Design of Stock Trading System for Historical Market Data Using Multiobjective Particle Swarm Optimization of Technical Indicators

Antonio C. Briza
Dept. of Computer Science
University of the Philippines-Diliman
Diliman, Quezon City, Philippines
acbriza@up.edu.ph

Prospero C. Naval Jr.
Dept. of Computer Science
University of the Philippines-Diliman
Diliman, Quezon City, Philippines
pcnaval@up.edu.ph

ABSTRACT

Stock traders consider several factors in making decisions. They also differ in the importance they attach to each of these objectives. This requires a tool that can provide an optimal tradeoff among different objectives, a problem aptly solved by a multiobjective optimization (MOO) system. However, the application of MOO to stock trading is very limited when compared with its existing applications in the fields of stock modeling and prediction, portfolio selection and portfolio optimization. Similarly, only a few real life applications have been proposed for multiobjective particle swarm optimization(MOPSO), an MOO algorithm based on particle swarm optimization which has experienced an increased popularity in recent years. In this paper, we present an application of MOO, specifically, of MOPSO, to stock trading. The system, using historical end-of-day market data, utilizes the trading signals from a set of financial technical indicators in order to develop a trading rule which is optimized for two objective functions, namely, Sharpe ratio and percent profit. The performance of the system was compared to the performance of the technical indicators and the market itself. The results show that the system performed well against the 5 technical indicators under study, outperforming them in terms of both objective functions in 3 training and testing periods. The system also performed competitively against the market. The system provided a diversity of solutions for the two objective functions and is found to be robust and fast. These results show the potential of the system as a tool for making stock trading decisions.

Categories and Subject Descriptors

I.2.1 [**Computing Methodologies**]: Artificial Intelligence—*Applications and Expert Systems*

General Terms

Algorithms

Keywords

Multiobjective optimization, particle swarm optimization, stock trading systems, technical indicators

1. INTRODUCTION

Success in trading stocks depends on timing the trades well. For years, stock traders have depended on two major tools: fundamental analysis, which relies on company performance and growth projection, and technical analysis which analyzes the trade history of a security through charts and mathematical formulas called technical indicators.

In recent years, Artificial Intelligence(AI) techniques have also been employed in timing the trades in stock market. AI can be applied in stock trading in two ways. First, as an aid in developing trading agents whose objective is to post buy and sell orders which are processed by an artificial stock exchange. These tools usually utilize intraday (real time) data and are validated by the agent's participation in a simulated stock trading exchange. Second, AI can be used as a tool to develop a trading system whose goal is to give trading signals using historical end-of-day market data. These trading systems are ordinarily validated by testing in an out-of-sample data.

The classification above is justified by the difference between one who is an intraday trader and one who is not. An intraday trader, as the terms imply, buys and sells securities within the day. For this kind of trader the first type of application is more appropriate since it is more attuned to the short term price movements and the trading dynamics that happen during stock trading sessions. But for longer-term traders who maintain their investment positions for weeks or months, the trading dynamics during the trading sessions will not be of much importance. For these traders, the second type of application is more appropriate.

An example of the first type of application is the study of Subramanian et al. [23] who designed agents that are based on composite trading rules trained by Genetic Algorithm (GA) and Genetic Programming (GP). The performance of the agents were evaluated by making them compete with other automated agents in the Penn-Lehman Automated Trading Project [13].

For the second type of application, we can cite the use of GAs to optimize parameters in technical indicators [9, 17] and the use of neural networks and GA to determine buy and sell points in commodities [22].

The above studies have shown significant results. However, it should also be noted that they optimize only one objective function- usually returns or risk. However, a stock trader's decision usually depends on several factors. Moreover, traders give different importance to these factors, depending on their trading personalities, e.g. the level of risk they could take on. This requires a tool that can provide an optimal tradeoff among different objectives. This problem is aptly solved by a Multiobjective Optimization (MOO) system.

Several applications of MOO exist in literature in the areas of stock prediction and modeling [16, 1], portfolio selection and optimization [7, 6, 2]. However, only Fukomoto and Kita's study [10] made use of MOO in timing the entry to the stock market. Their study used a GA-based multiobjective optimization approach to train intraday-trading agents on two objective functions, namely profit ratio and variance of profit. The agents were tested by running them in the U-Mart [15] artificial market simulator. The simulation obtained better trading strategies in terms of the two objective functions they have set.

Fukomoto and Kita's study demonstrates the success of by MOO in training an intraday agent for an artificial market. As mentioned earlier, this type of application is applicable only to intraday traders. The development of a MOO tool for end-of-day market data is therefore necessary if we are to also cater to the needs of the longer-term traders. This is the problem that we would like to address.

Specifically, we present a stock trading system that uses multiobjective particle swarm optimization (MOPSO) of financial technical indicators. Using end-of-day market data, the system optimizes the weights of several technical indicators over two objective functions, namely, percent profit and Sharpe ratio.

This study is significant in three respects. First, it demonstrates the effectiveness of MOO in analyzing end-of-day market data, thus providing long-term traders with a MOO tool that can be used in stock trading. Second, in using MOPSO as the tool for carrying out MOO, we present one of the few real world applications of MOPSO in literature. This is significant especially if we consider that the growing number of variants of the MOPSO algorithm demands a validation of MOPSO's effectiveness in solving real life problems, something pointed out by Reyes-Sierra and Coello Coello in their survey paper on MOPSO [21]. Third, the choice of a MOO tool that is based on Particle Swarm Optimization(PSO) rather than one based on GA can be considered as an improvement over Fukomoto and Kita's study. We can say this because it has been observed that PSO is more appropriate for optimizing continuous variables; moreover PSO is more computationally efficient than GA [11] while being at par with it in terms of the quality of solutions generated. This is significant since computation time is an important factor in designing a stock trading system or agent. Subramanian et al., for instance, mentioned that improving the computation time was a potential challenge in their paper.

The results show that the system performed well against the 5 technical indicators under study, beating them in terms of both objective functions in three training and testing periods. The system also performed competitively against the market.The system provided a diversity of solutions for the two objective functions and is found to be robust and fast. These results show the potential of the system as a tool for making stock trading decisions.

The remainder of this paper is organized as follows. In Section 2 we give a brief background of Multiobjective optimization and MOPSO. This is followed by a discussion of the proposed stock trading system in Section 3. The results are presented and discussed in Section 4. The paper concludes with the Conclusion in Section 5.

2. MULTIOBJECTIVE OPTIMIZATION USING PARTICLE SWARM OPTIMIZATION

In this section, we give a description of the multiobjective optimization problem and the particle swarm optimization algorithm. This is followed by a brief discussion of the adaptation of PSO to solve MOPs and the use of MOPSO in real world applications. Finally, we introduce , Multiobjective Particle Swarm Optimization - Crowding Distance (MOPSO-CD), the PSO-based MOO tool that we will use in the development of our trading system.

2.1 Multiobjective Optimization

Real world optimization problems are not just limited to single objectives. Many times, they require having a balance (or trade offs) among different interacting, and possibly conflicting objectives. Multiobjective optimization addresses this problem. Multiobjective optimization entails finding a set of solutions that optimizes several objectives. The notion of an optimum solution is different in multiobjective problems as compared to single objective ones since what is required is a set of tradeoff solutions rather than a single global optimum. This notion is commonly called Pareto optimality.

Pareto optimality is based on the concept of dominance. We say that one solution dominates another if it is not less than the second solution with respect to all objective functions and, at the same time, it is better than the second solution in at least one objective function. In multiobjective problems, the optimum solution is the set of all nondominated solutions. A nondominated solution is called a Pareto point while the set of all Pareto points (the optimal set of tradeoff solutions) is called the Pareto front.

2.2 Particle Swarm Optimization

Developed by Kennedy and Eberhart [14], Particle Swarm Optimization is a popular computational technique that is based on the social behavior of birds flocking to look for food. Reyes-Sierra and Coello Coello [21] note two reasons for PSO's popularity. First, since it is relatively simple, its implementation is straightforward; and second, it has been found to be very effective in a variety of applications, producing very good results at very low computational cost. PSO has been found to be effective in optimization problems requiring real-valued decision variables [3, 8]. Hassan et al.[11] report that while PSO's performance is comparable to GA, PSO is computationally more efficient than GA.

2.3 Multiobjective Particle Swarm Optimization

PSO has been extended PSO to solve MOP. The first proposal of such kind is MOPSO [4]. Less than a decade after MOPSO's introduction, several other variant MOPSO algorithms have already been proposed. In a survey of these algorithms, Reyes-Sierra and Coello Coello [21] cite two main algorithmic design aspects in adapting PSO to MOP. These are, first, the selection and updating of leaders (global best); and second, the creation of new solutions via updating of positions or mutation (or turbulence). The authors also note that, applications using MOPSO are still very few compared with those using other multiobjective evolutionary algorithms. They think that this may be due to MOPSO's relative novelty as compared to more known multiobjective genetic algorithms. The success obtained by the few applications that used MOPSO (for instance in molecular docking [12] and in blind color image fusion [18]) encourages research for other applications of this technique.

2.4 Multiobjective Particle Swarm Optimization - Crowding Distance

MOPSO's performance was compared with other multi-objective algorithms in [5]. In that study, MOPSO was the only algorithm which was able to cover the entire Pareto front for all the test functions that were presented. Citing the above study, Raquel and Naval [19] note that the success of MOPSO can be attributed to its use of an archive of nondominated solutions as well as to its new mutation operator. They further observed that while MOPSO is superior to other MOAs in converging to the true Pareto front, NSGA-II was better than it in terms of promoting diversity. This prompted them to propose a new algorithm that makes use of the specific strengths of the two algorithms. From MOPSO, they adopted the new mutation operator and the use of an external archive; and from NSGA-II,they made use of its diversity mechanism and its constraint-handling technique. Their new proposed algorithm, called MOPSO-CD, incorporates the crowding distance density estimator introduced in NSGA-II in selecting the global best and in the deletion of nondominated solutions in the archive.

3. MULTIOBJECTIVE OPTIMIZATION OF FINANCIAL TECHNICAL INDICATORS

In this section, we describe the trading system and how it employs the multiobjective optimization of financial technical indicators.

3.1 The Trading System

Subramanian et al. [23] optimized a set of weights associated with selected indicators. They employed GA and GP to optimize a weighted combination of 4 indicators (Moving Average , Price Channel Breakout, Price Trend, Order Book Volume imbalance). A decision was taken based on the combined weight of the indicators. In one experiment, they used the Sharpe ratio as their objective function, then in another, they used the sortino ratio.

Our trading system also optimized a set of weights associated with selected indicators; however, they were optimized over two objective functions using a MOO tool. We have chosen percent profit and Sharpe ratio as our objective functions. Put simply, the Sharpe ratio is the ratio of average returns to risk. The percent profit and Sharpe ratio measure two different trading criteria- profit and risk. They are not necessarily correlated: a trading system with a high profitability may, at the same time, carry with it a huge risk.

3.2 Input Data

The input data consisted of two items: first, the daily closing price of a stock index over a selected period; and second, the values of some selected technical indicators over the same period.

We chose to use a stock index to undertake the study since stock indices are more stable and are less susceptible to price speculations. Specifically, we investigated on the Dow Jones Industrial Average (DJIA) index from April 28, 1981 up to March 25, 2002 (3,960 observation points). The data file contained daily closing prices.

We selected 5 popular technical indicators- Directional Movement Index (DMI), Linear Regression(LIN), Moving Average Convergence- Divergence(MAC), Moving Average(MAV), Parabolic Stop and Reverse(PSR). These indicators were evaluated over the same period as the security data. The parameters that were used were the prescribed parameters according to literature.

3.3 Trade Parameters

Each technical indicator was associated with the usual trading rule defined in literature. The trading rule associated with an indicator generated a signal value, S_i, of 1 if the indicator is in a Long (or Buy) position, and a value of -1 if the indicator is in a Short (or Sell) position. Moreover, a weight w_i was attached to each technical indicator. The trading decision then was made to depend on the weighted decision value $\sum S_i w_i$. A trade was executed if this value exceeded $0.5 \sum S_i w_i$; a trade was terminated when this value went below $0.5 \sum S_i w_i$. The weighted decision value also determined the amount to be used in the trades. An initial investment value was used to trade the stock. Profits from the executed trades were not used for reinvestment. Transaction costs were not included in the design of the system.

3.4 Multiobjective optimization

Applied to our system, we can summarize the multiobjective optimization problem in the following manner:

Given

- [DMI,LIN,MAC,MAV,PSR] = [1,2,3,4,5]
- $\vec{S}_i$
  signal vector associated with the Indicator i
  where
  $\vec{S}_{i_j} = 1$ if position=Buy at jth trading day
  $\vec{S}_{i_j} = -1$ if position=Sell at jth trading day
- $\vec{w} = (w_1, w_2 \dots w_5), w_i \in \mathbb{R}$
  weights associated with the 5 indicators
- $WDTR(\vec{w})$ the weighted decision trading rule defined by:
  Buy $\vec{D}_j > 0.5 \sum w_i$
  Sell $\vec{D}_j < 0.5 \sum w_i$
  where
  $\vec{D}_j = \sum \vec{S}_{i_j} w_i$

Optimization Problem :

maximize
$$y = f(\vec{w}) = (f_1(\vec{w}), f_2(\vec{w}))$$

where
$$f_1(\vec{w}) = PercentProfit(WDTR(\vec{w}))$$

$f_2(\vec{w}) = SharpeRatio(WDTR(\vec{w}))$

We used Multiobjective Optimization with Crowding Distance (MOPSO-CD) [19] introduced by Raquel and Naval in the execution of the multiobjective optimization. The population size was set to 200 and the archive size to 100. The optimization was run for 100 generations.

3.5 Training and Out-of-sample Testing

Training and out-of-sample testing was carried out for a window size of 1320 trading days. We identified 3 adjacent training and testing periods (Table 1).

Table 1: Training and Testing Periods

Period	Training Range	Testing Range
A	4/28/1981 to 7/16/1986	7/17/1986 to 10/3/1991
B	7/17/1986 to 10/3/1991	10/4/1991 to 12/20/1996
C	10/4/1991 to 12/20/1996	12/23/1996 to 3/25/2002

4. RESULTS AND DISCUSSION

In this section, we discuss the system's performance in the training and testing phases, the distribution of the solutions it generated, and its running time.

4.1 Training Phase

We conducted 30 independent training and testing runs for each period; thus, we obtained 30 Pareto fronts. We then calculated PF_{best}, the average performance of the best points in the 30 Pareto fronts; and PF_{avg}, the average performance of all the points in the 30 Pareto fronts. A comparison of these two values (PF_{best} and PF_{avg}) is made with the performance of the indicators and the market, represented by the Buy-and-Hold (BH) strategy. The results of the training phase are presented in Table 2. It can be observed that the Pareto points performed very well during training. Both PF_{best} and PF_{avg} were able to beat all indicators in terms of percent profit and Sharpe ratio in all training periods. Additionally, PF_{best} beat the market in Training Periods A and B.

In Figure 1, we present a representative Pareto front from Training Periods A-C. The plots present the diversity of solutions from which a trader may select, depending on the tradeoff he wishes to get from the objective functions. The arguments above highlight two merits of the proposed system. First, it allows a trader to choose from a variety of solutions optimized for two objective functions; and second, it assures him that any solution he chooses will perform better than any indicator by itself, assuming that the system performs at least as well over testing data.

A question that could now arise is whether setting risk (represented by the standard deviation of profits) would perform better if it replaces the Sharpe ratio as the second objective function (f_2). This has been considered at the preliminary stage of the study. Table 4 shows the results of an earlier experiment in which we compared the performance of the system when f_2 is changed from Sharpe ratio to the standard deviation of profits. We observe while the PF_{best} values from the two experiments are comparable, PF_{avg} is smaller when f_2 is set to standard deviation of profits. This means that the quality of the solutions are better if f_2 is set to the Sharpe ratio.

4.2 Testing Phase

The results of the testing phase are presented in Table 3. Again, we observe that the PF_{best} beat all indicators in both objective functions. However, this time, it was not able to beat the market. On the other hand, PF_{avg} was able to beat all indicators except the Linear Regression indicator in Testing Period C. The good performance of the system over the test data suggests its effectiveness in finding acceptable solutions.

4.3 Distribution of solutions

To help out in the analysis of the distribution of the Pareto points over the 30 independent runs, we employ a bivariate extension of the boxplot called the bagplot [20]. The bagplot visualizes a two-dimensional data's location, spread, correlation and skewness. Its main components are the depth median, the bag, the fence and the loop. The depth median is the bivariate median, the location in the graph with the greatest depth. It is indicated in the graph with a red asterisk. The bag, indicated by a dark blue region, contains 50% of the observations. The fence, created by expanding the bag 3 times, serves to separate the inliers from the outliers. The loop is the region inside the fence but outside the bag; it is indicated in the graph by a light blue region.

The bagplot represents in two dimensions several characteristics of a bivariate data. The data's general location is indicated by the depth median, its spread is indicated by the size of the bag, the correlation of the two datasets can be deduced from the orientation of the bag, and finally, the skewness can be observed from the shape of the bag and the loop.

We now go to the analysis of the Pareto points using the bagplot. First, we plot the performance of the indicators on a plane whose axes represent the two objective functions. Then we superimpose the bagplot representing the performance of the Pareto points. These figures are presented in Figure 2.

From the bagplots we could draw out the following observations. The general location of the data, represented by the depth median, suggest the good performance of the system. The depth median is generally located to the upper right of the other indicators, indicating higher values for both objective functions. The differing orientations of the bags across the different periods confirm our claim that the two objective functions are not necessarily correlated. The size and shape of the bags indicate the existence of a diversity of solutions. The bagplots of the training and testing data for Periods A-C show that they have similar location, spread, correlation and skewness. This suggests that the system is robust.

4.4 Running time

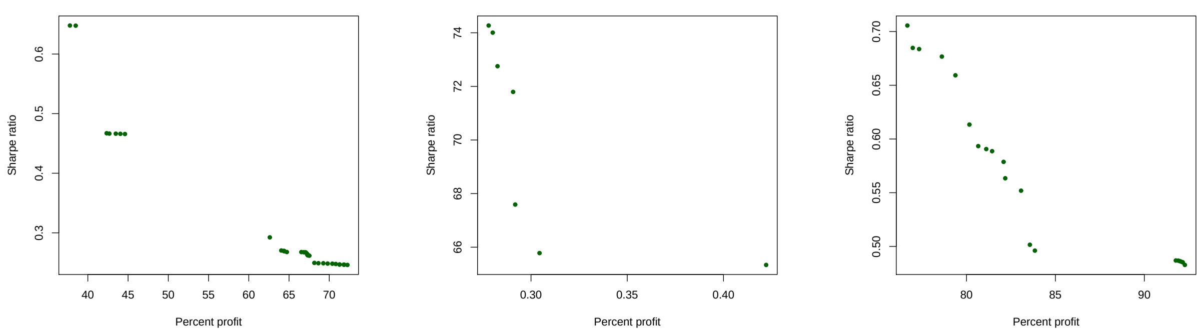
The system was fast enough to be used in actual decision making. An optimization of 1320 trading days only took 3 seconds in a computer with 1.6Ghz processor and 768MB memory. To present a rough comparison, Subramanian et al.'s experiment, optimized their system over 6 months of intraday data, took 2 hours and 30 minutes to train for 12

Table 2: Performance of the Pareto points over training data. A total of 30 Pareto fronts were obtained from 30 independent runs. PF_{best} refers to the average performance of the best points in the 30 Pareto fronts, while PF_{avg} refers to the average performance of all the points in the 30 Pareto fronts. A comparison is made with the performance of the indicators and the Buy-and-Hold.

	Training Period A		Training Period B		Training Period C	
	% Profit	Sharpe	% Profit	Sharpe	% Profit	Sharpe
Pareto Points						
PF_{best}	79.02	0.4628	74.52	0.3659	87.70	0.6349
PF_{avg}	73.27	0.2757	71.93	0.3302	83.16	0.5548
Indicators						
DMI	37.66	0.1672	24.01	0.1224	46.53	0.2619
LIN	70.52	0.2183	41.58	0.1368	49.08	0.2003
MAC	39.34	0.2119	27.82	0.1429	34.14	0.2604
MAV	39.73	0.1688	36.71	0.1517	47.40	0.2512
PSR	12.12	0.0508	-0.26	-0.0011	15.50	0.0780
Buy-and-Hold	74.46	-	67.52	-	118.94	-

Table 3: Performance of the Pareto points over testing data. A total of 30 Pareto fronts were obtained from 30 independent runs. PF_{best} refers to the average performance of the best points in the 30 Pareto fronts, while PF_{avg} refers to the average performance of all the points in the 30 Pareto fronts. A comparison is made with the performance of the indicators and the Buy-and-Hold.

	Testing Period A		Testing Period B		Testing Period C	
	% Profit	Sharpe	% Profit	Sharpe	% Profit	Sharpe
Pareto Points						
PF_{best}	63.97	0.3552	76.81	0.5205	49.49	0.2942
PF_{avg}	60.63	0.2974	72.15	0.4863	42.87	0.2130
Indicators						
DMI	24.01	0.1224	46.53	0.2619	11.35	0.0526
LIN	41.58	0.1368	49.08	0.2003	47.86	0.1374
MAC	27.82	0.1429	34.14	0.2604	17.65	0.0720
MAV	36.71	0.1517	47.40	0.2512	32.80	0.1249
PSR	-0.26	-0.0011	15.50	0.0780	2.11	0.0082
Buy-and-Hold	67.52	-	118.94	-	58.45	-



(a) Period A (b) Period B (c) Period C

Figure 1: Pareto fronts obtained from a representative run from Training Periods A-C. The representative run was chosen to be that run whose performance is closest to the median of the Pareto points.

Table 4: A comparison of the performance of the Pareto points when the second objective function (f_2) is a. the standard deviation of profits, and b.the Sharpe ratio. PF_{best} refers to the average performance of the best points in the 30 Pareto fronts, while PF_{avg} refers to the average performance of all the points in the 30 Pareto fronts. A constant amount was used to execute the trades, i.e. the weighted decision value was not used to determine the amount to be traded.

	Testing Period A		Testing Period B		Testing Period C	
	% Profit	Sharpe	% Profit	Sharpe	% Profit	Sharpe
f_2=Standard Dev						
Best	44.23	0.1892	49.64	0.3092	34.78	0.1967
Average	14.03	0.0723	27.34	0.2048	14.78	0.0945
f_2=Sharpe Ratio						
Best	44.64	0.2621	53.96	0.3394	37.93	0.1318
Average	25.64	0.2024	30.02	0.2690	24.34	0.1237

generations. We note that our optimization was run for 100 generations, and without the discretization of the decision variables.

5. CONCLUSION

This paper has presented a stock trading system based on multiobjective particle swarm optimization. Using historical end-of-day market data, the system utilized the trading signals from a set of financial technical indicators in order to develop a trading rule which is optimized for two objective functions, namely, Sharpe ratio and percent profit.

The system did well against the 5 technical indicators under study, outperforming them in terms of both objective functions in 3 training and testing periods. The system also performed competitively against the market. The plots of the performance of the Pareto points reveal that the system is robust and promotes a diversity of solutions. A further advantage of the system is its speed- it is able to carry out optimization across hundreds of generations in just a few seconds.

These results show the potential of the proposed system as a tool for making stock trading decisions and encourage further refinement in the system. Among the improvements could be explored in the future are the study of other technical indicators, the study of other objective functions such as length of trades and maximum drawdown and the addition more objective functions.

6. REFERENCES

[1] A. Abraham, C. Grosan, S. Y. Han, and A. Gelbukh. Evolutionary multiobjective optimization approach for evolving ensemble of intelligent paradigms for stock market modeling. In *Lecture Notes in Computer Science, 4th Mexican International Conference on Artificial Intelligence (MICAI'05)*, volume 3789, pages 673–681. Springer Verlag, Germany, 2005.

[2] R. Armananzas and J. A. Lozano. A multiobjective approach to the portfolio optimization problem. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)*, volume 2, pages 1388–1395, 2005. IEEE Press.

[3] M. Clerc and J. Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58-73, February 2002.

[4] C. Coello and M.Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Congress on Evolutionary Computation (CECŠ2002)*, volume 2, page 1051–1056, Piscataway, New Jersey, May 2002. IEEE Service Center.

[5] C. Coello, G. Pulido, and M. Salazar. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, June 2004.

[6] L. Diosan. A multi-objective evolutionary approach to the portfolio optimization problem. In *CIMCA '05: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-2 (CIMCA-IAWTIC'06)*, pages 183–187, Washington, DC, USA, 2005. IEEE Computer Society.

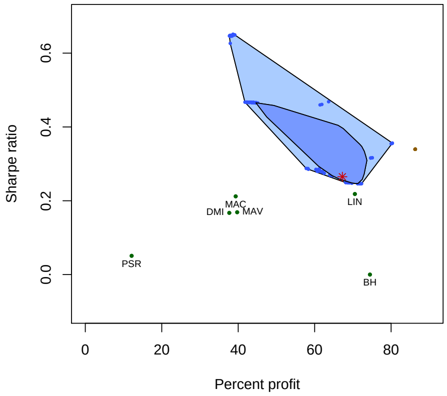
[7] K. Doerner, W. Gutjahr, R. Hartl, C. Strauss, and C. Stummer. Ant colony optimization in multiobjective portfolio selection. In *4th Metaheuristics Intl. Conf., MIC'2001*, pages 243–248, Porto,Portugal, July 2001.

[8] A. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2005.

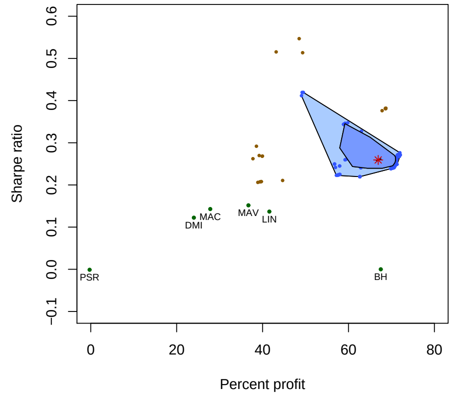
[9] F. Fernández-Rodríguez, C. González-Martel, and S. Sosvilla-Rivero. Optimisation of technical rules by genetic algorithms: Evidence from the madrid stock market. Fundacíon de Estudíos de Economía Aplicada, August 2001.

[10] R. Fukumoto and H. Kita. Designing trading agents for an artificial market with a multi-objective genetic algorithm. In *ICCIMA '01: Proceedings of the Fourth International Conference on Computational Intelligence and Multimedia Applications*, page 226, Washington, DC, USA, 2001. IEEE Computer Society.

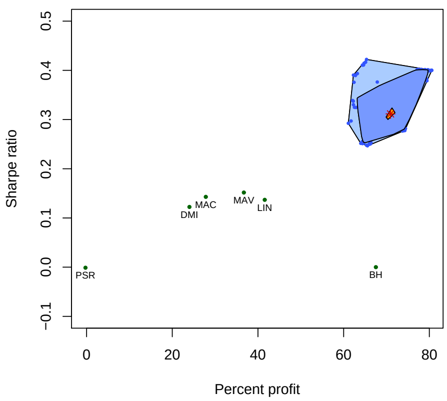
[11] R. Hassan, B. Cohanim, O. de Weck, and G. Venter. A comparison of particle swarm optimization and the genetic algorithm. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, number AIAA-2005-1897, Austin, Texas, April 2005.



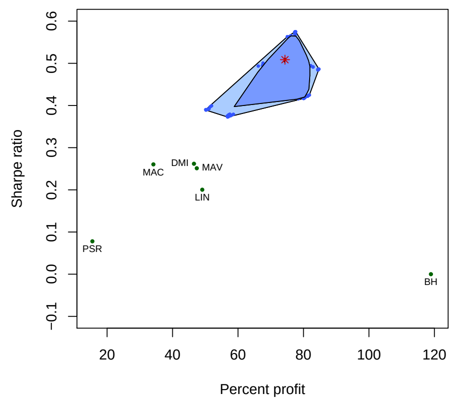
(a) Period A, Training



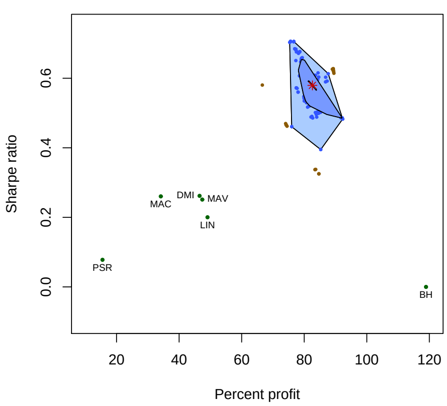
(b) Period A, Testing



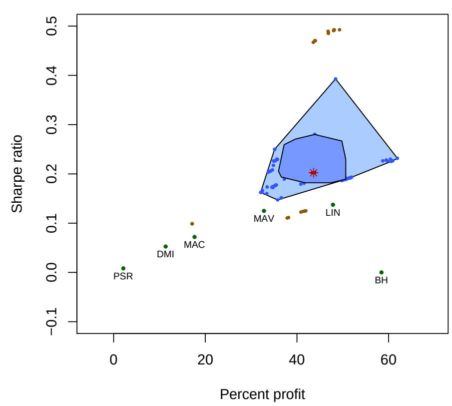
(c) Period B, Training



(d) Period B, Testing



(e) Period C, Training



(f) Period C, Testing

Figure 2: Bagplots of the training and testing performance of the Pareto points over 3 periods. The market is represented by BH (the Buy-and-hold trading strategy).The depth median, the bivariate median representing the general location of the distribution, is indicated by a red asterisk.

[12] S. Janson, D. Merkle, and M. Middendorf. Molecular docking with multi-objective particle swarm optimization. *Appl. Soft Comput.*, 8(1):666–675, 2008.

[13] M. Kearns and L. Ortiz. The Penn-Lehman automated trading project, 2003.

[14] J. Kennedy and R. Eberhart. Particle swarm optimization. In *1995 IEEE International Conference on Neural Networks*, page 1942-1948, Piscataway, New Jersey, 1995. IEEE Service Center.

[15] K.Kurumatani et al. U-mart: A virtual stock market as a forum for market structure analysis and engineering. In *5th joint Conderence on Information Science (JCIS'00), 1st Int'l Workshop on Computational Intelligence in Economics and Finance*, pages 957–960, 2000.

[16] J. Li and S. Taiwo. Enhancing financial decision making using multi-objective financial genetic programming. In *Proc.of 2006 IEEE Congress on Evolutionary Computation (CEC'06)*, Sheraton Vancouver Wall Centre, Vancouver, BC, Canada, July 16-21 2006.

[17] L. Lin, L. Cao, J. Wang, and C. Zhang. The applications of genetic algorithms in stock market data mining optimisation. In A. Zanasi, N. Ebecken, and C. Brebbia, editors, *Data Mining V: Data Mining, Text Mining and their Business Applications (DATA MINING 2004)*, page 448. WIT Press, Southampton, Boston, 2004.

[18] Y. Niu and L. Shen. The optimal multi-objective optimization using pso in blind color image fusion. In *International Conference on Multimedia and Ubiquitous Engineering*, 00:970–975, 2007.

[19] C. R. Raquel and J. Prospero C. Naval. An effective use of crowding distance in multiobjective particle swarm optimization. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 257–264, New York, NY, USA, 2005. ACM Press.

[20] P. J. Rousseeuw, I. Ruts, and J. W. Tukey. The bagplot: A bivariate boxplot. *The American Statistician*, 53(4):382–387, November 1999.

[21] M. R. Sierra and C. C. Coello. Multi-objective particle swarmoptimizers: a survey of the state-of-the-art. *Int. J. Comput. Intell. Res. 2*, 3:287-308, 2006.

[22] A. Skabar and I. Cloete. Neural networks, financial trading and the efficient markets hypothesis. In *ACSC '02: Proceedings of the twenty-fifth Australasian conference on Computer science*, pages 241–249, Darlinghurst, Australia, Australia, 2002. Australian Computer Society, Inc.

[23] H. Subramanian, S. Ramamoorthy, P. Stone, and B. J. Kuipers. Designing safe, profitable automated stock trading agents using evolutionary algorithms. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1777–1784, New York, NY, USA, 2006. ACM Press.

7. ACKNOWLEDGMENTS

AC Briza is supported by an Engineering Research and Development for Technology Grant. PC Naval is supported by an Exist Professorial Chair Grant.