Dynamic Multi-Swarm Particle Swarm Optimization for Multi-Objective Optimization Problems

J.J. Liang\textsuperscript{1}, B.Y. Qu\textsuperscript{2}
\textsuperscript{1}School of Electrical Engineering
Zhengzhou University
Zhengzhou, China
liangjing@zzu.edu.cn
\textsuperscript{2}School of Electrical and Information Engineering
Zhongyuan University of Technology
Zhengzhou, China
e070088@e.ntu.edu.sg

P. N. Suganthan\textsuperscript{3}, B. Niu\textsuperscript{4}
\textsuperscript{3}School of Electrical & Electronic Engineering
Nanyang Technological University
Singapore
epnsugan@ntu.edu.sg
\textsuperscript{4}College of Management
Shenzhen University
Shenzhen, China
driuben@gmail.com

Abstract— In this paper, Dynamic Multi-Swarm Particle Swarm Optimizer (DMS-PSO) which was first designed for solving single objective optimizations problems is extended to solve Multi-objective optimization problems with constraints. Through analysis, novel pbest and lbest updating criteria which are more suitable for solving Multi-objective optimization problems are proposed. By combining the external archive and the novel updating criteria, excellent performance is achieved by DMS-MO-PSO on eight benchmark test functions.

Keywords— Particle swarm optimizer; multi-objective optimization; dynamic multi-swarm optimizer

I. INTRODUCTION

We frequently encounter multi-objective problems with conflicting objectives. For example, in an automatic air-conditioning system, we have two objectives: keeping a comfort temperature and saving the energy. Then we must find a way to balance these two objectives. In fact, there are many more complex problems in engineering, business and so on with conflicting objectives. Thus, we need to find different tradeoffs and choose a proper one or more for those problems. We can mathematically express a multi-objective optimization problem as follows:

\begin{equation}
\text{Minimize} \quad \mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\
\text{s. t.} \quad g_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, J \\
\quad h_k(\mathbf{x}) = 0, \quad k = 1, \ldots, K \\
\mathbf{x} \in [\mathbf{X}_{\text{min}}, \mathbf{X}_{\text{max}}]
\end{equation}

\( \mathbf{X} \) is the decision vector, \( \mathbf{Y} \) is the objective vector. Different from the single objective optimization, there are two spaces to be considered. One is the decision space, we denoted it as \( \mathbf{X} \); another one is called the objective space, we denoted it as \( \mathbf{Y} \). In multi-objective optimization, for any two decision vectors \( \mathbf{u} \) and \( \mathbf{v} \), \( \mathbf{u} \) is said to dominate \( \mathbf{v} \) if \( \mathbf{u} \) is no worse than \( \mathbf{v} \) in all objectives or \( \mathbf{u} \) is strictly better than \( \mathbf{v} \) in at least one objective. Among a set of solutions \( \mathbf{P} \), the non-dominated set of solutions \( \mathbf{P}' \) are those that are not dominated by any member of the set \( \mathbf{P} \). When the set \( \mathbf{P} \) is the entire search space, or \( \mathbf{P} = \mathbf{S} \), the resulting non-dominated set \( \mathbf{P}' \) is called the Pareto-optimal set [1].

Like global and local optimal solutions in the case of single-objective optimization, there could be global and local Pareto-optimal sets in multi-objective optimization. The objective of multi-objective optimization is to find a set of solutions which can express the Pareto-optimal set well. Thus, there are two goals for the optimization, namely fast convergence to the Pareto-optimal front and good distribution of solutions along the front.

Development of evolutionary algorithms to solve multi-objective optimization problems has attracted much interest recently and a number of multi-objective evolutionary algorithms have been suggested. The main advantage of evolutionary algorithms (EAs) in solving multi-objective optimization problems is their ability to find multiple Pareto-optimal solutions in one single run [1][2].

Particle swarm optimization (PSO) is based on swarm intelligence which was proposed by Kennedy and Eberhart in 1995 [3][4]. As the PSO algorithm is simple in concept, easy to implement and computationally efficient, a number of proposals have been suggested to extend PSO to handle multi-objective problems. Ray and Liew [5] combined Pareto dominance and concepts of evolutionary techniques with the particle swarm. The approach uses crowding to maintain diversity and Pareto ranks to handle constraints. Better performing particles are recorded into a set of leaders based on non-dominated rank and the remaining particles move towards a leader randomly selected from the leaders. Leaders with fewer individuals around them have a high probability of being selected. Parsopoulos and Vrahatis [6] introduced two methods that extend the PSO to be able to handle multi-objective problems. They were a weighted aggregation approach and Vector evaluated PSO. In Hu and Eberhart [7], a dynamic neighborhood and a new pbest updating strategy were proposed. In each generation, the neighborhood best is dynamically chosen according to a particle’s distance to the other particles. The approach is then further improved by adding a secondary population, called extended memory, to store global Pareto optimal solutions to reduce computation.
time [8]. Although these approaches have been shown to find multiple non-dominated solutions on many test problems, researchers realized the need of introducing elitism as evidenced in many recent successful MOEAs. Further, recently more researchers are interested in incorporating an external archive into MOPSO to enhance the convergence properties.

Fieldsend and Singh [9] used dominated tree archive to select the best global individual based on a concept of closeness to members in the non-dominated set, and maintained a set of previous best solutions for each particle. Turbulence is incorporated to improve the performance of the multi-objective PSO. This approach uses an unbounded archive. However, some researchers bound the archive size to reduce the complexity of archive updating [10][11][12][13]. Mostaghim and Teich [12] proposed a sigma method in MOPSO for finding the best local guides for each particle in order to converge fast to the Pareto-optimal front with good diversity. In another paper [11], the same authors use \( \varepsilon \)-dominance to fix the archive size and compared \( \varepsilon \)-dominance to the clustering techniques. They used an initial archive instead of an empty archive for MOPSO. Li [14] extended the PSO to multi-objective problems with the non-dominated sorting concept of NSGA-II [15] and constructed the so called Non-dominated Storing Particle Swarm Optimizer (NSPSO). Instead of a single comparison between a particle's personal best and current position, it executes non-domination comparison among all particles' personal bests and their offspring in the entire population. Later, the author proposed a maxmin PSO for multi-objective optimization, which uses a fitness function derived from the maxmin strategy to determine Pareto-domination. One advantage is that no additional clustering or niching technique is needed since the maxmin fitness function provided the domination information and diversity information. Both algorithms showed competitive performance with the real-coded NSGA-II [16].

Bartz-Beielstein et al. [13] proposed DOPS that integrates the archiving technique into particle swarm optimization. They also analyzed several modifications and extensions of the archiving techniques. Coello and Lechuga [17] proposed MOPSO with an external repository and with an adaptive grid similar to PAES. This approach selects a global best based on roulette wheel selection of a hypercube. Coello et al. [18] also incorporated a special mutation operator to enhance the exploratory capabilities. Another improved version (called AMOPSO) is presented by Pulido and Coello [19], in which a clustering technique is used to divide the population of particles into several swarms in order to maintain a better distribution of solutions. A particle angle division method for finding the global best particle for each particle of the population was introduced in [20]. He et al. applied an improved quick sort method to construct the non-dominated solution set and introduced the crossover operator and mutation operator to increase the diversity of particle swarm [21]. Durillo el al. [22] studied the effect of parameter scalability in a number of state-of-the-art multi-objective metaheuristics and concluded that the two analyzed algorithms based on particle swarm optimization and differential evolution yield the best overall results. Zhao et al. introduced two lbests to guide the search direction in a novel multi-objective particle swarm optimizer to focus the search around small regions [23].

The paper is structured as follows. Section II introduces the basic Dynamic Swarm Optimizer. Section III describes the proposed novel method for multi-objective optimization in detail. Section IV presents experimental results on several test functions. And the conclusion is given in Section V.

II. DYNAMIC MULTI-SWARM PARTICLE SWARM OPTIMIZATION

The dynamic multi-swarm particle swarm optimizer (DMS-PSO) [24][25] is constructed based on the local version of PSO and a periodically changed neighborhood structure is used. In contrast with other evolutionary algorithms that prefer larger population, PSO needs a comparatively smaller population size. Especially for simple problems, a population with three to five particles can achieve satisfactory results. PSO with small neighborhoods performs better on complex problems. The population was divided into small sized swarms in the DMS-PSO and each swarm uses its own members to search for better area in the search space. Every R generations, the population is regrouped randomly and starts searching using a new configuration of small swarms. Here R is called regrouping period. In this way, the good information obtained by each swarm is exchanged among the swarms and the diversity of the population is increased simultaneously.

In the comprehensive learning particle swarm optimizer [26], each dimension of a particle can learn from its own pbest and the pbest of other particles. It has been shown that it can improve the performance of the algorithm when we assign some dimensions to learn from its own best positions and some dimensions to learn from other exemplars. Thus, a similar self-learning strategy is also introduced in dynamic multi-swarm particle swarm optimizer. Each Particle has a corresponding \( P_c \). Every R generation, \( keep_id \) is decided by \( P_c \). If this random number is larger than or equal to \( P_c \), this dimension will be set at the value of its own pbest, \( keep_id \) is set to 1 and otherwise \( keep_id \) is set to 0 and it will learn from the lbest and its own pbest as the PSO with constriction coefficient:

\[
\text{If } keep_id = 0 \quad \\
\text{otherwise } keep_id = 1
\]

\[
\begin{align*}
\text{If } keep_id &= 0 \\
\text{otherwise } keep_id &= 1 \\
\text{endif}
\end{align*}
\]

III. DYNAMIC MULTI-SWARM PARTICLE SWARM OPTIMIZATION FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

A. External Archive and Non-dominated Sorting

With the dynamic multi-swarm structure and self-learning strategy, the DMS-PSO possesses good diversity and good global search ability, thus a Dynamic Multi-swarm Particle
Swarm Optimization for Multi-Objective optimization problems (DMS-MO-PSO) is constructed based on it. An external archive is added to keep a historical record of the non-dominated solutions obtained during the search process. The maximum size of the archive $N_{\text{max}}$ is predefined. The technique of updating the external archive is similar to the NSGAII [16] and the schematic is presented in Figure 1.

**1) Non-dominated Sorting**

After adding the new solutions to the external archive, non-dominated sorting is performed on the external archive. We first find the best non-dominated solutions, non-dominated solutions of level 1, in the whole external archived population. We then find the next best non-dominated solutions, non-dominated solutions of level 2, in the remaining members of the external archive. In this way, we find all pareto fronts with different levels: $F_1, F_2, \ldots$.

**2) Crowding Distance Sorting**

If the external archived population reaches its maximum size $N_{\text{max}}$, set new external archive $P = \emptyset$, perform $P = P \cup F_i$ until $|P| + |F_{i+1}| = N_{\text{max}}$. We sort the solutions in $F_{i+1}$ according to one objective as in Figure 2, assign a large distance to the boundary solution and calculate the summation of the distances from the nearest two solutions in other members of $F_{i+1}$. We include the most widely spread $N_{\text{max}} - |P|$ solutions in $P$.

**B. Choose Local Best for Each Sub-Swarm**

Different from the DMS-PSO for single objective optimization problem, in DMS-MO-PSO, $\text{best}_k$ are chosen from $F_1$, the best non-dominated solutions set in the external archive. After the sub-swarms are regrouped, we sort the external archive based on one objective function (randomly chosen) then divide the external archive to $n$ (the number of sub-swarms) parts averagely according to that objective as Figure 3. For each sub-swarm, we randomly choose one member from one corresponding part. Thus, each part has one sub-swarm searching for it. This method can maintain the diversity of the population to obtain an external archive with good diversity.

**C. Update pbest and lbest**

An important characteristic of PSO is that it has $\text{pbest}$ and $\text{lbest}$ (or $\text{gbest}$ in the global version) to record the historical information of the particles. $\text{pbest}$ and $\text{lbest}$ guide the search of the particles. Thus, it is important to decide how to update them. In single objective optimization, the answer is straightforward and $\text{pbest}$ and $\text{lbest}$ will be updated if better solutions are found. A better solution here means a solution which has a larger fitness value. But, in the multi-objective optimization world, the answer is not straightforward.

In PSO variants for multi-objective optimization, there exist five updating methods:

- $\text{Phbest}$ is replaced if $x$ dominates $\text{pbest}$, otherwise if $x$ is mutually non-dominating with $\text{pbest}$, $\text{pbest}$ has 50% probability to be replaced.
- $\text{Pbest}$ is replaced if $x$ dominates $\text{pbest}$, otherwise if $x$ is mutually non-dominating with $\text{pbest}$.
- $\text{Phbest}$ is replaced only if $x$ dominates $\text{pbest}$.
- $\text{Pbest}$ is replaced if $x$ dominates any pareto optimal solution in the current generation.
- A set of mutually non-dominating $\text{pbests}$ is maintained for each $x$. 

---

**Figure 1.** External Archive Updating

**Figure 2.** The crowding distance calculation

**Figure 3.** Illustration of choosing local best for each sub-swarm (sorting according to $f_2(x)$)
Obviously, $x$ is better than $\text{pbest}$ if $x$ dominates $\text{pbest}$. But should we update $\text{pbest}$ if $X$ is not dominated by $\text{pbest}$? If we do not accept the non-dominated solution, in the end of evolutionary search, when the Pareto front has been found, the updating will seldom happen. But, should we accept the non-dominated solution? From Figure 4, we can observe that each time $\text{pbest}$ is updated by a non-dominated solution, obviously the $\text{pbest}$ moves away from the true Pareto Front. Though this is just an extreme example, this can happen in the search process. Thus, updating $\text{pbest}$ or $\text{lbest}$ when the new solution is not dominated by $\text{pbest}$ or $\text{lbest}$ is not a good idea. Sometime, it may lead the particle to fly in the wrong direction. It is because no information of external archive which contains the historical information of the whole search process is considered in the comparison.

Therefore, a new comparison criterion is proposed here. Best non-dominated solutions external archive will be used in the updating and in order to reduce the computation complexity, a reference front $RF$ is used instead of all member of $F_1$. The maximal size of the reference front $N_{\text{ref}}$ is predefined.

- If $|F_1| \leq N_{\text{ref}}$, $RF = F_1$;
- If $|F_1| > N_{\text{ref}}$, put the boundary solutions of in $RF$, we then randomly chose $N_{\text{ref}} - 2$ solutions from $F_1$ except the boundary solutions.

With the new comparison criterion, $\text{pbest}$ or $\text{lbest}$ is updated when

- $x$ dominates $\text{pbest}$ or $\text{lbest}$, or
- $x$ is not dominated by any member of the reference front $RF$.

Four possible scenarios of $\text{pbest}$ updating ($\text{lbest}$ updating is the same) are shown in Figure 5.

- $x_3$ is not dominated by $\text{pbest3}$ and $x_3$ is dominated by at least one member of $RF$, so $\text{pbest3}$ will not be updated;
- $x_4$ is dominated by $\text{pbest4}$, so $\text{pbest4}$ will not be updated;

Since constraints are frequently associated with real-world optimization problems, we also use constrained-domination to handle constraints [16]. A solution $i$ is said to constrained-dominate a solution $j$, if any of the following conditions is true.

- Solution $i$ is feasible and solution $j$ is not.
- Solution $i$ and $j$ are both infeasible, but solution $i$ has a smaller overall constraint violation.
- Solutions $i$ and $j$ are feasible and solution $i$ dominates solution $j$.

According to this constrained-domination principle, DMS-MO-PSO can deal with constrained problems without changing the modularity or computational complexity.

**D. Convergence Phase**

Different from the DMS-PSO for single objective optimization problems, in DMS-MO-PSO, in the convergence phase, the sub-swarms will not be grouped into one big swarm since the goals of multi-objective optimization are convergence to the Pareto Front and maintaining the diversity of the solutions. For the sake of improving the convergence to the Pareto Front, we set $P_c = 1$ for all particles to stop the self learning and speed up the convergence when $|F_1| > 0.5N_{\text{max}}$. The local search phase is removed since the objective of DMS-MO-PSO is to find a set of solutions not one solution. The convergence phase also plays the role of local search.

The flowchart of DMS-MO-PSO is given in Figure 6:
IV. EXPERIMENTS

A. Performance Measures

In order to measure the performance of MOEAs quantitatively, we need some performance metrics to evaluate and compare the algorithms. Three different metrics are employed to evaluate the performance of an MOEA:

1) Convergence Metric ($\gamma$):
This metric finds an average distance between non-dominated solutions found and the actual Pareto-optimal front, as follows:

$$\gamma = \frac{\sum_{i=1}^{N} d_i}{N}$$  \hspace{1cm} (3)

where $N$ is the number of non-dominated solutions obtained with an algorithm and $d_i$ is the Euclidean distance (in objective space) between the each of the non-dominated solutions and the nearest member of the actual Pareto optimal front. A smaller value of $\gamma$ demonstrates a better convergence performance.

2) Spread Metric ($\Delta$):

Deb et al. [16] proposed such a metric to measure the spread in solutions obtained by an algorithm. This metric is defined as

$$\Delta = \frac{\sum_{m=1}^{M} d^*_m + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{\sum_{m=1}^{M} d^*_m + (N-1)\bar{d}}$$  \hspace{1cm} (4)

Here, the parameters $d^*_m$ are the Euclidean distance between the extreme solutions of Pareto optimal front and the boundary solutions of the obtained non-dominated set corresponding to $m$th objective function. The parameter $d_i$ is the Euclidean distance between neighboring solutions in the obtained non-dominated solutions set and $\bar{d}$ is the mean value of these distances. $\Delta$ is zero for an ideal distribution when $d^*_m = 0$ and all $d_i$ equal to $\bar{d}$. Smaller the value of $\Delta$, the better the diversity of the non-dominated set is.

B. Experimental Settings

In the simulations, eight test problems are chosen from the standard MOEA literature. The problems are defined below.

1) Test problem 1 (SCH):
Min $f_1(x) = x^2$
Min $f_2(x) = (x-2)^2$

where $D = 1$ and $x \in [-10^5, 10^5]$.\hspace{1cm} (5)

2) Test problem 2 (FON):
Min $f_1(x) = 1 - \exp\left(-\sum_{i=1}^{3} (x_i - \frac{1}{\sqrt{D}})^2\right)$
Min $f_2(x) = 1 - \exp\left(-\sum_{i=1}^{3} (x_i + \frac{1}{\sqrt{D}})^2\right)$

where $D = 3$ and $x_i \in [-4, 4]$.\hspace{1cm} (6)

3) Test problem 3 (KUR):
Min $f_1(x) = \sum_{i=1}^{D-1} (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2}))$
Min $f_2(x) = \sum_{i=1}^{D} (|x_i|^{0.8} + 5\sin(x_i^2))$

where $D = 3$ and $x_i \in [-5, 5]$.\hspace{1cm} (7)

4) Test problem 4 (ZDT 1):
Min $f_1(x) = x_1$
Min $f_2(x) = g(x)[1 - \sqrt{x_1 / g(x)}]\hspace{1cm} (8)$
$g(x) = 1 + 9 \cdot \frac{\sum_{i=2}^{D} x_i}{(D-1)}$

where $D = 30$ and $x_i \in [0, 1]$.

5) Test problem 5 (ZDT 2):
Min $f_1(x) = x_1$
Min $f_2(x) = g(x)[1 - (x_1 / g(x))^2]\hspace{1cm} (9)$
$g(x) = 1 + 9 \cdot \frac{\sum_{i=2}^{D} x_i}{(D-1)}$

where $D = 30$ and $x_i \in [0, 1]$.\hspace{1cm} (9)
6) Test problem 6 (ZDT3):
\[
\begin{align*}
\min f_1(x) &= x_1 \\
\min f_2(x) &= g(x) \left[ 1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right] \\
g(x) &= 1 + 9 \sum_{i=2}^{D} \frac{x_i}{(D-1)} \\
\end{align*}
\]
where \(x_i \in [0, 1]\).

7) Test problem 7 (ZDT4):
\[
\begin{align*}
\min f_1(x) &= x_1 \\
\min f_2(x) &= g(x) \left[ 1 - \sqrt{\frac{f_1(x)}{g(x)}} \right] \\
g(x) &= 1 + 10(D-1) + \sum_{i=2}^{D} (x_i^2 - 10 \cos(2\pi x_i))^{0.25} \\
\end{align*}
\]
where \(D = 10\), \(x_i \in [0, 1]\) and \(x_i \in [-5, 5] \quad (i = 2, \ldots, D).\)

8) Test problem 8 (ZDT6):
\[
\begin{align*}
\min f_1(x) &= 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\
\min f_2(x) &= g(x) \left[ 1 - \left( \frac{f_1(x)}{g(x)} \right)^2 \right] \\
g(x) &= 1 + 9 \left( \sum_{i=2}^{D} x_i / (D-1) \right)^{0.25} \\
\end{align*}
\]
where \(D = 10\) and \(x_i \in [0, 1]\).

All MOEAs are run for a maximum of 25000 fitness function evaluations (FES). For real-coded NSGA-II, we use a population size of 100, crossover probability of 0.9 and mutation probability of \(1/n\), where \(n\) is the number of decision variables, distribution indexes for crossover and mutation operators as \(\eta_c = 20\) and \(\eta_m = 20\) as presented in [16].

The population obtained at the end of 250 generations is used to calculate the performance metrics. PAES uses a depth of four and an archive size of 100. MOPSO uses a population size of 50, a repository size of 100 and 30 divisions for the adaptive grid with mutation as presented in [18]. DMS-MO-PSO uses the following parameter values: 10 sub-swarms with 95% certainty, whereas h value of 0 implies that the performances are not statistically different.

C. Experimental Results

Table 1 and Table 2 show the means and variances of the convergence and diversity metrics values obtained using the four algorithms NSGA-II, PAES, MOPSO and DMS-MO-PSO by repeatedly running 30 times on each problem. The best mean result on each problem is emphasized in boldface. The nonparametric Wilcoxon rank sum tests are conducted between the DMS-MO-PSO’s result and the best result achieved by the other three MOEAs for each problem. The h values are presented in the last rows. An h value of 1 indicates that the performances of the two algorithms are statistically different with 95% certainty, whereas h value of 0 implies that the performances are not statistically different.

DMS-MO-PSO is able to converge better than the other three algorithms except on ZDT1 and ZDT2, where PAES yielded better convergence measure. But according to the diversity metric, it is observed PAES achieves a bad diversity for these two problems. With respect to the diversity measure, DMS-MO-PSO outperforms the other algorithms in all test problems. According to the statistical significance test, DMS-MO-PSO achieved the better performance on most problems for these metrics. The Pareto fronts generated by DMS-MO-PSO are presented in Figure 7.

Table 1. Table Convergence Metric (\(\gamma\)) Comparison of the Four Algorithms

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>SCH</th>
<th>FON</th>
<th>KUR</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>0.0043</td>
<td>0.0021</td>
<td>0.0324</td>
<td>0.0674</td>
<td>0.1897</td>
<td>0.6211</td>
<td>5.1219</td>
<td>3.1209</td>
</tr>
<tr>
<td>MOPSO</td>
<td>0.0045</td>
<td>0.0360</td>
<td>1.0955</td>
<td>0.0006</td>
<td>0.0005</td>
<td>0.0745</td>
<td>3.5907</td>
<td>7.5964</td>
</tr>
<tr>
<td>PAES</td>
<td>0.0004</td>
<td>0.0002</td>
<td>0.0107</td>
<td>0.0246</td>
<td>0.0615</td>
<td>0.0329</td>
<td>2.2526</td>
<td>0.3413</td>
</tr>
<tr>
<td>DMS</td>
<td>0.0004</td>
<td>0.0013</td>
<td>0.0252</td>
<td>0.0189</td>
<td>0.0162</td>
<td>0.0267</td>
<td>5.6413</td>
<td>0.7501</td>
</tr>
</tbody>
</table>

Table 2. Table Diversity Metric (\(\Delta\)) Comparison of the Four Algorithms

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>SCH</th>
<th>FON</th>
<th>KUR</th>
<th>ZDT1</th>
<th>ZDT2</th>
<th>ZDT3</th>
<th>ZDT4</th>
<th>ZDT6</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSGA-II</td>
<td>0.0043</td>
<td>0.0021</td>
<td>0.0324</td>
<td>0.0674</td>
<td>0.1897</td>
<td>0.6211</td>
<td>5.1219</td>
<td>3.1209</td>
</tr>
<tr>
<td>MOPSO</td>
<td>0.0045</td>
<td>0.0360</td>
<td>1.0955</td>
<td>0.0006</td>
<td>0.0005</td>
<td>0.0745</td>
<td>3.5907</td>
<td>7.5964</td>
</tr>
<tr>
<td>PAES</td>
<td>0.0004</td>
<td>0.0002</td>
<td>0.0107</td>
<td>0.0246</td>
<td>0.0615</td>
<td>0.0329</td>
<td>2.2526</td>
<td>0.3413</td>
</tr>
<tr>
<td>DMS</td>
<td>0.0004</td>
<td>0.0013</td>
<td>0.0252</td>
<td>0.0189</td>
<td>0.0162</td>
<td>0.0267</td>
<td>5.6413</td>
<td>0.7501</td>
</tr>
</tbody>
</table>

Test problem SCH is the simplest among the nine problems with only a single variable. All the four algorithms perform well on this problem, and almost get the same convergence measure. However, DMS-MO-PSO performs better than the other three algorithms with respect to the diversity measure. The FON is a two-objective optimization problem with three variables. The Pareto optimal front is a single non-convex curve. It is observed that the performances of the four algorithms are similar. The KUR problem has three disconnected Pareto-optimal regions, which may cause difficulty in finding non-dominated solutions in all regions. DMS-MO-PSO performs well as shown in Figure 7(c), obtaining non-dominated solutions in all regions. ZDT1 is probably the easiest of all of the ZDT problems, the only difficulty an MOEA may face in this problem is the large
number of variables. Though PAES achieves a small convergence metric, it achieves a bad diversity. Non-dominated solutions obtained in DMS-MO-PSO on ZDT2 are shown in Figure 7(e).

The problem ZDT6 is another hard problem. The adverse density of solutions across the Pareto-optimal front, together with the non-convex nature of the front, makes it difficult for many multi-objective optimization algorithms to maintain a well-distributed non-dominated set and converge to the true Pareto-optimal front. In the experiments, we observed that MOPSO, NSGA-II and PAES could not converge to the true Pareto front of ZDT6, while DMS-MO-PSO performs well by converging to the true front with a good spread of solutions along the front as presented in Figure 7(h). As shown in Tables Table 1 and Table 2, the average values of $\gamma$ and $\Delta$ obtained by DMS-MO-PSO on problem ZDT6 are much better than the corresponding performance metrics obtained by the other algorithms.

By comparing all four algorithms on eight test functions, it is observed that DMS-MO-PSO achieves fairly good results on all the eight functions. Except SCH problem, on which all four algorithms obtained good results, DMS-MO-PSO performs better than the other three algorithms especially on the complex problems ZDT4 and ZDT6.

About the computational complexity, DMS-MO-PSO has the similar complexity with NSGAII since the main time-consuming schedule for these two algorithms is updating the external archive and DMS-MO-PSO used the similar technique as NSGAII. Through it is a little slower than MOPSO and PAES, comparing the results, DMS-MO-PSO has much better performance. And in many real problems, the most time-consuming part is the cost functions calculations. In such a case, the computational complexity of these four algorithms can all be omitted.

V. CONCLUSION

In this paper, the DMS-PSO algorithm was extended to solve multi-objective optimization problems with an external archive and a novel phbest and lbest updating strategy. From the analysis, it is observed updating phbest or lbest just because the new solution is not dominated by the phbest or lbest is not enough and sometimes it will lead the particles to fly in a wrong direction. By judging the domination relationship of the new solution and iteratively updating the reference front, the phbest and lbest are updated in a more reasonable way. Non-dominated sorting and crowding distance sorting also help the DMS-MO-PSO to have a better external archive.

We evaluated the proposed approach on eight test problems currently adopted in the literature. The proposed DMS-MO-PSO significantly outperforms other three representative multi-objective evolutionary algorithms, mainly on larger
dimensional problems. It also demonstrates a good performance when solving a multimodal problem, ZDT4. Although NSGA-II has no external archive, it combines the parent and offspring populations, which has the same effect as external archive to avoid missing the non-dominated solutions. DMS-MO-PSO, MOPSO and PAES all incorporate external archive. To improve diversity of the non-dominated solutions, DMS-MO-PSO and NSGA-II use crowding distance. PAES and MOPSO use adaptive grids. But, DMS-MO-PSO performs the best among these four algorithms, which demonstrates that the self-learning strategy employed in DMS-MO-PSO is as effective as in single-objective optimization when dealing with multi-objective optimization problem. In brief, DMS-MO-PSO is an effective multi-objective evolutionary algorithm capable of converging to the true Pareto optimal front and maintaining a good diversity along the Pareto front. In the future work, a fast sorting mechanism proposed in [27] will be combined in DMS-MO-PSO to decrease the computational complexity. Decomposition-based method and ensemble strategy [28] will also be considered in the future study.

ACKNOWLEDGMENT

This work is partially supported by National Natural Science Foundation of China (Grant No. 60905039, 71001072) and Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) (No. 20114101110005).

REFERENCES