University "Alexandru Ioan Cuza" of Iaşi
Faculty of Computer Science

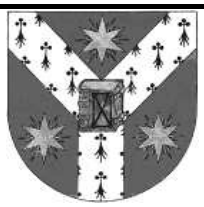
TECHNICAL REPORT

Importance Resampling with MRAS algorithm for Bermudan option pricing

Emanuel F. OLARIU

TR 12-01, March 2012

ISSN 1224-9327



Universitatea "Alexandru Ioan Cuza" Iaşi
Facultatea de Informatică
Str. Berthelot 16, 6600-Iaşi, Romania
Tel. +40-32-201090, email: bibl@infoiasi.ro

1 Introduction.

We consider here the American option with a finite number of early exercise opportunities, also called Bermudan options (somewhere between, but closer to, American and European options). Let us suppose that: K is the strike price, r is the interest rate, T is the maturity, n the number of early exercise opportunities (without maturity), the dates being

$$0 < t_1 < \ldots < t_{n+1} = T,$$

S_i^* are the early exercise thresholds, and S_i the stock prices.

For such a derivative the value of a *call option* is:

$$\max_{\mathbf{S}} \mathbb{E}\left[\mathcal{V}_{c,\mathbf{S}^*}\right], \tag{1}$$

where $\mathbf{S}^* = (S_1^*,\ S_2^*,\ \ldots,\ S_n^*)$, and ([Zhang06])

$$\begin{aligned}\mathcal{V}_{c,\mathbf{S}^*}(\mathcal{S}) = \sum_{i=1}^{n}(S_i - K)^+ \cdot e^{-rt_i} \cdot \mathbb{1}_{\left[S_1<S_1^*,\ \ldots,\ S_{i-1}<S_{i-1}^*,\ S_i>S_i^*\right]} + \\ +(S_{n+1} - K)^+ \cdot e^{-rT} \cdot \mathbb{1}_{\left[S_1<S_1^*,\ \ldots,\ S_n<S_n^*\right]}\end{aligned} \tag{2}$$

where $\mathcal{S} = (S_1, S_2, \ldots, S_n) \in \mathbb{R}^n$ is the vector of stock prices.

In a similar way the payoff for a *put option* is:

$$\begin{aligned}\mathcal{V}_{p,\mathbf{S}^*}(\mathcal{S}) = \sum_{i=1}^{n}(K - S_i)^+ \cdot e^{-rt_i} \cdot \mathbb{1}_{\left[S_1>S_1^*,\ \ldots,\ S_{i-1}>S_{i-1}^*,\ S_i<S_i^*\right]} + \\ +(K - S_{n+1})^+ \cdot e^{-rT} \cdot \mathbb{1}_{\left[S_1>S_1^*,\ \ldots,\ S_n>S_n^*\right]}\end{aligned} \tag{3}$$

In both equations (2) and (3) the second term on the right side is the payoff at the maturity time. For a *call option* the threshold prices form a decreasing sequence of non-negative numbers with the final threshold being the strike price:

$$S_1^* \geqslant S_2^* \geqslant \ldots \geqslant S_n^* \geqslant S_{n+1}^* = K \tag{4}$$

For a *put option* the threshold prices form an increasing sequence of non-negative numbers with the final threshold being the strike price:

$$S_1^* \leqslant S_2^* \leqslant \ldots \leqslant S_n^* \leqslant S_{n+1}^* = K \tag{5}$$

$\mathbf{S}^* = (S_1^*,\ S_2^*,\ \ldots,\ S_n^*)$ follow a multivariate gaussian distribution $N(\mu, \Sigma)$. In the case of the *call option* the restrictions on these prices are:

$$S_i^* \geqslant 0,\ i = \overline{1,n};\ \text{and}\ S_1^* \geqslant S_2^* \geqslant \ldots, \geqslant S_n^* \geqslant S_{n+1}^* = K.$$

defining a particular polytope

$$\mathbb{X}_c = \{\mathbf{x} \in \mathbb{R}_+^n\ :\ x_1 \geqslant x_2 \geqslant \ldots \geqslant x_n \geqslant K\}.$$

Similarly, the thresholds for a *put option* lie in the following polytope

$$\mathbb{X}_p = \{\mathbf{x} \in \mathbb{R}_+^n\ :\ x_1 \leqslant x_2 \leqslant \ldots \leqslant x_n \leqslant K\}.$$

Therefore, $\mathbf{S}$ the vector of threshold prices must be generated from a multivariate gaussian distribution truncated on one of the above polytopes: the pdf for such a distribution (with certain mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$) is

$$\varphi_{\mathbb{X}}\left(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\frac{1}{\left(\sqrt{2\pi}\right)^{n}\cdot\sqrt{|\boldsymbol{\Sigma}|}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{T}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]\cdot\mathbb{1}_{[\mathbb{X}]}(\mathbf{x}). \tag{6}$$

As the threshold prices are the optimized arguments in the algorithm, a fast and quality sampling procedure is crucial for the accuracy of our results. Therefore we avoid the accept-reject method for this truncated distribution and we used a Gibbs sampler combined with a Metropolis Chain. After all these prices are determined, the values of the desired option can be calculated by estimating the expectation of the value functions (3) or (2) (this is done by a forward simulation - knowing the equation which models the price dynamics).

2 Sampling importance resampling with MRAS

In the remaining sections we analyze only the *put options*; let us denote, following the notations from ([Hu07]), $\mathcal{H}(\mathbf{S}^{*})=\mathbb{E}\left[\mathcal{V}_{p,\mathbf{S}^{*}}\right]$; the problem at hand is to find

$$\widehat{\mathbf{S}}^{*}=\arg\max_{\mathbf{S}^{*}}\mathcal{H}(\mathbf{S}^{*})$$

The Model Reference Adaptive Search (MRAS) is an iterative procedure which, in our case, generates candidate solutions following the normal distribution $N(\boldsymbol{\mu},\boldsymbol{\Sigma})$, and updates the parameters of the distribution using:

$$\boldsymbol{\mu}_{t+1}=\frac{\mathbb{E}_{t}\left[s\left[\mathcal{H}(\mathbf{S}^{*})\right]^{t}/\varphi_{\mathbb{X}_{p}}(\mathbf{S}^{*};\boldsymbol{\mu}_{t},\boldsymbol{\Sigma}_{t})\cdot\mathbb{1}_{\left[\mathcal{H}(\mathbf{S}^{*})\geqslant\overline{\gamma}_{t+1}\right]}\cdot\mathbf{S}^{*}\right]}{\mathbb{E}_{t}\left[s\left[\mathcal{H}(\mathbf{S}^{*})\right]^{t}/\varphi_{\mathbb{X}_{p}}(\mathbf{S}^{*};\boldsymbol{\mu}_{t},\boldsymbol{\Sigma}_{t})\cdot\mathbb{1}_{\left[\mathcal{H}(\mathbf{S}^{*})\geqslant\overline{\gamma}_{t+1}\right]}\right]} \tag{7}$$

$$\boldsymbol{\Sigma}_{t+1}=\frac{\mathbb{E}_{t}\left[s\left[\mathcal{H}(\mathbf{S}^{*})\right]^{t}/\varphi_{\mathbb{X}_{p}}(\mathbf{S}^{*};\boldsymbol{\mu}_{t},\boldsymbol{\Sigma}_{t})\cdot\mathbb{1}_{\left[\mathcal{H}(\mathbf{S}^{*})\geqslant\overline{\gamma}_{t+1}\right]}\cdot\left(\mathbf{S}^{*}-\boldsymbol{\mu}_{t+1}\right)\left(\mathbf{S}^{*}-\boldsymbol{\mu}_{t+1}\right)^{T}\right]}{\mathbb{E}_{t}\left[s\left[\mathcal{H}(\mathbf{S}^{*})\right]^{t}/\varphi_{\mathbb{X}_{p}}(\mathbf{S}^{*};\boldsymbol{\mu}_{t},\boldsymbol{\Sigma}_{t})\cdot\mathbb{1}_{\left[\mathcal{H}(\mathbf{S}^{*})\geqslant\overline{\gamma}_{t+1}\right]}\right]} \tag{8}$$

where s is a continuous strictly increasing pozitive function, and the expectation $\mathbb{E}_{t}[\cdot]$ is taken under the truncated distribution $N_{\mathbb{X}_{p}}(\boldsymbol{\mu}_{t},\boldsymbol{\Sigma}_{t})$.

The above expectations are estimated by Monte Carlo simulation; we sample a sequence of i.i.d. gaussian vectors $\mathbb{S}=\left(\mathbf{S}^{(t,j)}\right)_{1\leqslant j\leqslant N_{t}}\subseteq\mathbb{X}_{p}$, and, for the sake of simplicity, define the following weights:

$$w_{j}\left(\mathbb{S};\boldsymbol{\mu},\boldsymbol{\Sigma},\gamma\right)=\frac{s\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\right]^{t}/\varphi_{\mathbb{X}_{p}}(\mathbf{S}^{(t,j)};\boldsymbol{\mu},\boldsymbol{\Sigma})\cdot\mathbb{1}_{\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\geqslant\gamma\right]}}{\sum_{j=1}^{N_{t}}s\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\right]^{t}/\varphi_{\mathbb{X}_{p}}(\mathbf{S}^{(t,j)};\boldsymbol{\mu},\boldsymbol{\Sigma})\cdot\mathbb{1}_{\left[\mathcal{H}\left(\mathbf{S}^{(t,j)}\right)\geqslant\gamma\right]}},$$

where $\boldsymbol{\mu}\in\mathbb{R}^{n}$, $\boldsymbol{\Sigma}\in\mathbb{R}^{n\times n}$, $\gamma\in\mathbb{R}$. The parameters for the truncated multivariate gaussian distribution are updated using the following formulas:

$$\boldsymbol{\mu}_{t+1}=\sum_{j=1}^{N_{t}}w_{j}\left(\mathbb{S};\boldsymbol{\mu}_{t},\boldsymbol{\Sigma}_{t},\overline{\gamma}_{t+1}\right)\cdot\mathbf{S}^{(t,j)} \tag{9}$$

$$\boldsymbol{\Sigma}_{t+1} = \sum_{j=1}^{N_t} w_j \left(\mathbb{S}; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \overline{\gamma}_{t+1}\right) \cdot \left(\mathbf{S}^{(t,j)} - \boldsymbol{\mu}_{t+1}\right) \cdot \left(\mathbf{S}^{(t,j)} - \boldsymbol{\mu}_{t+1}\right)^T \tag{10}$$

A smoothing coefficient is defined ($\alpha \in (0,1)$, $p \in \mathbb{N}^*$):

$$\alpha_t = \alpha - \alpha \left(1 - \frac{1}{t}\right)^p.$$

And the parameters are accordingly modified:

$$\widehat{\boldsymbol{\mu}}_t = \alpha_t \boldsymbol{\mu}_t + (1-\alpha_t)\widehat{\boldsymbol{\mu}}_{t-1}, \text{ and } \widehat{\boldsymbol{\Sigma}}_t = \alpha_t \boldsymbol{\Sigma}_t + (1-\alpha_t)\widehat{\boldsymbol{\Sigma}}_{t-1} \tag{11}$$

We describe a modified version of MRAS algorithm by including a sampling importance resampling phase; before calculate the sample $(1-\rho)$-quantile, we replace the actual samples using an importance sampling technique named resampling. First we can resample based on the natural weights: if $\mathbf{S} = \left(\mathbf{S}^j\right)_{1 \leqslant j \leqslant N}$ is the currently generated samples we have to determine the following weights:

$$u_j = \frac{\mathcal{H}(\mathbf{S}^j)}{\sum_{i=1}^{N} \mathcal{H}(\mathbf{S}^i)} \tag{12}$$

and generate a new sample using a multinomial distribution with these weights. That is, we give more importance to those samples which have greater payoffs.

In the following let us remember a few theoretical considerations concerning MRAS (exact version) global convergence. The parameters to update in each iteration are $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = \theta_t$; this merging parameter is chosen in such a way that $\varphi_{\theta_{t+1}}$, the next pdf, is "closer" to the corresponding distribution from sequence of so-called reference distribution $(g_k)_{k \geqslant 1}$, where:

$$g_{t+1}(x) = \frac{s[\mathcal{H}(x)] \cdot \mathbb{1}_{\left[\mathcal{H}(x) \geqslant \overline{\gamma}_{t+1}\right]} \cdot g_t(x)}{\mathbb{E}_{g_t}\left[s[\mathcal{H}(X)] \cdot \mathbb{1}_{\left[\mathcal{H}(X) \geqslant \overline{\gamma}_{t+1}\right]}\right]}, \forall\, t \geqslant 2 \text{ and} \tag{13}$$

$$g_1(x) = \frac{\mathbb{1}_{[\mathcal{H}(x) \geqslant \overline{\gamma}_1]}}{\mathbb{E}_{\theta_0}\left[\mathbb{1}_{[\mathcal{H}(X) \geqslant \overline{\gamma}_1]}\right]}. \tag{14}$$

Hence, for step t, a more natural resampling would be using the following weights:

$$v_j = \frac{\left[s(\mathcal{H}(\mathbf{S}^j))\right]^{t-1} \cdot \mathbb{1}_{[\mathcal{H}(\mathbf{S}^j) \geqslant \overline{\gamma}_t]}}{\sum_{i=1}^{N} \left[s(\mathcal{H}(\mathbf{S}^i))\right]^{t-1} \cdot \mathbb{1}_{[\mathcal{H}(\mathbf{S}^i) \geqslant \overline{\gamma}_t]}} \tag{15}$$

The algorithm follows:

- initialization: quantile level ρ_0, sample size N_0, $\boldsymbol{\mu}_0$, $\boldsymbol{\Sigma}$, sample size level λ, a limit parameter $\varepsilon > 0$, a weight $\beta \in (0,1)$, a continuous increasing pozitive function $s(\cdot)$, and $t = 0$;
- the general iteration of the algorithm:
  - generate a i.i.d. sample, $\mathbb{S}^t = \mathbf{S}^{(t,1)}, \mathbf{S}^{(t,2)}, \ldots, \mathbf{S}^{(t,N_t)}$ from density $\widehat{\varphi}_t = \beta\varphi_t + (1-\beta)\varphi_0$, where φ_t is the density of $N_{\mathcal{S}}\left(\widehat{\boldsymbol{\mu}}_t, \widehat{\boldsymbol{\Sigma}}_t\right)$;

then, resample using (12) or (15) and get another sample:

$$\mathbf{S}^{(t,(1))},\ \mathbf{S}^{(t,(2))},\ \ldots,\ \mathbf{S}^{(t,(N_t))}$$

- calculate the $(1-\rho_t)$ quantile, $\gamma_{t+1}(\rho_t, N_t)$, of the sample

$$\mathcal{H}\left(\mathbf{S}^{(t,(1))}\right),\ \mathcal{H}\left(\mathbf{S}^{(t,(2))}\right),\ \ldots,\ \mathcal{H}\left(\mathbf{S}^{(t,(N_t))}\right)$$

- if $t=0$ or $\gamma_{t+1}(\rho_t, N_t) \geqslant \gamma_t + \varepsilon$

$$\overline{\gamma}_{t+1} \leftarrow \gamma_{t+1}(\rho_t, N_t),\ \rho_{t+1} \leftarrow \rho_t,\ N_{t+1} \leftarrow N_t$$

else, if exists

$$\overline{\rho} = \max\left\{\rho' \ :\ \gamma_{t+1}(\rho', N_t) \geqslant \overline{\gamma}_t + \varepsilon,\ 0 \leqslant \rho' \leqslant \rho_t\right\},$$

then

$$\overline{\gamma}_{t+1} \leftarrow \gamma_{t+1}(\overline{\rho}, N_t),\ \rho_{t+1} \leftarrow \overline{\rho},\ N_{t+1} \leftarrow N_t$$

else

$$\overline{\gamma}_{t+1} \leftarrow \overline{\gamma}_t,\ \rho_{t+1} \leftarrow \rho_t,\ N_{t+1} \leftarrow \lambda N_t$$

- update and smooth $\boldsymbol{\mu}_{t+1}$ and $\boldsymbol{\Sigma}_{t+1}$, using (9), (10) and, respectively, (11)
- $t++$;

In our implementation the stopping criteria includes, besides the maximum number of steps ($N_t \geqslant N_{max}$), a minimum number of valid samples which will be used in the updating phase.

3 Models used

3.1 Geometric Brownian motion model

We say that the underlying stock price, $S(t)$, follows a geometric Brownian motion if

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \tag{16}$$

where $(W(t))_{t\geqslant 0}$ is a standard Wiener process (or Brownian motion), i.e. $W_0 \equiv 0$, $t \mapsto W(t)$ is continuous almost surely, and its increments are mutually independent and stationary ($W(t+s) - W(s) \sim N(0,t),\ \forall\, s > 0$).

For this model $\mu = r - \delta$, where δ is the dividend yield, and σ is the volatility – all supposed constants. Under these conditions, using Ito's lemma, equation (16) has the following solution:

$$S(t) = S(0) \cdot \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right] \tag{17}$$

from where the discrete counterpart used for simulation is

$$S_{\tau+\Delta\tau} = S_\tau \cdot \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta\tau + \sigma\sqrt{\Delta\tau}\cdot Z\right], \tag{18}$$

where Z is a normal standard distributed random variable.

3.2 Merton normal jump diffusion model

Practical evidences are that the geometric Brownian motion does not always accurately simulate the stock price behaviour. Therefore other models which allow jumps have been introduced – namely jump-diffusion models ([Cont04], [Tan09]). Merton ([Mert76]) proposed the following dynamic to model the underlying stock price:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) + S(t)dX(t), \tag{19}$$

$$X(t) = \sum_{i=1}^{N(t)} Y_i,$$

where $W(t)$ is a standard Wiener process, $X(t)$ is a compound Poisson process: $N(t)$ – the number of allowed jumps – is a Poisson process with parameter λ, and $Y_1, Y_2, \ldots$ is a sequence of independent and identical log-normal distributed, $LN(-\gamma^2/2, \gamma^2)$, random variables; here λ is the frequency and γ is the volatility of the jumps.

The discrete form of equation (19) is:

$$\frac{S_{\tau+\Delta\tau}}{S_\tau} = \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\cdot\Delta\tau + \sigma\sqrt{\Delta\tau}\cdot Z_0 + \sum_{i=1}^{N(\Delta\tau)}\left(\gamma Z_i - \frac{\gamma^2}{2}\right)\right], \tag{20}$$

where $Z_0, Z_1, \ldots$ are i.i.d normal standard random variables, and $N(\Delta\tau)$ is distributed Poisson with parameter $\lambda\Delta\tau$.

3.3 A double exponentially jump diffusion model

Kou proposed in [Kou02] another jump diffusion model for the asset price, which basically differs from the above model by the distribution of jump sizes – double exponentially:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) + S(t)dV(t), \tag{21}$$

$$V(t) = \sum_{i=1}^{N(t)} (V_i - 1),$$

where $W(t)$ is a standard Wiener process, $V(t)$ is a compound Poisson process: $N(t)$ is a Poisson process with rate λ, and $V_1, V_2, \ldots$ are independent and identical log-asymmetric double exponential distributed random variables, i. e. $Y = \log(V_i)$ has density:

$$f(x) = \begin{cases} p\cdot\alpha_1\cdot e^{-\alpha_1 x}, & x \geqslant 0 \\ (1-p)\cdot\alpha_2\cdot e^{\alpha_2 x}, & x < 0 \end{cases}, \tag{22}$$

p and $(1-p)$ being the probabilities of up and down jumps.

The solution to the equation (21) is

$$S(t) = S(0)\cdot\exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right]\cdot\prod_{i=1}^{N(t)} V_i. \tag{23}$$

The discrete form of equation (23) is:

$$S_{\tau+\Delta\tau} = S_\tau\cdot\exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\cdot\Delta\tau + \sigma\sqrt{\Delta\tau}\cdot Z_0 + \sum_{i=1}^{N(\Delta\tau)} Y_i\right], \tag{24}$$

where Z_0 is a normal standard distributed random variable, $Y_1, Y_1, \ldots$ are i.i.d asymmetric double exponentially distributed random variables (with density (22)), and $N(\Delta\tau)$ is distributed Poisson with parameter $\lambda\Delta\tau$.

4 Numerical results and concluding remarks

The numerical results are obtained in the following conditions: initial sample size is $N_0 = 200$, initial quantile level $\rho_0 = 0.5$, smoothing parameter $\alpha = 0.8$, sample size level correction $\alpha = 2$, weight parameter $\beta = 0.3$, $\varepsilon = 0.001$. The increasing pozitive function used is $s(x) = \exp(0.1x)$, initial mean is a vector having all components 10, and initial covariance matrix has diagonal elements 225. Option prices are obtained by using $50,000$ simulations, after the threshold prices are estimated.

Tables 1 to 3 show the results of our simulations; all models are tested for various early exercise dates and different first threshold price values. Every cell in these tables shows the mean obtained on 100 independent simulations, the standard error of the mean, and the average number of iterations performed by the algorithm.

The two sampling importance versions are: the uniform (u-SIR) and the reference distributions sampling importance resampling (rd-SIR). Both perform less steps, while rd-SIR performs at most half steps, than standard MRAS algorithm.

The prices we get are slightly smaller for importance resampling, although remaining very close to those obtained with the standard MRAS procedure. In almost all cases the standard error of the mean is similar for all three algorithms; an exception is for Kou and Merton models on 6 early exercices where standard error almost doubles, therefore remaining under 0.05. This means that importance resampling, especially that using the reference distributions from original MRAS, is a reliable and faster method compared with standard MRAS algorithm.

Number of exercices		S_0^* = 40	S_0^* = 60	S_0^* = 80	S_0^* = 100	
$n = 2$		10.898 0.009/23.58	10.901 0.009/23.02	10.889 0.009/23.30	10.904 0.011/24.04	mras
		10.903 0.011/16.78	10.880 0.010/16.64	10.895 0.010/17.18	10.894 0.008/16.96	u-SIR
		10.893 0.010/13.2	10.941 0.033/12.6	10.914 0.008/12.46	10.905 0.009/12.60	rd-SIR
$n = 4$		11.529 0.016/9.44	11.545 0.019/9.50	11.508 0.019/9.62	11.5488 0.015/10.58	mras
		11.379 0.017/9.28	11.403 0.016/9.32	11.396 0.015/8.76	11.388 0.018/8.76	u-SIR
		11.418 0.020/6.94	11.416 0.021/6.44	11.431 0.025/7.18	11.417 0.022/6.94	rd-SIR
$n = 6$		12.508 0.017/11.16	12.529 0.012/11.46	12.536 0.014/11.02	12.517 0.019/11.22	mras
		12.465 0.019/8.88	12.448 0.018/9.20	12.458 0.019/9.20	12.453 0.026/9.16	u-SIR
		12.471 0.022/6.62	12.455 0.024/6.40	12.450 0.017/6.84	12.434 0.023/6.14	rd-SIR

Table 1: Geometric Brownian model - prices for a 3-years put option with: $r = 0.05$, $\sigma = 0.25$, $\delta = 0$, $S_0 = K = 110$.

Number of exercices		S_0^* = 30	S_0^* = 50	S_0^* = 70	S_0^* = 90	
$n = 2$		10.410 0.009/24.66	10.401 0.008/25.08	10.383 0.008/24.2	10.372 0.010/24.34	mras
		10.402 0.010/16.80	10.404 0.009/18.20	10.375 0.009/16.52	10.379 0.008/17.76	u-SIR
		10.412 0.008/11.92	10.408 0.009/12.70	10.390 0.001/11.66	10.375 0.009/12.06	rd-SIR
$n = 4$		10.479 0.010/21.86	10.474 0.009/22.42	10.480 0.009/23.32	10.493 0.009/22.24	mras
		10.483 0.010/16.14	10.489 0.009/15.54	10.473 0.009/17.04	10.491 0.010/17.34	u-SIR
		10.481 0.009/11.74	10.481 0.010/10.58	10.488 0.007/11.42	10.467 0.009/10.78	rd-SIR
$n = 6$		10.643 0.008/23.44	10.656 0.008/23.18	10.645 0.008/23.46	10.645 0.009/21.94	mras
		10.652 0.011/17.18	10.664 0.011/16.18	10.652 0.009/16.18	10.651 0.007/16.68	u-SIR
		10.598 0.021/9.38	10.637 0.010/10.72	10.630 0.012/9.44	10.66 0.010/10.58	rd-SIR

Table 2: Jump diffusion model (Merton) - prices for an 1-year put option having: $r = 0.05$, $\sigma = 0.1$, $\delta = 0.02$, $S_0 = K = 110$, $\lambda = 2$, $\gamma = 0.2$.

Number of exercices		S_0^* = 30	S_0^* = 50	S_0^* = 70	S_0^* = 90	
$n = 2$		13.451 0.009/24.38	13.472 0.008/23.96	13.472 0.008/22.90	13.460 0.009/23.88	mras
		13.453 0.008/17.22	13.467 0.009/17.50	13.444 0.009/17.36	13.475 0.008/17.04	u-SIR
		13.439 0.008/12.62	13.454 0.009/11.84	13.470 0.010/12.64	13.460 0.007/12.86	rd-SIR
$n = 4$		13.440 0.010/23.78	13.425 0.009/24.24	13.418 0.009/23.56	13.423 0.006/24.06	mras
		13.430 0.010/17.46	13.419 0.007/17.04	13.409 0.009/17.36	13.424 0.008/16.50	u-SIR
		13.426 0.009/11.64	13.403 0.008/11.58	13.428 0.007/12.40	13.407 0.009/11.74	rd-SIR
$n = 6$		13.026 0.045/13.04	13.051 0.022/12.38	13.082 0.017/12.38	13.053 0.017/12.72	mras
		12.907 0.038/11.94	12.970 0.027/11.26	12.984 0.016/11.1	12.951 0.038/11.52	u-SIR
		12.9305 0.033/7.82	12.846 0.048/7.52	12.860 0.055/8.30	12.882 0.053/8.86	rd-SIR

Table 3: Double exponential jump diffusion model (Kou) - prices for an 1-year put option with the following characteristics: $r = 0.05$, $\sigma = 0.1$, $\delta = 0.02$, $S_0 = K = 110$, $\lambda = 3.0$, $p = 0.3$, $\alpha_1 = 40$, $\alpha_2 = 12$.

References

[Cont04] Cont, R., P. Tankov, *Financial Modelling with Jump Processes*, Chapman & Hall/CRC, 2004.

[Hu07] Hu, J., M. C. Fu, S. I. Marcus, *A Model Reference Adaptive Search Method for Global Optimization*, Operations Research, vol. 55, no. 3, pp. 549-568, 2007.

[Kou02] Kou, S., G., *A Jump-Diffusion Model for Option Pricing*, Management Science, INFORMS, vol. 48, no. 8, pp. 1086-1101, 2002.

[Mert76] Merton, R., C., *Option pricing when underlying stock returns are discontinuous*, Journal of Financial Economics, 3, pp. 125-144, 1976.

[Tan09] Tankov, P., E. Voltchkova, *Jump-diffusion models: a practitioner's guide*, Banque et Marchés, No. 99, March-April 2009.

[Zhang06] Zhang, H., M. C. Fu, *Applying model reference adaptive search to american-style option pricing*, Proceedings of the 2006 Winter Simulation Conference.