Gauging Agent-Based Trading of a Single Financial Asset

Definition of an Absolute Distance to the Best Behaviour

Olivier BRANDOUY[1], Philippe MATHIEU[2], and Iryna VERYZHENKO[3]

[1] LEM, UMR CNRS-USTL 8179, {olivier.brandouy}|{iryna.veryzhenko}@univ-lille1.fr
[2] LIFL, UMR CNRS-USTL 8022, philippe.mathieu@lifl.fr

Obtaining good (even acceptable) performances with active management strategies in finance is a fairly hard challenge. Theoretically speaking, tenants of the efficient market hypothesis claim, with strong arguments, that a rational investor should stick to a simple *”Buy and Hold”* strategy for a correctly diversified portfolio (see for example Sharpe (1991) or Malkiel (2004)). Said differently, an active management, linked to hypothetical skills allowing managers to make appropriate *market timing* and *stock picking* decisions, would mostly generate transaction costs without real benefit. Nevertheless, this debate among theoreticians is far from being closed (see for example Brock, Lakonishok, and LeBaron (1992), Shen (2003)) as well as the whole discussion on the profitability of active *versus* passive portfolio management styles.

In this paper, we do not discuss the opportunity of such active strategies based on *market timing* nor we describe an operational process allowing fund managers to find out how to identify states in the market where ”buying” or ”selling” is particularly appropriate. We neither propose a method that rank various active strategies in terms of risk-return performance (although our framework could be extended to this bi-criteria framework). We rather propose a new method delivering *an absolute performance indicator* geared towards the evaluation of a wide range of trading strategies.

We restrict our analysis to the evaluation of strategies involving the trading of a single asset (which could be a portfolio or a market index as well). The baseline in this research is to identify the maximum profit one can obtain in trading some financial commodity, under a predefined set of trading constraints and with a complete knowledge of its price motion. We show that this question is far from being trivial, even if this target immediately evokes many popular models that most frequently prove to be completely inefficient. Once this optimal strategy identified, we propose to compute a distance towards this best behaviour for any given investment strategy. One potential application for this distance is to gauge the *ex-post* performance of investment algorithms or fund management principles that are formulated *ex-ante* the realisation of prices over which these are deployed. It also provides an alternative to the classical relative rankings of investment behaviours delivered by traditional methods. We also believe it is scientifically interesting to merely know this maximum profit even if, to our knowledge, this task has never been clearly proposed before.

We show that this best investment behaviour can be defined using a linear programming framework and solved with a Simplex approach. Nevertheless, if this method is theoretically correct, it suffers from severe limitations in terms of computability (the underlying algorithm being non-polynomial in the worst case). We therefore propose to embed this question in a graph theory framework and show that the determination of the best investment behaviour is equivalent to the identification of an optimal path in an oriented, weighted, bipartite network. We illustrate these results real data as well as simulated algorithmic trading methods.

This paper is organised as follow. We first briefly review the literature to present where our research can be placed in the existing theoretical landscape *–not present in the log abstract version–*. In a second section, we formalise the framework we start from, and give some illustrations of the complexity of the optimisation task. In a third section we present the mathematical frameworks for solving the problem as well as a new algorithm for the identification of the best investment behaviour. In a last section we discuss this latter algorithmic method and provide a series of practical implementations to gauge the absolute performance of a set of automatic trading methods.

1 Elements of the game, formalisation and examples

Consider the idealised situation in which one investor has the complete knowledge of a finite series of financial prices $\overrightarrow{p} = \{p_t | t \in [0, n]\}$, for example daily closing prices for one given stock, index or portfolio.

Let's admit these prices, defined over a time window $[t = 1, t = n], n \in \mathbb{N}$ are those at which this investor has the opportunity to rebalance his portfolio. Let's now posit a price-taker framework, *i.e.*, agent' decisions cannot affect these closing prices and sufficient liquidity at these prices is assumed. We now define the rules of a game for this investor, or said differently, a series of rules constraining her behaviour:

- At date $t = 1$ –beginning of the game –, the investor's wealth equals W_0 and is completely composed of cash C_1; no financial asset is hold ($A_1 = 0$). In other terms, $C_1 = W_1$.
- Having the knowledge of the entire price series, the idealised investor must decide for each $t \in (1, n)$ one specific action with regards to the her portfolio, "buy", "sell" and "stay unchanged", resp. coded B, S and U. In other terms, the investor has to compose a "sentence" of size n using characters in S, U, B. The interpretation of each of these actions is as follows:
  Buy: One can write B *if and only if* $W_{t-1} = C_{t-1}$. If B is written at date t, all the investor's cash is converted into assets (delivering a new quantity for $A_t \neq 0$). Assuming transaction costs at a $c\%$ rate, $A_t = \frac{W_{t-1}}{p_t \times (1+c)}$. Additionally, the first character in any sentence must be B.
  Sell : *if and only if* $A_{t-1} \neq 0$, the investor can write S and convert his position into cash. Considering an identical rate of transaction costs c, $C_t = A_{t-1} \times (p_t \times (1 - c))$
  Stay unchanged: Whatever the nature of W_{t-1} (cash or assets), he can also decide to write U and let his position unchanged at date $t : W_t = W_{t-1}$.
- This "sentence" is one investment strategy S_i over $\overrightarrow{p}$ chosen in a set of strategies $\{S\}$.

Each instance S_i can be gauged in terms of relative performance using $S_{j,j \neq i}$. What we propose here is to determine an absolute performance indicator for each of these instances with respect to the best possible strategy in $\{S\}$ in terms of maximum profit $W_{t+n} - W_t$. As we will show later, this best strategy, denoted $S*$, is relatively easy to identify when transaction costs are not implemented. On the contrary, when this is the case, this identification is far more complex.

Basic illustration.

Let's consider the following (arbitrarily chosen) price series:

$$\{100, 120, 90, 160, 126, 150, 140, 160, 110, 170, 168, 180\}$$

It can be illustrated simply on this example that, when no transaction cost minor the benefits one can expect from trading, the best strategy consists in cumulating all *positive spreads* in the price sequence. This strategy is denoted $S1^*$ in Table 1. When transaction costs are implemented, the same strategy turns to be far less interesting (see $S4$, Table 1). Some of the trades are simply not profitable in this context of high transaction costs. The optimal strategy when such costs are supported is $S3^*$ (see same Figure and Table). One can observe it does not consist in realising all profitable trades as soon as they are observed in the price sequence (for example "Buy" in position 9 and "Sell" in position 10). It is obviously not the same as in the situation where there is no transaction costs. Notice that $S4$, which is similar to $S2^*$, but in a transaction-cost free framework, is not as interesting as $S1^*$.

Table 1. Some Strategies among all 2^{12} potential *sentences*

	t	1	2	3	4	5	6	7	8	9	10	11	12	$W_{12} - W_1$
	p_t	100	120	90	160	126	150	140	160	110	170	168	180	
$c = 0$	$S1^*$	B	S	B	S	B	S	B	S	B	S	B	S	480.61
	$S4$	U	U	B	S	B	U	U	S	B	U	U	S	369.41
$c = 0.1$	$S2^*$	U	U	B	S	B	U	U	S	B	U	U	S	202.33
	$S3$	B	S	B	S	B	S	B	S	B	S	B	S	144.18
	$S5$	U	U	B	U	U	U	U	U	U	U	U	S	163.64

A trivial method to solve this problem is to generate all possible sentences and to compute the net earning one can obtain with them to identify $S*$. This set is of finite size 2^n. As we will show now, there are at least two ways to improve efficiently the determination of the optimal strategy $S*$, whatever the level of transaction costs. One is based on a simplex method, the other is based on the search of an optimal path in an oriented bipartite network.

2 Mathematical models : a linear programming method and search in graphs

In this section, we show that the identification of $S*$ can be described as a linear programming problem with a classical Simplex solution. Unfortunately, this approach is relatively inefficient since the Simplex algorithm is non-polynomial in the worst case (*i.e.*, one can lack the necessary computing resources to obtain a result as soon as the size of $\overrightarrow{p}$ becomes important.)

Initial simplification

Before formal results are presented, we introduce two theorems that are necessary to find the solution of the $S*$ determination problem. These preliminary elements aims at simplifying the solution we propose. These simplifications are not presented in the short version paper. They allow to transform the price sequence $\overrightarrow{p}$, in a filtered price sequence $\overrightarrow{fp}$ only containing extremum points of $\overrightarrow{p}$ (that is, *peaks* and *troughs*.The second simplification allow to identify in $\overrightarrow{fp}$ two vectors of prices for *potential* "buy" and "sell" actions respectively denoted $\overrightarrow{fp_B}$ and $\overrightarrow{fp_S}$. These simplifications are grounded on theorems we prove.

A linear programming method for the identification of $S*$ *–not present in the short version–*.

Embedding the identification of $S*$ in a Graph structure

Let each price in $\overrightarrow{fp}$ be depicted as a vertex in a network. We can construct a bipartite, oriented and weighted network $\mathcal{N}(E, \overrightarrow{fp_B}, \overrightarrow{fp_S})$ connecting points in $\overrightarrow{fp_B}$ (where one can only "buy" *–cf. theorem 1–*) and $\overrightarrow{fp_S}$ (where one can only "sell" *–idem–*).

The $S* -determination$ algorithm

In this section we develop a new algorithm adapted to the determination of $S*$ in the graph framework. This algorithm derives from a technique exposed by Floyd (1969). We first introduce some notations and present the Floyd shortest-path algorithm; then we expose the $S* -determination$ algorithm itself.

(i) Identifying the shortest path in $\mathcal{N}$ with the Floyd algorithm –not present in the short version–.

(ii) Towards the $S -determination$ algorithm*

If the Floyd algorithm is performed with traditional maximisation instead of minimisation operation, this algorithm will produce the *maximum longest path*.
However, we must consider that the absence of an arc must be interpreted as a length $-\infty$, whereas in the shortest-path problem the absence of an arc is interpreted as a length of $+\infty$. Thus, for the longest path problem, the matrix D^0 initialising the procedure consists of arc lengths and $-\infty$ wherever no arc appears.
In order to simplify the algorithm, edges among elements of $\overrightarrow{fp_B}$ with length 0 are allowed. This convention is necessary to find the longest path between not every pair of vertices in the graph, but the longest path between the first vertex in $\overrightarrow{fp_B}$ to every other vertex.
The other modification we introduce is to prohibit backward loops in the network.
The pseudo-code of the $S* -determination$ algorithm is as follows:

```
     for k=1 to n
          for j=k to n
    path[0][j]=max ( path[0][j], path[0][k]+path[k][j])
```

(iii) Operating the $S -determination$ algorithm* –not present in the short version–.

3 Discussion and Empirical illustrations

We provide a series of illustrations on how to determine $S*$ over various financial time series (a series of stocks observed on EURONEXT-NYSE at time scales ranging from intra-day tick-by-tick data to weekly data). Figure 1 presents $S*$ computed using the closing value of the Dow-Jones index as it was each day between 2/12/1980 and 20/02/2009. One can notice the enormous potential profit one could obtain in behaving optimally in this time window. This is mainly due its length and to the perfect knowledge one have over the price sequence. Obviously, what is really hard for investors, is that the do not have such a perfect knowledge of the future. Therefore they just can build their own strategy S using expectations or predictions. Therefore, one cannot blame ex-ante trading rules not to perform as well as ex-post optimisation techniques such as the one we develop in this paper. Nevertheless, one can remark that : (i) even if they have had access to such a perfect knowledge, it is not so evident investors would have behave optimally (identifying $S*$ being far from a trivial task), (ii) even the best trading rules, the best algorithmic trading methods, should be gauged with respect to this maximum profit rule to judge how they absolutely perform.

From a practical point of view, the $S* -determination$ algorithm can be used to select automatic trading agents on a short-range investment horizon (less than one year). We therefore illustrate the absolute performance of a series of autonomous artificial agents trained to perform as well as possible over the same time series using various techniques (mainly chosen in the set of technical trading rules).

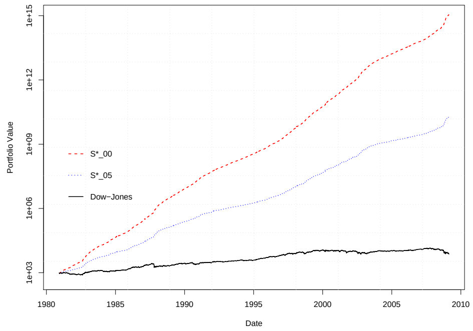

Fig. 1. $S*$ with resp. $c = 0\%$ and $c = 0.5\%$ and Dow-Jones Index (y axis in log scale)

References

ASANOV, M. O., V. BARANOVSKY, AND V. RASIN (2001): *Discrete Mathematics: Graphs, Matroids, Algorithms*. R&C Dynamics Izhevsk, Moscow.

BROCK, W., J. LAKONISHOK, AND B. LEBARON (1992): “Simple Technical Trading Rules and the Stochastic Properties of Stock Returns,” *Journal of Finance*, 47(5), 1731–1764.

CARHART, M. M. (1997): “On Persistence in Mutual Fund Performance,” *Journal of Finance*, 52(1), 57–82.

DANTZIG, G. (1966): “All Shortest Routes in a Graph,” in *Theory of Graphs, International Symposium, Rome*, pp. 91–92. Gordon and Breach, New York.

ELTON, E. J., M. J. GRUBER, AND C. R. BLAKE (1996): “The Persistence of Risk-Adjusted Mutual Fund Performance,” *Journal of Business*, 69(2), 133–57.

FLOYD, R. (1969): “Algorithm 97, Shortest Path Algorithms,” *Operations Research*, 17, 395–412.

GAREY, M. R., AND D. S. JOHNSON (1979): *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman.

JENSEN, M. C. (1968): “The performance of mutual funds in the period,” *Journal of Finance*, 23, 389–416.

MALKIEL, B. (2004): “Can Predictable Patterns in Market Returns be Exploited Using Real Money?,” *Journal of Portflio Management*, 30, 131–141.

MINIEKA, E. (1978): *Algorithms for Networks and Graphs*. Marcel Dekker.

SHARPE, W. F. (1991): “The Arithmetic of Active Management,” *The Financial Analysts’ Journal*, 47(1), 7–9.

SHEN, P. (2003): “Market timing strategies that worked – based on the E/P ratio of the S&P 500 and interest rates.,” *Journal of Portfolio Management*, 29, 57–68.

SHIER, D. (1973): “Iterative Methods for Determining the k Shortest Paths in a Network,” *Networks*, 6, 205–230.