MSCI

Research Insight

Treatment of Fixed Transaction Costs in Barra Optimizer

Leonid Kopman
Scott Liu

April 2011

1. Introduction

Controlling transaction costs is an essential element of good investment management practice. There are many types of transaction costs that should be considered; for example, brokerage commissions, order processing charges, market impact costs, and so forth. While most transaction costs depend on the number of shares or dollar amount traded, some do not. These are called fixed transaction costs. In this paper, we will consider fixed transaction costs in the context of portfolio optimization.

The structure of this paper is as follows. Section 2 will introduce the notion of fixed transaction costs and demonstrate how considering them during optimization is important. Section 3 will describe the types of transaction cost functions that can be used in Barra Optimizer. Section 4 will briefly discuss the methods utilized to solve a portfolio optimization problem with fixed transaction costs. Section 5 will present testing results.

2. Fixed Transaction Costs in Portfolio Optimization

2.1 What are Fixed Transaction Costs?

When trading, institutional money managers often encounter situations where parts of the trading costs do not depend on the monetary value of the trade or the number of shares traded. These costs are constant, regardless of the trade. These costs can arise from various government duties and taxes (e.g., stamp duty), exchange charges (e.g., transfer fee) and brokerage fees and commissions. Whereas government duties are typically up to hundreds of dollars, some brokerages in Europe and the Middle East can have fixed commissions for certain trade types, regardless of the amount traded. These commissions can reach tens of thousands of dollars, and hence are non-negligible. In the modeling community, it is customary to call costs that are not dependent on the amount of activity (and are present if and only if the activity is engaged in) fixed costs.

Mathematically, fixed transaction costs handled by Barra Optimizer can be represented as

$$TC_f(h,h^0) = \sum_{i:h_i>h_i^0} fbc_i + \sum_{i:h_i<h_i^0} fsc_i ,$$

where

h^0 is the initial portfolio, h is the final portfolio resulting from trading, and fbc_i and fsc_i are fixed costs of buying and selling asset i, respectively.

Figure 1 below illustrates a transaction cost function consisting of a fixed transaction cost and a piece-wise linear transaction cost.

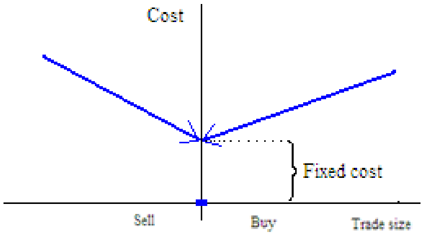

Figure 1: Fixed transaction cost

Since a transaction cost function containing a fixed transaction cost is discontinuous and non-convex, portfolio optimization problems with fixed transaction costs are hard to solve. We will discuss the algorithms used to solve portfolio optimization problems with fixed transaction costs in Section 4.

The size of institutional portfolios under management is typically on the order of tens of millions of dollars (low end) to hundreds of billions of dollars (high end). The size of fixed transaction costs, as explained above, can be on the order of hundreds to tens of thousands of dollars. Therefore, fixed transaction costs can be on the order of 10^{-3} to 10^{-8} of portfolio value and their order of magnitude is typically less than 10^{-4} of portfolio value. Nevertheless, this means that fixed transaction costs can have a non-negligible impact on the net return and utility function of a given portfolio. Hence, they cannot always be simply ignored; instead, for institutional investors using optimization, fixed transactions costs must be specifically handle in the optimization. In the next sub-section, we will use examples to further demonstrate this point.

2.2 The Importance of Fixed Transaction Costs in Optimization: Two Case Studies

In this sub-section, we will illustrate the importance of considering fixed transaction costs in the optimization process with two case studies that analyze two active portfolio rebalancing problems, based on the MSCI US 750 index. The initial portfolio contains 80 assets. Both problems are long-only, with return, risk term (with 0.0075 risk aversion) and transaction costs in the objective function.

Case 1: Besides the basic setting described above, Case 1 has transaction costs consisting of buy and sell rates of 100 bps per amount traded and fixed transaction cost of 1 bp per trade. Portfolio turnover is limited to 15%.

One could assume that ignoring fixed transaction costs during optimization could be a viable option in this situation: after all, how much trading would the optimum solution involve? Indeed, the presence of fairly high linear transaction costs and a turnover constraint, in addition to the fact that this is a rebalancing problem, should limit trading and transaction costs. However, it turns out this is not necessarily the case.

We have made two optimization runs: one run ignoring fixed transaction costs (but taking it into account when presenting results), the other taking it into consideration during the optimization, utilizing the algorithm in the Barra Optimizer. The results are indicated as Portfolio 1 and Portfolio 2, and their characteristics are summarized in Table 1.

Portfolio 2 has a net return 10 bps higher, a risk 10 bps lower, and as a result, a higher Sharpe Ratio. This illustrates the positive effects of accommodating fixed transaction costs in the optimization process, even in an instance where one would think it might not make a difference. Of course, the more trading involved, the higher these effects would be.

Table 1. Effects of ignoring fixed transaction costs in optimization

	Return, %	TC, %	Net Return,%	#Trades	Risk, %	Utility	Sharpe Ratio
Portfolio 1	**4.33**	**0.68**	**3.65**	**35**	**6.24**	**0.0336**	**0.585**
Portfolio 2	**4.25**	**0.5**	**3.75**	**22**	**6.17**	**0.0347**	**0.608**

Case 2: The second case differs from the first case as follows: there are no buy and sell rates; the fixed transaction costs are even smaller, i.e., 0.1bps; and there is no turnover limit.

Similar to Case 1, Portfolio 3 resulted from ignoring fixed transaction costs (but took the costs into account when presenting results); Portfolio 4 resulted from taking these costs into consideration during the optimization. Their characteristics are summarized in Table 2.

Because there are no piecewise linear transaction costs and no turnover limit, both optimization runs exhibit more trades than those in Case 1. However, the small fixed transaction costs (i.e., 0.1bps) still had the effect of reducing the number of trades (from 694 to 289). More importantly, Portfolio 4 has a net return that is higher by 40.5 bps, a risk that is higher by 6.1 bps, and as a result, a higher Sharpe Ratio. This again illustrates the positive effects of accommodating fixed transaction costs in the optimization process, even in an instance where one would think it might not make a difference.

Table 2: Effects of ignoring fixed transaction cost in optimization

	Return,%	TC, %	Net Return%	#Trades	Risk, %	Utility	Sharpe Ratio
Portfolio 3	**3.245**	**0.694**	**2.551**	**694**	**1.291**	**0.0130**	**1.976**
Portfolio 4	**3.245**	**0.289**	**2.956**	**289**	**1.352**	**0.0158**	**2.186**

3. Transaction Cost Function Types in Barra Optimizer

Transaction costs can be divided into two broad types: *direct* costs, which include brokerage commissions and other order processing charges, and *indirect* costs, or market impact costs, which are the costs associated with price changes due to a manager's own trade (in general, nonlinear). In particular, for institutional investors who have a large portfolio, low liquidity, high turnover and variability in cash inflow/outflows, it is crucial to incorporate these two types of transaction costs into the portfolio construction process.

Barra Optimizer users can utilize three types of transaction cost functions: piece-wise linear transaction cost, nonlinear transaction cost, and fixed transaction cost. Users can include one or more of these transaction cost function types in a single optimization problem. All three of these transaction cost functions are briefly described below.

3.1 Piecewise Linear Transaction Cost Function

A simple form of the piece-wise linear transaction cost function is:

$$TC_p(h,h^0) = \sum_{i=1}^{n} \left[c_{i\,buy} \cdot \max(h_i - h_i^0, 0) + c_{i\,sell} \cdot \max(h_i^0 - h_i, 0) \right]$$

where $c_{i\,buy}$ and $c_{i\,sell}$ are unit transaction costs for buying and selling, respectively. To ensure convexity, it must be true that $c_{i\,buy} \geq 0, c_{i\,sell} \geq 0$. Graphically, it can be represented as in Figure 2. More complicated piece-wise linear transaction cost functions may include multiple break points at the buy side, as well as the sell side.

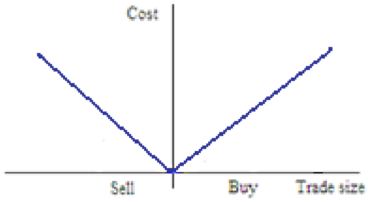

Figure 2: Simple piece-wise linear transaction cost

Since the market impact cost is a nonlinear function, users can model it by utilizing either a nonlinear transaction cost described in the next subsection, or a multiple breakpoint piece-wise linear transaction cost function. When users take advantage of Barra's Market Impact Model (MIM), the transaction cost function will be a piece-wise linear function with multiple break points, as illustrated in Figure 3. Users can also provide their own multiple break points piece-wise linear transaction cost function. To ensure convexity, the slopes between those break points for an asset must be non-decreasing.

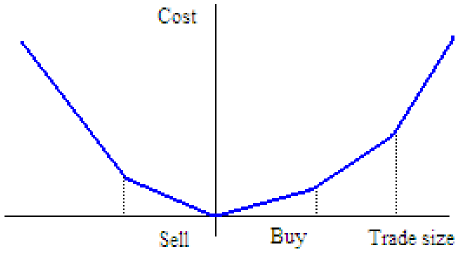

Figure 3: General piece-wise linear transaction cost

3.2 Nonlinear Transaction Cost Function

Barra Optimizer accommodates nonlinear transaction costs in the following form:

$$TC_n(h, h^0) = c \sum_{i=1}^{n} \left| h_i - h_i^0 \right|^p$$

where $c > 0$ is the nonlinear transaction cost multiplier and $p > 1$ is the exponent of the power function. The shape of the transaction cost function curve, as represented in Figure 4, will depend on the multiplier *c* and the exponent *p*.

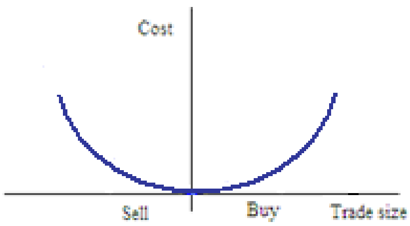

Figure 4: Nonlinear transaction cost

3.3 Fixed Transaction Cost Function

As discussed in Section 2, fixed transaction costs are, essentially, fixed charges for buying and selling, and can be defined as

$$TC_f(h, h^0) = \sum_{i: h_i > h_i^0} fbc_i + \sum_{i: h_i < h_i^0} fsc_i ,$$

where fbc_i and fsc_i are fixed costs of buying and selling asset *i*, respectively.

3.4 Standard Portfolio Optimization Problem with Transaction Costs

Mathematically, a simple Standard Portfolio Optimization problem can be represented as:

$$\underset{\mathbf{h}}{\text{Maximize}}: \quad r^T h - \lambda h^T Q h - \lambda_{TC} TC(h, h^0) \qquad (1)$$

Subject to:

$$e^{\mathrm{T}} h = 1 \qquad \text{(Holding Constraint)} \qquad (2)$$

$$TC(h, h^0) \le U_{tc} \qquad \text{(Transaction Cost Limit Constraint)} \qquad (3)$$

$$lb \le h \le ub \qquad \text{(Bound Constraints)} \qquad (4)$$

where

λ_{TC}	=	multiplier for the transaction cost term, with a default value of 1
r	=	$n \times 1$ vector of asset returns, where *n* is the number of assets
h	=	$n \times 1$ vector of portfolio holdings
h^0	=	$n \times 1$ vector of initial portfolio holdings
$h^T Q h$	=	risk term
λ	=	risk aversion
$TC(h, h^0)$	=	transaction cost function
e	=	$n \times 1$ vector of 1's
U_{tc}	=	upper bound on the transaction cost limit constraint
lb, ub	=	lower and upper bounds on assets

The transaction cost function $TC(h, h^0)$ can contain three optional components: piece-wise linear $TC_p(h, h^0)$, nonlinear $TC_n(h, h^0)$ and fixed $TC_f(h, h^0)$, and can be written as:

$$TC(h, h^0) = TC_p(h, h^0) + TC_n(h, h^0) + TC_f(h, h^0)$$

4. Approaches for Optimization with Fixed Transaction Costs

Fixed transaction cost is a discrete feature in a portfolio optimization problem. The possible methods for handling this problem can be divided into two broad categories. The first consists of formulating the problem as a discrete quadratic programming problem (with or without binary variables) and applying enumeration methods (such as branch-and-bound and its variations) to the problem. The second consists of various heuristics.

We have implemented an heuristic based on the algorithm described in Lobo, Fazel and Boyd (2007). It is founded on this simple idea: to iteratively solve the problem, and each time amortize the fixed transaction cost incurred by the solution in previous iteration (over the amount traded by that solution). For comparison purposes, we also incorporated fixed transaction costs in a CPLEX-based optimizer using binary variables.

4.1 Basic Algorithm

Let's say we are solving a portfolio optimization problem *P* in the following form (further simplified for the purposes of exposition):

$$\begin{aligned} &\max r'h - \lambda h'Qh - TC(h) \\ &\quad h'e = 1 \\ &\quad lb \le h \le ub \end{aligned}$$

where *h* is the vector of asset holdings, and *TC(h)* is the separable transaction cost function containing a piece-wise linear component and a fixed component:

$$TC(h) = \sum_i TC(h_i)$$

$$TC(h_i) = \begin{cases} a\,|h_i - h_i^0| + b, & |h_i - h_i^0| > 0 \\ 0, & h_i = h_i^0 \end{cases}$$

For each asset, if a trade is made, the cost incurred is the sum of fixed transaction cost b and a piece-wise linear transaction cost $a\,|h_i - h_i^0|$; no cost is incurred if no trade is made.

$TC(h_i)$ is discontinuous and non-convex, which is illustrated by Figure 1. However, its convex relaxation (also called outer approximation) $\widehat{TC}(h_i)$ can be obtained using bounds on *h*.

$$\widehat{TC}(h_i) = \begin{cases} \left(a + \dfrac{b}{ub_i - h_i^0}\right)(h_i - h_i^0), & h_i > h_i^0 \\ \left(a + \dfrac{b}{h_i^0 - lb_i}\right)(h_i^0 - h_i), & h_i < h_i^0 \end{cases}$$

Clearly, $\widehat{TC}(h_i)$ is continuous and convex, as illustrated by Figure 5.

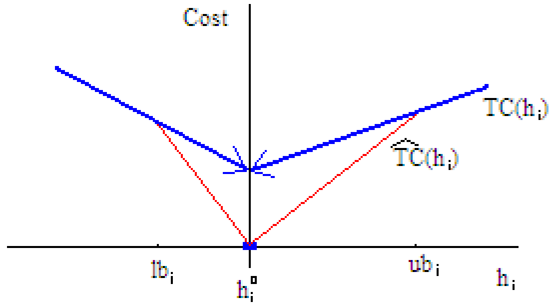

Figure 5: Fixed transaction cost and its outer approximation

For $lb_i \le h_i \le ub_i$, $\widehat{TC}(h_i) \le TC(h_i)$, since fixed transaction cost *b* is fully incurred only when $lb_i = h_i$ or $h_i = ub_i$.

The heuristic proposed by Lobo, Fazel and Boyd is based on the idea of amortizing the fixed transaction cost over traded amounts, similar to the idea used to construct $\widehat{TC}(h_i)$ above. It utilizes a small constant δ, which has a meaning of "tolerance for zero." That is, if $|h_i - h_i^0| < \delta$, the trade is assumed not to take place, and hence a fixed transaction cost is not incurred.

Algorithm 1

STEP 1. Set k=1. Solve problem P_1 obtained from *P* by substituting $TC(h)$ with

$$TC^1(h) = \widehat{TC}(h).$$

Let the solution be h^1.

STEP 2. Set k=k+1. Define $TC^k(h)$ as follows:

$$TC^k(h_i) = \left(a + \frac{b}{|h_i^{k-1} - h_i^0| + \delta} \right) |h_i - h_i^0|.$$

Obtain problem P_k from *P* by substituting $TC(h)$ with $TC^k(h)$ and solve it.

Let the solution be h^k.

STEP 3. If $h^k \approx h^{k-1}$, report h^k and exit. Otherwise, go to Step 2.

Algorithm 1, essentially, is built on the idea of amortizing fixed transaction costs evenly over the transaction amounts in the previous iteration, as illustrated by Figure 6. Fazel, et al, provide proof of convergence in their paper.

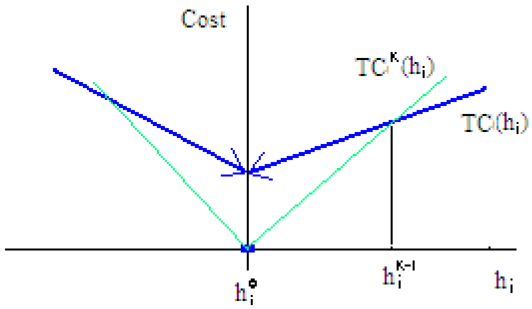

Figure 6: Amortizing fixed transaction cost.

Fazel, et al, report that the algorithm exhibits good performance on small-sized (10 assets) test problems: fast convergence (4 iterations or less) and very small differences between the optimum solution (obtained by enumerative methods) and the heuristic solution provided by the algorithm. Our computational results, presented in Section 5, confirm these findings.

4.2 Extensions and Modifications of the Basic Algorithm

We have extensively modified the original algorithm to allow its application to a broader class of transaction cost functions, as well as to improve its execution speed and solution quality.

Note that the original algorithm was written for the case where the non-fixed component of transaction costs consists of a piece-wise linear function that has just one breakpoint (h_i^0) and two line segments, which we call *slopes*. However, the basic idea of amortizing fixed transaction costs over traded amounts works for the case of multiple breakpoints as well, as long as one is being careful keeping the resulting approximation convex. Essentially, if we want to amortize fixed transaction cost b over the amount traded - $|h_i - h_i^0|$, we add $\frac{b}{|h_i - h_i^0|}$ to every slope of the piece-wise linear transaction cost. This keeps the modified piece-wise linear transaction cost convex, assuming the original one was convex. Likewise, if the piece-wise linear transaction cost is absent in the first place, we add a "dummy" cost (containing one breakpoint - h_i^0 , and two slopes - both zero, for each asset *i*) to the problem.

The algorithm was also modified to allow its application to a transaction cost function containing a nonlinear component (described in Section 3.2). The algorithm was modified to optionally accommodate counting crossover trades (i.e., trades that result in position changes from long to short, or vice versa) as two trades, as opposed to one trade.

Finally, numerous modifications were made in the interests of performance improvement, including replacing convergence criteria, allowing for hot starts from the previous iteration, and so forth. Computational results, presented in the next section, demonstrate the strong performance of the algorithm.

5. Computational Results

5.1 Performance with Various Types of Transaction Cost in the Objective Function

Large, randomly generated sets of portfolio optimization problems, based on real-world problems, were utilized to test the Barra Optimizer performance. We created a number of problem sets to test our heuristic for portfolio optimization problems with fixed transaction costs. The problem sets are based on the MSCI US 750 Index, and contain problems with and without general linear constraints, turnover constraints, factor constraints, and with varying risk aversions, with and without linked and composite assets, and so forth. Two runs were made on each problem set. We performed the first run using our heuristic, denoted below as approach H, standing for *amortization heuristic*. For comparison purposes, the second run was made using a leading commercially available general purpose optimizer utilizing branch-and-bound that is based on a formulation with binary variables. This approach is denoted below as BB, standing for branch-and-bound. For most problem sets, BB was allowed 20 seconds running time for each problem. We let it run longer on one problem set for a more detailed comparison. Below we describe the problem set features.

In Tables 3 and 4 below, “FTC only” means the transaction cost is "fixed only," i.e., there are no other types of transaction cost. “FTC with Buy/Sell” means a fixed transaction cost and simple linear rates for buys and sells. “FTC with MIM” means a fixed transaction cost plus transaction costs provided by the Barra Market Impact Model. Construction problems start with cash, while rebalancing problems start with an 80-asset portfolio.

The columns denoted “Average Time” display the CPU time used by the heuristic and BB algorithms. Although we limited the maximum CPU time for BB to 20 seconds, it can be seen that the heuristic is still faster by more than an order of magnitude. The next two columns contain the number of cases that exhibit performance differences: either heuristic solution is better than that of BB (denoted H>BB), or a BB solution is better than that of heuristic (denoted BB>H). Both the heuristic and BB obtain solutions of same or similar quality for the remaining cases.

The last two columns contain the maximum utility gap between the two algorithms (out of 256 cases for portfolio construction or out of 512 cases for rebalancing). Since CPU time used for BB was limited, BB did not get an optimal portfolio for some cases either; hence, heuristic solutions may perform better than BB solutions in many cases.

Table 3. Heuristic performance for portfolio construction problems (H=heuristic, BB=Branch&Bound).

TC	Average Time (Sec)		# of cases (out of 256)		Average Utility (256 cases)		Maximum Utility Gap	
	H	BB	#of cases H>BB	#of cases BB>H	H	BB	H-BB	BB-H
FTC only	**0.41**	**19.37**	**110**	**81**	**0.01509**	**0.01208**	**3.14e-02**	**3.64e-03**
FTC with buy and sell	**0.45**	**19.2**	**154**	**53**	**0.00356**	**-0.00108**	**4.09e-02**	**2.97e-03**
FTC with MIM	**0.73**	**20**	**187**	**14**	**0.01016**	**-0.00374**	**5.43e-02**	**1.36e-03**

Table 4. Heuristic performance for portfolio rebalancing problems.

TC	Average Time (Sec)		# of cases (out of 512)		Average Utility (512 cases)		Maximum Utility Gap	
	H	BB	#of cases H>BB	#of cases BB>H	H	BB	H-BB	BB-H
FTC only	**0.35**	**19.0**	**180**	**162**	**-0.0075**	**-0.00901**	**3.78e-02**	**5.33e-03 (20)**
FTC with buy and sell rates	**0.35**	**16.86**	**196**	**95**	**-0.0191**	**-0.01881**	**2.64e-02**	**7.43e-03 (31)**
FTC with MIM	**0.60**	**20.0**	**248**	**85**	**-0.0128**	**-0.01883**	**7.99e-2**	**1.22e-03 (7)**

5.2 The Quality of the Heuristic Solutions

In the last column of Table 4, the number in the brackets under the maximum utility gap denotes the number of cases for which the utility of the BB solution is better than that of the heuristic, and the difference is bigger than 5.0e-04 (i.e., 5 bps in terms of risk-adjusted net return).This demonstrates the quality of the solutions obtained by the heuristic. For example, among the 512 cases with FTC only, there are only 20 cases for which the utility of the BB solution is better than that of heuristic by more than 5 bps.

To evaluate how suboptimal the heuristic solutions are in the cases where the optimal solution is known, the following numerical experiment was performed. The BB was allowed to run for a maximum of 720 seconds per case on the rebalancing problems, with FTC and linear buys and sell transaction costs, to let it find optimal solutions for more cases. Out of 512 cases, 161 were solved to optimality (on the rest, timeout was reached). Table 5 demonstrates the results for these 161 cases with known optimal solutions. It can be seen that the heuristic solution loses to the optimal solution, at the most, 3.5e-4 in terms of the objective function value, or 0.07% of the optimal utility.

Table 5: Comparative performance of the heuristic solutions against known optimal solutions

# if Cases	Average CPU for Heuristic (sec)	Average CPU for B&B (Sec)	Maximum utility Gap (BB-H)	Max Utility Gap (BB-H)/|BB|
161	**0.35**	**515**	**3.58e-04**	**0.07%**

6. Summary

Besides transaction costs that depend on the amount of shares or dollar value traded, such as commissions or market impact costs, there often exists a fixed charge associated with buying or selling. These charges are independent of the amount traded, and are called fixed transaction costs. They may reach significant amounts. In such cases, they need to be considered in the portfolio optimization process, since ignoring them is likely to adversely affect portfolio performance. A heuristic procedure for the portfolio optimization with fixed transaction costs has been developed and incorporated into Barra Optimizer. Computational results demonstrate that the heuristic performance compares favorably with the branch-and-bound algorithm performance, if running time is limited, as is typical in practice.

References

Lobo, M., Fazel, M., and S. Boyd, "Portfolio Optimization with Linear and Fixed Transaction Costs," *Annals of Operations Research*, 152(1):341-365, July 2007.

Client Service Information is Available 24 Hours a Day

clientservice@msci.com

Americas

Americas	1.888.588.4567 (toll free)
Atlanta	+ 1.404.551.3212
Boston	+ 1.617.532.0920
Chicago	+ 1.312.675.0545
Montreal	+ 1.514.847.7506
Monterrey	+ 52.81.1253.4020
New York	+ 1.212.804.3901
San Francisco	+ 1.415.836.8800
Sao Paulo	+ 55.11.3706.1360
Stamford	+1.203.325.5630
Toronto	+ 1.416.628.1007

Europe, Middle East & Africa

Amsterdam	+ 31.20.462.1382
Cape Town	+ 27.21.673.0100
Frankfurt	+ 49.69.133.859.00
Geneva	+ 41.22.817.9777
London	+ 44.20.7618.2222
Madrid	+ 34.91.700.7275
Milan	+ 39.02.5849.0415
Paris	0800.91.59.17 (toll free)
Zurich	+ 41.44.220.9300

Asia Pacific

China North	10800.852.1032 (toll free)
China South	10800.152.1032 (toll free)
Hong Kong	+ 852.2844.9333
Seoul	+827.0768.88984
Singapore	800.852.3749 (toll free)
Sydney	+ 61.2.9033.9333
Tokyo	+ 81.3.5226.8222

About MSCI

MSCI Inc. is a leading provider of investment decision support tools to investors globally, including asset managers, banks, hedge funds and pension funds. MSCI products and services include indices, portfolio risk and performance analytics, and governance tools.

The company's flagship product offerings are: the MSCI indices which include over 148,000 daily indices covering more than 70 countries; Barra portfolio risk and performance analytics covering global equity and fixed income markets; RiskMetrics market and credit risk analytics; ISS governance research and outsourced proxy voting and reporting services; FEA valuation models and risk management software for the energy and commodities markets; and CFRA forensic accounting risk research, legal/regulatory risk assessment, and due-diligence. MSCI is headquartered in New York, with research and commercial offices around the world.