Solution of the Fuzzy Schrödinger Equation in Positron-Hydrogen Scattering Using Ant Colony Programming

N. Kumaresan,1,* M. Z. M. Kamali,2,† and Kuru Ratnavelu1
1Institute of Mathematical Sciences, Faculty of Science, University of Malaya, Kuala Lumpur 50603, Malaysia
2Centre for Foundation Studies in Science, University of Malaya, Kuala Lumpur 50603, Malaysia
(Received February 12, 2013; Revised May 13, 2015)

In this paper, a solution of the Takagi-Sugeno (T-S) fuzzy Schrödinger equation with linear potential in positron-hydrogen scattering is obtained using ant colony programming (ACP). The ACP solution is equivalent or very close to the exact solution of the problem. Accuracy of the solution computed by the ACP approach to the problem is qualitatively better. The solution of this novel method is compared with the traditional Runge Kutta (RK) method and Numerov’s method. An illustrative numerical example is presented for the proposed method.

DOI: PACS numbers: 34.80.Uv

I. INTRODUCTION

The Schrödinger equation is the fundamental equation of physics for describing non-relativistic quantum mechanical behavior. The exact solution of the one dimensional time independent Schrödinger equation for some quantum mechanical problems is not possible in complicated situations. But it can be solved analytically for a very limited number of potentials, such as the Coulomb, harmonic oscillator, and Poeschl-Teller potentials. It is also possible to find many works on quasi exact solvable cases [1–9]. Generally investigations in this area have been focused on finding some approximation methods, such as the Wentzel, Kramers, and Brillouin (WKB) [10], Hill determinant [11] super symmetric quantum mechanics [12], shifted 1/N expansion method, and some others [13, 14]. An asymptotic iteration method has been proposed for solving the second order linear differential equations and the Schrödinger equation [15–20]. In this work, the ACP approach is proposed to solve the Schrödinger equation.

Ant colony programming (ACP) is a metaheuristic approach that is inspired by the behaviour of real ant colonies, to find a good enough solution to the given problem in a reasonable amount of computation time. It allows the programmer to avoid the tedious task of creating a program to solve a well-defined problem [21]. ACP is a stochastic search technique that is carried out on a space graph where the nodes represent functions, vari-

*Electronic address: drnk2008@gmail.com, drnk2008@um.edu.my
†Electronic address: mzmk2000@yahoo.com
ables and constants. Functions are usually defined mathematically in terms of arithmetic operators, operands, and boolean functions. The set of functions defining a given problem is called a function set $P$ and the collection of variables and constants to be used are known as the terminal set $T$.

Ants are able to find their way efficiently from their nest to food sources. While searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the ant deposits a chemical pheromone trial on the ground. The quantity of pheromone deposited which may depend on the quantity and quality of the food, will guide other ants to the food source. If an ant has a choice of trails to follow, the preferred route is the trail with the highest deposit of pheromone [22]. This behaviour helps the ants to find the optimal route without any need for direct communication or central control. Therefore the artificial ants used in the ACP have some features taken from the behaviour of real ants, for example,

(a) Artificial ants move in a random fashion.
(b) Choice of a route of an artificial ant depends on the amount of pheromone.
(c) Artificial ants co-operate in order to achieve the best result.

A fuzzy system consists of linguistic IF-THEN rules that have fuzzy antecedent and consequent parts. It is a static nonlinear mapping from the input space to the output space. The inputs and outputs are crisp real numbers and not fuzzy sets. The fuzzification block converts the crisp inputs to fuzzy sets and then the inference mechanism uses the fuzzy rules to produce fuzzy conclusions or fuzzy aggregations and finally the defuzzification block converts these fuzzy conclusions into crisp outputs. The fuzzy system with singleton fuzzifier, product inference engine, center average defuzzifier and Gaussian membership functions is called the standard fuzzy system [23].

Two main advantages of fuzzy systems for the control and modelling applications are
(i) uncertain or approximate reasoning, especially difficult to express a mathematical model,
(ii) decision making problems with the estimated values under incomplete or uncertain information [24, 25].

Although parallel algorithms can compute the solutions faster than sequential algorithms, there have been no report on ant colony programming solutions for matrix Riccati differential equations [26, 27]. This paper focuses upon the implementation of an ant colony programming approach for solving the Schrödinger equation in order to get the best solution.

This paper is organized as follows. In Section II, the statement of the problem is given. In Section III, the solution of the Schrödinger equation is presented. In Section IV, a numerical example is discussed to illustrate the proposed method. The final conclusion section demonstrates the efficiency of the method.
II. STATEMENT OF THE PROBLEM

Consider the Takagi-Sugeno fuzzy Schrödinger equation (i.e., $V_i(r) = r$) that can be expressed in the form:

$$R_i : \text{If } F_j \text{ is } T_{ji}(m_{ji}, \sigma_{ji}), i = 1, \ldots, r \text{ and } j = 1, \ldots, n, \text{ then}$$

$$\ddot{F}(r) + V_i(r) F(r) - \frac{L_i(L_i + 1)}{r^2} F(r) = -k_i^2 F(r), \quad F(r) = F_r, \quad \dot{F}(r) = \dot{F}_r,$$

where $R_i$ denotes the $i$th rule of the fuzzy model, $m_{ji}$ and $\sigma_{ji}$ are the mean and standard deviation of the Gaussian membership function, $F(r)$ is the function of the scattered positron, $V_i(r)$ is the wave function of the atomic target, $k_i^2$ is the energy and $L_i$ is the angular momentum.

III. SOLUTION OF SCHRODINGER EQUATION

III-1. Runge Kutta method

Numerical integration is one of the oldest and most fascinating topics in numerical analysis. It is the process of producing a numerical value for the integration of a function over a set. Numerical integration is usually utilized when analytic techniques fail. Even if the indefinite integral of the function is available in a closed form, it may involve some special functions, which cannot be computed easily. In such cases also we can use numerical integration. RK algorithms have always been considered as the best tool for the numerical integration of ordinary differential equations (ODEs). The Schrödinger equation can be written in the following form:

$$\ddot{F}(r) = f(r, F(r)), \quad F(r) = F_r, \quad \dot{F}(r) = \dot{F}_r.$$

Hence the formula for RK method to solve (1) is given below:

$$F(r + h) = F(r) + h \times [\dot{F}(r) + \frac{1}{6}(k1 + k2 + k3)]$$

$$\dot{F}(r + h) = \dot{F}(r) + \frac{1}{6}(k1 + 2k2 + 2k3 + k4),$$

where

$$k1 = h \times f \left( r, F(r), \dot{F}(r) \right)$$

$$k2 = h \times f \left( r + \frac{h}{2}, F(r) + \frac{h}{2} \dot{F}(r) + \frac{h}{8}k1, \dot{F}(r) + \frac{1}{2}k2 \right)$$

$$k3 = h \times f \left( r + \frac{h}{2}, F(r) + \frac{h}{2} \dot{F}(r) + \frac{h}{8}k1, \dot{F}(r) + \frac{1}{2}k2 \right)$$

$$k4 = h \times f \left( r + h, F(r) + h \dot{F}(r) + \frac{h}{2}k3, \dot{F}(r) + k3 \right)$$
III-2. Numerov’s method

Numerov’s method [28] is an efficient algorithm for solving second order differential equations of the form

$$\ddot{y} = U(x) + V(x)y,$$  \hspace{1cm} (3)

Particular examples in physics of this type of equation are the one dimensional time independent Schrödinger equation,

$$\ddot{y} = \frac{2m}{\hbar} [V(x) - E] \psi y,$$  \hspace{1cm} (4)

and the equation of motion of an undamped forced harmonic oscillator,

$$m\ddot{y} = f_0 \cos(\omega x) - ky.$$  \hspace{1cm} (5)

In addition, Poisson’s equation may reduce to this form when the charge distribution is sufficiently symmetrical.

The important features of Eq. (1) for the application of Numerov’s method are that the first derivative is absent and the left hand side (LHS) is linear in \( y \). To obtain a finite difference scheme, one uses the centered difference equation,

$$y_{n+1} - 2y_n + y_{n-1} \approx 2\left( \frac{h^2}{2} \ddot{y} + \frac{h^4}{4!} + O(h^6) \right)$$  \hspace{1cm} (6)

where \( y_n = y(x_n) \) and we suppose that the \( x_n \) are uniformly spaced with a separation of \( h \). If we denote the LHS of Eq. (1) by

$$F = U(x) + V(x)y,$$  \hspace{1cm} (7)

then, by combining Eqs. (1) and (4), we have

$$y_{n+1} = 2y_n - y_{n-1} + h^2 F_n + \frac{h^4}{12} \ddot{F} + O(h^6).$$  \hspace{1cm} (8)

However, we can replace the second derivative of \( F \) by a difference equation similar to Eq. (4), giving the final result for Numerov’s algorithm,

$$y_{n+1} = \frac{2y_n - y_{n-1} + \frac{h^2}{12} (U_{n+1} + 10F_n + F_{n-1})}{1 - \frac{\nu_{n+1}h^2}{12}} + O(h^6).$$  \hspace{1cm} (9)

It is in this step that we require the LHS of Eq. (1) to be linear in \( y \).

III-3. Ant colony programming method

In this approach, ACP is used to obtain a set of expressions. If the required number expressions satisfy the fitness function, it will be the optimal solution of (3). The scheme of computing optimal solution is given in Figure 1. According to Boryczka and Wiezorek [21], the following four preparatory steps are essential for a searching process.
Choice of terminals and functions

- Construction of graph
- Defining fitness function
- Defining terminal criteria.

**Choice of Terminals and Functions**

A terminal symbol $t_i \in T$ can be a constant or a variable. Every function $f_i \in F$ can be an arithmetic operator \{+, −, ×, /\}, an arithmetic function (sin, cos, exp, log) and an arbitrarily defined function appropriate to the problem under consideration. The terminal symbols and functions have chosen such that they provided sufficient expressive power to express the solution to a problem. This means that the problem must be solved by a composition of functions and terminals specified. For solving the equation (1), terminal set and function
set are taken as $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, t\}$ and $F = \{+, -, \times, \div, \sin, \cos, \exp, \log\}$ for Example 1. and $F = \{+, -, \times, \div, (\,), \wedge, \sqrt{}, \sin, \cos\}$ for Example 2.

**Construction of graph**

In ant colony programming technique, the search space consists of a graph with $\ell$ nodes where the nodes are the functions or terminals and edges are weighted by pheromone. The examples of such a graph are given in Figures 2 and 3. Each node in the graph holds either a function or a terminal. This graph is generated by a randomized process.

*FIG. 2: Graph with functions and terminals*

*FIG. 3: Graph with functions and terminals*

**Fitness function**

The aim of the fitness function is to provide a basis for competition among available solutions and to obtain the optimal solution. Hence the fitness function for (2) is defined
as

\[ E_r = \left( \hat{F}(r) - f(r, F(r)) \right)^2. \]  

(10)

**Terminal criteria**

The group of ants and their collective tours form a generation. In each generation, a set of expressions are generated by the artificial ants. If the required number of expressions minimize the fitness function \( E_r \) tends to zero or very close to zero and they satisfy the terminal conditions, the process may be stopped; otherwise continue the ACP approach.

**ACP methodology**

Artificial ants build solutions by performing randomized tours on the completely connected graph \( G(\mathbb{V}, \mathbb{E}) \). In the graph, vertices \( \mathbb{V} \) are represented by Functions and Terminals and the set \( \mathbb{E} \) of edges connect the vertices. The ants move on the graph by applying a stochastic local decision policy that makes use of pheromone trials and heuristic information. In this way, ants incrementally build solutions to the given problem.

In the first generation, all edges are initialized by equal pheromone weight. Sent \( k \) (< \( \ell \) ) ants through the graph from \( k \) starting points in a random fashion. Each ant is initially put on a randomly chosen start node. Each ant is moving from the node \( r \) to node \( s \) in the graph at time \( t \) according to the following probability law [29]

\[ p_{rs}(t) = \frac{\tau_{rs}(t) \cdot [\gamma_s]^{\beta}}{\sum_{i \in J^k_r} [\tau_{ri}(t)] \cdot [\gamma_i]^{\beta}}, \]

where \( \gamma_s = (1/(2 + \pi_s))^d \), \( \pi_s \) is the power of symbol \( s \) which can be either a terminal symbol or a function, \( d \) is the current length of the arithmetic expression, \( \beta \) is a parameter which controls the relative weight of the pheromone trail and visibility and \( J^k_r \) is the set of unvisited nodes. The power of the symbols can be calculated from the following Table I. When an ant reaches a node, it determines whether the node is a terminal or a function node. If the ant is on a terminal node, the end of the tour has been reached for that ant.

After having found a tour of an ant, the ant deposits pheromone information on the edges through which it travelled. It constitutes a local update of the pheromone trail, which also comprises partial evaporation of the trail. The local update process is carried out according to the formula:

\[ \tau_{ij}(t + 1) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0, \]

**TABLE I: Power of terminal symbols and functions**

<table>
<thead>
<tr>
<th>Terminal symbol or function</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant, variable</td>
<td>−1</td>
</tr>
<tr>
<td>Functions</td>
<td>1</td>
</tr>
</tbody>
</table>
where \(1 - \rho\), \(\rho \in (0,1]\), is the pheromone decay coefficient (\(\rho\) is the concentration of pheromone on edge over time, \(\rho > 0.5\) yields good solution), \(\tau_{ij}\) is the amount of pheromone trail on edge \((i,j)\) and \(\tau_0\) is the initial amount of pheromone on edge \((i,j)\).

Each ant has a working memory that stores data about its tour. The ant’s memory is represented programmatically by a parse tree structure. In this tree, the root and branches are functions and leaves are terminals. The depth of the memory tree is limited according to the nature of the problem.

The tour \(e * t + 1 * +5\) of an ant is represented as parse trees in the Figure 4. The tour \(e * t/7 * +4/5\) of another ant is represented as parse trees in the Figure 5. The tours of the ants and their corresponding expressions extracted from the parse trees are given.
TABLE II: Tours and Expressions

<table>
<thead>
<tr>
<th>Tours of ants</th>
<th>Expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e \ast t \ast$</td>
<td>$e^t$</td>
</tr>
<tr>
<td>$e \ast t + 1 \ast + 5$</td>
<td>$e^{t+1} + 5$</td>
</tr>
<tr>
<td>$e \ast t/5 \ast + 0$</td>
<td>$e^{t/5} + 0$</td>
</tr>
<tr>
<td>$e \ast t/7 \ast + 4/5$</td>
<td>$e^{t/7} + 4/5$</td>
</tr>
<tr>
<td>$e \ast 3 \ast t - 2 \ast + + 5/2$</td>
<td>$e^{3(t-2)} + 5/2$</td>
</tr>
<tr>
<td>$e \ast 3 \ast t \ast + 5/2 - e \ast t \ast /7$</td>
<td>$e^{3t} + 5/2 - e^{t}/7$</td>
</tr>
</tbody>
</table>

TABLE III: Discarded tours

<table>
<thead>
<tr>
<th>Tours of ants</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+ e \ast t \ast + 1$</td>
</tr>
<tr>
<td>$e \ast t + 1 \ast + - 5$</td>
</tr>
<tr>
<td>$e \ast t/- * 2$</td>
</tr>
<tr>
<td>$- * e \ast t/7 * += - / + 0$</td>
</tr>
</tbody>
</table>

in the Table II. Some tours of the ants cannot be represented as the parse trees. Such type of tours are given in the following Table III. They are discarded when the parse tree construction process is carried out for the tours of the ants. This Parse tree construction is helpful to converge the solution quickly and also reduces the computation time by discarding the unnecessary tours. After each generation, a global update of pheromone trail is taken place. The level of pheromone is then changed as follows:

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

where $g$ is the number of generations, edges $(i, j)$ belong to the optimal tour found so far and $L$ is the length of this tour. The aim of the pheromone value update rule is to increase the pheromone values on the solution path. The update rule reduces the size of the searching region in order to find high quality solution with reasonable computation time. On the updated graph, the consecutive cycles of the ant colony algorithm are carried out by sending the ants through the best tour of the previous generation. The procedure is repeated until the fitness function (10) becomes zero or very close to zero. The optimal tour of the ACP and its corresponding tree are given in Figures 6 and 7 respectively.

ACP algorithm

Step 1. Construct a graph with $\ell$ nodes.

Step 2. Initialize the equal weight of pheromone in each edge of the graph.
Step 3. Pass $k$ ants through the graph from $k$ starting points and they move to the next node according to the probability law.

Step 4. Apply local update rule after the tour of each ant.

Step 5. Construct parse trees from the tours of $k$ ants.

Step 6. Extract the expressions from the trees.
Step 7. Evaluate the fitness function.

Step 8. If $E_r \to 0$ and they satisfy the initial conditions, then stop. Otherwise apply global update rule.

Step 9. Identify the best tour of the previous generation.

Step 10. Pass the same $k$ ants through the best tour and go to Step 4.

IV. NUMERICAL EXAMPLES

IV-1. Example 1
Consider the Takagi-Sugeno fuzzy Schrödinger equation with linear potential (ie $V_i(r) = r$) that can be expressed in the form:

$$
\frac{d^2 F(r)}{dr^2} + V_i(r)F(r) - \frac{L_i(L_i+1)}{r^2} F(r) = -k_i^2 F(r),
$$

(11)

where $V_1(r) = r$, $V_2(r) = r$, $k_i^2 = 1$, $L_i = 2$, $F(1) = 1$, and $\frac{dF(1)}{dr} = 3$.

Solution obtained using ant colony programming

The graph is generated randomly with 16 nodes. Let $\rho = 0.8$ and $\beta = 2$. Each edge is initialized by a pheromone weight of 1.0. Ten ants are taken for sending through the graph from 10 different points.

As the ant colony programming is carried out continuously, the solution of each generation will be improved by the pheromone updating rules. The construction of parse tree for the tour of the ants will converge the optimal solution quickly by discarding some unnecessary tours.

After 300 generations, the following expressions satisfy the fitness function and the value of the fitness function tends to 0. The expressions also satisfy the initial conditions. Hence the solution of the equation (11) is obtained.

Tours : $F(r) = \frac{5}{4}e^{2(t-1)} - \frac{1}{4}e^{2(1-t)}$

Expressions : $F(r) = 5/4 * e * 2t - 2 * -e * 2 - 2t * 1/4$

After 300 generations, the solutions are obtained from the graph. The numerical solutions of equation (1) are calculated and displayed in the Table IV using the ACP, RK method and Numerov’s method. Since this problem is having explicit solution, the ACP solution is equivalent to exact solution of the equation (11). The parse trees for the solutions are given in the Figures 8.

Similarly the solution of the above system with $V_2(r)$ can be found out using ant colony programming.
TABLE IV: Solutions of Schrödinger equation

<table>
<thead>
<tr>
<th>t</th>
<th>RK Solution</th>
<th>NM Solution</th>
<th>ACP Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>1.1</td>
<td>1.322067</td>
<td>1.313725</td>
<td>1.322070</td>
</tr>
<tr>
<td>1.2</td>
<td>1.697191</td>
<td>1.663590</td>
<td>1.697200</td>
</tr>
<tr>
<td>1.3</td>
<td>2.140429</td>
<td>2.050779</td>
<td>2.140445</td>
</tr>
<tr>
<td>1.4</td>
<td>2.669567</td>
<td>2.475872</td>
<td>2.669593</td>
</tr>
<tr>
<td>1.5</td>
<td>3.305843</td>
<td>2.939153</td>
<td>3.305882</td>
</tr>
<tr>
<td>1.6</td>
<td>4.074790</td>
<td>3.440761</td>
<td>3.305882</td>
</tr>
<tr>
<td>1.7</td>
<td>5.007269</td>
<td>3.980765</td>
<td>5.007350</td>
</tr>
<tr>
<td>1.8</td>
<td>6.140703</td>
<td>4.559198</td>
<td>6.140816</td>
</tr>
<tr>
<td>1.9</td>
<td>7.520580</td>
<td>5.176077</td>
<td>7.520734</td>
</tr>
<tr>
<td>2.0</td>
<td>9.202277</td>
<td>5.831410</td>
<td>9.202486</td>
</tr>
</tbody>
</table>

FIG. 8: Parse Tree for \( F(r) \)

IV-2. Example 2

Consider the Takagi-Sugeno fuzzy Schrödinger equation with linear potential (ie \( V_i(r) = r \)) that can be expressed in the form:

\[
R_i: \text{If } F_j \text{ is } T_{ji}(m_{ji}, \sigma_{ji}), i = 1, \ldots, r \text{ and } j = 1, \ldots, n, \text{ then }
\]

\[
\frac{d^2 F(r)}{dr^2} + V_i(r)F(r) - \frac{L_i(L_i + 1)}{r^2}F(r) = -k_i^2 F(r),
\]
where $V_1(r)=1$, $r=t$, $V_2(r)=1$, $k_i^2=1$, $L_i=2$, $F(1)=1$, and $\frac{dF(1)}{dr}=3$.

**Solution obtained using ACP**

In this modified ACP approach, the construction graph consists of 21 nodes $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, t\}$ and $F = \{+, -, \times, /, (\_), \wedge, \sqrt{\_}, \sin, \cos, \exp\}$. In this study, the ACP has been tested out to run for 10, 20, 40, 60, 80 and up to 110 nodes in order to search for the solution of the second order differential equations given below:

$$\ddot{F}(r) + (2 - 6t^2)F(r) = 0$$

with the terminal conditions $F(1) = 1$ and $\dot{F}(1) = 3$.

The $\rho = 0.5$, $\tau_{ij}(0) = 0.2$ and $\beta = 1$, are taken in the present calculations. In the first generation, 600-900 digital ants are sent out from any random nodes until the ants reach to the limit where the terminal condition is satisfied. Then these generated expression which fulfilled the terminal conditions will be tested and filtered by the fitness function. The fitness function may not be close to zero, but the path that have been taken by the ants might lead to the final solution. Therefore after completion of each generation, a global update of pheromone trail is taken place in order to increase the pheromone value on the solution path. This significant piece of information will be used for the next generation. The above process will be repeated several times until the final solution is obtained. This expression is obtained when the digital ants were sent out through 97 cities:

$$F(r) = \frac{\cos(\sqrt{2}(t - 1)) * (t \wedge 2 + 1) * t \wedge 2 + \sin(sqrt(2) * (t - 1)) * 3 * (2 * t \wedge 2 + (0 - 3) + (0 - (5 \wedge 2 + 1)))}{4\sqrt{2}t^2(-\sqrt{2}(\cos(\sqrt{2}(t - 1))) + (26t^2 + 9t - 39) + 3(2t^2 - 26t - 3)\sin(\sqrt{2}(t - 1)))} \times \frac{\sin(\sqrt{2}(\sqrt{2}t + 3) - 7 * 2 * t + (0 - 7 \wedge 2 * t) + (0 - 1/(4 \wedge 5))}{t \wedge 2} + \sin(sqrt(2) * t) * (5 \wedge 2 + (0 - t \wedge 2 * (8 \wedge 2 + 3/(5 \wedge 2))) + (0 - t/5 \wedge 2))).$$

For the terminal conditions, this expression predicted $F(1) = 1.0341$ whereas for the $F(1) = 2.7889$. Somehow it differs by about 0.0341 for the $F(1)$ and 0.2115 for the $\dot{F}(1)$. The parse trees is shown in Figure 9. This expression suggested approximate solutions to the exact ones. Finally at 109 cities, the ACP method predicted another expression which satisfies the terminal conditions as well as the fitness function.

$$F(r) = \frac{\cos(\sqrt{2}(t - 1)) * (t - 1)) * (0 - sqrt(2)) * (9 * t + (0 - 6 \wedge 2 + 3) + (5 \wedge 2 + 1) * t \wedge 2 + \sin(sqrt(2) * (t - 1)) * 3 * (2 * t \wedge 2 + (0 - 3) + (0 - (5 \wedge 2 + 1)))}{4\sqrt{2}t^2(-\sqrt{2}(\cos(\sqrt{2}(t - 1))) + (26t^2 + 9t - 39) + 3(2t^2 - 26t - 3)\sin(\sqrt{2}(t - 1)))}.$$
TABLE V: Comparison between ACP (109 and 97 cities) and the exact solutions.

<table>
<thead>
<tr>
<th>t</th>
<th>Exact $F(r)$</th>
<th>109 cities $F(r)$</th>
<th>97 cities $F(r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.00000</td>
<td>1.00000</td>
<td>1.03410</td>
</tr>
<tr>
<td>1.1</td>
<td>1.31992</td>
<td>1.31992</td>
<td>1.33483</td>
</tr>
<tr>
<td>1.2</td>
<td>1.67875</td>
<td>1.67875</td>
<td>1.67889</td>
</tr>
<tr>
<td>1.3</td>
<td>2.07382</td>
<td>2.07382</td>
<td>2.06315</td>
</tr>
<tr>
<td>1.4</td>
<td>2.50093</td>
<td>2.50093</td>
<td>2.48297</td>
</tr>
<tr>
<td>1.5</td>
<td>2.95447</td>
<td>2.95447</td>
<td>2.93244</td>
</tr>
<tr>
<td>1.6</td>
<td>3.42762</td>
<td>3.42762</td>
<td>3.40441</td>
</tr>
<tr>
<td>1.7</td>
<td>3.91248</td>
<td>3.91248</td>
<td>3.89066</td>
</tr>
<tr>
<td>1.8</td>
<td>4.40026</td>
<td>4.40026</td>
<td>4.38211</td>
</tr>
<tr>
<td>1.9</td>
<td>4.88148</td>
<td>4.88148</td>
<td>4.86896</td>
</tr>
<tr>
<td>2.0</td>
<td>5.34619</td>
<td>5.34619</td>
<td>5.34093</td>
</tr>
</tbody>
</table>

The analytical solution for this second order differential equation is given as:

\[
F(r) = \frac{1}{8(\cos(\sqrt{2})^2 + \sin(\sqrt{2})^2)t^2} \left( 13 \cos(\sqrt{2})\sqrt{2} - 3 \sin(\sqrt{2}) \right) \sqrt{2} (3 \sin(\sqrt{2})t \sqrt{2} - 2 \cos(\sqrt{2})t^2 + 3 \cos(\sqrt{2}t)) \]

\[
F(r) = \frac{1}{8(\cos(\sqrt{2})^2 + \sin(\sqrt{2})^2)t^2} \left( 13 \sin(\sqrt{2})\sqrt{2} + 3 \cos(\sqrt{2}) \right) \sqrt{2} (3 \cos(\sqrt{2})t \sqrt{2} + 2 \sin(\sqrt{2})t^2 - 3 \sin(\sqrt{2}t))
\]

From the Table V, the ACP approach at 109 cities are almost identical to the analytical solution. Initially, the pheromone values are equally distributed through out all the edges. As the ants move through out all the edges, there will be some path which will be favorable as these path satisfies the initial conditions. The best path can be obtained only if these path satisfies the fitness function. In Figure 11, we depicted all these by showing the differences in terms of colours. There are five colours shown in the space graph: red, green, orange, yellow, pink and blue. Initially, the pheromone is distributed equally through out all the edges in the space graph with the blue colour. Those paths with red colour show the highest density or the most favorable edges chosen by the ants. From this Figure 11, the favorable edges are listed down in Figure 12.

Similarly the solution of the above system with $V_2(r)$ can be found out using ant colony programming.
V. CONCLUSION

In this paper, solution of Takagi-Sugeno (T-S) fuzzy Schrödinger equation in Positron-Hydrogen scattering has been obtained using ant colony programming (ACP). The ACP solution is equivalent or very close to the exact solution of the problem. Accuracy of the solution computed by ACP approach to the problem is qualitatively better. The solution of this novel method is compared with the traditional Runge Kutta (RK) method and Numerov’s method. An illustrative numerical example is presented for the proposed method.

Acknowledgement

The funding of this work by the UMRG grant (Account No: RG099/10AFR) is gratefully acknowledged.
FIG. 10: Parse tree for $F(r)$ (109 cities).

References

FIG. 11: Pheromone density in the space graph.

1573(94)00080-M
### FIG. 12: List of the most favorable edges.

<table>
<thead>
<tr>
<th>Edges</th>
<th>Pheromone density</th>
</tr>
</thead>
<tbody>
<tr>
<td>( [\cdot,\cdot] )</td>
<td>( [\cdot,\cdot] )</td>
</tr>
<tr>
<td>( [\cdot, 2] ), ( [\cdot, 1] ), ( [\cdot, 0] ), ( [\cdot, 2] )</td>
<td>( [\cdot, 2] ), ( [\cdot, 1] ), ( [\cdot, 0] )</td>
</tr>
<tr>
<td>( [\cdot, 0] ), ( [\cdot, 0] )</td>
<td>( [\cdot, 0] ), ( [\cdot, 0] )</td>
</tr>
<tr>
<td>( [2, +] ), ( [\text{sqrt}, t] )</td>
<td>( [2, +] ), ( [\text{sqrt}, t] )</td>
</tr>
<tr>
<td>( [t, t] )</td>
<td>( [t, t] )</td>
</tr>
<tr>
<td>( [+, 1] ), ( [t, t] ), ( [t, t] ), ( [t, t] )</td>
<td>( [+, 1] ), ( [t, t] ), ( [t, t] )</td>
</tr>
<tr>
<td>( [\text{sqrt}, t] ), ( [t, 5] ), ( [t, +] ), ( [3, t] )</td>
<td>( [\text{sqrt}, t] ), ( [t, 5] ), ( [t, +] )</td>
</tr>
<tr>
<td>( [5, t] )</td>
<td>( [5, t] )</td>
</tr>
<tr>
<td>Others</td>
<td>Others</td>
</tr>
</tbody>
</table>

- density of the pheromone values increase