Formulation of the optimization model in Owl

Martin-D. Lacasse

April 22, 2025

1. Introduction

This document describes the mathematical model underlying 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, federal income tax, contributions, return rates, Medicare premiums, Roth conversions, and desired income amongst many other things.

The approach is described here mathematically and the Python implementation follows the structure and notation presented in this document. The intent of this document is to provide a guide to the source code for any individual desiring to extend the model to other cases. The mathmatical description is not tied to any specific mathematical programming language so that it remains as generic as possible.

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 introduced 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 constraint 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 Index of year being modeled. Period being modeled runs from the beginning of year 0 to the end of year $N_n - 1$, and therefore $N_n + 1$ years are actually considered. Year 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, the end of year $n_d - 1$ is when the first individual is assumed to pass while the survivor will decease at the end of year $N_n - 1$ of the plan.

q Index for income brackets related to Medicare premium adjustments. q goes from 0 to $N_q - 1$, from low to high. At the time of writing, there are $N_q = 6$ Medicare premium adjustments in the federal tax code.

t Federal income tax bracket. t goes from 0 to $N_t - 1$, from low to high. At the time of writing, there are $N_t = 7$ federal income tax brackets in the current federal tax code.

2.2 Variables

We will use lowercase roman letters to represent variables. This is done to separate variables from parameters, which will be represented using Greek letters or caligraphic fonts. All variables are assumed to take only non-negative values (≥ 0 inequality). Variables are resolved by the optimizer while parameters are prescribed by the user, historical data, or tax laws.

b_{ijn} Balance for individual i in savings account j at the beginning of year n. When we consider each asset class k, the variable b_{ijkn} is used instead.

d_{in} Deposit of year-n net spending surplus in taxable account of individual i. These deposits are coming from the surplus s_n, distributed to spousal taxable accounts depending on parameter η.

e_n Adjusted standard exemption for year n. This is a variable as the taxable income can sometimes be less that the standard exemption $\bar{\sigma}_n$, leading to a negative taxable income if the inflation-adjusted standard exemption is simply subtracted from the gross taxable income G_n in years of low income.

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_{tn} can be found in the section describing the parameters below.

g_n Net spending in year n.

m_n Medicare costs in year n.

s_n Surplus of funds during year n, most likely caused by required minimum distributions (RMDs) or influx of money from big-ticket items (inheritance, gifts, sale of a house, etc.).

w_{ijn} Withdrawal from account j belonging to individual i at the beginning of year n. For the ($j = 1$) tax-deferred savings account, w_{i1n} is referred to as a distribution for tax purposes as it is a taxable withdrawal, and will always satisfy required minimum distributions.

x_{in} Roth conversion performed by individual i during year n. These events are taxable as ordinary income.

z_* Binary variables will all be designated by the letter z, but a superscript will be used to distinguish the different families. For example, z_{qn}^m will be used for binary variables associated with Medicare calculations.

2.3 Parameters

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

β_{ij} Initial balances in savings accounts. These amounts are used to initialize b_{ij0}.

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. In other words, investing in cash yields constant dollars (return just matches inflation).

γ_n Cumulative inflation at the beginning of year n computed as the product

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

with $\gamma_0 := 1$, and where n' is a dummy index. As the time span of interest goes from the first year to the beginning of the year following the last year, variable γ_n will have $N_n + 1$ elements. Parameters indexed for inflation will be indicated by a bar on top as in $\bar{\sigma}_n$. See the next entry for a specific example.

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 exemptions after 65 of age, for example. It is a simple time series which can include any foreseeable changes in the tax code, or change in filing status due to the passing of one spouse for $n \geq n_d$. The value of $\bar{\sigma}_n$ is an upper bound for variable e_n.

ξ_n Spending profile. This is a time series that multiplies a basis for the desired net spending amount. It is $\xi_n = 1, \forall n$ for a flat profile, or can be a *smile* profile allowing for more money at the start of retirement and modulating it over 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 is adjusted accordingly.

χ Factor to reduce spending profile after the passing of one spouse. It is typically assumed to be 0.6. That is, we are assuming that the surviving spouse can live with 60% of the net spending amount.

ρ_{in} Required minimum distribution for individual i in year n. Expressed in fractions which are determined from IRS tables. These tables are simple if spouses are less than 10 years apart, but 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. An error message is generated in these cases and the calculation is aborted.

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

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, Γ_{tn} will be changing 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 changes after the passing of one spouse ($n \geq n_d$) and these brackets are 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 the 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. (2.19) 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

ratios typically involve two values, one at the beginning of the plan α_{ijk0} and the other at the end $\alpha_{ijkN_{n}-1}$, or $\alpha_{ijkn_{d}}$ for a spouse passing before the other. 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 can be adjusted by the user. Default values are $n_1 = 15$, and $n_2 = 5$, meaning that the transition center will occur in 15 years, taking place from $15 - 5$ years to $15 + 5$ years from now. Using $a = \alpha_{ijk0}$ and $b = \alpha_{ijkN_{n}-1}$ is an approximation as values of ± 1 are only reached at $\pm\infty$ for a hyperbolic tangent. More precise bounds a' and b' for matching the desired start and end values can be determined by solving a 2×2 system of equations leading to

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

where

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

These interpolation functions allow the allocation ratios to gradually change or *glide* during retirement. Fig. 2.2 provides an example of an *s-curve* gliding allocation ratios.

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.

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

$\mathcal{T}_{ijn}$ When the allocation ratios α_{ijkn} are prescribed, it is more convenient to express the return rates as

$$\mathcal{T}_{ijn} = \sum_k \alpha_{ijkn}\tau_{kn}. \tag{2.12}$$

$\Lambda_{in}^{\pm}$ 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).

λ Allowed deviation from the desired net spending profile during one year. Parameter λ can be better understood as a percentage. If $\lambda = 0.10$, then net spending amount is allowed to vary by up to 10% from the prescribed profile.

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. Pensions can optionally be indexed for inflation.

ζ_{in} Social security benefits for individual i in year n. Starting age and the passing of one

7

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 expressed in today's dollars. See parameter ν for the heirs tax rate.

κ_{ijn} Sum of contributions to savings account j made by individual i during year n. We assume that contributions are made at half-year to better represent periodic contributions made throughout the year. In practice, a contribution amount κ_{ijn} is specified in which case the contribution to each asset class is

$$\kappa_{ijkn} = \alpha_{ijkn}\kappa_{ijn}. \tag{2.13}$$

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

$\mathcal{C}_q$ Cost of Medicare for bracket q of modified adjusted gross income (MAGI). This includes Medicare part B premiums and any additional Income-Related Monthly Adjusted Amount (IRMAA). When adjusted for inflation, this becomes $\bar{\mathcal{C}}_{qn} = \gamma_n\mathcal{C}_q$. There are $N_q = 6$ brackets for IRMAA adjustments forming a piecewise constant function.

$\mathcal{L}_q$ Income brackets used to determine Medicare adjustments based on the modified adjusted gross income. When adjusted for inflation, this becomes $\bar{\mathcal{L}}_{qn} = \gamma_n\mathcal{L}_q$. While there are $N_q = 6$ brackets for IRMAA adjustments forming a piecewise constant function, $\mathcal{L}_q$ has $N_q + 1$ elements as the last element is an arbitrary large number bounding the last bracket. See Fig. 3.1 for a visual representation.

μ Dividend return rate for equities 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 surplus deposits, which goes from 0 to 1, as the fraction that goes to the $i = 1$ spouse's account. Therefore, a surplus s_n in year n will result in a deposit d in the taxable account of individual i as

$$\begin{aligned} d_{0n} &= (1 - \eta)s_n \\ d_{1n} &= \eta s_n. \end{aligned} \tag{2.14}$$

This choice is such that we can set a value depending on the surviving individual $\eta = i_s$ for $n \geq n_d$, after the passing of i_d. Default value is $(N_i - 1)/2$, i.e., 0.5 for couples and 0 for single individuals. When the beneficiary of the savings accounts is not the other spouse, i.e., when $\phi_j \neq 1, \forall j$, it is recommended that η be set to i_d so that all surplus get deposited to i_d's accounts, thus avoid loopholes when optimizing for the final bequest.

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, which are taxed as capital gains) in the $(j = 0)$ taxable account, including contributions κ, minus the standard deduction,

$$
\begin{aligned}
G_n \;=\; & \sum_i [\omega_{in} + 0.85\bar{\zeta}_{in} + \pi_{in}] - e_n + \\
& \sum_i [w_{i1n} + x_{in}] + \\
& \sum_{i,k} [(1 - \delta(k,0))(b_{i0n} - w_{i0n} + d_{in} + 0.5\kappa_{i0n})\alpha_{i0kn}\tau_{kn}]
\end{aligned}
\tag{2.15}
$$

Social security is indexed for inflation and is assumed to be taxed at 85%. Pensions can optionally be indexed for inflation. We use a discrete Kronecker δ function for selecting gains from non-equity assets in taxable accounts, but we could have equivalently excluded $k \neq 0$ in the sum. These gains are all taxed as ordinary income. Here, we assumed that withdrawals and deposits in the taxable account are taking place at the beginning of the year, while contributions, if any, are taking place in mid-year.

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

$$
Q_n = \sum_i \alpha_{i00n} \left[(b_{i0n} - w_{i0n} + d_{in} + 0.5\kappa_{i0n})\mu + w_{i0n}\max(0, \tau_{0(n-1)}) \right].
\tag{2.16}
$$

A formulation where only a fraction of dividends are qualified can easily be implemented with the addition of another parameter. Notice that we are using return rates from the previous year. The first terms on the right-hand side represent dividends generated by equities $(k = 0)$ in the $(j = 0)$ taxable savings account plus half the yearly contributions. The second term account for withdrawals w of equities assumed to have been purchased a year ago. It does not account for losses, but a market drop would most likely result in stock purchase rather than sale. 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. An implementation keeping track of stock purchases and sales is beyond the goal of providing a guide for retirement decisions.

P_n Amount of 10% early withdrawal penalty in year n,

$$
P_n = 0.10 \sum_{i,j \neq 0} (1 - \mathcal{H}(n - n_{i,59}))w_{ijn}.
\tag{2.17}
$$

Here, $H(n - n_{i,59})$ is a Heavyside step function which is 0 or 1, depending on the sign of its argument:

$$\mathcal{H}(x) := \begin{cases} 0 & x < 0 \\ 1 & x \geq 0. \end{cases} \tag{2.18}$$

The parameter $n_{i,59}$ is the year index when individual i turns 59, or 0 if the individual is already 59 years old or older at the beginning of the plan.

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

$$T_n = \sum_t f_{tn} \bar{\Delta}_{tn} \theta_{tn}. \tag{2.19}$$

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 guarantee to be ordered monotonically, a more practical 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, the lower brackets will be filled in first when optimizing. In that case

$$T_n = \sum_t \bar{F}_{tn} \theta_{tn}. \tag{2.20}$$

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

$$U_n = \psi Q_n. \tag{2.21}$$

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.

10

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 and robot advisers.

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 equations such as

$$\sum_k b_{ijk(n+1)} \;\; = \;\; \sum_k b_{ijkn}(1 + \tau_{kn}) + \dots, \tag{3.1}$$

for the annual evolution of account balances from year n to year $n+1$ as the simpler expression

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

where

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

This is a consequence that the allocation ratios are normalized to unity, i.e.,

$$\sum_k \alpha_{ijkn} = 1. \tag{3.4}$$

In this formulation where the α_{ijkn} are prescribed, we will use $\mathcal{T}_{ijn}$ to add the market returns to the savings balances.

11

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, and therefore,

$$w_{i1n} - \rho_{in}b_{i1n} \geq 0. \tag{3.5}$$

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. (3.5) 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.,

$$G_n = \sum_t f_{tn}\bar{\Delta}_{tn},$$
$$0 \leq f_{tn} \leq 1. \tag{3.6}$$

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 of rates with t to have the minimization algorithm naturally fill the lower tax brackets. For that purpose, we need to introduce the combined variable

$$\bar{F}_{tn} = f_{tn}\bar{\Delta}_{tn}, \tag{3.7}$$

implying that

$$G_n = \sum_t \bar{F}_{tn}, \tag{3.8}$$

with

$$0 \leq \bar{F}_{tn} \leq \bar{\Delta}_{tn}. \tag{3.9}$$

Then, ordinary income tax is easily calculated by Eq. (2.20),

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

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, surplus deposits d, and growth τ on the account through the year. For each spouse i, we track each savings account j separately, and tax-deferred accounts are coupled with the corresponding tax-exempt account through Roth conversions.

The timing of Roth conversions, withdrawals, and deposits brings additional coupling between these variables, and is worth a detailed discussion. First, the financial aspects, and then the algorithmic ones. For the former, some financial advisors would recommend making Roth conversions

at the beginning of the year, while making withdrawals at the end. Obviously, financial simulators would always yield higher numbers when using this scenario, as the moneys needed to pay the regular bills stayed in the bank until the end of the year. More realistically, however, it would be more accurate to assume withdrawals at mid-year, to better represent evenly distributed withdrawals. So, financially, conversions at the beginning of the year, and withdrawals at mid-year make good sense. Conversions are also typically best when timed with market downturns, which are obviously not always at the beginning of the year.

Now, let's look at the optimization side of these transactions. During years of positive returns, a direct withdrawal from the tax-deferred account at mid-year will always be unfavorable when compared to a Roth conversion at the beginning of the year, followed by a tax-exempt withdrawal later in the same year. This is because the second scenario involves gains which are tax-free over the half-year, while the first one does not. Moving account withdrawals at the beginning of the year, and the conversions in mid-year can only solve part of this artificial bias under specific conditions.

To solve these spurious scenarios, it would be desirable to make the following exclusions between surplus s_n, withdrawals w_{ijn}, and conversions x_{in}:

$$s_n \quad \texttt{XOR} \quad w_{i0n},$$
$$s_n \quad \texttt{XOR} \quad w_{i2n},$$
$$x_{in} \quad \texttt{XOR} \quad w_{i2n},$$

for $j \neq 1$, i.e., for all withdrawals except those from tax-deferred accounts. For example, to favor tax-deferred withdrawals in most reasonable situations, it is desirable to make Roth conversions and tax-exempt distributions exclusive events by introducing binary variables $z_{in}^R \in \{0, 1\}$ with the following constraints:

$$0 \leq z_{in}^R \mathcal{M} - x_{in} \leq \mathcal{M},$$
$$0 \leq z_{in}^R \mathcal{M} + w_{i2n} \leq \mathcal{M}. \tag{3.10}$$

Here, $\mathcal{M}$ is a large number such as 10^7, just slightly larger than what x and w can possibly be.

Another approach could be to perform Roth conversions at mid-year, while withdrawals could be made at the beginning of the year, and surplus deposits, if needed due to RMDs or receiving large sums of money, could be made at the end of the year. Let's formulate this approach in more detail and investigate for potential problems. Timing controls which terms get multiplied by the rate of return $(1 + \mathcal{T}_{ijn})$. Therefore, our current choice would yield

$$b_{ij(n+1)} = [b_{ijn} - w_{ijn} + 0.5\kappa_{ijn}](1 + \mathcal{T}_{ijn}) + [\delta(j,2) - \delta(j,1)]x_{in}(1 + \mathcal{T}_{ijn}/2)$$
$$+ \delta(j,0)d_{in} + 0.5\kappa_{ijn}, \tag{3.11}$$

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

$$b_{ij(n+1)} - (b_{ijn} - w_{ijn})(1 + \mathcal{T}_{ijn})$$
$$- [\delta(j,2) - \delta(j,1)]x_{in}(1 + \mathcal{T}_{ijn}/2) - \delta(j,0)d_{in} = \kappa_{ijn}(1 + \mathcal{T}_{ijn}/2). \tag{3.12}$$

When $j = 0$, this equation introduces a path to shelter negative returns by performing an over-withdrawal from the taxable account at the beginning of the year followed by a deposit in the same

account at the end of the year. This can be removed by using another binary variable, thus making these events exclusive by using the same strategy as Eq. (3.10).

A much simpler approach, while not so natural, is to move all transactions to be synchronous at the beginning or at the end of the year thus avoiding undesirable movements of funds. If we select the beginning of the year, this leads to

$$b_{ij(n+1)} - [b_{ijn} + \delta(j,0)d_{in} - w_{ijn} + (\delta(j,2) - \delta(j,1))x_{in}](1 + \mathcal{T}_{ijn}) = \kappa_{ijn}(1 + \mathcal{T}_{ijn}/2). \quad (3.13)$$

This is the current approach used in Owl.

Initial balances   The initial balances β_{ij} are one of the main inputs of the model. The initial savings account balances are imposed through additional constraints as

$$b_{ij0} = \beta_{ij}. \tag{3.14}$$

At this point, we assume that all accounts are balanced according to the desired allocation ratios α_{ijkn}.

We also introduce another constraint that might look unnecessary, but helps convergence, and prevents overdrafts during the year of passing of the first spouse. As withdrawals and conversions are at the beginning of the year we impose that

$$w_{ijn} + \delta(j,1)x_{in} \leq b_{ijn}, \tag{3.15}$$

which just states that account balances need to be larger than withdrawals and possible Roth conversions.

Roth Conversions   Roth conversions cannot be larger than the balance at the beginning of the year in the account:

$$x_{ijn} \leq b_{i1n}. \tag{3.16}$$

This constraint, however, is naturally satisfied when $b_{ijn} \geq 0$ non-negativity bounds are enforced. Additional maximum Roth conversion constraints x_{max} can be imposed by the user and then the previous equation becomes

$$x_{in} \leq \min(b_{i1n}, x_{max}). \tag{3.17}$$

This last equation will typically need to be coded as two separate constraints.

Net spending   For calculating the net spending g_n, we consider the cash flow of all withdrawals, wages, social security and pension benefits, and big-ticket items. Then we subtract potential surplus s_n and all taxes and Medicare premiums paid:

$$g_n = \sum_i [\omega_{in} + \bar{\zeta}_{in} + \pi_{in}] + \sum_{i,j} w_{ijn} + \sum_i \Lambda_{in}^\pm - s_n - P_n - T_n - U_n - m_n. \tag{3.18}$$

When both spouses are alive, surplus s_n gets deposited in the taxable accounts according to variable η as described in Eq. (2.14),

$$d_{in} = [\delta(i,0)(1-\eta) + \delta(i,1)\eta]s_n. \tag{3.19}$$

14

Otherwise, for $n \geq n_d$, variable η gets redefined as $\eta = \delta(1, i_s)$.

Notice how big-ticket items $\Lambda^{\pm}$ contribute directly to the cash flow. Replacing intermediate variables and bringing all variables to the left-hand side, we get

$$
g_n - \sum_{i,j} w_{ijn} + 0.1 \sum_{i,j \neq 0} (1 - \mathcal{H}(n - n_{i,59})) w_{ijn}
$$

$$
+ m_n + s_n + \sum_t \bar{F}_{tn} \theta_{tn}
$$

$$
+ \psi \sum_i \alpha_{i00n} \left[\mu(b_{i0n} - w_{i0n} + d_{in}) + w_{i0n} \max(0, \tau_{0(n-1)}) \right] = \sum_i [\Lambda_{in}^{\pm} - 0.5 \psi \alpha_{i00n} \mu \kappa_{i0n}]
$$

$$
+ \sum_i [\omega_{in} + \bar{\zeta}_{in} + \pi_{in}]. \quad (3.20)
$$

Notice that we do not consider market losses as we use $\max(0, \tau)$, and that rates from only the previous year are used. Tax-loss harvesting is beyond the scope of this model, as is the tracking of stocks purchased over the years.

We want the net spending to be predictable and smooth. For that purpose, we use

$$
g_n / \bar{\xi}_n = g_0 / \bar{\xi}_0, \quad (3.21)
$$

where the net spending is adjusted for inflation and where we use the time series of parameter ξ_n, allowing for additional adjustments to the overall desired spending. Note that $\bar{\xi}_0 = \xi_0$ as $\gamma_0 = 1$. This profile is used to lower the desired net spending amount by a reduction factor χ after the passing of one spouse and/or to allow for more realistic spending profiles, such as the *smile* profile described above. Eq. (3.21) can be rewritten as

$$
g_n \xi_0 - g_0 \bar{\xi}_n = 0, \quad (3.22)
$$

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 (2.15),

$$
\sum_t f_{tn} \bar{\Delta}_{tn} = \sum_i [\omega_{in} + 0.85 \bar{\zeta}_{in} + \pi_{in}]
$$

$$
+ \sum_i [w_{i1n} + x_{in}]
$$

$$
+ \sum_{i,k \neq 0} [(b_{i0n} - w_{i0n} + d_{in} + 0.5 \kappa_{i0n}) \alpha_{i0kn} \tau_{kn}] - e_n, \quad (3.23)
$$

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

$$
e_n + \sum_t \bar{F}_{tn} - \sum_i [w_{i1n} + x_{in}]
$$

$$
- \sum_{i,k \neq 0} [(b_{i0n} - w_{i0n} + d_{in}) \alpha_{i0kn} \tau_{kn}] = \sum_i [\omega_{in} + 0.85 \bar{\zeta}_{in} + \pi_{in}]
$$

$$
+ 0.5 \sum_{i,k \neq 0} \alpha_{i0kn} \tau_{kn} \kappa_{i0n}. \quad (3.24)
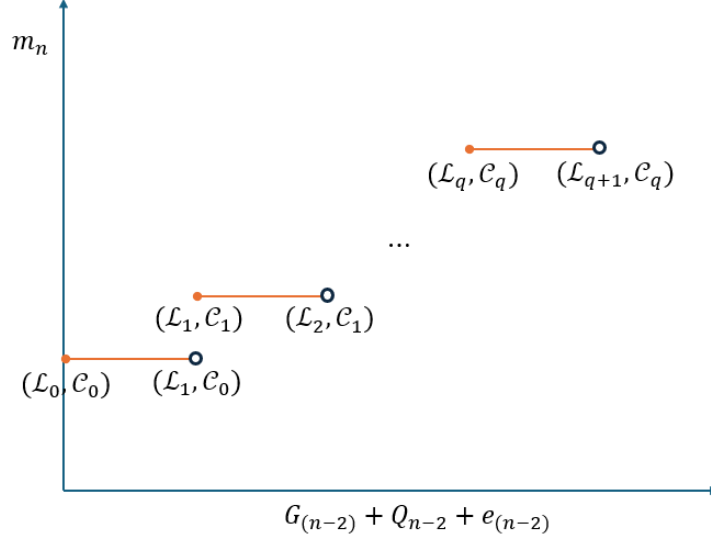$$

15

Figure 3.1: Modeling a piecewise constant function with special ordered set of type 1 (SOS1). The x-axis represents the MAGI from two years ago, while m_n is the Medicare adjusted premiums. A binary variable z_q^m is associated with each segment q.

Finally, the standard exemption is constrained by

$$0 \le e_n \le \bar{\sigma}_n. \tag{3.25}$$

Medicare Costs Annual Medicare costs m_n include the Income-Related Monthly Adjusted Amount commonly known as IRMAA. As this additional adjustment is a piecewise constant function, it can be computed using binary variables and mixed-integer linear programming. In the current tax code, this adjustment depends on the modified adjusted gross income (MAGI) from 2 years earlier. For the MAGI, we simply use $G_{(n-2)} + Q_{(n-2)} + e_{(n-2)}$, i.e., gross taxable ordinary income plus dividends plus standard exemption from 2 years ago. This approach ignores the additional IRS rules around tax-free interests which are insignificant in most cases. If the plan has its oldest individual currently either 64 or 65 years old, MAGI values for previous years must be provided by the user.

Including regular premiums, there are $N_q = 6$ brackets of MAGI indexed for inflation defined as $\bar{\mathcal{L}}_{qn} = \gamma_n \mathcal{L}_q$ and each step increases the annual Medicare cost to $\bar{\mathcal{C}}_{qn} = \gamma_n \mathcal{C}_q$, also adjusted for inflation. Notice that $\mathcal{L}_q$ has $N_q + 1$ elements, with the last element $\mathcal{L}_{N_q} = \mathcal{M}$, where $\mathcal{M}$ is an arbitrary large number bounding the last bracket. Points $(\bar{\mathcal{L}}_{qn}, \bar{\mathcal{C}}_{qn})$ form a piecewise constant function that can be modeled using a special ordered set of type 1 (SOS1). These points are shown in Fig. 3.1. When $q = 0$, the value $\bar{\mathcal{C}}_{qn}$ represents the inflation-adjusted base premium for Medicare part B in year n, while higher q values include the IRMAA additional costs.

We use binary variables $z_{qn}^m \in \{0,1\}$ with the constraints

$$\sum_q z_{qn}^m = 1, \tag{3.26}$$

16

valid for all $n \in \{n_m, \ldots, N_n\}$, where n_m is the index year when Medicare starts. For each of these years, we can then express Medicare costs m_n as

$$m_n = \sum_q z_{qn}^m \bar{\mathcal{C}}_{qn}, \tag{3.27}$$

as only one z_{qn}^m can be non-zero for a given n. The determination of which z_{qn}^m is active is achieved by enforcing the following $(N_n - n_m + 1)$ constraints

$$\sum_{q \neq 0} z_{qn}^m \bar{\mathcal{L}}_{qn} \leq G_{(n-2)} + Q_{(n-2)} + e_{(n-2)} < \sum_{q \neq N_q - 1} z_{qn}^m \bar{\mathcal{L}}_{(q+1)n}, \tag{3.28}$$

for all years $n \geq n_m$ when Medicare is required. The end points $\mathcal{L}_0 = 0$ and $\mathcal{L}_{N_q} = \mathcal{M}$ are not included as they generate constraints that are true for all values considered and do not bring additional information. Eq. (3.28) merely states that the MAGI from two years before must fall into the interval associated with the right premium. For plans where $n_m < 2$, we will request and use user-provided values for the first years MAGIs. Eq. (3.28) constraints can also be implemented individually with $(N_q - 1)(N_n - n_m + 1)$ constraints as

$$
\begin{aligned}
z_{qn}^m \bar{\mathcal{L}}_{qn} &\leq G_{(n-2)} + Q_{(n-2)} + e_{(n-2)}, & q \neq 0, \\
z_{qn}^m \bar{\mathcal{L}}_{(q+1)n} + (1 - z_{qn}^m)\mathcal{M} &> G_{(n-2)} + Q_{(n-2)} + e_{(n-2)}, & q \neq N_q - 1, \\
&\forall q \in 0, \ldots, N_q, \\
&\forall n \in n_m, \ldots, N_n.
\end{aligned}
\tag{3.29}
$$

As G and Q are intermediate variables, we need to expand the first inequality of Eq. (3.28) using Eqs. (2.15) and (2.16) as follows

$$
\begin{aligned}
\sum_{q \neq 0} z_{qn}^m \bar{\mathcal{L}}_{qn} &\leq \sum_{i,k \neq 0} \left[(b_{i0(n-2)} - w_{i0(n-2)} + d_{i(n-2)} + 0.5\kappa_{i0(n-2)})\alpha_{i0k(n-2)}\tau_{k(n-2)} \right] \\
&\quad + \sum_i \alpha_{i00(n-2)} \left[(b_{i0(n-2)} - w_{i0(n-2)} + d_{i(n-2)} + 0.5\kappa_{i0(n-2)})\mu + w_{i0(n-2)}\max(0, \tau_{0(n-3)}) \right] \\
&\quad + \sum_i [\omega_{i(n-2)} + 0.85\bar{\zeta}_{i(n-2)} + \pi_{i(n-2)} + w_{i1(n-2)} + x_{i(n-2)}].
\end{aligned}
\tag{3.30}
$$

For indices at the beginning of the time sequence, we use $\tau_{0,\max(0,n-x)}$, when $x > n$. Pulling all variables on the left-hand side yields

$$
\begin{aligned}
&\sum_{q \neq 0} z_{qn}^m \bar{\mathcal{L}}_{qn} \\
&- \sum_i \left[(b_{i0(n-2)} - w_{i0(n-2)} + d_{i(n-2)}) \left(\mu\alpha_{i00(n-2)} + \sum_{k \neq 0} \alpha_{i0k(n-2)}\tau_{k(n-2)} \right) \right] \\
&- \sum_i \left[w_{i1(n-2)} + x_{i(n-2)} + \alpha_{i00(n-2)}w_{i0(n-2)}\max(0, \tau_{0\max(0,n-3)}) \right] \\
&\leq \sum_i [\omega_{i(n-2)} + 0.85\bar{\zeta}_{i(n-2)} + \pi_{i(n-2)}] \\
&+ 0.5 \sum_i \left[\kappa_{i0(n-2)} \left(\mu\alpha_{i00(n-2)} + \sum_{k \neq 0} \alpha_{i0k(n-2)}\tau_{k(n-2)} \right) \right],
\end{aligned}
\tag{3.31}
$$

for all $n \in \{n_m, \ldots, N_n\}$, and $n_m \geq 2$, otherwise use numbers provided by user if $n_m < 2$. The second line of Eq. (3.31) represents the dividends and interests from taxable accounts, the third line contains distributions and Roth conversions from tax-deferred accounts and taxable gains resulting from the withdrawal of equity assets sold in taxable accounts. Last two lines are wages, social security, pensions, and gains (dividends and interests) generated from contributions in taxable accounts, which are assumed to be made at half-year.

Similarly, the second inequality in Eq. (3.28) becomes

$$
\sum_{q \neq N_q - 1} z_{qn}^m \bar{\mathcal{L}}_{(q+1)n}
$$

$$
- \sum_i \left[(b_{i0(n-2)} - w_{i0(n-2)} + d_{i(n-2)}) \left(\mu \alpha_{i00(n-2)} + \sum_{k \neq 0} \alpha_{i0k(n-2)} \tau_{k(n-2)} \right) \right]
$$

$$
- \sum_i \left[w_{i1(n-2)} + x_{i(n-2)} + \alpha_{i00(n-2)} w_{i0(n-2)} \max(0, \tau_{0 \max(0,n-3)}) \right]
$$

$$
> \sum_i [\omega_{i(n-2)} + 0.85 \bar{\zeta}_{i(n-2)} + \pi_{i(n-2)}]
$$

$$
+ 0.5 \sum_i \left[\kappa_{i0(n-2)} \left(\mu \alpha_{i00(n-2)} + \sum_{k \neq 0} \alpha_{i0k(n-2)} \tau_{k(n-2)} \right) \right], \tag{3.32}
$$

for all $n \in \{n_m, \ldots, N_n\}$, and $n_m \geq 2$, otherwise use numbers provided by user if $n_m < 2$.

4. Mapping of decision variables

At this point, one can use one of the many algebraic modeling languages such as AMPL, GAMS, MOSEK, AIMMS, and Gurobi, and code the equations above using that language, but most of these applications are proprietary and require a license and additional software installation. 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 can ease the process of solving these problems. For completeness, however, we present here a simple index mapping approach that allows solving this problem using a generic linear programming solver.

Using a simple interface for mapping sparse objects to dense ones, the approach described here has been successfully tested with both the HiGHS open-source solver and the MOSEK proprietary 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.

To bring all variables in a single block vector, we will simply use two generic index mapping functions defined as

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

and

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

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

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

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

where

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

and where n exceptionally runs from 0 to N_n inclusively, and therefore q_b runs from $C_b = 0$ to $C_d - 1$, where

$$C_d = q_C(C_b, N_i, N_j, N_n + 1) = N_i N_j (N_n + 1).$$

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

$$y[q_d(i,n)] = d_{in}, \qquad (4.5)$$

where

$$q_d(i,n) = q_*(C_d, i, n, 0; N_i, N_n, 1) \qquad (4.6)$$

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

$$C_f = q_C(C_d, N_i, N_n, 1) = N_i(N_j(N_n + 1) + N_n).$$

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

$$y[q_f(t,n)] = f_{tn}, \qquad (4.7)$$

where

$$q_f(t,n) = q_*(C_f, t, n, 0; N_t, N_n, 1) \qquad (4.8)$$

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

$$C_g = q_C(C_f, N_t, N_n, 1) = N_i(N_j(N_n + 1) + N_n) + N_t N_n.$$

Net spending (g)   For net spending g_n we will use

$$y[q_g(n)] = g_n, \qquad (4.9)$$

where

$$q_g(n) = q_*(C_g, n, 0, 0; N_n, 1, 1) = C_g + n, \qquad (4.10)$$

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

$$C_m = q_C(C_g, N_n, 1, 1) = N_i(N_j(N_n + 1) + N_n) + (N_t + 1)N_n.$$

Medicare costs (m)   For Medicare costs m_n we will use

$$y[q_m(n)] = m_n, \qquad (4.11)$$

where

$$q_m(n) = q_*(C_m, n, 0, 0; N_n, 1, 1) = C_m + n, \qquad (4.12)$$

with q_m running from C_m to $C_s - 1$, where

$$C_s = q_C(C_m, N_n, 1, 1) = N_i(N_j(N_n + 1) + N_n) + (N_t + 2)N_n.$$

Surplus (s)   Surplus can be generated if big-ticket items are received (inheritance, sale of a house, etc.) or due to RMDs. Surplus s is then deposited to taxable savings accounts according to variable η. We will use

$$y[q_s(n)] = s_n, \qquad (4.13)$$

where

$$q_s(n) = q_*(C_s, n, 0, 0; N_n, 1, 1) = C_s + n, \qquad (4.14)$$

with q_s running from C_s to $C_w - 1$, where

$$C_w = q_C(C_s, N_n, 1, 1) = N_i(N_j(N_n + 1) + N_n) + (N_t + 3)N_n.$$

Withdrawals (w)   For withdrawals w_{ijn} we will use

$$y[q_w(i, j, k, n)] = w_{ijn}, \tag{4.15}$$

where

$$q_w(i, j, n) = q_*(C_w, i, j, n; N_i, N_j, N_n) \tag{4.16}$$

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

$$C_x = q_C(C_w, N_i, N_j, N_n) = N_i(N_j(2N_n + 1) + (N_t + 3)N_n).$$

Roth conversions (x)   Finally, for Roth conversions x_{in} we will use

$$y[q_x(i, n)] = x_{in}, \tag{4.17}$$

where

$$q_x(i, n) = q_*(C_x, i, n, 0; N_i, N_n, 1) \tag{4.18}$$

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

$$C_* = q_C(C_x, N_i, N_n, 1) = N_i(N_j(2N_n + 1) + (N_t + N_i + 3)N_n). \tag{4.19}$$

With $N_i = 2$, $N_j = 3$, $N_k = 4$, $N_t = 7$ we have $24N_n + 6$ variables. For a 30-year plan, this results in 726 non-integer decision variables. If the time resolution is increased to months, that would result in 8,646 variables which is still solvable by today's standards. Adding binary variables z_{inz}^R can adds $N_z N_i N_n$ more decision variables, while adding z_{qn}^m adds $6N_n$ more, resulting in $34N_n + 6$, or 1,026 for 30 years.

4.1   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

$$z = q_*(C, i, j, k, n; N_i, N_j, N_k, N_n) := C + iN_jN_kN_n + jN_kN_n + kN_n + n, \tag{4.20}$$

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

$$
\begin{aligned}
n &= \ \texttt{mod}(\texttt{mod}(\texttt{mod}(z - C, N_jN_kN_n), N_kN_n), N_n), \\
k &= \ \texttt{mod}(\texttt{mod}(z - C - n, N_jN_kN_n), N_kN_n)/N_n, \\
j &= \ \texttt{mod}(z - C - n - kN_n, N_jN_kN_n)/(N_kN_n), \\
i &= \ (z - C - n - kN_n - jN_kN_n)/(N_jN_kN_n).
\end{aligned}
\tag{4.21}
$$

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.

5. Building constraint matrices

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

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

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_* (defined in Eq. (4.19)) columns but can have less rows, forming an underdetermined system to be optimized using linear programming. Function J is defined similarly for equality constraints, while I is used for building the rows of the matrix containing the inequality constraints. Additional indices found in columns and not in rows implies the existence of multiple elements on that row.

5.1 Inequality constraints

Required minimum distributions (RMDs)   We rewrite the inequality constraint on required minimum distributions Eq. (3.5) using matrix $A_u y \leq u$ starting with the following $N_i N_n$ rows,

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

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. (3.6) setting an upper limit on fractions $f_{tn} \leq 1$. Instead of using f,

however, we will use $\bar{F}_{tn} = f_{tn}\bar{\Delta}_{tn}$ of the same dimensions and therefore,

$$
\begin{aligned}
A_u[I_1(t,n), q_{\bar{F}}(t,n)] &= 1, \\
u[I_1(t,n)] &= \bar{\Delta}_{tn}, \\
&\quad \forall t \in \{0, \dots, N_t - 1\}, \\
&\quad \forall n \in \{0, \dots, N_n - 1\},
\end{aligned}
\tag{5.3}
$$

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

Medicare brackets Medicare's MAGI brackets determine the values of binary variables which are member of a special ordered set of type 1. We first code Eq. (3.31) and then Eq. (3.32) will be very similar. For establishing the segment lower bound we set

$$
\begin{aligned}
A_u[I_2(n), q_{z^m}(q,n)] &= +\bar{\mathcal{L}}_{qn}, \\
A_u[I_2(n), q_b(i,0,n-2)] &= -\mu\alpha_{i00(n-2)} - \sum_{k \neq 0} \alpha_{i0k(n-2)}\tau_{k(n-2)}, \\
A_u[I_2(n), q_d(i,n-2)] &= -\mu\alpha_{i00(n-2)} - \sum_{k \neq 0} \alpha_{i0k(n-2)}\tau_{k(n-2)}, \\
A_u[I_2(n), q_w(i,0,n-2)] &= +\mu\alpha_{i00(n-2)} + \sum_{k \neq 0} \alpha_{i0k(n-2)}\tau_{k(n-2)} - \alpha_{i00(n-2)}\max(0, \tau_{0\max(0,n-3)}), \\
A_u[I_2(n), q_w(i,1,n-2)] &= -1, \\
A_u[I_2(n), q_x(i,n-2)] &= -1, \\
u[I_2(n)] &= +\sum_i \left[\omega_{i(n-2)} + 0.85\bar{\zeta}_{i(n-2)} + \pi_{i(n-2)} \right] \\
&\quad + 0.5 \sum_i \left[\kappa_{i0(n-2)} \left(\mu\alpha_{i00(n-2)} + \sum_{k \neq 0} \alpha_{i0k(n-2)}\tau_{k(n-2)} \right) \right], \\
&\quad \forall i \in \{0, \dots, N_i - 1\}, \\
&\quad \forall k \in \{0, \dots, N_k - 1\}, \\
&\quad \forall q \in \{1, \dots, N_q - 1\}, \\
&\quad \forall n \in \{\max(n_m, 2), \dots, N_n - 1\}.
\end{aligned}
\tag{5.4}
$$

If $n_m < 2$, we must enforce these additional rows

$$
\begin{aligned}
A_u[I_2(n), q_{z^m}(q,n)] &= +\bar{\mathcal{L}}_{qn}, \\
u[I_2(n)] &= \mathcal{A}_n, \\
&\quad \forall q \in \{1, \dots, N_q - 1\}, \\
&\quad \forall n \in \{n_m, \dots, 2\},
\end{aligned}
\tag{5.5}
$$

where $\mathcal{A}_n$ is a user-provided array of MAGI values for the last two years in chronological order.

Determining which constant linear segment of Fig. 3.1 is active requires comparison with the

23

segment upper bounds which is enforced through similar constraints.

$$\begin{aligned}
A_u[I_3(n), q_{z^m}(q,n)] &= -\bar{\mathcal{L}}_{(q+1)n}, \\
A_u[I_3(n), q_b(i,0,n-2)] &= +\mu\alpha_{i00(n-2)} + \sum_{k \neq 0}\alpha_{i0k(n-2)}\tau_{k(n-2)}, \\
A_u[I_3(n), q_d(i,n-2)] &= +\mu\alpha_{i00(n-2)} + \sum_{k \neq 0}\alpha_{i0k(n-2)}\tau_{k(n-2)}, \\
A_u[I_3(n), q_w(i,0,n-2)] &= -\mu\alpha_{i00(n-2)} - \sum_{k \neq 0}\alpha_{i0k(n-2)}\tau_{k(n-2)} + \alpha_{i00(n-2)}\max(0, \tau_{0\max(0,n-3)}), \\
A_u[I_3(n), q_w(i,1,n-2)] &= +1, \\
A_u[I_3(n), q_x(i,n-2)] &= +1, \\
u[I_3(n)] &= -\sum_i \left[\omega_{i(n-2)} + 0.85\bar{\zeta}_{i(n-2)} + \pi_{i(n-2)}\right] \\
&\quad -0.5\sum_i \left[\kappa_{i0(n-2)}\left(\sum_{k \neq 0}[\alpha_{i0k(n-2)}\tau_{k(n-2)}] + \mu\alpha_{i00(n-2)}\right)\right], \\
&\quad\quad \forall i \in \{0, \ldots, N_i - 1\}, \\
&\quad\quad \forall k \in \{0, \ldots, N_k - 1\}, \\
&\quad\quad \forall q \in \{0, \ldots, N_q - 2\}, \\
&\quad\quad \forall n \in \{\max(n_m, 2), \ldots, N_n - 1\}.
\end{aligned}$$
(5.6)

When $n_m < 2$, we simply enforce these additional rows

$$\begin{aligned}
A_u[I_2(n), q_{z^m}(q,n)] &= -\bar{\mathcal{L}}_{(q+1)n}, \\
u[I_2(n)] &= -\mathcal{A}_n, \\
&\quad \forall q \in \{0, \ldots, N_q - 2\}, \\
&\quad \forall n \in \{n_m, \ldots, 2\},
\end{aligned}$$
(5.7)

where $\mathcal{A}_n$ has been defined above.

5.2 Equality constraints

Account balances For the equality constraints on account balances expressed in Eq. (3.13), we define an equality constraint matrix $A_e y = v$ starting with $N_i N_j N_n$ rows as

$$\begin{aligned}
A_e[J_0(i,j,n), q_b(i,j,n+1)] &= 1, \\
A_e[J_0(i,j,n), q_b(i,j,n)] &= -(1 + \mathcal{T}_{ijn}), \\
A_e[J_0(i,j,n), q_x(i,n)] &= -(\delta(j,2) - \delta(j,1))(1 + \mathcal{T}_{ijn}), \\
A_e[J_0(i,j,n), q_w(i,j,n)] &= (1 + \mathcal{T}_{ijn}), \\
A_e[J_0(i,j,n), q_d(i,n)] &= -\delta(j,0)(1 + \mathcal{T}_{ijn}), \\
&\quad\quad \forall i \in \{0, \ldots, N_i - 1\}, \\
&\quad\quad \forall j \in \{0, \ldots, N_j - 1\}, \\
&\quad\quad \forall n \in \{0, \ldots, N_n - 1\},
\end{aligned}$$
(5.8)

where v is

$$v[J_0(i, j, n)] = \kappa_{ijn}(1 + \mathcal{T}_{ijn}/2). \tag{5.9}$$

The initial account balances expressed in Eq. (3.14) are imposed through

$$
\begin{aligned}
A_e[J_1(i, j), q_b(i, j, 0)] &= 1, \\
v[J_1(i, j)] &= \beta_{ij},
\end{aligned}
\tag{5.10}
$$

$$
\begin{aligned}
&\forall i \in \{0, \ldots, N_i - 1\}, \\
&\forall j \in \{0, \ldots, N_j - 1\},
\end{aligned}
$$

$$\tag{5.11}$$

leading to $N_i N_j$ additional rows to A_e.

Medicare brackets and costs For binary variables z_{qn}^m to satisfy the special ordered set of type 1, we must enforce these N_n constraints

$$
\begin{aligned}
A_u[J_2(n), q_{z^m}(q, n)] &= 1, \\
u[J_2(n)] &= 1 \text{ if } n \geq n_m, 0 \text{ otherwise}, \\
&\forall q \in \{0, \ldots, N_q - 1\}, \\
&\forall n \in \{n_m, \ldots, N_n - 1\},
\end{aligned}
\tag{5.12}
$$

where n_m is the first index year when Medicare is activated. For $n < n_m$, we could also set $z_{qn}^m = 0$, for all q instead of using Eq. (5.12).

The costs for Medicare are easily obtained once the binary variables z_{qn}^m are determined.

$$
\begin{aligned}
A_u[J_3(n), q_m(n)] &= 1, \\
A_u[J_3(n), q_{z^m}(q, n)] &= -\bar{\mathcal{C}}_{qn}, \\
u[J_3(n)] &= 0, \\
&\forall q \in \{0, \ldots, N_q - 1\}, \\
&\forall n \in \{0, \ldots, N_n - 1\}.
\end{aligned}
\tag{5.13}
$$

Net spending For the equality constraint on net spending expressed in Eq. (3.20), we add N_n more rows to $A_e y = v$ as

$$
\begin{aligned}
A_e[J_4(n), q_g(n)] &= 1, \\
A_e[J_4(n), q_b(i, 0, n)] &= \psi \mu \alpha_{i00n}, \\
A_e[J_4(n), q_d(i, n)] &= 1 + \psi \mu \alpha_{i00n}, \\
A_e[J_4(n), q_{\bar{F}}(t, n)] &= \theta_{tn}, \\
A_e[J_4(n), q_m(n)] &= 1, \\
A_e[J_4(n), q_w(i, j, n)] &= -1 + 0.1(1 - \delta(j, 0))(1 - \mathcal{H}(n - n_{i,59})) + \delta(j, 0)\psi \alpha_{i00n}(\max(0, \tau_{0n-1}) - \mu), \\
&\forall t \in \{0, \ldots, N_t - 1\}, \\
&\forall i \in \{0, \ldots, N_i - 1\}, \\
&\forall j \in \{0, \ldots, N_j - 1\}, \\
&\forall n \in \{0, \ldots, N_n - 1\},
\end{aligned}
$$

where v is

$$v[J_4(n)] = \sum_i [\omega_{in} + \bar{\zeta}_{in} + \pi_{in} + \Lambda_{in}^{\pm} - 0.5\psi\mu\alpha_{i00n}\kappa_{i0n}]. \tag{5.14}$$

Here we used the fact that the sum of all deposits d_{in} must equal the surplus s_n,

$$s_n = \sum_i d_{in}.$$

The condition of having a predictable net spending expressed as an equality in Eq. (3.22) adds $N_n - 1$ more rows to $A_e y = v$ as

$$
\begin{aligned}
A_e[J_5(n), q_g(0)] &= -\bar{\xi}_n, \\
A_e[J_5(n), q_g(n)] &= \xi_0, \\
v[J_5(n)] &= 0, \\
&\forall n \in \{1, \ldots, N_n\}.
\end{aligned}
\tag{5.15}
$$

Taxable ordinary income Finally, for the equality constraint in Eq. (3.24) establishing taxable ordinary income, we add N_n rows to $A_e y = v$ as follows

$$
\begin{aligned}
A_e[J_6(n), q_e(n)] &= 1, \\
A_e[J_6(n), q_{\bar{F}}(t, n)] &= 1, \\
A_e[J_6(n), q_w(i, 1, n)] &= -1, \\
A_e[J_6(n), q_x(i, n)] &= -1, \\
A_e[J_6(n), q_b(i, 0, n)] &= -\sum_{k \neq 0} \alpha_{i0kn}\tau_{kn}, \\
A_e[J_6(n), q_w(i, 0, n)] &= \sum_{k \neq 0} \alpha_{i0kn}\tau_{kn}, \\
A_e[J_6(n), q_d(i, n)] &= -\sum_{k \neq 0} \alpha_{i0kn}\tau_{kn}, \\
&\forall t \in \{0, \ldots, N_t - 1\}, \\
&\forall i \in \{0, \ldots, N_i - 1\}, \\
&\forall k \in \{0, \ldots, N_k - 1\}, \\
&\forall n \in \{0, \ldots, N_n - 1\},
\end{aligned}
\tag{5.16}
$$

with

$$v[J_6(n)] = \sum_i [\omega_{in} + 0.85\bar{\zeta}_{in} + \pi_{in}] + 0.5 \sum_i \kappa_{i0n} \sum_{k \neq 0} \alpha_{i0kn}\tau_{kn}. \tag{5.17}$$

5.3 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.

These accounts typically use percentages to designate beneficiaries. 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 $n_d - 1 < N_n - 1$ following the year of passing. To account for that event in year n_d, Eq. (3.13) needs to be rewritten as

$$
\begin{aligned}
b_{ij(n+1)} \;=\; & [1 - \delta(n, n_d - 1)\delta(i, i_d)] \\
& \times \Big\{ [b_{ijn} + \delta(j,0)d_{ij} - w_{ijn} + (\delta(j,2) - \delta(j,1))x_{in}]\,(1 + \mathcal{T}_{ijn}) + \kappa_{ijn}(1 + \mathcal{T}_{ijn}/2) \Big\} \\
& + [\phi_j \delta(n, n_d - 1)\delta(i, i_s)] \\
& \times \Big\{ [b_{i_djn} + \delta(j,0)d_{i_dn} - w_{i_djn} + (\delta(j,2) - \delta(j,1))x_{i_dn}]\,(1 + \mathcal{T}_{i_djn}) \\
& + \kappa_{i_djn}(1 + \mathcal{T}_{i_djn}/2) \Big\}.
\end{aligned}
\tag{5.18}
$$

The first multiplier $[\ldots]$ 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 $[\ldots]$ before the second set of curly braces $\{\}$ 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

$$
\begin{aligned}
& b_{ij(n+1)} \\
& - [1 - \delta(n, n_d - 1)\delta(i, i_d)] \\
& \times \Big\{ [b_{ijn} + \delta(j,0)d_{in} - w_{ijn} + (\delta(j,2) - \delta(j,1))x_{ikn}]\,(1 + \mathcal{T}_{ijn}) \Big\} \\
& - [\phi_j \delta(n, n_d - 1)\delta(i, i_s)] \\
& \times \Big\{ [b_{i_djn} + \delta(j,0)d_{in} - w_{ijn} + (\delta(j,2) - \delta(j,1))x_{i_dkn}]\,(1 + \mathcal{T}_{i_djn}) \Big\} \\
=\; & [1 - \delta(n, n_d - 1)\delta(i, i_d)]\kappa_{ijn}(1 + \mathcal{T}_{ijn}/2) \\
& + [\phi_j \delta(n, n_d - 1)\delta(i, i_s)]\kappa_{i_djn}(1 + \mathcal{T}_{i_djn}/2).
\end{aligned}
\tag{5.19}
$$

27

We are now ready to replace Eq. (5.8) for $A_e y = v$ by

$$
\begin{aligned}
A_e[J_0(i,j,n), q_b(i,j,n+1)] &= 1, \\
A_e[J_0(i,j,n), q_b(i,j,n)] &= -[1 - \delta(n, n_d - 1)\delta(i, i_d)](1 + \mathcal{T}_{ijn}), \\
A_e[J_0(i,j,n), q_w(i,j,n)] &= [1 - \delta(n, n_d - 1)\delta(i, i_d)](1 + \mathcal{T}_{ijn}), \\
A_e[J_0(i,j,n), q_d(i,j,n)] &= -[1 - \delta(n, n_d - 1)\delta(i, i_d)]\delta(j, 0)(1 + \mathcal{T}_{ijn}), \\
A_e[J_0(i,j,n), q_x(i,n)] &= -[1 - \delta(n, n_d - 1)\delta(i, i_d)](\delta(j, 2) - \delta(j, 1))(1 + \mathcal{T}_{ijn}), \\
\text{when } N_i = 2 \text{ and } i = i_s, & \\
A_e[J_0(i,j,n), q_b(i_d, j, n)] &= -[\phi_j \delta(n, n_d - 1)\delta(i, i_s)](1 + \mathcal{T}_{i_d jn}), \\
A_e[J_0(i,j,n), q_w(i_d, j, n)] &= [\phi_j \delta(n, n_d - 1)\delta(i, i_s)](1 + \mathcal{T}_{i_d jn}), \\
A_e[J_0(i,j,n), q_d(i_d, n)] &= -[\phi_j \delta(n, n_d - 1)\delta(i, i_s)]\delta(j, 0)(1 + \mathcal{T}_{i_d jn}), \\
A_e[J_0(i,j,n), q_x(i_d, n)] &= -[\phi_j \delta(n, n_d - 1)\delta(i, i_s)](\delta(j, 2) - \delta(j, 1))(1 + \mathcal{T}_{i_d jn}),
\end{aligned}
$$

$$
\begin{aligned}
&\forall i \in \{0, \ldots, N_i - 1\}, \\
&\forall j \in \{0, \ldots, N_j - 1\}, \\
&\forall n \in \{0, \ldots, N_n - 1\},
\end{aligned}
$$

where v is

$$
\begin{aligned}
v[J_0(i,j,n)] &= [1 - \delta(n, n_d - 1)\delta(i, i_d)]\kappa_{ijn}(1 + \mathcal{T}_{ijn}/2) \\
&\quad + [\phi_j \delta(n, n_d - 1)\delta(i, i_s)]\kappa_{i_d jn}(1 + \mathcal{T}_{i_d jn}/2).
\end{aligned}
\tag{5.20}
$$

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

$$
\kappa_{ijkn} = \alpha_{ijkn}\kappa_{ijn}.
\tag{5.21}
$$

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

Assets allocation have been handled easily by assuming that the accounts are always rebalanced and only using a single multiplier $\mathcal{T}$, defined as

$$
\mathcal{T}_{ijn} = \sum_k \alpha_{ijkn}\tau_{kn},
\tag{5.22}
$$

to compute to return on the total balance of each savings account.

Spousal deposits and withdrawals While keeping the problem linear, a simple constraint can be imposed on surplus deposits in taxable savings accounts. One can specify a spousal ratio η such as

$$
d_{0n} = \eta d_{1n}.
\tag{5.23}
$$

28

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

$$w_{01n} = \eta w_{11n}, \tag{5.24}$$

but this can cause drawing from an empty account while the other spousal account is not. Only the deposit scheme has currently been implemented in Owl.

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

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

which is the value of the estate in nominal dollars at year n, taking into consideration the heir's marginal income tax rate ν on the $(j=1)$ tax-deferred account.

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

$$E_{N_n} = \bar{\epsilon}_{N_n} = \epsilon_{N_n}\gamma_{N_n}. \tag{6.2}$$

Fixing a bequest value amounts to adding the following constraint

$$\sum_{i,j} b_{ijN_n}(1 - \nu\delta(j,1)) = \epsilon_{N_n}\gamma_{N_n}, \tag{6.3}$$

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

$$
\begin{aligned}
A_e[I(0), q_b(i,j,N_n)] &= (1 - \nu\delta(j,1)) \\
v[I(0)] &= \epsilon_{N_n}\gamma_{N_n} \tag{6.4}\\
&\quad \forall i \in \{0,\ldots,N_i - 1\}, \\
&\quad \forall j \in \{0,\ldots,N_j - 1\},
\end{aligned}
$$
$$\tag{6.5}$$

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

$$c[q_g(0)] = -1, \tag{6.6}$$

and 0 otherwise. See Eq. (3.22).

Maximum variable net spending Instead of maximizing a basis for net spending that is multiplied by a profile $\bar{\xi}_n$, one could maximize the sum of net spending over the full duration of the plan, in today's dollars. The quantity to optimize is then

$$c[q_g(n)] \;=\; -1/\gamma_n, \tag{6.7}$$

and 0 otherwise. In that case, constraint equality Eq. (3.22) will need to be changed to an inequality. Instead of obeying

$$g_n \xi_0 - g_0 \bar{\xi}_n = 0, \tag{6.8}$$

we now impose the following inequality constraint

$$(1 - \lambda/100) g_0 \bar{\xi}_n / \xi_0 \; <= \; g_n \; <= \; (1 + \lambda/100) g_0 \bar{\xi}_n / \xi_0, \tag{6.9}$$

where λ is the percentage that the annual net spending is allowed to deviate from the desired profile. It should be noticed that when $\lambda = 0$ the two last equations are equivalent.

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

$$
\begin{aligned}
A_e[I(0), q_g(0)] &\;=\; 1, \\
v[I(0)] &\;=\; g_o,
\end{aligned}
\tag{6.10}
$$

subject to the net spending g_n obeying Eq. (3.22) over time.

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

$$
\begin{aligned}
c[q_b(i, j, N_n)] \;=\; &-(1 - \nu \delta(j, 1)), \tag{6.11} \\
&\forall i \in \{0, \ldots, N_i - 1\}, \\
&\forall j \in \{0, \ldots, N_j - 1\},
\end{aligned}
$$

$$\tag{6.12}$$

and 0 otherwise.