

Development and Calibration of Currency Market Strategies by Global Optimization

MUSTAFA ONUR ÇAĞLAYAN^{1,*} AND JÁNOS D. PINTÉR^{2,3}

1: Faculty of Economics and Administrative Sciences, 2: Faculty of Engineering
Özyeğin University, Kusbakisi Caddesi, No.2, 34662 Altunizade, Istanbul, Turkey
mustafa.caglayan@ozyegin.edu.tr, janos.pinter@ozyegin.edu.tr

3: Pintér Consulting Services, Inc., Canada; www.pinterconsulting.com.

* Corresponding author

Submitted for publication

Abstract. We have developed a new financial indicator – called the Interest Rate Differential Adjusted for Volatility (IRDAV) measure – to assist investors in currency markets. On a monthly basis, we rank currency pairs according to this measure and generate a basket of pairs with the highest IRDAV values. Under positive market conditions, an IRDAV based investment strategy (buying a currency with high interest rate and simultaneously selling a currency with low interest rate, after adjusting for volatility of the currency pairs in question) can generate significant returns. However, whenever the markets turn for the worse and crisis situations evolve, investors exit such money making strategies suddenly, and – as a result – significant losses can occur. In an effort to minimize these potential losses, our research work generates an aggregate risk metric that evaluates (estimates) the total risk by looking at various indicators across different markets. These risk indicators are used to get timely signals of evolving crises and to flip the strategy from long to short in a timely fashion, to prevent losses and make further gains even during crisis periods. Since our model is implemented in Excel as a highly nonlinear computational procedure, we utilize global optimization software (Excel-LGO) to maximize the performance of the currency basket, based on our selection of key decision variables. We introduce the new currency trading model and its implementation, and then present numerical results based on actual market data.

Keywords: Quantitative model development in the financial sector • IRDAV indicator, risk metric and trading strategy development for currency markets • Excel model implementation • Global optimization by Excel-LGO • Numerical results

1. Introduction

The differences in interest rates among countries have always played an important role for investors when making buy/sell investment decisions in currency markets. The pair-wise strategy of buying the currency of the country with the higher interest rate and simultaneously selling the currency of the country with the lower interest rate, which is also known in financial markets as the “carry trading strategy”, is often perceived as one of the most popular trading strategies in currency markets. Hence, it serves as a significant motivation among certain investors to put money (also) in currency markets.

While this type of a strategy can bring very good returns over the years, an other key financial factor that needs to be kept in mind at all times is the volatility of the currency pair in question. Therefore, a strategy that incorporates both the interest rate differentials and the volatility should better fit to the investors' spectrum of good investment opportunities in currency markets.

In an effort to fulfil this need, we have introduced an indicator, which we name Interest Rate Differentials Adjusted for Volatility (IRDAV), to capture both the interest rate differentials and the volatility of the currency pairs in one single indicator, and adopted it to a trading model for major developed market currencies. We measure IRDAV for each currency pair in our analysis as the ratio of the one-month London Inter-Bank Offer Rate (LIBOR) interest rate differential between the high-yielding and low-yielding currencies to the annualized standard deviation of the daily exchange rate returns. This measure uses the same concept as the well-known Sharpe ratio (SR) defined by

$$SR = \frac{(r_p - r_f)}{\sigma r_p} \quad (1)$$

In (1) r_p is the return on the asset (or portfolio) p , r_f is the risk-free interest rate and σr_p is the standard deviation of returns from p . For further background and applications, we refer to Sharpe (1966, 1975, 1992), Treynor and Black (1973), and Bodie, Kane, and Marcus (1993).

Under positive market conditions, the aforementioned investment strategy – buying a currency with high interest rate and simultaneously selling a currency with low interest rate, after adjusting for volatility of the currency pair in question – can generate significant returns. However, whenever the markets turn for the worse and crisis situations evolve, investors exit such money making strategies suddenly: as a result, significant losses can occur. In an effort to minimize these potential losses, we also developed an aggregated risk metric that evaluates the total risk by looking at various risk indicators across different markets. We use this risk metric to get timely signals of evolving crises and to flip the strategy from long to short in a timely fashion, to prevent losses and make further gains even during crisis periods.

In summary, the methodology pursued in this study entails to hold positive carry positions (i.e. buying high-yielding currencies and selling low-yielding currencies) in a currency basket composed of currency pairs with the highest IRDAV values during normal (positive) market conditions, but then to reverse these same positions (flip to negative carry, i.e. buying low-yielding currencies and selling high-yielding currencies) when our aggregated risk metric indicates the need for increased risk-aversion in global markets.

Since our model is highly nonlinear and for all practical purposes it is a “black box” computational procedure, we need to utilize suitable global optimization technology and software to maximize the performance of the currency basket, based on our selection of key decision variables. We have implemented our model in Excel, and used the Excel-LGO solver suite to handle it. LGO integrates several algorithm implementations for constrained global and local optimization: it does not require derivatives or other structural information, except basic model function continuity and local smoothness (to support theoretical convergence properties). Let us point out that similar “black box” situations often arise in practice, when an optimization engine has to handle a closed and/or confidential model developed by others.

Our paper briefly describes the proprietary currency trading model (Section 2), and its numerical solution (Section 3). For completeness, in Section 4 we summarize our approach to the optimization of “black box” system models, relegating most technical details to the

published literature. The recently completed Excel-LGO software implementation is described in Section 5, followed by conclusions and references.

2. Model Description

Let us start by defining the key indicator introduced in our study. For each currency pair included in this analysis, we calculate IRDAV as the interest rate differential between the high-yielding and low-yielding currencies, divided by the standard deviation of the exchange rate returns calculated over a six-month period. This measure is comparable to JPMorgan's Carry-to-Risk index measure, and it can be expressed by

$$IRDAV = \frac{(i_A - i_B)}{\hat{\sigma}_{FXa / FXb}} \quad (2)$$

In (2) i_A is the (higher) interest rate in country A, i_B is the (lower) interest rate in country B, and $\hat{\sigma}_{FXa / FXb}$ is the annualized standard deviation of the daily exchange rate returns between countries A and B. For further background, we refer to Çağlayan and Giacomelli (2003, 2005), Normand et al. (2004).

The major currency pairs included in our analysis are determined according to the liquidity and transaction volume of the Foreign Exchange (FX) pairs in currency markets. In other words, the most liquid FX pairs that have the highest transaction volumes among the major market currency pairs are selected. The 18 currency pairs considered in our study are: AUD/CAD, AUD/JPY, AUD/USD, EUR/CHF, EUR/GBP, EUR/JPY, EUR/NOK, EUR/SEK, EUR/USD, GBP/CHF, GBP/USD, NOK/SEK, NZD/USD, USD/CAD, USD/CHF, USD/JPY, USD/NOK, and USD/SEK. At the beginning of each month, we rank these currency pairs according to their IRDAV values, and we pick the top IRDAV-ranked pairs to compose the actual (monthly) currency basket, where each currency pair in the basket receives an equal portfolio weight. Later on, we will use global optimization to select the best fixed number of currency pairs that will be included in the basket, in order to maximize the overall performance of the currency portfolio. The optimized top IRDAV-ranked FX pairs constitute our “carry” basket for that month until new rankings are computed at the beginning of the following month. The positive “carry” positions in the basket are held until our risk metric (which is computed daily) reaches a certain threshold value, whereupon the basket is flipped to negative “carry” positions.

In currency markets, timing the change from normal markets' “*search for yield*” status to crises periods' “*flight to quality*” status is a crucial subject, especially so for “carry” models. It is these periods when investors suddenly empty their positive “carry” positions and cause a huge depreciation in high-yielding currencies and a significant appreciation in low-yielding currencies. Therefore, holding passive positive “carry” positions in a currency basket at all times does not make sense from a financial perspective, as these positions eventually could get hit very harshly, possibly wiping out long-term positive returns within a matter of weeks. Hence, investors need to be aware of sentiment changes in financial markets to better protect themselves against a sharp decline in the performance of positive “carry” trades.

We attempt to gauge the timing of these “sentiment shifts” in the financial markets with our *risk metric*. In order to assess an aggregated risk measure across a wide spectrum of markets, our risk metric includes six different proprietary risk measures, each representing a unique risk for liquidity, credit markets, emerging markets, equity markets, commodity

markets, and currency markets. All of these risk measures have proven in the past to indicate the periods of sentiment changes successfully. In other words, the risk measures chosen have given timely signals when to exit positive “carry” positions, and to reverse these positions — thereby making possible further gains even during crisis periods. For further background on risk indicators, we refer to Çağlayan and Kantor (2001, 2002) and Çağlayan (2003).

To be more specific regarding the construction of the risk metric, on a daily basis, we first calculate Z-scores (number of standard deviations away from a 3-month average) for each of the six individual risk measures mentioned above:

$$Z_i = \frac{(RM_t^i - \mu_t^i)}{\partial RM_t^i}. \quad (3)$$

In (3) RM_t^i is the risk measure i on day t , μ_t^i is the 3-month rolling average of the risk measure i on day t , and ∂RM_t^i is the 3-month standard deviation of the risk measure i on day t . Given the individual scores Z_i , we take their weighted average to obtain the final daily reading on our risk metric. Again, the weights of the individual risk measures that define each measure’s importance in the overall risk metric will be calculated by using global optimization software, to maximize the long-term performance of the currency basket. The threshold overall Z score level, upon which the currency basket positions are flipped to “negative carry” from “positive carry” is also determined by the optimization procedure.

One can argue that (due to their financial background) the six risk measures used in constructing the risk metric are not independent. The optimized weights attempt to take care of this problem, since optimization can potentially give a zero weighting to a risk indicator that is highly correlated with another measure but that does not perform as well as the other risk measure. Our generic optimization approach would work also for other risk metric constructions in which certain parameters need to be optimized after their overall effect is evaluated by a (“black box”) computational model.

In addition to the outlined general information about the currency basket and the risk metric, our model incorporates all the specifications for a full-fledged currency trading model, including transaction costs, stop-loss levels, and take-profit levels. The returns reported later on are all transaction cost adjusted. Specifically, the model deducts 0.05% return from each trade in the basket, a realistic assumption given the size of the bid-ask spreads for most of the major currency pairs. Moreover, the model calculates the drawdown from the peak of returns and the accumulated returns from the bottom of returns to use stop-loss and take-profit levels efficiently. We prefer to apply stop-losses and take-profits on the basket rather than on the individual currency pairs so that an extraordinary performance (negative or positive) in one of the currency pairs can be compensated by the performance of other pairs in the basket, without triggering stop-loss and take-profit signals too frequently. Once a stop-loss or take-profit signal is triggered on the basket portfolio, however, the model requires a certain number of days to wait before re-entering the trades again. Since from a financial perspective there is no definite “scientific” answer on what a stop-loss or take profit level should be, and on how many days should one wait before re-entering trades after a stop-loss or take-profit level, we will let the global optimization software determine those trade parameters as well, in order to maximize the overall performance of the currency basket.

We will refer to our proprietary model as the Currency Trading Model (CTM). We measure the overall performance of the CTM with the Information Ratio (IR), where IR is defined as:

$$IR = \frac{R_{IRDAV}}{\partial R_{IRDAV}} \quad (4)$$

Here R_{IRDAV} is the compounded annualized return from the CTM basket and ∂R_{IRDAV} is the annualized standard deviation of daily returns from the CTM basket. Using the introduced notation, and the background outlined above, our decision model can be summarized as follows:

Currency Trading Model (5)

Objective function

$$\max IR = \frac{R_{IRDAV}}{\partial R_{IRDAV}} \quad IR \text{ is calculated from the Excel implementation of our CTM.}$$

Weight factor normalization constraint

$$\sum_{i=1}^6 w_i = 1 \quad w_i \text{ is the weight that risk measure } Z_i \text{ receives in the risk metric.}$$

Decision variables, with corresponding (given) lower and upper bounds

$$ncpl \leq ncp \leq ncpu$$

$$rmtl \leq rmt \leq rmtu$$

$$0 \leq w_i \leq 1 \quad i=1, \dots, 6$$

$$sll \leq sl \leq sl_u$$

$$tpl \leq tp \leq tpu$$

$$ndsl \leq nds \leq ndsu$$

$$ndtl \leq ndt \leq ndtu.$$

Here ncp is the number of currency pairs to be included in the basket; rmt is the risk metric threshold level where positions are switched from “positive carry” to “negative carry”, or vice versa; w_i are the weights of our six risk measures that appear in the aggregated risk metric; sl is the stop-loss level; tp is the take-profit level; nds is the number of days to wait after stop-loss; ndt is the number of days to wait after take-profit. The corresponding lower and upper bounds are denoted by appending the symbols l and u to the name of the decision variable.

The CTM summarized by (5) has been implemented in Excel as a stand-alone computational procedure. For each selection of the decision variables listed above, the model can generate an overall performance measure (information ratio) value for the sample period using the historical data between 1998 and 2009. In other words, changing the input decision variable settings changes the value of the objective function. Some of the decision variables are discrete, while others are continuous: hence, we are facing a moderate size mixed integer-continuous nonlinear programming (MINLP) model. This implies a huge number of possible decision variable settings to test (even applying a “reasonable” discretization), to decide on which variable combination will generate the highest information value — an impossible task to conduct manually. In our numerical tests it also became obvious that the CTM is highly nonlinear. Due to its outlined characteristics, the solution of CTM requires derivative-free methodology and software. Specifically, we have used the Lipschitz Global Optimizer (LGO) software linked to Excel. We will briefly describe our approach and the Excel-LGO implementation later on, in Sections 4 and 5.

3. Solving the Model: Numerical Results and Discussion

In an effort to test the significance of the global optimization approach within the context of our CTM, we divide our 12-year total sample period (1998 to 2009) into two sub-samples, with 1998 through 2003 being the in-sample (training) period, and 2004 through 2009 being the out-of-sample (test, verification) period. We first run the global optimization for the in-sample period only, optimizing the decision variables introduced in the previous section. Based on the decision variable values found by optimization during the training period, we then check the resulting performance of the IRDAV basket during the test period. For comparison, later on we also compare the out-of-sample period performance of the optimized IRDAV basket to the performance of a basket where we manually pick the “best suggested” values for each of the key decision variables based on an expert “trial and error” method.

The numerical results (see Table 1) clearly show a significant improvement in the information ratio based performance of our currency basket after the optimization. Table 1 shows the globally optimized performance of the IRDAV basket during both the in-sample and the out-of-sample periods. The information ratio of the IRDAV basket during the training period 1998-2003 (the period for which we run Excel-LGO) is, as expected, an eye-opening 3.22. (Informally, this means 3.22 units of return for each unit of risk taken.) More importantly, the superior performance of the IRDAV basket essentially continues during the verification period 2004-2009 as well (where we use the values of the decision variables derived in the training period, without re-optimization). In fact, during the out-of-sample period, the IRDAV basket generated 14.3% annual compounded returns, just shy of the 15.7% annual compounded returns calculated during the optimized in-sample period. Similarly, the information ratio (*IR*) of the verification period is still very high at 2.30, compared to standard financial trading models’ average information ratios that typically lie between 1.00 and 1.50.

In addition, looking at the yearly performance of the IRDAV basket during the out-of-sample period in Table 1, we see that the outstanding performance of the basket during 2004-2009, was not due to a couple of outlier “star” years. In fact, the IRDAV basket generated positive, robust, and consistent returns between 2004-2009; four out of six years producing double-digit returns.

Table 1 Performance of the IRDAV basket after optimization using LGO

	Annual Return (%)	Std. Deviation (%)	Information Ratio
1998	24.49	5.88	4.17
1999	12.47	5.09	2.45
2000	13.58	4.13	3.29
2001	13.38	4.61	2.91
2002	15.32	4.33	3.53
2003	15.36	5.06	3.04
2004	16.29	5.49	2.97
2005	12.74	3.46	3.68
2006	11.84	4.16	2.85
2007	6.33	6.39	0.99
2008	7.28	7.67	0.95

2009	33.13	8.41	3.94
1998-2003 In-Sample	15.70	4.87	3.22
2004-2009 Out-of-Sample	14.27	6.20	2.30
1998-2009 Full Sample	14.98	5.58	2.69
<i>Key Decision Variables</i>		<i>Risk Metric Weights</i>	
Number of pairs in basket:	5	RM1:	0.245
Risk Metric threshold:	1.98	RM2:	0.047
Stop-loss level:	-3.66%	RM3:	0.231
Take-profit level:	9.95%	RM4:	0.260
Number of days to wait after stop-loss:	14	RM5:	0.105
Number of days to wait after take profit:	15	RM6:	0.112

Next, for comparison we evaluated the out-of-sample period performance of the optimized basket to the performance of a basket where we manually pick the “best proposed” values for each of the key decision variables based on intuition and some trial and error, without using any optimization technique. Table 2 below shows the performance of the IRDAV basket with the manually selected (best found) values for the key decision variables, without applying a formal optimization approach (by LGO or other tools).

By selecting values for the key decision variables both randomly and manually, we were able to produce information ratio values of the IRDAV basket for the verification period (2004-2009) up to 1.60: see Table 2. As shown above in Table 1, the information ratio during the same out-of-sample period after global optimization turns out to be 2.30. In the world of finance, this is an eye-opening (nearly 44%) improvement in the overall performance of the currency basket. This relatively simple exercise shows the major advantages of using global optimization technology and software.

Table 2 Performance of the IRDAV basket without formal optimization

	Annual Return (%)	Std. Deviation (%)	Information Ratio
1998	16.01	6.40	2.50
1999	9.18	4.91	1.87
2000	4.88	4.50	1.08
2001	7.59	5.26	1.44
2002	15.62	4.67	3.35
2003	12.95	5.00	2.59
2004	5.55	4.84	1.15
2005	8.51	3.96	2.15
2006	9.47	4.18	2.26
2007	13.28	7.63	1.74
2008	12.95	9.37	1.38
2009	15.52	8.50	1.83
1998-2003 In-Sample	10.96	5.14	2.13
2004-2009 Out-of-Sample	10.83	6.76	1.60
1998-2009 Full Sample	10.90	6.01	1.81
<i>Key Decision Variables</i>		<i>Risk Metric</i>	

		<i>Weights</i>	
Number of pairs in basket:	5	RM1:	0.1667
Risk Metric threshold:	1.25	RM2:	0.1667
Stop-loss level:	-2.50%	RM3:	0.1667
Take-profit level:	8.00%	RM4:	0.1667
Number of days to wait after stop-loss:	5	RM5:	0.1667
Number of days to wait after take profit:	1	RM6:	0.1667

The computational effort related to the optimization (by Excel-LGO) is fairly significant. We needed over 18 hours runtime on a current laptop computer, with over 130,000 iterations (each iteration taking about half a second). This is primarily due to two factors. First, the number of decision variables is 12 (some of these are integers), and our model proven to be multimodal. These facts dictate the need for a thorough global (numerical) search in a 12-dimensional decision space. Based on our extensive experience with LGO and other global optimization software, a direct sampling effort with tens or hundreds of thousands of sample points seems reasonable for such models. Second, the function evaluations themselves are quite expensive, since each decision vector setting needs to be evaluated by the Excel implementation of the CTM. To illustrate this point, for examples we display below just two Excel functions that are utilized in our analysis. (Formula (6) assists to determine the number of currency pairs to be held in the CTM portfolio, and (7) serves to calculate the daily performance of the basket based on whether the risk metric is above or below the risk-aversion threshold.)

$$\text{IF}(\text{B3413} \leq \text{\$EK\$4}, \text{IF}(\text{V3413} = \text{"Long"}, \text{CY3414}, -\text{CY3414}), 0) \quad (6)$$

$$(\text{IF}(\text{EW3413} > \text{\$EY\$2}, -1 * \text{EM3414}, \text{ES3414})) - ((\text{EX3414} * 0.05) / \text{\$EK\$4}) \quad (7)$$

As shown by these Excel formulas each decision variable contains “IF” functions, making them difficult in an optimization context. There are many other built-in functions of similar complexity in the model: together they make the CTM rather complicated. We also tried to use local optimization from selected starting points: the results clearly indicated that the “black box” model is indeed multi-modal.

In the next two sections we describe briefly the mathematical and computational background of our study.

4. Global Optimization of “Black Box” System Models

4.1. Problem Statement

We shall consider the continuous global optimization (CGO) model

$$\min f(x) \quad \text{subject to } x \in D := \{x: l \leq x \leq u \quad g_j(x) \leq 0 \quad j=1, \dots, m\}. \quad (8)$$

In (8) we apply the following notation and assumptions:

$x \in \mathbf{R}^n$ n -dimensional real-valued vector of decision variables

$f: \mathbf{R}^n \rightarrow \mathbf{R}$ continuous (scalar-valued) objective function

$D \subset \mathbf{R}^n$ non-empty set of feasible solutions, a proper subset of \mathbf{R}^n : D is defined by

$l \in \mathbf{R}^n, u \in \mathbf{R}^n$ component-wise finite lower and upper bounds on x , and

$g: \mathbf{R}^n \rightarrow \mathbf{R}^m$ an m -vector of continuous constraint functions.

Without going into technical details which are outside of the scope of the present discussion, we note that the terse model formulation (8) covers many special cases. Specifically, it also covers MINLP models defined by merely continuous functions: hence, our CTM will be treated as an instance of (8). The key “black box” characteristic of the CTM is that its objective function is evaluated by a numerical procedure implemented in Excel. Similar problems often arise in the real world of optimization: their solution requires robust and efficient derivative-free methodology.

4.2. The LGO Software Package for Nonlinear Optimization

Next, we briefly review the key features of the LGO solver suite that has been designed to address (also) “black box” problems as discussed above. Developed since the late 1980’s, LGO is currently available for a range of compiler platforms (C, C++, C#, Fortran), with seamless links to several prominent optimization modeling languages (AIMMS, AMPL, GAMS, MPL), and to the leading high-level technical computing systems Maple, Mathematica, and MATLAB. For algorithmic and implementation details not discussed here, we refer to Pintér (1996, 1997, 2001, 2002, 2005, 2007, 2009, 2010a), Pintér and Kampas (2005, 2009), Pintér et al. (2006).

The overall design of LGO is based on the combination of global and local (continuous nonlinear) optimization strategies, with corresponding theoretical convergence properties. LGO hence can be used for both global and local optimization. Based on our numerical experience, LGO can also handle MINLP models with a relatively small number of integer variables. This includes many of the smaller MINLP test problems collected in Floudas et al. (1999) that are easily solved by LGO. As our results indicate, LGO successfully handled also our CTM. (We do not wish to make unjustified claims regarding competitive performance for sizeable MINLP models.)

Let us point out that LGO’s overall derivative-free design is different from many other nonlinear optimization software packages that require explicit analytical model information to support model parsing and function type specific operations (such as rigorous bounding procedures). Of course, we do not claim that one design is “universally superior” to the other. A model function decomposition-parsing-bounding based approach supports the precise solution of a range of GO models which, however, typically have to be defined “only” in terms of a given set of possible component functions. In contrast to this approach, the design of LGO allows the handling of “black box” problems that will remain outside of the scope of the mentioned analytical global optimization approaches. To illustrate the latter point, consult e.g. Pintér (1996), Pintér and Kampas (2009), Pintér et al. (2006).

In numerical practice LGO’s global search options generate a global solution estimate(s) that is (are) refined by the seamlessly following local search mode(s). The expected practical result of using LGO – barring numerical problems caused by a “tricky” multimodal objective and/or difficult feasible set, poor model scaling, etc. which could impede any nonlinear optimization software – is a global and local search based high-quality feasible solution that meets at least the local optimality criteria. (The latter are theoretically guaranteed only under standard local smoothness conditions). At the same time, one should keep in mind that no global (or other) optimization software will “always” work satisfactorily, with default settings and under resource limitations related to time, model function evaluation, or to other preset usage limits.

Extensive numerical tests and an increasing range of practical applications demonstrate that LGO and its platform-specific implementations can find a close numerical approximation of the global solution not only when using “academic” GO test problems, but also in far more complicated, sizeable GO models. Examples of real-world optimization challenges handled by various LGO implementations are laser design (Isenor et al., 2003), intensity modulated cancer therapy planning (Tervo et al., 2003), integrated oil and gas production systems (Mason et al., 2007), vehicle design (Goossens et al., 2007), industrial packing problems (Castillo et al., 2008), and fuel processing technology analysis (Pantoleonos et al., 2009). Let us also mention (without revealing confidential information) that LGO has been used also in financial and other optimization applications.

5. The Excel-LGO Software Implementation

5.1. Hardware / Software Platform Requirements and Installation

Recently, we developed an LGO link to Excel with an easy-to-use worksheet interface to assist users in the model formulation process. Excel-LGO is described by the accompanying software documentation (Pintér, 2010b); we will present only the key points here. We wish to point out, however, that this Excel-LGO development is entirely independent from the Excel Premium Solver platform development by Frontline Systems.

In order to use Excel-LGO, users need to have access to a personal computer with the following software: Microsoft Windows XP Professional or Vista; Microsoft Office 2003 or 2007, installed with .NET Programmability Support; Microsoft .NET version 2; or a more recent version of any of the software products listed above.

A typical (recommended) Excel-LGO installation structure is shown below.

C:\EXCEL-LGO	
\Documentation	Includes the User’s Guide and optional technical notes
\Models	Test examples and user models, in subdirectories
\Work	Installation directory for the Excel-LGO runtime files

5.2. Model Development and Linking an Excel Model to LGO

We assume that the model to be solved by Excel-LGO is made available by the model developer as an Excel spreadsheet. Then the model linking procedure follows the steps below (described as instructions for users).

Step 1 Add a worksheet to your model development spreadsheet with the name “LGO”. The optimization will be carried out in this new worksheet: hence all dependency links (model function definitions) have to be properly imported here. This worksheet will serve to calculate the model objective and constraint function values based on the settings of the decision variables chosen by LGO.

Step 2 Write the following headers in the cells A1 to E1:

VARIABLE NAME
VARIABLE VALUE
LOWER BOUND

NOMINAL VALUE

UPPER BOUND

Step 3 For all decision variables insert row by row the relevant information as dictated by the headers shown in Step 2, except in the VARIABLE VALUE column. The values in this column will be overwritten by LGO during the optimization. For all other headers, enter the following information for each variable: VARIABLE NAME, LOWER BOUND, NOMINAL VALUE, and UPPER BOUND.

Step 4 Leave an empty row, and then add a new header row with the following entries:

CONSTRAINT NAME

CONSTRAINT TYPE

CONSTRAINT VALUE

Step 5 For all constraint functions insert row by row the relevant information as dictated by the headers shown in Step 4. Specifically, enter the CONSTRAINT NAME and then the following information:

CONSTRAINT TYPE

For each constraint the constraint type should be one of

LE, <= or -1, if the constraint's left-hand-side should be less than or equal to 0

EQ, =, or 0, if the constraint's left-hand-side should be equal to 0.

CONSTRAINT VALUE

In each column, insert the formula of the corresponding constraint function. You can tentatively use the VARIABLE VALUE column to check the formulas, by entering a vector(s) of test values there. Note that if there are no constraints, then only the header row should be inserted. This format also supports model extensions, e.g. by adding constraints incrementally to an initial box-constrained optimization model.

Step 6 Leave an empty row then add a new header row with the following entries:

OBJECTIVE NAME

OBJECTIVE VALUE

Enter in the next row with the name of the objective function and the function value formula reference. This function value will be minimized, in accordance with the model form (4). After the model is inserted in the LGO worksheet, it is ready to solve using Excel-LGO.

To assist users with this conceptually simple and transparent setup procedure, Excel-LGO is shipped with a set of illustrative test models. Users can directly modify these models for their own purposes.

Let us illustrate the model development and entry procedure by a very simple example. Consider the optimization model:

$$\begin{aligned} \min x^2 + y^2 \\ x + y = 1 \end{aligned} \tag{9}$$

$$0 \leq x \leq 1$$

$$0 \leq y \leq 1$$

Assuming that the cell C6 is defined as B2+B3 (to calculate the constraint value) and B9 is defined by B2*B2+B3*B3 (to calculate the objective function value), the corresponding LGO worksheet segment is displayed below.

	A	B	C	D	E
1	VARIABLE NAME	VARIABLE VALUE	LOWER BOUND	NOMINAL VALUE	UPPER BOUND
2	x	0.2	0	0.1	1
3	y	0.5	0	0.2	1
4					
5	CONSTRAINT NAME	CONSTRAINT TYPE	CONSTRAINT VALUE		
6	constraint 1	=	-0.3		
7					
8	OBJECTIVE NAME	OBJECTIVE VALUE			
9	objective function	0.29			

Figure 1 LGO worksheet setup for the simple model example (9)

5.3. Solving Optimization Models Using the Excel-LGO Solver Interface

We launch the Excel-LGO graphical user interface (GUI) by executing lgo_excel.exe in the \Work subdirectory. Next, we click the GUI's "Browse" button and select the spreadsheet (.xls) file that includes the LGO model description worksheet created as described in the previous section. It makes sense here to copy the model into the \Work directory, since the results will be written into this spreadsheet file after the solution procedure is completed.

If the optimization model is not described in the right format, then the model parser will report an informative error message that appears in the GUI. Otherwise, the interface will show the model, listing the variables and constraints as per the model formulation defined in the LGO worksheet. As an example, the Excel-LGO solver interface is shown by Figure 2, after invoking the simple model (9).

After loading the optimization model, users can either change some of the solver options or can directly start to solve the model. If there are no option changes, then LGO first tries to use the options file (.opt) that has an identical name with the model worksheet in question. If this file does not exist, then Excel-LGO will work with its preset (default) options. An example lgo.opt file is included with the Excel-LGO shipment, in the \Work directory. For illustration, Figure 3 shows the options used in solving model (9).

After the options are set, users should select the Solve tab and then press its Start Optimization button. The optimization results are shown by Figure 4.

Let us remark here that the option settings shown in Figure 3 are not "pre-optimized" at all. (For example, model (9) has been solved also using a smaller number of steps.)

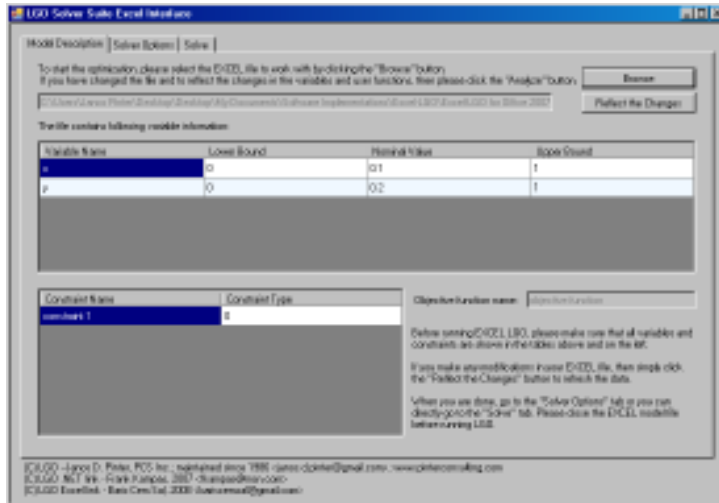


Figure 2 LGO worksheet interface: invoking model (9)

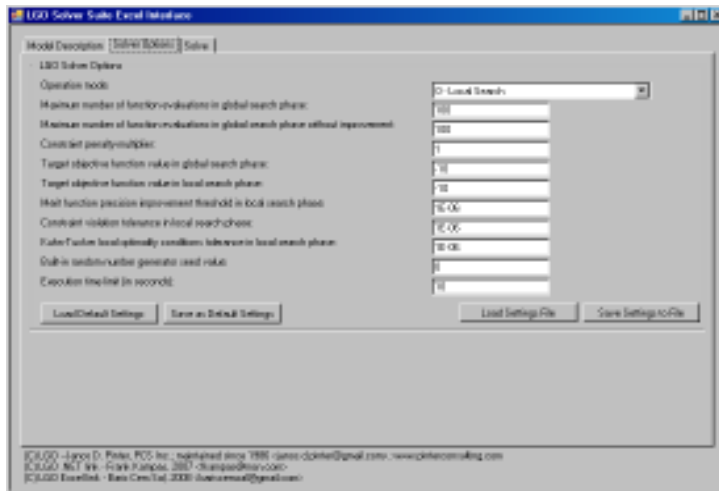


Figure 3 Excel-LGO options used in solving model (9)

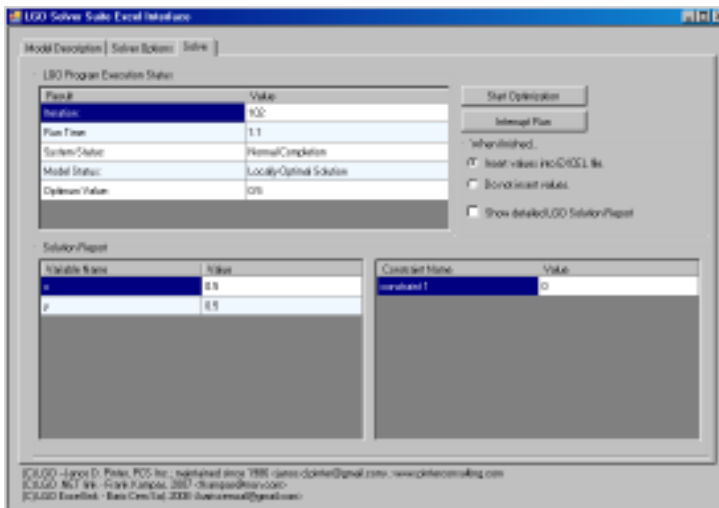


Figure 4 Excel-LGO result summary for model (9)

6 Conclusions

In this paper we have introduced a (proprietary) model to assist investors in currency markets and discuss its subsequent calibration by global optimization. Since the model has been developed using Excel as its platform, we use the Excel-LGO software to optimize our selection of key decision variables. After an introduction to the model, we present detailed numerical results based on actual market data. We also review the underlying LGO software and its Excel-LGO implementation.

From a practical point of view, our results show the need for and the validity of the global optimization approach followed. From the point of view of the targeted financial application area, we have been able to produce a high-quality trading strategy that is applicable in currency markets. Our forthcoming research work will expand upon the current model and its results, with the objective to make available a robust and efficient tool for financial decision making.

Acknowledgements

We wish to acknowledge the valuable contributions of Baris C. Sal and Frank J. Kampas to the Excel-LGO software link development project, and also thank them for their comments received during our numerical tests.

References

- Bodie, Z., Kane A., and Marcus A.J. (1993) *Investments*, 2nd Edn. Richard D. Irwin, Homewood, IL.
- Çağlayan, M.O. (2003) Alternative LCVI trading strategies. *JPMorgan Investment Strategies*, No.8 1-6.
- Çağlayan, M.O. and Giacomelli, D. (2003) Finding lower-risk FX carry trades. *JPMorgan Foreign Exchange Research Note*, 1-2.
- Çağlayan, M.O. and Giacomelli, D. (2005) Trading JPMorgan's carry-to-risk currency model in G-10 and emerging markets. *JPMorgan Foreign Exchange Research Note*, 1-16.
- Çağlayan, M.O. and Kantor, L. (2001) New LCPI trading rules: introducing FX CACI. *JPMorgan Investment Strategies*, No.3 1-6.
- Çağlayan, M.O. and Kantor, L. (2002) Using equities to trade FX: introducing the LCVI. *JPMorgan Investment Strategies*, No.7 1-4.
- Castillo, I., Kampas, F.J., and Pintér, J.D. (2008) Solving circle packing problems by global optimization: numerical results and industrial applications. *European Journal of Operations Research* 191, 786–802.
- Floudas, C.A., Pardalos, P.M., Adjiman, C.S., Esposito, W.R., Gümüs, Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A. and Schweiger, C.A. (1999) *Handbook of Test Problems in Local and Global Optimization*. Kluwer Academic Publishers, Dordrecht.
- Goossens, P., McPhee, J., Schmitke, C., Pintér, J.D., and Stahl, H. (2007) Driving innovation: how mathematical modeling and optimization increase efficiency and productivity in vehicle design. *Technical Memorandum*, Maplesoft, Waterloo, ON.
- Isenor, G., Pintér, J.D. and Cada, M. (2003) A global optimization approach to laser design. *Optimization and Engineering* 4, 177-196.
- Mason, T.L., Emelle, C., van Berkel, J., Bagirov A.M., Kampas, F. and Pintér, J.D. (2007) Integrated production system optimization using the Lipschitz Global Optimizer and the Discrete Gradient Method. *Journal of Industrial and Management Optimization* 3 (2), 257-277.
- Normand, J., Çağlayan, M.O., Ko, D., Panigirtzoglou, N., and Shen, L. (2004) JPMorgan's FX and commodity barometer. *JPMorgan Investment Strategies*, No.10 1-32.

- Pantoleonos, G., Basinas, P., Skodras, G., Grammelis, P., Pintér, J.D., Topis, S., and Sakellariopoulos, G.P. (2009) A global optimization study on the devolatilisation kinetics of coal, biomass and waste fuels. *Fuel Processing Technology* 90, 762–769.
- Pintér, J.D. (1996) *Global Optimization in Action*. Kluwer Academic Publishers, Dordrecht.
- Pintér, J.D. (1997) LGO — A program system for continuous and Lipschitz optimization. In: Bomze, I.M., Csendes, T., Horst, R. and Pardalos, P.M., Eds. *Developments in Global Optimization*, pp. 183-197. Kluwer Academic Publishers, Dordrecht.
- Pintér, J.D. (2001) *Computational Global Optimization in Nonlinear Systems*. Lionheart Publishing Inc., Atlanta, GA.
- Pintér, J.D. (2002) Global optimization: software, test problems, and applications. In: Pardalos and Romeijn, Eds. *Handbook of Global Optimization*, Volume 2, pp. 515-569. Kluwer Academic Publishers, Dordrecht.
- Pintér, J.D. (2005) Nonlinear optimization in modeling environments: software implementations for compilers, spreadsheets, modeling languages, and integrated computing systems. In: Jeyakumar, V. and Rubinov, A.M., Eds., *Continuous Optimization: Current Trends and Modern Applications*, pp. 147-173. Springer Science + Business Media, New York.
- Pintér, J.D. (2007) Nonlinear optimization with GAMS/LGO. *Journal of Global Optimization* 38, 79-101.
- Pintér, J.D. (2009) Software development for global optimization. In: Pardalos, P.M. and T. F. Coleman, eds. *Global Optimization: Methods and Applications*, pp. 183-204. Fields Institute Communications Volume 55. Published by the American Mathematical Society, Providence, RI.
- Pintér, J.D. (2010a) *LGO — A Model Development and Solver System for Nonlinear (Global and Local) Optimization. User's Guide*. (Current version) Distributed by Pintér Consulting Services, Inc., Canada. For professional information summary, see www.pinterconsulting.com.
- Pintér, J.D. (2010b) *Spreadsheet-Based Modeling and Nonlinear Optimization with Excel-LGO — User's Guide*. (With technical contributions by B.C. Sal and F.J. Kampas) Distributed by Pintér Consulting Services, Inc., Canada. www.pinterconsulting.com.
- Pintér, J.D. and Kampas, F.J. (2005) Nonlinear optimization in Mathematica with MathOptimizer Professional. *Mathematica in Education and Research* 10, 1-18.
- Pintér, J.D. and Kampas, F.J. (2009) MathOptimizer: A nonlinear optimization package for Mathematica users. (Submitted for publication)
- Pintér, J.D., Linder, D., and Chin, P. (2006) Global Optimization Toolbox for Maple: an introduction with illustrative applications. *Optimization Methods and Software* 21 (4), 565-582.
- Sharpe, W.F. (1966) Mutual fund performance. *Journal of Business*, January 1966, pp. 119-138.
- Sharpe, W.F. (1975) Adjusting for risk in portfolio performance measurement. *Journal of Portfolio Management*, Winter 1975, pp. 29-34.
- Sharpe, W.F. (1992) Asset allocation: management style and performance measurement, *Journal of Portfolio Management*, Winter 1992, pp. 7-19.
- Tervo, J., Kolmonen, P., Lyyra-Laitinen, T., Pintér, J.D., and Lahtinen, T. (2003) An optimization-based approach to the multiple static delivery technique in radiation therapy. *Annals of Operations Research* 119, 205-227.
- Treynor, J.L. and Black, F. (1973) How to use security analysis to improve portfolio selection. *Journal of Business*, January 1973, pp. 66-85.