**Article:**

Cheng, T., Chen, M., Fleming, P.J. orcid.org/0000-0001-9837-8404 et al. (2 more authors) (2017) A novel hybrid teaching learning based multi-objective particle swarm optimization. Neurocomputing, 222. pp. 11-25. ISSN 0925-2312

https://doi.org/10.1016/j.neucom.2016.10.001

# A novel hybrid teaching learning based multi-objective particle swarm optimization

Tingli Cheng[1]*, Minyou Chen[1], Peter J. Fleming[2], Zhile Yang[3], Shaojun Gan[3]
[1] School of Electrical Engineering, Chongqing University, Chongqing, China
[2] Department of Automatic Control & System Engineering, University of Sheffield, Sheffield, UK
[3] School of Electronics, Electrical Engineering and Computer Science, Queens University Belfast, Belfast, UK

**Abstract:** How to obtain a good convergence and well-spread optimal Pareto front is still a major challenge for most meta-heuristic multi-objective optimization (MOO) methods. In this paper, a novel hybrid teaching learning based particle swarm optimization (HTL-PSO) with circular crowded sorting (CCS), named HTL-MOPSO, is proposed for solving MOO problems. Specifically, the new HTL-MOPSO combines the canonical PSO search with a teaching-learning-based optimization (TLBO) algorithm in order to improve search ability and speed up search procedure. Also, CCS technique is developed to improve the diversity and spread of solutions when truncating the external elitism archive. The performance of HTL-MOPSO algorithm was tested on several well-known benchmarks problems and compared with other state-of-the-art MOO algorithms in respect of convergence and spread of final solutions to the true Pareto front. Also, the individual contributions made by the strategies of HTL-PSO and CCS are analyzed. Experimental results validate the effectiveness of HTL-MOPSO and demonstrate its superior ability to find solutions of better spread and diversity, while assuring a good convergence.

**Keywords**: Multi-objective optimization; particle swarm optimization; teaching learning based optimization; crowded sorting.

## 1 Introduction

Multi-objective optimization (MOO) problem is presented in a great variety of real-life optimization problems with more than one conflict objectives. The main goal of MOO is to achieve a set of optimal solutions that are: 1) as close as possible to the true Pareto front (PF), 2) have good diversity, 3) as well as to spread them evenly. In the past decades, evolutionary algorithms (EAs)[1][2][3][4] have been empirically shown to be suitable and efficient in addressing MOO problems due to their capacity to obtain a series of Pareto-optimal solution approximations in a single run [1]. Subsequently, various EA algorithms for MOO algorithms [6] [7] [8], have been developed and achieved great success.

More recently, Particle Swarm Optimization (PSO), another popular swarm intelligence optimization technique has been widely applied to optimization problems [9][10][11]. Benefits including simple implementation, low computational cost and high efficiency, render PSO effectively in dealing with single objective optimization compared with EAs, and have led to adaptation for tackling MOO problems [12] [13][14] [15]. However, there are problems to be addressed when PSO is extended to multi-objective particle swarm optimization (MOPSO)[16]. The high selection pressure presented in PSO often sacrifices diversity of population during the evolution process [17]and this can lead to more serious undesirable premature convergence or local optima in MOPSO.

To cope up with MOO problems, many strategies have been extensively developed in MOPSO from different respects. In the first category, the authors make use of the dynamic/flexible mechanisms of parameter selection instead of constant parameters in MOPSO to balance the convergence and diversity [15][18]. In [15] the values of inertia weight $w$ and acceleration coefficient $c_1$ decrease linearly while acceleration coefficient $c_2$ increase linearly to balance the exploitation and exploration. But these kinds of improvements do not depart from its roots and still keep the original property of PSO. In the second category, the multiple swarm concepts are incorporated into MOPSO to effectively enhance the exploration capacity and deal with multimodal problem

* Corresponding author.
E-mail address: tingli_cheng@126.com

[19][20][21]. In [20], the number of swarms is dynamically adjusted to help regulating the computational resources at different stages while multiple swarms are encouraged to explore different regions to preserve diversity. However, MOPSO variants with multiple populations need to establish communication channel between subwarms which promote the population diversity with the complicated algorithms structures. Also, the selection of leader for each particle in population provides great benefits to control convergence and promote diversity of solutions [22][23][24][25]. The external elitism archive is often used to store the nondominated solutions where global best (gbest) is chosen from. Random selection is the simplest and least expensive method but usually lacks convergence. The popular crowding distance technique can be applied not only to update the archive but also to select the gbest. But the extreme nondominated solutions with infinite value of crowding distance are given the highest priority to be gbests, which inadvertently decreases diversity in the swarm. At the last category, it is known that hybridizing PSO with other local search technique is an effective mechanism to balance the global and local search ability. In MOLS-MOPSO [26], multi-objective dichotomy line search (MOLS) strategy is activated periodically to enhance local search, improving convergence to the PF. Luo et al. in [27] presents a hybrid multi-objective optimization approach named MO-PSO-EDA. Based on the probability model, each sub-population reproduces new offspring the particle swarm optimization (PSO) algorithm and the estimation of distribution algorithm (EDA) to solve complex problem. Izuia et al. [28] proposes a new multi-objective optimization where gradient-based optimization method is combined with MOPSO to alleviate constraint-handling difficulties. A new memetic algorithm for multi-objective optimization is proposed in [29], which combines the global search ability of particle swarm optimization with a synchronous local search heuristic for directed local fine-tuning. The article [30] presents a new hybrid optimizer in which an innovative optimal particles local search strategy on basis of bound optimization by quadratic approximation algorithm and exterior penalty function method is integrated into MOPSO to improve the convergence performance of PSO, and preserve diversity of non-dominated set. However, many hybrids MOPSO with general-purposed local search still show low convergence precision and undesirable diversity.

Although above MOPSO works have shown a great improvement to search ability, there are still some remaining problems which need to be addressed. The particles in current PSO lack the sufficient information exchange within population. Most of them tend to avoid the premature at the cost of decreasing the convergence speed. They are poor when dealing with multi-frontal problems. Thus, it is difficult for them to simultaneously capture satisfactory optimal solutions with respect to MOO's good convergence and well-spread goals. The motivation behind this work is to obtain better convergence and spread PFs lies in that: 1) based on canonical PSO evolution process guided by personal best and gbest, may we use more population information and give more emphasis on the information exchange between all particles; 2) may the selection operator maintain more sufficient diversity to well spread the obtained solution along the whole PF? Inspired by learning-based optimization algorithm (TLBO) [31] [32], the average value (the center) of population can be used to push whole population to better search space. Also, the solution can learn from others rather than non-dominated solutions only to promote diversity. In addition, the selection operator may consider more diversity issue that the crowded distance impact of reminded solutions after other unselected candidates are deleted.

To these effects we propose a novel effective HTL-MOPSO that incorporated TLBO to PSO evolution, uses a modified selection based on crowded distance and other multiple strategies in order to better approximate real PF. The paper is organized as follows: Section 2 gives description of PSO, TLBO and HTL-PSO techniques. In Section 3, the algorithm details of HTL-MOPSO are introduced, followed by numerical simulations and an analysis of tests on well-known benchmarks in Section 4. Section 5 presents the conclusion and some possible paths for further research.

# 2 Hybrid TLBO based PSO strategy

## 2.1 Particle swarm optimization

As we know, the original idea of PSO algorithm was inspired by food searching process of population-based birds or fish [9]. The basic PSO search process is illustrated in (1) and (2). It is shown that the position of each particle, $x$, is regarded as a potential solution of dimensional space, and is modified by varying velocity, personal particle best (*pbest*) position and global best (gbest) position in whole swarm. pbest represents cognitive ability of every individual, whereas gbest is the individual with optimal fitness denotes experience of whole individuals in the population.

$$v_{new,i} = w*v_i + c_1*r_1*(pbest_i - x_i) + c_2*r_2*(gbest - x_i) \tag{1}$$

$$x_{new,i} = x_i + v_{new,i} \tag{2}$$

In (1) - (2) $x_i$ and $v_i$ present the position and velocity vector of $i_{th}$ solution; $w$ is the inertia weight; $c_1$ and $c_2$ are the acceleration coefficients; $r_1$ and $r_2$ are uniformly random numbers between (0, 1). The values of $w$, $c_1$ and $c_2$ in this study will be determined by using the method in TV-MOPSO [15].

### 2.2 Teaching learning based optimization

TLBO algorithm is a recently proposed technique inspired from the philosophy of teaching and learning process on learners[32]. The freedom of specific parameters determination and good convergence ability makes TLBO effective and efficient in solving optimization problem. Teacher phase and learner phase are two important phases in TLBO as follows:

1) Teacher phase

The best solution in each iteration will be chosen as the teacher $x_{teacher}$, who tries to shift mean of the class (the population), denoted by $M$, toward itself. The teacher will be new mean of the swarm, represented by $M\_{new}$. The solution changes its position according to the difference between the present and the new mean shown as (3) and (4).

$$difference\_mean_i = r_3*(M\_{new} - TF*M) \tag{3}$$

$$x_{new,i} = x_i + difference\_mean_i \tag{4}$$

where $r_3$ is a uniformly distributed random number between (0, 1); $TF$ is teacher factor which is used to decide the value of the mean to be changed, and could be chosen from either 1 or 2 randomly stochastically. All solutions should be re-evaluated after update of each teaching phase in order to determine the better solution from $x_i$ and $x_{new,i}$ for passing to learning phase.

2) Learner phase

Students can update their knowledge through interaction with other students in learning phase. Assume that $j^{th}$ solution in the swarm is randomly selected and the learning procedure of $i^{th}$ ($i \neq j$) solution in a minimization problem $f(x)$ is described as:

$$x_{new,i} = \begin{cases} x_i + r_4*(x_i - x_j), \text{if } f(x_i) < f(x_j) \\ x_i + r_5*(x_j - x_i), \text{otherwise} \end{cases} \tag{5}$$

where $r_4$ and $r_5$ are the random number between (0,1). If $x_{new,i}$ is better than $x_i$, we accept $x_{new,i}$; otherwise $x_i$ is accepted.

3) Modified TLBO

As descried in original TLBO, all solutions need to be evaluated twice every iteration, in teaching phase and learning phase respectively. In order to reduce evaluation complexity and combine the advantage of two phases, a modified TLBO (MTLBO) is proposed where teaching phase and learning phase are incorporated in (6).

$$x_{new,i} = x_i + D_1*diff\_T_i + D_2*diff\_L_i \tag{6}$$

$$diff\_T_i = r_3*(M\_{new} - TF*M) \tag{7}$$

$$diff\_L_i = \begin{cases} r_4*(x_i - x_j), \text{if } f(x_i) < f(x_j) \\ r_5*(x_j - x_i), \text{otherwise} \end{cases} \tag{8}$$

where $D_1$ and $D_2$ denote the coefficient factors of teacher phase and learner phase respectively, inspired from the coefficient factors $c_1$ and $c_2$ in (1) of PSO process.

### 2.3 HTL-PSO evolution process

To improve the search ability of the canonical PSO, our purpose in this section is to incorporate advantages of TLBO into PSO evolution, donated by HTL-PSO. In teacher phase, the random vector between nondominated space and the mean of population guides the population approaches the better search space on the functional landscape increasing exploration at the same times favoring convergence towards the promising basins of attraction. This phase in (3) and (4) is periodically invoked in the first half number of the iterations of PSO to enhance search ability without deteriorating the diversity in the early movements of the particles. Learner phase can improve the information exchange between particles and prevent the premature convergence of the population. In the later particles movement stages, smaller deviation of particle in search space is preferred [15]. Thus, the learner phase is periodically invoked in the second half number of the iterations of PSO to help the algorithm jump out of local optimum and find more diverse nondonimated solutions.

In HTL-PSO evolution, the effective global PSO[21] update is employed for main search and gives a good direction to optimal region; the teacher phase and learner phase in TLBO are activated periodically as auxiliary search techniques to use more population information and exchange more information between all solutions to improve search ability. As the TLBO is periodically invoked, not only globally, PSO is also not disturbed, but also enhances the ability of the whole algorithm. HTL-PSO updating scheme is shown in algorithm 1.

**Algorithm 1** Hybrid PSO/TLBO (HTL-PSO) search process

```
If (t mod INV）>0
    Activate PSO search according to Equation(1) and (2);
Else
   IF t<Gen
     D2=0;
     Select values  for D1;
   Else
     D1=0;
     Select values  for D2;
   Endif
   Activate MTLBO according to Equation (6);
Endif
```

where $t$ denotes current iteration number, Gen is the total iteration, $INV$ is a positive constant to stand the frequency of introducing MTLBO.

# 3 Hybrid teaching learning based multi-objective PSO

### 3.1 Get pbest, gbest and teacher

In PSO, pbest is the best position that an individual has had which corresponds to personal experiences. In HTL-MOPSO, if the present solution dominates the pbest solution, it replaces the latter; otherwise, the pbest solution is kept; if neither of them is dominated by each other, then we select one of them randomly.

To solve the conflicting objectives problem in MOO, nondominated solutions are often maintained in an external archive from which a solution is picked up as the gbest [12] [15]. A simple dynamically weighted sum method is usedto choose gbest for every particle from external archive [33]. The same method with gbest is used to select the teacher solution, i.e. *M_new* in TLBO algorithm.

### 3.2 Circular crowded sorting

The external elitism archive is often used to keep a record of the nondominated solutions obtained in the process of searching. In this study, the selection of gbest and teacher solution is done from this archive. At each iteration, the nondominated solutions from combined population of the present population and the archive are stored in archive. If the size of the archive exceeds the maximum size limit, it needs to be truncated considering diversity and spread preservation.

Deb et al [6] proposed a selector scheme using nondominated ranking and crowded distance sorting with respect to fitness and spread. However, the original crowded sorting (CS) method remains two problems: 1) The sort of density is fine for bi-objective problems, but it is not competitive enough for more than 2 objectives problems. 2) The impact of each solution on the crowded distance of remaining solutions is not taken into consideration. It is the fact that a solution deemed to be crowded may become less crowded when other more crowded solutions in the neighborhood are eliminated. Motivated by this idea, circular crowded sorting (CCS) strategy is proposed to select solutions from nondominated candidates set. For each candidate in archive $H$, we compute the first and second minimum Euclidean distance of it from all other candidates in $H$. The average of these two distances is used as the crowding distance of the candidate. The crowded distance here can avoid that the extreme nondominated solutions with infinite value of crowding distance are given the highest priority to be gbest.

The detailed procedure of CCS is described as follows:

*Step 1*: Calculate the crowding distances for all solutions, and keep a record of the two closest neighbors;

*Step 2*: All candidates are sorted by their crowding distance; the most crowded solution is deleted;

*Step 3*: Only update the crowding distances of the remaining candidates whose two closest neighbors is related with the deleted solution; jump to step 2.

Repeat step 2 and 3 until the size of the remaining solutions satisfies the final scale.

We consider the complexity of a single iteration of CCS. The basic operations and complexities of CCS are described as follows: Now suppose we need to select $N_0$ solutions from $N$ solutions that mean $N$-$N_0$ solutions are deleted. The computational complexity of crowding distance calculation for one solution in the first step is O($N$log$N$). The total complexity of distance calculation for all candidates is and find the most crowded candidate is O($N^2$log$N$+ $N$log$N$)=O( $N^2$log$N$). Hence, the complexity of CCS is less than O( $\sum_{i=0}^{N-N_0}(N-i)^2\log(N-i)$ ).

### 3.3 Implementation of HTL-MOPSO

As described above, the proposed HTL-MOPSO is implemented as follows algorithm 2:

**Algorithm 2** TLBO based multi-objective PSO (HTL-MOPSO) procedure

Initiate population $P$ with size $N$: position, velocity, pbest; External archive *Ex_list* and its maximum size is *EN*;
Save the nondominated solutions of the initial population in *Ex_list*;
**While** termination condition is not reached
  **For** each particle $x_i$ in $P$
    **If** $t$ mod *INV*>0
      Get gbest from *Ex _list*;
      Update the velocity $v_i$;
      Obtain offspring particle $x_{new,i}$ using PSO according to (1) and (2);
    **Else**
      Get the teacher from *Ex _list*;
      Obtain offspring particle $x_{new,i}$ using MTLBO according to (6);
      Get the velocity $v_i=x_{new,i}-x_i$;
    **Endif**
  **Endfor**
  If all($v$<=v_limit) , Polynomial mutation is applied;
  Evaluate all solutions in population $P$;
  Update *pbest* for all solutions in $P$ ;
  *Ex_list*=non_dominated( *Ex_list* $\cup$ $P$);
  Update the *Ex_list*;
  If(size(*Ex _list*)>*EN*), truncates *Ex_list* using CCS;
**Endwhile**
Output the solutions in archive *Ex_list* .

# 4 Simulation experiment

### 4.1 Performance Metric

A quantitative assessment for the performance of proposed HTL-MOPSO algorithm is shown in this section. Inverted generational distance (IGD) in (9) could measure both the convergence and spread of the obtained Pareto solutions along the true Pareto front in a sense [34]. The smaller IGD values imply the better performance.

$$IGD = \frac{\sum_{v \in P^*} d(v,P)}{|P^*|} \tag{9}$$

where $P$ presents the obtained Pareto solution set; $P^*$ is a set of uniformly distributed optimal Pareto solutions; $d(v,P)$ is the minimum Euclidean distance between $v$ and the solutions in $P$;$|P^*|$ is the cardinality of set $P^*$.

### 4.2 Benchmark Test

In this study, the well know benchmarks ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 [6], and DTLZ1, DTLZ2, DTLZ5, DTLZ7 [35] and UF2, UF3, UF7 [34] are used for algorithms assessment. In order to further investigate HTL-MOPSO, the method is compared with other popular MOO algorithms including NSGA-II [6] [36], TV-MOPSO [15], MOEA/D [7] and MOTLBO [31]. In order to evaluate the significance of the results, we need some statistical analysis methods [37][38]. Here, a non-parametric statistical test, the Wilcoxon's rank-sum tests [38] using Matlab statistical toolbox, have been carried out at 5% significant level. P-values calculated for Wilcoxon's rank-sum test on test problems are listed in Table 2, where N/A indicates "not applicable" which means that the corresponding algorithm could not statistically compare with itself in the rank-sum test.

All codes of mentioned algorithms are programmed in MATLAB R2014a by the authors according to corresponding references. The population size and iteration number of all algorithm are set to be 100 and 250 respectively when solving ZDT and DTLZ problem except that iteration number 500 for DTLZ1 . When solving the complicated UF problems, all the algorithms are with a population size of 100 and the iteration number of 1000. Other parameters setting of different algorithms are described in Table 1. To eliminate the randomness, all mentioned algorithms run 30 times independently. The experiment results are shown in Table2, where the sign '-' means that algorithm fails to approximate an acceptable Pareto front at most times among 30 runs.

**Table 1** Parameters setting of listed multi-objective algorithms

| | |
|---|---|
| **Archive size** | Archive size 100 for test problems |
| **Crossover probability** | Crossover probability 0.9 for NSGA-II[6] and 1.0 for MOEA/D[7] |
| **Mutation probability** | Mutation probability 1/*d* for NSGA-II[6], TV-MOPSO[9], MOEA/D[7] and HTL-MOPSO (where *d* is the number of decision variables) |
| **Other parameters** | TV-MOPSO [9], HTL-MOPSO: $w_1$=0.7, $w_2$=0.4,$c_{1i}$=2.5, $c_{1f}$=0.5, $c_{2i}$=0.5,$c_{2f}$=2.5.<br>HTL-MOPSO: *INV*=7 ;<br>1) t<=Gen/2,$D_1$=1.75 ,D2=0;<br>2) t>Gen/2, D1=0, $D_2$=1 |

### 4.3 Experimental results on test problems

**Table 2** Mean (M), variance (Var) values and p-values of IGD on test problems

| algorithms | | NSGA-II | TV-MOPSO | MOEA/D | MOTLBO | HTL-MOPSO |
|---|---|---|---|---|---|---|
| ZDT1 | IGD(M) | 4.88e-03 | 4.58e-03 | 4.60e-03 | 4.68e-03 | **3.88e-03** |
| | Var | 3.61e-08 | 2.15e-08 | 4.00e-08 | 4.12e-08 | 9.36e-10 |
| | p-value | 2.92e-11 | 2.92 e-11 | 2.92 e-11 | 2.91e-11 | N/A |
| ZDT2 | IGD(M) | 5.02e-03 | 4.79e-03 | 4.05e-03 | - | **3.85e-03** |
| | Var | 5.47e-08 | 3.14e-08 | 5.51e-09 | - | 5.29e-10 |
| | p-value | 2.74e-11 | 2.73 e-11 | 9.19e-10 | - | N/A |
| ZDT3 | IGD(M) | 7.40e-03 | 4.84e-03 | 1.06e-02 | 5.28e-03 | **4.82e-03** |
| | Var | 7.27e-05 | 1.11e-08 | 1.93e-08 | 8.47e-07 | 5.90e-09 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | p-value | 2.99e-11 | 5.84e-01 | 2.94e-11 | 1.25e-10 | N/A |
| ZDT4 | IGD(M) | 7.15e-03 | 5.19e+00 | 3.58e-02 | - | **3.99e-03** |
| | Var | 2.95e-06 | 1.65e+01 | 1.57e-03 | - | 3.22e-09 |
| | p-value | 4.88e-11 | 1.40e-11 | 1.88e-11 | - | N/A |
| ZDT6 | IGD(M) | 1.01e-02 | 3.91e-03 | **2.74e-03** | 3.76e-03 | 3.08e-03 |
| | Var | 2.70e-06 | 8.80e-08 | 3.45e-10 | 4.97 e-08 | 3.19e-09 |
| | p-value | 2.34e-12 | 2.18e-12 | N/A | 2.24e-12 | 1.15e-12 |
| DTLZ1 | IGD(M) | 2.57-02 | - | 2.89e-02 | - | **2.06e-02** |
| | Var | 1.02e-02 | - | 6.39e-06 | - | 7.15e-07 |
| | p-value | 2.99e-11 | - | 3.01e-11 | - | N/A |
| DTLZ2 | IGD(M) | 6.86e-02 | 8.26e-02 | 7.15e-02 | 9.11e-02 | **6.02e-02** |
| | Var | 9.56e-06 | 9.23e-06 | 8.95e-07 | 4.51e-05 | 2.69e-06 |
| | p-value | 2.98e-11 | 2.99e-011 | 2.97e-11 | 3.00e-11 | N/A |
| DTLZ5 | IGD(M) | 5.65e-03 | 6.65e-03 | 9.70e-03 | 2.0509e-02 | **3.78e-03** |
| | Var | 1.34e-07 | 4.25e-07 | 2.06e-07 | 3.8006e-06 | 1.04e-07 |
| | p-value | 2.73e-11 | 3.00e-11 | 2.76e-11 | 3.01e-11 | N/A |
| DTLZ7 | IGD(M) | 3.92e-01 | 5.64e-02 | 8.48e-02 | - | **4.35e-02** |
| | Var | 3.09e-02 | 9.83e-06 | 2.11e-04 | - | 8.89e-07 |
| | p-value | 2.99e-11 | 2.99e-11 | 2.98e-11 | 3.01e-11 | N/A |
| UF2 | IGD(M) | 3.03e-02 | 6.72e-02 | **2.85e-02** | 6.17e-01 | 4.71e-02 |
| | Var | 7.61e-05 | 6.12e-05 | 2.12e-05 | 4.19e-03 | 3.54e-05 |
| | p-value | 8.47e-01 | 2.99e-11 | N/A | 3.01e-11 | 7.00e-11 |
| UF3 | IGD(M) | **5.52e-02** | 1.71e-01 | 2.43e-01 | 5.25e-01 | 5.89e-02 |
| | Var | 5.65e-04 | 4.48e-05 | 5.79e-03 | 1.05e-02 | 1.99e-04 |
| | p-value | N/A | 3.01e-11 | 6.06e-11 | 3.02e-11 | 1.68e-04 |
| UF7 | IGD(M) | 2.02e-01 | 8.35e-02 | 1.96e-01 | 1.30e+00 | **3.06e-02** |
| | Var | 2.25e-02 | 3.02e-05 | 1.78e-02 | 1.89e-02 | 2.20e-05 |
| | p-value | 1.74e-05 | 3.00e-11 | 2.70e-2 | 3.02e-11 | N/A |

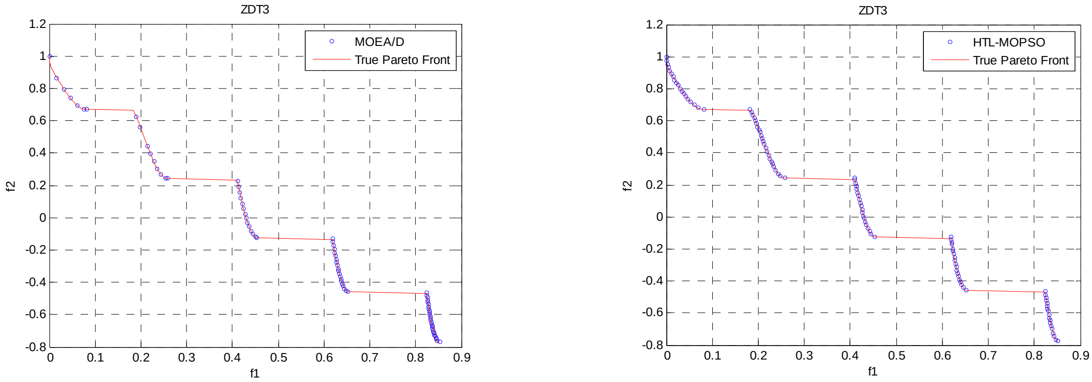
## 4.4 Discussion of the results

### 4.4.1 Experiments results on ZDT and DTLZ problems

Table 2 shows the mean and variance of IGD metrics as well as p-values (Wilcoxon rank-sum test) achieved by aforementioned algorithms including NSGA-II, TV-MOPSO, MOEA/D, MOTLBO and HTL-MOPSO. The reprehensive Pareto front in a run obtained by abovementioned algorithms on ZDT and DTLZ problems are shown in Fig.1-7.

For ZDT1 and ZDT2, all listed algorithms can easily converge to optimal Pareto front presented by low IGD values in Table 2. The p-values indicate HTL-MOPSO is significantly better than other counterparts on ZDT1 and ZDT2 functions, while other four algorithms have similar performance. The worst performance on ZDT2 is MOTLBO, which fails to get a set of solutions.

For ZDT3 whose Pareto-front is disconnected, all five algorithms are able to find the solutions near optimal Pareto front. MOEA/D has poor spread as illustrated by Fig.1 (a). From Table 2, it is observed HTL-MOPSO is the best with respect to IGD values which finds the ideal Pareto front as Fig.1 (b) while TV-MOPSO ranks second. Given that the p-value of TV-MOPSO is greater than 0.05 on ZDT3 in Table 2, the performance of HTL-MOPSO is similar to TV-MOPSO.




(a) (b)

Fig.1 Pareto fronts produces by (a) MOEA/D and (b)HTL-MOPSO on ZDT3

For ZDT4 with a number of local Pareto fronts, lead to significant difficulties to many MOO algorithms. The typical simulation results using all algorithms except MOTLBO on ZDT4 are shown in Fig.2. TV-MOPSO in Fig.2 (b) gets stuck at a local Pareto front far away from optimal Pareto front. MOEA/D in Fig.2 (c) gives a poor convergence even though the produced Pareto front is closer to optimal Pareto front compared with TV-MOPSO. Only NSGA-II in Fig.2 (a) and HTL-MOPSO in Fig.2 (d) are able to escape from different local optima. From Fig.2, we can see that HTL-MOPSO has found best convergence and better spread solutions along the entire Pareto optimal region. The capacity to find well-converged and well-spread of HTL-MOPSO are apparently the best as shown by the lowest values of IGD in Table 2 and the p-values indicate that HTL-MOPSO has superior performance than other counterparts ZDT4 functions.

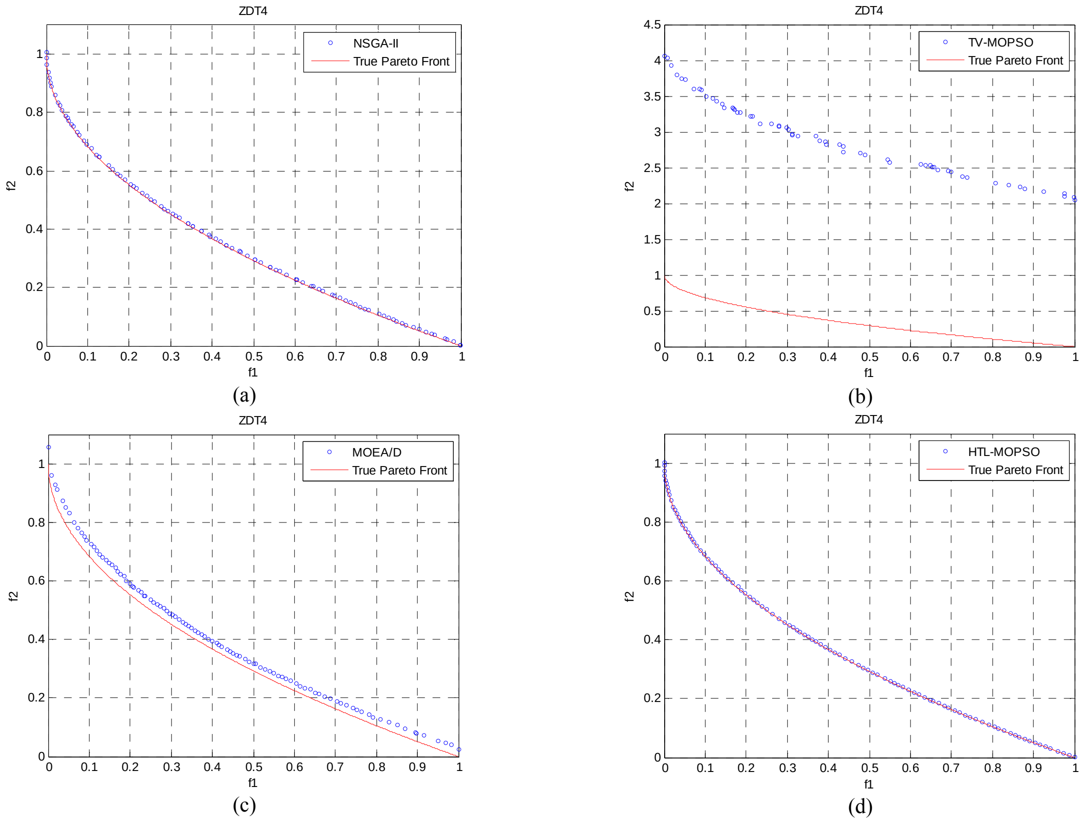



Fig.2. Pareto fronts produces by (a)NSGA-II, (b) TV-MOPSO, (c) MOEA/D and (d)HTL-MOPSO on ZDT4

ZDT6 has the non-convex and non-uniformly spaced Pareto front. The Pareto fronts produced by aforementioned algorithms are presented in Fig.3. It is clearly observed from Table 2 and Fig.3 (a) that NSGA-II gives the poor convergence of achieved Pareto front compared with HTL-MOPSO in Fig.3.(e). In Fig.3 (b), TV-MOPSO consistently fails to push parts of achieved solutions converge to true Pareto front. In most case with ZDT6, HTL-MOPSO displays an outstanding convergence and spread performance as Fig.3(e). Unfortunately, HTL-MOPSO failed to push all Pareto solutions close to true Pareto set in less than 10% among

30 simulation runs as demonstrated in Fig.3 (f). It is because few solutions far away from optimal Pareto front are stored in external elitism archive. The same situations occur in MOTLBO as shown in Fig.3 (d). However, the results in Table 2 and Fig.3 (c) signify that MOEA/D performs the best not only in obtaining close approximation of optimal Pareto set but also maintaining a uniform spread on the region. The HTL-MOPSO performs the second best on ZDT6.

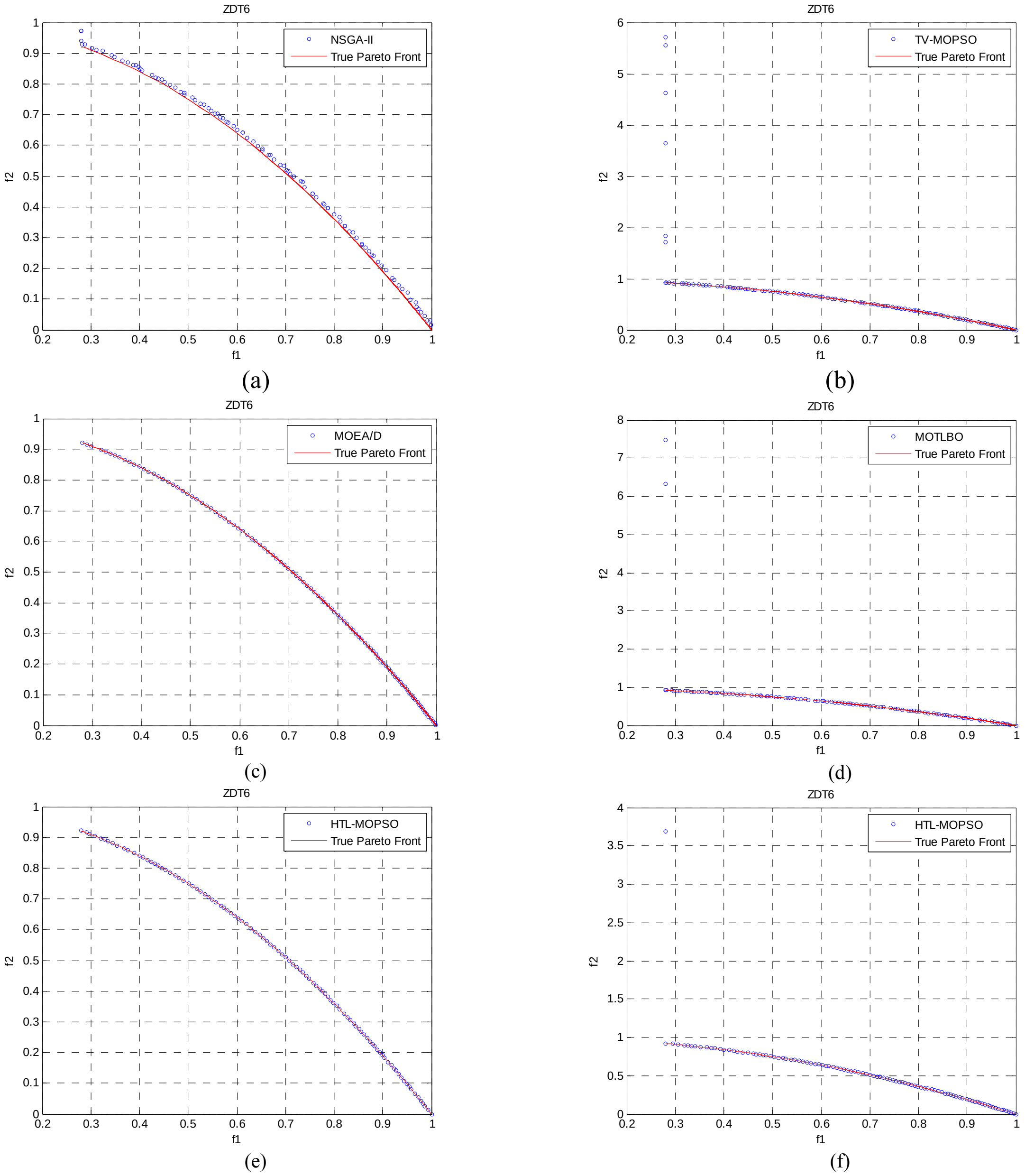
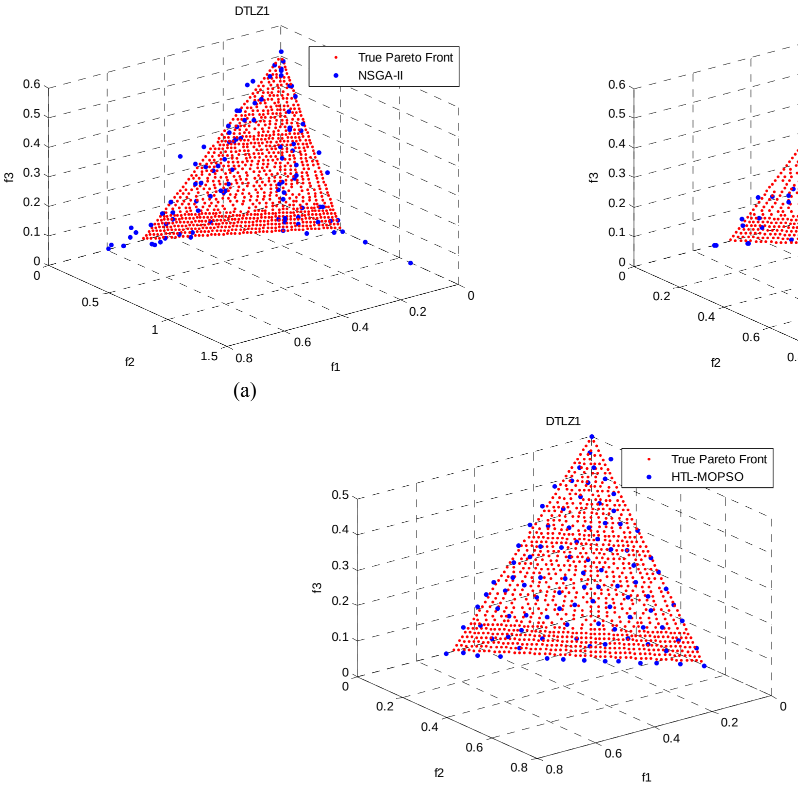


Fig.3. Pareto front produced by(a) NSGA-II,(b)TV-MOPSO,(c)MOEA/D,(d)MOTLBO,(e,f)HTL-MOPSO on ZDT6

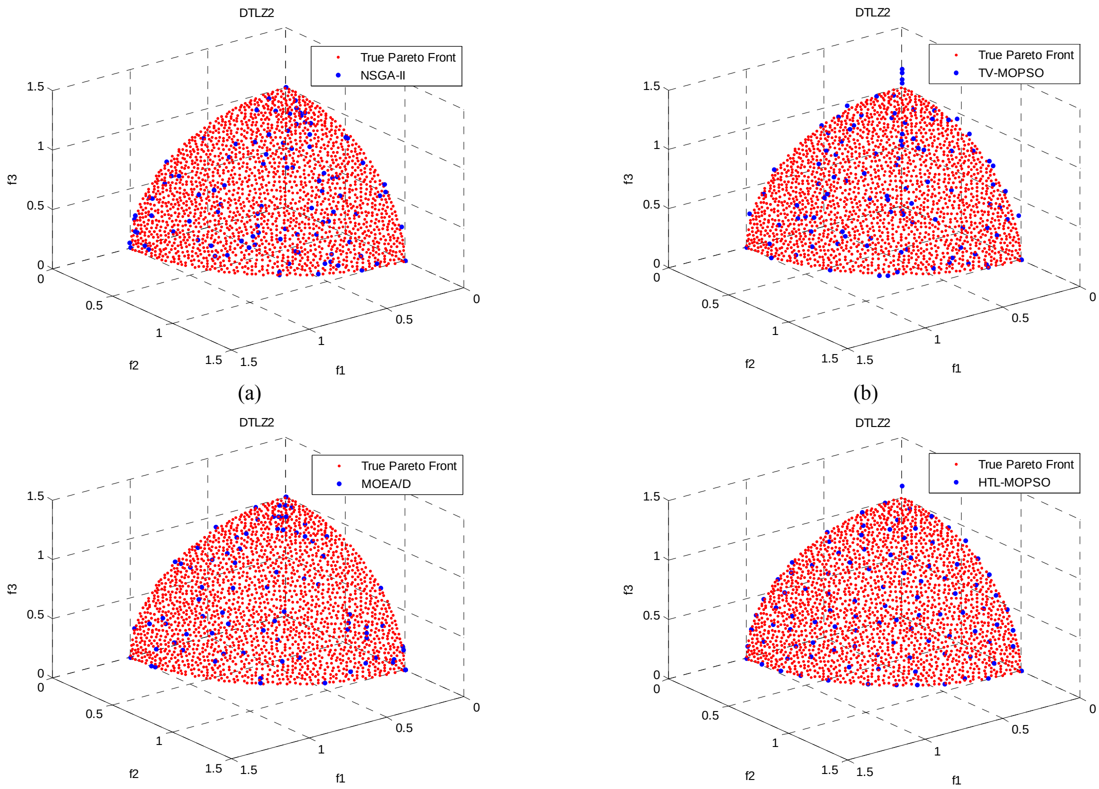
The benchmark problem DTLZ1 is a three-objective problem with a linear Pareto-optimal front. The search space contains ($11^5$-1) local Pareto hyper-planes which challenges the MOP algorithms' exploitation ability to deal with multi-modality. The Pareto front solutions produced by three competitive algorithms including NSGA-II, MOEA/D and HTL-MOPSO are illustrated in Fig.4. NSGA-II in Fig.4 (a) has failed in obtaining the non-dominated sets precisely with respect to IGD. Compared with NSGA-II, although MOEA/D has resulted in a better shape in Fig.4 (b), its IGD metrics in Table 2 are outperformed by HTL-MOPSO. HTL-MOPSO is the best among all mentioned algorithms in terms of the convergence and spread performance as demonstrated in Fig.4 (c). The p-values indicate that HTL-MOPSO is significantly better than other counterparts on DTLZ1.




(a) (b) (c)

Fig.4 Pareto front produced by(a) NSGA-II,(b)MOEA/D,(c)HTL-MOPSO on DTLZ1

The benchmark problem DTLZ2 with three objectives has a spherical Pareto-front. The Pareto fronts of DTLZ2 produced by NSGA-II, TV-MOPSO, MOEA/D and HTL-MOPSO are presented in Fig.5. With respect to spread performance in Fig.5, HTL-MOPSO is better than NSGA-II, TV-MOPSO, MOTLBO and MOEA/D. Also, the p-values comparison indicates HTL-MOPSO significantly outperforms other counterparts on DTLZ2.




(a) (b) (c) (d)

Fig.5 Pareto front produced by(a) NSGA-II,(b) TV-MOPSO,(c) MOEA/D,(d)HTL-MOPSO on DTLZ2

The benchmark problem DTLZ5 with three objectives tests the ability to converge into a degenerated curve. The Pareto front produced by NSGA-II, MOEA/D, MOTLBO and HTL-MOPSO are shown in Fig.6. From Table 2 and Fig.6 (c), it is observed that MOTLBO has the worst convergence especially at the bottom of the curve. HTL-MOPSO shows a more competitive convergence performance than other 4 algorithms. At the same time, HTL-MOPSO is able to provide the best spread solutions in obtaining Pareto front shown in Fig.6 (d). In terms of IGD metric, HTL-MOPSO is found to be the best followed by NSGA-II and TV-MOPSO. MOEA/D and MOTLBO are worse than other 3 algorithms with respect to IGD. The p-values again show that HTL-MOPSO is significantly better than other counterparts on DTLZ5.

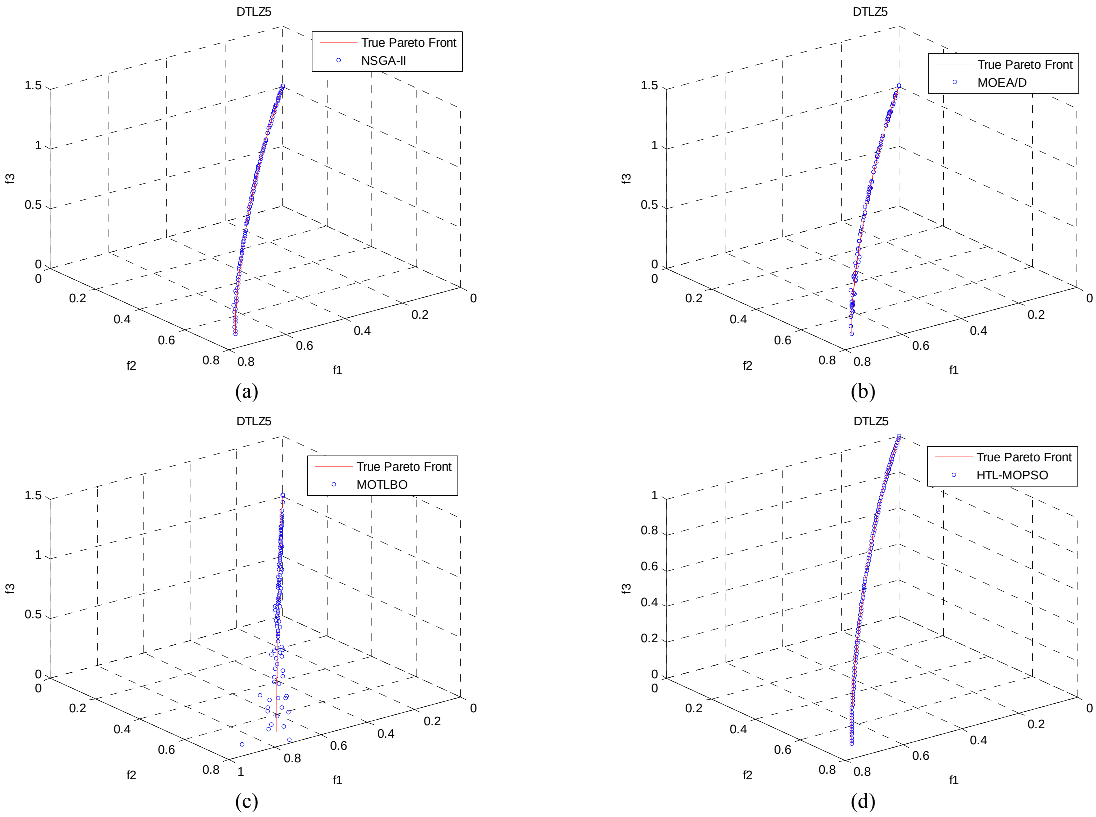


Fig.6 Pareto front produced by(a) NSGA-II,(b)MOEA/D,(c)MOTLBO,(d)HTL-MOPSO on DTLZ5

The test problem DTLZ7 has $2^{3-1}$ disconnected Pareto-optimal regions in the search space, which challenges the capacity of the algorithms to maintain subpopulation in different Pareto-optimal regions. The Pareto fronts produced by NSGA-II, TV-MOPSO, MOEA/D and HTL-MOPSO on DTLZ7 are presented in Fig.7. In Table 2, NSGA-II and MOEA/D provide better convergence performance than HTL-MOPSO. However, it is observed by Fig.7 (a) that NSGA-II fails to obtain all of the disconnected regions and MOEA/D in Fig.7 (c) suffers bad spread of obtained Pareto solutions among optimal Pareto regions. On the other hand, in Fig.7 (b) and Fig.7 (d), TV-MOPSO and HTL-MOPSO achieve better spread among true Pareto Front than NSGA-II and MODA/D. From Table 2, we can find that HTL-MOPSO and TV-MOPSO is the best and second-best performer in terms of IGD respectively. HTL-MOPSO is significantly better than other counterparts on DTLZ7 by comparing p-values in Table 2.

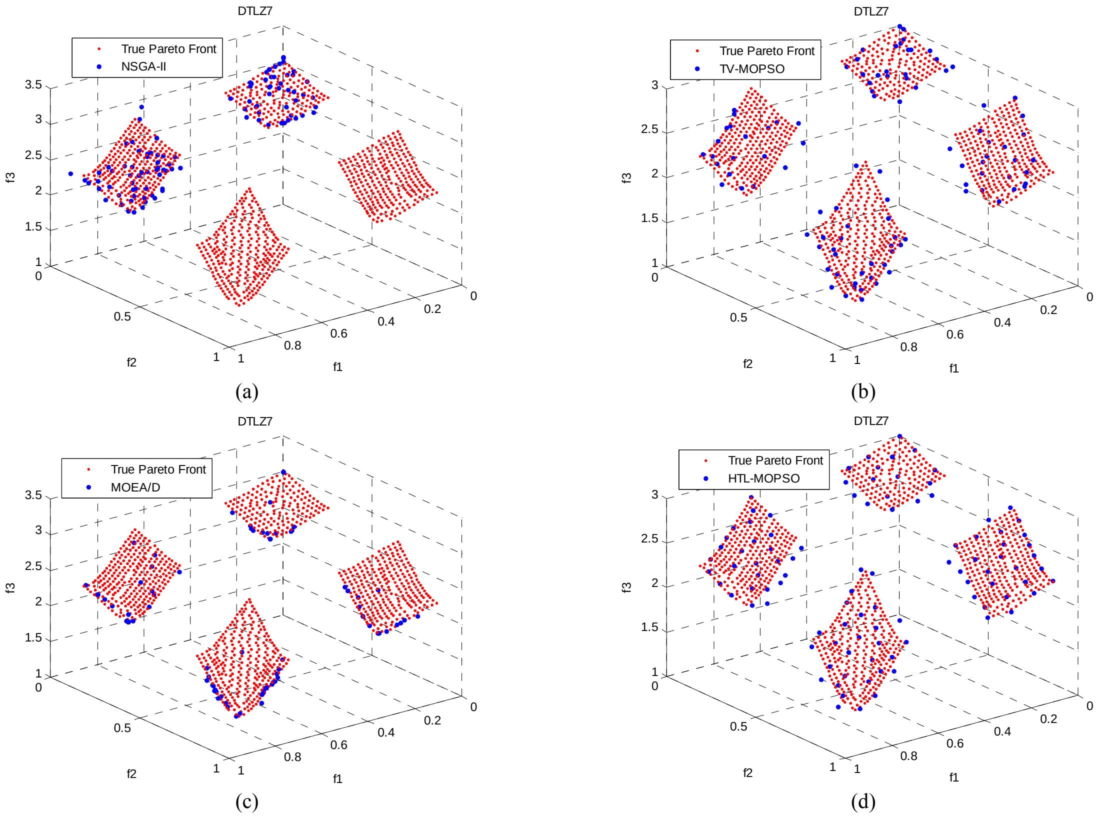



Fig.7 Pareto front produced by(a) NSGA-II,(b)TV-MOPSO,(b)MOEA/D,(d)HTL-MOPSO on DTLZ7

In conclusion, from the highlights in Table 2 and the discussion above, it is evident that HTL-MOPSO is a competitive algorithm on ZDT and DTLZ problems compared with some powerful algorithms. For most aforementioned benchmark functions, HTL-MOPSO allows the best IGD to be achieved in Table 2, being significantly better than the others, consistently for all test functions except for ZDT6. As for multimodal test functions, such as ZDT4 and DTLZ1, HTL-MOPSO are able to jump out the local optima and converge closely to the global Pareto front. Compared with TV-MOPSO, the search ability of HTL-PSO is better. The archive truncating strategy based on CCS allows better spread metric to be achieved.

### 4.4.2 Experiments results on UF problems

The previous section has demonstrated that HTL-MOPSO show a challenging performance on ZDT and DTLZ problems. In this part, the performance of HTL-MOPSO is further compared with other algorithms on the UF problems, which are the recently presented test problem with complicated Pareto set [34].

According to Table 2, on UF2 problem, MOEA/D is the best but not significantly different with NSGAII according to the p-values of Wilcoxon test. HTL-MOPSO is promising (the third best) and slightly worse than NSGA-II and MODA/D. The NSGA-II performs best with respect to UF3, followed by HTL-MOPSO. The results in Table 2 show that HTL-MOPSO performs best on UF7 followed by TV-MOPSO.

Fig.8 further compares HTL-MOPSO with best algorithm among NSGA-II, TV-MOPSO, MOEA/D and MOTLBO, by plotting the typical Pareto fronts on UF problems. In Fig.8 (a) (b), the Pareto solutions captured by HTL-MOPSO are not close to true Pareto front on the right side on UF2 and it is beaten by MOEA/D. On UF3 problem, Table 2 shows that NSGA-II has the smaller IGD value compared with HT-MOPSO. But HTL-MOPSO performs much better on the spread of Pareto solutions as illustrated in Fig.8(c), (d). For IGD metrics of UF7 problem in Table 2, HTL-MOPSO is best performer. According to Fig.8 (e), (f), the second best algorithm TV-MOPSO is observed to miss more parts of Pareto solutions and more far away from true PF. The results of UF problems test show that HTL-MOPSO is a promising algorithm and has reasonably performance on the UF benchmarks.

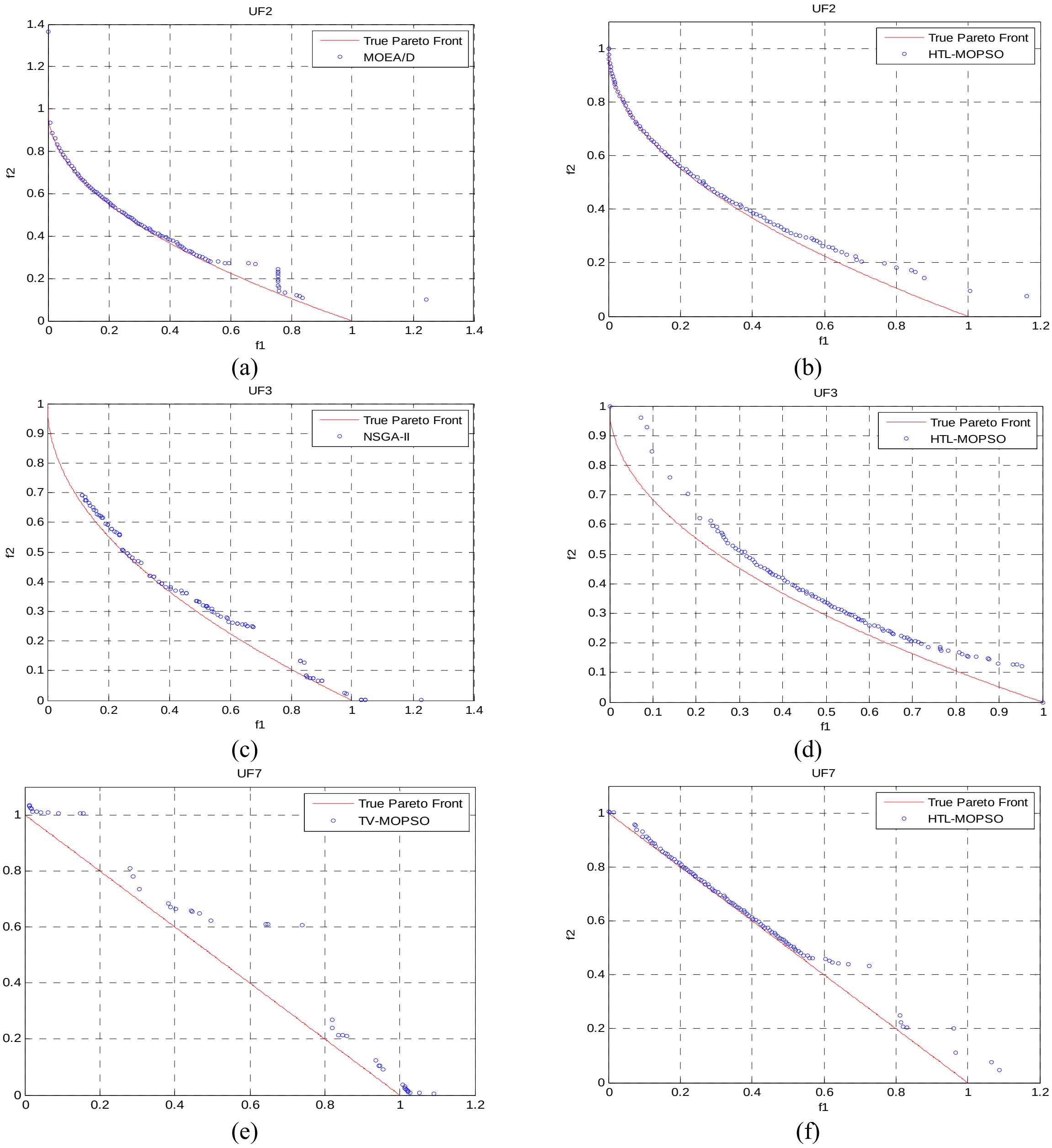



Fig.8 Pareto solutions in all 30 runs produced by(a) MOEA/D(UF2), (b)HTL-MOPSO(UF2), (c) NSGA-II (UF3), (d) HTL-MOPSO (UF3), ( e) TV-MOPSO(UF7), (f) HTL-MOPSO(UF7)

## 4.5 Comparison of HTL-PSO with PSO

The TLBO is incorporated within PSO (HTL-PSO) search process in order to enhance search ability. For unimodal problems, Fig. 9(a), (b) show the final nondominated solutions in all 30 runs by HTL-MOPSO and MOPSO (proposed HTL-MOPSO without TLBO) on ZDT2 and ZDT3. Note that it is easy to solve the ZDT2 and ZDT3 problems. But the MOPSO still misses the true Pareto front a few times while HTL-MOPSO always performs well.

For multimodal problems, HTL-PSO can help algorithm to avoid local optimum. In Fig.9(c), (d), a representative Pareto front of ZDT4 and DTLZ1 in a run by HTL-MOPSO and MOPSO are shown. The figures clearly show that MOPSO is easy to be trapped into local Pareto fronts and HTL-MOPSO has stronger global search ability to approximate the true Pareto fronts. The advantages of HTL-MOPSO are evident when solving MOPs with both unimodal and multimodal objective functions. The analysis reveals that search ability of HTL-PSO is better than single PSO without hybridization.

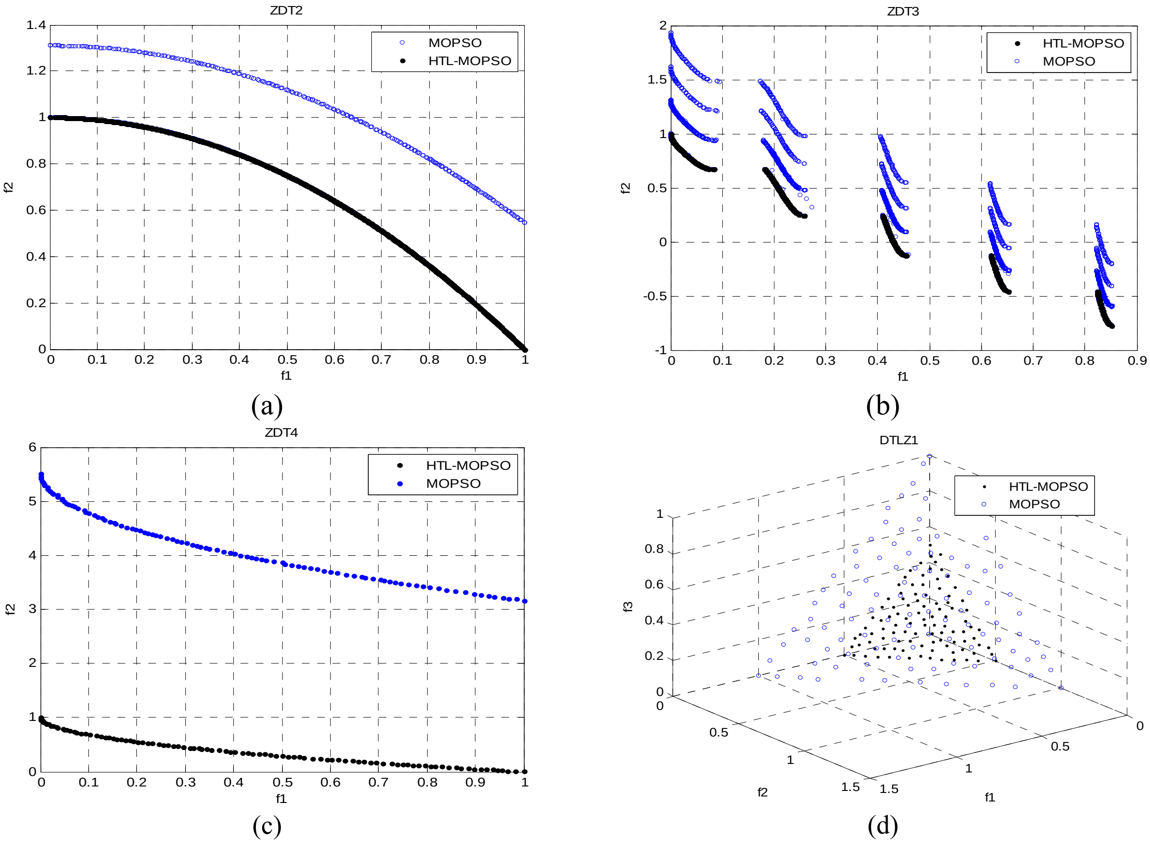


Fig. 9 Pareto fronts obtained by HTL-MOPSO and MOPSO on (a) ZDT2,(b) ZDT3,(c) ZDT4, (d) DTLZ1

### 4.6 Benefits of CCS

The HTL-MOPSO algorithm uses the circular crowded sorting (CCS) to truncate elitism archive instead of the popular CS in NSGA-II [6]. In this section, the benefits of CCS are investigated. The direct comparisons of CCS with CS are shown in Fig.10 and Fig.11. Here, Fig.10 (a) and Fig.11 (a) describe the original solutions of a two-objective problem and three-objective problem, respectively. The final solutions selected by CS and CCS are clearly shown in Fig.10 (b), (c) and Fig.11 (b), (c). Comparing Fig.10 (b) and Fig.10 (c) in same solutions scale, the final solutions by CCS have a better diversity and spread than that achieved by CS. Similarly, Fig.11 (c) has a much better spread than Fig.11 (b). According to the analysis, the archive truncating strategy based on CCS allows a better spread performance to be achieved.

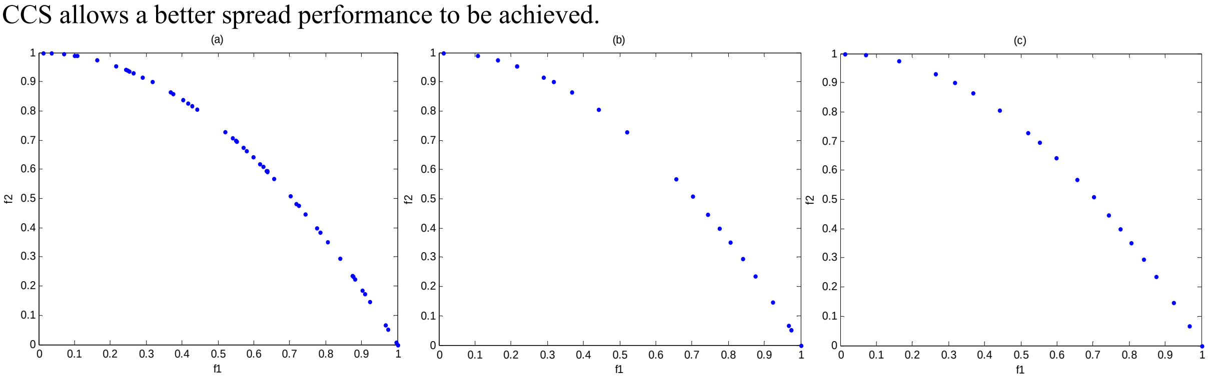


Fig.10 two-objective problem:(a) original solutions;(b)final solutions selected by CS;(c) final solutions selected by CCS

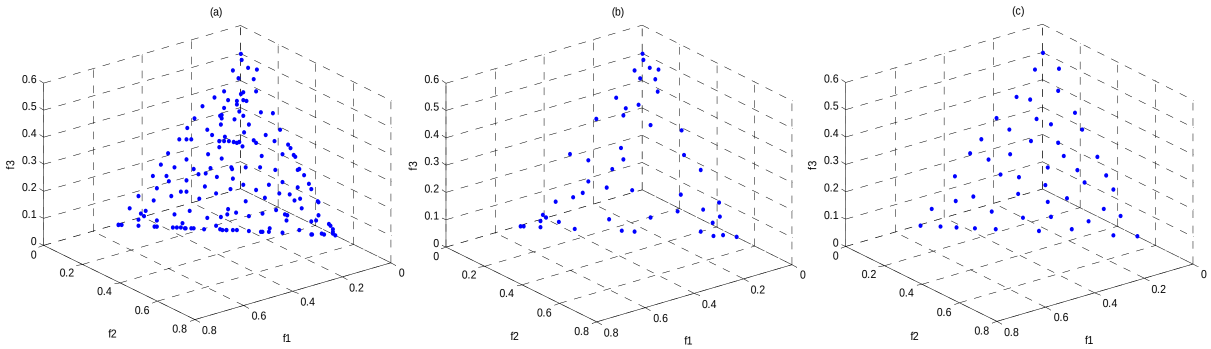


Fig.11 three-objective problem:(a) original solutions;(b)final solutions selected by CS;(c) final solutions selected by CCS

We further compare the IGD means and variance values on test problems obtained by HTL-MOPSO and HTL-MOPSO-CS (variant of HTL-MOPSO with CS instead of CCS) in Table 3. The comparisons Table 3 show that HTL-MOPSO outperforms HTL-MOPSO-CS on all 12 test problems. Specially, on multimodal ZDT4 and DTLZ1 problems, HTL-MOPSO is significantly better than HTL-MOPSO-CS with respect to mean and variance values. It is mean that CCS helps the algorithm jump out of local optimum and obtain a good spread of final PF. The results efficiently prove the advantage of the adoption of CCS technique.

Table 3 IGD means and variances of HTL-MOPSO and HTL-MOPSO-CS on test problems

| problems | HTL-MOPSO | | HTL-MOPSO-CS | |
|---|---|---|---|---|
| | Mean | Variance | Mean | Variance |
| ZDT1 | **3.88e-03** | **9.36e-10** | 4. 77e-03 | 8.77e-08 |
| ZDT2 | **3.85e-03** | **5.29e-10** | 4.94e-03 | 1.07e-07 |
| ZDT3 | **4.82e-03** | **5.90e-09** | 5.08e-03 | 2.94e-08 |
| ZDT4 | **3.99e-03** | **3.22e-09** | 2.17e-02 | 5.30e-03 |
| ZDT6 | **3.08e-03** | **3.19e-09** | 4.06e-03 | 1.18e-07 |
| DTLZ1 | **2.06e-02** | **7.15e-07** | 3.59e-02 | 2.47e-03 |
| DTLZ2 | **6.02e-02** | **2.69e-06** | 7.79e-02 | 2.71e-05 |
| DTLZ5 | **3.78e-03** | **1.04e-07** | 4.97e-03 | 1.43e-07 |
| DTLZ7 | **4.35e-02** | **8.89e-07** | 8.93e-02 | 1.22e-02 |
| UF2 | **4.71e-02** | 3.54e-05 | 4.86e-02 | **2.97e-05** |
| UF3 | **5.89ee-02** | 1.99e-04 | 5.98e-02 | **1.52e-04** |
| UF7 | **3.06e-02** | **2.20e-05** | 4.45e-02 | 2.66e-05 |

### 4.7 Impact of parameter setting

The parameters of inertia $w$, the accelerating coefficient $c_1$, $c_2$ referred to TV-MOPSO [15]. Here, interval generation *INV*, teaching factor $D_1$ and learning factors $D_2$ are investigated, where impact of different parameters values impact on the algorithm performance are investigated. The investigations are conducted on IGD mean of ZDT4 with multimodal objectives function and DTLZ2 with unimodal objectives functions in 30 runs.

Fig.12(a) and (b) shows the mean IGD values of ZDT4 and DTLZ2 when HTL-MOPSO uses different *INV* values. For ZDT4 problem with multimodal problem, when *INV* is smaller than 4, HTL-MOPSO fails to get Pareto solutions, which is not shown in Fig.12 (a). We can see in Fig. 12(a) that a relatively small *INV* values (between 4 and 10) are preferred. This may be due to the fact that a relatively small *INV* means that TLBO are invoked more frequently to enhance the search ability of PSO. For DTLZ2, it is evident that in Fig. 12(b) HTL-MOPSO is less sensitive to the values *INV* when it is greater than 4.

The impact of $D_1$ and $D_2$ on HTL-MOPSO's performance is also investigated on ZDT4 and DTLZ2 in Fig.12(c) and (d). For ZDT4 problem, the range of [0.75, 2] for $D_1$ and the range of [0.25, 1] for *D2* are preferred. For DTLZ2 problem, HTL-MOPSO is much less sensitive to $D_1$ and $D_2$ values, the results are not plotted in the Figures.

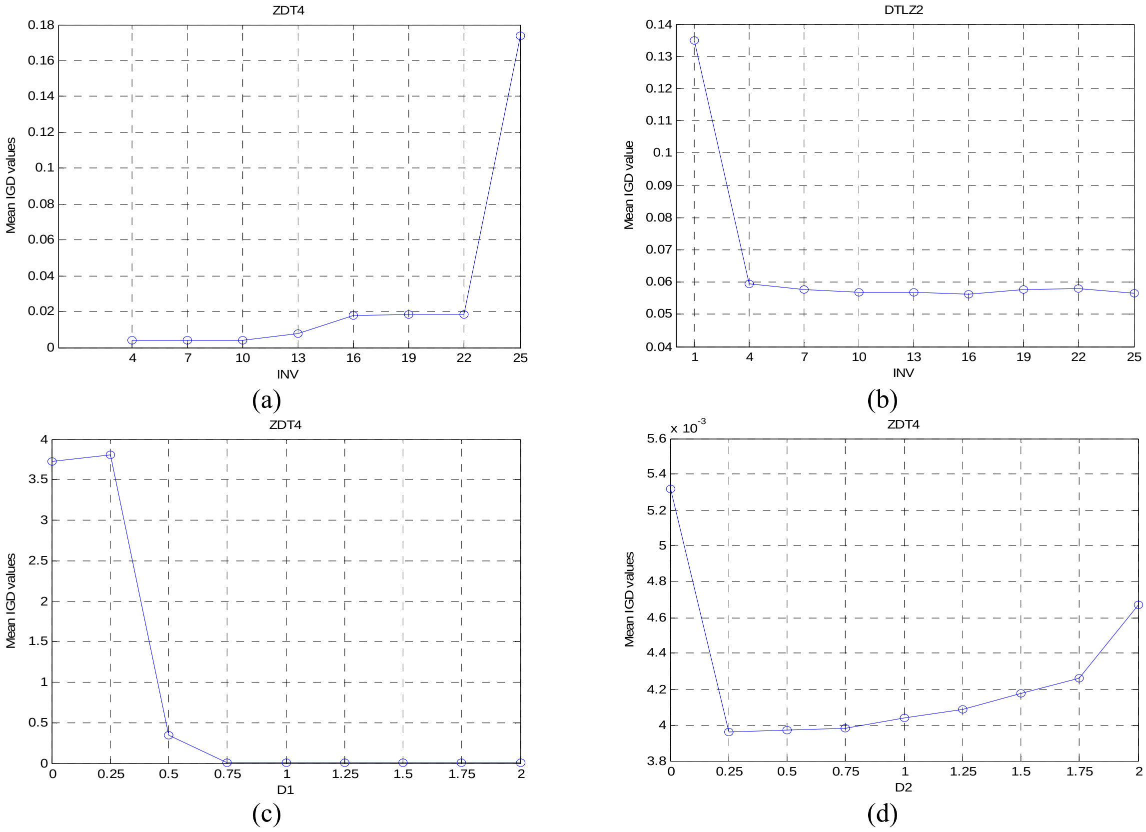



Fig.12. Mean IGD values of HTL-MOPSO ,(a) Different INV on DTLZ2,(b) Different INV on ZDT4,(c) Different D1 on ZDT4;(d) Different D2 on ZDT4

# 5 Conclusions

In this paper, a novel multi-objective PSO algorithm named HTL-MOPSO has been proposed. Several strategies including HTL-PSO, CCS and some other techniques are employed to enhance the performance of canonical MOPSO. HTL-MOPSO benefits from the following main aspects when solving MOPs and the contributions of this paper include: 1) As the particles in canonical MOPSO update their positions according to the personal best and the global best, insufficient shared information is transferred among different particles. In HTL-MOPSO, teacher phase and learner phase in TLBO are periodically invoked in PSO to pay more attention to whole information (mean positions) of all particles and encourage more interaction among different solutions, respectively. Hence, the search ability of HTL-MOPSO is improved by comparing with single PSO evolution and HTL-MOPSO has more capacity to jump out of local Pareto Front without deteriorating the main PSO convergence process. 2) As a section operator, circular crowded sorting is developed and performed on the elitism archive update process. The impact of each deleted solution on the crowded distance of remaining solutions is taken into consideration, which can maintain more diversity of candidates compared with existing crowding distance sorting. This is helpful for MOPs to maintain the diversity and the good distribution of non-dominated solutions along Pareto fronts. 3) The proposed HTL-MOPSO is complemented in this work to solve MOPs and the separate contributions of main strategies incorporated in HTL-MOPSO are confirmed.

The performance of HTL-MOPSO are tested on different MOPs benchmark functions includes ZDT, DTLZ and UF sets. Some state-of-the art algorithms, such as NSGA-II, MOEA/D, TV-MOPSO, MOTLBO, are used for comparisons. The experiments results show that HTL-MOPSO generally significantly outperforms all other algorithms on ZDT and DTLZ problems with respect to IGD metrics. For complicate UF problems, even though HTL-MOPSO does not always perform the best, it is shown promising performance compared with other time-consuming evolution algorithms (NSGA-II and MODA/D). Thus, experiment results support the good performance of HTL-MOPSO. Furthermore, the contributions of HTL-PSO evolution process and CCS section operator are investigated respectively. The analysis confirms that the benefit of HTL-PSO in bringing diversity to avoid local optimum and the benefits of CCS in maintaining diversity and good spread of solutions.

Although HTL-MOPSO with CCS has a better performance than that with CS, CCS is more time consuming. This is due the fact that the candidates are selected step by step according to the density. Meanwhile, the advantages of HTL-MOPSO on UF problems are not evident. Future work may focus on how to improve search ability of HTL-MOPSO on complicated Pareto front problems. The proposed efficient method is promising to be adopted in solving many objectives problems as well as more real world MOP problems.

## Acknowledge

This research is supported by the National Natural Science Foundation of China (No. 51177177, 51607177). Ms. Tingli Cheng, Mr. Zhile Yang and Mr. Shaojun Gan would like to thank the China Scholarship council for financially support their study in UK.

## References

[1] D. A. Van Veldhuizen, G. B. Lamont, Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. Evol. Comput. 8(2) (2000) 125-147.

[2] C. K. Chow and S. Y. Yuen, “A multiobjective evolutionary algorithm that diversifies population by its density,” IEEE Trans. Evol. Comput., vol. 16, no. 2, pp. 149-172, Apr. 2012.

[3] O. Abu Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. Information Sciences, 2014, 279:396-415.

[4] O. Abu Arqub, M. Al-Smadi, S. Momani, T. Hayat, Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method. Soft Computing, 2016, 20(8):3283-3302.

[5] M. Li, L. Liu, D. Lin, A fast steady-state ε-dominance multi-objective evolutionary algorithm. Comput. Optim. and Appl. 48(1) (2011) 109-138.

[6] K. Deb, A. Pratap, S. Agarwal, A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2) (2002) 182-197.

[7] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.

[8] S. Lin, F. Lin, H. Chen. A MOEA/D-based Multi-objective Optimization Algorithm for Remote Medical. Neurocomputing, 2016.

[9] J. Kenndy, R. C. Eberhart, Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. 4(1995) 1942-1948.

[10] J. Yu, S. Wang, L. Xi. Evolving artificial neural networks using an improved PSO and DPSO. Neurocomputing, 2008, 71(4): 1054-1060.

[11] R. Kundu, S. Das,R. Mukherjee. An improved particle swarm optimizer with difference mean based perturbation. Neurocomputing, 2014, 129(129):315-333.

[12] C. A. C. Coello, G. T. Pulido, M. S. Lechuga, Handling multiple objectives with particle swarm optimization. IEEE Trans. Evolut. Comput. 8(3) (2004) 256-279 .

[13] Y. Zhang, D. Gong, J. Zhang. Robot path planning in uncertain environment using multi-objective particle swarm optimization. Neurocomputing, 2013, 103: 172-185.

[14] S. Cheng, M.Y. Chen, P. J. Fleming. Improved multi-objective particle swarm optimization with preference strategy for optimal DG integration into the distribution system. Neurocomputing, 2015, 148: 23-29.

[15] P. K. Tripathi, S. Bandyopadhyay, S. K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Inf. Sci. 177(22) (2007) 5033-5049.

[16] M.Reyes-Sierra, C. A. C. Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art[J]. International Journal of Computational Intelligence Research, 2006, 2(3):287-308.

[17] Y.Sun, L. Zhang, X. Gu . A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. Neurocomputing, 2012, 98: 76-89.

[18] Y. Coorenm, M. Clerc, P. Siarry, MO-TRIBES: an adaptive multiobjective particle swarm optimization algorithm. Comput. Optim. Appl, 49(2) (2011) 379-400

[19] W. F. Leong and G. G. Yen, “PSO-based multiobjective optimizationwith dynamic population size and adaptive local archives,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 38, no. 5, pp. 1270–1293,Oct. 2008.

[20] G. G. Yen, W. F. Leong. Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization. IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans, 2009, 39(4):890-911.

[21] Z. H. Zhan, J. Li, J. Cao. Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems. IEEE Transactions on Cybernetics, 2013, 43(2): 445-463.
[22] Y. J. Wang, Y. P. Yang, "Particle swarm optimization with preference order ranking for multi-objective optimization," Information Sciences,149(12), (2009):1944-1959
[23] W. Hu., G. G.Yen. Adaptive Multiobjective Particle Swarm Optimization Based on Parallel Cell Coordinate System. IEEE Transactions on Evolutionary Computation, 2015, 19(1):1-18.
[24] M. Reyes Sierra and C. A. Coello Coello, "Improving PSO-based multiobjective optimization using crowding, mutation and ε-Dominance," in Proc. Evol. Multi-Criterion Optimization, 2005, pp. 505-519.
[25] J. D. Knowles and D.W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," Evol. Comput., vol. 8, pp. 149-172, Jun. 2000.
[26] G. Xu, Y. Yang, B. B. Liu, An efficient hybrid multi-objective particle swarm optimization with a multi-objective dichotomy line search. J. Comput. Appl. Math. 280 (2015) 310-326.
[27] J. Luo, Y. Qi, J. Xie, A hybrid multi-objective PSO–EDA algorithm for reservoir flood control operation. Appl. Soft Comput. 34(2015)526-538.
[28] K. Izui, S. Nishiwaki, M. Yoshimura, Enhanced multiobjective particle swarm optimization in combination with adaptive weighted gradient-based searching, Eng. Optim. 40 (9) (2008) 789–804.
[29] D. S. Liu, K.C. Tan, C.K. Goh, W.K. Ho, A multiobjective memetic algorithm based on particle swarm optimization, IEEE Trans. Syst. Man Cybern. B 37 (1) (2007) 42–50
[30] S. Cheng, H. Zhan, Z. Shu. An innovative hybrid multi-objective particle swarm optimization with or without constraints handling. Applied Soft Computing, 2016, 47:370–388.
[31] F. Zou, L. Wang, X. Hei, D. Chen, B. Wang, Multi-objective optimization using teaching-learning-based optimization algorithm. Eng. Appl. Artif. Intel. 26(4) (2013) 1291-1300 .
[32] F. Zou, L. Wang, X. Hei, Teaching–learning-based optimization with learning experience of other learners and its application. Appl. Soft Comput. 37 (2015) 725-736.
[33] M. Mahfouf, M. Y. Chen, D. A. Linkens, Adaptive weighted particle swarm optimisation for multi-objective optimal design of alloy steels[C]//Parallel problem solving from nature-ppsn viii. Springer Berlin Heidelberg, 2004: 762-771.
[34] Q. Zhang, A. Zhou, Zhao S, Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. University of Essex, 2008.
[35] K. Deb, L. Thiele, M. Laumanns, Scalable test problems for evolutionary multiobjective optimization. Springer London, 2005.
[36] http://www.iitk.ac.in/kangal/codes.shtml
[37] R. S. Barr, B. L. Golden, J. P. Kelly, M. G. C. Resende and W. R. Stewart (1995) "Designing and Reporting on Computational Experiments with Heuristic Methods," Journal of Heuristics, vol. 1, no. 1, pp. 9-32
[38] J. Derrac, S. García, D. Molina, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm & Evolutionary Computation, 2011, 1(1):3-18.

**Tingli Cheng** received the M.Sc. degree in school of Automation from Chongqing University in 2013. Now she is a Ph.D. candidate in school of Electrical Engineering, Chongqing University. Her research interests include multi-objective particle swarm optimization and optimal energy management of power system.

**Minyou Chen** received the M.Sc. degree in control theory and engineering from Chongqing University, China, in 1987, and the Ph.D. degree in control engineering from the University of Sheffield, Sheffield, U.K. in 1998. He is currently a Full Professor with Chongqing University. His current research interests include intelligent

modeling and control, multi-objective optimization, microgrid control, and state monitoring in power distribution systems. He has authored or co-authored over 180 papers.

**Peter J. Fleming** received the B.S. and Ph.D. degrees from the Queen's University, Belfast, U.K. Now he is a professor in the University of Sheffield. His control and systems engineering research interests include multi-criteria decision making, optimization, grid computing, and industrial applications of modeling, monitoring, and control. His current research interests include development of close links with a variety of industries in sectors such as aerospace, power generation, food processing, pharmaceuticals, and manufacturing. Prof. Fleming is a fellow of the Royal Academy of Engineering, the Institution of Electrical Engineers, and the Institute of Measurement and Control. He is an advisor with the International Federation of Automatic Control. He is the Editor-in-Chief of the International Journal of Systems.