SAA with Benders Cut for a maximal return

FUN FUN TALKS

boomer3boom

September 20, 2025

Problem Context

- Suppose we are a Pokemon card collector interested in a specific pikachu card.
- We want to buy at least 30 of these cards this year and plans to sell them much later in the future.
- Every day we are given the option to buy the card off eBay.
- However, every day the card's value changes, but we have an α level of confidence that Pikachu card will grow by 10% this year.
- How should we buy my pikachu card? (ie At what price should I buy them with what amount should I buy them?)

Problem Context

- Suppose we are a Pokemon card collector interested in a specific pikachu card.
- We want to buy at least 30 of these cards this year and plans to sell them much later in the future.
- Every day we are given the option to buy the card off eBay.
- However, every day the card's value changes, but we have an α level of confidence that Pikachu card will grow by 10% this year.
- How should we buy my pikachu card? (ie At what price should I buy them with what amount should I buy them?)

Problem Context

- Suppose we are a Pokemon card collector interested in a specific pikachu card.
- We want to buy at least 30 of these cards this year and plans to sell them much later in the future.
- Every day we are given the option to buy the card off eBay.
- However, every day the card's value changes, but we have an α level of confidence that Pikachu card will grow by 10% this year.
- How should we buy my pikachu card? (ie At what price should I buy them with what amount should I buy them?)

Problem Context

- Suppose we are a Pokemon card collector interested in a specific pikachu card.
- We want to buy at least 30 of these cards this year and plans to sell them much later in the future.
- Every day we are given the option to buy the card off eBay.
- However, every day the card's value changes, but we have an α level of confidence that Pikachu card will grow by 10% this year.
- How should we buy my pikachu card? (ie At what price should I buy them with what amount should I buy them?)

Problem Context

- Suppose we are a Pokemon card collector interested in a specific pikachu card.
- We want to buy at least 30 of these cards this year and plans to sell them much later in the future.
- Every day we are given the option to buy the card off eBay.
- However, every day the card's value changes, but we have an α level of confidence that Pikachu card will grow by 10% this year.
- How should we buy my pikachu card? (ie At what price should I buy them with what amount should I buy them?)

Component To Consider - Stochastic Component

- We want a way to capture a range of outcomes in the future since we don't know what will happen.
- How do we know if our solution works in what "actually" happens?

Component To Consider - Stochastic Component

- We want a way to capture a range of outcomes in the future since we don't know what will happen.
- How do we know if our solution works in what "actually" happens?

Solution - Stochastic Component

- We develop a range of samples that we deem as "valid".
- Our solution should be able to work on y% of the samples that we generated.

Solution - Stochastic Component

- We develop a range of samples that we deem as "valid".
- Our solution should be able to work on y% of the samples that we generated.

Solution - Stochastic Component

- We develop a range of samples that we deem as "valid".
- Let F(n) be a function that outputs a list of list. The outer list will contain n number of index (our samples), and the inner will be a list of 52 index (weeks) with the values representing the price of our pokemon card.
- Our solution should be able to work on y% of the samples that we generated.

Solution - Stochastic Component

- We develop a range of samples that we deem as "valid".
- Let F(n) be a function that outputs a list of list. The outer list will contain n number of index (our samples), and the inner will be a list of 52 index (weeks) with the values representing the price of our pokemon card.
- Our solution should be able to work on y% of the samples that we generated.

My Solution - Stochastic Component

- Monte carlo, Geometric Brownian Motion (GBM), or any variations of GBM.
- I have an alpha value whereby after after the GBM, i pull the price closer to target price: alpha_t * (target_price - next_price).
- I have an alpha level of confidence the price of the pokemon card will move towards the target price.

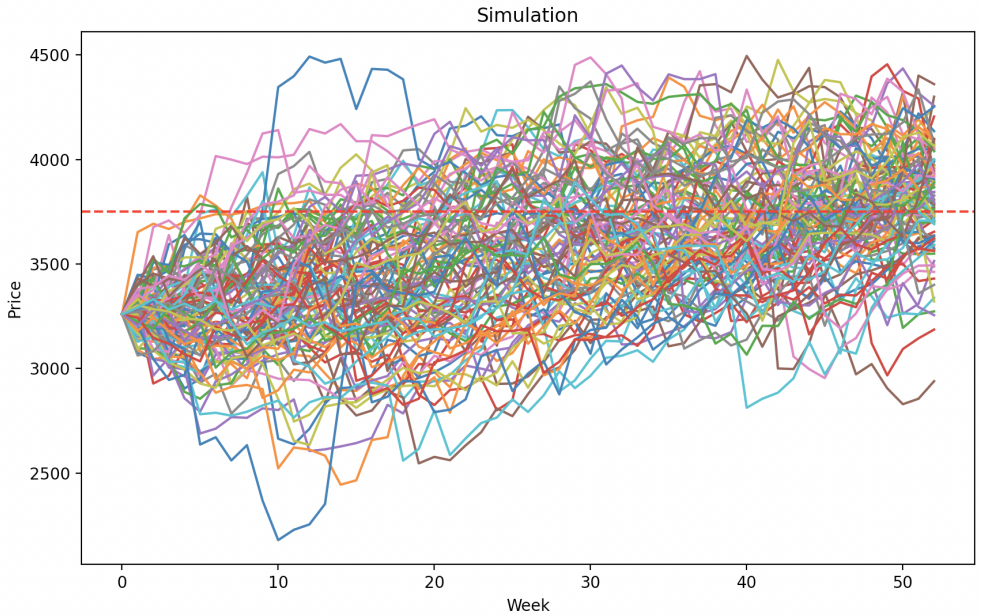
My Solution - Stochastic Component

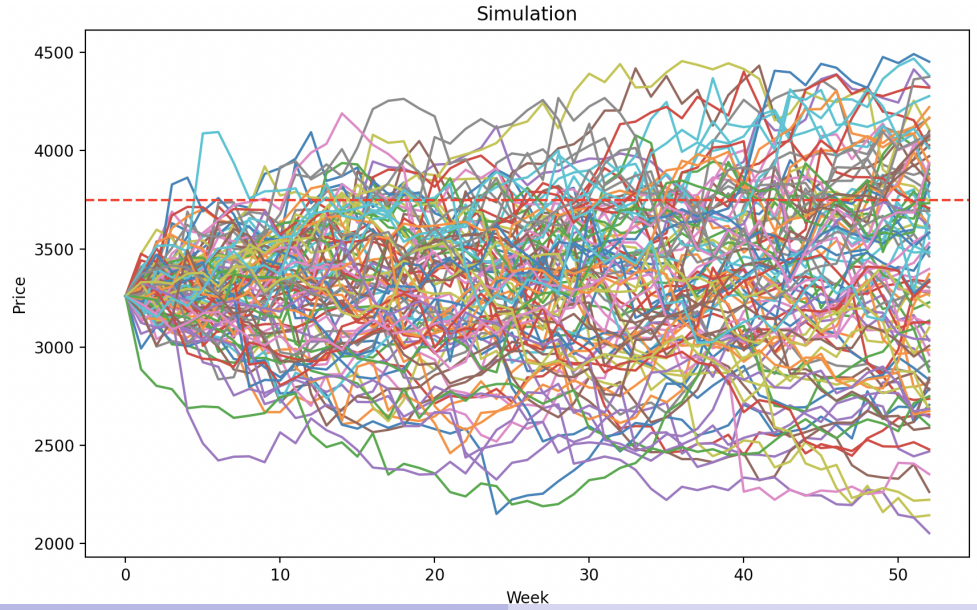
- Monte carlo, Geometric Brownian Motion (GBM), or any variations of GBM.
- I have an alpha value whereby after after the GBM, i pull the price closer to target price: alpha_t * (target_price - next_price).
- I have an alpha level of confidence the price of the pokemon card will move towards the target price.

My Solution - Stochastic Component

- Monte carlo, Geometric Brownian Motion (GBM), or any variations of GBM.
- I have an alpha value whereby after after the GBM, i pull the price closer to target price: alpha_t * (target_price - next_price).
- I have an alpha level of confidence the price of the pokemon card will move towards the target price.

Alpha vs No Alpha





Model?

So. What's our gurobi model?

Objective

- We want to maximize our average return across all scenarios!
- $Z_s \in R$: ROI for scenario s
- $\frac{1}{|S|} \sum_{s \in S} Z_s$

Objective

- We want to maximize our average return across all scenarios!
- $Z_s \in R$: ROI for scenario s
- $\frac{1}{|S|} \sum_{s \in S} Z_s$

Objective

- We want to maximize our average return across all scenarios!
- $Z_s \in R$: ROI for scenario s
- $\frac{1}{|S|} \sum_{s \in S} Z_s$

Variable

- What do we get to decide?
- $BA_{p,a} \in \{0, 1\}$: The amount, a, of stocks we buy at price p.

Variable

- What do we get to decide?
- $BA_{p,a} \in \{0, 1\}$: The amount, a, of stocks we buy at price p.

Constraints

- $\sum_{a \in A} BA_{p,a} = 1 \quad \forall p \in P$: One price can only have 1 a value.
- How do I link the BA and Z values?

Constraints

- $\sum_{a \in A} BA_{p,a} = 1 \quad \forall p \in P$: One price can only have 1 a value.
- How do I link the BA and Z values?

It's hard.... T ^ T

- At this period, we want to buy an amount of stock at price p, but ... how much have we spent on this simulation?
- At the end of the 52nd week, how much income have generated when we don't exactly know how much was spent and at what price and how many stocks?
- Basic Integer program (?) Maybe. But I'm dumb so BENDERS!

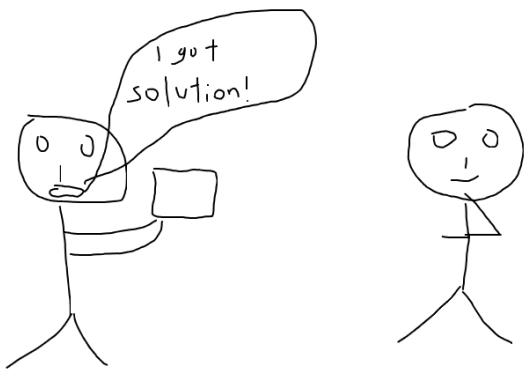
It's hard.... T ^ T

- At this period, we want to buy an amount of stock at price p, but ... how much have we spent on this simulation?
- At the end of the 52nd week, how much income have generated when we don't exactly know how much was spent and at what price and how many stocks?
- Basic Integer program (?) Maybe. But I'm dumb so BENDERS!

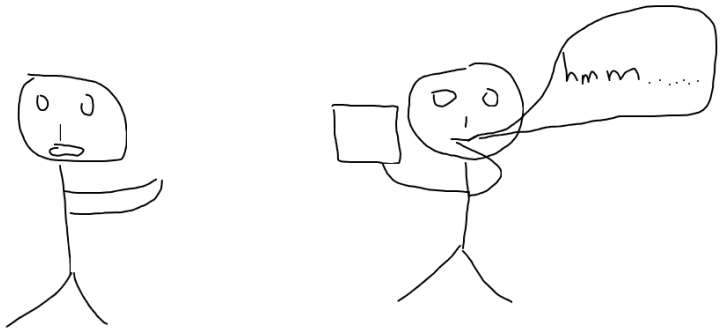
It's hard.... T ^ T

- At this period, we want to buy an amount of stock at price p, but ... how much have we spent on this simulation?
- At the end of the 52nd week, how much income have generated when we don't exactly know how much was spent and at what price and how many stocks?
- Basic Integer program (?) Maybe. But I'm dumb so BENDERS!

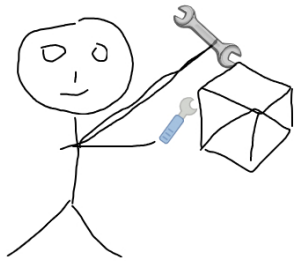
How to Bend This Pokemon card :P



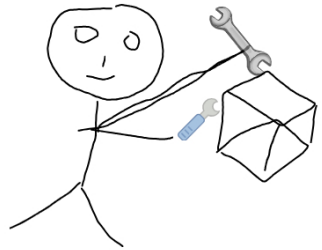
How to Bend This Pokemon card :P



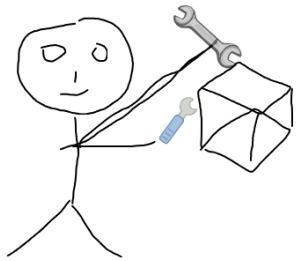
How to Bend This Pokemon card :P



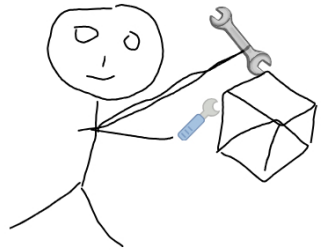
How to Bend This Pokemon card :P



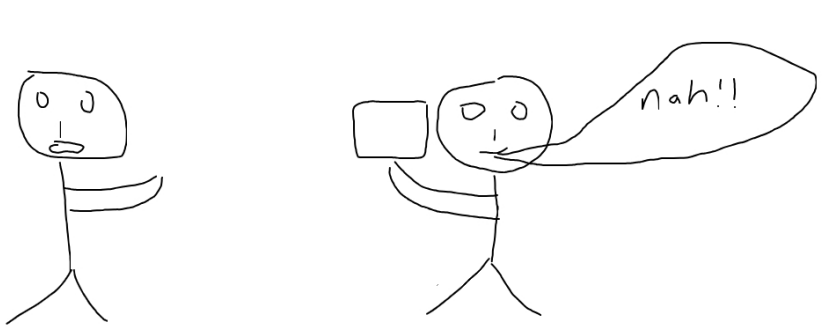
How to Bend This Pokemon card :P



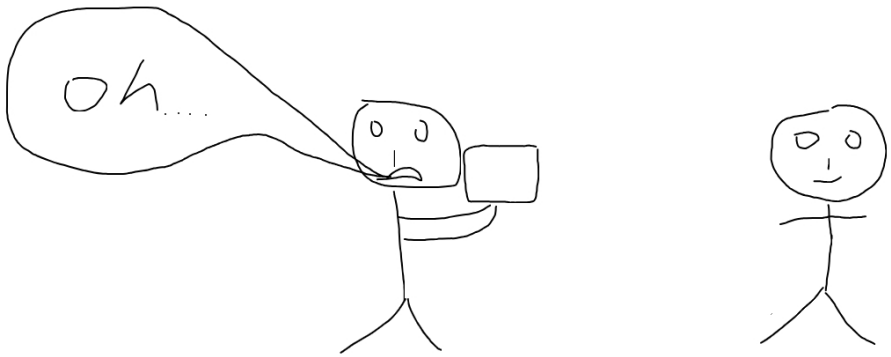
How to Bend This Pokemon card :P



How to Bend This Pokemon card :P



How to Bend This Pokemon card :P



How to Bend This Pokemon card :P

How to Bend This Pokemon card :P

How to Bend This Pokemon card :P

How to Bend This Pokemon card :P

Bender this Pokemon Card :P ft The Child (Master)

- $Asset_s$: The total value of all your card at the end of the year. The upper bound is Captial/min_price * $Price_{s,52}$.
- $Cost_s$: The cost of buying all your cards at the end of the year. The lower bound is $min(Price_s) * 30$.
- $Z_s \leq Asset_s - Cost_s \quad \forall s \in S$: An initial constraint to prevent the master problem from overshooting.

Bender this Pokemon Card :P ft The Child (Master)

- $Asset_s$: The total value of all your card at the end of the year. The upper bound is Captial/min_price * $Price_{s,52}$.
- $Cost_s$: The cost of buying all your cards at the end of the year. The lower bound is $min(Price_s) * 30$.
- $Z_s \leq Asset_s - Cost_s \quad \forall s \in S$: An initial constraint to prevent the master problem from overshooting.

Bender this Pokemon Card :P ft The Mum (Sub-Problem)

- Not a valid solution (feasibility cut) $->$ Provide a constraint that cuts (removes) this solution and (if possible) all the similar solutions that are non-valid.
- A valid solution (optimality cut) $->$ Provide a constraint that sets the true upper/lower bound on the objective value.

Bender this Pokemon Card :P ft The Mum (Sub-Problem)

- Not a valid solution (feasibility cut) $->$ Provide a constraint that cuts (removes) this solution and (if possible) all the similar solutions that are non-valid.
- A valid solution (optimality cut) $->$ Provide a constraint that sets the true upper/lower bound on the objective value.

Bender this Pokemon Card :P ft The Mum (Sub-Problem)

- Not a valid solution (feasibility cut) $->$ Provide a constraint that cuts (removes) this solution and (if possible) all the similar solutions that are non-valid.
- A valid solution (optimality cut) $->$ Provide a constraint that sets the true upper/lower bound on the objective value.

Bender this Pokemon Card :P ft The Mum (Sub-Problem)

- Not a valid solution (feasibility cut) $->$ Provide a constraint that cuts (removes) this solution and (if possible) all the similar solutions that are non-valid.
- A valid solution (optimality cut) $->$ Provide a constraint that sets the true upper/lower bound on the objective value.

Bender this Pokemon Card :P ft The Mum (Sub-Problem)

- let g(BA, Price) be a function of your choice that takes in the BA result from the child (master problem) and price, list of 52 entries.
- g should output a dictionary that contains 5 keys: cost (true cost), stocks (no of stocks), assets (final asset), invalids (list of violation), and ROI.
- list of violations might include: Need More Stocks, or Spend Too Much.

Bender this Pokemon Card :P ft The Mum (Sub-Problem)

- let g(BA, Price) be a function of your choice that takes in the BA result from the child (master problem) and price, list of 52 entries.
- g should output a dictionary that contains 5 keys: cost (true cost), stocks (no of stocks), assets (final asset), invalids (list of violation), and ROI.
- list of violations might include: Need More Stocks, or Spend Too Much.

Bender this Pokemon Card :P ft The Mum (Sub-Problem)

- let g(BA, Price) be a function of your choice that takes in the BA result from the child (master problem) and price, list of 52 entries.
- g should output a dictionary that contains 5 keys: cost (true cost), stocks (no of stocks), assets (final asset), invalids (list of violation), and ROI.
- list of violations might include: Need More Stocks, or Spend Too Much.

The "No-Good" Cut

- Let *current* be a list of (p, a) from the solution of BA.
- $\sum_{(p,a) \in current} BA_{p,a} \leq len(current) - 1$

The "No-Good" Cut

- Let *current* be a list of (p, a) from the solution of BA.
- $\sum_{(p,a) \in current} BA_{p,a} \leq len(current) - 1$

How to generate better cuts

BA = {1: 5, 2: 4, 3: 4, 6: 6, 7: 0}

Not Enough Cards:

- BA = {1: 4, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 3, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

Spend Over Capital:

- BA = {1: 7, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 10, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

How to generate better cuts

BA = {1: 5, 2: 4, 3: 4, 6: 6, 7: 0}

Not Enough Cards:

- BA = {1: 4, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 3, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

Spend Over Capital:

- BA = {1: 7, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 10, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

How to generate better cuts

BA $= \{1\text{: } 5, 2\text{: } 4, 3\text{: } 4, 6\text{: } 6, 7\text{: } 0\}$

Not Enough Cards:

- BA $= \{1\text{: } 4, 2\text{: } 4, 3\text{: } 4, 6\text{: } 6, 7\text{: } 0\}$?
- BA $= \{1\text{: } 5, 2\text{: } 4, 3\text{: } 4, 6\text{: } 3, 7\text{: } 0\}$?
- BA $= \{1\text{: } 5, 2\text{: } 4, 3\text{: } 8, 6\text{: } 3, 7\text{: } 0\}$?

Spend Over Capital:

- BA $= \{1\text{: } 7, 2\text{: } 4, 3\text{: } 4, 6\text{: } 6, 7\text{: } 0\}$?
- BA $= \{1\text{: } 5, 2\text{: } 4, 3\text{: } 4, 6\text{: } 10, 7\text{: } 0\}$?
- BA $= \{1\text{: } 5, 2\text{: } 4, 3\text{: } 8, 6\text{: } 3, 7\text{: } 0\}$?

How to generate better cuts

BA = {1: 5, 2: 4, 3: 4, 6: 6, 7: 0}

Not Enough Cards:

- BA = {1: 4, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 3, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

Spend Over Capital:

- BA = {1: 7, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 10, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

How to generate better cuts

BA = {1: 5, 2: 4, 3: 4, 6: 6, 7: 0}

Not Enough Cards:

- BA = {1: 4, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 3, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

Spend Over Capital:

- BA = {1: 7, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 10, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

How to generate better cuts

BA = {1: 5, 2: 4, 3: 4, 6: 6, 7: 0}

Not Enough Cards:

- BA = {1: 4, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 3, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

Spend Over Capital:

- BA = {1: 7, 2: 4, 3: 4, 6: 6, 7: 0}?
- BA = {1: 5, 2: 4, 3: 4, 6: 10, 7: 0}?
- BA = {1: 5, 2: 4, 3: 8, 6: 3, 7: 0}?

How to generate better cuts - Not Enough Cards

Not Enough Cards:

$$\sum_{\substack{(p,a)\in \text{current},\\ a'\in A,\, a'\leq a}} BA_{p,a'} \leq \text{len}(\text{current}) - 1$$

Spend Over Capital:

$$\sum_{\substack{(p,a)\in \text{current},\\ a'\in A,\, a'\geq a}} BA_{p,a'} \leq \text{len}(\text{current}) - 1$$

Valid Solution but wrong objective

$$Z_s \leq g(BA, prices)[ROI] + (500000 * \sum_{(p,a)inCurrent} 1 - BA_{p,a})$$