Basics of Algorithmic Trading: Concepts and Examples

By Shobhit Seth

AAA |

An algorithm is a specific set of clearly defined instructions aimed to carry out a task or process.

Algorithmic trading (automated trading, black-box trading, or simply algo-trading) is the process of using computers programmed to follow a defined set of instructions for placing a trade in order to generate profits at a speed and frequency that is impossible for a human trader. The defined sets of rules are based on timing, price, quantity or any mathematical model. Apart from profit opportunities for the trader, algo-trading makes markets more liquid and makes trading more systematic by ruling out emotional human impacts on trading activities.

Suppose a trader follows these simple trade criteria:

- Buy 50 shares of a stock when its 50-day moving average goes above the 200-day moving average
- Sell shares of the stock when its 50-day moving average goes below the 200-day moving average

Using this set of two simple instructions, it is easy to write a computer program which will automatically monitor the stock price (and the moving average indicators) and place the buy and sell orders when the defined conditions are met. The trader no longer needs to keep a watch for live prices and graphs, or put in the orders manually. The algorithmic trading system automatically does it for him, by correctly identifying the trading opportunity. (For more on moving averages, see: *Simple Moving Averages Make Trends Stand Out.*)

Algo-trading provides the following benefits:

- Trades executed at the best possible prices
- Instant and accurate trade order placement (thereby high chances of execution at desired levels)
- Trades timed correctly and instantly, to avoid significant price changes
- Reduced transaction costs (see the implementation shortfall example below)
- Simultaneous automated checks on multiple market conditions
- Reduced risk of manual errors in placing the trades
- Backtest the algorithm, based on available historical and real time data
- Reduced possibility of mistakes by human traders based on emotional and psychological factors

The greatest portion of present day algo-trading is high frequency trading (HFT), which attempts to capitalize on placing a large number of orders at very fast speeds across multiple markets and multiple decision parameters, based on pre-programmed instructions. (For more on high frequency trading, see: *Strategies and Secrets of High Frequency Trading (HFT) Firms*)

Algo-trading is used in many forms of trading and investment activities, including:

- Mid to long term investors or buy side firms (pension funds, mutual funds, insurance companies) who purchase in stocks in large quantities but do not want to influence stocks prices with discrete, large-volume investments.
- Short term traders and sell side participants (market makers, speculators, and arbitrageurs) benefit from automated trade execution; in addition, algo-trading aids in creating sufficient liquidity for sellers in the market.

- Systematic traders (trend followers, pairs traders, hedge funds, etc.) find it much more efficient to program their trading rules and let the program trade automatically.

Algorithmic trading provides a more systematic approach to active trading than methods based on a human trader's intuition or instinct.

Algorithmic Trading Strategies

Any strategy for algorithmic trading requires an identified opportunity which is profitable in terms of improved earnings or cost reduction. The following are common trading strategies used in algo-trading:

- **Trend Following Strategies:**

The most common algorithmic trading strategies follow trends in moving averages, channel breakouts, price level movements and related technical indicators. These are the easiest and simplest strategies to implement through algorithmic trading because these strategies do not involve making any predictions or price forecasts. Trades are initiated based on the occurrence of desirable trends, which are easy and straightforward to implement through algorithms without getting into the complexity of predictive analysis. The above mentioned example of 50 and 200 day moving average is a popular trend following strategy. (For more on trend trading strategies, see: *Simple Strategies for Capitalizing on Trends*.)

- **Arbitrage Opportunities:**

Buying a dual listed stock at a lower price in one market and simultaneously selling it at a higher price in another market offers the price differential as risk-free profit or arbitrage. The same operation can be replicated for stocks versus futures instruments, as price differentials do exists from time to time. Implementing an algorithm to identify such price differentials and placing the orders allows profitable opportunities in efficient manner.

- **Index Fund Rebalancing:**

Index funds have defined periods of rebalancing to bring their holdings to par with their respective benchmark indices. This creates profitable opportunities for algorithmic traders, who capitalize on expected trades that offer 20-80 basis points profits depending upon the number of stocks in the index fund, just prior to index fund rebalancing. Such trades are initiated via algorithmic trading systems for timely execution and best prices.

- **Mathematical Model Based Strategies:**

A lot of proven mathematical models, like the delta-neutral trading strategy, which allow trading on combination of options and its underlying security, where trades are placed to offset positive and negative deltas so that the portfolio delta is maintained at zero.

- **Trading Range (Mean Reversion):**

Mean reversion strategy is based on the idea that the high and low prices of an asset are a temporary phenomenon that revert to their mean value periodically. Identifying and defining a price range and implementing algorithm based on that allows trades to be placed automatically when price of asset breaks in and out of its defined range.

- **Volume Weighted Average Price (VWAP):**

Volume weighted average price strategy breaks up a large order and releases dynamically determined smaller chunks of the order to the market using stock specific historical volume profiles. The aim is to execute the order close to the Volume Weighted Average Price (VWAP), thereby benefiting on average price.

- **Time Weighted Average Price (TWAP):**

Time weighted average price strategy breaks up a large order and releases dynamically determined smaller chunks of the order to the market using evenly divided time slots between a start and end time. The aim is to execute the order close to the average price between the start and end times, thereby minimizing market impact.

- **Percentage of Volume (POV):**

Until the trade order is fully filled, this algorithm continues sending partial orders, according to the defined participation ratio and according to the volume traded in the markets. The related "steps strategy" sends orders at a user-defined percentage of market volumes and increases or decreases this participation rate when the stock price reaches user-defined levels.

- **Implementation Shortfall:**

The implementation shortfall strategy aims at minimizing the execution cost of an order by trading off the real-time market, thereby saving on the cost of the order and benefiting from the opportunity cost of delayed execution. The strategy will increase the targeted participation rate when the stock price moves favorably and decrease it when the stock price moves adversely.

- **Beyond the Usual Trading Algorithms:**

There are a few special classes of algorithms that attempt to identify "happenings" on the other side. These "sniffing algorithms," used, for example, by a sell side market maker have the in-built intelligence to identify the existence of any algorithms on the buy side of a large order. Such detection through algorithms will help the market maker identify large order opportunities and enable him to benefit by filling the orders at a higher price. This is sometimes identified as high-tech front-running. (For more on high-frequency trading and fraudulent practices, see: *If You Buy Stocks Online, You Are Involved in HFTs.*)

Technical Requirements for Algorithmic Trading

Implementing the algorithm using a computer program is the last part, clubbed with backtesting. The challenge is to transform the identified strategy into an integrated computerized process that has access to a trading account for placing orders. The following are needed:

- Computer programming knowledge to program the required trading strategy, hired programmers or pre-made trading software
- Network connectivity and access to trading platforms for placing the orders
- Access to market data feeds that will be monitored by the algorithm for opportunities to place orders
- The ability and infrastructure to backtest the system once built, before it goes live on real markets
- Available historical data for backtesting, depending upon the complexity of rules implemented in algorithm

Here is a comprehensive example: Royal Dutch Shell (RDS) is listed on Amsterdam Stock Exchange (AEX) and London Stock Exchange (LSE). Let's build an algorithm to identify arbitrage opportunities. Here are few interesting observations:

- AEX trades in Euros, while LSE trades in Sterling Pounds
- Due to the one hour time difference, AEX opens an hour earlier than LSE, followed by both exchanges trading simultaneously for next few hours and then trading only in LSE during the last hour as AEX closes

Can we explore the possibility of arbitrage trading on the Royal Dutch Shell stock listed on these two markets in two different currencies?

Requirements:

- A computer program that can read current market prices
- Price feeds from both LSE and AEX
- A forex rate feed for GBP-EUR exchange rate
- Order placing capability which can route the order to the correct exchange
- Back-testing capability on historical price feeds

The computer program should perform the following:

- Read the incoming price feed of RDS stock from both exchanges
- Using the available foreign exchange rates, convert the price of one currency to other
- If there exists a large enough price discrepancy (discounting the brokerage costs) leading to a profitable opportunity, then place the buy order on lower priced exchange and sell order on higher priced exchange
- If the orders are executed as desired, the arbitrage profit will follow

Simple and Easy! However, the practice of algorithmic trading is not that simple to maintain and execute. Remember, if you can place an algo-generated trade, so can the other market participants. Consequently, prices fluctuate in milli- and even microseconds. In the above example, what happens if your buy trade gets executed, but sell trade doesn't as the sell prices change by the time your order hits the market? You will end up sitting with an open position, making your arbitrage strategy worthless.

There are additional risks and challenges: for example, system failure risks, network connectivity errors, time-lags between trade orders and execution, and, most important of all, imperfect algorithms. The more complex an algorithm, the more stringent backtesting is needed before it is put into action.

The Bottom Line

Quantitative analysis of an algorithm's performance plays an important role and should be examined critically. It's exciting to go for automation aided by computers with a notion to make money effortlessly. But one must make sure the system is thoroughly tested and required limits are set. Analytical traders should consider learning programming and building systems on their own, to be confident about implementing the right strategies in foolproof manner. Cautious use and thorough testing of algo-trading can create profitable opportunities.