Formulation of the optimization model in Owl

Martin-D. Lacasse

June 20, 2024

Chapter 1

Introduction

This document describes the mathematical model undelying the optimization algorithms implemented in Owl. This code is a Python application optimizing retirement planning using linear programming. The goal of these calculations is to optimize the financial aspects of retirement planning, considering the types of savings accounts, income tax, contributions, return rates, Roth conversions, and desired income amongst many other things.

The approach is described here mathematically and the implementation follows the structure and notation presented in this document. The goal of this document is to provide a guide to the Python code intended for any individual desiring to extend the model to other cases.

Chapter 2

Indices, variables, and parameters

In the next sections, the indices, variables, and parameters are described in detail. Then the model constraints are introduced. For implementation in a linear programming solver, index mapping functions are proposed to map all variables into a single one-dimensional array that is optimized subject to inequality and equality constraints expressed in matrix form. Finally, the contraint matrices are built and so are some useful objective functions.

2.1 Indices

For all indices, we will follow the C array style (starting at 0), rather than the traditional mathematical standard starting at 1. This will facilitate the final sequential mapping of all the variables into a single one-dimensional array, and serve as a direct reference for better understanding the code implementation.

The indices used and their range are defined here, while we also introduce the characteristics and dimensions of the problem. Upper bounds on indices are indicated by the letter N, with the index name as a subscript, e.g., N_i for index i.

i — Individual. i runs from 0 to $N_i - 1$ where $N_i = 2$ for couples, or $N_i = 1$ for single individuals. The first individual to pass is denoted by i_d while the survivor is i_s.

j — Type of savings account. j goes from 0 to $N_j - 1$, for taxable, tax-deferred, and tax-exempt accounts respectively. Therefore $N_j = 3$.

k — Type of asset class. k goes from 0 to $N_k - 1$, for S&P 500, Baa corporate bonds, Treasury notes, and cash, respectively. $N_k = 4$. More asset classes could be considered at the cost of increasing the complexity of the problem while not generating much more insights.

n — Year being modeled. n runs from 0 to $N_n - 1$, and N_n is the first year following the passing of all individuals in the plan. The time period for all decision variables is annual. For spouses, $n_d - 1$ is the year in which the first individual passes while the survivor will decease in year $N_n - 1$ of the plan.

t — Federal income tax bracket. t goes from 0 to $N_t - 1$, from low to high. There are $N_t = 7$ federal income tax brackets.

2.2 Variables

We will use lowercase roman letters to represent variables. All variables are assumed to take only positive values (positive inequality).

b_{ijkn} Balance for individual i with asset class k in savings account j at the beginning of year n. When we only consider the sum in the savings account, without considering the distribution in each asset class k, the variable b_{ijn} is used.

$b^{\pm}_{ijkn}$ Amount to be added or subtracted to savings account balance value b in order to rebalance asset allocations of savings account j across different classes of assets k. We note variables not bound by positivity ≥ 0) by the superscript $^{\pm}$.

d_{ikn} Deposit of year-n net spending surplus in assets k of taxable account of individual i. Deposit is made at the end of year n.

f_{tn} Fraction of tax bracket t filled, so that taxable ordinary income G_n can be expressed as

$$G_n = \sum_t f_{tn}\bar{\Delta}_{tn}, \tag{2.1}$$

$$0 \leq f_{tn} \leq 1. \tag{2.2}$$

A definition of Δ can be found in the section describing the parameters below.

g_n Net spending in year n.

w_{ijkn} Withdrawal from assets k in account j belonging to individual i at the beginning of year n. For the ($j = 1$) tax-deferred savings account, w_{i1kn} is referred to as a distribution for tax purposes as it is a taxable withdrawal, and will always satisfy required minimum distributions.

x_{ikn} Roth conversion performed by individual i using asset class k at the beginning of year n. These events are taxable as ordinary income.

2.3 Parameters

For more easily distinguishing parameters from variables, all parameters will be expressed in Greek letters. Parameter values are either set by the user, historical data, or by the tax code.

βij Initial balances in savings accounts. These amounts are used to initialize b_{ijk0}.

τ_{kn} Annual rate of return for asset class k in year n. A time series of annual return rates for each class of asset. Here, inflation and the rate of return of ($k = 3$) cash are assumed to be the same.

γ_n Cumulative inflation at the beginning of year n computed as

$$\gamma_n = \prod_{n'=0}^{n-1} (1 + \tau_{3n'}), \tag{2.3}$$

with $\gamma_0 := 1$, and where n' is a dummy index. Parameters indexed for inflation will be indicated by a bar on top as in $\bar{\sigma}_n$.

σ_n Standard deduction. It can be adjusted for inflation as follows

$$\bar{\sigma}_n = \sigma_n \gamma_n, \tag{2.4}$$

and can be modified for additional age exemptions after 65 of age, for example. It is a simple time series to be provided which can include any foreseeable changes in the tax code, or change in filing status due to the passing of one spouse for $n \geq n_d$.

ξ_n Spending profile. This is a time series that multiplies the desired net spending amount. It is $\xi_n = 1$ for a flat profile, or can be a *smile* profile allowing for more money at the start of retirement. Parameter ξ_n can also contain spending adjustments typically made at the passing of one spouse. The *smile* can be implemented using a cosine superimposed over a gentle linear increase such as in

$$\xi_n = 1 + a_1 * cos(2n\pi/(N_n - 1)) + a_2 n/(N_n - 1), \tag{2.5}$$

and then normalized by factor $N_n/(\sum_n \xi_n)$ to be sum-neutral with respect to a flat profile. Values of $a_1 = 15\%$ and $a_2 = 12\%$ provide curves that are similar to realistic spending profiles reported in the literature. See Fig. 2.1 for an example. At the passing of one spouse, both profiles are reduced by a factor χ for $n \geq n_d$, and the normalizing factor needs to be adjusted accordingly.

χ Factor to reduce spending profile after the passing of one spouse. It is typically assumed to be 0.6.

ρ_{in} Required minimum distribution for individual i in year n. Expressed in fractions which are determined from IRS tables. Simple if spouses are less than 10 years apart, a little more complex otherwise, as the age of both spouses need to be taken into account. Current implementation only supports spouses being less that 10 years apart.

Γ_{tn} Bound for Federal income tax bracket. We define $\Gamma_{(-1)n} := 0$, so that Γ_{0n} is the upper bound for the 10% tax bracket in year n. As the filing status can change for couples, and so can the tax code, there will be changes over n.

Δ_{tn} Difference between upper bound Γ_t and lower bound Γ_{t-1} of a federal income tax bracket,

$$\Delta_{tn} = \Gamma_{tn} - \Gamma_{(t-1)n}. \tag{2.6}$$

Once adjusted for inflation, the taxable income can be expressed as in Eq. (2.1). These data are 7 time series. The filing status will change after the death of one spouse ($n \geq n_d$) and so these brackets need to be adjusted accordingly.

θ_{tn} Tax rate for tax bracket t in year n. Using N_t time series allows to adjust income tax rates in foreseeable future. For example, in 2024 the rates (in decimal) are .10, .12, .22, .24, .32, .35, and .37. It is speculated that the rates will revert back to 2017 rates in 2026 with .10, .15, .25, .28, .33, .35, and .396. See Eq. 2.15 for its use.

α_{ijkn} Desired asset allocation for savings account j of individual i in assets class k during year n. Allocation ratios come in many flavors as they could be specified globally between individuals and accounts as α_{kn}, for example. When specified by the user, allocation

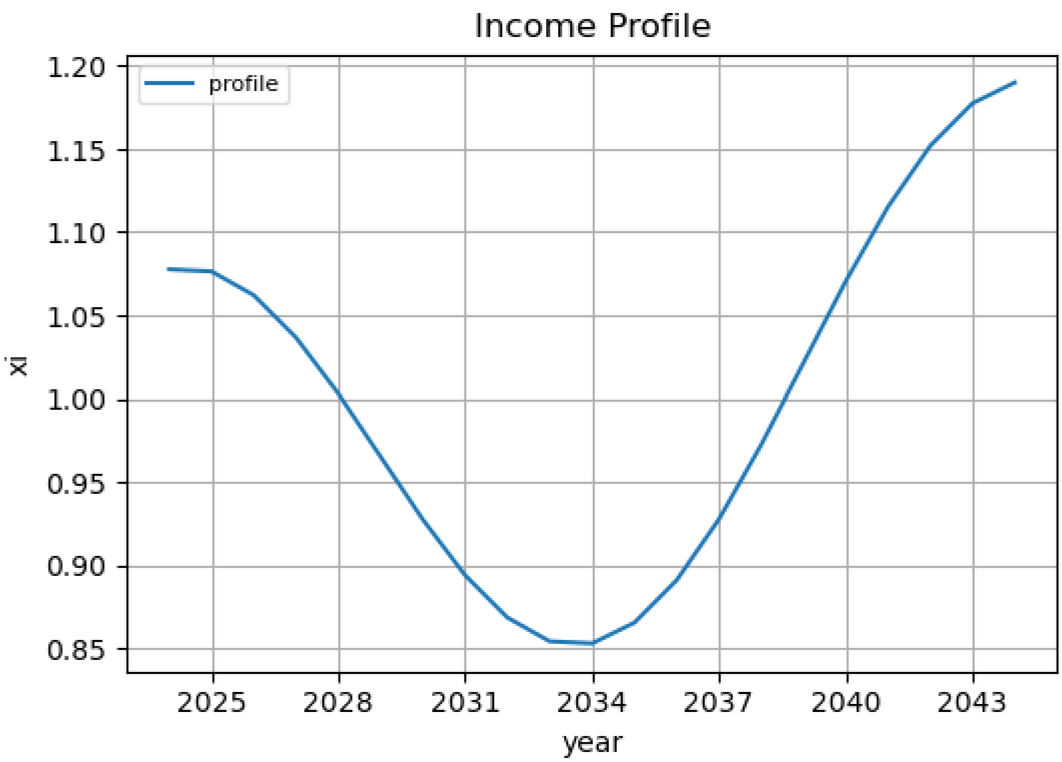

Figure 2.1: Example of a spending profile with 15% cosine factor and a 12% linear profile.

ratios typically involve two values, one at the beginning of the plan α_{ijk0} and the other at the end $\alpha_{ijkN_{n-1}}$. Then, intermediate values are interpolated either using a linear relation,

$$\alpha_{ijkn} = a + \frac{n}{N_n - 1}(b - a), \tag{2.7}$$

or an s-curve as in

$$\alpha_{ijkn} = a + \frac{(b - a)}{2}(\tanh((n - n_1)/n_2) + 1), \tag{2.8}$$

where n_1 is the number of years ahead when inflection point will occur, and n_2 is the width (in years) of the transition. Constants n_1 and n_2 are selected by the user. Using $a = \alpha_{ijk0}$ and $b = \alpha_{ijkN_n-1}$ is an approximation as values of ± 1 are only reached at $\pm\infty$. More precise bounds a' and b' can be determined by solving a 2×2 system of equations leading to

$$\begin{aligned} a' &= (a - k_{12}b')/k_{11} \\ b' &= ((b - (k_{21}/k_{11})a)/(k_{22} - (k_{21}/k_{11})k_{12}), \end{aligned} \tag{2.9}$$

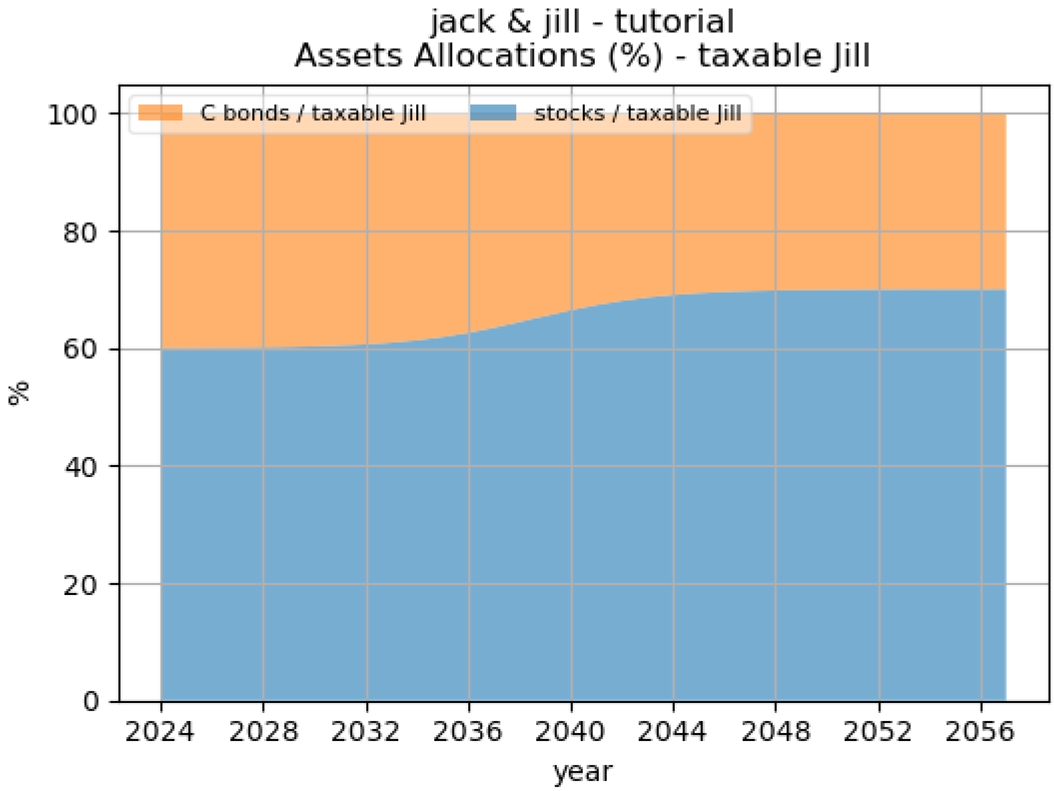

Figure 2.2: Example of an allocation portfolio with 60/40% stocks/bonds transitioning to 70/30%.

where

$$\begin{aligned} k_{11} &= \frac{1}{2}(1 - \tanh((0 - n_1)/n_2)) \\ k_{12} &= \frac{1}{2}(1 + \tanh((0 - n_1)/n_2)) \\ k_{21} &= \frac{1}{2}(1 - \tanh((N_n - 1 - n_1)/n_2)) \\ k_{22} &= \frac{1}{2}(1 + \tanh((N_n - 1 - n_1)/n_2)). \end{aligned} \tag{2.10}$$

These interpolation functions allow the allocation ratios to gradually change or *glide* during retirement.

It is also possible to have a coarser granularity on the portfolio by having an asset allocation scheme defined on a sum of accounts. For example, allocation can be coordinated between accounts leading to α_{ikn}, or even between spouses as α_{kn}. For any of these cases, it is assumed that weights are always properly scaled so that

$$\begin{aligned} \sum_k \alpha_{ijkn} &= 1, \\ \text{or} \quad \sum_k \alpha_{ikn} &= 1, \\ \text{or} \quad \sum_k \alpha_{kn} &= 1, \end{aligned} \tag{2.11}$$

depending on the scheme selected.

$\Lambda_{in}^{\pm}$ Big-ticket item requested by individual i in year n. These are large expenses or influx of money that can be planned. Therefore, $\Lambda^{\pm}$ can be positive (e.g., sell a house, inheritance) or negative (e.g., buy a house, large gifts).

π_{in} Sum of pension benefits for individual i in year n. These amounts are typically specified along with the ages at which these benefits begin.

ζ_{in} Social security benefits for individual i in year n. Starting age and the passing of one individual for spouses will determine the time series. $\bar{\zeta}_{in}$ is the same series adjusted for inflation.

ϵ_{N_n} Desired amount to leave as a bequest at the end of the final year of the plan, $N_n - 1$, which is the beginning of year N_n. This amount is the after-tax value of the estate for the heirs. See parameter ν.

κ_{ijkn} Sum of contributions to savings account j made by individual i during year n. We assume that contributions are made at half-year to balance regular contributions, and that individual selects asset class k when depositing. In practice, a contribution amount κ_{ijn} is specified in which case

$$\kappa_{ijkn} = \alpha_{ijkn}\kappa_{ijn}. \tag{2.12}$$

ω_{in} Sum of wages obtained by individual i during year n. Do not confuse wages ω with withdrawals w.

μ Dividend return rate in taxable accounts. Average is little above 2% for S&P 500.

ν Heirs income tax rate to be applied on the tax-deferred portion of the estate. This is not an estate tax but rather the federal income marginal tax rate for the heirs.

ϕ_j Fraction of savings account j that is left to surviving spouse i_s as a beneficiary at the death of individual i_d, the first spouse to pass.

ψ Income tax rate on long-term capital gain and qualified dividends, typically 15%.

η Spousal ratio for withdrawals.

2.4 Intermediate variables

We use intermediate variables for conciseness or clarity, but they are ultimately replaced in the final formulation. All intermediate variables are in uppercase letters.

G_n Taxable ordinary income in year n. Sum of wages, pension, social security benefits, all withdrawals from tax-deferred accounts, including Roth conversions, and gains from securities (i.e., all gains except those from the ($k = 0$) equities) in the ($j = 0$) taxable account, including contributions κ, minus the standard deduction,

$$\begin{aligned} G_n &= \sum_i [\omega_{in} + .85\bar{\zeta}_{in} + \pi_{in}] \\ &\quad + \sum_{ik} [w_{i1kn} + x_{ikn} + (1 - \delta(k, 0))(b_{i0kn} + .5\kappa_{i0kn})\tau_{kn}] - \bar{\sigma}_n. \end{aligned} \tag{2.13}$$

Social security is indexed for inflation and is assumed to be taxed at 85%. We use a discrete Kronecker δ function for selecting gains from non-equity assets in taxable accounts. These gains are taxed as ordinary income.

Q_n Qualified dividends and long-term capital gains obtained in year n. They only involve gains occurring in taxable savings accounts ($j = 0$) that were obtained from equities ($k = 0$), or sales of stocks for rebalancing allocations in taxable savings account. For simplicity, we assume that all sales only generate long-term capital gains and that all dividends are qualified, resulting in

$$Q_n = \sum_i \left[(b_{i00n} + .5\kappa_{i00n})\mu + (w_{i00n} + b^-_{i00n}) \max(0, \tau_{0n}) \right]. \tag{2.14}$$

A formulation where only a fraction of dividends are qualified can easily be implemented with the addition of another parameter. The first terms on the right-hand side represent the amount of equities ($k = 0$) in the ($j = 0$) taxable savings account plus half the yearly contributions. The last terms account for withdrawals w of equities assumed to have been purchased a year ago and others b^- sold at the end of the year as required for rebalancing the account. It does not account for losses, but a market drop would most likely result in stock purchase rather than sale (and therefore $b^- = 0$). For withdrawals, we make the assumption of selling the most recent stocks which would not be accurate in situations where the taxable savings account is being depleted slowly.

T_n Amount of income tax paid on taxable ordinary income G_n in year n. This is the taxes paid on ordinary income expressed as the sum of the amounts paid in each tax bracket as

$$T_n = \sum_t f_{tn} \bar{\Delta}_{tn} \theta_{tn}. \tag{2.15}$$

Notice that G_n is also defined by Eq. (2.1), and that optimal values of f_{tn} have to minimize T_n when either the bequest or the desired net spending are maximized. As the product $\bar{\Delta}_{tn}\theta_{tn}$ does not garrantee that the tax brackets will be favored monotonically, a more natural choice is to use the combined variable $\bar{F}_{tn} = f_{tn}\bar{\Delta}_{tn}$ in the optimization.

Given that the rates on tax brackets θ_{tn} are always increasing monotonically, we are then naturally filling in the lower brackets first when optimizing. In that case

$$T_n = \sum_t \bar{F}_t \theta_{tn}. \tag{2.16}$$

U_n Amount of income tax paid on long-term capital gains and qualified dividends in year n,

$$U_n = \psi Q_n. \tag{2.17}$$

Although it is not always the case, we assume that qualified dividends and long-term capital gains are taxed at the same preferential rate ψ.

Chapter 3

Formulation with imposed asset allocation ratios

We first present the case where the sums of assets in each savings accounts b_{ijn} are known over which we assume a prescribed asset allocation ratios. The amount in each asset class k for b_{ijkn} is simply obtained from $\alpha_{ijkn}b_{ijn}$ in this case. This formulation assumes that the accounts are always balanced. This is a reasonable assumption given the auto-balancing feature offered by many financial service providers.

The benefit of this approach is that it has less variables and that only the sums of all asset classes in each savings account need to be known. The rate of return of the account is then simply the product of the account balance with the sum of the rates of return weighted according to the desired allocation ratio. This approach allows us to eliminate k by summing over it and rewrite

$$\sum_k b_{ijkn+1} \quad = \quad \sum_k b_{ijkn}(1 + \tau_{kn}) + \ldots, \tag{3.1}$$

as the simpler expression

$$\begin{aligned} b_{ijn+1} \quad &= \quad b_{ijn} \sum_k \alpha_{ijkn}(1 + \tau_{kn}) + \ldots, \\ &= \quad b_{ijn}\mathcal{T}_{ijn} + \ldots, \end{aligned} \tag{3.2}$$

where

$$\mathcal{T}_{ijn} \quad := \quad 1 + \sum_k \alpha_{ijkn}\tau_{kn}. \tag{3.3}$$

3.1 Constraints

Required minimum distributions (RMDs) Withdrawals from the ($j = 1$) tax-deferred savings accounts must be larger or equal than the required minimum distributions, therefore

$$\sum_k [w_{i1n} - \rho_{in}b_{i1n}] \geq 0. \tag{3.4}$$

As b_{ijn} are the balances at the beginning of year n, they are also the balances at December 31 of the previous year, which is the amount from which the IRS bases the RMDs. Eq. 3.4 has to hold for each year n and each individual i, and therefore, there are $i \times N_n$ such equations (even when $\rho_{in} = 0$). These constraints avoid paying the 50% penalty on amounts not withdrawn when RMDs are required. Note that aggregate rules need to be considered separately as this approach only considers the sum of assets in a class with similar tax treatment (e.g., IRA and 401k).

Income tax brackets Taxable ordinary income is divided in tax brackets as defined in Eq. (2.1), i.e.,

$$\begin{aligned} G_n &= \sum_t f_{tn} \bar{\Delta}_{tn}, \\ 0 &\leq f_{tn} \leq 1. \end{aligned} \tag{3.5}$$

Given this definition, all bracket fractions f must be positive and smaller than or equal to 1, imposing an upper bound on f. Because $\theta_t > \theta_{t'}$ when $t > t'$, we can exploit this monotonically increasing series to fill the lower tax brackets using the minimization algorithm. For that purpose, we need to introduce the combined variable

$$\bar{F}_{tn} = f_{tn} \bar{\Delta}_{tn}, \tag{3.6}$$

implying that

$$G_n = \sum_t \bar{F}_{tn}, \tag{3.7}$$

with

$$\bar{F}_{tn} \leq \bar{\Delta}_{tn}. \tag{3.8}$$

Account balances Contributions are assumed to be made at half-year to better represent periodic contributions made throughout the year. As we already mentioned, the account balance at the end of a year is the same as the balance at the beginning of the following year. Changes include contributions κ, distributions and withdrawals w, conversions x, and growth τ. We track each savings account j separately, and the tax-deferred account is coupled to the tax-exempt account through Roth conversions.

The timing of Roth conversions, withdrawals, and deposits determines the coupling between these variables, and is worth a detailed discussion. First, the financial aspect, and then the algorithmic one. For the former, some financial advisors would recommend making conversions at the beginning of the year, while making withdrawals at the end. Obvioulsy, simulators would always yield higher numbers for this scenario, as the moneys needed to pay the regular bills stayed in the bank until the end of the year. More realistically, it would be more accurate to assume the withdrawals at mid-year, to better represent evenly distributed withdrawals. So, financially, conversions at the beginning of the year, and withdrawals at the end make good sense.

From an optimization point of view, any Roth conversion taking place in a year of positive returns and before account withdrawals will always favor a scenario of making a Roth conversion at the beginning of the year, followed by a tax-exempt withdrawal over the simpler scenario of having a direct withdrawal from the tax-deferred account. This is because the first scenario involves gains which are tax-free over the year, while the second does not. Moving account withdrawals at the beginning of the year, and the conversions in mid-year solves this artificial bias. Another option is the make the conversion and withdrawal events synchronous but introduce binary variables $z_{in} \in \{0, 1\}$ with the following constraints:

$$\begin{aligned} 0 \leq \quad & z_{in}M - x_{in} \quad \leq M, \\ 0 \leq \quad & z_{in}M + w_{i1j} \quad \leq M. \end{aligned} \tag{3.9}$$

Here, M is a large number such as 10^7, larger than what x and w can possibly be.

Roth conversions are assumed to be made at mid-year, while withdrawals are made at the beginning of the year, and surplus deposits, if needed, are made at the end of the year. Timing controls which terms get multiplied by the rate of return $(1 + \tau)$ of this particular asset. Therefore,

$$\begin{aligned} b_{ij(n+1)} \quad = \quad & [b_{ijn} + .5\kappa_{ijn}](1 + \mathcal{T}_{ijn}) + (\delta(j, 2) - \delta(j, 1))x_{in}(1 + \mathcal{T}_{ijn}/2) \\ & +.5\kappa_{ijn} - w_{ijn}(1 + \mathcal{T}_{ijn}) + \delta(j, 0)d_{in}, \end{aligned} \tag{3.10}$$

where we use discrete Kronecker δ functions for selecting the specific accounts involved in Roth conversions. These conversions are made such that asset allocation ratios in the sending and receiving accounts are unchanged.

Bringing all variables to the left-hand side, this gets rewritten as

$$\begin{aligned} & b_{ijk(n+1)} - [b_{ijkn} + (\delta(j, 2) - \delta(j, 1))x_{ikn}](1 + \tau_{kn}) \\ & \qquad - b^+_{ijkn} + b^-_{ijkn} + w_{ijkn} - \delta(j, 0)d_{ikn} \quad = \quad \kappa_{ijkn} \left(1 + \tau_{kn}/2\right). \end{aligned} \tag{3.11}$$

For each year, if the Roth conversions are at the beginning while all withdrawals are at the end of the year, there will be a tax advantage in performing a Roth conversion followed by a tax-exempt distribution at the end of the same year, when compared to a distribution from a tax-deferred account. To remove this unnecessary bias, the withdrawals from the tax-exempt accounts have to be synchronous with Roth conversions. This could be achieved by moving the $w_i 2kn$ to the beginning of the year simply by multiplying with $(1 + \delta(j, 2)\tau_{kn})$. However, this creates a condition in which the w and x variables are unbound. Solution might be to move to an MILP solver and ensure that either $x = 0$ or $w_{i2kn} = 0$.

There is also another equivalence involving an overwithdrawal from the taxable account immediately followed by a deposit in the same account. This can also easily be removed by post-processing the data as accounts balances are not affected.

The initial savings account balances are imposed through additional constraints as

$$b_{ijk0} = \alpha_{ijk0}\beta_{ij}, \tag{3.12}$$

where we applied the initial allocation ratios α on the initial account balances β_{ij}.

It is also possible to eliminate the account balance variables b through iterative substitutions, leading to

$$\begin{aligned} b_{ijkN} \quad = \quad & b_{ijk0} \prod_{q=0}^{N-1} (1 + \tau_{kq}) + (\delta(j, 2) - \delta(j, 1)) \sum_{n=0}^{N-1} x_{ikn} \prod_{q=n+1}^{N-1} (1 + \tau_{kq}) \\ & + \sum_{n=0}^{N-1} \kappa_{ijkn}(1 + \tau_{kn}/2) \prod_{q=n+1}^{N-1} (1 + \tau_{kq}) \\ & + \sum_{n=0}^{N-1} [b^+_{ijkn} - b^-_{ijkn} - w_{ijkn} + \delta(j, 0)d_{ikn}] \prod_{q=n+1}^{N-1} (1 + \tau_{kq}), \end{aligned} \tag{3.13}$$

where b_{ijk0} are the initial balances in year 0, and $\prod_{N}^{N-1} := 1$. This form highlights the fact that each amount, either withdrawn or deposited, can be accounted for arithmetically when taking into account the compounding effects of annual rates.

The amounts $b^{\pm}$ used for rebalancing must preserve money in each account j, every year n, and for each individual i, and therefore

$$\sum_k b^+_{ijkn} = \sum_k b^-_{ijkn}, \tag{3.14}$$

leading to the constraints

$$\sum_k [b^+_{ijkn} - b^-_{ijkn}] = 0. \tag{3.15}$$

When rebalancing, accounts cannot be negative, so we enforce the additional, and maybe unnecessary constraints that

$$b_{ijkn} - b^-_{ijkn} \geq 0. \tag{3.16}$$

The initial balances β_{ij} are one of the main inputs of this model. They are enforced as a constraint that depends on the asset allocation scheme chosen. For *account* prescribed allocations,

$$b_{ijk0} = \alpha_{ijk0}\beta_{ij}, \tag{3.17}$$

while for *individual* coordination between accounts,

$$\sum_j b_{ijk0} = \alpha_{ik0}\beta_{ij}, \tag{3.18}$$

and finally, for *spouses* type of coordination,

$$\sum_{ij} b_{ijk0} = \alpha_{k0}\beta_{ij}. \tag{3.19}$$

Maintaining the allocation is provided by the rebalancing variables $b^{\pm}$. After each rebalancing activity,

$$(b_{ijkn} + b^+_{ijkn} - b^-_{ijkn}) = \alpha_{ijkn} \sum_{k'} b_{ijk'n}, \tag{3.20}$$

where we used Eq. 3.15.

Roth Conversions Roth conversions cannot be larger than the balance in the account:

$$x_{ijn} \leq b_{i1kn}. \tag{3.21}$$

This constraint, however, is naturally satisfied when b_{ijkn} is enforced. Additional constraints x_{max} can be imposed on conversions and then the previous equation becomes

$$x_{ijn} \leq \min(b_{i1kn}, x_{max}). \tag{3.22}$$

Net spending For calculating the net spending g, we consider the cash flow of all withdrawals, wages, social security and pension benefits, and big-ticket items. Then we subtract potential surplus deposits d made to the taxable savings account and all taxes paid:

$$g_n = \sum_i [\omega_{in} + \bar{\zeta}_{in} + \pi_{in}] + \sum_{ijk} w_{ijkn} + \sum_i \Lambda_{in} - \sum_{ik} d_{ikn} - T_n - U_n. \tag{3.23}$$

Note that big-ticket items Λ can be negative or positive.

Replacing intermediate variables and bringing all variables to the left-hand side, we get

$$\begin{aligned} g_n - \sum_{ijk} w_{ijkn} + \sum_{ik} d_{ikn} + \sum_t \bar{\Delta}_{tn}\theta_{tn} f_{tn} & \\ +\psi \sum_i \left[\mu b_{i00n} + (w_{i00n} + b^-_{i00n}) \max(0, \tau_{0n})\right] &= \sum_i [\omega_{in} + \bar{\zeta}_{in} + \pi_{in}] \\ &\quad + \sum_i [\Lambda_{in} - .5\mu\psi\kappa_{i00n}]. \end{aligned} \tag{3.24}$$

Note that we do not consider market losses as we use $\max(0, \tau)$. Tax-loss harvesting is beyond the scope of this model.

We also want the net spending to be predictable and smooth. We will use

$$g_n/\bar{\xi}_n = g_0/\bar{\xi}_0, \tag{3.25}$$

where the net spending is adjusted for inflation and where we introduced parameter ξ_n allowing for additional adjustments to the overall desired spending. Note that $\bar{\xi}_0 = \xi_0$ as $\gamma_0 = 1$. This spending profile can also be used to lower the desired income at the passing of one spouse and/or to allow for more realistic spending profiles. Eq. (3.25) can be rewritten as

$$g_n\xi_0 - g_0\bar{\xi}_n = 0, \tag{3.26}$$

for the constraints to be enforced. Once g_0 is determined, the whole time series of net spending is determined.

Taxable ordinary income We connect the two definitions for G_n stated above in Eqs. (2.1) and (2.13),

$$\begin{aligned} \sum_t f_{tn}\bar{\Delta}_{tn} &= \sum_i [\omega_{in} + .85\bar{\zeta}_{in} + \pi_{in}] \\ &\quad + \sum_{ik} [w_{i1kn} + x_{ikn} + (1 - \delta(k, 0))(b_{i0kn} + .5\kappa_{i0kn})\tau_{kn}] - \bar{\sigma}_n, \end{aligned} \tag{3.27}$$

and re-arrange to move variables to the LHS as follows

$$\begin{aligned} \sum_t \bar{\Delta}_{tn} f_{tn} - \sum_{ik} [w_{i1kn} + x_{ikn} + (1 - \delta(k, 0))\tau_{kn} b_{i0kn}] &= \sum_i [\omega_{in} + .85\bar{\zeta}_{in} + \pi_{in}] - \bar{\sigma}_n \\ &\quad + .5 \sum_{ik} [(1 - \delta(k, 0))\tau_{kn}\kappa_{i0kn}]. \end{aligned} \tag{3.28}$$

Chapter 4

Mapping of decision variables

At this point, one can use one of the many algebraic modeling languages such as AMPL, GAMS, AIMMMS, and Gurobi, and code the equations above using that language, but of these applications are proprietary and require a license. These languages allow the problem to be stated at a high level and steps to cast the problem in a form suitable for solution are performed automatically. There are also object-oriented language extensions, such as Python's Pyomo and PuLP that ease the process of solving these problems. For completeness, however, we present here a simple index mapping approach that allow solving this problem using a generic linear programming solver.

To cast the problem in a form suitable for a linear programming solver, we will use a single block vector represented by the array $y[q()]$ with index-mapping functions $q()$. While this process can be achieved using slicing and reshaping in some programming languages, we will present a generic approach suitable for most programming languages.

The detailed approach presented here also allows us to determine the size of the problem to solve. We proceed alphabetically for all variables, and continue to use the convention of having index 0 for representing the first element.

First we define two generic mapping functions as

$$q_*(C, \ell_1, \ell_2, \ell_3, \ell_4; N_1, N_2, N_3, N_4) := C + \ell_1 N_2 N_3 N_4 + \ell_2 N_3 N_4 + \ell_3 N_4 + \ell_4, \tag{4.1}$$

and

$$q_C(C, N_1, N_2, N_3, N_4) := C + N_1 N_2 N_3 N_4, \tag{4.2}$$

with the constraint that $0 \leq \ell_i < N_i$.

Account balances ($\boldsymbol{b}$) For storing the savings account balances appropriately, variable b_{ijkn} needs to have one more entry $(N_n + 1)$ to store the end-of-life estate value. Therefore, we use

$$y[q_b(i, j, k, n)] = b_{ijkn}, \tag{4.3}$$

where

$$q_b(i, j, k, n) = q_*(C_b, i, j, k, n; N_i, N_j, N_k, N_n + 1) \tag{4.4}$$

and where n exceptionally runs from 0 to N_n inclusively, and therefore q_b runs from $C_b = 0$ to $C_{b^+} - 1$, where

$$C_{b^+} = q_C(C_b, Ni, N_j, N_k, N_n + 1) = [N_i N_j N_k (N_n + 1)].$$

Rebalancing amounts ($\boldsymbol{b}^{\pm}$) We have similar mappings for $b^{\pm}$ except for the range of indices,

$$y[q_{b^\pm}(i,j,k,n)] = b^{\pm}_{ijkn}, \tag{4.5}$$

where

$$q_{b^\pm}(i,j,k,n) = q_*(C_{b^\pm}, i, j, k, n; N_i, N_j, N_k, N_n), \tag{4.6}$$

and where q_{b^+} runs from C_{b^+} to $C_{b^-} - 1$, where

$$C_{b^-} = q_C(C_{b^+}, N_i, N_j, N_k, N_n) = [N_i N_j N_k(2N_n + 1)],$$

while q_{b^-} runs from C_{b^-} to $C_d - 1$, where

$$C_d = q_C(C_{b^-}, N_i, N_j, N_k, N_n) = [N_i N_j N_k(3N_n + 1)].$$

Surplus deposits ($\boldsymbol{d}$) For the surplus deposits in the taxable savings accounts d_{ikn} we will use

$$y[q_d(i,k,n)] = d_{ikn}, \tag{4.7}$$

where

$$q_d(i,k,n) = q_*(C_d, i, k, n, 0; N_i, N_k, N_n, 1) \tag{4.8}$$

with q_d running from C_d to $C_f - 1$, where

$$C_f = q_C(C_d, N_i, N_k, N_n, 1) = [N_i N_j N_k(3N_n + 1) + N_i N_k N_n].$$

Tax bracket fractions ($\boldsymbol{f}$) For tax bracket fractions f_{tn} we will use

$$y[q_f(t,n)] = f_{tn}, \tag{4.9}$$

where

$$q_f(t,n) = q_*(C_f, t, n, 0, 0; N_t, N_n, 1, 1) \tag{4.10}$$

with q_f running from C_f to $C_g - 1$, where

$$C_g = q_C(C_f, N_t, N_n, 1, 1) = [N_i N_j N_k(3N_n + 1) + (N_i N_k + N_t)N_n].$$

Net spending ($\boldsymbol{g}$) For net spending g_n we will use

$$y[q_g(n)] = g_n, \tag{4.11}$$

where

$$q_g(n) = q_*(C_g, n, 0, 0, 0; N_n, 1, 1, 1) = C_g + n, \tag{4.12}$$

with q_g running from C_g to $C_w - 1$, where

$$C_w = q_C(C_g, N_n, 1, 1, 1) = [N_i N_j N_k(3N_n + 1) + (N_i N_k + N_t + 1)N_n].$$

Withdrawals ($\boldsymbol{w}$) For withdrawals w_{ijkn} we will use

$$y[q_w(i,j,k,n)] = w_{ijkn}, \tag{4.13}$$

where

$$q_w(i,j,k,n) = q_*(C_w, i, j, k, n; N_i, N_j, N_k, N_n) \tag{4.14}$$

with q_w running from C_w to $C_x - 1$, where

$$C_x = q_C(C_w, N_i, N_j, N_k, N_n) = [N_i N_j N_k(4N_n + 1) + (N_i N_k + N_t + 1)N_n].$$

Roth conversions ($\boldsymbol{x}$) Finally, for Roth conversions x_{ikn} we will use

$$y[q_x(i,k,n)] = x_{ikn}, \tag{4.15}$$

where

$$q_x(i,k,n) = q_*(C_x, i, k, n, 0; N_i, N_k, N_n, 1) \tag{4.16}$$

with q_x running from C_x to $C_* - 1$, where

$$C_* = q_C(C_x, Ni, N_k, N_n, 1) = [N_i N_j N_k(4N_n + 1) + (2N_i N_k + N_t + 1)N_n].$$

With $N_i = 2, N_j = 3, N_k = 4, N_t = 7$ we have $120N_n + 24$ variables. For a 30-year plan, this results in 3,624 variables. If the time resolution is increased to months, that would result in 43,488 variables which is still solvable by today's standards.

4.0.1 Reverse mapping of indices

The inverse functions for the index-mapping functions will be derived for the most complex case encountered in this paper. If we have

$$z = q_*(C, i, j, k, n; N_i, N_j, N_k, N_n) := C + iN_jN_kN_n + jN_kN_n + kN_n + n, \tag{4.17}$$

then $(i,j,k,n) = q_*^{-1}(z; N_i, N_j, N_k, N_n, C)$ is obtained from

$$\begin{aligned}
n &= \mathtt{mod}(\mathtt{mod}(\mathtt{mod}(z - C, N_jN_kN_n), N_kN_n), N_n),\\
k &= \mathtt{mod}(\mathtt{mod}(z - C - n, N_jN_kN_n), N_kN_n)/N_n,\\
j &= \mathtt{mod}(z - C - n - kN_n, N_jN_kN_n)/(N_kN_n),\\
i &= (z - C - n - kN_n - jN_kN_n)/(N_jN_kN_n).
\end{aligned} \tag{4.18}$$

While this holds for all cases presented in the previous section, this can be easily simplified for cases having fewer active indices. However, some modern languages can accomplish this mapping rather easily by providing `reshape()` functions.

Chapter 5

Building constraint matrices

Let's first define generic index-mapping functions I and J as

$$\begin{aligned}
I_l(n) &= C_l + n, \\
I_l(i, n; N_n) &= C_l + iN_n + n, \\
I_l(i, j, n; N_j, N_n) &= C_l + iN_jN_n + jN_n + n, \\
\dots &= \dots
\end{aligned} \tag{5.1}$$

and so on, which would cumulatively increase row count C_l at each new instance l, similar to how we proceeded in the previous section. This allows us to build rectangular matrices by iteratively adding rows. These constraint matrices have $C_* = [N_iN_jN_k(4N_n + 1) + (2N_iN_k + N_t + 1)N_n]$ columns but will have less rows, forming an underdetermined system to be optimized using linear programming.

5.0.1 Inequality constraints

Required minimum distributions (RMDs) We rewrite the inequality constraint on required minimum distributions Eq. (3.4) using matrix $A_uy \leq u$ starting with the following N_iN_n rows,

$$\begin{aligned}
A_u[I_0(i, n), q_w(i, 1, k, n)] &= -1 \\
A_u[I_0(i, n), q_b(i, 1, k, n)] &= \rho_{in}, \\
u[I_0(i, n)] &= 0, \\
&\qquad \forall i \in \{0, \dots, N_i - 1\}, \\
&\qquad \forall k \in \{0, \dots, N_k - 1\}, \\
&\qquad \forall n \in \{0, \dots, N_n - 1\},
\end{aligned} \tag{5.2}$$

and all other elements in the same rows of A_u being 0. Notice that while b has $N_n + 1$ elements, the constraints for b go from 0 to $N_n - 1$ as there is no RMD required in the last year of the plan N_n. See Eq. (4.4).

Income tax brackets Similarly, we add $N_t N_n$ more rows to matrix $A_u y \leq u$ to express the inequality constraint in Eq. (3.5) setting an upper limit on fractions $f_{tn} \leq 1$. Therefore,

$$\begin{aligned} A_u[I_1(t,n), q_f(t,n)] &= 1, \\ u[I_1(t,n)] &= 1, \\ & \qquad \forall t \in \{0, \ldots, N_t - 1\}, \\ & \qquad \forall n \in \{0, \ldots, N_n - 1\}, \end{aligned} \tag{5.3}$$

and all other elements in the same rows of A_u being 0.

Account rebalance For Eq. (3.16), where we ensure that the rebalancing amounts do not overdraw, we add $N_i N_j N_k N_n$ rows to the upper-bound inequality matrix $A_u y \leq u$,

$$\begin{aligned} A_u[I_2(i,j,k,n), q_b(i,j,k,n)] &= -1, \\ A_u[I_2(i,j,k,n), q_{b^-}(i,j,k,n)] &= 1, \\ u[I_2(i,j,k,n)] &= 0, \\ & \qquad \forall i \in \{0, \ldots, N_i - 1\}, \\ & \qquad \forall j \in \{0, \ldots, N_j - 1\}, \\ & \qquad \forall k \in \{0, \ldots, N_k - 1\}, \\ & \qquad \forall n \in \{0, \ldots. N_n - 1\}. \end{aligned} \tag{5.4}$$

5.0.2 Equality constraints

Account balances For the equality constraint on account balances expressed in Eq. (3.11), we will define an equality constraint matrix $A_e y = v$ starting with $N_i N_j N_k N_n$ rows as

$$\begin{aligned} A_e[J_0(i,j,k,n), q_b(i,j,k,n+1)] &= 1, \\ A_e[J_0(i,j,k,n), q_b(i,j,k,n)] &= -(1 + \tau_{kn}), \\ A_e[J_0(i,j,k,n), q_x(i,k,n)] &= -(\delta(j,2) - \delta(j,1))(1 + \tau_{kn}), \\ A_e[J_0(i,j,k,n), q_{b^+}(i,j,k,n)] &= -1, \\ A_e[J_0(i,j,k,n), q_{b^-}(i,j,k,n)] &= 1, \\ A_e[J_0(i,j,k,n), q_w(i,j,k,n)] &= 1, \\ A_e[J_0(i,j,k,n), q_d(i,k,n)] &= -\delta(j,0), \\ & \qquad \forall i \in \{0, \ldots, N_i - 1\}, \\ & \qquad \forall j \in \{0, \ldots, N_j - 1\}, \\ & \qquad \forall k \in \{0, \ldots, N_k - 1\}, \\ & \qquad \forall n \in \{0, \ldots, N_n - 1\}, \end{aligned} \tag{5.5}$$

where v is

$$v[J_0(i,j,k,n)] = \kappa_{ijkn}(1 + \tau_{kn}/2). \tag{5.6}$$

The initial account balances expressed in Eq. 3.12 are imposed through

$$\begin{aligned} A_e[J_1(i,j,k), q_b(i,j,k,0)] &= 1, \\ v[J_1(i,j,k)] &= \beta_{ijk}, \\ &\quad\forall i \in \{0,\ldots,N_i-1\}, \\ &\quad\forall j \in \{0,\ldots,N_j-1\}, \\ &\quad\forall k \in \{0,\ldots,N_k-1\}, \end{aligned} \tag{5.7}$$

leading to $N_iN_jN_k$ additional rows to A_e.

For the constraint on rebalancing variables $b^{\pm}$ expressed in Eq. (3.15), we add $N_iN_jN_n$ more rows to $A_ey = v$, as

$$\begin{aligned} A_e[J_2(i,j,n), q_{b^+}(i,j,k,n)] &= 1, \\ A_e[J_2(i,j,n), q_{b^-}(i,j,k,n)] &= -1, \\ v[J_2(i,j,n)] &= 0, \\ &\quad\forall i \in \{0,\ldots,N_i-1\}, \\ &\quad\forall j \in \{0,\ldots,N_j-1\}, \\ &\quad\forall k \in \{0,\ldots,N_k-1\}, \\ &\quad\forall n \in \{0,\ldots .N_n-1\}. \end{aligned} \tag{5.8}$$

Net spending For the equality constraint on net spending expressed in Eq. (3.24), we add N_n more rows to $A_ey = v$ as

$$\begin{aligned} A_e[J_3(n), q_g(n)] &= 1, \\ A_e[J_3(n), q_w(i,j,k,n)] &= -1, \\ A_e[J_3(n), q_d(i,,k,n)] &= 1, \\ A_e[J_3(n), q_f(t,n)] &= \bar{\Delta}_{tn}\theta_{tn}, \\ A_e[J_3(n), q_b(i,0,0,n)] &= \psi\mu, \\ A_e[J_3(n), q_w(i,0,0,n)] &= \psi\max(0,\tau_{0n})/(1+\max(0,\tau_{0n})), \\ A_e[J_3(n), q_{b^-}(i,0,0,n)] &= \psi\max(0,\tau_{0n})/(1+\max(0,\tau_{0n})), \\ &\quad\forall t \in \{0,\ldots,N_t-1\}, \\ &\quad\forall i \in \{0,\ldots,N_i-1\}, \\ &\quad\forall j \in \{0,\ldots,N_j-1\}, \\ &\quad\forall k \in \{0,\ldots,N_k-1\}, \\ &\quad\forall n \in \{0,\ldots,N_n-1\}, \end{aligned} \tag{5.9}$$

where v is

$$v[J_3(n)] = \sum_i [\omega_{in} + \bar{\zeta}_{in} + \pi_{in} + \Lambda_{in} - .5\psi\mu\kappa_{i00n}]. \tag{5.10}$$

The condition of having a predictable net spending expressed as an equality in Eq. (3.26) adds

$N_n - 1$ more rows to $A_e y = v$ as

$$\begin{aligned} A_e[J_4(n), q_g(0)] &= -\bar{\xi}_n, \\ A_e[J_4(n), q_g(n)] &= 1, \\ v[J_4(n)] &= 0, \\ & \qquad \forall n \in \{1, \ldots, N_n\}. \end{aligned} \tag{5.11}$$

Taxable ordinary income Finally, for the equality constraint in Eq. (3.28) establishing taxable ordinary income, we add N_n rows to $A_e y = v$ as follows

$$\begin{aligned} A_e[J_5(n), q_f(t, n)] &= \bar{\Delta}_{tn}, \\ A_e[J_5(n), q_w(i, 1, k, n)] &= -1, \\ A_e[J_5(n), q_x(i, k, n)] &= -1, \\ A_e[J_5(n), q_b(i, 0, k, n)] &= -(1 - \delta(k, 0))\tau_{kn}, \\ & \qquad \forall t \in \{0, \ldots, N_t - 1\}, \\ & \qquad \forall i \in \{0, \ldots, N_i - 1\}, \\ & \qquad \forall k \in \{0, \ldots, N_k - 1\}, \\ & \qquad \forall n \in \{0, \ldots, N_n - 1\}, \end{aligned} \tag{5.12}$$

with

$$v[J_5(n)] = \sum_i [\omega_{in} + .85\bar{\zeta}_{in} + \pi_{in}] + .5 \sum_{ik} [(1 - \delta(k, 0))\tau_{kn}\kappa_{i0kn}] - \bar{\sigma}_n. \tag{5.13}$$

So far, A_u has $[(N_i N_j N_k + N_i + N_t)N_n]$ or $33N_n$ rows, while A_e has $[(N_k + 1)N_i N_j + 3)N_n + N_i N_j N_k - 1]$ or $33N_n + 23$ rows. For a 30-year plan, this means about 990 rows for each.

5.0.3 Other considerations

Beneficiaries Tax-exempt and tax-deferred accounts have special tax rules that allow giving part or the entire value of tax-exempt accounts to a spouse who can then consider it as his/her own. Let ϕ_j be the fraction of the account j that a spouse i_d wishes to leave to his/her surviving spouse i_s in the year $n_d < N_n - 1$ of passing. To account for that event in year n_d, Eq. (3.10) needs to be rewritten as

$$\begin{aligned} b_{ijk(n+1)} = {} & (1 - \delta(n, n_d - 1)\delta(i, i_d)) \\ & \times \Big\{ [b_{ijkn} + (\delta(j, 2) - \delta(j, 1))x_{ikn} + .5\kappa_{ijkn}] \, (1 + \tau_{kn}) \\ & + b^+_{ijkn} - b^-_{ijkn} + .5\kappa_{ijkn} - w_{ijkn} + \delta(j, 0)d_{ikn} \Big\} + (\phi_j \delta(n, n_d - 1)\delta(i, i_s)) \\ & \times \Big\{ [b_{i_d jkn} + (\delta(j, 2) - \delta(j, 1))x_{i_d kn} + .5\kappa_{i_d jkn}] \, (1 + \tau_{kn}) \\ & + b^+_{i_d jkn} - b^-_{i_d jkn} + .5\kappa_{i_d jkn} - w_{i_d jkn} + \delta(j, 0)d_{i_d kn} \Big\}. \end{aligned} \tag{5.14}$$

The first multiplier () on the right-hand side will always be one except for i_d in year $n_d - 1$ when it will be zero. This will result in emptying all accounts for i_d for years n_d and beyond. The second special multiplier () before the second set of brackets {} will always be zero except for the surviving spouse i_s in year $n_d - 1$, who will then inherit a fraction ϕ_j of account j that was scheduled to go into i_d's j account at the beginning of year n_d.

Rewriting the last equation as a constraint results in

$$
\begin{aligned}
&b_{ijk(n+1)} \\
&-(1 - \delta(n, n_d - 1)\delta(i, i_d)) \\
&\times\Big\{\left[b_{ijkn} + (\delta(j,2) - \delta(j,1))x_{ikn}\right](1 + \tau_{kn}) + b^{+}_{ijkn} - b^{-}_{ijkn} - w_{ijkn} + \delta(j,0)d_{ikn}\Big\} \\
&-(\phi_j \delta(n, n_d - 1)\delta(i, i_s)) \\
&\times\Big\{\left[b_{i_d jkn} + (\delta(j,2) - \delta(j,1))x_{i_d kn}\right](1 + \tau_{kn}) + b^{+}_{i_d jkn} - b^{-}_{i_d jkn} - w_{i_d jkn} + \delta(j,0)d_{i_d kn}\Big\} \\
&= \left[(1 - \delta(n, n_d - 1)\delta(i, i_d))\kappa_{ijkn} + (\phi_j \delta(n, n_d - 1)\delta(i, i_s))\kappa_{i_d jkn}\right](1 + \tau_{kn}/2).
\end{aligned}
\tag{5.15}
$$

We are now ready to replace Eq. (5.5) for $A_e y = v$ by

$$
\begin{aligned}
A_e[J_0(i,j,k,n), q_b(i,j,k,n+1)] &= 1, \\
A_e[J_0(i,j,k,n), q_b(i,j,k,n)] &= -(1 - \delta(n, n_d - 1)\delta(i, i_d))(1 + \tau_{kn}), \\
A_e[J_0(i,j,k,n), q_x(i,k,n)] &= -(1 - \delta(n, n_d - 1)\delta(i, i_d))(\delta(j,2) - \delta(j,1))(1 + \tau_{kn}), \\
A_e[J_0(i,j,k,n), q_{b^+}(i,j,k,n)] &= -(1 - \delta(n, n_d - 1)\delta(i, i_d)), \\
A_e[J_0(i,j,k,n), q_{b^-}(i,j,k,n)] &= (1 - \delta(n, n_d - 1)\delta(i, i_d)), \\
A_e[J_0(i,j,k,n), q_w(i,j,k,n)] &= (1 - \delta(n, n_d - 1)\delta(i, i_d)), \\
A_e[J_0(i,j,k,n), q_d(i,j,n)] &= -(1 - \delta(n, n_d - 1)\delta(i, i_d))\delta(j,0), \\
\text{when } N_i = 2 \text{ and } i = i_s & \\
A_e[J_0(i,j,k,n), q_b(i_d,j,k,n)] &= -(\phi_j \delta(n, n_d - 1)\delta(i, i_s))(1 + \tau_{kn}), \\
A_e[J_0(i,j,k,n), q_x(i_d,k,n)] &= -(\phi_j \delta(n, n_d - 1)\delta(i, i_s))(\delta(j,2) - \delta(j,1))(1 + \tau_{kn}), \\
A_e[J_0(i,j,k,n), q_{b^+}(i_d,j,k,n)] &= -(\phi_j \delta(n, n_d - 1)\delta(i, i_s)), \\
A_e[J_0(i,j,k,n), q_{b^-}(i_d,j,k,n)] &= (\phi_j \delta(n, n_d - 1)\delta(i, i_s)), \\
A_e[J_0(i,j,k,n), q_w(i_d,j,k,n)] &= (\phi_j \delta(n, n_d - 1)\delta(i, i_s)), \\
A_e[J_0(i,j,k,n), q_d(i_d,j,n)] &= -(\phi_j \delta(n, n_d - 1)\delta(i, i_s))\delta(j,0), \\
&\qquad \forall i \in \{0, \ldots, N_i - 1\}, \\
&\qquad \forall j \in \{0, \ldots, N_j - 1\}, \\
&\qquad \forall k \in \{0, \ldots, N_k - 1\}, \\
&\qquad \forall n \in \{0, \ldots, N_n - 1\},
\end{aligned}
$$

where v is

$$
v[I(i,j,k,n)] = [(1 - \delta(n, n_d)\delta(i, i_d))\kappa_{ijkn} + (\phi_j \delta(n, n_d - 1)\delta(i, i_s))\kappa_{i_d jkn}](1 + \tau_{kn}/2). \tag{5.16}
$$

While the last two equations may look cumbersome, their net effect is only to include a few more terms when $n = n_d - 1$.

Assets allocation ratios To avoid creating a quadratic problem while rebalancing the accounts, we track the account balances by asset classes using b_{ijkn}. We can then prescribe how to rebalance the accounts. The values of $b^{\pm}$ can be governed by prescribed asset allocation ratios α defined as

$$\alpha_{ijkn} = (b_{ijkn} + b^{+}_{ijkn} - b^{-}_{ijkn}) / \sum_{k'} b_{ijk'n}, \tag{5.17}$$

where k' here is just a dummy index. These asset allocation ratios can then be imposed as an equality constraint as

$$b_{ijkn} + b^{+}_{ijkn} - b^{-}_{ijkn} - \sum_{k'} \alpha_{ijkn} b_{ijk'n}] = 0, \tag{5.18}$$

which can be written as

$$\begin{aligned} A[J_6(i,j,k,n), q_b(i,j,k,n)] &= 1, \\ A[J_6(i,j,k,n), q_b^{+}(i,j,k,n)] &= 1, \\ A[J_6(i,j,k,n), q_b^{-}(i,j,k,n)] &= -1, \\ A[J_6(i,j,k,n), q_b(i,j,k',n)] &= -\alpha_{ijkn}, \\ v[J_6(i,j,k,n)] &= 0, \\ &\qquad \forall i \in \{0,\ldots,N_i - 1\}, \\ &\qquad \forall j \in \{0,\ldots,N_j - 1\}, \\ &\qquad \forall k,k' \in \{0,\ldots,N_k - 1\}, \\ &\qquad \forall n \in \{0,\ldots,N_n - 1\}. \end{aligned} \tag{5.19}$$

There are a few ways in which accounts can be coordinated to match overall asset allocation ratios. If one would like to coordinate across all savings accounts of each individual, using specified asset allocation ratios α_{ikn}, then Eq. (5.18) becomes

$$\sum_{jk'} [(\delta(k,k') - \alpha_{ikn}) b_{ijk'n}] = 0, \tag{5.20}$$

leading to

$$\begin{aligned} A[J_6(i,k,n), q_b(i,j,k',n)] &= \delta(k,k') - \alpha_{ikn}, \\ v[J_6(i,k,n)] &= 0, \\ &\qquad \forall i \in \{0,\ldots,N_i - 1\}, \\ &\qquad \forall j \in \{0,\ldots,N_j - 1\}, \\ &\qquad \forall k,k' \in \{0,\ldots,N_k - 1\}, \\ &\qquad \forall n \in \{0,\ldots,N_n - 1\}. \end{aligned} \tag{5.21}$$

Similarly, if coordination is desired between all savings accounts from both spouses specified by overall asset allocation ratios α_{kn}, then Eq. (5.21) becomes

$$\begin{aligned} A[J_6(k,n), q_b(i,j,k',n)] &= \delta(k,k') - \alpha_{kn}, \\ v[J_6(k,n)] &= 0, \\ & \qquad \forall i \in \{0,\ldots,N_i-1\}, \\ & \qquad \forall j \in \{0,\ldots,N_j-1\}, \\ & \qquad \forall k,k' \in \{0,\ldots,N_k-1\}, \\ & \qquad \forall n \in \{0,\ldots,N_n-1\}, \end{aligned} \tag{5.22}$$

reducing the number of constraints by $N_i N_j = 6$ when compared to Eq. (5.19).

If asset allocation ratios α are imposed, they should also be applied to how contributions amounts κ_{ijn} are invested, such that

$$\kappa_{ijkn} = \alpha_{ijkn}\kappa_{ijn}. \tag{5.23}$$

For other allocation schemes, just substitute $\alpha_{ijkn} = \alpha_{ikn}$ or α_{kn} depending on the scheme selected.

The same approach needs to be applied to surplus deposits, but in this case, however, it leads to additional constraints to be imposed,

$$\sum_{k'}[(\delta(k,k') - \alpha_{i0kn})d_{ik'n}] = 0, \tag{5.24}$$

adding $N_i N_k N_n$ additional rows to A_e.

Assets allocation could have been handled easily by assuming that the accounts are always rebalanced and only using a single multiplier Ξ, defined as

$$\Xi_{ijn} = \sum_k \alpha_{ijkn}\tau_{kn}, \tag{5.25}$$

to compute to return on the total balance of each savings account. However, a benefit of tracking all asset classes in each account separately is that it allows us to optimize over asset allocations through rebalancing, at the cost of solving a larger problem. If the constraints in Eq. (5.19), Eq. (5.21), or Eq. (5.22) are not imposed, the allocation ratios would then be optimized according to the chosen objective function, and calculated using Eq. (5.17).

Spousal deposits and withdrawals In order to keep the problem linear, a simple constraint that can be imposed on surplus deposits to be made in taxable savings accounts is to specify a spousal ratio η such as

$$d_{0kn} = \eta d_{1kn}. \tag{5.26}$$

A similar spousal ratio can be imposed on withdrawals from tax-deferred accounts

$$w_{01kn} = \eta w_{11kn}, \tag{5.27}$$

but this can cause drawing an account empty while the other spousal account is not.

Chapter 6

Objective functions

The objective function is a simple scalar defined as $c \cdot y$ that will be minimized.

Maximum net spending There are a few ways by which a retirement plan can be optimized. For maximizing the net spending under the constraint of a desired bequest, we introduce the following relation

$$E_n = \sum_{ijk} (1 - \nu\delta(j, 1)) b_{ijkn}, \tag{6.1}$$

which is the value of the estate in nominal dollars at year n, taking into consideration the heir's marginal income tax rate on the ($j = 1$) tax-deferred account. In this situation, the concept of surplus deposit is no longer valid, and therefore

$$d_{ikn} = 0, \tag{6.2}$$

implemented as

$$\begin{aligned} A_e[I(6), q_d(i, k, N_n)] &= 0 \\ v[I(0)] &= \epsilon_{N_n} \gamma_{N_n} \\ &\quad \forall i \in \{0, \ldots, N_i - 1\}, \\ &\quad \forall k \in \{0, \ldots, N_k - 1\}, \\ &\quad \forall n \in \{0, \ldots, N_n - 1\}. \end{aligned} \tag{6.3}$$

For a desired bequest ϵ_{N_n}, expressed in today's dollars, the final amount in year N_n will need to be

$$E_{N_n} = \bar{\epsilon}_{N_n} = \epsilon_{N_n} \gamma_{N_n}. \tag{6.4}$$

Fixing a bequest value amounts to adding the following constraint

$$\sum_{ijk} b_{ijkN_n} (1 - \nu\delta(j, 1)) = \epsilon_{N_n} \gamma_{N_n}, \tag{6.5}$$

which would add one more row to $A_e y = v$ as

$$\begin{aligned} A_e[I(0), q_b(i,j,k,N_n)] &= (1 - \nu\delta(j,1)) \\ v[I(0)] &= \epsilon_{N_n}\gamma_{N_n} \\ &\quad \forall i \in \{0,\ldots,N_i - 1\}, \\ &\quad \forall j \in \{0,\ldots,N_j - 1\}, \\ &\quad \forall k \in \{0,\ldots,N_k - 1\}, \end{aligned} \tag{6.6}$$

where $I(0)$ is used only to provide the proper row offset C_l. See Eq. (5.1).

For maximizing the net spending under the constraint of a fixed bequest, one has simply to minimize the inner product $c \cdot y$, where c is

$$c[q_g(0)] = -1, \tag{6.7}$$

and 0 otherwise. See Eq. 3.26.

Maximum bequest If, on the other hand, one would like to maximize the bequest under the constraint of a desired net spending g_o, one would add the following row to $A_e y = v$

$$\begin{aligned} A_e[I(0), q_g(0)] &= 1, \\ v[I(0)] &= g_o. \end{aligned} \tag{6.8}$$

The objective function would then be derived from Eq. (6.1) as minimizing the inner product $c \cdot y$, where c is

$$\begin{aligned} c[q_b(i,j,k,N_n)] &= -(1 - \nu\delta(j,1)), \\ &\quad \forall i \in \{0,\ldots,N_i - 1\}, \\ &\quad \forall j \in \{0,\ldots,N_j - 1\}, \\ &\quad \forall k \in \{0,\ldots,N_k - 1\}, \end{aligned} \tag{6.9}$$

and 0 otherwise.