A Novel Diversity Guided Particle Swarm Multi-objective Optimization Algorithm

Zhiyong Li*, Ransikarn Ngambusabongsopa, Esraa Mohammed, Ndatinya Eustache
School of Computer and Communication Hunan University, Changsha 410082, China
E-mail: zhiyong.li@hnu.edu.cn, ransikarn@hotmail.com, compubutterfly@hotmail.com, sure140@gmail.com
doi:10.4156/jdcta.vol5. issue1.29

Abstract
This paper presents a multi-objective diversity guided Particle Swarm optimization approach named MOPSO-AR which increases diversity performance of multi-objective Particle Swarm optimization by using Attraction and Repulsion (AR) mechanism. AR mechanism uses a diversity measure to control the swarm. Being attractive and repulsive will help to overcome the problem of premature convergence. AR mechanism together with crowding distance computation and mutation operator maintains the diversity of non-dominated set in external archive. The approach is verified by several test function experiments. Results demonstrate that the proposed approach is highly competitive in distribution of non-dominated solutions but still keeps convergence towards the Pareto front.

Keywords: Multi-objective Optimization, Particle Swarm Optimization, Attraction and Repulsion

1. Introduction

Multi-objective optimization is the process of simultaneously optimizing Multi-objective Optimization Problems (MOPs). A MOP is related to the problem when two or more objectives have to be optimized concurrently. Generally, such objectives are conflicting and are represented in different measurement units, preventing simultaneous optimizations of each one. MOPs can be found in various fields such as product and process design, finance, aircraft design, oil and gas industry, automobile design, or wherever optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives [15].

There are some methods that have been used to solve MOPs. Evolutionary Algorithms (EAs) are very popular approaches in multi-objective optimization. However some schemes based on Particle Swarm Optimization (PSO) and simulated annealing are significant.

Multi-objective evolutionary algorithms (MOEAs) have been the mainstream to solve MOP. Many MOEAs approaches were proposed such as the Pareto Archived Evolution Strategy (PAES) [1], A Niched Pareto Genetic Algorithm Multi-objective Optimization (NPGA) [2], The strength Pareto Approach II (SPEA2) [4] is an improved version of SPEA [3] and the extreme popular one is non-dominated Sorting Genetic Algorithm II (NSGAII) [5] which improved from NSGA. Since NSGAII proposed in 2002, plenty of afterward MOEAs adopted the innovations of NSGAII to their approaches, e.g. elitism mechanism, non-dominated Sorting Approach and crowding distance. Some researches were using NSGAII as based approach with hybrid or combination with another approach.

Particle Swarm optimization (PSO), several multi-objective optimization algorithms are based on PSO which was originally designed for solving single objective optimization problems. PSO is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. The initial population of particles is initialized with random solutions. For every generation, each solution moves toward the global Pareto front by updating its velocity. The particles follow the best solution achieved among the population of solutions [8, 14].

Applications of PSO to solve MOPs are Multi-objective Particle Swarm Optimization (MOPSO) [10], A Fast and Elitist Multi-objective Particle Swarm Algorithm (NSPSO) [12], Improving PSO-Based Multi-objective Optimization [9], Particle Swarm Optimization Method in Multi-objective Optimization [11].
Problems [11] and An Effective Use of Crowding Distance in Multi-objective Particle Swarm Optimization [8]. For the purpose of improving convergence to the true Pareto front and producing a well-distributed Pareto front, several mechanisms such as elitism, diversity operators, mutation operators, constraint handling and crowding distance have been incorporated into multi-objective optimization algorithms [8, 10, 12].

In [8] Carlo R. Raquel and Prospero C. Naval have presented an approach namely MOPSO-CD that extends the Particle Swarm Optimization algorithm to handle MOPs by incorporating the mechanism of crowding distance computation into the algorithm of PSO specifically on global best selection and in the deletion method of an external archive of non-dominated solutions. The diversity of non-dominated solutions in the external archive is maintained by using the mechanism of crowding distance together with a mutation operator which was proposed in MOPSO [10]. The performance of MOPSO-CD is evaluated on test functions and metrics from previous literature. The results show that MOPSO-CD [8] is highly competitive in converging towards the Pareto front and has generated a well-distributed set of non-dominated solution over MOPSO.

In this paper we propose MOPSO-AR which extends MOPSO-CD by adopting Attraction and Repulsion (AR) mechanism together with MOPSO-CD. AR mechanism was proposed by J. Riget and J.S.Vesterstrom [13]. It uses a diversity measure to control the swarm. Attraction and Repulsion mechanism together with crowding distance computation and mutation operator can obtain better performance in terms of diversity of non-dominated set.

2. Related work

Crowding distance computation, Global best selection, Mutation operation and Constraint handling were incorporated to MOPSO [10] and MOPSO-CD [8]. They were also consisted in our approach. The brief of those mechanisms are as follows.

2.1. Crowding distance computation

The crowding distance value of a solution provides an estimate of the density of solutions surrounding that solution. Crowding distance is calculated by first sorting the set of solutions in ascending objective function values. The crowding distance value of a particular solution is the average distance of its two neighboring solutions. The boundary solutions which have the lowest and highest objective function values are given an infinite crowding distance values so that they are always selected. This process is done for each objective function. The final crowding distance value of a solution is computed by adding the entire individual crowding distance values to each objective function. The pseudo code of crowding distance computation that was applied to MOPSO-CD is shown below [5, 8].

1. Get the number of non-dominated solutions in the external archive
   \[ n = |S| \]
2. Initialize distance
   (a) FOR \( i = 0 \) TO Max (Max is the number of non-dominated solution in archive)
   (b) \( S[i].distance = 0 \)
3. Compute the crowding distance of each solution
   (a) For each objective \( m \)
   (b) Sort using each objective value
   \[ S = \text{sort}(S, m) \]
   (c) For \( i = 1 \) to \( (n-1) \)
   (d) \( S[i].distance = S[i].distance + (S[i+1].m - S[i-1].m) \)
   (e) Set the maximum distance to the boundary points so that they are always selected
   \[ S[0].distance = S[n].distance = \text{maximum distance} \]

2.2. Global best selection
The selection of the global best guide of the particle swarm is a crucial step in a multi-objective-PSO algorithm. It affects both the convergence capability of the algorithm as well as maintaining a good spread of non-dominated solutions [8].

2.3. Mutation operator

The mutation operator of MOPSO was adapted because of the exploratory capability that it could give to the algorithm by initially performing mutation on the entire population then by rapidly decreasing its coverage over time [8, 10].

2.4. Constraint handling

MOPSO-CD [8] adapted the constraint handling mechanism used by NSGA-II [5] due to its simplicity in using feasibility and non-dominance of solutions when comparing solutions. A solution \(i\) is said to be constrained-dominate a solution \(j\) if any of the following conditions is true:

1. Solution \(i\) is feasible and solution \(j\) is not.
2. Both solutions \(i\) and \(j\) are infeasible, but solution \(i\) has a smaller overall constraint violation.
3. Both solutions \(i\) and \(j\) are feasible and solution \(i\) dominates solutions \(j\).

When comparing two feasible particles, the particle which dominates the other particle is considered a better solution. On the other hand, if both particles are infeasible, the particle with a lesser number of constraint violations is considered to be a better solution.

3. Proposed approach

In this section, we introduce A Novel Diversity Guided Particle Swarm Multi-objective Optimization Algorithm named MOPSO-AR approach. MOPSO-AR extends the algorithm of MOPSO-CD [8] by adopting AR mechanism which proposed the algorithm to be diversity guided of the single objective PSO [13].

3.1. Attraction and Repulsion mechanism concept

The intuition behind the PSO model is that by letting information about good solutions spread out through the swarm. At each time step \(t\) (generation) the velocity is updated and the particle is moved to a new position. This new position is simply calculated as the sum of the previous position and the new velocity as shown in equation (1).

\[
X_{[t+1]} = X_{[t]} + V_{[t+1]} \tag{1}
\]

The new velocity is computed from the previous velocity as in equation (2):

\[
V_{[t+1]} = W \times V_{[t]} + R_1 \times (PBEST_{[t]} - P_{[t]}) + R_2 \times (X_{[GBEST]} - P_{[t]}) \tag{2}
\]

Where \(R_1\) and \(R_2\) are real numbers chosen uniformly and at random in a given interval, usually \((0,1)\). \(R_1\) determines the significance of \(P_{[t]}\) and \(PBEST_{[t]}\) respectively while \(R_2\) determines the significance of \(X_{[GBEST]}\) and \(P_{[t]}\) respectively. The parameter \(W\) is the inertia weight and controls the magnitude of the old velocity \(V_{[t]}\) in the calculation of the new velocity \(V_{[t+1]}\).

AR mechanism consists of two phases; attraction phase and repulsion phase. The attraction phase is merely PSO algorithm. The particles will then attract each other, since in general they attract each other in the basic PSO algorithm because of the information flow of good solutions between particles. But the repulsion phase is by “inverting” the velocity-updates formula of the particles.

In the attraction phase the swarm is contracting, and consequently the diversity decreases. When the diversity drops below a lower bound \((d_{Low})\), it switches to the repulsion phase, in which the swarm
expands due to the above inverted update velocity. Finally, when a diversity of \( d_{\text{High}} \) is reached, it switches back to the attraction phase.

The result of this is an algorithm that alternates between phases of exploiting and exploring attraction and repulsion — low diversity and high diversity. \( \text{SetDirection} \) and \( \text{CalculateDiversity} \) are two functions of AR algorithm.

The pseudo-code for \( \text{SetDirection} \) is shown as follows:

\[
\begin{align*}
\text{if} \ (\text{Dir} > 0 \ \&\& \ \text{diversity} < d_{\text{Low}}) \ & \ \text{Dir} = -1; \\
\text{if} \ (\text{Dir} < 0 \ \&\& \ \text{diversity} > d_{\text{High}}) \ & \ \text{Dir} = 1;
\end{align*}
\]

The \( \text{SetDirection} \) determines which phase the algorithm is currently in. However, in the second function, \( \text{CalculateDiversity} \), the diversity of the non-dominated solutions is set according to the diversity-measure:

\[
\text{Diversity}(S) = \frac{1}{|S| \cdot |L|} \sum_{i=1}^{|S|} \sum_{j=1}^{N} (P_{ij} - \overline{P}_j)^2
\]

(3)

Where \( S \) is the swarm, \(|S|\) is the swarm size, \(|L|\) is the length of longest diagonal in the search space, \( N \) is the dimensionality of the problem, \( P_{ij} \) is the \( j \)th value of the \( i \)th particle and \( \overline{P}_j \) is the \( j \)th value of the Average point \( \overline{P} \).

Finally, the velocity-update formula, (2) is changed by multiplying the sign-variable \( \text{Dir} \) to the two last terms in it. This decides directly whether the particles attract or repel each other

\[
V_{[t+1]} = W \times V_{[t]} + \text{Dir} \times (R_1 \times (\text{PBEST}_{[t]} - P_{[t]}) + R_2 \times (X_{(\text{GBEST})} - P_{[t]}))
\]

(4)

3.2. MOPSO-AR algorithm

Our modification in the proposed algorithm MOPSO-AR is applied by adding two new steps 4 and 6.c; while modifying the formula of step 6.d.ii in the MOPSO-CD algorithm. The pseudo code of the proposed algorithm is demonstrated as follows:

1. For each particle in \( psize \) //\( psize \) is the population size
   a. Do initialization of variables
   b. Evaluate \( \text{pop}[i] \)
   End For
2. Iteration counter \( t = 0, \text{Dir} = 1 \)
3. Assign \( d_{\text{Low}}, d_{\text{High}} \)
4. Store the non-dominated vectors found in \( \text{pop} \) into external archive
5. Repeat
   a. Compute the crowding distance values
   b. Sort the non-dominated solutions in descending order according to crowding distance
   c. Calculate \( \text{Diversity} \) as (3)
   d. For each particle in \( psize \)
      i. Randomly select \( \text{GBEST} \) for \( \text{pop}[i] \) and store in \( \text{GBEST}[i] \).
      ii. Compute the new velocity using (4)
      iii. Calculate the new position of \( \text{pop}[i]: \text{pop}[i] = \text{pop}[i] + V[i] \)
      iv. If \( \text{pop}[i] \) goes beyond the boundaries, then its \( v[i] \) is multiplied by -1
      v. If \( t < (\text{MAXT} \times \text{PMUT}) \), then perform mutation on \( \text{pop}[i] \), and evaluate \( \text{pop}[i] \)
         // \( \text{MAXT} \) is the maximum number of iterations
   End For
f. Insert all new non-dominated solution in \( \text{pop} \) into the archive. If
   i. not dominated by the stored solutions, then solutions dominated by new solutions will be removed from the archive.
ii. the archive is full, the solution to be replaced is determined by:
- Computing crowding distance in the archive
- Sort them in descending order
- Randomly select a particle from the specified bottom portion, then replace with the new solution

6. Update PBEST of each particle in pop.
7. Until MAXT is reached

4. Experimentation

4.1. Performance measurement

In this paper, we used two metrics to assess the achievement of two goals of a multi-objective optimization. First metric is to convergence to the Pareto-optimal set; while second is the maintenance of diversity in the solution of Pareto set. Convergence to the Pareto-optimal is a spacing metric which proposed by Schott [16] to measure the spread (distribution) of vectors throughout the set of non-dominated solutions. It is defined as:

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

(5)

Where \( d_i = \min j \left( | f_i^1(x) - f_j^1(x) | + | f_i^2(x) - f_j^2(x) | \right) \); \( i, j = 1, \ldots, n \), \( \overline{d} \) is the mean of all \( d_i \), and \( n \) is the number of non-dominated vector found. The value of this measure is zero for a uniform distribution.

The second measure is the two set coverage metric (C) which was proposed by Zitzler et al [8]. This metric is defined as

\[ C(A, B) = \frac{| \{ b \in B : \exists a \in A : a \succ b \} |}{|B|} \]  

(6)

If \( C(A, B) = 1 \), this means that all solutions in \( B \) are weakly dominated by \( A \). Whereas, if it is zero, then none of the solutions in \( B \) are weakly dominated by \( A \). Other than it is not necessarily that \( C(A, B) \) is equal to \( 1-C(A, B) \). In case of \( 0<C(A, B)<1 \) and \( 0<C(B, A)<1 \), this means that neither \( A \) weakly dominates \( B \) nor \( B \) weakly dominates \( A \) or we can say that sets \( A \) and \( B \) are incomparable or \( A \) and \( B \) is not worse than each other.

4.2. Test Functions

In our experiments, we used three test functions. First test function is proposed by Kita [6] as follows:

Maximize \( F = (f_1(x, y), f_2(x, y)) \), where \( f_1(x, y) = -x^2 + y, f_2(x, y) = \frac{1}{2} x + y + 1 \)
Subject to \( 0 \geq \frac{1}{6} x + y - \frac{13}{2}, 0 \geq \frac{1}{2} x + y - \frac{15}{2}, 0 \geq 5 x + y - 30 \) And \( x, y \geq 0 \), the range is \( 0 \leq x, y \leq 7 \).

The second test function is proposed by Kursawe [7] as follows:

Minimize \( f_1(x) = \sum_{i=1}^{n-1} \left( -10 \exp\left( -0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right) \), Minimize \( f_2(x) = \sum_{i=1}^{n} |x_i|^{0.8} + 5 \sin(x_i) \)
where \( -5 \leq x_1, x_2, x_3 \leq 5 \).

And the third test function is proposed by Deb [9] as follows:
A Novel Diversity Guided Particle Swarm Multi-objective Optimization Algorithm
Zhiyong Li, Ransikarn Ngambusabongsopa, Esraa Mohammed, Ndatinia Eustache
International Journal of Digital Content Technology and its Applications. Volume 5, Number 1, January 2011

Minimize \( f_1(x_1, x_2) = x_1 \), Minimize \( f_2(x_1, x_2) = \frac{g(x_2)}{x_1} \)

where \( g(x_2) = 2.0 - \exp \left\{ -\frac{(x_2 - 0.2)^2}{0.004} \right\} - 0.8 \exp \left\{ -\frac{(x_2 - 0.6)^2}{0.04} \right\} \) and \( 0.1 \leq x_1, x_2 \leq 1.0. \)

4.3. Results

We compare MOPSO-AR, MOPSO-CD and MOPSO by computing each function 30 times. In all experiments, 100 particles were used, an external archive size of 500 particles and a mutation rate 0.5. In addition, the maximum number of iterations is set to 100. In MOPSO-AR, we assigned \( d_{Low} \) as \( 5.0 \times 10^{-3} \), \( d_{High} \) as 0.25.

Table 1 represents convergence of non-dominated solutions generated by the three algorithms MOPSO-AR, MOPSO-CD and MOPSO. The graphic results illustrate that computation of the three algorithms are able to converge to the true Pareto fronts of the three functions but the Pareto front of MOPSO is not steady as another the two algorithms.

Table 1. Pareto front of MOPSO-AR, MOPSO-CD and MOPSO

<table>
<thead>
<tr>
<th>Test function</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOPSO-AR</td>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
<td><img src="image3" alt="Graph" /></td>
</tr>
<tr>
<td>MOPSO-CD</td>
<td><img src="image4" alt="Graph" /></td>
<td><img src="image5" alt="Graph" /></td>
<td><img src="image6" alt="Graph" /></td>
</tr>
<tr>
<td>MOPSO</td>
<td><img src="image7" alt="Graph" /></td>
<td><img src="image8" alt="Graph" /></td>
<td><img src="image9" alt="Graph" /></td>
</tr>
</tbody>
</table>

Table 2 demonstrates the results of coverage metric C. The results of C illustrate that if \( 0 < C(MOPSO-AR, MOPSO-CD) < 1 \) and \( 0 < C(MOPSO-CD, MOPSO-AR) < 1 \), then this means that neither MOPSO-AR weakly dominates MOPSO-CD nor MOPSO-CD weakly dominates MOPSO-AR. Consequently, the average performance of the two algorithms is incomparable.

In the first and third test functions, \( C \) average of MOPSO-AR and MOPSO are \( 0 < C(MOPSO-AR, MOPSO) < 1 \) and \( 0 < C(MOPSO, MOPSO-AR) < 1 \). This means that neither MOPSO-AR weakly dominates MOPSO nor MOPSO weakly dominates MOPSO-AR. Thus, the average performance of the two algorithms is incomparable. In the second test function, \( C \) average of MOPSO-AR and MOPSO are \( C(MOPSO-AR, MOPSO)=0.0028 \) and \( C(MOPSO-AR, MOPSO)=... \)
A Novel Diversity Guided Particle Swarm Multi-objective Optimization Algorithm
Zhiyong Li, Ransikarn Ngambusabongsopa, Esraa Mohammed, Ndatinya Eustache
International Journal of Digital Content Technology and its Applications. Volume 5, Number 1, January 2011

0. As a result, no solution of MOPSO-AR algorithm is weakly dominated by MOPSO algorithm but some solutions of MOPSO are weakly dominated by MOPSO-AR algorithm.

In table 3, the results of spacing metric and computation time are listed. The spacing metric measures the performance in terms of diversity. The results of the three test functions show that MOPSO-AR has highly performance over MOPSO-CD and MOPSO. However, the computation time of the three test functions of MOPSO are better than MOPSO-AR and MOPSO-CD. The results of computation time in the first and second test functions MOPSO-CD are a little better computation time than MOPSO-AR, approximately about 0.01 and 0.04 seconds. Nevertheless in the third test function MOPSO-AR has a better computation time over MOPSO-AR almost 9 seconds, which is significantly big difference.

Table 2. Coverage (C) results of MOPSO-AR with MOPSO-CD and MOPSO-AR with MOPSO

<table>
<thead>
<tr>
<th>Test function</th>
<th>A : MOPSO-AR</th>
<th>B : MOPSO-CD</th>
<th>A : MOPSO-AR</th>
<th>B : MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C(A,B)</td>
<td>C(B,A)</td>
<td>C(A,B)</td>
<td>C(B,A)</td>
</tr>
<tr>
<td>1</td>
<td>0.0116</td>
<td>0.0136</td>
<td>0.0098</td>
<td>0.0130</td>
</tr>
<tr>
<td>2</td>
<td>0.2282</td>
<td>0.1853</td>
<td>0.0028</td>
<td>0.0000</td>
</tr>
<tr>
<td>3</td>
<td>0.0773</td>
<td>0.0751</td>
<td>0.0376</td>
<td>0.0134</td>
</tr>
</tbody>
</table>

Table 3. Spacing and Computation time results of MOPSO-AR, MOPSO-CD and MOPSO

<table>
<thead>
<tr>
<th>Result of</th>
<th>Test function</th>
<th>MOPSO-AR</th>
<th>MOPSO-CD</th>
<th>MOPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Spacing Metric</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.043670</td>
<td>0.068185</td>
<td>0.104172</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.030361</td>
<td>0.044352</td>
<td>0.068844</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.027285</td>
<td>0.030671</td>
<td>0.071405</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Computation time (seconds)</td>
<td>0.054945</td>
<td>0.091575</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.098901</td>
<td>0.086081</td>
<td>0.315018</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.161172</td>
<td>0.122711</td>
<td>0.091575</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.540293</td>
<td>12.516484</td>
<td>0.315018</td>
<td></td>
</tr>
</tbody>
</table>

5. Algorithm discussion

In the proposed algorithm, some important issues are considered and modified. These issues include how to place the parameter Dir, how to randomly select the global best guide for particle swarm and effect of mutation operator.

5.1. Parameter Dir discussion

Three types of position are considered to place parameter Dir to MOPSO-AR algorithm in step 6.d.ii when compute the new velocity.

In simply PSO formula, we divide the formula in three parts (a), (b) and (c).

\[ V[i] = W \times V[i] + R_1 \times (PBEASTS[i] - pop[i]) + R_2 \times (X[GBEST] - pop[i]) \]

(a) (b) (c)

Dir type 1: Let parameter Dir cover part (b) and (c)
\[ V[i] = W \times V[i] + \text{Dir}(R1 \times (PBESTS[i] - pop[i]) + R2 \times (X[GBEST] - pop[i])) \]

Dir type 2: Let parameter Dir cover part (b)

\[ V[i] = W \times V[i] + \text{Dir}(R1 \times (PBESTS[i] - pop[i]) + R2 \times (X[GBEST] - pop[i])) \]

Dir type 3: Let parameter Dir cover part (c)

\[ V[i] = W \times V[i] + R1 \times (PBESTS[i] - pop[i]) + \text{Dir}(R2 \times (X[GBEST] - pop[i])) \]

In this paper, results of coverage and computation time are not demonstrated because the computing time of the three types Dir parameters placing are approximately equal. In addition the results of coverage are similar with the study so far, as in section 5.1. This implies that the average performance of the coverage metric of the three types Dir parameters placing are incomparable. In other words, they do not weakly dominate each other. Table 4 shows that the diversity of non-dominated solutions generated by MOPSO-AR algorithm using Dir type 1 is better than the other two types in all three test functions. As a result, we applied Dir type 1 to MOPSO-AR algorithm.

<table>
<thead>
<tr>
<th>Test function</th>
<th>Spacing metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dir type 1</td>
</tr>
<tr>
<td>1</td>
<td>0.043670</td>
</tr>
<tr>
<td>2</td>
<td>0.030361</td>
</tr>
<tr>
<td>3</td>
<td>0.027285</td>
</tr>
</tbody>
</table>

5.2. Global best selection discussion

In step 6.d.i, MOPSO-AR algorithm randomly selects the global best guide for particle swarm and stores its position to GBEST. We considered three types of Global Best Selection by randomly dividing area to three specified portions. The non-dominated solutions in external archive(\(X\)) are ordered by crowding distance in values descending order. These three portions are Top Specified Portion which is 10% in top area of \(X\), Middle Specified Portion which is after 10% from top area and before 10% from bottom area of \(X\) and finally, Bottom Specified Portion that is 10% in bottom area of \(X\).

Whereas the three types of the Global Best Selection are GBest Type 1, GBest Type 2 and GBest Type 3. In GBest Type 1, all particles randomly select the global best from top specified portion. Secondly, in GBest Type 2, 60% of particles randomly select the global best from top specified portion, 20% of particles randomly select the global best from middle specified portion and 20% of particles randomly select global best from bottom specified portion. However in GBest Type 3, 60% of particles randomly select the global best from top specified portion, 40% of particles randomly select the global best from middle specified portion.

The results of coverage and computation time are not showed in this paper because computer time of three types global best selection approximately equal. In addition the results of coverage are similar with study so far in section 5.1, it means the average performance in coverage metric of the three types global best selection are incomparable or in other words they do not weakly dominates each other. Table 5 illustrates the diversity of non-dominated solutions generated by MOPSO-AR algorithm using global best selection GBEST Type 1 is better than other two types in all three test functions. Consequently MOPSO-AR algorithm is randomly selection the global best guide for particles from the top specified portion.

This simulation confirmed that the global best selection of MOPSO-CD [8] is the most advantage selection.
Table 5. Spacing metric results of Gbest type 1, Gbest type 2 and Gbest type 3

<table>
<thead>
<tr>
<th>Test function</th>
<th>Spacing metric</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gbest type 1</td>
</tr>
<tr>
<td>1</td>
<td>0.043670</td>
</tr>
<tr>
<td>2</td>
<td>0.030361</td>
</tr>
<tr>
<td>3</td>
<td>0.027285</td>
</tr>
</tbody>
</table>

5.3. Mutation operator discussion

Mutation operator is analyzed by comparing the results between generating of MOPSO-AR with and without mutation operator. Table 6 illustrates the performance in terms of diversity to non-dominant solutions and computation time of MOPSO-AR with mutation operator. Results show that MOPSO-AR with mutation operator is better than that without mutation operator. Hence mutation operator is incorporated to MOPSO-AR algorithm.

Table 6. Spacing and computation time results of MOPSO-AR with and without mutation operator

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Test function</th>
<th>Spacing metric</th>
<th>Computation time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOPSO-AR</td>
<td>1</td>
<td>0.043670</td>
<td>0.098901</td>
</tr>
<tr>
<td>With</td>
<td>2</td>
<td>0.030361</td>
<td>0.161172</td>
</tr>
<tr>
<td>Mutation</td>
<td>3</td>
<td>0.027285</td>
<td>3.540293</td>
</tr>
<tr>
<td>MOPSO-AR</td>
<td>1</td>
<td>0.050946</td>
<td>0.115385</td>
</tr>
<tr>
<td>Without</td>
<td>2</td>
<td>0.032667</td>
<td>0.236264</td>
</tr>
<tr>
<td>Mutation</td>
<td>3</td>
<td>0.028198</td>
<td>10.023810</td>
</tr>
</tbody>
</table>

6. Conclusion

In this paper, we have proposed an approach namely MOPSO-AR which incorporated Attraction and Repulsion mechanism into multi-objective particle swarm. AR mechanism is the diversity guided that uses a diversity measure to control the swarm. Attractive and repulsive features help to overcome the problem of premature convergence. We also considered about how to place the parameter Dir, how to randomly select the global best guide for particle swarm. In addition we discuss the effect of mutation operator on MOPSO-AR algorithm. The approach is verified by several test function experiments. AR mechanism together with crowding distance computation and mutation operator maintains the diversity of non-dominated set in external archive. Experimental results demonstrate MOPSO-AR is highly competitive in producing a well distributed Pareto front, in addition to keep convergence towards the Pareto front.

7. Acknowledgments

This work is supported by the “Fundamental Research Funds for the Central Universities” of Hunan University in China.
8. References


