A Reliable Global Calibration Method for Affine Jump-Diffusion Models

Seungho Yang [a] Jaewook Lee*

[a]*Department of Industrial and Management Engineering*
Pohang University of Science and Technology (POSTECH)
San 31 Hyoja Pohang 790-784 South Korea

Abstract

This study presents the efficiency and practical model calibration method to use lots' of affine jump-diffusion models in real option market and it gives a useful guideline to trader in the situation of selecting a model. A general affine jump-diffusion model which have three dynamics of asset price, variance of asset return and jump intensity is constructed and based on the general model twelve different affine jump-diffusion models are reconstructed. To calibrate twelve different affine jump-diffusion models the multi-basin particle swarm intelligence method is proposed which consists of three steps to conquer the local minimum problem and show that it can resolve it. To verify the efficiency of the proposed method we use S&P 500 index option prices. It is found that the optimization method calibrate the parameter set well and jumps should be included to have a good calibration performance.

Key words: Option markets, Affine jump-diffusion models, Optimization method.

[1] *Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, Kyungbuk 790-784, Korea. Email: *jaewookl@postech.ac.kr*

1 Introduction

Black and Scholes developed the methodology for pricing options for the first time in 1973, which is the well-known Black and Scholes (B-S) model. In the B-S model the dynamics of log stock prices follows a geometric Brownian motion with constant drift and diffusion parameters. According to numerous empirical evidences, it is clear that the B-S model can not explain the properties of option prices in real markets such as the "volatility smile", that is, the options with different strikes and time-to-maturities have different Black-Scholes implied volatilities (Balland, 2002; Cont & Fonseca, 2002; Dumas et al., 1998)[2,8,11]). To explain the smile effects many option pricing models have been developed and the models can be classified into non-parametric and parametric method. Non-parametric method does not have pre-assumed models and usually uses learning algorithms such as neural networks. Hutchinson (1994)[17] and Gencay (2001)[14] show the non-parametric method performs well with respect to out-of-sample error and hedging error. However it has serious drawbacks in that it cannot applied to pricing path-dependent exotic options because of its lack of underlying dynamics. To understand intrinsic properties of asset returns and volatility smile effects of traded option prices and obtain more realistic models of stock prices, researches have proposed various models in a parametric way such as exponential Lévy models and affine jump-diffusion models (Christopher, 2003[10]).

In this paper affine jump-diffusion models are treated intensively. Among affine jump-diffusion models most popular model was developed by Heston (1993)[16]. However stochastic volatility models have a serious drawback in that it cannot handle short-term smiles properly. At the same time the stock process which adds jumps to the dynamic of stock price has been developed to obtain more realistic models of stock prices. Jump-diffusion models can explain the volatility "smile" effect. They have disadvantages of producing increasing at-the-money-forward term structure, because jump models shifts up the overall volatility level. Including jumps in the stochastic volatility model can conquer those drawbacks of each models and proposed by Bates (1996)[4]. It can be constructed the affine jump-diffusion models which include a stochastic jump intensity rate in the Bates model, which is presented by Fang (2000)[13]. It's very challenging model, although the performance of this model is not known. Through the empirical evidences of the proposed method it is checked the performance of the model.

To calculate option prices of the affine jump-diffusion models the FFT approach is most widely used. Due to the singularity of the payoff function at $v = 0$ we can not directly apply the FFT approach. Carr and Madan (1999)[6] makes it possible introducing an damping parameter for the first time. When option prices are quoted on the market, a market-consistent risk-neutral pricing model $\mathbb{Q}$ can not be obtained only by an econometric analysis of the time series of the underlying but by looking at prices of contingent claims today ($t = 0$). One choose a risk-neutral model

such as to reproduce the prices of traded options, called **model calibration**, and then uses this model to price exotic, illiquid or OTC options and to compute hedge ratios. To conduct model calibration several methods proposed in parametric and nonparametric way. Cont (2004)[9] proposed nonparametric way to calibrate Lèvy measures using relative entropy. Bakshi (1997) [1] proposed parametric way to calibrate stochastic volatility models dividing the time to maturity and moneyness space. Schoutens (2004) [26] insists they can calibrate perfectly real option prices. Since model calibration is known as ill-posed **inverse problem**, the calibration results have lots of local minima and can be very different depending on the initial parameter set and optimization methods in a non-smooth way, and real market data have a noise, it's impossible to calibrate option prices perfectly. However, most paper does not consider those problems seriously.

In this article, Twelve different affine jump-diffusion models which have three dynamics which are the asset price, the volatility of the asset return and the jump intensity are reconstructed and the new optimization method, which has best performance to calibrate complex financial models robustly, is proposed. With the proposed method a parameter set of affine jump-diffusion models is calibrated using model-generated option prices and S&P 500 index option prices for presenting empirical evidences. It is found that it can solve the ill-posed inverse problems and calibrate the parameter set well with the proposed method and affine jump-diffusion models including jump have better performance without it.

The article is structured as follows: affine jump-diffusion models are reviewed and a brief overview of the Carr-Madan's FFT method in the section 2. In the section 3 it is presented new optimization method which consists of three steps. To check the robustness of the proposed method it is conducted simulation with model-generated option prices in section 4. With the robustness of the method the real market option prices, which are S&P 500 index option prices, are used. The description of the S&P 500 index option prices are explained and the calibration results of affine jump-diffusion models are shown in the section 5. The last section 6 draws the conclusions and gives final comments.

2 Affine Jump-Diffusion Models

The most widely used continuous models for asset dynamics are the affine jump diffusion models. This section briefly reviews these models and reconstruct them using the unified notations.

Consider first the following affine jump-diffusion models with three dynamics un-

der risk-neutral measure $\mathbb{Q}$:

$$\begin{aligned}
\frac{dS_t}{S_t} &= (r - q - \lambda_t \mu_J)dt + \sqrt{\upsilon_t} dW_{St}^{\mathbb{Q}} + (e^{J_t} - 1)dN_{St}^{\mathbb{Q}}, \\
d\upsilon_t &= \kappa(\eta - \upsilon_t)dt + \xi\sqrt{\upsilon_t} dW_{\upsilon t}^{\mathbb{Q}}, \\
d\lambda_t &= \kappa_\lambda(\eta_\lambda - \lambda_t)dt + \xi_\lambda\sqrt{\lambda_t} dW_{\lambda t}^{\mathbb{Q}}.
\end{aligned} \tag{1}$$

(i) For the asset price dynamics, S_t is the asset price, υ_t is the variance of asset return, r is the risk-free (domestic) interest rate, q is the continuous dividend yield (foreign interest rate), $W_{St}^{\mathbb{Q}}$ and $W_{\upsilon t}^{\mathbb{Q}}$ are correlated standard Wiener processes under $\mathbb{Q}$ with constant correlation ρ, defined by $\mathrm{Cov}(dW_{St}^{\mathbb{Q}}, dW_{\upsilon t}^{\mathbb{Q}}) = \rho dt$. $N_{St}^{\mathbb{Q}}$ is a Poisson process under $\mathbb{Q}$ with a stochastic intensity λ_t, J is a random jump size in the logarithm of the asset price with a probability density function (PDF) given by $\varpi(J)$. $\mu_J = \mathbb{E}[e^J - 1]$ is set to make the discounted asset process a martingale. (ii) For the variance dynamics, κ is the mean-reverting rate, η is the long-term mean variance, ξ is the volatility of volatility. (iii) For the jump intensity dynamics, κ_λ is the mean-reverting rate, η_λ is the long-term mean rate, ξ_λ is volatility of jump rate intensity, a Wiener process $dW_{\lambda t}^{\mathbb{Q}}$ is independent of $W_{St}^{\mathbb{Q}}$ and $W_{\upsilon t}^{\mathbb{Q}}$. It is also assumed that jump processes are independent of Wiener processes. Note that under risk-neutral measure $\mathbb{Q}$, $\mathbb{E}^{\mathbb{Q}}(dS) = (r - q)Sdt$.

Using the martingale pricing, option value can be represented as an integral of a discounted probability density times the payoff function and the Feynman-Kac theorem is applied to derive the PIDE satisfied by the value of an option. Variables are changed from S to $x = \ln S$ and from t to $\tau = T - t$. Applying Feynman-Kac theorem for the price dynamics (1), it is obtained that the value of a European-style claim denoted by $f(x, \upsilon, \lambda, \tau)$ satisfies the following partial integro-differential equation (PIDE)

$$\begin{aligned}
&-\frac{\partial f}{\partial \tau} + (r - q - \frac{1}{2}\upsilon - \lambda\mu_J)\frac{\partial f}{\partial x} + \frac{1}{2}\upsilon\frac{\partial^2 f}{\partial x^2} + \kappa(\eta - \upsilon)\frac{\partial f}{\partial \upsilon} + \frac{1}{2}\xi^2\upsilon\frac{\partial^2 f}{\partial \upsilon^2} \\
&+ \rho\xi\upsilon\frac{\partial^2 f}{\partial x \partial \upsilon} + \kappa_\lambda(\eta_\lambda - \lambda)\frac{\partial f}{\partial \lambda} + \frac{1}{2}\xi_\lambda^2\lambda\frac{\partial^2 f}{\partial \lambda^2} + \lambda\int_{-\infty}^{\infty}\varpi(J)dJ[f(x + J) - f(x)] = rf \\
&f(x, \upsilon, \lambda, 0) = g(e^x, K)
\end{aligned} \tag{2}$$

where g is the payoff function.

2.1 Solutions to Moment Generating Function

One of the nice property for the affine jump-diffusion models is that their option pricing formulas can be easily obtained with the aid of the moment generating

function (MGF). The MGF $G(u, x, \upsilon, \lambda, \tau)$ associated with the log of the terminal asset price $x(\tau) = \ln S(\tau)$ in the stochastic differential equation (1) is defined by

$$G(u, x, \upsilon, \lambda, \tau) = \mathbb{E}^{\mathbb{Q}}[e^{ux(\tau)}] = e^{-r\tau}\mathbb{E}^{\mathbb{Q}}[e^{r\tau}e^{ux(\tau)}]. \tag{3}$$

and can be considered as a contingent claim that pays off $e^{r\tau+ux}$ at time τ. Again through Feynman-Kac theorem the $G(u, x, \upsilon, \lambda, \tau)$ of (1) can be achieved by solving the following PIDE.

$$\begin{aligned} &- G_\tau + (r - q - \frac{1}{2}\upsilon - \lambda\mu_J)G_x + \frac{1}{2}\upsilon G_{xx} + \kappa(\eta - \upsilon)G_\upsilon + \frac{1}{2}\xi^2\upsilon G_{\upsilon\upsilon} \\ &+ \rho\xi\upsilon G_{x\upsilon} + \kappa_\lambda(\eta_\lambda - \lambda)G_\lambda + \frac{1}{2}\xi_\lambda^2\lambda G_{\lambda\lambda} \\ &+ \lambda\int_{-\infty}^{\infty}[G(x + J) - G]\varpi(J)dJ = 0, \qquad G(u, x, \upsilon, \lambda, 0) = e^{ux}. \end{aligned} \tag{4}$$

By the method of indetermined coefficients guessing of an affine-form $G = e^{A(\tau)+B(\tau)\upsilon+C(\tau)\lambda}$ PIDE (4) can be solved in closed-form as follows.

Proposition 1 *The solution to PIDE (4) is given by*

$$G(u, x, \upsilon, \lambda, \tau) = \exp\{xu + (r - q)\tau u + A(u, \tau) + B(u, \tau)\upsilon + C(u, \tau) + D(u, \tau)\lambda\} \tag{5}$$

where

$$\begin{aligned} &A(u,\tau) = -\frac{\kappa\eta}{\xi^2}[\psi_+\tau + 2\ln(\frac{\psi_- + \psi_+e^{-\zeta\tau}}{2\zeta})], \quad B(u,\tau) = -(u - u^2)\frac{1 - e^{-\zeta\tau}}{\psi_- + \psi_+e^{-\zeta\tau}}, \\ &A_d(u,\tau) = -(\kappa\tau + e^{-\kappa\tau} - 1)\frac{\eta(u - u^2)}{2\kappa}, \quad B_d(u,\tau) = -(1 - e^{-\kappa\tau})\frac{(u - u^2)}{2\kappa} \\ &C(u,\tau) = -\frac{\kappa_\lambda\eta_\lambda}{\xi_\lambda^2}[\chi_+\tau + 2\ln(\frac{\chi_- + \chi_+e^{-\epsilon\tau}}{2\xi})], \quad D(u,\tau) = 2\Lambda(u)\frac{1 - e^{-\epsilon\tau}}{\chi_- + \chi_+e^{-\epsilon\tau}}, \\ &C_d(u,\tau) = (\kappa_\lambda\tau + e^{-\kappa_\lambda\tau} - 1)\frac{\eta_\lambda\Lambda(u)}{\kappa_\lambda}, \quad D_d(u,\tau) = (1 - e^{-\kappa_\lambda\tau})\frac{\Lambda(u)}{\kappa_\lambda}, \\ &\psi_\pm = \mp(\kappa - \rho\xi u) + \zeta,\ \zeta = \sqrt{(\kappa - \rho\xi u)^2 + \xi^2(u - u^2)}, \\ &\chi_\pm = \mp\kappa_\lambda + \epsilon,\ \epsilon = \sqrt{\kappa_\lambda^2 - 2\xi_\lambda^2\Lambda(u)}, \\ &\Lambda(u) = \int_{-\infty}^{\infty}e^{Ju}\varpi(J)dJ - 1 - \mu_J u, \quad \mu_J = \int_{\infty}^{\infty}e^{J}\varpi(J)dJ - 1. \end{aligned}$$

where $A = A_d$, $B = B_d$ and $C = C_d$, $D = D_d$ in case of deterministic volatility and jump intensity, respectively.

The following table summarize the detailed parameters of the affine jump-diffusion models according to the dynamics of the variance and jump intensity and jump size distribution that follows log-normal distribution. **Const**, **Deter**, **Stoch**, **Vol** and **Inten** mean Constant, Deterministic, Stochastic, Volatility and Intensity.

Table 1
Summary of the Moment generating functions under Affine Jump-Diffusion Models

Model	Model Description	Moment Generating Function
CVNJ	Const Vol No Jump	$A(u,\tau)=0, B(u,\tau)=-(u-u^2)\tau/2,$ $C(u,\tau)=0, D(u,\tau)=0$
CVCI	Const Vol Const Inten	$A(u,\tau)=0, B(u,\tau)=-(u-u^2)\tau/2,$ $C(u,\tau)=0, D(u,\tau)=\tau\Lambda(u)$
CVDI	Const Vol Deter Inten	$A(u,\tau)=0, B(u,\tau)=-(u-u^2)\tau/2,$ $C(u,\tau)=(\kappa_\lambda\tau+e^{-\kappa_\lambda\tau}-1)\frac{\eta_\lambda\Lambda(u)}{\kappa_\lambda}, D(u,\tau)=(1-e^{-\kappa_\lambda\tau})\frac{\Lambda(u)}{\kappa_\lambda}$
CVSI	Const Vol Stoch Inten	$A(u,\tau)=0, B(u,\tau)=-(u-u^2)\tau/2,$ $C(u,\tau)=-\frac{\kappa_\lambda\eta_\lambda}{\xi_\lambda^2}[\chi_+\tau+2\ln(\frac{\chi_-+\chi_+e^{-\epsilon\tau}}{2\xi})], D(u,\tau)=(1-e^{-\kappa_\lambda\tau})\frac{\Lambda(u)}{\kappa_\lambda}$
DVNJ	Deter Vol No Jump	$A(u,\tau)=-(\kappa\tau+e^{-\kappa\tau}-1)\frac{\eta(u-u^2)}{2\kappa}, B(u,\tau)=-(1-e^{-\kappa\tau})\frac{(u-u^2)}{2\kappa},$ $C(u,\tau)=0, D(u,\tau)=0$
DVCI	Deter Vol Const Inten	$A(u,\tau)=-(\kappa\tau+e^{-\kappa\tau}-1)\frac{\eta(u-u^2)}{2\kappa}, B(u,\tau)=-(1-e^{-\kappa\tau})\frac{(u-u^2)}{2\kappa},$ $C(u,\tau)\equiv 0, D(u,\tau)=\tau\Lambda(u)$
DVDI	Deter Vol Deter Inten	$A(u,\tau)=-(\kappa\tau+e^{-\kappa\tau}-1)\frac{\eta(u-u^2)}{2\kappa}, B(u,\tau)=-(1-e^{-\kappa\tau})\frac{(u-u^2)}{2\kappa},$ $C(u,\tau)=(\kappa_\lambda\tau+e^{-\kappa_\lambda\tau}-1)\frac{\eta_\lambda\Lambda(u)}{\kappa_\lambda}, D(u,\tau)=(1-e^{-\kappa_\lambda\tau})\frac{\Lambda(u)}{\kappa_\lambda}$
DVSI	Deter Vol Stoch Inten	$A(u,\tau)=-(\kappa\tau+e^{-\kappa\tau}-1)\frac{\eta(u-u^2)}{2\kappa}, B(u,\tau)=-(1-e^{-\kappa\tau})\frac{(u-u^2)}{2\kappa},$ $C(u,\tau)=-\frac{\kappa_\lambda\eta_\lambda}{\xi_\lambda^2}[\chi_+\tau+2\ln(\frac{\chi_-+\chi_+e^{-\epsilon\tau}}{2\xi})], D(u,\tau)=(1-e^{-\kappa_\lambda\tau})\frac{\Lambda(u)}{\kappa_\lambda}$
SVNJ	Stoch Vol No Jump	$A(u,\tau)=-\frac{\kappa\eta}{\xi^2}[\psi_+\tau+2\ln(\frac{\psi_-+\psi_+e^{-\zeta\tau}}{2\zeta})], B(u,\tau)=-(u-u^2)\frac{1-e^{-\zeta\tau}}{\psi_-+\psi_+e^{-\zeta\tau}},$ $C(u,\tau)=0, D(u,\tau)=0$
SVCI	Stoch Vol Const Inten	$A(u,\tau)=-\frac{\kappa\eta}{\xi^2}[\psi_+\tau+2\ln(\frac{\psi_-+\psi_+e^{-\zeta\tau}}{2\zeta})], B(u,\tau)=-(u-u^2)\frac{1-e^{-\zeta\tau}}{\psi_-+\psi_+e^{-\zeta\tau}},$ $C(u,\tau)\equiv 0, D(u,\tau)=\tau\Lambda(u)$
SVDI	Stoch Vol Deter Inten	$A(u,\tau)=-\frac{\kappa\eta}{\xi^2}[\psi_+\tau+2\ln(\frac{\psi_-+\psi_+e^{-\zeta\tau}}{2\zeta})], B(u,\tau)=-(u-u^2)\frac{1-e^{-\zeta\tau}}{\psi_-+\psi_+e^{-\zeta\tau}},$ $C(u,\tau)=(\kappa_\lambda\tau+e^{-\kappa_\lambda\tau}-1)\frac{\eta_\lambda\Lambda(u)}{\kappa_\lambda}, D(u,\tau)=(1-e^{-\kappa_\lambda\tau})\frac{\Lambda(u)}{\kappa_\lambda}$
SVSI	Stoch Vol Stoch Inten	$A(u,\tau)=-\frac{\kappa\eta}{\xi^2}[\psi_+\tau+2\ln(\frac{\psi_-+\psi_+e^{-\zeta\tau}}{2\zeta})], B(u,\tau)=-(u-u^2)\frac{1-e^{-\zeta\tau}}{\psi_-+\psi_+e^{-\zeta\tau}},$ $C(u,\tau)=-\frac{\kappa_\lambda\eta_\lambda}{\xi_\lambda^2}[\chi_+\tau+2\ln(\frac{\chi_-+\chi_+e^{-\epsilon\tau}}{2\xi})], D(u,\tau)=(1-e^{-\kappa_\lambda\tau})\frac{\Lambda(u)}{\kappa_\lambda}$
	Other parameters	$\psi_\pm=\mp(\kappa-\rho\xi u)+\zeta,\ \zeta=\sqrt{(\kappa-\rho\xi u)^2+\xi^2(u-u^2)}$ $\chi_\pm=\mp\kappa_\lambda+\epsilon,\ \epsilon=\sqrt{\kappa_\lambda^2-2\xi_\lambda^2\Lambda(u)}$ $\Lambda(u)=e^{\mu_J u+\delta^2u^2/2}-1-u(e^{\mu_J+\delta_J^2/2}-1)$ (log-normal)

2.2 Option Pricing formulas

During the last decade, Fourier transform has been exclusively employed to express option pricing formula under the affine jump-diffusion models. Next three widely-used Fourier transform methods for option pricing are summarized ([16,4,6,21]). Here it is assumed that the characteristic function $\Phi_x(\cdot)$ defined by $\Phi_x(z)=\mathbb{E}[e^{izx_T}]=G(iz)$ (where G is the moment generating function (MGF)) is analytic and bounded in the strip $0\le\Im(z)\le 1$ where $x_T=\ln S_T$ and $k=\ln K$ to guarantee the existence of the Fourier transform.

- **The Black-Scholes-style Formula:** The current value of European call and put

options are given by

$$Call_t = e^{-q(T-t)} S_t \mathcal{P}_1^+ - e^{-r(T-t)} K \mathcal{P}_2^+. \tag{6}$$
$$Put_t = e^{-r(T-t)} K \mathcal{P}_2^- - e^{-q(T-t)} S_t \mathcal{P}_1^-. \tag{7}$$

where

$$\mathcal{P}_1^+ = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \mathrm{Re}\left[\frac{\Phi_x^{(1)}(z)}{iz}\right] dz, \quad \mathcal{P}_2^+ = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \mathrm{Re}\left[\frac{\Phi_x^{(2)}(z)}{iz}\right] dz$$
$$\mathcal{P}_1^- = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty \mathrm{Re}\left[\frac{\Phi_x^{(1)}(z)}{iz}\right] dz, \quad \mathcal{P}_2^- = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty \mathrm{Re}\left[\frac{\Phi_x^{(2)}(z)}{iz}\right] dz$$

and $\Phi_x^{(1)}(z) = e^{-\ln S_t - (r-q)(T-t)} \Phi_x(z-i) e^{-izk}$ and $\Phi_x^{(2)}(z) = \Phi_x(z) e^{-izk}$. Here $\mathcal{P}_1^+$ represents a delta, Δ, and $\mathcal{P}_2^+$ represents the real possibility for exercising an option. $\mathcal{P}_2^+$ is also equal to the value of a digital call option, so the Black-Scholes-style formula is widely used for many exotic structure embedded with digital options.

- **Lewis' Characteristic Formula:** The current value of European call and put options are given by

$$Call_t = e^{x(t)-q(T-t)} - \frac{e^{-r(T-t)} K}{\pi} \int_0^\infty \Re\left[\frac{Q(u)}{u^2 + 1/4}\right] du. \tag{8}$$
$$Put_t = e^{-r(T-t)} K - \frac{e^{-r(T-t)} K}{\pi} \int_0^\infty \Re\left[\frac{Q(u)}{u^2 + 1/4}\right] du. \tag{9}$$

where $Q(u) := e^{-(-iu+1/2)\ln K} \Phi_x(-u - i\frac{1}{2}) = e^{-(-iu+1/2)\ln K} G(-iu + \frac{1}{2})$.

- **Carr-Madan's Formula:** Let α be a positive constant such that the α-th moment of the stock price exists. The European call option price of T maturity and strike price K is then given by

$$C(k,T) = \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{+\infty} e^{-ivk} \psi_T(v) dv. \tag{10}$$

where $k = \ln(K)$ and

$$\psi_T(z) = \frac{e^{-rT} \Phi_x(u - (\alpha+1)i)}{\alpha^2 + \alpha - u^2 + i(2\alpha+1)u}.$$

The delta is then given by

$$\Delta = \frac{e^{-\alpha k}}{2 S_t \pi} \int_{-\infty}^{+\infty} e^{-ivk} (iu + (\alpha+1)) \psi_T(v) dv. \tag{11}$$

Now to compute (10) for N-log strike levels k ranging from $-b$ to b, i.e.

$$k_u = -b + \delta_k (u-1), \quad \text{for } u = 1, ..., N, \;\; \delta_k = 2b/N,$$

The fast Fourier transform (FFT) method is applied to (10) which is an $O(N \ln N)$ algorithm for computing

$$w(u) \approx \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)} x(j) \text{ for } u = 1, ..., N,$$

where N is a power of 2. Using the Simpson's rule for the integral part in (10) where the upper limit for the integration is set to be $2\pi/\delta_k$ and setting $h = 2\pi/(N\delta_k)$ and $v_j = h(j-1)$, the following equation is obtained:

$$\begin{aligned} C_T(k_u) &\approx \frac{e^{-\alpha k_u}}{\pi} \sum_{j=1}^{N} e^{-iv_j k_u} \psi_T(v_j) h, \quad \text{for } u = 1, ..., N \\ &\approx \frac{e^{-\alpha k_u}}{\pi} \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)} [e^{i\pi(j-1)} \psi_T(v_j) h] \end{aligned}$$

Among the three transform methods, Carr-Madan's Fourier transform method is shown to be very efficient in pricing several options with the same maturity.

3 The Proposed Method

The proposed method is composed of three steps. The first step is a gradient-based local-searching step. The second step is to look for a sup-local minimum from the local minimum of the first step. The third phase is to find the global optimal value based on the particle swarm optimization starting from the sup-local minimum. We will look for the parameter set which satisfies the Calibration Problem.

Calibration Problem *(Least squares calibration)* Given a parameter set (θ), the observed prices C_i of call options for maturities T_i and strikes $K_i, i \in I$, find a parameter vector θ which minimizes

$$\arg\min_{\mathbf{Q}_\theta \in \mathcal{Q}} \sum_{i=1}^{N} \omega_i |C^\theta(T_i, K_i) - C_i|^2 \tag{12}$$

where C^θ denotes the dollar adjusted call option price and $\mathcal{Q}$ is the set of martingale measures. ω_i is the weight of the difference of the option prices. In the proposed method we will set $\sum_{i=1}^{N} \omega_i |C^\theta(T_i, K_i) - C_i|^2$ as $f(\theta)$

Implied volatilities can be used to measure the differences between model and market instead of option prices. However, it takes lots of time and it is inefficient because implied volatilities are calculated in each step. In some parameter values of optimization procedure implied volatilities are unstable. Hence option prices are used to calibrate the parameter set of affine jump-diffusion models.

3.1 Step I: Local Search Phase

In Step I we look for a (1st order) local minimum. We construct a generalized negative gradient system represented by

$$\frac{d\theta}{dt} = -\mathrm{grad}_R f(\theta) \equiv -R(\theta)^{-1}\nabla f(\theta). \tag{13}$$

where $R(\theta)$ is a positive definite symmetric matrix for all $\theta \in \Re^n$. Such an R is called a *Riemannian metric* on $\Re^n$.

Many local search algorithms can be seen as a discretized implementation of process (13) depending on the choice of $R(\theta)$. We assume that f is twice differentiable to assure the existence of a unique solution (or trajectory) $\theta(\cdot) : \Re \to \Re^n$ for each initial condition $\theta(0)$. Note that it can be shown that the trajectory $\theta(\cdot)$ is defined on all $t \in \Re$ for any initial condition $\theta(0)$ under a suitable re-parametrization. A state vector $\bar{\theta}$ satisfying the equation $\nabla f(\bar{\theta}) = 0$ is called an equilibrium point (or critical point) of system (13). We say that an equilibrium point $\bar{\theta}$ of (13) is hyperbolic if the Hessian of f at $\bar{\theta}$, denoted by $H_f(\bar{\theta})$, has no zero eigenvalues (i.e., $H_f(\bar{\theta})$ is positive definite). Note that all the eigenvalues of the Jacobian of a gradient system are real since they are symmetric matrices. A hyperbolic equilibrium point is called a (asymptotically) stable equilibrium point (or an attractor) if all the eigenvalues of its corresponding Jacobian are positive. A hyperbolic equilibrium point $\bar{\theta}$ is called an index-k equilibrium point if the Jacobian of the hyperbolic equilibrium point has exactly k negative eigenvalues. The basin of attraction of a stable equilibrium point θ_s is similarly defined as

$$A(\theta_s) := \{\theta(0) \in \Re^n : \lim_{t\to\infty} \theta(t) = \theta_s\}.$$

The basin of attraction $A(\mathbf{x}_s)$ is an open and connected set. One nice property of this formulation is that every local minimum of the optimization problem (12) corresponds to a (asymptotically) stable equilibrium point of system (13). Hence, looking for the local minima of (12) can be achieved via locating corresponding stable equilibrium points of (13). Another nice property is that all the generalized gradient system have the equilibrium points at the same locations with the same type. Due to convenient properties we can design a computationally efficient algorithm to find stable equilibrium points of (13), and thus find local optimal solutions of (12).

With numerically integrating system (13) or by a steepest-descent method we can simply implement Step I. However, since we are primarily concerned with finding a stable equilibrium point, which is a limit point of a trajectory, rather than the trajectory itself, we can choose any robust local search algorithm if it can locate a local minimum nearby an initial guess. From the 'locality' of search in step I it is

important not to take too large steps initially for the line search employed by a local search method to avoid jumping out of the current basin of attraction.

3.2 Step II: Discrete Sup-Local Search Step

One important issue is how to escape from a local minimum of Step I and move on toward another neighboring local minimum. We can construct a graph $G = (V, E)$ describing the connections between the local minima with the following elements:

1. The vertices V of G are local minima $\theta_s^1, ..., \theta_s^p$ of (12), where p is the total number of stable equilibrium points.
2. The edge E of G can only connect two local minima adjacent to each other; θ_s^i is connected with θ_s^j if, and only if, θ_s^i is adjacent to $\mathbf{x}_s^j$.

The original optimization problem (12) can now be transformed into the following combinatorial optimization problem

$$\min_{\theta_s \in \mathcal{L}^1} f(\theta_s) \tag{14}$$

where $\mathcal{L}^1$ is the set of all the local minima. Utilizing the concept of a neighboring local minimum, Step II uses a discrete local search strategy on $\mathcal{L}^1$ to solve problem (14). Specifically, we first construct a user-defined neighborhood $\mathcal{N}$ where the neighborhood $\mathcal{N}(\theta_s) \subset \mathcal{L}^1$ of $\theta_s \in \mathcal{L}^1$ is defined as the set of multiple neighboring local minima adjacent to (or near) θ_s. With respect to the neighborhood $\mathcal{N}$, we will call a point $\theta^* \in \mathcal{L}^1$ a sup-local minimum if $f(\theta^*) \leq f(\mathbf{y}_s)$ whenever $\mathbf{y}_s \in \mathcal{N}(\theta_s)$. Then we perform a discrete local search strategy to locate a sup-local minimum.

We may apply the following direct strategy to locate a neighboring local minimum adjacent to a given local minimum: In order to escape from a local minimum, say θ_s, we first move in reverse time from θ_s to an index-1 equilibrium point, say θ_1. Then, starting from θ_1, we advance time to move towards an adjacent local minimum, say $\mathbf{y}_s$. The latter step can also be implemented using a local search algorithm as in Step I. The former step, although it is hard to conduct directly and takes lots of time, can also be achieved by using some of the techniques suggested in [20]. However this direct strategy can be inefficient since we are only concerned with locating an adjacent local minimum and do not need to find an exact index-1 equilibrium point, which is computationally intensive.

Indirect strategy can be used to speed up the search of neighboring local minimum: Choose an initial straight ray $\mathbf{r}(\lambda)$, emanated from $\mathbf{r}(0) = \mathbf{x}_s$ and follow the ray $\mathbf{r}(\lambda)$, with increasing λ, (or employ a mixed cubic and quadratic interpolation line search algorithm) until it arrives at its first local minimum (or it hits the boundary of search space) of $f(\mathbf{r}(\cdot))$. (The objective function f would increase, pass a local

maximum, and reach a local minimum along the ray.) Starting from the estimated point, locate a neighboring local minimum using a local search algorithm as in Step I.

3.3 Step III: PSO-based Global Search Step

When the random-generated particles fall off at a sup-local minimum through Step II, one important issue is how to escape from a local minimum and find the global minimum. In this multi-dimensional case, we employed the particle swarm optimization method, starting from the obtained points at Step II.

Particle swarm Optimization(PSO) is an efficient stochastic optimization technique developed by Kennedy and Eberhart (1995) [18,19]. It is simply based on the movement and intelligence of swarms. The sketchy concept of PSO is that particles in a space cooperate to find the optimal, and each particle has the velocities and moves toward the global optimum with remembering the past locations. In other words, each particle shares the global best value in one neighborhood space, and adjusts each velocity and location. The Step III follows several steps.

(1) Adjust a population size of particles with the number of obtained particles in Step II and initialize velocities on the multi-dimensions in the space of parameter sets which is consists of (θ).
(2) For each particle, evaluate the **Calibration Problem** which is the objective function of the parameters in each Lévy model.
(3) Calculate the each particle's *personal best (pbest)* through comparing the current value of fitness evaluation with the particle's *pbest*. If the current value is smaller than *pbest* then change the *pbest* into the current value, and also change the *pbest* position into the current position in the Lévy model's parameter space.
(4) Compare the current fitness with the *global best (gbest)* which means the smallest value among the population's previous best. If the current value is smaller than *gbest*then set the *gbest* equal to the current value and switch the *gbest* position into the current position in the Lévy model's parameter space.
(5) Change the velocity of the particle by the following equation 15:

$$\begin{aligned} v_{ij} =& K * [w * v_{ij} + a_1 * rand() * (p_{ij} - x_{ij}) \\ & + a_2 * rand() * (p_{gj} - x_{ij})] \\ & for \quad j = 1, ..., 5 \end{aligned} \tag{15}$$

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \tag{16}$$

where K is constriction function of a_1 and a_2 as equation 16 used the following condition $\varphi = a_1 + a_2$, $\varphi > 4$, and w means 'Inertia weight' and is linearly

decreased from 0.9 to 0.4 during iterations. Inertia weight w was used for improvement of performance by the better adjusting movement and eliminating the maximum velocity constraints. The use of the constriction factor K and φ has guaranteed of sufficient convergence of PSO algorithm. In our simulation, we set $K = 0.709$, $\varphi = 4.1$, and $a_1 = a_2 = 2.1$.

(6) Move to the position of the particle according to equation 6, and loop to step 2) and repeat until a criterion is satisfied.

$$x_{ij} = x_{ij} + v_{ij}, \qquad for \quad j = 1, ..., 5 \tag{17}$$

(7) After calibrating the parameter set we check a performance of the calibration result by calculating the average mean squared errors. First we calculate a mean squared error as follows:

$$\epsilon = \sum_{i=1}^{n} \frac{(C_o^i - C_e^i)^2}{n} \tag{18}$$

where C_o^i is the ith real option value, C_e^i is the ith estimated option value and n is the total number of data set. Then we average mean squared errors conducting several times.

4 Model Robustness Test

In this section, the stability of the multi-basin particle swarm intelligence method is verified through calibrating a parameter set of affine jump-diffusion models with model-generated option prices data. Through this section it is checked whether the proposed method finds well the true parameter set of the model-generated option prices or not. To calibrate a parameter set of affine jump-diffusion models Carr-Madan's method is applied and simple integration methods, such as Trapezoidal rule, Simpson's rule and Boole's rule, is utilized due to the equidistant spacing of grid in the integration domain of the Carr-Madan's method. Gaussian integration method which is an advance numerical integration method can not be applied since the grid points in the Gaussian integration method are pre-defined. Simpson' rule is used to implement the Carr-Madan's method.

Model-generated option prices are generated with the **SVNJ** and **SVSI** from a given true parameter set. The initial stock price is set, $S_0 = 100$, risk free interest rate, $r = 0.03$, and time to maturities, $\tau = 0.1, 0.3, 0.5, 0.7, 1$. The control parameters are also fixed, such as damping parameter $\alpha = 1.6$ and the weight of the difference of the option prices $\omega_i = 1$, for simplicity. The proposed method is applied to this model-generated option prices to calibrate a parameter set of the **SVNJ** and **SVSI**. Table 2 shows some results of Phase I and Phase III of **SVNJ** starting from randomly chosen 30 initial parameter set and Table 3 present the results of **SVSI**.

Table 2
Parameter calibration result of the Phase I & Phase III with **SVNJ**: **Estimated Result1** is the calibration results of Phase I and **Estimated Result2** is the calibration results of Phase III.

Parameter	σ	κ	η	ξ	ρ
True Parameter Set	0.04	3	0.02	0.6	-0.7
Initial Parameter Set	0.9199	0.7090	0.0848	0.8843	0.5650
Estimated Result1	0.04	3	0.02	0.6	-0.7
Estimated Result2	0.04	3	0.02	0.6	-0.7
True Parameter Set	0.04	3	0.02	0.6	-0.7
Initial Parameter Set	0.8160	0.1203	0.5884	0.4094	0.3584
Estimated Result1	0.0263	3.8503	0.0187	0.6962	-0.6940
Estimated Result2	0.04	3	0.02	0.6	-0.7
True Parameter Set	0.04	3	0.02	0.6	-0.7
Initial Parameter Set	0.9628	0.8298	0.2295	0.9946	0.2092
Estimated Result1	0.0651	0.0475	0.2837	4.2531e-7	0.1977
Estimated Result2	0.04	3	0.02	0.6	-0.7
True Parameter Set	0.04	3	0.02	0.6	-0.7
Initial Parameter Set	0.1241	0.3791	0.6405	0.9820	-0.9298
Estimated Result1	0.0190	4.5204	0.0184	0.9903	-8.1886
Estimated Result2	0.04	3	0.02	0.6	-0.7

Table 3
Parameter calibration result of the Phase I with **SVSI**: **Estimated Result1** is the calibration results of Phase I.

Parameter	σ	κ	η	ξ	ρ	κ_λ	η_λ	λ	μ_J	δ_J
True Parameter Set	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1
Initial Parameter Set	0.42	0.69	0.14	0.31	-0.55	0.54	0.76	0.44	0.55	0.72
Estimated Result1	0.14	-0.06	0.5	0.15	-0.99	0.72	0.19	0.001	0.43	1.02
Estimated Result2	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1
True Parameter Set	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1
Initial Parameter Set	0.94	-0.03	0.58	0.07	-0.49	0.10	0.72	0.34	0.06	0.32
Estimated Result1	0.06	-0.04	0.11	0.05	-1.25	0.15	1.25	0.03	0.09	4e-6
Estimated Result2	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1
True Parameter Set	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1
Initial Parameter Set	0.38	-0.85	0.13	0.88	-0.06	0.18	0.75	0.31	-0.78	0.13
Estimated Result1	0.05	-0.16	0.09	1.25	-0.07	0.25	0.47	0.28	-0.92	0.25
Estimated Result2	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1
True Parameter Set	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1
Initial Parameter Set	0.5	-0.56	0.68	0.14	0.22	0.55	0.29	0.89	-0.66	0.72
Estimated Result1	0.24	-0.59	1.14	0.15	0.3	0.24	0.43	0.08	-0.93	0.72
Estimated Result2	0.04	0.06	0.5	0.5	-0.7	-0.2	1	0.4	-0.06	0.1

Through Table 2 and 3 it can be checked there are lots of local minima after Phase I. Almost every paper finishes the model calibration with Phase I and uses the calibrated parameter set, which is a local minimum, to evaluate the model's performance without a serious consideration. It can be a serious problem since there are

lots of local minima. It is checked through Fig 1.

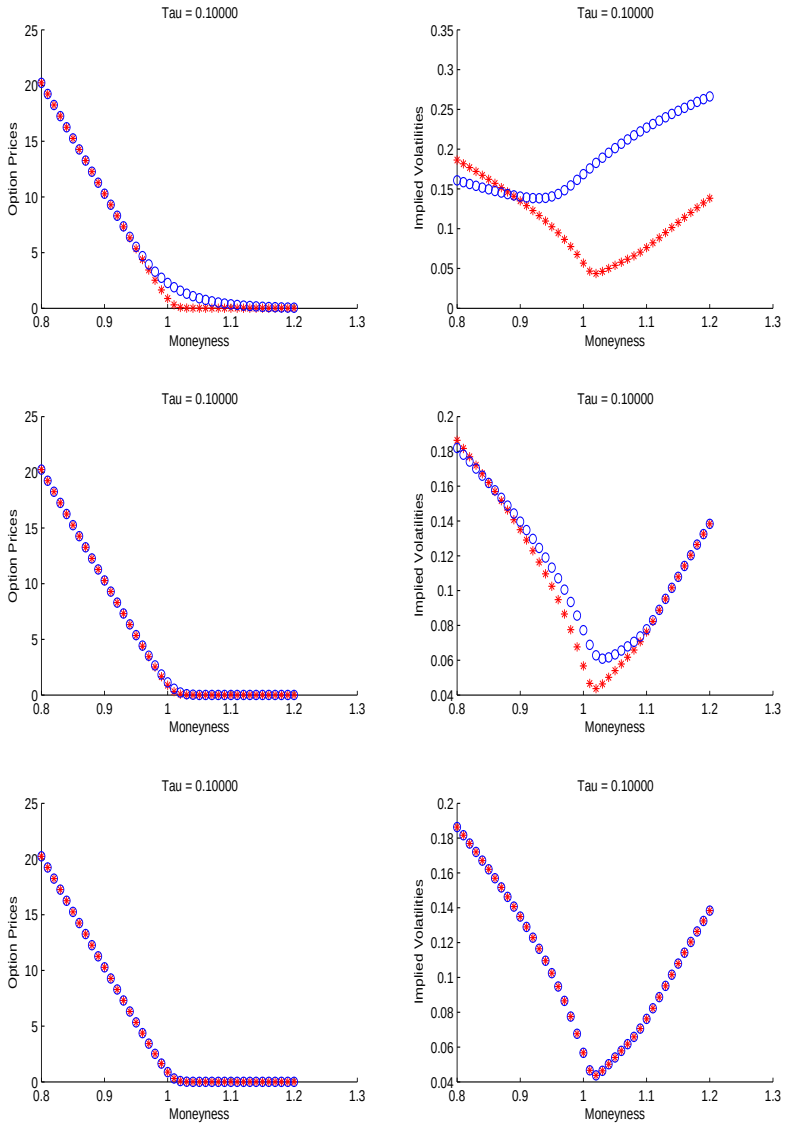

Fig. 1. The calibration results for the option prices (left panels) and implied volatilities (right panels) of the **SVNJ**: The panels in the 1st and the 2nd rows represent the calibration results of phase I (i.e. one of the 1st order local minima) and phase I & II (i.e. one of the sup-local minima) of the proposed method. The panels in the 3rd row represents the calibration results (i.e. a global minimum) of the whole three phases of the proposed method. Red star stands for the model-generated option prices and implied volatilities whereas the blue circle stands for the option prices and implied volatilities with calibration result.

Fig. 1 presents the calibration results of the proposed method with the option prices and implied volatilities of the **SVNJ**. As shown in the figure, the calibrated sup-

local minima through phase I&II often match closely to the option prices (the left panel in the 2nd row in Fig. 1) whereas the (1st) local minima obtained in phase I can not capture the option prices well (the left panel in the 1st row in Fig. 1). However, the sup-local minima still mismatch with the implied volatility despite of its seemingly perfect option price match (the right panel in the 2nd row in Fig. 1). The parameter set calibrated after the whole phases of the proposed method shows the perfect match to both option prices and implied volatilities.

To fix local minimum problem and find a best parameter set the optimization method which consists of three phases is proposed. After phase III every first order local minima converges to the true parameter set with the proposed method. Through model-generated option prices it is shown that the proposed method works well and finds the true parameter set well, although the affine jump-diffusion model has many parameters.

5 Empirical Results

With the stability of the multi-basin particle swarm intelligence method, most well-known S&P 500 index European option prices which are traded in the Chicago Board Options Exchange for the empirical analysis are chosen. S& P 500 index option prices and other informations, which are strike prices, trading volumes, option expiration dates, stock prices and implied volatilities, are collected from Thomson Datastream. We use the option prices of 2007. Unlike model-generated option prices, real market option prices reflect noise, thereby complicating model calibration. Hence, we apply the proposed global calibration method to the European S&P 500 index option prices to determine whether the calibration result accurately explains the observed properties of option data. To conduct experiments, we first eliminate outlier data through the following steps.

(1) Exclude options where the expiration date is less than 6 days, since the implied volatility changes rapidly with a small fluctuation of the option prices and expiration date less than 6 days can cause biases due to liquidity.
(2) Choose the expiration dates which have more than 10 strike prices with non-zero contracts traded.
(3) Eliminate the options with their prices less than their intrinsic values, since this is an obvious arbitrage opportunity.

we conduct numerical analysis with the option prices after preprocessing. The log strikes spaced equidistantly in the Carr-Madan's method can cause a problem since the real market option prices the strike prices are not spaced equidistantly. Hence applying the Carr-Madan's method directly to real market option prices is impossible. To solve it we interpolate option prices through the log strikes. Since the shapes of the option prices are different according to the time to maturities and it's hard

to define a certain function, we use the cubic spline interpolation method. First the parameter set of twelve affine jump-diffusion models is calibrated using the S&P 500 index option prices of May 18th, 2007 and then mean squared errors(MSE) is calculated to show the performance of the calibration result. The mean squared error is computed as follows:

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N}(C_o^i - C_e^i)^2 \tag{19}$$

where C_o^i is the ith real option value, C_e^i is the ith estimated option value and N is the total number of data set. Then mean squared errors are averaged through repeating several times. Following table 4 and 5 show the calibration result of **Phase I** and **Phase III** with the twelve affine jump-diffusion models.

Table 4
Parameter calibration result of **Phase I** with S&P 500 index option prices on May 18th, 2007

Parameter	**CVNJ**	**CVCI**	**CVDI**	**CVSI**	**DVNJ**	**DVCI**
υ_0	0.1208	0.1208	0.1240	0.1208	0.1119	0.1058
λ		1.8097e-15	2.738e-16	1.1387e-12		1.0121e-13
κ					-0.7322	4.6718
η					1.6105e-14	0.0181
ξ						
ρ						
κ_λ			-0.0124	0.7734		
η_λ			2.5953	9.2105e-17		
ξ_λ				0.8685		
μ_J		0.0162	0.4158	-0.1359		-2.4473
δ_J		0.4436	0.2271	0.0589		3.2872
MSE	14.2281	14.2281	13.8581	14.2281	13.0222	12.6110

Parameter	**DVDI**	**DVSI**	**SVNJ**	**SVCI**	**SVDI**	**SVSI**
υ_0	0.0763	0.0902	0.0936	0.1375	0.1065	0.1253
λ	0.7493	0.4811		1.8702e-18	0.0410	0.1114
κ	-0.3741	0.1398	0.0238	0.1600	-0.0143	-0.3555
η	0.0073	3.0883e-14	0.4242	0.2424	1.3134	1.0410e-04
ξ			0.3024	1.2080	0.0192	1.8820
ρ			0.7363	-0.2062	-0.0304	0.0840
κ_λ	-1.0652	0.0028			1.7372	0.0164
η_λ	0.0964	0.2367			1.0954	0.0195
ξ_λ		0.0969				1.2350
μ_J	-0.1006	-0.1655		3.0778	-0.1776	-0.5269
δ_J	0.0570	7.3793e-04		0.7938	0.0578	0.3108
MSE	2.1435	7.8961	70.2466	15.7114	2.7642	25.6224

Through Table 4 and 5 it is shown that the calibration results obtained by phase I have a poor performance comparing with the calibration results after conduct-

Table 5
Parameter calibration result of **Phase III** with S&P 500 index option prices on May 18th, 2007

Parameter	**CVNJ**	**CVCI**	**CVDI**	**CVSI**	**DVNJ**	**DVCI**
υ_0	0.1208	0.0794	0.0788	0.0916	0.0958	0.0802
λ		0.6367	0.5085	0.4654		0.6371
κ					10.9951	1.5010
η					0.0169	0.0057
ξ						
ρ						
κ_λ			18.8525	0.1039		
η_λ			0.7179	0.0072		
ξ_λ				0.0960		
μ_J		-0.1255	-0.1212	-0.1734		-0.1261
δ_J		0.0644	0.0604	5.04e-9		0.0644
MSE	14.2281	1.8390	1.7196	1.3718	12.6083	1.8318

Parameter	**DVDI**	**DVSI**	**SVNJ**	**SVCI**	**SVDI**	**SVSI**
υ_0	0.0843	0.0915	0.1067	0.0947	0.0956	0.0952
λ	0.9250	0.4681		0.6564	0.4209	0.1127
κ	2.8898	0.0238	6.0755	2.8649	1.5100	7.3500
η	5.361e-04	5.807e-12	0.0228	0.0177	0.0341	0.0233
ξ			0.6257	0.3009	0.3388	0.6625
ρ			-0.7686	-0.8913	-0.7413	-0.8041
κ_λ	2.5906	0.1018			-5.1275	-0.0381
η_λ	2.2474	0.0074			0.1850	10.8942
ξ_λ		0.0984				8.1461
μ_J	-0.0762	-0.1725		-0.0553	-0.1515	-0.050
δ_J	0.0611	1.803e-7		0.1049	0.0298	0.2867
MSE	1.3764	1.3717	1.0130	0.9608	0.8898	0.9285

ing all phases. Although the calibration results obtained by phase I do not have meaningful information, the final parameter set of the proposed method do have. Since option prices do not sensitive to κ, it can be calibrated fixing the κ within a certain range of proposed method. The initial variances of asset return(υ) of twelve affine jump-diffusion models are similar in a range from 8% to 12%. In general, the correlation(ρ) between the asset and the variance of asset return is negative and it is shown that the correlations between them are negative from -0.76 to -0.89. In every models, mean jump size(μ_J) is a negative value, which means the frequency of negative jump is more often than positive jump or the negative jump size is bigger than the positive jump size. Since the standard deviation(δ_J) of the jump size is very small, the distribution gathers around the mean value. As the number of parameters increases, the quality of calibration is getting better. **CVNJ**, which is same with the Black-Scholes model, has the worst performance among twelve affine jump-diffusion models. The best in-sample fit is the **SVDI** which has stochastic volatility and deterministic intensity.

The option prices and implied volatilities are calculated with the calibration result of Table 5 and are compared with real option prices and implied volatilities to show the the performance of the Multi-basin Particle Swarm Intelligence Method. Following Fig 2 and 3 show the simulation result of **SVCI** and **SVSI**.

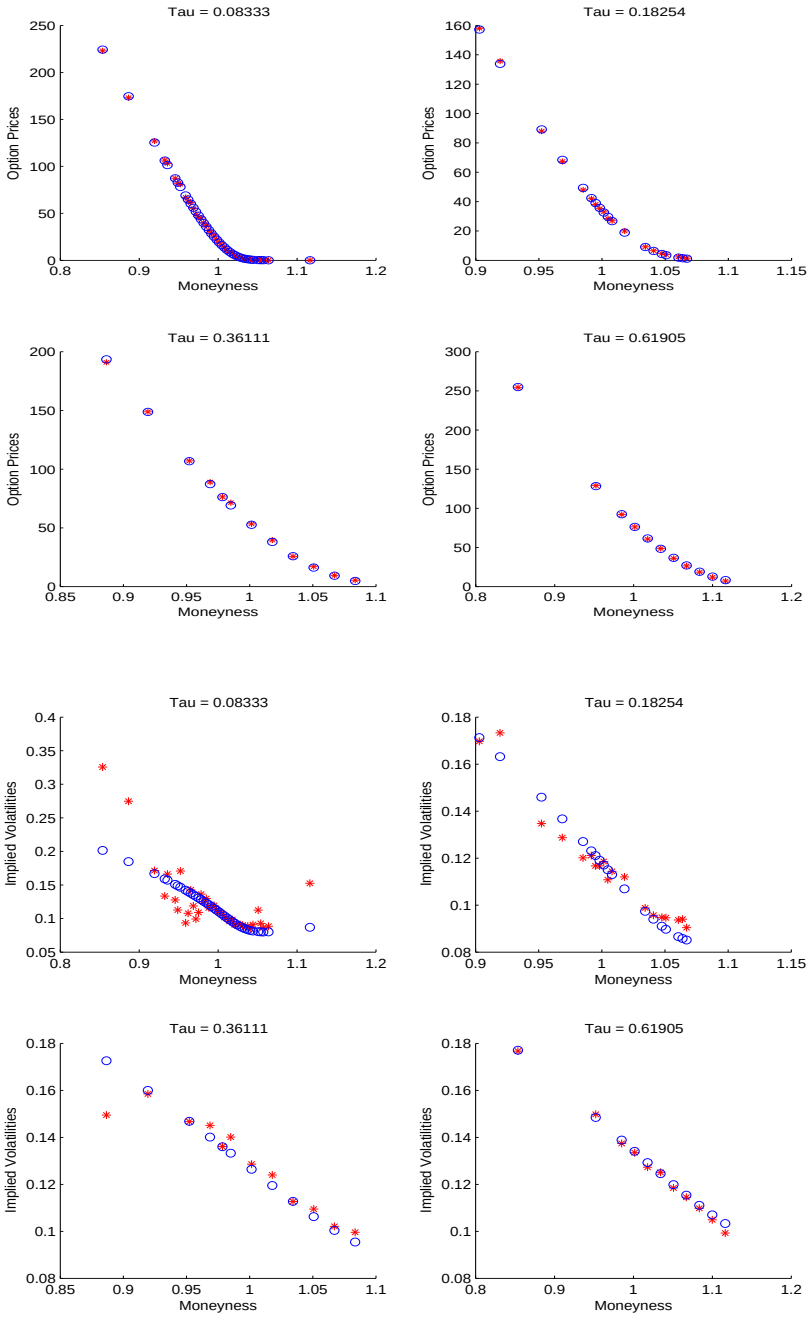

Fig. 2. The option prices and implied volatilities of the **SVCI** with the calibration result: red star stands for the real S& P 500 index option prices and implied volatilities and blue circle is the option prices and implied volatilities of the **SVCI**.

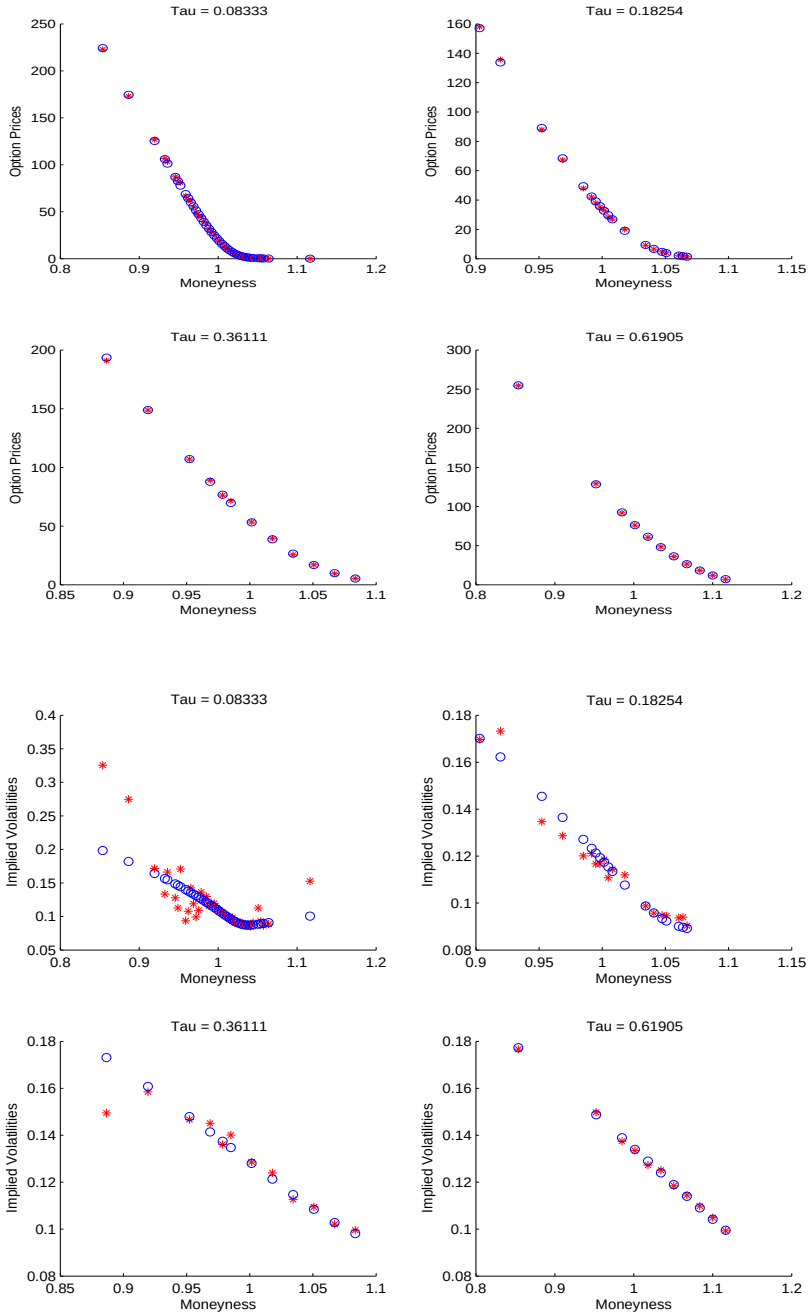

Fig. 3. The option prices and implied volatilities of the **SVSI** with the calibration result: red star stands for the real S& P 500 index option prices and implied volatilities and blue circle is the option prices and implied volatilities of the **SVSI**.

Through Fig 2 and 3 it is shown that the calibration result represent the real option prices well and **SVSI** captures the implied volatility smile well at a short maturity comparing with the **SVCI**.

6 Conclusion

In this paper, the global optimization method which consists of 3 phases is proposed to calibrate parameter set of twelve different affine jump-diffusion models which have three dynamics, the asset price, the variance of asset return and the jump intensity, and compare the performance with the model-generated option prices and real S&P 500 index option prices. It is shown that the multi-basin particle swarm intelligence method works significantly well to calibrate affine jump-diffusion models through model-generated option prices. With the calibration result of S&P 500 index option prices on May 18th, 2007 it is found that the parameters are calibrated reasonably. Through the empirical results several important features are found: First traders should not pay attention to use 1st order local minimum, since it causes a wrong option price. However it is found that local minimum problem can be solved with the proposed method. Second complex model can not become the best model. Third, to describe real market option prices well it is shown that the jumps should include. Examining the calibration performance of affine jump-diffusion models with option prices of different markets are left for further studies.

References

[1] Bakshi, G., Cao, C., and Chen Z. (1997). Empirical performance of alternative option pricing models. Journal of Finance, 52, 2003-2049.

[2] Balland, P. (2002). Deterministic implied volatility surfaces. Quantitative Finance, 2, 31-44.

[3] Barndorff-Nielsen, O. (1997). Processes of normal inverse Gaussian type. Finance and Stochastics, 2, 1432-1122.

[4] Bates, D. (1996). Jumps and stochastic volatility. Review of Financial Studies, 9, 69-107.

[5] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. New York: Springer-Verlag.

[6] Carr, P. and Madan, D. (1999). Option valuation using the fast Fourier transform. Journal of Computational Finance, 2, 61-73.

[7] Carr, P., Gaman, H., Madan, D. and Yor, M. (2003). Stochastic Volatility for Lévy Processes. Mathematical Finance, 13, 345-385.

[8] Cont, R. and Fonseca, J.D. (2002). Dynamics of implied volatility surface. Quantitative Finance, 2, 45–60.

[9] Cont, R. and Tankov, P. (2004). Financial Modelling with Jump Processes. London, UK: CRC Press.

[10] Jones Christopher, S. (2003). The dynamics of stochastic volatility: evidence from underlying and option markets. Journal of Econometrics, 116, 181-224.

[11] Dumas, B., Fleming, J., and Whaley, R.E. (1998). Implied volatility functions: empirical tests. Journal of Finance, 8, 2059–2106.

[12] Eberlin, E. (2001). Applications of generalized hyperbolic Lévy motion to Finance. In O. E. Barndorff-Nielsen, T. Mikosch, and S. Resnick, editors, Lévy Processes - Theory and Applications. Boston: Birkhauser.

[13] Fang, H. (2001). Option Pricing Implications of a Stochastic Jump Rate. Working Paper.

[14] Gencay, R., and Qi, M. (2001). Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. IEEE Transactions on Neural Networks, 12, 726-734.

[15] Gutiérrez, Ó. (2008). Option valuation, time-changed processes and the fast Fourier transform. Qunatitative Finance, 8, 103-108.

[16] Heston, S. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. The Review of Financial Studies, 6(2), 327–343.

[17] Hutchinson, J. M., Lo, A.W., and Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. Journal of Finance, 59, 851-889.

[18] Kennedy, J., and Eberhart, R., *Particle Swarm Optimization*, Proceedings IEEE International Conference on Neural Networks, Perth, Australia, Nov. 1995, pp. 1942–1948.

[19] Kennedy, J., and Eberhart, R., *A Discrete Binary Version of the Particle Swarm Algorithm*, 1997 IEEE Conference on Systems, Man, and Cybernetics, Orlando, FL, Oct. 1997, pp. 4104–4109.

[20] Lee, J., *Pseudobasin of attraction for combinatorial dynamical systems: Theory and its application to combinatorial optimization*, IEEE Trans. on Circuits and Systems II-Express Briefs, vol. 52, pp. 189–193, 2005.

[21] Lewis, A. (2001). A simple option formula for general jump-diffusion and other exponential Lévy processes. Working Paper.

[22] Madan, D., Carr, P., and Chang, E. (1998). The Variance Gamma Process and Option Pricing. European Finance Review, 2, 79-105.

[23] Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. Journal of Financial Economics, 3, 125-144.

[24] Minenna, M., and Verzella, P. (2008). A revisited and stable Fourier transform method for affine jump diffusion models. Journal of Banking and Finance, 32, 2064-2075.

[25] Sato, K. (1999). Lévy Processes and Infinitely Divisible Distributions. Cambridge, UK: Cambridge University Press.

[26] Schoutens, W., Simons, E., and Tistaert, J. (2004). A perfect calibration! Now what. Wilmott magazine, 4, 66-78.

[27] Kou, S. G. (2002). A jump diffusion model for option pricing. Management Science, 48, 1086-1101.