Solving Stochastic Differential Equation Systems by Particle Swarm Optimization Programming

Ali Sami Rashid[1] [0000-0003-1278-9443] Salah H Abid[2] [0000-0003-3617-6908] Sadiq A. Mehdi[2] [0000-0001-5244-7697]

[1] Mathematics Department, College of education, University of Misan, Misan, Iraq.
2 Mathematics Department, College of education, Mustansiriyah University, Baghdad, Iraq.
alisamirashid@uomisan.edu.iq

Abstract. Stochastic differential equations (SDE) represent dynamic systems subject to stochastic influences. This paper introduces particle swarm optimization programming (PSOP) as an innovative approach to solving SDE systems. The optimal solutions are reached by sending simulated particle to navigate in the search graph, and particle tours are saved and evaluated using a fitness function (*FF*). The PSOP has been developed as a programming algorithm in several directions, starting with construct the graph as a solution search space. the vision function is defined that depends on the simulated particle velocity in the swarm and the node position in the research graph, where the digital particle navigating in the research graph from the node to another according to the vision function. Finally, a fitness function is constructed to evaluate the expressions that represent possible solutions. The most important conclusion lies in obtaining symbolic solutions for the SDE systems studied. The PSOP method has been validated through simulations on multi-dimensional SDE systems, showing promising applications in stochastic processes, modeling, automatic programming, and artificial intelligence.

Keywords: Algorithms; Stochastic differential equation systems; Particle swarm optimization programming; Dynamical systems; Evolutionary algorithm.

1 Introduction

Stochastic differential equation (SDE) systems are dynamical systems that describe phenomena that include a random effect, whose effect may be internal to the system or external [1], [2]. SDE systems are of great importance in many fields, and are used to describe phenomena whose development is affected by random or noise factors over time, such as in finance, biology, engineering, physics, and many other fields [3]. The solution to the SDE represents a function of a stochastic process [4].

Previous studies have explored solutions and applications for some SDE. Research has focused on mathematical theories [5], practical applications [6], and numerical methods of SDE systems [7], [8], [9]. There are various numerical and analytical methods for solving a single SDE. Analytical methods are limited to solving some important examples of SDEs which are often based on the Itô lemma [10]. In terms of numerical methods for solving SDE, the Euler-Maruyama method [11], [12], [13], the Milstein method [14], the Monte Carlo method [15], [16], the Runge-Kutta method [17], and the finite difference methods [18], [19].

Particle swarm optimization (PSO) is a technique to solve optimization problems inspired by the collective behavior of molecules in nature. PSO has received wide attention from various researchers, due to simplicity of implementation and speed of convergence to acceptable solutions, and various aspects have been modified in the initial formulation [20]. PSO has been applied to address many problems such as function optimization, pattern classification, fuzzy control [21], [22], and artificial neural network training [23]. An important application of the PSO algorithm is the solution of systems of nonlinear algebraic equations [24]. Shi and Eberhart introduced the inertia weight parameter in the particle swarm algorithm, and simulations demonstrated its effective impact and importance in the particle swarm optimizer [25]. Because PSO has become a popular heuristic approach to optimization, Bratton and Kennedy designed the benchmark as an extension of the original algorithm and took into account developments to improve performance standard criteria [26].

Evolutionary algorithms are one of the automatic programming techniques, which works to enhance solutions and reduce errors over successive generations. Koza identified five basic steps that should be prepared to use an automatic programming approach to solve a problem [27], which are:

- Determining the terminals set,
- Determining the set of functions,
- Determining the fitness measure,
- Determining the controlling run parameters and
- Determining the terminal criteria for ending a run.

Most of these principles that are used in genetic programming (GP) and that Boryczka [28] used in ant colony programming (ACP) [29] will be adapted in particle swarm optimization programming (PSOP).

The PSOP will develop in several directions. First, construct a graph as a search space, define the vision function that depends on velocity and position of the particles in the graph, construct the particle tours according to the vision function, and finally construct a fitness function (FF) to evaluate the expressions representing possible solutions of the SDE system. Finally, the parse trees for optimal particle tours of PSOP solutions for SDE systems will be given.

The m-dimensional SDE system with respect to a 1-dimensional Wiener process $W = \{W_t, t \geq t_0\}$ is given by the form

$$dX(t) = f\big(t, X(t)\big)dt + g\big(t, X(t)\big)dW_t, \qquad t \in [t_0, T] \tag{1}$$

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

Where $X(t)$ is an m-dimensional stochastic process, the vector functions $f: [t_0, T] \times \mathbb{R}^m \to \mathbb{R}^m$ and $g: [t_0, T] \times \mathbb{R}^m \to \mathbb{R}^m$ are the drift and diffusion coefficients, respectively [30].

In matrix form, written as

$$\begin{bmatrix} dX_1(t) \\ dX_2(t) \\ \vdots \\ dX_m(t) \end{bmatrix} = \begin{bmatrix} f_1\big(t, X_1(t), \dots, X_m(t)\big) \\ f_2\big(t, X_1(t), \dots, X_m(t)\big) \\ \vdots \\ f_m\big(t, X_1(t), \dots, X_m(t)\big) \end{bmatrix} dt + \begin{bmatrix} g_1\big(t, X_1(t), \dots, X_m(t)\big) \\ g_2\big(t, X_1(t), \dots, X_m(t)\big) \\ \vdots \\ g_m\big(t, X_1(t), \dots, X_m(t)\big) \end{bmatrix} dW_t \quad (2)$$

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

The solution of SDE system (1) is

$$X(t) = X_0 + \int_0^t f\big(s, X(s)\big) ds + \int_0^t g\big(s, X(s)\big) dW(s) \quad (3)$$

In the system components form,

$$X_i(t) = X_i(0) + \int_0^t f_i\big(s, X_1(s), \dots, X_d(s)\big) ds + \int_0^t g_i\big(s, X_1(s), \dots, X_d(s)\big) dW_s \quad (4)$$

Where $\int_0^t f\big(s, X(s)\big) ds$ is Lebesgue integral, and $\int_0^t g\big(s, X(s)\big) dW_s$ is Itô's integral [31].

2 Particle swarm optimization programming (PSOP)

Particle swarm optimization (PSO) is a metaheuristic algorithm inspired by the particle swarm behavior of some natural models, such as birds, fish, and some insects [32]. The algorithm aims to find optimal solutions to various problems in mathematics based on the experience of each particle and the experience of the rest of the particles in the swarm. Such as finding optimal values for single variable or multivariable functions [33], [34].

The optimal solution to the SDE system is produced from the expressions resulting from particle tours through the graph. A dynamic graph $G = \{V, E\}$ is being constructed represents the search space. The nodes are represented by a set of functions and terminals. Functions are defined with $\sin, \cos, \log, \exp,$ and the arithmetic operations $+, -, /, *$. While the terminals are represented by the constants $0, 1, \dots, 9$ and variables. In principle, nodes are assigned random positions in the graph. Nodes are connected by edges E, which represent the distance between nodes. The particle velocity is $v_i = (v_{i1}, v_{i2}, \dots, v_{im})$, node position is $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ are where $x_{ij} \in [a, b]$ and m are the dimension of the SDE system.

Each particle in the swarm moves to several places and forms a mathematical expression during its tour of the graph. The mathematical expression produced by particle i consists of the names of nodes over which the particle passed. In the first generation, each particle starts from a random node and moves to the next node according to the vision function, and the tour of particle i ends until it reaches a terminal node. The expression for the particle is saved as P_i. After completing the tours of all particles in generation t, the parse tree for the particle tours is created, and mathematical expressions are generated. The expressions for each particle are evaluated using the fitness function FF. The best particle P_{best} and its corresponding tour that produces the minimum FF value are selected.

In PSOP, each particle will generate an arithmetic expression ϕ with a dimension equal to that of the SDE system. If ϕ satisfies FF equal or closed to zero, then ϕ will be the optimal solution. If the expression ϕ does not satisfy FF, the velocity v_i and position x_i will be updated, and regenerating the loop.

Sequential updating of the particle velocities and node positions during successive generation helps increase the particle velocity and reduce the distance between nodes on the best tours. The value of the vision function increases for the nodes that form an optimal solution and decreases for the other nodes, thus forming the optimal path for the particles. The flowchart of the PSOP algorithm is shown in Fig. 1.

Boryczka and Wiezorek identified four essential steps in the research process [29], choice of functions and terminals, graph construction, construction of the fitness function, and defining terminal criteria.

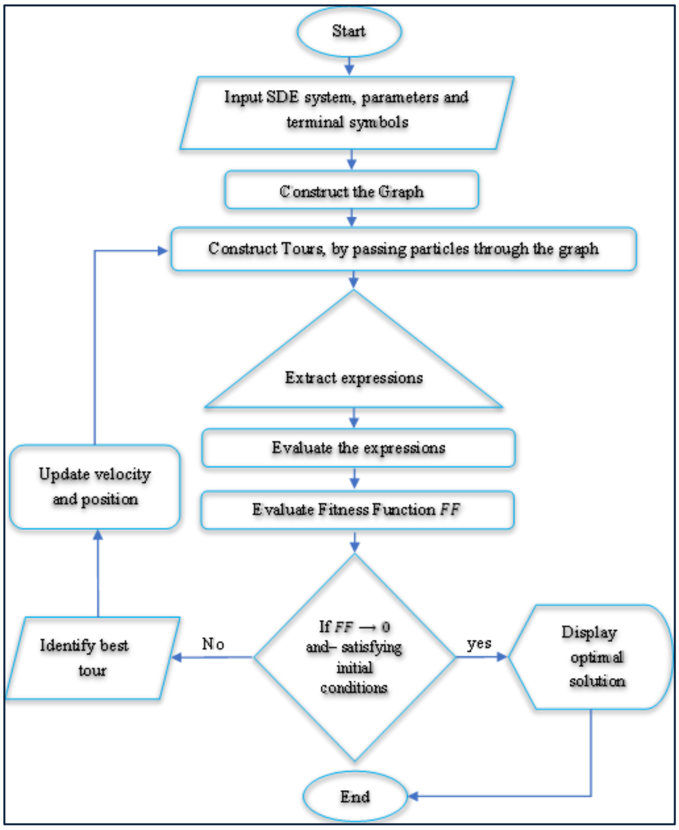

Fig. 1. PSOP Flowchart

2.1 Functions and terminals

The first important step in initializing the PSOP algorithm is the selection of functions and terminals, depending on the problem under study.

In the PSOP approach, the terminal symbols include variables $\{t, x, y, \ ...\}$, constants $\{0, 1, ..., 9, \pi, e\}$, 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.

In the SDE system formula (1), which includes dt and dW_t in the deterministic and stochastic parts, we note that the system variables are time t and Wiener process W_t, so will be included as variables in the terminal symbols. The functions and terminals are chosen as in Table 1.

Table 1. The terminal symbols and functions.

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

2.2 Construction of graph

In PSOP, the graph represents the search space, provides structure for particles to navigate, and search for solutions. A graph G consists of nodes and edges. Each node indicates a terminal symbol, which represents a function $f_i \in F$ or terminal $t_i \in T$. The edges in G connect the nodes and are weighted by the distance between the node positions. An illustrative graph in Fig. 2.

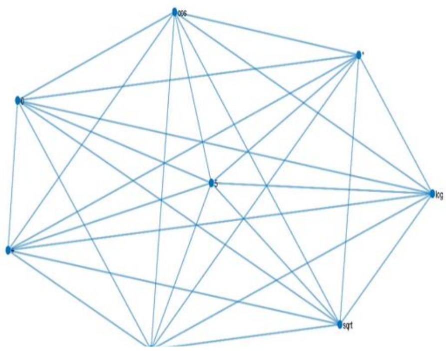

Fig. 2. An illustrative PSOP graph

2.3 Fitness function

The fitness function (FF) is the main focus of the PSOP algorithm, and its construction depends on the nature of the problem to be solved.

In the PSOP approach, FF is a norm used to evaluate the convergence or fitness of a solution generated by particle tours. In our study, the objective of FF is to filter out the best solution among the available solutions and to determine the fitness value of each solution and guide the search process towards finding the optimal solution. If a particle produces a stochastic process $\phi\big(t, X(t)\big)$, the Itô-formula for the component s of $\phi\big(t, X(t)\big)$ takes the form

$$d\phi_s = \hat{f}_s(t, X_1, \dots, X_m)dt + \sum_{r=1}^{m} \hat{g}_{sr}(t, X_r)dW_r(t) \tag{5}$$

The fitness function is defined by the form

$$FF = \sum_{s=1}^{m} \left(\hat{f}_s(t, X_1, \ldots, X_m) - f_s(t, X_1, \ldots, X_m)\right)^2 + \sum_{r=1}^{m} \left(\hat{g}_{sr}(t, X_r) - g_{sr}(t, X_r)\right)^2 \quad (6)$$

2.4 Terminal criteria

The terminal criteria refer to the conditions that determine when the PSOP algorithm stops searching for solutions. In each generation, a number of particles are sent to traverse the graph. If a particle finds an expression that gives a fitness function close to zero and satisfies the initial conditions, the program stops. Otherwise, velocity and position updating rules are applied and the process is repeated until a fitness function value equal to or close to zero is obtained.

3 PSOP methodology

To solve an SDE system using the PSOP algorithm, the initialization starts with:

- Choose the appropriate functions and terminals for the problem.
- Choose the number of nodes, and construct the graph.
- Choose the number of simulated particles.
- Determine the maximum number of generations.
- Define the values of the parameter.

Simulated particles are sent to search for available solutions of a given SDE system. The particle navigates through the graph $G(V, E)$, where the nodes V represent the functions f_i and terminal symbols t_i, and E is the set of edges connecting nodes, weighted by the distance between nodes. Each particle k moves from node i to node j on the graph G at time t according to the vision law:

$$\Upsilon_{i,sr}(t) = \frac{v_i}{d_{sr}} \quad (7)$$

Where v_i is the velocity of particle i and d_{sr} is the distance between positions s and r.

When each generation is finished and the particle tours are saved, the parse tree is performed for each tour. after that, the mathematical expressions are generated, evaluate the expressions, and exclude unwanted ones. For example, if a particle produces $log(2 * W_t)/t$ and another particle produce $W_t/) * e\hat{}t$, if we fixed the values of t and W_t are arbitrary, the first expression is evaluable while the second cannot. Table 3 illustrates the expressions corresponding for 6 particle tours and the possibility of their evaluation. Only evaluable mathematical expressions will be directed to substituted into FF. If the value of FF is equal to or close to zero and satisfies the initial conditions of the SDE system, then the generation of tours stops. Otherwise, perform an update of the velocity of particles and position of nodes on the graph according to the following laws:

$$v_i(t+1) = \theta v_i(t) + cr(FF_{best} - FF_i) \quad (8)$$

$$x_j(t+1) = \theta x_j(t) + (1-\theta) v_i(t+1) \quad (9)$$

Where c is a positive integer 2, r is a random value distributed uniformly in $[0, 1]$, t is the generation number, FF_{best} and FF_i are the fitness function values at the best particle and particle i, respectively. The parameter θ is an inertia weight that balanced the global exploration and local exploitation of the swarm [35], and descends linearly from 0.9 to 0.4.

After updating the velocity and position and determining the best tour in the previous generation, send the same particles back through the best tour.

Table 2. Tours and expressions.

Particle Tours	Expressions	Status
$\exp(W_t) * \sin)t$	$e^{W_t} * \sin)t$	Not evaluable
$\exp(W_t/sqrt\ 3) + \sin W_t$	$e^{\frac{W_t}{\sqrt{3}}} + \sin W_t$	Evaluable
$\cos(W_t * 3 * t/4)$	$\cos\left(\frac{3tW_t}{4}\right)$	Evaluable
$\sin +(5/W_t$	$\sin +(\frac{5}{W_t}$	Not evaluable
$\log\exp(W_t)$	W_t	Evaluable
$\exp(W_t/t) * 7 * t\char`^2$	$7t^2e^{\frac{W_t}{t}}$	Evaluable

The PSOP algorithm

Step 1. Start with input the SDE system, parameters, and terminal symbols.

Step 2. Construct the graph.

Step 3. As a starting point, set equal velocities for the particles and random positions for the nodes on the graph.

Step 4. Construct tours, particles moving to the next node according to the vision law (7).

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

Step 6. Extract 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, and identify FF_{best}

Step 9. Check, if FF close to zero and satisfying initial conditions; stop and go to Step 13.

Step 10. Otherwise, identify the best tour.

Step 11. Perform an update of the velocity and position by applying laws (8), (9).

Step 12. Passing the same particles through the best tour and go back to Step 4.

Step 13. Display the solution. End.

4 Simulation results of PSOP

In this work, a PSOP algorithm will be designed to simulate SDE systems with 1, 2, 3 and 4 dimensions. The parse trees generated by the PSOP algorithm for the optimal particle tours will be given, as well as the optimal solution with the corresponding FF value. Finally, the PSOP solutions will be proven to satisfy the SDE systems studied.

4.1 1-dimensional SDE

First example

. Consider a homogeneous SDE with multiplicative noise $\mathrm{W_t}$

With initial condition $X(0) = 1$, and $t \in [0,1]$.

$$dX_t = X_t dt + X_t\, dW_t \tag{10}$$

The optimal particle tour $1 * exp\ (1/2 * t + W_t)$ with $FF = 0$ is reached at generation 43, and the parse tree for the optimal tour is given in Fig. 3.

The PSOP algorithm produced the following optimal solution for the SDE (10):

$$X_t = \exp\left(\frac{1}{2}t + W_t\right) \tag{11}$$

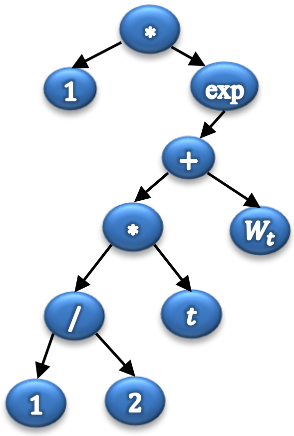

Fig. 3. Parse tree for the solution of SDE (10) by PSOP method

According to the Itô formula, the stochastic process (11) is an Itô process, and X_t satisfy

$$dX_t = \frac{1}{2}\exp\left(\frac{1}{2}t + W_t\right)dt + \exp\left(\frac{1}{2}t + W_t\right)dW_t + \frac{1}{2}\exp\left(\frac{1}{2}t + W_t\right)dW_t^2$$

where $dW_t^2 = dt$ then we get

$$dX_t = \exp\left(\frac{1}{2}t + W_t\right)dt + \exp\left(\frac{1}{2}t + W_t\right)dW_t$$

Thus, the stochastic process (11), is the solution of the SDE (10), and satisfy the initial conditions $X_t(0) = 1$.

Second example

Consider an SDE with respect to a 1-dimensional Wiener process $\mathrm{W_t}$

$$dX_t = -X_t(2\ln X_t + 1)dt - 2X_t\sqrt{-\ln X_t}\ dW_t \tag{12}$$

With initial condition $X(0) = 1$.

The optimal particle tour $exp\ (0 - (W_t - ln\ 1)$^2) with $FF = 0$ is reached at generation 52, and the parse tree for the optimal tour is given in Fig. 4.

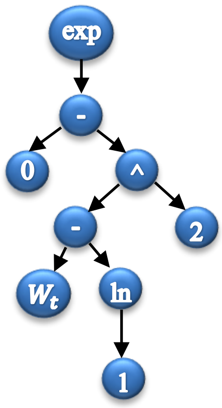

Fig. 4. Parse tree for the solution of SDE (12) by PSOP method

The PSOP algorithm produced the following optimal solution for the SDE (12):

$$X_t = \exp(-W_t^2) \tag{13}$$

According to the Itô formula, the stochastic process (13) is an Itô process, and X_t satisfy

$$dX_t = -2W_t \exp(-W_t^2)\, dW_t + \frac{1}{2}[(-2W_t)(-2W_t)\exp(-W_t^2) - 2\exp(-W_t^2)]dW_t^2$$

By simplifying the last equation, we obtain:

$$dX_t = -(-2W_t^2 + 1)\exp(-W_t^2)\, dt - 2W_t \exp(-W_t^2)\, dW_t$$

Therefore, the stochastic process (13) satisfies SDE (12) and the initial condition $X(0) = 1$.

Third example

Consider an SDE with respect to a 1-dimensional Wiener process W_t

$$dX_t = \frac{1}{3}\sqrt[3]{X_t}dt + \sqrt[3]{X_t^2}\, dW_t \tag{14}$$

With initial condition $X(0) = 1$.

The optimal particle tour $ln\ (exp(1 + \frac{1}{3}W_t)\ \hat{}\ 3)$ with $FF = 0$ is reached at generation 47, and the parse tree for the optimal tour is given in Fig. 5.

The PSOP algorithm produced the following optimal solution for the SDE (14):

$$X_t = \left(1 + \frac{1}{3}W_t\right)^3 \tag{15}$$

Applying the Itô formula to the stochastic process (15), we get

$$dX_t = \frac{1}{3}\left(1 + \frac{1}{3}W_t\right)dt + \left(1 + \frac{1}{3}W_t\right)^2 dW_t$$

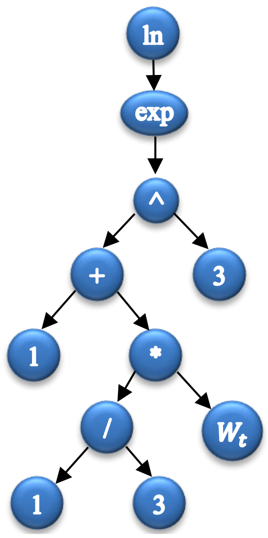

Fig. 5. Parse tree for the solution of SDE (14) by PSOP method

Hence, the stochastic process (15) satisfies SDE (14) and the initial condition $X(0) = 1$.

Therefore, Equation (15) is the exact solution of SDE (14).

4.2 2-dimensional SDE system

Consider the 2-dimensional SDE system with respect to a 1-dimensional Wiener process W_t,

$$\begin{aligned} dX_1(t) &= \frac{-1}{2} X_1(t)dt - X_2(t)dW_t \\ dX_2(t) &= \frac{-1}{2} X_2(t)dt + X_1(t)dW_t \end{aligned} \tag{16}$$

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

The optimal particle tours $cos(W_t)$, $sin(W_t)$ are reached at generation 57, and the parse tree for the optimal tours are given in Fig. 6.

The PSOP algorithm produced the following optimal solution for the SDE (16):

$$X = \begin{bmatrix} \cos W_t \\ \sin W_t \end{bmatrix} \tag{17}$$

According to the Itô formula, the stochastic process (17) is an Itô process, and the components X_1 , X_2 satisfy

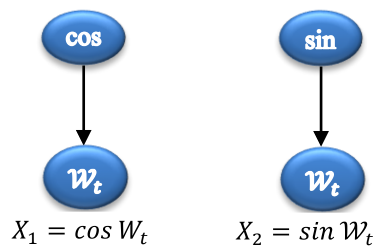

Fig. 6. Parse tree for the solution of SDE system (16) by PSOP method

$$dX_1(t) = -\tfrac{1}{2}\cos W_t\, dt \; - \sin W_t\, dW_t$$
$$dX_2(t) = -\tfrac{1}{2}\sin W_t\, dt + \cos W_t\, dW_t$$

Thus, the stochastic process $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$, is the solution of the SDE system (16), and satisfy the initial conditions $X_1(0) = 1$ and $X_2(0) = 0$.

4.3 3-dimensional SDE system

Consider the 3-dimensional SDE system with respect to a 1-dimensional Wiener process W_t,

$$\begin{aligned} dX_1(t) &= \left(\frac{2-t}{2t}\right) X_1(t)dt - X_2(t)dW_t \\ dX_2(t) &= \left(\frac{2-t}{2t}\right) X_2(t)dt + X_1(t)dW_t \\ dX_3(t) &= \left(\frac{1}{t}X_2(t) - \frac{1}{2}X_3(t)\right)dt + \left(X_1(t) - \frac{1}{t}X_2(t)\right)dW_t \end{aligned} \tag{18}$$

With initial conditions $X_1(0) = 0$, $X_2(0) = 0$ and $X_3(0) = 1$.

The optimal particle tours $t * cos(W_t)$, $t * sin(W_t)$, $cos(W_t) + t * sin(W_t)$ is reached at generation 61, and the parse tree for the optimal tours is given in Fig. 7.

The PSOP algorithm produced the following optimal solution for the SDE system (18):

$$X(t) = \begin{bmatrix} t \cos W_t \\ t \sin W_t \\ \cos W_t + t \sin W_t \end{bmatrix} \quad (19)$$

By the Itô formula, the stochastic process (19) is an Itô process. The components X_1, X_2 and X_3 satisfy

$$dX_1(t) = \left(\frac{2-t}{2t}\right) t \cos W_t \, dt - t \sin W_t \, dW_t$$

$$dX_2(t) = \left(\frac{2-t}{2}\right) \sin W_t \, dt + t \cos W_t \, dW_t$$

$$dX_3(t) = \left(\sin W_t - \frac{1}{2}\cos W_t - \frac{1}{2} t \sin W_t\right) dt + (t \cos W_t - \sin W_t) dW_t$$

Hence, the stochastic process (19) is the solution of the SDE system (18) and satisfies the initial conditions.

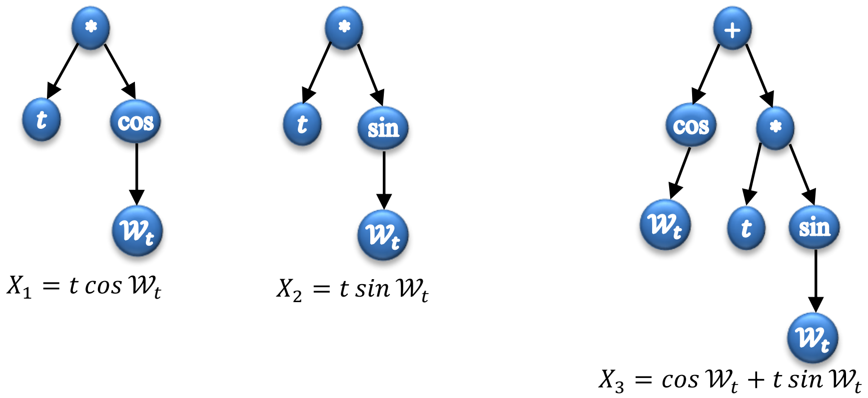

Fig. 7. Parse tree for the solution of SDE system (18) by PSOP method

4.4 4-dimensional SDE system

Consider the 4-dimensional SDE system with respect to a 1-dimensional Wiener process W_t,

$$\begin{aligned} dX_1 &= \left(X_3(t) - X_2(t) - \frac{1}{2}X_1(t)\right) dt + t\big(X_1(t) - X_4(t)\big) dW_t \\ dX_2 &= \left(X_4(t) - X_1(t) - \frac{1}{2}X_2(t)\right) dt + X_1(t) dW_t \\ dX_3 &= \left(\frac{1}{t}X_2(t) - \frac{1}{2}X_3(t)\right) dt + \left(X_4(t) - \frac{2}{t}X_2(t)\right) dW_t \\ dX_4 &= \left(\frac{1}{t}X_1(t) - \frac{1}{2}X_4(t)\right) dt + \left(\frac{2}{t}X_1(t) - X_3(t)\right) dW_t \end{aligned} \quad (20)$$

With initial conditions $X_1(0) = 0$, $X_2(0) = 0$, $X_3(0) = 1$ and $X_4(0) = 0$.
The optimal particle tours $t * cos(W_t)$, $t * sin(W_t)$, $cos(W_t) + t * sin(W_t)$, $sin(W_t) + t * cos(W_t)$ is reached at generation 61, and the parse tree for the optimal tours is given in Fig. 8.

The PSOP solution of the SDE system (20) is given by

$$X(t) = \begin{bmatrix} t\ \cos W_t \\ t\ \sin W_t \\ \cos W_t + t\ \sin W_t \\ \text{t} \cos W_t + \sin W_t \end{bmatrix} \tag{21}$$

By the Itô formula, the stochastic process (21) is an Itô process, and X_1 , X_2 , X_3, X_4 satisfy

$$dX_1(t) = \left(\cos W_t - \frac{1}{2} t \cos W_t \right) dt - t \sin W_t \, dW_t$$

$$dX_2(t) = \left(\sin W_t - \frac{1}{2} t \sin W_t \right) dt + t \cos W_t \, dW_t$$

$$dX_3(t) = \left(\sin W_t - \frac{1}{2} \cos W_t - \frac{1}{2} t \sin W_t \right) dt + (t \cos W_t - \sin W_t) dW_t$$

$$dX_4(t) = \left(\cos W_t - \frac{1}{2} (t \cos W_t + \sin W_t) \right) dt + (\cos W_t - t \sin W_t) dW_t$$

Therefore, the stochastic process (21) is a solution of the SDE system (20), and it satisfies the initial conditions.

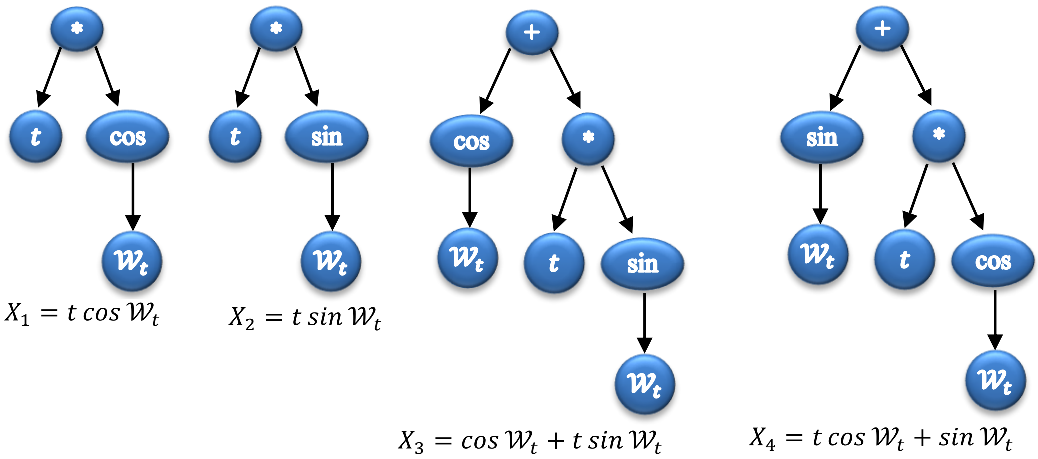

Fig. 8. Parse tree for the solution of SDE system (20) by PSOP method

5 Conclusions

This paper presents the PSOP method as an innovative computational approach to solving SDE systems, which has been applied using simulation and yielded significant results. Improvements to PSOP as an automatic programming algorithm have enabled to produce symbolic solutions for SDE systems. 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 PSOP method is appropriate for solving systems of multidimensional stochastic differential equations with respect to a 1-dimensional Wiener process.

The particle swarm algorithm can be developed as an automatic programming algorithm to produce and evaluate symbolic stochastic processes. More precisely, the PSOP algorithm can be used to generate symbolic solutions to various complex mathematical problems for which it is difficult to obtain analytical solutions that are appropriate to the nature of the problem to be addressed.

What distinguishes the PSOP method is that it generates symbolic mathematical expressions according to the functions that are chosen in the initialization step, so it is not limited to the type of SDE system, whether linear or non-linear, nor by the type of stochastic process. As future work, the PSOP algorithm can be developed to solve SDE systems with respect to an m-dimensional Wiener process.

Acknowledgments. The authors would like to thank the anonymous reviewers for their valuable comments.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. de Araujo Lobão, W. J., Pacheco, M. A. C., Dias, D. M., Abreu, A. C. A.: Solving Stochastic Differential Equations Through Genetic Programming and Automatic Differentiation. Engineering Applications of Artificial Intelligence **68**, 110–120 (2018).
2. Herdiana, R.: Numerical Simulation of Stochastic Differential Equations Using Implicit Milstein Method. Journal of Fundamental Mathematics and Applications **3**(1), 72–83 (2020)
3. Liu, Q., Khadidos, A. O., Wan, P.: Discretization Processing of Financial Risk Management Using Stochastic Differential Equation Simulation Method. Fractals **30**(2), 2240069 (2022)
4. Geiser, J.: Numerical Picard Iteration Methods for Simulation of Non-Lipschitz Stochastic Differential Equations. Symmetry **12**(3), 383–405 (2020)
5. Xia, M., Liu, L., Fang, J., Zhang, Y.: Stability analysis for a class of stochastic differential equations with impulses. Mathematics, **11**(6), 1541 (2023)
6. Abonongo, J., Chidzalo, P.: Synthesis Weibull Stochastic Differential Equation: Properties and application. Scientific African, **23**, e02074 (2024)
7. Armstrong, J., Brigo, D., Rossi Ferrucci, E.: Optimal Approximation of SDEs on Submanifolds: The Itô-vector and Itô-jet Projections. In: Proceedings of the London Mathematical Society, 176–213. London (2019)
8. Kruse R., Wu, Y.: A Randomized Milstein Method for Stochastic Differential Equations with Non-Differentiable Drift Coefficients. Discrete Continuous Dynamical Systems-B **24**(8), 3475–3502 (2019)
9. Allen, E.: Modeling with Itô Stochastic Differential Equations. Springer Science & Business Media (2007)

10. Kloeden, P. E., Platen, E.: Numerical Solution of Stochastic Differential Equations. Springer Berlin Heidelberg (1992)
11. Mao, X.: The Truncated Euler-Maruyama Method for Stochastic Differential Equations. Journal of Computational and Applied Mathematics **290**, 370–384 (2015)
12. Ngo, H. L., Taguchi, D.: Strong Rate of Convergence for The Euler-Maruyama Approximation of Stochastic Differential Equations with Irregular Coefficients. Mathematics of Computation, **85**(300), 1793–1819 (2015)
13. Neuenkirch, A., Szölgyenyi, M., Szpruch, L.: An Adaptive Euler-Maruyama Scheme for Stochastic Differential Equations with Discontinuous Drift and Its Convergence Analysis. SIAM Journal on Numerical Analysis **57**(1), 378–403 (2019)
14. Milstein, G. N.: Numerical Integration of Stochastic Differential Equations. Springer Science & Business Media (2013)
15. Anderson, D. F., Higham, D. J., Sun, Y.: Multilevel Monte Carlo for Stochastic Differential Equations with Small Noise. SIAM Journal on Numerical Analysis **54**(2), 505–529 (2016)
16. Müller, E. H., Scheichl, R., Shardlow, T.: Improving multilevel Monte Carlo for Stochastic Differential Equations with Application to the Langevin Equation. In: Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, pp. 1–20. (2015)
17. Wang, P.: Three-Stage Stochastic Runge-Kutta Methods for Stochastic Differential Equations. Journal of Computational and Applied Mathematics **222**(2), 324–332 (2008)
18. Dareiotis, K., Leahy, J. M.: Finite Difference Schemes for Linear Stochastic Integro-Differential Equations. Stochastic Processes and their applications **126**(10), 3202–3234 (2016)
19. Baccouch, M., Temimi, H., Ben-Romdhane, M.: A Discontinuous Galerkin Method for Systems of Stochastic Differential Equations with Applications to Population Biology, Finance, And Physics. Journal of Computational and Applied Mathematics **388**, 1–26 (2021)
20. Taherkhani, M., Safabakhsh, R.: A Novel Stability-Based Adaptive Inertia Weight for Particle Swarm Optimization. Applied Soft Computing **38**, 281–295 (2016)
21. Tian, N., Huang, S., He, L., Du, L., Yang, S., Huang, B.: Research on The Mill Feeding System of An Elastic Variable Universe Fuzzy Control Based on Particle Swarm Optimization Algorithm. Physicochemical Problems of Mineral Processing **59**(3), 1–13 (2023)
22. Tran, T. A.: The Optimization of Marine Diesel Engine Rotational Speed Control Process by Fuzzy Logic Control Based on Particle Swarm Optimization Algorithm. Future Internet, **10**(10), 99 (2018)
23. Bas, E., Egrioglu, E., Kolemen, E.: Training Simple Recurrent Deep Artificial Neural Network for Forecasting Using Particle Swarm Optimization. Granular Computing **7**(2), 411–420 (2022)
24. Jaberipour, M., Khorram, E., Karimi, B.: Particle swarm algorithm for solving systems of nonlinear equations. Computers & Mathematics with Applications **62**(2), 566-576 (2011)
25. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence, pp. 69-73. IEEE (1998)
26. Bratton, D., Kennedy, J.: Defining A Standard for Particle Swarm Optimization. In: 2007 IEEE swarm intelligence symposium, pp. 120–127. IEEE (2007)
27. Koza, J. R.: Survey of Genetic Algorithms and Genetic Programming. In: Wescon conference record, pp. 589–594. Western Periodicals Company (1995)
28. Boryczka, M., Czech, Z. J: Solving Approximation Problems by Ant Colony Programming. In: Late Breaking Papers at the Genetic and Evolutionary Computation Conference, pp. 39–46. Morgan Kaufmann Publishers, New York (2002)
29. Kamali, M. Z. M., Kumaresan, N., Ratnavelu, K.: Solving differential equations with ant colony programming. Applied Mathematical Modelling **39**(10-11), 3150-3163 (2015)

30. Kloeden, P. E., Platen, E., Schurz, H.: Numerical solution of SDE through computer experiments. Springer Science & Business Media, (2012)
31. Platen, E., Bruti-Liberati, N.: Numerical Solution of Stochastic Differential Equations with Jumps in Finance. Springer Science & Business Media, 2010.
32. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of international conference on neural networks, pp. 1942–1948. IEEE (1995)
33. Kennedy, J.: The Particle Swarm: Social Adaptation of Knowledge. In: Proceedings of 1997 IEEE International Conference on Evolutionary Computation, pp. 303–308. IEEE (1997)
34. Eberhart, R. C., Shi, Y., Kennedy, J.: Swarm Intelligence. Elsevier, 2001.
35. Hong, L., Yu, X., Tao, G., Özcan, E., Woodward, J.: A sequential quadratic programming based strategy for particle swarm optimization on single-objective numerical optimization. Complex & Intelligent Systems, **10**(2), 2421-2443 (2024)