IEOR 4530 Project: Modeling a Powerlifting Meet as a Game

Matheus Gramel
Industrial Engineering and Operations Research
Columbia Engineering
New York, NY
mg4379@columbia.edu

Abstract

The goal of this paper is to model a powerlifting meet as a game and use tools from game theory to find optimal weight selection for attempts. First, different ways to design a powerlifting meet as a game will be covered by thinking of how feasible and realistic each implementation is. Then, we will define a probability measure for how likely a lifter is to successfully lift a certain weight. Finally, once the probability measures for two lifters are defined, we can find the optimal weight selection for each by calculating an equilibrium.

Structure of a Powerlifting Meet

In a powerlifting meet, lifters have to complete three repetitions each of squat, bench press, and deadlift. The lifts have to happen in that order: first, three squats are attempted, then similarly for bench and deadlifts last.[1]

Generally, competitors who are in the same weight class will be in the same *flight*, which means that everyone in a given *flight* does their first attempts for a lift, then after everyone is done they chose a weight for the second attempt, and similarly, once everyone is done with their second attempts, weights are chosen for the third and last attempt.

Weight selection for each lift starts at 25kg and then can be increased any 2.5kg increment. Once the weight a specific exercise is chosen, further attempts have to be at that weight or higher, even if it is not successfully completed.[1]

The winner of a given weight class is whatever lifter has the highest *total*, which is defined as the sum of the heaviest weight successfully completed for each one of the three lifts.[1]

Some Ideas For How to Model a Meet as a Game

The general structure of the game will rely on the idea of weight selection by defining a probability measure for how likely a lifter is to successfully complete a lift at each weight. The way this is defined has to include some *public* information, which is some data that are available for both lifters. They will both have access to their own probability distribution as well as their opponent's distribution. This will then allow them to both optimize and reach an equilibrium in weight selection.

Assumptions and Simplifications

Modeling all three lifts is a hard task. Let's suppose we decide to start modeling the competition at the first squat. One might naively assume that each lifter should optimize to have the highest squat, but this is not the case, since they are optimizing for the highest total. This means that they would also have to take into account what probability distribution for the bench press and deadlift are too. For this reason, we will start modeling the competition only at the deadlift. This will greatly simplify the problem, since at this point the weights from the squat and bench press that will contribute to the total have already been determined, so we only have to worry about optimally choosing the deadlift that will lead to the highest chance of having the highest total.

Another useful simplification is modeling this as a two player game only. It is true that a weight class has more than two competitors, but in practice, a meet usually comes down to two lifters fighting for first place. By the time squats and bench press are done, it is very easy too see who still has a chance in the deadlift, and this will usually only be the top two lifters. The goal of both lifters is not to necessarily lift the most weight possible, but have the highest total relative to other competitors in your weight class, so you win first place.

As previously described, weights can only be chosen in 2.5kg increments starting at 25kg. This means we only have to define a discrete probability measure, which is how likely a lifter is to successfully each weight in 2.5kg increments. This is easier than if we were to have a continous space where any positive weight can be chosen.

Another simplification is that weights for an attempt are determined at the same time. In a powerlifting competition, that is not exactly true since you tell the referee your next attempt right after completing your lift, even if your opponents haven't gone yet. However, you are allowed to change your attempts, so we assume that they can both give the referee a *dummy* attempt, and then when both lifters finish lifting give their referee their real attempts.

Defining a Probability Measure

Training Data

The first idea for defining a probability measure was looking at training data. It is common for competitive powerlifters to share their training on social media, so by looking at their recent numbers from training, you could possibly define a good probability measure. The big advantage of this approach is that, given perfect training data, it would most likely generate the most realistic and accurate predictions.

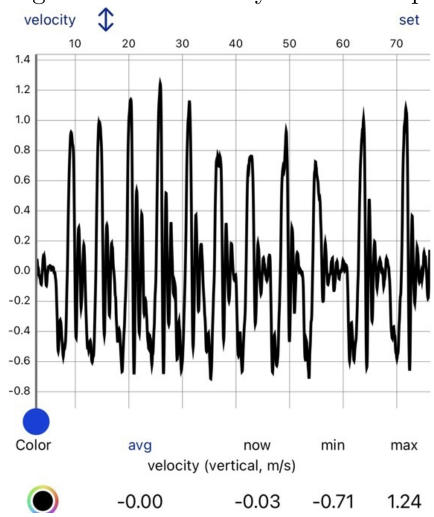
This approach, however, runs into a few issues, particularly with possible asymmetries in the data. It is possible for one lifter to not share their training numbers or even omit some training data, violating the condition we had previous set that all the data should be public and perfect. Considering some powerlifitng specific factors, how someone's training is structured can also affect the numbers they succeed in competition relative to training, possibly making training data unreliable.

Bar Velocity

The second idea considered involved looking at bar velocity. There are many applications that track the velocity of the barbell, which use velocity as one of the main factors in tracking how someone's training is progressing. One example of such an app is shown in Figure 1 below. The slower it is, the less likely someone is to successfully lift a higher weight. The idea would involve looking at the velocity, and then probabilities of success of other weights are set relative to the previous weight attempted and its velocity.

The main issue with this approach is the difficulty of data collection on competitions that already happened. While most competitions have livestreams, most apps that track velocity require very specific angles which cannot be retrieved after the fact. There are also external equipment that can be attached to the barbell to accurately track speed, such as the Vitruve Encoder, but we still run into the issue of data collection. Additionally, during a meet you may not be able to collect the data in real time, so you won't be able to do the optimization for both lifters in real time.

Figure 1: Bar Velocity Data Example



Historical Data

The final approach is to define the probability measure used historical data. USA Powerlifting (USAPL), the biggest drug-tested American powerlifitng federation, hosts a national level competition every year called Raw Nationals. Data for Raw Nationals (as well as every other USAPL competition, all the way from local to national level meets), are easily available online in a public database. The idea is to look at competitors who participated in multiple competitions, measure their improvement over a period of time, and then define a probability measure of how much someone is likely to improve over time. This is the approach that was chosen for the project, and will be explained in more details in the next section.

Data Collection and Analysis

As described previously, the data collection was based on looking at lifters in Raw Nationals in 2023[2] and Raw Nationals in 2024[3]. There 969 lifters competing in 2023, 951 lifters competing in 2024, and there was an overlap of 288 lifters who competed in both years. From this point on, it is important to note that Raw Nationals is comprised of experienced lifters, who most likely have many years of training, so the data might be different when looking at beginner or intermediate lifters.

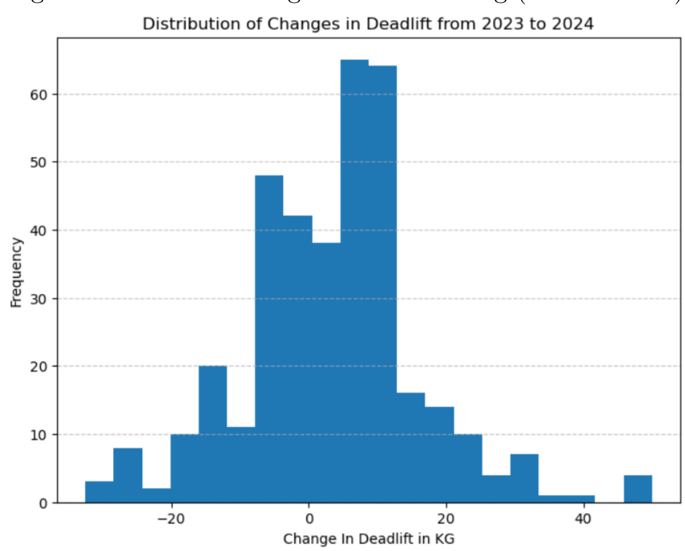
Data Overview and Analysis

First, we compute the change in highest successful deadlift from 2023 to 2024 for each lifter, looking only at the absolute value in kilograms. Note: it is possible for a lifter to fail all three attempts of a lift, in which case they are disqualified and receive a 0kg for that lift. These cases were ignored because they do not show the improvement year over year. We get the following data, which is described both in the Table 1 and Figure 2 below.

Table 1: Change in deadlift descriptive data (2023 to 2024)

Mean Increase	3.44kg
Standard Deviation	13.11kg
25% Percentile	-5.00kg
50% Percentile	5.00kg
75% Percentile	10.00kg

Figure 2: Absolute change in deadlift in kg (2023 to 2024)

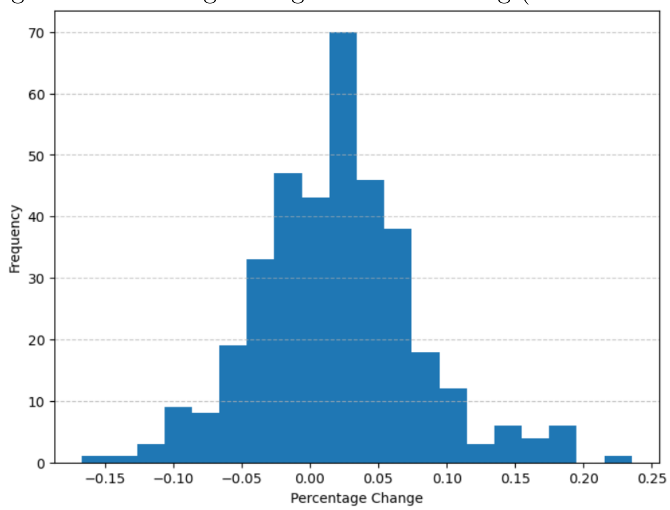


A better measure is using percent change instead of absolute change. Intuitively, if someone has lifted 100kg before, it will be much harder for them to add 10kg on top of that when compared to someone who has deadlifted 200kg before. Using percentages therefore can better take into account factors such as weight class, since those will highly affect the weight a person is able to realistically lift. On top of that, it is very common for powerlifitng training to calculate what weights someone should lift based on percentages of their previous lifts, making it more sport specific. The same descriptive data as before is available in Table 2 and the chart is available in Figure 3 but for percent change instead.

Table 2: Percent change in deadlift descriptive data (2023 to 2024)

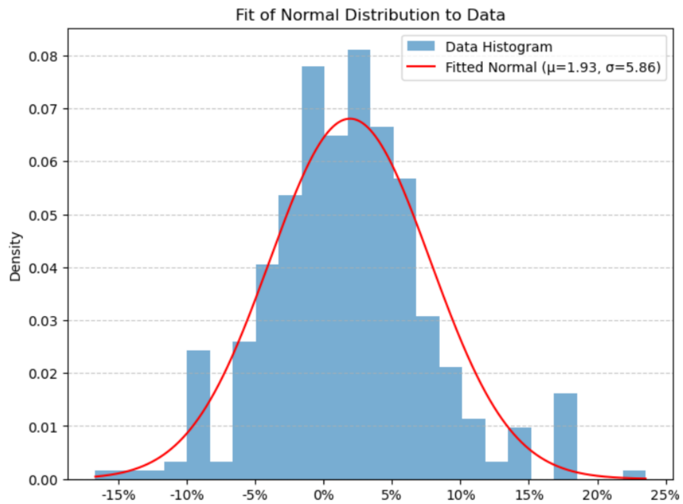
Mean Increase	1.93%
Standard Deviation	5.85%
25% Percentile	-1.75%
50% Percentile	1.75%
75% Percentile	5.15%

Figure 3: Percentage change in deadlift in kg (2023 to 2024)



Looking at the chart for percent change, it looks to be somewhat normally distributed. Using the mean and standard deviation from before, we fit a normal to the data $\sim Normal(\mu = 1.93\%, \sigma = 5.86\%)$. Simply by visually looking at the fit in Figure 4, it seems to be normally distributed and follow our given distribution above.

Figure 4: Percentage change in deadlift in kg (2023 to 2024) with fitted normal



To confirm this more rigorously, we run a Kolmogorov-Smirnof test to see if it is normally distributed. We get KS Statistic = 0.005498 and P-value = 0.20810. To interpret these result, a smaller KS-Statistic means the distribution (the normal), follows our data (raw data for percentage change) well. Also, the null hypothesis is that our data is normally distributed. Not being able to reject the null hypothesis contributes to the idea that the changes in deadlift percentage follow a normal distribution.

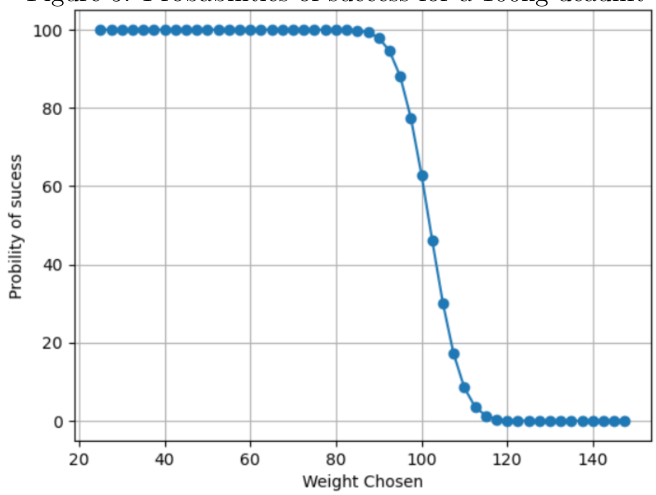
Calculating Probability Distribution

Now that we have a normal distribution modeling the percentage change in deadlift, we can extend this to finding how likely a lifter is to successfully complete a certain weight given their previous data. This can be used for competition other than Raw Nationals, but ideally this would be used in competitions that are roughly one year apart, since the collected data was in this time frame.

For each number in $\{25 + 2.5n \mid n \in \mathbb{Z}_0^+\}$ define the probability of successfully lifting the weight, given their previous deadlift. In order to do so, we will multiply the tail cdf of the fitted normal with the percent change with their previous deadlift. Note: in practice, we don't have to calculate the probabilities to infinity because after a certain point, all the probabilities either would be very close to 0 or even physically impossible to be lifted.

To make this more clear, an example is of someone who had lifted $100kg$ one year ago. If they were to attempt $100kg$ again, this would represent a 0% increase. The tail cdf of our normal evaluates to 62.9% for 0%, so the probability the successfully lift 100kg again is 62.9%. If they were to have tried 95kg or 105 kg instead, the it would represent a 5% increase or decrease, and the probabilities would then evaluate to 88.12% and 30.07%, respectively. Figure 5 shows the probabilities for many possible attempts.

Figure 5: Probabilities of success for a 100kg deadlift



Modeling the Game

While a weight class has many competitors, it usually comes down to the top two lifters in a weight class. This is convenient because it allows it to be modeled as a two player game. The action set of both players are the weights that can be loaded, each one having a chance of being successfully lifted. The winner of the game is whoever has the highest total, which includes the sum of the highest weight successfully lifted in the squat and bench press and the deadlift. At this point it is important to note that the goal of a lifter is not necessarily to maximize their own total, but to have a higher total than their opponent, winning their weight class.

Implementation of the Game

We are given two lifters, as well as their best deadlift data from one year ago. The probability distributions for each lift are available for both players, as well as each other's best squats and bench press. Now, each lifter must decide what weight they should choose for their attempt.

First Attempt

The first attempt has to be handled differently than the other two. In powerlifting, you only have a *total*, meaning that you eligible to place at all if you have at least one successful lift in each of the three modalities. Therefore, the first attempt should be something that you are very likely to successfully complete.

Here, the highest weight that has over 99% chance to be completed will be used. It is important to bring up that the 99% threshold is somewhat arbitrary. Ideally, you should chose a weight that has $100\% - \epsilon$ probability of success, where ϵ is very small. Since our distribution is not continuous, I landed on 99%, but an argument could be made for using a slightly different threshold. Regardless, changing the threshold is an easy change in the python implementation.

To keep our running example of a $100kg$ deadlift, in that case the weight chosen for the first attempt would be an $85kg$ deadlift. In practice, most lifters attempt something around 90% of their highest weight achieved in training [4]. This means our estimates might be slightly conservative, relative to what we observe in practice. However, this should not matter much because at this point lifters are still not optimizing relative to each other.

A rare edge case might happen when a lifter fails their first attempt, even when choosing a weight they are $> 99\%$ likely to achieve. If this were to happen, they would reattempt the same weight for their second attempt (and possible third attempt) to avoid failing all three lifts and being disqualified from competing for a position at all.

Second and Third Attempts

After the first deadlift attempts, we now have a total including a squat, bench press, and deadlift, and are trying to determine what are the optimal weight choices for both lifters. We will denote the current difference in attempts as Δ and the two lifters as lifters A and B.

In a powerlifting competition, in case of a tie whoever has the lower body-weight wins[1], so we will make it so lifter A has the lower body-weight and always wins in case of a tie.

There are four possible outcomes: 1. A and B both succeed in lifting their chosen weights; 2. A succeeds, B fails; 3. A fails, B succeeds; 4. Both fail.

We can calculate the probabilities of each outcome of A and B succeeding first. We use a function $p_X(w)$ that takes a weight as an input and returns how likely a lifter X is to successfully lift it given its previous best deadlift. Then, it is easy to calculate the probabilities for the four cases. We do that for every p pair of weights w_A and W_B.

1. Both succeed: $\mathbb{P} = p_A(w_A)p_B(w_B)$ and difference now $\Delta' = \Delta + w_A + w_B$
2. A succeeds, B fails: $\mathbb{P} = p_A(w_A)[1 - p_B(w_B)]$ and difference now $\Delta' = \Delta + w_A$
3. A fails, B succeeds: $\mathbb{P} = [1 - p_A(w_A)]p_B(w_B)$ and difference now $\Delta' = \Delta - w_B$
4. Both fail: $\mathbb{P} = [1 - p_A(w_A)][1 - p_B(w_B)]$ and difference now $\Delta' = \Delta$

Based on the outcomes above we can define the following utility function for A 1. This will evaluate to 1 if A ends up with a total greater than B's total in a given outcome otherwise, it evaluates to 0. Symmetrically, the utility function for B is the same but evaluates to 1 when Δ is less than 0, and evaluates to 0 when Δ is greater than 1.

$$U_A(w_A, w_B; \Delta) = \begin{cases} p_A(w_A)p_B(w_B)1\{\Delta + w_A - w_B \geq 0\}+ \\ \quad p_A(w_A)[1 - p_B(w_B)]1\{\Delta + w_A \geq 0\}+ \\ \quad [1 - p_A(w_A)]p_B(w_B)1\{\Delta - w_B \geq 0\}+ \\ \quad [1 - p_A(w_A)][1 - p_B(w_B)]1\{\Delta \geq 0\} \end{cases} \tag{1}$$

Lifter A chooses w_A to maximize $U_A(w_A, w_B; \Delta)$ and similarly lifter B chooses w_B to maximize $U_B(w_A, w_B; \Delta)$.

Both lifters will perform the maximization and choose their weights for the second attempt, and then do the same for the third attempt. Here, we are not considering the case that each lifter is also considering the possibilities for the third attempt when optimizing for the second. This means that the algorithm used for both attempts is the same.

In the python implementation, in order for the code to run in reasonable time we first begin by creating the payoff matrices for A and B. Then, we find the payoffs for each entry using the utility functions described above. The mixed equilibrium is then calculated using the Lemke Howson algorithm[5] from the Nashpy package.

This specific algorithm was used because it found the equilibrium in the most reasonable time out of all the available algorithms in the Nashpy package. In order for the other algorithms to run in a reasonable time, we had to truncate the probability space so the payoff matrices became smaller and the algorithm would run and produce results. However, this often required truncating it too much, and only having up to 6 possible attempts for each lifter, which lead to unrealistic results. The Lemke Howson algorithm gave the fastest results, even faster than the others with fewer possible attempts.

Example of Game Implementation

Now that we have determined how to implement the game, we can use an example to better understand this solution. Here, A has a total of 500kg so far, while B has a total of 495kg (this includes squat, bench press and the first deadlift for both lifters). One year ago A deadlifted 200kg, and B deadlifted 205kg. The mixed nash equilibrium for each lifter is see on Table 3 and Table 4.

Table 3: Mixed probability distribution for lifter A

Barbell Weight	$\mathbb{P}$ Weight is chosen by A	$\mathbb{P}$ Success
177.5	2.89%	98.76%
180.0	1.98%	97.89%
182.5	2.9%	96.56%
185.0	4.24%	94.59%
187.5	6.26%	91.83%
190.0	9.42%	88.12%
192.5	14.54%	83.35%
195.0	23.2%	77.49%
197.5	34.57%	70.62%

Table 4: Mixed probability distribution for lifter B

Barbell Weight	$\mathbb{P}$ Weight is chosen by B	$\mathbb{P}$ Success
182.5	36.32%	96.56%
190.0	1.17%	88.12%
192.5	1.76%	83.35%
195.0	2.63%	77.49%
197.5	3.94%	70.62%
200.0	5.98%	62.91%
202.5	9.26%	54.63%
205.0	14.72%	46.16%
207.5	24.22%	37.85%

In this specific example, it appears that for A most of the probability mass lies in 192.5kg to 197.5 kg, which is somewhat conservative with probabilities of sucess all over 70%. Intuitively, this makes sense because A is trying to advance its lead, while not being too agressive because they have a smaller deadlift compared to B.

The probability distribution for B is more interesting. There is a lot of mass put into the very conservative 182.5kg attempt, and then the next weights with high probability mass happen at 202.5kg to 207.5kg, which have a chance of winning of around 50% and under. Intuitively, since B is behind the way I interpret this is that it either stays conservative for the case that A tries a high attempt and misses it, or has to go very aggresive to make up the difference when A successfully completes their own deadlift.

Conclusion and Next Steps

Is this useful in practice? Yes and No. Lifters will usually have good information on what they are expected to successfully lift in competition themselves based on their recent training data. Therefore, they might be able to define a probability measure that is better than simply looking at their numbers one year ago. For other lifters in your weight class, you might have very limited or no data on their training, so defining their probabilities based on their older training numbers might be useful. What one could do is define their own probability measure, and then use the algorithm described here for their opponent's, and calculate the equilibrium based on those two distributions.

In general, this algorithm is as good as the probability measure is, so looking at other ways to calculate the probabilities is useful. In the future, the velocity and training data approaches could be revisited in order to find better distributions.

Another improvement would be by extending it for an n-number of players. In the n-case, we might be optimizing for position (not necessarily just winning the weight class). Alternatively, you could create an algorithm that also considers future attempts. For example you could consider both the second and third attempts when determining your second attempts for deadlift, or even consider all three lifts when determining weights for the squat.

References

[1] USA Powerlifting. USA powerlifting rulebook. https://www.usapowerlifting.com/wp-content/uploads/2023/09/USAPL-Rulebook-v2023.1.pdf, September 25, 2023.

[2] Open Powerlifting. 2023 USAPL raw nationals. https://www.openpowerlifting.org/m/usapl/NS-2023-03-K, 2023.

[3] Open Powerlifting. 2024 USAPL raw nationals. https://www.openpowerlifting.org/m/usapl/NS-2024-05, 2024.

[4] Swede Burns. Powerlifting meet manual (with formula for selecting attempts). https://www.elitefts.com/education/powerlifting-meet-manual-with-formula-for-selecting-attempts/, 2019.

[5] Nashpy. The lemke howson algorithm. https://nashpy.readthedocs.io/en/stable/text-book/lemke-howson.html#lemke-howson, 2018.