Kelly criterion generalized

BY

Lorenz Auer

Abstract

Ann, Bill, Carl and Dana are farmers. Dana and Bill are neighbors, and so are Ann and Carl. The growing conditions on neighboring fields are identical. They all start with the same amount of corn and work with the same amount of care and dedication. Yet after a couple of years Dana consistently harvests more corn than Bill, and Carl harvests more than Ann. (example from [3]) How is this possible? To answer this conundrum, this paper will show how to optimize placing bets. In particular, how to place wagers optimally on multiple bets with multiple outcomes, all running simultaneously. Proofs and a working algorithm are also provided.

Table of contents

1 Introduction

Going back to the example given in the abstract. Let's take a closer look at a harvest cycle to convert it into a more mathematical model. A farmer starts by planting their corn. Then the corn grows. As the corn is fully grown the farmer harvests it. The size of the harvest varies from year to year, since it depends on the amount of rain and sunshine, diseases or pests, etc. There are many different factors that can cut down the yield of the farmer, each of which is more or less random. The farmer has no way of knowing exactly what the conditions over the course of the year will be. Thus we can consider the result of the harvest to be random. In other words it is a bet. Each year the farmer places an amount of corn and at the end they will get a random return. Since Dana and Bill are neighbors and their field‘s growth conditions (weather, etc.) are the same, both have virtually the same bet to play with. The same goes for Ann and Carl. For further modelling assume that the bets are linear or in other words the amount of corn placed has no influence on the odds of the bet. Placing twice as many seeds should yield either twice the return or twice the loss. The last assumption is that the farmers have a rough understanding of the random distribution of results for their fields. For example: Ann knows that about 30% of the time she can harvest 3 times the corn she planted, with a 40% chance she yields 2 times the corn she planted, with a 20% chance she yields the same amount of corn planted, and the last 10% she yields virtually nothing due to some disaster like pests or frost killing all her plants.

2 Binary bet

Before analyzing multiple bets with more than two outcomes, let's considering the simpler problem of just one binary bet. Which means that we look at a single bet, which only has two outcomes, *win* or *loss*.

Definition 1. (Binary bet) *A binary bet has the following defining parameters:*

- *random variable X corresponding to a random event (i.e.: cointoss, diceroll, harvest ...)*
- *probability of winning the bet $p \in [0,1]$ (i.e.: 50%, $\frac{1}{6}$, ...)*
- *odds (= payout) $\phi > 0$ (i.e.: 2, 6, 17, ...)*
- *stake $0 \leqslant r \leqslant 1$ chosen by the gambler in percent of their assets (i.e.: gambler has 100€ and bets 20€ then $r = 20\%$)*

By placing r on such a bet, the gambler has to pay r upfront. Then the random event X occures. If its outcome matches the desired outcome, which has the given probability p of happening, the gambler is payed $\phi \cdot r$. If the outcome does not match the desired one, the gambler gets nothing.

For further analysis it is always assumed, that the gambler can play the bet many times consecutively ("rounds"). Example tossing a coin over and over again.

Warning 2. First of all it should be mentioned that just having a bet with an expected return $\Phi \cdot p \geqslant 1$ is not enough to turn a profit over many rounds of betting. For a demonstration consider the bet with $p = 0.5, \phi = 2.25$. Let's assume the gambler decides to bet 40% of all their assets each round. So in total the gambler either loses 40% or gains 50% with equal likelyhood. Now one might expect an average return of $+5\%$ each round and thus assume that the gamblers capital will grow in the long run. Here is a simulation of 100 seperate runs all starting at a value of 1, risking 40% each round, over 1000 rounds:

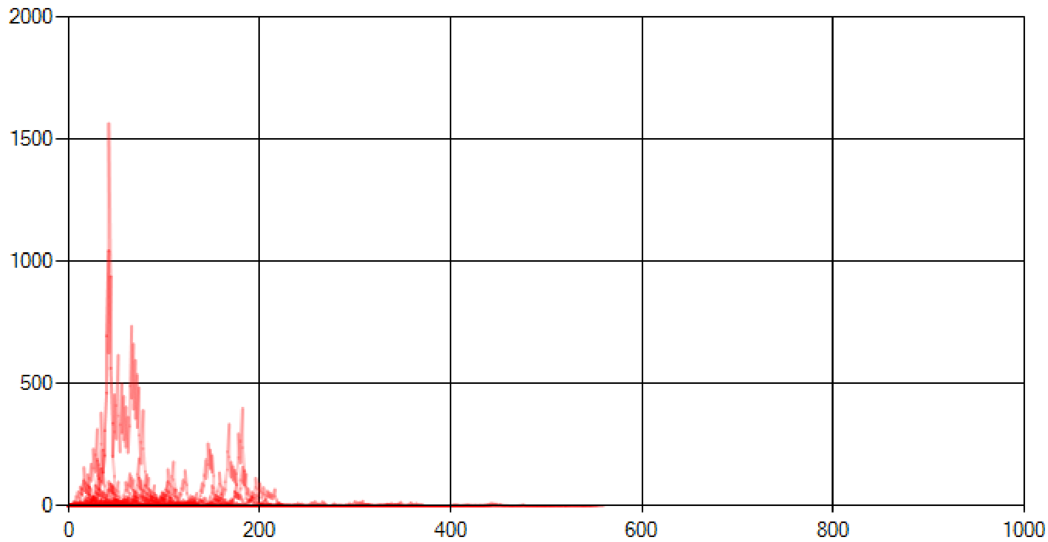

Figure 1. Simulation naive gambler

At first this result might seem quite surprising. Even the runs that seem to generate a lot of value at the beginning eventually lose it aswell. The average at the end of the 100 runs is practically 0 and nowhere close to the expected $1.05^{1000} \approx 1.5 \cdot 10^{21}$.

In 1956 J.L. Kelly published a paper *A new interpretation of information rate* [2] regarding this phenomenon, in which he derived the formula (1) nowadays often known as the *Kelly Criterion* to maximize growth. Following the Kelly criterion in the example above the gambler is best off only risking 10% each round instead of 40%:

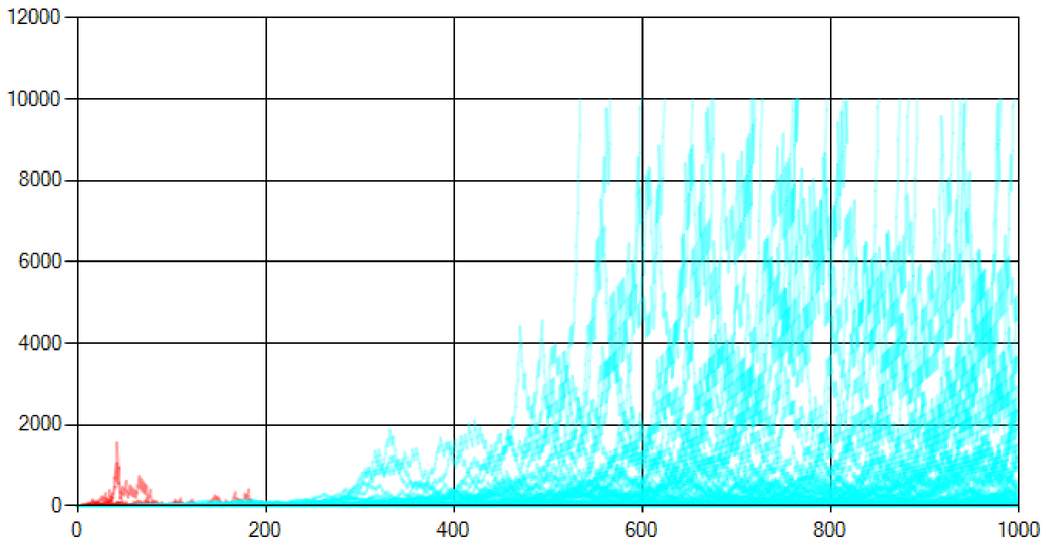

Figure 2. Simulation capped at 10^4 to prevent overflow errors

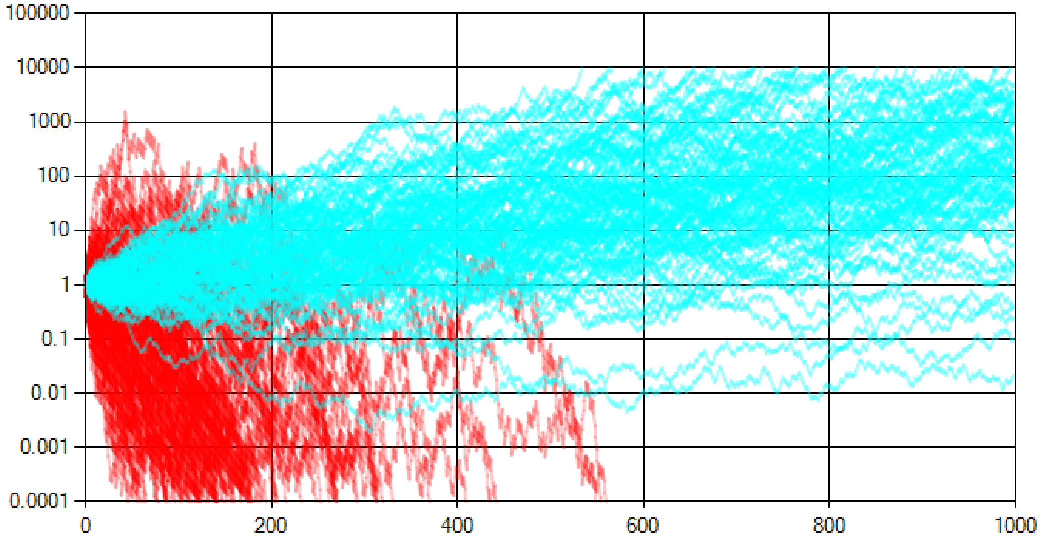

Figure 3. Logarithmic scale for better visualisation

Each round the assets of the gambler either grow by a factor of $1-r+\phi r$ or shrink by $1-r$. Note the order of wins and losses does not matter. Winning and the losing gives the same result as losing and then winning.

Due to the law of large numbers we expect, with $n\gg$, the gamblers assets to experience a growth of:

$$\underbrace{(1-r)}_{\text{loss}}{}^{\overbrace{(1-p)n}^{\#\text{losses}}}\cdot\underbrace{(1-r+\phi r)}_{\text{win}}{}^{\overbrace{pn}^{\#\text{wins}}}=(\text{loss}^{1-p}\cdot\text{win}^{p})^{n}$$

In order to provide the biggest growth for this exponential function (exponential in n) we have to maximize the base, which is equal to the geometric mean of the expected growth per round:

$$\underbrace{(1-r)}_{\text{loss}}{}^{1-p}\cdot\underbrace{(1-r+\phi r)}_{\text{win}}{}^{p}$$

By differentiation one can then derive the maximum at:

$$r=\frac{\phi p-1}{\phi-1} \tag{1}$$

This is also the result found by J.L. Kelly in his paper [2].

3 Finite bet

To generalize this result I will now consider bets with an arbitrary amount of outcomes.

Definition 3. (finite bet) *A* finite bet *has the following defining parameters:*

- *random variable X corresponding to a random event with $n<\infty$ different outcomes.*
- *probabilities for each possible outcome $p\in\mathbb{R}_{>0}^n, \|p\|_1=1$.*

- *odds (= payout) for each possible outcome* $\phi \in \mathbb{R}^n_{\geqslant 0}$.
- *stake* $0 \leqslant r \leqslant 1$ *chosen by the gambler in percent of their assets (i.e.: gambler has* 100€ *and bets* 20€ *then* $r = 20\%$*)*

By placing r *currency on such a bet, the gambler has to pay* r *upfront. Then the random event* X *occures and the index* $1 \leqslant i \leqslant n$ *is chosen according to* p*. The gambler then recieves* $\phi_i \cdot r$ *currency.*

The geometric mean of a finite bet:

$$(1 - r + \phi_1 r)^{p_1} * (1 - r + \phi_2 r)^{p_2} * \cdots = \prod_{i=1}^{n} (1 - r + \phi_i r)^{p_i}$$

If we now consider two such bets, which are independent of one another and executed simulaneously. We get a geometric mean of:

$$\prod_{\substack{i \in \mathbb{N}^2 \\ 1 \leqslant i_1 \leqslant n_1 \\ 1 \leqslant i_2 \leqslant n_2}} \Bigg(1 \underbrace{- r_1 + \phi_{1,i_1} r_1}_{\text{bet}\,1} \underbrace{- r_2 + \phi_{2,i_2} r_2}_{\text{bet}\,2}\Bigg)^{p_{1,i_1} p_{2,i_2}} \qquad \text{with}\, r_1 + r_2 \leqslant 1 \tag{2}$$

This considers all possible outcomes characterised by i, each having a probablity of $p_{1,i_1} \cdot p_{2,i_2}$ since they are independent.

Warning 4. Calculating r_1 and r_2 seperatly with (1) will not yield the best possible growth. Example: Consider two idependent bets both having $\Phi = (2,0)$ and $p = (0.75, 0.25)$. Both are binary bets, hence we can easily compute the best possible risk for each bet individually $r = \frac{2 \cdot 0.75 - 1}{2 - 1} = 0.5$. But if we choose $r_1 = r_2 = r$, there is a $25\%^2 = 6.25\%$ (both bets lose) chance of losing all assets. Which results in an expected growth of 0. Just betting nothing $r_1 = r_2 = 0$ would be a better strategy since it has an expected growth of 1.

3.1 Multiple finite bets

Let's consider m independent, simultaneous bets. Taking a look at (2) the expected growth of m such bets can be generalized to:

$$G(r) := \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1)\, r_j\right)^{\prod_{j=1}^{m} p_{j,i_j}} \qquad \text{with}\, \|r\|_1 \leqslant 1$$

where i is a vector describing the outcome of each bet. Let $R := \{r \in \mathbb{R}^m_{\geqslant 0} \mid \|r\|_1 < 1\}$, then $\overline{R}$ describes all possible bets the gambler could take.

The goal is now to proof that $\operatorname{argmax}_{r \in \overline{R}} G(r)$ can be found with the use of some algorithm or calculation.

Lemma 5. $\overline{R}$ *is convex.*

Proof. Let $r, r' \in \overline{R}$ and $t \in [0,1]$. $r'' = r\,t + r'(1-t)$.

$$r''_j = \underbrace{r_j\, t}_{\geqslant 0} + \underbrace{r'_j\,(1-t)}_{\geqslant 0} \geqslant 0$$

Hence $r'' \in \mathbb{R}^m_{\geqslant 0}$.

$$\|r''\|_1 = \sum_{j=1}^m r''_j = \sum_{j=1}^m r_j\, t + r'_j\,(1-t) = t\sum_{j=1}^m r_j + (1-t)\sum_{j=1}^m r'_j = t\,\underbrace{\|r\|_1}_{\leqslant 1} + (1-t)\underbrace{\|r'\|_1}_{\leqslant 1} \leqslant 1$$

Hence $r'' \in \overline{R}$ and $\overline{R}$ is convex. □

Lemma 6. *Let $a, b \in \mathbb{R}_{>0}$ and $p \in [0,1]$ then*

$$(1+a)^p (1+b)^{1-p} \geqslant 1 + a^p\, b^{1-p}$$

holds.

Proof.

Let $a, b \in \mathbb{R}_{>0}$. (w.l.o.g.: $a \geqslant b$)

$$\begin{aligned} f(p) &:= (1+a)^p(1+b)^{1-p} - 1 - a^p\, b^{1-p} \\ &= (1+b)\left(\frac{1+a}{1+b}\right)^p - 1 - b\left(\frac{a}{b}\right)^p \end{aligned}$$

by showing $f(p) \geqslant 0$ the statement will be proven. Note: $f(p) \in C^\infty(\mathbb{R})$.

For $a = b$:

$$f(p) = (1+b)^p(1+b)^{1-p} - 1 - b^p\, b^{1-p} = (1+b) - 1 - b = 0$$

For $a \neq b$:

$$f(0) = (1+b)\left(\frac{1+a}{1+b}\right)^0 - 1 - b\left(\frac{a}{b}\right)^0 = (1+b) - 1 - b = 0 \tag{3}$$

$$f(1) = (1+b)\left(\frac{1+a}{1+b}\right)^1 - 1 - b\left(\frac{a}{b}\right)^1 = (1+a) - 1 - a = 0 \tag{4}$$

It is also known that:

$$\begin{array}{rcl} a > & b & > 0 \\ \frac{a}{b} & > & 1 \\ \frac{a}{b} > & \frac{1+a}{1+b} & > 1 \\ \ln\left(\frac{a}{b}\right) > & \ln\left(\frac{1+a}{1+b}\right) & > 0 \end{array}$$

Given a and b using the first derivative

$$\frac{\partial}{\partial p} f(p) = \underbrace{(1+b)\ln\left(\frac{1+a}{1+b}\right)\left(\frac{1+a}{1+b}\right)^p}_{>0} - \underbrace{b\ln\left(\frac{a}{b}\right)\left(\frac{a}{b}\right)^p}_{>0}$$

one can explicitly calculate p' with:

$$\frac{\partial}{\partial p} f(p') = 0$$

It follows that:

$$\begin{aligned}
\frac{\partial^2}{\partial p^2} f(p') &= \underbrace{(1+b)\ln\left(\frac{1+a}{1+b}\right)^2 \left(\frac{1+a}{1+b}\right)^{p'}}_{>0} - b\ln\left(\frac{a}{b}\right)^2 \left(\frac{a}{b}\right)^{p'} \\
&< (1+b)\ln\left(\frac{1+a}{1+b}\right)\ln\left(\frac{a}{b}\right)\left(\frac{1+a}{1+b}\right)^{p'} - b\ln\left(\frac{a}{b}\right)^2\left(\frac{a}{b}\right)^{p'} \\
&= \ln\left(\frac{a}{b}\right)\left((1+b)\ln\left(\frac{1+a}{1+b}\right)\left(\frac{1+a}{1+b}\right)^{p'} - b\ln\left(\frac{a}{b}\right)\left(\frac{a}{b}\right)^{p'}\right) \\
&= \ln\left(\frac{a}{b}\right) * \frac{\partial}{\partial p} f(p') \\
&= 0
\end{aligned}$$

$f(p)$ has exactly one extremum which is a maximum at p'. Due to the mean value theorem it follows with (3) and (4) that $f(p)>0$ for $p\in(0,1)$. □

Lemma 7. *Let $n\in\mathbb{N}$. $a_1,\ldots,a_n\in\mathbb{R}_{>0}$. $b_1,\ldots,b_n\in\mathbb{R}_{>0}$. $p_1,\ldots,p_n\in\mathbb{R}_{\geqslant 0}$ with $\sum_{i=1}^n p_i=1$. Then*

$$\prod_{i=1}^n (a_i+b_i)^{p_i} \geqslant \prod_{i=1}^n a_i^{p_i} + \prod_{i=1}^n b_i^{p_i}$$

Proof. With $c_i=\frac{b_i}{a_i}>0$ we can simplify:

$$\begin{aligned}
\prod_{i=1}^n (a_i+b_i)^{p_i} &\geqslant \prod_{i=1}^n a_i^{p_i} + \prod_{i=1}^n b_i^{p_i} \\
\frac{\prod_{i=1}^n (a_i+b_i)^{p_i}}{\underbrace{\prod_{i=1}^n a_i^{p_i}}_{>0}} &\geqslant \frac{\prod_{i=1}^n a_i^{p_i}}{\prod_{i=1}^n a_i^{p_i}} + \frac{\prod_{i=1}^n b_i^{p_i}}{\prod_{i=1}^n a_i^{p_i}} \\
\prod_{i=1}^n \frac{(a_i+b_i)^{p_i}}{a_i^{p_i}} &\geqslant 1+\prod_{i=1}^n \frac{b_i^{p_i}}{a_i^{p_i}} \\
\prod_{i=1}^n \left(\frac{a_i+b_i}{a_i}\right)^{p_i} &\geqslant 1+\prod_{i=1}^n \left(\frac{b_i}{a_i}\right)^{p_i} \\
\prod_{i=1}^n \left(1+\frac{b_i}{a_i}\right)^{p_i} &\geqslant 1+\prod_{i=1}^n \left(\frac{b_i}{a_i}\right)^{p_i} \\
\prod_{i=1}^n (1+c_i)^{p_i} &\geqslant 1+\prod_{i=1}^n c_i^{p_i}
\end{aligned}$$

If $\exists i'\colon p_{i'}=1$:

$$\begin{aligned}
\prod_{i=1}^n (1+c_i)^{p_i} &\geqslant 1+\prod_{i=1}^n c_i^{p_i} \\
(1+c_{i'})^1 &\geqslant 1+c_{i'}^1 \\
1+c_{i'} &= 1+c_{i'}
\end{aligned}$$

Induction start $n=1$: Since $p_1=1$ the inequaltiy holds due to the statement above.

Assume $\forall i\colon p_i\neq 1$:

Induction step $n \to n+1$:

$$
\begin{aligned}
\prod_{i=1}^{n+1} (1+c_i)^{p_i} &= (1+c_{n+1})^{p_{n+1}} \prod_{i=1}^{n} (1+c_i)^{p_i} \\
&= (1+c_{n+1})^{p_{n+1}} \left(\prod_{i=1}^{n} (1+c_i)^{\overbrace{\frac{p_i}{1-p_{n+1}}}^{\sum_i = 1}} \right)^{\underbrace{1-p_{n+1}}_{\neq 0}} \\
\text{apply induction: } &\geqslant (1+c_{n+1})^{p_{n+1}} \left(1 + \prod_{i=1}^{n} c_i^{\frac{p_i}{1-p_{n+1}}} \right)^{1-p_{n+1}} \\
\text{Lemma 6: } &\geqslant 1 + c_{n+1}^{p_{n+1}} \cdot \prod_{i=1}^{n} c_i^{\frac{p_i \cdot (1-p_{n+1})}{1-p_{n+1}}} \\
&= 1 + c_{n+1}^{p_{n+1}} \cdot \prod_{i=1}^{n} c_i^{p_i} = 1 + \prod_{i=1}^{n+1} c_i^{p_i}
\end{aligned}
$$

□

Theorem 8. *G(r) is concave on* $\overline{R}$*.*

Proof. Let $r, r' \in R$ and $t \in (0,1)$. $r'' = r\,t + r'(1-t)$.

$$
\begin{aligned}
G(r'') &= \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(\underbrace{1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1)\, r''_j}_{>0} \right)^{\prod_{j=1}^m p_{j,i_j}} \\
&= \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1)(r_j t + r'_j (1-t)) \right)^{\prod_{j=1}^m p_{j,i_j}} \\
&= \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(1 + t \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r_j + (1-t) \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r'_j \right)^{\prod_{j=1}^m p_{j,i_j}} \\
&= \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(\underbrace{t \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r_j \right)}_{>0} + \underbrace{(1-t) \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r'_j \right)}_{>0} \right)^{\underbrace{\prod_{j=1}^m p_{j,i_j}}_{>0, \sum_i = 1}} \\
\text{Lemma 7 } &\geqslant \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(t \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r_j \right) \right)^{\prod_{j=1}^m p_{j,i_j}} + \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left((1-t) \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r'_j \right) \right)^{\prod_{j=1}^m p_{j,i_j}} \\
&= \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} t^{\prod_{j=1}^m p_{j,i_j}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r_j \right)^{\prod_{j=1}^m p_{j,i_j}} + \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} (1-t)^{\prod_{j=1}^m p_{j,i_j}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r'_j \right)^{\prod_{j=1}^m p_{j,i_j}}
\end{aligned}
$$

$$= t \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r_j\right)^{\prod_{j=1}^{m} p_{j,i_j}} + (1-t) \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1) r'_j\right)^{\prod_{j=1}^{m} p_{j,i_j}}$$

$$= t\,G(r) + (1-t)\,G(r')$$

$G(r)$ is concave on R. $G(r)$ consists of a finite amount of additions and multiplications, so $G(r)$ is continuous. Further $\lim_{\|r\|_1 \to 1} G(r) < \infty$, therefore the result over R can be continued to $\overline{R}$. $G(r)$ is concave on $\overline{R}$. □

Corollary 9. *Over $\overline{R}$ any local maximum of $G(r)$ is a global maximum.*

Proof. Implied directly by Lemma 5 and Theorem 8. □

Hence an algorithm like gradient descent can be implemented to approximate said maximum.

4 Implementation

I implemented such a gradient descent algorthim and will now outline its workings.

4.1 Staying inside bounds

Since each iteration step of gradient descent will change r_i by the gradient of $G(r_i)$ multiplied by some stepsize $t > 0$ $(=1)$

$$r_{i+1} = r_i + t \nabla G(r_i)$$

it has to be checked if $r_{i+1} \in \overline{R}$ still holds.

Warning 10. Just lowering the step size t until $r_{i+1} \in \overline{R}$ does not work. If $r_i \in \partial \overline{R}$ and $\nabla G(r_i)$ points outside, there is no such $t > 0$. And even if $t = 0$ the algorithm still fails, since $r_{i+1} = r_i$ would lead the algorithm to stop, but $\nabla G(r_i)$ might not be normal to the surface $\partial \overline{R}$ at r_i, which would mean the algorithm did not converging properly.

r_{i+1} will have to be placed along the projection of $r_i + t \nabla G(r_i)$ onto the surface $\partial \overline{R}$. An even bigger problem arrises if this projection is also out of bounds.

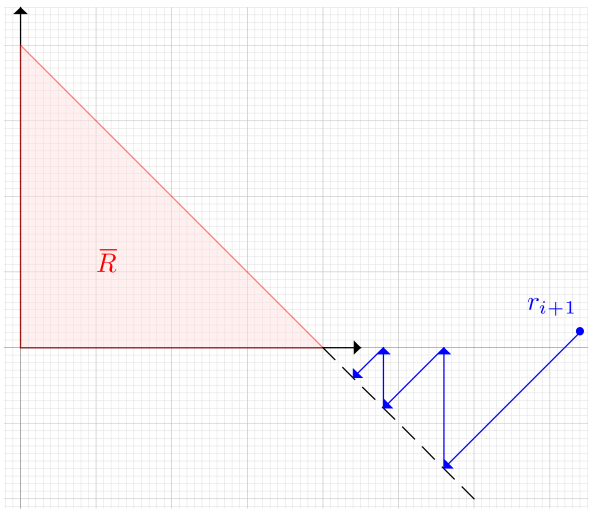

Figure 4. Sketch on how simple projection fails to place r_{i+1} inside the bounds

Another seemingly obvious solution: dividing r_{i+1} by its 1-norm $\|r_{i+1}\|$, straight up does not work, since the algorithm might then wrongly converge as scaling r_{i+1} does not correspond to the normal projection onto ∂R. To solve this I deviced the following algorithm:

Algorithm PLACEIOB(r)
Input: $r \in \mathbb{R}^m$
Output: $r \in \overline{R}$

```
for 1 ⩽ i ⩽ m:
  r_i ← max{r_i, 0};
while r ∉ R̄:
  n ← 0;
  for 1 ⩽ i ⩽ m:
    if r_i > 0:
      n ← n + 1;
  r̂ ← (‖r‖_1 − 1)/n;
  for 1 ⩽ i ⩽ m:
    if r_i > 0:
      r̂ ← min{r̂, r_i};
  for 1 ⩽ i ⩽ m:
    if r_i > 0:
      r_i ← r_i − r̂;
return r;
```

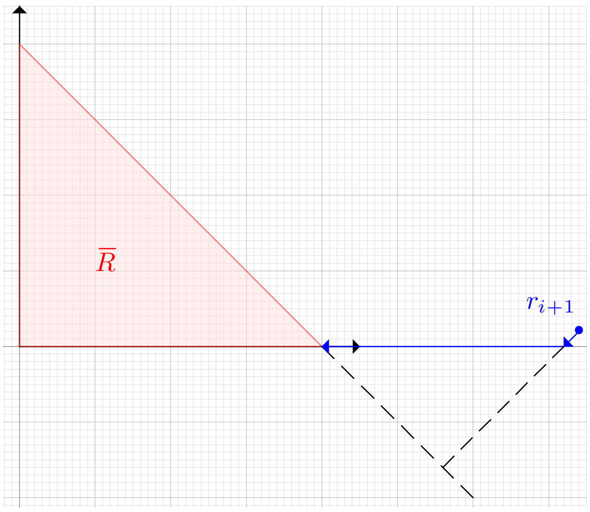

Figure 5. Sketch on how r_{i+1} is placed inside the bounds

4.2 Calculating gradient

$$\frac{\partial G}{\partial r_k}(r) = \sum_{\substack{i\in\mathbb{N}^m,\\ \forall j:1\leqslant i_j\leqslant n_j}} \left((\phi_{k,i_k}-1)\cdot\left(1+\sum_{j=1}^m (\phi_{j,i_j}-1)\,r_j\right)^{\prod_{j=1}^m p_{j,i_j}-1} \cdot \prod_{j=1}^m p_{j,i_j} \cdot \prod_{\substack{i'\in\mathbb{N}^m\\ \forall j:1\leqslant i'_j\leqslant n_j\\ i'\neq i}} \left(1+\sum_{j=1}^m (\phi_{j,i'_j}-1)\,r_j\right)^{\prod_{j=1}^m p_{j,i'_j}} \right)$$

$$= \sum_{\substack{i\in\mathbb{N}^m,\\ \forall j:1\leqslant i_j\leqslant n_j}} \left((\phi_{k,i_k}-1)\cdot\left(1+\sum_{j=1}^m (\phi_{j,i_j}-1)\,r_j\right)^{-1} \cdot \prod_{j=1}^m p_{j,i_j}\cdot G(r)\right)$$

$$= G(r) \cdot \sum_{\substack{i \in \mathbb{N}^m, \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left((\phi_{k,i_k} - 1) \cdot \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1)\, r_j \right)^{-1} \cdot \prod_{j=1}^{m} p_{j,i_j} \right)$$

Note $\frac{\partial G}{\partial r_k}(r)$ is only continuous on R. There could exist $r' \in \overline{R}$ such that.

$$\exists i' \in \mathbb{N}^m \colon 1 + \sum_{j=1}^{m} (\phi_{j,i'_j} - 1)\, r'_j = 0$$

For such r' we get

$$\begin{aligned} G(r') &= \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1)\, r'_j \right)^{\prod_{j=1}^{m} p_{j,i_j}} \\ &= \left(\underbrace{1 + \sum_{j=1}^{m} (\phi_{j,i'_j} - 1)\, r'_j}_{=0} \right)^{\prod_{j=1}^{m} p_{j,i'_j}} \prod_{\substack{i \in \mathbb{N}^m \\ \forall j: 1 \leqslant i_j \leqslant n_j \\ i \neq i'}} \left(1 + \sum_{j=1}^{m} (\phi_{j,i_j} - 1)\, r'_j \right)^{\prod_{j=1}^{m} p_{j,i_j}} \\ &= 0 \end{aligned}$$

By starting at $r_0 = 0, G(r_0) = 1$ and guaranteeing $G(r_{i+1}) > G(r_i)$, it is assured that $\frac{\partial G}{\partial r_k}(r)$ can always be evaluated. Ensuring $G(r_{i+1}) > G(r_i)$ is done by repeatedly halfing the stepsize until the inequality holds. Since $\overline{R}$ is convex a seperate check if the new $r_{i+1} \in \overline{R}$ does not have to be performed.

Algorithm ENSUREGROWTH(r_i,r_{i+1})
Input: $r_i, r_{i+1} \in \overline{R}, G(r_i) \neq 0, r_i \neq r_{i+1}$
Output: $r_{i+1} \in \overline{R}$

```
while G(r_{i+1}) ⩽ G(r_i):
    r_{i+1} ← (r_{i+1} + r_i)/2;
return r_{i+1};
```

4.3 Gradient descent

Outline of the gradient descent algorithm I implemented:

Algorithm OPTIMIZERISK(G,ε)
Input: $G \colon \overline{R} \to \mathbb{R}_{\geqslant 0}$
$\varepsilon > 0$
Output: $r \in \overline{R}$

```
r_0 = 0;
i = 0;
do:
    r_{i+1} ← r_i + ∇G(r_i);
    i ← i + 1
    r_i ← PLACEIOB(r_i);
    r_i ← ENSUREGROWTH(r_i);
while ‖r_i − r_{i−1}‖ > ε;
```

return $r = r_i$;

4.4 Interface

The GUI I implemented consists of two parts. On the left bets can be entered and optimized. On the other hand the right is used to simulate said bets from the left.

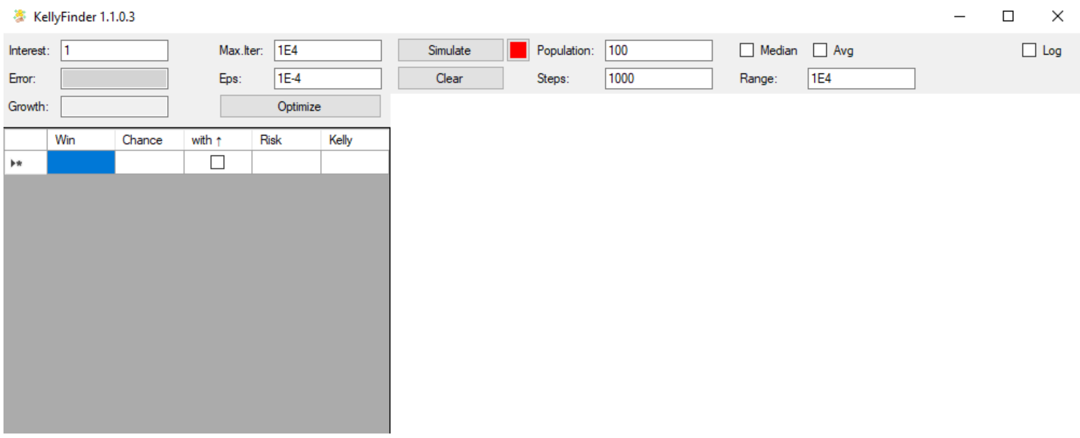

Figure 6. Picture of the GUI

If the *Error*-box is empty no error occured, otherwise there is an error message (ex.: "Error parsing Input"). *Max.Iter.* contains the maximum number of iterations for the optimization algorithm, if this number is exceeded an error will be shown. The stopping criterion ε is entered at *Eps*. In the table below bets can be entered. *Win* corresponds to ϕ, *Chance* to p and *Risk* to r. If *with*↑ is checked, the row is not a seperate bet and nothing can be enter at *Risk*. The row counts as an additional outcome of the last row above with *with*↑ unchecked. If the total *Chance* of a bet is below 1 it is assumed that $\phi = 0$ for the remaining percentile. The column *Kelly* shows the best possible growth when only utilizing the bet of this row. For bets consisting of a single line this entry is calculated using (1). *Growth* shows the current expected growth for given bets and risks. The button *Optimize* optimizes the risks using the algorithm from section 4.3 and changes their entries in the table. *Interest* provides the rate at which not utilized assets grow/decay each round (ex.: $1 = \text{no interest}, 1.01 = +1\%\,\text{interest}, 0.995 = -0.5\%\,\text{interest}$).

On the right side the big white area is reserved for a chart, which shows up as soon as the first simulation is performed. Pressing *Simulate*, *Population* many instances are created with a starting capital of 1. Next they take risks according to the given values in the table. Then for each instance the outcomes of the bets are created randomly. The new capital of the instances is drawn onto the graph for each instance seperatly unless *Median* or *Avg* is checked. For *Median* $25\%, 50\%$ and 75% lines are drawn. If an instance reaches *Range* or $\textit{Range}^{-1}$ it will be stopped to prevent overflow errors in the charting tool. This process is repeated *Steps* many times. Checking *Log* switches the chart to a logarithmic scale on the y-axis.

5 Examples

5.1 Binary bet

First let's test the algorithm on the example from Warning 2:

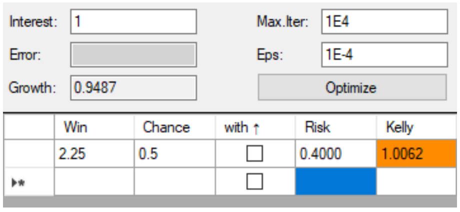

Figure 7. unoptimized risk for binary bet

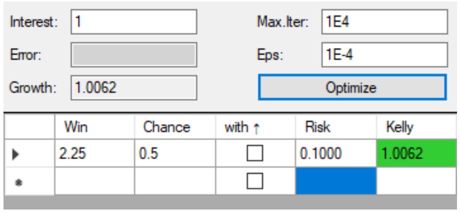

Figure 8. optimized risk for binary bet

In Figure 7 one can see that risking 40% each round leads to a loss of about 5% per round. As expected, pressing *optimize* (Figure 8) calculates that risking 10% yields the best growth. Which is the same result we got using (1).

5.2 Two binary bets

Next is the example from Warning 4:

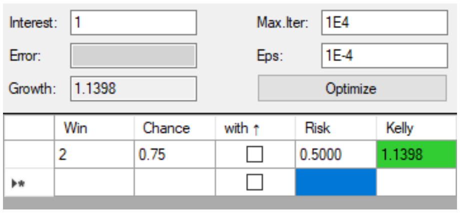

Figure 9. optimized risk for one bet

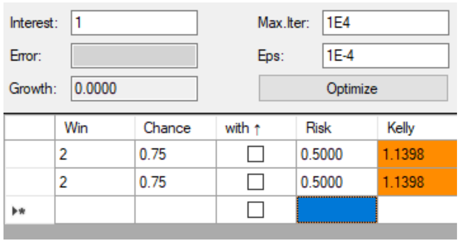

Figure 10. results do not carry over to multiple bets

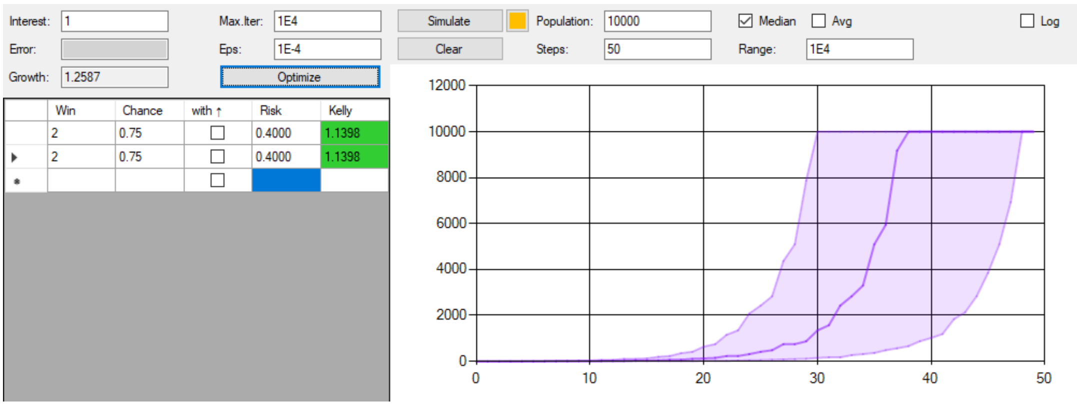

Figure 11. newly optimized risks & graph of resulting growth shown by medians

The results are as already expected in Warning 4. One should bet 40% each instead of 50%.

5.3 Farmers

Now, let's see how the farmers Dana and Carl had bigger harvests than Ann and Bill. As an example let the fields of Dana and Bill, have $\phi = (4,3,2,1,0)$ and $p = (0.2, 0.3, 0.25, 0.15, 0.1)$. And Ann and Carl's fields have $\phi = (3,2,1,0)$ and $p = (0.3, 0.4, 0.2, 0.1)$. Next, we assume Bill and Ann utilize their respective fields optimally (Figure 12 & Figure 13).

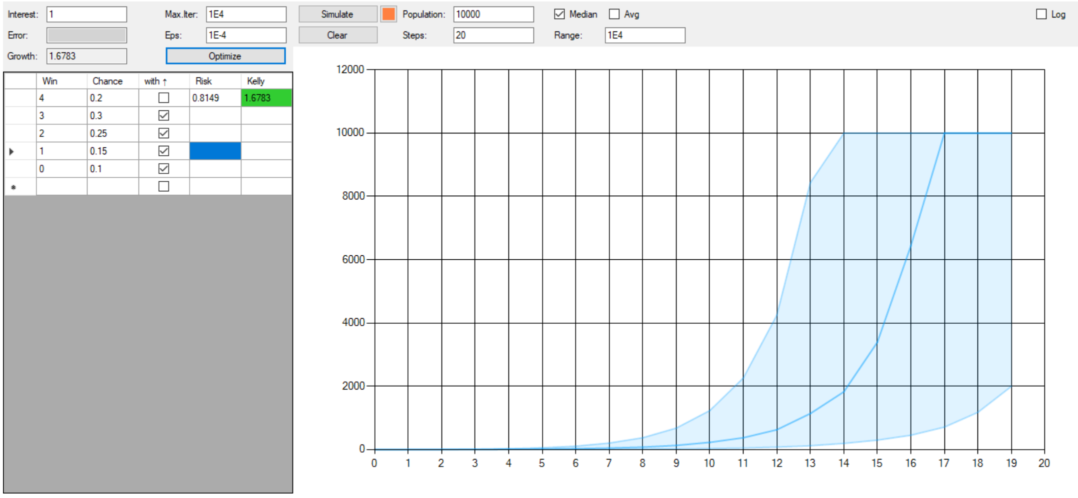

Figure 12. Bill's strategy

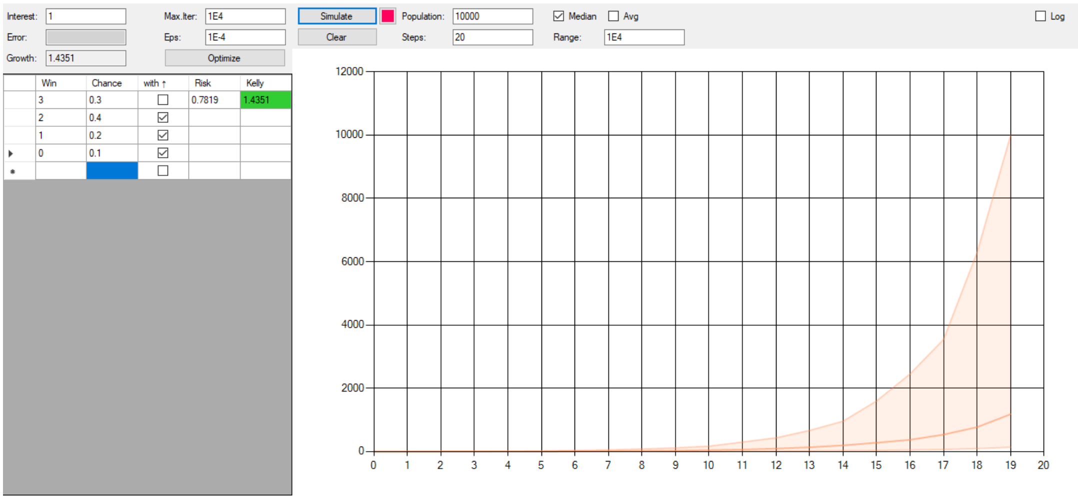

Figure 13. Ann's strategy

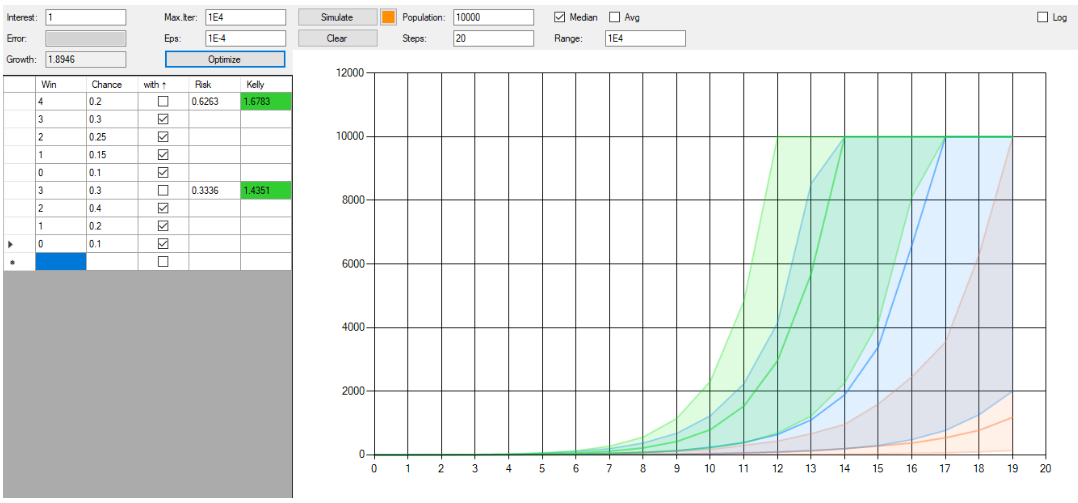

Figure 14. Dana and Carl's strategy (green)

By sharing their fields, and planting the according percentage of corn on each field Dana and Carl both outperform Bill and Ann (Figure 14). Even though the intuitive reaction would be, that Dana has nothing to gain from a cooperation, since Carl's field is obviously worse than Dana's.

5.4 Bonds

A more interesting example from a financial standpoint would be portfolio optimization. In this case let's assume we are given the opportunity to invest in a bond which promises a yield of +2%. Our research suggests that there is a 99% chance of actually receiving the promised payout while in case of the other 1% the investment is lost due to some cause, like the issuer going bankrupt. To invest optimally we can now adhere to the Kelly Criterion:

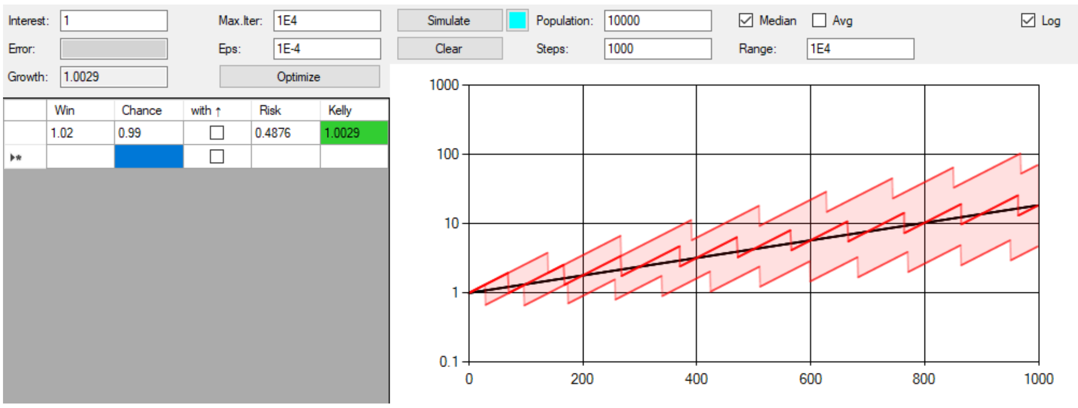

Figure 15. single bond example

The spikes in the median happen, since values needed for a smoother line are never reached. The black line shows the expected growth ($G(r_{\text{opt}}) = 1.0029$) which the algorithm optimized. As predicted by the law of large numbers, the median follows this line closely.

To better model the market let's add an interest rate of -1% for money not spent:

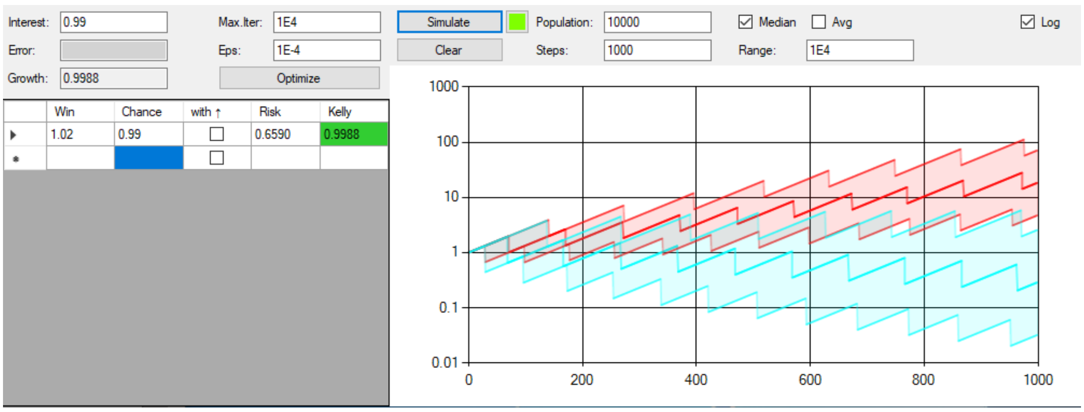

Figure 16. single bond with negative interest

Here one can see that, with negative interest the optimal investment strategy might not turn a profit anymore, but rather loses money as slowly as possible. Furthermore one should risk more than before, since not investing the money is basically losing it anyway.

To show even more interesting behaviours let's add a second bond:

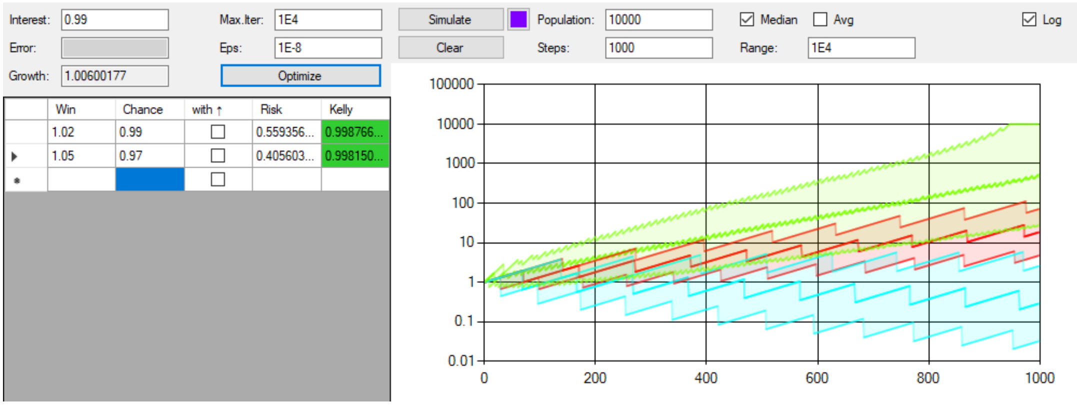

Figure 17. two bonds with negative interest

By investing optimally in either the first or the second bond, one can not achieve a growth of capital (column: Kelly). However, investing into both, results in a growth, even better than that we got with only one bond and without the negative interest.

In Figure 17 it seems like the first bond outperforms the second, as on its own it loses captial slightly slower than the second bond.

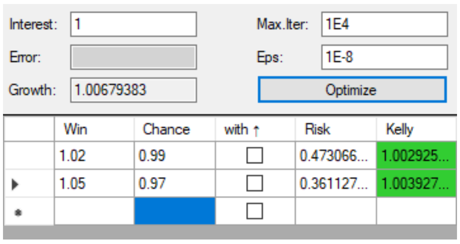

Figure 18. two bonds no interest

Interestingly enough when removing the interest the second bond suddenly outperforms the first. And even more unintutive is the fact that, when investing into both one should invest more into the first bond even though it seems to be worThis could be given by

$$\phi\colon [0,1] \rightarrow \mathbb{R}_{\geqslant 0}, \text{increasing}$$

for a single bet this would yield an expected growth of

$$G(r) = e^{\int_0^1 \ln(1-r+\phi(x)\cdot r)\mathrm{d}x}$$

se than the second one, when judging by their respective optimal growth values.

6 Further Questions

6.1 Interest

How would it affect the optimal betting strategy if assets not spent would generate interest (positive or negative)? I implemented this in the program, however I will not bother going into much detail, as this can simply be done by adjusting ϕ.

6.2 “Infinite” bets

How does this optimization problem evolve when facing infintely many different outcomes. This could be given by

$$\phi\colon [0,1] \to \mathbb{R}_{\geqslant 0}, \text{increasing}$$

for a single bet this would yield an expected growth of

$$G(r) = e^{\int_0^1 \ln(1-r+\phi(x)\cdot r)\mathrm{dx}}.$$

This problem can already be approximated with arbitrary precision by using the finite model given in this paper, assuming $\mathrm{argmax}_{r\in\overline{R}} G_\phi(r)$ is continuous over ϕ. Thus the results of Corollary 9 also hold for the “infinite” case. However the algorithm provided in section 4.3 generally scales very poorly when increasing precision of the approximation.

There are several papers about applying the Kelly Criterion to continuous models with the main goal of being used in the stock market. I will go into further detail on the methods and findings of these papers in section 7.

6.3 Asynchronous bets

How would it affect the model if bets happen asynchronously? As an example: Gambler bets on A. Gambler bets on B. A is payed out. Gambler bets on C. B is payed out. C is payed out.

Assuming the gambler does not know in advance what bets will be given in each timestep. One could argue, using the given algorithm with r_i fixed for some indices i, to model already placed bets, already yields the best possible result. However this might break when the time frames for bets vary greatly. Further research is needed.

6.4 Dependent bets

This would also be a very interesting topic for further investigation, since this together with 6.1 and 6.2 would be able to model a very wide array of different bets, including the stock market (assuming ϕ can be predicted).

Note the gradient descent algorithm would already work if the bets were dependent, by replacing p by an $n_1 \times n_2 \times \cdots \times n_m$ Tensor and picking the respective entry instead of using $\prod_{j=1}^m p_{j,i_j}$ for the probability of a certain result. As in 6.2 Corollary 9 still holds. The problem I faced with this was implementing a nice way to input such m-dimensional Tensors.

7 Related work

In the paper *Practical Implementation of the Kelly Criterion: Optimal Growth Rate, Number of Trades, and Rebalancing Frequency for Equity Portfolios* [1] the authors simplify a continuous bet with normal distribution to a discrete bet with only 2 outcomes ($\phi = (\sigma - \mu, \sigma + \mu), p = (0.5, 0.5)$), where they then are able to apply the standard Kelly criterion. They then go on to compare their results to common portfolio optimization methods such as minimal variance and tangent portfolio using real data examples.

The authors of the paper *Generalized framework for applying the Kelly criterion to stock markets* [6] solve the discrete model by setting $G'(r) = 0$ and consider this as a system of equations to be solved. This however is not trivial as it is not linear and the authors do not go into more detail on a solution. Instead they further analyse the continuous case (geometric Brownian motion) where with serveral assumptions and simplifications they construct a system of linear equations for an analytic solution of several, correlated stocks.

In the paper *Kelly Criterion for Multivariate Portfolios: A Model-Free Approach* [4] the author uses a Monte Carlo algorithm to numerically optimize the expected growth. As their approach is model free, they are able to apply it to continuous as well as dependent bets. As their optimization method is mainly based on randomness they have to manually tweak parameters to provide convergence.

The paper [5] *An explicit solution to the problem of optimizing the allocations of a bettor's wealth when wagering on horse races* covers a different style of general bet. The authors consider a singular horse race which is repeated over and over. The gambler can bet on horses individually, but only the bet with the winning horse gets payed out. They also provide a way to easily calculate the optimal strategy (according to the Kelly Criterion). Only a single racing track is considered at a time. This could also be modelled using multiple binary bets with negatively correlated outcomes.

8 Conclusion

I have developed a proof and a working algorithm for applying the Kelly Criterion to discrete bets. The proof is general in the sense that it can be applied to single or multiple bets, where each bet can have arbitary many outcomes. As mentioned through out section 6 the result of the proof (Corollary 9) is quite robust and can be applied to even bigger extent than shown with my implementation. I also provided examples to illustrate the use cases and effectiveness of the underlying theory as well as the algorithm itself. For a single binary bet the algorithm reproduces the result of the standard Kelly Criterion. The other examples show its application on multiple bets, where the general result is to distribute ones wager instead of "putting all the eggs in one basket". Even though I put little to no effort into optimizing the algorithm for speed, the optimal wagers were computed instantly through out.

Working on this paper was quite intriguing, especially since the last two examples show, that optimal strategies (according to the Kelly Criterion) often are unintuitive and comparsions between those strategies yield surprising results.

Bibliography

[1] Claudio Conversano Andrea Carta. Practical implementation of the kelly criterion: optimal growth rate, number of trades, and rebalancing frequency for equity portfolios. *Frontiers in Applied Mathematics and Statistics*, 6(577050), oct 2020. https://doi.org/10.3389/fams.2020.577050.

[2] J. L. Kelly. A new interpretation of information rate. *The Bell System Technical Journal*, 35(4):917–926, jul 1956. https://ieeexplore.ieee.org/document/6771227.

[3] Elsberg Marc. *Gier: Wie Weit Würdest du gehen?: Roman*. Blanvalet, 2020.

[4] Vasily Nekrasov. Kelly criterion for multivariate portfolios: a model-free approach. Sep 2014. https://dx.doi.org/10.2139/ssrn.2259133.

[5] Dave Tomkins Peter Smoczynski. An explicit solution to the problem of optimizing the allocations of a bettor's wealth when wagering on horse races. *Mathematical Scientist*, 35(1):10–17, 2010. https://www.researchgate.net/publication/268615950.

[6] Tristan Barnett Tim Byrnes. Generalized framework for applying the kelly criterion to stock markets. *International Journal of Theoretical and Applied Finance*, 21(1850033), jun 2018. https://arxiv.org/abs/1806.05293.