Diversity Guided Particle Swarm Optimization algorithm based on Search Space Awareness Particle Dispersion (DGPSO)

Anvar Bahrampour¹, Omid Mohamad Nezami ²
¹Department of computer engineering, Sanandaj Branch, Islamic Azad University, Sanandaj, Iran, Anvar.Bahrampour@gmail.com
²Department of computer engineering, Bijar Branch, Islamic Azad University, Bijar, Iran, Nezami@iaubijar.ac.ir

Abstract - Diversity control in the particle swarm optimization (PSO) algorithm is one of the important issues that influence the process of finding global optimal solution. In this study we create a historical process to find best area of the search space for population dispersion guide on PSO algorithm, and name Diversity Guided Particle Swarm Optimization algorithm (DGPSO) algorithm. Hence we propose a mechanism to guide the swarm based on diversity by using a diversifying process in order to detect suitable positions of the search space (points with fairly good fitness, and good distance from current distribution of the swarm particles) to disperse or relocating some of existing particles, hoped to increase diversity level of the swarm and escape from local optimal by detecting better area of the search space. This model uses a diversity measuring, and swarm dispersion mechanism to control the evolutionary process alternating between exploring and exploiting behavior. The numerical results show that the proposed algorithm outperforms other algorithms in most of the test cases taken in this study.

Keywords: Evolutionary Algorithm (EA), Particle Swarm Optimization (PSO) Algorithm, Population Diversity, and Premature Convergence.

1 Introduction

Particle Swarm Optimization (PSO) applies to concept of social interaction to problem solving and it was invented by Russ Eberhart and James Kennedy in 1995 [1, 2]. Each particle represents a point of search space or a solution of problem, denoted by \( X_i \). The PSO algorithm iteratively modifies the point and the velocity of each particle as it looks for the optimal solution based on Equation (1).

\[
\begin{align*}
V_{id} &= \omega \cdot V_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \\
X_{id} &= X_{id} + V_{id}
\end{align*}
\]

Where \( V_i \) in the first equation is the velocity of Particle \( i \) that represented as \( V_i \). The first part of the Equation (1) is the inertia of the previous velocity, \( \omega \) is predefined by the user, and the second part represents the cognition of the particle that shows personal thinking of the particle. The third part is the social component. In this equation \( c_1 \) and \( c_2 \) are acceleration constants. They represent the weighting of the stochastic acceleration terms that pull each particle toward personal best and global best positions. The constants \( r_1, r_2 \) are the uniformly generated random numbers in the range of (0, 1]. Although PSO is simple, but it is a powerful search technique, Many researchers have shown empirically in many studies that it works well [3].

The rate of convergence of particles in PSO is good through the fast information flowing among particles, so its diversity decreases very quickly in the successive iterations and lead to a suboptimal solution. This situation was said that an evolutionary process was trapped in a local optimal or premature convergence of the process.

The standard PSO algorithm can easily get trapped in the local optimal when solving complex multimodal problems. These weaknesses have restricted wider applications of the PSO [4, 5, and 6]. Some reasons cause to this problem, one of that is decreasing diversity of population. A number of variants of PSO algorithm have been proposed to overcome the problem of diversity loss. One of the common methods to increase the diversity is mutation. Mutation causes an improvement in exploration abilities, which can be applied to different elements of a particle swarm. The effect of mutation depends on which elements of the swarm are mutated [7]. Velocity vector mutation is equivalent to particle’s position vector mutation, under the condition that the same mutation operator is considered.

In [7] a negative feedback mechanism into particle swarm optimization has proposed and developed an adaptive PSO. This mechanism takes advantage of the swarm-
diversity to control the tuning of the inertia weight (PSO-DCIW), which in turn can adjust the swarm diversity adaptively and contribute to a successful global search. Some other methods exist that using diversity measuring and mutation in the particle's position, to promote the performance of the algorithm include Gaussian Mutation [7,9,10,11,12], Cauchy [12,13], and Chaos Mutation [14,15,16].

There are other different ways of introducing diversity and controlling the degree of diversity; Riget and Vesterstorm proposed an algorithm named ARPSO. In ARPSO if diversity is above the predefined threshold \( d_{high} \) then particles attract each other, and if it is below \( d_{low} \) then the particle repel each other until they meet the required high diversity \( d_{high} \) [17]; repulsion to keep particles away from the optimum proposed by Parsopoulos and Vrahatis [18]; LoZvbjerg and Krink made dispersion between particles that are too close to one another [19]; and Blackwell and Bentley have reduced the attraction of the swarm centre to prevent the particles clustering too tightly in one region of the search space in order to escape from local optimal [20]. J. J. Liang and P. N. Suganthan proposed a dynamic multi-swarm particle swarm optimizer (DMS-PSO)[21]. In this method whole population is divided into many small swarms, these swarms are regrouped frequently by using various regrouping schedules and information is exchanged among the swarms.

In this paper we propose a mechanism to guide the swarm based on diversity by using a diversifying process in order to detect suitable positions of the search space (points with fairly good fitness, and good distance from current distribution of the swarm particles) to disperse or relocating some of existing particles, hoped to increase diversity level of the swarm and escape from local optimal by detecting better area of the search space.

The rest of the paper is organized as follows: Section 2 we have a definition of diversity definition and measuring. The DGPSO described in section 3. Experimental results are discussed in section 4. Finally, this paper concludes in section 5.

2 Diversity defifinition and measuring

Population diversity is a way to monitor the degree of convergence or divergence in PSO search process [3]. Though, there are several measures have been used to detect diversity level of the population. Shi and Eberhart in [23], [24], [25], gave three definitions of PSO population diversity measurements that include: position diversity, velocity diversity, and cognitive diversity. Shi Cheng and Yuhui Shi gave new definition of population diversity measurement called L1 norm base on both element-wise and dimension-wise diversity [3], and they have shown that useful information on search process of an optimization algorithm could be obtained by using dimension-wise definition in L1 norm variant, so in this paper we use L1 norm of position diversity measurement. Let \( m \) shows the number of particles and \( n \) is the number of dimensions. Dimension-wise definition in L1 norm special is as Equation (2).

\[
\begin{align*}
\bar{x} &= \frac{1}{m} \sum_{i=1}^{m} x_i \\
D^p &= \frac{1}{m} \sum_{i=1}^{m} |x_i - \bar{x}| \\
D^p &= \frac{1}{n} \sum_{i=1}^{n} D^p_i
\end{align*}
\]

Where vector \( \bar{x} \) is mean of particle's position on each dimension, vector \( D^p \) is particle's position diversity vector based on L1 norm, and \( D^p \) is the whole population diversity value.

In [26], some other approaches for measuring of population diversity have been introduced in evolutionary computation including: hamming distance, Euclidean distance, information entropy, etc. In this paper we use Euclidean distance in selection process of particles to disperse in dispersion mechanism. The Euclidean distance is as Equation (3) for measures the distance between two particles \( X \) and \( Y \):

\[
D(X,Y) = \sqrt{\sum (x_i - y_i)^2}
\]

3 Diversity guided particle swarm optimization

The basic idea of this research is to measure the diversity level of the swarm during the evolutionary process and once the diversity of the population drops down to the predefined threshold level \( d \), then the system start to disperse or redistribute some of the swarm's particles by relocating them to new suitable positions which have fairly good fitness and relatively high distance from convergence position. Note, since we use previous personal best positions of dispersed particles, this is not a replacement of some particles by new generated particles. Though in this approach we relocate existing particles to new suitable positions instead of replacement them. Therefore by this process, the diversity level of the swarm will be increased up to certain degree. The evolutionary process will consistently reduce the diversity level again, and the dispersion process should be repeated once the diversity level drops down the \( d \) value. In PSO algorithm the speed of convergence is very high, so the swarm dispersion process should be repeated very often. On the other hand, repetition of this process is relatively time consuming, and in addition exploitation ability of the algorithm will be decreased by high frequency
swarm dispersion. Therefore, we introduce a new parameter \( T \), to define the duration that the dispersion system should be passive after each redistribution process. The following two sections illustrate the process in more details. The proposed method for dispersing or relocating swarm's particles was implemented independently from the problem characteristics to improve the global convergence behavior of PSO algorithms.

### 3.1 Swarm dispersion process

Figure 1 shows the process of Diversity Guided Particle Swarm Optimization algorithm (DGPSO); the steps of this process are the same as the steps of the standard PSO except steps 5 and 6. In order to detect target positions of selected particles for dispersion, we use the information that could be determined from previous generations of the PSO process using previous best particle's to determine good points in the search space. With this aim, we develop an external archive to store best particles of previous generations as good positions in the search space; hopefully, there are better points in the regions that this stored particles located in. In step 5 we update the External Archive if necessary; there is not necessary external archive to be updated in any iteration. When dispersion system is active, last dispersion took place more than \( T \) generations ago, in step 6 the swarm diversity is measured, and if diversity is higher than the predefined threshold this step didn’t do anything else, otherwise dispersion process starts to disperse some of the swarm's particles. Dispersion process will increase the swarm diversity by relocating idle particles to new potent positions. We define idle particle as a particle that there is no change in its personal best position for a long time. Process of determining target positions will describe in the following section. The final step of dispersion process is to reset velocity of dispersed particles to zero, because we want each dispersed particle search very carefully for better solutions in the vicinity of new location. In this study we found that nonzero particle's velocity probably causes to move away from new position rapidly, and lead to have unsearched area in that new region. Finally each idle particle has a period of \( T \) generation to change its personal best position, if no change took place in that duration it would be an idle particle in next idle selection process too.

### 3.2 Target positions of idle particles

In this section we describe a mechanism for determining target positions of the selected idle particles to disperse over the search space. In this research we established an external archive with 100 particles, and initialized it with random particles. Firstly, we gather particles with best fitness in the first generations (about 100 generations in this research) of the PSO process and replace particles in the external archive which have bad fitness. Then we should establish a replacement policy in order to gather effective particles in external archive. These particles should have good fitness and high distance from the center of current distribution of the external archive particles to avoid the convergence of external archive. In this study after first 100 generations, we only do replacement when fitness of global best particle changed notably, and remove one of the particles with low diversity. For detect low diversity particles to remove from external archive, we use a Euclidean distance described in section 2, and measure distance of each particle from the mean of external archive particles. One of the particles with less distance should be replaced by new particle.

![Figure 1. Steps of DGPSO Algorithm](image)

To determine new good positions for relocating of idle particles from information of the external archive, we add two new particles of \( x_{\text{Max}} \) and \( x_{\text{Min}} \) (vector of maximum or minimum value in each dimension) to external archive, for mutation purpose. Then we create a roulette wheel that weighted each particle based on fitness and distance of external archive particles from the center of the swarm. In order to generate new target location, for each dimension value we select one particle of the external archive randomly based on Roulette, and use value stored in the same dimension of selected particle. After value of each dimension was selected, we probably have new suitable position for dispersion process, but in this time we don’t use this point as good position with certainty. We collect all generated points in one matting pool, and add external archive particles as good points of the search space to the pool too. Then we apply operators such as genetic crossover and mutation to produce new points probably with good fitness and good diversity. Then we select a numbers of best points (45% of the swarm in this research) based on fitness and distance.
from the center of the swarm distribution, and return to dispersion process to relocate randomly the same numbers of selected idle particles of the swarm to this new positions. This process will increase diversity of the swarm notably and help to escape from local optimal trap. Figure 2 illustrates this mechanism.

![Figure 2](image2.png)

**Figure 2.** Mechanism of determining target positions of idle particles.

To illustrate impact of dispersion mechanism in diversity level of the swarm in DGPSO we use a 2-D Rastrigin function ($f_8$ in Table II) with 30 particles in 100 generations, and dispersion rate 45%. Figure 3 represents diversity curves of standard global PSO and DGPSO, and shows how diversity level of the swarm changes in DGPSO in each 15 generations.

![Figure 3](image3.png)

**Figure 3.** Diversity of the swarm of $f_8$ in GPSO and DGPSO Processes

## 4 Experimental setting and numerical results

For comparison some variants of PSO and DGPSO algorithms, we have used a collection of 10 standard benchmark problems. Mathematical models of the problems along with the true optimum value are given in TABLE 2. In this problem set, we have a unimodal functions such as $f_2$, $f_5$, and $f_7$, $f_8$ is a noisy quadric function, where random $[0, 1]$ is a uniformly distributed random variable in $[0, 1]$. The others are multimodal [27]. The entire set of test problems taken for this study is scalable i.e. the problems can be tested for any number of variables. However, for the present study we have tested the problems for dimensions 30 and 50.

In order to make a fair comparison of proposed DGPSO with some of other variants of PSO algorithm, we implement standard PSO algorithm in both global star structure and local ring structure named GPSO and LPSO, respectively. In addition to these comparisons we also implement PSO DCIW and DMS PSO algorithms, which proposed in [8, 21], and compared with DGPSO. We use the same initial population for all algorithms. The population size was taken as 20 while we have 30 variables (dimensions) for all the test problems, and 50 when problems should be tested with 50 variables. A linearly decreasing inertia weight is used which starts at 0.9 and ends at 0.4, with the user defined parameters $c_1=2.0$ and $c_2=2.0$. For each algorithm, the maximum number of iterations is set as 3000 iterations in the case of 30 dimensions, and 10000 for dimension 50. A new parameter $T$ for DGPSO algorithms is set as 30 and 50 in cases of 30 and 50 dimensions respectively, with the external archive of size 100, dispersion rate $R$ is set as 45%. In DMS-PSO we use group size 3 and regroup period 5. A total of 20 runs for each experimental setting were conducted, and the results are given in TABLE 1 and TABLE 3, in terms of mean of best fitness, standard deviation, and the improvement (%) of proposed DGPSO algorithm in comparison with original GPSO. Figure 5 through 6 show performance curves of the DGPSO in comparison with other variants of PSO for test functions $f_1$, $f_5$, and $f_8$ by mean fitness of best particles history found by the swarm in all runs. The numerical results show that the proposed algorithm outperforms other variants of PSO in most of the test cases taken in this study.

![Figure 4](image4.png)

**Figure 4.** Performance curves of GPSO, LPSO, DMS-PSO, PSO DCIW, and DGPSO for function $f_1$

## 5 Conclusion

Evolutionary algorithms (EAs) are best solutions for solving optimization problems; however those have different ability to investigate search space and attain optimum solution but have same behavior. One of the ideas for control
the behavior of these algorithms is a rein between exploration and exploitation, for this issue we should have good mechanism for control the diversity of population in different stages to achieve unsearched spaces. In order to control diversity to survey unsearched spaces we used search historical approach and implement that on the PSO algorithm, one of the powerful EAs. We propose a mechanism to guide the swarm based on diversity by using a diversifying process in order to detect suitable positions of the search space, this model uses a diversity measuring, and swarm dispersion mechanism to control the evolutionary process alternating between exploring and exploiting behavior and guide the algorithm to survey unsearched spaces we used search different stages to achieve unsearched spaces. In order to good mechanism for control th

![Graph](image)

**Figure 5.** Performance curves of GPSO, LPSO, DMS-PSO, PSO_DC1W, and DGPSO for function $f_5$

![Graph](image)

**Figure 6.** Performance curves of GPSO, LPSO, DMS-PSO, PSO_DC1W, and DGPSO for function $f_8$

6 References


**TABLE 1.** Comparison results of GPSO, LPSO, DMS_PSO, PSO_DCIW and DGPSO for 20 particles of 30 dimensions in 3000 iterations

<table>
<thead>
<tr>
<th>Test Function</th>
<th>GPSO</th>
<th>LPSO</th>
<th>DMS_PSO</th>
<th>PSO_DCIW</th>
<th>DGPSO</th>
<th>P-value, Improvement(%) of DGPSO with GPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean of Best Fitness</td>
<td>Std Dev</td>
<td>Mean of Best Fitness</td>
<td>Std Dev</td>
<td>Mean of Best Fitness</td>
<td>Std Dev</td>
</tr>
<tr>
<td>f1</td>
<td>0.149392</td>
<td>0.250608</td>
<td>0.010271</td>
<td>0.010999</td>
<td><strong>0.008371</strong></td>
<td>0.008079</td>
</tr>
<tr>
<td>f2</td>
<td>0.042299</td>
<td>0.167624</td>
<td>0.000407</td>
<td>0.000231</td>
<td>2.61e-06</td>
<td>1.79e-06</td>
</tr>
<tr>
<td>f3</td>
<td>40.8738</td>
<td>24.83168</td>
<td>26.59679</td>
<td>0.415812</td>
<td>26.27856</td>
<td>1.46758</td>
</tr>
<tr>
<td>f4</td>
<td>7084.364</td>
<td>770.4405</td>
<td>5307.758</td>
<td>662.3028</td>
<td>5031.983</td>
<td>828.1021</td>
</tr>
<tr>
<td>f5</td>
<td>0.056126</td>
<td>0.090293</td>
<td>0.049928</td>
<td>0.011532</td>
<td>0.018584</td>
<td>0.005755</td>
</tr>
<tr>
<td>f6</td>
<td>1.159241</td>
<td>1.399235</td>
<td>0.607834</td>
<td>0.779671</td>
<td>0.00709</td>
<td>0.024057</td>
</tr>
<tr>
<td>f7</td>
<td>0.875644</td>
<td>3.516198</td>
<td>6.75e-05</td>
<td>5.7r-05</td>
<td>4.18e-08</td>
<td>6.9r-08</td>
</tr>
<tr>
<td>f8</td>
<td>22.74307</td>
<td>4.910085</td>
<td>43.22946</td>
<td>8.51089</td>
<td>20.49267</td>
<td>4.969791</td>
</tr>
<tr>
<td>f9</td>
<td>22.32039</td>
<td>4.791212</td>
<td>42.08557</td>
<td>8.236476</td>
<td>21.55</td>
<td>4.773557</td>
</tr>
<tr>
<td>f10</td>
<td>2.255667</td>
<td>0.018616</td>
<td>2.129218</td>
<td>0.143674</td>
<td>1.411243</td>
<td>0.173116</td>
</tr>
</tbody>
</table>
### TABLE 2. Benchmark Functions used in our experimental study

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Function Definition</th>
<th>Range</th>
<th>Optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Griewank Function</td>
<td>$f_i(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i+1}}) + 1$</td>
<td>[-600,600]</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel function 2.22</td>
<td>$f_2(x) = \sum_{i=0}^{n-1} x_i + \prod_{i=0}^{n-1} \left</td>
<td>x_i \right</td>
<td>$</td>
</tr>
<tr>
<td>Rosenbrock Function</td>
<td>$f_3(x) = \sum_{i=0}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$</td>
<td>[-30,30]</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel Function</td>
<td>$f_4(x) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{</td>
<td>x_i</td>
<td>})$</td>
</tr>
<tr>
<td>Noisy Function</td>
<td>$f_5(x) = (\sum_{i=1}^{n} (i+1)x_i^4) + \text{rand}[0, 1]$</td>
<td>[-1.28,1.28]</td>
<td>0</td>
</tr>
<tr>
<td>Ackley Function</td>
<td>$f_6(x) = 20 + e - 20 \exp(-0.2 \left( \frac{1}{n} \sum_{i=1}^{n} x_i^2 \right) - \exp(-\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)))$</td>
<td>[-32,32]</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel function 1.2</td>
<td>$f_7(x) = \sum_{i=1}^{n} \left( \sum_{j=1}^{n} x_{ij} \right)^2$</td>
<td>[-100,100]</td>
<td>0</td>
</tr>
<tr>
<td>Rastrigin Function</td>
<td>$f_8(x) = \sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i))^2 + 10$</td>
<td>[-5.12,5.12]</td>
<td>0</td>
</tr>
<tr>
<td>Noncontinuous Rastrigin Function</td>
<td>$f_9(x) = \sum_{i=1}^{n} \left[ x_i^2 - 10 \cos(2\pi x_i) + 10 \right]$</td>
<td>[-5.12,5.12]</td>
<td>0</td>
</tr>
</tbody>
</table>

### TABLE 3. Comparison results of GPSO, LPSO, DMS_PSO, PSO_DCIW and DGPSO for 50 particles of 50 dimensions in 10000 iterations

<table>
<thead>
<tr>
<th>Test Function</th>
<th>Mean of Best Fitness (GPSO)</th>
<th>Std Dev (GPSO)</th>
<th>Mean of Best Fitness (LPSO)</th>
<th>Std Dev (LPSO)</th>
<th>Mean of Best Fitness (DMS_PSO)</th>
<th>Std Dev (DMS_PSO)</th>
<th>Mean of Best Fitness (PSO_DCIW)</th>
<th>Std Dev (PSO_DCIW)</th>
<th>Mean of Best Fitness (DGPSO)</th>
<th>Std Dev (DGPSO)</th>
<th>P-value (DGPSO)</th>
<th>Imp. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>0.171684</td>
<td>0.343501</td>
<td>5.07e-06</td>
<td>1.18e-05</td>
<td>0.011824</td>
<td>0.011451</td>
<td>0.026426</td>
<td>0.037654</td>
<td>0.000863</td>
<td>0.002685</td>
<td>0.167035</td>
<td>99.5%</td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.022719</td>
<td>0.083965</td>
<td>5.92e-07</td>
<td>3.42e-07</td>
<td>7.42e-11</td>
<td>4.84e-11</td>
<td>0.000197</td>
<td>0.000102</td>
<td>1.07e-20</td>
<td>1.75e-20</td>
<td>1.31e-02</td>
<td>100%</td>
</tr>
<tr>
<td>$f_3$</td>
<td>87.84807</td>
<td>53.03229</td>
<td>45.0903</td>
<td>39.2029</td>
<td>44.42762</td>
<td>1.419095</td>
<td>94.80813</td>
<td>40.12472</td>
<td>12.97845</td>
<td>6.417684</td>
<td>2.59e-08</td>
<td>85.23%</td>
</tr>
<tr>
<td>$f_4$</td>
<td>12610.62</td>
<td>1265.407</td>
<td>8427.603</td>
<td>587.5423</td>
<td>8654.23</td>
<td>737.3883</td>
<td>10831.61</td>
<td>1234.096</td>
<td>4780.807</td>
<td>438.2983</td>
<td>1.98e-21</td>
<td>62.09%</td>
</tr>
<tr>
<td>$f_5$</td>
<td>0.144703</td>
<td>0.016286</td>
<td>0.094603</td>
<td>0.019196</td>
<td>0.028865</td>
<td>0.006778</td>
<td>0.071137</td>
<td>0.013531</td>
<td>0.000553</td>
<td>0.000203</td>
<td>1.90e-10</td>
<td>99.62%</td>
</tr>
<tr>
<td>$f_6$</td>
<td>4.910699</td>
<td>4.209546</td>
<td>0.092147</td>
<td>0.309743</td>
<td>1.45e-06</td>
<td>2.16e-06</td>
<td>0.029624</td>
<td>0.117226</td>
<td>2.56e-14</td>
<td>3.15e-15</td>
<td>5.14e-19</td>
<td>100%</td>
</tr>
<tr>
<td>$f_7$</td>
<td>11.31171</td>
<td>20.10893</td>
<td>1.03e-08</td>
<td>5.07e-09</td>
<td>7.54e-15</td>
<td>1.32e-14</td>
<td>9.15e-07</td>
<td>1.26e-06</td>
<td>1.47e-33</td>
<td>2.14e-33</td>
<td>0.006363</td>
<td>100%</td>
</tr>
<tr>
<td>$f_8$</td>
<td>28.93487</td>
<td>7.574004</td>
<td>62.79516</td>
<td>11.89742</td>
<td>21.39162</td>
<td>4.705682</td>
<td>36.22156</td>
<td>11.4106</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$f_9$</td>
<td>30.34149</td>
<td>6.003359</td>
<td>65.30312</td>
<td>9.873662</td>
<td>24.65</td>
<td>5.751659</td>
<td>43.6</td>
<td>11.91814</td>
<td>1.4</td>
<td>3.574766</td>
<td>0.095997</td>
<td>95.39%</td>
</tr>
<tr>
<td>$f_{10}$</td>
<td>2.787375</td>
<td>0.019692</td>
<td>2.576146</td>
<td>0.117281</td>
<td>1.898071</td>
<td>0.180956</td>
<td>2.169315</td>
<td>0.750711</td>
<td>0.256608</td>
<td>0.33295</td>
<td>2.70e-03</td>
<td>90.79%</td>
</tr>
</tbody>
</table>