the occurrence of p is the event that Competitor A wins a point, then its complementary event that Competitor B wins the point is $q = 1 - p$. This satisfies that the probability P has the property that $P(p \cup q) = 1, \therefore p + q = 1$. This is used in probability distributions. Hence, we can examine if a table tennis game agrees with the conditions that could satisfy it to be a binomial distribution, by letting random discrete variable X represent the number of points to be won by Competitor A. Firstly, each point played is a Bernoulli trial, meaning that it has exactly two random outcomes: Competitor A wins or Competitor B wins.[2] Secondly, each of the point outcomes are independent of one another. Thirdly, each point has a constant probability p (for Competitor A).[3] Therefore, if we model table tennis game as a binomial distribution f, defined by parameters t points per game and probability p as $X \sim B(t, p)$, we have:

---

[2] Hu, Yan. *Research on the impact of competition system on badminton development based on curve fitting*. Trade Sciences Inc., 2014.

[3] Wathall, Jennifer, Harcet, Josip, Harrison, Rose, Heinrichs, Lorraine, and Torres-Skoumal, Marlene. *Mathematics: Analysis and Approaches Higher Level Course Companion.* Oxford University Press, 2019.

$$f(X = k) = C^{t}_{k} p^{t} q^{t-k},\ k \in [0, t].$$

However, the differing factor is the consideration that there is not a fixed number *t* points that each game is always played to, to be considered completed. In reality, a winner can be decided before all *t* total points available in a game have been played. For example, a game has a total of $t = 21$ points, then a winner is decided when a competitor first wins 11 points. Accordingly, the outcome of the remaining points are irrelevant.

Alternatively, we must find another way to model a probability distribution. Examining a table tennis game, let *n* denote the number of points won in order for a competitor to win, $n \in \mathbb{N}$. Therefore, the minimum points played in a game is *n*, and the maximum points played in a game is $2n - 1$ (this is explained in Section 3). If we let *k* denote the number of points that are played to in a game, then $k \in [n,\ 2n - 1]$. Competitor A with probability *p* to win a point must succeed *n* times within *k* points played — in a game, this probability is p^{n}. Likewise, Competitor B with probability *q* to win a point must succeed $k - n$ times within *k* points played — in a game, this probability is q^{k-n}. Together, the two probabilities occurring in a game is $p^{n} C^{k-n}_{k-1} q^{k-n}$. The combination C^{k-n}_{k-1} is present because there are various possibilities that Competitor A and B can win *n* points. However, Competitor B (q) cannot choose from *k* points because the k^{th} point must be won by Competitor A (p), so q is chosen from $k - 1$ points. If the probability of a game is modelled by $f(p)$, it is the sum within the domain of *k*. Thus, we have:

$$f(p) = \sum_{k=n}^{2n-1} p^{n} C^{k-n}_{k-1} q^{k-n}$$

and factoring constant p^{n},

$$f(p) = p^n \sum_{k=n}^{2n-1} C_{k-1}^{k-n} q^{k-n}$$

To tidy the function, let the iterative variable i be $i = k - n$. This allows for $i \in [0, n - 1]$, and $k = i + n$. Therefore, we have:

$$f(p) = p^n \sum_{i=0}^{n-1} C_{n-1+i}^{i} q^{i}$$

Since f is symmetrical about $p = 0.5$, it is more concise to restrict the domain D of f to be $D: p \in [0.5,\ 1]$. Any $p \in [0,\ 0.5]$ would mirror the domain, which would reverse the probabilities p and q of Competitors A and B. For $p \in [0.5,\ 1]$, we can see that f is a model for the stronger competitor. This is Competitor A, who has a higher win probability than Competitor B such that $p \geq q$, $q = 1 - p$, $\because\ q \in [0, 0.5]$.

When we consider a whole table tennis match, we can see that a match is an identical scenario to a game. Each game has exactly two outcomes: Competitor A wins or Competitor B wins. Each game is independent of one another. A match result can be determined before all available games have been played. Lastly, there exists a constant probability of outcome in each game. Specifically, we can say that a match is simply a composition of game probabilities, likewise to how a game is a composition of point probabilities. For example, if we consider Competitor A with a point win probability p and game win probability $f(p)$, and we let F be a Competitor A's match win probability, the probability distribution of a match would then be the composition of $F(f(p))$. In this case, F is the same formula as shown above because it is the same scenario, except with $f(p)$ as the constant probability. Therefore, we have:

$$F(f(p)) = (f(p))^m \sum_{j=0}^{m-1} C_{m-1+j}^{j} (1 - f(p))^j$$

In this function, m represents the number of games won in order for a competitor to win. When $f(p)$ is substituted, F is:

$$F(f(p)) = (p^n \sum_{i=0}^{n-1} C^i_{n-1+i} q^i)^m \sum_{j=0}^{m-1} C^j_{m-1+j} (1 - p^n \sum_{i=0}^{n-1} C^i_{n-1+i} q^i)^j$$

Given that we seek to model an optimization of a table tennis match through a set point-game system, we have previously defined n to denote the number of points won in order for a competitor to win a game, and m to denote the number of games won in order for a competitor to win a match. We will use (n, m) to represent this pair of $(points\ to\ win, games\ to\ win)$, where $n, m \in \mathbb{N}$. Logically, it can be hypothesized that the more total points a match is played to, the less probable the weaker competitor is able to win. Conversely, this implies that the less total points a match is played to, the more probable the weaker competitor is able to win. This minimum must be $(n, m) = (1, 1)$, as this is the lowest possible natural number pair. Next, consider the current table tennis regulations: $n = 11$ as it requires 11 points to win a game, and $m = 4$, as it requires 4 games to win a match; this pair is $(11, 4)$. If this is reversed, we have another $(n, m) = (4, 11)$. Lastly, for comparison purposes, consider the original table tennis rules, which would have $(n, m) = (21, 3)$. These point-game systems of different pairs of (n, m) have probability distribution F as shown in Figure 1. For example, for the pair $(11, 4)$, the game probability for all $p \in [0.5, 1]$ is

$$f(p) = p^{11} \sum_{i=0}^{10} C^i_{10+i} (1 - p)^i$$

If $p = 0.5$, then:

$$f(0.5) = (0.5)^{11} \sum_{i=0}^{10} C^i_{10+i} (1 - 0.5)^i = 0.5$$

This shows that for a game, if Competitor A has point win probability $p = 0.5$, Competitor A has a game win probability of 0.5. Considering a match with $p = 0.5$, we calculated that $f(0.5) = 0.5$. Using the function shown above, and $m = 4$ from $(11, 4)$,

$$F(f(0.5)) = (f(0.5))^4 \sum_{j=0}^{3} C_{3+j}^{j} (1 - f(0.5))^j = 0.5$$

This algorithm is repeated to calculate all values of *F* within the domain of $D: p \in [0.5, 1]$. This was done using Python 3.8.3 and the graphing library `matplotlib` to visualize *F* in Figure 1. The code and raw data can be found in Section 7 - Appendix. In Figure 1, it can be also seen that if $p = 0.5$, $F(f(p)) = 0.5$.

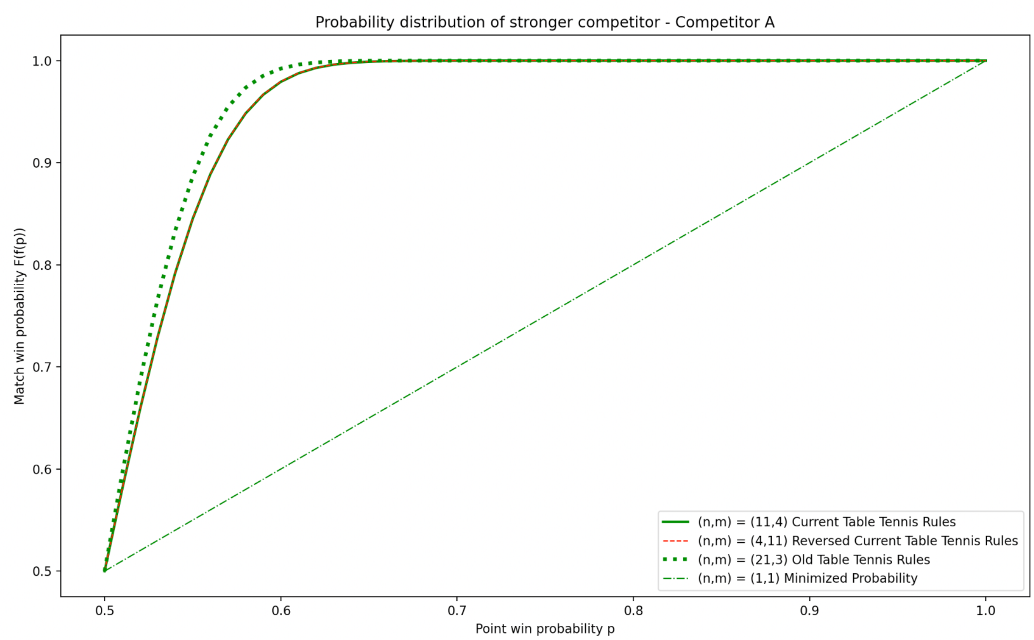

Figure 1 - Probability distribution $F(f(p)$ for various combinations of (n, m) for all $p \in [0.5, 1]$

Within this distribution of *F* with various (n, m) pairs in Figure 1, several conjectures can be noted. Since probabilities p within the domain $D: p \in [0.5, 1]$ are the probabilities of the stronger competitor (Competitor A), it can be seen that with the old table tennis rules, the match favours Competitor A, relative to the current table tennis rules. This is deduced because for lower values of p, there are higher values of F, that is, there is a higher match win probability. The second conjecture is that the case of $(n, m) = (11, 4)$, which is the point-match system of the current table tennis regulations, produces the same curve F as the case of $(4, 11)$, which is the 'reverse'. Therefore, this symmetry shows that both competitors A and B have unaffected win probabilities by inverses in the pair (n, m), such that the probabilities P have the property such that $P(n, m) = P(m, n)$. Lastly, the most prominent conjecture is that the plot of F is not arbitrarily affected by (n, m), but specifically the product of n and m. Not only is this shown in the symmetrical nature that $P(n, m) = P(m, n)$, $nm = mn$, but also when the logic of the table tennis match is considered. If each game is played to $n = 11$ points, and a match is won by first to $m = 4$ games, then there must be a minimum of mn total points that are won throughout the match. Whenever this condition is minimized, then the game would favour the weaker competitor. As correlated to Figure 1, the lower product approaches the probability curve where $(n, m) = (1, 1)$. At this curve, Competitor A has the lowest probability distribution, implying that the weaker competitor (Competitor B) has the most optimized probability to win a match. Logically speaking, the worse competitor will have their best probability of winning when there are fewer points and games played. Due to the multiplicative-iterative nature of probability, greater successions will only decrease the probability of an event occurring — in this case, the weaker competitor winning a game or a match. This further justifies why the aforementioned condition is aimed to be minimized. The next section (Section 3 - Probabilistic Optimization) deals with this optimization for any general total point product of (n, m), based on these conjectures.

Probabilistic Optimization

In the case of our table tennis probabilistic modelling, there must be some function that has parameters as the point system of a match. If we consider the modern table tennis rules with a 11-win 4-game format, $(n, m) = (11, 4)$, we also know that the maximum total points available are the case when each game has scores 10-11 (21 total points), and the game results are 3-4 (7 total games). Generally, let N denote the total points available per game, $N \in \mathbb{N}$, and let M denote the total games available per match, $M \in \mathbb{N}$. Alternatively, if we know that there are N total points available each game, and M total games available each match, we know that $n = \lceil \frac{N}{2} \rceil$, and $m = \lceil \frac{M}{2} \rceil$. This is because a competitor wins a game when they win just over half of the total points available. A match is the same case. Knowing that the ceiling function always evaluates the input to the nearest greater integer, therefore, we know that $N = 2n - 1$, and $M = 2m - 1$. Now, let T represent the total points played in a match. As discussed in Section 2, we deduced that the minimum total points played could be $T = nm$. Conversely, the maximum total points played would be $T = NM$. Hence, the function that models the point system of a match is T (maximum). We can define T to be a multivariable function of n and m such that $T(n, m) = NM = (2n - 1)(2m - 1)$. In multivariable calculus, a *partial derivative* evaluates the derivative of a multivariable function. Each partial derivative of some multivariable function examines the rate of change of the function by some change in each of the input variables. The function being partially differentiated with respect to a variable treats all other variables as constants.[4] If a multivariable function f is differentiable at all points, its gradient is denoted by nabla ∇f, which is a vector. For example, an input to a two-variable function

[4] For example, taking the partial derivative of $f(x,y) = xy$ with respect to x would treat y as a constant, vise versa.

generates a vector plane, which, in a 3-dimensional space, is a plane that describes the gradient at that input.[5] We can use partial derivatives to optimize multivariable function T at the point for which the gradient of T is a zero vector at a local minimum, that is:

$$\nabla T(n,m) = \begin{bmatrix} \frac{\partial}{\partial n}T \\ \frac{\partial}{\partial n}T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Firstly, we can expand $T(n, m)$

$$T(n,m) = 4nm - 2n - 2m + 1$$

Taking the partial derivative of T with respect to n and m:

$$\frac{\partial}{\partial n}(4nm - 2n - 2m + 1) = 4m - 2$$

$$\frac{\partial}{\partial m}(4nm - 2n - 2m + 1) = 4n - 2$$

Hence, the vector ∇T is

$$\nabla T(n,m) = \begin{bmatrix} 4m-2 \\ 4n-2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Setting each vector component equal to zero yields a simple system of equations:

$$4m - 2 = 0$$

$$4n - 2 = 0$$

yields that $m = \frac{1}{2}$ and $n = \frac{1}{2}$. Finding T at these values of m and n:

$$T(\frac{1}{2}, \frac{1}{2}) = (2(\frac{1}{2}) - 1)(2(\frac{1}{2}) - 1) = 0$$

so function T has a gradient of zero at $(\frac{1}{2}, \frac{1}{2}, 0)$. How can we verify that this is a local minimum? Finding the second partial derivative serves as the subsequent step.

[5] Strang, Gilbert, and Edwin "Jed" Herman. "Partial Derivatives," Mathematics LibreTexts, Massachusetts Institute of Technology (Strang) & University of Wisconsin-Stevens Point (Herman). Last Updated: May 28, 2021. Accessed: Nov. 10, 2021 https://math.libretexts.org/@go/page/2602.

Taking the second partial derivative of T with respect to n and m:

$$\frac{\partial^2}{\partial n^2}(4m - 2) = 0$$

$$\frac{\partial^2}{\partial m^2}(4n - 2) = 0$$

Given that the second derivative of T is zero with respect to both n and m, it is inconclusive, as it could be a local minimum, local maximum, or a saddle point.[6] However, in a table tennis match, the range of T must be non-negative because there cannot be negative total points in a match. As $T \geq 0$, and exactly one zero-gradient was evaluated at $(\frac{1}{2}, \frac{1}{2}, 0)$ where $T(n, m) = 0$, this point must also be a local minimum. At the local minimum, it is evident that $n = m = \frac{1}{2}$; when T is optimized at a zero gradient, $T(n, m) = (2n - 1)(2m - 1) = (2n - 1)^2 = (2m - 1)^2$. Now, solving for n (and m), $n = m = \frac{\sqrt{T}+1}{2}$. This implies that for any T total points in a table tennis match, the optimized pair (n, m) to minimize the probability that the stronger competitor (Competitor A) to win is $(\frac{\sqrt{T}+1}{2}, \frac{\sqrt{T}+1}{2})$. Alternatively, this pair represents a point-game system of a table tennis match, when the weaker competitor (Competitor B) has their most optimized probability of winning. Using current table tennis rules of $(n, m) = (11, 4)$, there is $T(11, 4) = 147$ total points available. With our result, Competitor B should have an optimized match win probability at $(\frac{\sqrt{147}+1}{2}, \frac{\sqrt{147}+1}{2})$, realistically rounded to the nearest integers because $n, m \in \mathbb{N}$. In the next section, we will simulate a table tennis match with various combinations of (n, m) to simulatively verify if the weaker competitor (Competitor B) truly has an optimized match win probability at $(\frac{\sqrt{T}+1}{2}, \frac{\sqrt{T}+1}{2})$.

[6] In multivariable calculus, *Saddle Points* are points of a function which have zero slope in all orthogonal directions (imagine a zero-slope/flat plane in 3-dimensions), but is not an extremum.

Computational Simulation

In order to evaluate the optimization of a point-game system of a table tennis match, we will use graphical and statistical analysis of a simulation. In any given match, there must be a total point value *T*, which can be arbitrarily chosen. Considering the realistic nature of sports, *T* should not exceed a value that is deemed 'too high' or 'too low'. For calculation purposes, we will let $T = 225$. 225 was chosen due to the following properties: it is an odd positive integer, it is a perfect square, it has multiple factor pairs, and it is sufficiently large for a match. This will allow for straight-forward calculations.

As defined earlier, the maximum point total in a match is modeled by $T = NM$, $N, M \in \mathbb{N}$. Again, N is the total games available per match, and M is the total points available per game. Since *T* is a constant of 225, we can see that N and M are simply factor pairs of 225. The factor pairs (N, M), of 225 are each an element of the set *Z*:

$$Z = \{(1, 225), (3, 75), (5, 45), (9, 25), (15, 15)\}$$

Now, from each factor pair $(N, M) \in Z$, we can also calculate point-game pairs (n, m). Let *g* be an operator such that $g(\lambda) = (\lceil \frac{N}{2} \rceil, \lceil \frac{M}{2} \rceil), \forall\, \lambda = (N, M),\ \lambda \in Z$, because we know that $n = \lceil \frac{N}{2} \rceil$, and $m = \lceil \frac{M}{2} \rceil$. g produces a new set of pairs in a new set *S* that denotes the table tennis match point-game system:

$$S = \{g(\lambda) \colon (N, M) \in Z\} = \{(1, 113), (2, 38), (3, 23), (5, 13), (8, 8)\}$$

In *S*, $(n, m) = (8, 8)$ is the point-game system for which the weaker competitor (Competitor B) has an optimized probability to win. As computed, for $(8, 8)$, $n = \lceil \frac{15}{2} \rceil = \frac{\sqrt{225}+1}{2} = 8$,

$m = \lceil \frac{15}{2} \rceil = \frac{\sqrt{225}+1}{2} = 8$. Creating a probability distribution for $p \in [0.5, 1]$ using $F(f(p)) = (f(p))^m \sum_{j=0}^{m-1} C^j_{m-1+j}(1 - f(p))^j$ as previously defined in Section 2, for each $(n, m) \in S$, Figure 2 can be produced with `matplotlib`.

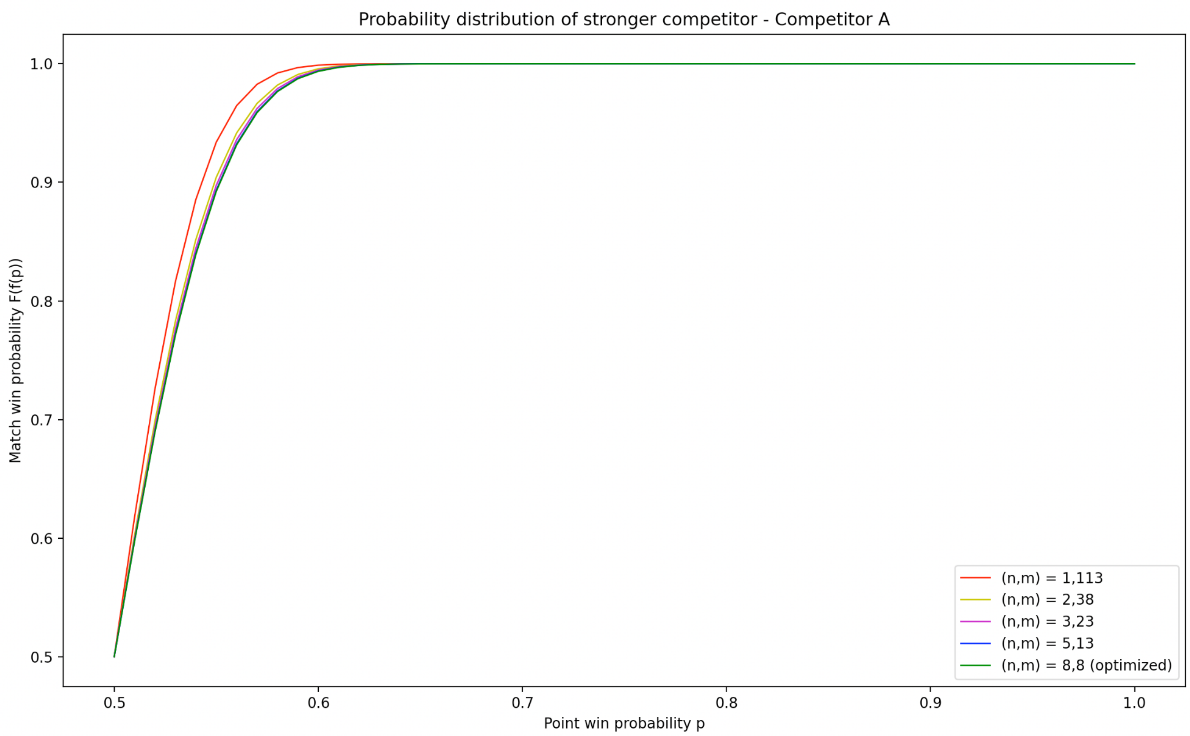

Figure 2 - Probability distributions $F(f(p)$ for various combinations of (n, m) for all $p \in [0.5, 1]$ where $T = 225$

Observationally, asymptotic behaviour can be observed as each curve (approximately) approaches values of $p > 0.65$. Accordingly, this implies that Competitor A almost has a guaranteed match-win probability when they have a point win probability greater than (approximately) 0.65. Thus, we should restrict the domain again. $p \in [0.6, 0.65]$ can display a clearer graph in Figure 3. We use the lower bound $p = 0.6$ and not 0.5 to simply show clearly noticeable differences between each curve.

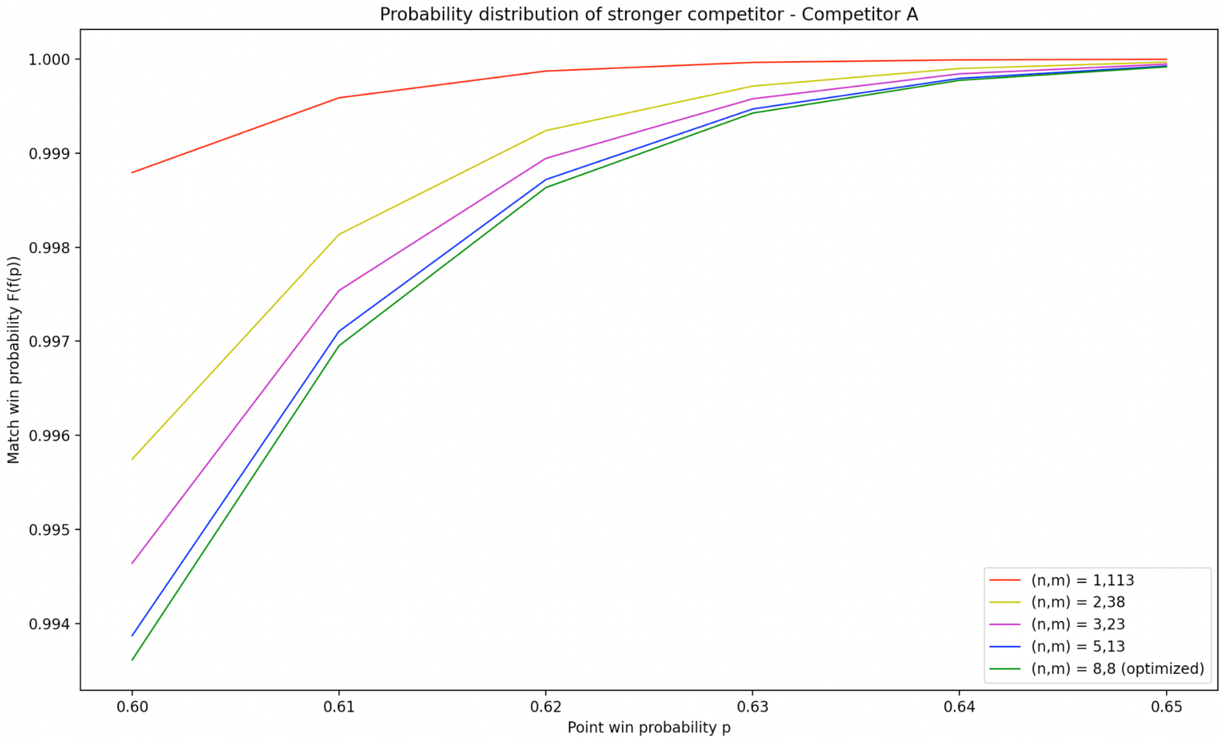

Figure 3 - Probability distributions $F(f(p)$ for various combinations of (n, m)

for all $p \in [0.6, 0.65]$ where $T = 225$

The green curve represents the point-game system of $(n, m) = (8, 8)$ as shown in the legend. Mathematically, this was deduced to be the table tennis system that would optimize the match-win probability of Competitor B. This is evident in the graph, where the green curve is lower than all other pairs of $(n, m) \in S$. The probability distribution suggests that Competitor A has the lowest probability to win a match under the system that $(n, m) = (8, 8)$. Therefore, Competitor B's match-win probability is simultaneously optimized.

To simulate a table tennis match using Python 3.8.3 under the conditions of: set S, $T = 225$, and $p \in [0.5, 1]$, allows for the generation of datasets. This would model the wins of the weaker competitor. This code can be found in the Section 7 - Appendix. For this domain, each $(n, m) \in S$ set the point-game system for a simulated match. As we denote Competitor B to be the weaker competitor, the simulation counted the number of occurrences B would win a match, for all probability p in its domain. The simulation was run 1000 times, thus each average occurrence is the average of 1000 data points. The table (Figure 4) is the data generated from the simulation. Data is presented to four significant figures because that was the maximum amount present when the data was generated.

(n, m) **Pair**	**Average Occurrence of** B **for** $p \in [0.5, 1]$
$(1, 113)$	2.530
$(2, 38)$	2.737
$(3, 23)$	2.729
$(5, 13)$	2.780
$(8, 8)$	2.863

Figure 4 - Tabular data generated from a table tennis simulation

As shown in Figure 4, it is evident that Competitor B was able to win the most matches for all its probabilities $q = 1 - p$, $q \in [0, 0.5]$ when the point-game system was set to $(n, m) = (8, 8)$. Compared to the other point-game systems, Competitor B was able to win a match with the optimized system (8, 8): 2.98% more often than (5, 13), 4.91% more often than (3, 23), 4.60% more often than (2, 38), and 13.16% more often than (1, 113).

Sample calculation: $\frac{|2.863 - 2.780|}{2.780} \times 100\%$ is the relative occurrence of B with (8, 8) more often than (5, 13).

This result can be seen in Figure 3, where the curve of $(n, m) = (1, 113)$ is far greater than the minimized $(8, 8)$ as it deviates furthest from our optimized $(n, m) = (\frac{\sqrt{T}+1}{2}, \frac{\sqrt{T}+1}{2})$. Similar visuals can be observed for other points relative to $(8, 8)$.

Conclusion

Table tennis, a seemingly simple sport between two competitors, has a deep probabilistic nature when considering the system of points and games in a match. In this exploration, it became evident that these changes in the match would alter the expected win-probabilities of competitors, shown through probability distributions and a simulation. By optimizing a multivariable function, it could be deduced that $(n, m) = (\frac{\sqrt{T}+1}{2}, \frac{\sqrt{T}+1}{2})$ could produce a match most favoured for the weaker competitor to win. With this, further exploration into this research could investigate the opposite — modelling a table tennis match to optimize the win probability of the stronger competitor. This could allow for a probabilistic analysis in both spectrums of table tennis. In addition, this exploration omitted the rule that a table tennis game must be won by 2 points. This would undoubtedly affect the probability distribution of the game as the amount of points to win a game is not a constant. A deeper investigation into this aspect of a table tennis game could produce vastly different, yet more realistic and applicable results. Unfortunately, as sports are influenced by many more factors than mathematical principles, limitations to include these variables does not allow for a perfect model of sporting matches. This exploration, though focused on table tennis, comprises an investigation that can be applied to other scenarios. Other sports, not limited to: volleyball, badminton, and tennis, all bear the same fundamental nature as a table tennis match — from a mathematical standpoint. Despite both positive and negative implications involved with this contrivance, ultimately, it shows the significance of regulations that are often overlooked. By understanding the effect of point-game systems on probability distributions, regulators are able to model matches that favour a competitor, or neither competitor. In practice, research into the underlying mathematics behind a deceivingly simple concept serves as an imperative focus to improve the design of conflicts, not limited to the optimization of sports, but the optimization of the world of mathematics around us.

Works Cited

Hu, Yan. *Research on the impact of competition system on badminton development based on curve fitting*. Trade Sciences Inc., 2014.

Wathall, Jennifer, Harcet, Josip, Harrison, Rose, Heinrichs, Lorraine, and Torres-Skoumal, Marlene. *Mathematics: Analysis and Approaches Higher Level Course Companion.* Oxford University Press, 2019.

Strang, Gilbert, and Edwin "Jed" Herman. "Partial Derivatives," Mathematics LibreTexts, Massachusetts Institute of Technology (Strang) & University of Wisconsin-Stevens Point (Herman). Last Updated: May 28, 2021. Accessed: Nov. 10, 2021 https://math.libretexts.org/@go/page/2602.

Appendix

Computing System	Description	Specification
Computer and Operating System	Graphing and simulation conducted on a MacBook Pro (2017 13 in.).	Version: 12.01 macOS Monterey Processor: 3.1 GHz Dual-Core Intel Core i5 RAM: 16 GB 2133 MHz LPDDR3
Integrated Development Environment (IDE)	Graphing and simulation conducted on one IDE.	IDE: Visual Studio Code version 1.47.3 Commit: 91899dcef7b8110878ea59626991a18c8
Programming Language	Graphing and simulation performed using one programming language.	Programming Language: Python 3.8.3 64-bit

Graphing Figure 1

```
from math import factorial
import matplotlib.pyplot as plt

def c_n_m(total_count, valid_count):
   fact_n    = factorial(total_count)
   fact_m    = factorial(valid_count)
   fact_n_m  = factorial(total_count - valid_count)
   c_n_m_num = fact_n / (fact_m * fact_n_m)
   return c_n_m_num

def main():
   p   = 0.5
   q   = 1 - p
   pg  = 0
   qg  = 1 - pg
   pm  = 0
   qm  = 1 - pm
```

```
pl   = [i/100 for i in range(50,101,1)]

ptgm = [
        (11, 4,'-','g',2,'(n,m) = (11,4) Current Table Tennis Rules'),
        (4, 11,'--','r',1,'(n,m) = (4,11) Reversed Current Table Tennis Rules'),
        (21, 3,':','g',3,'(n,m) = (21,3) Old Table Tennis Rules'),
        (1,  1,'-.','g',1,'(n,m) = (1,1) Minimized Probability')
        ]

file = open("/Users/XXXX/Desktop/Coding/Math IA/Math IA Data/mathiadata.csv","w")

file_header = "p"
for p in pl:
    file_header += "," + str(p)
file.write(file_header)

for n,m,ls,col,lw,lb in ptgm:
    pgl = []
    pml = []

    for p in pl:
        q = 1 - p

        a = 0
        for i in range(n):
            a += c_n_m(n-1+i,i) * (q**i)

        pg = (p**n) * a
        qg = 1 - pg
        pgl.append(pg)

        b = 0
        for j in range(m):
            b += c_n_m(m-1+j,j) * (qg**j)

        pm = (pg**m) * b
        pml.append(pm)

    file_line = f"\nn={n}"
    for p in pgl:
        file_line += "," + str(p)
```

```
        file.write(file_line)

        file_line = f"\nm={m}"
        for p in pml:
            file_line += "," + str(p)
        file.write(file_line)

        plt.plot(pl, pml, linestyle=ls, color=col, linewidth=lw, label = lb)
        plt.xlabel("Point win probability p")
        plt.ylabel("Match win probability F(f(p))")
        plt.title("Probability distribution of stronger competitor - Competitor A")

    file.write("\n")
    file.close()

    plt.legend()
    plt.show()

if __name__ == '__main__':
    main()
```

Graphing Figure 2 and Figure 3

```
from math import factorial
import matplotlib.pyplot as plt

def c_n_m(total_count, valid_count):
    fact_n     = factorial(total_count)
    fact_m     = factorial(valid_count)
    fact_n_m   = factorial(total_count - valid_count)
    c_n_m_num = fact_n / (fact_m * fact_n_m)
    return c_n_m_num

def main():
    p    = 0.5
    q    = 1 - p
    pg   = 0
    qg   = 1 - pg
    pm   = 0
    qm   = 1 - pm

    pl    = [i/100 for i in range(50,66,1)]
```

```python
    ptgm = [
            (1, 113,'-','r',1,'(n,m) = 1,113'),
            (2,  38,'-','y',1,'(n,m) = 2,38'),
            (3,  23,'-','m',1,'(n,m) = 3,23'),
            (5,  13,'-','b',1,'(n,m) = 5,13'),
            (8,   8,'-','g',1,'(n,m) = 8,8 (optimized)')
            ]

    for n,m,ls,col,lw,lb in ptgm:
        pgl = []
        pml = []

        for p in pl:
            q = 1 - p

            a = 0
            for i in range(n):
                a += c_n_m(n-1+i,i) * (q**i)

            pg = (p**n) * a
            qg = 1 - pg
            pgl.append(pg)

            b = 0
            for j in range(m): #M
                b += c_n_m(m-1+j,j) * (qg**j)

            pm = (pg**m) * b
            pml.append(pm)

        plt.plot(pl, pml, linestyle=ls, color=col, linewidth=lw, label = lb)
        plt.xlabel("Point win probability p")
        plt.ylabel("Match win probability F(f(p))")
        plt.title("Probability distribution of stronger competitor - Competitor A")

    plt.legend()
    plt.show()

if __name__ == '__main__':
    main()
```

Simulating and Generating Figure 4

```
import math
import numpy as np
import random
import csv

def save(results):
   with open("/Users/XXXXX/Desktop/Coding/Math IA/Math IA
Data/ttgamesim.csv","w",newline="") as file:
       writer = csv.writer(file)
       writer.writerows(results)

def main():
   x = 1000
   B_results = [0,0,0,0,0]

   for i in range(x):
       probabilityA = [i for i in range(50,101)]

       tg_tm = [(1,225),(3,75),(5,45),(9,25),(15,15)]

       results = []
       i = 0

       for pgsystem in tg_tm:
           n = math.ceil(pgsystem[0]/2)
           m = math.ceil(pgsystem[1]/2)

           pointsA = 0
           pointsB = 0
           gamesA  = 0
           gamesB  = 0

           B_count = 0

           for p in probabilityA:
               playing = True
               while playing:
                   if random.randint(1,101) < p:
                       pointsA += 1
```

```
            else:
                pointsB += 1

            if pointsA == n:
                gamesA += 1
                pointsA = 0
                pointsB = 0
                if (gamesA or gamesB) == m:
                    if gamesA > gamesB:
                        winner = 'A'
                    else:
                        winner = 'B'
                        B_count += 1
                    result = [p, n, m, winner]
                    results.append(result)
                    playing = False
                    gamesA = 0
                    gamesB = 0
                    break

            elif pointsB == n:
                gamesB += 1
                pointsA = 0
                pointsB = 0
                if (gamesA or gamesB) == m:
                    if gamesA > gamesB:
                        winner = 'A'
                    else:
                        winner = 'B'
                        B_count += 1
                    result = [p, n, m, winner]
                    results.append(result)
                    playing = False
                    gamesA = 0
                    gamesB = 0
                    break

        B_results[i] += B_count
        i += 1

    #save(results)
```

```
    B_results = [i/x for i in B_results]
    print(B_results)

if __name__ == "__main__":
    main()
```