PORTFOLIO OPTIMIZER

Authors: Alper Atamtürk and Marty O'Brien

Date: October 2019

Version: 3.21.6

Bloomberg

PORT<GO> Portfolio and Risk Analytics

Bloomberg Portfolio Optimizer is a part of the suite of portfolio and risk analytics PORT<GO> of the Bloomberg Professional® service. PORT<GO> provides a comprehensive set of investment portfolio analytics including performance attribution, characteristics, risk, scenario analysis and portfolio construction.

Portfolio Optimization Highlights

- Multi-asset class support (equities, fixed income, currencies, FX forwards, commodities, index and bond futures, listed equity options, CDS, IRS)
- Support for portfolios of composite securities, e.g., ETFs, mutual funds, indices
- Optimization backtesting using point-in-time data
- Efficient frontier optimization
- Long-only as well as long-short optimization
- Support for separate constraints on the long and short sides of the portfolio
- Risk (volatility), Value-at-Risk, Conditional Value-at-Risk goal and/or constraints, possibly relative to a benchmark
- Sharpe ratio, information ratio maximization and/or constraint
- Cash-flow matching, net-present-value constraint/goal for asset-liability management
- Gross active weight (active share) goal and/or constraint
- Custom goal and/or constraint fields, e.g., user-defined alphas, factors
- Factor exposure/characteristics objective and/or constraints
- Multiple goals with user-defined trade-offs
- Weight constraints on groups of securities, e.g. sectors, countries, issuers, ratings
- UCITS 5/10/40 diversification compliance rule
- Hard as well as soft constraints
- Optimization with Bloomberg and/or custom scenarios
- User-defined as well as Bloomberg TCA/LQA transaction cost with market impact
- Round position and round trade lots
- Bounds on the number of trades, buys, sells, positions, longs, shorts
- Nonconvex position and trade threshold constraints
- Seamless integration with the wealth of Bloomberg data and portfolio analytics
- Sophisticated preprocessing and search algorithms and latest second-order conic mixed-integer optimization technology for speed and numerical accuracy

Table of Contents

1 Bloomberg Portfolio Optimizer

Bloomberg Portfolio Optimizer is a part of the suite of portfolio and risk analytics PORT<GO> of the Bloomberg Professional® service. Bloomberg Portfolio Optimizer offers portfolio managers advanced capabilities for constructing and re-balancing global multi-asset-class portfolios with tailored risk, expected return and exposure characteristics. With Bloomberg Portfolio Optimizer users can not only find optimal weights for each security to control their risk and expected return, but also select a small set of securities to achieve this goal from a large investment universe. Bloomberg Portfolio Optimizer is available for multi-asset-class portfolios including equities, fixed income, currencies, FX forwards, commodities, index and bond futures, listed equity options, credit default swaps, interest rate swaps.

This document describes the concept of portfolio optimization, the functionality of the Bloomberg Portfolio Optimizer, and examples of its application. It is organized as follows. Section 2 provides an introduction to portfolio optimization and illustrates optimal risk-return profiles with a simple example. Section 3 outlines the different components that comprise a portfolio optimization task, including the objective and different types of constraints that may be specified. Section 4 shows the application of portfolio optimization in passive and active investment strategies, where the performance of the constructed portfolios is evaluated using historical backtesting. Appendix A defines the types of *optimization fields*, which are quantities that make up the definition of the portfolio objective or a portfolio constraint. Finally, Appendix B briefly describes the Bloomberg risk factor model used by Bloomberg Portfolio Optimizer.

2 Portfolio Optimization

Portfolio optimization is an advanced process for making intelligent choices among a vast set of possible investment decisions often involving trade-offs between conflicting goals and limited resources. It is a process of systematically evaluating investment decisions so as to allocate capital most efficiently to achieve investor's goals, views, and mandates.

The basic portfolio optimization model used for constructing portfolios by trading-off risk and return expectations is due to Markowitz [15]. Letting w denote an n-dimensional vector of normalized security weights in the target portfolio, Markowitz's *mean-variance model* minimizes the variance of the portfolio return for a given expected return:

$$\begin{array}{lrl} \text{min} & w'Qw & \text{(variance)} \\ \text{s.t.} & r'w = \rho & \text{(expected return)} \\ & 1'w = 1 & \text{(budget)} \end{array}$$

In this model Q is the $n \times n$ *covariance matrix* of security returns, r is the $n \times 1$ vector of *expected returns* for the securities, and ρ is the target expected return of the portfolio to be constructed. Here portfolio return variance is used as the measure of risk. By varying ρ, one can construct a trade-off curve between risk and expected return. The second constraint is referred to as the normalized *budget constraint*, as normalized weights sum up to one.

This basic model is instructive as it highlights the fundamental trade-off between risk and expected return and the benefits of portfolio diversification, and it lends itself to a simple closed-form solution. However, its practical utility is limited as it ignores many considerations that are important to portfolio managers and traders. When constructing and re-balancing portfolios a portfolio manager faces many constraints due to mandates and compliance rules. The manager may be interested not only in the total risk of a portfolio but also its exposures to several sources of risk, often relative to a benchmark, and may want to control other risk indicators such as portfolio leverage. Moreover, portfolios resulting from the simple mean-variance model may be far from being implementable due to excessive transaction costs, tax consequences, liquidity considerations, round-lot (minimum piece and increment) constraints or because the resulting portfolio contains too many securities having very small weights.

The Bloomberg Portfolio Optimizer is designed as an easy-to-use, yet powerful and flexible decision support system to address many of the needs of a modern portfolio manager for long-only as well as long-short global multi-asset class portfolios. It allows the user to specify a variety of goals, incorporate proprietary data such as custom factors, expected returns or alpha, constrain portfolio attributes such as active risk, exposures, leverage, and weights with respect to a benchmark potentially over custom and/or predefined nested classifications. It has mixed-integer optimization capabilities in order to limit the number of securities in the portfolio or trades to a small number; and it supports threshold constraints to avoid small positions in the portfolio or small trades (min piece), and to avoid odd lots (min increment). Bloomberg Portfolio Optimizer employs sophisticated preprocessing and branch-and-cut algorithms based on recent academic research on conic mixed-integer optimization [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] to produce high quality solutions fast.[1]

An illustration

A rational investor is interested in achieving the largest expected return for a given risk budget or, alternatively, minimizing her risk for a target expected return. As higher return is often (if not always) associated with higher risk, it is a non-trivial task to hit the right balance between risk and expected return. An optimization model can be used to quantify this trade-off and construct portfolios with a tailored risk-return relationship. In this section, we demonstrate the effect of practical considerations on feasibility and risk-return profiles.

For illustration we consider a portfolio of only four securities, and for simplicity we assume that the trading universe also consists of the same securities. The initial portfolio weights

[1] Non-convex mixed-integer portfolio optimization problems are intractable and, therefore, it is impossible to guarantee optimality for them at large scale. For instance, there are 2×10^{107} different ways of choosing 100 stocks out of 500, whereas the estimated number of atoms in the observable universe is only around 10^{82}! For such problems, the power of the optimization algorithms is crucial in producing near-optimal solutions within a reasonable compute time.

The models and algorithms behind the Bloomberg Portfolio Optimizer are continiously improved and rigourously tested on hundreds of portfolios. The current version of the optimizer supports mixed-integer portfolio problems with up to 12,000 securities.

and benchmark portfolio weights are listed in the following table.

Stock	Initial weight	Benchmark weight	Active weight
Johnson & Johnson	43.1%	28.9%	14.2%
JPMorgan Chase	31.8%	31.4%	0.4%
Dow Chemical	19.8%	7.0%	12.8%
AT&T	05.3%	32.7%	-27.4%

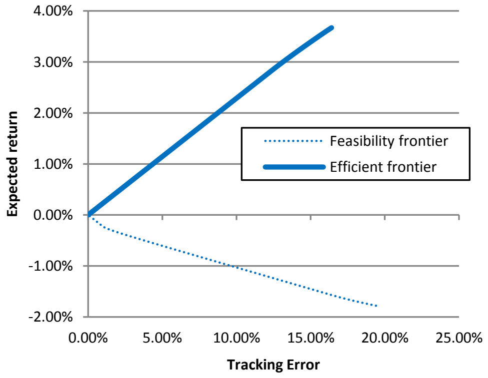
We are interested in constructing an active risk (benchmark tracking error) versus active expected return profile and exploring the effects of practical constraints. In order to do so, we start with the following long-only optimization model (LO):

$$\begin{aligned} \text{min} \quad & \sqrt{(w-w^b)'Q(w-w^b)} && \text{(active total risk)} \\ \text{s.t.} \quad & r'(w-w^b) = \rho && \text{(expected active return)} \\ & \sum_i w_i = 1 && \text{(budget)} \\ & w \geq 0 && \text{(long-only)} \end{aligned}$$

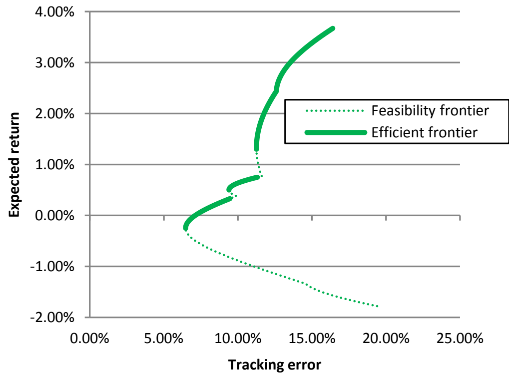
This formulation describes a model with the *goal* of constructing a portfolio with minimum active risk to the benchmark with weights w^b. The set of all possible portfolios under consideration form the *feasible set* of optimization and is determined by the *constraints* of the model. In this case, we specify that only those portfolios with positive security weights w (long only (LO) restriction) that add up to one (budget constraint) be considered. Moreover, the allowed portfolios must have an expected *active* return equal to ρ. Note that for some values of ρ there may be no portfolios satisfying all of the constraints simultaneously. In that case, the optimization model has no solution, and it is said to be *infeasible*.

By plotting the graph of minimum risk for all possible values of return we obtain a *feasibility frontier* (see Figure 1(a)). Each point on the feasibility frontier corresponds to a portfolio of minimum active risk for a given level of expected active return. However, a rational investor would prefer a higher expected return for a given risk. Eliminating the dominated points of the feasibility frontier, one is left with the *efficient frontier* in Figure 1(a). The efficient frontier has a positive slope by construction, reflecting the fact that one needs to take on riskier positions to attain a higher expected return.

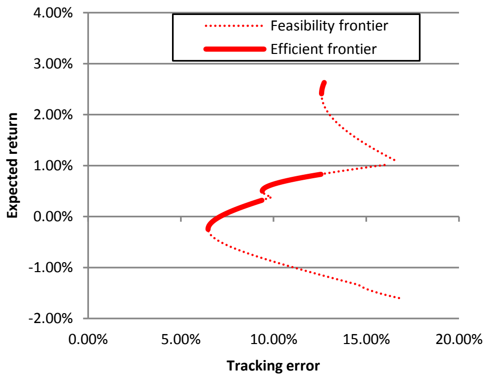
The shape of the efficient frontier depends on the constraints that define the collection of feasible portfolios. In order to illustrate the effect of imposing realistic constraints on efficient frontiers, we modified the LO model by restricting the maximum number (MaxNum) of positions, as well as a portfolio turnover (TO):



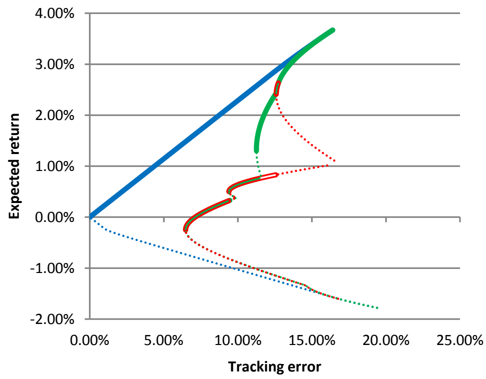
(a) Long only frontiers



(b) LO & MaxNum constrained frontiers



(c) LO, MaxNum & TO constrained frontiers

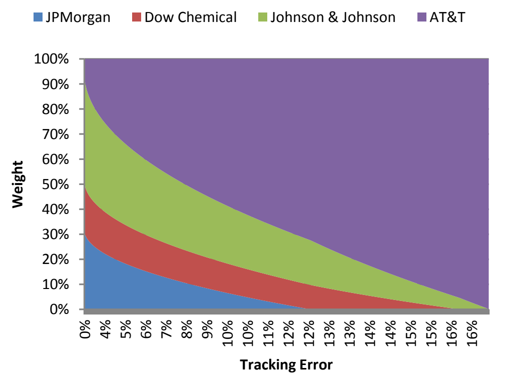


(d) Frontiers (a), (b), (c) overlayed

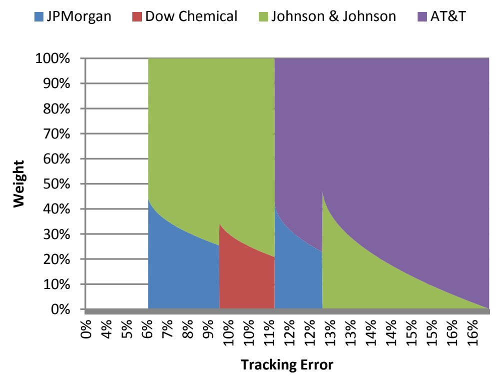
Figure 1: Efficient and feasibility frontiers for the portfolio optimization models LO, LO-MaxNum, LO-MaxNum-TO 2. The vertical axis specifies the expected active return in percent units and the horizontal axis the annualized active risk (tracking error). Dotted lines represent the feasibility frontiers (points to the left of those frontiers are infeasible), solid lines represent the efficient frontiers. Note that the feasibility frontiers contain all points of the efficient frontiers. Panel (d) shows, for comparison, all the other charts overlaid together.

$$
\begin{array}{lrll}
\min & \sqrt{(w-w^b)'Q(w-w^b)} & & \text{(active total risk)} \\
\text{s.t.} & r'(w-w^b) & = \rho & \text{(expected active return)} \\
 & \sum_i w_i & = 1 & \text{(budget)} \\
 & w & \geq 0 & \text{(long-only)} \\
 & \text{num}(w) & \leq \kappa & \text{(max number of positions)} \\
 & \sum_i |w_i - w_i^0| & \leq \tau & \text{(turnover)}
\end{array}
$$

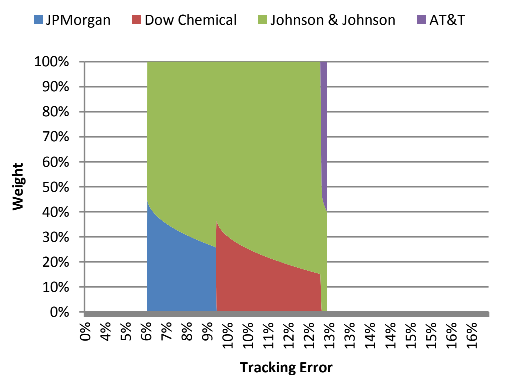
Here w^0 represents the initial portfolio weights. We set maximum number of positions to $\kappa = 2$ and turnover limit to $\tau = 110\%$ for constructing the frontiers below.



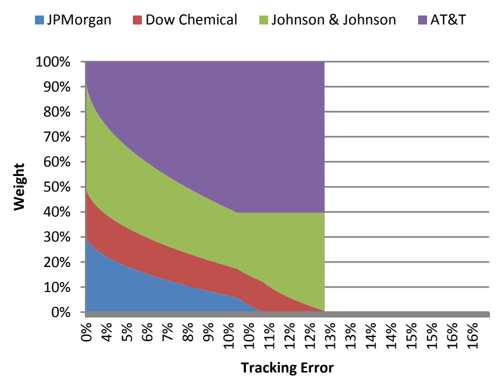
(a) LO portfolios



(b) LO & MaxNum constrained portfolios



(c) LO, MaxNum & TO constrained portfolios



(d) LO & TO constrained portfolios

Figure 2: Composition of the portfolios at the feasibility frontier. The vertical axis is the stacked weight of the portfolio securities. The horizontal axis is the annualized active risk. Panels (a), (b) and (c) show the portfolio weights corresponding to the frontiers in Figure 1. Panel (d), on the other hand, shows the portfolio weights on a frontier corresponding to a turnover-only constrained problem (not shown in Figure 1). Note that the low active risk portfolios are eliminated by restricting the number of positions to two. Similarly, large active risk portfolios are eliminated by the turnover restriction.

Results on the efficient and feasibility frontiers for the LO, LO-MaxNum and LO-MaxNum-TO models are displayed in Figure 1. Panel (a) shows the frontiers corresponding to the LO model. The bullet-like (concave) shape is characteristic of the classic Markowitz mean-variance optimization. Figure 1(b) shows the frontiers corresponding to the LO-MaxNum model. It is seen that restricting the number of positions in a portfolio may significantly alter the efficient frontier. As a new constraint is added, the frontier moves to the right indicating a degradation in the portfolio's efficiency (the minimum attainable active risk becomes larger). In particular, the zero active risk is unattainable as the benchmark portfolio is no longer feasible with the addition of the maximum number of positions constraint.

Another consequence of the number of positions constraint is that the efficient frontier is no longer concave, and is not even continuous. To understand this effect, consider all possible two-stock subsets of the trading universe. For each subset, one could solve the LO optimization model and obtain several concave frontiers. When the optimizer solves the number of positions constrained model, it picks the best two-securities subset satisfying the constraints for each value of expected return. Thus the LO-MaxNum efficient frontier is formed by non-dominated portfolios of the union of all two-security LO efficient frontiers.

Figure 1(c) illustrates how the turnover constraint further modifies the efficient frontier. The most striking difference between the LO-MaxNum-TO constrained and the LO-MaxNum constrained frontiers occurs at the expected return level of about 1% to 2% (active total risk of around 12% to 13%) where the LO-MaxNum-TO frontier is lower than the LO-MaxNum frontier by up to 1.5 percentage points in expected return. This is due to the fact that the corresponding two-stock subset of the trading universe is forbidden by the turnover constraint. Figure 2 shows the optimal portfolio weights for the LO, LO-MaxNum and LO-MaxNum-TO models as a function of active total risk (tracking error). The LO-MaxNum optimal portfolio for active total risk of 0.014 consists of only AT&T and JPMorgan stocks. Since the initial portfolio invests in all four stocks, in order to achieve the optimal portfolio Dow Chemical and Johnson & Johnson positions must be sold. However, that transaction would incur a turnover of 125.8% (see trading universe table), which is not allowed by the turnover constraint.

With a simple example, we have illustrated the construction of risk/return profiles and the effect of practical constraints on the profiles. In the following sections we will describe in more detail the different components of an optimization model and the various types of goals and constraints that can be set up within the Bloomberg Portfolio Optimizer.

3 Model Components

In order to set up the portfolio optimization process, one needs to select an initial portfolio and, optionally, a benchmark in PORT before launching the Bloomberg Portfolio Optimizer. In the optimizer *Setup* screen users may specify one or more trading universes, one or more goals, zero or more constraints at the portfolio or security group level, and security level properties and bounds.

In addition, users may optionally place trading restrictions on lists of securities, such as "No-trade," "No-buy" and "No-short" lists. Each of these components are explained in detail in the following subsections.

3.1 Portfolio

Typically the main use of portfolio optimization is to make adjustments to an existing portfolio in response to changing market conditions or investment strategies. Optimization suggests trades by taking into consideration updated risk, exposure, and return expectations as well as portfolio turnover and transactions costs involved in rebalancing.

Portfolio value (v_{port}) is the quantity used as the denominator for percentage weight reporting and it must be positive. Portfolio value of the target portfolio is always equal to the portfolio value of the initial portfolio. By default, the portfolio value is the net market value of the initial portfolio. For long-short or levered portfolios users can specify an alternative portfolio value such as the gross market value or a custom postive value. If a new portfolio needs to be constructed from scratch the initial portfolio must be set to *all cash* to be used as a portfolio budget.

If cash is infused to the portfolio, the initial portfolio value is updated by the infused amount before optimization starts. This will alter the initial weights. Cash infusion may be positive or negative.

3.2 Benchmark

The performance of a portfolio is often measured relative to a benchmark portfolio, such as the S&P 500 Index, Russell 3000 Value Index, or the Barclays Capital Aggregate Bond Index. Therefore, in addition to total risk and exposure, one is interested in measuring risk and exposures of the active portfolio, which is the difference between the portfolio and the benchmark. Bloomberg Portfolio Optimizer allows users to specify active risk (tracking error) and exposures in the goal as well as constraints to help portfolio managers control their active exposures and risk relative to a benchmark. Portfolio risk and active risk are computaed based on Bloomberg's multi-factor risk models.

3.3 Trade universe

The initial step of an optimization process is to select a universe of securities from which to build portfolios. Depending on the investment style the trade universe may be a broad index such as Russell 1000, a particular sector, or a small set of securities that is the outcome of fundamental research. Users can specify a trade universe in Bloomberg Portfolio Optimizer by selecting from a list of indices and custom portfolios. Alternatively, users can utilize Bloomberg's equity, fixed-income, or funds screens, EQS, SRCH, FSRC, to create custom trade universes. Prior to optimization it is advisable to perform an initial screening of the trade universe based on characteristics such as market cap, liquidity, price, yield, etc., in order to eliminate from consideration securities that do not match the desired criteria for inclusion. Doing so will lead to faster and better quality solutions. The trade universe always includes the initial portfolio and cash, which is denominated in the *portfolio reporting currency*. Before running an optimization task, users must choose a unique currency as the portfolio reporting currency.

3.4 Trade exclusion lists

When a large trade universe is selected, it is often necessary to eliminate securities from trade consideration due to compliance rules and applicable investment strategies and mandates. Bloomberg Portfolio Optimizer supports seven different types of exclusion lists in order to conveniently eliminate securities from trade consideration or portfolio positions.

- *No-buy list*: No additional position will be added to securities in the no-buy list.

- *No-sell list*: Positions of securities in a no-sell list will not be reduced.
- *No-trade list*: Positions of securities in a no-trade list will be kept constant.
- *No-long list*[2]: Securities in a no-long list can only be considered for short position.
- *No-short list*[2]: Securities in a no-short list can only be considered for long position.
- *No-hold list*: Securities in a no-hold list cannot be held either long or short. If they are in the initial portfolio, they will be liquidated.
- *Hedge list*: The hedge list serves as an override to the lists above. Any security in the hedge list is allowed to be traded, even if it is in one of the above trade exclusion lists.

Portfolio reporting currency is always included in the trade universe even if it is in a trade exclusion list above.

3.5 Optimization goal

The goal of the portfolio optimization model drives the optimization process. It refers to the specification of maximizing or minimizing a field of interest. For example, the portfolio manager may be interested in minimizing the portfolio active risk relative to a benchmark, or maximizing the expected return or yield to maturity.

A typical goal is to minimize the active risk of the portfolio. In this case one would select active portfolio risk as the goal `FIELD`, and specify that it should be minimized. If we denote the target portfolio weights by w and the benchmark weights by w^b, this goal field is expressed as

$$\text{min } \sqrt{(w - w^b)' Q (w - w^b)}.$$

Another example of a goal is to maximize the dividend yield of a portfolio. For this one would select the dividend yield as the goal `FIELD` and specify that it needs to be maximized. This goal may be expressed as follows

$$\text{max } \sum_i \text{divyld}_i w_i,$$

where divyld_i denotes the dividend yield of the ith security.

An optimization goal is expressed, in general, as

$$\text{max } \texttt{FIELD}$$

or

$$\text{min } \texttt{FIELD}$$

[2] Enabled only for long-short portfolios.

where FIELD refers to the term that we wish to maximize or minimize. One specifies the goal by selecting a FIELD from the field picker and specifying whether to maximize or minimize it. A list of fields that may be used in defining the goal is shown in Table 1. Certain fields have restrictions on how they may be used in the goal: e.g., active risk can only be minimized. Such restrictions, when they exist, are also noted in Table 1. We provide detailed definitions of the fields in Appendix A.

Multiple goals

The Bloomberg Portfolio Optimizer allows users to optimize multiple goals simultaneously, whereby one effectively finds an optimal trade-off between multiple and, often, conflicting goals. A typical example would be maximizing the expected return while minimizing the risk of the portfolio. In order to accomplish the simultaneous optimization, Bloomberg Portfolio Optimizer elicits from the user her indifference between the goals. To do so, the user specifies a *trade-off* for each goal, which is then used to scale each goal field appropriately. For instance, a trade-off of 2 for expected return and 3 for risk indicates that the user is willing to trade-off 2% of expected return for 3% of risk, in which case the optimizer maximizes a composite goal after normalizing the two goal fields by the respective trade-offs:

$$\max \quad \frac{\text{exp. return}}{2} - \frac{\text{risk}}{3}.$$

It is important to note that the unit of a trade-off is the same as the unit of its field. For instance, a risk field trade-off would be expressed in percentage, whereas a equity style exposure trade-off would be expressed in standard deviation. Adhering to consistent trade-off units leads to a normalized unitless utility as expressed in the example above: each 2% expected return has the same utility as each 3% risk. On the other hand, if the user is more risk averse and is indifferent between 2% expected return and only 1% risk, then the utility maximization changes to

$$\max \quad \frac{\text{exp. return}}{2} - \frac{\text{risk}}{1}.$$

giving twice as much weight to risk than expected return.

3.6 Constraints

In addition to the goal, a portfolio manager would typically want to specify a number of conditions that a portfolio must satisfy for it to be considered acceptable. Such conditions on feasible portfolios are called constraints. For example, the manager may wish impose a constraint that the long leverage of the target portfolio does not exceed 130%, that the total weight of technology stocks in the portfolio is within $\pm 10\%$ of the benchmark, that the turnover during the rebalance does not exceed 5%, or that the growth factor exposure of the portfolio is at least 0.3 standard deviations.

Constraints may also be placed on the whole portfolio as well as on groups of securities. For instance, a portfolio manager may be interested in limiting the amount invested in the energy sector to less than 10% of the total portfolio value, or may wish to allocate at least 30% of the portfolio to investment grade debt.

Bloomberg Portfolio Optimizer offers users numerous classification schemes, including GICS industries, country of domicile, Moody's credit ratings, etc., for grouping securities. Moreover, users may create their own custom (potentially nested) groups through Bloomberg's Unified Classsfications UNCL<GO>. Once a custom classfication is created, it becomes automatically available in the optimizer.

3.6.1 Generic constraints

A generic constraint accepted by Bloomberg Portfolio Optimizer is of the form

$$\text{minval} \quad \leq \; \texttt{FIELD} \; \leq \quad \text{maxval},$$

where `FIELD` refers to the quantity on which we wish to impose a condition, and minval and maxval are the lower and upper bounds on the field, respectively. It is not necessary to provide both lower and upper bounds: the user may specify only the lower bound or only the upper bound.

The quantities that may be used as fields to define constraints is the same as those used to define the goal, and are listed in Table 1. The table also notes certain restrictions on how the fields may be used to define constraints, e.g., one cannot specify a minval when active total risk is selected as `FIELD`. The reader is referred to Appendix A for detailed definitions of fields.

To define a constraint, one would specify a `FIELD`, the minval and/or maxval, and whether the constraint applies at the *portfolio level*, at the *group level* or at the *security level*.

Portfolio level constraints

Portfolio level constraints apply to all securities in the trading universe (U). A typical portfolio level constraint is the (two-way) turnover constraint

$$\sum_{i \in \mathsf{U}^*} |w_i - w_i^0| \leq \tau,$$

which specifies a upper bound τ on the sum of buy trades and sell trades by adding the absolute value of difference between the target portfolio (w) and the initial portfolio (w^0). To be precise, here U^* is slightly smaller than U as the turnover calculation does not include currency trades.

Group level constraints

Group level constraints can be used to set limits on the fields over one or more groups of securities. The user specifies a security group (G) and minimum and maximum values for the field over the group, e.g.:

$$\mathsf{minval}_\mathsf{G} \leq \sum_{i \in \mathsf{G}} \mathsf{YIELD}_i w_i \leq \mathsf{maxval}_\mathsf{G}.$$

For instance, the following constraint stipulates that the allocation to Asia Pacific stocks be at least 20% of the total portfolio value:

$$\sum_{i \in \mathsf{ASIAPAC}} w_i \geq 0.2.$$

3.6.2 Security level constraints

Security weight bounds: Security level weight bounds specify the minimum and maximum weight of any security:

$$\mathsf{minval}_i \leq w_i \leq \mathsf{maxval}_i, \ i \in \mathsf{U}.$$

The minimum and maximum values may be specified as absolute quantities or as relative to the initial portfolio or the benchmark.

An example of a security bound constraint is to limit individual security weights to a maximum short position of 5% and a maximum long position of 10%. To specify this constraint one would select minval $= -0.05$ and maxval $= 0.1$. For long-only portfolios the default values for minval and maxval are 0 and 1, respectively. For long-short portfolios, the default values for minval and maxval are -1 and 1, respectively.

Round position lot constraint: When this constraint is set, the number of shares held in each security is forced to be an integer multiple of a user-specified lot size, ℓ_i. In other words, the number of lots of the position for security i,

$$k_i = \frac{w_i v_{\text{port}}}{p_i \ell_i},$$

is constrained to be an integer number, where v_{port} is the portfolio value and p_i is the market price of the ith security. The integer number of lots k_i is a quantity that is determined by the optimizer.

Round trade lot constraint: Round trade lots refer to trade shares that are integer multiples of a fixed lot size. When this constraint is set, the number of shares bought or sold for each security will be an integer multiple of the user specified lot size, ℓ_i. That is,

$$k_i = \frac{(w_i - w_i^0) v_{\text{port}}}{p_i \ell_i}$$

is constrained to be an integer number.

Exclusion list conflicts. Sometimes securities in trade exclusion lists may lead to inconsistencies with initial portfolio positions and round lot constraints. For instance, if 150 shares of a security is held in the initial portfolio and the security is in a no-trade list, then it is not possible to satisfy position round lot of 100 shares for this security. Similarly, if the same security is in the no-hold list and a trade round lot of 100 shares is specified, it is not possible to liquidate it. Therefore, if the round lot constraint for a security is inconsistent with the initial portfolio position and a trade exclusion list, Bloomberg Portfolio Optimizer ignores the round lot constraint for that security and issues a warning.

3.6.3 Special constraints

Certain constraints cannot be expressed in the generic form described in Section 3.6.1, but are nevertheless essential to any practical portfolio optimization process. We list such constraints in this section.

Budget constraint: The budget constraint ensures that the market value of the optimal portfolio equals the market value of the initial portfolio. Because the optimizer uses notional weights, security weights are multiplied by the ratio of the market price to the notional price to convert notional exposures to market value exposures. Every optimization includes the budget constraint automatically.

$$\sum_{i \in \mathsf{U}} \frac{p_i^m}{p_i^n} w_i = \sum_{i \in \mathsf{U}} \frac{p_i^m}{p_i^n} w_i^0.$$

Number of positions: One of the direct consequences of minimizing risk or having a risk constraint is a well-diversified portfolio with nonzero exposure to almost all securities in the trade universe. While diversification is beneficial for risk reduction, a portfolio with thousands of securities may be hard-to-manage and undesirable. Therefore, Bloomberg Portfolio Optimizer allows users to restrict the number of securities in the optimal portfolio or in the trade list and avoid small nonzero positions or trades. Such constraints are non-convex and computationally challenging, and are typically referred to as mixed-integer constraints.

If an upper bound κ and a lower bound η on the number of (long or short) positions is specified, the optimal portfolio will have at most κ and at least η securities:

$$\eta \leq \mathsf{num}(w) \leq \kappa$$

Number of long/short positions: Users can set a minimum value and a maximum value on the number of long positions as well as short positions, separately.

Number of trades: Users may also restrict the number of (buy or sell) trades needed to get from the initial portfolio to the optimal portfolio. When an upper bound κ and a lower bound η is specified, the number of trades will be restricted to between η and κ:

$$\eta \leq \text{num}(w - w^0) \leq \kappa$$

Number of buy/sell trades: Users can set a minimum value and a maximum value on the number of buy trades as well as sell trades, separately.

Minimum (absolute) position size: In order to eliminate small nonzero positions, users can specify a threshold value, which is the minimum absolute value allowed for a security to be included in the portfolio. When specified the optimal portfolio excludes nonzero positions in the range $(-\tau, +\tau)$. A typical value for τ is 50 bps. Under this constraint, one of the following three conditions must be satisfied by each security weight w_i:

$$w_i = 0, \quad w_i \geq \tau, \text{ or } \ w_i \leq -\tau$$

Minimum (absolute) trade size: Users may aslo specify a threshold value on trades, which is the minimum absolute value allowed for a trade if a security is traded. Specifying a trade threshold of τ disallows nonzero trades in the range $(-\tau, +\tau)$: e.g., τ may be set to 10 bps to prohibit trades less than 10 bps.

$$w_i = w_i^0, \quad w_i \geq w_i^0 + \tau, \text{ or } \ w_i \leq w_i^0 - \tau$$

Exclusion list conflicts. Sometimes securities in trade exclusion lists may lead to inconsistencies with the initial portfolio positions and the minimum absolute size constraints. For instance, if a security has 25 bps weight in the initial portfolio and belongs to a no-trade list, then it is not possible to maintain a minimum position size of 50 bps for this security. Similarly, if the same security is in the no-hold list and a minimum trade size of 50 bps is specified, then it is not possible to liquidate it. Therefore, Bloomberg Portfolio Optimizer ignores the minimum absolute position or trade size constraint for a security if it is inconsistent with the initial portfolio position and a trade exclusion list.

3.6.4 Soft constraints

Portfolio managers are at times faced with situations when it is not entirely clear what minimum and maximum values to assign for the constraints. For instance, a PM controlling portfolio momentum exposure may want to limit the portfolio exposure to the momentum factor to be smaller than 0.2 standard deviations, but may be willing to tolerate a growth factor exposure of 0.25 standard deviations if that would help to find a feasible solution or if it would result in a significant improvement of the goal.

To handle such situations, Bloomberg Portfolio Optimizer allows the user "soften" any generic constraint (see Section 3.6.1). In contrast to the usual "hard" constraints, a soft constraint is allowed to be violated in a manner consistent with the user's tolerance to such violations and the impact the constraint has on the goal and other soft constraints. The use of soft constraints is particularly helpful in cases where there is no portfolio satisfying all (hard) constraints, and in cases where the goal must be severely compromised in order to satisfy a set of (hard) constraints.

In order to elicit the user's tolerance for violation for soft constraints, the user is asked to input her *trade-off unit* for each soft constraint. Any constraint with a positive trade-off is treated as a soft constraint and all other constraints are treated as hard constraints. For example, suppose that the user would like the optimal portfolio to have an exposure of 1 standard deviation (std) to the growth factor and a weighted average market cap of at least $500 MM, but is willing to tolerate some violation of these constraints. Accordingly, she specifies trade-off units of 0.05 std for the exposure constraint and $10 MM for the market cap constraint:

$$
\begin{aligned}
&1 \text{ std} \le \sum_i F_{ig} w_i \le 1 \text{ std} && [0.05 \text{ std}] \quad \text{(growth exposure)} \\
&\sum_i \text{MarketCap}_i w_i \ge \$500 \text{ MM} && [\$10 \text{ MM}] \quad \text{(market cap)}
\end{aligned}
$$

The trade-off units above are interpreted as the user is indifferent between 0.05 std of violation to the growth exposure constraint and $10 MM of violation to the market cap constraint. Bloomberg Portfolio Optimizer then defines quadratic penalties for violations of the soft constraints. The aggregate penalty of all soft constraint violations is simply the sum of the violation penalties for all soft constraints.

Finally, when soft constraints are used or when multiple goals are specified, the user also needs to enter a trade-off unit for each goal in the task. Bloomberg Portfolio Optimizer defines the utility of each goal field as the ratio of the field value to its trade-off unit. The soft constraint penalties are subtracted from the aggregate goal utility. As an example, consider an optimization task maximizing a goal with two soft constraints. Mathematically, the penalized utility for this task is expressed as

$$
\frac{\text{goal value}}{\text{goal trade-off}} - \left(\frac{\text{constraint\#1 violation}}{\text{constraint\#1 trade-off}}\right)^2 - \left(\frac{\text{constraint\#2 violation}}{\text{constraint\#2 trade-off}}\right)^2.
$$

Bloomberg Portfolio Optimizer finds a portfolio that maximizes the above penalized utility, subject to security bounds and hard constraints, if any.

Observe that, the smaller the trade-off unit, the more important the goal field or the soft constraint is, as the contribution to utility or penalty increases with a smaller denominator.

Table 1: The list of goal and constraint fields. For a detailed description, please see Appendix A.

Field Name	Use as Goal	Use as Constraint	Soft Constraint
Execution Fields			
Number of trades	Not allowed	MinVal, MaxVal	Not allowed
Number of buys	Not allowed	MinVal, MaxVal	Not allowed
Number of sells	Not allowed	MinVal, MaxVal	Not allowed
Number of positions	Not allowed	MinVal, MaxVal	Not allowed
Number of longs	Not allowed	MinVal, MaxVal	Not allowed
Number of shorts	Not allowed	MinVal, MaxVal	Not allowed
Position size (%)	Not allowed	MinVal	Not allowed
Trade size (%)	Not allowed	MinVal	Not allowed
TCA/LQA transaction cost	Minimize	MaxVal	Allowed
User-def linear trans. cost	Minimize	MaxVal	Allowed
User-def nonlinear trans. cost	Minimize	MaxVal	Allowed
Active share	Min. or Max.	MinVal, MaxVal	Allowed
Turnover	Minimize	MaxVal	Allowed
Long leverage	Not allowed	MinVal, MaxVal	Allowed
Short leverage	Not allowed	MinVal, MaxVal	Allowed
UCITS/RIC diversification rule	Not allowed	MaxVal	Not allowed
Risk Fields			
Portfolio total risk	Minimize	MaxVal	Allowed
Portfolio factor risk	Minimize	MaxVal	Allowed
Portfolio non-factor risk	Minimize	MaxVal	Allowed
Portfolio VaR	Minimize	MaxVal	Allowed
Portfolio CVaR	Minimize	MaxVal	Allowed
Active total risk	Minimize	MaxVal	Allowed
Active factor risk	Minimize	MaxVal	Allowed
Active non-factor risk	Minimize	MaxVal	Allowed
Active VaR	Minimize	MaxVal	Allowed
Active CVaR	Minimize	MaxVal	Allowed
Risk factor exposure	Min. or Max.	MinVal, MaxVal	Allowed
Beta to benchmark	Min. or Max.	MinVal, MaxVal	Allowed
Gross active weight	Min. or Max.	MinVal, MaxVal	Allowed
Sharpe/information ratio	Maximize	MinVal	Not allowed
Risk contribution	Minimize	MaxVal	Allowed
Cash-Flow Fields			
Cash flows	Not allowed	MinVal, MaxVal	Not allowed
Net present value	Min. or Max.	MinVal, MaxVal	Allowed
Characteristic Fields			
Any PORT characteristic	Min. or Max.	MinVal, MaxVal	Allowed
Security Level Data			
Expected return	Maximize	MinVal, MaxVal	Allowed
Bloomberg fields	Min. or Max.	MinVal, MaxVal	Allowed
User-defined fields	Min. or Max.	MinVal, MaxVal	Allowed
Weight			
Security group weights e.g., GICS sectors, countries	Not allowed	MinVal, MaxVal	Allowed
Individual security weights	Not allowed	MinVal, MaxVal	Not allowed

4 Case Studies

In this section we illustrate the use of Bloomberg Portfolio Optimizer in passive as well as active investment strategies. Under passive portfolio management, we cover the fundamental problem of tracking a large index with a portfolio of small number of securities and the construction of investable factor mimicking portfolios. Under active portfolio management, we show how the optimizer may be used in conjunction with an alpha model to construct long-short dollar-neutral growth portfolios. In both cases we study the historical performance of the weekly-rebalanced portfolios by backtesting over five years between 2006 and 2010.

4.1 Passive portfolio management

4.1.1 Index tracking

Motivated by the efficient capital market hypothesis, a passive investment strategy aims to replicate the performance of a benchmark index rather than outperform it. This can be done by "full replication," *i.e.,* investing in all members of the index at their index weights. However, such a strategy is usually far too expensive to implement due to the high cost of holding a large number of securities with very small weights, the cost of trading of relatively illiquid index members, and the costs of frequent transactions associated with index turnover and the flow of capital in and out of the fund. Therefore, fund managers often prefer to replicate an index with a fewer number of liquid securities that are cheaper to trade and unlikely to be delisted. These easier-to-manage, lower-cost portfolios naturally incur an index tracking error.

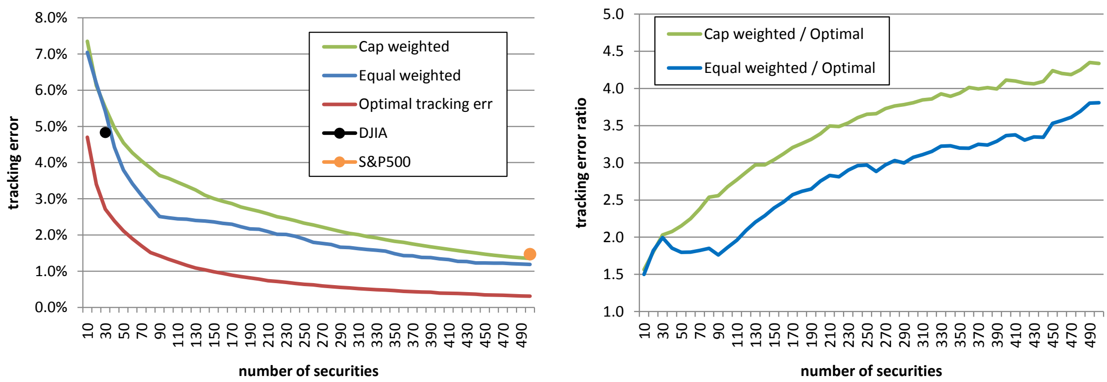
In the following suppose that we would like to replicate a capitalization-weighted benchmark. The main problem faced by an index tracking fund manager is to decide (1) which subset of securities to include in the portfolio and (2) what fraction of capital to invest in each of them so as to keep the tracking error low. As diversification reduces risk and one aims to replicate the performance of the benchmark, it is tempting to construct an index tracking portfolio by sorting securities in decreasing order of capitalization (benchmark weight) and including them in the portfolio in this order with the same relative weights as in the benchmark until a target tracking error is achieved. However, capitalization weighting can be far from optimal when tracking an index with a small number of securities. As shown in Figure 3 it can even be worse than equally weighting the securities.

Using the Russell 3000 Index as the benchmark, in Figure 3 we compare the forecast tracking error of three construction methods: capitalization weighted, equal weighted and optimization, for different number of securities included in the portfolio. The optimal portfolios are constructed with Bloomberg Portfolio Optimizer by minimizing the tracking error subject to budget and a maximum number of positions constraints[3]:

[3] A real implementation would include many other constraints, including bounds on sector weights, security weight, thresholds, etc. For simplicity of the discussion we have not included them in this example.

$$
\begin{array}{llll}
\text{min} & \sqrt{(w-w^b)'Q(w-w^b)} & & \text{(tracking error)} \\
\text{s.t.} & \sum_i w_i & = 1 & \text{(budget)} \\
& \text{num}(w) & \leq \kappa & \text{(max number of positions)}
\end{array}
$$

Unlike cap weighting, optimization does not use any particular ordering of the securities. For each value of κ it finds the optimal combination of κ securities that minimizes the tracking error. Observe that for each choice of κ optimization produces tracking error several times smaller than cap weighting and equal weighting.



(a) Tracking Error (b) Tracking Error Ratio

Figure 3: Performance comparison for the market-capitalization-weighted, equal-weighted and optimization methods. Tracking errors are annualized.

4.1.2 Investable factor mimicking portfolios

Bloomberg Portfolio Optimizer allows users to adjust their portfolio exposures in a way that more accurately reflects their market views and investment strategies. For instance, one may want to construct a portfolio tracking a benchmark index, and at the same time place a bet on a particular style factor, while keeping active exposures to the remaining factors to a minimum.

For example, suppose a portfolio manager is interesting in tracking Russell 3000 with a long-only portfolio of 150 securities and having a positive tilt in the *dividend yield style factor* only. Furthermore, assume that she likes to keep the turnover at each weekly rebalance is limited to 2.5%. This can be readily achieved with Bloomberg Portfolio Optimizer by solving the following optimization model:

$$
\begin{array}{rlll}
\text{min} & \sqrt{(w-w^b)'Q(w-w^b)} & & \text{(tracking error)} \\
\text{s.t.} & \sum_i f_{id}(w_i - w_i^b) & = 1 & \text{(dvd yield exposure)} \\
 & -0.1 \le \sum_i f_{ij}(w_i - w_i^b) & \le 0.1 & \text{(all other exposures } j\text{)} \\
 & \text{num}(w) & \le 150 & \text{(cardinality)} \\
 & \sum_i w_i & = 1 & \text{(budget)} \\
 & \sum_i |w_i - w_i^0| & \le 2.5\% & \text{(turnover)} \\
 & w & \ge 0 & \text{(long-only)}
\end{array}
$$

Here f_{id} denotes the dividend yield exposure for the i-th stock and f_{ij} represents the jth remaining exposure. We set the active dividend yield exposure to 1 standard deviation above the benchmark. All other exposures are kept within a band of ± 0.1 standard deviation around the benchmark. Note that a constraint has been added to restrict the portfolio turnover and portfolio weights are restricted to be positive. We will refer to this portfolio as a *dividend yield factor-mimicking* portfolio.

Backtesting is a standard method used to assess the performance of a given strategy over time. It applies the strategy to historical data as if it had actually been run in the past. In that way, the portfolio manager can evaluate how the strategy would have performed.

Here we backtest over a period of about five years - from Jan 04, 2006 to Sep 04, 2010. The dividend yield tilt strategy is applied by re-optimizing the portfolio on a weekly basis. The backtesting procedure generates historical portfolios with active exposure almost exclusively to the dividend yield factor. Note that as a result of the periodic rebalancing (carried out by running the optimizer on each selected date), the positions in the portfolio change throughout the backtesting period in response to changes in the covariance matrix and factor exposures. The amount of trading needed to change those positions is limited by the turnover constraint.

Figure 4 illustrates the time series of values corresponding to the factor-mimicking portfolio and the benchmark. The portfolio and the benchmark were scaled to have a market value of one million dollars at the beginning of the backtesting period.

In order to show how well the portfolio tracks the benchmark, we plot in Figure 5 the realized portfolio active returns against the factor-model projected tracking error (representing, by definition, one standard deviation active returns). It can be seen that the level of tracking error to the Russell 3000 Index projected for the factor-mimicking portfolio ranges from about 0.25% weekly (2% annual) in 2006 to about 0.75% weekly (6% annual) in 2009. To assess the quality of these projections, we define the scaled portfolio active returns as

$$\tilde{R}_i = R_i/\sigma_i,$$

where R_i is the portfolio active return on the i-th backtesting date and σ_i is the corresponding

Figure 4: Realized historical values of the dividend yield factor-mimicking portfolio and Russell 3000. The portfolio and the benchmark are scaled to be worth $1M at the beginning of the test.

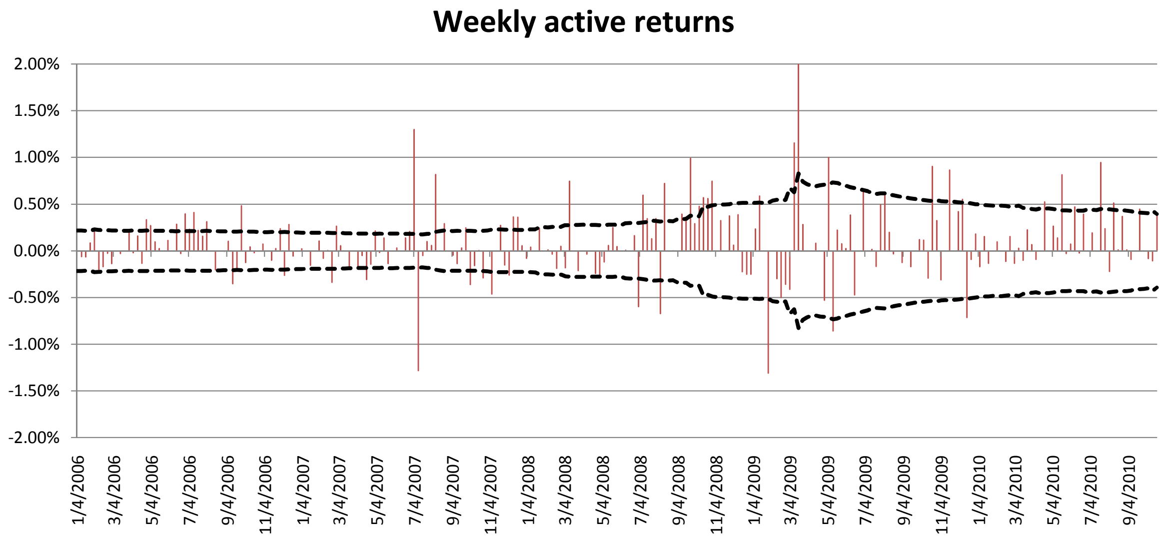


Figure 5: Realized weekly active returns of the factor-mimicking portfolio. The dotted line indicates the portfolio tracking error forecast (one standard deviation), which is minimized on each rebalance date.

tracking error projection. If the tracking error estimate is unbiased, the scaled returns must have close to unit volatility. In the backtesting experiment, the sample volatility was measured to be 1.17, signaling a slight under forecasting of active risk for the optimized portfolio, but still in a fairly good agreement with the expected value.

The optimization at each weekly rebalance date resets the dividend yield factor exposure to 1.0, and close to zero for the other factors. During the week after rebalance, those exposures change as the prices, and therefore the weights, change. Furthermore, individual securities' exposures vary as well. However, one would expect exposures to remain stable over a relatively short period of time. Figure 6 shows the factor mimicking-portfolio style exposures right before the optimization (for clarity, industry exposures are not shown). Note that, as expected, exposures do not drift much in-between the rebalance dates.

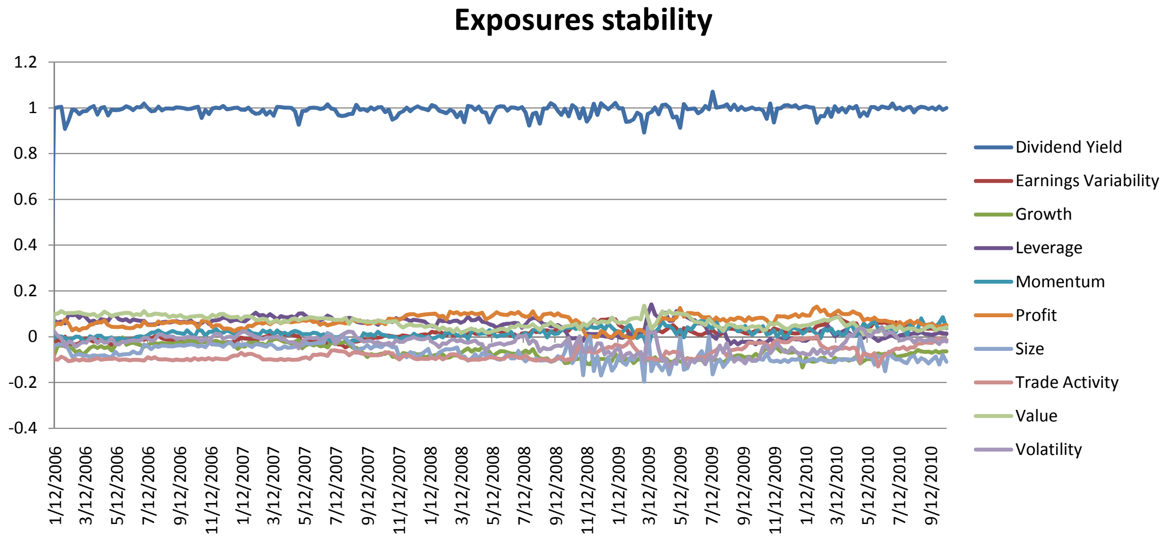


Figure 6: Portfolio exposures to style factors *before* optimization. Optimization resets the dividend yield exposure equal to 1.0 standard deviation above the benchmark, while keeping all other exposures in the range of ± 0.1 standard deviations around the benchmark. Drift from the reset values is caused by changing security prices and variations in individual securities' exposures in-between the rebalance dates.

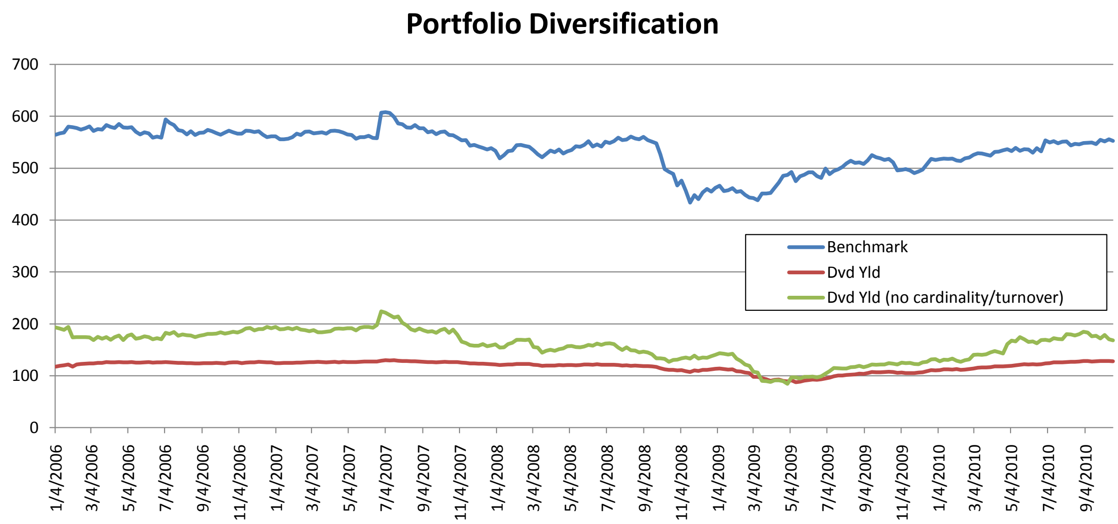


Figure 7: A measure of diversification $\mathrm{card}_{\mathrm{eff}}$ can be interpreted as an effective number of securities in the portfolio. For Russell 3000, it varies around 500 whereas for the dividend yield portfolio it ranges between 100 and 200.

A consequence of restricting the portfolio to be exposed to only one factor is a reduction in diversification with respect to the benchmark. The strategy is, in fact, tilting the portfolio by favoring higher dividend yield stocks. To quantify this effect for long-only portfolios, one can use entropy, a measure of diversification based on information theory:

$$\mathsf{card}_{\mathsf{eff}} = \exp\left(-\sum_i w_i \ln w_i \right)$$

where w_i is the weight of security i in the portfolio. This quantity represents a measure diversification. For an equally weighted portfolio, $\mathsf{card}_{\mathsf{eff}}$ equals to the number of securities in the portfolio, whereas for a portfolio with a single stock, $\mathsf{card}_{\mathsf{eff}}$ equals one. This diversification measure is shown in Figure 7. The chart shows the time series of the effective number of securities for the dividend yield-mimicking portfolio, the benchmark, and the cardinality-unconstrained portfolio from Figure 4. The dividend yield portfolio has a $\mathsf{card}_{\mathsf{eff}}$ ranging from 100 to 200. It is interesting to note that the dividend yield-mimicking portfolio without the cardinality requirement has a slightly higher level of diversification, suggesting that the concentration increase is mostly due to the exposure constraints and not the cardinality constraint.

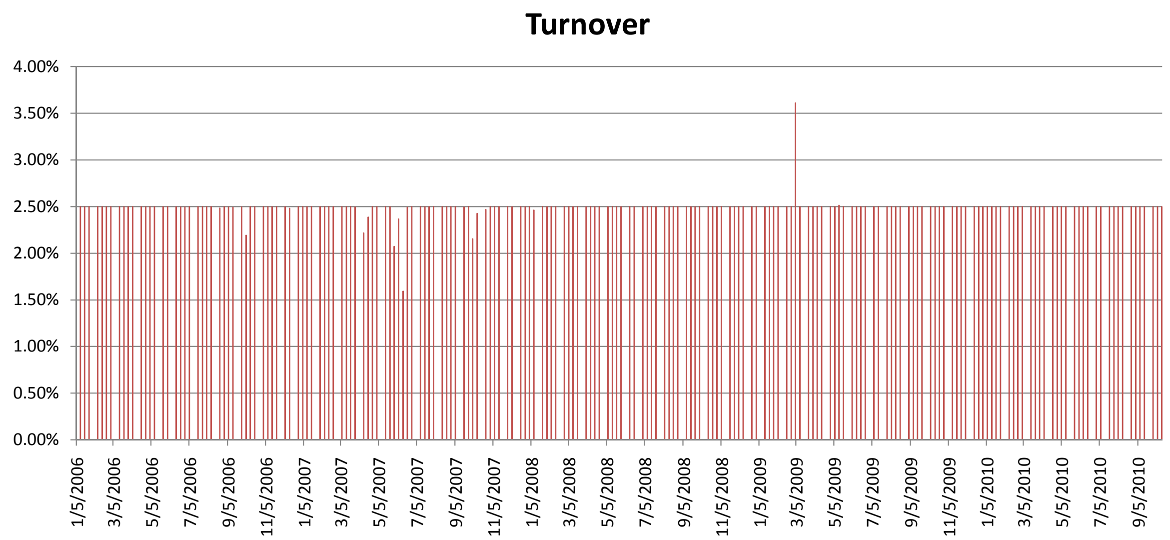


Figure 8: Weekly portfolio turnover over time. The turnover limit was set to 2.5% in the optimizer. On a couple of dates, the constraint was relaxed to avoid infeasibilities due to intrinsic variations in the individual securities' exposures.

It was observed that on some dates the turnover constraint was impossible to satisfy, rendering the optimization problem infeasible. This was possibly caused by intrinsic turnover due to changes in the securities' exposures. In those weeks, the turnover constraint was relaxed so that turnover takes on the minimum possible value that makes the optimization feasible. Figure 8 shows the actual portfolio turnover incurred by the optimization process. It can be seen that the limit of 2.5% is met in 99% of the weeks.

4.2 Active portfolio management

4.2.1 Alpha-driven long-short growth portfolios

In this section we illustrate the use of Bloomberg Portfolio Optimizer as an active portfolio rebalancing tool when expected future returns are forecast via an alpha model. We consider long-short, dollar-neutral strategies on the constituents of the Russell-3000 index for this illustration.

To emulate the effect of a true alpha model we use a *noisy look-ahead* alpha model, in which the alpha signal is a noisy observation of the true next-period excess return:

$$\alpha = R + \varepsilon$$
$$\varepsilon \sim N(0, Q/\gamma),$$

where α is the vector of alpha signals for all securities, R is the true (unknown) next-period excess return vector, and ε is the normally-distributed noise vector which is uncorrelated with the true return vector R. Note that, for simplicity, we set the noise covariance matrix to be a scaled version of the return covariance matrix, Q. The signal-to-noise-ratio γ, defined as the ratio of the return covariance matrix and the noise covariance matrix, quantifies the predictive power of the alpha model: the larger the signal-to-noise ratio γ, the smaller the noise and the more informative the alpha signal. The limiting case $\gamma \to 0$ corresponds to a completely uninformative alpha signal (it is purely noise), and the limiting case $\gamma \to \infty$ corresponds to perfect prediction of realized next-period excess returns.

We note that α by itself is not an expectation of the next-period excess return. We derive a model for the conditional expected excess return vector μ given the alpha signal by assuming that the unconditional distribution of the true next-period return is normal:

$$R \sim N(0, Q),$$

which gives

$$\mu = \mathbb{E}(R|\alpha) = \frac{\gamma}{1+\gamma}\alpha.$$

Using back-testing, we compare the historical performance of the three rebalancing strategies described in the sub-sections below, each of which uses the above alpha signal. For each strategy the entire available capital at the beginning of each time period is invested in cash, such that all long and short positions in stocks sum to zero. Denoting the cash weight by w_0 and weights in securities by w_i, $i = 1, \cdots, n$, where n is the number of securities in the Russell 3000 index at that time, implies

$$
\begin{aligned}
w_0 &= 1 \\
\sum_{i=1}^{n} w_i &= 0.
\end{aligned}
$$

Deciles, equal-weighted This strategy does not use the Bloomberg Portfolio Optimizer, and is the baseline strategy for comparison. At the beginning of each period we sort all n securities in the Russell-3000 index on that day by their conditional expected returns, μ. We then construct the portfolio for the next period by taking long positions with equal weights in all securities in the top decile and equal short positions in all securities in the bottom decile:

$$
w_i = \begin{cases} 10\ell/n & \text{if } i \in \mathcal{D}_1 \\ -10\ell/n & \text{if } i \in \mathcal{D}_{10} \\ 0 & \text{otherwise.} \end{cases}
$$

In the above, $\mathcal{D}_k$ is the set of all securities in the kth highest decile, and ℓ is a leverage parameter. Note that the *long size* or long leverage of the portfolio is given by

$$
\sum_{i=1}^{n} \mathsf{max}(w_i, 0) = \ell.
$$

For the purpose of this illustration we set $\ell = 1$, which means that the aggregate long positions and the aggregate short positions is each equal to the portfolio total value.

Deciles, optimal-weighted With this strategy, as with the baseline, we assign zero weight to all securities in the middle eight deciles after sorting by the conditional expected return. However, instead of using equal weights in the top and bottom deciles, we use optimal weights computed by the Bloomberg Portfolio Optimizer. The optimizer is configured to maximize the expected portfolio return, and the volatility and leverage of the portfolio are constrained to be no larger than the respective values from the baseline solution. The following is the optimization problem that we present to the Bloomberg Portfolio Optimizer (the vector w in the following refers to $[w_1 \ w_2 \ \cdots w_n]'$; it does not include cash):

$$
\begin{array}{llr}
\text{max} & \mu' w & \text{(expected return)} \\
\text{s.t.} & \sqrt{w'Qw} \le \sigma_{\text{max}} & \text{(volatility)} \\
& \sum_{i=1}^{n} \text{max}(w_i, 0) \le \ell & \text{(leverage)} \\
& w_0 + \sum_{i=1}^{n} w_i = 1 & \text{(budget)} \\
& w_0 = 1 & \text{(dollar-neutral)} \\
& -1 \le w_i \le 1, \quad i \in \mathcal{D}_1, \mathcal{D}_{10} & \text{(bounds)} \\
& w_i = 0 \quad i, \in \mathcal{D}_2, \cdots, \mathcal{D}_9. &
\end{array}
$$

where $\mathcal{D}_k$ denotes the kth decile, and the upper-bounds on volatility and leverage, denoted respectively by σ_{max} and ℓ, are the volatility and leverage of the baseline (equal-weighted) strategy described above.

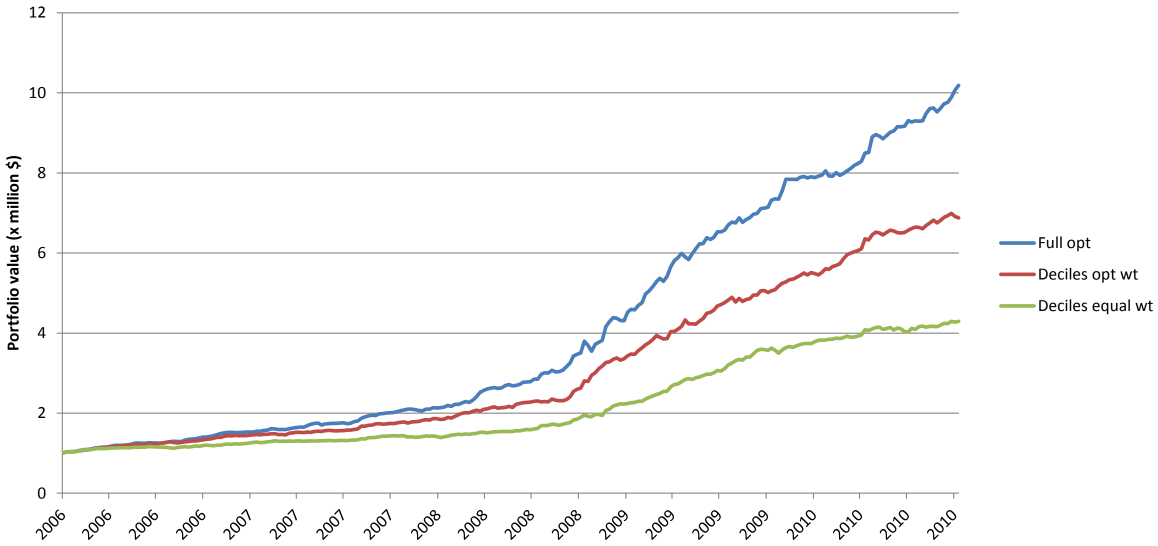
Full optimal This strategy is similar to the optimal weights strategy above, except that we relax the security bounds to allow non-zero positions to be taken in all n securities. The optimization problem formulation for this strategy is the same as the one specified above, with the security bounds replaced by the following:

$$
-1 \le w_i \le 1 \quad \text{for all } i \ne 0. \qquad \text{(bounds)}
$$

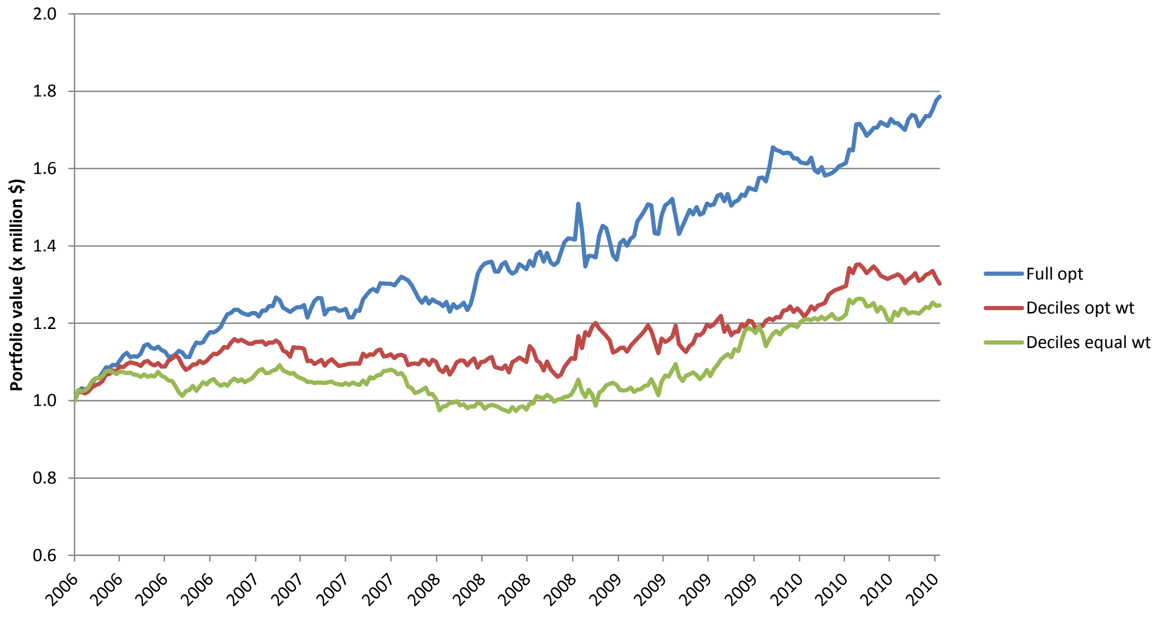
Figure 9 shows the portfolio value over time from back-testing the above three strategies from the beginning of 2006 to October 2010. Figure 9(a) corresponds to a signal-to-noise ratio of $\gamma = 1/50$, which means the noise covariance matrix is 50 times the return covariance matrix. In this scenario the baseline strategy has a compounded annual return of 35.6%, and those of the deciles optimal and full optimal strategies are 49.6% and 62.4%, respectively.

Figure 9(b) corresponds to a signal-to-noise ratio of $\gamma = 1/1000$, which means the noise covariance matrix is 1000 times the return covariance matrix. This represents the case when the alpha signal is very weak. In this scenario the baseline strategy has a compounded annual return of 4.7%, and those of the deciles optimal and full optimal strategies are 5.7% and 12.9%, respectively.

The above results show that substantial performance gains may be obtained by using the Bloomberg Portfolio Optimizer for alpha-driven active portfolio management. The benefit due to optimizing the weights of securities in the top and bottom deciles is significant when the alpha signal is strong, but not as significant when the alpha signal is weak. The performance improvement obtained by optimizing the weights over the entire security universe, on the other hand, is substantial in both cases. This additional gain in this case, however, comes at the expense of holding a larger portfolio.



(a) Alpha model with noise covariance = 50× return covariance ($\gamma = \frac{1}{50}$)



(b) Alpha model with noise covariance = 1000× return covariance ($\gamma = \frac{1}{1000}$)

Figure 9: Realized historical values of long-short dollar-neutral portfolios over the backtest period.

A Optimization Fields

In this appendix we provide a list of optimization fields that can be used to specify the optimization goal and constraints. We also specify any restrictions that may apply in the use of a field: e.g., we cannot impose a lower-bound on active risk. We use the following notation in our definitions of the fields.

w^0 : vector of initial portfolio weights
w^b : vector of benchmark portfolio weights
w : vector of target portfolio weights

Portfolio weights are computed by dividing the basis (notional) value of the securities by the *portfolio value* (v_{port}).

A.1 Execution fields

This category encompasses constraints on the number of positions, trades, buys, sells, longs and shorts; long size, short size, constraints on the smallest non-zero position or trade; and other trade execution-related characteristics such as turnover and transaction costs. Please see Table 1 for a summary of their usage in goals and constraints.

Number of positions

The number of non-zero (long or short) positions in the final (optimized) portfolio.

Number of longs

The number of non-zero long positions in the final portfolio. (Applies only to long/short optimization.)

Number of shorts

The number of non-zero short positions in the final portfolio. (Applies only to long/short optimization.)

Number of trades

The number of non-zero (buy or sell) trades between the initial and final (optimized) portfolios.

Number of buys

The number of non-zero buy trades.

Number of sells

The number of non-zero sell trades.

Minimum position size

The absolute smallest non-zero position weight in the final portfolio:

$$\text{min}_{i \in \mathsf{P}} |w_i| : w_i \neq 0$$

where P is the set of non-zero position weights.

Minimum trade size

The absolute smallest non-zero trade weight:

$$\text{min}_{i \in \mathsf{T}} |w_i - w_i^0| : w_i \neq w_i^0$$

where T is the set of non-zero trades.

Long leverage (size)

Long leverage (size) refers to the sum of long weights in the target portfolio excluding cash assets:

$$\sum_{i \in \mathsf{U} \setminus C} \text{max}(w_i, 0)$$

where $\mathsf{U} \setminus C$ is the non-cash assets in the trade universe.

Short leverage (size)

Short leverage (size) refers to the sum of absolute value of the short weights in the target portfolio excluding cash assets:

$$\sum_{i \in \mathsf{U} \setminus C} |\text{min}(w_i, 0)|$$

where $\mathsf{U} \setminus C$ is the non-cash assets in the trade universe. Note that the absolute value is taken so that this field is always nonnegative.

Cash leverage (size)

Cash leverage (size) refers to the sum of cash positions, including currencies, margins, and FX forwards, in the target portfolio:

$$\sum_{i \in \mathsf{C}} w_i$$

where C is the cash assets in the trade universe.

Active share

The active share is defined as half of the sum of absolute deviations from the benchmark weights:

$$0.5 \times \sum_{i \in \mathsf{U}} |w_i - w_i^b|$$

where U is the trade universe.

Turnover

The turnover is defined as the sum of non-cash buys and sells required to obtain the optimal portfolio:

$$\sum_{i \in \mathsf{U} \setminus \mathsf{C}} |w_i - w_i^0|$$

where U is the trade universe and C is the set of cash assets.

Transaction cost

The transaction cost field equals the sum of the transaction cost over all non-cash securities. Cash securities are assumed to have negligible trading cost. Bloomberg Portfolio Optimizer supports four types of transaction cost functions. The first is the user-defined linear cost on buy and sell trades; the second one is the user-defined nonlinear transaction cost model, which incorporates bid-ask spread, volatility, and average daily volume of the securities using internal data fields; the third one is the proprietary Bloomberg equity transaction cost model (TCA), which estimates temporary market impact and permanent market impact for each exchange separately as a function of a participation rate; whereas the fourth one is the proprietary Bloomberg fixed-income transaction cost model (LQA), which estimates the liqudity cost of as a function of trade size.

User-defined linear transaction cost

The user-defined linear transaction cost is linear in buy and sell trades and is computed by simply multiplying the trades by the user-defined buy cost and sell cost, depending on the side of the trade:

$$\sum_{i \in \mathsf{U} \setminus \mathsf{C}} \left(\mathsf{buycost}_i \cdot \mathsf{max}(w_i - w_i^0, 0) + \mathsf{sellcost}_i \cdot \mathsf{max}(w_i^0 - w_i, 0) \right)$$

where U \ C is the set of non-cash securities. Coefficients $\mathsf{buycost}_i$ and $\mathsf{sellcost}_i$ are specified for each security by the user.

User-defined nonlinear transaction cost

User-defined nonlinear transaction cost field is available for equities only. This field incorporates the bid-ask spread as well as volatility and liquidity of the securities in order to estimate market impact cost of the trades. The total transaction cost is the sum of transaction cost estimate for each non-cash security:

$$\sum_{i \in \mathsf{U} \setminus \mathsf{C}} \mathsf{UD\text{-}NLTC}(t_i, v_{\text{port}}),$$

where U \ C is the set of non-cash securities, t_i is the trade weight, $w_i - w_i^0$, for security i, v_{port} is the portfolio value, and

$$\mathsf{UD\text{-}NLTC}(t_i, v_{\text{port}}) = \left(\alpha_i \cdot \mathsf{spread}_i + \beta_i \cdot \mathsf{vol}_i \cdot \sqrt{\frac{v_{\text{port}} \cdot |t_i| / \mathsf{price}_i}{\mathsf{ADV}_i}} \right) \cdot |t_i| \cdot$$

The functional form of the user-defined nonlinear transaction cost model above is the standard one used in the industry for estimating the trade market impact cost (see e.g. Grinold & Kahn [14]). Here the first term is α_i times the bid-ask spread, spread_i, whereas the second term is the market impact cost estimate. The *unit* market impact cost estimate is a constant β_i times 90-day volatility vol_i of the security and the square root of the number of shares traded as a fraction of 30-day average daily volume, adv_i . Thus the total transaction cost is a convex, increasing function of the absolute value of the trade. For speed and accuracy, Bloomberg uses an exact second-order conic reformulation [1] of the nonlinear transaction cost, rather than a piece-wise linear approximation for optimization. By default, $\alpha_i = 0.5$ and $\beta_i = 0.032$ for each security i. Users may update α_i and β_i per security if their transaction cost estimates differ from the ones computed with these default values.

If a data field necessary for computing the UD-NLTC function is missing for a security, Bloomberg Portfolio Optimizer substitutes UD-NLTC for that security with a linear cost function computed with a user-defined default unit cost.

Bloomberg equity transaction cost (TCA)

The Bloomberg equity transaction cost (TCA) field is available for equities. It incorporates the bid-ask spread, volatility, liquidity, as well as the participation rate per security in order to estimate market impact cost of the trades. The total transaction cost is the sum of Bloomberg transaction cost estimate for each non-cash security:

$$\sum_{i \in \mathsf{U} \setminus \mathsf{C}} \mathsf{TCA}(t_i, v_{\text{port}}),$$

where $\mathsf{U} \setminus \mathsf{C}$ is the set of non-cash securities, t_i is the trade weight, $w_i - w_i^0$, for security i, v_{port} is the portfolio value, and

$$\mathsf{TCA}(t_i, v_{\text{port}}) = \left(\lambda_i \cdot \mathsf{spread}_i + \alpha_i \cdot \mathsf{vol}_i \cdot \mathsf{part}_i^{\beta_{i1}} \cdot \mathsf{dur}_i^{\beta_{i2}} + \gamma_i \cdot \mathsf{vol}_i \left(\frac{v_{\text{port}}|t_i|/\mathsf{price}_i}{\mathsf{adv}_i}\right)^{\eta_i}\right) \cdot |t_i| \cdot$$

Bloomberg's TCA transaction cost model is described in detail in Rashkovich and Verma [17]. Here the first term caputures the bid-ask spread cost, the second tem is the tremporary market impact cost estimate, whereas the third term is the permananet market impact estimate. The temporary impact is a function of the participation rate part_i, which the user may specify per security (the default value is 10%). dur_i stands for the duration of the trade and for the purpose of optimization, it is approximated as

$$\mathsf{dur}_i = \frac{v_{\text{port}} \cdot |t_i|/\mathsf{price}_i}{\mathsf{adv}_i \cdot \mathsf{part}_i}$$

Note that the numerator equals the number of shares traded. The coefficients $\alpha_i, \beta_{i1}, \beta_{i2}, \gamma_i, \eta_i$ are derived from nonlinear regressions performed using Bloomberg's trading data for each exchage. The total transaction cost is a convex, increasing function of the absolute value of the trade. For speed and accuracy, Bloomberg uses a second-order conic reformulation [1] of the nonlinear transaction cost, rather than a piece-wise linear approximation for optimization.

If a data field necessary for computing the TCA function is missing for a security, Bloomberg Portfolio Optimizer substitutes TCA for that security with a linear cost function computed with a user-defined default unit cost.

Bloomberg fixed-income transaction cost (LQA)

The Bloomberg fixed-income transaction cost (LQA) field is available for government, securitized, agency, and corporate (GSAC) bonds in the optimizer. It is a percentage liquidity cost estimate as a function of the trade size. The total transaction cost for the portfolio is the sum of transaction cost estimate for each non-cash security:

$$\sum_{i \in \mathsf{U} \setminus \mathsf{C}} \mathsf{LQA}(t_i, v_{\text{port}}),$$

where $\mathsf{U} \setminus \mathsf{C}$ is the set of non-cash securities, t_i is the trade weight, $w_i - w_i^0$, for security i, v_{port} is the portfolio value, and

$$\mathsf{LQA}(t_i, v_{\text{port}}) = \left(\alpha_i \cdot \left(\frac{v_{\text{port}}|t_i|/\mathsf{price}_i}{\mathsf{adv}_i}\right)^{\beta_i} \cdot \mathsf{TH}^{\gamma_i}\right) \cdot |t_i|,$$

where adv_i is the available daily volume for security i and TH is the trade horizon. The default value for trade horizon is one day. The users can override the trade horizon to speed up or slow down the trade execusion. The total transaction cost is a convex, increasing function of the absolute value of the trade. For speed and accuracy, Bloomberg uses a second-order conic reformulation [1] of the nonlinear transaction cost, rather than a piece-wise linear approximation for optimization. The details of the Bloomberg's proprietary liquidity cost model (LQA) are described on the Help Page of the the LQA<GO> function.

If a data field necessary for computing the LQA function is missing for a security, Bloomberg Portfolio Optimizer substitutes LQA for that security with a linear cost function computed with a user-defined default unit cost.

UCITS/RIC issuer sum

Bloomberg Portfolio Optimizer supports diversification rules of the European Directive on UCITS (Undertakings for Collective Investment in Transferable Securities). The UCITS 5/10/40 rule states that issuers exceeding 5% (threshold) weight cannot make up more than 40% (max weight) of a portfolio. A separate per issuer min/max weight constraint is also supported. In the optimizer, threshold, max weight as well as the per issuer min/max weight parameters are configurable so that the users may apply the rule more or less conservatively than the default values. Indeed, a similar Regulated Investment Company (RIC) 5/25/50 diversification rule of the United States Internal Revenue Service can be implemented by adjusting the diversification parameters appropriately.

Let w_i be the weight of issuer $i \in \mathsf{ISS}$. Then the UCITS/RIC diversification constraint limits the sum of all issuer weights higher than threshold τ to a maximum value κ:

$$\sum_{i \in \mathsf{ISS}: w_i > \tau} w_i \leq \kappa$$

UCITS/RIC diversification rule and threshold τ value can be set from the Bloomberg Portfolio Optimizer settings menu.

A.2 Risk fields

Users may use most risk fields in the context of a goal or a constraint; see Table 1 for a summary of the usage of risk fields. Except for factor exposures and beta to benchmark, it is not allowed to maximize a risk field or to set a lower bound on it.

Portfolio factor risk

The portfolio total risk (systematic volatility) refers to the annualized standard deviation of the return explained by the common risk factors for the target portfolio:

$$\sqrt{w'FCF'w}$$

Portfolio non-factor risk

The portfolio non-factor risk (residual volatility) refers to the annualized standard deviation of the return that is *not* explained by the common risk factors for the target portfolio:

$$\sqrt{w'Dw}$$

Portfolio total risk

Portfolio total risk (portfolio volatility) refers to the annualized standard deviation of the return of the target portfolio:

$$\sqrt{w'Qw},$$

where $Q = FCF' + D$.

Active factor risk

The active factor risk (systematic tracking error) refers to the annualized standard deviation of the active portion of the target portfolio explained by the common risk factors:

$$\sqrt{(w - w^b)'FCF'(w - w^b)}$$

Active non-factor risk

The active non-factor risk (residual tracking error) refers to the annualized standard deviation of the active portion of the target portfolio that is *not* explained by the common risk factors:

$$\sqrt{(w - w^b)'D(w - w^b)}$$

Active total risk

Active total risk (tracking error) refers to the annualized standard deviation of the return of the active portion of the target portfolio with respect to a benchmark portfolio:

$$\sqrt{(w - w^b)'Q(w - w_b)},$$

where $Q = FCF' + D$.

Portfolio VaR

The value-at-risk (VaR) implementation in the optimizer uses the parametric value-at-risk method assuming a jointly normal distribution of security returns. VaR horizon and confidence level are specified in PORT. At confidence level α%, portfolio losses higher than α%-VaR occurs 100-α % of the time.

Portfolio CVaR

The portfolio conditional value-at-risk (CVaR) implementation in the optimizer uses the parametric conditional value-at-risk method assuming a jointly normal distribution of security returns. α%-CVaR is the expectation of the portfolio losses higher than α%-VaR, where α is the confidence level.

Active VaR

The active value-at-risk is the parametric value-at-risk of the portfolio relative to the benchmark.

Active CVaR

The active conditional value-at-risk is the parametric conditional value-at-risk of the portfolio relative to the benchmark.

Factor risk exposures

The exposure to the j-th factor is the weighted average of the security exposures to the j-th factor:

$$\sum_{i \in U} f_{ij} w_i$$

Beta-to-Benchmark

Beta-to-Benchmark refers to the exposure of the portfolio to the benchmark. It is computed from the factor model as

$$\frac{(w^b)' Q w}{(w^b)' Q w^b}$$

Gross active weight

Gross active weight (also referred to as active share) is defined as the sum of absolute weight differences from the benchmark:

$$\sum_{i \in \mathsf{U}} |w_i - w_i^b|$$

where U is the trade universe.

Sharpe ratio

Sharpe ratio of the portfolio is a measure of risk adjusted expected return. It is simply the expected return of the portfolio divided by the ex-ante volatility of the portfolio:

$$\frac{r'w}{\sqrt{w'Qw}}$$

Bloomberg portfolio optimizer assumes that users input annualized expected return (r) net of risk-free return as percentage.

Information ratio

Information ratio of the portfolio is a measure of risk adjusted expected return relative to the benchmark. It is the expected return of the active portfolio divided by the ex-ante tracking error (active total risk) of the portfolio:

$$\frac{r'(w - w^b)}{\sqrt{(w - w^b)'Q(w - w^b)}}$$

Bloomberg portfolio optimizer expects that users input annualized expected return (r) net of risk-free return as percentage.

Risk parity and contribution to total risk

Bloomberg Portfolio Optimizer supports the construction of risk-parity portfolios and, more generally, portfolios with custom risk contribution targets. Given the portfolio weights w, using Euler's formula, the portfolio's total risk, $\sigma(w)$, can be decomposed into contribution from each security, $\sigma_i(w)$, as follows:

$$\sigma(w) = \sqrt{w'Qw} = \sum_{i=1}^{n} \sigma_i(w) = \sum_{i=1}^{n} w_i \frac{\partial \sigma(w)}{\partial w_i}.$$

The risk contribution of security i is its weight w_i times its marginal risk, i.e., the partial derivative of the risk with respect to i. In a risk-parity portfolio, each security (with the exception of the risk-free cash) has equal contribution to the risk:

$$\sigma_i(w) = \sigma_j(w) \text{ for all } i, j.$$

Setting the relative risk contribution to $1/(n-1)$ ensures that each risk security contributes to the portfolio total risk equally:

$$rrc_i(w) = \sigma_i(w)/\sigma(w) = 1/(n-1).$$

The optimizer automatically scales the relative risk contributions entered by the user so that the sum, excluding of the risk-free cash, is always 100%. Therefore, as long as the same (positive) number is entered as the relative risk contribution for each security, risk parity will be achieved without the need to pay attention to the number of securities in the universe. This feature is particularly convenient when backtesting risk-parity strategies as the size of the universe may change at each rebalance date.

It is important to note that there is a unique (long-only) risk-parity portfolio and it will be returned by the optimizer if the optimization task has no constraints. In the presence of constraints, the risk-parity portfolio may not be feasible. In that case, the optimizer returns a portfolio that (a) satisfies the constraints, and (b) has the smallest tracking error to the risk-parity portfolio. Therefore, the contribution to total risk displayed is the tracking error to the risk-parity portfolio, which will be zero if a risk-parity portfolio is returned.

It is also possible to construct portfolios with varying, custom relative risk contribution targets — risk parity being just a special case with equal risk contributions. For instance, if the user enters 0.5 for security ABC and 1 for all other securities, then security ABC will contribute to the risk half of the contribution of every other security in the portfolio. The treatment of constraints is similar: if the risk contribution target portfolio is infeasible with the constraints, then a feasible portfolio with the smallest tracking error to it is returned by the optimizer.

Risk parity and risk contribution targeting are not supported for long-short portfolios.

A.3 Cash-flow fields

Bloomberg Portfolio Optimizer supports cash-flow matching for asset-liability management (ALM). Users can specify min and/or max values on *the cash flows net of benchmark liabilities, interest payments, and re-investments* over a sequence of time periods. The sequence of time periods are specified by a start date, an end date and a periodicity, which can be daily, monthly, quarterly, semi-annual, or annual.

Cash flows

Without borrowing: The net cash flow in time period t equals the sum of cash flows of the portfolio in time period t ($\mathsf{CF}_t'w$), minus the liability in period t (ℓ_t), plus the amount re-invested in period $t-1$ and its one-period interest at the re-investment rate ρ ($i_{t-1}+\rho i_{t-1}$), and minus the excess amount re-invested in period t (i_t):

$$\mathsf{CF}_t'w - \ell_t + (1+\rho)i_{t-1} - i_t$$

With borrowing: If borrowing is allowed, the the net cash-flow term is extended with the amount borrowed in time period t (b_t), minus the payment of the borrowed amount in period $t-1$ and its one-period interest at the borrow rate β ($b_{t-1}+\beta b_{t-1}$):

$$\mathsf{CF}_t'w - \ell_t + (1+\rho)i_{t-1} - i_t + b_t - (1+\beta)b_{t-1}$$

Users are expected to enter re-investment and borrow rates per annum. The optimizer automatically adjusts the rates to reflect the periodicity of the cash-flow constraint. For example, if the cash-flow constraint periodicity is monthly, then monthly rates are applied in the calculations above. A cash security is assumed to have a cash flow equal to its value in the first time period and no cash flow in the subsequent time periods. This assumption allows the cash available in the portfolio to be applied toward paying the liabilities.

Net present value

The net present value field represents the net present value of the cash left on-hand in the last period n (i_n), discounted at the re-investment rate (ρ):

$$\sum_{s\in \mathsf{C}} p_s w_s + \frac{1}{(1+\rho)^n} i_n,$$

Note that the interest is compounded at the cash-flow constraint periodicity. Therefore, a cash-flow constraint must be specified to use the net present value field.

In order to construct a least-cost dedication portfolio to meet a stream of liabilities, users should maximize the net present value subject to a minimum cash-flow constraint relative to the liability benchmark.

A.4 Characteristic fields

Bloomberg Portfolio Optimizer supports numeric characteristic fields available on PORT<GO> Characteristics tab. In addition to these Bloomberg fields, users may upload custom fields, such as expected returns or proprierty factors, and use them as goals or constraints. Custom fields can be uploaded into the Bloomberg terminal using the BBU<GO> function as CDE<GO> fields or dropped from an Excel sheet directly into Section 4 of the optimizer setup screen. By default, characteristic values at the portfolio level are calculated as the weighted average of the security characteristic values:

$$\alpha_{\mathsf{P}} = \sum_{i \in \mathsf{P}} \alpha_i w_i$$

where α_{P} is the portfolio-level characteristic value and α_i is the value of the characteristic for the ith security. Other aggregation methods are described in the next section.

Characteristic fields can be used to define either a maximization or minimization goal. Users can also set maximum and/or minimum values on any characteristic field as constraints. Treatment of missing (NA) values: For securities with missing characteristics values, the NA override value specified for the field in PORT<GO> Characteristic tab is applied. If no such default is specified in the PORT<GO> Characteristic tab, then the default value specified for the field in the optimization setup screen is applied. In order to match PORT<GO> field aggregation values in the optimizer, it is important to specify NA override values and utilize consistent aggregation methods in the PORT<GO> Characterstic tab and the optimizer.

A.5 Field aggregation

A characteristic field or risk exposure field may be aggregated using net, gross, long, or short weights. Given a field α and a group of securities G, the following aggregation methods are supported:

Contribution:

$$\sum_{i \in \mathsf{G}} \alpha_i w_i$$

Gross value:

$$\sum_{i \in \mathsf{G}} \alpha_i |w_i| \Big/ \sum_{i \in \mathsf{G}} |w_i|$$

Long value:

$$\sum_{i \in \mathsf{G}} \alpha_i \max(w_i, 0) \Big/ \sum_{i \in \mathsf{G}} \max(w_i, 0)$$

Short value:

$$\sum_{i \in \mathsf{G}} \alpha_i \mathsf{max}(-w_i, 0) \Big/ \sum_{i \in \mathsf{G}} \mathsf{max}(-w_i, 0)$$

Note that a constraint defined using gross, long, or short value aggregation is assumed to be satisfied for any portfolio, where the denominator of the above formulas is zero valued, in which case the aggregation is undefined.

B Risk Factor Models

Linear risk factor models are a useful way to estimate and model the risk statistics of a large universe of securities in a parsimonious manner. A risk factor model defines a set of common factors – typically of a much smaller size than the security universe – that drive the systematic portion of risk for any portfolio of securities, and quantifies the exposure of every security to these common factors. By decomposing risk and return into several common factors and residuals, they provide valuable insight into sources of portfolio risk and return. Moreover, factor models provide not only a more efficient, but a more robust way of building a covariance matrix of returns. The naïve approach of computing all the pair-wise covariances for a large trading universe often subjects the covariance matrix estimate to significant estimation errors, which are generally amplified by the optimization process.

In a factor model, the covariance matrix Q is decomposed into factor and non-factor components:

$$Q = FCF' + D,$$

where F is an $n \times k$ factor exposure (loading) matrix, C is a $k \times k$ covariance matrix of the factors, and D is the $n \times n$ matrix of non-factor covariances. D is typically modeled a diagonal or block diagonal matrix, which is the reason that the non-factor risk is sometimes referred to as idiosyncratic or security-specific risk.

Bloomberg Portfolio Optimizer uses Bloomberg fundamental factor models to represent the security correlations. Below we give a very brief introduction to Bloomberg equity fundamental and fixed income fundamental factor models. For details on Bloomberg factor models the reader is referred to [12, 13, 11, 16].

B.1 Equity fundamental factors

Bloomberg equity factor models include regional (US, Europe, Asia ex Japan, Japan), integrated regional model and a global model.

B.1.1 Style factors

It is well established that stocks with different fundamentals exhibit different risk and return characteristics. Bloomberg uses the following fundamental style factors to differentiate stocks:

Momentum	Differentiates between stocks that have risen over the past year from those that fell
Value	A composite value metric that differentiates between rich and cheap stocks
Dividend Yield	Another dimension of value, but is also distinct in its behavior from the value factor
Size	An aggregate measure distinguishing between large cap and small cap stocks
Trading Activity	A turnover-based measure of liquidity
Growth	Captures the distinction between stocks with high expected growth from stocks with low expected growth
Leverage	A composite metric of different measures of leverage
Profitability	Analyzes profit margins and measures such as ROE to differentiate between money makers and money losers
Volatility	Distinguishes between more volatile and less volatile stocks by using different indicators of variability
Earnings Variability	Analyzes how consistent the earnings, cash flows, and sales have been based on the last several years of data.

B.1.2 Industry factors

Industry factors are based on the 24 GICS Level 2 (GICS Industry Group) membership[4]:

Energy, Materials, Capital Goods, Commercial & Professional Services, Transportation, Automobiles & Components, Consumer Durables & Apparel, Consumer Services, Media, Retailing, Food & Staples Retailing, Food, Beverage & Tobacco, Household & Personal Products, Health Care Equipment & Services, Pharmaceuticals, Biotechnology & Life Sciences, Banks, Diversified Financials, Insurance, Real Estate, Software & Services, Technology Hardware & Equipment, Semiconductor Equipment, Telecommunication Services, Utilities.

If security belongs to a given industry, it is assigned the exposure value of 1 to this industry, and 0 for all other industries. In cases when GICS data is not available, Bloomberg infers the GICS industry group on the basis of the Bloomberg Industry Codes, or BICS.

B.1.3 Country and currency factors

In addition to style and industry factors, international models incorporate country and currency factors to capture country and currency specific risk. Each equity security has a unit exposure to the respective country/currency depending on its country of domicile and trading currency. For example, if a company is listed in London and is domiciled in Germany, it has exposure of 1 to Germany country factor and exposure of 1 to British Pound currency factor. The remaining country and currency exposures are set to zero.

[4]Industry classification for the Japan model is different – please see the white paper for the Japan equity model.

B.2 Fixed-income fundamental factors

Current version of the Bloomberg fixed-income factor model coverage extends to all fixed income sovereign, agency, corporate bonds in investment grade as well as high yield grade denominated in 42 currencies.

B.2.1 Yield curve factors

Bloomberg fixed-income fundamental factor model uses the following key rate factors for the G6 markets: 6M, 1Y, 2Y, 3Y, 5Y, 7Y, 10Y, 20Y, and 30Y and the square of the average curve change. Yield curve factors are explicit factors, i.e., their values are directly observed in the market. Exposures to the yield curve factors are key rate durations and option-adjusted convexity. Base curve for sovereign bonds is the sovereign curve, whereas the base curve for corporate bonds and agency bonds is the swap curve.

B.2.2 Spread factors

Spread factors capture systematic sources of risk for securities trading above the base curve. Spread factors are constructed for each model and asset class separately. Spread factors returns are implicit and their values are obtained by cross-sectional regression. We refer the reader to [13] for modeling details of the spread factors.

B.2.3 Currency factors

In addition to the yield curve and spread factors, for portfolios that contain securities with different denomination than the portfolio base currency, currency factors capture currency specific risk. Similar to the yield curve factors, currency factors are observable in the market and, thus, are explicit factors.

References

[1] M. S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Operations Research Letters*, 37:187–191, 2009.

[2] A. Atamtürk and A. Bhardwaj. Supermodular covering knapsack polytope. *Discrete Optimization*, 18:74–86, 2015.

[3] A. Atamtürk and A. Goméz. Submodularity in conic quadratic mixed 0-1 optimization, 2016. BCOL Research Report 16.02, UC Berkeley, Forthcoming in *Operations Research*.

[4] A. Atamtürk and A. Goméz. Strong formulations for quadratic optimization with m-matrices and indicator variables. *Mathematical Programming*, 170:141–176, 2018.

[5] A. Atamtürk and A. Gómez. Simplex QP-based methods for minimizing a conic quadratic objective over polyhedra. *Mathematical Programming Computation*, 11:311–340, 2019.

[6] A. Atamtürk and H. Jeon. Lifted polymatroid inequalities for mean-risk optimization with indicator variables. *Journal of Global Optimization*, 73:677–699, 2019.

[7] A. Atamtürk, L. F. Muller, and D. Pisinger. Separation and extension of cover inequalities for conic quadratic knapsack constraints with generalized upper bounds. *INFORMS Journal of Computing*, 25:420–431, 2013.

[8] A. Atamtürk and V. Narayanan. Polymatroids and risk minimization in discrete optimization. *Operations Research Letters*, 36:618–622, 2008.

[9] A. Atamtürk and V. Narayanan. Conic mixed-integer rounding cuts. *Mathematical Programming*, 122:1–20, 2010.

[10] A. Atamtürk and V. Narayanan. Lifting for conic mixed-integer programming. *Mathematical Programming*, 126:351–363, 2011.

[11] N. Baturin and E. Cahan. Commodities Factor Model. Bloomberg Portfolio & Risk Analystics Research Report, March 2014.

[12] N. Baturin, E. Cahan, S. Persad, and X. Xu. Global Equity Fundamental Factor Model. Bloomberg Portfolio & Risk Analystics Research Report, May 2011. Version 1.0, Bloomberg L.P.

[13] Y. Gan. Fixed Income Fundamental Factor Model. Bloomberg Portfolio & Risk Analystics Research Report, December 2016.

[14] R. C. Grinold and R. N. Kahn. *Active Portfolio Management: A Quantitative Approach for Providing Superior Returns and Controlling Risk*. McGrow-Hill, New York, 2nd edition, 2000.

[15] H.M. Markowitz. *Portfolio selection: Efficient diversification of investments*. Wiley, 1991.

[16] J. Menchero and L. Ji. Multi-asset Class Risk Model (MAC2). Bloomberg Portfolio & Risk Analystics Research Report, August 2016.

[17] V. Rashkovich and A. Verma. Trade cost: Handicapping on PAR. *Journal of Trading*, 7:47–54, 2012.

Index

> CONTACT US

To learn more about Bloomberg's Portfolio & Risk Analytics, contact your Bloomberg account representative or press the <HELP> key twice on the Bloomberg Professional service.

>>

BEIJING +86 10 6649 7500	**FRANKFURT** +49 69 9204 1210	**LONDON** +44 20 7330 7500	**NEW YORK** +1 212 318 2000	**SÃO PAULO** +55 11 2395 9000	**SYDNEY** +61 2 9777 8600
DUBAI +971 4 364 1000	**HONG KONG** +852 2977 6000	**MUMBAI** +91 22 6120 3600	**SAN FRANCISCO** +1 415 912 2960	**SINGAPORE** +65 6212 1000	**TOKYO** +81 3 3201 8900

bloomberg.com