CHAPITRE 4

DSO FOR IV FORECASTING

As shown in Section 2.2, GA's have been used to extract patterns from the volatility time series and enable analysts to achieve a high forecasting accuracy. In this chapter we attempt to demonstrate that a Markov chain based discrete stochastic optimization (DSO) method could provide the theoretical support for applying GA's to forecast IV, if the IV time series is properly converted into a Markov chain. By employing this DSO method introduced by Andradottir (1995, 1999), we demonstrate the feasibility and convergence of GA's in case of time non-homogeneity. Viewed differently, the current work demonstrates the efficiency improvement which GA brings to the application of the stochastic optimization process in forecasting IV.

4.1 Literature Review

In order to model and forecast volatility, a wide variety of methods have been attempted in the last decade. Among literatures that use the traditional approaches, Barndorff-Nielsen & Shephard (2002) formed a general stochastic volatility model to estimate IV so that model based approaches can potentially lead to significant reductions of the mean square error. Working from the concepts of realized power variation and realized volatility, they provided a limiting distribution theory to strengthen the consistency results (Barndorff-Nielsen & Shephard 2003). They went on to provide a systematic study of kernel-based estimators of the integrated variance in the presence of market microstructure noise by deriving the optimal kernel-based estimator under an assumption that the noise is without memory and independent of the efficient price (Barndorff-Nielsen & Shephard 2004). Refer to Ma *et al.* (2004*a*, *b*) for a list of the related literature on this subject.

As discussed in the earlier chapters, it was concluded that GA's could prove to be the more practical and effective approach at present in tackling the stochastic optimization problems. Its advantage lies in the ease of variable coding and its inherent parallelism. The use of genotypes instead of phenotypes to travel in the search space makes them less likely to get stuck in local maxima. The GA approach employed in Ma *et al.* (2004*b*) satisfied some stringent criteria and yielded forecasting accuracy that is higher than those derived from other publicly available research. GA methods have, however certain drawbacks, *e.g.* GA's are not guaranteed to give an optimal solution and they lack convergence proof. In comparison to other stochastic optimization techniques such as simulated annealing, GA's lack rigorous supporting mathematical theory such as the one based on the principle of Markov chain (Pinto, 2000).

The recent advancements in discrete stochastic optimization methods provide the theoretical foundation to solidify the GA approach. For example, Andradottir (1995, 1999) demonstrated the feasibility of applying the Markov chain method when the transitional matrix is initially non-time homogeneous and asymptotically approaches time homogeneous state, unlike Duan *et al.* (2003) and most other work in the field, which are confined to time-homogeneous cases. However, the main difficulty while applying Markov chain theory to solve time series problems is that data in time series problems are typically correlated, while Markov chain by definition does not concern about the historical states prior to the current one. This is exemplified by the application of Markov chain method on the non-linear asymmetric GARCH(1,1) process, as done in Duan *et al.'s* research publication (2003). Therefore, one needs to transform a time series into a Markov chain while maintaining the necessary characteristics of the original data, in order to make use of the rigorous mathematical theory to substantiate the stochastic GA operation.

Duan *et al.* (2003) introduced the use of a time-homogenous Markov chain for the valuation of options, in which volatility determination is a key. The Markov chain approach allows one to decouple the partitioning of time and state. In other words, one can use time steps suitable for a particular contingent claim without being unduly constrained to have a particular set of state values, unlike other option valuation methods such as binomial tree or lattice and finite difference methods. Such a characteristic motivates the current IV data conversion into the overlapping four-lag recursive data groups, thus enabling the joining force of both Markov chain and GA's for the purpose of optimization. And this is another key contribution of this work, *i.e.* to substantiate the GA operation with Markov chain when applied to optimize the forecast of a volatility time series.

In the following sections we attempt to apply a Markov chain based Discrete Stochastic Optimization (DSO) method to substantiate the use of GA's in Ma *et al.*'s (2004*b*) published paper, since GA based DSO typically lacks rigorous mathematical proof. The key in such a process is to transform the IV time series data set into a cross-sectional one. It turns out that the 4-lag recursive transformation in the TSDM framework as described in Chapter 3 fulfils precisely such a purpose. In Section 4.2, Andradottir's (1999) global search DSO method is introduced, whereas Section 4.3, the method is applied to substantiate the use of GA's for volatility forecast.

4.2 Discrete Stochastic Optimization Method

In this section, a typical form of DSO or Discrete Stochastic Optimization problems is outlined while the basic concept, procedure as well as advantages and disadvantages of DSO are introduced. As shown in Ma *et al.* (2004*b*), the calculation of IV converts the volatility from a latent variable into an "observable" one. Upon certain conversion as shown in the next sections, the IV time series could become the random variable of a DSO process, thus allowing us to make use of the DSO method. This is demonstrated in the following sections.

4.2.1 Typical Markov Chain Approach

The following is the general form of a DSO problem for which the current approach we take, needs to determine global optimal solutions:

$$\max_{\theta\in\Theta} f(\theta), \quad \text{where } f(\theta)=E\{X(\theta)\}, \forall\theta\in\Theta. \tag{9}$$

a) Here, $f: \Theta \rightarrow \Re$ is the objective function, where $\Re$ is the domain of real numbers.

b) Θ is the discrete feasible region containing at least two states; in the current case, for a finite feasible set, $\Theta^* \neq 0$, where $\Theta^* = \{\theta \in\Theta : f(\theta)\geq f(\theta')$ for all $\theta \in\Theta\}$ $(\theta' \in\Theta\backslash\{\theta\})$ is the set of global optimal solutions to the optimization problem; since f: $\Theta \rightarrow \Re$, the optimal value $f^* = \max_{\theta\in\Theta} f(\theta)$ is finite and can be achieved.

c) $\{X(\theta):\theta \in\Theta\}$ is a collection of random variables having the property that $E\{X(\theta)\}$ cannot be evaluated analytically but estimated or measured.

d) θ is a random variable in a stochastic process.

e) And $X_l(\theta)$, …, $X_L(\theta)$ are independent and identically distributed (IID) observations of $X(\theta)$ for all $\theta\in\Theta$.

In seeking the solution for the next step, many traditional random search algorithms estimate the optimal solution by using either the feasible solution the method is currently exploring or the feasible solution visited most often so far. A feasible solution is one that corresponds to the state within $\tilde{\Theta}$. On the other hand, Andradottir (1996*b*) believe that their performance can be improved and proposed an alternative approach. Further details are given in the following section.

4.2.2 More Contemporary Approch

Andradottir proposed using *all the observed objective function values* generated as the random search method moves around the feasible region to obtain increasingly more accurate estimates of the objective function values at different points. At any given time, the feasible solution that has the best estimated objective function value, *e.g.* the largest one for maximization problems is used as the estimate of the optimal solution. At the same time, Andradottir specified the rate of convergence of this method and proved that it is guaranteed to converge almost surely to the set of global optimal solutions. Numerical evidence presented by Andradottir (1996*b*) and by Alrefaei and Andradottir (1996*a*, *c*) suggests that this approach for estimating the optimal solution appears to help yield improved performance relative to other approaches for estimating the optimal solution.

Andradottir's (1999) Lemma 3.1 assumes that P_m, $m = 0, 1, 2, \ldots$ and P are Markov matrices on the state space Θ such that P is irreducible and aperiodic and $P_m \rightarrow P$ as $m \rightarrow \infty$. If q: $\Theta \rightarrow \Re$, then as $M \rightarrow \infty$

$$\frac{1}{M}\sum_{m=1}^{M} q(\theta_m) \rightarrow \sum_{d=1}^{J} \pi_d q(d), \quad as\ M \rightarrow \infty, \tag{10}$$

where $\pi^T = (\pi_1, \ldots, \pi_J)$ is the steady-state distribution corresponding to P, while $\{X_m\}$ is a non-homogeneous Markov chain with transition probabilities

$$\begin{gathered} P\{\theta_{m+1} = d \mid \theta_0, \mathrm{K}\ \theta_m\} = P_m(\theta_m, d) \\ \forall d \in \Theta \wedge m = 0,1,2,\mathrm{K} \end{gathered} \tag{11}$$

In other words, at iteration $m+1$, θ_{m+1} has $d=J$ possible states. Here the number of states is countable and limited. At the limit, the transitional matrix becomes time-homogeneous, *e.g.* stable. Andradottir's (1999) preferable approach would involve maintaining two variables for each point $\theta \in \Theta$. One of these, say $\mathrm{K_m}(\theta)$ would count how many estimates of $f(\theta)$ have been generated in the first m iterations for the respective θ, while the other one $\sum_m(\theta)$ would contain the sum of all $\mathrm{K}_m(\theta)$ estimates of $f(\theta)$ that have been generated in the first m iterations. The specific procedure is outlined in Algorithme 1.

Algorithm 1 – Modified Global Search Method

Step 0: Select a starting point $\theta_0 \in \Theta$. Let $K_{-1}(\theta) = \Sigma_{-1}(\theta) = 0 \ \forall \ \theta \in \Theta$. Let $m = 0$ and $\theta^*_m = \theta_0$ and go to Step 1.

Step 1: Given the value of θ_m, generate a uniform random variable θ'_m on $N(\theta_m)$ independently of the past (so that $\forall \ \theta \in \Theta$, $\theta \neq \theta_m$, we have that $\theta'_m = \theta$ with probability $1 / (|\Theta| - 1)$). Go to Step 2.

Step 2: Given the value of θ_m and θ'_m, generate observations $X_{m,l}(\theta)$ of $X(\theta)$, for $l = 1, \ldots, L$ and $\theta = \theta_m, \theta'_m$ independently of the past. Let $R_m = \Sigma^L_{l=1} (X_{m,l}(\theta_m) - X_{m,l}(\theta'_m)/L$. if $R_m > 0$, then let $\theta_{m+1} = \theta_m$. Otherwise let $\theta_{m+1} = \theta'_m$. Go to Step 3.

Step 3: Let $K_m(\theta) = K_{m-1}(\theta) + L$ for $\theta = \theta_m, \theta'_m$, and $K_m(\theta) = K_{m-1}(\theta) \ \forall\theta \in \Theta \backslash \{\theta_m, \theta'_m\}$. Moreover, let $\sum_m(\theta) = \sum_{m-1}(\theta) + \sum^L_{l=l} X_{m,l}(\theta)$ for $\theta = \theta_m, \theta'_m$, and $\sum_m(\theta) = \sum_{m-1}(\theta) \ \forall\theta \in \Theta \backslash \{\theta_m, \theta'_m\}$. Let $\theta^*_m \in \arg\max_{\theta \in \Upsilon_m} \sum_m(\theta)/K_m(\theta)$, where $\Upsilon_m = \{\theta \in \Theta : K_m(\theta) > 0\}$. Let $m = m + 1$ and go to Step 1.

There is no particular requirement how θ, the solution should behave. On the other hand, $\{X(\theta):\theta \in \Theta\}$ should be a collection of random variables having the feature that $E\{X(\theta)\}$ is the unbiased and consistent estimation of $f(\theta)$. Details regarding the rationale of unbiased and consistent estimation of $f(\theta)$ are given in Section 4.4. The main issue in applying Algorithm 1 will be the way to use the state data generated by a random search method in order to obtain an estimate of the optimal solution. Here, solutions are first compared against each other pair by pair. Those solutions that have higher averaged $X(\theta)$ will be retained for the next generation, *i.e.* the selection of θ_{m+1} based on the value of R_m. Andradottir's (1999) approach requires the search of optimized solution to be identified in Step 3, where $K_m(\theta)$ and $\sum_m(\theta)$ for each $\theta \in \Theta$ are stored, accumulated and compared for maximization. At the last generation, among thousands of solutions in the memory the optimization is performed with $\theta_m^* \in \arg\max_{\theta \in \Upsilon_m} \sum_m(\theta)/K_m(\theta)$. The top solutions could be selected and used for validation by testing against other set of data. The fact that all values of θ_{m+1} are kept in memory while the optimization is ongoing makes it the key difference between Andradottir's method and others including the one used in Ma *et al.* (2004*b*). The detailed procedure to incorporate Andradottir's method with GA will be given in the next section, which can further accelerate the optimization process.

4.3 Proposed Methodology

To apply Algorithm 1, we classify the respective values in the IV time series into four ranges, *e.g.* $(-\infty, -a]$, $(-a, b]$, $(b, c]$ and (d, ∞). Therefore, all data will become a sequence of numbers, *e.g.* 1, 2, 3, & 4. We then define a set $\theta_j = \{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4}\}$, where $\theta \in \{1, 2, 3, 4, *\}$ is a state in Θ with * = *don't care* and $j = 1, \ldots, J$, where J is the total number of states as indicated in Eq. (11). In other words, θ could be defined as the successively overlapped 4-lag recursive data set that has been converted from the original IV time series. We then generate rules randomly in the form of <IF [(($\theta_t = I$) AND/OR ($\theta_{t+1} = K$) AND/OR (($\theta_{t+2} = L$)) AND/OR ($\theta_{t+3} = M$))], THEN ($\theta_{t+4} = N$)>, where the "IF" part {θ_t, θ_{t+1}, θ_{t+2}, θ_{t+3}, AND, OR, *} is used as the qualifying criteria and the "THEN" part {θ_{t+4}, *} is for predicting the subsequent IV value. Each rule will work as a sliding window to pass across the entire IV time series point by point. In light of Andradottier's Lemma 3.1 as shown in Eq. (9) and (10), such a plan is sound in our GA operation, because in practical sense it is acceptable to assume that there is a limited number of patterns existing in the IV time series, *e.g.* J. And we are looking for the rules that most frequently match with the overlapped 4-lag IV data. Those patterns that appear more often tend to be caught by rules derived from crossover and/or mutation, and will gradually lead to more successful estimates. Moreover, Theorem 1 as shown in Section 4.4 demonstrates when each estimate is obtained from a single trial (in our case, each generation in GA generates 100 rules per group of total 100 groups in which each rule is independent of each other) the random search method is of first order convergence (Andradottir, 1999). Upon satisfying the two key conditions, we could define the stochastic function as :

$$X(\theta) = \begin{cases} 1, & \text{if } \{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4}\} \text{ matches the data sequence;} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

where $\{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4}$, AND, OR, *$\}$ represents the rules. Consequently, $f(\theta)$ would be the expected prediction accuracy $E\{X(\theta)\}$ for a rule θ and could be any real value in [0, 1]. Each rule will be independent of any other rules or at least treated so in view of GA, thus satisfies the requirement of Markov chain operation, and the nature of $X(\theta)$ makes it IID as required in Eq. (9). The problem is therefore, converted into a search of rules that best fit the four-point patterns in the IV data set so that the immediate fifth IV value could be forecasted upon knowing the previous four points. In a more general sense, a time series problem is thus converted into a set of random data that could be approached with the Markov chain method.

To extend Andratottir's strategy of comparing rules pair by pair, we make use of the GA technique such as the tournament/elitist selection criterion to improve the chance of reaching the optimal objective function. As defined by Langdon (1996), tournament selection is

> *"a mechanism for choosing individuals from a population. A group (typically between 2 and 7 individuals) are selected at random from the population and the best (normally only one, but possibly more) is chosen. An elitist GA is one that always retains in the population the best individual found so far. Tournament selection is naturally elitist."*

In every generation, new rules in the groups that have been derived from crossover and mutation in the previous generation will be put back into the pool to be compared with those retained from the last generation. Only those new ones that have higher prediction rates will replace the respectively selected peers for the next generation. Either accepted or rejected they are recorded in memory together with other existing rules. Here, L is the smaller number of the possible matches derived by comparing θ_m and $\theta_m^{'}$ and is at maximum equals the number of data points in the IV time series minus four, while m is the number of generations to perform GA. One important feature GA incorporates in

Step 2 is the way of generating $X_{m,l}(\theta)$ of $X(\theta)$, for $l = 1, \ldots, L$ and $\theta = \theta_m, \theta'_m$ independently of the past. By applying GA, θ'_m are generated through crossover or mutation, while $X(\theta)$ depend on whether the qualified rule predicts correctly. With the value of R_m we could choose from θ_m and θ'_m, to make one of them go through further GA manipulation such as crossover or mutation. At the last generation, we could retain θ^*_m as the optimal solution for the m^{th} generation by carrying out the optimization process. Note that the calculation of $K_m(\theta)$ could be modified as

$$K_m(\theta) = \begin{cases} K_{m-1}(\theta) + L, & \text{if } \{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4}\} \\ & \text{matches the data sequence;} \\ K_{m-1}(\theta), & \text{otherwise.} \end{cases} \tag{13}$$

where $\{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}\}$ is again the qualifying part of the rule.

4.4 Proposed Procedure to Apply DSO with GA

When applying Algorithm 1 to solve the current discrete stochastic optimization problem, we obtain the following Algorithm 2:

Algorithm 2 – Discrete Stochastic Optimization with GA

Step 0: Randomly assign any one value of {1, 2, 3, 4, *} to the first four fields in $\theta = (\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4})$, randomly assign operators "AND" and "OR" to join these four fields and then assign $\theta_{t+4} = 1$ for the first 25 rules. Repeat the same process with $\theta_{t+4} = 2$, 3 and 4 respectively to form a total of 100 rules. Repeat the operation to generate another 99 such groups. Then randomly select 50 rules in each group as θ_0's. Set all counters to zeros.

Step 1: The rest of 50 rules in each group that have been generated in Step 0 will become θ'_m's. Or when $m > 0$ θ'_m are derived by applying crossover or mutation on the first four points and the three joining operators of rules in those ones rejected in Step 2 during the previous generation.

Step 2: Generate the random variable $X_{m,l}(\theta)$ by running the pair of rules respectively selected from θ_m and θ'_m sequentially through the entire IV data set. L would be the smaller of the two corresponding total matches for each θ_m and θ'_m. $X_{m,l}(\theta) = 1$ when predict correctly, 0 otherwise. Let $R_m = \sum^{L}_{l=1}(X_{m,l}(\theta_m) - X_{m,l}(\theta'_m))/L$. If $R_m > 0$, then let $\theta_{m+1} = \theta_m$. Otherwise let $\theta_{m+1} = \theta'_m$. Select another pair rules from θ_m and θ'_m and repeat the comparison procedure until obtaining 50 θ_{m+1} rules. 25 of the rejected rules will be used for crossover and the other 25 mutation at Step 1 in the next generation. Repeat the entire process for the rest of the 99 groups.

Step 3: $K_m(\theta)$ would be the total number of matches in the qualifying part of rules θ_m and θ'_m up to generation m, while $\sum_m(\theta)$ is the number of correct predictions for the corresponding rules. Increase the counter by 1 until reaching the preset limit. At the last generation, optimize among all rules stored in the memory based on the given criteria and retain the top 100 θ^*_m that could best forecast in the given data set, i.e. maximize the percentage of correct forecast by letting $\theta^*_m \in \arg\max_{\theta\in\Upsilon_m} \sum_m(\theta)/K_m(\theta)$, where $\Upsilon_m = \{\theta \in \Theta : K_m(\theta) > 0\}$. In ranking all stored rules, among those rules that are numerically identical, qualified and predicting correctly only the one has minimum "*don't care*" fields and "OR" operators will be retained.

At Step 0 generation 0, first rule is generated to take a value of θ_0 and the success rate of prediction to be zero. For whatever value of θ_0 we generate a different rule based on criteria given in Step 1. At Step 1 we apply the GA techniques such as tournament/elitist selection criterion, crossover and mutation to generate rules for comparison. At Step 2, we generate the expected outcome $X_m(\theta)$ for both rules by comparing each rule with all data points in the IV series. In carrying on the same process to the next point in the data set till completion, we find the respective L. For generation m>1, we only need to go through this process for θ'_m while values of $X_m(\theta)$ and L for θ_m have been derived in the previous generation. If θ'_m have higher rates of success, replace the current rules with the more successful ones and keep them in memory as θ_{m+1}. In such an operation, the same θ_{m+1} from different groups could appear more than once as indicated in Step 1, and it will yield the same $X(\theta)$ as before due to the nature of the data set. But only one of them should be registered when they predict better than the current best θ_m. In order to comply with Algorithm 1, we could incorporate a screening mechanism firstly to reject rules that are the same as those currently exist in the memory and secondly to reject rules that are identically qualified and correctly predicting in the current generation. This is necessary because in Andratottir's algorithm (1995), θ'_m that is the same as previous θ's will be rejected in Step 2. This process is repeated in parallel for all 100 groups. At step 3, we calculate for the optimal solution θ^*_{m+1} at the last generation based on the corresponding number of correct predictions, *i.e.* determine the rules that maximizes the prediction among all retained rules. Once the top 100 rules are derived, we could use them to predict another set of IV data especially those at an immediately subsequent time period in order to confirm the validity of the approach.

4.5 Rate of Convergence

Yan and Mukai (1992) defined the rate of convergence of the algorithm to be the rate at which the distribution of θ_m in Algorithm 1 converges to an optimal distribution, *i.e.* only puts a positive mass on elements of Θ^*. In other words, rate of convergence of a random search method for DSO is the rate at which the estimated value of the objective function at the estimated optimal solution converges to the optimal values of the objective function.

Theorem 1: Rate of convergence of Random Search Methods [1]. Assume that

a) $\Theta^* \neq 0$ and is finite;

b) The estimate of the optimal solution $\theta^*_{m+1} \in \arg\max_{\theta \in \Upsilon_m} \sum_m(\theta)/K_m(\theta)$ in Algorithm 2 converges almost surely to the set Θ^* as $m \to \infty$. Since $\sum_m(\theta)$ is the number of correct predictions while $K_m(\theta)$ is the number of hits, i.e. the number of matches between the first four points of the rule and the 4-lag recursive points in the IV data set, as $m \to \infty$, $K_m(\theta) \to \infty$. From the Strong Law of Large Numbers, consistent and unbiased solutions exist [2];

c) For all $\theta \in \Theta^*$, the estimate of $f(\theta)$ (obtained from single trials, i.e. at a certain value of m) are independent and identically distributed with mean $-\infty < f(\theta) < \infty$ and variance $0 < \sigma^2 < \infty$; If Θ is finite and for all $\theta \in \Theta$ we have $|f(\theta)| < \infty$, the estimates of $f(\theta)$ here are IID [2]. Since the rules are initially randomly generated, and each rule is independent of each other; rules after randomly crossover and mutated are also independent. Moreover, they are generated in a similarly random fashion, therefore it is understandable that the rate of correct prediction for all rules at each iteration is IID.

d) The estimates $f(\theta)$ are independent of the estimates of $f(\theta')$ for all $\theta' \in \Theta^* \backslash \{\theta\}$ (when each estimate is obtained from a single trial); and there exists a constant $0 < c(\theta) < \infty$ and a sequence $\{a_m\}$ of constants such that as $m \to \infty$, $a_m \to \infty$ and $K_m(\theta) / a_m \to c(\theta)$. (i.e. $K_m(\theta)$ can be tracked so that it is feasible for each θ to have a distinguishable value of $K_m(\theta)$.) We then have

$$\sqrt{a_m}\left(\frac{\sum_m\left(\theta^*_{m+1}\right)}{K_m\left(\theta^*_{m+1}\right)} - \min_{\theta\in\Theta} f(\theta)\right) \Rightarrow \min_{\theta\in\Theta^*} Z(\theta), \text{ as } m \to \infty \qquad (14)$$

where $\forall\theta \in \Theta^*$, the random variables $Z(\theta)$ are independent and

$$Z(\theta) \sim N(0, \frac{\sigma^2(\theta)}{c(\theta)}). \qquad (15)$$

The recently available real time data base for equity, indices, foreign exchanges or even fixed incomes makes it sensible to assume $m \to \infty$. However, one may need to take into account properties associated with the nature of financial markets. For example, the micro-structure of the equity bid-ask prices makes it difficult to use data that have higher frequency than say one reading in every 15 minutes for the purpose of volatility evaluation (Andersen, 1998 & 2001, Barndorff-Nielsen, 2004). Moreover, data patterns that occurred more than one or two years earlier may have little influence on the recent data, thus may not be applicable in the current volatility forecasting process. Further evidence and discussion on this issue is given in Chapter 6.