Decumulation of Retirement Savings: *The Nastiest, Hardest Problem in Finance* Part I: Introduction and Results

Peter Forsyth[1]

[1]Cheriton School of Computer Science
University of Waterloo

Woudschoten
September 27-29, 2023
Thursday 10:00

Motivation

Defined Benefit Plans (DB) are disappearing

→ Corporations/governments no longer willing to take risk of DB plans

Recent survey[1] P7 countries[2]

- Defined Contribution (DC)[3] plan assets: 55% of all pension assets
- Some examples
  - → Australia 87% DC
  - → US 65% DC
  - → Canada 43% DC
  - → ...
  - → Japan 5% DC

Netherlands → *Collective* DC plan (2027)

[1] Thinking Ahead Institute (2023)
[2] Australia, Canada, Japan, Netherlands, Switzerland, UK, US
[3] DC plan: retiree takes on all investment risk

The retiree dilemma (Defined Contribution (DC))

A retiree with savings in a DC plan[4] [5] has to decide on

- An investment strategy (stocks vs. bonds)
- A decumulation schedule

The retiree now has two major sources of risk

- Investment risk
- Longevity risk (running out of cash before death)

William Sharpe (Nobel Laureate in Economics) calls this

"The nastiest hardest problem in finance"

[4] In a DC plan, the retiree is responsible for investment/decumulation

[5] RRSP (Canada), SIPP (UK), 401(k)(US), Super Fund (Australia)

The Four per Cent Rule

Based on rolling 30-year historical periods, Bengen (1994) showed:

A retiree who

- Invested in a portfolio of 50% bonds, 50% stocks (US), rebalanced annually
- Withdrew 4% of initial capital (adjusted for inflation) annually

→ Would never have run out of cash, over any rolling 30-year period (from 1926)

Criticism

- Simplistic asset allocation strategy
- Simplistic withdrawal strategy
- Rolling 30 year periods contain large overlaps
  → Underestimates risk of portfolio depletion

Bengen rule

> *"Play the long game. A retirement income plan should be based on planning to live, not planning to die. A long life will be expensive to support, and it should take precedence over death planning." Pfau (2018)*

Note that Bengen rule is based on assumption that 65-year old will live to be 95

- Should we mortality weight the cash flows (as in an annuity)?
- Example: median life expectancy of 65-year old male $\simeq$ 87.
  - $\rightarrow$ Effectively, mortality weighting will weight minimum cash flow of 87-year old by $1/2$
  - $\rightarrow$ If I am 87, and alive, I need 100% of my minimum cash flows
  - $\rightarrow$ If I am dead, I need zero dollars
- We will consider an individual investor, not averaging over a population
  - $\rightarrow$ 30 year retirement, no mortality weighting
  - $\rightarrow$ Consistent with Bengen approach

Fear of running out of cash

Recent survey[6]

- Majority of pre-retirees fear exhausting their savings in retirement more than death

In Canada, a 65-year old male

- Probability of 0.13 of living to be 95
- Probability of 0.02 of living to be 100

Conservative strategy:

$\rightarrow$ Assume 30 year retirement (as in Bengen (1994)).

Other assets can be used to hedge extreme longevity[7]

---

[6] 2017 Allianz Generations Ahead Study - Quick Facts #1. (2017), Allianz
[7] Real estate

Objective of this talk

Determine a decumulation strategy which has

- Variable withdrawals (minimum and maximum constraints)
- Minimizes risk of portfolio depletion
- Maximizes total expected withdrawals
- Allows for dynamic, non-deterministic asset allocation

We will treat this as a problem in optimal stochastic control

Formulation

Investor has access to two funds

- A broad stock market index fund
  - *Amount* in stock index S_t
- A constant maturity bond index fund
  - *Amount* in bond index B_t

$$\text{Total Wealth } W_t = S_t + B_t \tag{1}$$

Model the returns of both indexes

- Parametric, jump diffusion
- Non-zero stock-bond correlation

• Fit parameters to market data 1926:1-2019:12
$\hookrightarrow$ All returns adjusted for inflation

Notation

Withdraw/rebalance at discrete times $t_i \in [0, T]$

The investor has two controls at each rebalancing time

$$\begin{aligned} q_i &= \text{Amount of withdrawal} \\ p_i &= \text{Fraction in stocks after withdrawal} \end{aligned} \tag{2}$$

At t_i, the investor withdraws q_i

$$\begin{aligned} W_i^- &= \overbrace{S_i^- + B_i^-}^{\text{wealth before withdrawal}} \\ W_i^+ &= W_i^- - q_i \end{aligned} \tag{3}$$

Then, the investor rebalances the portfolio

$$\begin{aligned} S_i^+ &= p_i W_i^+ \\ B_i^+ &= (1 - p_i) W_i^+ \end{aligned} \tag{4}$$

Can show that

$$q_i = q_i(W_i^-) \quad ; \quad p_i = p_i(W_i^+)$$

Controls

Constraints on controls

$$
\begin{aligned}
q_i &\in [q_{\min}, q_{\max}] \quad ; \quad \text{withdrawal amount} \\
p_i &\in [0, 1] \quad ; \quad \text{fraction in stocks} \\
&\Rightarrow \text{ no shorting, no leverage}
\end{aligned}
$$

Set of controls

$$
\mathcal{P} = \{(q_i(\cdot), p_i(\cdot))) : i = 0, \ldots, M\} \tag{5}
$$

Reward and Risk

Reward: Expected total (real) withdrawals (EW)

$$\text{EW} = E\Big[\overbrace{\sum_i q_i}^{\textit{total withdrawals}} \Big]$$

$$E[\cdot] = \text{ Expectation}$$

Risk measure: Expected Shortfall *ES*

$$ES(5\%) \equiv \Big\{ \text{Mean of worst 5\% of } W_T \Big\}$$

$$W_T = \text{ terminal wealth at } \; t = T$$

ES defined in terms of final wealth, *not losses*[8]

→ Larger is better

[8] ES is basically the negative of CVAR

Objective Function

Multi-objective problem $\rightarrow$ scalarization approach for Pareto points

Find controls $\mathcal{P}$ which maximize (scalarization parameter $\kappa > 0$)[9]

$$\sup_{\mathcal{P}} \Big\{ EW + \kappa \ ES \Big\}$$

$$\sup_{\mathcal{P}} \Bigg\{ \overbrace{E_{\mathcal{P}}[\sum_i q_i]}^{total \ \ withdrawals} + \kappa \overbrace{\left(\frac{E_{\mathcal{P}}[W_T \ 1_{W_T \leq W^*}]}{.05} \right)}^{mean \ \ worst \ \ 5\% \ \ outcomes} \Bigg\}$$

$$\text{s.t.} \quad Prob[W_T \leq W^*] = .05$$

Varying κ traces out the efficient frontier in the (EW, ES) plane

[9] $E_{\mathcal{P}}[\cdot] \equiv$ expectation under control $\mathcal{P}$.

EW-ES Objective Function

Given an expectation under control $E_{\mathcal{P}}[\cdot]$ (Rockafellar and Uryasev, 2000)

$$\text{ES}_{5\%} = \sup_{W^*} E_{\mathcal{P}}\Big[G(W_T, W^*) \Big]$$

$$G(W_T, W^*) = \left(W^* + \frac{1}{.05} \left[\min(W_T - W^*, 0)\right] \right)$$

Reformulate objective function:

$$\sup_{\mathcal{P}} \sup_{W^*} \; E_{\mathcal{P}} \Bigg\{ \overbrace{\sum_i q_i}^{\color{red}{total\ \ withdrawals}} + \kappa \overbrace{G(W_T, W^*)}^{\color{red}{mean\ \ worst\ \ 5\%\ \ W_T}} + \overbrace{\epsilon W_T}^{\color{red}{Stabilization}} \Bigg\}$$

Why do we need the stabilization term?
$\hookrightarrow$ More later

Time Consistency

The EW-ES objective function is not formally *time consistent*

Time inconsistency

$\Rightarrow$ Investor has incentive to deviate from initial optimal policy at later times

EW-ES policy computed at time zero
$\hookrightarrow$ Pre-commitment policy

Induced time consistent policy

At t_0 we compute the pre-commitment EW-ES control

- For $t > t_0$ we assume that the investor follows the *induced time consistent* control (Strub et al (2019))
- This control is identical to the pre-commitment control at t_0
- No incentive to deviate from this control at $t > t_0$

Induced time consistent control determined from (fixed W^*)

$$\sup_{\mathcal{P}} \; E_{\mathcal{P}} \left\{ \sum_i q_i + \kappa G(W_T, W^*) + \epsilon W_T \right\}$$

W^* from pre-commitment solution at time zero

Alternative: equilibrium mean-ES control
$\hookrightarrow$ Does not actually control tail risk! (Forsyth(2020)) [10]

[10] For more discussion of time consistency, induced time consistency, pre-commitment, see Bjork et al (2021), Vigna (2020, 2022), Strub et al (2019), Forsyth (2020)

Withdrawal Control: limiting case

Theorem 1 (Bang-bang withdrawal control: continuous limit)

Assume that

- *the stock and bond indexes follow a parametric jump-diffusion*
- *the portfolio is continuously rebalanced, and withdrawals occur at the continuous (finite) rate* $\hat{q} \in [\hat{q}_{\text{min}}, \hat{q}_{\text{max}}]$

then the optimal control is bang-bang, i.e. the optimal withdrawal $\hat{q}^*$ *is either* $\hat{q}^* = \hat{q}_{\text{min}}$ *or* $\hat{q}^* = \hat{q}_{\text{max}}$.

Proof.

See Forsyth (North American Actuarial Journal (2022)) □

But of course, in real life, we do not withdraw/rebalance continuously.

Scenario: all amounts indexed to inflation

- DC account at $t = 0$ (age 65) \$1,000K (one million)
- Minimum withdrawal from DC account \$35K per year[11]
- Maximum withdrawal from DC \$60K per year
- No shorting, no leverage ($p \in [0, 1]$)
- Annual rebalancing/withdrawals
- Retiree owns mortgage-free real estate worth \$400K

Investment Horizon

- $T = 30$ years, i.e. from age 65 to 95

  $\Rightarrow$ Plan to live long and prosper

[11] Assume gov't benefits of 22K/year. Minimum income $\simeq 22K + 35K = 57K$/year.

Scenario II

Why do we include real estate in the scenario?

Since $q_{\min} = 35K$ per year, W_t can become negative

- When $W_t < 0$, assume retiree is borrowing, using a reverse mortgage[12]
    - Reverse mortgages allow borrowing of 50% of home value
    - In our case: \$200K
- Once $W_t < 0$
    - All stocks are liquidated
    - Debt accumulates at borrowing rate
- If $W_T > 0$, then real-estate is a bequest
- Real estate is a hedge of last resort: not fungible with other wealth
    - This mental bucketing of real estate is a well-known behavioral finance result.[13]

---

[12] See Pfeiffer et al, Journal of Financial Planning (2013)

[13] I also observe this with my fellow retirees: real-estate is a separate bucket

Numerical Method I

Pre-commitment control at t_0 (same as induced time consistent control)

Interchange $\sup \sup (\ldots)$

$$\underbrace{\sup_{W^*} \overbrace{\sup_{\mathcal{P}} E_{\mathcal{P}} \left\{ \sum_i q_i + \kappa G(W_T, W^*) + \epsilon W_T \right\}}^{Solve\ \ using\ \ Dynamic\ \ Programming\ \ (fixed\ \ W^*)}}_{maximize\ \ over\ \ W*}$$

Solve inner DP problem using PIDE methods

Numerical Method II

Inner maximization: dynamic programming

- Conditional expectations at t_i^+
  - Solve linear 2-d PIDE
  - Use δ-monotone Fourier method (Forsyth and Labahn (2019))
- Optimal controls at each rebalancing time
  - Discretize controls
  - Find maximum by exhaustive search
- Guaranteed to converge to the solution as discretization parameters $\rightarrow 0$

Outer maximization over W^*

- Discretize W^*, use coarse PIDE grid

  $\rightarrow$ Find optimal W^* by exhaustive search
- Use coarse grid W^* as starting point for 1-d optimization on finer grids

Data

Center for Research in Security Prices (CRSP) US

- Cap weighted index, all stocks on all major US exchanges 1926:1-2019:12
- US 10 year Treasury index
- Monthly data, inflation adjusted by CPI

Synthetic Market

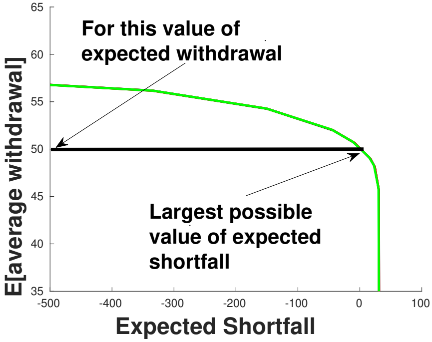
- Stock/bond returns driven by parametric jump-diffusion model, calibrated to data
- Optimal controls computed in the *synthetic* market

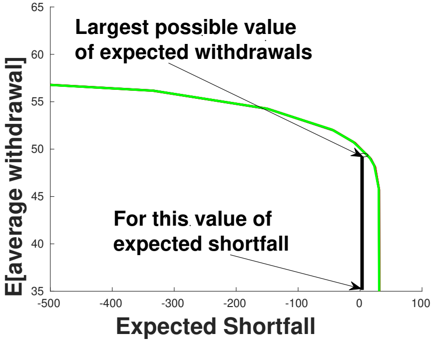
Historical market

- Stock/bond returns from stationary block bootstrap resampling of actual data[14]
- No assumptions about stock/bond processes
- Used to test control robustness computed in the synthetic market

[14] Dichtl et al (2016, Appl. Econ.), Anarkulova et al (JFE,2022)

Pareto optimal points (Units: Thousands)



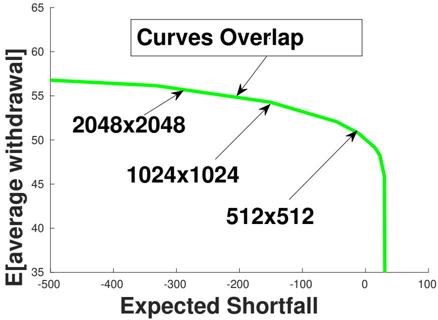


Varying scalarization parameter κ

$\rightarrow$ Traces out efficient frontier

- y-axis is annual average expected withdrawals
- E.g.: 50K ($W_0 = 1000K$) corresponds to 5% withdrawal rate
- Recall ES is mean of worst 5% $W_T \Rightarrow$ larger is better

EW-ES efficient frontier (Units: thousands)

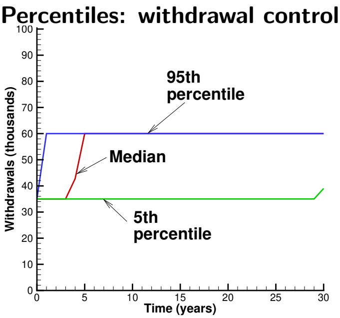


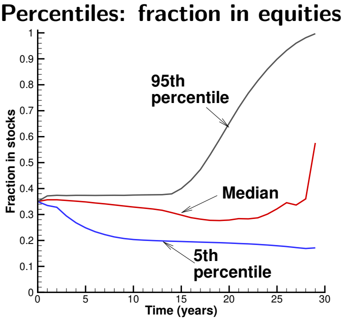
- Solutions with different PIDE grids
- ES is the mean of the worst 5% of outcomes
- Each pt on curve, different κ
- Reverse mortgage hedge
  - $\rightarrow$ Any point $ES > -200K$ is acceptable

Note Efficient Frontier almost vertical at right hand end

- Base case: constant withdrawal 35K/year
- Tiny increase in risk (smaller ES)
  - $\Rightarrow$ Average withdrawal 50K per year (never less than 35K)

Point on Frontier: (EW,ES) = (52K/year, -42K)



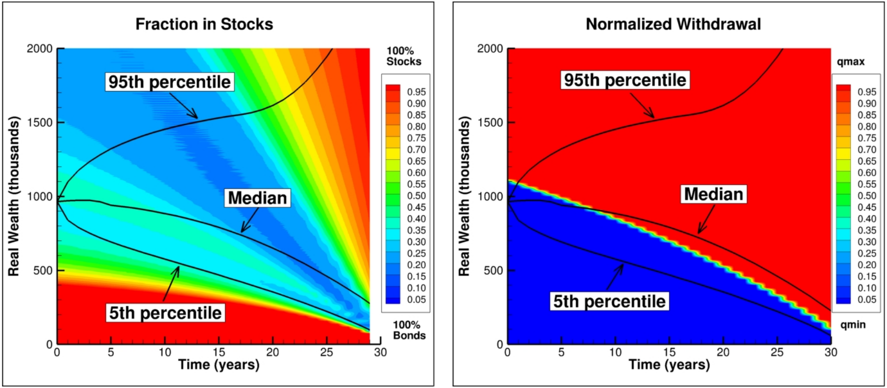


→ $ES \simeq -42K$

→ 5th percentile wealth at $t = 30 \simeq$ 58K

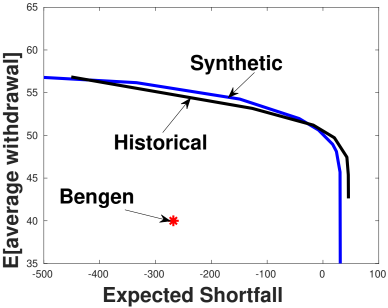
→ Average withdrawal ≃52K/year

Point on Frontier: (EW,ES) = (52K/year, -42K)



- Withdrawal controls $\simeq$ *bang-bang*, i.e. only withdraw either q_{min} or q_{max}.
- Median $W_t \simeq 1000K \rightarrow 300K$

Robustness Check: Efficient Frontier (Units: thousands)



Bengen 4% rule: bootstrapped historical market[a] [b]

⇒ very inefficient

⇒ More risky than advertised, ES ≃ -270K

[a] Bengen suggests 50% in stocks.
[b] Experimentally, 40% in stocks maximized ES.

Controls computed and stored in the *synthetic* market

- Parametric model calibrated to historical data

Controls tested[15] in the bootstrapped historical market

→ Controls are robust to parametric model misspecification

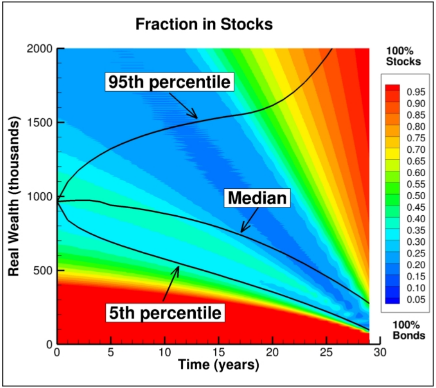
[15] "Out-of-sample" test.

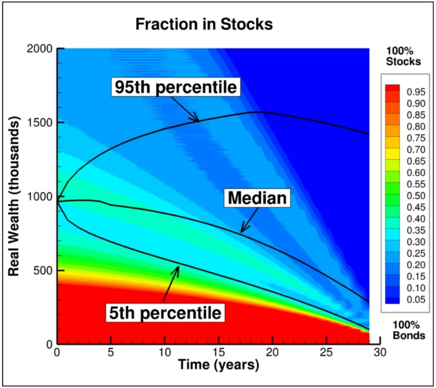
Stabilization term (EW,ES) = (52K/year, -42K)

Recall objective function:

$$\sup_{\mathcal{P}} \sup_{W^*} \Big\{ \overbrace{EW}^{total\ \ withdrawals} + \overbrace{\kappa\ G(W_T, W^*)}^{mean\ \ worst\ \ 5\%\ \ outcomes} + \overbrace{\epsilon W_T}^{Stabilization} \Big\}$$

$\epsilon = +10^{-6}$ $\qquad\qquad$ $\epsilon = -10^{-6}$

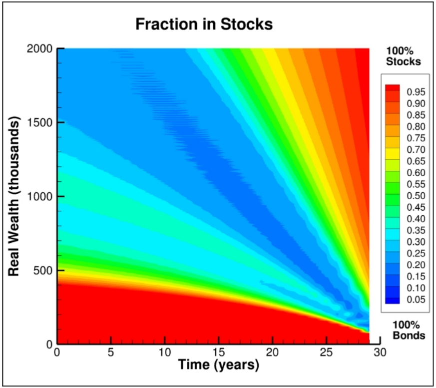




Stabilization term

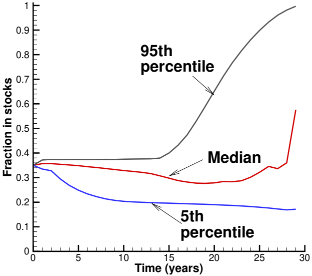
Plots of efficient EW-ES frontiers overlap for $\epsilon = \pm 10^{-6}$
Recall that we are assuming the investor follows the induced time consistent strategy

$\epsilon = +10^{-6}$

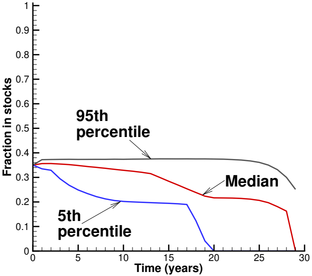


- $W^* = 58K$
- Suppose that $t = 25$, i.e. 90 years old
- W = 2000K, you will never run out of cash with $q_{max} =$ 60K/year
- It does not matter whether you invest 100% in stocks or bonds

If you are Warren Buffet, this problem is ill-posed



$\epsilon = +10^{-6}$



$\epsilon = -10^{-6}$

Fraction Stocks < 0.4 at 95th percentile

If you are rich and old, then it does not matter what you do

- $\epsilon = +10^{-6}$ invest 100% in stocks
- $\epsilon = -10^{-6}$ invest 100% in bonds

But these lucky large wealth outcomes $\Rightarrow$ no effect on (EW,ES) frontier

Peter Ponzo: Canasta Strategy

Peter Ponzo (retired Applied Math Professor from Waterloo)

- Retired: 1993; passed away: 2020
- In 1993, took commuted value of his pension
  - One-half $\rightarrow$ annuity (interest rate: 9.8%)
  - One-half $\rightarrow$ self-directed investments
  - Wrote a blog about his attempts to *"beat the market"*
- It turned out that beating the market was not easy!

But: he summarized his withdrawal strategy: "Canasta Strategy"

> *"If we have a good year, we take a trip to China,...,if we have a bad year, we stay home and play canasta."*

This is a bang-bang control!

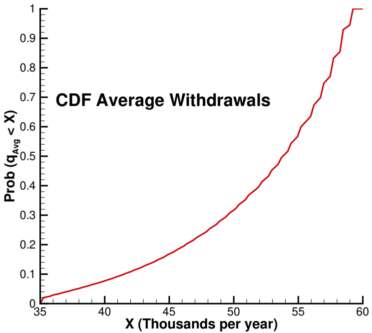
Conclusions

- Optimal strategy: flexible withdrawals, dynamic stock-bond allocation
  - $\rightarrow$ Less risk, higher average withdrawals[16] compared to 4% rule
  - $\rightarrow$ Bootstrap resampling $\Rightarrow$ controls are robust
- In the continuous withdrawal limit
  - $\rightarrow$ Optimal withdrawals are *bang-bang*, i.e. only withdraw at either maximum or minimum rate
- Discrete rebalancing: withdrawal controls are very close to bang-bang
- Intuition: if you are lucky, and make money in stocks, take money off the table and go on a cruise
  - $\rightarrow$ Otherwise: sit tight

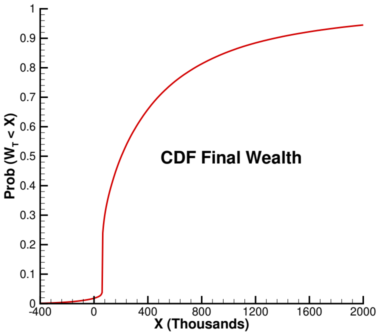
[16] Optimal: 5% EW, with ES $\simeq 0$; Bengen: 4% EW, with ES $\simeq -270K$.

Cumulative Distribution Functions: (EW,ES) = (52K/year, -42K)

Average withdrawal



Wealth at $T = 30$ years



Bootstrap resampled historical data (blksize = 3 months)

- $>$ 94% probability: average withdrawals $> 40K$ per year
- $>$ 98% probability: $W_T > 0$

Decumulation of Retirement Savings:
The Nastiest, Hardest Problem in Finance
Part II: Numerical Algorithms

Peter Forsyth[1] Y. Li[1] M. Chen[1] M. Shirazi[1]

[1]Cheriton School of Computer Science
University of Waterloo

Woudschoten
September 27-29, 2023
Friday 9:00

Decumulation of Retirement Savings

Recall from the first talk

- Retiree wants to maximize total withdrawals
- Minimize risk of running out of cash (30 year retirement)
- Can invest in a mix of stocks and bonds
- At each (yearly) rebalancing time
  - Choose amount to withdraw q
  - Fraction in stocks p
- No shorting/leverage for investments
- $q \in [q_{\min}, q_{\max}]$

Stochastic Process: Stock Index

Let S_t be the real (inflation adjusted) *amount* in a stock index
S_t follows a jump diffusion process

$$\frac{dS_t}{S_{t^-}} = (\mu - \lambda\gamma)\, dt + \sigma\, dZ + d\left(\sum_{i=1}^{\pi_t^s}(\xi_i - 1)\right),$$

$$\sigma^s = \text{ volatility}$$

$$dZ = \text{ increment of Wiener process}$$

$$\pi_t^s = \text{ Poisson process with intensity } \lambda$$

$$S_{t^-} \rightarrow \xi_i S_t \text{ at jump times}$$

$$\gamma = E[\xi - 1]$$

$$\xi \simeq \text{ double exponential distribution} \qquad (1)$$

Stochastic Process: Bond Index

Let B_t be the real (inflation adjusted) *amount* in a constant maturity bond index

Model real returns of the bond index directly as a stochastic process

- Common practitioner approach (Lin et al, IME (2015))
- Avoids modelling interest rates, inflation
- Easy to calibrate to historical data

B_t follows a jump diffusion process

$$\frac{dB_t}{B_{t^-}} = \ldots \text{ similar to stock process} \tag{2}$$

Parameters for both processes calibrated to historical data

Recall

Withdraw/rebalance at discrete times $t_i \in [0, T]$
The investor has two controls at each rebalancing time

$$\begin{aligned} q_i &= \text{Amount of withdrawal} \\ p_i &= \text{Fraction in stocks after withdrawal} \end{aligned} \tag{3}$$

At t_i, the investor withdraws q_i

$$\begin{aligned} W_i^- &= \overbrace{S_i^- + B_i^-}^{\text{wealth before withdrawal}} \\ W_i^+ &= W_i^- - q_i \end{aligned} \tag{4}$$

Then, the investor rebalances the portfolio

$$\begin{aligned} S_i^+ &= p_i W_i^+ \\ B_i^+ &= (1 - p_i) W_i^+ \end{aligned} \tag{5}$$

Can show that

$$q_i = q_i(W_i^-) \quad ; \quad p_i = p_i(W_i^+)$$

Controls

Constraints on controls

$$
\begin{aligned}
q_i &\in [q_{\min}, q_{\max}] \quad ; \quad \text{withdrawal amount} \\
p_i &\in [0, 1] \quad ; \quad \text{fraction in stocks} \\
&\quad \text{no shorting, no leverage}
\end{aligned}
$$

Set of controls

$$
\mathcal{P} = \{(q_i(\cdot), p_i(\cdot)) : i = 0, \ldots, M\} \tag{6}
$$

$$
\mathcal{P}_n = \{(q_i(\cdot), p_i(\cdot)) : i = n, \ldots, M\} \quad \text{tail of the controls} \tag{7}
$$

EW-ES Objective Function

Objective function:

$$\sup_{\mathcal{P}} \sup_{W^*} E_{\mathcal{P}} \left\{ \overbrace{\sum_i q_i}^{total\ \ withdrawals} + \overbrace{\kappa\ G(W_T, W^*)}^{mean\ \ worst\ \ 5\%\ \ outcomes} + \overbrace{\epsilon W_T}^{Stabilization} \right\}$$

Numerical Method I

Interchange $\sup\sup(\ldots)$

$$\underbrace{\sup_{W^*} \overbrace{\sup_{\mathcal{P}} E_{\mathcal{P}}\left\{\sum_i q_i + \kappa G(W_T, W^*) + \epsilon W_T\right\}}^{Solve\ \ using\ \ Dynamic\ \ Programming\ \ (fixed\ \ W^*)}}_{maximize\ \ over\ \ W*}$$

Solve inner DP problem using PIDE methods

Inner problem: value function

$$V(s, b, W^*, t_n^-) = \sup_{\mathcal{P}_n} \Bigg\{ E_{\mathcal{P}_n}^{(S_n^-, B_n^-), t_n^-} \Bigg[\sum_{i=n}^{M} q_i + \kappa \left(W^* + \frac{1}{\alpha} \min((W_T - W^*), 0) \right) \Bigg| (S_n^-, B_n^-)) = (s, b) \Bigg] \Bigg\} .$$

Where:

$$\text{Subject to} \quad \begin{cases} (S_t, B_t) \text{ follow processes (1) and (2);} \\ W_\ell^+ = S_\ell^- + B_\ell^- - q_\ell \\ S_\ell^+ = p_\ell(\cdot) W_\ell^+ ; \; B_\ell^+ = (1 - p_\ell(\cdot)) W_\ell^+ \\ t_\ell = \text{ rebalancing times} \end{cases} .$$

Dynamic Programming Approach

Terminal condition at $t_M = T$

$$V(s, b, W^*, T^+) = \kappa\left(W^* + \frac{\min((s + b - W^*), 0)}{.05}\right) .$$

At any rebalancing time t_n
$\hookrightarrow$ Advance the solution backwards $t_n^+ \rightarrow t_n^-$

$$V(s, b, W^*, t_n^-) = \sup_{(p,q)} \left\{ q + \left[V(w^+ p \ , \ w^+(1-p), W^*, t_n^+) \right] \right\}$$
$$w^- = s + b$$
$$w^+ = w^- - q$$

$t_n^+ = t_n + \epsilon, \ t_n^- = t_n - \epsilon, \ \epsilon \uparrow 0^+$

Between rebalancing times

For $t \in (t_{n-1}^{+}, t_n^{-})$
$\hookrightarrow$ No cashflows, no discounting, for $h \rightarrow 0$
$\hookrightarrow$ Tower property

$$V(s, b, W^*, t) = E\Big[V(S(t+h), B(t+h), W^*, t+h) \Big| S(t) = s, B(t) = b \Big]$$

Apply Ito's Lemma for jump/diffusion processes

$\rightarrow$ 2-D Partial Integro Differential Equation (PIDE)

$\rightarrow$ Independent variables (s, b, t)

Numerical Algorithm: Details

Discretize state space (s, b)

$\hookrightarrow$ 2-D grid, with mesh parameter h

Solve PIDE, using Fourier method

- Standard Fourier methods may not be monotone
- Example: Two possible controls $\mathcal{P}^A, \mathcal{P}^B$ are such that

$$\begin{aligned} \mathcal{P}^{\mathcal{A}} &= \{(q_i(\cdot), p_i(\cdot))) : i = 0, \ldots, M\} \in \mathcal{A} \\ \mathcal{P}^{\mathcal{B}} &= \{(q_i(\cdot), p_i(\cdot))) : i = 0, \ldots, M\} \in \mathcal{B} \end{aligned}$$

- Assume $\mathcal{A} \subset \mathcal{B}$

Then we should have the monotonicity property (optimal control maximizes V)

$$V^{\mathcal{A}}(s, b, t) \leq V^{\mathcal{B}}(s, b, t) \ ; \ \forall (s, b, t)$$

We use a δ-monotone Fourier method $\rightarrow$ guarantees

$$V^{\mathcal{A}}(s, b, t) \leq V^{\mathcal{B}}(s, b, t) \ + \delta$$

Given fixed h, δ can be made arbitrarily small

Numerical Details II

At rebalancing times:

- Discretize the controls with spacing $O(h)$
- Find optimal (p, q) by exhaustive search
- For off-grid points
  - $\rightarrow$ Use linear interpolation of discretized value function

Actual value function $\hat{V}(s_0, b_0, t_0)$

$$\hat{V}(s_0, b_0, t_0) = \sup_{W^*} \overbrace{V(s_0, b_0, W^*, t_0)}^{Inner\ \ PIDE\ \ Solve}$$

Solve problem on sequence of grids

- On coarse grid, discretize W^*, maximize by exhaustive search
- On finer grids, use coarse grid estimate for W^* as starting point
  - $\rightarrow$ Find optimal W^* using 1-d optimization algorithm

Numerical Details III

Solve control problem on grid

- At each rebalancing time, store optimal controls

Determine statistical quantities

- **Synthetic Market:** use stored controls, do Monte Carlo simulations with parametric SDE model of stocks and bonds
- **Historical Market:** use stored controls, do bootstrap resampling of historical stock, bond returns

Bootstrap simulations

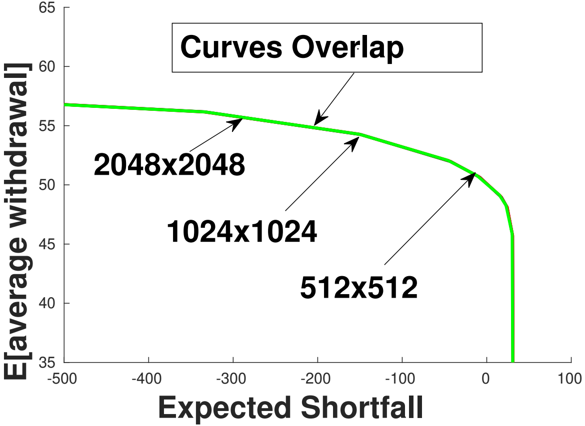
- *Out of sample* test
- No assumptions about market stochastic processes

Numerical Example

- DC account at $t = 0$ (age 65) \$1,000K (one million)
- Minimum withdrawal from DC account \$35K per year[2]
- Maximum withdrawal from DC \$60K per year
- No shorting, no leverage ($p \in [0, 1]$)
- Annual rebalancing/withdrawals
- Retiree owns mortgage-free real estate worth \$400K
  - $\rightarrow$ Hedge of last resort if account exhausted
- Investment horizon: age 65 to 95

[2] Assume gov't benefits of 22K/year. Minimum income $\simeq 22K + 35K = 57K$/year.

Convergence Check: Synthetic Market



$\Rightarrow$ Even coarse grid gives good solution

Alternative Approach: Machine Learning

- Does not use dynamic programming
  - Efficient in cases where performance criteria is high dimensional
    - → Control is low dimensional (see van Staden, Forsyth, Li, SIFIN (2023))
  - Can be used in cases where no dynamic programming principle exists (e.g. mean semi-variance)
- Does not require a parametric model of stochastic processes for stock and bond
- Can be extended to higher dimensional problems (e.g. more assets)

Basic idea [3]

- Go back to original problem formulation
- Approximate control directly using a Neural Network (NN)
- Approximate expectations by sampling paths
- Optimize w.r.t. NN parameters

[3] See also Han (2016), Andersson, Oosterlee (2023).

NN Framework

Approximate controls

$$
\begin{aligned}
q_i(W_i^-, t_i^-) &\simeq \hat{q}(W_i^-, t_i^-; \theta_q) \\
p_i(W_i^+, t_i^+) &\simeq \hat{p}(W_i^+, t_i^+; \theta_p) \\
\mathcal{P} &\simeq \hat{\mathcal{P}} = \{\hat{q}(\cdot), \hat{p}(\cdot)\}
\end{aligned}
$$

$\{\hat{q}(W_i^-, t_i^-; \theta_q),\ \hat{p}(W_i^+, t_i^+; \theta_p)\}$

- fully connected feedforward NNs, parameterized by (θ_q, θ_p)
- Separate NN for $\hat{q}$ and $\hat{p}$.
- Note that using time t as input
  - $\rightarrow$ recurrent network
- Wealth is only state variable needed in this case

Solve for control directly (Policy Function Approximation)

Recall Objective function

$$\sup_{\mathcal{P}} \sup_{W^*} E_{\mathcal{P}} \left\{ \overbrace{\sum_i q_i}^{total\ \ withdrawals} + \overbrace{\kappa\ G(W_T, W^*)}^{mean\ \ worst\ \ 5\%\ \ outcomes} + \overbrace{\epsilon W_T}^{Stabilization} \right\}$$

Generate M sample paths (use stochastic model)

$$\begin{aligned} W_T^j &= \text{Final wealth along } j^t h \text{ path} \\ q_i^j &= \text{Withdrawal at time } t_i \text{ along } j^t h \text{ path} \end{aligned}$$

Approximate $E[\cdot]$ by mean of samples

$$\sup_{W^*, \theta_q, \theta_p} \frac{1}{M} \sum_{j=1}^{M} \left\{ \sum_i q_i^j + \kappa\ G(W_T^j, W^*) + \epsilon W_T^j \right\}$$

Simultaneously maximize over $(W^*, \theta_p, \theta_q)$

NN Method

Each NN has output activation function that encodes constraints

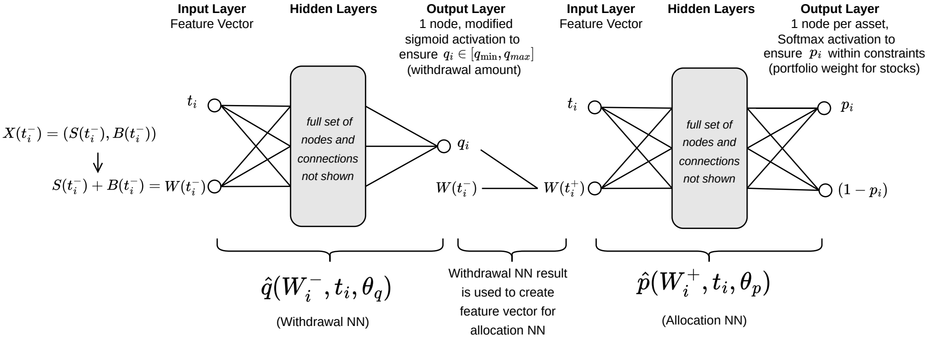
$\rightarrow$ Allows unconstrained optimization (i.e. SGD)

No need to have inner/outer optimization

$\rightarrow$ W^* maximized along with (θ_q, θ_p)

- A single network $\hat{q}(W^-, t; \theta_q)$ approximates the q control for all t
- Similarly for the p control
  - $\rightarrow$ Contrasts with *stacked NN* approach used previously
- Note: we generate paths using parameterized SDEs
  - $\rightarrow$ We are agnostic to method used to generate paths

NN Framework Diagram



Output of $\hat{q}$ network

$\Rightarrow$ Input to $\hat{p}$ network

Withdrawal Control Heatmaps

Withdrawal control is 'bang-bang': Switches abruptly between q_{min} and q_{max}.

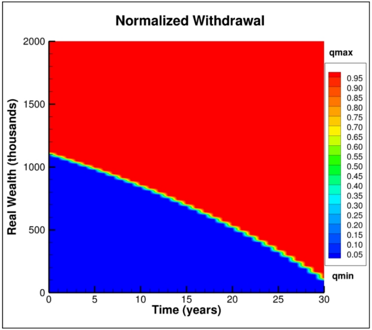


Figure: Withdrawal amount, PDE Control, $\epsilon = 10^{-6}$

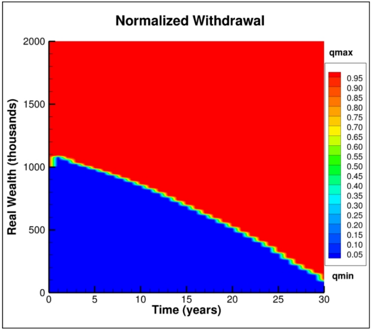


Figure: Withdrawal amount, NN Control, $\epsilon = 10^{-6}$

Units: thousands of dollars

Stock Allocation Control Heatmaps (1)

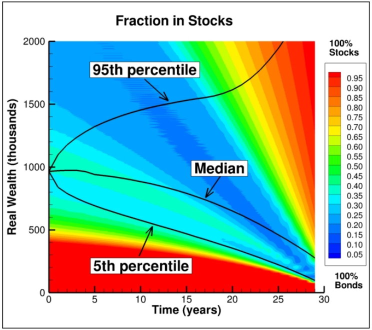


Figure: Fraction in stocks, PDE Control, $\epsilon = 10^{-6}$

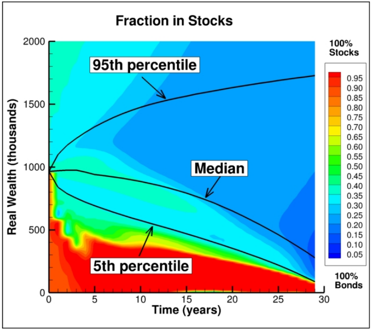


Figure: Fraction in stocks, NN Control, $\epsilon = 10^{-6}$

Effect of stabilization term clearly shown in PDE heatmap, but NN is not sensitive enough (ϵ is tiny). *Units: thousands of dollars*

Stock Allocation Control Heatmaps (2)

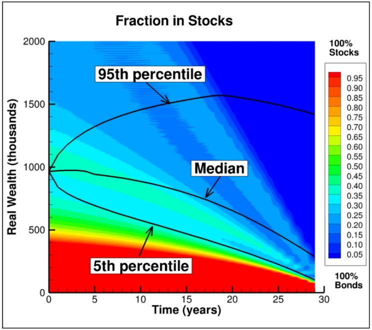


Figure: Fraction in stocks, PDE Control, $\epsilon = -10^{-6}$

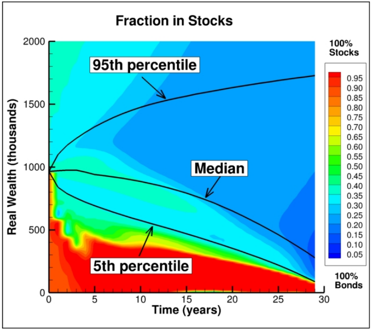


Figure: Fraction in stocks, NN Control, $\epsilon = 10^{-6}$

Making **stabilization term negative** shows that NN control is somewhere in between +/- epsilon versions of PDE control. *Units: thousands of dollars*

Efficient Frontier Comparison: Synthetic Market

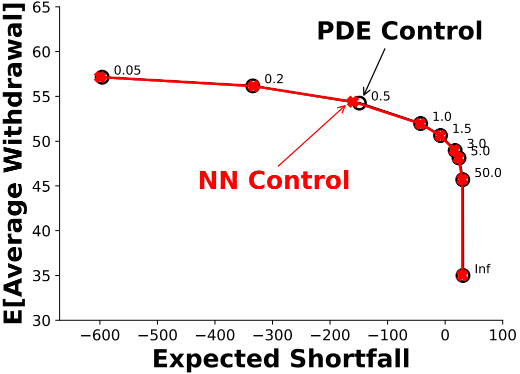


Figure: Comparison of EW-ES frontier for NN and PDE methods. Labels on nodes are the κ values. *Units: thousands of dollars*

PDE frontier virtually the same, $\epsilon = \pm 10^{-6}$

Bootstrap Resampling

Stationary Block Bootstrap resampling

- Monthly historical data: 1926:1-2020:1
- Draw blocks of data (with replacement) from historical data
  - → Simultaneously draw stock and bond returns
  - → Sampling in blocks preserves serial correlation
- Blocksizes are drawn from a geometric distribution
  - → Random blocksizes reduce edge effects, preserve stationarity
- Concatenate blocks to form a single path of T years
- Dubious algorithm available to determine expected blocksize

Typical parameters

- 10^5 training samples, 10^5 test samples
- Probability of a single identical train, test path $< 10^{-29}$

The universe is 10^{18} seconds old.

Train on Synthetic Data, Test on Historical Data

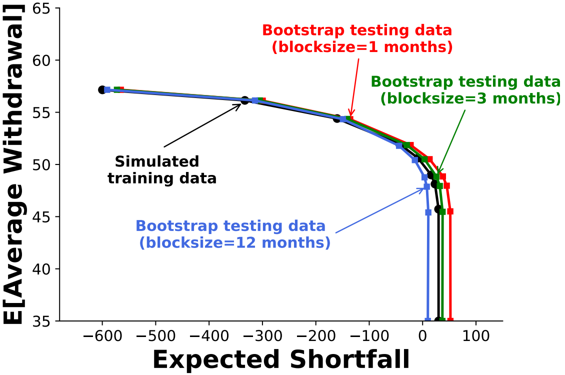


Figure: Comparison of EW-ES frontier for NN training performance vs. tests on resampled historical data. *Units: thousands of dollars*

Train with Historical Data, Test on Synthetic Data

Demonstrates NN framework's ability to use other datasets and still yield good results.

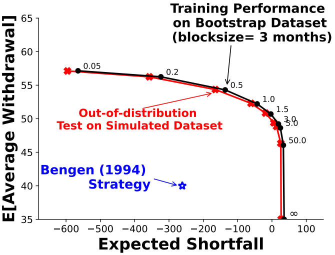


Figure: Historical training data, block size = 3 months

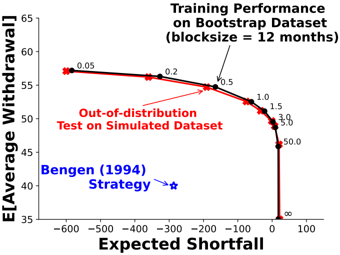


Figure: Historical training data, block size = 12 months

Labels on nodes: κ values. Units: thousands of dollars

Conclusions

- Train/test combinations → multi-period optimization is robust
- NN method → accurate results compared to ground truth

  → Even for bang-bang controls
- Advantages of NN
  - Does not depend on parametric SDE model (data driven)
  - Can solve high dimensional problems
  - Can be used for problems which do not have DP principle

But

CPU time for computing a single point on the efficient frontier

- PDE: medium grid (C++) $\simeq$ 400 sec (laptop)
- NN: 2 hours (Pytorch + GPUs)

Low dimensional problem, parametric model for stochastic processes

→ PDEs win