# Multi-population Evolution based Dynamic Constrained Multiobjective Optimization under Diverse Changing Environments —Supplementary Material

Qingda Chen, *Member, IEEE*, Jinliang Ding, *Senior Member, IEEE*, Gary G. Yen, *Fellow, IEEE*, Shengxiang Yang, *Senior Member, IEEE*, and Tianyou Chai, *Life Fellow, IEEE*

This is the supplementary material to the paper entitled "Multi-population Evolution based Dynamic Constrained Multiobjective Optimization under Diverse Changing Environments", submitted to **IEEE Transactions on Evolutionary Computation.**

## S-I. A DCMOP Including Distance and Position Variables

The DCMOP including distance and position variables is described as follows:

$$\begin{cases} \min\ f_1(\mathbf{x},t) = 1 + x_1 + (x_2 - 0.01t)^2 \\ \min\ f_2(\mathbf{x},t) = t - x_1 + (x_2 - 0.03t)^2 \end{cases}$$
$$subject\ to: \begin{cases} f_1 + f_2 \le 3 \\ x_i \in [0,1],\ i = 1,\ 2. \end{cases} \tag{S-1}$$

For any fixed $x_2$, changing the value of $x_1$ in $\mathbf{x}$ only generates a set of incomparable or equivalent solutions. Therefore, $x_1$ is considered to be a position variable. However, for any fixed $x_1$, only changing the value of $x_2$ in $\mathbf{x}$ will never result in incomparable solutions. Therefore, $x_2$ is called a distance variable.

## S-II. Penalty Function Method

The penalty function method is shown in Equation (S-2). Each modified objective value (denoted as $f'_m(\mathbf{x}_i,t)$) consists of the "distance" and "penalty" values of $\mathbf{x}_i$ in the $m$th objective function (represented as $d_m(\mathbf{x}_i,t)$ and $p_m(\mathbf{x}_i,t)$).

$$f'_m(\mathbf{x},\ t) = d_m(\mathbf{x}_i,\ t) + p_m(\mathbf{x}_i,\ t) \tag{S-2}$$

The distance value of each dimension of the objective space is calculated according to the effect of a solution's $CV$ value on its objective function, and it can be formulated as follows:

$$d_m(\mathbf{x}_i,\ t) = \sqrt{\overline{f_m(\mathbf{x},\ t)}^2 + \overline{v(\mathbf{x}_i,\ t)}^2} \tag{S-3}$$

$$\overline{f_m(\mathbf{x},t)} = \frac{f_m(\mathbf{x},t) - f_{m,t}^{\min}}{f_{m,t}^{\max} - f_{m,t}^{\min}} \tag{S-4}$$

$$\overline{v(\mathbf{x}_i,\ t)} = \frac{1}{H}\sum_{k=1}^{H}\frac{h_k(\mathbf{x}_i,\ t)}{h_{k,\ t}^{\max}} + \frac{1}{G-H}\sum_{k=H+1}^{G}\frac{g_k(\mathbf{x}_i,\ t)}{g_{k,\ t}^{\max}} \tag{S-5}$$

where $f_{m,t}^{max}$ and $f_{m,t}^{min}$ represent the maximum and minimum values of all solutions on the $m$th objective function in environment $t$, respectively. $\overline{f_m(\mathbf{x},\ t)}$ and $\overline{v(\mathbf{x}_i,\ t)}$ are the $m$th normalized objective function and $CV$ values of $\mathbf{x}_i$ at $t$, respectively, and they can be calculated by Equations (S-4) and (S-5), respectively.

The penalty value is added to the fitness of each infeasible solution to identify its performance, and the value of $\mathbf{x}_i$ in the $m$th objective function is calculated as follows:

$$p_m(\mathbf{x}_i,\ t) = (1 - r_f) \times \overline{v(\mathbf{x}_i,\ t)} + r_f \times Y_m(\mathbf{x}_i,\ t) \tag{S-6}$$

$$r_f = \frac{\text{number of feasible solutions in current population}}{\text{population size}} \tag{S-7}$$

$$Y_m(\mathbf{x}_i,\ t) = \begin{cases} 0, & \text{if } \overline{v(\mathbf{x}_i,\ t)} = 0 \\ \overline{f_m(\mathbf{x}_i,\ t)}, & \text{otherwise} \end{cases} \tag{S-8}$$

where $r_f$ denotes the feasible ratio of the current population,

The final modified objective function values of $\mathbf{x}_i$ are obtained, and each solution is assigned a fitness value according to Equation (S-2).

## S-III. Crowding Distance Calculation Operator

The crowding distance values of all solutions in a set $I$ are calculated as the sum of distance values corresponding to each objective. The pseudocode of this operator is presented in Algorithm S-1.

This work was supported in part by the National Natural Science Foundation of China under Grant 62203101, Grant 61988101, Grant 61991400, Grant 61991403, in part by the Fundamental Research Funds for the Central Universities under Grant N2224004-01, Grant N2224002-23, in part by the Central Government Guides Local Science and Technology Development Foundation under Grant 2022JH6/100100055, in part by the Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education under Grant 202109241, and in part by the 111 Project 2.0 under Grant B08015. (*Corresponding Authors: Jinliang Ding, Gary G. Yen*)

Q. Chen, J. Ding, and T. Chai are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China (e-mail: cqd0309@126.com; jlding@mail.neu.edu.cn, tychai@mail.neu.edu.cn). (*Corresponding authors*: *Jinliang Ding, Gary G. Yen*)

G. G. Yen is with the school of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74074, USA (gyen@okstate.edu).

S. Yang is with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China, and also with the Centre for Computational Intelligence. School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, U. K. (syang@dmu.ac.uk).

**Algorithm S-1**: Crowding Distance Calculation Operator

```
Input: A set I
Output: Distances of all solutions in a set I.
Sort using each objective value of solutions in the set I.
For i=1: |I|
   if i=1 or i=|I| then
      d[i]=∞;
   else
      For j=1: m;
         d[i]= d[i]+(f_j(x_{i+1},t) -f_j(x_{i-1},t))
      end for
   end if
end for
```

**Algorithm S-2**: Offspring Generation Operator

```
Input: Two parent solutions (i.e., p_1 and p_2), the crossover probability
   (Cr), the mutation probability (mp), D, L_j and U_j,
Output: Two children solutions (i.e., c_1 and c_2)
for j=1: D
   Generate a random number rd that belongs to [0,1].
   if rd<Cr then
      Generate the jth decision variable according to Equations (S-9) and
      (S-10).
      if c_{1,j}>U_j then
         c_{1,j}=U_j
      else if c_{1,j}<L_j then
         c_{1,j}=L_j
      end if
      if c_{2,j}>U_j then
         c_{2,j}=U_j
      else if c_{2,j}<L_j then
         c_{2,j}=L_j
      end if
   end if
end for
for i=1:2
   for j=1: D
      Generate a random number md that belongs to [0,1].
      if md<mp then
         Generate a Δ according to Equation (S-11), and let c_{i,j}= c_{i,j}+Δ.
         if c_{i,j}>U_j then
            c_{i,j}=U_j
         else if c_{i,j}<L_j then
            c_{i,j}=L_j
         end if
      end if
   end for
end for
```

## S-IV. Offspring Generation Operator

The simulated binary crossover (SBX) and polynomial mutation (PM) operators are to generate the children solutions (i.e., $c_1$ and $c_2$) from two parent solutions (i.e., $p_1$ and $p_2$). The SBX operator is described as follows.

$$\begin{cases} c_{1,j} = 0.5\times\left[(1+\beta)\,p_{1,j} + (1-\beta)\,p_{2,j}\right] \\ c_{2,j} = 0.5\times\left[(1-\beta)\,p_{1,j} + (1+\beta)\,p_{2,j}\right] \end{cases} \tag{S-9}$$

TABLE S-I
The Approximate Feasibility Retios of Eight Test Problems at Environment $T$

| Environment index \ Instance No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.044% | 0.289% | 0.649% | 0.786% | 0.042% | 0.279% | 0.616% | 0.769% |
| 1 | 0.081% | 0.494% | 0.722% | 1.061% | 0.082% | 0.481% | 0.705% | 1.025% |
| 2 | 0.139% | 0.742% | 0.773% | 1.381% | 0.131% | 0.737% | 0.747% | 1.372% |
| 3 | 0.238% | 1.112% | 0.783% | 1.722% | 0.230% | 1.091% | 0.768% | 1.655% |
| 4 | 0.112% | 0.525% | 0.222% | 0.675% | 0.113% | 0.522% | 0.212% | 0.669% |
| 5 | 0.409% | 1.415% | 0.425% | 1.419% | 0.411% | 1.367% | 0.413% | 1.419% |
| 6 | 0.898% | 2.535% | 0.470% | 2.088% | 0.874% | 2.521% | 0.463% | 2.025% |
| 7 | 1.628% | 3.838% | 0.400% | 2.415% | 1.591% | 3.755% | 0.398% | 2.343% |
| 8 | 2.388% | 4.756% | 0.295% | 2.284% | 2.418% | 4.688% | 0.285% | 2.276% |
| 9 | 1.220% | 1.918% | 0.083% | 0.735% | 1.170% | 1.854% | 0.077% | 0.702% |
| 10 | 2.301% | 2.901% | 0.081% | 0.915% | 2.209% | 2.804% | 0.081% | 0.889% |
| 11 | 2.463% | 3.834% | 0.168% | 1.487% | 2.395% | 3.757% | 0.158% | 1.452% |
| 12 | 2.543% | 5.105% | 0.297% | 2.412% | 2.500% | 4.995% | 0.284% | 2.380% |
| 13 | 1.837% | 4.770% | 0.438% | 2.855% | 1.770% | 4.559% | 0.407% | 2.761% |
| 14 | 0.203% | 0.518% | 0.119% | 0.432% | 0.200% | 0.513% | 0.114% | 0.413% |
| 15 | 0.226% | 0.661% | 0.222% | 0.669% | 0.221% | 0.674% | 0.229% | 0.663% |
| 16 | 0.279% | 0.969% | 0.489% | 1.172% | 0.285% | 0.937% | 0.470% | 1.142% |
| 17 | 0.263% | 1.258% | 0.890% | 1.910% | 0.282% | 1.240% | 0.898% | 1.847% |
| 18 | 0.225% | 1.420% | 1.506% | 2.768% | 0.231% | 1.421% | 1.477% | 2.751% |
| 19 | 0.007% | 0.053% | 0.086% | 0.138% | 0.008% | 0.052% | 0.094% | 0.138% |
| 20 | 0.006% | 0.042% | 0.104% | 0.130% | 0.005% | 0.040% | 0.094% | 0.123% |

$$\beta = \begin{cases} (rand\times 2)^{\frac{1}{1+dc}} & rand \le 0.5 \\ \left(\dfrac{1}{2-rand\times 2}\right)^{\frac{1}{1+dc}} & otherwise \end{cases} \tag{S-10}$$

where $rand$ is a random number belonging to [0,1]. $c_{1,j}$ and $p_{1,j}$ are the $j$th decision variable of $c_1$ and $p_1$, respectively. $\beta$ and $\eta$ denote the spread factor and the distribution index.

The PM operator is to obtain a mutation individual (i.e., $mc$) from a children solution (i.e., $c$), which is described as follows.

$$\Delta = \begin{cases} (2\times rand)^{\frac{1}{1+dm}} - 1 & rand < 0.5 \\ 1-\left[2(1-rand)\right]^{\frac{1}{1+dm}} & otherwise \end{cases} \tag{S-11}$$

$$mc = c + \Delta \tag{S-12}$$

where $dm$ is the distribution index, and $\Delta$ is a mutation value.

The pseudocode of the offspring generation operator for obtaining two children solutions is presented in Algorithm S-2.

## S-V. Feasibility Ratios and True DPOFs and DPOSs of Test Problems

It is difficult to know the true feasibility ratios of the test problems due to their complex characteristics. In this paper, the feasibility ratio is defined as the percentage of feasible solutions out of 1,000,000 randomly sampled individuals. The approximated feasibility ratios of eight test problems at environment $t$ are given in Table S-I.

Table S-I shows that the maximum and minimum feasibility ratios of eight test problems among all environments are 5.105% and 0.005%, respectively. Each test problem includes both small and slightly large feasible ratios. Therefore, the designed test problems can measure the constraint handling capability of a dynamic constrained algorithm.

The true DPOFs and DPOSs of each test problem in environment $t$ are given in Table S-II.

TABLE S-II
THE TRUE DPOFS AND DPOSS OF EACH TEST PROBLEM

| Instance No. | Description | Remarks |
|---|---|---|
| 1 | $s(t)=\max(3.5-0.14\times t, 0.7+0.14\times t)$; $m(t)=\max(1.43-0.05\times t, 0.43+0.05\times t)$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(2\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(2\pi x_1))$<br>DPOS($t$): $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(2\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs have one mode, and they first move downward with time $t$ and then upward. The true DPOFs change from continuous to disconnected and finally back to continuous. |
| 2 | $s(t)=\max(2.5-0.05\times t, 1.5+0.05\times t)$; $m(t)=\max(1.16-0.075\times t, -0.34+0.075\times t)$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(2\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(2\pi x_1))$<br>DPOS($t$) $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(2\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs have one mode, and they first move downward with time $t$ and then upward. The true DPOFs change from disconnected to continuous and finally back to disconnected. |
| 3 | $s(t)=\min(2.1+0.14\times t, 4.9-0.14\times t)$; $m(t)=\min(0.93+0.05\times t, 1.93-0.05\times t)$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(2\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(2\pi x_1))$<br>DPOS($t$) $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(2\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs have one mode, and they first move upward with time $t$ and then downward. The true DPOFs change from disconnected to continuous and finally back to disconnected. |
| 4 | $s(t)=\min(2+0.05\times t, 3-0.05\times t)$; $m(t)=\min(0.41+0.075\times t, 1.91-0.075\times t)$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(2\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(2\pi x_1))$<br>DPOS($t$) $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(2\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs have one mode, and they first move upward with time $t$ and then downward. The true DPOFs change from continuous to disconnected and finally back to continuous. |
| 5 | $s(t)=\max(3.5-0.14\times t, 0.7+0.14\times t)$; $m(t)=\max(1.43-0.05\times t, 0.43+0.05\times t)$; $W_t=6\times\sin(0.2\times\pi(t+1))$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(W_t\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(W_t\pi x_1))$<br>DPOS($t$) $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(W_t\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs oscillate among ten optimization modes, and they first move downward with time $t$ and then upward. The true DPOFs change from continuous to disconnected and finally back to continuous. |
| 6 | $s(t)=\max(2.5-0.05\times t, 1.5+0.05\times t)$; $m(t)=\max(1.16-0.075\times t, -0.34+0.075\times t)$; $W_t=6\times\sin(0.2\times\pi(t+1))$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(W_t\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(W_t\pi x_1))$<br>DPOS($t$) $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(W_t\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs oscillate among ten optimization modes, and they first move downward with time $t$ and then upward. The true DPOFs change from disconnected to continuous and finally back to disconnected. |
| 7 | $s(t)=\min(2.1+0.14\times t, 4.9-0.14\times t)$; $m(t)=\min(0.93+0.05\times t, 1.93-0.05\times t)$; $W_t=6\times\sin(0.2\times\pi(t+1))$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(W_t\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(W_t\pi x_1))$<br>DPOS($t$) $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(W_t\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs oscillate among ten optimization modes, and they first move upward with time $t$ and then downward. The true DPOFs change from disconnected to continuous and finally back to disconnected. |
| 8 | $s(t)=\min(2+0.05\times t, 3-0.05\times t)$; $m(t)=\min(0.41+0.075\times t, 1.91-0.075\times t)$ $W_t=6\times\sin(0.2\times\pi(t+1))$; $t\in[0,1,\dots,20]$;<br>$g(\mathbf{x}_{\mathrm{II}},t)=\sum_{j=2}^{4}(x_j-\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)})^2$; $g(\mathbf{x}_{\mathrm{III}},t)=\sum_{j=5}^{10}(x_j-(0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor))^2$<br>$f_1(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(x_1+0.05\times\sin(W_t\pi x_1))$; $f_2(\mathbf{x},t)=(1+g(\mathbf{x}_{\mathrm{II}},t))(1+g(\mathbf{x}_{\mathrm{III}},t))(s(t)-x_1+0.05\times\sin(W_t\pi x_1))$<br>DPOS($t$) $0\le x_1\le1$; $x_j=\frac{2}{e^{2.1}}\times e^{t\%5\times0.5+0.1\sin(j\times t\times0.52136\pi)}, j\in[2,4]$; $x_j=0.05\sin(j\times(t+1)\times0.52136\pi)+0.38\lfloor\frac{t+6}{5}\rfloor, j\in[5,10]$<br>DPOF($t$) without constraints: $f_1+f_2=s(t)+0.1\sin(W_t\pi\frac{f_1-f_2+s(t)}{2})$<br>*s.j.* $\cos\frac{-\pi}{4}\times(f_2-1)-sin\frac{-\pi}{4}\times f_1-m(t)\ge0.2\times\lvert\sin(4\pi\times(sin\frac{-\pi}{16}\times(f_2-1)+cos\frac{-\pi}{16}\times f_1))\rvert^{0.5}$;<br>$f_1+f_2-6<0$; $m(t)<6$ | The unconstrained DPOFs oscillate among ten optimization modes, and they first move upward with time $t$ and then downward. The true DPOFs change from continuous to disconnected and finally back to continuous. |

TABLE S-III
COMBINATIONS OF KEY PARAMETER VALUES

| Parameters | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 |
|---|---|---|---|---|---|
| *Cr* | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
| *dc* | 1 | 3 | 5 | 7 | 9 |
| *mp* | 0.01 | 0.05 | 0.1 | 0.15 | 0.2 |
| *dm* | 10 | 20 | 30 | 40 | 50 |

## S-VI. ALGORITHM PARAMETER SELECTION

The proposed *m*EDCMOA adopts the popular simulated binary crossover (SBX) and polynomial mutation (PM) operators to generate offspring. The main parameters in the SBX operator are the crossover probability (*Cr*) and the distribution index (*dc*), and those in the PM operator are the mutation probability (*mp*) and the distribution index (*dm*).

Considering that a set of solutions can satisfy the optima of a given DCMOP, this paper uses two performance metrics (i.e., MHV and MIGD) to evaluate the performance of an algorithm. To facilitate the analysis of the other four parameters (i.e., *Cr*, *dc*, *mp*, and *dm*) on the performance of *m*EDCMOA, we use the average regularization indicator (ARI) to represent these two metrics, which is calculated according to Equations (S-13) and (S-14).

$$RI_{cb} = \frac{MHV_{cb}}{MHV_{\min}} - \left(\frac{MIGD_{cb}}{MIGD_{\max}}\right) \tag{S-13}$$

$$ARI_c = \frac{1}{8}\sum_{b=1}^{8} RI_{cb} \tag{S-14}$$

where $c$ represents the index of the parameter combination, and $b$ is the index of the test problem. $RI_{cb}$ denotes the regularization value of $\mathrm{MHV}_{cb}$ and $\mathrm{MIGD}_{cb}$ obtained by *m*EDCMOA that uses the $c$th parameter combination to solve the $b$th test problem. $ARI_c$ is the average regularization indicator of *m*EDCMOA that uses the $c$th parameter combination to solve eight test problems. The larger the value of $ARI_c$, the better the performance of an algorithm.

Table S-III gives the five levels of *Cr*, *dc*, *mp*, and *dm* based on our preliminary experiments. For example, *Cr*, *dc*, *mp*, and *dm* are set to 0.7, 5, 0.1, and 30, respectively, when their levels are 3. An orthogonal array $L_{25}$ with four parameters and five levels was used to examine the performance of *m*EDCMOA with different parameter combinations. For each parameter combination, *m*EDCMOA was independently executed 30 times on each benchmark. Subsequently, the $ARI_c$ of *m*EDCMOA is calculated according to Equations (S-13) and (S-14). Last, this paper uses the Taguchi method proposed in [s1] to examine the impacts of *m*EDCMOA with different parameter combinations. The factor-level trends of $ARI_c$ of four parameters can be obtained, which are given in Fig. S-1.

Table S-III and Fig. S-1 show that *Cr* with a level of 4, *dc* with a level of 3, *mp* with a level of 2, and *dm* with a level of 4 can yield the best result. Thus, *Cr*, *dc*, *mp*, and *dm* were set to 0.8, 5, 0.05, and 40, respectively. According to our experimental results, there is no statistically significant difference when *m*EDCMOA uses some other parameter combinations (e.g., *Cr*=0.6, *dc*=7, *mp*=0.04, and *dm*=40).

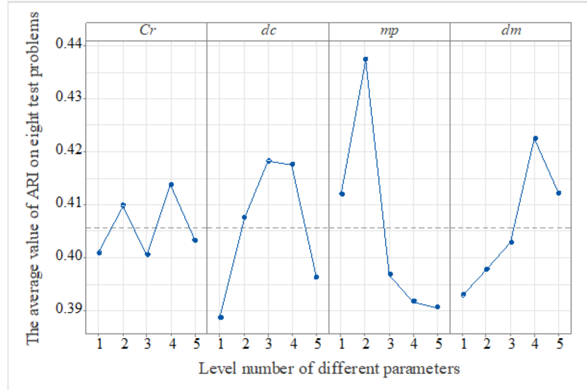


Fig. S-1. Factor-level trends of four parameters

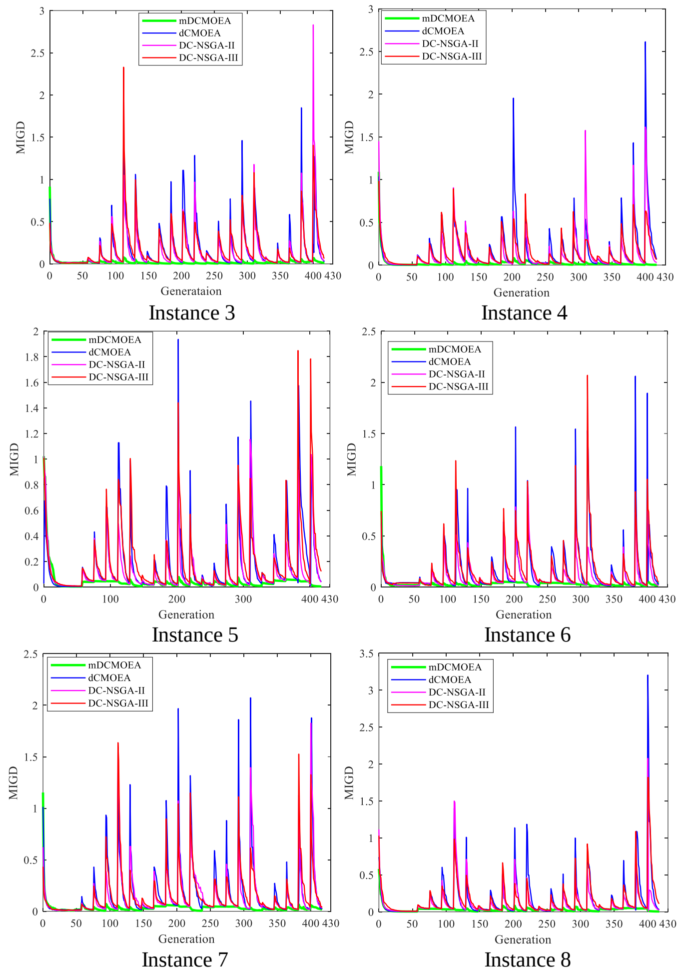


Fig. S-2. Evolution curves of average MIGD values for the third to eighth test problems with $\tau_t$=18 and $n_t$=21.

## S-VII. EVOLUTION CURVES OF AVERAGE MIGD VALUES

Fig. S-2 shows the evolution curves of average MIGD values for the third to eighth test problems with $\tau_t$=18 and $n_t$=21.

## S-VIII. DPOFS OBTAINED BY *M*EDCMOA AND COMPARED ALGORITHMS

The DPOFs obtained by *m*EDCMOA and compared algorithms on the second to eighth test problems with $\tau_t$=18 and $n_t$=21 are given in Figures S-3–S-9.

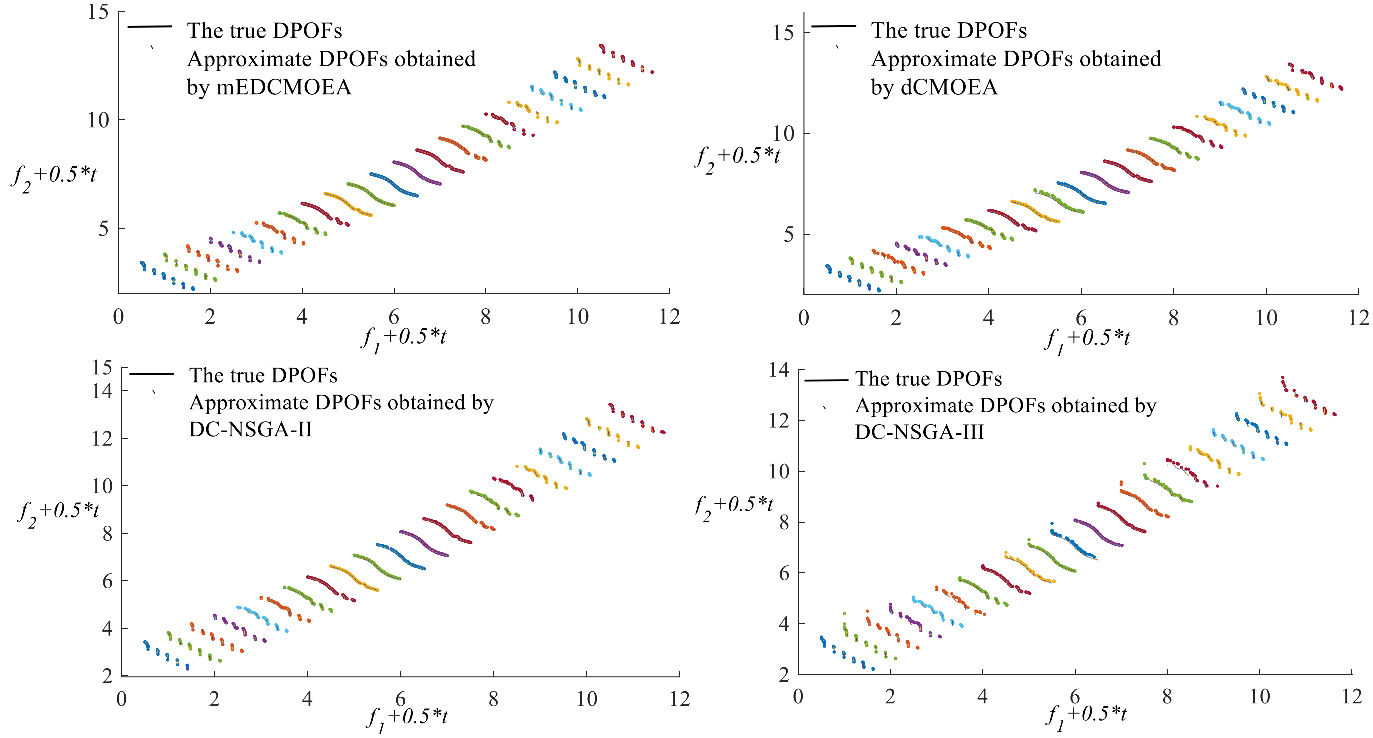


Fig. S-3. Obtained DPOFs obtained by the *m*EDCMOA (top left), *d*CMOEA (top right), DC-NSGA-II (bottom left), and DC-NSGA-III (bottom right) on the second test problem with $\tau_t$=18 and $n_t$=21.

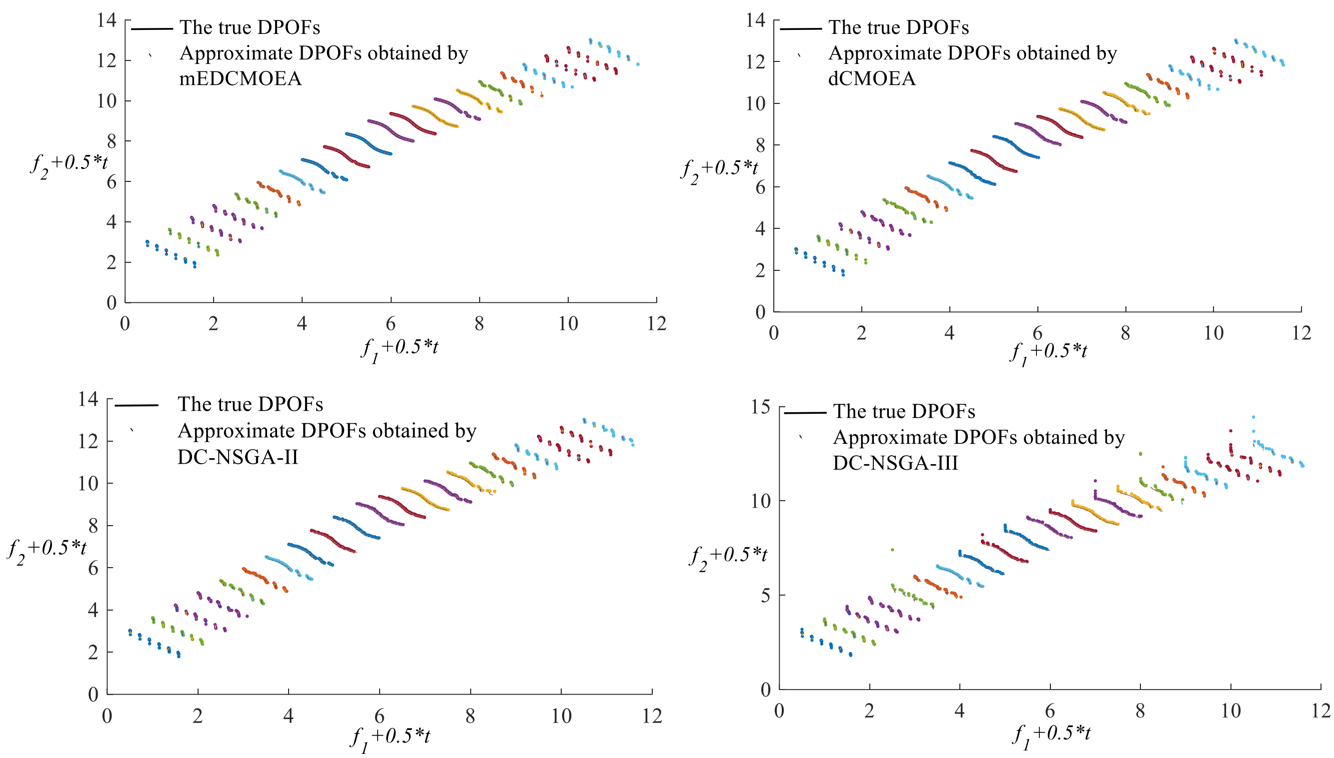


Fig. S-4. Obtained DPOFs obtained by the *m*EDCMOA (top left), *d*CMOEA (top right), DC-NSGA-II (bottom left), and DC-NSGA-III (bottom right) on the third test problem with $\tau_t$=18 and $n_t$=21.

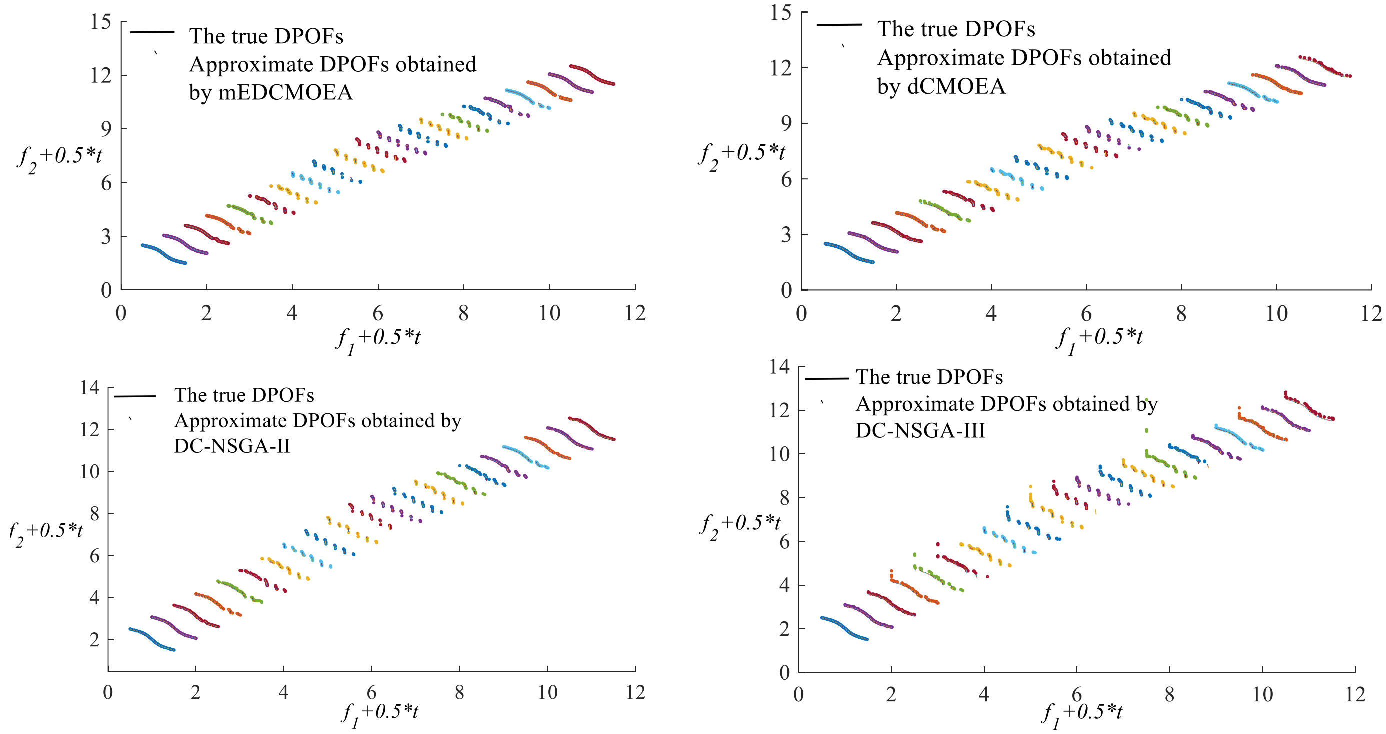


Fig. S-5. Obtained DPOFs obtained by the *m*EDCMOA (top left), *d*CMOEA (top right), DC-NSGA-II (bottom left), and DC-NSGA-III (bottom right) on the fourth test problem with $\tau_t$=18 and $n_t$=21.

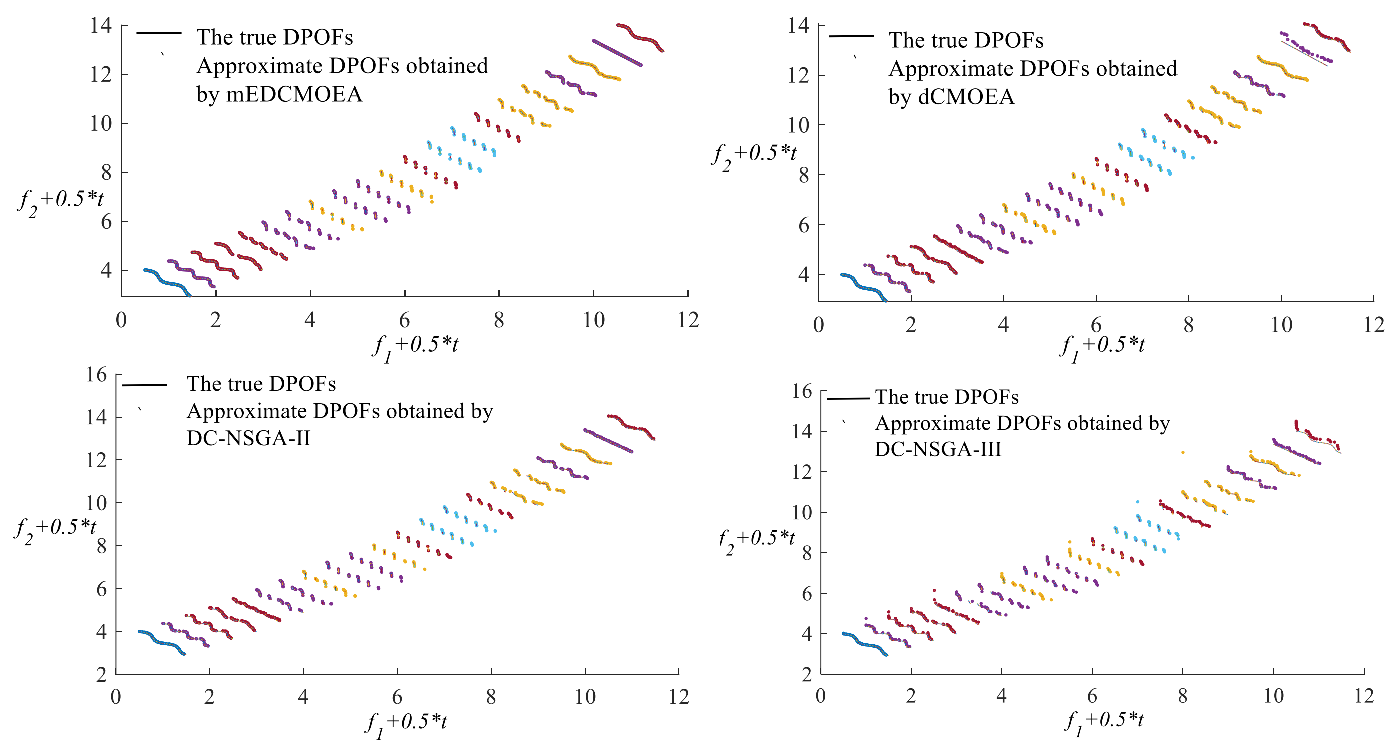


Fig. S-6. Obtained DPOFs obtained by the *m*EDCMOA (top left), *d*CMOEA (top right), DC-NSGA-II (bottom left), and DC-NSGA-III (bottom right) on the fifth test problem with $\tau_t$=18 and $n_t$=21.

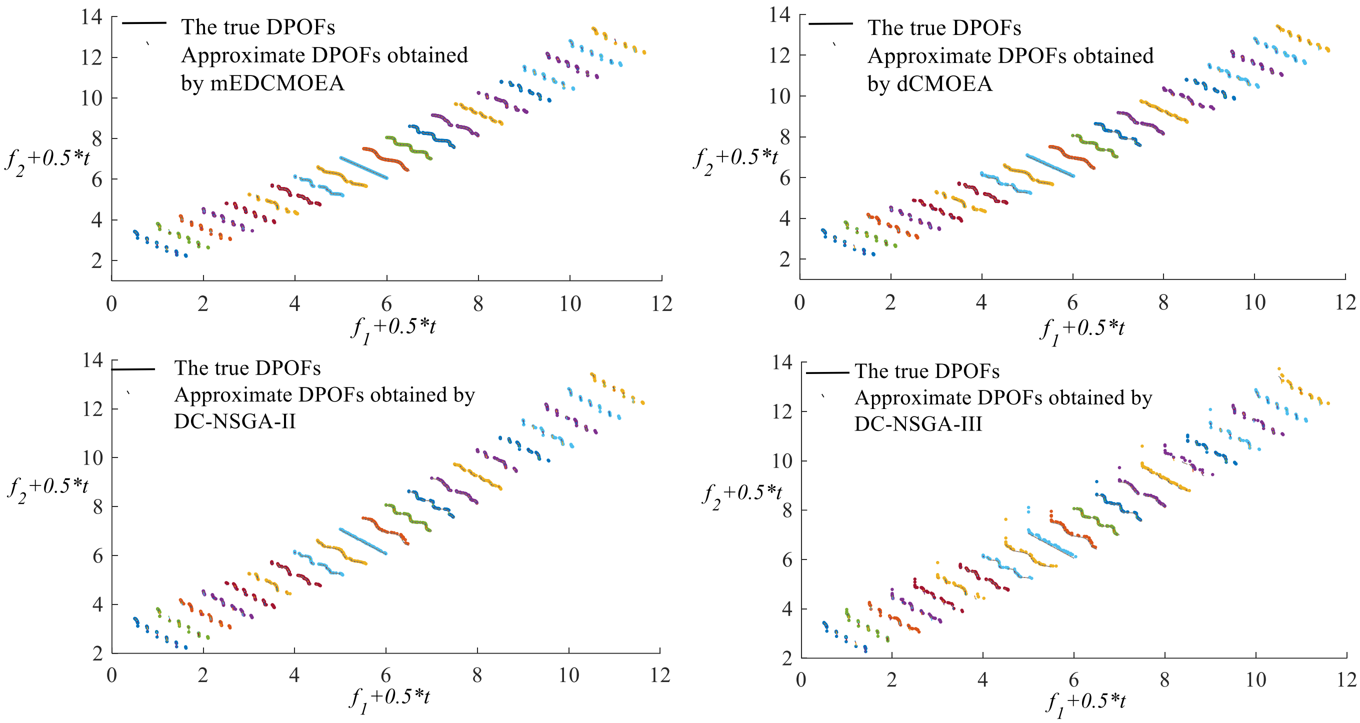


Fig. S-7. Obtained DPOFs obtained by the *m*EDCMOA (top left), *d*CMOEA (top right), DC-NSGA-II (bottom left), and DC-NSGA-III (bottom right) on the sixth test problem with $\tau_t$=18 and $n_t$=21.

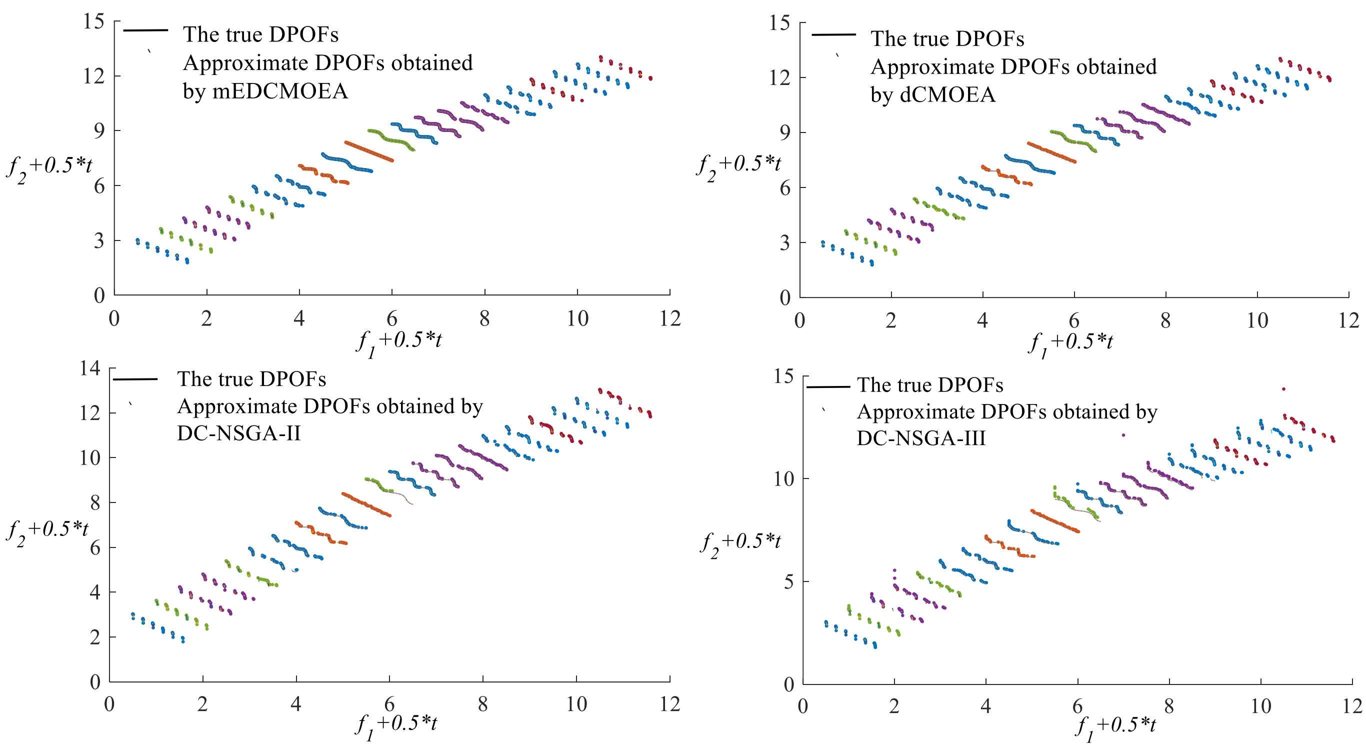


Fig. S-8. Obtained DPOFs obtained by the *m*EDCMOA (top left), *d*CMOEA (top right), DC-NSGA-II (bottom left), and DC-NSGA-III (bottom right) on the seventh test problem with $\tau_t$=18 and $n_t$=21.

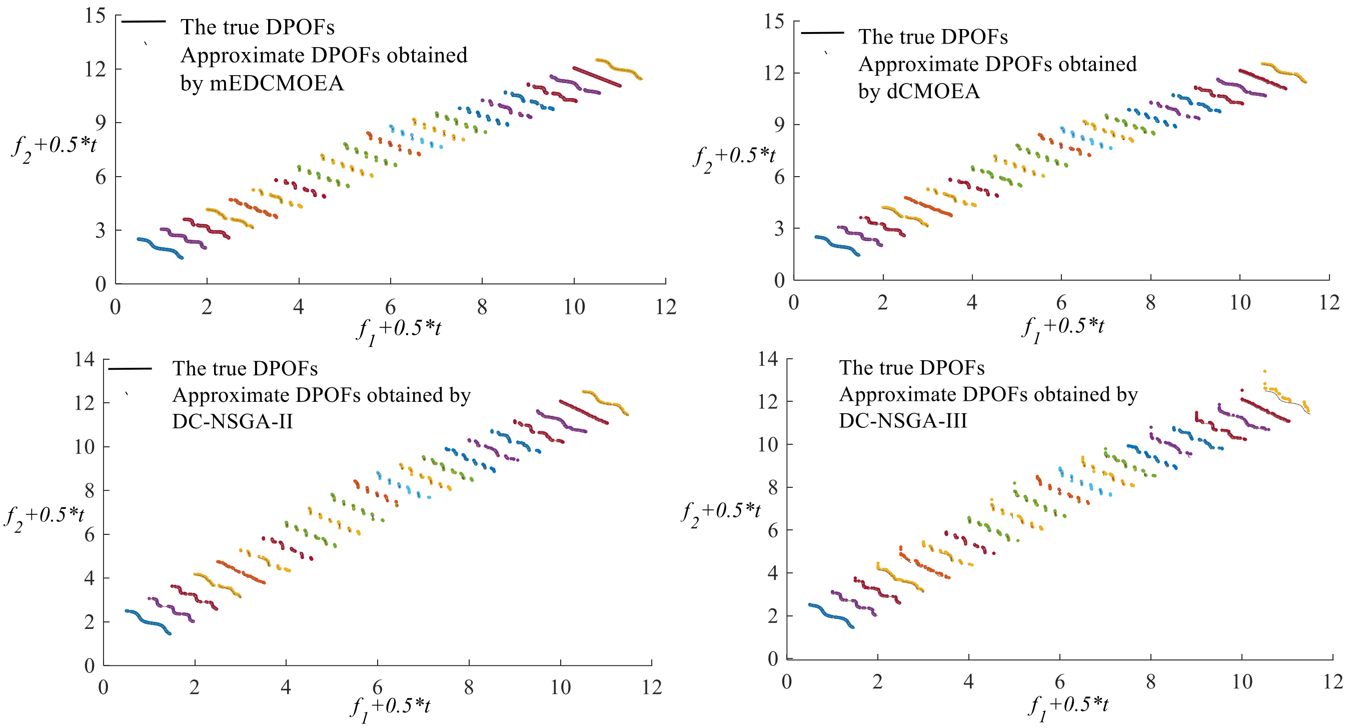


Fig. S-9. Obtained DPOFs obtained by the *m*EDCMOA (top left), *d*CMOEA (top right), DC-NSGA-II (bottom left), and DC-NSGA-III (bottom right) on the eighth test problem with $\tau_t$=18 and $n_t$=21.

TABLE S-IV
MEAN AND STANDARD DEVIATIONS OF MHV METRIC OBTAINED BY SEVEN ALGORITHMS

| Ins | ($\tau_t$, $n_t$) | *m*EDCMOA-V1 | *m*EDCMOA-V2 | *m*EDCMOA-V3 | *m*EDCMOA-V4 | *m*EDCMOA-V5 | *m*EDCMOA-V6 | *m*EDCMOA |
|---|---|---|---|---|---|---|---|---|
| 1 | (10, 21) | 3.51802(0.00227)‡ | 3.34464(0.00393)‡ | 3.38984(0.02595)‡ | 3.34604(0.00333)‡ | 3.21516(0.01814)‡ | 3.12235(0.08505)‡ | **3.51923(0.00224)** |
| | (14, 21) | 3.52519(0.00212)‡ | 3.51255(0.00278)‡ | 3.39073(0.01847)‡ | 3.50860(0.00413)‡ | 3.25920(0.03959)‡ | 3.29580(0.05254)‡ | **3.52701(0.00146)** |
| | (18, 21) | 3.52976(0.00175)‡ | 3.52893(0.00172)‡ | 3.39492(0.01679)‡ | 3.51514(0.00372)‡ | 3.48731(0.00602)‡ | 3.40020(0.03350)‡ | **3.53070(0.00143)** |
| 2 | (10, 21) | 3.41507(0.00183)† | 3.22775(0.00751)‡ | 3.37622(0.01042)‡ | 3.23010(0.00273)‡ | 3.13026(0.01617)‡ | 2.99118(0.08785)‡ | **3.41524(0.00163)** |
| | (14, 21) | 3.41911(0.00349)‡ | 3.41122(0.00182)‡ | 3.38163(0.00534)‡ | 3.39679(0.00615)‡ | 3.21258(0.02518)‡ | 3.19326(0.05275)‡ | **3.42030(0.00121)** |
| | (18, 21) | 3.42182(0.00140)‡ | 3.42116(0.00092)‡ | 3.38503(0.00852)‡ | 3.40362(0.00421)‡ | 3.39183(0.00235)‡ | 3.30277(0.02140)‡ | **3.42259(0.00183)** |
| 3 | (10, 21) | 3.51856(0.00217)‡ | 3.34502(0.00308)‡ | 3.48113(0.00521)‡ | 3.34614(0.00517)‡ | 3.21847(0.01982)‡ | 3.07655(0.08586)‡ | **3.51969(0.00244)** |
| | (14, 21) | 3.52522(0.00345)‡ | 3.51414(0.00268)‡ | 3.49105(0.00790)‡ | 3.50348(0.00378)‡ | 3.27159(0.03077)‡ | 3.28593(0.04849)‡ | **3.52703(0.00209)** |
| | (18, 21) | 3.53035(0.00254)‡ | 3.52999(0.00202)‡ | 3.49310(0.00697)‡ | 3.51415(0.00394)‡ | 3.49304(0.00395)‡ | 3.40918(0.03540)‡ | **3.53175(0.00179)** |
| 4 | (10, 21) | 3.40756(0.00151)† | 3.22766(0.00339)‡ | 3.28962(0.01489)‡ | 3.22928(0.00382)‡ | 3.12377(0.01774)‡ | 2.99638(0.07778)‡ | **3.40792(0.00120)** |
| | (14, 21) | 3.41137(0.00140)‡ | 3.40415(0.00119)‡ | 3.30424(0.01788)‡ | 3.39326(0.00366)‡ | 3.18194(0.03677)‡ | 3.17313(0.04774)‡ | **3.41214(0.00133)** |
| | (18, 21) | 3.41362(0.00147)‡ | 3.41356(0.00113)‡ | 3.29895(0.01618)‡ | 3.39641(0.00329)‡ | 3.38072(0.00520)‡ | 3.29058(0.03755)‡ | **3.41503(0.00094)** |
| 5 | (10, 21) | 3.53473(0.00189)‡ | 3.35607(0.00375)‡ | 3.41339(0.01544)‡ | 3.35778(0.00317)‡ | 3.22518(0.02018)‡ | 3.15692(0.06172)‡ | **3.53629(0.00206)** |
| | (14, 21) | 3.54203(0.00167)† | 3.53058(0.00288)‡ | 3.41417(0.01938)‡ | 3.52162(0.00376)‡ | 3.27107(0.02966)‡ | 3.33717(0.03935)‡ | **3.54228(0.00151)** |
| | (18, 21) | 3.54649(0.00127)‡ | 3.54513(0.00183)‡ | 3.41918(0.01659)‡ | 3.52881(0.00407)‡ | 3.50512(0.00602)‡ | 3.43015(0.01964)‡ | **3.54773(0.00180)** |
| 6 | (10, 21) | 3.44214(0.00190)‡ | 3.26013(0.00295)‡ | 3.40914(0.00597)‡ | 3.26149(0.00275)‡ | 3.16218(0.01449)‡ | 3.02541(0.06687)‡ | **3.44291(0.00181)** |
| | (14, 21) | **3.44861(0.00083)**† | 3.43871(0.00151)‡ | 3.41899(0.00946)‡ | 3.42781(0.00428)‡ | 3.24075(0.02558)‡ | 3.22567(0.05822)‡ | 3.44843(0.00129) |
| | (18, 21) | 3.45104(0.00090)† | 3.45021(0.00093)‡ | 3.42573(0.00737)‡ | 3.43232(0.00478)‡ | 3.41996(0.00354)‡ | 3.32179(0.03445)‡ | **3.45112(0.00139)** |
| 7 | (10, 21) | 3.53446(0.00213)‡ | 3.35934(0.00329)‡ | 3.50073(0.00930)‡ | 3.35851(0.00425)‡ | 3.23161(0.01782)‡ | 3.11419(0.06750)‡ | **3.53560(0.00187)** |
| | (14, 21) | 3.54153(0.00101)‡ | 3.52976(0.00193)‡ | 3.50605(0.01126)‡ | 3.51826(0.00286)‡ | 3.28706(0.02690)‡ | 3.30185(0.04128)‡ | **3.54243(0.00226)** |
| | (18, 21) | 3.54591(0.00158)‡ | 3.54474(0.00157)‡ | 3.51460(0.00741)‡ | 3.52383(0.00333)‡ | 3.51092(0.00461)‡ | 3.41620(0.03444)‡ | **3.54703(0.00161)** |
| 8 | (10, 21) | **3.43814(0.00180)**† | 3.25843(0.00314)‡ | 3.32588(0.02197)‡ | 3.25898(0.00295)‡ | 3.15356(0.01823)‡ | 3.01296(0.09941)‡ | **3.43758(0.00156)** |
| | (14, 21) | 3.44232(0.00125)‡ | 3.43251(0.00210)‡ | 3.34176(0.01530)‡ | 3.42496(0.00330)‡ | 3.21556(0.02378)‡ | 3.20184(0.05049)‡ | **3.44294(0.00124)** |
| | (18, 21) | **3.44624(0.00107)**† | 3.44417(0.00111)‡ | 3.34382(0.02101)‡ | 3.42948(0.00285)‡ | 3.41044(0.00622)‡ | 3.32252(0.03305)‡ | 3.44574(0.00103) |

‡ and † indicate *d*CMOEA performs significantly better than and equivalently to the corresponding algorithm, respectively.

## S-IX. AVERAGE VALUES AND STANDARD DEVIATIONS OF MHV AND MIGD METRICS OF THE *M*EDCMOA AND ITS VARIANTS

Tables S-IV and S-V summarize the average values and standard deviations of MHV and MIGD metrics over 30 runs for *m*EDCMOA and its variants on test problems. The results show that *m*EDCMOA outperforms its variants except for *m*EDCMOA-V1 on all test problems. *m*EDCMOA-V1 achieves the best MHV values on the sixth test problem with $\tau_t$=14 and the eighth test problem with $\tau_t$=10 and $\tau_t$=18. In terms of the MIGD metric, *m*EDCMOA-V1 outperforms *m*EDCMOA on the fifth test problem with $\tau_t$=18, and the sixth test problem with $\tau_t$=10. It is worth noting that there is no statistical difference in performance between *m*EDCMOA-V1 and *m*EDCMOA when *m*EDCMOA-V1 performs better than *m*EDCMOA. *m*EDCMOA-V1 adopts the advantages of nondominated infeasible solutions and shows comparable performance with

TABLE S-V
MEAN AND STANDARD DEVIATIONS OF MIGD METRIC OBTAINED BY SEVEN ALGORITHMS

| Ins | ($\tau_t$, $n_t$) | *m*EDCMOA-V1 | *m*EDCMOA-V2 | *m*EDCMOA-V3 | *m*EDCMOA-V4 | *m*EDCMOA-V5 | *m*EDCMOA-V6 | *m*EDCMOA |
|---|---|---|---|---|---|---|---|---|
| 1 | (10, 21) | 0.01275(0.00129)‡ | 0.02119(0.00155)‡ | 0.07128(0.01188)‡ | 0.02000(0.00174)‡ | 0.05659(0.00624)‡ | 0.14879(0.03010)‡ | **0.01169(0.00090)** |
| | (14, 21) | 0.01086(0.00108)‡ | 0.01355(0.00108)‡ | 0.07131(0.00934)‡ | 0.02130(0.00126)‡ | 0.09884(0.01571)‡ | 0.08797(0.01627)‡ | **0.00945(0.00088)** |
| | (18, 21) | 0.00972(0.00074)‡ | 0.00900(0.00111)‡ | 0.07255(0.00793)‡ | 0.01991(0.00157)‡ | 0.01991(0.00199)‡ | 0.05116(0.01205)‡ | **0.00840(0.00088)** |
| 2 | (10, 21) | 0.01209(0.00151)‡ | 0.01784(0.00146)‡ | 0.03127(0.00191)‡ | 0.01694(0.00127)‡ | 0.05302(0.00606)‡ | 0.17252(0.03404)‡ | **0.01135(0.00094)** |
| | (14, 21) | 0.01058(0.00135)† | 0.01274(0.00102)‡ | 0.03086(0.00201)‡ | 0.01960(0.00172)‡ | 0.08530(0.00920)‡ | 0.09515(0.01863)‡ | **0.01046(0.00104)** |
| | (18, 21) | 0.01007(0.00131)‡ | 0.01001(0.00129)‡ | 0.03052(0.00196)‡ | 0.01884(0.00157)‡ | 0.01821(0.00116)‡ | 0.05306(0.00710)‡ | **0.00936(0.00124)** |
| 3 | (10, 21) | 0.01297(0.00120)‡ | 0.02104(0.00177)‡ | 0.03609(0.00214)‡ | 0.02018(0.00190)‡ | 0.05659(0.00666)‡ | 0.16486(0.02846)‡ | **0.01236(0.00103)** |
| | (14, 21) | 0.01174(0.00139)‡ | 0.01434(0.00107)‡ | 0.03549(0.00205)‡ | 0.02324(0.00241)‡ | 0.09673(0.01038)‡ | 0.09244(0.01770)‡ | **0.01075(0.00128)** |
| | (18, 21) | 0.01054(0.00150)‡ | 0.01028(0.00095)‡ | 0.03602(0.00146)‡ | 0.02162(0.00133)‡ | 0.01856(0.00142)‡ | 0.04910(0.01063)‡ | **0.00949(0.00091)** |
| 4 | (10, 21) | 0.01149(0.00147)‡ | 0.01811(0.00135)‡ | 0.06114(0.00890)‡ | 0.01746(0.00159)‡ | 0.05372(0.00675)‡ | 0.16279(0.02770)‡ | **0.01033(0.00085)** |
| | (14, 21) | 0.01003(0.00087)‡ | 0.01200(0.00104)‡ | 0.05865(0.00743)‡ | 0.01777(0.00130)‡ | 0.09246(0.01227)‡ | 0.09707(0.01759)‡ | **0.00941(0.00093)** |
| | (18, 21) | 0.00943(0.00105)‡ | 0.00924(0.00136)‡ | 0.06110(0.00675)‡ | 0.01672(0.00123)‡ | 0.01953(0.00220)‡ | 0.05287(0.01254)‡ | **0.00861(0.00118)** |
| 5 | (10, 21) | 0.02643(0.00073)‡ | 0.03657(0.00153)‡ | 0.08070(0.00766)‡ | 0.03525(0.00159)‡ | 0.06047(0.00626)‡ | 0.14119(0.02086)‡ | **0.02616(0.00122)** |
| | (14, 21) | 0.02481(0.00098)‡ | 0.02759(0.00096)‡ | 0.08184(0.01094)‡ | 0.03635(0.00180)‡ | 0.10016(0.00964)‡ | 0.08089(0.01366)‡ | **0.02407(0.00090)** |
| | (18, 21) | **0.02337(0.00082)**† | 0.02398(0.00084)‡ | 0.08240(0.00950)‡ | 0.03609(0.00177)‡ | 0.03029(0.00192)‡ | 0.05156(0.00648)‡ | 0.02339(0.00083) |
| 6 | (10, 21) | **0.02599(0.00117)**† | 0.03332(0.00129)‡ | 0.04304(0.00168)‡ | 0.03232(0.00176)‡ | 0.05343(0.00540)‡ | 0.16554(0.02529)‡ | 0.02611(0.00114) |
| | (14, 21) | 0.02514(0.00095)‡ | 0.02669(0.00096)‡ | 0.04263(0.00180)‡ | 0.03516(0.00157)‡ | 0.08566(0.00952)‡ | 0.09245(0.02072)‡ | **0.02459(0.00106)** |
| | (18, 21) | 0.02450(0.00143)‡ | 0.02456(0.00126)‡ | 0.04215(0.00180)‡ | 0.03549(0.00139)‡ | 0.02733(0.00116)‡ | 0.05943(0.01260)‡ | **0.02387(0.00078)** |
| 7 | (10, 21) | 0.02695(0.00113)‡ | 0.03709(0.00197)‡ | 0.04722(0.00261)‡ | 0.03558(0.00157)‡ | 0.06047(0.00566)‡ | 0.15656(0.02456)‡ | **0.02637(0.00090)** |
| | (14, 21) | 0.02541(0.00098)‡ | 0.02807(0.00103)‡ | 0.04616(0.00374)‡ | 0.03815(0.00192)‡ | 0.09762(0.00869)‡ | 0.09330(0.01387)‡ | **0.02490(0.00091)** |
| | (18, 21) | 0.02463(0.00075)‡ | 0.02439(0.00089)† | 0.04574(0.00179)‡ | 0.03959(0.00198)‡ | 0.02897(0.00114)‡ | 0.05671(0.01137)‡ | **0.02402(0.00091)** |
| 8 | (10, 21) | 0.02528(0.00093)‡ | 0.03323(0.00154)‡ | 0.07391(0.00995)‡ | 0.03248(0.00144)‡ | 0.05376(0.00592)‡ | 0.16377(0.03633)‡ | **0.02471(0.00112)** |
| | (14, 21) | 0.02420(0.00113)‡ | 0.02605(0.00095)‡ | 0.06871(0.00627)‡ | 0.03345(0.00173)‡ | 0.09005(0.00921)‡ | 0.09797(0.01791)‡ | **0.02363(0.00070)** |
| | (18, 21) | 0.02332(0.00105)† | 0.02314(0.00074)† | 0.07142(0.00917)‡ | 0.03359(0.00147)‡ | 0.02860(0.00181)‡ | 0.05533(0.01118)‡ | **0.02310(0.00085)** |

*m*EDCMOA. In test problems, there are many nondominated infeasible solutions when the unconstrained DPOFs include infeasible regions, and exploring the potentiality of nondominated infeasible solutions is conducive to pushing the population to the DPOFs. Additionally, it is worth noting that *m*EDCMOA-V1 takes approximately one-third more time than *m*EDCMOA for each test problem.

The proposed *m*EDCMOA and *m*EDCMOA-V2 have no statistical difference on the MIGD metric on the seventh and eighth test problems with $\tau_t$=18. This may be because *m*EDCMOA-V2 requires a long time to converge, and many feasible and infeasible solutions far from the DPOFs may be used to generate offspring, obtaining children with poor performance and slowing down population convergence. The population selection strategies in *m*EDCMOA-V3 and *m*EDCMOA-V4 may choose some feasible solutions with large objective values, leading to more solutions far away from the DPOFs reserved in the population.

In *m*EDCMOA-V5, the memory strategy stores some previously found solutions, but they are not updated according to the current problem characteristics, which may increase the number of infeasible solutions in the new environment. Therefore, *m*EDCMOA-V5 cannot track the new DPOF. In contrast, *m*EDCMOA-V6 updates the reserved solutions according to the feasibility ratio when a new environment arises. However, simply updating solutions far from the DPOFs may still not be enough to approach the true DPOFs.

## S-X. EXPERIMENTAL RESULTS ON *M*EDCMOA WITH OTHER NUMBERS OF THE OPTIMAL DISTANCE VARIABLES

The values and standard deviations of the two chosen performance metrics (i.e., MHV and MIGD) for these algorithms on the test problems with *ju*=6 and 8 are shown in Tables S-VI and S-VII, respectively.

It can be seen from Tables S-VI and S-VII that *m*EDCMOA achieves the best performance on the test problems among the compared algorithms, thus undoubtedly demonstrating that *m*EDCMOA can successfully solve DCMOPs with different numbers of the optimal distance variables that have drastic changes under a dynamic changing environment.

TABLE S-VI
STATISTIC RESULTS OF THE FOUR COMPARED ALGORITHMS ON TEST PROBLEMS WITH *JU*=6

| Ins | Indicator | *d*CMOEA | DC-NSGA-II | DC-NSGA-III | *m*EDCMOA |
|---|---|---|---|---|---|
| 1 | *MHV* | 3.46307(0.00944)‡ | 3.46173(0.01103)‡ | 3.38766(0.01308)‡ | **3.53126(0.00155)** |
| | *MIGD* | 0.02881(0.00339)‡ | 0.02483(0.00179)‡ | 0.04378(0.00311)‡ | **0.00856(0.00100)** |
| 2 | *MHV* | 3.37433(0.00690)‡ | 3.37124(0.00699)‡ | 3.31102(0.01292)‡ | **3.42255(0.00125)** |
| | *MIGD* | 0.02538(0.00249)‡ | 0.02263(0.00130)‡ | 0.03849(0.00320)‡ | **0.00944(0.00134)** |
| 3 | *MHV* | 3.47383(0.00616)‡ | 3.46181(0.01028)‡ | 3.40156(0.01262)‡ | **3.53135(0.00244)** |
| | *MIGD* | 0.02553(0.00240)‡ | 0.02519(0.00153)‡ | 0.04142(0.00312)‡ | **0.00962(0.00087)** |
| 4 | *MHV* | 3.35748(0.00980)‡ | 3.35842(0.01097)‡ | 3.29284(0.01476)‡ | **3.41462(0.00102)** |
| | *MIGD* | 0.02869(0.00368)‡ | 0.02312(0.00181)‡ | 0.04038(0.00360)‡ | **0.00840(0.00106)** |
| 5 | *MHV* | 3.47956(0.01013)‡ | 3.47485(0.01662)‡ | 3.40610(0.01940)‡ | **3.54777(0.00143)** |
| | *MIGD* | 0.03706(0.00281)‡ | 0.03651(0.00469)‡ | 0.04905(0.00429)‡ | **0.02365(0.00092)** |
| 6 | *MHV* | 3.40116(0.00598)‡ | 3.39564(0.01140)‡ | 3.33124(0.01255)‡ | **3.45058(0.00122)** |
| | *MIGD* | 0.03280(0.00201)‡ | 0.03146(0.00182)‡ | 0.04393(0.00267)‡ | **0.02373(0.00097)** |
| 7 | *MHV* | 3.49118(0.00883)‡ | 3.47109(0.01776)‡ | 3.41158(0.01842)‡ | **3.54594(0.00175)** |
| | *MIGD* | 0.03419(0.00205)‡ | 0.03672(0.00411)‡ | 0.04799(0.00335)‡ | **0.02379(0.00077)** |
| 8 | *MHV* | 3.38889(0.00746)‡ | 3.38866(0.01276)‡ | 3.32649(0.02002)‡ | **3.44593(0.00092)** |
| | *MIGD* | 0.03462(0.00268)‡ | 0.03195(0.00221)‡ | 0.04428(0.00408)‡ | **0.02310(0.00097)** |

TABLE S-VII
STATISTIC RESULTS OF THE FOUR COMPARED ALGORITHMS ON TEST PROBLEMS WITH *JU*=8

| Ins | Indicator | *d*CMOEA | DC-NSGA-II | DC-NSGA-III | *m*EDCMOA |
|---|---|---|---|---|---|
| 1 | *MHV* | 3.46674(0.01244)‡ | 3.45761(0.01833)‡ | 3.39409(0.01831)‡ | **3.53120(0.00174)** |
| | *MIGD* | 0.02742(0.00424)‡ | 0.02534(0.00376)‡ | 0.04302(0.00372)‡ | **0.00813(0.00073)** |
| 2 | *MHV* | 3.37935(0.00691)‡ | 3.36887(0.01031)‡ | 3.30820(0.01529)‡ | **3.42220(0.00124)** |
| | *MIGD* | 0.02389(0.00283)‡ | 0.02262(0.00189)‡ | 0.03846(0.00328)‡ | **0.00959(0.00095)** |
| 3 | *MHV* | 3.47287(0.01840)‡ | 3.46154(0.01634)‡ | 3.39593(0.01599)‡ | **3.53109(0.00173)** |
| | *MIGD* | 0.02571(0.00578)‡ | 0.02555(0.00335)‡ | 0.04209(0.00353)‡ | **0.00966(0.00096)** |
| 4 | *MHV* | 3.36397(0.00810)‡ | 3.36078(0.00997)‡ | 3.29763(0.01953)‡ | **3.41471(0.00096)** |
| | *MIGD* | 0.02584(0.00258)‡ | 0.02234(0.00152)‡ | 0.03928(0.00373)‡ | **0.00863(0.00084)** |
| 5 | *MHV* | 3.49073(0.01299)‡ | 3.46928(0.01897)‡ | 3.40280(0.02050)‡ | **3.54835(0.00145)** |
| | *MIGD* | 0.03462(0.00493)‡ | 0.03752(0.00443)‡ | 0.04914(0.00476)‡ | **0.02369(0.00094)** |
| 6 | *MHV* | 3.40401(0.00832)‡ | 3.39232(0.01330)‡ | 3.33343(0.01391)‡ | **3.45020(0.00122)** |
| | *MIGD* | 0.03162(0.00203)‡ | 0.03171(0.00202)‡ | 0.04248(0.00275)‡ | **0.02402(0.00102)** |
| 7 | *MHV* | 3.48953(0.01910)‡ | 3.46989(0.01824)‡ | 3.39672(0.02664)‡ | **3.54610(0.00198)** |
| | *MIGD* | 0.03453(0.00872)‡ | 0.03665(0.00365)‡ | 0.05143(0.00699)‡ | **0.02418(0.00077)** |
| 8 | *MHV* | 3.39237(0.00939)‡ | 3.38656(0.01110)‡ | 3.32903(0.01557)‡ | **3.44632(0.00103)** |
| | *MIGD* | 0.03343(0.00265)‡ | 0.03188(0.00194)‡ | 0.04283(0.00251)‡ | **0.02340(0.00103)** |

TABLE S-VIII
FEASIBLE TIME RATIOS OBTAINED BY SEVEN ALGORITHMS

| Ins | ($\tau_t$, $n_t$) | *d*CMOEA | DC-MOEA | DC-NSGA-II-A | DC-NSGA-II | DC-NSGA-III | DC-TAEA | *m*EDCMOA |
|---|---|---|---|---|---|---|---|---|
| 1 | (10, 21) | 0.9720 | 0.8880 | 0.9440 | 1.0000 | 1.0000 | 0.9760 | 1.0000 |
| | (14, 21) | 0.9790 | 0.9341 | 0.9641 | 0.9970 | 1.0000 | 0.9671 | 1.0000 |
| | (18, 21) | 0.9833 | 0.9330 | 0.9761 | 1.0000 | 1.0000 | 0.9785 | 0.9976 |
| 2 | (10, 21) | 0.9760 | 0.9560 | 0.9800 | 1.0000 | 1.0000 | 0.9840 | 1.0000 |
| | (14, 21) | 0.9820 | 0.9611 | 0.9910 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (18, 21) | 0.9857 | 0.9857 | 0.9904 | 1.0000 | 1.0000 | 0.9976 | 1.0000 |
| 3 | (10, 21) | 0.9720 | 0.9200 | 0.9640 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (14, 21) | 0.9790 | 0.9072 | 0.9701 | 1.0000 | 1.0000 | 0.9940 | 1.0000 |
| | (18, 21) | 0.9833 | 0.9474 | 0.9880 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 4 | (10, 21) | 0.9760 | 0.9480 | 0.9880 | 1.0000 | 1.0000 | 0.9920 | 1.0000 |
| | (14, 21) | 0.9790 | 0.9761 | 0.9940 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (18, 21) | 0.9857 | 0.9737 | 0.9880 | 1.0000 | 1.0000 | 0.9880 | 1.0000 |
| 5 | (10, 21) | 0.9720 | 0.9000 | 0.9480 | 0.9920 | 0.9960 | 0.9600 | 1.0000 |
| | (14, 21) | 0.9761 | 0.9461 | 0.9641 | 1.0000 | 0.9970 | 0.9671 | 1.0000 |
| | (18, 21) | 0.9833 | 0.9187 | 0.9785 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 6 | (10, 21) | 0.9800 | 0.9520 | 0.9840 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (14, 21) | 0.9790 | 0.9611 | 0.9820 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (18, 21) | 0.9833 | 0.9809 | 0.9904 | 1.0000 | 1.0000 | 0.9928 | 1.0000 |
| 7 | (10, 21) | 0.9720 | 0.9120 | 0.9640 | 1.0000 | 1.0000 | 0.9920 | 1.0000 |
| | (14, 21) | 0.9790 | 0.9431 | 0.9671 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (18, 21) | 0.9833 | 0.9426 | 0.9785 | 1.0000 | 1.0000 | 0.9880 | 1.0000 |
| 8 | (10, 21) | 0.9720 | 0.9680 | 0.9800 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (14, 21) | 0.9820 | 0.9641 | 0.9970 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| | (18, 21) | 0.9833 | 0.9833 | 1.0000 | 1.0000 | 1.0000 | 0.9904 | 1.0000 |

## S-XI. EXPERIMENTAL RESULTS ON OTHER PERFORMANCE MEASURES

Two new measures are introduced to evaluate the capability of an algorithm to deal with the dynamic changes of DCMOPs, i.e., the feasible time ratio and offline error.

The feasible time ratio (denoted as $\gamma_f$) is designed to reflect the ability of an algorithm to locate feasible regions for problems that are difficult to obtain feasible solutions. It is formulated as follows:

$$r_f = \frac{\sum_{t=1}^{n_t}\sum_{g=1}^{\tau_t} fs(g,t)}{\tau_t \times n_t} \tag{S-15}$$

where $fs(g,t)=1$ when at least a feasible solution is obtained at generation $g$ at time $t$. $r_f$ denotes the feasible time ratio.

The offline error for dynamic constrained single optimization problems (DCSOPs) is calculated as the average over, at every evaluation, the error of the best solution of feasible solutions found since the last change of the environment according to [s2], [s3], which is as follows.

$$EV_{MO} = \frac{1}{ng}\sum_{j=1}^{ng} e_{MO}(j) \tag{S-16}$$

where $e_{MO}(j)$ is the offline error of the best solution at generation $j$, and $ng$ is the number of generations. $EV_{MO}$ is the offline error.

Unlike single-objective optimization problems, a set of solutions can satisfy the optima of a given DCMOP, and they are called Pareto-optimal solutions. Therefore, there are an infinite number of optimal solutions in DCMOPs, and Equation (S-16) cannot be directly used for evaluating the Pareto-optimal solutions of DCMOPs. The hypervolume (HV) and inverted generational distance (IGD) metrics are usually used to evaluate the performance of Pareto-optimal solutions in a stationary environment. However, the global optimum value of the HV metric is unknown. Therefore, to use the offline error to evaluate the capability of an algorithm to track true DPOFs in dynamic multiobjective optimization problems, this paper modifies Equation (S-16) based on the IGD metric.

This measure is modified as follows:

$$ED_{MO}^t = \frac{1}{ng_t}\sum_{j=1}^{ng_t}\left(IGD^t(j) - IGD_{global}^t(j)\right) \tag{S-17}$$

where $IGD^t(j)$ is the best IGD of Pareto-optimal solutions obtained by an algorithm at generation $j$ of the $t$th environment. $IGD_{global}^t$ is the global optimum value of the IGD at generation $j$ of the $t$th environment. $ng$ is the number of generations at environment $t$.

Because the global optimum value of the IGD metric (i.e., $IGD_{global}^t$) in each generation is zero. Therefore, Equation (S-17) can be written as follows:

$$ED_{MO}^t = \frac{1}{ng_t}\sum_{j=1}^{ng_t} IGD^t(j) \tag{S-18}$$

Additionally, a test problem includes $n_t$ environments. Therefore, we adopt the average value of $n_t$ offline errors to evaluate the performance of an algorithm, which is as follows:

$$MED_{MO}^t = \frac{1}{n_t}\sum_{t=1}^{n_t} ED_{MO}^t \tag{S-19}$$

where $MED_{MO}^t$ denotes the modified offline error.

The IGD values in the first environment are not considered so that the effect of static optimization can be minimized. The feasible time ratios obtained by seven algorithms are given in Table S-VIII, and the values and standard deviations of the modified offline error based on the IGD are shown in Tables S-IX.

Table S-VIII shows that the values of $\gamma_f$ of *m*EDCMOA are

TABLE S-IX
MEAN AND STANDARD DEVIATIONS OF THE MODIFIED OFFLINE ERROR BASED ON MIGD METRIC OBTAINED BY SEVEN ALGORITHMS

| Ins | ($\tau_t$, $n_t$) | *d*CMOEA | DC-MOEA | DC-NSGA-II-A | DC-NSGA-II | DC-NSGA-III | DC-TAEA | *m*EDCMOA |
|---|---|---|---|---|---|---|---|---|
| 1 | (10, 21) | 0.28125(0.02088) | 0.65980(0.04193) | 0.39169(0.03274) | 0.21890(0.01622) | 0.25561(0.01346) | 0.33372(0.04134) | **0.02013(0.00090)** |
| | (14, 21) | 0.21949(0.01556) | 0.46765(0.03747) | 0.29755(0.02222) | 0.16487(0.01037) | 0.20470(0.01193) | 0.26827(0.03759) | **0.01742(0.00091)** |
| | (18, 21) | 0.18164(0.01450) | 0.36761(0.02698) | 0.24467(0.01328) | 0.13942(0.01009) | 0.16611(0.00972) | 0.22757(0.03493) | **0.01555(0.00090)** |
| 2 | (10, 21) | 0.28527(0.02153) | 0.57287(0.05350) | 0.40605(0.03364) | 0.20474(0.01581) | 0.24451(0.01591) | 0.30642(0.03378) | **0.01722(0.00099)** |
| | (14, 21) | 0.21739(0.01124) | 0.43245(0.03655) | 0.31147(0.02525) | 0.15728(0.01399) | 0.19774(0.01204) | 0.24782(0.02805) | **0.01556(0.00098)** |
| | (18, 21) | 0.17167(0.01348) | 0.35392(0.03166) | 0.24074(0.01793) | 0.12788(0.01125) | 0.15799(0.01198) | 0.20291(0.02392) | **0.01378(0.00101)** |
| 3 | (10, 21) | 0.29329(0.02113) | 0.49937(0.06103) | 0.42340(0.03831) | 0.22159(0.01903) | 0.25751(0.01475) | 0.35484(0.03169) | **0.02041(0.00108)** |
| | (14, 21) | 0.23168(0.01958) | 0.40310(0.04210) | 0.31935(0.02061) | 0.17279(0.01047) | 0.20852(0.01334) | 0.28380(0.03079) | **0.01811(0.00110)** |
| | (18, 21) | 0.18698(0.01207) | 0.33054(0.02658) | 0.25109(0.01711) | 0.13920(0.00749) | 0.16783(0.01034) | 0.23503(0.02720) | **0.01619(0.00097)** |
| 4 | (10, 21) | 0.28751(0.02350) | 0.49237(0.05122) | 0.42278(0.02605) | 0.20258(0.01636) | 0.25014(0.01913) | 0.31472(0.02757) | **0.01696(0.00085)** |
| | (14, 21) | 0.22299(0.01712) | 0.38523(0.03331) | 0.31206(0.02756) | 0.15956(0.01308) | 0.19157(0.01357) | 0.25992(0.02928) | **0.01536(0.00075)** |
| | (18, 21) | 0.18119(0.01133) | 0.32463(0.03017) | 0.25639(0.02695) | 0.12632(0.00795) | 0.15548(0.00914) | 0.20184(0.02178) | **0.01389(0.00108)** |
| 5 | (10, 21) | 0.28648(0.03001) | 0.66287(0.05430) | 0.39604(0.03312) | 0.22274(0.01669) | 0.26443(0.01801) | 0.34280(0.04166) | **0.03448(0.00082)** |
| | (14, 21) | 0.21875(0.01575) | 0.47470(0.03948) | 0.30868(0.01651) | 0.17669(0.01349) | 0.20586(0.01027) | 0.27686(0.03902) | **0.03181(0.00095)** |
| | (18, 21) | 0.18252(0.01238) | 0.37611(0.03550) | 0.24794(0.01824) | 0.14242(0.00932) | 0.17018(0.01172) | 0.23400(0.03067) | **0.03070(0.00064)** |
| 6 | (10, 21) | 0.28327(0.01799) | 0.58687(0.05457) | 0.40650(0.02920) | 0.21083(0.01596) | 0.24156(0.01370) | 0.31754(0.04578) | **0.03266(0.00114)** |
| | (14, 21) | 0.21350(0.01358) | 0.44415(0.03755) | 0.31395(0.02880) | 0.15829(0.01208) | 0.19092(0.01365) | 0.25460(0.02849) | **0.03024(0.00108)** |
| | (18, 21) | 0.17438(0.01421) | 0.35152(0.02651) | 0.25699(0.01519) | 0.13260(0.00809) | 0.15930(0.00913) | 0.21093(0.02802) | **0.02895(0.00075)** |
| 7 | (10, 21) | 0.30740(0.02327) | 0.51585(0.05207) | 0.41207(0.02563) | 0.22942(0.01874) | 0.26174(0.02078) | 0.35781(0.03853) | **0.03443(0.00100)** |
| | (14, 21) | 0.23706(0.01551) | 0.40148(0.03690) | 0.32190(0.02418) | 0.17725(0.01419) | 0.21182(0.01299) | 0.27899(0.03249) | **0.03236(0.00086)** |
| | (18, 21) | 0.18733(0.01222) | 0.34324(0.03590) | 0.25489(0.01612) | 0.14398(0.01102) | 0.17658(0.01253) | 0.23550(0.03050) | **0.03071(0.00071)** |
| 8 | (10, 21) | 0.28791(0.02038) | 0.52398(0.05599) | 0.41884(0.03847) | 0.20657(0.01723) | 0.24416(0.01590) | 0.32358(0.03912) | **0.03200(0.00124)** |
| | (14, 21) | 0.22347(0.01851) | 0.41092(0.03414) | 0.30427(0.02222) | 0.16021(0.01088) | 0.19325(0.01557) | 0.25623(0.03547) | **0.02998(0.00082)** |
| | (18, 21) | 0.17508(0.01344) | 0.34427(0.03600) | 0.25690(0.02156) | 0.13147(0.00840) | 0.15437(0.00997) | 0.20995(0.02404) | **0.02871(0.00093)** |

1.0 except the first test problem with $\tau_t$=18 and $n_t$=21. The values of $\gamma_f$ of DC-NSGA-II and DC-NSGA-III are 1 on almost test problems. DCMOEA performs poorly in tracking feasible regions, followed by *d*CMOEA and DC-NSGA-II-A.

From these comparison results of the modified offline errors shown in Table S-IX, it is clear that *m*EDCMOA performs well in eight test instances, demonstrating that *m*EDCMOA can deal with the dynamic changes of DCMOPs.

## REFERENCES

[S1] D. C. Montgomery, Design and Analysis of Experiments. Hoboken, NJ, USA: Wiley, 2005.

[S2] T. T. Nguyen and X. Yao, "Continuous dynamic constrained optimization—the challenges," *IEEE Trans. Evol. Comput*., vol. 16, no. 6, pp. 769–786, Deb. 2012.

[S3] C. Bu, W. Luo, and L. Yue. "Continuous dynamic constrained optimization with ensemble of locating and tracking feasible regions strategies," *IEEE Trans. Evol. Comput*., vol. 21, no. 1, pp. 14–33, Feb. 2017.