Modeling POMDPs for Generating and Simulating Stock Investment Policies

Augusto Cesar Espíndola Baffa
UNIRIO - Dep. Informática Aplicada
Av. Pasteur, 458 - Térreo
Rio de Janeiro - Brazil
augusto.baffa@uniriotec.br

Angelo E. M. Ciarlini
UNIRIO - Dep. Informática Aplicada
Av. Pasteur, 458 - Térreo
Rio de Janeiro - Brazil
angelo.ciarlini@uniriotec.br

ABSTRACT

Analysts and investors use Technical Analysis tools to create charts and price indicators that help them in decision making. Chart patterns and indicators are not deterministic and even analysts may have different interpretations, depending on their experience, background and emotional state. In this way, tools that allow users to formalize these concepts and study investment policies based on them can provide a more solid basis for decision making. In this paper, we present a tool we have built to formally model stock investment contexts as Partially Observable Markov Decision Processes (POMDP), so that investment policies in the stock market can be generated and simulated, taking into consideration the accuracy of Technical Analysis techniques. In our models, we assume that the trend for the future prices is part of the *state* at a certain time and can be "partially observed" by means of Technical Analysis techniques. Historical series are used to provide probabilities related to the accuracy of Technical Analysis techniques, which are used by an automated planning algorithm to create policies that try to maximize the profit. The tool also provides flexibility for trying and comparing different models.

Categories and Subject Descriptors

I.2.8 [**Control Methods**]: Plan execution, formation, generation.
D.4.8 [**Performance**] Measurements, Modeling and prediction, Monitors, Simulation.

General Terms

Algorithms, Design, Economics and Experimentation

Keywords

POMDP, Simulation, Stock Market, Technical Analysis

SAC'10, March 22-26, 2010, Sierre, Switzerland.
Copyright 2010 ACM 978-1-60558-638-0/10/03…$10.00

1. INTRODUCTION

Stock Market Analysis can be seen as a nondeterministic and partially observable problem because there is no way to be sure about the results of a certain decision. Even experienced analysts are not able to predict all factors that might affect stock prices.

In Technical Analysis techniques [9], it is assumed Dow's hypothesis [8], which states that time series are the only relevant source of information to estimate the best time to buy or sell stocks. Technical Analysis provides many statistical methods to study price series such as chart patterns, indicators and other concepts, but there are several ways to interpret data. Investors can be either in *long* positions, in which they bet that prices of a certain stock will increase or in *short* positions, in which they bet that prices will fall. Long positions can be assumed by simply buying assets, expecting to sell them at higher prices. Short positions can be assumed by borrowing assets and selling them with the intention of buying them back later at lower prices. Ideally, Technical Analysis techniques could indicate when the investor should assume either a long or a short position. The problem is that Technical Analysis techniques are neither deterministic nor perfect.

A tool for modeling, testing and combining the concepts should help in decision making, considering the risks and potential benefits of a certain decision. Past decisions can influence future results and decisions. In this way, the estimation of bull or bear periods is not enough to make the right decision at a certain time. Buying after a bull market, for example, should consider the chances of a continued bull market to increase profits. It is then necessary to adopt policies that take into consideration risks and benefits of combining decisions in the long term.

In this paper, we describe a tool we have developed to support modeling a context of investment in the stock market as a Partially Observable Markov Decision Process (POMDP) [3], so that investment policies can be automatically generated, simulated and evaluated. The tool allows the user to formally model what composes a state at a certain time and to formally specify Technical Analysis sensors that indicate the occurrence of patterns in time series.

In order to model the process as a POMDP, we have to assume a Markov hypothesis, establishing that the next state depends only on the current state. In this way, we try to encapsulate within a state all information that could be relevant for decision making. A state contains data that can be directly observed such as the investor's position and price data from the past (up to the point that we consider useful). Additionally, we assume that there is always a trend for the future prices. In order to incorporate this idea, states also contain information about prices in the near future. This part of the state cannot be directly observed, but we can try to "partially observe it" via our Technical Analysis sensors. Based on the formal specifications for states and sensors, time series are investigated to establish probabilities of changing from one state to another and of observing a Technical Analysis pattern at a certain state. Probabilities can also be artificially specified by experienced analysts and compared to occurrences in the time series.

Having a probabilistic model for our POMDP, the tool uses automated planning techniques [16] to generate policies. These policies define which actions should be executed according to the observations obtained from the environment with the sensors. The same sensors that were previously used to learn the policy provide observations to allow the selection of the corresponding action. Policies are evaluated in different market scenes (*bull*, *bear*, *flat*). A simulator is used to evaluate policies using either real time market information or market series from a database. The tool allows the user, to test various Technical Analysis concepts and check their reliability in different scenarios.

Related works on Artificial Intelligence research have also aimed to provide support for decision-making in stock markets. Zahedim and Chong [23] showed how to define states to manage portfolios and proposed a model to adjust the quantity of shares based on a benchmark. Elder [7] proposed a reinforcement learning approach. Lin, Cao, Wang and Zhang [14] proposed a data mining approach using genetic algorithms to analyze shares. Egeli, Ozturan and Badur [6], Lezos and Tull [13], and Davey, Hunt and Frank [5] showed different techniques of prediction and forecast using neural networks. The main difference of our approach from previous work is that we try to create a platform for modeling alternative POMDPs incorporating different Technical Analysis techniques and based on historic data. Policies are then generated and simulated taking into consideration the efficacy of each technique to predict price trends, so that the best techniques for each period and asset can be chosen.

2. POMDP

A Partially Observable Markov Decision Process (POMDP) is a framework to describe a process in which an agent has to execute actions without being able to directly observe the underlying state. The agent has then to reason based on some local observations and on probability distributions over the states of the world being modeled. The set of probabilities of being in specific states at a certain time is called a *belief*. The domain should be modeled as a stochastic system with nondeterministic state transitions caused by actions. Transitions from states by means of actions are also associated with specific probabilities. More formally, a POMDP is represented by the tuple $\sum = (S, A, O, T, P, C, R)$ where:

- S is a finite set of states, A is a finite set of actions and O is a finite set of observations.
- $T_a(s'|s)$ is a probability distribution. For any $a \in A$, $s \in S$ and $s' \in S$, $T_a(s'|s)$ is the probability of reaching state s' after executing action a in s. If, for any state s and action a, $T_a(s'|s) \neq 0$, then $\sum_{s' \in S} T_a(s'|s) = 1 \cdot$
- $P_a(o \mid s)$ is a probability distribution. For any $a \in A$, $s \in S$ and $o \in O$, $P_a(o|s)$ is the probability of observing o in state s after executing a. If, for any action a and observation o, $P_a(o \mid s) \neq 0$, then $\sum_{o \in O} P_a(o \mid s) = 1 \cdot$
- $C : S \times A \rightarrow \Re$ is a function that assigns a cost to each state and action.
- $R : S \rightarrow \Re$ is a function that assigns each state to a reward.

As the current state is unknown, it is necessary to reason based on a belief, that is, a distribution probability of being at each state. Let us call the set of all possible beliefs B. If $b \in B$ is a belief state, the probability of being in state s is denoted by $b(s)$. The function $b_a(s)$ calculates the probability of reaching state s by executing action a from the current belief state b.

$$b_a(s) = \sum_{s' \in S} T_a(s \mid s') b(s')$$

The function $b_a(o)$ calculates the probability of observing o, after having executed action a, from the current belief state b.

$$b_a(o) = \sum_{s \in S} P_a(o \mid s) b_a(s)$$

So, given a belief state b, the execution of an action a results in a new belief state defined by b_a^o after having observed o, as described by the expression:

$$b_a^o(s) = \frac{P_a(o \mid s) b_a(s)}{b_a(o)}$$

The reward for a belief state b when action a is executed is calculated by the function:

$$\rho(b, a) = \sum_{s \in S} b(s)(R(s) - C(s, a))$$

A solution for a POMDP is a policy $\pi : B \rightarrow A$ that maps belief states into actions. A discount factor γ is normally used to generate policies that give more importance to recent benefits and costs. An optimal policy is a policy that maximizes the possibility of future benefits, determined by the difference between rewards and costs. The POMDP planning problem is classified as an optimization problem. Its objective is the generation of a policy that determines, for each belief state b, the best action to maximize the expected benefit *E(b)* as described by the function:

$$E(b) = \max_{a \in A} \{ \rho(b, a) + \gamma \sum_{o \in O} b_a(o) E(b_a^o) \}$$

A possible method for solving POMDP problems corresponds to algorithms that convert POMDPs to completely observable Markov Decision Processes (MDPs) based on belief states instead of domain states. The resulting MDP is then solved by means of algorithms like Policy Iteration [19] or Value Iteration [2]. However, as the set of belief states is usually infinite and continuous, POMDPs are computationally very hard to solve. Algorithms that return optimal policies work only with very small problems. Algorithms that generate approximations are then usually adopted.

3. ARCHITECTURE AND METHODOLOGY

In order to study policies for investment in the stock market, we propose a tool in which the user can specify POMDP models. Probabilities are then collected from past data (usually from a long time interval) and a policy is obtained by means of an algorithm that generates approximate optimal solutions. Finally, the efficacy of the model and the corresponding generated policy is evaluated by means of simulating the adoption of the policy to other time intervals.

All the process is based on the idea that, at any time, there is a trend for the prices of a certain asset. In this way, a state corresponding to a certain time can be composed of recent data, the current investor's position and the current trend (which corresponds to prices in the near future). We assume that we might not know the trend, but it is always confirmed and can be partially observed by means of Technical Analysis sensors. In this way, prices in the near future are part of the underlying state at each time. At real-time we never know future prices, but, when we examine the historic, prices after a certain time in the past are known and can be compared to Technical Analysis indications at that time. Probabilities relating indications to future prices can then be collected.

When we model the problem as a POMDP, we assume that Technical Analysis concepts might not be perfect when compared to real data, which is modeled by the collected probabilities. By using these probabilities, we are able to create policies that try to combine actions in order to increase profits.

The influence of past data on future prices may vary and the best interpretation for Technical Analysis concepts may also vary. In this way, our tool tries to provide flexibility both for modeling the composition of a state and for creating and using Technical Analysis sensors. Various POMDPs can then be specified and solved in parallel, generating various alternative policies that can be evaluated before being used to support decision making at real-time.

As the solution generated by planners that solve POMDPs is usually an approximation of the optimal solution, we also try to provide flexibility for experimenting with different POMDP planners.

Figure 1 presents the overall architecture of the tool. The *Context Controller* module allows the user to create a POMDP model, by defining states, actions, transitions and the way observations are to be produced by sensors. Based on this model, specific modules are activated by the *Context Controller* to solve the POMDP. Besides market data, states contain information related to the investment position (long, short or off). This is the part of the state that is directly modified by actions. As mentioned earlier, actions have costs and states have rewards. The model should be specified in such a way that states that generate profits are assigned to positive rewards and those that cause losses are assigned to negative rewards.

The *Probability Generator* module analyzes a time series stored in the database to assign probabilities to transitions between states and to observations at each state. It is usually analyzed a long time interval for estimating the probabilities, which should not overlap with time intervals used for evaluating policies. Probabilities are obtained by counting the number of times a certain state transition occurred and the number of times an observation is verified at a certain state. Probabilities can also be adjusted or inserted manually by users, according to their experience.

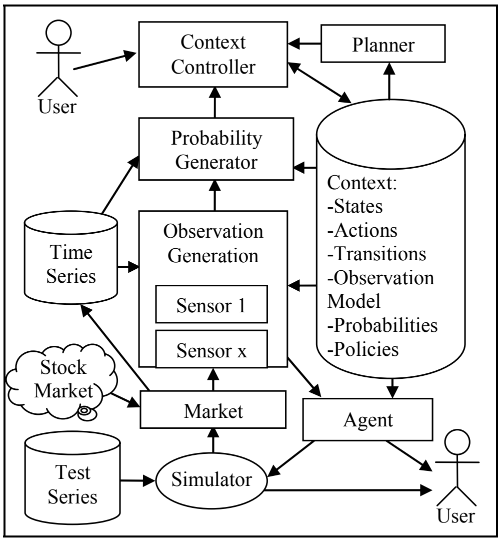

Figure 1- Overall Architecture

The *Planner* module is responsible for the automated planning process that generates policies to be stored in the database and used in future simulations and at real-time. Different algorithms can be used to solve POMDPs according to the user's preference.

Observations are generated by means of sensors that implement the Technical Analysis concepts. Sensors translate market data to discrete observations, such as estimates of high, bull market, low, bear market, flat market, oscillation and trend reversal, according to the model provided by the user. Observations can be generated by a single indicator or chart pattern or by a combination.

The *Observation Generator* module produces observations in accordance with user definitions and data provided by the *Market* module. Observations are used during the generation of probabilities and for applying policies either in simulations or at real-time for decision support.

The *Market* module is responsible for acquiring data from a broker or stock exchange. It can also simulate the stock market information using historical data provided by the *Simulator* module.

The *Agent* module is responsible for executing the policies in accordance with observations provided by the *Observation Generator*. When an observation is informed, the corresponding action is selected and a new belief state is reached. Actions can be informed to the user at real-time as a suggestion. They can also be sent to the *Simulator* module, which simulates the execution of actions and evaluates performances. Either real-time market data or test time-series can be used in this simulation. Reports about the quality of the policies are automatically generated.

The basic indicators that can be used in the sensors are: Moving Averages, Moving Average Oscillators and [15], MACD (Moving

Average Convergence/Divergence) [1], DUAL CCI (Commodity Channel Index) [11], Momentum and ROC (Rate of Change) [9], Parabolic SAR and RSI (Relative Strength Index) [22] and Stochastic Oscillator (SO)[12]. Candlesticks models [17], pivot supports and resistances [9] can also be used to produce observations.

The tool has incorporated open-source planners for solving POMDPs. We have resorted so far to the algorithms implemented by Cassandra [4], which are available at http://www.pomdp.org. As the computational complexity of the problem is high, the execution of these algorithms for obtaining approximate solutions usually takes hours. We are currently investigating the incorporation of other more recent algorithms, such as those defined in [10, 20, 21]. In addition, a Grid Computing Framework was implemented to do various planning jobs in parallel. This architecture enables the tool to process different assets and/or test different models simultaneously. During our tests, 4 PCs have been used, each one was able to process 2 jobs in parallel. The average time for obtaining a policy for the investment model described in the next section has been 8 hours.

4. BASIC INVESTMENT MODEL

In our approach, we do not propose a single POMDP model. The right model might depend on the specific stocks and on the time period it is applied to. We tried instead to provide means to create and evaluate models. Nevertheless, in order to test our architecture and methodology, we proposed an initial basic model.

Given the complexity of POMDP planning, the model must have few states, otherwise the automatic generation of investment policies might not be tractable. We decided to initially work only on the spot market, analyzing a single stock or index.

In this way, we proposed a state composed of only three variables: (a) market trend (*bull*, *bear* or *flat*) in the previous 6 days; (b) market trend in the following 6 days; and (c) investor's position (*long*, *short* or *off*). As each variable can assume 3 different values, we have 27 different states.

Each trend is defined by a range of price variation. Ranges can be adjusted by the user. For a 6-day period, we assumed the following trends in accordance with price variation: *bull* market for *variation ≥1%*, *bear* market for variation *≤-1%* and flat market for variation in between.

Investors can be either in *short* or *long* positions towards the stock being analyzed or can be *off* the market. In a long position, the investor will have bought stock shares corresponding to a fixed amount of money *M*. In a short position, the investor will have borrowed and immediately sold stock shares corresponding to the amount of money *M*. In long positions, profit is proportional the stock price variation and in short positions it is proportional to the symmetric of price variation. For the sake of simplicity, we neither consider fees nor the combination with the purchase of call options to limit losses in short positions. We assume that profits and losses are accumulated when the investor clears one position and assumes another, so that the amount of money invested is always *M*, that is, profits are not reinvested and the investor always have money to invest the amount *M*, even if he or she had previous losses.

The possible actions are *Buy*, *Sell*, *DoubleBuy*, *DoubleSell* and *Nothing*. *Buy* and *Sell* actions correspond, respectively, to buying and selling an amount of money *M* in stock shares. *DoubleBuy* and *DoubleSell* actions correspond to an amount *2M* and are used to directly change the investor's position from *short* to *long* and vice-versa. Action *Nothing* corresponds to keeping the investor's position.

The assignment of rewards to states is essential to specify how profitable each state is. It is highly desirable to be in a long position in a bull market. In contrast, in a bear market, we should try to assume a bear position. When the market is flat, it might be preferable to stay off the stock market, avoiding clearing costs and loss of other investment opportunities. As shown in Table 1, we assigned high positive rewards to states where the investor is certainly in the most profitable position and high negative rewards to states in which the investor is in a position that certainly causes losses. When trends are not so clear, smaller rewards were assigned. Rewards when the investor is off the market are close to zero, being slightly positive if the market is really flat and slightly negative when there is a clear trend that the investor could have taken advantage. The initial assignment shown in Table 1 was used in the application of the model described in Section 6, but they are not fixed and can be changed in order to study models that better correspond to a given context. For the sake of simplicity, we also decided not to assign costs to the actions, but they can be assigned typically to take into consideration that fees have to be paid to brokers when stocks are bought and sold.

Table 1 – Rewards for states based on a triple <previous 6-day trend, following 6-day trend, investor's position>

bull, bull, long	10	bull, bull, short	-10	bull, bull, off	-2
bull, flat, long	5	bull, flat, short	-2	bull, flat, off	0
bull, bear, long	0	bull, bear, short	5	bull, bear, off	0
flat, bull, long	5	flat, bull, short	-5	flat, bull, off	0
flat, flat, long	0	flat, flat, short	0	flat, flat, off	2
flat, bear, long	-5	flat, bear, short	5	flat, bear, off	0
bear, bull, long	5	bear, bull, short	0	bear, bull, off	0
bear, flat, long	-2	bear, flat, short	5	bear, flat, off	0
bear, bear, long	-10	bear, bear, short	10	bear, bear, off	-2

There are some transition restrictions due to the investor's position. Actions *Buy* or *Sell* can be executed only if the investor's position is not already *long* or *short*, respectively. On the other hand, actions *DoubleBuy* and *DoubleSell* can be executed only if the investor's position is *short* or *long*, respectively. The set of states that can be reached after an action is also limited. After an action *DoubleSell*, for instance, the investor's position is certainly *short*. This means that from one state, other 9 distinct states can be reached, varying only on the trends for the previous and following days. As already mentioned, the probability for each transition is estimated based on the historic of stock prices.

The model described in this section was simple but enabled us to already obtain interesting results as shown in the next section. More sophisticated models can however be applied to study the importance of various variables for decision making. Among them, we can point out daily volumes, correlations between different stocks and indexes, and correlations between price variations in long-term and short-term intervals.

5. APPLYING THE BASIC MODEL

The experiments were undertaken using Cassandra's POMDP Planner. We used the Finite Grid POMDP algorithm (instance of PBVI) [18]. Two periods from Bovespa[1] Index (IBOV) were selected. The first one, from Jan. 2nd., 2000 to Dec. 30th, 2007 was used during the statistical learning to estimate probabilities. During this period, many important events impacted on the market, such as the dot-com bubble, Sept. 11th. terrorist attacks and the Brazilian presidential election crisis in 2002. The other period, from Jan. 2nd, 2008 to Jun, 30th, 2009 was chosen for the simulation. It was selected based on the fact that 3 different trends occurred. In the first semester of 2008 there was a flat trend. In the second semester of 2008, the subprime crisis occurred and there was a bearish trend. In the first semester of 2009 there was a crisis recovery (a bullish period again).

We modeled the states of our POMDP as described in Section 5. In this evaluation, we chose to experiment with different sensors, each one based on a different Technical Analysis indicator. In this way, we intended to compare results for different indicators and to compare the adoption of the generated policies with the immediate obedience to the classical interpretation of each indicator. We performed simulations based on the following indicators: two versions of the Moving Average Convergence/Divergence indicator (MACD1 and MACD2), two versions of the Relative Strength Index (RSI1 and RSI2) and two versions of the Stochastic Oscillator (SO1 and SO2). Different versions of a same indicator vary based on different strategies of interpretation.

Simulations generated policies with a number of belief states close to 1000, each one establishing probability ranges for each underlying state. The policy maps each belief state and observation to an action and indicates the new belief state that will be reached after executing the action. During the simulation, observations are continuously provided for each day and the execution of the action determined by the policy is simulated. Whenever a long or a short operation is closed, the corresponding result is accumulated. If, on the last day, there is an open position the simulation forces its closure.

The results obtained for each indicator as well as the variation of the Bovespa Index (IBOV) are described in Table 2. All indicators, except SO2, performed better than IBOV, but some indicators did much better than others. In particular, in simulations based on RSI2 and SO1 we were able to make profits even during the crisis. Operations in which the investor assumed a short position were important to minimize losses and even make profits during the crisis.

Operations in which the investor assumed long positions showed to be important in particular in the recovery period after the crisis. Tables 3 and 4 present the results of long and short operations, respectively.

It is interesting to note that, for all indicators, the results obtained by using our POMDP model were much better than simply immediately following the indications of the sensors, as demonstrated in Table 5.

Table 2 – Planner Results x Ibovespa Index

Indicators	2008 Jan-Jun	2008 Jul-Nov	2009 Dec-Jun	Total
RSI1	2.06%	-10.50%	55.72%	47.27%
RSI2	14.57%	14.39%	38.88%	67.84%
MACD1	20.80%	0.00%	4.38%	25.18%
MACD2	19.77%	-59.46%	28.72%	-10.97%
SO1	10.06%	28.78%	51.83%	90.67%
SO2	2.14%	-84.37%	55.44%	-26.79%
IBOV	-1.35%	-36.88%	39.69%	-10.80%

Table 3 – Results for Long Operations

Indicators	2008 Jan-Jun	2008 Jul-Nov	2009 Dec-Jun	Total
RSI1	1.19%	-30.47%	52.22%	22.94%
RSI2	9.22%	-19.96%	42.56%	31.82%
MACD1	15.64%	0.00%	2.80%	18.44%
MACD2	19.61%	-63.79%	37.50%	-6.69%
SO1	4.42%	-11.32%	49.52%	42.62%
SO2	-0.03%	-64.39%	52.44%	-11.98%

Table 4– Results for Short Operations

Indicators	2008 Jan-Jun	2008 Jul-Nov	2009 Dec-Jun	Total
RSI1	0.87%	19.96%	3.50%	24.33%
RSI2	5.35%	34.34%	-3.68%	36.02%
MACD1	5.16%	0.00%	1.58%	6.74%
MACD2	0.16%	4.33%	-8.78%	-4.29%
SO1	5.64%	40.10%	2.31%	48.05%
SO2	2.17%	-19.98%	3.00%	-14.81%

Table 5 – Results for immediate obedience to indicators

Indicators	2008 Jan-Jun	2008 Jul-Nov	2009 Dec-Jun	Total
RSI1	1.67%	0.00%	-5.93%	-4.26%
RSI2	14.01%	-28.14%	25.38%	11.25%
MACD1	1.98%	0.00%	-6.36%	-4.38%
MACD2	14.71%	-65.53%	11.80%	-33.13%
SO1	-7.37%	-69.06%	5.98%	-70.45%
SO2	4.67%	-86.77%	0.00%	-82.10%

6. CONCLUDING REMARKS

In this paper, we proposed an approach to generate and simulate stock investment policies. The approach is based on modeling investment contexts as POMDPs and using Technical Analysis to provide observations about market trends. We have implemented a prototype tool that allows users to study various models and Technical Analysis concepts. We have also run the tool with a basic investment model that we proposed to test our approach and obtained some interesting results.

The combined use of Technical Analysis with POMDPs has produced much better results than simply following Technical

[1] Bovespa (Bolsa de Valores de São Paulo) is the main Brazilian stock exchange which is based in São Paulo city.

Analysis indicators. Results have confirmed so far our hypothesis that POMDPs can enhance profitability, because indicators are not perfect. In our approach, we combine actions taking into account the probabilities they have shown of identifying market trends. Our approach has also shown to be useful to compare Technical Analysis concepts, so that an investor can choose the one that seems to produce better results for a certain stock.

In the continuation of this project, we intend to apply the tool to other stocks and indexes and check whether other models for the states of our POMDP can produce better results. It deserves special attention the combination of long-term and short-term trends for composing a state. Other Technical Analysis indicators and graphical patterns as well as their combination to create sensors are also still to be investigated. In addition, we intend to study how our tool can be adapted to cope with more complex investment contexts, such as those in which the user has limited resources, wants to limit risks or wants to combine investments in various stocks and options.

7. REFERENCES

[1] Appel, G. *The Moving Average Convergence-Divergence Trading Method(Advanced Version)*. Traders Pr, 1985

[2] Bellman, R. *Dynamic Programming*. Princeton University Press, 1957

[3] Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. *Acting optimally in partially observable stochastic domains.* In Proceedings of the Twelfth National Conference on Artificial Intelligence, (AAAI) Seattle, WA, 1994

[4] Cassandra, A.R. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. Ph.D. Thesis. Brown University, Department of Computer Science, Providence, RI, 1998

[5] Davey, N., Hunt, S.P. and Frank, R.J. *Time Series Prediction and Neural Networks*, Kluwer Academic Publishers Journal of Intelligent and Robotic Systems archive, Volume 31: May /July,2001

[6] Egeli, B., Ozturan, M. and Badur, B. *Stock Market Prediction Using Artificial Neural Networks*, Hawaii International Conference on Business, Honolulu, Hawaii, USA*,* 2003

[7] Elder, T. *Creating Algorithmic Traders with Hierarchical Reinforcement Learning*, MSc. Dissertation, School of Informatics, University of Edinburgh, 2008

[8] Hamilton, W. *The Stock Market Barometer: A Study of its Forecast Value Based on Charles H. Dow's Theory of the Price Movement*.. New York, NY: John Wiley & Sons,Inc., 1998 (reprint of 1922 edition)

[9] Kirkpatrick, C. and Dahlquist, J. *Technical Analysis: the Complete Resource for Financial Market Technicians*. First. FT Press, 2006

[10] Kurniawati, H., Hsu, D., and Lee, W.S. *SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces*. In *Proc. Robotics: Science and Systems*, 2008

[11] Lambert, D. *Commodity channel index: Tool for trading cyclic trends*, Technical Analysis of Stocks & Commodities, Volume 1: July/August, 1983

[12] Lane, G. C.. *Lane's stochastics: the ultimate oscillator.* Journal of Technical Analysis, May, 1985

[13] Lezos, G., and Tull, M. *Neural network & fuzzy logic techniques for time series forecasting*. Conference on Computational Intelligence for Financial Engineering, New York, NY, 1999

[14] Lin, L., Cao, L., Wang, J. and Zhang, C. *The Applications of Genetic Algorithms in Stock Market Data Mining Optimisation*, Fifth International Conference on Data Mining, Text Mining and their Business Applications, Malaga, Spain, 2004

[15] Murphy, J.J. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Prentice Hall Press, 1999

[16] Nau, D., Ghallab, M., and Traverso, P. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc. , 2004

[17] Nison, S. *Japanese Candlestick Charting Techniques, Second Edition*. Prentice Hall Press., 2001

[18] Pineau, J., Gordon, G., and Thrun, S.. *Point-based value iteration: An anytime algorithm for POMDPs*. In Proc. Int. Joint Conf. on Artificial Intelligence, Acapulco, Mexico, 2003

[19] Puterman, M.L. *Markov Decision Processes : DiscreteStochastic Dynamic Programming*. New York : John Wiley & Sons, 1994

[20] Smith, T. and Simmons, R. *Heuristic search value iteration for POMDPs*. In Proceedings of the 20th Conference on Uncertainty in Artificial intelligence (Banff, Canada, July 07 - 11, 2004). ACM International Conference Proceeding Series, vol. 70. AUAI Press, Arlington, Virginia, 520-527, 2004

[21] Smith, T., Thompsom,D.R., and Wettergreen,D.S. *Generating exponentially smaller POMDP models using conditionally irrelevant variable abstraction*. In Proc. Int. Conf. on Applied Planning and Scheduling (ICAPS), 2007

[22] Wilder, J.W. *New Concepts in Technical Trading Systems*. Greensboro, SC: Trend Research, 1978

[23] Zahedim, R. and Chong, E., *Portfolio Management using Partially Observable Markov Decision Processes,* Colorado State University Information Science and Technology Research Colloquium, 2005