Chapter 4. Computational results of Test Functions for Function Optimization

4.1 Test Functions

Results of all the three algorithms namely Simulated Annealing Algorithm, Threshold Acceptance Algorithm and Genetic Algorithm are numerically investigated on nine test functions. The selected test functions are discussed below with their boundary range.

During simulation in matlab 2009a the function names used are ca_min1 to ca_min9. In the thesis, now onwards in place of ca_min1 to ca_min9, function names F1 to F9 are used to denote the functions as shown in Table 4.1 for the ease of representation.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Nine Test Functions</th>
<th>Constraints</th>
<th>Function Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 (ca_min1)</td>
<td>f(x1, x2) = x1 + x2</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F2 (ca_min2)</td>
<td>f(x1, x2) = x1^2 + x2^2</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F3 (ca_min3)</td>
<td>f(x1, x2) = 20 + x1^2 + x2^2</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F4 (ca_min4)</td>
<td>f(x1, x2) = x1^4 + x2^4 + (2 × x1 × x2)</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F5 (ca_min5)</td>
<td>f(x1, x2) = x1^2 + x2^2 + (20 × x1 × x2)</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F6 (ca_min6)</td>
<td>f(x1, x2, x3, x4) = x1^2 + x2^2 + x3^2 + x4^2</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F7 (ca_min7)</td>
<td>f(x1, x2) = (x1 × sin√x1) + (x2 × sin√x2)</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F8 (ca_min8)</td>
<td>f(x1, x2) = x1^2 - (10 × cos(2 × π × x1)) + 10 + (10 × cos(2 × π × x2)) + 10</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
<tr>
<td>F9 (ca_min9)</td>
<td>f(x1, x2) = (4 × x1^4) + (2.1 × x2^4) + (x1^6 / 3) + (4 × x1 × x2) + (4 × x1^2) + (4 × x2^2)</td>
<td>[0,1]</td>
<td>Minimize</td>
</tr>
</tbody>
</table>
4.2 Simulation Results for functions F1 to F9 using Simulated Annealing Algorithm (SA)

Here simulated annealing algorithm is used to minimize functions F1 to F9. The functions have input variables $x_1, x_2, x_3$ and $x_4$. The range for variables varies from 0 to 1. The output which we get to minimize functions F1 to F9 is number of iterations. The below given figures shows simulation results for function F1 to F9. Figures describe the total number of iterations taken by simulated annealing algorithm to minimize functions F1 to F9.
For Function F1

![Simulation result for function F1 using SA](image)

**Figure 4.1 Simulation result for function F1 using SA**

As shown in Figure 4.1, total number of iterations taken to minimize function F1 using SA are 5924.
For Function F2

![Simulation result for function F2 using SA](image)

**Figure 4.2 Simulation result for function F2 using SA**

As shown in Figure 4.2, total number of iterations taken to minimize function F2 using SA are 1632.
For Function F3

![Problem Setup and Results](image)

**Figure 4.3 Simulation result for function F3 using SA**

As shown in Figure 4.3, total number of iterations taken to minimize function F3 using SA are 1468.
For Function F4

![Simulation result for function F4 using SA](image)

**Figure 4.4 Simulation result for function F4 using SA**

As shown in Figure 4.4, total number of iterations taken to minimize function F4 using SA are 2652.
For Function F5

![Problem Setup and Results](image)

Figure 4.5 Simulation result for function F5 using SA

As shown in Figure 4.5, total number of iterations taken to minimize function F5 using SA are 5942.
For Function F6

As shown in Figure 4.6, total number of iterations taken to minimize function F6 using SA are 3353.
For Function F7

![Problem Setup and Results](image)

**Figure 4.7 Simulation result for function F7 using SA**

As shown in Figure 4.7, total number of iterations taken to minimize function F7 using SA are 2280.
For Function F8

Figure 4.8 Simulation result for function F8 using SA

As shown in Figure 4.8, total number of iterations taken to minimize function F8 using SA are 1514.
For Function F9

Figure 4.9 Simulation result for function F9 using SA

As shown in Figure 4.9, total number of iterations taken to minimize function F9 using SA are 2455.
4.3 Simulation Results for functions F1 to F9 using Threshold Acceptance Algorithm (TA)

Here threshold acceptance algorithm is used to minimize functions F1 to F9. The functions have input variables $x_1, x_2, x_3$, and $x_4$. The range for variables varies from 0 to 1. The output which we get to minimize functions F1 to F9 is number of iterations. The below given figures shows simulation results for function F1 to F9. Figures describe the total number of iterations taken by threshold acceptance algorithm to minimize functions F1 to F9.
For Function F1

As shown in Figure 4.10, total number of iterations taken to minimize function F1 using TA are 1506.
For Function F2

As shown in Figure 4.11, total number of iterations taken to minimize function F2 using TA are 1000.
For Function F3

As shown in Figure 4.12, total number of iterations taken to minimize function F3 using TA are 1000.
For Function F4

![Problem Setup and Results]

Figure 4.13 Simulation result for function F4 using TA

As shown in Figure 4.13, total number of iterations taken to minimize function F4 using TA are 1000.
Chapter 4 Computational results of Test Functions for Function Optimization

For Function F5

![Problem Setup and Results](image)

**Figure 4.14 Simulation result for function F5 using TA**

As shown in Figure 4.14, total number of iterations taken to minimize function F5 using TA are 5916.
For Function F6

As shown in Figure 4.15, total number of iterations taken to minimize function F6 using TA are 2000.
For Function F7

![Simulation result for function F7 using TA](image)

**Figure 4.16 Simulation result for function F7 using TA**

As shown in Figure 4.16, total number of iterations taken to minimize function F7 using TA are 1044.
For Function F8

![Figure 4.17 Simulation result for function F8 using TA](image)

As shown in Figure 4.17, total number of iterations taken to minimize function F8 using TA are 1131.
For Function F9

![Figure 4.18 Simulation result for function F9 using TA](image)

As shown in Figure 4.18, total number of iterations taken to minimize function F9 using TA are 1000.
4.4 Simulation Results for functions F1 to F9 using Genetic Algorithm (GA)

Here genetic algorithm is used to minimize functions F1 to F9. The functions have input variables $x_1, x_2, x_3$ and $x_4$. The range for variables varies from 0 to 1. The output which we get to minimize functions F1 to F9 is number of iterations. The below given figures show simulation results for function F1 to F9. Figures describe the total number of iterations taken by genetic algorithm to minimize functions F1 to F9.
For Function F1

![Problem Setup and Results](image)

As shown in Figure 4.19, total number of iterations taken to minimize function F1 using GA are 100.
For Function F2

![Simulation result for function F2 using GA](image)

**Figure 4.20 Simulation result for function F2 using GA**

As shown in Figure 4.20, total number of iterations taken to minimize function F2 using GA are 51.
For Function F3

![Simulation result for function F3 using GA](image)

**Figure 4.21 Simulation result for function F3 using GA**

As shown in Figure 4.21, total number of iterations taken to minimize function F3 using GA are 51.
For Function F4

![Problem Setup and Results](image)

**Figure 4.22 Simulation result for function F4 using GA**

As shown in Figure 4.22, total number of iterations taken to minimize function F4 using GA are 52.
For Function F5

Figure 4.23 Simulation result for function F5 using GA

As shown in Figure 4.23, total number of iterations taken to minimize function F5 using GA are 100.
For Function F6

Figure 4.24 Simulation result for function F6 using GA

As shown in Figure 4.24, total number of iterations taken to minimize function F6 using GA are 64.
For Function F7

![Simulation result for function F7 using GA](image)

Figure 4.25 Simulation result for function F7 using GA

As shown in Figure 4.25, total number of iterations taken to minimize function F7 using GA are 51.
For Function F8

Figure 4.26 Simulation result for function F8 using GA

As shown in Figure 4.26, total number of iterations taken to minimize function F8 using GA are 51.
For Function F9

Figure 4.27 Simulation result for function F9 using GA

As shown in Figure 4.27, total number of iterations taken to minimize function F9 using GA are 51.
4.5 Computational Results by Comparative Analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) in terms of number of iterations.

Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) are applied to minimize the test functions. From the simulation results the comparative analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) in terms of number of iterations to minimize the functions F1 to F9 is as shown below in Table 4.2.

Table 4.2 Comparative Analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) in terms of number of iterations.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SA</td>
</tr>
<tr>
<td>F1</td>
<td>5924</td>
</tr>
<tr>
<td>F2</td>
<td>1632</td>
</tr>
<tr>
<td>F3</td>
<td>1462</td>
</tr>
<tr>
<td>F4</td>
<td>2652</td>
</tr>
<tr>
<td>F5</td>
<td>5942</td>
</tr>
<tr>
<td>F6</td>
<td>3353</td>
</tr>
<tr>
<td>F7</td>
<td>2280</td>
</tr>
<tr>
<td>F8</td>
<td>1514</td>
</tr>
<tr>
<td>F9</td>
<td>1168</td>
</tr>
</tbody>
</table>
4.6 Graphical Results of Comparative Analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) in terms of number of iterations.

For Function F1

![Function F1 Graph](image)

**Figure 4.28 Comparative Analysis of SA, TA and GA for function F1 in terms of iterations.**

Here the function F1 is $f(x_1, x_2) = x_1 + x_2$. A simple linear function with two variables $x_1$ and $x_2$ are taken. Summation of $x_1$ and $x_2$ is performed. As shown in figure 4.28, X-axis describes number of iterations (0 to 6000) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.28 show the comparative analysis of SA, TA and GA for function F1 in terms of number of iterations.

To minimize function F1 number of iterations taken by SA = 5924, TA = 1506 and GA = 100. The result show that GA takes far less number of iterations to minimize function F1 than SA and TA.
For Function F2

![Bar chart showing comparative analysis of SA, TA, and GA for function F2 in terms of number of iterations.]

Figure 4.29 Comparative Analysis of SA, TA and GA for function F2 in terms of iterations.

Here the function F2 is \( f(x_1, x_2) = x_1^2 + x_2^2 \). A function with two variables \( x_1 \) and \( x_2 \) are taken. Summation of square of \( x_1 \) and \( x_2 \) is performed. As shown in figure 4.29, X-axis describes number of iterations (0 to 1800) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.29 show the comparative analysis of SA, TA and GA for function F2 in terms of number of iterations.

To minimize function F2 number of iterations taken by SA = 1632, TA = 1000 and GA = 51. The result show that GA takes far less number of iterations to minimize function F2 than SA and TA.
For Function F3

Here the function F3 is $f(x_1, x_2) = 20 + x_1^2 + x_2^2$. A function with two variables $x_1$ and $x_2$ are taken. Summation of square of $x_1$, $x_2$ a constant is performed. As shown in figure 4.30, X-axis describes number of iterations (0 to 1600) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.30 show the comparative analysis of SA, TA and GA for function F3 in terms of number of iterations.

To minimize function F3 number of iterations taken by SA = 1462, TA = 1000 and GA = 51. The result show that GA takes far less number of iterations to minimize function F3 than SA and TA.
For Function F4

Here the function F4 is $f(x_1, x_2) = x_1^4 + x_2^4 + (2 \times x_1 \times x_2)$. A function with two variables $x_1$ and $x_2$ are taken. Summation of $x_1$ to the power of four, $x_2$ to the power of four and a constant is performed. As shown in figure 4.31, X-axis describes number of iterations (0 to 3000) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.31 show the comparative analysis of SA, TA and GA for function F4 in terms of number of iterations.

To minimize function F4 number of iterations taken by SA = 2652, TA = 1000 and GA = 52. The result show that GA takes far less number of iterations to minimize function F4 than SA and TA.
For Function F5

![Function F5](image)

**Figure 4.32 Comparative Analysis of SA, TA and GA for function F5 in terms of iterations.**

Here the function F5 is \( f(x_1, x_2) = x_1^2 + x_2^2 + (20 \times x_1 \times x_2) \). A function with two variables \( x_1 \) and \( x_2 \) are taken. Summation of square of \( x_1 \), \( x_2 \) and a constant is performed. As shown in figure 4.32, X-axis describes number of iterations (0 to 6000) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.32 show the comparative analysis of SA, TA and GA for function F5 in terms of number of iterations.

To minimize function F5 number of iterations taken by SA = 5942, TA = 5916 and GA = 100. The result show that GA takes far less number of iterations to minimize function F5 than SA and TA.
For Function F6

![Figure 4.33 Comparative Analysis of SA, TA and GA for function F6 in terms of iterations.](image)

Here the function F6 is \( f(x_1, x_2, x_3, x_4) = x_1^2 + x_2^2 + x_3^2 + x_4^2 \). A function with four variables \( x_1, x_2, x_3 \) and \( x_4 \) are taken. Summation of square of \( x_1, x_2, x_3 \) and \( x_4 \) is performed. As shown in figure 4.33, X-axis describes number of iterations (0 to 3500) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.33 show the comparative analysis of SA, TA and GA for function F6 in terms of number of iterations.

To minimize function F6 number of iterations taken by SA = 3353, TA = 2000 and GA = 64. The result show that GA takes far less number of iterations to minimize function F6 than SA and TA.
For Function F7

Here the function F7 is $f(x_1, x_2) = (x_1 \times \sin\sqrt{x_1}) + (x_2 \times \sin\sqrt{x_2})$. A function with two variables $x_1$ and $x_2$ are taken. This function also includes Sine function. To make the function complex the square root of $x_1$ and $x_2$ with sine function is used in the function. As shown in figure 4.34, X-axis describes number of iterations (0 to 2500) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.34 show the comparative analysis of SA, TA and GA for function F7 in terms of number of iterations.

To minimize function F7 number of iterations taken by SA = 2280, TA = 1044 and GA = 51. The result show that GA takes far less number of iterations to minimize function F7 than SA and TA.
For Function F8

![Function F8](image)

**Figure 4.35 Comparative Analysis of SA, TA and GA for function F8 in terms of iterations.**

Here the function F8 is $f(x_1, x_2) = x_1^2 - (10 \times \cos(2 \times \pi \times x_1)) + 10 + (10 \times \cos(2 \times \pi \times x_2)) + 10$. A function with two variables $x_1$ and $x_2$ are taken. To make the function complex cosine function is used in the function. As shown in figure 4.35, X-axis describes number of iterations (0 to 1600) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.35 show the comparative analysis of SA, TA and GA for function F8 in terms of number of iterations.

To minimize function F8 number of iterations taken by SA = 1514, TA = 1131 and GA = 51. The result show that GA takes far less number of iterations to minimize function F8 than SA and TA.
For Function F9

Here the function F9 is 

\[ f(x_1, x_2) = 4 \times x_1^2 + (2.1 \times x_1^4) + \left( x_1^6 / 3 \right) + (4 \times x_1 \times x_2) + (4 \times x_2^2) + (4 \times x_2^4). \]

A function with two variables \( x_1 \) and \( x_2 \) are taken. To make the function complex floating point number is used in the function. As shown in figure 4.36, X-axis describes number of iterations (0 to 1200) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.36 show the comparative analysis of SA, TA and GA for function F9 in terms of number of iterations.

To minimize function F9 number of iterations taken by SA = 1168, TA = 1000 and GA = 51. The result show that GA takes far less number of iterations to minimize function F9 than SA and TA.
4.6.1 Graphical Results of Comparative Analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) for functions F1 to F9 in terms of iterations.

For Functions F1 to F9

![Comparative Analysis of SA, TA and GA for functions F1 to F9 in terms of iterations.](image)

As shown in figure 4.37, X-axis describes number of iterations (0 to 6000) and Y-axis describes nine different functions i.e. F1 to F9. After the rigorous comparison of SA, TA and GA with each function from F1 to F9, figure 4.37 shows the overall comparison of SA, TA and GA for functions F1 to F9 in terms of iterations.

From the above results it is found that number of iterations taken by GA to minimize functions F1 to F9 is far less than SA and TA.
4.7 Graphical Results of Comparative Analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) with respect to time (in seconds)

Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) are applied to minimize the test functions. From the simulation results the comparative analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) with respect to time to minimize the functions F1 to F9 is as shown below in Table 4.3.

Table 4.3 Comparative Analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) with respect to time.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SA</td>
</tr>
<tr>
<td>F1</td>
<td>30</td>
</tr>
<tr>
<td>F2</td>
<td>14</td>
</tr>
<tr>
<td>F3</td>
<td>11</td>
</tr>
<tr>
<td>F4</td>
<td>19</td>
</tr>
<tr>
<td>F5</td>
<td>31</td>
</tr>
<tr>
<td>F6</td>
<td>24</td>
</tr>
<tr>
<td>F7</td>
<td>17</td>
</tr>
<tr>
<td>F8</td>
<td>12</td>
</tr>
<tr>
<td>F9</td>
<td>10</td>
</tr>
</tbody>
</table>
For Function F1

Here the function F1 is \( f(x_1, x_2) = x_1 + x_2 \). A simple linear function with two variables \( x_1 \) and \( x_2 \) are taken. Summation of \( x_1 \) and \( x_2 \) is performed. As shown in figure 4.38, X-axis describes time in seconds (0 to 30) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.38 show the comparative analysis of SA, TA and GA for function F1 with respect to time.

To minimize function F1 time taken by SA = 30, TA = 12 and GA = 5. The result show that time taken by GA to minimize function F1 is less than SA and TA.
For Function F2

![Function F2](image)

Figure 4.39 Comparative Analysis of SA, TA and GA for function F2 with respect to time.

Here the function F2 is \( f(x_1, x_2) = x_1^2 + x_2^2 \). A function with two variables \( x_1 \) and \( x_2 \) are taken. Summation of square of \( x_1 \) and \( x_2 \) is performed. As shown in figure 4.39, X- axis describes time in seconds (0 to 14) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.39 show the comparative analysis of SA, TA and GA for function F2 with respect to time.

To minimize function F2 time taken by SA = 14, TA = 7 and GA = 2. The result show that time taken by GA to minimize function F2 is less than SA and TA.
For Function F3

Here the function F3 is \( f(x_1, x_2) = 20 + x_1^2 + x_2^2 \). A function with two variables \( x_1 \) and \( x_2 \) are taken. Summation of square of \( x_1, x_2 \) a constant is performed. As shown in figure 4.40, X-axis describes time in seconds (0 to 12) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.40 show the comparative analysis of SA, TA and GA for function F3 with respect to time.

To minimize function F3 time taken by SA = 11, TA = 7 and GA = 2. The result show that time taken by GA to minimize function F3 is less than SA and TA.
For Function F4

Here the function F4 is \( f(x_1, x_2) = x_1^4 + x_2^4 + (2 \times x_1 \times x_2) \). A function with two variables \( x_1 \) and \( x_2 \) are taken. Summation of \( x_1 \) to the power of four, \( x_2 \) to the power of four and a constant is performed. As shown in figure 4.41, X-axis describes time in seconds (0 to 20) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.41 show the comparative analysis of SA, TA and GA for function F4 with respect to time.

To minimize function F4 time taken by SA = 19, TA = 7 and GA = 2. The result show that time taken by GA to minimize function F4 is less than SA and TA.
For Function F5

![Function F5](image)

**Figure 4.42 Comparative Analysis of SA, TA and GA for function F5 with respect to time.**

Here the function F5 is \( f(x_1, x_2) = x_1^2 + x_2^2 + (20 \times x_1 \times x_2) \). A function with two variables \( x_1 \) and \( x_2 \) are taken. Summation of square of \( x_1 \), \( x_2 \) and a constant is performed. As shown in figure 4.42, X-axis describes time in seconds (0 to 35) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.42 show the comparative analysis of SA, TA and GA for function F5 with respect to time.

To minimize function F5 time taken by SA = 31, TA = 29 and GA = 5. The result show that time taken by GA to minimize function F5 is less than SA and TA.
For Function F6

![Figure 4.43 Comparative Analysis of SA, TA and GA for function F6 with respect to time.](image)

Here the function F6 is \( f(x_1, x_2, x_3, x_4) = x_1^2 + x_2^2 + x_3^2 + x_4^2 \). A function with four variables \( x_1, x_2, x_3 \) and \( x_4 \) are taken. Summation of square of \( x_1, x_2, x_3 \) and \( x_4 \) is performed. As shown in figure 4.43, X- axis describes time in seconds (0 to 25) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.43 show the comparative analysis of SA, TA and GA for function F6 with respect to time.

To minimize function F6 time taken by SA = 24, TA = 15 and GA = 3. The result show that time taken by GA to minimize function F6 is less than SA and TA.
For Function F7

Here the function F7 is \( f(x_1, x_2) = (x_1 \times \sin \sqrt{x_1}) + (x_2 \times \sin \sqrt{x_2}) \). A function with two variables \( x_1 \) and \( x_2 \) are taken. This function also includes Sine function. To make the function complex the square root of \( x_1 \) and \( x_2 \) with sine function is used in the function. As shown in figure 4.44, X-axis describes time in seconds (0 to 18) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.44 show the comparative analysis of SA, TA and GA for function F7 with respect to time.

To minimize function F7 time taken by SA = 17, TA = 8 and GA = 2. The result show that time taken by GA to minimize function F7 is less than SA and TA.
For Function F8

Here the function F8 is \( f(x_1, x_2) = x_1^2 - (10 \times \cos(2 \times \pi \times x_1)) + 10 + (10 \times \cos(2 \times \pi \times x_2)) + 10 \). A function with two variables \( x_1 \) and \( x_2 \) are taken. To make the function complex cosine function is used in the function. As shown in figure 4.45, X-axis describes time in seconds (0 to 12) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.45 show the comparative analysis of SA, TA and GA for function F8 with respect to time.

To minimize function F8 time taken by SA = 12, TA = 9 and GA = 2. The result show that time taken by GA to minimize function F8 is less than SA and TA.
For Function F9

Here the function F9 is $f(x_1, x_2) = (4 \times x_1^2) + (2.1 \times x_1^4) + (x_1^6 / 3) + (4 \times x_1 \times x_2) + (4 \times x_2^2) + (4 \times x_2^4)$. A function with two variables $x_1$ and $x_2$ are taken. To make the function complex floating point number is used in the function. As shown in figure 4.46, X-axis describes time in seconds (0 to 10) and Y-axis describes three different algorithms i.e. SA, TA and GA. Figure 4.46 show the comparative analysis of SA, TA and GA for function F9 with respect to time.

To minimize function F9 time taken by SA = 10, TA = 7 and GA = 2. The result show that time taken by GA to minimize function F9 is less than SA and TA.
4.7.1 Graphical Results of Comparative Analysis of Simulated Annealing Algorithm (SA), Threshold Acceptance Algorithm (TA) and Genetic Algorithm (GA) for functions F1 to F9 with respect to time.

For Functions F1 to F9

![Graphical Results](image)

**Figure 4.47 Comparative Analysis of SA, TA and GA from functions F1 to F9 with respect to time.**

As shown in figure 4.47, X-axis describes time in seconds (0 to 35) and Y-axis describes nine different functions i.e. F1 to F9. After the rigorous comparison of SA, TA and GA with each function from F1 to F9, figure 4.47 shows the overall comparison of SA, TA and GA for functions F1 to F9 with respect to time.

From the above results it is found that time taken by GA to minimize functions F1 to F9 is far less than SA and TA.