Engineering Optimization
Vol. 00, No. 0, Month 2010, 1–17

Taylor & Francis
Taylor & Francis Group

Two-*lbests* based multi-objective particle swarm optimizer

S.Z. Zhao and P.N. Suganthan*

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

(*Received day month year; final version received day month year*)

The global best (*gbest*) or local best (*lbest*) of every particle in state-of-the-art multi-objective particle swarm optimization (MOPSO) implementations is selected from the non-dominated solutions in the external archive. This approach emphasizes the elitism at the expense of diversity when the size of the current set of non-dominated solutions in the external archive is small. This article proposes that the *gbest*s or *lbests* should be chosen from the top fronts in a non-domination sorted external archive of reasonably large size. In addition, a novel two local bests (*lbest*) based MOPSO (2LB-MOPSO) version is proposed to focus the search around small regions in the parameter space in the vicinity of the best existing fronts unlike the current MOPSO variants in which the *pbest* and *gbest* (or *lbest*) of a particle can be located far apart in the parameter space thereby potentially resulting in a chaotic search behavior. Comparative evaluation using 19 multi-objectives test problems and 11 state-of-the-art multi-objective evolutionary algorithms overall ranks the 2LB-MOPSO as the best while two state-of-the-art MOPSO algorithms are ranked the worst with respect to other multi-objective evolutionary algorithms.

Keywords: multi-objective evolutionary algorithm; multi-objective particle swarm optimization; two *lbest*s multi-objective particle swarm optimization (2LB-MOPSO); external archive

1. Introduction

Particle swarm optimization (PSO) is a heuristic search technique proposed by Kennedy and Eberhart (1995) who were inspired by the choreography of bird flocks, animal herds and fish schools. The c_1 and c_2 in the velocity update equation in Figure 1 are the acceleration constants, which represent the weighting of stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. *rand1()* and *rand2()* are two uniform random numbers between 0 and 1.

$$X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$$

represents the position of the ith particle;

$$pbest_i = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD})$$

represents the best previous position of the ith particle with the best fitness value;

$$gbest = (gbest_1, gbest_2, \ldots, gbest_D)$$

*Corresponding author. Email: epnsugan@ntu.edu.sg

ISSN 0305-215X print/ISSN 1029-0273 online

DOI: 10.1080/03052151003686716
http://www.informaworld.com

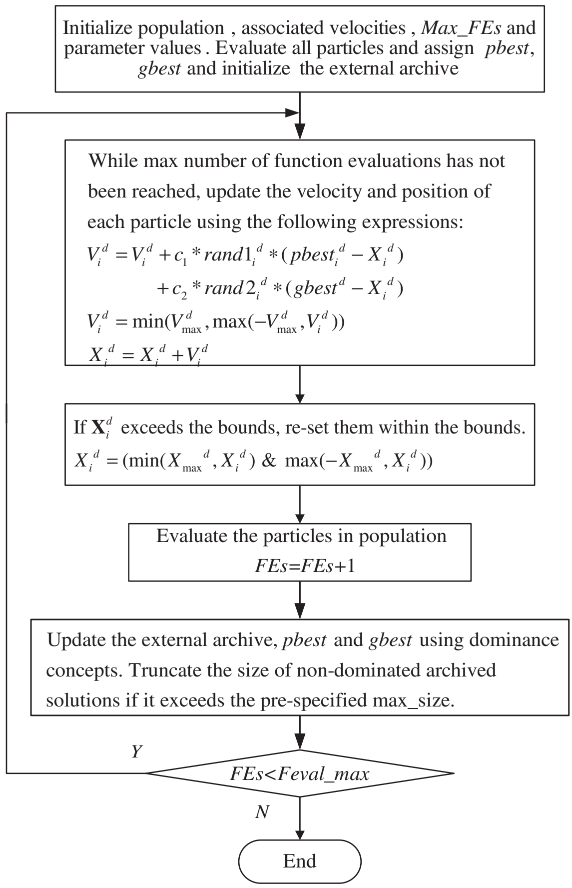

Figure 1. The flowchart of a typical MOPSO implementation.

represents the best position obtained by the swarm so far;

$$V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$$

represents the rate of the positional change (velocity) of particle i.

The relative simplicity of the PSO and the fact that it is a population-based technique has made it a natural candidate to be extended for solving multi-objective optimization problems.

In this article, it is pointed out that all MOPSO implementations select the global best (*gbest*) or local best (*lbest*) from only the non-dominated solutions in the external archive in the current iteration (Abido 2007, Branke and Mostaghim 2006, Coello Coello *et al*. 2004, Huang *et al*. 2006, Ireland *et al*. 2006, Sierra and Coello Coello 2005, 2006, Zielinski and Laur 2007). This approach emphasizes the elitism at the expense of diversity with possible adverse consequences due to the potential loss of diversity if the size of the non-dominated solutions is very small for several consecutive iterations. This article proposes that the *gbest*s or *lbest*s should be selected from the top fronts in a non-domination sorted external archive of reasonably large size. A novel two local bests (*lbest*s) based MOPSO version (2LB-MOPSO) is also proposed to effectively focus the search around a small region in the parameter space in the vicinity of the best existing fronts whereas in the current MOPSO variants, the *pbest* and *gbest* (or *lbest*) of a particle can be far apart in the parameter space due to the random nature of their selection process thereby potentially resulting in a chaotic search process.

The remainder of the article is organized as follows. Section 2 reviews the literature on MOPSO in order to highlight that the *gbest* selection is predominantly confined to the current set of non-dominated solutions in the external archive. In Section 3, the proposed two *lbest*s based multi-objective particle swarm optimizer (2LB-MOPSO) is presented. Section 4 presents experimental procedures while simulation results are presented in Section 5. Section 6 draws conclusions.

2. Multi-objective particle swarm optimization (MOPSO)

Several optimization problems have many objectives conflicting with each other. As there is no single solution for these problems, the aim is to find Pareto optimal trade-off solutions representing the best possible compromises among all objectives. A multi-objective optimization problem (MOP) with m conflicting objectives can be defined as in Deb (2001):

Minimize/maximize

$$\mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})), \quad \mathbf{x} \in [\mathbf{Xmin}, \mathbf{Xmax}]$$

subject to:

$$g_j(\mathbf{x}) \leq 0, \; j = 1, \ldots, J$$

$$h_k(\mathbf{x}) = 0, \; k = 1, \ldots, K$$

where $\mathbf{x}$ is the decision vector and $\mathbf{y}$ is the objective vector. Different from the single objective optimization, there are two spaces to be considered. One is the decision space denoted as $\mathbf{x}$ and the other is the objective space denoted as $\mathbf{y}$. Some basic concepts with respect to multi-objective optimization are defined below:

Definition 1 Pareto dominance *For any two-objective vectors* $\mathbf{u}$ *and* $\mathbf{v}$, $\mathbf{u}$ *is said to dominate* $\mathbf{v}$, *if*

(1) $\mathbf{u}$ *is no worse than* $\mathbf{v}$ *in all objectives.*
(2) $\mathbf{u}$ *is strictly better than* $\mathbf{v}$ *in at least one objective.*

Definition 2 Non-dominated set *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* $\mathbf{P}$.

Definition 3 Pareto optimality *When the set* $\mathbf{P}$ *is the entire feasible search space, the resulting non-dominated set* $\mathbf{P}'$ *is called the Pareto-optimal solution set.*

In iteration, all solutions are sorted based on non-domination concept into different fronts. The solutions belonging to the first front (front 1) are not dominated by any solution in the current iteration. The solutions belonging to the second front are not dominated by any solution in the current iteration except solutions belonging to front 1. After sorting all solutions into fronts, each solution is assigned with a rank value as the front number to which the solution belongs.

Development of multi-objective evolutionary algorithms (MOEAs) has attracted much interest and several MOEAs have been proposed (Coello Coello *et al.* 2002, Deb 2001, Deb *et al.* 2002, Knowles and Corne 2000). While these algorithms were developed to achieve fast convergence to the Pareto-optimal front with good distribution of solutions along the front, each algorithm employs a combination of techniques to achieve these goals. The main advantage of EAs in solving multi-objective optimization problems is their ability to find multiple Pareto-optimal solutions in one single run. As particle swarm optimizers (PSO) also have this ability, several multi-objective particle swarm optimizers (MOPSOs) (Abido 2007, Branke and Mostaghim 2006, Coello Coello *et al*. 2004, Huang *et al*. 2006, Ireland *et al*. 2006, Ray and Liew 2002, Sierra and Coello Coello 2005, 2006, Zielinski and Laur 2007) have been proposed.

When solving single-objective optimization problems, the guides (*i.e. pbest*, *lbest*, *gbest*) used by each particle to update its position are completely determined once a neighbourhood topology is chosen. However, when solving multi-objective optimization problems, due to the existence of numerous non-dominated solutions, each particle can have a set of guides from which one or more can be selected in order to update its position. All MOPSO implementations maintain an external archive to hold the current set of non-dominated solutions. In order to hold the best solutions of a fixed size in the external archive, the non-domination sorting combined with either the crowding distance sorting (Deb 2002) or clustering (Coello Coello *et al.* 2004) or binning (Sierra and Coello Coello 2005) are applied to the external archive in MOPSO variants. The non-dominated solutions contained in the external archive are used as *lbest* or *gbest* guides when the positions of the particles in the swarm are updated. Figure 1 shows a typical MOPSO implementation. The external archive only stores the non-dominated front 1 solutions in all current MOPSO implementations. Since the size of the repository is usually limited to the finally required approximation solution set size, whenever the front 1 solution set size exceeds the finally required approximation solution set size, an appropriate strategy (Coello Coello *et al.* 2004, Sierra and Coello Coello 2005) is used to truncate the archive size by giving priority to solutions located in less populated areas of objective space over those in highly populated regions.

Branke and Mostaghim (2006) allowed each particle to memorize all non-dominated personal best particles it has encountered. If the updated personal best position did not dominate the old one, they kept both in the personal archive. Ireland *et al*. (2006) compared a new principal ‘Centroid’ method with the Sigma method for selecting the guide particles from front 1 solutions only. Jiang *et al.* (2007) introduced a particle angle division method for finding the global best particle from the current set of non-dominated solutions for each particle in the population.

Abido (2007) presented a two-level of non-dominated solutions approach to MOPSO using more than one external archive. All external archives store only non-dominated individuals and all dominated solutions are deleted from the external archives. To avoid the effort associated with choosing control parameter settings, Zielinski and Laur (2007) presented an adaptive approach for parameter setting of an MOPSO. Multi-objective comprehensive learning particle swarm optimizer (MOCLPSO) (Huang *et al.* 2006), which extended the CLPSO (Liang *et al*. 2006) has also demonstrated competitive performance against several other MOEAs (Coello Coello *et al.* 2004, Deb *et al.* 2002, Knowles and Corne 2000). All these MOPSO implementations select guides from external archives having only the non-dominated solutions in the current iteration.

The vector evaluated particle swarm optimization (VEPSO) algorithm (Parsopoulos *et al.* 2004) has been inspired by the concept of the vector evaluated genetic algorithm (VEGA) (Schaffer 1985). In VEPSO, there are M swarms corresponding to each of the M objectives. Each swarm optimizes just one objective very much like the single objective PSO, except the *gbest* of one swarm is selected from the *gbest*s of other swarms. However, once a solution has the best objective value with respect to one objective, that solution must belong to the current non-dominated solution set. Hence, VEPSO also selects the *gbest*s from non-dominated solutions only. Further, instead of considering all non-dominated solutions as the potential *gbest*s, VEPSO considers only the end points of the current non-dominated solutions.

In Ray and Liew (2002), the individuals of a swarm update their flight directions through their neighbouring leaders selected from the non-dominated solutions in the population. When the number of front 1 solutions is no larger than half of the population size, they assign individuals with less than or equal to the average rank of the whole population to be the set of leaders (SOL); when the number of front 1 solutions is larger than half of the population size, the SOL is selected only from front1 solutions. This algorithm without an external archive is referred to as SMM (Ray and Liew 2002) and included in comparison with the proposed algorithm in Subsection 5.4. It should be observed that the SMM may be regarded as a hybrid MOPSO because, in SMM, leaders and new positions are combined, non-domination sorted and truncated to the size of population. The top half of this truncated population is selected as the SOL and not evaluated in the next generation, the other half acquire information from the SOL and move to new positions through a simple generational operator. In all MOPSOs, all particles continuously fly through the search space without a selection process and non-dominated solutions are captured in the external archive.

3. Two-*lbests* based multi-objective particle swarm optimizer

In this section, first the non-domination sorted external archive is briefly presented. Subsequently, the proposed novel two-*lbests* based MOPSO with non-domination sorted external archive will be presented.

3.1. *Non-domination sorted external archive*

An external archive is commonly used to store the non-dominated solutions obtained in the search process by an MOPSO. Since the size of the non-dominated set can be very large and the complexity of updating the archive increases with the archive size, the size of the archive is usually restricted to a pre-specified size which is usually the same as the finally required approximation solution set size. To truncate the external archive in the proposed MOPSO implementation, the procedure used in the NSGA–II (Deb *et al.* 2002) to truncate NSGA-II's population is employed.

In the proposed MOPSO implementation, the initialized archive includes all initialized solutions at iteration 1. In every iteration, all new positions $Q(t)$ generated in iteration t is combined with the members in the archive $P(t)$ to obtain the mixed-temporary external archive. The non-domination sorting is applied to this mixed-temporary archive to obtain the sorted archive $R(t)$. During this process, all the sorted solutions retain two indicators, namely, the front rank and crowding distance value (Deb *et al.* 2002). The sorted solutions with the lowest front rank (Deb *et al.* 2002) will be included in the archive $P(t+1)$ first. When the size of the archive hits the permitted maximum size of archive $|P(t+1)|(=|P(t)|)$, the crowding distance (Deb 2001) is applied to select the required number of members to be included in $P(t+1)$ from the lowest front that still remains unselected in archive $R(t)$.

3.2. *Two-lbests based multi-objective particle swarm optimizer*

In most MOPSO algorithms, particles fly in directions in the search space determined by *pbest* and *gbest* positions. In reality, these positions can be far apart from each other in the parameter space. Therefore, guiding the directions of acceleration of particles by using two far apart *pbest* and *gbest* positions may not be very effective when solving hard problems. Therefore, this article proposes two *lbest*s based MOPSO variant. In this variant, each objective function range in the external archive is divided into a number of bins (*n_bins*). The two *lbest*s are chosen from the external archive members located in two neighbouring bins, so that they are near each other in the parameter space. In order to select the first *lbest* for a particle, an objective is first randomly selected followed by a random selection of a non empty bin of the chosen objective. Within this bin, the archived member with the lowest front number and among these with the highest crowding distance is selected as the first *lbest*. The second *lbest* is selected from solutions in the neighbourhood non empty bins with the lowest front number and the smallest Euclidean Distance in the parameter space to the first *lbest*. As each particle is guided by two *lbest*s from

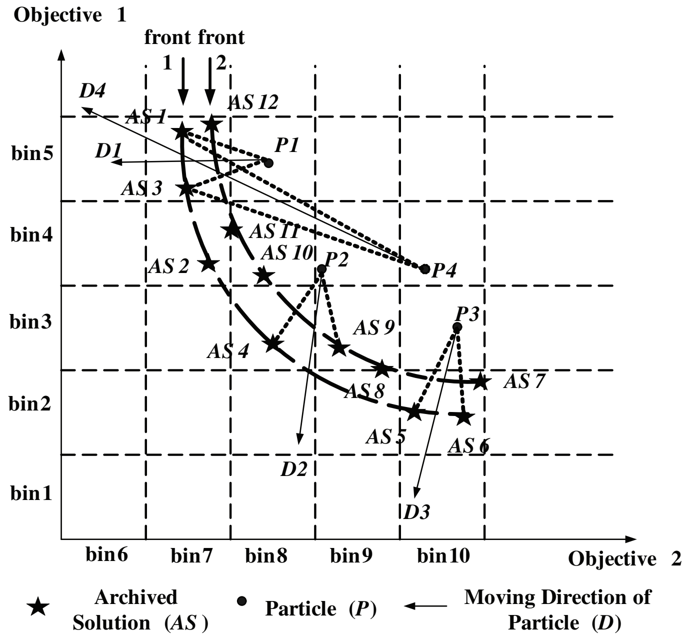

Figure 2. An illustration of two *lbests* based search process with minimization objectives.

a neighbourhood in parameter and objective spaces with the smallest front number, the velocity updating of each particle will be in the direction between the positions of two *lbest*s and oriented to improve upon the current non-dominated solutions as shown in Figure 2.

Given that the first *lbest* of every particle is chosen somewhat randomly, it will not be effective if every particle is assigned with a new pair of *lbest*s which come from the different pair of bins in every iteration, because the flight of each particle will be almost random in this case. Therefore, after assigning a pair of *lbest*s to a particle, the number of iterations the particle fails to contribute a solution to the archive $P(t)$ is counted. If the *count* exceeds a pre-specified threshold, the particle is reassigned with another pair of *lbest*s. During the initialization stage and when *count* is larger than the pre-specified threshold during the iterative optimization stage, the first *lbest* for a particle is chosen randomly by selecting an objective and one bin of the objective. The second *lbest* is chosen from the neighbourhood of the first *lbest* in the parameter space. Otherwise, when *count* is less than or equal to the pre-specified threshold during the iterative optimization stage, two *lbest*s are chosen from the same assignment of the objective and the bin as used in the last iteration. The particle will accelerate potentially in a direction between the two *lbest*s. Hence, this particle may explore the region of the two *lbest*s. Figure 2 illustrates both of the two cases during the iterative search.

In Figure 2, a two-objective minimization problem with **Objective1** and **Objective2** is shown. Each objective range in the archive $P(t)$ is equally divided. $P(t)$ has 12 archived solutions in two fronts, among them, archived solutions (***AS***) 1 to 6 are non-dominated front 1 solutions, the other six are front 2 solutions.

The four particles namely ***P1*** to ***P4*** fly in directions guided by their corresponding two *lbest*s. For example, assuming that the *count* for ***P4*** exceeds the pre-specified threshold in the current iteration, when selecting the two *lbests* for the particle ***P4***, it is best to randomly select one objective and one bin for its first *lbest*. Here, objective 1 and bin 5 are assumed to be selected for the first *lbest* of ***P4***. Among the two candidates ***AS1*** and ***AS3*** with the lowest front number in bin5, the ***AS1*** is chosen as the first *lbest* as it has the larger crowding distance. The ***AS3*** is selected to be the second *lbest* of ***P4*** as it is nearest to the ***AS1*** in the neighborhood in the parameter

Table 1. The pseudo-code of the 2LB-MOPSO.

Initialize the swarm: Initialize *NP* number of particles randomly and uniformly in the D-dimensional search space. Evaluate the fitness values of all particles. Set all current positions to be $P(0)$, the external archive. Select *lbests* from the $P(0)$ external archive using the steps within 'If $count(j) > 5$' below.

Optimization Loop:

For $i = 1$ to *Feval_max* // *iteration loop*

For $j = 1$ to *NP* // *iteration loop for updating velocity, position of each particle*

If *count(j)* > 5 // *updating lbests* (*more details in Subsection 3.2*)

For particle j, the first *lbest* is selected by (a) randomly selecting an objective, (b) randomly choosing a bin of the selected objective, (c) choosing the solution with the lowest front number and largest crowding distance in the chosen bin. The second *lbest* is selected from the neighbourhood of the first *lbest* in the parameter space.

Else

For particle j, update the two *lbests* in the same bins used in the last iteration. The first *lbest* should have the lowest front number and the largest crowding distance in the bin. The second *lbest* is selected from the neighbourhood of the first *lbest* in the parameter space.

Endif

$V(j) = \omega * (V(j) + c_1 * rand() * (lbest(j) - particle(j)) + c_2 * rand() * (lbest(j + NP) - particle(j)))$ // *updating velocity*

$V_i(d) = (\min(V_{\max}(d), V_i(d))\ \&\ \max(-V_{\max}(d), V_i(d)))$ //*Limit the velocity for each dimension*

$X_i(d) = X_i(d) + V_i(d)$ //*updating position of each particle*

If any dimension exceeds the search space

Set the corresponding bound value to be the dimension value of *particle*(j);

Endif

Evaluate the fitness values of*particle*(j)

feval_count = *feval_count* + 1

EndFor j

Do non-dominated sorting on combined $P(t)$ and new particles $Q(t)$ to obtain external archive $P(t + 1)$.

Update *count* of each particle (*count* is the number of iterations the particle fails to contribute a solution to the external archive as explained in Subsection 3.2).

Stop if a stop criterion is satisfied (here is when *feval_count* reached the *Feval_max*).

EndFor i

space as well as the objective space with the lowest front number. Hence, the ***P4*** will accelerate in direction ***D4*** in the next iteration. Assuming the first *lbest* of ***P2*** is assigned to objective 1 and bin3 in the previous iteration, and the *count* for ***P2*** does not exceed the pre-specified threshold in the current iteration. ***AS4*** is selected as the first *lbest* because it has the lowest front member in bin3. From the neighbouring solutions with the lowest front members in each bin, ***AS9*** with the smallest parameter space Euclidean Distance to***AS4*** will be chosen as the second *lbest.* Hence, ***P2*** will accelerate in ***D2*** guided by the ***AS4*** and ***AS9*** as the first *lbest* and second *lbest* respectively. Assuming that the *count*s for ***P1 and P3*** also do not exceed the pre-specified threshold, following the same rules, ***AS1*** and ***AS3*** can be chosen as the first *lbest* and second *lbest* for ***P1*** as well as ***AS6*** and ***AS5*** can be chosen as the first *lbest* and second *lbest* for ***P3***. Table 1 presents the pseudo-code of the 2LB-MOPSO.

4. Experimental procedures

4.1. *MOEA variants and benchmark problems*

19 multi-objective problems (Huband *et al*. 2006, Huang *et al.* 2007, Rudolph *et al.* 2007) are used in the experiments. Among these problems, the first seven are two-objective problems, the next

six are three-objective problems and the last six are five-objective problems. The proposed 2LB-MOPSO is compared with two MOPSO variants and seven MOEA variants. Eight of these MOEA variants were from the CEC07 Special Session and Competition on 'Performance Assessment on Multi-objective Optimization Algorithms' (Huang *et al.* 2007) on the multi-objective problems. The algorithms used in the experimental comparisons are:

(1) Proposed 2LB-MOPSO: Two *Lbests* based MOPSO. 2LB-MOPSO was coded in Matlab language. Population size:

$$NP = 50, \omega = 0.729; c_1 = c_2 = 2.05; V_{\max} = 0.25(X_{\max} - X_{\min}); n\_bin = 10$$
$$V = (2 * rand(NP, 1) - 1) * (0.25. * (X_{\max} - X_{\min})).$$

(2) MOCLPSO (Multi-objective Comprehensive Learning Particle Swarm Optimizer) (Huang *et al.* 2006): The same parameter settings as in the original MOCLPSO article were used. Population size:

$$N = 50. \text{ Inertia weight updating, for } k = 1 \text{ to } maxgen$$
$$\omega(k) = \frac{(\omega_0 - \omega_1) \times (\max gen - k)}{\max gen} + \omega_1 \quad and \quad \omega_0 = 0.9, \omega_1 = 0.2.$$

Acceleration rate, $c1 = c2 = 2.05$. Velocity definition, $V_{\max} = 0.25 * (X_{\max} - X_{\min})$. Learning probability $P_c = 0.1$, elitism probability $P_m = 0.4$. The original Matlab codes of MOCLPSO were used. As MOCLPSO outperforms the MOPSO, MOCLPSO is chosen to replace MOPSO (Coello Coello *et al.* 2004) from the comparisons.

(3) MO_PSO: In this article, adaptive control parameter setting is employed with a multi-objective particle swarm optimization algorithm (Zielinski and Laur 2007).
(4) NSGA2_SBX: In this work, NSGA-II with SBX crossover is employed (Sharma *et al.* 2007). The sequential quadratic programming (SQP) method is used as a local search procedure.
(5) NSGA2_PCX: A hybrid multi-objective optimization approach using NSGA-II and PCX crossover is employed (Kumar *et al.* 2007). Sequential quadratic programming (SQP) is used as the local search procedure.
(6) GDE3: In this work, the generalized differential evolution 3 is applied to solve the multi-objective optimization problems (Kukkonen and Lampinen 2007).
(7) MO_DE: A multi-objective differential evolution algorithm with an adaptive approach for setting control parameters is used (Zielinski and Laur 2007).
(8) MOSaDE: This multi-objective optimization algorithm uses self-adaptive differential evolution algorithm which automatically adapts the trial vector generation strategies and their associated parameters according to their previous experience of generating promising or inferior individuals (Huang *et al.* 2007).
(9) DEMOwSA: This multi-objective differential evolution algorithm employs self adaptation strategies to adapt F and CR parameters of DE (Zamuda *et al.* 2007).
(10) MTS: Multiple trajectory search for multi-objective optimization (MTS) uses multiple agents to search the solution space concurrently (Tseng and Chen 2007). Each agent performs an iterated local search using one of four candidate local search methods.

4.2. *Parameter setting*

In order to be comparable to the CEC07 'Performance Assessment on Multi-objective Optimization Algorithms', for the other two MOPSOs (2LB-MOPSO and MOCLPSO (Huang *et al.* 2006)) which are not included in the CEC07 Special Session and Competition, the maximum of function evaluations (MAX_FES) with 25 independent runs is set at 500,000 and the results are recorded three steps after 5000, 50,000 and 500,000 function evaluations. The archive sizes are set as 100 for two-objective problems, 150 for three-objective problems and 800 for five-objective problems. The maximum number of non-dominated solutions used to calculate the performance metrics is the same as the archive size for all the MOEAs. All the above parameter settings are exactly the same as in the other MOEA variants from the CEC07 Special Session and Competition.

4.3. *Performance measures*

In order to compare the performances of the above MOEAs, quantitative performance metrics are needed. There are two goals in a multi-objective optimization: (1) convergence to the true Pareto-optimal front and (2) distribution of approximated solutions. The following two metrics were used and the metrics were computed using normalized objective values. The same performance measures were used in the CEC07 'Performance Assessment on Multi-objective Optimization Algorithms' (Huang *et al.* 2007).

(1) R indicator (I_{R2}) (Knowles *et al.* 2006):

$$I_{R2} = \frac{\sum_{\lambda \in \Lambda} u^*(\lambda, A) - u^*(\lambda, R)}{|\Lambda|}$$

where R is a reference set, u^* is the maximum value reached by the utility function u with weight vector λ on an approximation set A, *i.e.*, $u^* = \max_{z \in A} u_\lambda(z)$. The augmented Tchebycheff function was used as the utility function. The R indicator shows how much the approximation set A has to be translated/scaled so that it covers the reference set R.

(2) Hypervolume difference to a reference set ($I_{\bar{H}}$) (Knowles *et al.* 2006): The hypervolume indicator I_H measures the hypervolume of the objective space that is weakly dominated by an approximation set A, and is to be maximized. Here, the hypervolume difference to a reference set R is used and refered to this indicator as $I_{\bar{H}}$, which is defined as $I_{\bar{H}} = I_H(R) - I_H(A)$ where smaller values correspond to higher quality – in contrast to the original hypervolume I_H. The hypervolume indicator possesses the following two properties. On the one hand, it is sensitive to any type of improvements, *i.e.* whenever an approximation set A dominates another approximation set B, then the measure yields a strictly better quality value for the former than for the latter set. On the other hand, the hypervolume measure guarantees that any approximation set A that achieves the maximally possible quality value for a particular problem contains all Pareto-optimal objective vectors.

The R indicator and the hypervolume indicator capture both convergence to the Pareto front and diversity of the approximated set of solutions.

5. Discussions of experimental results

The simulation results of tuning parameters *count* and *n_bin* are reported in Subsection 5.1. In Subsection 5.2, the main experimental results obtained by using the proposed 2LB-MOPSO on all the 19 test problems are presented and compared with the results of the CEC 2007 competition. The competition considered the speed of convergence and the quality of the final solutions by recording the results after 5000, 50,000 and 500,000 function evaluations. Different algorithms are compared by ranking the mean results. In order to perform the comparison, the results as reported in the corresponding publications of algorithms 3 to 10 in the CEC 2007 competition are copied. The associated parameter values of these eight algorithms are presented in the respective publications. The parameter settings related to the multi-objective optimization problems are presented in Subsection 4.2 and these are the same for all 10 algorithms. M indicates the number of objectives in the test problem in each comparison. Additional experiments were conducted in Subsection 5.3 to compare the performance when the two *lbest*s are selected from (a) a non-domination sorted external archive of a reasonably large size, and (b) only non-dominated (front 1) solutions. Experiments were also conducted to show the benefit of selecting *gbest* (or *lbest*) from the neighbourhood in both parameter space and objective space and the advantage of retaining the *gbest* or *lbest* for every particle in the same neighbourhood for a few iterations (*i.e. count*). Finally, in Subsection 5.4, SMM (Ray and Liew 2002) and 2LB-MOPSO are compared by testing on the same test problems as used in the SMM. In Tables 2 to 6, the best results and the best ranks are highlighted in bold.

5.1. *Parameter sensitivity studies*

In 2LB-MOPSO, there are *count* and *n_bins* parameters defined in Section 3.2 to be tuned. In this section, out of these 19 multi-objectives problems, six problems are used to investigate the impact of the parameters. The six problems are No. 2 SYMPART (two-objectives), No. 7 S_ZDT6 (two-objectives), No. 9 R_DTLZ2 (three-objectives), No. 10 S_DTLZ3 (three-objectives), No. 18 WFG8 (five-objectives) and No. 19 WFG9 (five-objectives). The 2LB-MOPSO is run for a maximum of 500,000 function evaluations (FES) with 25 independent runs. The results are recorded after 5000, 50,000 and 500,000 function evaluations. For each of these test problems, the mean value of the R and H indicator achieved by the 25 runs are ranked among all different parameter values used.

5.1.1. *Parameter sensitivity study on count*

As explained in Section 3.2, the *count* value is used as a pre-specified threshold to decide when to reassign a particle to another pair of *lbest*s if the particle fails to generate a solution to enter the external archive consecutively for *count* number of iterations. Experiments were conducted with *count* set to 3, 4, 5 and 6 with 25 independent runs and results are recorded after 5000, 50,000 and 500,000 function evaluations. Mean values of R and H indicators obtained with different *count* on all six test functions are ranked. Two indicators on three searching steps sum up to six sets of comparisons. In order to save space, only one of them is shown in Table 2. In addition, Figure 3 shows the summation of all the rankings of R and H indicators on three searching steps of each *count* value on all six test functions. It can be observed that *count* can influence the results. But, the results are not highly sensitive as the values are approximately similar as can be seen from Table 2. For the six test functions, better results were obtained when *count* is around 5 as shown in Figure 3. Hence, in the proposed 2LB-MOPSO, the *count* is set at 5 for all test problems.

Table 2. R_indicator of parameter study on count of 2LB-MOPSO on 500 k function evaluation.

FEs	*Count*	2.SYMPART	7.S_ZDT6	9.R_DTLZ2	10.S_DTLZ3	18.WFG8	19.WFG9	Rank
5e + 5	3	2.5709e−06(5)	5.5397e−04(5)	2.4273e−04(5)	1.2059e−05(4)	−5.3581e−03(5)	8.0232e−04(2)	26
	4	2.1907e−06(4)	4.0070e−04(4)	2.2154e−04(4)	1.1977e−05(3)	−5.5447e−03(4)	1.0624e−03(4)	23
	5	1.3453e−06(3)	3.5901e−04(3)	1.8229e−04(**1**)	1.1170e−05(**1**)	−5.7425e−03(3)	7.4545e−04(**1**)	12
	6	9.0367e−07(**1**)	2.9691e−04(2)	1.9760e−04(2)	1.1204e−05(2)	−6.2084e−03(**1**)	8.7012e−04(3)	**11**
	7	9.6147e−07(2)	2.8393e−04(1)	2.1844e−04(3)	1.4117e−05(5)	−6.0251e−03(2)	1.2411e−03(5)	18

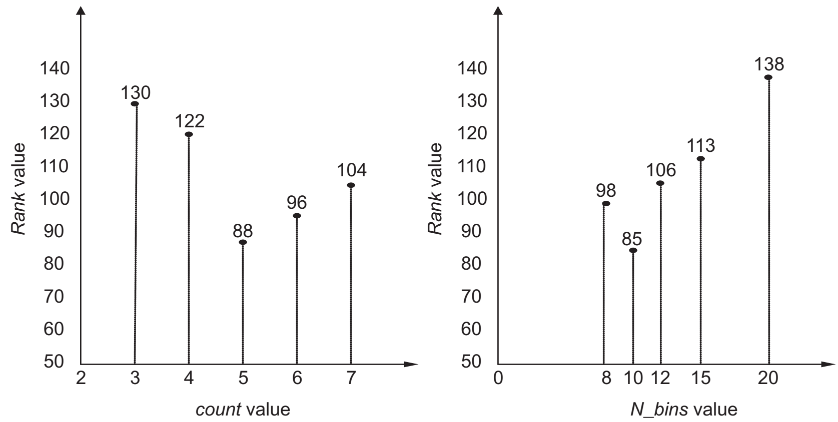

Figure 3. Ranking results of parameter sensitivity investigation on count and *n_bins*.

Table 3. R_indicator of parameter study on *n_bins* of 2LB-MOPSO on 500 k function evaluation.

FEs	*n_bins*	2.SYMPART	7.S_ZDT6	9.R_DTLZ2	10.S_DTLZ3	18.WFG8	19.WFG9	Rank
5e + 5	8	1.3878e−06(5)	2.8428e−04(**1**)	2.3503e−04(5)	1.4967e−05(2)	−4.4631e−03(5)	9.5691e−04(2)	20
	10	1.3453e−06(4)	3.5901e−04(2)	1.8229e−04(**1**)	1.1170e−05(**1**)	−5.7425e−03(**1**)	7.4545e−04(**1**)	**10**
	12	1.2506e−06(2)	3.8362e−04(3)	1.9750e−04(2)	1.7923e−05(3)	−5.7410e−03(2)	1.1002e−03(5)	17
	15	1.3317e−06(3)	4.1344e−04(4)	2.0156e−04(3)	1.8445e−05(5)	−5.0425e−03(3)	9.8550e−04(3)	21
	20	1.1305e−06(**1**)	4.2775e−04(5)	2.1909e−04(4)	1.8279e−05(4)	−4.8530e−03(4)	1.0535e−03(4)	22

5.1.2. *Parameter sensitivity study on n_bin*

The number of bins has to be assigned with an appropriate numerical value to coarsely partition each objective for the selection of the *lbest*s. Experiments were conducted with the *n_bins* set to 8, 10, 12, 15 and 20 with 25 independent runs. The results are recorded after 5000, 50,000 and 500,000 function evaluations. From Figure 3 and Table 3, it can be observed that *n_bins* also influences the results, whereas they are not very sensitive as the differences among the values are even smaller than those in *count* investigation in Subsection 5.1.1. For the six test problems, better results were obtained when the *n_bins* is 10. Hence, in the proposed 2LB-MOPSO, the *n_bins* is set at 10 for all test problems.

5.2. *Comparison with MOEAs in CEC 2007 competition*

The mean values of R and H indicators among all MOEAs on the 19 test problems are ranked in the comparison results (the full comparison results and the source codes of 2LB-MOSPO in Matlab are available from: http://www3.ntu.edu.sg/home/EPNSugan/). According to CEC 2007 competition, the results at three steps (5000, 50,000 and 500,000 function evaluations) are considered in the comparison. The ranks of 5000 and 50,000 function evaluations show that 2LB-MOPSO not only performs with fast convergence speed, but also has a good diversity of the solutions which is the main shortcoming of most MOPSO variants. From the ranks of 500,000 function evaluations, the 2LB-MOPSO can almost approximate the Pareto optimal front on most of these 19 test problems except one difficult problem S_ZDT4. It can also be observed that the proposed 2LB-MOPSO overall achieves the best solution quality over all MOEAs.

Table 4 shows all ranks of indicators for every algorithm on two, three and five-objective test problems. Out of the nine cases, obviously, the proposed 2LB-MOPSO can achieve three first

Table 4. Summation of ranks based on $R\&H$ indicators on all test problems. For example, the total '22' for 2LB-MOPSO at FEs $=$ 5000 and M $=$ 2 is obtained by adding '11' (A1, R and FEs $=$ 5000) and '11' (A1, H and FEs $=$ 5000) in the last column. The 'Overall Summation' of 114 (M $=$ 2) for the 2LB-MOPSO is obtained by adding 22 (M $=$ 2, FEs $=$ 5000), 40 (M $=$ 2, FEs $=$ 50,000) and 52 (M $=$ 2, FEs $=$ 500,000). In (), the ranks vertically across different MOEAs are shown.

	FEs = 5000 $(R + H)$			FEs = 50,000 $(R + H)$			FEs = 500,000 $(R + H)$			Overall summation		
Algorithm	M = 2	M = 3	M = 5	M = 2	M = 3	M = 5	M = 2	M = 3	M = 5	M = 2	M = 3	M = 5
2LB-MOPSO	**22**	**18**	**37**	**40**	**30**	**40**	**52**	**42**	**52**	**114**	**90**	**129**
	(1)	**(1)**	**(2)**	**(2)**	**(1)**	**(3)**	**(2)**	**(2)**	**(3)**	**(1)**	**(1)**	**(2)**
MOCLPSO	120	79	60	118	92	80	129	100	87	369	271	227
	(10)	(7)	(5)	(10)	(10)	(8)	(10)	(10)	(9)	(10)	(10)	(6)
MO_PSO	84	80	96	106	88	95	111	88	96	301	256	287
	(6)	(8)	(9)	(9)	(9)	(10)	(9)	(9)	(10)	(9)	(9)	(10)
NSGA2_SBX	**35**	**49**	**33**	**42**	**49**	**27**	63	**50**	**18**	**140**	**148**	**78**
	(2)	**(3)**	**(1)**	**(3)**	**(3)**	**(1)**	(4)	**(3)**	**(1)**	**(2)**	**(3)**	**(1)**
NSGA2_PCX	94	68	**41**	93	73	77	71	71	81	258	212	199
	(8)	(5)	**(3)**	(6)	(5)	(6)	(6)	(7)	(7)	(6)	(5)	(5)
GDE3	77	61	68	**32**	**31**	**37**	**46**	**27**	**40**	**155**	**119**	**145**
	(5)	(4)	(6)	**(1)**	**(2)**	**(2)**	**(1)**	**(1)**	**(2)**	**(3)**	**(2)**	**(3)**
MO_DE	93	81	85	98	79	78	78	70	84	269	230	247
	(7)	(9)	(7)	(8)	(7)	(7)	(7)	(6)	(8)	(7)	(6)	(8)
MOSaDE	75	103	99	59	88	91	**58**	63	66	192	254	256
	(4)	(10)	(10)	(4)	(8)	(9)	**(3)**	(5)	(4)	(4)	(8)	(9)
DEMOwSA	108	**49**	94	97	54	72	65	62	66	270	165	232
	(9)	**(2)**	(8)	(7)	(4)	(5)	(5)	(4)	(5)	(8)	(4)	(7)
MTS	**61**	72	47	91	76	63	99	87	70	251	235	180
	(3)	(6)	(4)	(5)	(6)	(4)	(8)	(8)	(6)	(5)	(7)	(4)

rank positions, four second rank positions and two third rank positions on both R indicator and H indicator among all MOEAs on these 19 test problems. According to the ranks in Table 4, except the proposed 2LB-MOPSO, there is no other MOEA to be within the top three always. In addition, the other two recent MOPSO variants (MOCLPSO and MOPSO) almost perform the worst when compared with the other MOEAs. Therefore, the original contributions have made the 2LB-MOPSO attain the best ranking among all the MOEAs while the two other state-of-the-art MOPSO variants were the worst among all the MOEAs in the CEC 2007 competition. NSGA2_SBX and GDE3 were declared the joint winners. In Table 4 within 'Overall Summation' column, NSGA2_SBX and GDE3 have overall aggregate rankings of 6 and 8 respectively, while the 2LB-MOPSO has an overall aggregate ranking of 4.

5.3. *Comparison among 2LB-MOPSO variants*

In order to demonstrate the benefits of selecting the *gbest* or *lbest* from a neighbourhood in both parameter and objective spaces of a non-domination sorted external archive of large size, as well as the advantage to retaining the *gbest* or *lbest* of every particle in the same neighbourhood for a few iterations, further experiments are conducted on three modified versions of the proposed 2LB-MOPSO.

(1) 2LB-MOPSO_M1: Two *Lbests* based MOPSO with *count* $= 1$ as all current MOPSO variants in the literature. Selecting the two *lbest*s from two random bins, and the archive only includes non-dominated front 1 solutions. This version is similar to most MOPSO variants except having two *lbest*s instead of one *gbest* and one *pbest* for each particle.
(2) 2LB-MOPSO_M2: Two *Lbests* based MOPSO with *count* $= 1$. Selecting the two *lbest*s from two random bins of the non-domination sorted external archive of a reasonably large size. This version can show the benefit of non-domination sorted external archive with a reasonably

Table 5. R_indicator obtained by the variants *of* 2LB-MOPSO at 500 k function evaluations.

Variants	2.SYMPART	7.S_ZDT6	9.R_DTLZ2	10.S_DTLZ3	18.WFG8	19.WFG9	Rank
2LB-MOPSO_M1	1.1222e−04(4)	6.4552e−02(4)	3.8002e−04(4)	2.1906e−04(4)	1.2664e−03(4)	1.8550e−03(4)	24
2LB-MOPSO_M2	6.8550e−05(2)	2.8462e−03(3)	1.6950e−04(**1**)	9.0054e−05(3)	−4.1225e−03(3)	1.7650e−03(3)	15
2LB-MOPSO_M3	7.2885e−05(3)	2.3142e−03(2)	1.7703e−04(2)	8.1804e−05(2)	−8.6358e−03(**1**)	1.3703e−03(2)	12
2LB-MOPSO	1.3453e−06(**1**)	3.5901e−04(**1**)	1.8229e−04(3)	1.1170e−05(**1**)	−5.7425e−03(2)	7.4545e−04(**1**)	**9**

large size when comparing to 2LB-MOPSO_M1 with only front 1 solutions in the external archive.

(3) 2LB-MOPSO_M3: Two *Lbests* based MOPSO with *count* = 1. The two *lbest*s are selected from two neighbouring bins of the non-domination sorted external archive. This version can show the benefit of selecting the *gbest* or *lbest* from the external archive within a neighbourhood in both parameter space and objective space. When comparing the proposed 2LB-MOPSO to the 2LB-MOPSO_M3, the advantage of retaining the *gbest* or *lbest* of every particle in the same neighbourhood for a few iterations as determined by the *count* parameter can be observed.

The mean values of 25 runs for *R* indicator achieved by the three modified versions of 2LB-MOPSO and the proposed 2LB-MOPSO at 5e+5 function evaluations are compared on six problems, which are No. 2 SYMPART (two-objectives), No. 7 S_ZDT6 (two-objectives), No. 9 R_DTLZ2 (three-objectives), No. 10 S_DTLZ3 (three-objectives), No. 18 WFG8 (five-objectives) and No. 19 WFG9 (five-objectives). Table 5 ranks these four different versions based on the mean values of *R* indicator and add the rankings together as presented in the last column. From the summation of rankings in the last column of Table 5, improvements due to the three contributions in the proposed 2LB-MOPSO can be easily observed. Hence, it is obvious that each of the novel contributions plays a clear role in making the proposed 2LB-MOPSO to perform substantially better than the state-of-the-art MOPSO variants.

5.4. *Comparison with SMM*

An experimental comparison is performed to highlight the performance differences between SMM and 2LB-MOPSO algorithms. Four test problems from SMM (Ray and Liew 2002) are considered in this comparison. The first function is a two-objective minimizing problem with a discontinuous Pareto optimal front. The second problem is the modelling for the design of a welded beam, which is a two-objective minimization problem. The third problem designs a multiple disc brake, which is a two-objective minimizing problem. The last problem deals with the designing of a four bar truss which is a two-objective minimizing problem. All problem definitions are given in Appendix A. As some of these problems are constrained, the epsilon-constraint handling method presented by Laumanns *et al.* (2006) is applied to solve them. To compare the SMM and the proposed 2LB-MOPSO, the population size, the number of Pareto optimal solutions found and the function evaluations required are shown in Table 6. As 100 is used as the archive size for two-objective problems, 2LB-MOPSO easily achieves full size of the Pareto optimal solutions by using a population size of 50 and smaller number of function evaluations. However, the SMM fails to evolve the whole population to be the front 1 solutions. The obtained non-dominated fronts are show in Figure 4 in which for ease of comparison, the non-dominated fronts published in SMM are reproduced. The proposed 2LB-MOPSO can obtain better quality front 1 curves with good spread, while the front 1 curves presented in SMM comparatively lack diversity frequently. For the first problem, 2LB-MOPSO can obtain four discontinuous front 1 curves with good spreads, while the SMM finds 40 front 1 solutions out of 100 population with almost six times function

Table 6. Comparison between SMM and 2LB-MOPSO.

	Population size		Number of Front 1 solution		Max function evaluations	
Problem	SMM	2LB-MOPSO	SMM	2LB-MOPSO	SMM	2LB-MOPSO
1	100	**50**	40	**100**	29,572	**5000**
2	100	**50**	42	**100**	18,389	**8000**
3	100	**50**	52	**100**	6385	**2000**
4	100	**50**	91	**100**	2525	**1000**

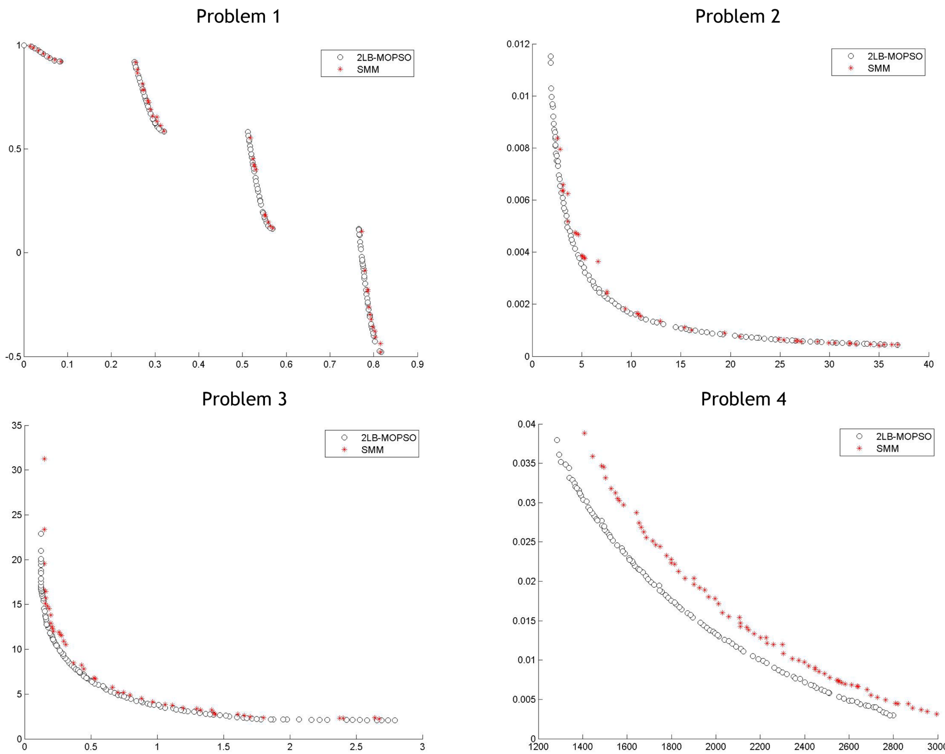

Figure 4. Pareto fronts obtained by the 2LB-MOPSO and SMM.

evaluations of 2LB-MOPSO. For the second problem, with less than half of function evaluations used in SMM, 2LB-MOPSO is able to find the required number of front 1 solutions, while SMM find only 52 feasible solutions. For the third problem, it can easily be found that 2LB-MOPSO can achieve more feasible solutions with better diversity than SMM by using much smaller number of function evaluations. For the last problem, 2LB-MOPSO obtains totally dominating solutions when comparing to the SMM with only 0.2 times the function evaluations used in SMM.

6. Conclusion

This article pointed out that almost all multi-objective particle swarm optimization (MOPSO) implementations currently available in the literature used exclusively the non-dominated solutions in the external archive of the current iteration as the *gbest* or *lbest* of every particle. As this approach emphasizes the élitism at the expense of diversity, this approach has the potential to converge on a small section of the true Pareto front if the size of the non-dominated set is small for several iterations during the evolution. In order to resolve this flaw, the *gbest* or *lbest* of every particle are chosen from a non-domination sorted archive of reasonably large size. As no current MOPSO variant takes into consideration the actual positions of the two guides, it is possible that the two guides can be located far apart in the parameter and/or objective space. In this situation, the search behaviour of the MOPSO can potentially become chaotic. To resolve this, a novel two local best (*lbest*s) based MOPSO is proposed in which the two *lbest*s are located close to each other in the vicinity of the best front sections. Since in all MOPSO variants, the *gbest* or *lbest* of each particle was chosen independently in each iteration. This process also results in chaotic search behaviour. The threshold *count* was included in the 2LB-MOPSO to keep the *gbest*s or *lbest*s in a particular region for a few iterations. Therefore, the 2LB-MOPSO was able to focus the search around small regions in the vicinity of the best front sections while maintaining the diversity due to the usage of two *lbest*s and binning. Simulation results demonstrate the improved performances due to the original contributions.

Acknowledgements

The authors acknowledge the financial support offered by A * Star (Agency for Science, Technology and Research) under grant number 052 101 0020.

References

Abido, M.A., 2007. Two-level of non-dominated solutions approach to multi-objective particle swarm optimization. *Proceedings of the genetic and evolutionary computation conference (GECCO-2007)*, date, London, UK. Place: Publisher, 00–00. **Q1**

Branke, J. and Mostaghim, S., 2006. About selecting the personal best in multi-objective particle swarm pptimization. *Proceedings of the 9th international conference on parallel problem solving from nature (PPSN IX)*, date, Reykjavik, Iceland, Place: Publisher, 523–532. **Q2**

Coello Coello, C.A., Van Veldhuizen, D.A. and Lamont, G.B., 2002. *Evolutionary algorithms for solving multi-objective problems*. New York: Kluwer Academic Publishers.

Coello Coello, C.A., Pulido, G.T. and Lechuga, M.S., 2004. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8 (3), 256–279.

Deb, K., 2001. *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: Wiley.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2), 182–197.

Schaffer, J.D., 1985. Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of 1st international conference on genetic algorithms*, date, place. Place: Lawrence Erlbaum, 93–100. **Q3**

Huband, S., Hingston, P., Barone, L. and While, L., 2006. A review of multi-objective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10 (5), 477–506.

Huang, V.L., Suganthan, P.N. and Liang, J.J., 2006. Comprehensive learning particle swarm optimizer for solving multi-objective optimization problems. *International Journal of Intelligent Systems*, 21, 1–18.

Huang, V.L., Qin, A.K., Suganthan, P.N. and Tasgetiren, M.F., 2007. Multi-objective optimization based on self-adaptive differential evolution algorithm. *Proceedings of the IEEE congress on evolutionary computation*, date, Singapore. Place: Publisher, 3601–3608. **Q4**

Huang, V.L., Qin, A.K., Deb, K., Zitzler, E., Suganthan, P.N., Liang, J.J ., Preuss, M. and Huband, S., 2007. Problem definitions for performance assessment on multi-objective optimization algorithms [online]. Technical Report. Nanyang Technological University, Singapore. Available from: http://www.ntu.edu.sg/home/epnsugan [Accessed day month year]. **Q5**

Ireland, D., Lewis, A., Mostaghim, S. and Lu, J.W., 2006. Hybrid particle guide selection methods in multi-objective particle swarm optimization. *2nd IEEE international conference on e-science and grid computing*, date, place. Place: Publisher, 116–116. **Q6**

Jiang, Q., Huang, M. and Wang, C., 2007. A novel method for finding good local guides in multi-objective particle swarm optimization. *3rd international conference on natural computation*, 3, date, place. Place: Publisher, 737–741. **Q7**

Knowles, J.D. and Corne, D.W., 2000. Approximating the nondominated front using the Pareto archive evolutionary strategy. *Evolutionary Computation*, 8 (2), 149–172.

Knowles, J., Thiele, L. and Zitzler, E., 2006. A tutorial on the performance assessment of stochastic multi-objective optimizers. TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland.

Kukkonen, S. and Lampinen, J., 2007. Performance assessment of Generalized Differential Evolution 3 (GDE3) with a given set of problems. *Proceedings of the IEEE congress on evolutionary computation*, date, Singapore. Place: Publisher, 3593–3600.

Kumar, A., Sharma, D. and Deb, K., 2007. A hybrid multi-objective optimization procedure using PCX based NSGA-II and sequential quadratic programming. *Proceedings of the IEEE congress on evolutionary computation*, date, Singapore. Place: Publisher, 3011–3018.

Eberhart, R.C. and Kennedy, J., 1995. A new optimizer using particle swarm theory. *Proceedings of the 6th international symposium on micromachine and human science*, date, Nagoya, Japan. Place: Publisher, 39–43.

Liang, J.J., Qin, A.K., Suganthan, P.N. and Baskar, S., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10 (3), 281–295.

Laumanns, M., Thiele, L. and Zitzler, E., 2006. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169 (3), 932–942.

Parsopoulos, K.E., Tasoulis, D.E. and Vrahatis, M.N., 2004. Multi-objective optimization using parallel vector evaluated particle swarm optimization. *Proceedings of international conference on artificial intelligence and applications (IASTED)*, date, Innsbruck, Austria. Place: Publisher, 00–00.

Ray, T. and Liew, K.M., 2002. A swarm metaphor for multi-objective design optimization. *Engineering Optimization*, 34 (2), 141–153.

Rudolph, G., Naujoks, B. and Preuss, M., 2007. Capabilities of EMOA to detect and preserve equivalent Pareto subsets. *Evolutionary multi-criterion optimization (EMO 2007)*, 4403, date, place. Berlin: Springer, 36–50.

Sharma, D., Kumar, A., Deb, K. and Sindhya, K., 2007. Hybridization of SBX based NSGA-II and sequential quadratic programming for solving multi-objective optimization problems. *Proceedings of the IEEE congress on evolutionary computation*, date, Singapore. Place: Publisher, 3003–3010.

Sierra, M.R. and Coello Coello, C.A., 2005. Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. *3rd international conference on evolutionary multi-criterion optimization (EMO 2005)*, LNCS 3410, date, Mexico. Place: Springer-Verlag, 505–519.

Sierra, M.R. and Coello Coello, C.A., 2006. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2 (3), 287–308.

Tseng, L.-Y. and Chen, C., 2007. Multiple trajectory search for multiobjective optimization. *Proceedings of the IEEE congress on evolutionary computation (CEC 2007)*, date, Singapore. Place: Publisher, 3609–3616.

Zamuda, A., Brest, J., Boskovic, B. and Zumer, V., 2007. Differential evolution for multiobjective optimization with self adaptation. *Proceedings of the IEEE congress on evolutionary computation (CEC 2007)*, date, Singapore. Place: Publisher, 3617–3624.

Zielinski, K. and Laur, R., 2007. Adaptive parameter setting for a multi-objective particle swarm optimization algorithm. *Proceedings of IEEE congress of evolutionary computation (CEC 2007)*, date, Singapore. Place: Publisher, 3585–3592.

Zielinski, K. and Laur, R., 2007. Differential evolution with adaptive parameter setting for multi-objective optimization. *Proceedings of IEEE congress of evolutionary computation (CEC 2007)*, date, Singapore. Place: Publisher, 3585–3592.

Appendix A

Function 1

$$
\begin{aligned}
&\textit{Minimize} && f_1(x) = x_1 \\
&\textit{Minimize} && f_2(x) = (1 + 10x_2)\left(1 - \left(\frac{x_1}{1 + 10x_2}\right)^2 - \frac{x_1 \sin(8\pi x_1)}{1 + 10x_2}\right) \\
&\textit{subjective to} && 0 \le x \le 1
\end{aligned}
$$

Function 2

$$
\begin{aligned}
&\textit{Minimize} && f_1(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2) \\
&\textit{Minimize} && f_2(x) = \delta(x) = \frac{2.1952}{x_3^3 x_4} \\
&\textit{subjective to} && \tau(x) - 13{,}600 \le 0 \\
& && \sigma(x) - 30{,}000 \le 0 \\
& && x_1 - x_4 \le 0 \\
& && P - P_C(x) \le 0
\end{aligned}
$$

$$
\tau(x) = \sqrt{(\tau')^2 + (\tau'')^2 + (x_2 \tau' \tau'')/\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}}
$$

$$\tau' = \frac{6,000}{\sqrt{2}x_1x_2}$$

$$\tau'' = \frac{6,000(14+0.5x_2)\sqrt{0.25(x_2^2+(x_1+x_3)^2)}}{2\{0.707x_1x_2(x_2^2/12+0.25(x_1+x_3)^2)\}}$$

$$\sigma(x) = \frac{504,000}{x_3^2x_4}$$

$$P_C(x) = 64,746.022(1-0.0282346x_3)x_3x_4^3$$

where $0.125 \leq x_1 \leq 5.0$, $0.1 \leq x_1 \leq 10.0$, $0.1 \leq x_1 \leq 10.0$ and $0.125 \leq x_1 \leq 5.0$.

Function 3

$$\begin{aligned}
&\textit{Minimize} && f_1(x) = 4.9\times10^{-5}(x_2^2-x_1^2)(x_4-1)\\
&\textit{Minimize} && f_2(x) = \frac{9.82\times10^6(x_2^2-x_1^2)}{x_3x_4(x_2^3-x_1^3)}\\
&\textit{subjective to} && (x_2-x_1)-20\geq 0\\
& && 30-2.5(x_4+1)\geq 0\\
& && 0.4-\frac{x_3}{3.14(x_2^2-x_1^2)}\geq 0\\
& && 1-\frac{2.22\times10^{-3}x_3(x_2^3-x_1^3)}{(x_2^2-x_1^2)^2}\geq 0\\
& && \frac{2.66\times10^{-2}x_3x_4(x_2^3-x_1^3)}{(x_2^2-x_1^2)}-900\geq 0
\end{aligned}$$

where $55 \leq x_1 \leq 80$, $75 \leq x_1 \leq 110$, $1000 \leq x_1 \leq 3000$ and $2 \leq x_1 \leq 20$.

Function 4

$$\begin{aligned}
&\textit{Minimize} && f_1(x) = L(2x_1+\sqrt{2}x_2+\sqrt{x_3}+x_4)\\
&\textit{Minimize} && f_2(x) = \frac{FL}{E}\left(\frac{2}{x_1}+\frac{2\sqrt{2}}{x_2}-\frac{2\sqrt{2}}{x_3}+\frac{2}{x_4}\right)\\
&\textit{subjective to} && F/\sigma \leq x_1 \leq 3F/\sigma\\
& && \sqrt{2}F/\sigma \leq x_2 \leq 3F/\sigma\\
& && \sqrt{2}F/\sigma \leq x_3 \leq 3F/\sigma\\
& && F/\sigma \leq x_4 \leq 3F/\sigma
\end{aligned}$$

where $F = 10\,kN$, $E = 2.0e05kn/cm^2$, $L = 200cm$ and $\sigma = 10kN/cm^2$