A retirement planner and withdrawal optimization model

(Including both 2017 and 2018 tax implications)

Introduction:

What is the problem we are concerned with here?

For most of us the amount we are able to build up for our retirement is less than what we would like. This creates challenges for us in our retired years. In order to make the most of what we are able to do with these funds we look to our financial advisors. The financial community has understood for a long time that our specific withdrawal patterns can make a significant impact on the amount available to us and how long our funds will last.

One of the first articles I read[[1]](#footnote-1) along these lines discussed 15 separate withdrawal strategies, chief among them is the Common Rule (CR) and its comparison with the article’s winner TDD. The common rule, simply stated, is to withdraw from after tax accounts first, then the tax deferred accounts and finally the tax free accounts. The idea is to allow as much growth as possible in accounts sheltering that growth. The TDD rule on the other hand shuffles the CR’s order to start with some portion of the tax deferred accounts, followed by after tax accounts, next the tax free accounts and finally the rest from the tax deferred accounts. Here the idea is to withdraw the amount equal to the level of deductions from the tax deferred account effectively withdrawing taxable money tax free. Additionally this lowers the amount that will have to be withdrawn starting at age 70 ½ for the Required Minimum Distribution (RMD) required by the IRS, which may prevent some creeping into higher tax brackets. The paper finds TDD outperforms CR by $400K more remaining and paying $225K less taxes when starting with $2M (IRA 70%, After Tax Account 20%, Roth 10%) and a $50K yearly withdrawal.

A primary tool for their work as discussed in their paper was a Linear Programming (LP) model that fully accounted for tax implications. However, they do not go into the details of the model. Thankfully, there is a substantial example[[2]](#footnote-2) from 1994 that can serve as a guide for those of us not completely comfortable with LP models.

The main focus in the above work is on where to withdraw funds for each period of retirement in order to get the most from what you have. Another line of thinking that is even more important is focused on how much to withdraw while ensuring your funds do not run out. The 4% rule[[3]](#footnote-3) seems to have been the first systematic approach to this line of thinking and is well worth the read. The basic idea is to withdraw 4% of your funds the first year of retirement. Each subsequent year, increase the 4% by the previous year’s inflation rate. In many cases (dare I say most) following this advice will leave a substantial amount for the estate. In a few cases (have some examples for this) however it may still fail to fund the retirement in its entirety.

Looking to improve the guidance for where to make withdrawals while also ensuring funds will last throughout retirement has pushed LP models forward to be used for retirement planning and year to year execution as with 3-PEAT.[[4]](#footnote-4) ---Now what to say ---

(Give refs and examples…) and a later ref

Current environment and the rules that govern it:

In order for such models to be useful they must take into account current tax laws and the special nature of retirement accounts, their laws and rules. All of these are regularly modified and updated each year so the models also need to be updated to continue to be accurate. Let’s look at each area the models have to address.

Ordinary Income Tax: Ordinary income includes taxes for earned income, earned interest and up to 85% of social security as well as taxing Tax Deferred Retirement Account (TDRA) withdrawals as normal income. The 2017 tax code defines 7 tax brackets progressing from a 10% marginal rate up to a 39.6% marginal rate. The actual bracket definitions vary depending on filing status: single, married filing jointly, married filing separately… The IRS defines deductions and exemptions allowed from income to reach a taxable income amount. These deductions can be complicated so the IRS also defines a Standard Deduction and personal exemptions which are also dependent on filing status.

Capital Gains Tax: Capital gains tax is the return on capital investments that have been purchased and held for a year or more before selling to recoup the investment and any gain or loss. Such gains from items not held for a year or more are taxed as ordinary income. The 2017 tax code defines three capital gains tax brackets from a 0% marginal rate up to a 20% marginal rate. As with ordinary tax brackets, these vary depending on filing status. High earners also pay a Medicare Net Investment Income tax of 3.8% with threshold amounts that are not indexed to inflation.

Bring 2017 info up to 2018 for this paper

Company Managed Retirement Plans: 401(k), 403(b), 457(b)…, Pensions, ???? (401K… fall into what I will call a Tax Deferred Retirement Account (TDRA))

Individual Retirement Accounts (IRA): IRAs are also TDRAs but not managed by an employer. They have a limit to the yearly contribution that is the minimum of the defined maximum contribution level and your actual ordinary income. The defined maximum contribution is shared between all IRA and Roth IRA accounts. For 2017 it is $5,500 with a $1,000 catchup adder if you are over 50 years old. The yearly contribution to an IRA account is tax deductible (i.e., pretax) in most cases but whether you participate in a company retirement plan and your earnings level may subject you to a graduated scale till no deduction is allowed. No contributions are allowed after age 70.

(Add table to compare 2017 and 2018 data)

IRA withdrawals before age 59 ½ are both taxed and receive a 10% tax penalty for early withdrawal. Withdrawals made after 59 ½ are taxed as ordinary income with no additional penalty. Once the IRA account owner reaches age 70 ½ they are required to withdraw a Required Minimum Distribution (RMD) each year. The RMD is the sum of all the owners IRA account balances divided by their life expectancy as defined by the IRS. If the RMD is not withdrawn the portion remaining will be taxed at 50%.

Roth IRA: A Roth IRA is not a TDRA. Contributions to a Roth IRA are made with after tax money. The advantage they provide is that the profits made in a Roth IRA are not taxed. As mentioned above a maximum yearly contribution is shared between all your IRA and Roth IRA accounts. However, the maximum that can be contributed to a Roth IRA starts to be restricted based on filing status to the point that no contributions can be made for the highest earners.

Roth IRA withdrawals of contributions (as opposed to profits/gains) have no restrictions. Withdrawals of profits/gains, on the other hand, taken before age 59 ½ or before your oldest Roth IRA account has been in place for 5 years will result in the profits being taxed as ordinary income and if before age 59 ½ a 10% penalty tax. After age 59 ½ there will be no penalty and after your oldest Roth account is 5 years old there will be no taxes. Withdrawals from a Roth IRA are defined by the IRS to be in the following order:[[5]](#footnote-5) first contributions then conversions (oldest conversions first) finally profits. Because of this ordering of withdrawal funds it is often the case that withdrawals that are made before 59 ½ and / or before the oldest Roth account has reach 5 years sense opening still do not require any tax or penalty to be paid.

TDRA to Roth Conversions: not yet a part of this model

Estate Taxes: … (need some discussion here)

* Early withdrawal
  + IRA, 401K…
  + Roth (a good source with good links for the 2 five year rules is: https://www.kitces.com/blog/understanding-the-two-5-year-rules-for-roth-ira-contributions-and-conversions/ )

Now, what about the model and its mathematical representation?

A little background may be required at this point. Linear Programming (LP) is a mathematical technique for optimization. The name was chosen prior to the wide spread use and development of software and the programming involved to create it. Rather, when using LP we define an object function to be optimized, a set of real valued variables to be determined, and a set of constraint expressions that set requirements on the optimal solution. This gives us everything we need to set up and solve our matrix expression:

(0)

where x is our vector of variables, each row of A is represents a constraint such that when this row is multiplied by x the corresponding element of b appears on the Right Hand Side (RHS) of the expression. Once the problem has been defined in this manor, expression (0) can be solved for the optimal value of the object function c; ct is the transpose of c. With that out of the way, let’s move on.

OK, so we are looking at two similar object functions. One (R1) inspired by the Ragsdale model (see footnote 2). In the Ragsdale model the overall effect of the model was to optimize for the largest Estate possible. To get it to work on optimizing the way funds are removed from the various accounts it needs a desired spending amount. Without this, the model will not remove any funds from any account that is not required, like the RMD (Require Minimum Distribution). If we are not inclined to maximize our final estate, but would rather maximize the funds we can spend each year we need to emphasize the yearly spending component for the object function. With such an object function we might want to specify a maximum spending level, above which we’d rather keep the funds in the estate. Let’s start with an object function (S1) to maximize possible yearly spending and then we will rewrite it to maximize the estate as does Ragsdale’s object function (R1):

(S1)

(R1)

Sy is gone with piecewise linear tax calculations

The idea here is that we want to maximize the spendable (si) dollars across the retirement years i from 1 to n. For Ragsdale we also want to maximize the remaining account balances. However, we don’t want to require that all the available account balances are use up in cases where we set a maximum spending amount. So, in both the cases we define the object function to include both the sum of the yearly spendable amount and the sum of the account balances at the end of the period times a discount rate for each account (DjA) for all accounts j from 1 to nl. For (S1) though we include a “balancer” to lower the significance of the final balances in the optimization such that spendable funds will be favored. The balancer is using a heuristic of multiplying by 0.001. The account discount rate is applied to the final balances to suggest the value of the final balance at plan end given how the account is taxed; TDRA at 0.85, ROTH at 1.0, and the after tax investment account at 1.0. With regards to the after tax investment account you might expect the discount rate to be lower but tax law steps the basis value up to the value at death removing the capital gains. Finally we want to put pressure on our tax brackets in such a way as to force, as much as possible, the ordinary taxable funds into the lowest brackets first with bracket k from 1 to Bt. We do this by subtracting all the brackets for every year with the higher brackets having higher values (higher btk). We do the same for a shadow bracket set, to be discussed later, using a b,tcg that increases with each shadow bracket.

We model three account types, two of which must be tied to their owner while the third can be shared. These accounts are not homogeneous so this often complicates the notation. Despite this we will, whenever possible, treat them as just a set of accounts from 1 to nl. There is up to one account per person for the TDRA and Roth Retirement Account (RothRA) type accounts and one (shared) after tax account. The account types include:

1. Tax Deferred Retirement Accounts (TDRA) including 401(k), traditional IRA and similar plans (one per person),
2. Roth Retirement Accounts (RothRA) including after tax contributions to 401(k) and similar as well as Roth IRAs (one per person) and
3. After Tax Retirement Savings/Investing Accounts (ATRSI) (shared).

Each account has several associated yearly variables: account balance (bij), account withdrawals (wij) and account deposits (Dij). We will model the wij and Dij as a single variable (or as if) transaction that can be positive or negative. Of course as a single variable it can’t be both positive and negative which eliminates much of the issue with modeling them as completely separate variables.

Note: This model uses one account for each type of account for each person such that the balance and withdrawals and deposits for the model account represent the sum of the balances / withdrawals / deposits for any number of accounts of that type being modeled. One sticky point here is when, for example, a 401(k) contains both tax deferred and after tax contributions. In this case the balance must be split and added to the correct modeled account types.

To summarize, the objective specified by expression (S1) is to maximize the spendable amounts (si) and to a lesser degree remaining balances (bij) while helping to reinforce the proper filling of the tax brackets. Similarly, (R1) does the same while emphasizing the remaining account balances more than the sum of the yearly spendable amounts.

The spendable amount (si) is the sum of all withdrawals (wij) minus any penalty for early withdrawal (pij) plus Social Security (SSi) and other income (oi) minus special expenses (ei), income tax (xik subject to tk) and capital gains tax (yil subject to tcgl) as well as money deposited back into the accounts (Dij). The following constraint is used to assign the spendable amount to si.

(2)

(2#)

Piecewise branch work:

In (2) above replace the sigma tk xik with new var ITi (income tax year i represented here as xi2) and replace sigma tlcg yil with CGTi (capital gains Tax year i represented here as yi2)

In the above expression the tkxik, where tk monotonically increases with each tax bracket, should force the funds into the lowest brackets first. However, in practice it does not always do so which is the reason we added the btkxik to (S1) and (R1) above; this gives it a little more of a nudge. From expression (2) we get n constraint expressions in the model, one for each year.

---

Note to think about: if create new variables such that

oiAs = oiA – oiAD where As (Assets spendable), AD (Assets deposited) A(Assets from sale).

Then I could add oiAs to (2) above and remove oiA from (19) below which would then to limit wij from transferring asset sales income to si and improving the tax issue (if this really works )

(17) would need to use oiAD rather than oiA.

Think about trying this

---

In general we want allowable spending to increase with inflation and to remain steady year to year so we have the following constraint:

(3)

As with expression (2), expression (3) generates n constraint expressions for the model.

(4)

An additional constraint can be given to allow for a minimum level of income per year. To this end we define expression (4) to ensure that the first year’s spendable amount (s1) is at least as much as the desired income (d). Each subsequent year in the modeled retirement period will follow from expression (3). This is only applicable with (R1) as (S1) will maximize the spendable amount and should surpass d whenever possible. When not possible the optimization will fail.

(5)

Similarly, we add a constraint for limiting the yearly spendable amount (s1) to be less than the desired maximum (dm). This constraint is only applicable with (S1) as (R1) will maximize for the ending balance and would find smaller spendable amounts independent of this constraint. Expressions (4) and (5) specify one constraint each for our model.

A number of constraints are required to constrain the objective function in expression (R1) and (S1) to optimal values while ensuring that IRS rules are followed and taxes and penalties are properly accounted for.

(6)

(7)

(8)

(9)

~~(N’)~~

Deposits (Dij) to all IRA and Roth accounts must not exceed either other taxable income (oti) or the sum of the IRS defined maximum contribution(s) (mcir) for each retiree (r) or the users specified contribution (ucij) amount unless a user switch () nullifies this last constraint as in (6); no TDRA account with owner over age 70 ½ can have contributions. In addition, in the ‘married joint’ case we need to ensure that each person’s accounts do not exceed their personal maximums as in (7). On the other hand we want deposits to at least match the user specified contribution level (ucij) (8). Notice that oti in expression (6,7) must be greater than or equal to the sum of the applicable ucij in expression (8) or there will be no model solution. We also need to ensure that no deposits are made to an IRA account after the owner is age 70 or above (9). From these expressions we get n constraints from (6), n times the number of retirees from (7) and an indeterminate number of constraints from (8) and (9). However, this can create a number of redundant constrains, especially when there is no taxable income (oti). For this reason the code to generate these constrains refrains from generating the redundant constrains.

The user switch (-z) allows the optimizer to choose deposit levels above those the user specifies for both TDRA and RothRA accounts, that is those defined in (ucij) as long as they remain within the legal limits.

~~We define a constraint (N’) that is included unless a user switch is used (-z) that limits all deposits to both TDRA and RothRA account to the amount explicitly specified for contribution (uc~~~~ij~~~~). This allows us to compare the difference in the optimized results of allowing the optimizer to choose to make deposits vs. not allowing them.~~

Another constraint required to meet IRS rules for Required Minimum Distribution (RMD) is:

(10)

Of the three account types, only the TDRA has an IRS requirement for a required minimum distribution. This requirement applies to all such accounts but the sum of the RMD can be withdrawn from each account, any one of the accounts or some combination of TDRA accounts as long as the full amount of the RMD is withdrawn from TDRA accounts for each account owner. To ensure withdrawals are at least as much as the IRS minimum required, expression (10), requires withdrawals, starting at age 70 (n70j), exceed the balance in the TDRA (bij) divided by an IRS defined life expectancy value (aij). N70j is the year in which the account owner for account j turns 70 years old. IRS requirement is the year they turn 70 ½ years old. aij is a life expectancy from the IRS tables that includes information of the owners age and the age of the spouse / beneficiary of the j account. If the original owner has died we assume the spouse will own the account and their life expectancy is used.

Piecewise branch work:

Let ITi === xi2, tabable amount as xi1

Replace (11) by replacing the sigma xik with just xi1 and (12) with New piecewise convex constraints to define ITi.

ITi >= btik \*infadj + mrk \* xi1 , I = 1..n, k=1..Bt-1 (baseTax ik \* inflation + marginal rate \* alltaxable<= IncomeTax

Transforming into the desired form is:

ITi – mrk\*xi1 >= btik\*infadj 🡺

Mrk\*xi1 – ITi <= -btik\*infadj, I = 1..n, k=1..Bt-1

(11)

(12)

Equation (11) constrains the variable representing the amount of income in tax bracket k (xik) to take on an amount related to the total taxable income (the TDRA withdrawals (wij) minus TDRA deposits (Dij) which are not taxable and other taxable income, oti) minus the deductions (standard deduction and exemptions, sdi). The source of the funds used for TDRA deposits should be included in other taxable income. Expression (2) and the object functions force taxable income into the lowest possible brackets through the applied tax rate and the need to be maximized. That is, tk is monotonically increasing as k increases in expression (2), which forces the xik in the lowest brackets to fill first. Expression (12) ensures that the xik portion of the taxable income does not exceed the bracket amount (mik). mik and sdi are inflation adjusted. Expression (11) generates n constraints for the model while (12) generates n \*(Bt-1) constraints.

Piecewise branch work:

Below in (13) replace sigma yil with yi1 to place all taxable capital gains in yi1 rather than a set of bin. Yi2 will be used for the actual capital gains:

Let CGTi === yi2, taxable amount is yi1

(13)

(14-2017)

In the same manor (13, 14-2017) fill the capital gains tax brackets (yil) with the non-basis fraction (fi) of the investment account (ATRSI) withdrawals (winAT) and any other taxable capital gains from assets (oiAt). However a quirk of this model is that “withdrawals” are first sourced from Asset Sales (oiA) which is NOT really a withdrawal from the After tax account and does not create any additional tax burden. For this reason we would like to lower these withdrawals for tax purposes by the amount coming from sales of assets but no further than zero. For this a zerofloor() function would be nice but this is not linear so we must do without and allow the over estimation of taxes by the amount f\*w for years with sales of assets.

Also, the capital gains tax bracket fill must start where the ordinary income bracket fill stopped and continue up from there. In order to do this we subtract the amounts in the tax brackets (xik) that overlap with the capital gains bracket from the size of the capital gains bracket. Expressions (13) and (14) generate n + n\*(Bcg-1) or n\*Bcg constraints. Equation (14-2017) works for the 2017 tax code because the tax brackets for ordinary income form perfect subsets for the capital gains brackets. That is, yi1 is a superset of the union of xi1 and xi2, yi2 is a superset of xi3, xi4, xi5 and xi6, and yi3 matches xi7. As the legislature has not done the same thing for the 2018 tax code we need a different solution.

The solution we have chosen is to create a shadow bracket set that is filled with ordinary taxable fund at the same time as xik but with the same bracket constraints as yil. This will give us the amounts that need to be removed from yil to properly fill them independent of the way yil and xik brackets overlap.

-=-=-

Problem: Need to ensure there are NO gaps in tax brackets

Solution: Create variable to capture the gap value(s) and give them a very high penalty in the Objective function.

Goal: have these variable(s) all equal zero in the solution

-=-=-

(15)

(16)

Here (15, 16) are the additions that should follow (11, 12) closely. Notice that the changes deal only with changing xik to syil and related brackets. Syil is our new shadow brackets. With syil we can modify (14-2017) to (14-2018).

Piecewise branch work:

Below in (13) replace sigma yil with yi1 to place all taxable capital gains in yi1 rather than a set of bin. Yi2 will be used for the actual capital gains:

Let CGTi === yi2, taxable amount is yi1

Replace (14-2018) with a new set of piecewise linear convex constraints to define:

CGTi (yi2) as CGTAi (xi3) – CGTIi (xi4)

where CGTAi (Capital Gains Tax ALL year i) and CGTI (Capital Gain Tax on Taxable Income year i)

– not part of CGTi) are:???

CGTAi >= yintercepti + mrk \* yi1 + mrk\* xi1, I = 1..n, k=1..Bt;

* yintercept ik + mrk \* yi1 + mrk \* xi1 <= Taxable Capital gains (yi3)
* mrk \* yi1 + mrk xi1 – yi3 <= yintercepik for all i,k

CGTLi >= yintercept + mrk \* xi1 , I = 1..n, k=1..Bt;

* yintercept ik + mrk \* xi1<= Taxable Capital gains L (yi4)
* mrk \* xi1 – yi4 <= - yintercept ik for all i,k

CGTi(yi2) >= CGTAi(yi3) – GCTLi (yi4)

* yi3 – yi4 – yi2 <= 0 for all i

f(yi1+xi1) -🡪

* CGTAi >= cgtik \*infadj + mrk \* (yi1+xi1) , I = 1..n, k=1..Bt-1 (baseCGTax ik \* inflation + marginal rate \* alltaxable<= CGT(IncomeTax+TaxCG)
* Transforming into the desired form is:
* CGTAi – mrk\*xi1 – mrk\*yi1 >= cgtik\*infadj 🡺
* Mrk\*xi1+mrk\*yi1 – CGTAi <= -cgtik\*infadj, I = 1..n, k=1..Bt-1

f(yi1) -🡪

* CGTIi >= cgtik \*infadj + mrk \* xi1 , I = 1..n, k=1..Bt-1 (baseCGTax ik \* inflation + marginal rate \* taxableIncome<= CGTI(TaxCG)
* Transforming into the desired form is:
* CGTIi – mrk\*xi1 >= cgtik\*infadj 🡺
* mrk\*xi1 – CGTIi <= -cgtik\*infadj, I = 1..n, k=1..Bt-1

CGTi >= CGTAi – CGTIi 🡪

-CGTi <= CGTIi – CGTAi 🡪

CGTAi – CGTIi – CGTi <= 0

(14-2018)

Adding (15, 16) and changing (14-2017) to (14-2018) should remove dependence of the model on matching bracket boundaries. Note that this means (14-2018) should work as well for the 2017 tax code as it does for the 2018 tax code.

(17)

Equation (17) ensures that the balance for each account at the beginning of the year (bi+1,j) is equal to the balance of the account at the start of the previous year (bij), minus the previous year’s withdrawals (wij)) plus the deposits and any income from the sale of assets (oiA) in the previous year (all modeled as occurring at the beginning of the year) times the return on the investment for the year at the rate of return (rij). This is somewhat pessimistic because withdrawals are usually not taken out in one transaction and optimistic as neither are deposits made at the beginning of the year but this is a small effect for our purposes here. Expression (17) generates n\*nl constraints.

(18)

(19)

We also what to make sure that the withdrawals do not exceed their account balance with the one exception that the After tax withdrawal variable (winat) is also used to transfer some amount of money from the sale of assets to the spendable (si) amount. Constraints (18) and (19) ensure these bounds. For both these we get n \* nl constraints.

(20)

Expression (20) sets the beginning account balances to qj.

(21)

Finally, expression (21) constrains the model variables to be greater than or equal zero.

Model Assumptions:

* The standard deduction is assumed along with any allowed personal exemption
* All withdrawals from the investment account are taxed based on gains only and are assumed long term capital gains
  + Fraction gains needs improvement
* The percent of Social Security income that is subject to tax varies by income, here we assume the maximum percentage of 85%
* Not including the Medicare Net Investment Income tax
* All IRA contributions are with pretax money, this ignores income limits at the high end
* Roth withdrawals never incur tax on profits but do incur a penalty before age 59 ½ on the full amount (not quite right)
* No Roth contribution restrictions based on income levels are modeled
* Roth 5 year restriction is always assumed to be met
* RMD table all use the same expectancy table which assumes the spouses are within 10 years age of each other

Possible additions for Second release:

* Remove Roth assumption by adding a basis for roth account and handling its special case taxing correctly

Transforming our model into a python implementation:

OK for our current work we will use the python scipy library, specifically the function scipy.optimize.linprog(). This requires the model to conform to the following template:[[6]](#footnote-6)

Object function: Minimize ct x

With constraints: A x <= b, and x >= 0

Given this we transform our model expressions to match scipy template form as follows:

(S1’)

(R1’)

(2’)

Pij == 0.1 for i <60,j<nl Otherwise pij ==0

(3a’)

(3b’)

(4’)

(5’)

(6’)

(7’)

(8’)

(9’)

(N’)

(10’)

(11’)

(12’)

(13a’)

(13b’)

~~(14-2017’)~~

(14-2018’)

(15’)

(16’)

(17a’)

(17b’)

Bij supports an extra year

(18’)

(19’)

(20a’)

(20b’)

(21’)

To transform our model into the scipy equivalent we have only to use a few operations. To transform the object function, expression (R1’) or (S1’), we minimize the opposite of (R1) or (S1). The rest of the expressions are transformed by multiplying by minus one (-1) to change greater than or equal (≥) to less than or equal (≤), moving constants to the Right Hand Side (RHS) of the expression and doubling up equations to convert from equal (=) into two relations, one with greater than or equal (≤) and the other with less than or equal (≥) (properly transformed) to bring all constraints into standard form (i.e., A x ≤ b). In (17a’ and 17b’) we also multiplied out the c(bij – wij + Dij + oAi) to (cbij – cwij + cDij + coAi) to more closely match the matrix coding. Constraint (20a’ and 20b’) set the initial account balances to qj.

Expression Key:

aij IRA life expectancy at age in year i for account j

Bcg Number of capital gains tax brackets

Bt Number of tax brackets

bij balance of account j in year i

d desired minimum after tax income

dm desire maximal after tax income

Dij deposits to j account in year i

DjA The discount rate for the jth account

ei Special expenses that have a limited duration. These will not be counted as part of si

fi the capital gains fraction of investments (i.e., fraction that does not include the basis)

Flcg floor of the capital gains bracket l

i index for number of retirement years

inf inflation rate

j index for the number of accounts

k index for the tax brackets

l index for capital gains tax brackets

mik size of the kth tax bracket in year i

mcir maximum contribution to TDRA for retiree r

mcgil size of the lth capital gains bracket in year i

n Number of retirement years

n70,j year number that retiree owning the jth account is age 70

nl Number of accounts (i.e., number TDRA + number Roth+ aftertax accounts)

nAT Aftertax account index (when there is an aftertax account, nAT will equal nl)

oi Other income in the ith year. This includes income defined in the configuration income sections as well as returns from asset sales.

oti Other taxable income in the ith year (oti is the ordinary taxable subset of oi)

oAi Other income assets sale in the ith year

oAti Other taxable capital gains from asset sale in the ith year (taxable capital gains related to oAi)

pi the penalty cost of accessing a retirement account prior to age 60 (59½), Age <60 10% else 0%

pvi Present Value in year i

qj balance for account j at the start of retirement

R set of retirees

rij rate of return for account j in year i

rcg capital gains tax rate (temp until cg tax brackets are working)

sdi Standard deduction in year i

si Spendable amount in year i

SSi Social Security income in year i

SSt Social Security faction that is taxable

tlcg marginal capital gains tax rate in bracket l

tk marginal tax rate in tax bracket k

ucij user defined contribution in year i to account j

wij withdrawal from account j in year i

xik ordinary taxable income in year i and bracket k

yil capital gains income in year i and bracket l

EXPERIMENTAL:

Consolidated Todo:

1. Create a case to use Aeq x == beq as well as Aub x <= bub
2. NEED TO ENSURE SS WORKS correct if it has already started (i.e., we are already retired) (Version 1) (done)
3. Add a discussion of the withdrawal schedule for ATRSI accounts specifically that the withdrawn money has to match the fraction for basis vs. non-basis. (this is the IRS assumed withdrawal order rules)
4. Bugs found while porting to go:
   1. maxContribution() inflates to years but should inflate to prePlanYears+years (Version 1)

To Add:

* Need to determine if accounts need to be explicitly moved to new owner after the death of the first spouse. This may be an issue for the TDRA and RothRA accounts!! (Version 1 or 2) (done)
  + In go version I am shifting ownership for RMD needs (done)
* Add joint start [expense] section (Version 1)??? What was I thinking here???
* Add to top of account summary the rate of return for each account (Version 1)
* Add to top of income and expense summary (rate of return or inflation rate “ror/infl” (Version 1)
* Add switch to force off the early withdrawal penalty for each account (Version 2)
  + Add a check box to not use this is an exception applies (add an attribute in the accountable?)
* Like to enable the user to plug in a value for the standard deduction + exemptions. (Version 2)
* Should I assume any interest yields??? Currently it’s just capital gains (Version 2)
* Other taxes (Medicare tax, ???) (No?)
* What about fsic (social security tax) (No?)
* Another problem with this is it DOES NOT ALLOW FOR A CHANGE IN THE BASIS while optimization is happening. Why is this important? Because, I want to be able to have excess withdrawals placed in ATRSI (bi3) which would need a corresponding change to the basis bmi this would require it to be a variable (not a constant) but I don’t think this method allows for variable to be multiplied. NEED TO VERIFY (Version 2)
  + Add a discussion of this to the paper
* Break out the consistency checking code into a separate file (Version 2)
* Make all output headings consistent and useful (uniform across the different ones (Taxable done the same everywhere)) (not necessary for prototype Version 2)
* Add state taxes (Version 2)
* Add checks to eliminate constraints where not needed
* What mix for Investment account? (100% stock, stock bond mix?) (Version 2)
  + Capital gains taxes vs. interest vs. income tax
  + Currently assuming 100% -- But this is not most people or standard advice
* Like to optimize start year of SS (Version 2)
* ORP use a list of time x amount tuples
  + Fplan has an expense vector (implementing this! Has model stability problems)
  + Emergency fund (6months – 1year, family emergencies,…)
* Ability to run in simulation mode against a defined return rate for each year (ie., a portion of the historical S&P 500 record). (Version 2)
  + Compare against other strategies:
    - 4% initial amount forever, or with inflation, or adjust to 4% each year[[7]](#footnote-7) [[8]](#footnote-8) [[9]](#footnote-9)
    - CR
    - TDD
    - Autopilot (<http://www.marketwatch.com/story/theres-a-better-way-to-plan-retirement-withdrawals-2015-01-13?page=2> )
    - RMD all the way

1. Sumutka, Alan R., Andrew M Sumutka, and Lewis W. Coppersmith. (2012). Tax Efficient Retirement Withdrawal Planning Using a Comprehensive Tax Model. Journal of Financial Planning; April, Vol. 25, Issue 4. Retrieved from: <https://www.onefpa.org/journal/Pages/Tax-Efficient%20Retirement%20Withdrawal%20Planning%20Using%20a%20Comprehensive%20Tax%20Model.aspx> [↑](#footnote-ref-1)
2. Ragsdale, Cliff T., Andrew F. Seila, and Philip L. Little. 1994. “An Optimization Model for Scheduling Withdrawals from Tax-Deferred Retirement Accounts.” Financial Services Review 3, 2: 93–108. Retrieved from [www.twenty-first.com/pdf/RagsdaleAn\_Opt\_Model.pdf](http://www.twenty-first.com/pdf/RagsdaleAn_Opt_Model.pdf) [↑](#footnote-ref-2)
3. Bengen, William P (October 1994) Determining withdrawal rates using historical Data. Journal of Financial Planning Pages 171-180, Retrieved from: <http://www.retailinvestor.org/pdf/Bengen1.pdf> [↑](#footnote-ref-3)
4. Welch, James S Jr. A 3-Step Procedure for computing sustainable retirement savings withdrawals. Journal of Financial Planning 30 (8): 45-55. Retrieved from: <https://www.onefpa.org/journal/Pages/AUG17-A-3-Step-Procedure-for-Computing-Sustainable-Retirement-Savings-Withdrawals.aspx> [↑](#footnote-ref-4)
5. IRS 2017 publication 590b. [↑](#footnote-ref-5)
6. The original implementation was done in Python but has been rewritten in go. This includes a rewrite of the simplex solver. [↑](#footnote-ref-6)
7. Bengen, William P (October 1994) Determining withdrawal rates using historical Data. Journal of Financial Planning Pages 171-180, Retrieved from: <http://www.retailinvestor.org/pdf/Bengen1.pdf> [↑](#footnote-ref-7)
8. Bengen, William P (May 1, 2012) How much is enough? Financial Advisor, Retrieved from: <http://www.fa-mag.com/news/how-much-is-enough-10496.html> [↑](#footnote-ref-8)
9. Bengen, William P (June 1, 2016) Is 4.5% Still Safe? Financial Advisor, Retrieved from: <http://www.fa-mag.com/news/is-4-5---still-safe-27153.html> [↑](#footnote-ref-9)