ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2022.2.076

Approximate Solution for Barrier Option Pricing Using Adaptive Differential Evolution With Learning Parameter

Werry Febrianti, **Kuntjoro Adji Sidarto**, **Novriana Sumarti**

Industrial and Financial Mathematics Research Group, Faculty of Mathematics and Natural Sciences, Ganesha 10 Bandung, Indonesia

werry_febrianti@s.itb.ac.id, werry.febrianti@gmail.com, sidarto@itb.ac.id, novriana@itb.ac.id

Abstract

The Black-Scholes (BS) equation, which has the form of a partial differential equation, is a fundamental equation in mathematical finance, especially for option pricing. Even though there exists an analytical solution of the standard form, it is not straightforward to solve the equation numerically. An effective and efficient numerical method will be useful to solve advanced and non-standard forms of the BS equation in the future. In this paper, we propose a method to solve BS equations using a metaheuristic optimization algorithm to find the best approximate solution. Here we propose the Adaptive Differential Evolution with Learning Parameter (ADELP) algorithm. The BS equations being solved are meant to find values of European option pricing equipped with barrier option. The results of our approximation method fit well with the analytical approximation solutions.

Keywords: *Adaptive differential evolution, Approximation solution, Black-Scholes, Metaheuristic optimization, Partial differential equations.*

Received: 16 October 2022
Accepted: 13 December 2022
Online: 14 December 2022
Published: 20 December 2022

1 Introduction

In asset trading activities such as stock trading, options offer a guarantee for investors to avoid large losses. Black and Scholes [3] together with Merton [19] revolutionized financial markets by introducing the Black-Scholes (BS) equation also known as the Black-Scholes-Merton (BSM) equation. The BS equation, which is in the form of a partial differential equation, provided a new approach to calculating financial market options.

Many different numerical calculations have been used to solve the BS equation, especially finite difference methods. For example, Jeong et al. [14] proposed a finite difference method for solving the BS equation without boundary conditions. Kim et al. [17] used a nonuniform finite difference method for three-dimensional (3D) time-fractional BS equations. He and Zhang [11] proposed the Fractional Black–Scholes Model (FBSM) of option pricing in a fractal transmission system. Gulen et al. [10] proposed the discrete behavior of linear and nonlinear BS European option pricing models using a sixth-order finite difference (FD6) scheme and a third-order stability.

In this paper, we present an approximation solution to the BS partial differential equation (PDE) using a metaheuristic optimization method, which is the novelty of this study, as it is a mesh-free approach. This is based on Chen and Lee [6], who introduced solving the BS equation with a metaheuristic approach using genetic algorithms. Further, we present the solution of BS equations with a trial solution, building on the concept from Khan et al. [16] and Eskiizmirliler et al. [7].

The contribution of this study is to confirm the proposed method for option pricing, especially for barrier option pricing. We use the proposed optimization method to approximate the solution of the BS PDE to determine the option pricing and the BS PDE equipped with barrier option pricing. First, we change the BS PDE into an optimization problem using a residual method. Before using this residual method to solve the BS PDE, we use this method to solve an ordinary differential equation (ODE) as presented in Febrianti et al. [8] and Febrianti et al. [9]. This method was inspired by Babaei [1] and Sadollah et al. [22], who used a weighted residual method to solve ODEs. Chaquet et al. [5] and Panagant et al. [21] solved the PDE using a weighted residual method. Their approach is used here to approximate the solution of the BS PDE with a novel metaheuristic optimization method, called Adaptive Differential Evolution with Learning Parameter (ADELP). The results of the approximate solution of the BS PDE are then compared with the analytical approximation solution as a form of validation of the performance of the proposed method. Finally, the residual method and the ADELP algorithm are used to calculate the approximate solution of the BS PDE for barrier option pricing.

Barrier option pricing is an exotic option that depends on the path-line movement of stock prices in the financial market. The cheapness and marketability it offers investors has made it more popular than other exotic options. Another advantage is that it is more flexible than hedging and speculation. Speculators can choose a variety of barrier options that can help them to monitor possible asset price movement, which in turn, reduces potential losses [24]. The payoff of barrier options depends on the specified barrier level that can be reached by the underlying asset price.

Generally, barrier options are classified either as knock-in options, i.e., the barrier option is activated once the underlying price reaches the barrier level, or knock-out options, i.e., the barrier option is extinguished if the barrier is reached [24]. Barrier option pricing can be calculated using a binomial method, Monte-Carlo simulation, or a BS PDE. In this paper, we focus on a Black-Scholes PDE equipped with barrier option pricing.Previous studies mostly used finite difference methods to approximate BS PDEs equipped with barrier option pricing, for example, [4, 13, 25]. Babasola, et al. [2] used the Crack-Nicolson approach for the valuation of barrier options.

This paper consists of five sections. The first section is the introduction. The second section introduces the trial solution for the BS equation using the neural network concept and a model of the BS equation as an optimization problem. In the third section, we describe the Adaptive Differential Evolution algorithm. The fourth section presents the results and their discussion, and the fifth section contains the conclusion.

2 Neural Network Model for Solving the Black-Scholes Equation

When using optimization to solve differential equations, a trial solution is needed, which can be a series with unknown series coefficients. This trial solution is then substituted in the optimization form of the differential equation. This section discusses a trial solution for the BS, applying the neural network concept and modelling the Black-Scholes equation as an optimization problem. The Black-Scholes differential equation European call option is given as follows:

$$\begin{gathered}\frac{\partial C}{\partial t}+\frac{1}{2}\sigma^2 S^2\frac{\partial^2 C}{\partial S^2}+rS\frac{\partial C}{\partial S}-rC=0\\ 0<S<\infty,\quad 0<t<T\end{gathered}\tag{1}$$

with the following final condition and boundary conditions:

$$\begin{aligned}C(S,T)&=\max\{S(T)-K,0\}\\ C(0,t)&=0,\quad \text{for all } 0\le t\le T\\ C(S,t)&\approx S,\quad \text{for large } S \text{ and } 0\le t\le T\end{aligned}\tag{2}$$

where r and σ, are constants that state the volatility and risk-free interest rate, respectively; $C=C(S,t)$ is the European call option pricing, whose value depends on stock value S and time t; K is the strike price; and T is the maturity time.

Based on the BS differential equation for the European call options (see Eq. 1), we can obtain an approximate analytical solution by transforming it into the standard heat equation $C_\tau = C_{xx}$ using well-known transformations of independent variables $S = Kexp(x)$, $t = T - 2\tau/\sigma^2$. The approximate analytical solution of the BS problem (see Eq. 1) with boundary conditions (see Eq. 2) is found as follows:

$$C(S,t)=SN(d_1)-Ke^{-r(T-t)}N(d_2)\tag{3}$$

where

$$\begin{aligned}d_1&=\frac{\ln\left(\frac{S}{K}\right)+\left(r+\frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}}\\ d_2&=d_1-\sigma\sqrt{T-t}\\ N(x)&=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x}e^{-s^2/2}ds\end{aligned}\tag{4}$$

with N(x) being the normalized normal distribution cumulative function.

2.1 Trial Solution of the Black-Scholes Equation based on the Neural Network Concept

Based on Eskiizmirliler et al. [7], the function $G(x)$ is constructed from the final and boundary conditions (see Eq. 2) such that in general we can write:

$$G(S,t)=\frac{t}{T}\max\{S(t)-K,0\}\tag{5}$$

Eskiizmirliler et al. [7] used the activation function $NN(x,p)$ and a sigmoid-like function $f(z) = 1/(1 + exp(-z))$ stated as follows:

$$NN(X,p)=\sum_{i=1}^{m}\frac{\alpha_i}{1+e^{\mu_i S-\omega_i t-\beta_i}}\tag{6}$$

where X stands for the two PDE variables; p is the unknown parameter vector to be determined; m is the total number of neurons in the hidden layer of the neural network; α_i is the synaptic weight of the i-th hidden neuron of the output; μ_i is the synaptic coefficient from the spatial inputs to the i-th hidden neuron; ω_i is the synaptic coefficient from the time input to the i-th hidden neuron; and β_i is the bias value of the i-th hidden neuron. Based on Eq. (6) and Eskiizmirliler et al. [7], function $F(X, NN(X,p))$ is in the following form:

$$F(X,NN(X,p))=S\cdot(T-t)\cdot\sum_{i=1}^{m}\frac{\alpha_i}{1+e^{\mu_i S-\omega_i t-\beta_i}}\tag{7}$$

Based on Eq. (5), Eq. (6) and Eq. (7), the trial solution for Eq. (1) is given by Eq. (8):

$$\begin{aligned}C_{\text{trial}}(S,t)=&\,t/T\cdot\max(S(t)-K,0)+\\ &S\cdot(T-t)\cdot\sum_{i=1}^{m}\frac{\alpha_i}{1+e^{\mu_i S-\omega_i t-\beta_i}}\end{aligned}\tag{8}$$

2.2 Solving the Black-Scholes Equation as an Optimization Problem

In the previous section, a trial solution for European call options was derived as given in Eq. (1). If Eq. (9) is substituted in Eq. (1), we get the following residual equation:

$$R_n = \Big[\frac{\partial C_{\text{trial}}(S,t)}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C_{\text{trial}}(S,t)}{\partial S^2} + rS\frac{\partial C_{\text{trial}}(S,t)}{\partial S} - rC_{\text{trial}}(S,t)\Big]_{t=t_n, S=S_n} \tag{9}$$

where we take $0 < S < \infty$, and $0 \le t \le T$. In general $R \neq 0$, but we can force R in Eq. (9) to approach 0, which can be satisfied when $C_{\text{trial}} \approx C$. To solve this problem, we need to find the set of coefficients $\{\alpha_i, \mu_i, \omega_i, \beta_i, \mid i = 1, \ldots, m\}$ which minimizes:

$$\sum_{n_S, n_t=1}^{N_S, N_t} R_n^2 \tag{10}$$

where $N_S = ((S_max - S_0))/h_S$ and $N_t = T/h_t$, h_S is the step length of S and h_t is the step length of t. Therefore, we change the problem into an optimization problem in the following way:

$$\text{minimize} \sum_{n_S, n_t=1}^{N_S, N_t} R_n^2 \tag{11}$$

$$\begin{aligned} \text{subject to: } & [C_{\text{trial}}(S,t)]_{t=T, S=S_n} = \max\{S-K, 0\} \\ & [C_{\text{trial}}(S,t)]_{t=t_n, S=S_0} = 0 \end{aligned} \tag{12}$$

Eq. (11) and Eq. (12) together are formulated as an optimization problem with a boundary value problem (BVP) (1)-(2). In Section 3, we discuss the optimization algorithm used in this work for solving Eq. (11) and Eq. (12), which is the Adaptive Differential Evolution with Learning Parameter (ADELP) algorithm.

2.3 European Barrier Option Pricing

The payoff of barrier options depends on whether the asset crosses a pre-defined barrier level. A down-and-out call option has a payoff equal to zero when the asset crosses some pre-defined barrier $B < S_0$ at some time in $[0, T]$ [12].

In the other condition, when the asset does not cross the pre-defined barrier $B < S_0$, the payoff becomes equal to the European call option $\max\{S(T) - K, 0\}$.

Let $C^B(S,t)$ denote the value of a down-and-out call option at asset price S and time t. Then, the BS equation for European down-and-out options given as follows:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS\frac{\partial C}{\partial S} - rC = 0 \qquad 0 < S < \infty, \quad 0 < t < T \tag{13}$$

with the following boundary and final conditions:

$$\begin{aligned} & C^B(B,t) = 0, \quad 0, \le t \le T \\ & C^B(S,T) = \max\{S(T) - K, 0\}, \quad B \le S \end{aligned} \tag{14}$$

The analytical approximation solution for the European down-and-out call in [12] is stated as follows:

$$C^B(S,t) = C(S,t) - \left(\frac{S}{B}\right)^{1-\frac{2r}{\sigma^2}} C\left(\frac{B^2}{S}, t\right) \tag{15}$$

When the probability of hitting the barrier decreases, the initial asset prices increase such that the European barrier down-and-out call option value approaches the value of European call options. Based on Eq. (5), Eq. (6) and Eq. (7), we propose a trial solution to approximate the value of the European down-and-out barrier as in Eq. (16):

$$C_{\text{trial}}(S,t) = t/T \cdot \max(S(t) - K, 0) + (S - B)\cdot(T - t)\cdot \sum_{i=1}^{m} \frac{\alpha_i}{1 + e^{\mu_i S - \omega_i t - \beta_i}} \tag{16}$$

for $S > B$ and $S \le B$, the value of $C^B_{\text{trial}}(S,t) = 0$.

3 Adaptive Differential Evolution with Learning Parameter

The Differential Evolution (DE) algorithm was first introduced by Storn and Price [24]. DE is an evolutionary algorithm inspired by the genetic algorithm. DE has a vector that distinguishes it from other algorithms so that it robust in finding optimal solutions.

DE is a metaheuristic algorithm that is known for the speed with which it finds the optimal solution. Generally, DE uses three different vectors in the optimization process. One vector is called the base vector and the other two vectors are called the difference vectors. The process of updating individuals in DE is as follows: the sum of the difference vectors is added to the base vector after multiplying the sum of the difference vectors with the mutation scale (F). Panagant and Bureerat [20] describe about DE in an oscillation equation where the best value for the mutation scale is 0.5. Generally, the mutation scale value in DE is chosen as a constant value. However, if we think further, this mutation scale factor (F) should be able to change to compensate for shifts in the basis vector so that individual updates always go to the best individual. In this paper, we propose the following new mutation scheme:

$$v^t_{(i,G+1)} = x^t_{(best,G)} + w(x^t_{(best,G)} - x^t_{(i,G)}) \tag{17}$$

where the parameter w is obtained with a learning process that is similar to the process in the Neural Network Algorithm (NNA) from Sadollah et al. [23]. NNA is packaged in such a way that the parameters in it are able to adjust themselves. However, NNA does not converge sufficiently in every program running

Table 1: Parameter settings for NNA, DE and ADELP.

NNA Parameters	DE Parameters	ADELP Parameters
Beta = 1	F = 0.50	Cr = 0.90
Max_Iteration = 1,000	Cr = 0.90	Max_Iteration = 1,000
Population = 200	Max_Iteration = 1,000	Population = 200
	Population = 200	

Table 2: RMSE for European call options with NNA, DE, and ADELP.

Type of errors	Result NNA	Result DE	Result ADELP
RMSE	1.515e+01	6.770e-02	5.802e-02

NNA. Therefore, we propose the concept of a mutation scheme with learning parameter w (related to the mutation scale), so that we obtain a learning process that is similar to that of the NNA. This learning parameter can improve the mutation result compared to only using the ordinary mutation scheme.

Next, the crossover constant (Cr) is chosen large enough (close to 1) to enable a comparison between the results of the previous population and the current one. Then, we use the root mean square error (RMSE) in order to do performance analysis with the following equation:

$$||V-\psi||_{\mathrm{RMSE}} = \left(\frac{1}{N_S\times N_t}\sum_{i=1}^{N_S}\sum_{j=1}^{N_t}|V(S_i,t_j)-\psi(S_i,t_j)|^2\right)^{\frac{1}{2}} \qquad (18)$$

4 Results

In this section, we approximate the BS solution for European call options using ADELP. The algorithm uses a population size of 200 and the maximum number of iterations in these computations is 1,000. All computations are run with MATLAB R2018a on an HP Pavilion laptop, model 14-dv0067TX equipped with an Intel Core TM i7 processor (4.70 GHz) and 8 GB RAM, running Windows 10.

We use the example data from [12]. We consider the value of a call option with strike price $K = 4$. The risk-free interest rate per year is 3% continuously compounded, so $r = 0.03$. The time to expiration is $T = 1$ measured in years, and the volatility, the standard deviation per year on the return of the stock, is $\sigma = 0.3$. The value of the call option at maturity is plotted over the range of stock prices $0 \leq S \leq 10$. First, we calculate the European call option without barrier, then we calculate European call option with a barrier equal to 2.

4.1 European Call Option

In this subsection, we show the result for the European call option using three different optimization algorithms. The first one uses the DE algorithm, the second one uses NNA, and the third one uses ADELP. The parameter settings for DE, NNA, and ADELP are shown in Table 1. The results of the approximation solution are shown in Table 2. The surface of each problem is shown in Fig. 1 and a graph of each problem shows in Fig. 2.

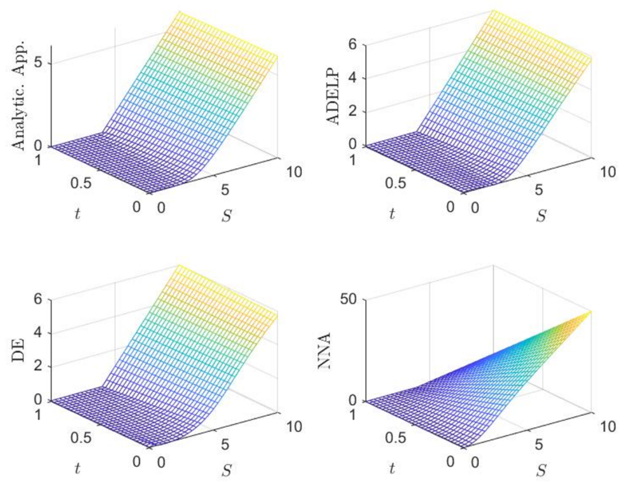

Figure 1: Surface comparison of the approximation solutions of the Black-Scholes equation for European call options from ADELP, DE, and NNA with the analytical approximation solution.

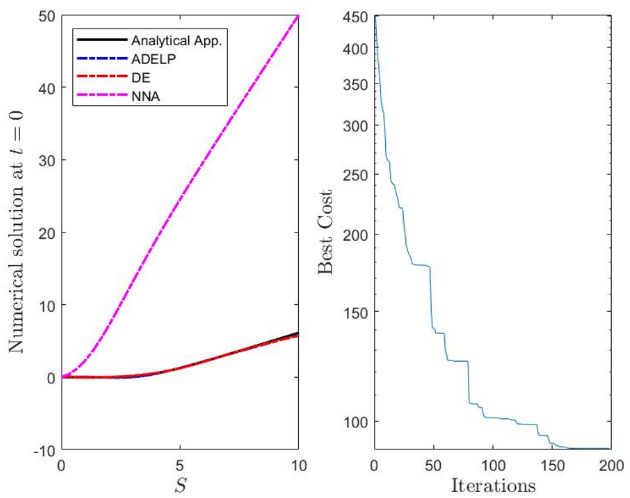

Figure 2: Graph comparison of the approximation solutions of the Black-Scholes equation for European call options from ADELP, DE, and NNA with the analytical approximation solution along with a graph of the best cost value with the best iterations.

Based on the results in Table 2, the RMSE of the

Table 3: RMSE for European down-and out barrier call options with NNA, DE, and ADELP.

Type of errors	Result NNA	Result DE	Result ADELP
RMSE	1.778e+00	5.181e-02	4.417e-02

result of ADELP was the smallest compared to DE and NNA. Therefore, we used ADELP next to approximate BS and BS equipped with barrier option pricing.

In Fig. 1, we show the surface comparison result of ADELP, DE, NNA to analytical approximation solution. Based on the surface results, we conclude that ADELP for European call options could approximate the analytical approximation solution better than DE and NNA. Therefore, the ADELP algorithm can be used to approximate the solution of BS for European call option pricing.

Next, we compare graphs of the ADELP, DE, and NNA result with a graph of the analytical approximation solution (Fig. 2). It can be seen that the graph of ADELP fits well with the graph of the analytical approximation solution. Therefore, ADELP can be a good tool to approximate the analytical approximation solution.

4.2 Down-and-Out Barrier

In this subsection, we show the result for the European call option equipped with down-and-out barrier using ADELP, DE, and NNA. The results of the approximation solution are given in Table 3. The surface of each problem is shown in Fig. 3 and a graph of each problem is shown in Fig. 4.

Based on the results in Table 3, the RMSE of the result of ADELP was the smallest compared to DE and NNA. Therefore, we used ADELP next to approximate BS equipped with barrier option pricing.

In Fig. 3, we show a comparison of the surface results of ADELP, DE, and NNA with the analytical approximation solution. Based on the surface results, we conclude that ADELP for European down-and-out barrier call option pricing could approximate the analytical approximation solution better than DE and NNA. Therefore, the ADELP algorithm can be a good tool to approximate the solution of BS European down-and-out barrier call option pricing.

Next, we compared graph of the results of ADELP, DE, and NNA with a graph of the analytical approximation solution in Fig. 4. It can be seen that the graph of ADELP fits well with the graph of the analytical approximation solution. The graph of ADELP better approximates the graph of the analytical approximation solution than the graphs of DE and NNA. Thus, ADELP can be a good tool to approximate the analytical approximation solution of European down-and-out barrier call option pricing.

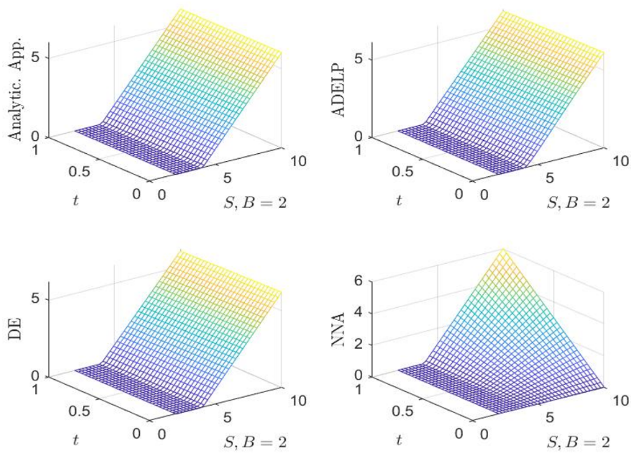

Figure 3: Surface comparison of the approximation solutions of the Black-Scholes equation equipped with down-and-out barrier from ADELP, DE and NNA with the analytical approximation solution.

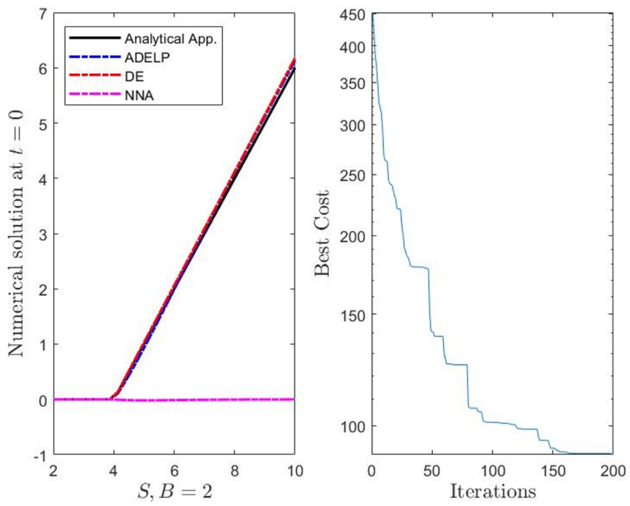

Figure 4: Graph comparison of the approximation solutions of the Black-Scholes equation equipped with down-and-out barrier from ADELP, DE, and NNA with the analytical approximation solution along with the graph of the best cost value with the best iterations.

5 Discussion

In Table 1, we show the parameter settings for NNA, DE and ADELP. Based on Kazikova et al. [15], it is essential to pay attention to the quality of conducted experiments, especially when comparing several different algorithms. Parameter tuning of a metaheuristic algorithm should be an integral part of the development and testing process because it can influence the performance of the algorithm. Therefore, we chose to use a learning parameter with the ADELP algorithm to improve its result compared to DE and NNA. We did not choose a mutation scale as input for the ADELP algorithm because ADELP algorithm find the value of the mutation scale itself during the iteration process. The process of finding the mutation scale is called the learning process and the mutation scale is called the learning parameter. It is appropriate to use a learning parameter with BS partial differential equations because the learning process is appropriate to find a mutation scale that fits the iteration for solving barrier option pricing. ADELP with learning parameter is the novelty of this paper and it fits well with the BS PDE for barrier option pricing.

In metaheuristic algorithms like ADELP, DE, and NNA, the population is initialized randomly. Matousek et al. [18] proposed a way to select a good starting solution. One possibility is to start from a random solution in the hope that after a sufficiently large number of tries, one gets a 'good enough' solution. In our experiment, we used 1,000 iterations and a population of 200.

The method proposed by Matousek et al. [18] solves the quadratic assignment problem (QAP) by using a fusion of two approaches, whereby the solutions from the computation of the lower bounds are used as the starting points for a metaheuristic optimization algorithm, called HC12, which is implemented on a GPU CUDA platform. Adding lower bound techniques in constructing the starting point has a significant impact on the quality of the resulting solutions.

Barrier option pricing is an NP-hard optimization problem because it uses a BS PDE. Therefore, it is attractive to use a metaheuristic algorithm that can find high-quality solutions within an acceptable computation time. We used zero as the lower bound of the strike price and as the lower bound of the maturity time. We used this lower bound to make sure that the result of our exercise was always positive, because it is related to the option price.

6 Conclusion

In this paper, we proposed using a residual function to solve the BS equation with given boundary conditions in order to predict the value of European call options and European call options equipped with barrier option pricing. We also described how to transform the option pricing problem into an optimization problem using this residual function. Then, we used Adaptive Differential Evolution with a learning parameter to find the approximation solution for the BS PDE to calculate the value of European call options and European call options equipped with barrier options.

This optimization method can be used in the calculation of the approximation solution of BS equipped with barrier options or without barrier options. The benefit of using this method is that we can approximate the solution without using a mesh method. It also enables us to find the approximate solution without transformation, so that all variables are defined well.

In the future, we will try to apply the proposed method to options that do not have an exact solution, such as other types of exotic options.

Acknowledgement: This research was funded by Indonesian Minister of Research and Technology/National Research and Innovation Agency, Indonesia, Grant Number 2/E1/KP.PTNBH/2021.

References

[1] Babaei, M. A general approach to approximate solutions of nonlinear differential equations using particle swarm optimization. *Applied Soft Computing 13*, 7 (2013), 3354–3365.

[2] Babasola, O., Ngare, P., and Owoloko, E. Crank nicolson approach for the valuation of the barrier options. *International of Journal of Applied Mathematical Sciences (JAMS) 11* (2018), 7–21.

[3] Black, F., and Scholes, M. The pricing of options and corporate liabilities. *Journal of political economy 81*, 3 (1973), 637–654.

[4] Boyle, P. P., and Tian, Y. An explicit finite difference approach to the pricing of barrier options. *Applied Mathematical Finance 5*, 1 (1998), 17–43.

[5] Chaquet, J. M., and Carmona, E. J. Solving differential equations with fourier series and evolution strategies. *Applied Soft Computing 12*, 9 (2012), 3051–3062.

[6] Chen, S.-H., and Lee, W.-C. Option pricing with genetic algorithms: The case of european-style options. In *ICGA* (1997), pp. 704–711.

[7] Eskiizmirliler, S., Günel, K., and Polat, R. On the solution of the black–scholes equation using feed-forward neural networks. *Computational Economics 58*, 3 (2021), 915–941.

[8] Febrianti, W., Sidarto, K. A., and Sumarti, N. Solving some ordinary differential equations numerically using differential evolution algorithm with a simple adaptive mutation scheme. In *AIP Conference Proceedings* (2021), vol. 2329, AIP Publishing LLC, p. 040003.

[9] Febrianti, W., Sidarto, K. A., and Sumarti, N. Solving systems of ordinary differential equations using differential evolution algorithm with the best base vector of mutation scheme. In *AIP Conference Proceedings* (2021), vol. 2423, AIP Publishing LLC, p. 020006.

[10] Gulen, S., Popescu, C., and Sari, M. A new approach for the black–scholes model with linear and nonlinear volatilities. *Mathematics 7*, 8 (2019), 760.

[11] He, J., and Zhang, A. Finite difference/fourier spectral for a time fractional black–scholes model with option pricing. *Mathematical Problems in Engineering 2020* (2020).

[12] Higham, D. J. An introduction to financial option valuation: mathematics, stochastics and computation.

[13] Ioffe, G., and Ioffe, M. Application of finite difference method for pricing barrier options. *Paper available at www. egartech. com* (2003).

[14] Jeong, D., Yoo, M., and Kim, J. Finite difference method for the black–scholes equation without boundary conditions. *Computational Economics 51*, 4 (2018), 961–972.

[15] Kazikova, A., Pluhacek, M., and Senkerik, R. Why tuning the control parameters of metaheuristic algorithms is so important for fair comparison? *Mendel 26*, 2 (2020), 9–16.

[16] Khan, N. A., Razzaq, O. A., and Hameed, T. A metaheuristic approach to optimize european call function with boundary conditions. *Journal of Systems Science and Information 6*, 3 (2018), 260–268.

[17] Kim, S., Lee, C., Lee, W., Kwak, S., Jeong, D., and Kim, J. Nonuniform finite difference scheme for the three-dimensional time-fractional black–scholes equation. *Journal of Function Spaces 2021* (2021).

[18] Matousek, R., Dobrovsky, L., and Kudela, J. How to start a heuristic? utilizing lower bounds for solving the quadratic assignment problem. *International Journal of Industrial Engineering Computations 13*, 2 (2022), 151–164.

[19] Merton, R. C. Theory of rational option pricing. *The Bell Journal of economics and management science* (1973), 141–183.

[20] Panagant, N., and Bureerat, S. Differential evolution algorithm for solving a nonlinear single pendulum problem. In *Advanced Materials Research* (2014), vol. 931, Trans Tech Publ, pp. 1129–1133.

[21] Panagant, N., and Bureerat, S. Solving partial differential equations using a new differential evolution algorithm. *Mathematical Problems in Engineering 2014* (2014).

[22] Sadollah, A., Eskandar, H., Kim, J. H., et al. Approximate solving of nonlinear ordinary differential equations using least square weight function and metaheuristic algorithms. *Engineering Applications of Artificial Intelligence 40* (2015), 117–132.

[23] Sadollah, A., Sayyaadi, H., and Yadav, A. A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm. *Applied Soft Computing 71* (2018), 747–782.

[24] Storn, R., and Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization 11*, 4 (1997), 341–359.

[25] Umeorah, N., and Mashele, P. A crank-nicolson finite difference approach on the numerical estimation of rebate barrier option prices. *Cogent Economics & Finance* (2019).