

Volatility Forecast by Discrete Stochastic Optimization and Genetic Algorithms

Irwin Ma¹ Tony Wong¹ Thiagas Sankar¹ Lisa Li²

¹Department of Automated Manufacturing Engineering
École de technologie supérieure, University of Québec, Montréal (Québec), Canada

²Positron Canada Ltd

Abstract

The daily volatility is crucial in the study of financial risks. In an earlier attempt, a group of rules is successfully trained by Genetic Algorithms (GA) to extract patterns from the Integrated Volatility (IV) time series enabling analysts to achieve a forecasting accuracy of 75%. The current paper substantiates such a use of GA with a Markov chain based discrete stochastic optimization method. By transforming the IV time series into a Markov chain, it proves the link between the two methods in case of time non-homogeneity and convergence. Viewed differently, it demonstrates the efficiency improvement GA brought to the application of the stochastic optimization method in the forecast of IV.

Keywords: Volatility, financial index, forecasting, stochastic optimization, evolutionary algorithm

1. Introduction

To an equity option trader, volatility is a measurement of an underlying stock's fluctuation in price, and calculated as the stock's standard deviation of returns. It is a critical factor in calculating an option's current theoretical value. With the forecast of volatility, one could directly trade it by longing and shorting the Chicago Board Options Exchange's Volatility Index Futures. Volatility is typically an unobservable random variable. It is thus not easy to predict where it heads in the next time step; let alone what value it is going to take.

In order to model and forecast volatility, a wide variety of methods have been attempted in the last decade. Refer to Ma et al. [1] for a list of the related literature. It was concluded that Genetic Algorithms (GA) seem to be the more practical and effective approach at present in tackling the stochastic optimization problems. The GA approach employed in [1] 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. Compared with other stochastic optimization techniques such as simulated annealing, it is less rigorous.

The current advancement in discrete stochastic optimization methods provides the theoretical foundation to solidify the GA approach. For example, Andradottir [2] demonstrated the feasibility of applying the Markov chain method when the transitional matrix is initially non-time homogeneous and asymptotically approaches time homogeneous, unlike Duan [3] 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 problem 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 point. 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 [3] research. Therefore, one needs to transform a time series into a Markov chain while maintaining the necessary characteristic of the original data, in order to make use of the rigorous mathematical theory to substantiate the stochastic optimization such as GA's.

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 method. Such a characteristic motivates the current IV data conversion into the overlapping four lag data groups thus making the optimization feasible when use both Markov chain and GA together. And that is the 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 method (DSOM) to substantiate the use of GA's in Ma et al. [1], which typically lacks rigorous mathematical proofs. In Section 2 Andradottir's [2] global search method for discrete stochastic optimization is

introduced. In Section 3, the method is applied to substantiate the use of GA's for volatility forecast. In Section 4, we generally discuss the approach with respect to its limitations and potential applications. Finally, conclusions are drawn and recommendations for future work are outlined in Section 5.

2. Discrete Stochastic Optimization

The following is the general form of a discrete optimization problem that needs to determine global optimal solutions:

$$\max_{\theta \in \Theta} f(\theta), f(\theta) = E\{X(\theta)\}, \forall \theta \in \Theta. \quad (1)$$

In Eq. (1), θ is a random variable in a stochastic process, Θ is the discrete feasible region containing at least two states and $f: \Theta \rightarrow P$ is the objective function. In the current case, for a finite feasible set, $\Theta^* \neq \emptyset$, where $\Theta^* = \{\theta : f(\theta) \geq f(\theta'), \forall \theta' \in \Theta \wedge \theta' \in \Theta \setminus \{\theta\}\}$ and Θ^* is the set of global optimal solutions to the optimization problem. Since $f: \Theta \rightarrow P$, the optimal value $f^* = \max_{\theta \in \Theta} f(\theta)$ is finite and can be achieved. The collection of random variables $\{X(\theta) : \theta \in \Theta\}$ has the property that $E\{X(\theta)\}$ cannot be evaluated analytically but estimated or measured. Rather than sequentially using either the current point or the most frequently visited point to estimate the optimal solution, Andradottir [2] 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. Numerical evidence presented by Andradottir [2] suggests that the use of this approach for estimating the optimal solution appears to yield improved performance relative to other approaches for estimating the optimal solution.

2.1. Contemporary approach

Andradottir's [2] Lemma 3.1 assumes that P_m , $m = 0, 1, 2, \dots$, 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 $g: \Theta \rightarrow P$, then as $M \rightarrow \infty$

$$\frac{1}{M} \sum_{m=1}^M g(\theta_m) \rightarrow \sum_{d=1}^J \pi_d g(d) \quad (2)$$

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

$$P\{\theta_{m+1} = d | \theta_0, \dots, \theta_m\} = P_m(\theta_m, d), \quad \forall d \in \Theta \text{ and } m = 0, 1, 2, \dots, \quad (3)$$

In other words at iteration $m+1$, θ_{m+1} has $d = J$ possible states. Here the number of states is countable while θ

could be, in the case of Ma et al. [1] the successively overlapped 4-lag recursive data set that has been converted from the original IV time series. And it takes the states in the form of numerical values $\theta \in \{0, 1, 2, 3, 4, *\}$, and $j = 1, \dots, J$. θ is a typical rule with numerical values joined by logical operators "AND" and "OR". Details about the structure of the rules will be given in section 3. At the limit, the transitional matrix becomes time-homogeneous. By using Theorem 1 shown in section 3.2 and our GA operation, we look for rules that most frequently match with the actual 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.

Andradottir's method [2] needs to maintain two variables for each point $\theta \in \Theta$, i.e. $K_m(\theta)$ would count how many estimates of $f(\theta)$ have been generated in the first m iterations, while $\Sigma_m(\theta)$ would contain the sum of all $K_m(\theta)$ estimates of $f(\theta)$ that have been generated in the first m iterations. The specific procedure is outlined as follows:

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 $l = 1, \dots, L$ and $\theta = \theta_m, \theta'_m$ independently of the past. Let $R_m = \Sigma_{l=1}^L (X_{m,l}(\theta_m) - X_{m,l}(\theta'_m)) / L$. 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 $K_m(\theta) = K_{m-1}(\theta) \forall \theta \in \Theta \setminus \{\theta_m, \theta'_m\}$. Moreover, let $\Sigma_m(\theta) = \Sigma_{m-1}(\theta) + \sum_{l=1}^L X_{m,l}(\theta)$ for $\theta = \theta_m, \theta'_m$, and $\Sigma_m(\theta) = \Sigma_{m-1}(\theta) \forall \theta \in \Theta \setminus \{\theta_m, \theta'_m\}$. Let $\theta_m^* \in \arg \max_{\theta \in \Upsilon_m} \Sigma_m(\theta) / K_m(\theta)$, where $\Upsilon_m = \{\theta \in \Theta : K_m(\theta) > 0\}$. Let $m = m + 1$ and go to Step 1.

The main issue in using 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. There is no particular requirement how θ , the rule 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 $f(\theta)$, i.e. the prediction accuracy recorded at the current generation among all best rules. Details regarding the rationale of unbiased and consistent estimation of $f(\theta)$ are given in Section 3. GA operation in Ma et al [1] repetitively

loops through Step 1 through 3 until the last generation. The main differences include a) evaluating far more rules per iteration, b) ranking instead of tournament as the selection method and c) limited number of rules instead of all historical rules involved in fitness value optimization. Here, rules in each group are first compared against each other pair by pair. Those rules that have better prediction rate will be retained for the next generation, i.e. the selection of θ_{m+1} based on the value of R_m . Andradottir's approach [2] 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. This is the key difference between Andradottir's method and others including the one used in Ma et al [1], i.e. all values of θ_{m+1} are kept in memory while GA's are ongoing. 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 compare 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. At the last generation, among thousands of rules 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 100 non-identical rules will be used for validation on another set of IV data. The detailed procedure in applying Andradottir's method with GA will be given in Section 3.1.

3. Proposed methodology

Algorithm 1 is quite general form of optimization without explicit assumption on the variables. To apply Algorithm 1, we classify the IV time series into four ranges, e.g. $(-\infty, -a]$, $(-a, b]$, $(b, c]$, (d, ∞) according to the respective values. Therefore all data will become a sequence of numbers, e.g. 1, 2, 3 & 4. Generate rules randomly in the form $\langle \text{IF } [((\theta_t = \text{I}) \text{ AND/OR } (\theta_{t+1} = \text{J}) \text{ AND/OR } ((\theta_{t+2} = \text{K})) \text{ AND/OR } (\theta_{t+3} = \text{L}))], \text{ THEN } (\theta_{t+4} = \text{M}) \rangle$, where the first four elements in the "IF" part are used as the qualifying criteria and the "THEN" part is for predicting the subsequent IV value. We define a set $\theta = \{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4}\}$, where $\theta \in \{1, 2, 3, 4, *\}$, with $*$ = *don't care*. As a result, the collection of random variable

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

where $\{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4}\}$ represents the complete rule including the qualifying part $\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}$ and the prediction part θ_{t+4} . The nature of $X(\theta)$ makes it IID as required in Eq. (1). 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. Each rule with five recursive points in θ 's will be independent of each other or at least treated as independent in the eye of GA's, thus satisfies the requirement of Markov chain operation. A time series problem is thus converted into a set of random data that could be approached with the Markov chain method. Here, L is the smaller number of the possible matches derived by comparing θ_m and θ'_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's incorporate in Step 2 is the way of generating $X_{m,l}(\theta)$ of $X(\theta)$, for $l=1, \dots, L$ and $\theta = \theta_m, \theta'_m$ independently of the past. By applying GA's, θ'_m are generated through crossover or mutation, while $X(\theta)$ depends on whether the qualified rule predicts correctly. With the value of R_m we could choose either θ_m or θ'_m to go through further GA manipulation, i.e. 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

$$K_m(\theta) = \begin{cases} K_{m-1}(\theta) + L, & \text{if } \{\theta_t, \theta_{t+1}, \theta_{t+2}, \theta_{t+3}, \theta_{t+4}\} \text{ matches the data sequence;} \\ K_{m-1}(\theta), & \text{otherwise.} \end{cases} \quad (5)$$

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

3.1. Applying DSOM with GA

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

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 . $X_{m,i}(\theta) = 1$ when predict correctly, 0 otherwise. Let $R_m = \sum_{i=1}^L (X_{m,i}(\theta_m) - X_{m,i}(\theta'_m))/L$. 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 would make use of the GA technique such as the tournament/elitist selection criterion to improve the chance of reaching the optimal objective function. 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. 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, θ'_m which 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, at the last generation we calculate for the optimal solutions θ^*_{m+1} based on the corresponding number of correct predictions, i.e. to determine the rules that maximize 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 a subsequent time period in order to confirm the validity of the approach.

3.2. Rate of convergence

Yan and Mukai [3] 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 discrete stochastic optimization 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

— $\Theta^* \neq \emptyset$ and is finite;

— 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 $m \rightarrow \infty$. 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 $m \rightarrow \infty$, $K_m(\theta) \rightarrow \infty$. From the Strong Law of Large Numbers, consistent and unbiased solutions exist [2];

— 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.

— The estimates $f(\theta)$ are independent of the estimates of $f(\theta')$ for all $\theta' \in \Theta^* \setminus \{\theta\}$ (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 $m \rightarrow \infty$, $a_m \rightarrow \infty$ and $K_m(\theta) / a_m$

→ $c(\theta)$. (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

$$\sqrt{a_m} \left(\frac{\sum_m (\theta_{m+1}^*)}{K_m(\theta_{m+1}^*)} - \min_{\theta \in \Theta} f(\theta) \right) \Rightarrow \min_{\theta \in \Theta^*} Z(\theta) \quad (6)$$

as $m \rightarrow \infty$

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

$$Z(\theta) \sim N\left(0, \frac{\sigma^2(\theta)}{c(\theta)}\right). \quad (7)$$

4. Limitations of the proposed method

Indeed, Andradottir's local and global search methods are based on the assumptions of initially non-homogeneous but asymptotically time homogeneous Markov transition matrix as $m \rightarrow \infty$, while other assumptions are easy to satisfy, i.e. irreducible, aperiodic, etc. [2]. Such a principle of time averages for non-homogeneous Markov chains may be applicable to our case, because the 100 rules found by GA are derived from matching the 100 most popular patterns in the IV data set through an evolutionary process. The 100 popular patterns may not necessary be at the steady-state because the limited number of generations and size of the available data set. In other words, the required conditions for applying Andradottir's method are stronger than what we actually possess. This could be seen from the following test results. The S&P500 data sets, first 2002 then 2002~2003 are processed with the Matlab programs based on Algorithm 2 in order to forecast 2004 IV's. The averaged forecast accuracies after 75 generations converge to 80 and 70% respectively, i.e. shorter training time yields better forecasts. However, Andradottir's approach provides at least a basic mathematical foundation for further development. The second limitation is closely related with the first one. Andradottir assumed that unbiased estimates of the objective function values are available. In particular, if $X_1(\theta), \dots, X_L(\theta)$ are IID observations of $X(\theta)$ for all $\theta \in \Theta$, then $\sum_{l=1}^L X_l(\theta)/L$ is an unbiased estimate of $f(\theta)$ for all $L \in \mathbb{N}$ and $\theta \in \Theta$. And at current time we will need to accept such an assumption prior to the GA operation with a limited data set, which leads to limited m .

5. Conclusions

As detailed in [1], our GA approach provides the flexibility and guideline in determining the forecasting horizon based on the values of entropy of the respective wavelet coefficients by incorporating the wavelet transform. This will be more reasonable than

arbitrarily select the forecasting horizon based on users' experience or requirement in other prevalent approaches. The method's concentration on abrupt changes and the use of 4-lags does not limit it for analysis of longer term volatility activities. The current 4-lag could be configured to deal with hourly, weekly or even monthly data. Moreover, 5-lag, 6-lag or more could be used to provide forecast for longer time horizon. With an improved prediction of volatility, we could proceed to trade volatility itself. One way to do so would be to use the newly formed VIX Futures in the CFE (Chicago Futures Exchange), namely VXB.

The current research took one step further to substantiate the application of GA's for forecasting of random variables such as the volatility of a financial index by making use of the more vigorous mathematics such as Markov chain. Viewed differently, GA's are a tool applied to improve the efficiency of Markov chain approaches in solving the discrete stochastic optimization problems. The discrete stochastic optimization method introduced by Andradottir [2] provides the guideline to search for the optimal solution in a random process. The GA operation accelerates the optimization process by performing the search in a larger scale, parallel and progressively targeted fashion. Consequently, the two techniques complement each other. The Markov chain method serves as the mathematical foundation while GA's improve the efficiency and confidence in the search of the optimized solutions.

Further clarification may be needed to alleviate the limitations listed in Section 4. For example, we still need vigorous proof for applying the current method to a data set that has limited number of data points, which leads to a limited L .

6. References

- [1] I. Ma, T. Wong, T. Sankar, "Forecasting the volatility of financial index by evolutionary algorithms," *IEEE International Conference on systems, Man & Cybernetics*, October 10 – 13, The Hague, Netherlands, pp. 5824–5829, 2004.
- [2] S. Andradottir, "Accelerating the convergence of random search methods for discrete stochastic optimization," *ACM Trans. Model. Comput. Simul.*, vol. 9, no. 4, pp. 349–380, 1999.
- [3] J. Duan, G. Gautier, J. G. Simonato, "A Markov chain method for pricing contingent claims," *Stochastic Modeling and Optimization with applications in queues, finance and supply chains*, Springer-Verlag, New-York, 2003.
- [4] D. Yan, H. Mukai, "Stochastic discrete optimization," *SIAM Journal Control and Optimization*, vol. 30, no. 3, pp. 594–612, 1992.