Experimental Analysis of a Threat-based Online Trading Algorithm

Working Paper

Günter Schmidt[1+2], Esther Mohr[1] and Mike Kersch[1]

Abstract. Trading decisions in financial markets can be supported by the use of online algorithms. We evaluate the empirical performance of a threat-based online algorithm and compare it to a reservation price algorithm, an average price algorithm and to buy-and-hold. The effectiveness of the algorithms is analyzed with historical DAX prices for the years 1998 to 2007. Performance measures are geometric return and period return. The performance of the threat-based algorithm found in the simulation runs dominates all other investigated algorithms. We also compare its performance to results from worst case analysis and conduct a *t*-test.

Keywords: Investment analysis, Heuristics, OR in banking, Simulation, Uncertainty modelling

1 Introduction

Many major financial markets are electronic market places where trading is carried out automatically. Trading algorithms which have the potential to operate without human interaction are of great importance in electronic financial markets. Very often such algorithms are based on data from technical analysis as described in *Brock et al.* [BLL92], *Mils* [Mil98], *Ratner and Leal* [RL99], *Kwon and Kish* [KK02], and *Shen* [She03]. Many researchers have also studied trading algorithms from the perspective of artificial intelligence, software agents or neural networks cf. *Feng et al.* [FRS04], *Silaghi and Robu* [SR05] and *Chavarnakul and Enke* [CE08].

In order to carry out trading policies automatically they have to be converted into trading algorithms. Before a trading algorithm is applied one should be interested in its performance. The performance analysis of trading algorithms can basically be

---

[1] Saarland University
P.O. Box 151150
D-66041 Saarbrücken, Germany
E-mail: {gs,em,mk}@itm.uni-sb.de

[2] University of Liechtenstein
Fürst - Franz - Josef - Strasse
9490 Vaduz, Liechtenstein

carried out by three different approaches. One is Bayesian analysis where a given probability distribution for asset prices is a basic assumption. Another one is assuming strict uncertainty about asset prices and analyzing the trading algorithm under worst case outcomes. This approach is called worst case competitive analysis. The third one is a heuristic approach where trading algorithms are tested on historic data by simulation runs. In this paper we apply the third approach and compare the results to these of the second one. Note, that we do not deal with probabilities and we do not calculate risk measures based on probabilities. We do not want to explain market behaviour. We want to compare the performance of algorithms when we assume uncertainty of asset prices which are drawn from a given interval of minimum and maximum prices.

We consider single and multiple trade problems and analyze a threat-based online trading algorithm from a worst case and an empirical case point of view based on experimental data. For the empirical case the actually observed performance is calculated and for the worst case the worst possible performance which could have been occurred is calculated when the experimental data is considered. Moreover we compare its performance to this of an optimal trading algorithm, and three other online algorithms based on reservation prices, average prices, and buy and hold.

The reminder of this paper is organized as follows. In the next section the problem is formulated and worst case competitive analysis of the reservation price and the threat-based trading algorithms are performed. Section 3 gives a literature overview on heuristic trading rules for multiple trade problems. In Section 4 new trading rules based on online algorithms for this problem are introduced. Section 5 presents detailed experimental findings from our simulation runs. We finish with some conclusions and suggestions for future research in the last section.

2 Problem Formulation

If we trade in markets we are interested in buying at low prices and selling at high prices. Let us consider the single trade problem and the multiple trade problem with a finite trading horizon. In a single trade problem we search for the minimum price m and the maximum price M in a time series of prices once. At best we buy at m and sell later at M. Buying and selling can be interpreted as exchanging some asset d (e.g. cash) to some other asset y (e.g. stock). In a multiple trade problem we exchange assets more than once. If we buy and sell (exchange) assets p times we call the problem p-trade problem with $p \geq 1$.

As we do not know future asset prices the decisions to be taken are subject to uncertainty. How to handle uncertainty for trading problems is discussed in *El-Yaniv et al.* [YFKT01]. Trading is represented by search. To solve financial search problems a trader which owns some asset at time $t = 0$ obtains price quotations $q(t)$ with $m \leq q(t) \leq M$ at points of time $t = 1, 2, \ldots, T$. For each $q(t)$ the trader must decide which fraction $s(t)$ of his current asset he wants to sell at time t. At the last price $q(T)$ the trader must sell all the remaining fractions of the asset he holds at the last point of time T of the trading horizon. It is assumed that the time interval $[1, T]$ and the possible minimum and maximum prices m and M of the interval are known to the trader. The problem to determine $s(t)$ for $t = 1, 2, \ldots, T$ is solved by online algorithms.

An algorithm *ON* computes online if for each j = 1, …, t-1, it computes an output for j before the input for j+1 is given. An algorithm computes offline if it computes a feasible output given the entire input sequence j = 1, …, t-1.We denote an optimal offline algorithm by *OPT*. An online algorithm *ON* is c-competitive if for any input I

$$ON(I) \geq 1/c \cdot OPT(I). \qquad (2\text{-}1)$$

If the competitive ratio is related to a performance guarantee it must be a worst case measure. In such a case any c-competitive online algorithm can guarantee a value of at least the fraction $1/c$ of the optimal offline value $OPT(I)$ no matter how unfortunate or uncertain the future will be. As we have a maximization problem $c \geq 1$ the smaller c the more effective is *ON*. Later we will define a competitive ratio also for the empirical case.

We analyse the competitive ratio of two online algorithms based on a reservation price policy ($s(t) \in \{0,1\}$) and on a threat based policy ($0 \leq s(t) \leq 1$). We differ between a worst case competitive ratio for search c_s and a worst case competitive ratio for trading c_t.

(1) *Reservation Price Policy*
For the search problem the selling rule introduced by *El-Yaniv* [Yan98]

sell at the first price greater or equal to reservation price $q = \sqrt{M \cdot m}$

has a competitive ratio $c_s = \sqrt{M / m}$ where M and m are upper and lower bounds of prices $q(t)$ with $q(t) \in [m, M]$. c_s measures the worst case in terms of maximum and minimum prices. This result can be transferred to a single trade problem if we modify the rule to

buy the asset at the first price smaller or equal and sell the asset at the first price greater or equal to the reservation price $q = \sqrt{M \cdot m}$.

In the single trade problem we have to carry out the search twice. In the worst case we get a competitive ratio of c_s for buying and the same competitive ratio of c_s for selling resulting in an overall competitive ratio for the single trade problem of $c_t = c_s \cdot c_s = M/m$.

In general for the p-trade problem we get a worst case competitive ratio of

$$c_t(p) = \prod_{i=1,\ldots,p} (M(i)/m(i)) \qquad (2\text{-}2)$$

If $m(i)$ and $M(i)$ are constant for all trades $c_t(p) = (M/m)^p$. The ratio $c_t(p)$ can be interpreted as the geometric return we can achieve by buying and selling assets sequentially as stated in *Mohr and Schmidt* [MS08].

(2) *Threat-based Policy*
For the search problem the following procedure is suggested in *El-Yaniv et al.* [YFKT01]:
Choose a competitive ratio c and select a trading algorithm which can guarantee c.

(2.1) Consider exchanging asset d for asset y only when the current exchange rate $q(t)$ (number of assets y for one asset d) is the highest seen so far;

(2.2) Whenever you exchange asset d for asset y at time t convert enough to ensure that the given c would be obtained if an adversary dropped the next exchange rate $q(t+1)$ to the minimum possible rate m and kept it there until the end of the time horizon T, i.e. that this threat exists.

Let $k \leq T$ be the remaining exchange rates in the time series. Let q'(1) be the first exchange rate of this time series. Let $c^k(q'(1))$ be a competitive ratio which is achievable on a sequence of k exchange rates $q'(1), \ldots, q'(k)$. The achievable competitive ratio $c^k(q'(1))$ for k remaining trading days is determined by

$$c^k\,(q'(1))= 1+((q'(1)-m)/q'(1))*(k-1)*(1-[(q'(1)-m)/(M-m)]\hat{}(1/(k-1)) \qquad (2\text{-}3)$$

The optimal competitive ratio for the search problem is calculated by $c = sup\ c^k\,(q(1), q(2), \ldots, q(k) \mid k \leq T)$, cf. *El-Yaniv et al.* [YFKT92 and 01]. For each trade we conduct the threat-based algorithm twice, once for buying and once for selling. The competitive ratio of the threat-based algorithm for the trading problem can be calculated in the same way as it is done for the reservation price algorithm described in 2.1.

In the following we apply the reservation price and the threat-based algorithms to multiple trade problems and compare it to two other trading algorithms. Before doing that we review experimental results for heuristic trading rules from the literature.

3 Related Work

Experimental analysis of online algorithms in the field of trading is a new area. *Mohr and Schmidt* [MS08] investigate the empirical and the worst case performance of a reservation price policy introduced by *El Yaniv* [Yan98] and compare it to Buy and Hold. Experimental analysis of online algorithms in other fields can be found in *Karlin* [Kar98].

There are more results on experimental studies of trading algorithms. Unfortunately, this analysis is restricted to empirical results and does not take into account worst case results. We give a brief overview on the experimental studies on heuristic trading policies for multiple trade problems from the literature. Here the comparison to Buy and Hold is of prime interest.

Two trading algorithms suggested by *Brock et al.* [BLL92], the Moving Average (MA) crossover and the Trading Range Breakout (TRB) (also known as Momentum), are of major interest in the literature. *Mills* [Mil98], *Ratner and Leal* [RL99], *Kwon and Kish* [KK02], *Chang et al.* [CLT04] and *Tabak and Lima* [TL09] also investigate these rules.

Brock et al. [BLL92] investigate the rules MA crossover and TRB by conducting experiments with a price-weighted index (Dow Jones Industrial Average (DJIA)) for the time period from 1897 to 1986. The returns on buy (sell) signals on the DJIA are compared to returns from simulated comparison series generated by the following models: autoregressive (AR(1)), generalized autoregressive conditional heteroskedasticity in mean (GARCH-M) and an exponential GARCH. The returns obtained from buy (sell) signals of the trading rules are not likely to be generated by these three models. The results provide empirical support for utilizing technical trading rules where returns outperform not only Buy and Hold but also the autoregressive (AR(1)), generalized autoregressive conditional heteroskedasticity in mean (GARCH-M), and an exponential GARCH model.

Mills [Mil98] investigates the same two types of trading rules as in *Brock et al.* [BLL92] by conducting experiments on daily data of the London Stock Exchange FT30 index for the time intervals 1935-1954 and 1975-1994. In addition, trading signals generated by the geometric MA and the arithmetic MA are calculated. All rules are compared to Buy and Hold. The geometric MA gave an almost identical set of signals as the conventional (arithmetic) MA. Until 1980 all investigated trading rules outperform Buy and Hold. The results of *Mills* [Mil98] are consistent, in almost every respect, with those of *Brock et al.* [BLL92] until 1980. But from then on Buy and Hold clearly dominates all other trading rules. The sample used in *Brock et al.* [BLL92] ended in 1986; so there was not the data to analyze structural shifts that might have taken place starting in 1982.

Ratner and Leal [RL99] compare the MA crossover rules introduced by *Brock et al.* [BLL92] to Buy and Hold. Trading rules are investigated in ten emerging equity markets in Latin America and Asia from January 1, 1982 to April 1, 1995. The average returns considering transaction costs for each rule and country are compared to Buy and Hold the S&P500 and Nikkei225 indices. Results show that these trading rules applied to emerging markets do not have the ability to outperform Buy and Hold.

Kwon and Kish [KK02] extend the work of *Brock et al.* [BLL92] in two ways. First by investigating the predictive ability of historical data to forecast future prices for the New York Stock Exchange index (NYSE) and the National Association of Security Dealers Automatic Quotations index (NASDAQ). Second by including another MA trading rule called Moving Average with Trading Volume (MAV). The experimental study uses historical data from July 1962 to December 1996 for NYSE and from January 1972 to December 1996 for NASDAQ. The results show that the trading rules have potential to outperform Buy and Hold. *Kwon and Kish* [KK02] compare buy, sell, and buy-sell returns with returns from the random walk model. The results support the price-weighted index (Dow Jones Industrial Average (DJIA)) analysis of [BLL92] by showing that the technical trading rules add value by capturing profit opportunities when compared to Buy and Hold.

Chang et al. [CLT04] test whether returns for emerging stock markets in US and Japan are predictable. Predictability is analyzed by means of multivariate variance ratios using heteroscedastic robust bootstrap procedures. The MA and TRB rules introduced by *Brock et al.* [BLL92] are employed and compared to buy and hold. Results show that there is some evidence of forecasting power. When trading costs are taken into account only a few rules generate positive excess returns. *Chang et al.* [CLT04] check for robustness by analyzing returns from 1559 different trading rules, testing different sub-samples, analyzing returns in bear and bull markets, and also comparing results found for emerging markets to the US and Japan. For the US the MA trading rules do not seem to have forecasting power for the recent sample used by *Chang et al.* [CLT04].

Tabak and Lima [TL09] also investigate the predictive power of the MA and TRB rules introduced by *Brock et al.* [BLL92] for the Brazilian exchange rate for the 2003 to 2006 period. A bootstrap procedure is employed to test for the predictability of exchange rates. Furthermore, the ability of the trading rules to generate significant higher returns compared to the buy and hold returns is tested. Results show that the excess return generated by the MA and TRB rules is not significant, suggesting that such predictability is not economically significant. Their results are consistent with those of *Chang et al.* [CLT04].

Shen [She03] compares simple market-timing heuristics to Buy and Hold. Trading signals are generated by the value of the short spread between the Earning-Price (EP) ratio and selected interest rates using S&P500 data from 1970 to 2000. Trading rules either invest in the S&P500 index or in treasury bills over a period of one month depending on predefined thresholds. If the spread is above some threshold level, the rule invests in the S&P500 index for the next month and if the spread is below this level, the portfolio is liquidated at the end of the month and the money is invested in 30-day treasury-bills for the next month. At the end of each month spreads are considered again. The portfolio return is compared with these of S&P500 index Buy and Hold for 1970 to 2000. Results show that the trading rule outperforms the S&P500 index generating higher mean returns. In particular, the rule based on the spread between the EP ratio and a short-term interest rate beats the S&P500 index even when transaction costs are taken into account.

Feng et al. [FRS04] and *Silaghi and Robu* [SR05] evaluate different heuristic trading rules but unfortunately do not compare the results to Buy and Hold.

Feng et al. [FRS04] evaluate stock trading rules in the context of the Penn-Lehman Automated Trading simulator. Two rules suggested by *Ronggang and Stone* [RS03] are used. The first rule is a market-making rule exploiting market volatility without predicting the direction of the stock price movement. The second rule is a reverse rule based on technical analysis. Both rules trade the Microsoft Corp. (MSFT) asset over 15 days from February 24, 2003 to March 18, 2003. The market-making rule fixes a selling price x and a buying price y for MSFT. When prices go beyond x a sell order is placed and when prices drop on y a buy order is placed. The reverse rule sells when prices tend to move upwards and buys when prices tend to move downwards. The experimental analysis is designed as a tournament with three rounds, each lasting one week. Both rules survived the first round; the market-making rule did not survive the second round. The reverse rule won the tournament but without achieving any profit.

Silaghi and Robu [SR05] compare traditional price-based rules to rules based on order book information. Tested rules are called Static Order Book Imbalance (SOBI), Volume Average Weighted Prices (VWAP), Trend Following (TF) and Reverse Policy (RP). SOBI buys (sells) if order book sell prices are greater (smaller) than the order book buy prices. VWAP buys (sells) if the markets average buying (selling) prices are greater (smaller) than VWAP buying (selling) prices. TF calculates a long and a short trend line from ticker prices and buys (sells) if slopes of long (short) and short (long) match (both negative / positive). The fourth rule implemented is the reverse rule discussed by *Feng et al.* [FRS04]. All four rules were tested over a 15-

day period from January 5, 2004 to January 23, 2004 with NASDAQ order book data. Three mixed policies which combine two, three or all of the four rules were considered: SOBI+VWAP+RP+TF, SOBI+RP and SOBI+RP+TF. Results compare achieved returns and the Sharpe ratio. For a period length of 15 days the best combined rule is SOBI+RP+TF in terms of achieved return; the reverse rule is the overall winner in terms of the Sharpe ratio.

Chavarnakul and Enke [CE08] compare combinations of moving averages to rules based on individual moving averages and to Buy and Hold. The trading rules are based on the Volume Adjusted Moving Average (VAMA) and the Ease of Movement (EMV) indicators. VAMA is a moving average, where prices are replaced by volume. EMV illustrates the relationship between the rate of price and volume change of an asset. Trading is simulated over a time horizon of 1508 days from January 1998 to December 2003. At each point of time only one asset of the S&P500 index is in the portfolio. Different types of period lengths are investigated: 1 week (5-days), 4 weeks (21-days) and 13 weeks (55-days). Trading signals are generated by VAMA and EMV with and without the use of a Neural Network (NN). Transaction costs are not considered. The VAMA rule buys if the price of the asset is smaller than the VAMA and sells if the price is greater. The EMV trading rule buys when the smoothing value of EMV crosses above zero from below and sells when the smoothing value of EMV crosses below zero from above. Trading rules might not be executed depending on the results of the NN which predict the next day's VAMA and EMV. Different combinations of trading rules are tested. VAMA+NN, VAMA+NN+Filter, VAMA+NN+SMA, and EMV+NN+VAMA. Benchmarks are VAMA, EMV, a Single Moving Average (SMA) and Buy and Hold. Results show that trading with NN support is helpful to generate better trading decisions. The combined rule EMV+NN+VAMA outperforms all benchmarks in terms of average returns.

4 Experiments

Our experiments are based on the DAX 30 index for the time interval 01-01-1998 to 12-31-2007. We excluded weekends from this interval resulting in 260 days for each year. Trading is carried out by exchanging cash into the index (buying) and by exchanging the index back into cash (selling). No other assets than cash and index are considered for trading.

For the multiple trade problem we divide the time horizon into several trading periods of different length. Each trading period of length K consists of two sub-periods T_u = $\lceil K/2 \rceil$ for buying (buying period u) and T_v = $\lfloor K/2 \rfloor$ for selling (selling period v) with $K = T\_u + T\_v$. We differ between trading periods with 260, 130, 65, 20, and 10 days: one year (K=260 days, 130 days for buying (selling)), six months (130 / 65 (65)), three months (65 / 33 (32)), one month (20 / 10 (10)), and two weeks (10 / 5 (5)). With this arrangement we exclude trading on weekends but other country-specific non-trading days are not excluded. E.g. in 2007 we have only 252 trading days for the K = 260 days period. For the 130 buying days we have only T_u = 127 days where buying is possible and for the 130 selling days we have only T_v = 125 days where selling is possible. The number of possible trading days in each period is always smaller or equal than the period length. When we do not need to differ between buying and selling periods we denote the number of days in one period simply by T.

Within each buying period we must exchange all cash into the index and within each selling period we must exchange all money invested in the index back into cash. At the end of the last day of each buying (selling) period all cash (index) has been exchanged into the index (cash). We assume that for each buying (selling) period there are precise estimates of the possible maximum price M and the possible minimum price m.

In our experiments we investigate the following five trading algorithms:

(1) Optimal Trading

Optimal Trading (*OPT*) is an offline algorithm which achieves the best possible return in each trading period. It is assumed that *OPT* knows all prices of a period. In each buying period u *OPT* will buy at the minimum realized price $p_{min} \geq m(u)$ and will sell in each selling period v at the maximum realized price $p_{max} \leq M(v)$. *OPT* carries out only two transactions in every trading period.

(2) Threat-based Trading

At every time an exchange is carried out the threat-based algorithm (*Threat*) calculates the achievable competitive ratio for each period and buys or sells the corresponding quantities such that the achievable competitive ratio is also realized. There might be as many transactions as there are days T in a trading period.

In our implementation of the algorithm we must ensure that the competitive ratio for each period is never smaller than one and that not more than the available asset values are traded.

(3) Reservation Price Trading

For every period the reservation price algorithm (*Square*) calculates reservation prices $RP(t)$ for each day t. In case *Square* has to buy (sell) the index the first price $q(t)$ with $q(t) \leq (\geq) RP(t)$ is accepted for buying (selling). If there was no such price then buying (selling) has to be done on the last day T of a period. There are only two transactions in every trading period.

(4) Average Price Trading

The average price algorithm (*Constant*) buys (sells) the index with the constant fraction $1/T\_u$ ($1/T\_v$) in every buying (selling) period. There are $T\_u + T\_v$ transactions in every trading period.

(5) Buy and Hold

Buy and Hold (*BH*) buys the index on the first day of the trading period and sells it on the last day, i.e. it is invested in the index from the first day of the buying period until the last day of the following selling period. There are only two transactions in every trading period.

The following assumptions apply for all tested algorithms.

1. There is an initial cash value greater zero.
2. Possible transaction prices are daily closing prices.
3. In each buying period all cash is exchanged into the index and in each selling period the index is exchanged into cash completely.
4. Transaction costs are not considered.
5. Minimum price *m*, maximum price *M*, and the lengths *T_u* of each buying and *T_v* of each selling period are known.
6. Interest rate on cash is assumed to be zero.

The performance measures of the algorithms are the annualized geometric return (*GR*) and the average trading period return (*AR*). Let d_i and D_i be the amount of cash at the beginning and at the end of a trading period *i*. Return r_i generated in a trading period *i* is calculated according to

$$r_i = D_i / d_i \tag{4-1}$$

Let *n* be the number of trading periods considered. The geometric return rate is based on the assumption that we reinvest the portfolio of each trading period *i* completely for trading in the next period i+1, i = 1, …, n-1, until the end of the investment horizon. If the investment horizon is $h \geq 1$ year *GR* is calculated according to

$$GR(n) = (\prod_{i=1,\ldots,n} r_i)^{(1/h)} \tag{4-2}$$

The geometric return tells us which annualized performance the algorithms could achieve in the investment horizon. The average period return assumes that we only invest in a trading period of given same length and averages the result over all trading periods of the same length.

$$AR(n) = (\prod_{i=1,\ldots,n} r_i)^{(1/n)} \tag{4-3}$$

The average period return tells us which average performance we could expect within a trading period of given length.

We also calculate the worst case competitive ratio and the empirical case competitive ratio. The competitive ratios are calculated according to

$$c \geq OPT(I) \,/\, ON(I) \text{ where } ON \in \{Threat, Square, Constant, BH\} \tag{4-4}$$

Let c_w be the worst case competitive ratio and let c_e be the empirical case competitive ratio. For the worst case competitive ratio $ON(I)$ is the worst case return which could have been achieved taking the data of the problem instance into account; for the empirical case competitive ratio $ON(I)$ is the empirical case return which actually was achieved by an online algorithm and is calculated according to (4-2) and (4-3).

Worst case competitive ratios we only consider for algorithms *Threat* and S*quare*. For *Threat* we use its empirical case competitive ratio as its worst case competitive ratio because the empirical ratio can be achieved also in the worst case. Thus, c_w of *Threat* is the same as its c_e and it is calculated according to (2-3).

For *Square* we must calculate the worst case return; let $m(u)$ and $M(u)$ be the bounds for the buying period u and let $m(v)$ and $M(v)$ be the bounds for the selling period v; the worst case competitive ratio for buying is $m(u)$/SQRT[$m(u)\cdot M(u)$] = SQRT[$m(u)/M(u)$] and for selling is $M(v)$/SQRT[$m(v)\cdot M(v)$] = SQRT[$M(v)/m(v)$]. For the entire trading period we get a worst case competitive ratio c_w = SQRT[$M(u)\cdot M(v)/m(u)\cdot m(v)$].

In order to find out how *Threat* and *Square* behave relative to each other in the empirical and in the worst case we calculate for the empirical case the ratio of the achieved returns by *Threat* and *Square*.

$$GR_{Threat}\,(n) \,/\, GR_{Square}\,(n) \tag{4-5}$$

$$AR_{Threat}\,(n) \,/\, AR_{Square}\,(n) \tag{4-6}$$

For the worst case we want to know the ratio of the worst case return of *Threat* and the worst case return of *Square*, i.e. worst case returns *Threat* / *Square*. As $c(Square) = OPT(I) \,/\, Square(I)$ and $c(Threat) = OPT(I) \,/\, Threat(I)$ we can calculate

$$\begin{aligned} c(Square) \,/\, c(Threat) &= (OPT(I) \,/\, Square(I)) \,/\, (OPT(\mathrm{I}) \,/\, Threat(I)) \\ &= Threat(I) \,/\, Square(I) \end{aligned} \tag{4-7}$$

to find the worst case ratio of the returns where $Threat(I)$ and $Square\,(I)$ relate to worst case performances of both algorithms.

5 Experimental Results

Clearly, all online algorithms (2) - (5) cannot beat the benchmark algorithm *OPT*. We carried out simulation runs in order to find out how the following measures compare:

(1) the empirical performance of the algorithms,

(2) the empirical case competitive ratios found in the experiments,
(3) the worst case competitive ratios which could have been possible from the experimental data,
(4) the performance ratios *Threat* / *Square* in the empirical case and in the worst case.

We answer these questions using the DAX 30 index data for the 10 year interval [1998, 2007]. We conducted experiments for the whole interval and for each year of the interval. We calculated annualized geometric returns and we calculated average period returns. Clearly, the answers generated from the interval data must basically be the same as these generated from the yearly data. Therefore we only report on results based on the average period returns from the interval data in detail.

Question 1:
How does the empirical performance of the algorithms compare?

We calculated the experimental performance of the four online algorithms *Threat*, *Square*, *BH*, and *Constant* and compared it to *OPT* according to (4-2) and (4-3). The results are presented in Table 1. *Threat* dominates all other online algorithms. *Square* dominates *BH* and *Constant*. *Constant* is dominated by all other online algorithms except for the 65 days trading period; here *Constant* has a better performance than *BH*. In most cases we see that the longer the trading periods the better the performance of *OPT*, *Threat*, *Square*, and *BH*.

1998-2007	**Empirical case: Average period return**				
Period Length	**10 days**	**20 days**	**65 days**	**130 days**	**260 days**
OPT	1.0308	1.0562	1.1320	1.2110	1.2923
Threat	1.0236	1.0376	1.0807	1.0981	1.1636
Square	1.0218	1.0302	1.0602	1.0528	1.1220
BH	1.0024	1.0050	1.0137	1.0242	1.0568
Constant	1.0005	1.0028	1.0154	1.0099	0.9930

Table 1: *Average period returns for the interval 1998-2007*

If we take annualized geometric return (cf. (4-2)) into account we could see that all algorithms generate better returns for more (shorter) trading periods. This observation can be generalized to such algorithms which generate positive period returns. We conclude that in our experiments it is better to have *more* trading periods than *longer* ones applying algorithms generating positive returns. To answer this question more generally we have to compare r_l^x and r_m^y where r_l (r_m) is the average return of a period with length l (m) and x (y) is the number of trading periods of length l (m) for the whole trading horizon.

Question 2:
How do the empirical case competitive ratios found in the experiments compare?

Clearly, the answers to Question 1 regarding the relative performance comparison of the algorithms are also true for Question 2 because the numerator in (4-4) is constant for all algorithms in each period. We calculated the numerical values of the empirical competitive ratios achieved by all algorithms according to (4-4). The results are shown in Table 2. The shorter the trading period length the better is the empirical case competitive ratio of the algorithms, i.e. the algorithms loose performance compared to *OPT* the longer the periods are. Taking the results for the annualized geometric return into account all algorithms loose performance compared to *OPT* the more trading periods are considered.

1998-2007	**Empirical case: Competitive ratio average period return**				
Period Length	**10 days**	**20 days**	**65 days**	**130 days**	**260 days**
OPT/Threat	1.0070	1.0179	1.0475	1.1028	1.1106
OPT/Square	1.0088	1.0252	1.0677	1.1504	1.1517
OPT/BH	1.0283	1.0509	1.1167	1.1824	1.2228
OPT/Constant	1.0302	1.0532	1.1148	1.1991	1.3014

Table 2: *Empirical case competitive ratios for the interval 1998-2007*

Question 3:
How do the worst case competitive ratios which could have been possible from the experimental data compare?

We calculated the worst case competitive ratios for *Threat* and *Square* which are possible from the experimental data set. The results are shown in Table 3. Using the worst case criteria *Threat* clearly outperforms *Square*, i.e. if we like to minimize worst case returns we choose *Threat*. Moreover the performance of *Square* gets worse compared to *Threat* the longer the trading periods are.

1998-2007	**Worst case: Competitive ratio average period return**				
Period Length	**10 days**	**20 days**	**65 days**	**130 days**	**260 days**
OPT/Threat	1.0070	1.0179	1.0475	1.1028	1.1106
OPT/Square	1.0302	1.0529	1.1109	1.1962	1.2913

Table 3: *Worst case competitive ratios for the interval 1998-2007*

Question 4:
What are the performance ratios Threat / Square in the empirical case and in the worst case?

Comparing *Threat* and *Square* by their worst case competitive ratio we know that *Threat* outperforms *Square* (Table 3). This is also true for the empirical case competitive ratio we found in the experiments (Table 2). Answering Question 4 we want to know how the ratios of the worst case and of the empirical case differ, i.e. in which case the out-performance is greater. The answer is given in Table 4. Using average period return as performance measure the ratio is between 2.3% and 16.3% in the worst case and only between 0.18% and 4.31% in the experiments. So we conclude that trading with *Square* is a good alternative to *Threat* in practical applications especially if we want to reduce the number of transactions which are generated by *Threat*.

1998-2007	**Empirical and worst case ratio average period return: *Threat* / *Square***				
Period Length	**10 days**	**20 days**	**65 days**	**130 days**	**260 days**
Empirical Case	1.0018	1.0072	1.0193	1.0431	1.0370
Worst Case	1.0230	1.0343	1.0605	1.0847	1.1627

Table 4: *Empirical case versus worst case ratio*

Question 5:
Can the answers to Questions 1 and 2 be confirmed by a statistical t-test?

We use a student *t*-test to test for significance. The following input data is used:

- For trading algorithms *Threat*, *Square*, and *Constant* all period returns are used; e.g. for period length 10 days within the 10 year interval 260 period returns are generated, for period length 20 days 130 period returns are generated, etc.
- For *BH* the same number of period returns are calculated using daily returns q_t / q_{t-1} with $t > 1$.

The *t*-test generates useful output if the sample size (number of period returns) is greater 30 or the period returns are normally distributed. To test for normality we use the Jarque-Bera (JB) test. The null hypothesis is that the period returns of each algorithm and each trading period length are normally distributed, i.e. for four algorithms and five different period lengths we conduct 20 JB tests. The JB test tests the normality of large samples using both skewness and kurtosis measures, since samples from a normal distribution have an expected skewness and an expected kurtosis of 0, cf. *Known and Kish* [KK02], *Gunasekarage and Power* [GP01]. Results of the JB test are shown in Table 5. The “yes” entries mean that the null hypothesis cannot be rejected; the “no” entries mean that the null hypothesis could be rejected, i.e. the period returns are not normally distributed.

1998-2007	**Jarque-Bera Test for Normal Distribution**				
Period Length	10 days	20 days	65 days	130 days	260 days
Threat	no	no	yes	yes	yes
Square	no	no	yes	yes	yes
BH	no	no	no	no	no
Constant	no	no	yes	yes	yes

Table 5: *Jarque-Bera Test for Normality Results*

We use a *t*-test to test the algorithms against each other. The null hypothesis is that the 10 year average of the period returns of one algorithm A_1 is less or equal ($\leq$) than these of another algorithm A_2. Before running a t-test we have to check if the period returns of the compared two algorithms (t-test samples) have equal variances or not. If data is normally distributed, the Bartlett test is used to test the variances; if not we use the Levene test (*Brock et al.* [BLL92], *Known and Kish* [KK02], *Mills* [Mil98]). Both tests test the null hypothesis that the variances across the t-test samples are equal against the alternative that at least two variances are different. Equal variance across t-test samples is called homoscedastic or homogeneity of variances; non-equality is called heteroskedastic. Depending on the results for the variances different kinds of t-tests are used.

The t-test statistics are calculated for the 10 year interval depending on the results of the normality test and the variance equality test for the algorithms. We use a significance level of 5 percent. The following empirical findings are tested for each trading period length, i.e. for each pair of algorithms five t-tests for each null hypothesis (10, 20, 65, 130, 260 days). As we test six pairs of algorithms 30 t-tests were conducted. Sample sizes for each trading period refer to the number of trading periods in the interval 01-01-1998 to 12-31-2007, i.e. for trading period length of 10 days we have a sample of 260 returns, for trading period length 20 we have a sample of 130 returns, etc. Results are shown in Table 6. The lower the p-value, the more "significant" is the result of the t-test concerning the rejection of H_0. The “no proof” entries mean that the null hypothesis H_0 cannot be rejected, i.e. we cannot prove that $A_1 > A_2$; the “true” entries mean that the null hypothesis H_0 could be rejected, i.e. $A_1 > A_2$ is true.

Period Length	**10 days**	**20 days**	**65 days**	**130 days**	**260 days**
(1)	H_0: ***Threat*** $\leq$ ***Square***				
Threat* > *Square	no proof	no proof	no proof	no proof	no proof
***p*-value**	24.55%	6.99%	9.49%	6.70%	27.77%
(2)	H_0: ***Threat*** $\leq$ ***BH***				
Threat* > *BH	true	true	true	true	true
***p*-value**	0.00%	0.00%	0.00%	0.00%	0.41%
(3)	H_0: ***Threat*** $\leq$ ***Constant***				
Threat* > *Constant	true	true	true	true	true
***p*-value**	0.00%	0.00%	0.01%	0.19%	1.92%
(4)	H_0: ***Square*** $\leq$ ***BH***				
Square* > *BH	true	true	true	true	true
***p*-value**	0.00%	0.00%	0.00%	0.81%	1.19%
(5)	H_0: ***Square*** $\leq$ ***Constant***				
Square* > *Constant	true	true	true	no proof	no proof
***p*-value**	0.00%	0.00%	0.27%	8.57%	5.26%
(6)	H_0: ***BH*** $\leq$ ***Constant***				
BH* > *Constant	no proof	no proof	no proof	no proof	no proof
***p*-value**	65.64%	85.23%	95.31%	79.94%	56.96%

Table 6: *Comparing the Trading Algorithms*

When testing the null hypothesis that the 10 year average of the period returns of *Constant* is less or equal than the 10 year average of the period returns of BH (H_0: *Constant* ≤ *BH*) the null hypothesis cannot be rejected in four cases (10, 20, 130, 260 days). For 65 days the *t*-test result is highly significant, rejecting the null hypothesis with *p*-value 4.69%.

6 Conclusions

In order to answer the five questions raised in this paper simulation runs with different number and lengths of trading periods were performed. We assumed the precise values for *m*, *M*, and *T* to be known. The answers to Questions 1, 2, and 5 are summarized in Table 7. The "no" entries in column "*t*-test" mean that the null hypothesis could not be rejected; the "yes" entry means that the null hypothesis could not be rejected for two period lengths.

10 Year Interval 1998-2007			
	Average Period Return	**Simulation**	***t*-test**
(1)	*Threat* dominates *Square*	yes	no
(2)	*Threat* dominates *BH*	yes	yes
(3)	*Threat* dominates *Constant*	yes	yes
(4)	*Square* dominates *BH*	yes	yes
(5)	*Square* dominates *Constant*	yes	(yes)
(6)	*BH* dominates *Constant*	yes	no

Table 7: *Summary of simulation and t-test results*

The table shows that the results found in the simulation runs could be confirmed clearly in three cases and weakly in one case. This is not only true for average period returns but also for the corresponding competitive ratio. Where the results from the simulation runs cannot be confirmed by the *t*-test the return values generated by the two algorithms are too close to produce significance. Since the *t*-ratios assume normality, stationarity, and time-independent distributions it would be interesting to perform a bootstrap procedure to calculate critical values when dealing with small samples, cf. *Tabak and Lima* [TL09].

The conclusion is that *Threat* clearly outperforms *BH* and *Constant*. This result was achieved without considering transaction costs. If transaction costs have to be considered *Threat* still outperforms *Constant* because it never generates more transactions. If we want to reduce transaction costs experimental results show that *Square* is a good alternative to *Threat*, i.e. it also outperforms *BH*. The worst performance found in the simulation is achieved by *Constant. BH* looses performance relative to *Threat* and *Square* the shorter the periods are. For the worst case ratio average period return values are increasing the longer the periods are. The worst case performance is the greater the greater the difference in *m* and *M* which gets greater with longer periods.

One might argue that the comparison of *Threat* and *Square* with *BH* and *Constant* is not appropriate because the first two algorithms use information about future prices while the latter two do not. It is also to be expected that algorithms which use more information should perform better than these which do not. Note, that *BH* and *Constant* could also be modified such that they would use information about m and M. But in reality no algorithm can rely on the correctness of information about future prices and forecasts for m and M. A suitable procedure for estimating m and M is an important factor to provide a good online algorithm. It also would be of interest to assume that we do not have forecasts for m and M. One approach is to observe a certain number k of the T prices within a time horizon with $k < T$ and then trade to the next best price $q(t) >$ max (<min) $\{q(t) \mid t = 1, \ldots, k\}$ (cf. the approach to the secretary's problem in *Freeman* [Fre83] and *Ferguson* [Fer89]).

It also would be interesting to analyse the performance of *Threat* compared to *Square* and *BH* in further experiments taking transaction costs into account (possibly including discounting the payments). Moreover one could investigate how *Threat* performs in comparison to other popular trading rules like *Moving Average* and *Trading Range Breakout*.

References

[BLL92]
Brock, W.A, Lakonishok, J., LeBaron, B., Simple Technical Trading Rules and the Stochastic Properties of Stock Returns, *Journal of Finance* 47, 1731-1764, 1992

[CE08]
Chavarnakul, T., Enke, D., Intelligent technical analysis based equivolume charting for stock trading using neural networks, *Expert Systems and Applications* 34, 1004-1017, 2008

[CLT04]
Chang, E.J., Lima, E.J.A., Tabak, B.M., Testing for predictability in emerging equity markets, *Emerging Markets Review* 5 (3), 295-316, 2004

[FRS04]
Feng, Y., Ronggang, Y., Stone, P., Two stock trading agents: Market making and technical analysis, in: Faratin, P., Parkes, D., Rodriguez-Aguilar, J., Walsh, W. (eds.), *Lecture Notes in Artificial Intelligence*, Agent Mediated Electronic Commerce V: Designing Mechanisms and Systems, 18-36, Springer 2004

[Fre83]
Freeman, P.R., The Secretary Problem and Its Extensions: A Review, *International Statistical Review* 51 (2), 189-206, 1983

[Fer98]
Ferguson, T.S., Who solved the secretary problem? *Statistical Science* 4, 282-296, 1989

[GP01]

Gunasekarage, A., Power, D.M., the profitability of moving average trading rules in South Asian stock markets, *Emerging Markets Review* 2, 17-33, 2001

[Kar98]
Karlin, A., On the Performance of Competitive Algorithms in Practice, in: Fiat, A., Woeginger, G.J. (eds.), *Online Algorithms – The State of the Art*, Lecture Notes in Computer Science 1442, 373-384, Springer 1998

[KK02]
Kwon, K.-Y., Kish, R.J., A Comparative Study of Technical Trading Strategies and Return Predictability: An Extension of Brock, Lakonishok, and LeBaron (1992) Using NYSE and NASDAQ Indices, *Quarterly Review of Economics and Finance* 42 (3), 613-633, 2002

[Mil98]
Mills, T.C., Technical analysis and the London Stock Exchange: testing trading rules using the FT30, *International Journal of Finance & Economics* 2 (4), 319-331, 1998

[MS08]
Mohr, E., Schmidt, G., Empirical Analysis of an Online Algorithm for Multiple Trading Problems, in: Hoai An, Le Thi; Bouvry, Pascal; Pham Dinh, Tao (Eds.), *Modelling, Computation and Optimization in Information Systems and Management Sciences* 14, 299-308, Springer 2008

[RL99]
Ratner, M., Leal, R.P., Tests of technical trading strategies in the emerging equity markets of Latin America and Asia, *Journal of Banking and Finance* 23, 1887-1905, 1999

[RS03]
Ronggang, Y., Stone, P., Performance analysis of a counter-intuitive automated stock trading stategy, in: Proceedings of the 5th International Conference on Electronic Commerce, *ACM International Conference Proceedings* 50, 40-46, 2003

[She03]
Shen, P., Market-timing strategies that worked, *Journal of Portfolio Management* 29(2), 57-68, 2003

[SR05]
Silaghi, G., Robu, V., An agent policy for automated stock market trading combining price and order book information, in: *ICSC Congress on Computational Intelligence Methods and Applications*, 4-7, 2005

[TL09]
Tabak, B.M., Lima, E.J.A., Market efficiency of Brazilian exchange rate: Evidence from variance ratio statistics and technical trading rules, *European Journal of Operational Research* 194, 814-820, 2009

[YFKT92]
El-Yaniv, R., Fiat, A., Karp, R., Turpin, G., Competitive analysis of financial games, *Technical Report* TR-92-064, 1992

[YFKT01]
El-Yaniv, R., Fiat, A., Karp, R., Turpin, G., Optimal search and one-way trading algorithm, *Algorithmica* 30, 101-139, 2001

[Yan98]
El-Yaniv, R. (1998), Competitive solutions for online financial problems, *ACM Computing Surveys* 30(1), 28-69, 1998