Tax Effects on Investment Portfolios and Financial Markets: Large-Scale Portfolio Optimization Under Stochastic and Integer Constraints

Shijie Liu, Andrew Adams and Boulis M. Ibrahim
Department of Accountancy, Economics and Finance, Heriot Watt University, Edinburgh EH14 4AS, United Kingdom, sl452@hw.ac.uk (Shijie Liu), a.t.adams@hw.ac.uk (Andrew T. Adams), b.m.ibrahim@hw.ac.uk (Boulis M. Ibrahim).

Abstract

This paper investigates the quantitative effects of income and capital gains tax on large investment portfolios. As in Bonami and Lejeune (2009), portfolios are constrained to meet or exceed a prescribed return threshold with a high confidence level and satisfy buy-in threshold and diversification constraints. However, the model of Bonami and Lejeune (2009) is improved upon by incorporating complex tax trading rules with withdrawal features that enhance those considered by Osorio et al. (2004, 2008). The integer variables used result in Mixed Integer Non-linear Programming (MINLP) problems. To implement this on large scale applications, we propose a solution based on Greedy methods with newly introduced dynamic ranking and integer evaluation rules. Performance comparisons with extant MINLP branch and bound and approximation method solvers show that, while the latter fair well for small-scale MINLP problems of less than 72 assets, our proposed method retains good performance with up to 288 assets. A study on individual portfolio composition finds substantial non-linear tax effects on riskier assets and enhanced effects of withdrawal tax only when tax rates are high. The developed framework better enables investors to react to tax changes, and tax policy-makers to quantify the influence of tax changes on private investment preferences.


Area for review: Finance.
1. Introduction

Portfolio optimization has been studied using single and multistage stochastic programming with discrete asset choice constraints. An important issue relating to mean-variance optimization is the uncertainty in problem parameters or the so-called estimation risk or uncertainty in the estimation of expected returns. Bonami and Lejeune (2009) minimize portfolio variance while simultaneously considering uncertainty in expected returns (estimation risk) and trading restrictions modeled with integer constraints. They incorporate uncertainty in expected returns through a probabilistic constraint that follows Roy's (1952) safety first criterion. This identifies as optimal the portfolio for which the probability of its return falling below a prescribed threshold is minimized. The portfolio's expected return is guaranteed to be above a prescribed minimal level with a high probability, typically [0.7,1). They also consider the following three trading restrictions: diversification, which ensures investments in a number of industrial sectors; buy-in threshold, which prevents investors from holding small positions; and round-lot purchasing, which incorporates even-lot block trading behavior of institutional investors. Bonami and Lejeune (2009), however, ignore the substantial effects of taxation, which are important to investors. We extend their work in this important direction by solving portfolio optimization problems that incorporate their three trading restrictions as well as income and capital gains tax under a realistic set of tax rules.

Taxation complicates portfolio optimization problems, but can have significant and important effects on investor wealth (Feldstein, 1976; Constantinides and Scholes, 1980; Constantinides, 1983, 1984; Hubbard, 1985; Dybvig and Koo, 1996; Dammon et al., 2001, 2004; De Miguel and Uppal, 2005; Birge and Yang, 2007). Most of this cited prior work, however, does not consider real-market variations in tax rules within different investment accounts and across different countries and regions. In reality, investment returns arise mainly in the form of income or capital gains and these are subject to different tax rates. Investors may withdraw funds as income either from returns or from initial invested capital. Tax rates also differ across investment accounts, investment assets and global regions. For example, to maximize tax advantage some investment accounts have restrictions on the amount, timing and source of withdrawal from income or initial capital that investors can make. Further, some withdrawal limits increase over the investment horizon, while others are constant, and some taxes are payable immediately upon encashment of a certain type of income, while others can be deferred to the
end of the investment horizon. Moreover, tax rates and policies also vary across countries. These tax and withdrawal rules complicate portfolio optimization problems mainly by introducing constraints that may involve binary or integer variables, which require integer programming, and by causing an indirect mapping between the control variables (asset weights) and the portfolio optimization objective function, which results in an objective function that is neither linear nor perfectly convex with respect to asset weights.

Some recent papers on post-tax portfolio optimization improve long-term investment models by adding real-market features such as tax withdrawals (Osorio et al., 2002, 2004a) and bank taper relief (Osorio et al., 2008b). These papers focus on the effect of taxes on portfolio allocation and deal with return uncertainty through scenario trees. We also extend this post-tax portfolio optimization work further in five ways. First, we apply a probabilistic constraint as in Bonami and Lejenue (2009) to consider return uncertainty and estimation risk simultaneously with integer and other constraints that incorporate taxation and withdrawal rules. Second, we enhance the diversification constraint of Osorio et al. (2004a) by requiring a portfolio to maintain a minimum number of assets, which is a regulatory requirement for some institutional investors. Third, we also enhance the framework of Osorio et al. (2004a,b) by introducing a buy-in threshold constraint that avoids small investments in individual assets that are disallowed, cannot be purchased or are costly to maintain. Fourth, we relax the withdrawal rules of Osorio et al. (2008b) to allow for transactions between accounts, rather than within accounts only. Finally, and in contrast to Osorio et al. (2008b), we allow the tax rate to vary between 0 and 0.7 rather than maintain a constant value. This facilitates consideration of non-linear relations between optimal asset weights and tax rates in a complex and realistic tax and trading environment. It also enables investors and policy-makers to estimate the effects of tax rate changes.

The combination of integer and probabilistic constraints result in mixed-integer nonlinear programming (MINLP) problems. These are challenging to solve, especially for the large-scale problems that we consider. Much work has been done on improving the efficiency of algorithms used to solve the mean-variance Markowitz model under MINLP (see, for example, Bienstock, 1996; Konno and Yamamoto, 2005; Jobst et al., 2001; Corazza and Favaretto, 2007; Gondzio and Grothey, 2007; Bonami and Lejeune, 2009; and Lejeune and Samath-Paç, 2012). However, certain features can limit the applicability of these proposed algorithms to general post-tax portfolio problems. First, many
ignore the uncertainty in problem parameters (estimation risk) (Bienstock, 1996). Second, the objective function considered is specific (Bonami and Lejeune, 2009). Third, the trading rules considered are simplified (Lejeune and Samatlı-Paç, 2012). A new method is therefore required for more complex portfolio optimization problems.

Previous work shows that branch and bound (henceforth B&B) methods, such as BONMIN’s, under most fractional branching rules exhibits higher precision for MINLP than approximating methods such as CPLEX (Bonami and Lejeune, 2009). However, since B&B methods search all possible solutions under a branching tree, it requires many more iterations to reach the optimal solution, and this reduces algorithm efficiency, particularly for large-scale MINLP. We propose a new method based on Greedy heuristics that shortens computing time but retains reasonable precision for our large-scale post-tax portfolio optimization problems. Investors can expect to use their personal PCs rather than super computers to obtain a portfolio solution that, in most cases, is no less accurate than that found by the B&B and the approximating methods considered. The classical Greedy algorithm, however, has its own disadvantages. In particular, its scope is limited to specific problems and its precision is highly dependent on the order of iteration. In response, we present a modification to Greedy and compare its performance with that of BONMIN and CPLEX.

We use our framework to investigate tax effects at the micro levels. Specifically, we analyse the effects of changes in tax rates across products in optimal portfolios of personal investors. Three types of products that are subject to different tax and withdrawal rules are considered, namely, offshore bonds, onshore bonds and unit trusts. Total post-tax return is maximized subject to the constraints mentioned above. We also test whether these effects vary if the investor is rebalancing an existing portfolio or starting up a new portfolio. Effects of withdrawal tax on the optimal portfolio composite are also investigated.

The remainder of this paper is organized as follows. Section 2 presents the optimization problem and relevant settings for the micro level analysis. Section 3 presents the objective functions and constraints of the model. Section 4 presents the modified Greedy heuristic and tests on its performance. Section 5 presents the empirical analyses of tax effects. Section 6 summarizes and concludes.

2. Post-tax Personal Investment Portfolio Optimization – Micro-level

We test tax effects by optimizing personal portfolios over a single period, and, for a given level of risk,
maximize return net of taxes, management fees, and transaction costs. Investors diversify by both allocating their wealth across risky asset classes and locating their wealth across three investment 'accounts' that follow different tax and cash withdrawal rules. These accounts are offshore investment bonds, onshore investment bonds and unit trusts. Offshore bonds is a generic umbrella account for investments that benefit from certain tax concessions such as deferment, while unit trusts are assumed to contain only equity investments. Different tax rates and rules apply to income, capital gains and withdrawals. The general UK tax framework of Osorio (2004a,b, 2008a,b) is adopted, but constraints are enhanced in a number of ways and flexibility is added to allow this setup to be applicable in other countries. This is discussed in Section 3.

Investors generally have two decisions to consider: initializing new portfolios or rebalancing existing portfolios. We represent these by two separate settings (see Table 2). At initialization of new portfolios, investors are assumed to start with cash in hand, and the model's function is to optimize buying decisions in forming new risky asset portfolios within and across the three accounts. There is no demand for interim cash withdrawals and only new cumulative taxes need to be deducted from total return at the end of the period. At rebalancing, investors are assumed to hold an existing portfolio that is bequeathed from the previous period with no cash in hand. Thus, the model’s function is to optimize buy and sell decisions. New cumulative taxes, as well as old taxes accumulated from previous periods, need to be deducted from end-of-period total return.

In line with Osorio et al. (2004a) the tax structures for the three accounts are as follows:

(a) Offshore (investment) bonds

- All taxes are cumulated and paid on total return at the end of the investment horizon.

- Annual withdrawals up to 5% of the original investment are permitted, and associated taxes are deferred until the end of investment (encashment). Unused withdrawal allowances may be carried forward indefinitely.

- Additional withdrawals beyond the annual 5% allowance limit may be made subject to an immediate tax at the encashment rate of \( t_{off} \).

- Withdrawals from the original capital are permitted only when all positive returns have been withdrawn. These withdrawals are not taxed.

(b) Onshore (investment) bonds
Part of the tax on total return is cumulated to the end of the investment and the rest is paid annually at the end of each period.

Tax on withdrawal is the same as offshore bonds, except that the encashment (only) tax rate is \( t_{on} \).

(c) Unit trusts

- Tax on capital gains is paid at the end of investment and this rate changes as its holding time increases. Thus, different assets in a portfolio may be subject to different tax rates depending on their time of purchase.
- Tax on income (e.g., dividends) is paid annually at the end of each period.
- Only return from the previous period is available for withdrawal at the beginning of the decision period.
- Withdrawals from the last period’s income are not taxed at the encashment rate.
- Withdrawals from last year’s capital gains are subject to an immediate tax at the encashment rate of \( CG_T \), where the optimizing period is counted as \( T + 1 \).
- Capital withdrawals follow the same rules as for offshore bonds.

All notation is described in Table 1.

3. Problem Constraints and Objective Functions

3.1. Basic Trading Constraints

**Internal trading budget.** An internal trading budget (balance) constraint ensures that for every account \( k = 1,2,3 \) the total selling proceeds from all three asset classes \( j = 1,2,3 \), \( 1'it_{kj}^s \), are equal to the total buying costs, \( 1'it_{kj}^b \), so that

\[
\sum_j (1'it_{kj}^b - 1'it_{kj}^s) = 0, \quad \forall k = 1,2,3.
\]

**Diversification.** This constraint sets an upper bound on the total value of each asset in a portfolio.

\[
\sum_{k=1}^3 1'w_{kj1} \leq U_k \sum_{j=1}^3 \sum_{k=1}^3 1'w_{kj1} \quad \forall k = 1,2,3; j = 1,2,3
\]

where \( U_k \sum_{j=1}^3 \sum_{k=1}^3 1'w_{kj1} \) is the upper bound and \( \sum_{k=1}^3 1'w_{kj1} \) is the total wealth of each asset in all three accounts. By also setting a lower bound on the total number of assets in a portfolio, \( N_{min} \), firm specific risk can be minimized in the portfolio,

\[
\sum_{j=1}^3 \sum_{k=1}^3 1'\delta_{kj} \geq N_{min}.
\]
where the sum of the binary variables, \( \delta_{kj} \in \{0,1\} \), count this number. In their diversification constraint, Osorio et al. (2004a) do not stipulate a minimum number of assets, which is a common requirement in portfolio management and, for some investing institutions, is imposed by regulation.

**Buy-in threshold.** This constraint requires a minimum purchasing volume for each asset, as small holdings are costly to maintain, disallowed or cannot be purchased. Thus,

\[
w_{kj} \leq \delta_{kj} \sum_{j=1}^{3} \sum_{k=1}^{3} w_{kj1} \quad \forall k = 1,2,3; j = 1,2,3
\]

(4)

\[
w_{\text{min}} \delta_{kj} \leq w_{kj1} \quad \forall k = 1,2,3; j = 1,2,3.
\]

(5)

If investors want to hold one asset in the new portfolio, the corresponding binary variable in (4) must be valued at 1. In (5), \( w_{\text{min}} \delta_{kj} \) defines the buy-in threshold requirement. This constraint does not feature in Osorio et al. (2004a,b, 2008a,b). The large number of integer variables introduced by constraints (3), (4) and (5) (equal to the number of assets, which could be hundreds) contributes to the complexity of the problem and, together with the non-linear stochastic risk constraint discussed below in Section 3.3, necessitates a new algorithm to solve the large-scale MINLP.

### 3.2. Taxation

The total tax liability is built up by calculating the impact of different tax rules on cumulative returns, withdrawals and wealth.

**Cumulative returns.** The remaining returns available for withdrawal in each account are²:

\[
CR_{111} = CR_{110} - 1'wd_{11}^1 - 1'wd_{11}^2/(1 - g_{b:e})
\]

(6a)

\[
CR_{121} = CR_{120} - 1'wd_{12}^1 - 1'wd_{12}^2/(1 - g_{b:b})
\]

(6b)

\[
CR_{131} = CR_{130} - 1'wd_{13}^1 - 1'wd_{13}^2/(1 - g_{b:c})
\]

(6c)

\[
CR_{211} = CR_{210} - 1'wd_{21}^1 - 1'wd_{21}^2/(1 - g_{r:e})
\]

(7a)

\[
CR_{221} = CR_{220} - 1'wd_{22}^1 - 1'wd_{22}^2/(1 - g_{r:b})
\]

(7b)

\[
CR_{231} = CR_{230} - 1'wd_{23}^1 - 1'wd_{23}^2/(1 - g_{r:c})
\]

(7c)

\[
CR_{311} = CR_{310} - 1'wd_{31}^1 - 1'wd_{31}^2/(1 - G_{T:e})
\]

(8a)

\[
CR_{321} = CR_{320} - 1'wd_{32}^1 - 1'wd_{32}^2/(1 - G_{T:b})
\]

(8b)

\[
CR_{331} = CR_{330} - 1'wd_{33}^1 - 1'wd_{33}^2/(1 - G_{T:c})
\]

(8c)

Here, both \( 1'wd_{kj}^1 \) and \( 1'wd_{kj}^2 \) represent total amounts of cash withdrawal. The difference is that the cash in \( wd_{kj}^1 \) is free of tax at encashment, while the cash in \( wd_{kj}^2 \) is subject to an immediate tax
payment. In addition, an upper bound on total withdrawals is set at \( CR_{kj1} \geq 0 \) (\( \forall k = 1,2,3 \)).

**Withdrawals.** There are two types of withdrawal: one from returns and the other from initial capital. For return withdrawals from investment bonds there are immediate tax and withdrawal allowance limits, which increase with the time horizon.

\[
\sum_j 1'wd_{kj}^1 \leq 0.05 \times T \times WL_k - CW_k \quad \forall k = 1,2
\]

Unlike investment bonds, withdrawals from the unit trust account are free from immediate tax and are only available from last year’s income, \( CR_{jni} \).

\[
wd_{3j}^1 \leq CR_{jni} \quad \forall j = 1,2,3
\]

\[
wd_{3j}^2 \leq CR_{jng} \quad \forall j = 1,2,3
\]

Initial capital, however, is available for encashment if, and only if, all available returns have been used up. Binary variables, \( y_k \in \{0,1\} \), are subject to the following restrictions.

\[
\sum_j 1'wd_{kj}^3 - WL \times y_k \leq 0 \quad \forall k = 1,2,3
\]

\[
\sum_j CR_{kj1} + WL \times y_k \leq WL \quad \forall k = 1,2,3
\]

According to (12) and (13), if investors wish to withdraw initial capital (\( wd_{3j}^3 \) is positive), the binary variable \( y_k \) will be 1, and \( CR_{kj1} \) will then be equal to, or less than, 0. Thus, (12) and (13) introduce additional binary variables to the problem.

**Wealth and external trading budget.** Next, we calculate the total wealth in each account after trading, and at the end of the period. In calculating the former, transactions both within and between accounts are counted.

\[
w_{111} = w_{110} - \left[ wd_{11}^3 + wd_{11}^2 / (1 - t_{off,c}) + wd_{11}^2 \right] + (1 - tc)(it_{11}^b + I_{11}) - it_{11}^z
\]

\[
w_{121} = w_{120} - \left[ wd_{12}^3 + wd_{12}^2 / (1 - t_{off,b}) + wd_{12}^2 \right] + (1 - tc)(it_{12}^b + I_{12}) - it_{12}^z
\]

\[
w_{131} = w_{130} - \left[ wd_{13}^3 + wd_{13}^2 / (1 - t_{off,c}) + wd_{13}^2 \right] + (1 - tc)(it_{13}^b + I_{13}) - it_{13}^z
\]

\[
w_{211} = w_{210} - \left[ wd_{21}^3 + wd_{21}^2 / (1 - t_{on,e}) + wd_{21}^2 \right] + (1 - tc)(it_{21}^b + I_{21}) - it_{21}^z
\]

\[
w_{221} = w_{220} - \left[ wd_{22}^3 + wd_{22}^2 / (1 - t_{on,b}) + wd_{22}^2 \right] + (1 - tc)(it_{22}^b + I_{22}) - it_{22}^z
\]

\[
w_{231} = w_{230} - \left[ wd_{23}^3 + wd_{23}^2 / (1 - t_{on,e}) + wd_{23}^2 \right] + (1 - tc)(it_{23}^b + I_{23}) - it_{23}^z
\]
Apart from the balance for internal, within account, trading, there is a balance for external trading across accounts, as shown in (17). With regard to the trading budget constraints (14) to (17), Osorio et al. (2004a) assume that all withdrawals during subsequent periods are held in cash and no transactions are allowed between accounts (Eq. (18) in their paper). In contrast, we include cross-account transactions and cash withdrawal re-investments, which is more realistic. In calculating wealth at the end of the period both expected capital gains and income are considered, and corresponding annual tax payments and management fees are deducted.

\[
w_{112} = (1 - mf_1) \left[(1 + \bar{dv}_e + \bar{cg}_e) \cdot w_{111}\right]
\]

\[
w_{122} = (1 - mf_1) \left[(1 + \bar{dv}_b + \bar{cg}_b) \cdot w_{121}\right]
\]

\[
w_{132} = (1 - mf_1) \left[(1 + \bar{dv}_c + \bar{cg}_c) \cdot w_{131}\right]
\]

\[
w_{212} = (1 - mf_2) \left[(1 + (1 - t_{ane})\bar{dv}_e + \bar{cg}_e) \cdot w_{211}\right]
\]

\[
w_{222} = (1 - mf_2) \left[(1 + (1 - t_{anb})\bar{dv}_b + \bar{cg}_b) \cdot w_{221}\right]
\]

\[
w_{232} = (1 - mf_2) \left[(1 + (1 - t_{ane})\bar{dv}_c + \bar{cg}_c) \cdot w_{231}\right]
\]

\[
w_{312} = (1 - mf_3) \left[(1 + (1 - t_{ine}) \cdot \bar{dv}_e + \bar{cg}_e) \cdot w_{311}\right]
\]

\[
w_{322} = (1 - mf_3) \left[(1 + (1 - t_{inb}) \cdot \bar{dv}_b + \bar{cg}_b) \cdot w_{321}\right]
\]

\[
w_{332} = (1 - mf_3) \left[(1 + (1 - t_{inc}) \cdot \bar{dv}_c + \bar{cg}_c) \cdot w_{331}\right]
\]

Furthermore, since the tax rate on unit trusts decreases with the holding period, a set of rates, \(t_{ine}\), \(t_{inb}\), \(t_{inc}\), are used to account for this feature.

**Cumulative taxes.** Finally, the total tax liability is calculated by adding deferred tax from previous periods to that of the current period.

\[
CT_{112} = CT_{110} + t_{offe}(1 - mf_1) [(\bar{dv}_e + \bar{cg}_e) \cdot w_{111}] - \{t_{offe}/(1 - t_{offe})\} 1' wd_{11}^2
\]

\[
CT_{122} = CT_{120} + t_{offb}(1 - mf_1) [(\bar{dv}_b + \bar{cg}_b) \cdot w_{121}] - \{t_{offb}/(1 - t_{offb})\} 1' wd_{12}^2
\]

\[
CT_{132} = CT_{130} + t_{offc}(1 - mf_1) [(\bar{dv}_c + \bar{cg}_c) \cdot w_{131}] - \{t_{offc}/(1 - t_{offc})\} 1' wd_{13}^2
\]
\[ CT_{212} = CT_{210} + t_{one}(1 - mf_2)\left[ (\overline{dv}_e + \overline{cg}_e)'w_{211} \right] - \left\{ t_{one}/(1 - t_{one}) \right\} 1'wd^2_{31} \] 

\[ CT_{222} = CT_{220} + t_{onb}(1 - mf_2)\left[ (\overline{dv}_b + \overline{cg}_b)'w_{221} \right] - \left\{ t_{onb}/(1 - t_{onb}) \right\} 1'wd^2_{32} \] 

\[ CT_{232} = CT_{230} + t_{one}(1 - mf_2)\left[ (\overline{dv}_c + \overline{cg}_c)'w_{231} \right] - \left\{ t_{one}/(1 - t_{one}) \right\} 1'wd^2_{33} \] 

\[ CT_{312} = CT_{310} + CG_{T+1e}(1 - mf_3)\left[ \overline{cg}_e'w_{311} \right] - \left\{ CG_{T+1e}/(1 - CG_{Te}) \right\} 1'wd^2_{31} \] 

\[ CT_{322} = CT_{320} + CG_{T+1b}(1 - mf_3)\left[ \overline{cg}_b'w_{321} \right] - \left\{ CG_{T+1b}/(1 - CG_{Tb}) \right\} 1'wd^2_{32} \] 

\[ CT_{332} = CT_{330} + CG_{T+1c}(1 - mf_3)\left[ \overline{cg}_c'w_{331} \right] - \left\{ CG_{T+1c}/(1 - CG_{Te}) \right\} 1'wd^2_{33} \] 

For unit trusts, the tax rate changes over time. We therefore assume separate rates for cumulative and immediate taxes, \( CG_{T+1} \) and \( CG_T \), and calculate the final net return of each account by subtracting all deferred tax liabilities from account wealth.

\[ TR_{kj} = 1'w_{kj2} - CT_{kj2} \quad \forall k = 1,2,3; j = 1,2,3 \] 

3.3. Estimation (stochastic) risk constraint

The portfolio optimization literature discusses different approaches to measuring risk and uncertainty (Artzner et al., 1999). Goldfarb and Iyengar (2003), for example, propose a robust factor model to manage risk. Others use historical data of asset returns to represent future risk (Bonami and Lejeune, 2009; Lejeune, 2010) and assume that asset returns follow a normal distribution (Bodnar and Schmid, 2007). In our analysis, the risk measurement falls within the Markowitz mean-variance framework. The classic Markowitz framework relies on the perfect knowledge of the expected returns and the variance-covariance matrix (variance risk) of the assets. This assumes that there is no estimation error. However, expected returns, variances and covariances matrix are unobservable and unknown. Obtaining accurate estimates of them is a challenge. Indeed, many possible sources of errors (e.g., impossibility to obtain a sufficient number of data observations, instability of data and differing personal views of decision-makers on future returns) affect estimation and lead to the so-called estimation risk in portfolio selection (Mulvey and Erkan, 2003; Bawa et al., 1979). Estimation risk has been shown to be the source of erroneous decisions. As pointed out in Ceria and Stubbs (2006) and Cornuejol and Tütüncü (2007) the composition of the optimal portfolio is very sensitive to estimates of the moments of the return distribution, and minor perturbations in these estimates can result in the construction of different portfolios.

Broadie (1993), Chopra and Ziemba (1993) and Ceria and Stubbs (2006) show that estimation
risk is due mainly to errors in estimating the mean of the return distribution. Hence, we focus on the estimation risk of expected returns (as Bonami and Lejeune, 2009) rather than the variance-covariance matrix of returns (as Lejeune and Samatlı-Paç, 2012). This makes the algorithm proposed by Lejeune and Samatlı-Paç (2012), which is based on the reformulation of estimation risk of the variance-covariance matrix unsuitable for our problem. This issue will be discussed further in Section 4. The error in estimating expected returns has attracted renewed interest, and several approaches to incorporate it into portfolio selection have recently been developed. As in Bonami and Lejeune (2009) we adopt Roy’s (1959) safety first risk criterion, which identifies as optimal the portfolio for which the probability of its return falling below a prescribed threshold is minimized, to incorporate the estimation risk of expected returns.

\[ P (\sum_{j=1}^{3} \sum_{k=1}^{3} \xi_{kj} w_{kj1} \geq R_{\min}) \geq p_{\min} \]

The constraint ensures that total expected return, \( \sum_{j=1}^{3} \sum_{k=1}^{3} \xi_{kj} w_{kj1} \), exceeds a prescribed minimal level \( R_{\min} \) with a minimal probability \( p_{\min} \). The account returns (income and capital gains), \( \xi_{k1} = d v_e + c g_e \), \( \xi_{k2} = d v_b + c g_b \), \( \xi_{k3} = d v_c + c g_c \), which are multiplied by the decision variables \( w \), are stochastic and not necessarily independent across accounts. The constraint, however requires transformation prior to incorporation in the model. First, for simplicity, we follow Bodnar and Schmid (2007) and assume that the gross return is normally distributed \( N(\mu, \sigma^2) \), but other, perhaps skewed, distributions are also possible (Bonami and Lejeune, 2009). Second, since \( \sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj} w_{kj1} \) is the average return, \( \mu \), and \( w_1' \sum w_1 \) is equal to the variance \( \sigma^2 \), where \( w_1 \) is a vector \( (w_{111}, w_{211}, w_{311}, w_{121}, w_{221}, w_{321}, w_{131}, w_{231}, w_{331}) \) of all asset weights at period 1, we define

\[ \varphi \equiv (\sum_{j=1}^{3} \sum_{k=1}^{3} \xi_{kj} w_{kj1} - \sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj} w_{kj1}) / \sqrt{w_1' \sum w_1} \]

where \( \mu_{k1} = d v_e + c g_e \), \( \mu_{k2} = d v_b + c g_b \) and \( \mu_{k3} = d v_c + c g_c \), as the normalized portfolio returns. It follows that

\[ P (\sum_{j=1}^{3} \sum_{k=1}^{3} \xi_{kj} w_{kj1} \geq R_{\min}) = P (\varphi \geq R_{\min} - \sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj} w_{kj1} / \sqrt{w_1' \sum w_1}) \]

\[ = 1 - F_{(w)} \left( R_{\min} - \sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj} w_{kj1} / \sqrt{w_1' \sum w_1} \right) \]

where \( F_{(w)} \) is the cumulative probability distribution of the normalized portfolio return and \( F_{(w)}^{-1} \) is its inverse. The probabilistic constraint is thus transformed into the following deterministic equivalent:
\[ 1 - F(w) \left( R_{\min} - \sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj}' w_{kj} / \sqrt{w_{1}' \sum w_{1}} \right) \geq p_{\min} \]

\[ \Leftrightarrow F(w) \left( R_{\min} - \sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj}' w_{kj} / \sqrt{w_{1}' \sum w_{1}} \right) \leq 1 - p_{\min} \]

\[ \Leftrightarrow \sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj}' w_{kj} + F^{-1}_{(w)}(1 - p_{\min}) \sqrt{w_{1}' \sum w_{1}} \geq R_{\min} \]  

(25)

Note that constraint (25) also incorporates the variance risk and, hence, can be used to replace the basic variance risk constraint in our portfolio optimization model.

In our applications we set \( R_{\min} = R_f \), the risk-free rate. Further, as only extreme poor performance needs to be considered, we use a one-tail test. For instance, if 95% confidence is required then \( p_{\min} \) is set at 0.95 and \( F^{-1}_{(w)}(1 - p_{\min}) = -1.65 \). Osorio et al. (2004a) incorporate return uncertainty through a scenario tree, but do not explicitly minimize risk, and this may lead to risk-taking trading strategies. Osorio et al. (2004b, 2008a,b), however, explicitly minimize risk, but in stochastic quadratic programming problems. In our paper, estimation risk rather than the basic variance risk is controlled by the probabilistic constraint to remain under a certain level. Note, that (25) can also be interpreted as a constraint that imposes a pre-specified risk-return tradeoff, where the (negative) term \( F^{-1}_{(w)}(1 - p_{\min}) \) measures this tradeoff in a manner similar to ‘risk tolerance’ of a utility function in mean-variance space. Specifically, it is the marginal rate of substitution between risk and return, and is allowed to vary according to the risk tolerance of the investor.

### 3.4. Single-period MINLP optimization

In contrast to the model of Bonami and Lejeune (2009), which minimizes variance, our objective is to maximize portfolio net (of tax) terminal wealth while holding estimation risk below a given level. This objective function resonates more clearly with investors who want to earn as much risk-adjusted net returns as possible as long as possible while controlling estimation below a certain level. This enables us to investigate better the effects of tax on investment portfolios. It is important to mention that in Bonami and Lejeune (2009) and Lejeune and Samath-Paç (2012) portfolio variance risk is also set as the objective function. This is a convex function of the control variables (asset weights), and perturbations in these variables map directly to the objective function (i.e., variances and covariances do not change with asset weights). This simplifies the model considerably. With taxation, however, this direct mapping breaks down, and the objective function of terminal wealth has to be re-evaluated every time the control variables are perturbed, since tax liabilities differ for different asset weights.
This is expanded upon in Section 4.

As shown in Fig. 10, the estimation risk constraint (with normally-distributed returns) provides an upper boundary on portfolio risk (vertical dotted line). This boundary will move to the left (right) when the required confidence level, $p_{min}$, is higher (lower). The proposed model will return the portfolio at the intersection of the boundary and the efficient frontier. In addition, the change of the objective function increases the sensitivity of portfolios to changes in tax rates. This enables us to better investigate the effects of tax on investment portfolios.

Combining all constraints, the optimization model for individual investors is formulated as the following MINLP problem:

\[
\begin{align*}
\text{Maximize} & \quad \sum_{j=1}^{3} \sum_{k=1}^{3} TR_{kj} \\
\text{Subject to} & \quad \text{constraints (1)–(25)}; \\
& \quad TR_{kj} \geq 0, k, j = 1, 2, 3; \\
& \quad CT_{kj2}, CR_{kj1} \geq 0, k, j = 1, 2, 3; \\
& \quad w_{kj1}, \mathbf{it}_{kj}^{b}, \mathbf{it}_{kj}^{g}, \mathbf{I}_{kj} \geq 0, k, j = 1, 2, 3; \\
& \quad \mathbf{wd}_{kj1}, \mathbf{wd}_{kj2}, \mathbf{wd}_{kj3} \geq 0, k, j = 1, 2, 3; \\
& \quad y_{kj}, \delta_{kj} \in \{0,1\}, k, j = 1, 2, 3;
\end{align*}
\]

4. Solution method

We initially attempt a non-linear B&B (BONMIN version 1.5) solver in OptiToolbox v.1.34, MATLAB version) to solve our MINLP problem in Section 3.4. This, however, is found to take a prohibitive amount of time, with no solution returned in over ten hours. Algorithms for solving MINLP problems are often based on relaxation schemes. For the standard mean-variance portfolio problem, different approaches based on nonlinear B&B algorithms (Bonami et al., 2008; Bonami and Lejeune, 2009) and outer approximations (Lejeune and Samath-Paç, 2012) have been proposed. However, the increased complexity of our optimization problems due to the inclusion of taxes, probabilistic returns and the increased number of assets (up to 288) limits the use of these algorithms. The tax withdrawal rules necessitate the re-evaluation of the entire objective function ($TR_A$) and its constraints (1)–(25) at every iteration in which the control variables, $w_1$, are perturbed, or integrality restrictions on the integer variables restored. In Bonami and Lejeune (2009) portfolio variance is the objective function, and their integer variable scoring process depends on a function of the specific...
contribution of each variable to the overall risk of the portfolio. This is estimated through the Lagrangian function, and for their simple variance-minimising objective function there is a direct link (mapping) between the control variables (asset weights) and the objective function (portfolio variance). This allows them to calculate the effect of a small change to the control variables with two simple equations (22 in their paper). In our model, however, this is not possible since a change in the control variables does not map directly to the objective function. Changes in asset weights lead to complex changes in tax and this in turn changes the total post-tax return used in the objective function.

We introduce a solution method based on a Greedy algorithm with a new dynamic ranking procedure. As optimality is not guaranteed with Greedy heuristics, we perform a comparison and report the optimality gap against BONMIN (B&B) and CPLEX (approximation method). The relaxation schemes and dynamic iterating processes of our solution method are now discussed.

In theory, the B&B algorithm, reviewed and used by Bonami and Lejeune (2009), Lejeune and Samath-Paç (2012) and others, always returns the global optimal solution for all MINLP problems. However, its tree search enumeration process is sometimes too time consuming and has to be terminated before completion, returning a solution without a proof of its optimality, particularly for large-scale applications. This hinders many investors to use solvers that apply B&B methods due to the lack of access to super computers. One of the objectives of the proposed new solution method is to allow investors to find optimal solutions that are, in most cases, guaranteed to be no worse than those obtained by applying the B&B algorithm, and are obtainable within reasonable periods of time using personal computers. Another objective is for this method to be flexible enough to solve large-scale post-tax portfolio optimization problems.

We start by decomposing our MINLP in Section 3.4 into two independent sub-problems, an integer programming problem and a non-linear programming problem, by non-linear relaxation. The non-linear component of the probabilistic constraint (25) is relaxed (removed).

\[
\sum_{j=1}^{3} \sum_{k=1}^{3} \mu_{kj} w_{kj1} \geq R_{min}
\]  

The proposed new solution method based on the Greedy algorithm is used to solve the integer component and the interior point solver Ipopt (Gondzio and Grothey, 2007) is used to solve the non-linear component.

The classical [basic] Greedy algorithm is generally used for resource allocation problems. Given
units of resources and \( m \) addresses for allocation, the algorithm begins by evaluating and ranking all possible pairs of unallocated resources and addresses, choosing the pair with the highest rank for the allocation and then removing the allocated unit and address from subsequent iterations. The algorithm repeats this process until all resource units are removed and an optimal integer solution is returned.

As the algorithm solves integer programming problems dynamically, it reduces the computing time sharply. However, it makes choices dependent only on choices made so far and not on future choices (non-anticipatory). It iteratively makes one greedy choice after another, reducing each given problem into a smaller one. This property makes the quality of solution highly sensitive to the ranking process. In fact, the order of allocation that depends on the ranking process determines whether the algorithm can return a global optimal solution. A proper ranking process is, therefore, required. This is the first major issue we need to solve. Further, to implement the Greedy algorithm for our MINLP problem, we also need to reprogram the iterating process to make it suitable for our portfolio optimization problem, to deal with non-linear relaxation and to solve special cases when enumeration is pruned (stopped) by infeasibility. Solutions to these issues are described below.

### 4.1. Improved Greedy for portfolio optimization

The main Improved Greedy (IG) program is described in pseudo code in Table 3. The description focuses on the iteration and ranking processes (inserted comments are preceded by the percentage sign, '%'). The program starts by relaxing integer constraints only, thus reformulating the problem as a non-linear optimization (NonOpt). It then also relaxes the non-linear constraints and reformulates the problem as a linear optimization (LinOpt). LinOpt is solved first, returning an optimal objective function value (LinObj) of the linear problem (pseudo code line 03 in Table 3). All integer variables are then gathered into an 'Unsolved' set (line 04, Table 3). The program then calculates an 'impact score' (explained in the next paragraph) for each 'Unsolved' integer variable and selects the variable with the highest impact score (lines 05-12, Table 3). This is the variable that would be selected for integer restoration at this iteration. LinOpt is then solved twice, once when this selected variable is assigned an integer value of 0, and once when it is assigned an integer value of 1. The integer value that returns a higher solution is then assigned to the variable as its integer restoration at this iteration (lines 13-17, Table 3). The variable is then deemed 'Solved' and is removed from 'Unsolved'. Fig. 1 is a graphical depiction of the integer restoration process of lines 04-20 of Table 3. The program then
moves forward to the next iteration and picks another 'Unsolved' variable for integer restoration (line 19, Table 3). This process is repeated until all 'Unsolved' integer variables are 'Solved' and their integer restrictions restored (line 20, Table 3). All integer solutions 'Solved' are then fed into the original MINLP problem and a final optimal solution is obtained by the interior point optimizer (Ipopt) in MATLAB v.7 (line 21, Table 3). As the integer part has already been solved by the improved Greedy algorithm, the solution of the remaining problem can be found quickly by the solver, even for a large number of variables.

In ranking 'Unsolved' integer variables for selection (lines 08-10, Table 3) a formula similar to that used by Linderoth and Savelsbergh (1999) is adopted to evaluate how the restoration of the integrality condition impacts (decreases) the objective function of the portfolio optimization problem. This formula is \( \text{Impact}(i) = (\text{LinObj} - \text{Max}(i)) + 2 \times (\text{LinObj} - \text{Min}(i)) \), where \( \text{Max}(i) \) is the higher of two LP solutions of LinOpt obtained when the unsolved integer variable is assigned a value of 0 or 1. \( \text{Min}(i) \) is the second (lower-value) solution.

4.2. Improved Greedy for infeasibility (Improved Greedy)

Running the above algorithm, we find that, in some cases, no \( \text{Impact}(i) \) is returned for some variables and the iterations stop. This occurs when the integer restoration (at 0 or 1) for some integer variables renders the remaining LP problem infeasible. These cases affect the evaluating and ranking process and reduce the algorithm’s precision and efficiency. To deal with these challenges, supplementary code is added for the two possible cases of infeasibility: one-sided and two-sided. One-sided infeasibility occurs when only one of the integers (0 or 1) assigned to an integer variable renders the remaining LP problem infeasible. Two-sided infeasibility is rare and would occur if both integers (0 and 1) that could be assigned to a certain integer variable render the problem infeasible.

We deal with one-sided infeasibility by switching the integer value for the problematic variable. If the LP problem is feasible when the variable is valued at 1 (or 0) but infeasible if it is valued at 0 (or 1), then the variable is integer valued at 1 (or 0) directly before the ranking process. This variable is then recorded as 'Solved' and removed from 'Unsolved'. The algorithm then continues the current iteration by evaluating and ranking all other 'Unsolved' variables. In this manner no more future iteration is required for this variable, and the resulting reduction in the total number of iterations improves the algorithm’s efficiency. Table 4 describes the supplementary pseudo code and Fig. 2
depicts an example of this process.\textsuperscript{5} Fig. 2 shows that while restoring the integrality restriction for a selected variable 2 in the first iteration, the algorithm finds variable 5 to be infeasible at the integer value of 1 but feasible at 0. It therefore chooses to evaluate variable 5 at 0 and moves it to 'Solved'. This is considered as a solution to an 'iteration' that takes precedence and is, thus, recorded as iteration 1. The algorithm then continues to solve the integrality restriction of variable 2 as iteration 2.

A more involved procedure is adopted to deal with two-sided infeasibility. If in the evaluation process the algorithm finds a variable infeasible at both 0 and 1, it reverts to the prior iteration (to the previous 'Solved' variable). It then switches the integer restoration of that iteration's variable to 0 (or 1) if it had been previously 'Solved' or restored to 1 (or 0) and treats the integer 1 (or 0) as 'cancelled' for this variable. If, however, the algorithm attempts to switch the integer to an already 'cancelled' value, it then reverts further back one iteration and repeats the check on the variable of that iteration. This process continues until an integer restoration is found to solve the infeasibility. All integer allocations and references to subsequent iterations will then be cleaned, and the algorithm is allowed to resume its iterations. In the extreme case where all previous iterations have 'cancelled' integers due to infeasibility, an error will be returned by the algorithm and the original MINLP problem is considered infeasible. Table 5 presents the relevant supplementary pseudo code, and Fig. 3 depicts an example of this idea. In Fig. 3, variable 6 at iteration 4 is found to be infeasible at both 0 and 1 (two-sided infeasibility). The algorithm clears this iteration and reverts back to the preceding iteration, 3. It switches the integer solution of variable 1 but finds the variable infeasible with this switch. The algorithm then clears this iteration and reverts back to iteration 2. It finds variable 4 not previously tested at integer 0 (as it had been previously 'Solved' at 1). It changes the integer restoration of this variable to 0 and refers to the previous solution it had of 1 as 'cancelled'. The algorithm then resumes the 'forward' iteration process. This improves the algorithm’s precision by preventing it from returning an incorrect solution that the model is infeasible.

### 4.3. Improved Greedy with precision (Improved Precision)

There are still other cases in which no feasible solution is returned by the program. We think that these are caused by the initial non-linear relaxation imposed when searching for integer solutions. The relaxation used in our algorithm is simple, and the two sub-problems, integer and non-linear relaxations, are separated completely. However, the gap between these two separate steps leads to
some errors in integer valuation, especially when an extremely high confidence level ($p_{\text{min}} \geq 99.5\%$) is set to control risk. As these errors may downgrade the algorithm’s performance, we apply a more advanced relaxation method to combine these two sub-problems and to improve precision.

This is carried out by adding a new test in the code whenever the variable with the highest impact is found. This test checks whether the integer solution obtained from a certain iteration is feasible and optimal with non-linear restoration. In the test, we re-optimize, at the end of each ranking process, the top-ranked variable with non-linear, rather than linearized, constraints to ensure the variable is integer valued properly. In other words, the second evaluation stage within each iteration is carried out by using an NLP rather than the LP of the improved Greedy described above. Thus, ranking is carried out by LP and integer evaluation is carried out by NLP. Table 6 shows the relevant code. The infeasibility test in the previous section is copied while changing the examined problem from "LinOpt" to "NonOpt". This code is used to eliminate the gap between integer and non-linear programming. Although the coding here is not complicated, this new relaxation method improves the algorithm’s quality even though the repeated non-linearity test doubles the computing time.

4.4. Computational Results

We compare the two new methods, the Improved Greedy and the Improved Precision, with BONMIN B&B v.1.5, which is a NLP-based branch-and-bound algorithm from MATLAB/OPTI Toolbox v.1.34, and TOMLAB/CPLEX v.12.1, which is an approximation branch-and-bound approach that utilizes an interior point algorithm to solve second-order cone optimization problems. For NLP problems our two new algorithms use the interior point optimizer Ipopt v.3.10 in MATLAB/OPTI Toolbox v.1.34. All empirical work is performed on an IBM X201 with Intel Dual-Core i5 2.4GHz CPU, 3GB of RAM, and running Windows 7 and MATLAB 7.

The investigation on the effect of taxes in Section 5 below uses actual financial market data. The performance comparison of algorithms conducted in this section, however, is carried out by constructing a test bed that uses simulated data to ensure that extreme cases that may not be present in the actual data are considered. The test bed itself consists of 108 portfolio optimization experiments/portfolios/instances divided into 6 size groups by number of assets: 18 instances/portfolios with 9 assets each, 18 instances with 18 assets each, 18 with 36, 18 with 72, 18 with 144, and 18 with 288. In each instance the prescribed minimum return level, $R_{\text{min}}$, is set equal to
5%, the tax rates for assets are all set equal to 40% or 60%, and the prescribed reliability level, $p_{\text{min}}$, is set between 60% and 99%. For simplicity, asset returns are assumed to follow a normal distribution. All problem instances are modeled using the AMPL modeling language. Table 7 presents problem statistics of the six portfolio groups. The largest problem considered contains 288 assets, 2604 variables, of which 291 are binary, and 4814 total constraints, of which 874 are non-linear inequalities and 589 are non-linear equalities (linear constraints with binary variables are classified as non-linear).

The comparison is conducted on both efficiency and precision, based on computing time and solution’s quality, respectively. The main results are in Table 8. The 'Name' column lists the portfolio instances where one or more algorithm exhibits an optimality gap (digits to the left of the underscore denote the number of assets in a problem instance, and digits to the right denote the number, out of 18, of that portfolio instance). In each problem instance, four optimal solutions are returned by four algorithms. We define an algorithm's solution to a particular problem instance as 'high quality' if it is not worse than any of those returned by the other three algorithms. In these cases there is a zero optimality gap (recorded as “*” in the columns of Table 8 entitled “Optimality Gap”). We define a solution to a particular problem instance as 'low quality' if it exhibits a non-zero optimality gap relative to the high-quality solution of that problem instance. In Table 8 these are reported as a percentage, or denoted by 'NS' or 'INF'. A reported percentage figure measures how much lower the optimal objective function value of a low quality solution is relative to that obtained by a high quality solution. 'NS' denotes no solution returned within 9 hours, and 'INF' denotes an infeasible problem. Fig. 4 shows a summary of the algorithms solution quality reported in Table 8. The vertical axis is the proportion of experiments in which each respective algorithm returns a high-quality solution, and the horizontal axis is the number of assets in every set of experiments. BONMIN and CPLEX return high quality solutions in all experiments with 36 assets or less, while our two new algorithms return a high quality solution for around 81% of experiments of this size. The Improved Greedy exhibits a maximum optimality gap of 8% and an average of only 3.3% and when $p_{\text{min}}$ of the stochastic opportunity constraint is set at a high level of 95% (Table 8). It returns an INF only in cases when $p_{\text{min}}$ is set at the extremely high level of 99%. Such a high confidence level for stochastic risk enlarges the gap between integer and non-linear programming under linear relaxation and misleads the algorithm to a wrong solution. In these two cases the Improved Precision outperforms the Improved Greedy but
still underperforms BONMIN and CPLEX. However, when the number of assets increase beyond 36 a sudden decline is observed in the relative performance of BONMIN and CPLEX. With 72 assets, the Improved Greedy retains its performance at 81%. BONMIN and CPLEX match this performance, but the Improved Precision outperforms all three algorithms returning a high quality solution in 100% of cases. With 144 assets BONMIN's and CPLEX's performance decline to 43% and 61%, respectively, while the Improved Greedy and the Improved Precision perform at 89%. With 288 assets, only the Improved Greedy returns solutions, and it does so at its consistent level of performance of above 81%.

Thus, although the Improved Greedy does not attain the quality of BONMIN and CPLEX for small portfolio problems, it consistently provides high quality solutions even for large portfolio problems in over 81% of cases. Similar to Bonami and Lejeune (2009) and Lejeune and Samatlı-Paç (2012) our results show that neither BONMIN's B&B nor CPLEX are efficient for large-scale portfolio optimizations. The solution quality of the former decreases sharply from 100% to 0% as the number of assets increases from 36 to 288, mainly because time requirements terminate its iterating process before convergence, and no solution is returned. The solution quality of CPLEX also decreases with increasing number of assets. This is because the general approximation method can only return a local optimal solution in a reasonable period of time as the number of integers increase. This holds an optimality gap from the solution found by our new algorithm.

Table 8 summarizes the quality of all solutions against number of assets for the six groups of experiments. Fig. 5 presents the average computing time expended in finding a high quality solution (only). An upper bound on computing time is set at 32400 seconds (9 hours). The figure shows that all four algorithms return a high-quality solution in a reasonably short time when no more than 72 assets are considered in the optimization. When this number doubles to 144, however, differences between the algorithms become apparent. Improved Greedy retains its efficiency while the time for the other three methods, particularly for BONMIN's B&B, increases dramatically. When the number of assets is increased to 288, only the Improved Greedy is able to return a solution and with an average computing time of 12129.88 seconds (3.4 hours), which would normally be acceptable for a tax or a portfolio balancing exercise. None of the other algorithms could return a solution within an extended upper limit of 72000 seconds (20 hours).

In summary, similar to the conclusion reached by Bonami and Lejeune (2009), a dynamic
iterating process can solve large-scale portfolio optimizations under MINLP more efficiently than general B&B and approximation methods. In our paper, we redesign the dynamic process in a Greedy framework and tailor it to more complex post-tax portfolio optimization problems. From the computational results, we conclude that our new solution method, Improved Greedy, is more reliable and efficient than BONMIN’s B&B and CPLEX for our large-scale portfolio problems. Investors can obtain a high-quality solution within a reasonable period of time in most instances using PCs.

5. Influence of taxation on personal portfolio management

In this section, influences of tax policy are examined under the set-up outlined in Sections 2 and 3 above. Market data are obtained from Thomson Reuters Datastream for these investigations. The details are now presented.

5.1. Personal portfolio: Data

Each equity, bond (corporate) and commodity segment of the U.K. market is divided into several subclasses. Equities are categorized by industry sector. All corporate bonds currently active in the market are divided into two groups first: investment grade and high yield. Each group is further divided into industrial subclasses (airline, technology, telecommunications, … etc.). Commodities are categorized by product type (e.g., oil, gold, copper, corn, … etc.). This generates 30 classes of U.K. shares, 7 of corporate bonds and 18 of commodities for each account (onshore investment bond, offshore investment bond and unit trust). Historical annual dividends and capital gains of the asset classes are obtained from Datastream for the period 1990 through 2011. These are then used to calculate average returns and the covariance of related assets within their respective asset classes. All the relevant settings of the parameters in the model are presented in Table 2.

5.2. Personal portfolio: Results and analysis

Initially, all tax rates are kept the same as specified in the initial settings. Each tax rate is then gradually varied in each round in order to test its effect on overall portfolio composition.

Figs. 6–8 show the resulting weights (change in wealth) at different tax rates for equities, corporate bonds and commodities, respectively. Figs. 11-13 show the resulting weights (change in wealth) at different tax rates for offshore bonds, onshore bonds and unit trusts, respectively. The horizontal dotted lines show the cases when optimization ignores taxes. The word 'new' denotes a set of applications with initial (beginning-of-period) wealth being held in cash and a new portfolio is
constructed by the investor through optimization, while the word 'existing' (in Fig. 9) denotes a bequest portfolio that requires rebalancing at the beginning of the investment horizon.

Two main observations can be made from Figs. 6–8. First, the inclusion of taxes in portfolio optimization has a large effect, on average, on portfolio composition (asset weights). This general finding is consistent with previous research. What this analysis adds, however, is that when the effective tax rate is similar across assets (e.g., 40%) little or no difference is observed between the results of post-tax and pre-tax optimizations, but this difference changes dramatically when taxes for different products vary around 40%. Accordingly, optimal investment strategy is sensitive to different tax rates across products. For example, in our setting when the corresponding tax decreases to 20%, investments in corporate bonds and commodities increase far more rapidly than those in equities. When taxes decrease further to 0% it is the investments in equities instead, that increase more rapidly.

Further, the variation across different tax rates in the optimal weights for commodities is higher than that for corporate bonds and equities, and this is a direct result of their relatively higher, on average, return and risk. Taxation, therefore, plays a more critical role in portfolio optimization for riskier investment assets. Second, a decrease in tax rates usually increases the optimal weights of assets, but this relationship is neither linear nor perfectly convex over the entire range of possible tax rates. The implication is that earlier studies that assume a perfectly convex relationship between tax rates and portfolio weights by simplifying tax rules impose simplistic tax consequences and err in their consideration of the true effect of taxes (e.g., Elton and Gruber, 1978).

Secondly, Fig. 9 shows the difference in optimal portfolio composition between new and rebalanced investments. This represents the difference in weights of the three asset classes between the 'new' and 'existing' portfolios at different tax rates. The figure shows that the difference is small. Given that the two main factors that distinguish these two investments are transaction fees and withdrawal tax, it can be concluded that neither of these factors has a large effect on the final investment strategy. However, a difference exists at high tax rates and rises (especially for corporate bonds and commodities) as tax rates increase. Thus, even in a single-period model, there is a potential loss from instant withdrawal tax when the rate of this type of tax is high.

Finally, Figs. 11-13 show that the inclusion of different tax rules through investment accounts (onshore and offshore investment bonds and unit trusts) has a very limited effect on this single period
portfolio optimization. As the tax rates on asset classes change (in analysis it is assumed that both income and capital gains tax rates on an asset class in each account are perturbed in equal increments), the total investment in each account remains almost the same. For example, in Fig. 11 the total investment in the onshore investment bond account remains around 35% no matter how the tax rate on equities changes. The reason could be that in a single period optimization (new or rebalance), the long-term advantages of tax rules in onshore and offshore investment bond accounts are ignored. Thus as the tax rates change for all three asset classes, only the portfolio composition in each account changes while the total investment in each account remains more or less the same. In conclusion, long-term investors who use multi-stage optimization should consider investment accounts with different tax treatments, but short-term investors who use single-stage optimization can safely exclude them to simplify the model.

In summary, there is evidence that taxation plays a significant role in portfolio management, particularly when tax rates differ across products. Further, the relationship between tax and optimal portfolio composition is neither linear nor convex. Thus, mathematical programming methods have an obvious advantage over theoretical methods in this area. In addition, withdrawal tax should not be ignored when related tax rates are high, even in a single-period investment horizon. Finally, investors’ preference for certain assets is significantly influenced by its tax rate. Accordingly, tax policy setters can use our framework to estimate quantitative effects of the change of tax policy to avoid excessive capital outflow of, or ‘overheat’ in the demand for, relevant financial products.

6. Conclusions

We develop a post-tax portfolio optimization model with integer-based trading constraints. In order to examine the real influence of income tax on portfolio management, we incorporate a number of realistic trading constraints. These include the need for diversification, requirements on both the number of assets in a portfolio and the maximum holdings in single assets, round-lot buying, and taxation on cash withdrawals peculiar to the specific personal investments considered (the last two are modeled with integer variables). We also account for the risk in estimating expected asset returns through a stochastic constraint that ensures the expected return of the portfolio exceeds a pre-specified threshold with a high confidence level. We are unaware of other research that considers so broad a range of trading rules for post-tax portfolio optimization in a single model. Hence, this is a more
realistic simulation than prior work, and quantifies better the influence of tax on personal investments.

One key contribution of our paper is that it innovates on the basic Greedy algorithm, making it available for post-tax portfolio optimization problems in which stochastic risk and realistic market restrictions modeled with integer constraints are simultaneously considered. The combination of integer and nonlinear constraints explains the complexity involved in solving such problems under large-scale applications for which very few solvers are efficient. We evaluate the efficacy of our approach on more than 50 problems containing up to 288 assets, and the computational results provide evidence of its efficiency in two aspects: precision of solution and required computing time.

Regarding the role of tax in financial markets, we find that income tax has an obvious impact on portfolio optimization for investors, which supports the conclusion of existing theoretical work. Our mathematical programming shows that with realistic trading constraints, tax rates and portfolio composition have a complex relationship that is neither linear nor convex. Convexity assumptions often made in the literature to guarantee optimality, therefore, are not only unrealistic but also erroneous simplifications. This is the main advantage of our work over prior theoretical research on post-tax portfolio optimization. In addition, in the investigation on effects of withdrawal tax, we find that for single-period optimization, this factor has a limited influence. Investors can simplify the optimization model by ignoring withdrawal tax without changing the optimal solution significantly. Finally, our analysis shows that investors’ preference for a certain asset is significantly influenced by its tax rate. The model is, therefore, useful to tax policy setters.
Table 1. Notation

<table>
<thead>
<tr>
<th>Input data</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 = (1,1,…,1)’</td>
<td>$CR_{kj1}$ cumulative returns for account $k$ after withdrawal of class $j$</td>
</tr>
<tr>
<td>$u^\prime v = u_{1}v_{1} + u_{2}v_{2} + \ldots + u_{n}v_{n}$ (Inner product)</td>
<td>$CT_{kj2}$ final accumulated tax in account $k$ of class $j$</td>
</tr>
<tr>
<td>$u \circ v = (u_{1}v_{1}, u_{2}v_{2}, \ldots, u_{n}v_{n})'$ (Hadamard product)</td>
<td>$TR_{kj}$ net redemption value obtained from account $k$ of class $j$</td>
</tr>
<tr>
<td>$n_{k}$ number of investment assets in account $k$</td>
<td>$w_{kj1}$ amount of money held in each asset after rebalance in account $k$ of class $j$</td>
</tr>
<tr>
<td>$N_{min}$ minimum number of assets with non-zero wealth in portfolio</td>
<td>$w_{kj2}$ final amount of money held in each asset in account $k$ of class $j$</td>
</tr>
<tr>
<td>$WL_k$ amount of wealth at the beginning of investment in account $k$</td>
<td>$it_{kj}^b$ amount of money spent to buy an asset in account $k$ of class $j$</td>
</tr>
<tr>
<td>$m_{f_{k}}$ percentage paid in management fee for account $k$</td>
<td>$it_{kj}^c$ amount of money obtained when selling an asset in account $k$ of class $j$</td>
</tr>
<tr>
<td>$d_{v_{k}}$ sets of historical dividends or income returns of each asset in class $k$</td>
<td>$wd_{kj1}$ first withdrawal from account $k$ of class $j$</td>
</tr>
<tr>
<td>$c_{g_{k}}$ sets of possible capital gains of each asset in class $k$</td>
<td>$wd_{kj2}$ excess withdrawal from account $k$ of class $j$</td>
</tr>
<tr>
<td>$\overline{d_{v_{k}}}$ expected dividends or income returns in class $k$</td>
<td>$wd_{kj3}$ withdrawal taken from the original investment in account $k$ of class $j$</td>
</tr>
<tr>
<td>$\overline{c_{g_{k}}}$ expected capital gains in class $k$</td>
<td>$L_{kj}$ money spent to buy an asset of class $j$ in account $k$ when using withdrawal</td>
</tr>
<tr>
<td>$t_{off}$ cumulative tax on gross returns of offshore bond</td>
<td>$y_k \in {0, 1}$, binary variable for account $k$,</td>
</tr>
<tr>
<td>$t_{on}$ cumulative tax on gross returns of onshore bond</td>
<td>$\delta_k \in {0, 1}$, binary variable for assets in account $k$</td>
</tr>
</tbody>
</table>
Table 2
Investment settings: portfolio management

<table>
<thead>
<tr>
<th>Taxation</th>
<th>Amount (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offshore</td>
<td>$t_{off}$</td>
</tr>
<tr>
<td>Onshore</td>
<td>$t_{on}$</td>
</tr>
<tr>
<td>Onshore</td>
<td>$t_{on}$</td>
</tr>
<tr>
<td>unit trust</td>
<td>$t_{in}$</td>
</tr>
<tr>
<td>unit trust</td>
<td>$CG_T$</td>
</tr>
<tr>
<td>unit trust</td>
<td>$CG_{T+1}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fees</th>
<th>Amount (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annual Fees</td>
<td>$mf_1$ (Offshore)</td>
</tr>
<tr>
<td></td>
<td>$mf_2$ (Onshore)</td>
</tr>
<tr>
<td></td>
<td>$mf_3$ (Unit trust)</td>
</tr>
<tr>
<td>Transaction Costs</td>
<td>$tc$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Initial Wealth</th>
<th>Amount (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Investment</td>
<td>$w_{10}$, $w_{20}$, $w_{30}$ (Portfolio)</td>
</tr>
<tr>
<td></td>
<td>$C_0$ (Cash)</td>
</tr>
<tr>
<td>Rebalance</td>
<td>$w_{10}$, $w_{20}$, $w_{30}$ (Portfolio)</td>
</tr>
<tr>
<td></td>
<td>$C_0$ (Cash)</td>
</tr>
</tbody>
</table>

Table 3
Main Code of Algorithm

01 Relax integer constraints only and record as "NonOpt"
02 Relax both non-linear and integer constraints and record as "LinOpt"
03 LinObj=Opti(LinOpt)  
  % return optimal solution of LinOpt
04 Group all integer variables into set "Unsolved"
05 iteration=1
06 While iteration<=$N$
07 for i=1:$N$
08 if i$\in$Unsolved
09 Impact(i)=Gains(Var(i), Solved, LinOpt)  
  % compute the impact of integer restoration of variable i
10 end
11 end
12 j=find(Max(Impact))  
  % return the variable with the highest impact
13 if Opti(LinOpt, Solved, Var(j)=1)<=$Opti(LinOpt, Solved, Var(j)=0)$
14 Solved(j)=0
15 else
16 Solved(j)=1
17 end
18 Remove j from Unsolved
19 iteration=iteration+1
20 end
21 Obj=Opti(NonOpt, Solved)  
  % return final solution with given integer values
Table 4
Additional Code: One-Sided Infeasibility

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>08.01</td>
<td><code>inf=0</code></td>
</tr>
</tbody>
</table>
| 08.02 | `if Opti(LinOpt, Solved, Var(i)=1)==infeasible`  
% Check whether the selected variable is feasible at 1 for linear problem |
| 08.03 | `solved(i)=0` |
| 08.04 | `inf=inf+1` |
| 08.05 | `end` |
| 08.06 | `if Opti(LinOpt, Solved, Var(i)=0)==infeasible`  
% Check whether the selected variable is feasible at 0 for linear problem |
| 08.07 | `solved(i)=1` |
| 08.08 | `inf=inf+1` |
| 08.09 | `end` |
| 08.10 | `if inf==0`  
% Two-sided feasibility |
| 09.01 | `elseif inf==1`  
% One-sided infeasibility only |
| 09.02 | `iteration=iteration+1` |
| 09.03 | `Remove i from Unsolved` |
| 09.04 | `end` |

Table 5
Additional Code: Two-Sided Infeasibility

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
</table>
| 09.03.01 | `elseif inf==2`  
% Two-sided infeasibility |
| 09.03.02 | `Revert back to the closest iteration which has no record of cancellation` |
| 09.03.03 | `Turn back to the status at the end of that iteration` |
| 09.03.04 | `Assign the opposite integer value to the variable of that iteration` |
| 09.03.05 | `Record this cancellation` |
| 09.03.06 | `Clean all records subsequent to this iteration` |
| 09.03.07 | `Terminate current iteration` |
Table 6
Additional And Replacement Code To Improve Precision

12.01 inf=0
12.02 if Opti(NonOpt, Solved, Var(j)=1)==infeasible
    %Check whether selected variable is feasible for non-linear problem at 1
12.03 solved(j)=0
12.04 inf=inf+1
12.05 end
12.06 if Opti(NonOpt, Solved, Var(j)=0)==infeasible
    %Check whether selected variable is feasible for non-linear problem at 0
12.07 solved(j)=1
12.08 inf=inf+1
12.09 end
12.10 if inf==2
12.11 Follow the same action as in the case of two-sided infeasibility
12.12 elseif inf==0
13. if Opti(NonOpt, Solved, Var(j)=1)<=Opti(NonOpt, Solved, Var(j)=0)
17.01 end

Table 7
Problem Statistics

<table>
<thead>
<tr>
<th>Total Assets</th>
<th>Total Variables</th>
<th>Total Constraints</th>
<th>Binary Variables</th>
<th>Non-linear Inequality</th>
<th>Non-linear Equality</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>93</td>
<td>191</td>
<td>12</td>
<td>37</td>
<td>31</td>
</tr>
<tr>
<td>18</td>
<td>174</td>
<td>341</td>
<td>21</td>
<td>64</td>
<td>49</td>
</tr>
<tr>
<td>36</td>
<td>336</td>
<td>641</td>
<td>39</td>
<td>118</td>
<td>85</td>
</tr>
<tr>
<td>72</td>
<td>660</td>
<td>1241</td>
<td>75</td>
<td>226</td>
<td>157</td>
</tr>
<tr>
<td>144</td>
<td>1308</td>
<td>2441</td>
<td>147</td>
<td>442</td>
<td>301</td>
</tr>
<tr>
<td>288</td>
<td>2604</td>
<td>4814</td>
<td>291</td>
<td>874</td>
<td>589</td>
</tr>
</tbody>
</table>
### Table 8: Computational Results for Instances Showing an Optimality Gap

<table>
<thead>
<tr>
<th>Name</th>
<th>Confidence Level ((p_{\text{min}}, %))</th>
<th>Tax Rate (%)</th>
<th>Optimality Gap (%)</th>
<th>Optimality Gap (%)</th>
<th>Optimality Gap (%)</th>
<th>Name</th>
<th>Confidence Level ((p_{\text{min}}, %))</th>
<th>Tax Rate (%)</th>
<th>Optimality Gap (%)</th>
<th>Optimality Gap (%)</th>
<th>Optimality Gap (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9_15</td>
<td>95</td>
<td>40</td>
<td>*</td>
<td>*</td>
<td>7.1</td>
<td>144_16</td>
<td>95</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>9_16</td>
<td>95</td>
<td>60</td>
<td>*</td>
<td>*</td>
<td>8.0</td>
<td>144_17</td>
<td>99</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>INF</td>
</tr>
<tr>
<td>9_17</td>
<td>99</td>
<td>40</td>
<td>*</td>
<td>INF</td>
<td>9.8</td>
<td>144_18</td>
<td>99</td>
<td>60</td>
<td>NS</td>
<td>INF</td>
<td>NS</td>
</tr>
<tr>
<td>9_18</td>
<td>99</td>
<td>60</td>
<td>*</td>
<td>INF</td>
<td>15.1</td>
<td>288_01</td>
<td>60</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>18_15</td>
<td>95</td>
<td>40</td>
<td>*</td>
<td>*</td>
<td>2.5</td>
<td>288_02</td>
<td>60</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>18_16</td>
<td>95</td>
<td>60</td>
<td>*</td>
<td>*</td>
<td>4.1</td>
<td>288_03</td>
<td>65</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>18_17</td>
<td>99</td>
<td>40</td>
<td>*</td>
<td>INF</td>
<td>6.5</td>
<td>288_04</td>
<td>65</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>18_18</td>
<td>99</td>
<td>60</td>
<td>*</td>
<td>INF</td>
<td>9.1</td>
<td>288_05</td>
<td>70</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>36_15</td>
<td>95</td>
<td>40</td>
<td>*</td>
<td>*</td>
<td>1.4</td>
<td>288_06</td>
<td>70</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>36_16</td>
<td>95</td>
<td>60</td>
<td>*</td>
<td>*</td>
<td>1.8</td>
<td>288_07</td>
<td>75</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>36_17</td>
<td>99</td>
<td>40</td>
<td>*</td>
<td>INF</td>
<td>5.0</td>
<td>288_08</td>
<td>75</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>36_18</td>
<td>99</td>
<td>60</td>
<td>*</td>
<td>INF</td>
<td>5.8</td>
<td>288_09</td>
<td>80</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>72_15</td>
<td>95</td>
<td>40</td>
<td>*</td>
<td>11.3</td>
<td>0.7</td>
<td>288_10</td>
<td>80</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>72_16</td>
<td>95</td>
<td>60</td>
<td>*</td>
<td>8.6</td>
<td>1.1</td>
<td>288_11</td>
<td>85</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>72_17</td>
<td>99</td>
<td>40</td>
<td>NS</td>
<td>15.2</td>
<td>INF</td>
<td>288_12</td>
<td>85</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>72_18</td>
<td>99</td>
<td>60</td>
<td>NS</td>
<td>6.8</td>
<td>INF</td>
<td>288_13</td>
<td>90</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>144_11</td>
<td>85</td>
<td>40</td>
<td>NS</td>
<td>9.9</td>
<td>*</td>
<td>288_14</td>
<td>90</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>144_12</td>
<td>85</td>
<td>60</td>
<td>NS</td>
<td>*</td>
<td>*</td>
<td>288_15</td>
<td>95</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>144_13</td>
<td>90</td>
<td>40</td>
<td>NS</td>
<td>15.3</td>
<td>*</td>
<td>288_16</td>
<td>95</td>
<td>60</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
</tr>
<tr>
<td>144_14</td>
<td>90</td>
<td>60</td>
<td>NS</td>
<td>8.8</td>
<td>*</td>
<td>288_17</td>
<td>99</td>
<td>40</td>
<td>NS</td>
<td>INF</td>
<td>NS</td>
</tr>
<tr>
<td>144_15</td>
<td>95</td>
<td>40</td>
<td>NS</td>
<td>NS</td>
<td>*</td>
<td>288_18</td>
<td>99</td>
<td>60</td>
<td>NS</td>
<td>INF</td>
<td>NS</td>
</tr>
</tbody>
</table>
**Fig. 1** Process of Integer Allocation
(Note to be deleted: All colour figures are supplied for reproduction on the Web, and for printing in greyscale. Quality for printing in greyscale from the supplied colour versions has been checked)

**Fig. 2** Process For One-Sided Infeasibility (inf: infeasible; fea: feasible)

**Fig. 3** Process For Two-Sided Infeasibility
**Fig. 4** Solution Quality: Percentage of Instances in which a High-Quality Solution is Returned Within Nine Hours

<table>
<thead>
<tr>
<th>Number of Assets</th>
<th>BONMIN</th>
<th>Improved Greedy</th>
<th>Improved Precision</th>
<th>CPLEX 12.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>100%</td>
<td>81%</td>
<td>81%</td>
<td>100%</td>
</tr>
<tr>
<td>18</td>
<td>100%</td>
<td>81%</td>
<td>81%</td>
<td>100%</td>
</tr>
<tr>
<td>36</td>
<td>100%</td>
<td>81%</td>
<td>81%</td>
<td>100%</td>
</tr>
<tr>
<td>72</td>
<td>81%</td>
<td>81%</td>
<td>88%</td>
<td>81%</td>
</tr>
<tr>
<td>144</td>
<td>43%</td>
<td>89%</td>
<td>100%</td>
<td>61%</td>
</tr>
<tr>
<td>288</td>
<td>0%</td>
<td>89%</td>
<td>81%</td>
<td>0%</td>
</tr>
</tbody>
</table>

**Fig. 5** Speed and Scalability: Average Computing Time For a High-Quality Solution

<table>
<thead>
<tr>
<th>Number of Assets</th>
<th>BONMIN</th>
<th>Improved Greedy</th>
<th>Improved Precision</th>
<th>CPLEX 12.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>13.72</td>
<td>3.04</td>
<td>11.04</td>
<td>6.78</td>
</tr>
<tr>
<td>18</td>
<td>170.72</td>
<td>8.46</td>
<td>132.04</td>
<td>72.03</td>
</tr>
<tr>
<td>36</td>
<td>710.23</td>
<td>29.50</td>
<td>509.19</td>
<td>360.13</td>
</tr>
<tr>
<td>72</td>
<td>2243.68</td>
<td>105.81</td>
<td>1525.90</td>
<td>1203.11</td>
</tr>
<tr>
<td>144</td>
<td>65210.45</td>
<td>891.35</td>
<td>25430.24</td>
<td>28213.23</td>
</tr>
<tr>
<td>288</td>
<td></td>
<td>12129.88</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 6** Influence of Tax on Equities

<table>
<thead>
<tr>
<th>Tax rate of Equities (new)</th>
<th>Equities</th>
<th>Bonds</th>
<th>Commodities</th>
<th>Equities</th>
<th>Equities(no tax)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>0.1</td>
<td>90%</td>
<td>90%</td>
<td>90%</td>
<td>90%</td>
<td>90%</td>
</tr>
<tr>
<td>0.2</td>
<td>80%</td>
<td>80%</td>
<td>80%</td>
<td>80%</td>
<td>80%</td>
</tr>
<tr>
<td>0.3</td>
<td>70%</td>
<td>70%</td>
<td>70%</td>
<td>70%</td>
<td>70%</td>
</tr>
<tr>
<td>0.4</td>
<td>60%</td>
<td>60%</td>
<td>60%</td>
<td>60%</td>
<td>60%</td>
</tr>
<tr>
<td>0.5</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>0.6</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
<td>40%</td>
</tr>
<tr>
<td>0.7</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
<td>30%</td>
</tr>
</tbody>
</table>
Fig. 7 Influence of Tax on Corporate Bonds

![Influence of Tax on Corporate Bonds](image1)

Fig. 8 Influence of Tax on Commodities

![Influence of Tax on Commodities](image2)

Fig. 9 Difference between New and Existing Portfolio

![Difference between New and Existing Portfolio](image3)

Fig. 10 Optimal Portfolio Obtained by Proposed Model

![Optimal Portfolio Obtained by Proposed Model](image4)
**Fig. 11** Influence of Equity Tax on Investment Account

**Fig. 12** Influence of Corporate Bond Tax on Investment Account

**Fig. 13** Influence of Commodity Tax on Investment Account
Endnotes

1. \(j=1\) is the equity asset class; \(j=2\) is the bonds asset class; \(j=3\) is the commodities asset class.

2. The extra letters, ‘e’, ‘b’ and ‘c’, in the parameter of tax rate (e.g. \(t_{geb}\)) refer to the equity, bond and commodity classes, respectively.

3. As the exact code in MATLAB v.7 is too long, only the pseudo code of the major idea is presented in Tables 3, 4, 5, and 6. Investors can apply it in MATLAB with minor changes.

4. For linear optimization we use lp_solve in OPTI Toolbox v.1.34 on MATLAB.

5. The digits on the left hand side of Table 4 are code line numbers relating to those in Table 3. For example, line 08.01 is line 01 of supplementary code to be inserted after line 08 in Table 3.

6. This assumption can be relaxed to perhaps include positively-skewed distributions without altering the nature of the problem (see Bonami and Lejeune, 2009).

References


Bodnar, T., Schmid, W., 2007. The distribution of the sample variance of the global minimum


