Metaheuristic Optimisation of Parameterised Betting Exchange Strategies

P.Tsirimpas* and W.J. Knottenbelt**

* *Department of Computing, Imperial College London, London, United Kingdom: pt307@doc.ic.ac.uk*

** *Department of Computing, Imperial College London, London, United Kingdom: wjk@doc.ic.ac.uk*

Abstract. Stochastic optimisation algorithms have been growing rapidly in popularity over the last three decades, with a number of methods now playing a significant role in the analysis, design, and execution of betting strategies. This paper presents the optimisation platform of SPORTSBET, an event-driven tool for the quantitative evaluation of generic betting exchange trading strategies. Strategy parameters are automatically refined using a stochastic search heuristic in order to improve strategy performance. Walk Forward Analysis is employed to avoid overfitting. To demonstrate the applicability and effectiveness of the platform, case studies are presented for betting strategies in horse racing.

1. Introduction

Gambling and mathematics have a long mutual history. Though the ability to analyse and act on high-frequency real time betting markets is relatively new and very challenging. Since their introduction in June 2000, betting exchanges have revolutionised the nature and practice of betting. Betting exchange markets share some similarities with financial markets in terms of their operation. However, in stark contrast to financial markets, there are very few quantitative analysis tools available to support the development of automated betting exchange trading strategies.

SPORTSBET (Specification and Performance Optimisation of Real-time Trading Strategies for Betting Exchange plaTforms) (Tsirimpas & Knottenbelt, 2011) is a toolset developed to specify, execute, back-test and optimise parameterised betting strategies for a wide range of sports. SPORTSBET allows the definition of betting strategies in a novel generic betting strategy specification language (UBEL) as sets of concurrent processes which make use of event-calculus-like operators. Strategy performance is quantified by synchronizing multiple real time or historical data streams with a dynamic market reconstruction. The development of a trading strategy is a complex process consisting of a number of different stages such as: formulation, specification in a computer testable form, back-testing, optimisation, evaluation, real time trading, monitoring trading performance and finally refinement and evolution.

This paper confronts challenges related to the back-testing, optimisation and execution of parameterised automated trading strategies for betting exchange markets, and presents the optimisation platform of SPORTSBET. The organisation of the rest of this paper is as follows. Section 2 explains the back-testing and optimisation process in general. Section 3 explores the practical impacts the different types of search and evaluation methods have upon the outcome and quality of the historical simulation and on the optimisation process and presents the optimisation platform of SPORTSBET. Section 4 presents a case study in horse racing and finally, Section 5 summarises this paper and gives directions to future works.

2. Back-testing and Optimisation preliminaries

Back-testing is a specific type of historical testing that calculates how a strategy would have performed if it had actually been applied in the past. This requires the back-test to replicate the market conditions of the time in question in order to get an accurate result. While back-testing does not allow one to predict how a strategy will perform under future conditions, its primary benefit lies in understanding the vulnerabilities of a strategy as it encountered real world conditions of the past (Wikipedia, 2013). The more accurate the back-testing, the better the real-time trading results are likely to be.

To optimise a trading strategy is to obtain its peak trading performance. Most trading strategies have a set of parameters that highly affect their performance. Any strategy that can accept different values for these parameters is eligible for optimisation. An optimisation algorithm is an algorithmic method that can be applied to solve optimisation problems. Numerous optimisation algorithms are available but choosing one for solving a given optimisation problem depends much on the characteristics of the optimisation problem at hand. Many optimisation methods are especially designed for specific types of search spaces, objective and constraint functions. This work focuses on optimisation methods that are not dependent on any knowledge about the system or model of the optimisation problem. For example: imagine if you are trying to find an optimal set of parameters for a betting exchange strategy. You have a simulator for the betting strategy and can test any given set of parameters and assign it a quality. And you have come up with a definition for what a strategy parameter sets look like in general. But you have no idea what the optimal parameter set is, not even how to go about finding it. These optimisation problems are known as black box optimisation problems (see Figure 1).

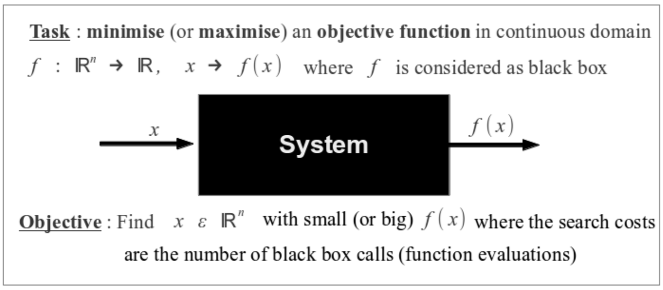

Figure 1Black box optimisation

Optimisation has many drawbacks and this is because there are many ways that it can be done incorrectly. Usually an optimisation done incorrectly is overfitted. Overfitting occurs when the optimisation process identifies parameters that produces good trading performance on historical data but produces poor trading performance on unseen data. This is because someone can always find a combination of rules and trading parameters that fits perfectly to the available historical data, resulting in exceptional trading results based on those tests. However, when those rules are tested on a live market, they can fail and lose money very quickly. There are also degrees of overfitting. A trading strategy, where the degree of overfitting is not extreme, can still produce real-time profit but it will certainly underperform its optimisation results. On the other hand, a highly overfitted trading strategy will produce disastrous real-time trading losses. Thus to avoid overfitting, it is essential to further test any strategy with the optimized parameters on a set of historical data that is distinct from that used in the optimisation process. Some of the most well known and used techniques are the K-fold Cross Validation (Rodriguez, et al., 2009), Regularization (Gencay & Qi, 2001) and Walk-Forward Analysis (Pardo, 2008). In SPORTSBET optimisation platform Walk-Forward Analysis is employed to avoid overfitting.

3. SPORTSBET Optimisation Platform

During the optimisation process, a historical simulation (back-testing) will be calculated for a large number of different values of the key strategy parameters. In order to do that, all optimisation processes use some type of search method. The methods adopted in stochastic optimisation attempt to model the uncertainty in data by assuming that the input is specified in terms of a probability distribution. Metaheuristics are the most general of these kinds of algorithms, and can be applied to a wide range of problems. The search method will determine the number of back-tests to be performed and therefore the amount of processing time required to complete the process. Moreover, the search method will guide the search in productive directions. However, the directed search methods have some drawbacks. Since a direct search method does not evaluate every possible candidate solution, there is a potential for a certain lack of thoroughness. This can be minimised by selection of the appropriate search method.

In order to retrieve from the optimisation process the trading strategy parameters that are most likely going to produce real-time and long term trading profits, we need to understand the impact of the objective function. The objective function is used during the optimisation process to assign a score in each candidate solution.

3.1 SPORTSBET search method

Evolution strategies (ESs) are robust stochastic search algorithms designed to minimize objective functions f that map a continuous search space $\Re^n$ into $\Re$. An Evolution Strategy is broadly based on the principle of biological evolution. In each generation (iteration) new candidate solutions (denoted as x) are generated by variation, usually in a stochastic way, and then some individuals are selected for the next generation based on their objective function value $f(x)$. Like this, over the generation sequence, individuals with better and better $f(x)$ are generated. An Evolution Strategy follows these steps:

- **Initialisation**: The initial population can be based on known good solutions or can be generated randomly.
- **Evaluation**: The evaluation uses the objective and constraint functions of the optimisation problem to assign a quality score to each individual. Individuals with a higher score will have a higher probability of surviving and passing on their genetic material (i.e., the candidate solution) to future generations.
- **Selection**: There are two types of selection, the parental selection and the survivor selection. Parental selection is a stochastic selection type that selects the parents that are used for the recombination of a new offspring. In this selection type, the fitter individuals have a higher probability to be selected as parent for recombination. Survivor selection is a deterministic selection that selects the μ fittest individuals either out of the λ offspring (elitist selection) or out of the λ offspring and the μ old parents (non-elitist). This selection type is commonly referred as $(\mu + \lambda)$ when denoting elitist selection, and as (μ, λ) when denoting non-elitist selection.
- **Mutation and recombination**: Mutation operators add small perturbations to the individuals in the population. Recombination operators recombine two or more individuals in the population into a new individual.
- **Termination**: The termination condition can depend on the available computation time the available number of evaluations/generations, or on convergence criteria such as a predefined target fitness that is to be reached. After termination, the best solution(s) is found throughout the evolution cycle.

The simplest Evolution Strategy is the (1+1)-ES (one parent, one offspring) (Hoffmeister & Back, 1991). SPORTSBET optimisation platform uses the Covariance Matrix Adaptation Evolution Strategy (CMA- ES) (Hansen, 2006). The CMA-ES is a $(\mu / \mu_W, \lambda)$-ES in which all offspring are generated from the same recombinant, computed as the weighted centre of mass of the μ selected individuals. For more details see (Beyer & Sendho, 2008).

3.2 Objective Function

A search method is continually accepting or rejecting trading strategies in the process of seeking the best parameter set in the least time possible. Thus, it is critical to use a proper objective function, which correctly characterises the quality of a trading strategy. As an example if the optimisation process is using as objective function the highest net profit, caution must be exercised in the event that a large proportion of the profit arises from a single large and likely unrepeatable trade with a favourable outcome. Furthermore, using the highest net profit in isolation completely ignores the question of risk. The strategy with the highest net profit could also have a very large and unacceptable drawdown. Or, the strategy parameters selected may have a very small number of trades, which brings into question the statistical validity of these parameters. All of these criteria are very crucial and cannot be ignored when building the objective function.

SPORTSBET's back-testing process returns as its result an array of values representing the evaluation of several objective functions, which the user may use as they are, or combine them to produce a new one. The array contains:

- **Profit**: the difference between the winning and the losing trades.
- **Profit after commission**: the difference between the winning and the losing trades including commission.

- **Maximum Drawdown (MDD)**: measures the largest single drop from peak to bottom in an account balance during the life of a strategy.

$$MDD = PV - LV, \qquad MDD(\%) = MDD/PV \tag{1}$$

where PV is the peak value before the largest drop and LV is the lowest value before new high established.

- **Maximum Run Up (MRU)**: measures the largest single increased from bottom to peak in an account balance during the life of a strategy.

$$MRU = PV - LV, \qquad MRU(\%) = MRU/LV \tag{2}$$

where LV is the lowest account before the largest increased and PV is the highest value before new lowest established.

- **Return On Investment (ROI)**: evaluate the efficiency of an investment. Formula:

$$ROI(\%) = Profit\ after\ commission/\ Total\ Investment \tag{3}$$

- **Risk-Adjusted Rate of Return (RAR)**: measures the amount of risk involved in an investment's return.

$$RAR(\%) = Net\ profit\ after\ commission/(Risk + Initial\ Account),\ \ where\ Risk = 2 \times MDD \tag{4}$$

- **Reward to Risk Ratio (RRR)**: provides an easy comparison of reward to risk.

$$RRR(\%) = Net\ profit\ after\ commission/MDD \tag{5}$$

- **Perfect Profit**: it is the total profit produced if the strategy was winning the highest net profit in each market during the historical period.
- **Number of winning trades**: the number of winning trades. A winning trade is the one where the net profit in a market is positive.
- **Number of losing trades**: the number of losing trades. A losing trade is the one where the net profit in a market is negative.
- **Pessimistic return on margin (PROM)**: a measure that pessimistically assumes that a trading strategy will win less and lose more in real-time trading than it did in its historical simulation.

$$PROM(\%) = \left(\left[\bar{W} \times \left(WT - \sqrt{WT}\right)\right] - [\bar{L} \times (LT - \sqrt{LT})]\right)/margin \tag{6}$$

where $\bar{W}$ is the average winnings, $\bar{L}$ is the average losses, *WT* the number of winning trades, *LT* the number of losing trades, and margin the initial account balance. *PROM* is a robust measure because it takes in account a number of significant performance statistics as the ones mentioned above. Moreover, *PROM* penalises the small trade samples because of the adjustment of gross profit and loss by the square root of their respective number.

- **Strategy Efficiency (SE)**: measures how efficiently a trading strategy converts the perfect potential profit into realised trading profits.

$$SE(\%) = Net\ profit\ after\ commission/Perfect\ profit \tag{7}$$

3.3 Walk-Forward Analysis

The primary purpose of a Walk-Forward Analysis is to determine the consistency of a trading strategy's performance. This can be achieved by judging the performance of a trading system exclusively on data which were never part of the optimisation process which is a far more reliable measure than performance based only on in-sample simulation. The automatic Walk-Forward Analysis is a system design and validation technique in which you optimise the strategy parameter values on a past segment of market data ("in-sample"), then verify the performance of the strategy by testing it forward in time on data following the optimisation segment ("out-of-sample"). The evaluation of the trading strategy is based on how well it performs on the test data ("out-of-sample"), not the data it was optimised on. The process is repeated over subsequent time segments (see Figure 2).

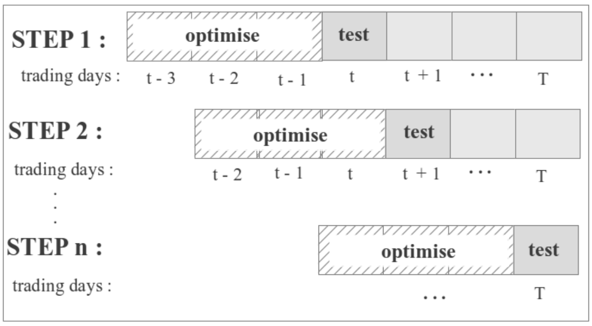

Figure 2 Walk-Forward Analysis

In each step the *Walk-Forward Efficiency (WFE)* is calculated and saved.

$$WFE(\%) = Annualised\ net\ profit\ from\ testing / Annualised\ net\ profit\ from\ optimisation \qquad (8)$$

The average WFE over the Walk-Forward Analysis can be used to provide some estimation of the rate of profit to be earned during real-time trading. Research has clearly demonstrated that robust trading strategies have WFEs greater than 50-60 percent. Finally another thing to take in consideration is the consistency of the trading strategy. Example:

- **Consistency on profits**: 70 percent of the Walk-Forward windows were profitable.
- **Distribution of profits**: no individual time window contributes more than 50 percent.
- **Maximum Drawdown**: no individual time window had a drawdown of more than 40 percent of initial capital.

4. A Horse Racing Case Study

A horse racing case study was used as a means of demonstrating and evaluating the results yielded by the SPORTSBET optimisation platform. The trading strategy is: 1 minute before a race start, back a range of horses $[X_2, X_3]\ where\ X_2 \leq X_3,\ \ X_2 \geq 0,\ \ X_3 \leq 5,\ \ X_2, X_3\ \in Z$ for £40 if the odds of the favourite are more than X_1 where $X_1 \in \Re$, and spread the profit across all the backed horses. So we want to find the best combination of $X_1,\ X_2,\ X_3$. The strategy was tested in 990 markets using 10 time windows. The chosen objective function was the PROM and each time window consisted 99 markets. The step size of the Walk-Forward Analysis was three time windows. Table 1 shows an optimisation sample of the time windows 4, 5 and 6. Table 2 shows the Walk-Forward Efficiency of the strategy in each time window, where *W* is the current optimisation window, I the iteration of the searching algorithm, *Ev.* the number of evaluation(how many times you performed a back-test), CW is the maximum number of consecutive winnings and CL is the maximum number of consecutive losses.

Table 1 Optimisation sample of time windows 4-5-6

W	I.	Ev.	X	Fitness	Prof.After.	MDD	ROI	Profit	MDD(%)	MRU	MRU(%)	Perf.Profit	PROM	RAR	RRR	SE	CW	CL	W	L
6	1	1	[4.6, 3, 5]	25.66	829.58	280	48.23	934.29	28.46	909.58	757.99	5910.23	25.66	32.40	296.28	14.0364	2	6	14	29
6	1	2	[4.6, 1, 5]	14.63	316.85	218.03	18.42	367.21	35.416	471.09	325.95	1595.83	14.63	13.01	145.33	19.855	4	4	27	16
6	1	3	[5.6, 4, 4]	14.82	853.6	280	118.55	928	56.49	1053.6	0	6338.4	14.82	33.34	304.86	13.4671	2	6	4	14
6	1	4	[5.2, 2, 5]	20.98	552.25	120	55.22	604.48	28.01	592.26	370.16	1846.6	20.98	24.65	460.21	29.9066	3	3	14	11
6	1	5	[4.6, 3, 4]	-21.44	7.6	738.8	0.44	88	78.06	826.4	688.66	12288	-21.44	0.21	1.02	0.061849	2	13	5	38
6	1	6	[5, 2, 4]	-1.63	110.34	317.54	9.85	158.25	77.21	376.61	401.77	3339.08	-1.63	4.187	34.74	3.30455	3	6	8	20
6	1	7	[5.7, 4, 5]	5.27	536.6	200	95.82	588	56.56	696.6	1741.5	4681.6	5.27	22.35	268.3	11.4619	1	4	3	11
6	1	8	[6.4, 3, 4]	-1.17	201.2	80	167.67	216	16.62	321.2	200.75	889.2	-1.17	9.31	251.5	22.6271	1	2	1	2
6	1	9	[4.8, 2, 5]	23.49	607.79	159.99	47.48	671.35	34.86	647.79	404.87	2354.48	23.49	26.19	379.89	25.8142	3	4	17	15
6	1	10	[4.9, 3, 4]	-7.42	271.2	440	24.21	336	57.20	609.2	380.75	8126.96	-7.42	9.42	61.63	3.33704	1	11	4	24
6	2	11	[3.7, 2, 5]	-7.58	-64.79	388.21	-1.88	47.58	109.15	339.73	1143.95	6045.35	-7.58	-2.33	-16.69	-1.07184	3	6	31	55
6	2	12	[6, 4, 4]	4.76	503	160	179.64	540	34.63	623	778.75	2428.2	4.76	21.68	314.37	20.7149	1	3	2	5
6	2	13	[4.9, 2, 5]	25.57	657.34	120	58.69	717.2	28.01	697.34	435.84	2054.65	25.57	29.34	547.78	31.9929	3	3	16	12
6	2	14	[5, 2, 4]	-1.63	110.34	317.54	9.85	158.25	77.21	376.61	401.77	3339.08	-1.63	4.19	34.74	3.30455	3	6	8	20
6	2	15	[5,1, 3, 5]	36.13	1050.62	200	93.8	1139.6	19.81	1090.62	681.64	3870	36.13	43.77	525.31	27.1477	3	5	12	16
6	2	16	[5.8, 3, 4]	7.91	582.4	240	121.33	632	35.27	662.4	414	3526.4	7.91	23.48	242.67	16.5154	1	6	3	9
6	2	17	[5,1, 3, 5]	36.13	1050.62	200	93.8	1139.6	19.8	1090.62	681.636	3870	36.13	43.78	525.31	27.1477	3	5	12	16
6	2	18	[5.1, 2, 5]	25.57	657.34	120	58.69	717.2	28	697.34	435.84	2054.65	25.57	29.35	547.78	31.9929	3	3	16	12
6	2	19	[4.7, 2, 4]	-2.02	110.92	330.02	7.92	169.39	65.43	390.92	244.33	4113.06	-2.02	4.17	33.612	2.69688	3	7	10	25
6	2	20	[5.8, 4, 5]	-3.38	303	280	63.12	340	140	583	828.75	3990	-3.38	11.83	108.21	7.59398	1	6	2	10
6	3	21	[5.9, 3, 5]	12.26	478.59	160	99.70	518.52	45.34	518.59	324.12	1645.13	12.26	20.62	299.12	29.0916	2	4	5	7
6	3	22	[5.4, 4, 5]	52.15	1776.46	200	211.48	1899.44	32.01	1856.46	1547.05	7261.26	52.15	74.01	888.23	24.4649	2	4	7	14
6	3	23	[5.1, 2, 5]	25.57	657.34	120	58.69	717.2	28.01	697.34	435.84	2054.65	25.57	29.34	547.78	31.9929	3	3	16	12

Table 2 Walk-Forward Efficiency

Time Window	*X*	*Profit After Commission*	*MDD*	*Winning Percentage*	*WFE*
4	[3.95, 1, 5]	-116.51	236.11	50	-196.39
5	[5.4, 4, 5]	758	80	66.66	128.01
6	[3.4, 3, 5]	-33.85	373.43	22.58	-8.18
7	[3.5, 0, 3]	30.35	200	63.33	13.7
8	[2.6, 0, 4]	442.38	239.6	44	33.3
9	[2.8, 0, 3]	123,29	265.39	39.5	19.4
10	[3.1, 0, 4]	218.43	178.6	42.43	35.36

So we have $\overline{WFE} = 3.6\%$, which means the strategy didn't pass the Walk-Forward Analysis test. A robust strategy must have $\overline{WFE}$ more than 50%. In case the $\overline{WFE}$ is more than 50% it means we have a robust trading strategy. In that case, in order to choose which set of parameters will be used for the real time trading, we use another metric called the Sharpe ratio (Wikipedia, 2013).

4. Conclusion – Future Work

In this paper we have presented a way to find near-optimal parameterisation of betting strategies for betting exchange markets. As illustrated in the case study, the SPORTSBET optimisation platform implements Walk-Forward Analysis for the robust parameterisation of betting exchange trading strategies without overfitting. Future work includes the usage and comparison of different metaheuristics inside the SPORTSBET platform and the automated evolution of entirely new betting strategies.

References

Beyer, H. & Sendho, B., 2008. Covariance Matrix Adaptation Revisited. The CMSA Evolution Strategy. In: *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, pp. 123-132.

Gencay, R. & Qi, M., 2001. Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *Neural Networks, IEEE Transactions,* 12(4), pp. 726-734.

Hansen, N., 2006. The CMA Evolution Strategy: A Comparing Review. In: *Towards a New Evolutionary Computation.* s.l.:Springer Berlin Heidelberg, pp. 75-102.

Hoffmeister, F. & Back, T., 1991. Lecture Notes in Computer Science. In: Heidelberg, ed. *Genetic Algorithms and evolution strategies: Similarities and differences.* Springer Berlin, pp. 455-469.

Pardo, R., 2008. *The Evaluation and Optimization of Trading Strategies.* 2 ed. s.l.:Wiley.

Rodriguez, J., Perez, A. & Lozano, J., 2009. Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions,* 32(3), pp. 569-575.

Tsirimpas, P. & Knottenbelt, W. J., 2011. *SPORTSBET: A Tool for the Quantitative Evaluation and Execution of Betting Exchange Trading Strategies.* Aachen, pp. 155-156.

Wikipedia, 2013. *Backtesting.* [Online] Available at: http://en.wikipedia.org/wiki/Backtesting

Wikipedia, 2013. *Sharp ratio.* [Online] Available at: http://en.wikipedia.org/wiki/Sharpe_ratio