Particle swarm optimisation with adaptive neighbourhood search for solving multi-objective optimisation problems

Layak Ali*
School of Engineering,
Central University of Karnataka,
Gulbarga, Karnataka – 585311, India
Email: informlayak@gmail.com
Email: layakali@cuk.ac.in
*Corresponding author

Samrat L. Sabat
Center for Advanced Studies in Electronics Science and Technology,
School of Physics,
University of Hyderabad,
Hyderabad – 500046, India
Email: slssp@uohyd.ernet.in

Abstract: This paper proposes a new variant of PSO algorithm with adaptive neighbourhood search of Pareto solutions to solve multi-objective optimisation problems. This algorithm essentially consists of identifying the particles responsible for worst solution and accelerating these particles towards the best solution in an adaptive manner. This concept is hybridised with the updation strategy of non-dominated solution using crowding distance. Performance of the proposed algorithm is demonstrated by solving standard multi-objective benchmark problems including CEC 2009. In particular we compare our method with two recent algorithms by considering the inverted generational distance, spacing, convergence and diversity as the performance metrics. The proposed algorithm is also compared with recent state of art methods in terms of inverted generational distance. The results demonstrate that the proposed algorithm is quite competitive in terms of the mentioned performance indicators as compared to other algorithms.

Keywords: multi-objective optimisation; Pareto optimality; particle swarm optimisation; PSO.

Reference to this paper should be made as follows: Ali, L. and Sabat, S.L. (2016) ‘Particle swarm optimisation with adaptive neighbourhood search for solving multi-objective optimisation problems’, Int. J. Swarm Intelligence, Vol. 2, No. 1, pp.1–42.

Biographical notes: Layak Ali received his PhD in Electronics Science from the University of Hyderabad in 2012. He obtained his Master’s in Computer Science and Bachelor in Electronics and Communication Engineering during 2005 and 1999 respectively. Presently, he is working as an Assistant Professor of Electronics and Communication Engineering in the School of Engineering, Central University of Karnataka. His research areas include reactive power compensation using swarm intelligence, global optimisation, swarm intelligence and robots for agricultural sector.
Samrat L. Sabat received his PhD in Electronics Science from Berhampur University, India, in 2004. He currently is an Associate Professor in the Center for Advanced Studies in Electronics Science and Technology, School of Physics with the University of Hyderabad, India. His research interests include signal processing algorithm for cognitive radio, sensor signal processing, swarm intelligence techniques and its applications, and the hardware implementation of signal processing algorithms.

This paper is a revised and expanded version of a paper entitled ‘Adaptive accelerated exploration particle swarm optimizer for global multimodal functions’ presented at Proceedings of IEEE World Congress on Nature & Biologically Inspired Computing, NaBIC 2009, 9–11 December 2009.

1 Introduction

Most of the real world optimisation problems demand efficient algorithms that can simultaneously optimise conflicting objectives (Silva et al., 2008; Agrawal et al., 2008; Herrero et al., 2009; Liu et al., 2010). Several evolutionary algorithms such as VEGA (Schaffer, 1985), linear (Fonseca and Fleming, 1995) and nonlinear (Coello Coello and Christiansen, 1998) aggregating methods have been proposed to solve such multi-objective optimisation problems (MOPs). In solving MOPs, a set of trade-off solutions called Pareto optimal solutions that has a correspondence with a set in objective space known as Pareto front (Goldberg, 1989; Srinivas and Deb, 1994; Horn et al., 1994; Zitzler et al., 2001, 2002; Deb and Pratap, 2002) is obtained. Pareto-based methods are popularly being used in solving MOPs. In the recent past bioinspired algorithms are also used in solving MOPs (Wang and Yang, 2009; Yang et al., 2009; Coello et al., 2004; Chen et al., 2010; Yang and Deb, 2013; Venske et al., 2014; Zhou et al., 2011; Long, 2014). Particle swarm optimisation (PSO) (Eberhart and Kenedy, 1995a) is one of the bioinspired technique that has been successfully applied to solve single objective optimisation problems (Sabat et al., 2009a, 2009b). PSO is popular because of its low algorithmic complexity and fast convergence. In the recent past PSO is used together with non-dominated sorting (Parsopoulos and Vrahatis, 2002), Pareto optimal searching (Yang et al., 2009), Pareto dominance ranking (Wang and Yang, 2009) and competitive co-evolution (Goh et al., 2010) to solve MOPs.

The basic PSO algorithm suffers from premature convergence while solving complex multiple objectives. Improper selection of personal best (pbest) and global best (gbest) among the Pareto solutions is also responsible for its poor performance. Multi-objective comprehensive learning particle swarm optimisation (MOCLPSO) (Huang et al., 2006) is an improvement over PSO and performs better than multi-objective self-adaptive differential evolution (MOSaDE) for solving MOPs. MOSaDE has also been proposed and investigated in Huang et al. (2009) for solving MOPs.

In this work, we propose a new strategy for the PSO to avoid premature convergence and improve the quality by solving different instances of MOPs. In the proposed approach, the set of particles with inferior solutions are identified and accelerated towards the current best solution. During the initial phase, the particle will have higher search patch size to search around the best particle and the patch size decreases exponentially with iteration. This approach helps to control the diversity measure of the
swarm. In the proposed approach, we have employed this adaptive, accelerated exploration strategy with non-dominated sorting concept to solve multi-objective problems, and named the approach as multi-objective adaptive accelerated exploration particle swarm optimisation (MOAAEPSO). The archive size is maintained based on the crowding distance as in Deb and Pratap (2002). Thus, the MOAAEPSO algorithm has the following attractive features.

1. Integrates Pareto optimality and uses bounded external archive to store the non-dominated solutions.
2. Identifies and accelerates diverged particles towards the best solution.
3. Explores adaptive neighbourhood patch around the best solution.

The remainder of the paper is organised as follows: in Section 2, brief description of MOP and concept of Pareto optimal front concept is given. Section 3 describes an overview of PSO algorithm and state of the art techniques. Section 4 describes the proposed MOAAEPSO algorithm. Section 5 presents the simulation setup adopted for performance comparison. Section 6 describes simulation results followed by conclusion in Section 7.

2 Multi-objective optimisation

A MOP essentially minimises

\[ f(x) = [f_1(x), f_2(x), f_3(x), \ldots, f_k(x)] \]

where \( x = (x_1, x_2, x_n) \in R \) is an \( n \)-dimensional decision vector, \( R \) is the decision space and \( k \) is the number of objective functions. MOP involves solving a set of conflicting objectives simultaneously. Hence, there can be a set of optimal solutions to the given problem. The trade-offs amongst them is defined in terms of Pareto optimality.

2.1 Pareto optimality

\( u \in R \) is said to dominate \( v \in R \), denoted by \( u \prec v \), iff \( \forall i \in 1, 2 \ldots k: f_i(u) \leq f_i(v) \) and \( \exists j \in 1, 2 \ldots k: f_j(u) < f_j(v) \). A decision vector \( x \in R \) is said to be Pareto optimal with respect to \( R \) iff there is no other decision vector that dominates \( x \) in \( R \). The set of all Pareto optimal solutions in the decision space is termed as Pareto optimal set and the corresponding set of objective vector is termed as Pareto optimal front. The aim of multiobjective optimisation algorithm is to obtain Pareto optimal front accurately.

3 Particle swarm optimisation

PSO is a decade old population-based Swarm intelligence algorithm inspired by the food foraging behaviour of birds (Eberhart and Kenedy, 1995b; Robinson and Rahmat-Samii, 2004). Each individual (particle) in the swarm represents a potential solution in \( n \)-dimensional search space. The particles are initialised with velocities and positions randomly and independently within a search range. During the search process, each
particle remembers its own so-far best position known as (pbest); the so-far-best position of whole swarm known as (gbest). The velocity (V) and position (X) of each particle are updated as

\[
V_{t+1} = V_t + c_1 \xi_1 \left( X_{pbest} - X_t \right) + c_2 \xi_2 \left( X_{gbest} - X_t \right),
\]

\[
X_{t+1} = X_t + V_{t+1},
\]

where \(c_1\) and \(c_2\) are cognitive and social component respectively. \(\{\xi_i\}\) are independent variables and randomly sampled from a uniform distribution in the range (0, 1). To increase the search capability the basic PSO equation may be incorporated with inertia weight. The inertia weight, cognitive and social components can be varied as in Sabat et al. (2009a) and is given below.

\[
w', c_1', c_2' = w_{\text{max}} - \left( \frac{(w_{\text{max}} - w_{\text{min}})}{\text{Maxgen}} \right) t \]

where \(w_{\text{max}} = 0.9, w_{\text{min}} = 0.2\) is iteration.

The basic PSO equation now becomes as follows

\[
V'^{t+1}_{i,d} = w'V'^t_{i,d} + c_1' \cdot \text{rand} \cdot \left( \text{pbest'}_{i,d} - X'^t_{i,d} \right) + c_2' \cdot \text{rand} \cdot \left( \text{gbest'}_{i,d} - X'^t_{i,d} \right)
\]

\[
X'^{t+1}_{i,d} = X'^t_{i,d} + V'^{t+1}_{i,d}
\]

### 3.1 Multi-objective comprehensive learning particle swarm optimisation (MOCLPSO)

In the classical comprehensive learning particle swarm optimisation (CLPSO) velocity of a particle is updated on the basis of gbest of the Swarm, pbest of the particle, and the pbests of some other random particle (Liang et al., 2004). Huang et al. (2006) extended CLPSO to multi-objective optimisation and named as MOCLPSO. MOCLPSO incorporates non-dominated sorting, bounded external archive and updates particle position and velocity as in CLPSO.

### 3.2 Multi-objective self-adaptive differential evolution (MOSaDE)

Differential evolution (DE) is one of the popular evolutionary algorithm. In DE, a trial vector is generated with a predefined strategy; the parameters of the vector are tuned by trial and error. In this classical DE, tuning of the control parameters is a tedious task and require an additional effort. To address this self-adaptive DE was proposed that avoids tuning by trial and error (Huang et al., 2009). Self-adaptive DE has been extended to solve MOPs (Huang et al., 2009).

### 4 Multi-objective adaptive accelerated exploration particle swarm optimisation (MOAAEPSO)

In the recent past, evolutionary algorithms have been applied to solve MOPs. Many of these focus on the tuning and ordering of the Pareto optimal fronts in combination with the basic evolutionary algorithm. Not much attention has been paid towards to improve the PSO and use it to solve MOPs. This could be due to the fact that basic PSO suffers
from premature convergence for complex objectives. To overcome this problem an accelerated exploration particle swarm optimiser (AAEPSO) is proposed for complex single objective problems (Sabat et al., 2009a). The inertia weight, social and cognitive factor for MOAAEPSO are varied as per Algorithm 1. The key features of this algorithm is to identify the particles that have inferior solutions that do not improve their solution for many iteration, and accelerate them in the direction of the best particle in the Swarm (i.e. the global solution). This helps the particles to escape the deep local minima in the search space. These particles are available for searching better solution. This mechanism is carried out at the end of every tenth (randomly chosen) iteration to maintain a balance between exploration and exploitation. In this work, we present MOAAEPSO as an improved variant of PSO for solving multi-objective problems. The three major steps of the proposed algorithm are selection, acceleration and exploration coupled with non-dominated sorting.

Algorithm 1 MOAAEPSO

Initialisation
Initialise the swarm of size NP:
Initialise position \( X \) and velocity \( V \) of the particles randomly in \( D \)-dimensional search range \( (X_{\text{min}}, X_{\text{max}}) \).
Initialise selection factor \( S_f \) for worst particles with random permutation in \([10-90]\).
Evaluate the fitness values of all particles.
Set the current position as \( p_{\text{best}} \) and the particle with the best fitness value as \( g_{\text{best}} \).
Initialise the bounded archive size to 100
Set \( \text{count}_{A_i} = 0 \)

Optimise
for \( t \leftarrow 1 \), to Maxgen do

\[
\begin{align*}
  w^t, c_1^t, c_2^t &= w_{\text{max}} - \left( \left( w_{\text{max}} - w_{\text{min}} \right) / \text{Maxgen} \right) \times t \\
  \text{where } w_{\text{max}} &= 0.9, w_{\text{min}} &= 0.2
\end{align*}
\]  

Decrease \( AE_f \) (acceleration and exploration factor) exponentially, equation (1).
Update velocity and position of each particle as

\[
\begin{align*}
  V^t_{i,d} &= w \cdot V^t_{i,d} + c_1^t \cdot \text{rand}_1 \cdot (p_{\text{best}}_{i,d} - X^t_{i,d}) + c_2^t \cdot \text{rand}_2 \cdot (g_{\text{best}}_{i,d} - X^t_{i,d}) \\
  X^t_{i,d} &= X^t_{i,d} + V^t_{i,d} 
\end{align*}
\]  

Evaluate fitness function for each particle.
Update \( p_{\text{best}} \): If current fitness dominates the previous then set current position as \( p_{\text{best}} \) else retain \( p_{\text{best}} \).
Update \( g_{\text{best}} \): If best of all the current \( p_{\text{best}} \) dominates the previous \( g_{\text{best}} \) then set best \( p_{\text{best}} \) as \( g_{\text{best}} \) else retain \( g_{\text{best}} \).
Find the Pareto-optimal solutions.
Check the domination status of new solution with the solutions in the archive.
if \( X \) dominates any member of Archive then
Delete the dominated member and insert \( X \) in the archive
else if \( X \) is neither dominated by the members of archive nor by the members by \( X \) then

\[
\begin{align*}
  X^t_{i,d} &= X^t_{i,d} + V^t_{i,d} \\
  X^t_{i,d} &= X^t_{i,d} + V^t_{i,d} \\
  X^t_{i,d} &= X^t_{i,d} + V^t_{i,d} \\
  X^t_{i,d} &= X^t_{i,d} + V^t_{i,d} \\
\end{align*}
\]
Insert X in the archive
else if X is dominated by the members of archive then
    reject X
end if
if size of archive exceeds maximum size then
    use crowding distance to eliminate less dominated particles.
end if
if Mod(t, 10) = 1 then
    Find distance of each particle from gbest.
    Sort the particles based on their fitness value and distance from gbest.
    Select \( S_f \) number of worst particles from sorted particles.
    Reinitialise these selected particles around gbest using \( AE_f \) as
    \[
    count_{si} = count_{si} + 1
    \]
    \[
    a = gbest - A_f(count_{si}) \text{ and } b = gbest + A_f(count_{si})
    \]
    for \( i \leftarrow 1, S_f \) do
        for \( d \leftarrow 1, D \) do
            \[
            X_{i,d} = a_d + (b_d - a_d) \ast \text{rand}
            \]
        end for
    end for
end if
continue optimising until stopping criteria or exceeding maximum iteration
Report results
Terminate

4.1 Selection

In this phase, MOAAEPSO select the particles that do not improve their solution. These are called as the worst particles. They are defined as the particles
1. with poor fitness value
2. that do not contribute to improve the search process,
which may be because they are trapped in local minima. These are also termed as diverged particles. The diverged particles do not participate in further searching.

4.2 Acceleration

MOAAEPSO algorithm selects a certain fraction of the selected population as diverged/inferior particles and allow to search them near to the converged particle (gbest). The selected diverged particles are accelerated towards gbest particle. The acceleration factor is calculated based on the Euclidian distance between the gbest and the position of diverged particle. It is quite possible that gbest may also get trapped in
local minima during the search process, resulting to poor convergence. To avoid this, the chosen diverged particles are initialised with random positions in the vicinity of $\text{gbest}$.

4.3 Exploration

The accelerated diverged particles are allowed to explore a patch around the $\text{gbest}$. It is assumed that around the best particle position, better solutions are present. Thus the particles which were not participating in the search process are now eligible to search and explore the region around best particle of the swarm. To maintain the diversity, MOAAEPSONO declares a pre determined fraction of the particles as diverged. With iteration, the whole population tend to converge towards global solution; hence the neighbourhood patch size is decreased exponentially in an adaptive manner with increase of iteration. In MOAAEPSONO the exploration region of the diverge particles decreases exponentially with iteration. At the beginning of search process the patch size is set as the search range of the problem; with increase in iteration, the search range decays as

$$AE_j^t = R_{\text{max}} \cdot e^{-\frac{t}{u}}$$

where $u = 10$ a constant and $R_{\text{max}}$ is the maximum search range of the problem. The decay in exploration patch size is shown in Figure 1. This strategy gives complete freedom to the particles to explore the search space independently around the best particle. The complete process of selection, acceleration and exploration is shown in Figure 2. Here, only two particles out of ten are selected as diverged particles. The small empty circle inside the big one shows the position of diverged particles after acceleration; the big circle denotes the patch size. The MOAAEPSONO is a Pareto-based algorithm that maintains an external archive to store non-dominated solutions. The two solutions are said to be non-dominated if none of them has less value in all objectives (in a minimisation problem). The external archive is updated as per the strategy shown in Algorithm 1.

**Figure 1** Exponential decrease of exploration region
Figure 2  The search process (exploration region) around global solution

5 Simulation

The simulations are carried out using Pentium Core2Duo, 2 GHz with 2 GB RAM. Algorithms are coded in Matlab 7.2 in Windows XP platform.

5.1 Benchmark functions

The performance of the proposed MOAAEPSO algorithm is compared with MOSaDE (Huang et al., 2009) and MOCLPSO (Huang et al., 2006) algorithms on seventeen different benchmark functions which include CEC 2009 complex test problems (Zhang et al., 2009). These functions have different characteristics in terms of convexity, discontinuity and non-uniformity (Zhang et al., 2009). The KUR, FUN, SCH, ZDT1-ZDT4 and UF1UF7 problems are unconstrained two objectives test problems, while UF8UF10 are unconstrained three objectives test problems. The mathematical representation of these problems is given in Appendix.

5.1.1 Performance metric

A set of performance metric (Huang et al., 2006) is used to measure the efficiency of algorithms.
Inverted generational distance metric (IGD): IGD is used to estimate the closeness of elements in dominated solutions with the true Pareto optimal set. It is defined as (Huang et al., 2006):

\[
GD = \frac{\sum_{i=1}^{n} d_i^2}{n}
\]

where \(n\) is number of non-dominated solutions generated by the algorithm and \(d_i\) is the Euclidean distance between the elements in Pareto optimal set found so far (by the algorithm) and its nearest neighbour in known Pareto optimal solution. A smaller IGD value indicates better performance.

Spacing (S): Spacing measures the range variance of the neighbouring solutions in the Pareto front obtained from the algorithm. It is defined as (Huang et al., 2006):

\[
S = \sqrt{\frac{1}{(n-1)} \sum_{i=1}^{n} (\overline{d} - d_i)^2}
\]

where \(n\) is the number of non-dominated solutions generated by the algorithm and 

\[
d_i = \min\left(\sum_{k=1}^{m} f_k^d(\bar{x}) - f_k^d(\bar{x})\right) \quad i, j = 1, 2, \ldots, n
\]

and \(\overline{d}\) is the mean distance of all \(d_i\). \(m\) is the number of objectives. A zero value for this metric indicates that all elements of Pareto fronts are equi-spaced.

Convergence metric (CM): The CM measures the extent of convergence of Pareto solutions to the known optimal Pareto front. It is defined as (Huang et al., 2006):

\[
CM = \frac{1}{n} \sum_{i=1}^{n} d_i
\]

where \(n\) is the number of non-dominated solutions obtained by the algorithm and \(d_i\) is the Euclidean distance (in objective space) between the \(i^{th}\) non-dominated solution and the nearest member of the known Pareto optimal front. A smaller value of CM denotes a better convergence performance.

Spread or diversity metric (DM): The DM measures the extent of spread among the solutions. It is defined as (Huang et al., 2006):

\[
DM = \frac{\sum_{m=1}^{M} d_{m}^2 + \sum_{i=1}^{n-1} |d_i - \overline{d}|}{\sum_{m=1}^{M} d_{m}^2 + (n-1)\overline{d}}
\]

where \(d_m\) is the Euclidean distance between the two extreme solutions of Pareto optimal front and the boundary solutions of the non-dominated solutions corresponding to \(m^{th}\) objective function. \(d_i\) is the Euclidean distance between \(i^{th}\) non-dominated solution and corresponding true Pareto front. \(\overline{d}\) is the mean value of
all $d_r$. Zero value indicates an ideal distribution. Smaller DM indicates better diversity of the non-dominated set.

5 Convergence graphs: The convergence characteristics of optimising algorithms can be found out using Pareto front graphs. If the obtained Pareto front of the algorithm is closer to the true Pareto front, then algorithm is said to converged.

6 Experimental results

This section compares the performance of the proposed algorithm with MOSaDE and MOCLPSO on a set of benchmark functions (Appendix). The algorithmic parameters for MOSaDE are as Huang et al. (2009). The algorithmic parameters for MOCLPSO are as follows: population size ($NP$) = 50, archive size ($A$) = 100, learning probability ($P_c$) = 0.1, and elitism probability ($P_m$) = 0.4. The experimental results are the average of 25 independent runs where each run is of 1,000 iterations. The results are presented in two different form; numerical and graphical. Table 1 and Table 2 gives numerical results for different metrics (IGD, S, CM and DM). Figures 3 to 19 represents convergence graphs of different test bench functions. The comprehensive comparisons among optimising algorithms are done under the following headings.

Table 1 Results for MO functions: mean(std)

<table>
<thead>
<tr>
<th></th>
<th>IGD</th>
<th>Convergence metric</th>
<th>Spacing</th>
<th>Diversity metric (spread)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KUR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPSO</td>
<td>1.09e-3(2.19e-4)</td>
<td>1.63e-2(1.33e-3)</td>
<td>3.75e-2(4.87e-3)</td>
<td>1.32(3.73e-4)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>2.89e-3(1.81e-3)</td>
<td>4.08e-2(1.95e-2)</td>
<td>4.14e-2(9.74e-3)</td>
<td>1.33(1.83e-4)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.09e-3(2.17e-3)</td>
<td>4.98e-2(5.79e-3)</td>
<td>8.85e-2(1.31e-2)</td>
<td>1.32(2.30e-5)</td>
</tr>
<tr>
<td>FON</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPSO</td>
<td>1.96e-3(1.63e-4)</td>
<td>1.10e-2(8.70e-4)</td>
<td>1.28e-2(1.09e-3)</td>
<td>7.94e-1(2.36e-4)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>2.08e-3(1.97e-4)</td>
<td>1.13e-2(1.24e-3)</td>
<td>1.36e-2(1.75e-3)</td>
<td>7.94e-1(6.94e-5)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.09e-3(1.09e-005)</td>
<td>1.24e-2(8.49e-005)</td>
<td>9.77e-2(5.84e-005)</td>
<td>2.61(1.48e-002)</td>
</tr>
<tr>
<td>SCH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPSO</td>
<td>8.67e-4(4.96e-5)</td>
<td>9.71e-4(3.35e-4)</td>
<td>1.28e-3(1.10e-3)</td>
<td>3.70e-1(2.26e-4)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>9.05e-4(5.07e-5)</td>
<td>1.01e-2(5.99e-4)</td>
<td>1.31e-2(5.77e-4)</td>
<td>3.73e-1(1.14e-4)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>6.35e-3(1.39e-4)</td>
<td>2.84e-3(3.06e-4)</td>
<td>3.71e-3(5.90e-4)</td>
<td>2.46(4.33e-2)</td>
</tr>
<tr>
<td>ZDT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPSO</td>
<td>4.69e-5(8.83e-7)</td>
<td>8.62e-4(2.10e-5)</td>
<td>1.24e-3(3.68e-5)</td>
<td>2.74e-1(2.67e-4)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>5.37e-5(9.02e-7)</td>
<td>1.07e-3(1.99e-5)</td>
<td>1.14e-3(6.56e-5)</td>
<td>2.78e-1(4.22e-4)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.30e-2(2.19e-3)</td>
<td>5.12e-1(4.85e-2)</td>
<td>1.27e-1(9.86e-2)</td>
<td>7.51e-1(4.21e-2)</td>
</tr>
<tr>
<td>ZDT2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPSO</td>
<td>7.71e-2(1.08e-1)</td>
<td>1.72(2.41)</td>
<td>8.21e-3(4.71e-4)</td>
<td>8.80e-1(5.41e-2)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>5.20e-5(1.79e-6)</td>
<td>1.00e-3(3.63e-5)</td>
<td>1.21e-3(4.69e-5)</td>
<td>2.26e-1(8.20e-6)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>4.77e-2(3.33e-3)</td>
<td>1.05(7.38e-2)</td>
<td>1.28e-1(9.91e-2)</td>
<td>7.53e-1(2.80e-2)</td>
</tr>
</tbody>
</table>
Table 1  Results for MO functions: mean(std) (continued)

<table>
<thead>
<tr>
<th></th>
<th>IGD</th>
<th>Convergence metric</th>
<th>Spacing</th>
<th>Diversity metric (spread)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>3.92e-4(2.71e-5)</td>
<td>3.64e-3(2.97e-4)</td>
<td>5.17e-3(8.06e-4)</td>
<td>6.89e-1(9.89e-5)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>4.00e-4(1.88e-5)</td>
<td>3.88e-3(1.51e-4)</td>
<td>4.84e-3(4.05e-4)</td>
<td>6.88e-1(4.86e-5)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>5.77e-2(3.30e-3)</td>
<td>6.51e-1(3.39e-2)</td>
<td>9.45e-2(3.52e-2)</td>
<td>8.62e-1(1.06e-2)</td>
</tr>
<tr>
<td>ZDT4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>6.63e-3(2.03e-2)</td>
<td>8.59e-2(2.64e-2)</td>
<td>2.73e-1(8.51e-1)</td>
<td>1.22(5.26e-2)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>4.15e-2(5.24e-2)</td>
<td>5.54e-1(7.26e-1)</td>
<td>8.93e-1(8.54e-1)</td>
<td>1.22(1.65e-2)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>6.58e-2(9.10e-3)</td>
<td>9.23e-1(1.26e-1)</td>
<td>2.21(2.47)</td>
<td>1.15(4.38e-2)</td>
</tr>
</tbody>
</table>

Table 2  Results for MO functions: mean(std)

<table>
<thead>
<tr>
<th></th>
<th>IGD</th>
<th>Convergence metric</th>
<th>Spacing</th>
<th>Diversity metric (spread)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>9.73e-3(9.22e-3)</td>
<td>2.69e-1(2.66e-1)</td>
<td>1.46e-2(2.86e-2)</td>
<td>4.11e-1(1.38e-1)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>4.82e-3(2.20e-3)</td>
<td>1.35e-1(5.63e-2)</td>
<td>3.42e-2(1.58e-2)</td>
<td>4.40e-1(8.91e-2)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.12e-2(3.02e-4)</td>
<td>8.41e-1(9.23e-2)</td>
<td>7.05e-2(1.69e-2)</td>
<td>5.01e-1(6.73e-2)</td>
</tr>
<tr>
<td>UF2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>2.10e-3(7.30e-5)</td>
<td>6.54e-2(2.41e-3)</td>
<td>1.52e-2(8.62e-3)</td>
<td>3.98e-1(4.87e-2)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>5.10e-3(3.09e-4)</td>
<td>1.54e-1(1.01e-2)</td>
<td>9.46e-2(3.02e-2)</td>
<td>4.49e-1(2.39e-2)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.21e-2(4.23e-4)</td>
<td>3.02e-1(1.08e-1)</td>
<td>1.19e-2(1.25e-2)</td>
<td>5.59e-1(5.98e-2)</td>
</tr>
<tr>
<td>UF3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>1.62e-2(5.1e-3)</td>
<td>4.67e-1(1.6e-1)</td>
<td>1.0(0.23)</td>
<td>9.07e-1(1.02e-1)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>1.81e-2(4.23e-4)</td>
<td>5.70e-1(1.33e-2)</td>
<td>1.25e-1(8.08e-2)</td>
<td>6.64e-1(4.13e-2)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>1.42e-2(1.70e-4)</td>
<td>4.47e-1(5.0e-3)</td>
<td>3.63e-2(1.75e-2)</td>
<td>5.96e-1(3.43e-2)</td>
</tr>
<tr>
<td>UF4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>9.14e-1(5.87e-1)</td>
<td>2.89e+1(1.86e+1)</td>
<td>9.27e-1(7.03e-2)</td>
<td>9.88e-1(7.45e-3)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>6.63e-3(4.29e-4)</td>
<td>2.06e-1(1.35e-2)</td>
<td>9.26e-2(1.90e-2)</td>
<td>3.79e-1(1.77e-2)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>3.08e-3(5.52e-5)</td>
<td>9.67e-2(1.88e-3)</td>
<td>6.95e-3(1.19e-3)</td>
<td>3.44e-1(9.47e-3)</td>
</tr>
<tr>
<td>UF5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>1.43e-1(5.15e-2)</td>
<td>6.37e-1(2.27e-1)</td>
<td>2.60e-2(5.48e-2)</td>
<td>1.37e-1(2.69e-1)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>3.64e-1(1.28e-1)</td>
<td>1.66(5.89e-1)</td>
<td>2.05e-1(8.70e-2)</td>
<td>7.37e-1(6.84e-2)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>3.84e-1(1.70e-2)</td>
<td>1.76(7.88e-2)</td>
<td>1.91e-1(1.26e-1)</td>
<td>7.69e-1(3.26e-2)</td>
</tr>
<tr>
<td>UF6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOAAEPO</td>
<td>4.56e-3(5.42e-4)</td>
<td>1.18e-1(2.72e-2)</td>
<td>1.02e-1(1.63e-1)</td>
<td>9.96e-1(2.33e-4)</td>
</tr>
<tr>
<td>MOCLPSO</td>
<td>2.25e-2(4.81e-3)</td>
<td>6.82e-1(1.26e-1)</td>
<td>5.33e-2(9.37e-2)</td>
<td>9.98e-1(1.67e-4)</td>
</tr>
<tr>
<td>MOSaDE</td>
<td>2.73e-2(1.47e-3)</td>
<td>8.59e-1(4.48e-2)</td>
<td>1.84e-1(1.44e-1)</td>
<td>9.99e-1(2.52e-4)</td>
</tr>
</tbody>
</table>
Table 2 Results for MO functions: mean(std) (continued)

<table>
<thead>
<tr>
<th>Function</th>
<th>IGD</th>
<th>Convergence metric</th>
<th>Spacing</th>
<th>Diversity metric (spread)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF7</td>
<td>MOAEPSO</td>
<td>1.10e-2(2.34e-3)</td>
<td>2.67e-1(5.41e-2)</td>
<td>2.85e-2(3.25e-2)</td>
</tr>
<tr>
<td></td>
<td>MOCLPSO</td>
<td>9.15e-2(7.43e-2)</td>
<td>3.40e-1(1.87e-1)</td>
<td>5.10e-2(3.70e-2)</td>
</tr>
<tr>
<td></td>
<td>MOSaDE</td>
<td>7.21e-2(9.89e-3)</td>
<td>7.07e-1(2.57e-2)</td>
<td>5.18e-2(2.45e-2)</td>
</tr>
<tr>
<td>UF8</td>
<td>MOAEPSO</td>
<td>3.85e-3(8.93e-4)</td>
<td>3.23e-1(7.50e-2)</td>
<td>9.10e-2(6.47e-2)</td>
</tr>
<tr>
<td></td>
<td>MOCLPSO</td>
<td>5.07e-3(1.19e-3)</td>
<td>4.35e-1(9.16e-2)</td>
<td>1.14e-1(3.94e-2)</td>
</tr>
<tr>
<td></td>
<td>MOSaDE</td>
<td>4.39e-3(3.96e-4)</td>
<td>4.17e-1(4.23e-2)</td>
<td>9.63e-2(3.21e-2)</td>
</tr>
<tr>
<td>UF9</td>
<td>MOAEPSO</td>
<td>2.33e-3(4.15e-4)</td>
<td>2.13e-1(3.78e-2)</td>
<td>1.26e-1(2.65e-1)</td>
</tr>
<tr>
<td></td>
<td>MOCLPSO</td>
<td>6.08e-2(5.63e-4)</td>
<td>5.63e-1(5.69e-2)</td>
<td>2.04e-1(1.09e-1)</td>
</tr>
<tr>
<td></td>
<td>MOSaDE</td>
<td>4.83e-2(3.17e-4)</td>
<td>4.65e-1(2.72e-2)</td>
<td>1.69e-1(5.29e-2)</td>
</tr>
<tr>
<td>UF10</td>
<td>MOAEPSO</td>
<td>9.02e-3(1.47e-3)</td>
<td>8.49e-1(1.50e-1)</td>
<td>1.81e-3(2.07e-3)</td>
</tr>
<tr>
<td></td>
<td>MOCLPSO</td>
<td>3.23e-1(3.03e-3)</td>
<td>3.22(3.03e-1)</td>
<td>4.65e-1(1.01e-1)</td>
</tr>
<tr>
<td></td>
<td>MOSaDE</td>
<td>3.92e-1(2.60e-3)</td>
<td>3.92e+0(2.61e-1)</td>
<td>5.37e-1(1.91e-1)</td>
</tr>
</tbody>
</table>

6.1 IGD results

The first column in Table 1 show the results obtained by algorithms for IGD metrics on KUR, FUN, SCH and ZDT1-ZDT4 functions. The best results obtained by the algorithms are shown in bold. The standard deviation of the IGD for 25 independent run is also tabulated in Table 2. From this table, it is evident that the proposed MOAEPSO performs better in terms of IGD on almost all the functions, except ZDT2, where MOCLPSO outperforms. The standard deviation achieved by MOAEPSO is also comparable, except on FON, ZDT2 and ZDT3. The MOSaDE shows more stability on FON, MOCLPSO on ZDT2 and ZDT3 function. IGD results on functions UF1 to UF10 are tabulated in Table 2. From this table it is clear that MOAEPSO algorithm shows better performance on all the functions except UF3 and UF4 where MOSaDE outperforms. The MOAEPSO ceases to give any good results for UF3 function. Table 5 gives the comprehensive comparisons of IGD metrics with the present state-of-the-art algorithms (Reza et al., 2012). This table shows that MOAEPSO is quite competitive with majority of the state-of-the-art techniques.
6.2 CM results

The second column in Table 1 show the results obtained by all the three algorithms for CM metrics on functions $KUR$, $FUN$, $SCH$ and $ZDT1$-$ZDT4$. This table clearly shows that proposed MOAAEPSO gives superior performance for almost all the functions, except $ZDT2$, where MOCLPSO outperforms. Similarly, Table 2 gives CM results on functions $UF1$ to $UF10$. The MOAAEPSO shows better results on $UF2$, $UF5$ to $UF10$, MOSaDE gives good results on $UF3$ and $UF4$, where as MOCLPSO on only $UF1$.

6.3 Spacing results

The third column in Table 1 show the results obtained by all the three algorithms for spacing metrics on functions $KUR$, $FUN$, $SCH$ and $ZDT1$-$ZDT4$. The MOAAEPSO shows good results on almost all the functions except $ZDT1$, $ZDT2$ and $ZDT3$, where MOCLPSO dominates. The stability of the MOAAEPSO is comparable on $KUR$, $ZDT1$ and $ZDT4$, where as MOSaDE on $FON$ and MOCLPSO on all other functions. Table 2 gives spacing results for functions $UF1$ to $UF10$. The proposed MOAAEPSO shows better results on $UF2$, $UF5$ to $UF10$, where as MOSaDE give good results $UF3$ and $UF4$ functions.

6.4 DM results

The fourth column in Table 1 show the DM results on functions $KUR$, $FUN$, $SCH$ and $ZDT1$-$ZDT4$. The MOAAEPSO shows good results on almost all the functions except $ZDT2$, $ZDT3$ and $ZDT4$, where MOCLPSO and MOSaDE dominates. The stability of the MOAAEPSO is comparable only on $FON$ and $ZDT1$. The MOSaDE is stable only on $KUR$ and MOCLPSO on all other functions. The fourth column in Table 2 gives DM results on functions $UF1$ to $UF10$. The MOAAEPSO shows better results on $UF2$, $UF5$ to $UF10$, MOSaDE gives good results on $UF3$ and $UF4$. The algorithmic stability of MOAAEPSO is seen only on $UF7$ and $UF10$, MOSaDE dominates on $UF1$, $UF3$, $UF4$ and $UF5$, otherwise MOCLPSO shows good results.

6.5 Convergence results

Figures 3 to 19 presents the Pareto front generated using the algorithms on 17 benchmark functions including CEC 2009 (Zhang et al., 2009). Symbol ‘o’ represents true Pareto front obtained analytically; ‘*’ denotes the non-dominated solutions obtained by the algorithm. From these figures it is evident that MOAAEPSO finds optimal solutions closer to true Pareto optimal front as compared to MOCLPSO and MOSaDE for all the 17 benchmark functions.
Figure 3 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem KUR
Figure 4  Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem FON
Figure 5  Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem SCH
Figure 6 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem ZDT1
Figure 7 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem ZDT2
Figure 8 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem ZDT3.
Figure 9  Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem ZDT4
Figure 10  Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem UF1.
Figure 11 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem UF2
Figure 12 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem UF3
Figure 13 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem UF4.
Figure 14 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEP SO, on benchmark problem UF5
Figure 15 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPso, on benchmark problem UF6
Figure 16 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem UF7.
Figure 17 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem UF8
Figure 18 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSEO, on benchmark problem UF9.
Figure 19 Pareto fronts obtained by MOSaDE, MOCLPSO and MOAAEPSO, on benchmark problem UF10
The summary of performances of all the three algorithms for all 17 benchmark functions are tabulated in Table 3 and Table 4 respectively. From these tables, it is evident that MOAAEPSO algorithm is a competitive algorithm for solving multi-objective test bench function. works better than other two algorithms for all the benchmark functions except UF3 and UF4 for all the performance metric. In case of UF3 and UF4 functions, MOSaDE algorithm is found to be the best of the three algorithms investigated.

**Table 3** Conclusion of mean results from Table 1

<table>
<thead>
<tr>
<th>Metric</th>
<th>KUR</th>
<th>FON</th>
<th>SCH</th>
<th>ZDT1</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGD</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Convergence metric (CM)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Spacing (S)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOCLPSO</td>
</tr>
<tr>
<td>Divergence metric (DM)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGD</td>
<td>MOCLPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Convergence metric (CM)</td>
<td>MOCLPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Spacing (S)</td>
<td>MOCLPSO</td>
<td>MOCLPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Divergence metric (DM)</td>
<td>MOCLPSO</td>
<td>MOCLPSO</td>
<td>MOSaDE</td>
</tr>
</tbody>
</table>

**Table 4** Conclusion of mean results from Table 2

<table>
<thead>
<tr>
<th>Metric</th>
<th>UF1</th>
<th>UF2</th>
<th>UF3</th>
<th>UF4</th>
<th>UF5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGD</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOSaDE</td>
<td>MOSaDE</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Convergence metric (CM)</td>
<td>MOCLPSO</td>
<td>MOAAEPSO</td>
<td>MOSaDE</td>
<td>MOSaDE</td>
<td>MOCLPSO</td>
</tr>
<tr>
<td>Spacing (S)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOSaDE</td>
<td>MOSaDE</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Divergence metric (DM)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOSaDE</td>
<td>MOSaDE</td>
<td>MOAAEPSO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric</th>
<th>UF6</th>
<th>UF7</th>
<th>UF8</th>
<th>UF9</th>
<th>UF10</th>
</tr>
</thead>
<tbody>
<tr>
<td>IGD</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Convergence metric (CM)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Spacing (S)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
<tr>
<td>Divergence metric (DM)</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
<td>MOAAEPSO</td>
</tr>
</tbody>
</table>
Table 5

Comparison between the state-of-the-art algorithms on mean of inverted generational distance metric

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>UF1</th>
<th>UF2</th>
<th>UF3</th>
<th>UF4</th>
<th>UF5</th>
<th>UF6</th>
<th>UF7</th>
<th>UF8</th>
<th>UF9</th>
<th>UF10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOAAEPSO</td>
<td>0.0097</td>
<td>0.0021</td>
<td>0.016</td>
<td>0.9410</td>
<td>0.1430</td>
<td>0.0046</td>
<td>0.0110</td>
<td>0.0039</td>
<td>0.0023</td>
<td>0.0090</td>
</tr>
<tr>
<td>MOABC</td>
<td>0.00618</td>
<td>0.00484</td>
<td>0.05120</td>
<td>0.05801</td>
<td>0.07758</td>
<td>0.06537</td>
<td>0.03573</td>
<td>0.06726</td>
<td>0.06150</td>
<td>0.19499</td>
</tr>
<tr>
<td>MOEAD</td>
<td>0.00435</td>
<td>0.00679</td>
<td>0.00742</td>
<td>0.06385</td>
<td>0.18071</td>
<td>0.00587</td>
<td>0.00444</td>
<td>0.05840</td>
<td>0.07896</td>
<td>0.47415</td>
</tr>
<tr>
<td>GDE3</td>
<td>0.00534</td>
<td>0.01195</td>
<td>0.10639</td>
<td>0.02650</td>
<td>0.03928</td>
<td>0.25091</td>
<td>0.02522</td>
<td>0.24855</td>
<td>0.08248</td>
<td>0.43326</td>
</tr>
<tr>
<td>MOEADGM</td>
<td>0.00620</td>
<td>0.00640</td>
<td>0.04290</td>
<td>0.04760</td>
<td>1.79190</td>
<td>0.55630</td>
<td>0.00760</td>
<td>0.24460</td>
<td>0.18780</td>
<td>0.5646</td>
</tr>
<tr>
<td>MTS</td>
<td>0.00646</td>
<td>0.00615</td>
<td>0.05310</td>
<td>0.02356</td>
<td>0.01489</td>
<td>0.05917</td>
<td>0.04079</td>
<td>0.11251</td>
<td>0.11442</td>
<td>0.15306</td>
</tr>
<tr>
<td>LiuLi</td>
<td>0.00785</td>
<td>0.01230</td>
<td>0.01497</td>
<td>0.04350</td>
<td>0.16186</td>
<td>0.17555</td>
<td>0.00730</td>
<td>0.08235</td>
<td>0.09391</td>
<td>0.44691</td>
</tr>
<tr>
<td>DMOEADD</td>
<td>0.01038</td>
<td>0.00679</td>
<td>0.03337</td>
<td>0.04268</td>
<td>0.31454</td>
<td>0.06673</td>
<td>0.01032</td>
<td>0.06841</td>
<td>0.04896</td>
<td>0.32211</td>
</tr>
<tr>
<td>NSGAIIILS</td>
<td>0.01153</td>
<td>0.01237</td>
<td>0.10603</td>
<td>0.05840</td>
<td>0.56570</td>
<td>0.31032</td>
<td>0.02132</td>
<td>0.08630</td>
<td>0.07190</td>
<td>0.84468</td>
</tr>
<tr>
<td>OWMOSaDE</td>
<td>0.01220</td>
<td>0.00810</td>
<td>0.10300</td>
<td>0.05130</td>
<td>0.43030</td>
<td>0.19180</td>
<td>0.03850</td>
<td>0.09450</td>
<td>0.09830</td>
<td>0.74300</td>
</tr>
<tr>
<td>ClusMOEA</td>
<td>0.02999</td>
<td>0.02280</td>
<td>0.05490</td>
<td>0.05850</td>
<td>0.24730</td>
<td>0.08710</td>
<td>0.02230</td>
<td>0.23830</td>
<td>0.29340</td>
<td>0.41110</td>
</tr>
<tr>
<td>AMGA</td>
<td>0.03588</td>
<td>0.01623</td>
<td>0.06998</td>
<td>0.04062</td>
<td>0.09405</td>
<td>0.12942</td>
<td>0.05707</td>
<td>0.17125</td>
<td>0.18861</td>
<td>0.32418</td>
</tr>
<tr>
<td>MOEP</td>
<td>0.05960</td>
<td>0.01890</td>
<td>0.09900</td>
<td>0.04270</td>
<td>0.22450</td>
<td>0.10310</td>
<td>0.01970</td>
<td>0.42300</td>
<td>0.34200</td>
<td>0.36210</td>
</tr>
<tr>
<td>DECMOSA</td>
<td>0.07702</td>
<td>0.02834</td>
<td>0.09350</td>
<td>0.03392</td>
<td>0.16713</td>
<td>0.12604</td>
<td>0.02416</td>
<td>0.21583</td>
<td>0.14111</td>
<td>0.36985</td>
</tr>
<tr>
<td>OMOEAI</td>
<td>0.08564</td>
<td>0.03057</td>
<td>0.27141</td>
<td>0.04624</td>
<td>0.16920</td>
<td>0.07338</td>
<td>0.03554</td>
<td>0.19200</td>
<td>0.23179</td>
<td>0.62754</td>
</tr>
</tbody>
</table>
7 Conclusions

In this paper, we have proposed a new PSO-based multi-objective optimisation technique. Combining our technique with Pareto optimality yielded a simple, effective and stable multi-objective optimisation algorithm named as MOAAEPSO. Various performance measures and CEC 2009 test bench functions were considered to quantify the efficiency of the proposed algorithm. Results show that the proposed algorithm is better than other popular multi-objective optimisation algorithms in almost all benchmark functions.

References


Appendix

**KUR**

Definition:

\[
  f_1(x) = \sum_{i=1}^{n-1} \left( -10 \exp \left( -0.2 \left( \sqrt{x_i^2 + x_{i+1}^2} \right) \right) \right)
\]

\[
  f_2(x) = \sum_{i=1}^{n} \left( |x_i|^{0.8} + 5 \sin \left( x_i \right) \right)
\]

Search space: \(-5 \leq x_i \leq 5, \ i = 1, 2, \ldots, n\)
Dimension: \(n = 3\)

**FON**

Definition:

\[
  f_1(x) = 1 - \exp \left( \sum_{i=1}^{n} \left( x_i - \frac{1}{\sqrt{3}} \right)^2 \right)
\]

\[
  f_2(x) = 1 - \exp \left( -\sum_{i=1}^{n} \left( x_i + \frac{1}{\sqrt{3}} \right)^2 \right)
\]

Search space: \(-4 \leq x_i \leq 4, \ i = 1, 2, \ldots, n\)
Dimension: \(n = 3\)

**SCH**

Definition:

\[
  f_1(x) = x^2
\]

\[
  f_2(x) = (x - 2)^2
\]

Search space: \(-10^3 \leq x_i \leq 10^3, \ i = 1, 2, \ldots, n\)
Dimension: \(n = 1\)

**POL**

Definition:

\[
  f_1(x) = \left[ 1 + (g_1 - h_1)^2 + (g_2 - h_2)^2 \right]
\]

\[
  f_2(x) = \left[ (x_1 + 3)^2 + (x_2 + 1)^2 \right]
\]
\[
\begin{align*}
g_1 &= 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2 \\
g_2 &= 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2 \\
h_1 &= 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2 \\
h_2 &= 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2
\end{align*}
\]

Search space: \(-\pi \leq x_i \leq \pi, i = 1, 2, \ldots, n\)

Dimension: \(n = 2\)

**ZDT1**

Definition:

\[
\begin{align*}
f_1(x) &= x_i \\
f_2(x) &= g(x) \left[ 1 - \sqrt{\frac{x_i}{g(x)}} \right]
\end{align*}
\]

where \(g(x) = 1 + 9 \sum_{i=2}^{n} \frac{x_i}{n-1}\)

Search space: \(0 \leq x_i \leq 1, i = 1, 2, \ldots, n\)

Dimension: \(n = 30\)

**ZDT2**

Definition:

\[
\begin{align*}
f_1(x) &= x_i \\
f_2(x) &= g(x) \left[ 1 - \left( \frac{x_i}{g(x)} \right)^2 \right]
\end{align*}
\]

where \(g(x) = 1 + 9 \sum_{i=2}^{n} \frac{x_i}{n-1}\)

Search space: \(0 \leq x_i \leq 1, i = 1, 2, \ldots, n\)

Dimension: \(n = 30\)

**ZDT3**

Definition:

\[
f_1(x) = x_i
\]
f_2(x) = g(x) \left[ 1 - \frac{x_1}{g(x)} \left( \frac{x_1}{g(x)} \sin(10\pi x_1) \right) \right]

where \( g(x) = 1 + 9 \sum_{i=2}^{n} \frac{x_i}{n-1} \)

Search space: \( 0 \leq x_i \leq 1, ~ i = 1, 2, \ldots, n \)

Dimension: \( n = 30 \)

\textit{ZDT4}

Definition:

\( f_1(x) = x_1 \)

\( f_2(x) = g(x) \left[ 1 - \frac{x_1}{g(x)} \right] \)

where \( g(x) = 1 + 9(n-1) + \sum_{i=2}^{n} \left[ x_i^2 - 10 \cos(4\pi x_i) \right] \)

Search space: \( 0 \leq x_i \leq 1, ~ i = 1, 2, \ldots, n \)

Dimension: \( n = 30 \)

\textit{UF1}

Definition:

\( f_1(x) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left[ x_j - \sin(6\pi x_j + \frac{j\pi}{n}) \right]^2 \)

\( f_2(x) = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left[ x_j - \sin(6\pi x_j + \frac{j\pi}{n}) \right]^2 \)

where \( J_1 = \{j \mid j \text{ is odd and } 2 \leq j \leq n\} \) and \( J_2 = \{j \mid j \text{ is even and } 2 \leq j \leq n\} \)

Search space: \( 0 \leq x_1 \leq 1, -1 \leq x_i \leq 1, ~ i = 2, 3, \ldots, n \)

Dimension: \( n = 30 \)

\textit{UF2}

Definition:

\( f_1(x) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} y_j^j \)
\[ f_2(x) = 1 - \sqrt{n_1} + \frac{2}{|J_2|} \sum_{j \in J_2} y_j^2 \]

where \( J_1 = \{ j \mid j \text{ is odd and } 2 \leq j \leq n \} \) and \( J_2 = \{ j \mid j \text{ is even and } 2 \leq j \leq n \} \) and

\[ y_j = x_j - \left[ 0.3x_j^2 \cos\left( 24\pi x_j + \frac{4}{n} \right) + 0.6x_j \right] \cos\left( 6\pi x_j + \frac{j\pi}{n} \right) \quad j \in J_1 \]

\[ y_j = x_j - \left[ 0.3x_j^2 \cos\left( 24\pi x_j + \frac{4}{n} \right) + 0.6x_j \right] \sin\left( 6\pi x_j + \frac{j\pi}{n} \right) \quad j \in J_2 \]

Search space: \( 0 \leq x_1 \leq 1, -1 \leq x_i \leq 1, i = 2, 3, \ldots, n \)

Dimension: \( n = 30 \)

**UF3**

Definition:

\[ f_1(x) = x_1 + \frac{2}{|J_1|} \left( 4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left( \frac{20y_j\pi}{\sqrt{j}} \right) + 2 \right) \]

\[ f_2(x) = 1 - \sqrt{n_1} + \frac{2}{|J_2|} \left( \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left( \frac{20y_j\pi}{\sqrt{j}} \right) + 2 \right) \]

where \( J_1 = \{ j \mid j \text{ is odd and } 2 \leq j \leq n \} \) and \( J_2 = \{ j \mid j \text{ is even and } 2 \leq j \leq n \} \) and

\[ y_j = x_j - x_1 \frac{0.5(\cos(\frac{3(j-2)\pi}{n-2}) - 1)}{\sqrt{j}} \quad j = 2, 3, \ldots, n \]

Search space: \( 0 \leq x_1 \leq 1, i = 1, 2, \ldots, n \)

Dimension: \( n = 30 \)

**UF4**

Definition:

\[ f_1(x) = x_1 + \frac{2}{|J_1|} \left( \sum_{j \in J_1} h(y_j) \right) \]

\[ f_2(x) = 1 - x_1^2 + \frac{2}{|J_2|} \sum_{j \in J_2} h(y_j) \]

where \( J_1 = \{ j \mid j \text{ is odd and } 2 \leq j \leq n \} \) and \( J_2 = \{ j \mid j \text{ is even and } 2 \leq j \leq n \} \) and

\[ y_j = x_j - \sin\left( 6\pi x_j + \frac{j\pi}{n} \right) \quad j = 2, 3, \ldots, n \]

and
Particle swarm optimisation with adaptive neighbourhood search

\[ h(t) = \frac{|r|}{1 + e^{2|t|}} \]

Search space: \(0 \leq x_1 \leq 1, -2 \leq x_i \leq 2, i = 2, 3, \ldots, n\)

Dimension: \(n = 30\)

**UF5**

Definition:

\[ f_1(x) = x_1 + \left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1) + \frac{2}{|J_1|}\left(2 \sum_{j \in J_1} h(y_j)\right) \]

\[ f_2(x) = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1) + \frac{2}{|J_2|}\left(2 \sum_{j \in J_2} h(y_j)\right) \]

where \(J_1 = \{ j \mid j \text{ is odd and } 2 \leq j \leq n \}\) and \(J_2 = \{ j \mid j \text{ is even and } 2 \leq j \leq n \}\) and

\[ y_j = x_j - \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) \quad j = 2, 3, \ldots, n \]

and

\[ h(t) = 2t^2 - \cos(4\pi t) + 1 \]

Search space: \(0 \leq x_1 \leq 1, -1 \leq x_i \leq 1, i = 2, 3, \ldots, n\)

Dimension: \(n = 30\)

**UF6**

Definition:

\[ f_1(x) = x_1 + \max\left\{0, 2\left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1)\right\} \]

\[ + \frac{2}{|J_1|}\left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right) \]

\[ f_2(x) = 1 - x_1 + \max\left\{0, 2\left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1)\right\} \]

\[ + \frac{2}{|J_2|}\left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right) \]

where \(J_1 = \{ j \mid j \text{ is odd and } 2 \leq j \leq n \}\) and \(J_2 = \{ j \mid j \text{ is even and } 2 \leq j \leq n \}\) and

\[ y_j = x_j - \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) \quad j = 2, 3, \ldots, n \]
Search space: $0 \leq x_1 \leq 1$, $-1 \leq x_i \leq 1$, $i = 2, 3, \ldots, n$
Dimension: $n = 30$

**UF7**

Definition:

\[
\begin{align*}
    f_1(x) &= \sqrt[3]{x_1} + \frac{2}{|J_1|} \sum_{j \in J_1} y_j^2 \\
    f_2(x) &= 1 - \sqrt[3]{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} y_j^2
\end{align*}
\]

where $J_1 = \{ j \mid j \text{ is odd and } 2 \leq j \leq n \}$ and $J_2 = \{ j \mid j \text{ is even and } 2 \leq j \leq n \}$ and

\[
y_j = x_j - \sin \left( 6\pi x_1 + \frac{j\pi}{n} \right) \quad j = 2, 3, \ldots, n
\]

Search space: $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $-2 \leq x_i \leq 2$, $i = 3, 4, \ldots, n$
Dimension: $n = 30$

**UF8**

Definition:

\[
\begin{align*}
    f_1(x) &= \cos (0.5x_1 \pi) \cos (0.5x_2 \pi) + \frac{2}{|J_1|} \sum_{j \in J_1} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right) \right)^2 \\
    f_2(x) &= \cos (0.5x_1 \pi) \sin (0.5x_2 \pi) + \frac{2}{|J_2|} \sum_{j \in J_2} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right) \right)^2 \\
    f_3(x) &= \sin (0.5x_1 \pi) + \frac{2}{|J_3|} \sum_{j \in J_3} \left( x_j - 2x_2 \sin \left( 2\pi x_1 + \frac{j\pi}{n} \right) \right)^2
\end{align*}
\]

where

\[
J_1 = \{ j \mid 3 \leq j \leq n \text{ and } j-1 \text{ is a multiplication of 3} \},
J_2 = \{ j \mid 3 \leq j \leq n \text{ and } j-2 \text{ is a multiplication of 3} \},
J_3 = \{ j \mid 3 \leq j \leq n \text{ and } j \text{ is a multiplication of 3} \}
\]

Search space: $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1$, $-2 \leq x_i \leq 2$, $i = 3, 4, \ldots, n$
Dimension: $n = 30$
Particle swarm optimisation with adaptive neighbourhood search

**UF9**

**Definition**

\[ f_1(x) = 0.5 \left[ \max \left\{ 0, \left(1 + \varepsilon \right) \left(1 - 4 \left(2 x_1 - 1\right)^2 \right) \right\} + 2 x_1 \right] x_2 + \frac{2}{|J_1|} \sum_{j \in J_1} \left( x_j - 2 x_2 \sin \left( 2 \pi x_j + \frac{j \pi}{n} \right) \right)^2 \]

\[ f_2(x) = 0.5 \left[ \max \left\{ 0, \left(1 + \varepsilon \right) \left(1 - 4 \left(2 x_1 - 1\right)^2 \right) \right\} - 2 x_1 + 2 \right] x_2 + \frac{2}{|J_2|} \sum_{j \in J_2} \left( x_j - 2 x_2 \sin \left( 2 \pi x_j + \frac{j \pi}{n} \right) \right)^2 \]

\[ f_3(x) = 1 - x_2 + \frac{2}{|J_3|} \sum_{j \in J_3} \left( x_j - 2 x_2 \sin \left( 2 \pi x_j + \frac{j \pi}{n} \right) \right)^2 \]

where

\( J_1 = \{ j \mid 3 \leq j \leq n \text{ and } j - 1 \text{ is a multiplication of 3} \} \),

\( J_2 = \{ j \mid 3 \leq j \leq n \text{ and } j - 2 \text{ is a multiplication of 3} \} \),

\( J_3 = \{ j \mid 3 \leq j \leq n \text{ and } j \text{ is a multiplication of 3} \} \)

and \( \varepsilon = 0.1 \); \( \varepsilon \) can take any other positive value

Search space: \( 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1, -2 \leq x_i \leq 2, i = 3, 4, \ldots, n \)

Dimension: \( n = 30 \)

**UF10**

**Definition**

\[ f_1(x) = \cos(0.5 x_1 \pi) \cos(0.5 x_2 \pi) + \frac{2}{|J_1|} \sum_{j \in J_1} \left[ 4 y_j^2 - \cos(8 \pi y_j) + 1 \right] \]

\[ f_2(x) = \cos(0.5 x_1 \pi) \cos(0.5 x_2 \pi) + \frac{2}{|J_2|} \sum_{j \in J_2} \left[ 4 y_j^2 - \cos(8 \pi y_j) + 1 \right] \]

\[ f_3(x) = \sin(0.5 x_1 \pi) \frac{2}{|J_3|} \sum_{j \in J_3} \left[ 4 y_j^2 - \cos(8 \pi y_j) + 1 \right] \]

where

\( J_1 = \{ j \mid 3 \leq j \leq n \text{ and } j - 1 \text{ is a multiplication of 3} \} \),

\( J_2 = \{ j \mid 3 \leq j \leq n \text{ and } j - 2 \text{ is a multiplication of 3} \} \),

\( J_3 = \{ j \mid 3 \leq j \leq n \text{ and } j \text{ is a multiplication of 3} \} \)
\[ y_j = x_j - 2x_2 \sin \left( \frac{2\pi x_1 + j\pi}{n} \right) \quad j = 3, \ldots, n \]

Search space: \(0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1, \quad -2 \leq x_i \leq 2, \quad i = 3, 4, \ldots, n\)

Dimension: \(n = 30\)