International Journal of Analysis and Applications

Application of Ant Colony Programming Approach for Solving Systems of Stochastic Differential Equations

Ali Sami Rashid[1,*], Salah H Abid[2], Sadiq A. Mehdi[2]

[1]*Mathematics Department, College of education, University of Misan, Misan, Iraq*

[2]*Mathematics Department, College of education, Mustansiriyah University, Baghdad, Iraq*

**Corresponding author:* alisamirashid@uomisan.edu.iq

ABSTRACT. Stochastic differential equations (SDE) have wide applications in natural phenomena, engineering, finance, and biological models. Obtaining analytic solutions for an SDE is often complex, and the complexity increases for an SDE system. The paper introduces ant colony programming (ACP) as a novel approach for solving SDE system. Ant colony programming was developed in two directions, the first is to add the Wiener process W_t as a variable to the terminals and functions, and the second is to construct the appropriate fitness function FF. ACP constructs mathematical expressions and evaluates them using the fitness function FF. The ACP proposed effectiveness has been demonstrated by applying to 2,3 and 4-dimensional SDE systems. The most important finding of this work is that ACP generates symbolic stochastic processes that represent solutions for SDE system. Methods for solving SDE systems are important tools for study phenomena that involve noise or randomness.

1. Introduction

A stochastic differential equation (SDE) is a dynamical system that describes the behavior of systems under random influences, the influence of which may be internal to the system or external [1]. The stochastic term in SDE represents the randomness in the system [2]. SDEs are widely used in various scientific [3]. The solution to SDE is a stochastic process characterized by probability distributions [4].

Received May 20, 2024

2020 Mathematics Subject Classification. 60H10, 65C30, 37H10.

Key words and phrases. stochastic differential equation system; ant colony programming; evolutionary algorithm; algorithms; computational methods.

https://doi.org/10.28924/2291-8639-22-2024-105

ISSN: 2291-8639

Previous literature on SDEs has dealt with analytical methods limited to solving some important examples of a single SDEs, for instance Geometric Brownian Motion, Ornstein-Uhlenbeck Process, Stochastic Logistic Growth and others [5], [6], [7]. And numerical methods give approximate solutions with an acceptable error rate, such as Euler-Maruyama method [8], Milstein method [9], Runge-Kutta method [10], Monte Carlo method and finite difference methods [11], [12].

The research problem is based on finding optimal solutions for SDE systems using the ant colony programming (ACP), which is considered one of the important evolutionary algorithms used in finding the optimal path. ACP algorithm was developed by Boryczka and Czech and has been used to solve approximation problems [13], [14]. Kumaresan and Balasubramaniam developed the ACP to solve a stochastic linear singular system [15], Kamali modified the ACP to solve differential equations [16].

The researchers will develop the ACP algorithm in two axes. The first is to choose node names in the search graph space, that formed of arithmetic operations, functions, variables and constants. Second, constructing a fitness function FF for evaluating the mathematical expressions formed by the ants' tours in a way that suits the algebraic formulation of SDE system under study. Finally, it will be proven that the optimal solutions generated by the ACP algorithm represent exact solutions for the SDE systems studied.

2. Stochastic differential equation system

An Itô process X_t is a stochastic process with respect to a Wiener process W_t has the form

$$X(t) = a(X_t)dt + b(X_t)dW_t,\ \ t \geq 0 \tag{2.1}$$

With initial value $X_0 = X(t_0)$, and continuous functions a and b.

Consider a function $\varphi: [0, T] \times \mathbb{R} \to \mathbb{R}$ with continuous partial derivatives $\frac{\partial \varphi}{\partial t}, \frac{\partial \varphi}{\partial x}, \frac{\partial^2 \varphi}{\partial x^2}$ are exist. The Itô formula is given by

$$\begin{aligned} d\varphi(t, X_t) = &\left(\frac{\partial \varphi(t, X_t)}{\partial t} + a(X_t)\frac{\partial \varphi(t, X_t)}{\partial x} + \frac{1}{2}b^2(X_t)\frac{\partial^2 \varphi(t, X_t)}{\partial x^2}\right)dt \\ &+ b(X_t)\frac{\partial \varphi(t, X_t)}{\partial x}dW_t \end{aligned} \tag{2.2}$$

Consider a stochastic process $X(t)$ satisfies (2.1). The general form of SDE is:

$$dX(t) = f\big(t, X(t)\big)dt + g\big(t, X(t)\big)dW_t \tag{2.3}$$

Where $f\big(t, X(t)\big)$ and $g\big(t, X(t)\big)$ are functions that determine the drift and diffusion coefficients, respectively, and W_t is a Wiener process [17]. SDE (2.3) describes the evolution of the stochastic process $X(t)$ over time $[0, T]$ and includes both deterministic and stochastic terms. The stochastic term represents the randomness or noise in the system, while the deterministic term represents the drift or trend [18].

The set m of SDEs is known as SDE system with dimension m, and written as [19],

$$dX(t) = f\big(t, X(t)\big)dt + g\big(t, X(t)\big)dW(t)\,, \qquad t \in [t_0, T] \tag{2.4}$$

Where vector function $f: [t_0, T] \times \mathbb{R}^m \to \mathbb{R}^m$, a matrix function $g: [t_0, T] \times \mathbb{R}^m \to \mathbb{R}^{m\times m}$ and $W = \{W_t, t \geq t_0\}$ is a m- dimension Wiener process, or rewrite in matrix form as,

$$\begin{aligned}\begin{bmatrix} dX_1(t) \\ dX_2(t) \\ \vdots \\ dX_m(t) \end{bmatrix} &= \begin{bmatrix} f_1\big(t, X_1(t), \dots, X_m(t)\big) \\ f_2\big(t, X_1(t), \dots, X_m(t)\big) \\ \vdots \\ f_m\big(t, X_1(t), \dots, X_m(t)\big) \end{bmatrix} dt \\ &+ \begin{bmatrix} g_{11}\big(t, X_1(t)\big) & g_{12}\big(t, X_2(t)\big) & \cdots & g_{1m}\big(t, X_m(t)\big) \\ g_{21}\big(t, X_1(t)\big) & g_{22}\big(t, X_2(t)\big) & \cdots & g_{2m}\big(t, X_m(t)\big) \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1}\big(t, X_1(t)\big) & g_{m2}\big(t, X_2(t)\big) & \cdots & g_{mm}\big(t, X_m(t)\big) \end{bmatrix} \begin{bmatrix} dW_1(t) \\ dW_2(t) \\ \vdots \\ dW_m(t) \end{bmatrix}\end{aligned} \tag{2.5}$$

With vector initial condition $X(t_0) = X_0$.

The solution of SDE system (2.5) is

$$X(t) = X_0 + \int_0^t f\big(u, X(u)\big)du + \int_0^t g\big(u, X(u)\big)dW(u) \tag{2.6}$$

In system components form,

$$X_i(t) = X_i(0) + \int_0^t f_i\big(u, X_1(u), \dots, X_d(u)\big)du + \sum_{j=1}^{m} \int_0^t g_{ij}\left(u, X_j(u)\right) dW_j(u) \tag{2.7}$$

where $\int_0^t f\big(u, X(u)\big)du$ is the Lebesgue integral, and $\int_0^t g\big(u, X(u)\big)dW_j(u)$ is the Itô's integral.

As especial case, for a 1- dimensional Wiener process W_t, then the formula of SDE system (2.5) is given by

$$\begin{bmatrix} dX_1(t) \\ dX_2(t) \\ \vdots \\ dX_m(t) \end{bmatrix} = \begin{bmatrix} f_1\big(t, X_1(t), \dots, X_m(t)\big) \\ f_2\big(t, X_1(t), \dots, X_m(t)\big) \\ \vdots \\ f_m\big(t, X_1(t), \dots, X_m(t)\big) \end{bmatrix} dt + \begin{bmatrix} g_1\big(t, X_1(t), \dots, X_m(t)\big) \\ g_2\big(t, X_1(t), \dots, X_m(t)\big) \\ \vdots \\ g_m\big(t, X_1(t), \dots, X_m(t)\big) \end{bmatrix} dW_t \tag{2.8}$$

Where $\Delta W_{t_i} = W_{t_i} - W_{t_j} \sim N(0, \Delta t)$, for each $t_0 \leq t_i, t_j \leq T$.

3. Ant colony programming

ACP was proposed as a metaheuristic algorithm used to address optimization problems, inspired by the foraging behavior of ants [20]. ACP utilizes a graph-based representation of the problem space, where nodes represent variables, functions, arithmetic operations and constants. Edges represent connections between the nodes [13].

In ACP, a population of "simulated ants" is employed to explore the graph and find optimal solutions. Each ant constructs a solution to the problem under study by traversing the search graph, starting from an initial node and moving to neighboring nodes according to a probabilistic decision rule. The decision rule is influenced by the pheromone trails deposited on the edges of the graph [14].

The pheromone trails represent the collective knowledge of the ant population and are updated based on the quality of the solutions found. Ants deposit pheromone on the edges of the graph corresponding to the nodes they visit during their search. The amount of pheromone deposited is proportional to the quality of the solution found. Over time, the pheromone evaporates, allowing exploration of new paths.

The probabilistic decision rule used by the ants is influenced by both the pheromone trails and a heuristic value associated with each edge. The heuristic value represents the attractiveness of a particular edge based on problem-specific information. The combination of pheromone trails and heuristic values guides the ants towards promising nodes of the search space.

Successive iteration of the algorithm, which includes updating the pheromone trails and evaporation leads to enhancing the pheromone level at the path of promising nodes, thus increasing the probability of ants moving to them, making the ants' expressions closer to the optimal solution. The algorithm stops when terminal criteria are met, such as reaching a maximum number of iterations or finding an optimal solution. The process flowchart of the ACP algorithm is shown in Fig. 1.

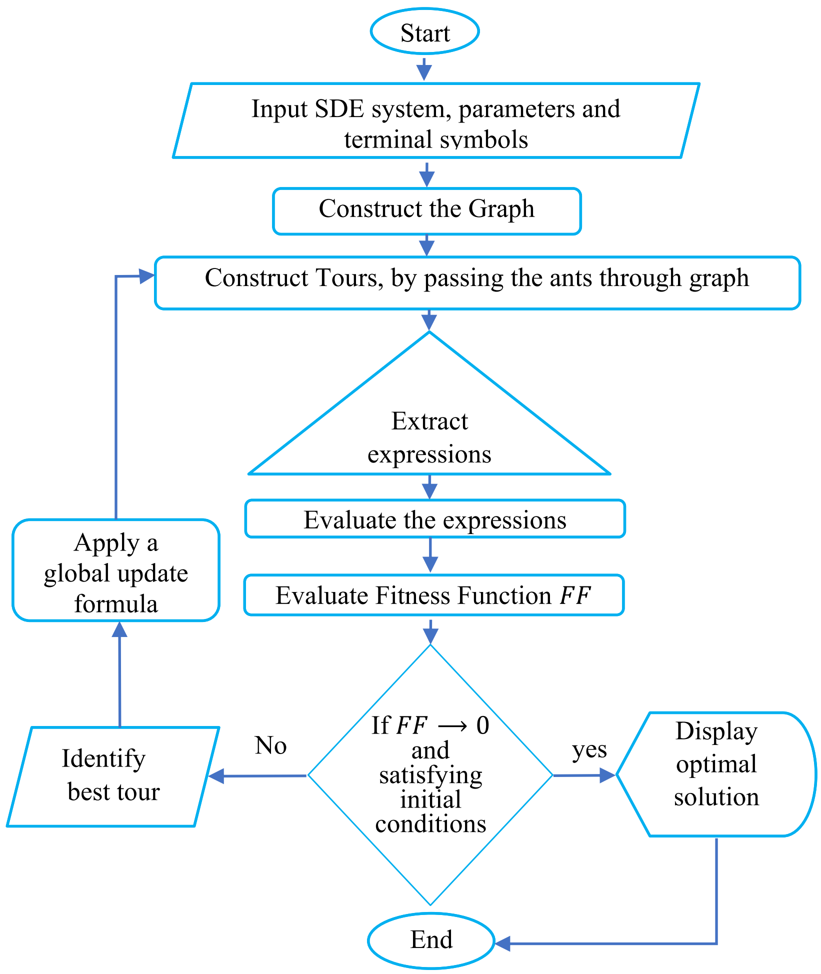

Figure 1: ACP algorithm flowchart

Boryczka and Wiezorek identified four essential steps in researching process [14], [21],

- Choice of functions and terminals,
- Graph Construction,
- Construction of fitness function, and
- Defining terminal criteria.

3.1. Terminals and functions

Choice of functions and terminal symbols in the ACP algorithm depends on the problem studied. In ACP approach, terminal symbols include constants $\{0, 1, \dots, 9, \pi, e\}$, variables $\{t, x, y, W_t, \dots\}$, and functions $\left\{sin, cos, exp, log, \wedge\frac{m}{n}\right\}$ where m, n are integers and $n \neq 0$. Functions can be designated as arithmetic operations $\{+, -, *, \div\}$, Boolean operations $\{\vee, \wedge, \Rightarrow, \Leftrightarrow\}$, or functions defined with a particular form appropriate to the problem.

The choice of terminals and functions is crucial in representing the problem and finding the optimal solution. In SDE system problems, by choosing the appropriate combination of terminals and functions, ACP can efficiently explore the search space and generate expressions forming stochastic processes that solving the given SDE system. The terminals and functions are chosen as in Table 1.

Table 1: The functions and terminal symbols.

Terminal symbol or function	
$t_i \in \mathbb{T}$	$\mathbb{T} = \{0,1,2,3,4,5,6,7,8,9, t, W_t\}$
$f_i \in \mathbb{F}$	$\mathbb{F} = \{ +, -, *, /,), (, sin, cos, exp, log, sqrt\}$

3.2. Graph construction

The construction of the graph G is an initial step in ACP as it provides structure for the ants to navigate and search for solutions, involves construct a graph G with L nodes and edges to represent the search space. Each node in G represents either a function $f_i \in \mathbb{F}$ or a terminal symbol $t_i \in \mathbb{T}$. The edges in G connect the nodes and are weighted by pheromone. Fig. 2 is illustrative Graph of ACP.

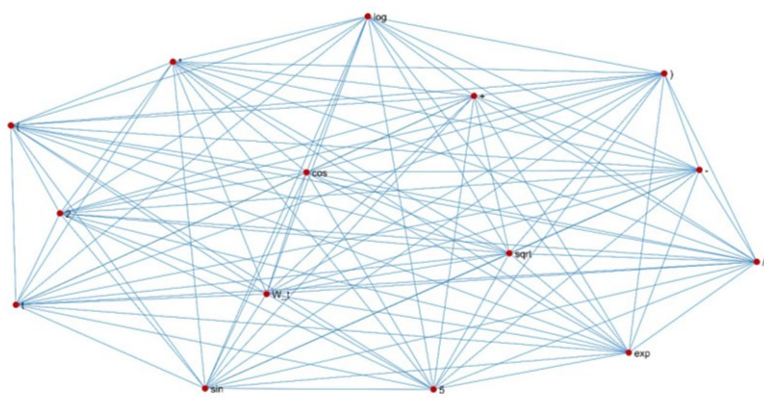

Figure 2: Illustrative ACP graph.

3.3. Fitness function construction

The fitness function FF is a form of the objective function. In ACP algorithm FF plays a crucial role in evaluating and selecting the solutions of given problem, the formulation of FF depends on the nature of the problem to be solved [14].

In ACP algorithm, FF is utilized to evaluate the quality or fitness of the solutions generated by the ant tours. It is a norm of how well the solution satisfies the desired criteria or objectives of the problem. In solving SDE system, the objective of FF is to filter out the best solution among the generated solutions and to determine the fitness value of each solution and guide the search process towards finding the optimal solution.

If ACP produces a m −dimensional stochastic process $\phi\big(t, X(t)\big)$, then the Itô formula for the r component takes the form,

$$d\phi_r = \hat{f}_r(t, X_1, \dots, X_m)dt + \sum_{s=1}^{m} \hat{g}_{rs}(t, X_s)dW_s(t) \tag{3.1}$$

The fitness function is,

$$FF = \sum_{r=1}^{m} \Big(\hat{f}_r(t, X_1, \dots, X_m) - f_r(t, X_1, \dots, X_m)\Big)^2 + \sum_{s=1}^{m} \big(\hat{g}_{rs}(t, X_s) - g_{rs}(t, X_s)\big)^2 \tag{3.2}$$

Where f_r, g_{rs} are drift and diffusion functions of the component r in the SDEs system (2.5).

3.4. Terminal criteria

In the context of the ACP technique, the terminal criteria refer to the conditions that determine when the program stops searching for solutions. In each generation, the ants are sent to traverse through the graph. If an ant finds an expression that gives a fitness function value of zero and satisfies the initial conditions, the program stops. Otherwise, the tour with the minimum FF value is identified as the best ant tour, and the global update rule is applied to update the pheromone values in the graph, and the process is repeated until FF value equal or close to zero is obtained.

4. ACP methodology

The ACP algorithm starts with

- Choose the appropriate terminals and functions for the problem.
- Determine the number of nodes, and create the graph.
- Choose the number of simulated ants.
- Determine the maximum number of iterations.
- Define parameter values.

Indexed ants are sent to search for available solutions for the SDE system under study. The ants navigate through nodes in the graph $G(V, E)$, where the nodes V represent the functions f_i and terminal symbols t_i, and E the set of connecting edges between the nodes that are weighted by the pheromone concentration.

Each ant k navigates from node i to node j on graph G at time t according to the probability law:

$$p_{ij,k}(t) = \frac{\tau_{ij}(t) \cdot [\gamma_j]^\beta}{\sum_{r \in J_i^k} [\tau_{ir}(t)] \cdot [\gamma_r]^\beta} \tag{4.1}$$

Where τ_{ij} represents pheromone concentration at the edge (i, j), $\gamma_s = \left(\frac{1}{2+\pi_j}\right)^d$, and π_j is power of node j that can be belongs T or F, value π_j is shown in Table 2, d is the current length of the expression, the pheromone concentration and visibility on the ant trail are controlled by parameter $\beta = 0.8$ and $J_i^k$is the set of nodes not visited by ant k from node i.

After completing the ant tours, a parse tree is performed for each tour, the mathematical expressions are generated, evaluate the expressions and exclude expressions that do not forming valid mathematical functions. For example, if one ant produces $sin\,(W_t/5 * e) + t$ and another ant produces $W_t +) * e^t$, the first expression is evaluable while the second cannot. Table 3 showing the evaluable mathematical expressions corresponding to Virtual tours. The mathematical expressions (evaluable) are directed to be substituted into FF. If the value of FF is equal to or close to zero and the initial conditions of the SDE system under study are satisfied, then the generation of additional ant tours stops. Otherwise, the global update of the pheromone values on the edges of the graph is performed according to the following law:

$$\tau_{ij}(t+1) = (1-\rho).\,\tau_{ij}(t) + \rho.\frac{1}{L} \tag{4.2}$$

Where t is the generation number, ρ is the parameter of pheromone decay coefficient with the rang (0,1], and L is the length of the best tour.

After updating the pheromone and determining the best tour in the previous generation, send the ants back through the best tour.

Table 2: Power of functions and terminal symbols

function of terminal symbol	Power
Variable or constant	-1
Functions or closing parenthesis)	0
Arithmetic operations +, - , * , / or opening parenthesis (	1

Table 3: Virtual tours and corresponding expressions

Ant tours	Expressions	Status
$\exp(W_t) * t$	e^{tW_t}	evaluable
$\sin)t + W_t$	$\sin)t + W_t$	Non-evaluable
$\cos(W_t * t * 2/5)$	$\cos\left(\frac{2tW_t}{5}\right)$	evaluable
$\sin(W_t\, *)$	$\sin(W_t\, *)$	Non-evaluable
$2 * t * \exp(W_t/t)$	$2te^{\frac{W_t}{t}}$	evaluable
$\exp\log(W_t)$	W_t	evaluable

The ACP algorithm described with the following,

Step 1. Start with input parameters and terminal symbols.

Step 2. Construct the graph.

Step 3. As a starting point, set equal values of pheromone on all edges.

Step 4. Construct tours, by passing N ants through nodes, and moving from node to the other according to Equation (4.1).

Step 5. Save the ant tours and construct parse trees for it.

Step 6. Extract the expressions and exclude unwanted ones.

Step 7. Evaluate the expressions, substitute the value of time T and Wiener process W_t into the expressions.

Step 8. Evaluate fitness function FF.

Step 9. If $FF \rightarrow 0$ and satisfying initial conditions; stop, and go to Step 13.

Step 10. Otherwise, identify the tour with minimum FF (best tour).

Step 11. Perform a global update of the pheromone values on the graph edges by applying law (4.2).

Step 12. Go to Step 4.

Step 13. Display the solution. End.

5. Simulation results of ACP

In this work, an ACP algorithm will be designed to simulate 2, 3 and 4-dimensional SDE systems (5.1), (5.4) and (5.7) on time interval [0,1]. The mathematical expressions generated by ACP algorithm and the parse trees for the best ant tours, as well as the optimal solution corresponding to each component of SDE systems will be shown figures. Finally, the ACP solution for each SDE system will be proven to be the exact solution.

According to previous literature [16], [22], the control parameters for the ACP method were selected in Table 4.

Table 4: ACP control parameters

Parameters	value
Population size	100
Terminals	$\{0,1,\ldots,9,t,W_t\}$
Functions	$\{+,-,*,/,sin,cos,exp,log,sqrt\}$
Max generation	2000

5.1. 2-dimensional SDE system

Consider $Y_t = W_t$ is the 1-dimensional Wiener process, and the 2-dimensional SDE system

$$\begin{aligned} dX_1(t) &= \frac{-1}{2} X_1(t)dt - X_2(t)dW_t \\ dX_2(t) &= \frac{-1}{2} X_2(t)dt + X_1(t)dW_t \end{aligned} \tag{5.1}$$

With initial condition $X(0) = \begin{bmatrix} X_1(0) \\ X_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $t \in [0,1]$.

Fig. 3 shows the parse trees corresponding to the optimal ant tours with $FF = 0$, for each component of the solution for SDE system (5.1). The ACP algorithm produced the following optimal solution for the SDE system (5.1):

$$X = \begin{bmatrix} \cos W_t \\ \sin W_t \end{bmatrix} \tag{5.2}$$

As Y_t is an Itô process, the Itô formula (2.2) for the components X_1 and X_2 of the stochastic process (5.2) satisfy the following,

$$\begin{aligned} dX_1(t) &= -\frac{1}{2} \cos W_t \, dt - \sin W_t \, dW_t \\ dX_2(t) &= -\frac{1}{2} \sin W_t \, dt + \cos W_t \, dW_t \end{aligned} \tag{5.3}$$

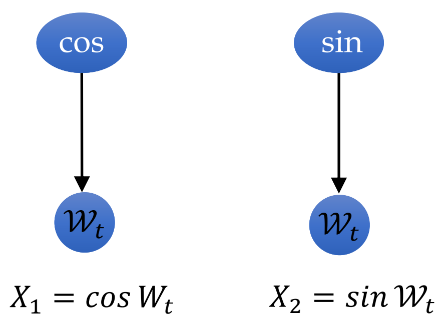

Figure 3: Parse tree for the solution of SDE system (5.1) by ACP method

Therefore, the stochastic process (5.2), is the exact solution of the SDE system (5.1), and X_1 and X_2 are satisfy the initial conditions $X_1(0) = 1$ and $X_2(0) = 0$.

The optimal solution and best ant tours generated by the ACP algorithm for the components of the SDE system (5.1) are plotted in Fig. 4.

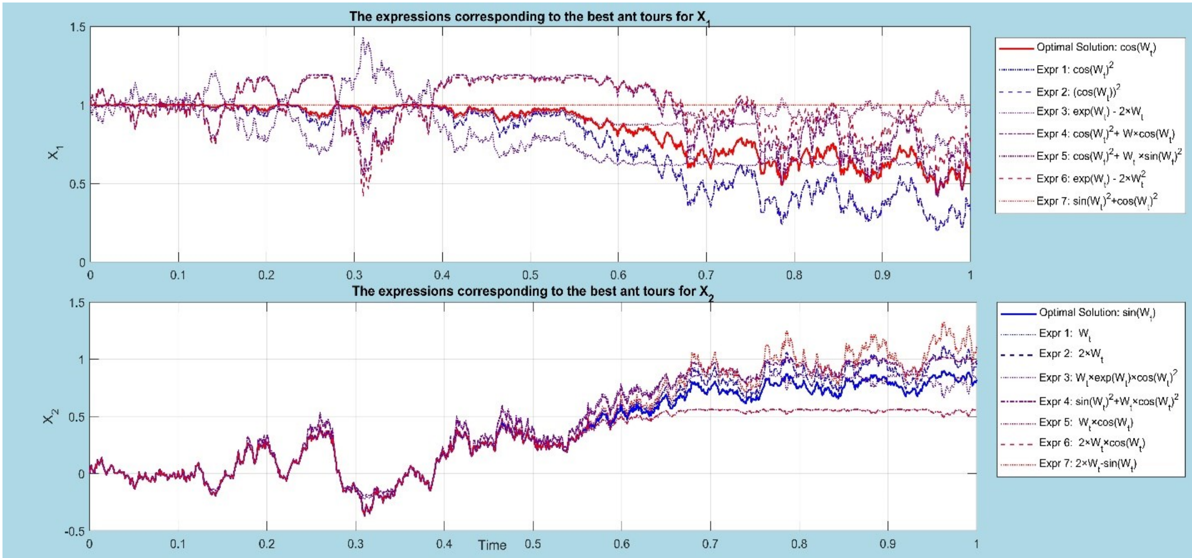

Figure 4: Exact solution, optimal solution and last 7 best ant tours for the SDE system (5.1)

5.2. 3-dimensional SDE system

Consider the following 3-dimensional SDE system with respect to the 1-dimensional Wiener process W_t.

$$\begin{aligned} dX_1(t) &= \left(\frac{2-t}{2t}\right) X_1(t)dt - X_2(t)dW_t \\ dX_2(t) &= \left(\frac{2-t}{2t}\right) X_2(t)dt + X_1(t)dW_t \\ dX_3(t) &= \left(\frac{1}{t}X_2(t) - \frac{1}{2}X_3(t)\right)dt + \left(X_1(t) - \frac{1}{t}X_2(t)\right)dW_t \end{aligned} \tag{5.4}$$

With initial conditions $X_1(0) = 0$, $X_2(0) = 0$, $X_3(0) = 1$ and $t \in [0,1]$.

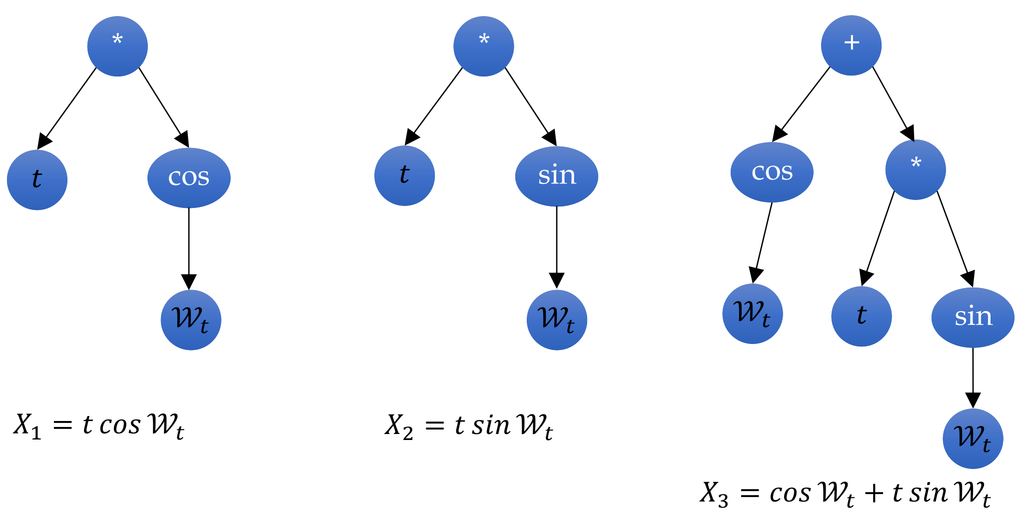

Figure 5: Parse tree for the solution of SDE system (5.4) by ACP method

Fig. 5 shows the parse trees corresponding to the optimal ant tours with $FF = 0$, for each component of the solution for SD E system (5.4).

The ACP algorithm produced the following optimal solution for the SDE system (5.4):

$$X(t) = \begin{bmatrix} t\ \cos W_t \\ t\ \sin W_t \\ \cos W_t + t\ \sin W_t \end{bmatrix} \tag{5.5}$$

By the Itô formula (2.2) for the components X_1 ,X_2 and X_3 of the stochastic process (5.5), we obtain

$$\begin{aligned} dX_1(t) &= \left(\frac{2-t}{2t}\right) t \cos W_t\, dt - t\ \sin W_t\, dW_t \\ dX_2(t) &= \left(\frac{2-t}{2}\right) \sin W_t\, dt + t\ \cos W_t\, dW_t \\ dX_3(t) &= \left(\sin W_t - \frac{1}{2}\cos W_t - \frac{1}{2} t\ \sin W_t\right) dt + (t \cos W_t - \sin W_t) dW_t \end{aligned} \tag{5.6}$$

Therefore, the stochastic process (5.5), is the exact solution of the SDE system (5.4), and the components X_1, X_2, X_3 are satisfy the initial conditions $X_1(0)$ =0, $X_2(0) = 0$ and $X_3(0) = 1$.

The optimal solution (5.5) and best ant tours generated by the ACP algorithm for the components of the SDE system (5.4) are plotted in Fig. 6.

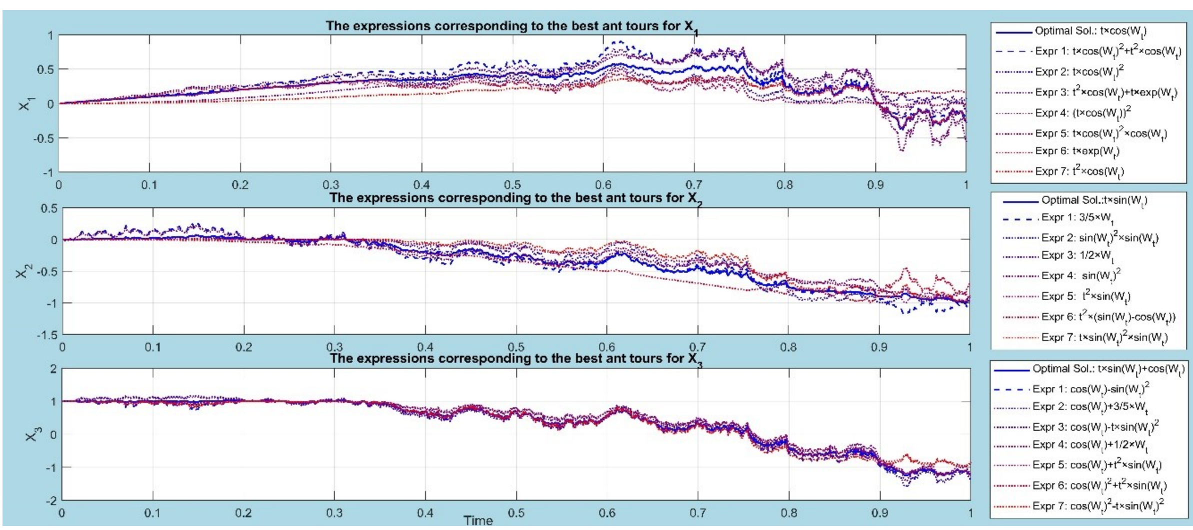

Figure 6: Exact solution, optimal solution and last 7 best ant tours for the SDE system (5.4)

5.3. 4-dimensional SDE system

Consider the following 4-dimensional SDE system with respect to the 1-dimensional Wiener process W_t.

$$\begin{aligned} dX_1 &= \left(X_3(t) - X_2(t) - \frac{1}{2}X_1(t)\right) dt \ + \ t\big(X_1(t) - \ X_4(t)\big) dW_t \\ dX_2 &= \left(X_4(t) - X_1(t) - \frac{1}{2}X_2(t)\right) dt \ + \ \ X_1(t) dW_t \end{aligned} \tag{5.7}$$

$$dX_3 = \left(\frac{1}{t}X_2(t) - \frac{1}{2}X_3(t)\right)dt \; + \left(X_4(t) - \frac{2}{t}X_2(t)\right)dW_t$$
$$dX_4 = \left(\frac{1}{t}X_1(t) - \frac{1}{2}X_4(t)\right)dt \; + \left(\frac{2}{t}X_1(t) - X_3(t)\right)dW_t$$

With initial conditions $X_1(0) = 0$, $X_2(0) = 0$, $X_3(0) = 1$, $X_4(0) = 0$, and $t \in [0,1]$.

Fig. 7 gives the parse trees corresponding to the optimal ant tours with $FF = 0$, for each component of the SDE system (5.7) solution. The ACP algorithm produced the following optimal solution for the SDE system (5.7):

$$X(t) = \begin{bmatrix} t\ \cos W_t \\ t\ \sin W_t \\ \cos W_t + t\ \sin W_t \\ \mathrm{t} \cos W_t + \sin W_t \end{bmatrix} \tag{5.8}$$

The Itô formula (2.2) for the components X_1 , X_2 , X_3 and X_4 of the stochastic process (5.8) satisfies the following:

$$\begin{aligned} dX_1(t) &= \left(\cos W_t - \frac{1}{2}t\cos W_t\right)dt - t\sin W_t\, dW_t \\ dX_2(t) &= \left(\sin W_t - \frac{1}{2}t\sin W_t\right)dt + t\cos W_t\, dW_t \\ dX_3(t) &= \left(\sin W_t - \frac{1}{2}\cos W_t - \frac{1}{2}t\sin W_t\right)dt + (t\cos W_t - \sin W_t)dW_t \\ dX_4(t) &= \left(\cos W_t - \frac{1}{2}(t\cos W_t + \sin W_t)\right)dt + (\cos W_t - t\sin W_t)dW_t \end{aligned} \tag{5.9}$$

Therefore, the stochastic process (5.8), is the exact solution of the SDE system (5.7), and the components X_1, X_2, X_3, X_4 are satisfy the initial conditions $X_1(0) = 0$, $X_2(0) = 0$, $X_3(0) = 1$ and $X_4(0) = 0$.

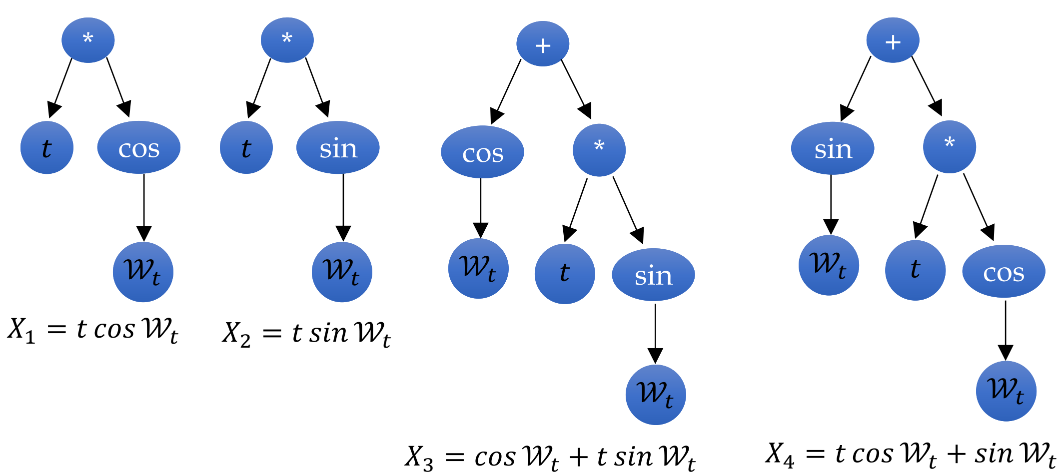

Figure 7: Parse tree for the solution of SDE system (5.7) by ACP method

The optimal solution and best ant tours generated by the ACP algorithm for the components of the SDE system (5.7) are plotted in Fig. 8.

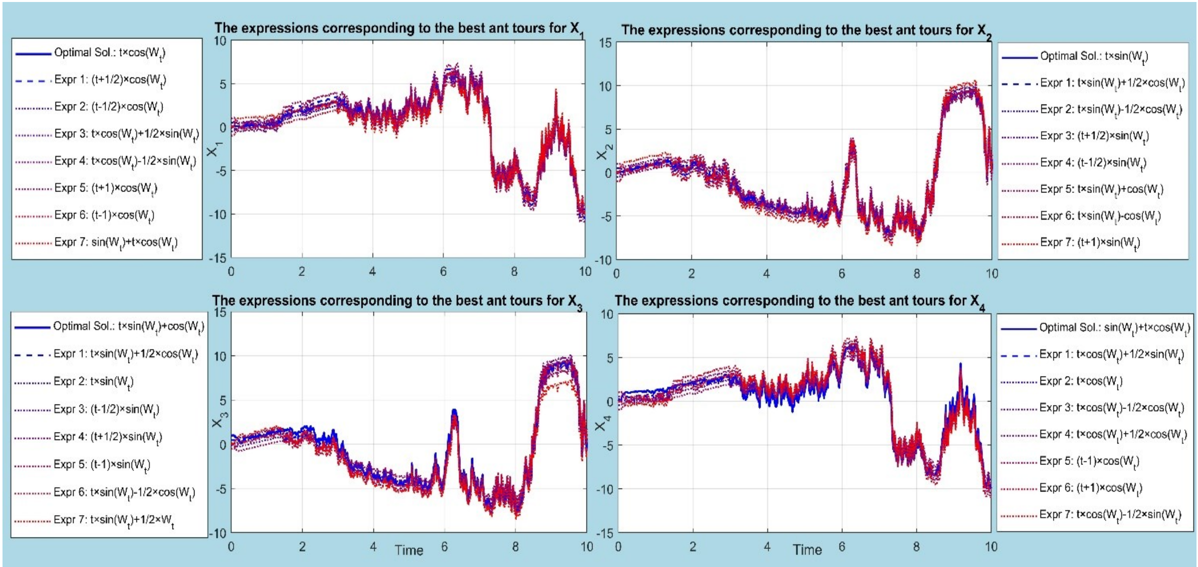

Figure 8: Exact solution, optimal solution and last 7 best ant tours for the SDE system (5.7)

6. Conclusions

This paper presented the ACP algorithm as a novel approach to solve SDE systems with respect to a 1-dimensional Wiener process. Simulation programs by MATLAB were construct to solve SDE systems by the ACP method, and the simulations yielded important results. Improvements to ACP algorithm enable it to produce and evaluate stochastic processes. More precisely, the ACP algorithm can be used to generate symbolic solutions to complex mathematical problems for which analytical solutions are difficult to obtain. The FF values for the optimal solutions of the studied SDE systems are equal to zero, which means that the optimal solutions are exact. Therefore, the ACP method is appropriate to solve systems of multidimensional stochastic differential equations. The symbolic mathematical expressions generated by the ACP algorithm are based on the functions chosen in the initialization step, so they are not limited to the type of SDE system, whether linear or nonlinear, nor to the type of stochastic process. As a promising tool, the ACP method may be of great interest in solving scientific and finance problems involving stochastic dynamics. As a future work, ACP and other optimization and evolutionary algorithms could be developed as automatic programming algorithms to produce symbolic mathematical expressions to address and solving various of mathematical problems, as appropriate to the nature of the problem to be solved.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] W.J. de A. Lobão, M.A.C. Pacheco, D.M. Dias, A.C.A. Abreu, Solving Stochastic Differential Equations Through Genetic Programming and Automatic Differentiation, Eng. Appl. Artif. Intell. 68 (2018), 110–120. https://doi.org/10.1016/j.engappai.2017.10.021.

[2] S. Lototsky, B. Rozovskii, Stochastic Differential Equations: A Wiener Chaos Approach, in: From Stochastic Calculus to Mathematical Finance, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006: pp. 433–506. https://doi.org/10.1007/978-3-540-30788-4_23.

[3] C.C. Chen, K. Yao, Stochastic-Calculus-Based Numerical Evaluation and Performance Analysis of Chaotic Communication Systems, IEEE Trans. Circuits Syst. I 47 (2000), 1663–1672. https://doi.org/10.1109/81.899918.

[4] L. Evans, An Introduction to Stochastic Differential Equations, American Mathematical Society, Providence, Rhode Island, 2013. https://doi.org/10.1090/mbk/082.

[5] S.D. Jacka, B. Oksendal, Stochastic Differential Equations: An Introduction with Applications, J. Amer. Stat. Assoc. 82 (1987), 948. https://doi.org/10.2307/2288814.

[6] M. Bayram, T. Partal, G. Orucova Buyukoz, Numerical Methods for Simulation of Stochastic Differential Equations, Adv. Differ. Equ. 2018 (2018), 17. https://doi.org/10.1186/s13662-018-1466-5.

[7] A.F. Fareed, M.S. Semary, H.N. Hassan, Two Semi-Analytical Approaches to Approximate the Solution of Stochastic Ordinary Differential Equations with Two Enormous Engineering Applications, Alexandria Eng. J. 61 (2022), 11935–11945. https://doi.org/10.1016/j.aej.2022.05.054.

[8] X. Mao, The Truncated Euler–Maruyama Method for Stochastic Differential Equations, J. Comput. Appl. Math. 290 (2015), 370–384. https://doi.org/10.1016/j.cam.2015.06.002.

[9] Z. Yin, S. Gan, An Improved Milstein Method for Stiff Stochastic Differential Equations, Adv. Differ. Equ. 2015 (2015), 369. https://doi.org/10.1186/s13662-015-0699-9.

[10] P. Wang, Three-stage stochastic Runge-Kutta methods for stochastic differential equations, J. Comput. Appl. Math. 222 (2008), 324–332. http://doi.org/10.1016/j.cam.2007.11.001.

[11] D.F. Anderson, D.J. Higham, Y. Sun, Multilevel Monte Carlo for Stochastic Differential Equations with Small Noise, SIAM J. Numer. Anal. 54 (2016), 505–529. https://doi.org/10.1137/15M1024664.

[12] A.R. Soheili, F. Soleymani, Iterative Methods for Nonlinear Systems Associated with Finite Difference Approach in Stochastic Differential Equations, Numer. Algor. 71 (2016), 89–102. https://doi.org/10.1007/s11075-015-9986-5.

[13] M. Boryczka, Z.J. Czech, Solving Approximation Problems by Ant Colony Programming, in: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, 2002.

[14] M. Boryczka and W. Wiezorek, Solving Approximation Problems Using Ant Colony Programming, in: Proceedings of AI-METH, pp. 55–60, 2003.

[15] N. Kumaresan, P. Balasubramaniam, Singular Optimal Control for Stochastic Linear Quadratic Singular System Using Ant Colony Programming, Int. J. Comput. Math. 87 (2010), 3311–3327. https://doi.org/10.1080/00207160903026634.

[16] M.Z.M. Kamali, N. Kumaresan, K. Ratnavelu, Solving Differential Equations with Ant Colony Programming, Appl. Math. Model. 39 (2015), 3150–3163. https://doi.org/10.1016/j.apm.2014.11.003.

[17] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer Berlin Heidelberg, 1992. https://doi.org/10.1007/978-3-662-12616-5.

[18] K. Burrage, P.M. Burrage, T. Tian, Numerical Methods for Strong Solutions of Stochastic Differential Equations: An Overview, Proc. R. Soc. Lond. A. 460 (2004), 373–402. https://doi.org/10.1098/rspa.2003.1247.

[19] E. Platen, N. Bruti-Liberati, Numerical Solution of Stochastic Differential Equations with Jumps in Finance, Springer, Berlin, Heidelberg, 2010. https://doi.org/10.1007/978-3-642-13694-8.

[20] M. Boryczka, Ant Colony Programming: Application of Ant Colony System to Function Approximation, in: R. Chiong (Ed.), Intelligent Systems for Automated Learning and Adaptation, IGI Global, 2010: pp. 248–272. https://doi.org/10.4018/978-1-60566-798-0.ch011.

[21] M. Boryczka, Ant Colony Programming for Approximation Problems, in: M.A. Kłopotek, S.T. Wierzchoń, M. Michalewicz (Eds.), Intelligent Information Systems 2002, Physica-Verlag HD, Heidelberg, 2002: pp. 147–156. https://doi.org/10.1007/978-3-7908-1777-5_15.

[22] Y. Boudouaoui, H. Habbi, C. Ozturk, D. Karaboga, Solving Differential Equations with Artificial Bee Colony Programming, Soft Comput. 24 (2020), 17991–18007. https://doi.org/10.1007/s00500-020-05051-y.