Portfolio Optimisation with Sequential Monte Carlo

Yow Tzu Lim

MSc in Statistics

Imperial College, London

Department of Mathematics

August 7, 2014

Abstract

As computer system size and complexity grow, formulating effective policies require more sophistication. There are many risk factors need to be considered, some of which may be in conflict. Inevitably, unpredictable circumstances that demand decisions will arise during operation. In some cases an automated response may be imperative; in other cases these may be ill-advised. Manual decisions made to override the default ones, in a sense, redefining the underlying policy. This matter is further complicated in highly dynamic operational environments like mobile ad-hoc networks, in which the risk factors may be changing continually. Thus, security policies must be able to change and adapt to the operational needs.

This study investigates the potential of evolutionary algorithms as a tool in determining the optimal security policies that suit such environments. This thesis reviews some fundamental concepts in related domains. It presents three applications of evolutionary algorithms in solving problems that are of direct relevance. These include the inference of security policies from decision examples, the dynamic adaptation of security policies and the optimisation of security policies for a specific set of missions. The results show that the inference approaches based on evolutionary algorithms are very promising. These approaches are also sufficiently generic to be used as general dynamic classification algorithms.

The thesis concludes with an evaluation on the work done, the extent to which the work justify the thesis hypothesis and some possible directions on how evolutionary algorithms can be applied to address a wider range of relevant problems on the domain of concern.

Contents

1 Introduction **1**

1.1 Technical Approach . . . 2
1.2 Thesis hypothesis . . . 2
1.3 Thesis organisation . . . 3

2 Monte Carlo Methods **4**

2.1 Bayesian Inference . . . 4
2.2 Perfect Monte Carlo . . . 5
2.3 Rejection sampling . . . 6
2.4 Importance sampling . . . 6
2.5 Sequential Monte Carlo . . . 7
2.5.1 Hidden Markov Models . . . 8
2.5.2 Kalman Filter . . . 9
2.5.3 Sequential Important Sampling (SIS) . . . 10
2.5.4 Optimal Proposal Distribution . . . 11
2.5.5 Sequential Importance Resampling (SIR) . . . 12
2.5.5.1 Resample-Move Algorithm . . . 12
2.5.6 Effective sample size (ESS) . . . 13
2.5.7 Rao-blackwellised (Marginal) Important sampling . . . 13
2.6 Conclusion . . . 13
2.7 Markov chain Monte Carlo (MCMC) . . . 14
2.7.1 Markov Chain . . . 14
2.7.2 Metropolis-Hastings sampling . . . 15
2.7.3 Gibbs sampling . . . 16

3 Portfolio optimisation **17**

3.1 Technical Approach . . . 17
3.2 Portfolio Optimisation . . . 18

3.3 Portfolio Optimisation as a Stochastic Control Problem 20
3.4 Technical Approach 20
3.5 Objective in portfolio optimisation 20
3.6 Technical approach 21
3.7 Problem formulation 22
3.8 Numerical example on stationary oscillating wave 22
3.8.1 Period length performance 23
3.8.2 Increasing the power 23
3.9 Different reference signals 23
3.10 Discussion and possible extension 23
3.11 Possible parallel computation at different phases 23
3.12 Evidence for the thesis and future work 23
3.13 Conclusions 24

4 Evaluation and conclusions **26**
4.1 Evaluation 26
4.1.1 Static policy inference 26
4.1.2 Dynamic policy inference 27
4.1.3 Mission-specific policy discovery 28
4.1.4 Thesis contributions 28
4.2 Envisaged future work 29
4.2.1 Policy fusion 30
4.2.2 The robustness of a security policy 30
4.2.3 Scalability with the training set size 30
4.2.4 More complex security policies 31
4.2.5 More complex scenarios 31
4.3 Closing remarks 32

References **33**

List of Figures

2.1 Hidden Markov Models 8

List of Tables

Acknowledgements

I would like to express my gratitude to all who have provided support in accomplishing this thesis. I would like to thank my supervisor, Dr. Nikolas Kantas, for the opportunity given to carry out this research and also for his guidance, expertise and encouragement during the course of this research. I would like to thank the support staff, especially XYZ and ABC, who have maintained the Tesla grid servers, on which my experiments have been carried out.

I would also like to express my appreciation to my wife Ka Ki Lai for her endless support and care in my personal life during the course of this master degree study. I would like to thank God Almighty, who has given me the strength and peace to work hard on this project. I would like to express my gratitude to all who have provided support in accomplishing this thesis.

Declaration

The work contained in this thesis is my own work unless otherwise stated.

I am the primary author of all work reported in this thesis. Advice on specific aspects of the work was provided by my supervisor, Dr. Nikolas Kantas.

Chapter 1

Introduction

Resource allocation is a common challenge for every investor. In the investment decision making process, investors decide how much resources are allocated to different investable assets to form a portfolio with the aim to optimise the performance of the overall portfolio is better off than any other according to some criterion. The criterion can be different to the investors. (Some investors may have different considerations such as tax considerations and legal restriction on investment assets or holding periods.)

Two common objectives, often contradicting, are the financial return and the investment risk. The Markowitz's modern portfolio theory [1] proposes a portfolio selection framework. In Markowitz's model, it is assumed that investor attempt to maximize a portfolio's return and minimize the risk (measured by the variance of the portfolio return. Based on this criteria, the set of non-dominance portfolio is known as the *efficient porfolios*. Using variance as a risk measure has its limitation. Variance is a symmetric measure; an out performing asset (than the expected return) is deemed to as risky as an under perfoming one. Many alternative risk measurements have been proposed, e.g. Sortino ratio, Conditional Value at Risk (CVaR), etc.. Refer [2] for details.

Surprisingly, there are some investment managers who have no interest on maximising their portfolio's return. Instead, the main objective of portfolio management for such a fund is simply track and replicate the exposure of a benchmark index as close as possible. These funds are attractive as it provides investors the exposure to the market, at the same time, minimal active management required makes the fund less vulnerable to change of management and has the advatanges of lower fees and taxes. The performance of these funds are often assessed in term of how well the fund tracks the benchmark index using some pre-defined metrics.

1.1 Technical Approach

Traditionally, portfolio optimization have been explored in an analytical fashion, adopting necessary assumption as necessary. This seems rather restrictive; there are many instances where numerical method has been used to derive an approximate or even more effective solution to the problem in question. For example, Monte Carlo technique is used to do integral, evolutionary techniques applied in engineering domains, etc..

Our approach to the problem in this thesis is a radical one. We view a portfolio optimisation as a *stochastic* control problem. We adopt the Bayesian view and treat tehse parameters as random variables. The objective is to find the sequence of control parameters that optimise the control objective defined in terms of portfolio return and financial risk. We investigate the potential of using SMCs as the means of determining the optimal strategy, or at least excellent, strategies for the portfolio optimisation problem in question. The main reason of choosing SMCs is its ability to carry out *sequential* update on the posterior distribution over time fit well with parameter inference in stochastic process. Moreoever, these techniques have achieved significant success in their applications on many domains. Of course, other heuristic search tecniques are also potentially applicable.

To investigate this approach, we first applied the technique on to a simpled determinstic reference model. This model is doubly useful. It demonstrates the concept nicely and serves as a basic model to allow us to gain further understanding on the tunable parameters. We then considered a simplied a market model with two assets: one risky asset with its price modelled as Brownian motion with drift, and one zero interest risk-free asset (constant) which has an analytical solution. Using SMCs, we search for the optimal strategy (the set of control parameters at each time point) that optimise against the optimisation criteria (a.k.a. reward) in terms of expected return over the investment period, and minimize the financial risk (variance of the return) using SMC techniques. The results are then evaluated against with the analytic solution described in [**?**].

1.2 Thesis hypothesis

Formally, the hypothesis of the thesis is stated as follows:

> Sequential Monte Carlo (SMCs) have the potential to be an effective means of searching the optimal strategy for portfolio problem.

We attempt to examine this hypothesis from three different perspectives:

1. Exploring the potential of SMCs in searching the optimal strategy from mean-variance criteria with constraints.

2. Exploring the sensitivity of the techniques in terms of the parameter settings.

3. Exploring the trade-off between the estimation accuracy, the complexity of the problem complexity and the computational efforts numerically and providing suggestions for real-world problem.

Given the time frame, we fully understand it is impossible to evaluate our approach on full scale strategy. The aim is to establish the plausibility or, at the very least, a greater understanding of the strengths and weaknesses of the above approach.

1.3 Thesis organisation

The subsequent chapters of this thesis are organised as follows:

- Chapter 2 reviews some fundamental concept in Monte Carlo method that are related to this thesis. It begins with a brief introduction to basic methods such as perfect Monte Carlo sampling, rejection sampling, importance sampling. It then details two common Markov Chain Monte Carlo (MCMC) techniques, namely Metropolis-Hastings and Gibbs Sampling. Lastly, it introduces the Sequential Monte Carlo (SMC) technique used in this thesis.

- Chapter 3 briefly review the state of the art of portfolio optisation problem. It then discusses how a portfolio problem can be transformed naturally into standard parameter estimation in Sequential Monte Carlo frameowrk.

- Chapter 4 details the toy experiment in which we attempt to use SMC to track a reference signal generated by a known synthetic model.

- Chapter 5 details the experiment in using SMC to infer the optimal control for portfolio that tracks real-world indices. In particular, we focus on the major stock indices accross the continient. It first discusses the market model that is used. It then details the problem form ulation. Lastly, it discusses the experimental results, in comparison to the theorical results.

- Chapter 6 concludes the thesis by evaluating the degree to which the hypothesis has been justfied and outlines potential work for the future.

Chapter 2

Monte Carlo Methods

This chapter reviews some fundamental concept in Monte Carlo method that are related to this thesis. It begins with a brief introduction to basic methods such as perfect Monte Carlo sampling, rejection sampling, importance sampling. It then details two common Markov Chain Monte Carlo (MCMC) techniques, namely Metropolis-Hastings and Gibbs Sampling. Lastly, it introduces the Sequential Monte Carlo (SMC) technique used in this thesis.

2.1 Bayesian Inference

In Bayesian inference framework, each unknown parameter in the model is assumed to be random variable and is associated with prior distribution that characterises the initial belief. The inference process is merely updating the belief with new observable evidence in a systematic fashion using Bayes theorem.

Formally, let M be the Bayes model of interest, θ be the set of parameters of the model, $p(\theta \mid M)$ be the prior distribution (initial belief) and $p(x \mid \theta, M)$ be the probability of observing an observation x given the model, then posterior distribution (updated belief) is given as follows:

$$\begin{aligned} p(\theta \mid x, M) &= \frac{p(x \mid \theta, M)\ p(\theta \mid M)}{p(x \mid M)} \\ &\propto p(x \mid \theta, M)\ p(\theta \mid M) \qquad (2.1) \end{aligned}$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior} \qquad (2.2)$$

This problem formulation is elegant, but there remains some subtle issues in practice. One particular issue is about the calculation of the normalisation constant

$p(x \mid M)$ in (2.1), which requires us to have the ability to carry out the following integral analytically:

$$p(x \mid M) = \int p(x \mid \theta, M)p(\theta \mid M)\ d\theta \tag{2.3}$$

, which is often infeasible[1].

Instead of a closed form solution, the Method Carlo methods offer a numerical solution in estimating the integral using sampling technique. The need of calculating integral that does not posess analytic solution also arises in the marginalisation process of nuisance parameters, calculating expectation of a function, etc..

2.2 Perfect Monte Carlo

Consider the calculation the expectation of a function, I of the following form:

$$I = E[f(x)] = \int f(x)p(x)\ dx \tag{2.4}$$

assuming we are able to sample N independent and identically distributed (i.i.d.) points of x from $p(\cdot)$, denote these as $\{x^{(i)}\}$ where i is from $1 \dots N$, a Monte Carlo estimate of I using the the point masses of the samples is:

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} f(x^{(i)}) \tag{2.5}$$

One can view this approximation as a discretisation of the continuous distribution with *random* support. This estimate has been shown to be unbiased and converge almost surely to be unbiased and converge almost surely to I as $N \to \infty$ by the Law of Large number.

Morever, if the variance of $f(\cdot)$ is bounded ($\sigma_f^2 < \infty$), then the following central limit theorem holds:

$$\sqrt{N}(\hat{I} - I) \Longrightarrow N(0, \sigma_f^2) \tag{2.6}$$

as $N \to \infty$, where $\Longrightarrow$ denotes convergence in distribution. It is important to note that the converengence rate of $\frac{1}{\sqrt{N}}$ is independent of the dimensions of x. This is in constrast with any determinstic method that has a rate that decreases a the integral dimension increases [3]. This is the main advantage of Monte Carlo integration.

[1] Traditionally, the need of normalisation constant is often circumvented by making use of conjugate prior (the choices of certain prior distributions that yield posterior distributions from the same family) in an analytical fashion.

2.3 Rejection sampling

However, it is not always possible to sample directly from the distribution $p(\cdot)$. Suppose we can find an instrumental distribution, $q(\cdot)$, that is easy to sample from and has the property such that $cq(x)$ dominates $p(x)$ for all x, i.e., $cq(x) \geq p(x) \geq 0$ for all x, then to get a random sample from $p(\cdot)$, we can first sample from $q(\cdot)$ instead and accept the sample with acceptance probability $\alpha(x) = \frac{p(x)}{cq(x)}$. If the sample is rejected, repeat the process until success. The algorithm is summarised in Algorithm 1.

Algorithm 1 Rejection Sampling

function REJECTIONSAMPLING(n)
    $r = [\ ]$
    repeat
        sample $x \sim q(\cdot)$
        sample $u \sim \mathcal{U}(0, 1)$
        if $u \leq \frac{p(x)}{cq(x)}$ **then**
            $r \leftarrow [r, x]$
        end if
    until len(r)=n
end function

One could easily see that the optimal choice of instrumental distribution, q^*, is that one minimize the space bounded by $cq(x)$ subject to the constraint that it still dominates the target density $p(x)$. As the dimension of x increases, this algorithm becomes very ineffecient because the acceptance ratio that is defined as the ratio of two embedded spaces tends towards zero and therefore many generated examples would be rejected.

2.4 Importance sampling

Instead of making a binary accept-reject decision on each sample, the main idea behinds important sampling is to weight each sample x based on how well the sample from the instrumental distribution, $q(\cdot)$, resembles the target distribution, $p(\cdot)$. More formally, assume we have an instrumental distribution, $q(\cdot)$, that is easy to sample from which

has support that includes $p(\cdot)$, we can re-write (2.4) as:

$$\begin{aligned} I &= \int f(x)\frac{p(x)}{q(x)}q(x)\ dx \\ &= \int f(x)w(x)q(x)\ dx \end{aligned} \tag{2.7}$$

where $w(x) = \dfrac{p(x)}{q(x)}$, which is commonly referred as the importance weight. This reformulation leads to the following Monte Carlo estimate of I:

$$\begin{aligned} \hat{I} &= \frac{\frac{1}{N}\sum_{i=1}^{N} w(x^{(i)})f(x^{(i)})}{\frac{1}{N}\sum_{j=1}^{N} w(x^{(j)})} \\ &= \sum_{i=1}^{N} \tilde{w}(x^{(i)})f(x^{(i)}) \end{aligned} \tag{2.8}$$

where $\tilde{w}(x^{(i)}) = \dfrac{w(x^{(i)})}{\sum_{j=1}^{N} w(x^{(j)})}$ is the normalised importance weight. This estimate is biased as it consists of the ratio of two estimates, yet it is still asymtoptically consistent.

To obtain samples from the target distribution, $p(\cdot)$, an additional resampling step can be introduced. In the first step, we sample a sample set of size M, $\tilde{x}^{(i)}_{0\leq i\leq N}$ from the instrumental distribution, each of whcih is associated with a weight $\tilde{w}(x^{(i)})$ as discussed. In the resampling step, a sample set of N, $\tilde{x}^{(i)}_{0\leq i\leq N}$ is draw from this intermediate set with the weightings taken into account.

There are many ways of implementing the resampling stage. A simple direct implementation would be select the sample from the intermediate stage according to a Multinomial distribution with with the success probability parameter set to the weighting vector, $\tilde{w}(x^{(i)})_{0\leq i\leq N}$, i.e., the chance of a sample point being replicated is proportional to its weightings. Note that this resampling step introduces extra variance to the estimators, but it is crucial in the sequential scheme that we shall look in the following section to avoid sampling degeneracy over time.

2.5 Sequential Monte Carlo

Even if we were able to sample exactly from the target distribution $\pi_n(x_{1:n})$, the minimal computational complexity of the sampling scheme would be at least linear in n. Sequential Monte Carlo (SMC) techniques provides a way to obtain samples for each sequential time step in a fixed computational complexity time in Hidden Markov Models (HMMs). We shall briefly some background on HMMs in the next section that is

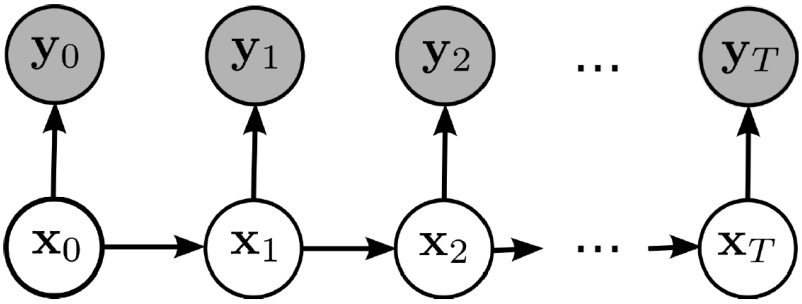

Figure 2.1: Hidden Markov Models

necessary to understand SMC. Refer [4] for further details of inference techniques for HMMs in general.

2.5.1 Hidden Markov Models

HMMs can be seen as a class of models that consist of two related processes: an underlying Markov process, X_t, which is the target process of interest, and a observable process, Y_t, which its state can be measured and therefore provides some information about X_t. Moreover, it is assumed that these two processes have conditional independence properties as shown using the graphical model representation in Figure 2.1. These properties can be summarised as follows:

$$\begin{aligned} p(x_t \mid x_{1:t-1}) &= f(x_t \mid x_{t-1}) \\ p(y_t \mid x_{1:t}, y_{1:t-1}) &= g(y_t \mid x_t) \end{aligned} \tag{2.9}$$

where $f(x_t \mid x_{t-1})$ is the transition density and $g(y_t \mid x_t)$ is the likelihood. It is worth emphasizing here that this model is designed to capture systems that evolve from one state to another over time, generating observation after each state move. The inference problem is typically about estimating the state(s) in *real-time* given the observations. This imposes an implicit requirement from the computation perspective that the estimate calculation cost should remain constant over time, i.e., the calculation cost does not increase with the increasing number of states.

Arguably, the most common inference problem in HMMs is the smoothing distribution, $p(x_{1:t} \mid y_{1:t})$, that is estimating the states $x_{1:t}$ based on the sequence of observations up to time t, $y_{1:t}$. Using Bayes rules, we can write the density of the distribution of interest as follows:

$$\begin{aligned} p(x_{1:t} \mid y_{1:t}) &\propto p(x_{1:t} \mid y_{1:t-1}) g(y_t \mid x_t) \\ &= p(x_{1:t} \mid y_{1:t-1}) f(x_t \mid x_{t-1}) g(y_t \mid x_t) \end{aligned} \tag{2.10}$$

This recursion is often re-written into two separate steps: the prediction step (the estimation of distribution of t states given only $t-1$ states):

$$p(x_{1:t} \mid y_{1:t-1}) = p(x_{1:t-1} \mid y_{1:t-1}) f(x_t \mid x_{t-1})$$

and the update step (the correction of the predicted distribution taking into account the new observation as follows:

$$p(x_{1:t} \mid y_{1:t}) = \frac{p(x_{1:t} \mid y_{1:t-1}) g(y_t \mid x_t)}{\int p(x_{1:t} \mid y_{1:t-1}) g(y_t \mid x_t)\ dx_{1:t}} \tag{2.11}$$

Moreover, the estimate of any smoothing distribution $p(x_{j:k} \mid y_{1:t})$ where $j \leq k \leq l$) can be obtained by integrating out x that are not interested in as follows:

$$p(x_{j:l} \mid y_{1:t}) = \int p(x_{1:t} \mid y_{1:t}) dx_{1:j,l+1:t} \tag{2.12}$$

One particular smoothing distribution of interest is the final marginal distribution $p(x_t \mid y_{1:t})$, which is often referred to as the filtering distribution.

Another distribution of interest is the prediction distribution, that is the estimation of the distribution of any unseen *future* state based on the sequence of observations up to time. If we let $j = 1$ and $l \geq n$ in (2.12), we obtain the following equation:

$$p(x_{j:l} \mid y_{1:t}) = p(x_{j:t} \mid y_{1:t}) \prod_{i=t+1}^{k} f(x_i \mid x_{i-1}) \tag{2.13}$$

Therefore, any prediction density can be obtained by simply integrating out the variables of not interest from the above equation.

While the problem of distribution estimations as discussed appears to be simple, but the problem is in fact far from being resolved in practice. The integral appear in the above equations are often intractable and can only be estimated except in the a very specific setting discussed below.

2.5.2 Kalman Filter

In the linear Gaussian setting in which the transition density and likelihood are each a Gaussian distribution with center lied at a point of a linear combination of the known conditional variables, u_t of the following form:

$$\begin{aligned} f_t(x_t \mid x_{t-1}, u_t) &= N(A_t(u_t)x_{t-1} + F_t(u_t), B_t(u_t)B_t(u_t)^T) \\ g_t(y_t \mid x_t, u_t) &= N(C_t(u_t)x_t, D_t(u_t)D_t(u_t)^T) \end{aligned} \tag{2.14}$$

where A_t, B_t, C_t, D_t are appropriate known matrix or vector operations, that may depend on the given conditional variable u_t.

Using the properties of Gaussian distribution, the integral can be resolved analytically. This leads the widely used *Kalman Filter* [5], which has the following recursive solution as follows:

$$\mu_{t|t-1} = A_t(u_t)(\mu_{t-1|t-1})X_{t-1} + F_t(u_t) \tag{2.15}$$
$$\Sigma_{t|t-1} = A_t(u_t)\Sigma_{t-1|t-1}A_t(u_t)^T + B_t(u_t)B_t(u_t)^T \tag{2.16}$$
$$S_t = C_t(u_t)\Sigma_{t|t-1}C_t(u_t)^T + D_t(u_t)D_t(u_t)^T \tag{2.17}$$
$$y_{t|t-1} = C_t(u_t)\mu_{t|t-1} + G_t(u_t) \tag{2.18}$$
$$\mu_{t|t} = \mu_{t|t-1} + \Sigma_{t|t-1}C_t(u_t)S_t^{-1}(y_t - t_{t|t-1}) \tag{2.19}$$
$$\Sigma_{t|t} = \Sigma_{t|t-1} - \Sigma_{t|t-1}C_t(u_t)S_t^{-1}C_t(u_t)\Sigma_{t|t-1} \tag{2.20}$$

where $\mu_{t|t-1}$ and $\Sigma_{t|t-1}$ are the predicted mean and covariance of the state x_t, $y_{t|t-1}$ and S_t are the mean and covariance of the measurement at time t and $\mu_{t|t}$ and $\Sigma_{t|t}$ are the estimated mean and covariance of the state x_t after seeing the observation y_t.

There are various extensions have been developed to this approach. For example, the Extended Kalman Filter (EKF) which uses Taylor Series expansion to linearise at the conditional variables locally, Unscentend Kalman Filter, etc. Refer [6] for further details.

2.5.3 Sequential Important Sampling (SIS)

In more general setting, here is however no analytical solution for this estimation problem. Sequential Monte Carlo provides a systematic way to approximate the solution to this estimation problem. Assuming that it is possible to decompose the selected importance distribution in the following form:

$$\begin{aligned} q_n(x_{1:t}) &= q_n(x_{1:t-1})q_n(x_t \mid x_{1:t-1}) \\ &= q_1(x_1)\prod_{i=2}^{n} q_i(x_i \mid x_{1:t-1}) \end{aligned} \tag{2.21}$$

we can then obtain sample of $X_{1:n} \sim q_n(x_{1:t-1})$ at time t by first sample $X_1 \sim q_1$ at time 1 then $X_i \sim q_i(x_i \mid x_{1:t-1})$ for time i from $2 \ldots n$. The corresonding weights associated to each sample $X_{1:n}$ can also calculated in a similar recursion fashion using

the following decomposition:

$$\begin{aligned} w_n(x_{1:t}) &= \frac{p_t(x_{1:t})}{q_t(x_{1:t})} \\ &= \frac{p_{t-1}(x_{1:t-1})}{q_{t-1}(x_{1:t-1})} \frac{p_t(x_{1:t})}{p_{t-1}(x_{1:t-1}) q_n(x_t \mid x_{1:t-1})} \qquad (2.22) \\ &= w_1(x_1) \prod_{i=2}^{n} \frac{p_i(x_{1:i})}{p_{i-1}(x_{1:i-1}) q + n(x_i \mid x_{1:i-1})} \\ &= w_1(x_1) \prod_{i=2}^{n} \alpha_k(x_{1:i}) \qquad (2.23) \end{aligned}$$

where $\alpha_k(x_{1:i})$ is often referred to as incremental importance weight function. This algorithm is summarised in X.

2.5.4 Optimal Proposal Distribution

While SIS is attractive, it is nothing but a specialised version of importance sampling introduced earlier in 2.4. As the state space increases with the number of time step t. Direct importance sampling on a state space with increasing size is not efficient. The weights of the samples start to degenerate quickly, in the sense that the weights start to concentrate on only a small number of samples, i.e., many of the samples have negligible weights and therefore rendered useless in estimating the expectation. It can be shown that the importance weights will increase with every iteration, and therefore the quality of the estimators will decrease over time [].

To alleviate this weight degeneracy issue, we can rewrite (2.22) as follows:

$$\begin{aligned} w_n(x_{1:t}) &= \frac{p_{t-1}(x_{1:t-1})}{q_{t-1}(x_{1:t-1})} \frac{p_t(x_{1:t})}{p_{t-1}(x_{1:t-1}) q + n(x_t \mid x_{1:t-1})} \\ &= 1 + 1 \qquad (2.24) \end{aligned}$$

Looking at (2.24), it is obvious that the proposal distribution, q_{t+1} that minimise the variance of the importance weight, $w_n(x_{1:t})$ takes the following form:

$$q_{t+1}(x + 1 \mid x_{t-1} \propto f_{t+1}(x_{t+1} \mid \qquad (2.25)$$

This is often referred to as the optimal proposal distribution.

In general, it is not always possible to sample from this optimal proposal distribution. Yet, the knowledge of its form can be used to guide the design of a reasonable good proposal distribution, which can be sampled from. Using a better proposal distribution reduces the amount of variance introduced, but does not totally eliminate degeneracy problem.

2.5.5 Sequential Importance Resampling (SIR)

The variance in importance weight accumulates over iterations. This suggests a possible solution is to "reset" the weights associated to the samples somehow during the iterations. Sequential Importance Resampling (SIR) introduces an additional resampling step to SIS step in a similar fashsion as discussed in Section 2.4. After resampling, the weight of each samples are reset to be equal, i.e., $\frac{1}{N}$. This algorithm is summarised in X.

Besides the simpliest multinomial resampling scheme, many different resampling schemes have been proposed in the literature. For example, stratified resampling [] as the name suggested splitting the samples in stratums to ensure the good coverage on the resulting sample set, residual resampling [] that has an effect in reducing the variance of the weights, etc. See [] for further details on the comparison of these sampling schemes.

However, resampling is not a silver bullet for sampling imporishment. Essentially, resampling provides a mechanism to eliminate low weight samples to give way to replicate *copies* of high weight samples. This allows all samples to participate and contribute to forma good estimation of the distribution of interest. This is obvious for the case of estimating filtering distribution and predictive distribution.

Over time, this replication reduces the number of distinct values available for previous time steps. The start of the trajectory will eventually become the same. This phenomena is known as sample impoverishment. This is a fundamental weakness of SMC, in which the history of the path is not re-written. The lose of diversity in the sample set will have a negative impact when it comes to estimating snoothing distribution.

2.5.5.1 Resample-Move Algorithm

To counteract this sample impoverishment, Resample-Move Algorithm [] is proposed to introduce some perturbance to the samples (so to diversify them) without changing the distribution they represent. This is accomplished by using MCMC steps with a Markov Kernel, K that is invariant to the target distribution.

In the original paper, this is done in a way by introducing an additional MCMC "move" step to each sample after resampling step according to a Markov kernel that is invariant to the target distribution. This algorithm is summarised in X.

This does not entirely solve the smoothing distribution estimation problem. To apply Markov Kernels with invariant distribution corresponding to the smoothing dis-

tribution, the space that Markov kernel is defined has to increase at each iteration. This implies the computation time increases linearly with time. Moreover, fast mixing high dimension Markov kernel in itself is not easy to design. In practice, one could use a sliding windows approach, in which MCMC Kernels which diversify the samples of the previous n time step at each iteration. This has a *fixed* additional cost to each iteration.

2.5.6 Effective sample size (ESS)

Resampling step induces additional Monte Carlo variance to the weights. Yet, this step is necessary to avoids accumulation of estimation variance on weights over time and therefore result in a more stable approximation to the filtering and preditive distribution.

To trading off these two competing requirements, one possible way to monitor the effective sample size (ESS) which provides a measure on the quality of the weighted samples. Two possible estimation are as follows:

$$ESS \approx \frac{1}{E[w^2]} \approx \frac{\left(\sum_{i=0}^{N} w_i\right)^2}{\sum_{i=0}^{N} w_i^2} \tag{2.26}$$

A possible implementation is that resampling step is only triggered if the ESS_t faill below certain threshold at time t, say $N/2$. See [] for detail discussion on ESS.

2.5.7 Rao-blackwellised (Marginal) Important sampling

2.6 Conclusion

This chapter presents a review of Monte Carlo method, with a particular focus on Sequential Monte Carlo method that is used extensively in this thesis for portfolio optimisation. It begins to describe traditional Monte Carlo sampling techniques. It then introduce a simple SMC algorithm, and couple of extensions that have been proposed to improve the performance of the algorithm. Lastly, the chapter presents a simple concrete example, in which SMC is used to estimate the filtering and smoothing distribution for the problem in question.

It is worth nothing the algorithm is not restricted to sequential filtering problem. For example, it has been established that it is possible to use SMC within MCMC framework (pMCMC, where p stands for particle) [**?**] to solve other problems. In the

next chapter, we will show Sequential Monte Carlo is used as a maximiser to search for a optimal strategy for a portfolio, given a multiplicative reward function.

Chapter 3

Portfolio optimisation

3.1 Technical Approach

Traditionally, multi-period mean-variance portfolio optimization have been explired in an analytical fashion, adopting necessary assumption as necessary. This seems rather restrictive; there are many instances where numerical method has been used to derive an approximate or even more effective solution to the problem in question. For example, Monte Carlo technique is used to do integral, evolutionary techniques applied in engineering domains, etc..

Our approach to the problem in this thesis is a radical one. We view a portfolio optimisation as a *stochastic* control problem. We adopt the Bayesian view and treat tehse parameters as random variables. The objective is to find the sequence of control parameters that minimize the control objective defined in terms of portfolio return and financial risk. We investigate the potential of using SMCs as the means of determining the optimal strategy, or at least excellent, strategies for multi-period mean-variance portfolio optimisation problem. The main reason of choosing SMCs is its ability to carry out *sequential* update on the posterior distribution over time fit well with parameter inference in stochastic process. Moreoever, these techniques have achieved significant success in their applications on many domains. Of course, other heuristic search tecniques are also potentially applicable.

We investigate in this chapter how SMCs can be used to search for the optimal control parameters for portfolio optimisation problem in financial market, i.e., what decision a portfolio manager need to do to minimize the portfolio's tracking error to the target's stock index over a finite time horizon. This chapter is organised as follows: In Section 3.2 shows how a portfolio optimisation problem can be re-casted into a parameter estimation problem in SMC framework. Section **??** details the experiments in

accessing how well this approach perform to track various target signals generated by some known models. Section **??** presents the experiments of examining the approach using real-world data collected from Yahoo finance. Section **??** details possible extensions that can be easily achieved with this framework with some concrete examples. Finally, Section **??** concludes this chapter with a summary of results.

3.2 Portfolio Optimisation

Resource allocation is a common challenge for every investor. In the investment decision making process, investors decide how much resources are allocated to different investable assets to form a portfolio with the aim to optimise the performance of the overall portfolio is better off than any other according to some criterion. The criterion can be different to the investors. (Some investors may have different considerations such as tax considerations and legal restriction on investment assets or holding periods.)

Two common objectives, often contradicting, are the financial return and the investment risk. The Markowitz's modern portfolio theory [**?**] proposes a portfolio selection framework. In Markowitz's model, it is assumed that investor attempt to maximize a portfolio's return and minimize the risk (measured by the variance of the portfolio re turn. Based on this criteria, the set of non-dominance portfolio is known as the *efficient porfolios*. Using variance as a ris measure has its limitation. Variance is a symmetric measure; an out performing asset (than the expected return) is deemed to as risky as an under perfoming one. Many alternatve risk measurements have been proposed, e.g., Sortino ratio, Conditional Value at Risk (CVaR), etc.. Refer [2] for details.

In the original Marowitz model, the investment decision problem is viewed and solved as a single time-step problem. In practice, investment often span across multi time-step period and adjustments may be made to the allocation periodically to achive better performance as needed. This problem is much more difficult to deal with because it is time inconsistent in the sense that an investment strategy that is optimal over the whole period may not be the optimal one over a sub-interval of the period. This violates the Bellman's Priciple of Optimality [10]. Consequently, dynamic programming approach is not applicable here.

For index tracker fund manager, the main objective of portfolio management is to track and replicate the exposure of a benchmark index[1]. Different investors have dif-

[1]The lack of active management generally makes the fund less vulnerable to change of management and has the advatanges of lower fees and taxes

ferent risk appetite and goals, yet it is safe assumed that investor attempt to maximize a portfolio's return and minimize the risk (measured by the variance of the portfolio return. However, there are other constraint an investor need to consider in practice, e.g., asset type, holiding periods, etc.

Different metrics have been introduced to quantify the mismatch between the performance of a fund and its benchmark. For example, tracking difference is the sum of absolute difference in returns between of a fund and its benchmark. Here, we adopt the tracking error as our metric, which is defined to be the standard deviation of the absolute difference of the returns of the fund and the benchmark defined in [**?**]. Formally, tracking error is defined to be:

$$\epsilon = \sqrt(E[(r_p - r_b)^2]) \tag{3.1}$$

The tracking error can be caused by differnet factors: some of which can be summarised as follows:

1. Benchmark index rebalance — the benchmark index is re-weighting its constitutents periodically to reflex the changes on the market based on its methodology. To track the index, the fund has to adjust its portfolio accordingly. This will incur some trasaction costs. During the index rebalance period, cash drag can happen between the liquidation of the constituents that have weights reduced/dropped and the addition of the constituents that have weights increased/added. This cash is essentially not participating in the market and therefore do not reflex the changes on the benchmark index.

2. different assumption on dividend trinvestment and taxation — This is best illustrate with examples. For example, the benchmark index calculation may assume immediate reinvestment of dividends on ex-dividend dates but the fund may only able to reinvest the dividend after receiving it. The tax treatment may on the dividends amy also different too.

3. Replication and Sampling techniques — funds may choose too replicate the benchmakr index by selecting a subset of the constituents (often the ones with larger weights and more liquid) in an effort to minimize the transaction costs. This exclusion of the smaller, less liquid constitutents may introduce another source od tracking error, especially under stressed market.

4. Total Expense Ratio — the average annual expense that is charged to the fund on daily basis to cover the management cost.

This list is by no mean exclusive. See [] for further detail.

3.3 Portfolio Optimisation as a Stochastic Control Problem

Traditionally, the state space models used in portfolio management are deterministic. For example, forcasting and predict f[29, 30], valuing electricity contracts for hedging in deregulated electricity markets, the short term available wind power and the temperature driven consumer demand (see [28, 29, 30] and the references within), or when examining the power transfer fluctuations across transmission lines [27]. TeIt seems there is a pressing interest for stochastic modelling from computational statistics when analytical solution is not available.

We focus here the problem in minimizing the tracking error between a portfolio and its benchmark index using a stochastic control modelling approach. Our aim is to determine what investment actions (buy or sell) on the necessary constituents a portfolio manager has to do on a daily basis across the investment horizon to minimize the tracking error of the fund managed. We will proceed by presenting the stochastic state space model that we assume throughout this thesis. This model is by no mean to compete the state of the art model in realistic portfolio optimisation, but to motivate further work in this direction.

MOdel and objective function

3.4 Technical Approach

The technical approach is to

Portfolio managers is facing constant challenge

3.5 Objective in portfolio optimisation

We adopt the conditional linear Gaussian model with the following form:

$$X_t = A_t(U_t)X_{t-1} + B_tU_tW_t + F_tU_t, W_t \sim N(0, I) \tag{3.2}$$

$$Y_t = C_t(U_t)X_{t-1} + D_tU_tV_t + G_tU_t, N_t \sim N(0, I) \tag{3.3}$$

where A_n,B_n, C_n, D_n, F_n, and G_n are appropriate matrix/vector functions, $U_{tt\geq 0}$ is a determinstic control input sequence that is used regulate the hidden states, $X_{tt\geq 0}$, which are assumed to be a discrete time hidden Markov process that has an initial value x_0 and admit Gaussian transition density $f_t(x_t \mid x_{t_1}, u_t$ and $Y_{tt\geq 0}$ is the only observable process which has a Gaussian conditional likelihood density $g_n(y_n \mid x_n, u_n)$.

With this model, the objective is to search for a sequence of controls $u_{1:t}$ that would result in a sequence of observations $y_{1:t}$ is closed to a reference signal $y_{1:T}^{ref}$. This problem is often known as stochastic regulation problem. We adopt here the following finite horizon multiplicative reward function:

$$1 + 1 \tag{3.4}$$

where the expectation is take with respect to the whole path of the Markov Chain $X_{0:T}$, i.e., $E_{x_0}[f(X_{1:t})] = \int f(X_{1:T}) \prod f_t(x_t \mid x_{t-1})\ dx_{1:t}$, with Q_n, L_n are assumed to be known. The corresponding optimal open loop policy is:

$$u_{1:T}^* = \arg\max_{u_{1:T}} J(u_{1:T}; y_{1:t}^{ref}; x_0) \tag{3.5}$$

3.6 Technical approach

Under the Bayesian inference framework, we treat the control inputs as random variables that admit a prior distribution. Assuming the sequenc

The key point to note is that for the model becomes a linear Gussian model for a given u_t, which allows us to solve x_t analytically using Kalman Filter algorithm.

Under the realm of Bayesian inference framework, we treat U as a random variable and admit a probability distribution. The objective is to compute the marginal posterior distribution density $p(u_{0:t} \mid y_{0:t}$.

This desntiy function can be derived as follows:

$$-1 + 1 \tag{3.6}$$

We can solve this equation using SMC by consider the pair etc.

However, a more efficient algorithm can be derived by considering the following factorisation:

$$p(x_{0:t}, u_{0:t} \mid y_{1:t}) = p(x_{0:t} \mid u_{0:t}, y_{1:t}) p(u_{0:t} \mid y_{1:t}) \tag{3.7}$$

Note that density $P(x_{0:t} \mid u_{0:t}, y_{1:t}$ is Gaussian mixture model, which can be computed analytically usign Kalman Filfter given the density $p(u_{0:t} \mid y_{1:t}$, which has the following recursion form:

$$1 + 1 \tag{3.8}$$

This can be resolved using SMC by using particles to do this.

3.7 Problem formulation

Assume for the time being that the control inputs are set as U1:n = u1:n and remain fixed. Recall Xnn0 and Ynn1 are assumed to be stochastic processes obeying a Markov transition density fn(xn—xn1,un) and a conditionally independent likelihood density gn(yn—xn,un) respectively. Given any observed y1:n realisation, inference about the states X1:n may be based on the following posterior density

show the algorithm

3.8 Numerical example on stationary oscillating wave

We will consider a simple linear Gaussian state space model as presented earlier as (3.3), with $A_t = B_t = C_t = D_t = I$, $F_t u_t = u_t$, $G_t u_t = 0$, $Xi = 0$, $u = 0$. This model can be re-written as follows:

$$X_t = X_{t-1} + W_t + U_t, W_t \sim N(0, I) \tag{3.9}$$

$$Y_t = X_{t-1} + V_t, N_t \sim N(0, I) \tag{3.10}$$

with the target reference is set to be an oscillating wave: $y_t = \cos(0.2\pi t + 0.3)$. This toy example is first introduced in **??**. It serves two purposes here. Firstly, it provides a simple example to verify our implementation[1] Secondly, it severs as benchmark for the following experiments in which we attempt to use more complicated reference signals and models.

Setting the maximum time period, $T = 50$,

We proceed by examining the algorithm for the following different implementations: (a) qn = fn without using the MCMC move (Step 2(d)), (b) qn = fn with the MCMC move and (c) qn being the optimal importance density of [13] without the MCMC case. In the last case the MCMC step was omitted because when the optimal importance density is used the improvement in performance was marginal. In the MCMC move we will use a random walk proposal. For = 100, 1000 and N = 200, 500, 1000, 5000, 10000 we present box plots for logT(UI1:T)inFigure1after30independentrunsofthealgorithm,whereUI1:T istheestimatorofu1:T ineach run. Similarly, in Figure 2 we plot U I1:T and the particle population nU i1:T oNi=1 taken from one run of the each of the same cases, but this time we show results only for N = 10000. Simulations took roughly 3, 70 and 4 seconds per 1000 particles for (a), (b), (c) respectively when implemented in Matlab using a

[1] Strictly speaking, testing increases confidence but does not prove no bug, which is almost impossible in practice.

2.4 GHz processor. The algorithm seems to perform quite well in most settings and very well when the optimal importance distribution is used. For the case where qn = fn, MCMC seems to improve the performance of the algorithm. The improvement is more evident when = 1000 both in the box plots and when plotting nUi1:T oNi=1, for which the degeneracy is apparent without the MCMC step.

3.8.1 Period length performance

We extend the time step to be 90 and 250 steps and looks at the corresponding performance. Due to the increase of time step, it makes sense to have more particles to track them and also look at the performance of performance of different order of γ. Based on the same metric, the results are summarised in Figure **??**.

3.8.2 Increasing the power

3.9 Different reference signals

Given the initial result looks promising, we attempt to investigate with the following more complicated reference signals:

1. reference signal treding osccilating wave —
2. two un-correlated bi-variate signals —
3. two correlated bi-variate signals —
4. ten different signals —

3.10 Discussion and possible extension

3.11 Possible parallel computation at different phases

3.12 Evidence for the thesis and future work

Formulating an optimal security policy is difficult. Current research work attempts to alleviate this issue by looking for ways to analyse and refine security policies in a top-down manner. We propose an alternative view on this issue: inferring security policies from decision examples. This idea is entirely novel. There is no previous work

to my knowledge in the application of EAs or machine learning techniques in inferring security policies.

In this chapter, we presents some experiments that have been carried out to validate this proposal using EAs. Three different ways of representing security policies and the use of two different EAs are demonstrated. The results show that the inference process is largely independent of many parameters. We also show how the fuzzy set ensemble based approaches can be easily integrated into the policy inference framework to enhance the inference ability, yet it remains an interesting research topic to search for the optimal ways of defining the underlying target fuzzy membership functions.

EAs have shown several potentials in determining the security policies in the experiments. In particular, EAs are found to be able to quickly infer security policies with considerable complexity. The performance of these inferred policies is comparable to the original reference models that are used to generate the training sets. These techniques are also able to scale well with the range of input/output variables and to tolerate "wrong" examples in a training set. An obvious way forward is to validate this concept with other inference approaches and make a recommendation on which approach is better for what circumstance.

Being a data driven approach, the representativeness of the training set is crucial. Indeed, the experiments show that even the inference of the simple MLS Bell-LaPadula model may fail because of this. Inference summarises rather than speculates; the techniques do not know how to handle an unseen case.

As in other applications of EAs, the fitness function used is vital in guiding the search. Poor fitness function may result in policies that are suboptimal. Interesting future work would be to examine how to design a fitness function in a principled manner that is suitable for cost sensitive learning, in which different types of prediction errors are not equally costly. This is likely to be appropriate in security policy in which leaking of high sensitivity information is obviously far more severe than leaking of low sensitivity information.

3.13 Conclusions

This chapter presents some proof-of-concept experiments that have been carried out to validate our proposal: inferring security policies from decision examples using EAs. It first presents the experiments on inferring some simple binary decision policies and continues with the experiments on inferring the Fuzzy MLS model, which is a more complicated multi-decision policy model. In all cases, the results show that EAs are

able to infer policies that can approximate (if not refine) the original reference models that are used to generate the training sets. The technique is also shown to be able to scale with the range of input/output variables and to tolerate "wrong" examples in the training set.

For a dynamic environment, the ability to infer policy from examples alone is not sufficient. The inferred and learnt policies will eventually become suboptimal over time as the operational requirements change. The policy needs to be updated continually to maintain its optimality. The next chapter demonstrates how multi-objective evolutionary algorithms (MOEAs) can be used to achieve this goal.

Chapter 4

Evaluation and conclusions

The work reported in previous chapters provides evidence to support the thesis hypothesis stated in Section, namely:

> Evolutionary algorithms (EAs) have the potential to be an effective means of determining the security policies that suit dynamic challenging environments.

This chapter reviews the work that has been done, evaluates the extent to which they justify the thesis hypothesis and concludes the thesis by addressing the directions for future work.

4.1 Evaluation

In previous three chapters, We have detailed several experimentations that serves to support the thesis hypothesis from three different strands of research. We explored the potential of EAs in inferring optimal security policies, dynamically updating security policies with new decision examples and searching for policies with optimal trade offs between objectives using simulation runs. This section summarises the work completed in each strand of research and outlines the contributions and novelty of the work presented in this thesis.

4.1.1 Static policy inference

Current security policy is often developed in a top-down approach. High-level security goals are first determined, after which they undergo a series of refinement processes to obtain the low-level executable rules. Although some work has been done in applying

machine learning techniques to aid the policy refinement process, there is no previous work to my knowledge in the application of EAs or machine learning techniques in inferring security policies.

Chapter details the experiments in using EAs to infer security policies from decision examples. Here EAs is used as a tool to generalise a set of low-level examples to a set of high-level rules. Various simple security policies have been attempted and inferred successfully. These include the traditional MLS Bell-LaPadula policy model, the budgetised MLS policy model and the Fuzzy MLS policy model. Two different EAs, namely GP and GE are used. In all cases, the results show that a minimal amount of design effort and domain knowledge are required to infer the reference policy or a close approximator of it. The only requirements are to have a good fitness function and training examples that form a good representation of the target policy.

The last part of the chapter presents how other machine techniques can be incorporated into the policy inference framework created. Fuzzy set concept is used as an example here. Multiple policies are learnt independently; each of which focuses on inferring a fuzzy rule for a particular class of decisions (fuzzification). The ultimate output policy, which is an ensemble of all these policies, is formed using a weighted voting mechanism (defuzzification). Various experiments have been carried out to examine different fuzzification and defuzzification techniques. The results show that these approaches can consistently infer policies that closely match with the original reference models used.

4.1.2 Dynamic policy inference

There will inevitably be times when unseen circumstances demand a decision during operation. In some cases the default automated response may be imperative; in other cases this may be ill-advised. Manual decisions made to override the default one essentially define a new policy. Furthermore, even if the optimal security policy can be developed or inferred automatically, it would eventually become suboptimal due to the changes in either the operational environment or security requirements, or both. Therefore, a security policy has to be able to continually change and be updated to suit the operational needs to maintain its optimality.

on dynamic security policy inference. As there is no dynamic security policy model available and therefore no decision example is available for us to work with, we designed a dynamic security policy model. This model is used to generate time varying decision examples for training and evaluation purposes.

To infer this dynamic security policy model, two novel dynamic learning frameworks based upon MOEAs are designed: one that based on Fan's intuition [11] and DOO. In DOO, an n-objective optimisation problem is treated as a $2n$-objective optimisation problem by adding an opposing objective for each of the original objectives. With such a setting, DOO is able to maintain the diversity among the individuals in the population whilst optimising the intended objectives. This diversity can aid in preventing the population from premature convergence and allows the concept drift in the policy to be continually relearnt. The results show that these frameworks are very promising. Reasonably good approximators to the model are able to be inferred from the examples using these frameworks.

4.1.3 Mission-specific policy discovery

Chapter introduces the notion of mission-specific policy discovery. EAs are used to search for the security policies that can provide the optimal, or at least excellent, tradeoffs among security objectives for a specific mission. Here, EAs serve as an optimisation tool to synthesise the optimal policies, in terms of achieving the mission as well as security objectives without violating the constraints given.

We demonstrate here how simulation can be used to obtain the fitnesses of the policy candidates that are used to guide the policy search. To evaluate the fitness of an individual (policy) for a mission, the policy is first plugged into a simulated mission, then the simulated mission is executed and the outcome of it is measured. This is very different from the practice of fitting a policy a priori without the details of the specific mission being taken into account. This concept of "mission-specific policy" is entirely novel.

Various EA baed techniques are used here to discover the optimal policies. These include GP/MOGP and DOO. In all cases, the results show that these techniques are able to discover the set of policies that are optimal for the mission of concern. By using MOGP (and MOEAs in general), tradeoffs between a variety of criteria can be explored. Such information can be valuable to policy makers to select and apply the optimal policy that best fits the current operation.

4.1.4 Thesis contributions

In summary, we demonstrate how:

- EAs can be used to infer static security policies from a set of decision examples. Three different ways of representing security policies and two different EAs are investigated. The results show that this idea is feasible.

- the fuzzy set concept can be integrated into the policy inference framework to improve the policy inference performance. The idea is sufficiently generic to be applied to other classification problems, provided that there is a partial ordering among the classes.

- multi-objective evolutionary algorithms (MOEAs) can be used to infer dynamic security policies from a set of decision examples. Two novel dynamic learning frameworks based upon MOEAs are developed: one that is based on Fan's intuition and DOO. Both of them can be used as general dynamic classification algorithms.

- an ensemble policy model can be constructed from multiple models in a single EA run to achieve better performance. The improvement is especially significant in the DOO setting.

- MOEAs can be used to infer a set of Pareto optimal policies that fit a specific mission (or at least a specific family of missions).

- simulation runs can be used in place of a set of decision examples to provide feedback in evaluating the fitness of a policy with respect to the specified high-level objectives.

- MOEAs can be used as a decision making tool where tradeoffs between objectives exists. The Pareto front of the security policies discovered using MOEAs can reveal useful information about the relationship among different objectives, which may be difficult to obtain otherwise. Such information provides useful insight for policy makers to select and apply the optimal policy that fits the needs of current operational environment on a case-by-case basis.

4.2 Envisaged future work

Having discussed the contributions of the thesis, we now outline numerous possible directions for future work that have been identified during the course of this research.

4.2.1 Policy fusion

In dynamic coalitions, parties with different policies can come together to collaborate. Prior to the formation of dynamic coalitions, each party may have its own security policy. An interesting step forward would be to investigate how well EAs could be used to combine these security policies together. One possible way is to generate decision examples from both existing policies and use these examples as the training input for the policy inference framework. MOEAs can also be used to discover the Pareto optimal set of policy candidates, which are then chosen depending on the security requirements. However there are still issues that require further investigation. These include:

- Understanding how to deal with policies that consists of different sets of decision-making factors, which may be measured using different scales.
- Understanding what the implicit priorities that EAs have assigned to the conflicting rules are, what the factors that influence the priorities are and how to control these priorities, etc.

4.2.2 The robustness of a security policy

The framework proposed in this thesis has been shown to be effective in dynamically inferring the optimal policy. However, the optimality of a policy is not always the only factor of concern; the robustness in performance of a security policy in different environments may be equally important. This is especially so in a pervasive operating environment where the deployment of a new policy can be a difficult or expensive process. To incorporate this factor into the proposed framework, a way to quantify the robustness in performance of a security policy is required.

This measure also provides a way to determine the invariant part of the optimal policies for different operational environments of concern. The determination of this invariant part is doubly useful: Firstly, it can serve as a template or testing target in the policy development process. Secondly, it can help to protect the security policy inference framework from poisoning attack, which attempts to mislead the inference process in the favour of the attacker by the injection of specially crafted decision examples.

4.2.3 Scalability with the training set size

Scalability is a subtle issue. We have addressed some aspects of this issue. For example, we have shown the method scales well with the size of the training set. In the

experiments presented in Chapter we have increased the size of the training set from 100 examples to 1000 examples and the results still remain consistent. Obviously, the fitness evaluation time would increase; 1000 examples take ten times longer than 100 examples to evaluate. This is unlike to be an issue in practice as the fitness evaluation of each individual can be executed in parallel if necessary. In Chapterwe have shown that DOO is able to evolve and update policies with decision examples in an incremental manner. However, there are still some issues remaining with these frameworks that need to be investigated. This includes searching for appropriate techniques to sample old decision examples and examining the generality of the DOO framework.

4.2.4 More complex security policies

The security policies used in this thesis are rather simple. This can be potentially an issue. However, note that these policies are either real-world policies or proposals from major research institutes for real world use. They are simple, but by no means "toy" policies. Ultimately, we should strive for simple policies wherever is possible, but at the same time, we should also need to acknowledge that MANET policies may need legitimately to be much more complicated. To cope with complexity, instead of attempting to extract and discover the policies as a whole, we could simply target the areas that we need help. Humans produce security policies sequentially too, i.e., they consider in turn authentication policy, file access control, audit policy, etc. In practice, it is also often that there are some rules of thumbs and constraints that are dictated from on high. We do not need to extract these bits of a policy. Yet, there is still much to answer here, for example:

- Can EAs be used to evolve more complex policies or policies of other types, e.g., obligation policies? If not, how can we divide the security policies into smaller components in a systematic manner?
- How to incorporate the constraints imposed from on high into the policy inference framework to form a continuous learning loop in an efficient manner? Should we take such constraints into consideration in the evolution process? If so, how?

4.2.5 More complex scenarios

The scenario used in Chapter is relatively simple. It has only one type of agent in each team and one type of information. A real test of this approach would be to embed it within a more realistic simulated scenario, with more sophisticated information types,

and realistic consequence models. Note that the simulated scenario may be much more complex but we are really interested in some of the measurable properties, which may only be few. For example, how many properties would an operational commander be interested in feasibly ready trading off? The techniques proposed here should be able to scale well with it.

4.3 Closing remarks

The work reported in this thesis demonstrates a considerable degree of originality supported by extensive experimentation. The case studies are necessarily limited given the limited amount of time frame. However, the results demonstrate that portfolio optimisation approaches using Sequential Monte Carlo techniques have very considerable promise. Everyone accepts that portfolio optimisation is difficult, and things are to worsen as the marketfinancial become more complex environments with increasing sophistication and subtlety of decision-making process. We recommend these approaches to the research community for further investigation.

References

[1] Harry Markowitz. Portfolio selection*. *The journal of finance*, 7(1):77–91, 1952. 1

[2] R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000. 1, 18

[3] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 5, 14, 15, 16

[4] Olivier Cappé, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 8

[5] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. 10

[6] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995. 10

[7] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953. 15

[8] W.K. Hastings. Monte carlo samping methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970. 15

[9] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6(6):721–741, November 1984. 16

[10] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957. 18

[11] Wei Fan. Systematic data selection to mine concept-drifting data streams. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 128–137, New York, NY, USA, 2004. ACM. 28