Pooling Trades in a Quantitative Investment Process

A competitive equilibrium in the market for liquidity ensures fairness.

Colm O'Cinneide, Bernd Scherer, and Xiaodong Xu

Modern asset management increasingly relies on large-scale portfolio construction technology in order to provide investment management services to large numbers of clients as efficiently as possible. Using quantitative methods, managers can readily accommodate clients with different investment universes, risk appetites, constraints, benchmarks, and legal requirements, within a unified process.

Efficiency often dictates that the trades of diverse clients be pooled and executed simultaneously. In this case, the transaction costs for a given client may depend on the overall level of trading and not just on that client's trading requirements. How do we formulate a portfolio optimization problem that takes this simultaneous trading into account?

The problematic interaction between client trades arises when the cost of trading per share varies with the number of shares traded. We describe these transaction costs as non- linear because the total cost of a trade is a non-linear function of the size of the trade. Non-linear transaction costs are the rule rather than the exception. Costs per share traded typically increase with the size of the trade because of market impact.

In this case, if the firm optimizes a client's portfolio based on her trades only, failing to take into consideration the higher transaction costs due to trades for other accounts, it imposes more trading on that client than is appropriate. If an active investment process is to be scalable, it must take into account the effects of aggregate trading. Otherwise, the benefits sought through trading may be lost because of underestimated transaction costs.

The question of how to deal with simultaneous trading is not a purely technical one. In fact, it necessitates

Colm O'Cinneide
is a senior research analyst at Deutsche Asset Management in New York.
colm.o-cinneide@db.com

Bernd Scherer
is head of research at Deutsche Asset Management in New York.
bernd.scherer@db.com

Xiaodong Xu
is a research analyst at Deutsche Asset Management in New York.
xiaodong.xu@db.com

a fundamental change to the traditional goal of portfolio selection. Because the trading choices made for one client affect the outcomes for other clients, we must devise a process that serves to mediate between clients with regard to decisions about trading, rather than simply optimizing for each client individually.

Decisions on trading must take into account fairness and the common good of all clients. The themes of fairness and mediation can be addressed by bringing ideas from microeconomics to bear on the portfolio selection problem.

The importance of an asset manager's role as mediator and guarantor of fairness in trading decisions is well understood, but we don't know that anyone has yet proposed a methodology for expressing the goal of fairness in a quantitative portfolio selection context. We show how to formulate an optimization problem that determines the portfolios of all clients in a way that is optimal in an overall sense and also ensures fairness to each client in how trading decisions are made. We call this process *multi-account optimization.*

Multi-account optimization combines the objectives of all clients in a simple way and evaluates transaction costs according to aggregate trading needs. The procedure determines all positions and trades in a single optimization. We argue that multi-account optimization is the correct solution to the problem of aggregate trading because it identifies the unique competitive equilibrium in the market for liquidity. That is to say, multi-account optimization makes the same decisions for the clients as they would make for themselves if they were trading competitively in the market for liquidity. The asset manager's mediation in trading decisions is equivalent to setting prices of liquidity according to a market equilibrium.

Multi-account optimization has several other properties. It maximizes the total welfare of all clients. This means that the asset manager is playing the role of a social planner in allocating liquidity among clients (O'Connell [1982]). The allocation of liquidity under multi-account optimization is Pareto-optimal, so that liquidity cannot be redistributed to the benefit of one client without making things worse for another client. Finally, the portfolio that each client receives is optimal, given the marginal transaction costs—or the prices of liquidity—based on aggregate trading.

We illustrate multi-account optimization by developing a special formulation that incorporates the ERISA rule forbidding cross-trading. We give examples showing how multi-account optimization behaves in various situations, and we demonstrate how performance of a strategy is eroded as more and more clients are taken on when transaction costs are non-linear.

The problem of simultaneous trading may be circumvented by folding the separate accounts into a single fund, but this is often impossible because of clients' diverse requirements. Another approach is to trade different accounts at different times. This may have the advantage of reducing market impact, but it comes at the expense of increased dispersion between accounts, delays that may erode value-added due to alpha decay, and increased demands on operational resources.

FORMULATING THE MULTI-ACCOUNT OPTIMIZATION PROBLEM

We must first specify a formulation of a single client's optimization problem that is general enough to allow risk to be measured in various ways and to appear either as a constraint or as a penalty in the objective function. It must also exhibit the transaction cost model explicitly in the objective function. Moreover, the separate utilities of the clients must be compatible with one another and with the transaction cost model, since ultimately these must all be combined in a meaningful way.

We address these requirements as follows. We assume that each client's (expected) utility function U_k is in the form of an expected dollar return penalized for risk. We are agnostic as to how risk is measured, but require that U_k be concave. The transaction cost model τ gives the cost of trading in dollars; this is assumed to be a general convex increasing function.

Convexity implies increasing marginal cost, a property that arises from the market impact of trading rather than brokerage commissions. The transaction cost model need not be additive across assets and so may have the feature that trading in one asset can affect transaction costs in another.

We can represent the portfolio selection problem of client k, one of a group of K clients whose accounts are to be traded simultaneously, as:

$$\begin{aligned} &\text{Maximize } U_k(x_k) - \tau(t_k) \\ &\text{subject to } x_k \in C_k \end{aligned} \tag{1}$$

where x_k is a vector of optimal holdings, or weights, for client k's account, which we wish to determine; t_k is a vector of trades taking the portfolio from the initial holdings x_k^0 to the optimal holdings x_k; and C_k denotes the client's feasible region, describing all of her constraints. The entries of x_k are denoted by x_{ki} and represent client k's holdings of asset i, and so on.

The objective in (1) may be described as *net utility*—utility penalized for transaction costs. Since utility here is expected dollar return less a penalty for risk, and transaction costs τ are also in dollars, this objective may also be described as expected dollar return penalized for risk and net of transaction costs. Note that an increase of one dollar in transaction costs is exactly equivalent to a reduction of one dollar in portfolio expected return, because these have exactly the same effect on a client's return; we have made this equivalence explicit in the way we formulate the objective function. The fact that each objective is expressed in dollars will be important in enabling us to combine objectives across clients.

For some portfolio selection problems, t_k is not an independent decision variable but rather is determined uniquely by x_k and the fixed initial positions x_k^0. This is the case when there is only one way to trade from x_k^0 to x_k. In other situations, there may be many ways to implement the desired trades. For example, if a euro-pound forward could also be implemented as a euro-dollar forward plus a dollar-pound forward, then t_k would be a decision variable capturing this possibility.

Multi-Account Optimization

We introduce a portfolio optimization problem combining the optimization problems of our K clients in a direct manner:

$$\begin{aligned}&\text{Maximize } U_1(x_1) + U_2(x_2) + \cdots + U_K(x_K) - \tau(t)\\ &\text{subject to } x_k \in C_k \text{ for } k = 1, 2, \ldots, K\end{aligned} \tag{2}$$

Here we simply add the utility functions of the K separate problems defined in (1), and we replace the separate-account transaction cost model with a model that is applied to aggregate trading t. The vector t summarizes all the trading requirements of the clients in a parsimonious way and reflects how the trades are to be implemented. At a minimum, t must specify the total amount of buying and selling required in each asset, but it may also be necessary for t to convey information about how the trading is to be carried out. Typically, t will take the form $(t_1^+, t_2^+, \ldots, t_n^+, t_1^-, t_2^-, \ldots, t_n^-)$, specifying aggregate buys t_i^+ and sells t_i^- in each asset $i = 1, 2, \ldots, n$.

The transaction cost model τ at this point is a general increasing convex function, and its exact form in a given problem will depend, as t does, on the details of how trading is implemented. The function τ was not previously assumed to be additive across different assets, allowing trades in one asset to influence trading costs in another. Let us now simplify this model by assuming that it is additive across assets. This means that we have:

$$\tau(t) = \sum_{i=1}^{n} \tau_i\,(t_i^+, t_i^-) \tag{3}$$

where $\tau_i\,(t_i^+, t_i^-)$ is the cost of trading asset i as a function of the aggregate buys and sells for that asset.

When trades cannot be crossed, we may have to simultaneously buy a particular asset for one client and sell that asset for another client, so that both t_i^+ and t_i^- are positive. It may be tempting to decompose $\tau_i\,(t_i^+, t_i^-)$ into a sum of a cost of buying and a cost of selling, so that $\tau_i(t_i^+, t_i^-) = \tau_i^+(t_i^+) + \tau_i^-(t_i^-)$, but this is unreasonable because buying and selling in the same asset tends to have offsetting effects on prices, even if the trades are executed through different brokers.

The transaction cost model must capture this phenomenon, and to do this we must go back to first principles and extend the mathematical modeling and market data analysis underlying the single-account transaction cost model to cover the case of simultaneous buying and selling of an asset.

Is the multi-account optimization (2) just a naive combination of the separate optimizations (1)? We argue that not only is this approach reasonable, but also that it is the best solution to the problem of simultaneous trading.

Justification: Competitive Equilibrium in the Market for Liquidity

Multi-account optimization may be described as maximizing the clients' total welfare. Our main insight is that this is equivalent to a market equilibrium where clients compete for liquidity. Multi-account optimization describes how clients would behave if they sought liquidity independently in an efficient market.

These two interpretations—maximization of total welfare and market equilibrium—form the basis of the claim that multi-account optimization is the best solution to the problem of simultaneous trading. Total welfare maximization assures us that the solution is good, while the equilibrium interpretation assures us that it is fair.

To explain these ideas in more detail, consider client k's problem (1), but now assume a linear transaction cost function with arbitrary marginal transaction costs π, so that $\tau(t_k) = t_k \cdot \pi$. This model means that the entries of π

give the prices of liquidity—the prices per unit of buying and selling in the different assets.

The solution to problem (1) with this transaction cost model determines the client's trading requirements, or demand, when liquidity prices are given by π, and we write this demand as $D_k(\pi)$. The supply of transactions at these prices $S(\pi)$ can be identified as the level of trading at which marginal transaction costs equal the entries of π.[1]

Market equilibrium is determined by setting supply equal to total demand:

$$\begin{aligned} S(\pi) &= D(\pi) \\ \text{where } D(\pi) &= D_1(\pi) + D_2(\pi) + \cdots + D_K(\pi) \end{aligned} \tag{4}$$

But this equilibrium condition is also the first-order condition for optimality of the multi-account optimization problem (2). This is an informal demonstration of the equivalence of multi-account optimization and market equilibrium.[2]

The objective function of (2) is the total welfare of all clients, measured in dollars. It is an example of what is called a *social welfare function* in the microeconomics literature (see Mas-Colell, Whinston, and Green [1995]). Since multi-account optimization maximizes this welfare, it follows that the resulting allocation of trading resources is Pareto-optimal: We cannot improve the lot of one client by reallocating trades without worsening the lot of another. The proof of this fact is that if such a reallocation were possible, it would increase total welfare, contradicting optimality of the solution to (2).

To eliminate a weakness in the argument that the multi-account optimization (2) is fair to all clients, we must require that it have a unique solution. Otherwise, there is a possibility that a particular solution could be selected to favor one client over another. Uniqueness of the solution is assured by assuming strict convexity of the objective functions in (1), which in most practical problems follows from the no-arbitrage condition that there is only one riskless portfolio.

We have given short shrift to one market participant here, namely, the broker who provides liquidity. Her utility does not appear in our analysis. It is not her welfare that is our concern, rather only the prices she charges for a given level of trading. In multi-account optimization, our goal with respect to the broker is simply to minimize payments to her, not to enhance her welfare. So the market we have described is competitive only with respect to the clients' interactions with one another. The clients then form an oligopsony, a group of buyers with pricing power, collectively determining prices favorable to them in the market for liquidity (Pindyck and Rubinfeld [2005]).

An Iterative Algorithm

It is interesting to observe that multi-account optimization can also be motivated through a natural iterative algorithm for determining trading. We took this approach first, before settling on the multi-account optimization framework. Its appeal is that it is easily implemented because it leverages an algorithm for solving separate account optimizations into an algorithm for solving multi-account optimizations.

In this approach, we first solve each client's portfolio selection problem (1) in isolation, assuming a linear transaction cost function with arbitrary initial liquidity prices π_0, so that $\tau(t_k) = t_k \cdot \pi_0$. Next, we aggregate the trading needs of all clients resulting from this optimization and compute the actual marginal transaction costs π_1 based on this level of trading. Then we optimize the accounts in isolation again, but using the new marginal transaction costs π_1. This results in updated trading requirements and corresponding updated marginal transaction costs π_2. This process can be repeated iteratively.

Under certain assumptions, these iterations converge to the same result as multi-account optimization. To see this, note that the marginal transaction costs in the iterative algorithm may be viewed as the prices of liquidity in the various assets. The iterations constitute an algorithm for setting these prices so that the level of trading services provided by the broker matches the level demanded by the clients—i.e., so that (4) holds. This algorithm is analogous to the Walrasian auctioneer's *tatonnement* process for identifying market-clearing prices (see, for example, Sawyer [1989]).

As a practical matter, the iterative algorithm is not computationally reliable, because there is no mathematical guarantee of convergence; and even if there is convergence, it may be slow.[3] The multi-account optimization problem, however, is convex under most reasonable models of utility and transaction costs. Convex optimization is a robust, reliable technology, for which there are many excellent computer packages available. This optimization technology enables us to solve the large-scale problems, with many thousands of variables, that arise from multi-account portfolio selection.[4]

Level of Trading in Multi-Account Optimization

Multi-account optimization determines the vector of aggregate trades, which we denote by t^*, and associated with this level of trading there is a vector π^* of liquidity prices or marginal costs of trading. Suppose client k decides to trade further at these marginal prices, for her own account and independently of the manager. She will find she can gain no advantage from such trading.

To see this, consider this client's separate account optimization problem (1) above, but take τ to be the linear transaction cost model with marginal transaction costs π^* so that $\tau(t_k) = t_k \cdot \pi^*$. Now the optimal portfolio for this problem is simply the optimal portfolio from multi-account optimization. This is because the first-order conditions for optimality in the multi-account problem (2) imply the first-order conditions for optimality in each separate account problem (1) with this linear transaction cost model.

This is a key feature of multi-account optimization: that it produces portfolios that are optimal for each client separately, given the actual marginal transaction costs π^* associated with the aggregate trading t^*. Any client who tries to improve her lot by trading independently—buying or selling liquidity at the going marginal prices π^*—would gain no benefit.

So, given the aggregate level of trading, each client's portfolio is optimal at prices π^*. Therefore, the only reason a client may be unhappy with the resulting portfolio is that the trading demands of other clients have driven the individual transaction costs up. The same complaint could be made whenever several clients invest in a single fund, or, more generally, whenever there are multiple market participants competing for liquidity.

We can ensure that a client will not trade inordinately at a high marginal cost by introducing a transaction cost tolerance parameter $v_k < 1$ into the utility function, replacing U_k by $v_k U_k$ in (1) and (2), but of course this would force the client to forgo the full benefit of the available investment information. If this client trades less, the other clients will get the benefit of reduced transaction costs. This is not consistent with the fairness goal of multi-account optimization.

To further challenge multi-account optimization, suppose a client complains she is unhappy with the current portfolio and would like to trade more of a particular asset. This might happen if the client has a transaction cost tolerance parameter greater than 1 ($v_k > 1$). The client does not want to trade independently at market prices, π^*, but rather wants to trade within the multi-account framework so that she pays the average cost per share.

This poses two questions. First, would it benefit the client if we were to accommodate her? And second, is it appropriate for us to accommodate her? The answers are yes, and no. Since marginal costs rise with trade size, paying the average cost per share rather than the marginal cost is a bargain for the client. Since her marginal benefit from trading is currently equal to marginal costs, if we were to reduce her costs in this way she would benefit from trading more, as would any client.

But to allow her to do this would immediately reduce total welfare by driving up trading costs for everyone else, destroying the competitive equilibrium that is the basis of the fairness of the model. Why should one client get the benefit of average costs while also exercising pricing power against the other clients (Pindyk and Rubinfeld [2005])?

MULTI-ACCOUNT OPTIMIZATION WHEN TRADES CANNOT BE CROSSED

An ERISA regulation forbids the crossing of trades.[5] That is, when an asset is to be sold on behalf of one client and bought on behalf of another, it is not permissible simply to transfer the asset from one account to the other—the separate buy and sell transactions must be executed in the market. The multi-account formulation we develop observes this rule.

In aggregating the trades of different clients, the buy orders of all accounts in each asset will be combined and traded together, and the sell orders of all accounts in each asset will be combined and traded together. Because trades cannot be crossed, it is possible—although perhaps unlikely—that an asset will be bought on behalf of some accounts but sold on behalf of other accounts at the same time.

Fortunately, we can explain explicitly how the separate account problems can be brought together without having to make detailed assumptions about the form of those problems. To formulate the multi-account optimization problem, various auxiliary variables must be introduced to take account of aggregate trading (see Rardin [1998]). We call these variables *buy variables, aggregate buy variables*, and *aggregate sell variables*.

We first deal with the units of the variables x_{ki}, which could represent dollar holdings, number of shares, or weights. Constants n_{ki} convert the x_{ki} to dollars, in the sense that $n_{ki} x_{ki}$ is always client k's dollar holdings of asset i. If the x_{ki} represent shares or dollars, n_{ki} will represent the share price of asset i or 1. If the x_{ki} represent weights, then in a long-only portfolio n_{ki} will represent the total portfolio value for client k, while in a long-short portfolio n_{ki} will represent the notional value for client k.

Buy Variables

For each asset i traded by client k we introduce a non-negative buy variable $x_{ki}^{+} \geq 0$ into the optimization that provides an upper bound on the weight of asset i that client k will buy:

$$x_{ki} - x_{ki}^{0} \leq x_{ki}^{+} \quad \text{for } i = 1, 2, \ldots, n \text{ and } k = 1, 2, \ldots, K \tag{5}$$

In the optimal solution, x_{ki}^{+} will equal the actual amount of asset i that client k will buy. If the client is selling rather than buying, then x_{ki}^{+} will be 0.

Aggregate Buy Variables

The buy variables for each asset i are combined into a variable $t_i^{+} \geq 0$ representing the total dollar amount of buying in that asset for all clients, which we call an aggregate buy variable. This is defined mathematically by the equation:

$$t_i^{+} = \sum_{k=1}^{K} n_{ki} x_{ki}^{+} \qquad \text{for } i = 1, 2, \ldots, n \tag{6}$$

Aggregate Sell Variables

We could first introduce sell variables x_{ki}^{-} in the same way as the buy variables x_{ki}^{+} in (5), but this would substantially increase the number of variables and constraints in the multi-account optimization. Instead we use the fact that buys and sells in a given asset must total to trades, so we must have:

$$t_i^{+} - t_i^{-} = \sum_{k=1}^{K} n_{ki} (x_{ki} - x_{ki}^{0}) \quad \text{for } i = 1, 2, \ldots, n \tag{7}$$

where $t_i^{-} \geq 0$ is the aggregate dollar amount of selling in asset i.

Non-Negativity Constraints on Trade Variables

Buy variables, aggregate buy variables, and aggregate sell variables are all non-negative:

$$x_{ki}^{+} \geq 0, \; t_i^{+} \geq 0, \; t_i^{-} \geq 0 \tag{8}$$
$$\text{for } i = 1, 2, \ldots, n \quad \text{and} \quad k = 1, 2, \ldots, K$$

With these definitions and constraints, the multi-account portfolio optimization problem under the ERISA no cross-trading rule may be formulated as follows. We assume again that transaction costs are additive across assets, so that the transaction cost function satisfies Equation (3):

$$\text{Maximize } \sum_{k=1}^{K} U_k(x_k) - \sum_{i=1}^{n} \tau_i \; (t_i^{+}, t_i^{-})$$
$$\text{subject to (5)-(8) and } x_k \in C_k, k = 1, 2, \ldots, K \tag{9}$$

Cross-Trades Allowed

When we are allowed to cross trades, there is no need for the buy variables x_{ki}^{+}, or for constraints (5) and (6), because (7) is all we need to determine the aggregate buy and sell variables t_i^{+} and t_i^{-} in the presence of the non-negativity constraints t_i^{+}, $t_i^{-} \geq 0$. Moreover, in this case we never have to buy and sell an asset simultaneously, so the transaction cost model developed for the single-account case will be adequate as long as it remains accurate for the higher levels of trading resulting from aggregating the clients' transactions.

EXAMPLES

We illustrate multi-account optimization with examples assuming an investment universe of $n = 10$ uncorrelated assets. Expected returns μ_i range from 11% to 20% in ten equal steps, and volatilities σ_i range from 5.5% to 10.0%, also in equal steps. Transaction costs are taken to be the same for all assets, and are specified as $\tau(t) = \theta t^{\gamma}$, where t denotes the trade size, and θ and γ are constants satisfying $\theta \geq 0$ and $\gamma \geq 1$. In most of our examples there are $K = 2$ clients, each with a goal of maximizing expected return net of transaction costs, subject to various constraints. We take the x_{ki} to represent dollar holdings so that $n_{ki} = 1$ for all assets and clients. To further simplify, we assume we are starting from zero holdings in all assets for all clients.

This allows us to write the objective functions of the separate accounts $k = 1, 2$ as follows:

$$\text{Maximize } \sum_{i=1}^{n} x_{ki} \mu_i - \tau_k \tag{10}$$
$$\text{where } \tau_k = \theta \sum_{i=1}^{n} |x_{ki}|^{\gamma}$$

In terms of the general single-account optimization problem of Equation (1), the expected return term here is the utility function U_k, and of course τ_k is the transaction cost function. Each account has an upper bound on risk, which we denote by $\overline{\sigma}_k, k = 1, 2$, that is required to be cash-neutral:

$$\sum_{i=1}^{n} x_{ki}^2 \sigma_i^2 \le \overline{\sigma}_k^2 \quad \text{and} \quad \sum_{i=1}^{n} x_{ki} = 0 \tag{11}$$

In some of the scenarios discussed below, we add a constraint forbidding the first account to hold asset 10, which is the asset with the highest alpha:

$$x_{1,10} = 0 \tag{12}$$

We introduce this constraint to highlight the way multi-account optimization treats clients with different investable universes. Constraints (11) and (12) define the feasible sets C_k of the individual clients. To set up the multi-account optimization under the ERISA no cross-trading rule, we roughly follow the formulations in Equations (5)-(9). We begin by introducing the buy variables x_{ki}^+, and analogous sell variables x_{ki}^-, via the constraints:

$$x_{ki} = x_{ki}^+ - x_{ki}^- \quad \text{and} \quad x_{ki}^+,\ x_{ki}^- \ge 0 \quad \text{for } k = 1, 2 \text{ and } i = 1, 2, \cdots, n \tag{13}$$

Aggregate buy and sell variables for each asset are given by:

$$t_i^+ = x_{1i}^+ + x_{2i}^+ \quad \text{and} \quad t_i^- = x_{1i}^- + x_{2i}^- \quad \text{for } i = 1, 2, \cdots, n \tag{14}$$

The sell variables x_{ki}^- help to make the formulation a little easier to follow. With this notation, the multi-account optimization problem is to maximize expected returns net of transaction costs, which may be written as:

$$\sum_{i=1}^{n} \left(x_{1i} + x_{2i}\right)\mu_i - \tau \tag{15}$$

$$\text{where } \tau = \theta \sum_{i=1}^{n} \left(\left(t_i^+\right)^{\gamma} + \left(t_i^-\right)^{\gamma} \right)$$

subject to (11), (13), and (14), with constraint (12) added in some cases. To keep the analysis simple, we have ignored any offsetting effects of simultaneous buying and selling in the same asset, and so have assumed additivity of buy and sell costs for each asset.

Performance Measures

For each optimization, we report several measures related to performance and transaction costs, including the information ratio (IR), turnover, and total transaction costs. We also partition total transaction costs into the costs incurred by buying and the costs incurred by selling. The information ratio is computed as portfolio expected return net of transaction costs, divided by portfolio risk. We refer to this as *net IR*.

The transaction costs τ_k incurred for client k in the single-account optimization are given explicitly in (10). In the multi-account optimization, the total transaction costs are given by the term τ defined in (15), and these costs must be apportioned between the two accounts.

A given client's share of the costs of buying or selling an asset is simply the proportion of the trading done on her behalf. In other words, all clients pay the average cost per share.

This implies that we can decompose a client's transaction costs into separate buy and sell costs as follows:

$$\tau_k = \tau_k^+ + \tau_k^-$$

$$\text{where } \tau_k^+ = \theta \sum_{i=1}^{n} x_{ki}^+ \left(t_i^+\right)^{\gamma-1}$$

$$\text{and } \tau_k^- = \theta \sum_{i=1}^{n} x_{ki}^- \left(t_i^-\right)^{\gamma-1}$$

In addition to computing the transaction costs incurred by each account separately, we also report the total cost of buying $\tau^+ = \tau_1^+ + \tau_2^+$ and the total cost of selling $\tau^- = \tau_1^- + \tau_2^-$.

Computational Results

We begin with three experiments, each involving two clients trading simultaneously. Transaction costs are zero ($\theta = 0$) in the first experiment, linear ($\gamma = 1$) in the second, and quadratic ($\gamma = 2$) in the third. In each of these experiments we show that the multi-account optimization treats the two accounts in a reasonable way. Each experiment explores three scenarios:

Scenario A assumes a risk limit of 5% for both accounts. This is the base case against which the other two cases are compared.

Scenario B enforces a smaller risk budget of 3% for account number 1. The goal is to see how the multi-account optimization treats accounts that differ in trading needs.

Scenario C is identical to Scenario A except that constraint (12) is imposed on client number 1. The goal is to see how the multi-account optimization treats accounts that differ in their investment universes.

For each scenario, we solve three optimizations: the two separate account optimizations and the multi-account optimization. Because the transaction costs are low compared to the expected returns, the risk limits are attained in all optimizations below. All calculations for the examples are carried out using NuOPT for S-Plus. Details on NuOPT and its use in portfolio optimization can be found in Scherer and Martin [2005].

Experiment 1: No transaction costs. When $\theta = 0$ the transaction costs are zero, and there is nothing to connect the two accounts. In each scenario, the multi-account optimization produces exactly the same portfolios as the separate account optimizations, and this of course is exactly how it should behave in this case.

In Scenario A, the accounts have identical constraints, each having risk budgets of 5%. Because of this, their optimal portfolios (not shown) are also identical, as are their performance measures reported in Exhibit 1.

Scenario B, for which we set $\bar{\sigma}_1 = 3\%$ and $\bar{\sigma}_2 = 5\%$, shows what happens if the accounts have different risk budgets, reflecting different levels of client aggressiveness. In fact, the optimal positions here are proportional to risk. The net IRs—which are simply IRs in this case as transaction costs are zero—are identical, as they are unaffected by leverage. For the same reason, turnover is proportional to risk, as we see from Exhibit 1: $0.5548 / 0.9246 = 0.6 = \bar{\sigma}_1 / \bar{\sigma}_2$.

In Scenario C we impose constraint (12), which forbids account number 1 to hold the asset with the highest expected return. The net IR of account number 1 drops, but again each account gets the same optimal portfolio, whether it is optimized on its own or with the other account.

Experiment 2: Linear transaction costs. Setting $\gamma = 1$, our transaction cost model becomes linear. By taking $\theta = 0.1$, we calibrate the costs to a reasonable level; they result in a modest reduction in trade sizes compared to Experiment 1. Superficially, the multi-account optimization now appears to forge a link between the two accounts via the transaction cost term. As marginal transaction costs are constant here, trades in one account will not alter the marginal costs in the other account. Again the results of a combined optimization will be identical to the results of two separate optimizations in every scenario, and this is confirmed in Exhibit 2. Again, the multi-account optimization is behaving exactly as we would wish.

The increased transaction costs are manifest in lower turnover, smaller positions, and lower net IRs across all nine optimizations, compared to Experiment 1. The net IR has fallen from 0.3 to 0.2817 in all cases except Scenario C, where account number 1 cannot hold asset 10, and there the drop is from to 0.2634 to 0.2457.

In Exhibit 2 we also see that buy costs (τ^+) and sell costs (τ^-) are equal in all optimizations, a property that will be lost under non-linear costs as we shall see in Experiment 3.

EXHIBIT 1
Optimization of Two Accounts with Zero Transaction Costs ($\theta = 0$)

Scenario	Scenario A			Scenario B			Scenario C		
Constraints	$\bar{\sigma}_1 = \bar{\sigma}_2 = 5\%$			$\bar{\sigma}_1 = 3\%, \bar{\sigma}_2 = 5\%$			$\bar{\sigma}_1 = \bar{\sigma}_2 = 5\%, x_{1,10} = 0$		
Account	#1	#2	#1 & #2	#1	#2	#1 & #2	#1	#2	#1 & #2
Net IR #1	-	0.3000	0.3000	0.3000	-	0.3000	0.2634	-	0.2634
Net IR #2	0.3000	-	0.3000	-	0.3000	0.3000	-	0.3000	0.3000
$\sum_i \lvert x_{1i} \rvert$	-	0.9246	0.9246	0.5548	-	0.5548	0.8946	-	0.8946
$\sum_i \lvert x_{2i} \rvert$	0.9246	-	0.9246	-	0.9246	0.9246	-	0.9246	0.9246
τ_1	-	-	-	-	-	-	-	-	-
τ_2	-	-	-	-	-	-	-	-	-
τ^+	-	-	-	-	-	-	-	-	-
τ^-	-	-	-	-	-	-	-	-	-

EXHIBIT 2
Optimization of Two Accounts with Linear Transaction Costs ($\theta = 1/10, \gamma = 1$)

Scenario	Scenario A			Scenario B			Scenario C		
Constraints	$\bar{\sigma}_1 = \bar{\sigma}_2 = 5\%$			$\bar{\sigma}_1 = 3\%, \bar{\sigma}_2 = 5\%$			$\bar{\sigma}_1 = \bar{\sigma}_2 = 5\%, x_{1,10} = 0$		
Account	#1	#2	#1 & #2	#1	#2	#1 & #2	#1	#2	#1 & #2
Net IR #1	0.2817	-	0.2817	0.2817	-	0.2817	0.2457	-	0.2457
Net IR #2	-	0.2817	0.2817	-	0.2817	0.2817	-	0.2817	0.2817
$\sum_i \lvert x_{1i} \rvert$	0.9037	-	0.9037	0.5422	-	0.5422	0.8762	-	0.8762
$\sum_i \lvert x_{2i} \rvert$	-	0.9037	0.9037	-	0.9037	0.9037	-	0.9037	0.9037
τ_1	0.0904	-	0.0904	0.0542	-	0.0542	0.0876	-	0.0876
τ_2	-	0.0904	0.0904	-	0.0904	0.0904		0.0900	0.0904
τ^+	0.0452	0.0452	0.0904	0.0270	0.0452	0.0723	0.0438	0.0450	0.0890
τ^-	0.0452	0.0452	0.0904	0.0270	0.0452	0.0723	0.0438	0.0450	0.0890

Experiment 3: Non-linear transaction costs. This is the most interesting case because here the accounts actually interact. For this experiment we set $\gamma = 2$, making the transaction cost model quadratic. By taking $\theta = 1$ we calibrate the model to bring about a modest reduction in trading compared to Experiment 1. The results are presented in Exhibit 3. For the multi-account optimizations, any trading in one account will affect the marginal transaction costs, and therefore the optimal portfolio, of the other account.

While individual optimizations under Scenario A lead to a net IR of 0.2760 for both accounts, slightly lower than the 0.2817 of Experiment 2, the net IR under multi-account optimization drops further, to 0.2529. High transaction costs due to non-linearity are the reason for this drop. While transaction costs are 0.1178 for each account when traded separately, they jump to 0.2277 under aggregate trading.

In Scenario B, the drop in net IR is mitigated by the reduced trading demands of client number 1, which reflect her smaller risk budget. Perhaps surprisingly, net IR is the same for both clients here too, despite the non-linearity of the transaction cost function. This is due to the special nature of quadratic transaction costs, which makes the underlying positions (not shown) of the two clients again proportional.

The drop in net IR in Scenario C due to simultaneous trading is about 9.0%, in relative terms, for account number 1 (from 0.2388 to 0.2174) but only 7.9% (from 0.2758 to 0.2540) for account number 2. Because account number 1 is not allowed to hold the asset with the highest alpha, it attains its 5% risk budget by taking larger positions in the other assets. The non-linearity forces up the marginal costs of trading these assets. Account number 2, on the other hand, can invest in asset 10 without having to worry about account number 1 driving up transaction costs for that asset. While transaction costs are higher for both accounts, the impact appears to be greater for account number 1 than account number 2.

Is the multi-account solution in Scenario C reasonable, despite the differential impacts of aggregate trading on net IR? In fact there is no reason why net IRs should decline proportionally for the two clients under multi-account optimization.

The clients' objectives are in fact to maximize the net expected return within a 5% risk budget, and the net expected return falls by 10.9 basis points [$0.0109 = 5\% \times (0.2388 - 0.2174)$] for the first account and 10.7 basis points [$0.0107 = 5\% \times (0.2758 - 0.2540)$] for the second. Optimizing the accounts together has had virtually no effect on client number 2's advantage over client number 1. Client number 2 derives more benefit than client number 1 under multi-account optimization because she has a larger universe, not because she has been given an unfair advantage in access to liquidity.

But ultimately what assures us that the multi-account portfolios of Scenario C are fair is the fact that the same result would have been achieved had the two clients gone directly to the market and traded competitively, under our model of transaction costs and client utility.

In all scenarios of Experiment 3, the transaction

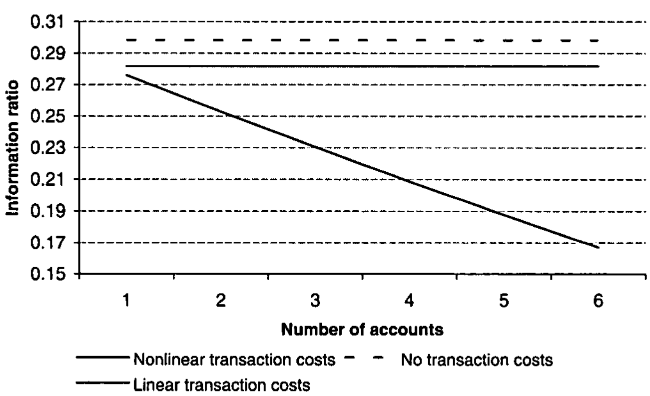
Exhibit 3

Optimization of Two Accounts with Quadratic Transaction Costs ($\theta = 1, \gamma = 2$)

Scenario	Scenario A			Scenario B			Scenario C		
Constraints	$\bar{\sigma}_1 = \bar{\sigma}_2 = 5\%$			$\bar{\sigma}_1 = 3\%, \bar{\sigma}_2 = 5\%$			$\bar{\sigma}_1 = \bar{\sigma}_2 = 5\%, x_{1,10} = 0$		
Account	#1	#2	#1 & #2	#1	#2	#1 & #2	#1	#2	#1 & #2
Net IR #1	0.2760	-	0.2529	0.2800	-	0.2620	0.2388	-	0.2174
Net IR #2	-	0.2760	0.2529	-	0.2700	0.2620	-	0.2759	0.2540
$\sum_i \lvert x_{1i} \rvert$	0.9200	-	0.9115	0.5500	-	0.5492	0.8948	-	0.8911
$\sum_i \lvert x_{2i} \rvert$	-	0.9200	0.9115	-	0.9200	0.9154	-	0.9196	0.8968
τ_1	0.1178	-	0.2277	0.0430	-	0.1108	0.1233	-	0.2225
τ_2	-	0.1178	0.2277	-	0.1180	0.1846	-	0.1148	0.2146
τ^+	0.0433	0.0433	0.1770	0.0150	0.0430	0.1126	0.0470	0.0430	0.1626
τ^-	0.0746	0.0746	0.2776	0.0270	0.0700	0.1828	0.0767	0.0750	0.2745

Exhibit 4

Net Information Ratio versus Number of Accounts



costs for sales, τ^+, are substantially higher than for purchases, τ^-, indicating that the short positions are generally more concentrated than the long positions. The reason for this is that the assets with the lowest means—which are most desirable for short positions—are doubly attractive to short because their volatilities are also the lowest. On the other hand, the assets with the highest means—attractive for long positions—are made less attractive because of their higher volatilities. Thus assets 1, 2, and 3 tend to be heavily shorted, while long positions are spread fairly evenly over assets 6–10.

Experiment 4: Adding new accounts. As we add ever more accounts to a given strategy, market impact and transaction costs inevitably rise. Depending on the size and liquidity of a given market, the decay in net IR can be substantial. We extend Scenario A of Experiment 3 to investigate the question of capacity of an investment management strategy as there are more clients. We successively add identical accounts with 5% risk. The results are summarized in Exhibit 4.

We see that net IR decays steadily. By the time we have reached five accounts, the net IR is approximately halved. This is hardly a scalable approach. Naive portfolio construction, ignoring the increased transaction costs due to pooling trades when transaction costs are non-linear, would have led us to anticipate a much higher net IR. The manager would be well advised to limit the number of clients in the strategy in order to justify a given level of fee.

CONCLUSIONS

When many accounts are traded simultaneously, transaction costs typically increase for each account. This presents a technical problem: keeping track of the trades of all clients as we optimize a given client's portfolio. It also presents the philosophical problem of how to mediate between clients and ensure fairness with respect to trading decisions.

Our solution to these problems is to combine the separate optimizations into a single multi-account optimization in a natural way. The justification of this approach is that it leads to trading choices that the clients would make for themselves if they were to act independently in their own best interests in an efficient market for liquidity. The marginal costs of trades in each asset under multi-account optimization are precisely the market-clearing prices in an oligopsony—a buyer-dominated market—where clients attain a competitive equilibrium in the market for liquidity provided by a price-taking broker.

In addition to the efficient markets interpretation of multi-account optimization, the procedure has a number of other properties:

- Total welfare of all clients is maximized.
- The allocation of trading resources is Pareto-optimal, so that no reallocation of liquidity among the clients will improve the lot of one without worsening the lot of another.
- Each client's portfolio is optimal, given the prices of liquidity resulting from the aggregate trades, so there is no advantage for any client in doing further trading for her own account.

Multi-account optimization is fundamentally different from traditional portfolio optimization. Its purpose is not simply to make the best choice for each individual client. Its purpose is rather to make the best choices for each client while ensuring fairness in the allocation of liquidity. The goal of ensuring fairness, when there is interaction between clients due to trading, is what makes equilibrium ideas from microeconomics relevant to portfolio selection.

While there are some technical details involved in formulating multi-account optimization under ERISA rules, and despite the extent of some multi-account portfolio selection problems, modern convex optimization technology enables us to solve them quickly and accurately.

ENDNOTES

The authors thank Joshua Feinman, Petter Kolm, Slava Malkin, Mikhail Solodukhin, Ronghui Tao, Greg van Inwegen, and Robert Wang for helpful discussions and insights.

[1]We do not attempt a formal mathematical treatment here. For the sake of exposition, we assume there is always a unique solution to problem (1) and that for any set of prices π there is a unique level of trading t for which π is the vector of marginal costs of trading. A more formal treatment would rest on ideas on general equilibrium pioneered in the 1950s by Arrow and Debreu—see, for example, Debreu [1955] and Starr [1997].

[2]A simple insight into (4) is as follows. If client k requires trades $t_k = D_k(\pi)$, then by definition the marginal benefit of more trading to her is π. When the broker provides the aggregate trading services $t = S(\pi)$, then the marginal cost of trading is also π. So if we can choose π so that markets clear, meaning that $t = t_1 + t_2 + \cdots + t_n$, which is the same as Equation (4), buyers and sellers all agree on the price. Agreeing on the price is just the first-order condition for optimality in the multi-account optimization problem (2).

[3]The approach described here is essentially a fixed-point iteration algorithm for (1), given by $\pi_{n+1} = S^{-1}(D(\pi_n))$. This is guaranteed to converge only under special conditions, such as if the function $\pi \rightarrow S^{-1}(D(\pi))$ is a contraction mapping (Granas and Dugundji [2003]).

[4]See Boyd and Vandenberghe [2004] for an up-to-date treatment of convex optimization. Of course, many portfolio optimization problems are non-convex, including, for example, problems with round lot constraints, and problems with constraints on minimum position or trade sizes. The features that lead to non-convexity usually arise from technical trading and portfolio management considerations rather than investment considerations. Such features are best thought of as minor but technically challenging refinements of problems that are in essence convex. The usual approach to solving such non-convex problems is to add mixed-integer heuristics to a convex optimizer.

[5]Section 406(b)(3) of the Employee Retirement Income Security Act of 1974 (ERISA) prohibits cross-trading.

REFERENCES

Boyd, Stephen, and Lieven Vandenberghe. *Convex Optimization*. New York: Cambridge University Press, 2004.

Debreu, Gerard. *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*. New Haven: Cowles Foundation, 1955.

Granas, Andrzej, and James Dugundji. *Fixed Point Theory*. New York: Springer-Verlag, 2003.

Mas-Colell, Andreu, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. New York: Oxford University Press, 1995.

O'Connell, J. *Welfare Economic Theory*. Boston: Auburn House Publishing, 1982.

Pindyck, Robert S., and Daniel L. Rubinfeld. *Microeconomics*, 6th ed. New York: Prentice Hall, 2005.

Rardin, Ronald L. *Optimization in Operations Research*. New York: Prentice-Hall, 1998.

Sawyer, John A. *Macroeconomic Theory*. Philadelphia: University of Pennsylvania Press, 1989.

Scherer, Bernd, and Douglas R. Martin. *Modern Portfolio Optimization with NuOPT for S-Plus*. New York: Springer-Verlag, 2005.

Starr, Ross M. *General Equilibrium Theory*. Cambridge: Cambridge University Press, 1997.

To order reprints of this article, please contact Dewey Palmieri at dpalmieri@iijournals.com or 212-224-3675.

©Euromoney Institutional Investor PLC. This material must be used for the customer's internal business use only and a maximum of ten (10) hard copy print-outs may be made. No further copying or transmission of this material is allowed without the express permission of Euromoney Instituitonal Investor PLC. Copyright of Journal of Portfolio Management is the property of Euromoney Publications PLC and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.
The most recent two editions of this title are only ever available at http://www.euromoneyplc.com